СОДЕРЖАНИЕ

Данный выпуск посвящается столетию академика
Ивана Георгиевича ПЕТРОВСКОГО,
выдающегося русского ученого и ректора
Московского государственного университета

УДК 517.956.2

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ РАВНОВЕСНЫХ КОНФИГУРАЦИЙ ЗВЕЗД НЕБОЛЬШОЙ МАССЫ

© 2007 г. И. С. Барская, С. И. Мухин, В. М. Чечеткин

irina_smet@mail.ru

Кафедра Вычислительных методов

Введение. Одной из актуальных, но недостаточно изученных и требующих дальнейшего исследования проблем математического моделирования эволюции звезд является построение равновесных конфигураций звезд небольшой массы [1-6]. Такие конфигурации важны для моделирования газодинамических процессов на этапах образования нейтронных звезд или черных дыр, а также для моделирования коллапса и вспышек сверхновых.

В работе [19] нами было показано, как строится модель звезды и находятся ее равновесные конфигурации. Объект рассматривается как самогравитирующее газовое облако и описывается с помощью физических параметров, распределение которых находится из системы уравнений газовой динамики.

Основной трудностью при нахождении равновесной конфигурации звезды является нахождение гравитационного потенциала. Гравитационное поле вне тела со сферически симметричным распределением вещества зависит только от полной массы притягивающего тела. Распределение гравитационного потенциала внутри звезды описывается уравнением Пуассона [7]. Проблемой при решении этого уравнения является выбор граничных условий [2, 8, 11]. В работе [19] для численного моделирования самогравитирующей вращающейся звезды и получения ее равновесных конфигураций исследовались три различных типа граничных условий для уравнения Пуассона и проводилось сравнение полученных решений. Было показано, что наиболее предпочтительным является использование граничных условий третьего типа – интегральных граничных условий.

Опираясь на результаты, полученные для модельной звезды, мы покажем в данной работе, как можно найти равновесную конфигурацию реального астрофизического объекта – звезды с массой порядка $1.8\mathrm{M}_\Theta$.

1. Математическая модель. Как было показано в [19], равновесные конфигурации звезды можно получить за счет равновесия силы собственного тяготения и градиента газового давления, решив систему трехмерных уравнений газовой динамики [8]:

$$\frac{\partial(r\rho)}{\partial t}+\frac{\partial(r\rho u)}{\partial r}+\frac{1}{r}\frac{\partial(r\rho v)}{\partial\varphi}+\frac{\partial(r\rho w)}{\partial z}=0,$$

$$\frac{\partial(r\rho u)}{\partial t}+\frac{\partial(r\rho u^2+rp)}{\partial r}+\frac{1}{r}\frac{\partial(r\rho uv)}{\partial\varphi}+\frac{\partial(r\rho uw)}{\partial z}=p+\rho v^2+r\rho\mathbf{F}_r,$$

$$\frac{\partial(r\rho v)}{\partial t}+\frac{\partial(r\rho vu)}{\partial r}+\frac{1}{r}\frac{\partial(r\rho v^2+rp)}{\partial\varphi}+\frac{\partial(r\rho vw)}{\partial z}=-\rho uv+r\rho\mathbf{F}_\varphi, \qquad (1)$$

$$\frac{\partial(r\rho w)}{\partial t}+\frac{\partial(r\rho wu)}{\partial r}+\frac{1}{r}\frac{\partial(r\rho wv)}{\partial\varphi}+\frac{\partial(r\rho w^2+rp)}{\partial z}=r\rho\mathbf{F}_z,$$

$$\frac{\partial(r\rho e)}{\partial t}+\frac{\partial(r\rho uh)}{\partial r}+\frac{1}{r}\frac{\partial(r\rho vh)}{\partial\varphi}+\frac{\partial(r\rho wh)}{\partial z}=r\rho(\mathbf{F},\mathbf{v}),$$

$$e=\varepsilon+u^2/2+v^2/2+w^2/2,\quad h=e+p/\rho.$$

Уравнения (1) записаны в цилиндрической системе координаты (r, φ, z). Для замыкания системы (1) используем уравнение состояния идеального газа:

$$p = (\gamma - 1)\rho\,\varepsilon. \tag{2}$$

Здесь r - радиус, φ - полярный угол, t- время, ρ - плотность газа, p - давление, ε - удельная внутренняя энергия, e - полная удельная энергия, γ - показатель адиабаты, h - полная удельная энтальпия, $\mathbf{v} = (u, v, w)^T$ - скорость газа, u - ее радиальная компонента, v - азимутальная, w - компонента скорости по оси z, $\mathbf{F}$ - суммарная удельная сила с компонентами $(\mathbf{F}_r, \mathbf{F}_\varphi, \mathbf{F}_z)$, действующая на частицу газа. Сила $\mathbf{F}$ в данной задаче – это сила гравитации, которая выражается через гравитационный потенциал Ф:

$$\mathbf{F} \;=\; -grad(\,\Phi) \quad ,$$

а гравитационный потенциал удовлетворяет уравнению Пуассона: 1

$$\Delta\,\Phi \;= 4\pi\,G\rho\,, \tag{3}$$

где $\boldsymbol{G}$ – гравитационная постоянная [7].

Будем решать задачу в безразмерном виде. В качестве масштабных величин выберем $\{\,\rho_0, l_0\,, t_0\,\}$, где l_0 - характерный пространственный размер задачи, ρ_0 - плотность в центре звезды, t_0 - характерный временной масштаб. В безразмерных переменных, выбранных согласованным образом, система уравнений (1), уравнение (2) и (3) сохраняют прежний вид.

Систему уравнений (1), (2), (3) в силу симметрии относительно плоскости $z = 0$ будем решать в области Ω (Рис.1):

$$\Omega = (0 < r \le r_2) \times (0 \le \varphi \le 2\pi) \times (0 \le z \le z_2). \tag{4}$$

На “внешней” границе этой цилиндрической области $(\,r = r_2, z = z_2)$ в качестве граничных условий поставим “свободные” условия:

$$\left.\frac{\partial f}{\partial r}\right|_{r=r_2} = 0, \qquad \left.\frac{\partial f}{\partial z}\right|_{z=z_2} = 0 \quad , \tag{5}$$

где f – это одна из функций ρ, p, u, v, w, h. На “нижней” $(z = 0)$ и “внутренней” $(r = 0)$ границе - условия симметрии.

Поскольку мы ищем стационарные конфигурации, будем полагать, что сила гравитации обладает осевой симметрией, что позволит ограничиться рассмотрением двумерного уравнения Пуассона для гравитационного потенциала в двумерной области $G = (\,0 < r \le r_2\,,\, 0 \le z \le z_2)$ (Рис.1). При этом расчет газодинамических параметров будем проводить в трехмерном пространстве.

В области G задача для гравитационного потенциала имеет вид:

$$\begin{cases} \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial \Phi}{\partial r}\right) + \frac{\partial^2 \Phi}{\partial z^2} = 4\pi\,G\rho(r,z,t)\,, \quad (r,z) \in G \\ \frac{\partial \Phi}{\partial r}\big|_{\Gamma_1} = 0\,, \\ \frac{\partial \Phi}{\partial z}\big|_{\Gamma_4} = 0\,, \\ \Phi|_{\Gamma_2,\Gamma_3} = \mu(r,z,t) \end{cases} . \tag{6}$$

В качестве краевых условий для этой задачи на границе Γ_2 и Γ_3 возьмем функцию $\mu(r, \varphi, z)$, представляющую собой точное значение гравитационного потенциала:

$$\mu(r,\varphi,z) = -G \int\limits_{\Omega} \frac{\rho(\bar{x}')}{|\bar{x} - \bar{x}'|}\, d\Omega \quad . \tag{7}$$

Система (1), (2), (6) решается совместно, так как для нахождения силы гравитации, компоненты которой входят в правую часть системы уравнений газовой динамики (1), используется функция плотности.

В качестве начальных данных для системы уравнений (1), (2), (6) возьмем гидростатически равновесную стационарную конфигурацию, которую можно найти, задав центральную плотность, из системы уравнений [7]:

$$\begin{cases} \frac{dp}{dr} = -\frac{GM(r)\rho}{r^2} \\ \frac{dM}{dr} = 4\pi r^2 \rho \\ p = p(\rho, \varepsilon) \end{cases} . \tag{8}$$

2. Вычислительный алгоритм. Систему уравнений (1), (2), (6) будем решать численно, разбиением по процессам: газодинамическому и гравитационному. В начальный момент времени найдем распределение всех газодинамических параметров, решив систему уравнений (8). Затем, зная распределение плотности, найдем распределение гравитационного потенциала, численно решив задачу (6). Это распределение потенциала используем для нахождения силы гравитации, стоящей в правой части системы уравнений (1), и, наконец, численно решив систему (1), найдем распределение физических параметров в следующий момент времени.

Запишем систему уравнений газовой динамики (1) в цилиндрической системе координат в виде:

$$\frac{\partial q}{\partial t} + \frac{\partial F}{\partial r} + \frac{1}{r}\frac{\partial G}{\partial \varphi} + \frac{\partial H}{\partial z} = \Psi, \text{где} \tag{9}$$

$$q = \{\, r\rho,\, r\rho u,\, r\rho v,\, r\rho w,\, r\rho e\}^T \tag{10}$$

$$F = \left\{\, r\rho\, u,\, r\rho\, u^2 + rp,\, r\rho\, uv,\, r\rho uw,\, r\rho uh\right\}^T$$

$$G = \left\{\, r\rho\, v,\, r\rho\, uv,\, r\rho\, v^2 + rp\,,\, r\rho vw,\, r\rho vh\right\}^T$$

$$H = \left\{\, r\rho\, w, r\rho uw,\, r\rho vw,\, r\rho\, w^2 + rp, r\rho wh\right\}^T$$

$$\Psi = \left\{\, 0, p + \rho v^2 + r\rho F_r,\, -\rho vu + r\rho F_\varphi,\, r\rho F_z, r\rho(\vec{F}, \vec{v})\right\}^T$$

Для решения этой системы уравнений в области Ω построим сетку $\bar{\omega} = \omega_r \times \omega_\varphi \times \omega_z$, где

$$\omega_r = \left\{ r_i;\ r_i = r_1 + i \cdot \Delta r\ ;\ i = 0, ..., N_r + 1\ ; \Delta r = \frac{r_2}{N_r} \right\}$$

$$\omega_\varphi = \left\{ \varphi_i;\ \varphi_i = i \cdot \Delta\varphi\ ;\ i = 0, ..., N_\varphi + 1\ ; \Delta\varphi = \frac{2\pi}{N_\varphi} \right\} \tag{11}$$

$$\omega_z = \left\{ z_i;\ z_i = i \cdot \Delta z\ ;\ i = 0, ..., N_z + 1\ ; \Delta z = \frac{z_2}{N_z} \right\}$$

Будем использовать явную разностную схему первого порядка аппроксимации - схему Роу, которая в цилиндрической системе координат представляется в виде [14]:

$$\frac{\hat{q}_{i,j,k}-q_{i,j,k}}{\tau}+\frac{F_{i+1/2,j,k}-F_{i-1/2,j,k}}{\Delta r}+\frac{G_{i,j+1/2,k}-G_{i,j-1/2,k}}{r\Delta\varphi}+\frac{H_{i,j,k+1/2}-H_{i,j,k-1/2}}{\Delta z}=\Psi_{i,j,k}\quad,$$

где $\hat{q}_{i,j,k}=q(r_i,\,\varphi_j,\,z_k,\,t_{n+1})$ - функции с верхнего временного слоя, а потоки в полуцелых точках вычисляются по следующим формулам:

$$A=\frac{\partial F}{\partial q},B=\frac{\partial G}{\partial q},C=\frac{\partial H}{\partial q}$$

$$v=\{u,v,w\}^T,$$

$$e=\varepsilon+\frac{v^2}{2},h=\varepsilon+\frac{P}{\rho}+\frac{v^2}{2}$$

$$R_A=\begin{pmatrix}1 & 0 & 0 & 2 & 1\\ u-c & 0 & 0 & 2u & u+c\\ v & 2c & 0 & 2v & v\\ w & 0 & 2c & 2w & w\\ h-uc & 2vc & 2wc & v^2 & h+uc\end{pmatrix},$$

$$\Lambda_A=\operatorname{diag}\{u-c,u,u,u,u+c\},$$

$$F_{i+1/2,j,k}=\frac{F_{i,j,k}+F_{i+1,j,k}}{2}-\frac{1}{2}\sum_m\left|\lambda_A^m(q^*_{i+1/2,j,k})\right|\Delta s^m_{i+1/2,j,k}r_A^m(q^*_{i+1/2,j,k}),$$

$$\Delta s^{1,5}_{i+1/2,j,k}=\frac{1}{2c^{*2}_{i+1/2,j,k}}\left[(P_{i+1,j,k}-P_{i,j,k})\mp\rho^*_{i+1/2,j,k}c^*_{i+1/2,j,k}(u_{i+1,j,k}-u_{i,j,k})\right],$$

$$\Delta s^{2}_{i+1/2,j,k}=\frac{1}{2c^{*2}_{i+1/2,j,k}}\left[\rho^*_{i+1/2,j,k}c^*_{i+1/2,j,k}(v_{i+1,j,k}-v_{i,j,k})\right],$$

$$\Delta s^{3}_{i+1/2,j,k}=\frac{1}{2c^{*2}_{i+1/2,j,k}}\left[\rho^*_{i+1/2,j,k}c^*_{i+1/2,j,k}(w_{i+1,j,k}-w_{i,j,k})\right],$$

$$\Delta s^{4}_{i+1/2,j,k}=\frac{1}{2c^{*2}_{i+1/2,j,k}}\left[c^{*2}_{i+1/2,j,k}(\rho_{i+1,j,k}-\rho_{i,j,k})-(P_{i+1,j,k}-P_{i,j,k})\right],$$

$$\rho^*_{i+1/2,j,k}=\sqrt{\rho_{i+1,j,k}\rho_{i,j,k}},\,v^*_{i+1/2,j,k}=\frac{\sqrt{\rho_{i,j,k}}v_{i,j,k}+\sqrt{\rho_{i+1,j,k}}v_{i+1,j,k}}{\sqrt{\rho_{i,j,k}}+\sqrt{\rho_{i+1,j,k}}},$$

$$h^*_{i+1/2,j,k}=\frac{\sqrt{\rho_{i,j,k}}h_{i,j,k}+\sqrt{\rho_{i+1,j,k}}h_{i+1,j,k}}{\sqrt{\rho_{i,j,k}}+\sqrt{\rho_{i+1,j,k}}},$$

$$c^*_{i+1/2,j,k}=\sqrt{\frac{\sqrt{\rho_{i,j,k}}c^2_{i,j,k}+\sqrt{\rho_{i+1,j,k}}c^2_{i+1,j,k}}{\sqrt{\rho_{i,j,k}}+\sqrt{\rho_{i+1,j,k}}}+\frac{\gamma-1}{2}\frac{\sqrt{\rho_{i+1,j,k}\rho_{i,j,k}}}{(\sqrt{\rho_{i,j,k}}+\sqrt{\rho_{i+1,j,k}})^2}(v_{i+1,j,k}-v_{i,j,k})^2}.$$

Аналогично записываются выражения для $G_{i,j+1/2,k}$ и для $H_{i,j,k+1/2}$.

Компоненты силы гравитации $(\mathbf{F}_r,\mathbf{F}_\varphi,\mathbf{F}_z)$, входящие в правую часть Ψ на каждом временном слое, находятся численным решением системы уравнений для гравитационного потенциала (6).

Для решения системы (6) аппроксимируем уравнение Пуассона и граничные условия следующим образом [9]:

$$\frac{1}{r_{i,j}}\frac{1}{\Delta r}\left[r_{i+\frac{1}{2},j}\frac{1}{\Delta r}\left(\Phi_{i+1,j}-\Phi_{i,j}\right)-r_{i-\frac{1}{2},j}\frac{1}{\Delta r}\left(\Phi_{i,j}-\Phi_{i-1,j}\right)\right]+$$

$$+\frac{1}{\Delta z^2}\left(\Phi_{i,j+1}-2\Phi_{i,j}+\Phi_{i,j-1}\right)=4\pi\ G\rho_{i,j}\quad , \tag{12}$$

$$i=1,..N_r,\quad j=1,..N_z$$

$$\frac{1}{\Delta r}\left(\Phi_{i+1,j}-\Phi_{i,j}\right)=0\quad ,\quad i=0,\quad j=1,..N_z$$

$$\frac{1}{\Delta z}\left(\Phi_{i,j+1}-\Phi_{i,j}\right)=0\quad ,\quad j=0,\quad i=1,..N_r$$

Правая часть в уравнении (12) берется с нижнего временного слоя.

3. Результаты расчетов. В рамках описанной выше модели рассматривалась сферически симметричная звезда с центральной плотностью $5\bullet 10^{9}$ г/см^{3} и температурой 10^{8} K. Дальнейшее распределение температуры по радиусу подбиралось таким образом, чтобы при нахождении начальной одномерной равновесной конфигурации звезды из системы уравнений (8) масса этой звезды (далее $\mathrm{M_З}$) получалась не меньше чем $1.5\mathrm{M_\Theta}$, а масса вещества в оставшейся расчетной области – не больше 1% от получившейся $\mathrm{M_З}$. Границей звезды при этом считалась точка, в которой происходило падение центральной плотности на два порядка. Эти условия достигаются, если взять распределение температуры так, как показано на рис. 2. При этом $\mathrm{M_З}$ получается равной 1.868 $\mathrm{M_\Theta}$, масса фона 0.014 $\mathrm{M_\Theta}$, что составляет 0.7% $\mathrm{M_З}$. Соответствующие одномерные распределения плотности и давления, найденные путем численного решения системы уравнений (8), изображены на рис.3 и 4.

Найденное распределение физических параметров используется в качестве начальных данных для численного решения системы уравнений (1) в области Ω и уравнения (6) в области G со свободными граничными условиями (5) на внешней границе, условиями симметрии на нижней и внутренней границе и следующими граничными условиями для гравитационного потенциала:

$$\left.\frac{\partial\Phi}{\partial r}\right|_{r=0}=0\ ,\quad \left.\frac{\partial\Phi}{\partial z}\right|_{z=0}=0\ ,\quad \Phi|_{r=r_2,\,z=z_2}=\mu(r,\varphi,z)\quad . \tag{13}$$

Как было показано нами в работе [19], для нахождения гравитационного потенциала на внешней границе Γ_2 и Γ_3 следует использовать его точное интегральное представление (7), которое аппроксимируется по формуле:

$$\mu_{m\,l\,n}=-G\sum_{\substack{i\ \neq\ m\\ j\ \neq\ l\\ k\ \neq\ n}}\frac{M_{i\,j\,k}}{R^{m\,l\,n}_{i\,j\,k}}=-G\sum_{\substack{i\ \neq\ m\\ j\ \neq\ l\\ k\ \neq\ n}}\frac{\rho_{i\,j\,k}\ r_i\Delta r\,\Delta\varphi\ \Delta z}{\sqrt{r_i^2+r_m^2-2r_ir_m\cos(\varphi_j-\varphi_l)+(z_j-z_n)^2}} \tag{14}$$

В этой формуле $R^{m\,l\,n}_{i\,j\,k}$ - это расстояние от ячейки сетки с координатами i,j,k и массой $M_{i\,j\,k}$, которая создает потенциал, до ячейки с координатами m,l,n, в которой ищем потенциал.

На рис.5 и 6 изображены линии уровня плотности и давления, соответствующие начальному распределению плотности (в сечении $\phi=\pi/4$), которое было получено путем сферически симметричного распространения решения одномерной системы уравнений (8) в трехмерную расчетную область.

Во всех расчетах в области Ω (4) строилась равномерная сетка $\bar{\omega}$ (11) с числом ячеек $N_r = N_z = N_\varphi = N$, где N варьировалось от 50 до 150. Проведенные расчеты показали, что при увеличении числа ячеек сетки результат расчета качественно и количественно практически не изменяется.

На рис.7–10 представлены характерные картины процесса установления равновесия в области Ω на два момента времени: $t_1 = 7.976$ и $t_2 = 13.34$. Время t_1 соответствует середине расчета, а t_2 - окончанию расчета, когда равновесная конфигурация считается найденной. В процессе расчетов, исходя из критерия Куранта, шаг по времени определялся автоматически по формуле:

$$\tau = \sigma \min_{i,m} \{\Delta r / |\lambda_i^m|\} \quad , \tag{15}$$

где $0 < \sigma \leq 1$ - параметр схемы, в связи с чем картины течения на различных сетках по пространству сравнивались на ближайшие моменты времени.

Расчеты показали, что в процессе установления равновесия отклонение плотности составляло от 0.001 до 0.019% от начального состояния (линии уровня плотности на различные моменты времени построены на рис.7 и 9). Отклонение полной массы (массы звезды и массы фона) доходило до 0.04%, но оно появлялось не за счет перераспределения плотности внутри звезды, а за счет свободных граничных условий, которые позволяли изменяться массе фона из-за перетекания вещества через внешнюю границу. Граница звезды в процессе расчета практически не передвигалась относительно начального положения, ее максимальное отклонение наблюдалось на одну пространственную ячейку при $N = 50$ и на две – при $N = 100$.

На рис.11 и 12 показано, как распределены плотность и давление по радиусу в последний момент времени. Сравнение этих графиков с графиками начального распределения плотности и давления (рис.3 и 4) и с соответствующими графиками, построенными на различные моменты времени в процессе расчета, позволяет сделать вывод о том, что одномерная равновесная конфигурация звезды, найденная путем решения системы уравнений (8) и взятая в качестве начальных данных для трехмерной системы уравнений (1)(6), остается почти неизменной (стационарной) в течении длительного расчетного времени.

Заключение. В настоящей работе путем проведения серии численных экспериментов на основе трехмерной нестационарной газодинамической модели получены равновесные конфигурации звезды с массой равной 1.868 M_Θ на различных пространственных сетках. Найденные равновесные конфигурации могут быть использованы в качестве начальных данных при численном трехмерном моделировании газодинамических процессов на поздних этапах эволюции звезд.

Рис. 1. Область Ω и подобласть G.

Рис. 2. Распределение температуры по радиусу звезды.

Рис. 3. Начальное распределение плотности по радиусу звезды.

Рис. 4. Начальное распределение давления по радиусу звезды.

Рис. 5.

Рис. 6.

Рис. 7.

Рис. 8.

Рис. 9.

Рис. 10.

Рис. 11. Распределение плотности по радиусу звезды в момент времени t_2 .

Рис. 12. Распределение давления по радиусу звезды в момент времени t_2 .

СПИСОК ЛИТЕРАТУРЫ

1. *Тассуль Ж.Л.* Теория вращающихся звезд // М.: Наука. Физматлит. 1982. 472с.
2. *Абакумов М.В., Мухин С.И., Попов Ю.П., Чечеткин В.М.* Исследование равновесных конфигураций газового облака вблизи гравитирующего центра // препринт ИПМ им. М.В.Келдыша РАН № 33, 1995.
3. *Баранов В.Б.* Устойчивость течений в гидроаэромеханике // Соросовский образовательный журнал, № 9, 1999, с.106-111.
4. *Богоявленский О.И.* Автомодельные адиабатические движения самогравитирующего газа в звездах// Письма в ЖЭТФ, том 27, вып.2, с.91-94.
5. *Aksenov, A.G., Blinnikov, S.I.* A Newton iteration method for obtaining equilibria of rapidly rotating stars // Astronomy and Astrophysics 290, 1994, p.674-681.
6. *Hachisu I.* A versatile method for obtaining structures of rapidly rotating stars // The Astrophysical Journal Supplement Series, 61, 1986, p.479-507.
7. *Ландау Л.Д., Лифшиц Е.М.* Теория поля // М. Наука, 1962, 568с.
8. *Самарский А.А., Попов Ю.П.* Разностные методы решения задач газовой динамики // М. Наука, 1992, 423с.
9. *Самарский А.А.* Теория разностных схем // М. Наука, 1971, 656с.
10. *Самарский А.А., Николаев Е.С.* Методы решения сеточных уравнений // М. Наука, 1978, 378 с.
11. *Дубошин Г.Н.* Теория притяжения // М. Государственное издательство физико-математической литературы, 1961, 288 с.
12. *Джордж А., Лю Дж.* Численное решение больших разреженных систем уранений // М. Мир, 1984, 77 с.
13. *Гончаров А.Л.* Комплекс программ для решения разреженных систем алгебраических уравнений // отчет ИПМ за 1986 г.
14. *Кузнецов О.А.* Численное исследование схемы Роу с модификацией Эйнфельдта для уравнений газовой динамики // препринт ИПМ им. М.В. Келдыша РАН № 43, 1998.
15. *Абакумов М.В.* О некоторых методах визуализации сеточных данных // препринт ИПМ им. М.В.Келдыша РАН № 72, 2004.
16. *Лихтенштейн Л.* Фигуры равновесия вращающейся жидкости. // М.: Наука,1965.
17. *Blinnikov, S.I.* Self-consistent field method in the theory of rotating stars // Astronomicheskii Zhurnal 52, 1975, p.243-254.
18. *Blinnikov, S.I., Sasorov, P.V., Woosley, S.E.* Self-Acceleration of Nuclear Flames in Supernovae // Space Science Reviews 74, 1995, p.299-311.
19. *Барская И.С., Мухин С.И., Чечеткин В.М.* Математическое моделирование равновесных конфигураций самогравитирующего газа // препринт Института Прикладной Математики им. М.В. Келдыша № 41, 2006г.

УДК 519.7:004.855.5

ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА АНАЛИЗА И ФИЛЬТРАЦИИ ИНТЕРНЕТ ИНФОРМАЦИИ

© 2007 г. В. В. Глазкова, В. А. Масляков, И. В. Машечкин, М. И. Петровский

glazv@mail.ru, maslyakov@gmail.com, mash@cs.msu.su, michael@cs.msu.su

Кафедра Автоматизации систем вычислительных комплексов
Лаборатория Технологий программирования

Введение.

Развитие современного образования и науки без активного использования ресурсов Интернет в учебных и научных учреждениях абсолютно невозможно. В настоящее время Интернет ресурсы использует подавляющее большинство как государственных, так и коммерческих организаций любого уровня. Однако использование этих ресурсов, как правило, не контролируемо, в связи с чем возникает необходимость анализировать и фильтровать входящий и исходящий Интернет трафик в этих организациях.

Проблема контроля доступа к Интернет ресурсам актуальна и имеет важное значение по следующим основным причинам:

- блокирование доступа к нелегальной (экстремистской, антисоциальной и другой) информации;
- предотвращение использования Интернет ресурсов не по назначению, в частности, ограничение и контроль доступа к развлекательным и другим ресурсам для личного пользования;
- предотвращение утечки конфиденциальной информации через Интернет.

В результате исследования, проведённого аналитиками Европейского Союза в рамках проекта “Safer Internet. Internet Filtering and Rating”, было опрошено порядка 600 крупных и мелких организаций и установлено, что 90% организаций хотели бы использовать специализированное программное обеспечение для контроля Интернет трафика [1]. Согласно ещё одному исследованию, проведенному компанией Salary.com [2] средний американский сотрудник тратит 2.1 из 8 часов своего рабочего времени на личное пользование. Порядка 45% этого времени он тратит на нецелевое использование Интернет ресурсов, что ежегодно обходится американским работодателям в 759 миллиардов долларов.

В России данная проблема не менее актуальна. Так, к примеру, в рамках национального проекта “Образование” планируется подключить к Интернету свыше 10000 школ и учебных заведений, в которых необходимо будет осуществлять контроль и фильтрацию Интернет трафика в связи с указанными выше причинами. О несовершенстве современных систем фильтрации говорилось в рамках круглого стола (сентябрь 2005 года) на тему “Безопасность детей в Интернете”.

Авторами настоящей статьи предлагается новый подход, который заключается в применении методов машинного обучения и интеллектуального анализа данных (Data Mining) при построе-нии системы анализа и фильтрации Интернет трафика. Такие методы позволят разрабатываемой системе легко адаптироваться к постоянно изменяющейся природе Интернет ресурсов и учитывать специфику анализа сетевого трафика для различных организаций и стран.

Применение разрабатываемой интеллектуальной системы анализа и фильтрации Интернет трафика должно обусловить совершенствование технологических процессов в организациях (как с точки зрения снижения издержек производства, так и с точки зрения повышения

информационной безопасности). К социальным эффектам использования разрабатываемой системы можно отнести ограничение доступа к ресурсам, содержащим нелегальную информацию (экстремистскую, антисоциальную и другую), а также развитие учебных и научных сетевых сообществ (за счёт контроля и анализа посещаемых ими ресурсов).

1. Обзор существующих систем фильтрации Интернет информации.

В силу обозначенной актуальности проблемы и интереса к подобным программным средствам со стороны учебных, научных и других организаций, разработка систем анализа и фильтрации Интернет трафика проводилась, как в прошлом, так и проводится в настоящее время. Попытки решения проблемы контроля доступа к Интернет информации предпринимаются по сей день как отдельными коммерческими и некоммерческими организациями, так и правительствами целых стран, например, таких как Китай, Иран, Саудовская Аравия.

На сегодняшний день существует множество как коммерческих, так и некоммерческих решений. К наиболее распространённым коммерческим продуктам можно отнести: SurfControl (http://www.surfcontrol.com), WebSense (http://www.websense.com) и CyberPatrol (http://www.cyberpatrol.com). Среди open-source решений стоит отметить: Poesia (http://www.poesia-filter.org) и SIFT (http://www.sift-platform.org).

Перед рассмотрением характеристик существующих систем введём терминологию. Термин *пользователь* системы понимается в расширенном смысле, то есть это может быть как человек, пытающийся получить доступ к некоторому Интернет ресурсу, так и некоторая программа-агент, запрашивающая Интернет информацию. Под *ресурсом* понимается блок информации, который может быть запрошен пользователем. *Контент* – это содержательная часть ресурса. Помимо содержательной части ресурса у него может быть *мета информация* – информация, описывающая ресурс и не входящая в его контент. Примером мета информации может быть размер ресурса, его тип, дата последней модификации, кодировка и прочее. *URI – uniform resource identifier* (уникальный идентификатор ресурса) – это уникальная метка, позволяющая однозначно идентифицировать каждый Интернет ресурс [3].

Так как на сегодняшний день не существует чёткой классификации существующих систем [20], авторами было проведено исследование и предложена следующая классификация систем фильтрации Интернет трафика (рис. 1).

Рис.1 Классификация систем фильтрации Интернет трафика.

Три наиболее важных критерия – это масштаб системы, способ и время анализа трафика.

По масштабу системы можно разделить на системы:

- комплексные и внедряемые в масштабах целой страны;
- средней сложности, рассчитанные на использование большим количеством пользователей и предоставляемые, как правило, в качестве отдельной услуги интернет провайдерами; и
- независимые системы, устанавливаемые и настраиваемые в рамках отдельных локальных сетей или организаций.

По способу анализа все системы можно разбить на два больших класса:

- анализирующие лишь общую(мета) информацию о ресурсе и
- анализирующие в том числе и содержимое ресурса.

По времени анализа все системы можно также разбить на два класса:

- анализирующие информацию в реальном времени (онлайн), то есть во время запроса пользователем Интернет ресурса;
- анализирующие информацию в отложенном режиме (оффлайн), то есть после того, как пользователь получил доступ к ресурсу.

Важным критерием для систем масштаба локальных сетей также является способ блокирования трафика. Так одни системы используют для этого статичные списки доменов и адресов, другие древовидные каталоги ресурсов (вроде directory.google.com), третьи – статические списки меток, которыми помечается каждый из ресурсов. Последний вид систем фильтрации так же часто называют системами классификации трафика.

Авторам наиболее интересен класс систем фильтрации масштаба локальных сетей, осуществляющих фильтрацию http трафика на основе мета информации и контента ресурсов с использованием Data Mining методов; с локальной базой данных ресурсов и результатов классификации и с возможностью анализировать трафик как в реальном режиме времени (онлайн), так и в отложенном режиме (оффлайн).

Традиционно в существующих системах анализа и фильтрации Интернет информации применяется так называемый сигнатурный подход, основанный на использовании экспертной базы знаний адресов Интернет ресурсов. Такая база знаний содержит адреса ресурсов, с каждым из которых связан набор тем (категорий), к которым, по мнению экспертов, относится данный Интернет ресурс.

Однако системы, основанные на экспертных базах знаний адресов, обладают рядом недостатков, основные из которых перечислены ниже.

- Невозможность анализировать трафик в реальном времени (онлайн). Анализ в реальном времени необходим, когда содержимое (контент) одного и того же ресурса может динамичес-ки изменяться во времени, а на сегодняшний день это свойственно подавляющему большинс-тву Интернет ресурсов.
- При анализе Интернет ресурсов никак не учитывается их содержимое, что приводит к существенному снижению точности таких систем.
- Невозможность анализа исходящего Интернет трафика (для предотвращения утечки конфиденциальной информации).
- Необходимость использования внешних баз знаний о ресурсах, что может быть недопустимо по соображениям безопасности.

- Качество функционирования таких систем существенно зависит от качества и оперативности компаний, поддерживающих постоянное обновление баз знаний. Как правило, для поддержа-ния баз знаний в актуальном состоянии требуется большое количество экспертов. В связи со стремительными темпами роста Интернета, осуществлять обновление баз знаний становится всё сложней и сложней как с технической, так и с экономической точки зрения.

- Ориентация на обработку конкретного набора языков, что делает систему неприменимой для анализа других языков.

Таким образом, применение сигнатурного подхода для анализа трафика имеет ряд существенных недостатков, связанных с неспособностью этого подхода адаптироваться к постоянной динамике изменения Интернет ресурсов.

Актуальным состоянием мировых исследований рассматриваемой проблемы является разработка и привлечение методов интеллектуального анализа данных (Data Mining) для анализа Интернет информации. Интеллектуальные методы обладают следующими достоинствами:

- самообучаемость и адаптируемость – способность автоматически оперативно подстраиваться к динамически изменяющемуся содержимому Интернет ресуров;

- автономность – независимость от внешних баз знаний и экспертов.

В силу перечисленных достоинств в настоящее время данное направление исследований интенсивно развивается как в России, так и за рубежом. Ежегодно публикуются сотни работ, проводятся специализированные конференции и симпозиумы (например, “IEEE/WIC/ACM International Conference on Web Intelligence” http://www.comp.hkbu.edu.hk/iwi06/wi/, “Atlantic Web Intelligence Conference” http://www.awic2007.net/), появляются исследовательские проекты (Poesia http://www.poesia-filter.org). Однако на сегодняшний день пока не существует промышленных систем анализа и фильтрации Интернет информации, основанных на методах интеллектуального анализа данных.

Основными количественными показателями систем фильтрации Интернет трафика являются:

- точность анализа - процент верно отфильтрованных Интернет ресурсов;

- излишнее блокирование или ложно-положительные ошибки - процент “хороших” ресурсов, ошибочно запрещенных системой фильтрации;

- недостаточное блокирование или ложно-отрицательные ошибки - процент “плохих” ресурсов, ошибочно разрешенных системой фильтрации;

- скорость анализа - максимальный объем данных, который система может проанализировать в единицу времени.

На сегодняшний день качество систем фильтрации трафика по-прежнему остается достаточно низким: при максимально достижимой точности анализа 90% системы имеют очень большой процент ложно-положительных ошибок (2-5%) и достаточно медленную скорость, что вызывает заметные для пользователей задержки.

2. Предложенное решение.

Авторами предлагается новый подход, который заключается в применении методов машинного обучения и интеллектуального анализа данных при построении системы анализа и фильтрации Интернет трафика. Такие методы позволят разрабатываемой системе легко адаптироваться к постоянно изменяющейся природе Интернет ресурсов и учитывать специфику анализа сетевого трафика для различных организаций и стран (в частности, поддерживать работу с произвольными языками), что без сложной и дорогостоящей доработки не позволяет сделать ни одно из существующих на сегодняшний день решений.

Предлагаемый подход позволит организациям определять произвольные категории классификации Интернет ресурсов и задавать гибкие политики фильтрации, что обеспечивает адаптацию функционирования разрабатываемой системы к потребностям конкретных организаций. Также отметим, что в предлагаемом подходе учитывается, что Интернет ресурсы, как правило, являются многотемными, то есть каждый Интернет ресурс может быть отнесен более чем к одной уместной теме или категории.

Основная идея предлагаемого подхода к решению задачи фильтрации Интернет трафика состоит в следующем. На основе обучающей совокупности, состоящей из заранее рубрицированных ресурсов, строится математическая модель классификации, которая позволит определять уместные категории для произвольных ресурсов схожей природы. Впоследствии, эта математическая модель может уточняться за счёт пошагового дообучения на новых ресурсах, для которых известны уместные категории.

Таким образом, предлагается осуществлять категоризацию ресурсов и содержимого Интернет трафика на основе не только статических правил, заданных экспертом, но и на основе построения и применения Data Mining моделей классификации гипертекстовой информации, что позволяет сделать систему адаптивной и обучаемой. Исследование существующих методов классификации многотемных объектов применительно к задаче фильтрации Интернет информации показало, что эти методы не имеют возможности дообучения, которая очень важна для рассматриваемой задачи. Авторами предлагается новый метод классификации многотемных объектов с возмож-ностью дообучения.

Применение такого интеллектуального метода классификации с возможностью дообучения соответственно потребовало разработки новых подходов к организации архитектуры системы. В рамках настоящего исследования предлагается новый тип архитектур систем фильтрации Интернет трафика, обладающих следующими основными свойствами:

- анализ трафика в режиме реального времени (онлайн);
- анализ ресурсов на основе общей (мета) информации и их содержимого с использованием дообучаемых Data Mining методов;
- динамический список категорий;
- поддержка анализа исходящего трафика;
- автономность и локальность баз данных;
- масштабируемость, расширяемость, переносимость.

2.1. Архитектура интеллектуальной системы анализа и фильтрации Интернет информации.

Авторами настоящей статьи была предложена и специфицирована система, использующая интеллектуальные методы анализа трафика, а именно multi-label класификацию трафика. При этом подходе, в системе задаётся некоторый список классов ресурсов, а каждому из пользователей локальной сети ставится в соответствие список разрешённых и запрещённых классов ресурсов. При запросе пользователем очередного Интернет ресурса, который ранее не был классифицирован, ресурс классифицируется, и система на основе этой и другой информации принимает решение о том, разрешать или запретить пользователю доступ к данному ресурсу. На рисунке 2 представлена предложенная архитектура интеллектуальной системы анализа и фильтрации Интернет информации.

Рис. 2 Архитектура системы.

В текущей реализации планируется использовать стабильный и достаточно эффективный open-source прокси-сервер Squid. Для общения с ядром будет использоваться стандартый протокол ICAP [4], для этого на Squid будет установлен специальный ICAP-клиент, а в ядро системы интегрирован ICAP-сервер. ICAP – это HTTP-подобный протокол, используемый для переадресации запросов с прокси сервера на различные системы фильтрации, анти-вирусы, спам фильтры и прочее.

Ядро будет реализовано в качестве Java приложения с back-end базой данных XML BerkleyDB, в которой будут хранится списки пользователей, меток, запросов пользователей, результаты классификации и прочее. Выбор XML Berkley DB продиктован эффективностью и хранением данных в XML-виде, так как для описания ресурсов будут использоваться стандарты RDF и OWL [5]

Основные части системы подключаются к ядру в виде модулей и могут общаться с ядром через XML-RPC API [21]. В свою очередь ядро общается с остальными модулями через единственный модуль принятия решений (Decision Making Module) через его XML-RPC API, вся роль которого сводится к тому, чтобы для определённого пользователя и конкретного ресурса сказать блокировать доступ к нему или нет.

Модуль же принятия решений может пользоваться любой информацией о ресурсе и пользователе, в частности, он может запрашивать у модуля классификации (если понадобится), дополнительную информацию о метках данного ресурса. Если же ресурс был ранее проклассифицирован, модуль принятия решения может запросить у ядра результаты из кеша.

В целом работа системы абсолютно прозрачна для пользователя и не требует от него никаких дополнительных настроек.

2.2. Методы интеллектуального анализа данных для фильтрации Интернет информации.

Рассматриваются две основные подзадачи, которые должен решать модуль классификации разрабатываемой системы анализа и фильтрации интернет информации: представление web-документов и multi-label классификация. Для решения рассматриваемых проблем авторами используются методы интеллектуального анализа данных.

Задача multi-label классификации.

Исследование проблемы анализа интернет информации показало, что большинство возникающих практически значимых задач классификации являются задачами с существенно перекрывающимися классами. Web-документы, как правило, являются многотемными (англ. multi-label), т.е. они могут быть отнесены более чем к одной релевантной теме (или классу). Соответственно, для классификации web-документов необходимы эффективные методы решения задачи классификации многотемных документов (multi-label классификации). Эта задача заключается в нахождении набора тем, релевантных для заданного классифицируемого документа.

Задача multi-label классификации является расширением традиционной задачи multi-class классификации, которая состоит в отнесении классифицируемого объекта к одному из нескольких предопределённых взаимоисключающих классов. В задаче же multi-label классификации классифицируемый объект может принадлежать сразу к нескольким релевантным классам, и классы не являются взаимоисключающими. С задачей multi-label классификации тесно связана задача ранжирования, которая заключается в упорядочении всех классов (из предопределённого набора классов) по степени их релевантности для заданного классифицируемого объекта.

Задача multi-label классификатора состоит в предсказании всех релевантных классов для заданного документа. Среди существующих методов multi-label классификации можно выделить три основные группы: методы, основанные на декомпозиции multi-label проблемы в набор независимых подпроблем бинарной классификации [6,7,8]; методы multi-label классификации, основанные на оценке апостериорных вероятностей наборов классов [9,10] и методы multi-label классификации, основанные на сведении проблемы ранжирования к проблеме multi-label классификации [6,7,11,12,13].

Решение проблемы multi-label классификации на основе ранжирования включает два этапа. Первый этап состоит в обучении алгоритма ранжирования, который упорядочивает все классы по степени их релевантности для заданного классифицируемого объекта. Второй этап заклю-чается в построении функции multi-label классификации, отделяющей высоко ранжированные релевантные классы от нерелевантных. Как правило, на втором этапе используются пороговые функции [6,7,11,12,13]. Проблема заключается в том, что порог обычно зависит от классифицируемых объектов, поэтому использование фиксированного порога (одинакового для всех объек-тов) [6,7,11] может приводить к низкой точности классификации. Elisseeff и J. Weston предложили находить пороговую функцию в пространстве признаков классифицируемых объектов [12,13]. Однако для многих практических приложений (например, для задачи текстовой клас-сификации) временная и пространственная сложность вычисления пороговой функции в пространстве признаков очень велика (так как, например, размерность пространства признаков текстовых документов порядка количества слов в словаре).

Авторами предлагается новый подход для сведения проблемы ранжирования к проблеме multi-label классификации [14,15,16,17]. В то время как в существующих методах сложные решающие функции строятся в пространстве признаков, в представленном подходе предлагается строить простые линейные функции multi-label классификации в пространстве релевантностей классов, используя результат работы алгоритмов ранжирования как новое множество признаков анализируемого электронного документа. Пространство релевантностей классов – это прос-транство векторов, координатами которых являются значения релевантностей классов для документов. Обычно размерность пространства релевантностей классов значительно меньше размерности пространства признаков. В рамках предложенного подхода разработано

два новых метода. В первом методе TCRS (Threshold function on the Class Relevance Space) строится линейная пороговая функция, определённая в пространстве релевантностей классов, а во втором методе LORM (Linear Operator from class Ranks to Multi-label decision functions, $f{=}Ar$) находится линейный оператор, отображающий ранги классов в набор значений решающих вектор-функций multi-label классификации. По сравнению с существующими методами, предложенные методы вычислительно более эффективны и, в то же время, демонстрируют сравнимую, а в некоторых случаях значительно лучшую точность классификации.

Исследование существующих методов классификации многотемных объектов применительно к задаче фильтрации Интернет информации показало, что эти методы не имеют возможности дообучения, которая очень важна для рассматриваемой задачи. Авторами предлагается новый метод классификации многотемных объектов для решения рассматриваемой задачи [18], обладающий следующими свойствами:

- возможность дообучения;
- высокая точность классификации;
- возможность классификации новых ресурсов в режиме реального времени;
- возможность динамического изменения списка тем (добавление/удаление тем);
- относительно небольшое время обучения (дообучения) и эффективность по памяти.

Задача представления web-документов.

Перед подачей web-документов на вход алгоритму multi-label классификации, очень важно выбрать для них некоторое формальное представление. Так как мы имеем дело с гипертекстовыми документами, необходимо, чтобы в этом представлении учитывались как текст, содержащийся в документах, так и наличие гиперссылок между документами. Это связано с тем, что, в отличие от задачи текстовой классификации, в которой классифицируемый объект – это обособленный документ (т.е. просто последовательность терминов), в рассматриваемой задаче классификации web-документов классифицируемый объект не является обособленным. Web-документы содержат навигационные ссылки, ссылки на цитаты, списки ресурсов и т.д. Web-документ, как правило, имеет входящих соседей (документы, цитирующие данный) и исходящих соседей (документы, которые данный документ цитирует). Поэтому учёт гиперссылок между документами должен позволить получить более точное (для классификации) представление документа, по сравнению с учётом только локального текста классифицируемого документа. Основная идея представления web-документов с учётом гиперссылок следующая [19]. Сначала классифицируем неклассифицированные документы-соседи тестового документа (используя, например, просто текстовый классификатор), а затем каждую гиперссылку в тестовом документе заменяем идентификаторами классов соответствующего документа-соседа. Таким образом, такое представление документов основывается на информации, которая является как локальной, так и нелокальной по отношению к данному документу.

3. Заключение.

Авторами настоящей статьи предлагается новый подход, который заключается в применении методов машинного обучения и интеллектуального анализа данных при построении системы анализа и фильтрации Интернет трафика. Такие методы позволят разрабатываемой системе легко адаптироваться к постоянно изменяющейся природе Интернет ресурсов и учитывать специфику анализа сетевого трафика для различных организаций и стран. Систему такого уровня можно применять для анализа и фильтрации Интернет трафика в локальных сетях любых государственных, коммерческих и общественных организаций. Основными конкурентными преимуществами разрабатываемой системы являются её адаптируемость и точность работы.

Настоящая работа поддерживается грантами РФФИ 05-01-00744, 06-01-00691 и 06-07-08035-офи; грантом поддержки ведущих научных школ НШ №02.445.11.7427; грантом Президента РФ МК-4264.2007.9.

СПИСОК ЛИТЕРАТУРЫ

1. *Karl Donert End-user* Requirements (http://www.poesia-filter.org/pdf/Deliverable_2_1.pdf).
2. *Dan Malachowski,* Salary.com Wasted Time At Work Costing Companies Billions (http://www.sfgate.com/cgi-bin/article.cgi?file=/g/a/2005/07/11/wastingtime.TMP&type= printable).
3. *R. Fielding* Uniform Resource Identifier (URI): General Syntax (http://www.ietf.org/rfc/rfc3986.txt).
4. Internet Content Adaptration Protocol (http://www.i-cap.org/).
5. W3C Consortsium. Semantic Web (http://www.w3.org/2001/sw/).
6. *Schapire R. E., Singer. Y.:* Improved boosting algorithms using confidence-rated predictions. Machine Learning, 37(3), 1999, pp. 297-336.
7. *Schapire R. E., Singer. Y.:* BoosTexter: A boosting-based system for text categorization. Machine Learning, 39(2-3), 2000, pp. 135-168.
8. *Boutell M. R., Luo J., Shen X., Brown C.M.:* Learning multi-label scene classification. Pattern Recognition, 37, 2004, pp. 1757-1771.
9. *McCallum A.:* Multi-label text classification with a mixture model trained by EM. Working Notes of the AAAI'99 Workshop on Text Learning, Orlando, FL, 1999.
10. *Zhang M.-L., Zhou Z.-H.:* A k-nearest neighbor based algorithm for multi-label classification. Proceedings of the 1^{st} IEEE International Conference on Granular Computing (GrC'05), Beijing, China, 2005, pp. 718-721.
11. *Comite F. D., Gilleron R., Tommasi M.:* Learning multi-label alternating decision tree from texts and data. Machine Learning and Data Mining in Pattern Recognition, MLDM 2003 Proceedings, Lecture Notes in Computer Science 2734, Berlin, 2003, pp. 35–49.
12. *Elisseeff A., Weston J.:* A kernel method for multi-labelled classification. Proceedings of the 14^{th} Neural Information Processing Systems (NIPS) Conference, Cambridge, 2002.
13. *Elisseeff A., Weston J.* Kernel methods for multi-labelled classification and categorical regression problems. Technical report, BIOwulf Technologies, 2001.
14. *Mikhail Petrovskiy, Valentina Glazkova.* Linear Methods for Reduction from Ranking to Multilabel Classification // Springer-Verlag, Lecture Notes in Artificial Intelligence, 2006, vol. 4304, pp. 1152-1156.
15. *Глазкова В.В.* Исследование и разработка методов классификации многотемных докумен-тов // Сборник тезисов лучших дипломных работ 2006 года, сс. 75-76, М., 2006.
16. *Глазкова В.В., Петровский М.И.* Метод быстрой классификации многотемных текстовых документов // Сборник статей молодых учёных факультета ВМиК МГУ, №3, М., 2006, сс. 55-64.
17. *Глазкова В.В., Петровский М.И.* Методы классификации многотемных документов" // Сборник тезисов XIII Международной конференции студентов, аспирантов и молодых учёных "ЛОМОНОСОВ", секция ВМиК, 2006, сс. 16-17.
18. *Глазкова В. В., Петровский М.И.* Дообучаемый метод классификации многотемных доку-ментов для анализа и фильтрации Интернет информации. // Программные системы и инструменты. Тематический сборник № 7, М.: Изд-во факультета ВМиК МГУ, 2006.
19. *Chakrabarti S., Dom B.E., Indyk P.* Enhanced hypertext categorization using hyperlinks", Proceedings of the ACM International Conference on Management of Data, SIGMOD 1998, pages 307-318.
20. *Ronald J.Deibert and Nart Villeneuve.* Opennet Initiative Introduction to Internet Filtering [HTML] (http://www.opennetinitiative.org/modules.php?op=modload&name=Archive&file=index&req=viewarticle&artid=5).
21. *Joe Johnston.* Using XML-RPC for Web services (http://www-128.ibm.com/developerworks/webservices/library/ws-xpc2/).

УДК 517.95

ИНТЕРАКТИВНЫЕ МЕТОДЫ КОРРЕКЦИИ ПОЛУМЕТРИК

© 2007 г. И. А. Громов

igor_gromov@mail.ru

Кафедра Математических методов прогнозирования

Введение. Работа посвящена исследованию свойств функций расстояния и разработке методов преобразования метрической информации, используемых в интеллектуальном анализе данных. Одним из начальных этапов решения прикладной задачи интеллектуального анализа данных является ее декомпозиция на фундаментальные задачи, такие как классификация, кластеризация и др. Следует отметить, что, как правило, *в постановке* прикладной задачи не бывает указаний на необходимость использовать тот или иной метод попарного сравнения объектов, тем более, не фиксируется функция расстояния (метрика или полуметрика) на множестве объектов распознавания. При решении задач классификации и кластеризации часто возникает потребность формализовать понятие сходства объектов путем задания метрики. Т.е. метрика, введенная на множестве объектов, является не свойством изначальной задачи, а *свойством ее решения*. Следовательно, метрика есть субъективная сущность. Для эффективного использования методов метрического анализа необходимо создать комплекс технических приемов для преобразования метрик.

В интеллектуальном анализе данных предметом исследований часто является конечное фиксированное множество объектов распознавания. Это обусловливает использование *метрик на конечных множествах*. Специфика задач, для решения которых используются методы интеллектуального анализа данных, такова, что выполнение аксиом метрики является слишком жестким требованием, поэтому принято исследовать более широкий класс функций сходства, например, полуметрики.

Как отмечалось выше, выбор полуметрики для решения конкретной задачи субъективен, а значит, можно допустить возможность модифицировать полуметрику с тем, чтобы она точнее отражала характерные сходства и различия исследуемых объектов. Было предложено использовать в качестве источника требований о модификации полуметрик *мнение эксперта*, определенным образом формализованное. В работе рассмотрен следующий случай: эксперт может высказать свое требование об изменении *одного* расстояния. Следует, однако, отметить, что такое (сравнительно простое) пожелание эксперта допускает ряд трактовок и возможность использовать множество методов коррекции полуметрик. В этом заключается интерактивный характер решения задачи: каждый раз строится алгоритм коррекции, наиболее полно и точно интепретирующий пожелания эксперта и при этом минимально деформирующий большую часть расстояний между объектами.

Прежде чем дать формальную постановку задачи, решению которой посвящена статья, определим основные понятия и введем обозначения, используемые далее в тексте. В работе используется стандартное определение метрики. Пусть X - произвольное множество.

Определение. Функция $\rho : X \times X \mapsto \mathbb{R}$ называется *метрикой на* X, если удовлетворяет следующим условиям:

1. $\forall\ x \in X\ \rho(x, x) = 0$;
2. $\forall\ x_1, x_2 \in X\ \rho(x_1, x_2) = \rho(x_2, x_1)$;
3. $\forall\ x_1, x_2 \in X\ \rho(x_1, x_2) \geqslant 0$;
4. $\forall\ x_1, x_2, x_3 \in X\ \rho(x_1, x_3) \leqslant \rho(x_1, x_2) + \rho(x_2, x_3)$;
5. $\forall\ x_1, x_2 \in X\ \rho(x_1, x_2) = 0 \Rightarrow x_1 = x_2$.

Если функция ρ удовлетворяет только условиям (1)-(4), то она называется *полуметрикой на* X. Если функция ρ удовлетворяет только условиям (1)-(3), то она называется *расстоянием, или функцией расстояния, на* X. Условие (4) принято называть *неравенством треугольника*; в дальнейшем для краткости будем обозначать его $(\triangle)$.

Будем рассматривать конечные множества V_N объектов, заданных номерами $1, 2, \ldots, N$. $\rho(i,j) \stackrel{\text{def}}{=} \rho_{ij}$.

R - матрица метрики (полуметрики) ρ, определенной на V_N, и $r_{ij} \stackrel{\text{def}}{=} \rho_{ij}$.

Чтобы избежать дублирования элементов в матрице R, введем множество неупорядоченных пар индексов $E_N = \{(i,j) \mid i,j \in \{1,2,\ldots,N\},\ i \neq j\}$. Нетрудно видеть, что $|E_N| = C_N^2 = N(N-1)/2$.

Функцию расстояния, полученную в результате экспертной модификации, будем обозначать ρ', а функцию расстояния, полученную в результате последующей коррекции - $\widetilde{\rho}$. $\rho'(i,j) = \rho'_{ij} = r'_{ij}$, $\widetilde{\rho}(i,j) = \widetilde{\rho}_{ij} = \widetilde{r}_{ij}$. R' и $\widetilde{R}$ - матрицы функций расстояния ρ' и $\widetilde{\rho}$ соответственно.

Не ограничивая общности рассуждений, будем впредь считать, что эксперт всегда изменяет расстояние между объектами i и j: $\rho(i,j) \mapsto \rho'(i,j)$.

$P_\rho(kl)$ - множество (отрезок) допустимых значений расстояния между объектами k и l в полуметрике ρ. Напомним, что

$$P_\rho(kl) = \left[\max_{i \in V_N} |\rho(i,k) - \rho(i,l)|,\ \min_{i \in V_N} (\rho(i,k) + \rho(i,l))\right]$$

и $\forall (k,l) \in E_N \quad P_\rho(kl) \neq \varnothing$.

$\min(kl)_\rho$ - левая граница множества $P_\rho(kl)$.

$\max(kl)_\rho$ - правая граница множества $P_\rho(kl)$.

Треугольник, построенный на попарно несовпадающих вершинах (объектах) $i,j,k \in V_N$ с длинами сторон, равными ρ_{ij}, ρ_{ik}, ρ_{jk} будем обозначать $\triangle(ijk)$. Когда в рассмотрение вводятся новые объекты, предполагается, что они не совпадают с уже введенными в задаче.

1. Постановка задачи. Дано конечное множество объектов V_N с заданной на нем полуметрикой ρ. Эксперт произвольно изменяет расстояние между одной парой объектов (i,j): $\rho(i,j) \mapsto \rho'(i,j) \ \Rightarrow \ R \mapsto R'$. В общем случае это влечет нарушение неравенств треугольника. Требуется предложить методы коррекции $A: \ \rho' \mapsto \widetilde{\rho}$, которые позволят построить полуметрику $\widetilde{\rho}$ и при этом будут

- универсальны (т.е. применимы для всех ρ и ρ');
- проблемно-ориентированы (т.е. будут «наиболее полно» соответствовать требованиям эксперта к вносимым в полуметрику изменениям);
- интерпретируемы;
- строить полуметрику $\widetilde{\rho}$, «максимально близкую» к исходной полуметрике ρ.

2. Интерпретация изменения расстояния ρ_{ij}. Прежде чем приступать к поиску методов коррекции полуметрики следует *формализовать понятие близости,* или *сходства,* двух полуметрик. Поскольку будем корректировать расстояние ρ', стараясь получить полуметрику $\widetilde{\rho}$, минимально отличную от исходной ρ, но в то же время учитывающую изменения эксперта, то надо оценить различие двух полуметрик в контексте проведенной экспертом модификации.

Рассмотрим два взгляда на природу изменения одного расстояния в матрице R:

1. Эксперт хотел показать, насколько расстояние между объектами i и j должно отличаться по абсолютной величине от существующего значения;
2. Эксперт хотел изменить пропорции межточечных расстояний. Изменив $r_{ij} \mapsto r'_{ij}$, он тем самым высказал требование изменить

2.a) $\dfrac{r_{ij}}{r_{ik}} \mapsto \dfrac{r'_{ij}}{r_{ik}}$, $\dfrac{r_{ij}}{r_{jk}} \mapsto \dfrac{r'_{ij}}{r_{jk}}$, $k \in V_N$;

2.b) $\dfrac{r_{ij}}{r_{kl}} \mapsto \dfrac{r'_{ij}}{r_{kl}}$, $k, l \in V_N$, $k, l \notin \{i, j\}$.

Второй подход, по-видимому, более адекватен. Действительно, расстояния между различными парами объектов едва ли можно рассматривать по отдельности, как самостоятельные величины. Гораздо информативнее соотношения этих величин. Именно они показывают, насколько объекты i и j близки, а, например, j и k далеки. Следующий пример (см. рис. 1) иллюстрирует данное утверждение. Эксперт проводит следующее преобразование: $r'_{ij} = 10$.

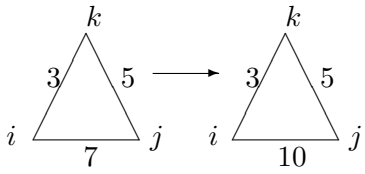


Рис. 1.

Как можно это преобразование интерпретировать? Пусть эксперт хотел просто увеличить расстояние между объектами i и j . Но на основании чего он назначил $r'_{ij} = 10$? Очевидно, что эксперт рассматривает матрицу расстояний в целом, чтобы найти типичные расстояния, характеризующие объекты как «схожие» и «различные» и затем анализирует соотношения между *парами* объектов. Т.е. его интересуют пропорции $\frac{r_{ij}}{r_{ik}}$, $\frac{r_{ik}}{r_{jk}}$ и т. д. Следовательно, изменив расстояние r_{ij} , эксперт выразил требование увеличить пропорцию $\frac{r_{ij}}{r_{jk}}$ с 1.4 до 2, и/или увеличить пропорцию $\frac{r_{ij}}{r_{ik}}$ с 2.3 до 3.3.

В дальнейшем будем использовать данный вывод и применять в большей степени пропорциональный подход при коррекции полуметрик.

При этом необходимо обратить внимание на то обстоятельство, что если ρ - полуметрика, то существует предел $\frac{r_{ij}}{r_{ik}} \xrightarrow[r_{ik} \to +0]{} +\infty$. О том, какие техниические трудности при коррекции может повлечь существование данного предела, будет сказано в следующем разделе.

3. Подходы к определению понятия сходства полуметрик. Далее при исследовании вопроса сходства полуметрик ограничимся рассмотрением функций $\rho = \rho(k, l)$ таких, что

$$\min_{(k,l) \in E_N} \rho(k, l) = \varepsilon > 0.$$

ε играет роль минимального расстояния между объектами из V_N . Очевидно, что при $\varepsilon \to +0$ множество метрик $\{\rho\}$ аппроксимирует множество полуметрик, определенное на $V_N \times V_N$.

Пусть известны матрицы метрик R и $\widetilde{R}$.

1. Минимизация суммы квадратов отклонений.

   В качестве функционала различия предлагается использовать

$$Q_u(R, \widetilde{R}) = \sum_{(kl) \in E_N} \left(\frac{\widetilde{r}_{kl} - r_{kl}}{\widetilde{r}_{kl}} \right)^2 .$$

2. Сохранение пропорций между расстояниями. Рассмотрим две метрики (см. рис. 2).

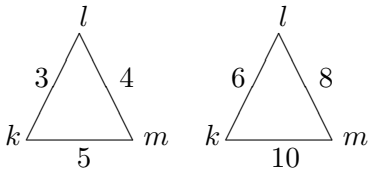


Рис. 2.

Очевидно, что, несмотря на то, что соответствующие расстояния в них неравны, их можно назвать «схожими»: отличаются они лишь масштабом. Поэтому становится разумным применение следующего функционала различия:

$$Q_p(R,\widetilde{R}) = \sum_{\triangle(klm)} \left(\frac{\widetilde{r}_{kl}}{\widetilde{r}_{km}} - \frac{r_{kl}}{r_{km}}\right)^2 + \left(\frac{\widetilde{r}_{kl}}{\widetilde{r}_{lm}} - \frac{r_{kl}}{r_{lm}}\right)^2 + \left(\frac{\widetilde{r}_{km}}{\widetilde{r}_{lm}} - \frac{r_{km}}{r_{lm}}\right)^2$$

3. Более гибкий подход к проблеме коррекции метрики должен предполагать возможность комбинировать требования сохранения пропорций между расстояниями и минимизации суммы квадратов отклонений полученных от исходных расстояний. Для этой цели предлагается использовать функционал различия

$$Q_w(R,\widetilde{R}) = w_u Q_u(R,\widetilde{R}) + w_p Q_p(R,\widetilde{R}), \text{ где } w_u, w_p \geqslant 0.$$

Комбинация функционалов может также подразумевать возможность применять один из них на одном подмножестве всех треугольников метрики, а другой - на другом подмножестве.

При использовании описанных выше функционалов может возникнуть проблема, вызванная присутствием в знаменателе дроби малой величины ε. Это приведет к тому, что некоторые слагаемые в функционале качества будут много больше всех остальных. А значит, коррекция основной массы расстояний будет мало влиять на результирующее значение $Q(R,\widetilde{R})$, которое останется неадекватно большим.

В связи с этим встает задача проблемно-ориентированного выбора величины ε. Представляется разумным назначать ε, исходя из рассмотрения *конкретной* задачи и расстояний, характерных для нее. Предлагается использовать следующий подход к определению минимального расстояния ε и преобразования полуметрики в метрику.

1. В данной матрице R найти величину $\overline{d}$ - медиану расстояний между объектами (для этого достаточно упорядочить все расстояния по возрастанию и выбрать в качестве $\overline{d}$ средний элемент полученного вариационного ряда (см. [1])) и величину

$$D = \max_{(k,l)\in E_N} r_{kl}\ .$$

$\overline{d}$ можно интерпретировать как *типичное* расстояние для данной задачи.

2. Пусть $\alpha = \dfrac{D}{\overline{d}}$. Тогда $\varepsilon = k\cdot\dfrac{\overline{d}}{\alpha} = k\cdot\dfrac{\overline{d}^2}{D}$, $k>0$. Коэффициент k предполагается использовать для более тонкой настройки величины ε. По построению при $k=1$ величина «типичного» расстояния $\overline{d}$ равноудалена от ε и D. Выбор $0<k<1$ смещает $\overline{d}$ к «схожим», а $k>1$ - к «различным» объектам.

3. $\forall(k,l)\in E_N : r_{kl} < \varepsilon \quad r_{kl} := r_{kl} + \varepsilon.$

4. Организация диалога с экспертом. Эксперт определяет значение r'_{ij}, выбирает функционал различия метрик и множества треугольников, на которых данный функционал

будет применяться (напомним, что эксперт имеет возможность требовать использовать различные функционалы на различных множествах треугольников). В соответствии с выбранным функционалом будут использованы методы коррекции.

Следующее утверждение накладывает некоторые ограничения на возможности коррекции в рамках «пропорционального» подхода.

Лемма 1. *Если в* $\triangle(ijk)$ *нарушено неравенство* $(\triangle)$, *то скорректировать это нарушение, сохранив хотя бы две пропорции между его сторонами, нельзя.*

Доказательство. Пусть $r'_{ij} > r_{ik} + r_{jk}$.

1. Надо сохранить пропорции r_{ik}/r'_{ij}, r_{jk}/r'_{ij}. Тогда

   $\dfrac{r_{ik}}{r'_{ij}} + \dfrac{r_{jk}}{r'_{ij}} = \dfrac{r_{ik} + r_{jk}}{r'_{ij}} < 1$, т.е. неравенство $(\triangle)$ останется нарушено.

2. Надо сохранить пропорции r_{ik}/r'_{ij}, r_{ik}/r_{jk}. Если известны значения r_{ik}/r'_{ij} и r_{ik}/r_{jk}, то известно и значение $\dfrac{r_{jk}}{r'_{ij}} = \dfrac{r_{jk}/r'_{ij}}{r_{ik}/r'_{ij}}$, поэтому случай 2 сводится к случаю 1.

3. Для пропорций r_{jk}/r'_{ij}, r_{ik}/r_{jk} все аналогично случаю 2. ■

5. Этапы коррекции полуметрики. Понятие величины нарушения неравенства треугольника. В дальнейшем в работе будет использована графовая интерпретация метрики, определенной на конечном множестве, как нагруженной клики на N вершинах ([2]). Множество вершин графа - это множество объектов, множество дуг - это множество попарных расстояний между объектами, а расстояния между объектами соответствуют весам, приписанным дугам. В литературе ([3] - [5]) описан метод представления метрики в виде нагруженного дерева. Однако возможность построить соответствующее дерево была доказана лишь для метрик специального вида (в частности, для ультраметрик). В общем случае вопрос аппроксимации заданной метрики с помощью дерева еще не решен.

В настоящей работе предлагается трехэтапная схема построения методов коррекции метрики. Вообще говоря, она применима для любых полуметрик, т.к. не опирается на использование свойства метрики (5). Как отмечено выше, данная схема не является единственной возможной для решения поставленной задачи синтеза полуметрики (и метрики в частности). Пусть изменено расстояние r_{ij}. В общем случае это повлечет нарушение неравенств $(\triangle)$ в $N-2$ треугольниках, а именно, во всех треугольниках вида (ijk), $k \in V_N$.

1-й этап коррекции: коррекция $\triangle(ijk)$, в которых неравенства $(\triangle)$ нарушены. Результатом 1-го этапа может быть появление нарушений неравенств $(\triangle)$ в $\triangle(ikl)$ и $\triangle(jkl)$, вызванное изменением расстояний r_{ik}, r_{jk}.

2-й этап коррекции: коррекция $\triangle(ikl)$ и $\triangle(jkl)$, в которых неравенства $(\triangle)$ нарушены. На втором этапе коррекции надо использовать такие методы, которые не вызовут нового нарушения неравенств $(\triangle)$ в $\triangle(ijk)$.

3-й этап коррекции: коррекция $\triangle(klm)$, в которых неравенство $(\triangle)$ нарушено. Требование к методам, используемым на третьем этапе такие же, как и на втором этапе.

В результате описанного процесса будет получена полуметрика $\widetilde{\rho}$.

Стоит обратить внимание на два возможных случая экспертной модификации метрики, ведущих к нарушению неравенства $(\triangle)$ в произвольном $\triangle(ijk)$:

$(*)$ расстояние r_{ij} увеличено таким образом, что $r'_{ij} > r_{ik} + r_{jk}$;

$(**)$ расстояние r_{ij} уменьшено таким образом, что либо $r_{ik} > r'_{ij} + r_{jk}$, либо $r_{jk} > r'_{ij} + r_{ik}$.

Оказывается, что каждый из этих случаев имеет свою специфику и требует применения несколько разных формул коррекции в рамках одного подхода.

Рассмотрим $\triangle(ijk)$. Введем в рассмотрение величину

$$q_{\triangle(ijk)} = \max\{(r_{ij} - r_{ik} - r_{jk}),\ (r_{ik} - r_{jk} - r_{ij}),\ (r_{jk} - r_{ik} - r_{ij})\}$$

Очевидно, что если в $\triangle(ijk)$ неравенства $(\triangle)$ выполняются, то $q_{\triangle(ijk)} \leqslant 0$.

Пусть в $\triangle(ijk)$ эксперт преобразовал $r_{ij} \mapsto r'_{ij}$ и неравенство $(\triangle)$ при этом нарушилось. При $r'_{ij} > r_{ij}$ это означает, что $q_{\triangle(ir_{jk})} = r'_{ij} - r_{ik} - r_{jk} > 0$. А если $r'_{ij} < r_{ij}$, то $q_{\triangle(ir_{jk})} = \max\{r_{ik}, r_{jk}\} - \min\{r_{ik}, r_{jk}\} - r'_{ij} > 0$.

Определение. Величину q будем называть *величиной нарушения неравенства* $(\triangle)$, или, для краткости, *нарушением* $(\triangle)$.

При коррекции этого нарушения можно использовать два подхода:

▼ корректировать расстояния r'_{ij}, r_{ik}, r_{jk} так, чтобы $q_{\triangle(ijk)} = 0$.

▽ корректировать расстояния r'_{ij}, r_{ik}, r_{jk} так, чтобы $q_{\triangle(ijk)} < 0$.

Коррекция, учитывающая условие $(\blacktriangledown)$, является в некотором смысле минимальной, т.к. в этом случае расстояния $\widetilde{r}_{ik}, \widetilde{r}_{jk}$ (при фиксированном $\widetilde{r}_{ij} = r'_{ij}$) будут изменены, чтобы удовлетворить строгому равенству в неравенстве $(\triangle)$. На меньшую величину их просто нельзя изменить. В ряде предложенных ниже методов коррекции метрик условие $(\blacktriangledown)$ используется. Однако в общем случае необходимость его использования неочевидна.

В следующих разделах формулируется ряд *алгоритмов коррекции полуметрик*. Их область применения в равной степени распространяется и на метрики, т.к. свойство «быть полуметрикой» нигде в них не используется.

6. Универсальный алгоритм коррекции полуметрик $\mathcal{A}$ при требовании эксперта $r'_{ij} = \widetilde{r}_{ij}$. Эксперт модифицировал в матрице R полуметрики ρ одно расстояние: $r_{ij} \mapsto r'_{ij}$ и дополнительно требует сохранить указанное им значение r'_{ij}. Тогда для того, чтобы скорректировать возникшие вследствие этого нарушения неравенств $(\triangle)$ в ρ', достаточно провести следующие два этапа коррекции.

1-й этап: коррекция $\triangle(ijk)$, в которых неравенства $(\triangle)$ нарушены. Какой именно метод коррекции при этом должен быть применен, определяет эксперт.

2-й этап: коррекция $\triangle(ikl)$ и $\triangle(jkl)$, в которых неравенства $(\triangle)$ нарушены.

При этом коррекция проводится по следующему правилу: $\forall(k,l) \in E_N \quad \widetilde{r}_{kl} = \min(kl)_{\widetilde{\rho}}$, либо $\forall(k,l) \in E_N \quad \widetilde{r}_{kl} = \max(kl)_{\widetilde{\rho}}$.

Прежде чем обосновать данный алгоритм, докажем два вспомогательных утверждения.

Лемма 2. *В рамках предложенного алгоритма* $\mathcal{A} \quad \forall\, \widetilde{r}_{kl} \quad P_{\widetilde{\rho}}(kl) \neq \varnothing$, *т.е.*

$$\max\{|\widetilde{r}_{ik} - \widetilde{r}_{il}|, |\widetilde{r}_{jk} - \widetilde{r}_{jl}|\} \leqslant \min\{(\widetilde{r}_{ik} + \widetilde{r}_{il}), (\widetilde{r}_{jk} + \widetilde{r}_{jl})\}\ .$$

Доказательство. Докажем от противного. При этом достаточно рассмотреть следующие два случая (в силу симметричности):

1. $|\widetilde{r}_{ik} - \widetilde{r}_{il}| > \widetilde{r}_{ik} + \widetilde{r}_{il} \ \Leftrightarrow\ (\widetilde{r}_{il} < 0) \bigvee (\widetilde{r}_{ik} < 0)$, что невозможно.
2. $|\widetilde{r}_{jk} - \widetilde{r}_{jl}| > \widetilde{r}_{ik} + \widetilde{r}_{il}$

   (a) $\begin{cases} \widetilde{r}_{jk} - \widetilde{r}_{jl} > \widetilde{r}_{ik} + \widetilde{r}_{il}, \\ \widetilde{r}_{jk} \geqslant \widetilde{r}_{jl} \end{cases} \Leftrightarrow \begin{cases} \widetilde{r}_{jk} - \widetilde{r}_{ik} > \widetilde{r}_{jl} + \widetilde{r}_{il} \geqslant r'_{ij}, \\ \widetilde{r}_{jk} \geqslant \widetilde{r}_{jl} \end{cases} \Rightarrow\ \widetilde{r}_{jk} > \widetilde{r}_{ik} + r'_{ij}\,,$

   т.е. после 1-го этапа $\triangle(ijk)$ остался нескорректированным.

$$\text{(b)}\quad \begin{cases}\widetilde{r}_{jl}-\widetilde{r}_{jk}>\widetilde{r}_{ik}+\widetilde{r}_{il},\\ \widetilde{r}_{jl}\geqslant\widetilde{r}_{jk}\end{cases}\Leftrightarrow\begin{cases}\widetilde{r}_{jl}-\widetilde{r}_{il}>\widetilde{r}_{jk}+\widetilde{r}_{ik}\geqslant r'_{ij},\\ \widetilde{r}_{jl}\geqslant\widetilde{r}_{jk}\end{cases}\quad\Rightarrow\ \widetilde{r}_{jl}>\widetilde{r}_{il}+r'_{ij}\,,$$

т.е. после 1-го этапа $\triangle(ijl)$ остался нескорректированным.

Таким образом, ни один из рассмотренных случаев не может возникнуть на 2-м этапе выполнения алгоритма $\mathcal{A}$. Лемма доказана. ■

На основании леммы 2 сформулируем следующее утверждение.

Лемма 3. *Пусть дана система объектов* $V_N=\{i,\ j,\ k_1,\ k_2,\ \ldots,\ k_{N-2}\}$. *Пусть, кроме того, введена функция* $\rho(k,l)$, *отвечающая аксиомам полуметрики одновременно на всех тройках объектов вида* $\{i,j,k_i\},\ i=1,2,\ldots,(N-2)$. *Если на множестве пар объектов* $(k_p,k_q),\ p\neq q$ *функция* $\rho(k,l)$ *строится по правилу*

$$\rho(k_p,k_q)=max\ \{|\rho(i,k_p)-\rho(i,k_q)|,|\rho(j,k_p)-\rho(j,k_q)|\},\ p,q=1,2,\ldots,(N-2),\ p\neq q \qquad (1)$$

то ρ *есть полуметрика на системе объектов* V_N.

Аналогичное утверждение справедливо и в случае

$$\rho(k_p,k_q)=min\ \{(\rho(i,k_p)+\rho(i,k_q)),(\rho(j,k_p)+\rho(j,k_q))\},\ p,q=1,2,\ldots,(N-2),\ p\neq q. \qquad (2)$$

Прежде чем привести доказательство, считаем необходимым дать следующее

Пояснение: не ограничивая общности, будем считать, что множество V_N упорядочено и $k_1<k_2<\ldots<i<j$. Формируем матрицу расстояний:

1-й шаг (дано в условии леммы 3):

$$\begin{array}{c} \\ k_1\\ k_2\\ k_3\\ \\ i\\ j\end{array}\begin{array}{c}\begin{array}{cccccc}k_1 & k_2 & k_3 & & i & j\end{array}\\ \left(\begin{array}{cccccc} 0 & & & \cdots & \bullet & \bullet\\ & 0 & & \cdots & \bullet & \bullet\\ & & 0 & \cdots & \bullet & \bullet\\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots\\ & & & \cdots & 0 & \bullet\\ & & & \cdots & & 0\end{array}\right)\end{array}$$

где элементы, обозначенные $(\bullet)$, образуют множество троек расстояний, для которых неравенства $(\triangle)$ выполняются.

2-й шаг (построение по формуле (1) или (2)):

$$\begin{array}{c} \\ k_1\\ k_2\\ k_3\\ \\ i\\ j\end{array}\begin{array}{c}\begin{array}{cccccc}k_1 & k_2 & k_3 & & i & j\end{array}\\ \left(\begin{array}{cccccc} 0 & + & - & \cdots & \oplus\ominus & \oplus\ominus\\ & 0 & \times & \cdots & \oplus\otimes & \oplus\otimes\\ & & 0 & \cdots & \ominus\otimes & \ominus\otimes\\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots\\ & & & \cdots & 0 & \bullet\\ & & & \cdots & & 0\end{array}\right)\end{array}$$

где элементы, обозначенные
$\oplus$ - это «область зависимости» значения расстояния $(+)$;
$\ominus$ - это «область зависимости» значения расстояния $(-)$;
$\otimes$ - это «область зависимости» значения расстояния $(\times)$.
В тройках вида $\{+,\oplus_{ik_1},\oplus_{ik_2}\}$, $\{+,\oplus_{jk_1},\oplus_{jk_2}\}$ неравенства $(\triangle)$ выполняются (аналогично для $-$ и $\times$). Кроме того, неравенства $(\triangle)$ выполняются в треугольниках, полученных на

шаге 1.

Тогда

$$\begin{array}{c} \\ k_1 \\ k_2 \\ k_3 \\ \\ i \\ j \end{array} \begin{array}{c} \begin{array}{cccccc} k_1 & k_2 & k_3 & & i & j \end{array} \\ \begin{pmatrix} 0 & \bullet & \bullet & \cdots & \bullet & \bullet \\ & 0 & \bullet & \cdots & \bullet & \bullet \\ & & 0 & \cdots & \bullet & \bullet \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ & & & \cdots & 0 & \bullet \\ & & & & & 0 \end{pmatrix} \end{array}$$

в $\triangle(k_1k_2k_3)$ неравенство $(\triangle)$ также верно; полученная матрица есть матрица полуметрики.

Доказательство. Для краткости записи математических выкладок выделим во множестве $\{k_1,\ k_2,\ \ldots,\ k_{N-2}\}$ некоторые три попарно несовпадающие объекта и обозначим их k, l, m. Рассмотрим матрицу R полуметрики ρ.

В треугольниках вида $\triangle(ijk)$, $\triangle(ikl)$ и $\triangle(jkl)$ неравенства $(\triangle)$ верны по построению. Остается показать, что они также верны в $\triangle(klm)$. Будем это доказывать «от противного». Для того, чтобы показать, что в любом $\triangle(klm)$ неравенства $(\triangle)$ верны, достаточно рассмотреть $3 \cdot 8$ случаев для формул (1) и (2) соответственно. Но поскольку на расстояния $r_{kl},\ r_{km},\ r_{lm}$ нет дополнительных ограничений, то все три неравенства $(\triangle)$ в $\triangle(klm)$ рассматриваются аналогично. Исследуем подробно только неравенство

$$r_{kl} \leqslant r_{km} + r_{lm}. \tag{3}$$

Для доказательства утверждения достаточно проверить, всегда ли выполняется (3).

В зависимости от значения функции max в (1) возникают следующие случаи:

1. $|r_{ik} - r_{il}| \leqslant |r_{ik} - r_{im}| + |r_{il} - r_{im}|$;
2. $|r_{ik} - r_{il}| \leqslant |r_{ik} - r_{im}| + |r_{jl} - r_{jm}|$;
3. $|r_{ik} - r_{il}| \leqslant |r_{jk} - r_{jm}| + |r_{il} - r_{im}|$;
4. $|r_{ik} - r_{il}| \leqslant |r_{jk} - r_{jm}| + |r_{jl} - r_{jm}|$;
5. $|r_{jk} - r_{jl}| \leqslant |r_{ik} - r_{im}| + |r_{il} - r_{im}|$;
6. $|r_{jk} - r_{jl}| \leqslant |r_{ik} - r_{im}| + |r_{jl} - r_{jm}|$;
7. $|r_{jk} - r_{jl}| \leqslant |r_{jk} - r_{jm}| + |r_{il} - r_{im}|$;
8. $|r_{jk} - r_{jl}| \leqslant |r_{jk} - r_{jm}| + |r_{jl} - r_{jm}|$;

Нетрудно видеть, что случаи 1 и 8, 2 и 7, 3 и 6, 4 и 5 можно рассматривать совместно, т.к. эти пары неравенств представляют собой один и тот же случай с точностью до переименования объектов i и j. Аналогично можно упростить перебор для формулы (2). Благодаря этому общее число случаев уменьшилось до 4 для формул (1) и (2) соответственно. Чтобы сократить объем выкладок, приведем по одному примеру раскрытия модулей в каждом из случаев.

1. $\begin{cases} r_{ik} - r_{il} > r_{ik} - r_{im} + r_{il} - r_{im} \ \Leftrightarrow\ r_{im} > r_{il} \\ r_{ik} \geqslant r_{il} \geqslant r_{im} \end{cases} \Rightarrow$ система несовместна;

2. $\begin{cases} r_{ik} - r_{il} > r_{ik} - r_{im} + r_{jm} - r_{jl} \ \Leftrightarrow\ |r_{jm} - r_{jl}| < |r_{im} - r_{il}| \\ r_{ik} \geqslant r_{il},\ r_{ik} \geqslant r_{im},\ r_{jm} \geqslant r_{jl} \\ |r_{jm} - r_{jl}| \geqslant |r_{im} - r_{il}| \end{cases} \Rightarrow$ система несовместна;

3. $\begin{cases} r_{ik} - r_{il} > r_{jm} - r_{jk} + r_{im} - r_{il} \;\Leftrightarrow\; r_{ik} - r_{im} > r_{jm} - r_{jk} \\ r_{ik} \geqslant r_{il},\; r_{im} \geqslant r_{il},\; r_{jm} \geqslant r_{jk} \\ |r_{jm} - r_{jk}| \geqslant |r_{im} - r_{ik}| \end{cases} \Rightarrow$ система несовместна;

4. $\begin{cases} r_{ik} - r_{il} > r_{jm} - r_{jk} + r_{jm} - r_{jl} \\ r_{ik} \geqslant r_{il},\; r_{jm} \geqslant r_{jk},\; r_{jm} \geqslant r_{jl} \\ r_{jm} - r_{jk} \geqslant |r_{ik} - r_{im}| \\ r_{jm} - r_{jl} \geqslant |r_{im} - r_{il}| \end{cases}$

Необходимо раскрыть модули в последних двух неравенствах, для чего придется рассмотреть 4 случая:

(a) $(++)$ $\begin{cases} r_{ik} \geqslant r_{im} \geqslant r_{il} \\ r_{ik} - r_{il} > 2r_{jm} - r_{jk} - r_{jl} \geqslant r_{ik} - r_{im} + r_{im} - r_{il} = r_{ik} - r_{il} \;\Leftrightarrow\; 0 > 0 \end{cases} \Rightarrow$ система несовместна;

(b) $(+-)$ $\begin{cases} r_{ik} \geqslant r_{il} \geqslant r_{im} \\ r_{ik} - r_{il} > 2r_{jm} - r_{jk} - r_{jl} \geqslant r_{ik} - 2r_{im} + r_{il} \;\Leftrightarrow\; r_{im} > r_{il} \end{cases} \Rightarrow$ система несовместна;

(c) $(--)$ $\begin{cases} r_{im} \geqslant r_{ik} \geqslant r_{il} \geqslant r_{im} \;\Rightarrow r_{im} = r_{ik} = r_{il} \\ r_{ik} - r_{il} > 2r_{jm} - r_{jk} - r_{jl} \geqslant r_{im} - r_{ik} + r_{il} - r_{im} \;\Leftrightarrow\; r_{ik} > r_{il} \end{cases} \Rightarrow$ система несовместна;

(d) $(-+)$ $\begin{cases} r_{im} \geqslant r_{ik} \geqslant r_{il} \\ r_{ik} - r_{il} > 2r_{jm} - r_{jk} - r_{jl} \geqslant 2r_{im} - r_{ik} - r_{il} \;\Leftrightarrow\; r_{ik} > r_{im} \end{cases} \Rightarrow$ система несовместна;

Таким образом, доказано, что в условиях утверждения применение формулы (1) строит полуметрику.

Для доказательства «от противного» возможности применения формулы (2) для построения полуметрики достаточно рассмотреть, как указывалось выше, 4 случая, соответствующие различным комбинациям значений функции min в левой и правой частях неравенства (3). Вот они:

1. $r_{ik} + r_{il} > r_{ik} + r_{im} + r_{il} + r_{im} \;\Rightarrow\; r_{im} < 0 \;\Rightarrow$ система несовместна;

2. $\begin{cases} r_{ik} + r_{il} > r_{ik} + r_{im} + r_{jl} + r_{jm} \\ r_{ik} + r_{il} \leqslant r_{jk} + r_{jl} \end{cases} \Rightarrow\; 0 > r_{ik} + r_{im} + r_{jm} - r_{jk} \;\Leftrightarrow$
$\Leftrightarrow\; r_{jk} > r_{ik} + r_{im} + r_{jm} \geqslant r_{ik} + ij \geqslant r_{jk} \;\Rightarrow$ система несовместна;

3. $\begin{cases} r_{ik} + r_{il} > r_{jk} + r_{jm} + r_{il} + r_{im} \\ r_{ik} + r_{il} \leqslant r_{jk} + r_{jl} \end{cases} \Rightarrow\; 0 > r_{jm} + r_{il} + r_{im} - r_{jl} \;\Leftrightarrow\; r_{jl} > r_{il} + r_{im} + r_{jm} \geqslant$
$\geqslant r_{il} + ij \geqslant r_{jl} \;\Rightarrow$ система несовместна;

4. $\begin{cases} r_{ik} + r_{il} > r_{jk} + r_{jm} + r_{jl} + r_{jm} \\ r_{ik} + r_{il} \leqslant r_{jk} + r_{jl} \end{cases} \Rightarrow\; r_{jk} + r_{jl} \geqslant r_{ik} + r_{il} > r_{jk} + r_{jm} + r_{jl} + r_{jm} \;\Leftrightarrow$
$\Leftrightarrow\; r_{jm} < 0 \;\Rightarrow$ система несовместна;

Проверка остальных случаев была выполнена автором, но в силу объемности в данной работе не приводится. Лемма доказана полностью. ■

Теорема 1. *Процедура коррекции, описанная в алгоритме* $\mathcal{A}$*, строит полуметрику* $\widetilde{\rho}$*.*

Доказательство. Условия леммы 3 конструктивно отражают схему коррекции 2-го этапа алгоритма $\mathcal{A}$. В соответствии с утверждением этой леммы, в результате использованной схемы

будет построена полуметрика ρ. Следовательно, алгоритм $\mathcal{A}$ также строит полуметрику $\widetilde{\rho}$. ■

7. Алгоритмы коррекции полуметрик при требовании эксперта $r'_{ij} = \widetilde{r}_{ij}$. В описанном универсальном алгоритме $\mathcal{A}$ не накладывается никаких ограничений на методы коррекции, используемые на 1-м этапе, что, однако, компенсируется весьма жесткими условиями, налагаемыми на процедуру коррекции на 2-м. В данном разделе предпринята попытка перераспределить мощность требований между этапами выполнения алгоритма в духе алгебраического подхода синтеза алгоритмов распознавания ([6], [7]). Постараемся смягчить условия 2-го этапа, сузив множество методов коррекции на 1-м этапе. Будем искать такие методы коррекции на 1-м этапе алгоритма $\mathcal{A}$, которые отвечали бы условию (▼). Ниже приведены некоторые методы коррекции и обозначены функционалы различия метрик, которые они минимизируют.

Значение r'_{ij} «неприкосновенно», т.е. $r'_{ij} = \widetilde{r}_{ij}$. Тогда

$$(u): \quad r'_{ij} > r_{ij} \Rightarrow \widetilde{r}_{ik} = r_{ik} + \frac{q_{\triangle(ijk)}}{2}, \quad \widetilde{r}_{jk} = r_{jk} + \frac{q_{\triangle(ijk)}}{2};$$

$$r'_{ij} < r_{ij} \Rightarrow \widetilde{r}_{ik} = r_{ik} + \frac{q_{\triangle(ijk)}}{2}, \quad \widetilde{r}_{jk} = r_{jk} - \frac{q_{\triangle(ijk)}}{2}, \text{ где } r_{ik} < r_{jk};$$

$$(p): \quad r'_{ij} > r_{ij} \Rightarrow \widetilde{r}_{ik} = r_{ik} + \frac{r_{ik}}{r_{ik} + r_{jk}}\, q_{\triangle(ijk)}, \quad \widetilde{r}_{jk} = r_{jk} + \frac{r_{jk}}{r_{ik} + r_{jk}}\, q_{\triangle(ijk)};$$

$$r'_{ij} < r_{ij} \Rightarrow \widetilde{r}_{ik} = r_{ik} - \frac{r_{ik}}{r_{jk} - r_{ik}}\, q_{\triangle(ijk)}, \quad \widetilde{r}_{jk} = r_{jk} - \frac{r_{jk}}{r_{jk} - r_{ik}}\, q_{\triangle(ijk)};$$

$$(w): \quad r'_{ij} > r_{ij} \Rightarrow \widetilde{r}_{ik} = r_{ik} + \frac{w_1}{w_1 + w_2}\, \frac{q_{\triangle(ijk)}}{2} + \frac{w_2}{w_1 + w_2}\, \frac{r_{ik}}{r_{ik} + r_{jk}}\, q_{\triangle(ijk)},$$

$$\widetilde{r}_{jk} = r_{jk} + \frac{w_1}{w_1 + w_2}\, \frac{q_{\triangle(ijk)}}{2} + \frac{w_2}{w_1 + w_2}\, \frac{r_{jk}}{r_{ik} + r_{jk}}\, q_{\triangle(ijk)};$$

$$r'_{ij} < r_{ij} \Rightarrow \widetilde{r}_{ik} = r_{ik} + \frac{w_1}{w_1 + w_2}\, \frac{q_{\triangle(ijk)}}{2} - \frac{w_2}{w_1 + w_2}\, \frac{r_{ik}}{r_{jk} - r_{ik}}\, q_{\triangle(ijk)},$$

$$\widetilde{r}_{jk} = r_{jk} - \frac{w_1}{w_1 + w_2}\, \frac{q_{\triangle(ijk)}}{2} - \frac{w_2}{w_1 + w_2}\, \frac{r_{jk}}{r_{jk} - r_{ik}}\, q_{\triangle(ijk)}, \text{ где } r_{ik} < r_{jk};$$

Алгоритм $\mathcal{A}_1$ коррекции полуметрики для $r'_{ij} > r_{ij}$:

Эксперт модифицировал в матрице полуметрики ρ одно расстояние: $r_{ij} \mapsto r'_{ij}$, причем $r'_{ij} > r_{ij}$ и дополнительно требует сохранить указанное им значение r'_{ij}. Тогда для того, чтобы скорректировать возникшие вследствие этого нарушения неравенств $(\triangle)$ в ρ', достаточно провести следующие два этапа коррекции:

1-й этап: коррекция $\triangle(ijk)$, в которых неравенства $(\triangle)$ нарушены, должна быть проведена таким образом, чтобы $\widetilde{r}_{ik} \geqslant r_{ik}$, $\widetilde{r}_{jk} \geqslant r_{jk}$.

2-й этап: коррекция $\triangle(ikl)$ и $\triangle(jkl)$, в которых неравенства $(\triangle)$ нарушены. При этом коррекция проводится по следующему правилу:

$$\widetilde{r}_{kl} = \begin{cases} r_{kl}, & \text{если } r_{kl} \in P_{\widetilde{\rho}}(kl); \\ \mathrm{P}(kl), & \text{если } r_{kl} \notin P_{\widetilde{\rho}}(kl), \end{cases}$$

где $\mathrm{P}(kl)$ - это проекция точки r_{kl} на множество

$$P_{\widetilde{\rho}}(kl) = [\max\{|\widetilde{r}_{ik} - \widetilde{r}_{il}|, |\widetilde{r}_{jk} - \widetilde{r}_{jl}|\}, \min\{(\widetilde{r}_{ik} + \widetilde{r}_{il}), (\widetilde{r}_{jk} + \widetilde{r}_{jl})\}] \ .$$

Лемма 4. *Изменение $r_{ij} \mapsto r'_{ij}$ может вызвать во всех треугольниках полуметрики ρ нарушения только одного и того же рода.*

Пояснение: лемма утверждает, что экспертное увеличение r_{ij} не может повлечь в некотором непустом подмножестве треугольников полуметрики ρ деформации вида $(*)$, а в другом подмножестве треугольников - деформации вида $(**)$.

Доказательство.

1. Пусть $r'_{ij} > r_{ij}$. Рассмотрим $\triangle(ijk)$, $\triangle(ijl)$ такие, что
$$\begin{cases} r'_{ij} > r_{ik} + r_{jk}, \\ r_{jl} > r'_{ij} + r_{il}, \\ r'_{ij} > r_{ij} \end{cases} \Rightarrow r_{jl} > r_{ij} + r_{il},$$ т.е. в исходной полуметрике ρ неравенство $(\triangle)$ нарушалось.
2. Пусть $r'_{ij} < r_{ij}$ и
$$\begin{cases} r'_{ij} > r_{ik} + r_{jk}, \\ r_{jl} > r'_{ij} + r_{il}, \\ r'_{ij} < r_{ij} \end{cases} \Rightarrow r_{ij} > r'_{ij} > r_{ik} + r_{jk},$$ тоже пришли к противоречию. ■

Следствие. *Если $r'_{ij} > r_{ij}$, то $r_{kl} \leqslant \max(kl)_{\widetilde{\rho}}$, где $\widetilde{\rho}$ - из алгоритма $\mathcal{A}_1$.*

Доказательство. По условию алгоритма $\mathcal{A}_1$

$$\begin{cases} \widetilde{r}_{ik} + \widetilde{r}_{il} \geqslant r_{ik} + r_{il}, \\ \widetilde{r}_{jk} + \widetilde{r}_{jl} \geqslant r_{jk} + r_{jl} \end{cases}$$

Далее от противного:

$$\begin{cases} \widetilde{r}_{ik} + \widetilde{r}_{il} \geqslant r_{ik} + r_{il}, \\ \max(kl)_{\widetilde{\rho}} = \widetilde{r}_{ik} + \widetilde{r}_{il}, \\ r_{kl} > \max(kl)_{\widetilde{\rho}} \end{cases} \Rightarrow r_{kl} > \widetilde{r}_{ik} + \widetilde{r}_{il} \Rightarrow r_{kl} > r_{ik} + r_{il},$$

т.е. в исходной метрике ρ неравенство $(\triangle)$ нарушалось. ■

Теорема 2. *Процедура коррекции, описанная в алгоритме $\mathcal{A}_1$, строит полуметрику $\widetilde{\rho}$.*

Доказательство. По утверждению леммы 2, $\forall (k,l) \in E_N \ P_{\widetilde{\rho}}(kl) \neq \varnothing$, следовательно, 2-й этап алгоритма $\mathcal{A}_1$ реализуем. Остается показать, что в построенных таким образом $\triangle(klm)$ неравенства $(\triangle)$ верны. Возможны 4 случая:

1. $\widetilde{r}_{kl} \neq r_{kl}$, $\widetilde{r}_{lm} \neq r_{lm}$, $\widetilde{r}_{km} \neq r_{km} \Rightarrow$ по следствию из леммы 4, $\widetilde{r}_{kl} = \min(kl)_{\widetilde{\rho}}$, $\widetilde{r}_{lm} = \min(lm)_{\widetilde{\rho}}$, $\widetilde{r}_{km} = \min(km)_{\widetilde{\rho}}$, а для такой тройки расстояний неравенство $(\triangle)$ верно по лемме 3.

2. $\widetilde{r}_{kl} \neq r_{kl}$, $\widetilde{r}_{lm} \neq r_{lm}$, $\widetilde{r}_{km} = r_{km}$.

   Пусть при этом $r_{km} = \widetilde{r}_{km} > \widetilde{r}_{kl} + \widetilde{r}_{lm} > r_{kl} + r_{lm} \ \Leftrightarrow r_{km} > r_{kl} + r_{lm}$, что невозможно, ибо ρ - полуметрика.

   Пусть $\widetilde{r}_{kl} > \widetilde{r}_{km} + \widetilde{r}_{lm} \ \Leftrightarrow \min(kl)_{\widetilde{\rho}} > \min(lm)_{\widetilde{\rho}} + \min(km)_{\widetilde{\rho}}$, что противоречит лемме 3. Случай $\widetilde{r}_{lm} > \widetilde{r}_{km} + \widetilde{r}_{kl}$ рассматривается аналогично.

3. $\widetilde{r}_{kl} \neq r_{kl}$, $\widetilde{r}_{lm} = r_{lm}$, $\widetilde{r}_{km} = r_{km}$. Все аналогично случаю 2.

4. $\widetilde{r}_{kl} = r_{kl}$, $\widetilde{r}_{lm} = r_{lm}$, $\widetilde{r}_{km} = r_{km}$. Очевидно, что в этом случае неравенства $(\triangle)$ выполнены автоматически, т.к. исходная функция ρ есть полуметрика. ■

Теорема 3. *Пусть* $r'_{ij} > r_{ij}$. *Тогда применение формул (u) на 1-м этапе коррекции посредством алгоритма* $\mathcal{A}_1$ *уже в ходе 1-го этапа строит полуметрику* $\widetilde{\rho}$.

Доказательство. Достаточно показать, что после проведения 1-го этапа коррекции в $\triangle(ikm)$, $\triangle(jkm)$ неравенство $(\triangle)$ не будет нарушено. Проверять $\triangle(klm)$ нет смысла, т.к. они не корректировались вовсе.

1. Пусть $\triangle(ijk)$ и $\triangle(ijm)$ были скорректированы. Тогда

$$\widetilde{r}_{ik} = r_{ik} + \frac{r'_{ij} - r_{ik} - r_{jk}}{2}\,, \quad \widetilde{r}_{jk} = r_{jk} + \frac{r'_{ij} - r_{ik} - r_{jk}}{2}$$

$$\widetilde{r}_{im} = r_{im} + \frac{r'_{ij} - r_{im} - r_{jm}}{2}\,, \quad \widetilde{r}_{jm} = r_{jm} + \frac{r'_{ij} - r_{im} - r_{jm}}{2}$$

Пусть в $\triangle(ijk)$ $\widetilde{r}_{ik} > \widetilde{r}_{im} + r_{km}$. Поскольку $r'_{ij} = \widetilde{r}_{ik} + \widetilde{r}_{jk} = \widetilde{r}_{im} + \widetilde{r}_{jm}$, то $\widetilde{r}_{ik} - \widetilde{r}_{im} = \widetilde{r}_{jm} - \widetilde{r}_{jk} \;\Rightarrow \widetilde{r}_{jm} > \widetilde{r}_{jk} + r_{km}$. Складывая эти неравенства, получим

$$\widetilde{r}_{ik} + \widetilde{r}_{jm} > \widetilde{r}_{im} + \widetilde{r}_{jk} + 2r_{km} \;\Leftrightarrow$$
$$\Leftrightarrow\; r_{ik} + r_{jm} > r_{im} + r_{jk} + 2r_{km} \;\Leftrightarrow\; r_{ik} + r_{jm} > (r_{im} + r_{km}) + (r_{jk} + r_{km}) > r_{ik} + r_{jm}$$

пришли к противоречию. Значит, в $\triangle(ikm)$ нарушений $(\triangle)$ быть не может. Случай с $\triangle(jkm)$ рассмотрен по ходу доказательства.

2. Пусть $\triangle(ijk)$ был скорректирован, а $\triangle(ijm)$ - нет. Тогда по лемме 4 нарушение неравенств $(\triangle)$ в $\triangle(ikm)$ и $\triangle(jkm)$ может иметь только вид $\widetilde{r}_{ik} > r_{im}+r_{km}$, $\widetilde{r}_{jk} > r_{jm}+r_{km}$ соответственно. $r_{ik} + r_{jk} < r'_{ij} \leqslant r_{im} + r_{jm}$ - по условию. Рассмотрим подробнее случай с $\triangle(ikm)$:

$$\begin{cases} \widetilde{r}_{ik} > r_{im} + r_{km}, \\ \widetilde{r}_{ik} + \widetilde{r}_{jk} = r'_{ij} \leqslant r_{im} + r_{jm} \end{cases} \Leftrightarrow \begin{cases} \widetilde{r}_{ik} > r_{im} + r_{km}, \\ \widetilde{r}_{ik} + \widetilde{r}_{jk} \leqslant r_{im} + r_{jm} \end{cases} \Rightarrow\; -\widetilde{r}_{jk} > -r_{jm} + r_{km} \;\Rightarrow$$
$$\Rightarrow\; r_{jm} > \widetilde{r}_{jk} + r_{km} > r_{jk} + r_{km},$$

что означает нарушение неравенства $(\triangle)$ в $\triangle(jkm)$ в исходной метрике ρ , чего не может быть. Аналогично рассматривается случай с $\widetilde{r}_{jk} > r_{jm} + r_{km}$. Следовательно, $\triangle(ikm)$ и $\triangle(jkm)$ не могут иметь нарушений $(\triangle)$. Утверждение доказано. ■

Следствие. Метод (u) имеет сложность $2N$, т.е. является *линейным* относительно числа объектов.

Рассмотрение различных полуметрик показывает, что методы (p) и (w) имеют квадратичную сложность $O(N^2)$ и, таким образом, работают быстрее алгоритмов, в которых предусмотрена проверка *всех* треугольников (т.е., по нашей терминологии, работающие в 3 этапа и имеющие, таким образом, сложность $O(N^3)$).

Исследование показало, что при $r'_{ij} < r_{ij}$ алгоритм, подобный $\mathcal{A}_1$, не строит полуметрику.

Исходя из утверждения теоремы 1, гарантировано существуют две полуметрики, построенные в соответствии с алгоритмом $\mathcal{A}$: в одной из них $\forall (k,l) \in E_N$, $\widetilde{r}_{kl} = \min(kl)_{\widetilde{\rho}}$, а в другой $\forall (k,l) \in E_N$, $\widetilde{r}_{kl} = \max(kl)_{\widetilde{\rho}}$. Возникает предположение: если $\min(kl)_{\widetilde{\rho}} < \max(kl)_{\widetilde{\rho}}$ и в ходе коррекции положить $\widetilde{r}_{kl} = \min(kl)_{\widetilde{\rho}} + \varepsilon$, $\varepsilon > 0$, а затем «скомпенсировать» это ε -отклонение, соответствующим образом изменив $\widetilde{r}_{km}$ и $\widetilde{r}_{lm}$, то будет построена полуметрика $\widetilde{\rho}$. При этом за счет настройки величин ε для каждого корректируемого расстояния можно достичь лучшего соответствия метрики $\widetilde{\rho}$ требованиям эксперта, чем в случае использования универсального алгоритма $\mathcal{A}$.

Таким образом, приходим к оптимизационному алгоритму коррекции $\mathcal{A}_{op}$, который можно использовать как в случае $r'_{ij} < r_{ij}$, так и в случае $r'_{ij} > r_{ij}$.

Оптимизационный алгоритм коррекции $\mathcal{A}_{op}$ **:**

Эксперт модифицировал в матрице R полуметрики ρ одно расстояние: $r_{ij} \mapsto r'_{ij}$ и дополнительно требует сохранить указанное им значение r'_{ij}. Тогда для того, чтобы скорректировать возникшие вследствие этого нарушения неравенств $(\triangle)$ в ρ', достаточно провести следующие три этапа коррекции.

1-й этап: коррекция $\triangle(ijk)$, в которых неравенства $(\triangle)$ нарушены. Какой именно метод коррекции при этом должен быть применен, определяет эксперт.

2-й этап: коррекция $\triangle(ikl)$ и $\triangle(jkl)$, в которых неравенства $(\triangle)$ нарушены.
При этом коррекция проводится по следующему правилу:

$$\widetilde{r}_{kl} = \begin{cases} r_{kl}, & \text{если } r_{kl} \in P_{\widetilde{\rho}}(kl); \\ \mathrm{P}(kl), & \text{если } r_{kl} \notin P_{\widetilde{\rho}}(kl), \end{cases}$$

где $\mathrm{P}(kl)$ - это проекция точки ρ_{kl} на множество

$$P_{\widetilde{\rho}}(kl) = [\max\{|\widetilde{r}_{ik} - \widetilde{r}_{il}|, |\widetilde{r}_{jk} - \widetilde{r}_{jl}|\}, \min\{(\widetilde{r}_{ik} + \widetilde{r}_{il}), (\widetilde{r}_{jk} + \widetilde{r}_{jl})\}] \ .$$

3-й этап: провести оптимизацию

$$\begin{cases} Q(R, \widetilde{R}) \to \min_{\widetilde{R}}, \\ r_{kl} - r_{km} - r_{lm} \geqslant 0, \ \text{где } r_{kl} = \max\{r_{kl}, r_{km}, r_{lm}\}, \ \forall \triangle(klm). \end{cases}$$

На каждой итерации необходимо проверять выполнение неравенств $(\triangle)$ в треугольниках вида $\triangle(ikl)$, $\triangle(jkl)$ и $\triangle(klm)$.

Алгоритм $\mathcal{A}_{op}$ даст на выходе полуметрику (по построению).

8. Использование процедуры обучения для настройки весов в методе коррекции (w). Пусть эксперт показал на некотором множестве $\{\triangle(ijl)\}$, какие значения $\widetilde{r}_{il}, \widetilde{r}_{jl}$ он хочет получить в результате коррекции, т.е. провел вручную коррекцию этих треугольников. Будем использовать значения $\widetilde{r}_{il}, \widetilde{r}_{jl}$ как прецеденты ([8]), настраивая по ним веса w_1 и w_2, и затем применим полученный взвешенный метод коррекции (формулы (w)) ко *всему* множеству $\{\triangle(ijk)\}$, $\forall\ k \in V_N$.

Процедура настройки весов:

1. Инициализация весов $x_0 := \dfrac{1}{2}$, $y_0 := \dfrac{1}{2}$;

2. Вычисление функционала качества $Q(x, y) = \sum\limits_{\{\triangle(ijl)\}} \left[(\widetilde{r}_{il} - r'_{il})^2 + (\widetilde{r}_{jl} - r'_{jl})^2\right]$,
   где r'_{il}, r'_{jl} заданы экспертом, а $\widetilde{r}_{il}, \widetilde{r}_{jl}$ вычисляются по формулам, используемым в данном методе коррекции, причем $\frac{w_1}{w_1+w_2} = x_i$, $\frac{w_2}{w_1+w_2} = y_i$.

3. Вычисление очередных x_{i+1}, y_{i+1} методом градиентного спуска

$$\begin{cases} x_{i+1} = x_i - \dfrac{1}{\mu} \cdot \dfrac{\partial Q(x, y)}{\partial x}, \quad \mu = \dfrac{1}{i} \\ y_{i+1} = 1 - x_{i+1} \end{cases}$$

4. Если x_{i+1}, y_{i+1} отвечают критерию останова, то ВЫХОД, иначе ПЕРЕХОД к п.2.

В качестве критерия останова можно, например, использовать условие $|Q(x_{i+1}, y_{i+1}) - Q(x_i, y_i)| < \delta$, где $\delta > 0$. В результате итерационного процесса получим $x_n \xrightarrow[n\to\infty]{} \dfrac{w_1}{w_1 + w_2}$,

$y_n \xrightarrow[n\to\infty]{} \dfrac{w_2}{w_1+w_2}$. Можно получить явное выражение для w_1 и w_2: $w_1 = 1,\ w_2 = y_n/x_n$ (но для коррекции метрики это не требуется).

В методе градиентного спуска частные производные имеют следующий вид: для $r'_{ij} > r_{ij}$

$$\frac{\partial Q(x,y)}{\partial x} = \sum_{\{\triangle(ijl)\}} \left[(\widetilde{r}_{il} - r'_{il}) + (\widetilde{r}_{jl} - r'_{jl})\right] q_{\triangle(ijl)} \ ,$$

$$\frac{\partial Q(x,y)}{\partial y} = 2 \sum_{\{\triangle(ijl)\}} \frac{(\widetilde{r}_{il} - r'_{il})\widetilde{r}_{il} + (\widetilde{r}_{jl} - r'_{jl})\widetilde{r}_{jl}}{\widetilde{r}_{il} + \widetilde{r}_{jl}} \ .$$

Для $r'_{ij} < r_{ij}$ эти формулы выписываются похожим образом.

Заключение. В работе рассмотрена задача коррекции полуметрики при заданном экспертом значении $\rho(i,j)$, которое требуется сохранить; введен ряд функционалов различия метрик; описана трехэтапная схема построения методов коррекции полуметрик. Предложен ряд алгоритмов, не требующих рассмотрения в процессе коррекции всех троек объектов и имеющих квадратичную, а в специальном случае - линейную по количеству объектов, сложность. Доказана корректность данных алгоритмов. Предложен алгоритм коррекции, предусматривающий возможность применения процедуры обучения с целью улучшения качества коррекции.

СПИСОК ЛИТЕРАТУРЫ

1. *Королев В. Ю.* Вероятностные модели: введение в асимптотическую теорию случайного суммирования. // М.: Диалог-МГУ. 1997.
2. *Зыков А. А.* Теория конечных графов. // Новосибирск: Наука. 1969.
3. *Миркин Б. Г., Родин С. Н.* Графы и гены. // М.: Наука. 1977.
4. *Юшманов С. В.* Методы графов в эволюции. // Математическая кибернетика и ее приложения к биологии. - М.: Издательство Московского Университета. 1987.
5. *Юшманов С. В.* Восстановление филогенетического древа по поддеревьям, порожденным четверками его висячих вершин. // Математическая кибернетика и ее приложения к биологии. - М.: Издательство Московского Университета. 1987.
6. *Рудаков К. В.* Об алгебраической теории универсальных и локальных ограничений для задач классификации // Распознавание, классификация, прогноз. М.: Наука. 1989. С. 176-201.
7. *Журавлев Ю. И., Рудаков К. В.* Об алгебраической коррекции процедур обработки (преобразования) информации // Проблемы прикладной математики и информатики. М.: Наука. 1987. С.187-198.
8. *Дуда Р., Харт П.* Распознавание образов и анализ сцен. // М.: Мир. 1976.

УДК 517.956.2

ОЦЕНКА ПАРАМЕТРОВ РАЗРЯДА ПЛАЗМЫ МЕТОДОМ ОПОРНЫХ ВЕКТОРОВ

© 2007 г. Ф. М. Жданов

zhdanov_f_m@mail.ru

Кафедра Автоматизации научных исследований

Введение. Настоящая работа посвящена применению нового подхода для оценки параметров удержания термоядерной плазмы в установках токамак. Для проведения разряда плазмы важно определить, в какой моде окажется разряд - в H-mode или L-mode. В настоящей работе для решения этих задач предлагается использовать современный статистический подход теории data mining - метод опорных векторов.

С тех пор как была предложена первая схема магнитного термоядерного реактора прошло более 50 лет. За это время было проведено множество экспериментов по удержанию плазмы в токамаках, и накопилось большое количество данных об осуществленных разрядах. В настоящей работе предлагается использовать эти данные для определения одной из наиболее важных характеристик разряда: его моды. Для решения поставленной задачи предлагается использовать современный статистический подход теории data mining - метод опорных векторов. Понятие data mining буквально означает “извлечение знаний из данных”, и может быть также определено как "процесс поддержки принятия решений, основанный на поиске в данных скрытых закономерностей". В качестве исходной информации для наших исследований использовались базы данных, в которых собраны параметры разрядов, осуществленных на различных установках; например, общедоступная база PID[1].

Целью данной статьи является построение классификатора, который должен определять L-или H-моду разряда плазмы в токамаке по вектору входных значений его параметров. Одним из наиболее важных применений этого классификатора является предсказание моды работы ITER – проектирумого международного токамака-реактора. Для решения этой задачи мы предлагаем использовать линейный классификатор, так как он обладает наилучшей способностью к экстраполяции. Значения параметров ITER далеки от значений параметров других токамаков, поэтому сохранение способности классификатора к экстраполяции является важным для прогнозирования.

Метод опорных векторов (Support Vector Machine) [2] является в настоящее время одним из наиболее мощных методов построения линейного классификатора. Используя этот метод, нам удалось достичь значения точности классификации 92,66%, используя восемь наиболее важных параметров разряда. В настоящей работе было проведено сравнение с методом Линейного Дискриминантного анализа Фишера (ЛДА)[3], используемого ранее для классификации разрядов на L-H моды, и показано, что при предлагаемом нами подходе удается достичь существенно лучшей точности классификации.

В первой части статьи приведены основы метода опорных векторов, необходимые для построения линейного классификатора. Во второй части описывается численный эксперимент, а также анализируются полученные результаты. В частности показано, что токамак-реактор ITER при индуктивном сценарии разряда будет работать в H-моде.

1. Основы метода опорных векторов. Основной идеей метода опорных векторов является то, что он находит оптимальное линейное разделение двух множеств гиперплоскостью таким образом, чтобы расстояние между этими множествами и гиперплоскостью было максимальным [2,4]. Представим, что имеется так называемое обучающее множество векторов, состоящее из N точек данных: $\{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \ldots, (\vec{x}_N, y_N)\}$, где $\vec{x}_i \in R^n$ это известные параметры, а $y_i \in \{-1, +1\}$ - известные значения, определяющие принадлежность вектора

тому или иному множеству. Допустим, множества линейно разделимы. Целью ставится поиск линейной разделяющей гиперплоскости H, в дальнейшем называемой классификатором, такой чтобы

$$f(\vec{x}) = sgn((\vec{w} \cdot \vec{x}) - b), \vec{w} \in R^n, b \in R, \tag{1}$$

где $\vec{w}$ и b - параметры классификатора, подлежащие определению.

Предположим, нами уже построена оптимальная разделяющая гиперплоскость H. Рассмотрим две гиперплоскости H_1 и H_2, параллельные H

$$\begin{array}{l} H_1 : y = (\vec{w} \cdot \vec{x}) - b = +1, \\ H_2 : y = (\vec{w} \cdot \vec{x}) - b = -1, \end{array} \tag{2}$$

, проходящие через точки разделяемых множеств и такие, что между ними отсутствуют точки разделяемых множеств. H_1 содержит как минимум одну точку с $y = +1$, а H_2 - как минимум одну точку с $y = -1$. Расстояние между множествами определяется как расстояние между этими гиперплоскостями, а оптимальная гиперплоскость H является равноудаленной от H_1 и H_2 - см. рисунок 1.

Рисунок 1. Жирная прямая соответствует гиперплоскости H (классификатору), построенному так, чтобы максимизировать расстояние между множествами закрашенных и незакрашенных кружков. По осям X_1 и X_2 отложены координаты точек разделяемых множеств.

Точки, через которые проходят гиперплоскости H_1 и H_2, называются опорными векторами. Метод опорных векторов находит их линейную комбинацию, с помощью которой строится разделяющая гиперплоскость. Это точки x_1, x_2, x_3 на рисунке 1. Расстояние от H_1 до параллельной гиперплоскости $H : (\vec{w} \cdot \vec{x}) - b = 0$, равно

$$\frac{|(\vec{w} \cdot \vec{x}) - b|}{\|w\|} = \frac{1}{\|w\|}. \tag{3}$$

Расстояние между H_1 и H_2 равно $\frac{2}{\|w\|}$. Чтобы максимизировать расстояние от H_1 до H_2, минимизируется $\|w\|$ с условиями, что между H_1 и H_2 нет точек данных:

$$\begin{array}{l} (\vec{w} \cdot \vec{x}) - b \geq +1, y = +1, \\ (\vec{w} \cdot \vec{x}) - b \leq -1, y = -1. \end{array} \tag{4}$$

Неравенства (4) можно записать в виде:

$$y_i\left((\vec{w}\cdot\vec{x})-b\right)\geq 1 \tag{5}$$

и решать задачу

$$\min_{\vec{w},b}\frac{1}{2}\vec{w}^T\vec{w} \tag{6}$$

с линейными ограничениями (5). Таким образом, задача свелась к хорошо известной задаче квадратичного программирования, которая решается минимизацией функционала Лагранжа

$$L\left(\vec{w},b;\,\vec{\alpha}\right)=\frac{1}{2}\left\|w\right\|^2-\sum_{i=1}^{N}\alpha_i\left(y_i\left((\vec{w}\cdot\vec{x})-b\right)-1\right)\rightarrow\min \tag{7}$$

, где $\alpha_i\geq 0$ - множители Лагранжа. Так как Лагранжиан является выпуклой функцией, и минимум ищется по выпуклому множеству, можно свести задачу к эквивалентной двойственной задаче: максимизации L по α, исходя из того, что производные L по $\vec{w}$ и b равны нулю [2]:

$$\frac{\partial L}{\partial\bar{w}}=0,\frac{\partial L}{\partial b}=0.$$

откуда следует, что $\vec{w}=\sum\limits_{i=1}^{N}\alpha_i y_i\vec{x}_i$ и $\sum\limits_{i=1}^{N}\alpha_i y_i=0$.

Двойственная задача формулируется следующим образом: максимизировать

$$L_D(\vec{\alpha})=\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j(\vec{x}_i,\vec{x}_j) \tag{8}$$

с ограничениями

$$\alpha_i\geq 0,\sum_{i=1}^{N}\alpha_i y_i=0, i=1,\ldots,N \tag{9}$$

Значение b определяется из условия Куна-Таккера[2,5]:

$$\alpha_i\left[y_i\left((\vec{w}\cdot\vec{x}_i)-b\right)-1\right]=0, i=1,\ldots,N \tag{10}$$

Решения не существует для линейно неразделимых данных, так как в этом случае не существует гиперплоскостей H_1 и H_2. Однако можно ослабить ограничения (5) введя штраф C. Задача переформулируется так:

$$\max_{\vec{\alpha}}L_D(\vec{\alpha})=\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j(\vec{x}_i,\vec{x}_j) \tag{11}$$

с условиями

$$0\leq\alpha_i\leq C,\sum_{i=1}^{N}\alpha_i y_i=0, i=1,\ldots,N \tag{12}$$

Эта задача отличается от (8),(9) тем, что имеется ограничение сверху на α_i (12). Мы выбирали C таким образом, чтобы минимизировать ошибку классификации. Весовые коэффициенты $\vec{w}$ разделяющей гиперплоскости могут быть определены из формулы

$$\vec{w}=\sum_{i=1}^{N}\alpha_i y_i\vec{x}_i \tag{13}$$

2. Численный эксперимент. Для проведения численного эксперимента нами были использованы данные из общедоступной базы PID [1], содержащей результаты экспериментов на различных токамаках за много лет. Все 9706 разрядов, представленные в этой базе данных, были разделены на два класса, соответствующие H-моде и L-моде. Разряды с модами H, HGELM, HSELM, HGELMH, HSELMH, LHLHL были отнесены к множеству H-моды, а разряды с модами L, OHM, RI - к множеству L-моды. Рисунок 2 представляет количество разрядов каждого токамака в базе. Можно увидеть, что наибольшее число разрядов соответствуют токамаку JET, самому большому токамаку в настоящее время.

Следуя [6], мы выбрали 8 наиболее важных параметров из всей совокупности параметров разрядов. Это

I - текущий в плазме ток (измеряется в мега-амперах, MA);

B_T - вакуумное тороидальное магнитное поле (измеряется в тесла, T);

P_{loss} - оцененные потери мощности (измеряется в мегаваттах, MW);

n_e - средняя плотность электронов плазмы на центральной линии (измеряется в шт.*10^{19} м $^{-3}$) ;

R - большой радиус плазмы (измеряется в метрах, m);

a - горизонтальный малый радиус плазмы измеряется в метрах, m);

K_a - эллиптичность (растяжение) (безразмерная величина);

M - эффективная атомная масса (измеряется в AMU).

Мы использовали множество пар $(\vec{x}, y)$ в качестве тренировочного набора для SVM, где $\vec{x} = (I, B_T, P_{loss}, n_e, R, a, K_a, M)$, $y = \{+1, -1\}$, здесь $+1$ соответствует H-моде, –1 - L-моде. Восемь параметров были нормализованы к отрезку [-1,1]. Размер обучающего множества достаточно велик: из 9706 векторов 5837 лежат в H-моде, 3869 – в L-моде.

Рисунок 2. Число разрядов соответствующих каждому токамаку в обучающем множестве.

Обучение классификатора проводилось методом последовательной оптимизации **S**equential **M**inimal **O**ptimization (SMO), основанной на сведении исходной задачи к последовательности задач максимизации функции двух переменных [5,7]. SMO является одним из наиболее эффективных методов обучения.

Методом опорных векторов была получена следующая формула для разделяющей гиперплоскости между множествами разрядов H и L-мод:

$$\begin{aligned} &1.6142 \cdot I - 0.0925 \cdot B_T + 1.7215 \cdot P_{loss} + 0.1347 \cdot n_e - \\ &-2.04 \cdot R + 2.24066 \cdot a + 11.18 \cdot K_a - 11.5165 \cdot M - 6.8086 = 0 \end{aligned} \tag{14}$$

Равенство (14) содержит ненормализованные параметры, измеряемые в указанных выше величинах. Эта гиперплоскость разделяет множества с точностью 92.66%. Существует множество гиперплоскостей со значениями коэффициентов, близкими к значениям в (14), в за-

висимости от принятого значения штрафа C , и точности выполнения условий (10). Из всех допустимых гиперплоскостей мы выбрали ту, которая показывает наибольшую точность как на обучающем, так и на тестовом множествах.

Рисунок 3 представляет число неправильно классифицированных разрядов для каждого токамака (слева), и процентное содержание таких разрядов в общем числе разрядов выбранного токамака (справа).

Рисунок 3. Распределение неправильно классифицированных разрядов по токамакам: число разрядов для каждого токамака (слева), и процентное содержание таких разрядов в общем числе разрядов выбранного токамака (справа).

Из рисунка 3 можно увидеть, что процент ошибок классификации на самом большом на сегодняшний день токамаке JET всего лишь 5.11%, тогда как ошибка для PDX - около 70%. Это позволяет сделать вывод, что построенный классификатор хорошо работает для больших токамаков.

Расстояние до разделяющей гиперплоскости показывает, насколько "глубоко"разряды находятся в H- или L- моде, т.е. на сколько стабильно токамак работал в этом режиме. Расстояния между разделяющей гиперплоскостью и вектором в 8-мерном пространстве, показаны на рисунке 4. Здесь расстояние определяется по отношению к выбранной стороне гиперплоскости, и может быть отрицательным:

$$\rho = \frac{w_1 x_1 + w_2 x_2 + ... + w_n x_n - b}{\sqrt{w_1^2 + w_2^2 + ... + w_n^2}}$$

Расстояния считались для векторов, значения координат которых были предварительно приведены к отрезку [-1,1], с помощью максимального и минимального значений по каждой компоненте

$$u_i^{norm} = \frac{2 \cdot u_i - (\min\limits_{l=1..N} x_i^l + \max\limits_{l=1..N} x_i^l)}{\max\limits_{l=1..N} x_i^l - \min\limits_{l=1..N} x_i^l}$$

Здесь min и max берутся по всему обучающему набору. Коэффициенты нормализации компонент представлены в таблице 1.

Рисунок 4. Расстояние до разделяющей гиперплоскости от разрядов токамака. Точки на горизонтальной оси соответствуют различным разрядам. Вертикальная ось показывает расстояние. Отрицательная полуплоскость соответствует разрядам в L-моде, положительная – в H-моде, темные вертикальные черточки – неправильно классифицированным векторам.

Установки на рисунке 4 упорядочены по убыванию относительной ошибки классификации, показанной на рисунке 3 (правом). PDX имеет самый худший процент правильно классифицированных векторов, а PBXM – самый лучший. Разряды для каждого токамака упорядочены по возрастанию расстояния до гиперплоскости. Вертикальная ось представляет Евклидово расстояние в 8-мерном пространстве от вектора до гиперплоскости. Кажется, что кривые на рисунке непрерывны, но этот эффект происходит из-за того, что количество разрядов очень велико.

Положение темных точек позволяет понять структуру множеств L и H мод. Оставаясь достаточно хорошо разделимыми, в них есть неправильно классифицированные точки, не только близко лежащие к гиперплоскости, но и точки, достаточно далеко от нее отдаленные.

Построенный классификатор применялся нами к для предсказания режима работы ITER. Рассматривались параметры индуктивного сценария работы ITER с 400MW мощности [6,8]

$$I = 15MA, B_T = 5.3T, P_{loss} = 87MW, n_e = 10.1e + 19m^{-3}, R = 6.2m, a = 2m, K_a = 1.7, M = 2.5 \tag{15}$$

Расстояние от разделяющей гиперплоскости для этих параметров после нормализации составило $\rho = 6.125$. Это означает, что метод SVM предсказывает устойчивую работу ITER в H-моде, см. рисунок 4.

В [9] предложено использовать Линейный Дискриминантный Анализ чтобы предсказать вероятность попадания в H- или L-моду. Мы реализовали этот метод и сравнили его с SVM.

Линейный Дискриминатный Анализ (ЛДА), или Дискриминантный анализ Фишера, представляет решение проблемы классификации для случая двух классов[3]. Идея метода заклю-

чается в том, что векторы обучающего множества проектируются на некоторую прямую, на которой и проводится классификация (это эквивалентно поиску линейной комбинации их компонентов). Прямая выбирается так, чтобы максимизировать расстояние между классами, и минимизировать дисперсию в каждом классе. Для случая двух классов максимизируется отношение $\frac{|P_1^0 - P_2^0|}{\sigma_1 + \sigma_2}$, где P_i^0 - проекция центра i-го класса (средний вектор) на прямую, а σ_i - разброс проекции векторов i-го класса.

Точность, полученная методом ЛДА, равна 85.6%. Это примерно на 7% хуже, чем точность SVM. Достоинством же этого метода является быстрая по сравнению с SVM процедура обучения. Но по точности классификации SVM превосходит ЛДА.

Заключение В настоящей работе была исследована задача определения моды разряда плазмы в установках токамак по его параметрам. Для решения поставленной задачи использовался метод опорных векторов. С его помощью была достигнута точность классификации 92,66% при обработке около 10 тысяч разрядов, что позволяет сделать заключение о высокой эффективности данного метода. Одним из наиболее важных результатов, полученных в настоящей работе, является предсказание моды работы токамака-реактора ITER - построенный классификатор показывает, что ITER при индуктивном сценарии разряда будет функционировать в H-моде.

Благодарности Автор выражает благодарность научному руководителю А.А. Лукьянице, Ф.С. Зайцеву, G. Maddison и другим за полезные замечания и помощь в проведении исследования.

СПИСОК ЛИТЕРАТУРЫ

1. The International Global H-mode Confinement Database // http://efdasql.ipp.mpg.de/HmodePublic
2. *Vapnik V.N.* An Overview of Statistical Learning Theory. // IEEE transactions on neural networks 10 5 pp 988-999 September. 1999
3. *Balakrishnama S. and Ganapathiraju A.* Linear Discriminant Analysis - a brief tutorial.// Institute for Signal and Information Processing Mississippi. 1998
4. *Merkov A.B.* О статистической тоерии обучения. // http://www.recognition.mccme.ru/pub/RecognitionLab.html/slt.html. 2006
5. *Platt J.C.* Sequential Minimal Optimization for SVM. // http://citeseer.ist.psu.edu/. 1998
6. *Cordey J.G., Kardaun O., Thomsen K. et al* Recent developments in the staistical analysis of the Tokamak confinement data and the implications for ITER and Power Plants. // Fusion Theory Colloquium 2004
7. *Platt J.C.* Fast Training of Support Vector Machines using Sequential Minimal Optimization. // Advances in Kernel Methods - Support Vector Learning. MIT Press. pp 41-65. 1998
8. ITER Technical Basis. // Plant description document. Chapter 4 pp 1-39. http://www.iter.org/. 2000
9. *Martin Y.* Prediction of the ITER H-mode power threshold by means of various statistical techniques. // ICPP&25th EPS Conf. on Contr. Fusion and Plasma Physics Praha. 1998

УДК 517.977.58

ЧИСЛЕННОЕ РЕШЕНИЕ П-СИСТЕМ ДЛЯ ЗАДАЧ ОПТИМАЛЬНОГО УПРАВЛЕНИЯ С ФАЗОВЫМИ И СМЕШАННЫМИ ОГРАНИЧЕНИЯМИ

© 2007 г. С. С. Жулин

zstep@yandex.ru

Кафедра Оптимального управления

Введение. Настоящая статья является продолжением работы [8], здесь решаются проблемы, связанные с применением предложенного метода к классу задач оптимального управления с фазовыми и смешанными ограничениями. Предлагается метод нахождения управления, максимизирующего функцию Гамильтона-Понтрягина, посредством численного решения нелинейного уравнения. Приводятся выкладки нахождения производных максимизатора и множителей Лагранжа для построения системы уравнений первой вариации по начальным данным. Это позволяет применить метод продолжения по параметру к поиску экстремали Понтрягина в указанном классе задач.

Многие задачи оптимального управления помимо ограничений на управление содержат ограничения на фазовую переменную, а также смешанные ограничения. Эти ограничения обычно отражают физические рамки, в которых система функционирует по заданному дифференциальному закону.

Приведем пример. Рассмотрим задачу о тележке, в ней динамика задается следующей системой

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = u \end{cases}.$$

В классической задаче ставятся ограничения на управление

$$a \le u \le b.$$

Если рассматривать физический объект, описываемый системой, эти неравенства ограничивают силу тяги двигателя тележки. Но в реальных условиях также не может быть бесконечной и мощность двигателя, поэтому напрашивается условие

$$x_2 \cdot u \le c.$$

Подобные условия, включающие как управление, так и фазовую переменную называются смешанными и являются существенными для многих практических задач.

Реальные задачи, содержащие подобные условия, как правило, не поддаются аналитическому решению. Таким образом, возникает задача построения универсальных эффективных численных методов для данного класса задач. Один подход к этой проблеме, показавший себя эффективным, описан в данной работе.

1. Аффинные задачи со смешанными ограничениями. Рассмотрим задачу оптимального управления

$$\begin{cases} \dot{x} = f(x,u,t) \equiv F_1(x,t)u + f_2(x,t), \quad t \in [0,T] \\ h_i(x,u,t) \equiv \langle n_i(x,t), u\rangle + m_i(x,t) \le 0, \quad 1 \le i \le d(h) \\ K(x(0),x(T)) = 0 \\ u \in U \\ J(u) \to \min\limits_{u} \end{cases}. \tag{1}$$

Область управления U является гладким выпуклым компактом или многогранником и задается опорной функцией $c(\psi) = \max\limits_{u \in U}(u, \psi)$.

Определение. Под **гладким выпуклым компактом** будем понимать такой выпуклый компакт, опорная функция которого удовлетворяет следующим условиям:

1. Условие гладкости $\exists c''(\psi), \quad \forall \psi \neq 0$.
2. Условие максимальности ранга гессиана $\operatorname{rank}[c''(\psi)] = \dim(u) - 1$.

Предлагаемый метод решения данной задачи базируется на необходимом условии оптимальности (принципе максимума), которое является частным случаем теоремы (44, стр. 126) из [5]. С помощью принципа максимума задача оптимального управления сводится к краевой задаче (П–системе) (см. [8,12]), которую можно трактовать как некоторое уравнение относительно начального значения расширенной переменной

$$\Phi(z) = 0, \text{ где } \quad z = (x(0), \psi(0)), \quad \Phi : \mathbb{R}^{2n} \to \mathbb{R}^{2n}. \tag{2}$$

Первая проблема заключается в том, что управление, максимизирующее функцию Гамильтона-Понтрягина, (максимизатор) не находится аналитически, следовательно необходимо находить его численно. Применение численных методов (градиентного [4], Ньютона [9,11], продолжения по параметру [1,8,13]) к уравнению (2) порождает вторую проблему: нахождение производной(якобиана) левой части уравнения. Это потребует решения задачи Коши для уравнения первой вариации, в которое войдет производная максимизатора и множителей Лагранжа, соответствующих смешанным ограничениям.

Решение сформулированных проблем предлагается в данной работе.

1.1. Нахождение максимизатора функции Гамильтона-Понтрягина. Функция Гамильтона-Понтрягина в этой задаче является аффинной по управлению

$$H(x, \psi, u, t) = \langle \psi, f(x, u, t) \rangle \equiv \langle \psi, F_1(x, t)u + f_2(x, t) \rangle \equiv \langle a(x, \psi, t), u \rangle + b(x, \psi, t).$$

При нахождении ее максимизатора возникает задача

$$\begin{cases} \langle a, u \rangle \to \max\limits_{u} \\ u \in U \\ \langle n_i, u \rangle + m_i \leq 0, \quad 1 \leq i \leq d(h) \end{cases}. \tag{3}$$

Результатом пересечения выпуклого множества U с выпуклыми полупространствами, порожденными линейными ограничениями, является выпуклое множество (вырожденный случай пустого множества, не рассматриваем). Будем рассматривать случай общего положения, предполагая, что задача максимизации линейного функционала на таком множестве имеет единственное решение. Для численного метода представляет интерес лишь регулярный случай, т.к. при малейшем изменении входных данных вырожденность пропадает.

Условие $u \in U$ можно переписать в терминах опорной функции

$$\langle \psi, u \rangle \leq c(\psi), \quad \forall \psi.$$

Если область управления является многогранником, то имеем конечное число аффинных ограничений вида неравенств и задача превращается в классическую задачу линейного программирования

$$\begin{cases} \langle a, u \rangle \to \max\limits_{u} \\ \langle \psi_i, u \rangle - c(\psi_i) \leq 0, \quad 1 \leq i \leq k \\ \langle n_i, u \rangle + m_i \leq 0, \quad 1 \leq i \leq d(h) \end{cases}$$

Эту задачу можно решать, например, симплекс-методом, приведя сначала к каноническому виду. В дальнейшем будем считать, что в такой задаче ограничения вида $u \in U$ отсутствуют, а присутствуют лишь смешанные ограничения.

В случае гладкого выпуклого компакта напрашивается метод построения многогранных аппроксимаций для этого множества и сведение задачи к предыдущему случаю. Однако, для размерности управления больше двух задача линейного программирования для нахождения максимизатора с требуемой точностью будет крайне велика.

Предложим другой метод решения задачи нахождения максимизатора для гладкого выпуклого компакта U. В случае активности ограничения $u \in U$ решение, получаемое предлагаемым методом, характеризуется тем, что удовлетворяет этому ограничению точно. Алгоритм состоит из трех этапов.

1 этап. Определение активности смешанных ограничений.

1. Нахождение кандидата на роль максимизатора

$$\tilde{u}^* = c'(a).$$

2. Проверка, удовлетворяет ли кандидат $\tilde{u}^*$ смешанным ограничениям

$$\langle n_i, \tilde{u}^* \rangle + m_i \le 0, \quad 1 \le i \le d(h).$$

Если ограничения удовлетворяются, смешанные ограничения неактивны, и максимизатор найден $u^* = \tilde{u}^*$, в противном случае, смешанные ограничения активны, переходим к следующему этапу.

2 этап. Определение активности ограничения $u \in U$ (активность ограничения означает принадлежность точки границе множества).

Решение задачи линейного программирования с конечным числом аффинных ограничений

$$\begin{cases} \langle a, u \rangle \to \max\limits_{u} \\ \langle n_i, u \rangle + m_i \le 0, \quad 1 \le i \le d(h) \end{cases}.$$

Как уже упоминалось, будем рассматривать случай общего положения, в котором решение данной задачи единственно. Если максимум достигается в некоторой точке $\bar{u}^*$, производится проверка этой точки на удовлетворение ограничению $\bar{u}^* \in U$ (алгоритм проведения этой проверки будет описан ниже). Если оно удовлетворяется, ограничение $u \in U$ неактивно, и максимизатор найден $u^* = \bar{u}^*$. В противном случае, если ограничение не удовлетворяется, или максимум не достигается, ограничение $u \in U$ активно, переходим к следующему этапу.

3 этап. Нахождение максимума по границе множества.

Предложим метод решения задачи нахождения максимума по границе множества с учетом аффинных ограничений

$$\begin{cases} \langle a, u \rangle \to \max\limits_{u \in dU} \\ \langle n_i, u \rangle + m_i \le 0, \quad 1 \le i \le d(h) \end{cases}. \tag{4}$$

Так как область управления замкнута и строго выпукла, а максимум достигается на его границе, очевидно, что решение задачи максимизации существует и единственно. Таким образом, необходимые условия оптимальности, выполняющиеся в единственной точке, являются и достаточными.

Применим правило множителей Лагранжа (теорему Куна-Таккера). Максимум функции Лагранжа

$$L(u) = \langle a, u \rangle - \sum_{1 \le j \le d(h)} \lambda_j \left(\langle n_j, u \rangle + m_j \right) = \left\langle a - \sum_{1 \le j \le d(h)} \lambda_j n_j, u \right\rangle + \sum_{1 \le j \le d(h)} \lambda_j m_j$$

при условии $u \in dU$ достигается на векторе

$$u = c' \left(a - \sum_{1 \le j \le d(h)} \lambda_j n_j \right).$$

Подставляя это выражение в условие, дополняющее нежесткости, получим систему уравнений относительно $\{\lambda_i\}$

$$\lambda_i \left(\left\langle n_i, c' \left(a - \sum_{1 \le j \le d(h)} \lambda_j n_j \right) \right\rangle + m_i \right) = 0, \quad 1 \le i \le d(h). \tag{5}$$

Необходимо найти нетривиальный корень данной системы $\{\lambda_i \geqslant 0\}$, удовлетворяющий ограничениям

$$\left\langle n_i, c' \left(a - \sum_{1 \le j \le d(h)} \lambda_j n_j \right) \right\rangle + m_i \le 0, \quad 1 \le i \le d(h). \tag{6}$$

Рассматриваемый случай общего положения подразумевает, что число активных ограничений меньше размерности вектора управлений и матрица, составленная из вектора a и векторов активных ограничений n_i, имеет полный ранг. В этих предположениях полученное необходимое условие будет также достаточным, т.к. сделанные предположения на опорную функцию гарантируют, что некоторому значению

$$u^* = c' \left(a - \sum_{1 \le j \le d(h)} \lambda_j n_j \right)$$

может соответствовать не более одного набора λ_j. Алгоритм поиска точки, удовлетворяющей данному условию, будет описан ниже.

1.2. Множители Лагранжа и сопряженное уравнение. При активности смешанных ограничений сопряженная система приобретает вид

$$\dot{\psi} = -\left(f_x\right)^T \psi + \left(h_x\right)^T \lambda.$$

В нее входят множители Лагранжа, соответствующие смешанным ограничениям. Если ограничение $u \in U$ активно, λ_i определяются из системы уравнений (5), если же нет, их можно определить из системы линейных уравнений

$$\begin{cases} \sum\limits_{j \in I_h} \lambda_j n_j = a \\ \lambda_j = 0, \quad j \notin I_h \end{cases}, \tag{7}$$

где I_h - множество активных индексов смешанных ограничений.

Таким образом, расширенная система ОДУ, образующая вместе с краевыми условиями краевую задачу выглядит следующим образом

$$\begin{cases} \dot{x} = f(x, u^*(x, \psi)) \\ \dot{\psi} = -\left(f_x(x, u^*(x, \psi))\right)^T \psi + \left(h_x(x, u^*(x, \psi))\right)^T \lambda(x, \psi) \end{cases}. \tag{8}$$

Как упоминалось выше, в численных методах, таких как метод Ньютона или метод продолжения по параметру используется также система уравнений первой вариации данной системы

$$\begin{cases} \dfrac{X_x}{dt} = f_x(x,u^*(x,\psi)) + f_u(x,u^*(x,\psi))u^*_x(x,\psi) \\ \dfrac{X_\psi}{dt} = f_u(x,u^*(x,\psi))u^*_\psi(x,\psi) \\ \dfrac{\Psi_x}{dt} = \dfrac{\partial}{\partial x}\left(-\left(f_x(x,u^*(x,\psi))\right)^T\psi + \left(h_x(x,u^*(x,\psi))\right)^T\lambda(x,\psi)\right) \\ \dfrac{\Psi_\psi}{dt} = \dfrac{\partial}{\partial \psi}\left(-\left(f_x(x,u^*(x,\psi))\right)^T\psi + \left(h_x(x,u^*(x,\psi))\right)^T\lambda(x,\psi)\right) \\ X_x(0) = I, \Psi_\psi(0) = I, X_\psi(0) = 0, \Psi_x(0) = 0 \end{cases} . \tag{9}$$

Как видно, в нее входят производные $u^*(x,\psi)$ и $\lambda(x,\psi)$ по (x,ψ). Следующий пункт посвящен их нахождению.

1.3. Нахождение производной максимизатора и множителей Лагранжа по фазовому вектору. Итак, необходимо вычисление производных максимизирующей функции $u^*(z)$ и функции множителей Лагранжа $\lambda(z)$ по вектору $z = (x,\psi)$.

Случай А) Ограничение $u \in U$ активно. Производные $\frac{d}{dz}u^*(z)$ и $\frac{d}{dz}\lambda(z)$ можно определить, продифференцировав по z систему уравнений

$$\begin{cases} h_i(z,u^*(z)) \equiv \langle n_i(z), u^*(z)\rangle + m_i(z) = 0, \quad i \in I_h \\ u^*(z) - c'\left(a(z) - \sum\limits_{j\in I_A}\lambda_j(z)n_j(z)\right) = 0 \\ \lambda_i(z) = 0, \quad i \notin I_h \end{cases} ; \tag{10}$$

здесь $n_i(z)$ зависят лишь от x, и считаются зависящими от ψ формально, для соблюдения единообразия записи. В результате дифференцирования получим систему линейных уравнений для нахождения $\frac{d}{dz}u^*(z)$ и $\frac{d}{dz}\lambda(z)$

$$\begin{cases} u^*\cdot\dfrac{\partial}{\partial z}n_i + \dfrac{\partial}{\partial z}m_i + n_i\left[\dfrac{d}{dz}u^*\right] = 0, \quad i \in I_h \\ \left[\dfrac{d}{dz}u^*\right] - c''\left(a - \sum\limits_{j\in I_h}\lambda_j n_j\right)\cdot\left(\dfrac{\partial}{\partial z}a - \sum\limits_{j\in I_h}\left(\lambda_j\dfrac{\partial}{\partial z}n_j + n_j\cdot\left[\dfrac{d}{dz}\lambda_j\right]\right)\right) = 0 . \\ \left[\dfrac{d}{dz}\lambda_i\right] = 0, \quad i \notin I_h \end{cases} \tag{11}$$

В матричной форме это выглядит так

$$\begin{pmatrix} N & 0 \\ I & c''N^T \end{pmatrix}\begin{pmatrix} \dfrac{du}{dz} \\ \dfrac{d\lambda}{dz} \end{pmatrix} = \begin{pmatrix} -\dfrac{\partial h}{\partial z} \\ c''\cdot\dfrac{\partial}{\partial z}\left(a - N^T\lambda\right) \end{pmatrix}, \tag{12}$$

где $N = \left(\frac{\partial \tilde{h}}{\partial u}\right)$ — прямоугольная матрица активных ограничений, I — единичная матрица.

Отсюда, учитывая следующую из условия общего положения невырожденность матрицы

$$\begin{pmatrix} N & 0 \\ I & c''N^T \end{pmatrix}$$

получим

$$\begin{pmatrix} \dfrac{du}{dz} \\ \dfrac{d\lambda}{dz} \end{pmatrix} = \begin{pmatrix} N & 0 \\ I & c'' N^T \end{pmatrix}^{-1} \begin{pmatrix} -\dfrac{\partial h}{\partial z} \\ c'' \cdot \dfrac{\partial}{\partial z}\left(a - N^T\lambda\right) \end{pmatrix}. \tag{13}$$

Случай Б) Ограничение $u \in U$ неактивно. Здесь случай общего положения предполагает, что число активных ограничений равно размерности вектора управлений u и матрица N невырождена.

В этом случае $\frac{d}{dz}u^*(z)$ и $\frac{d}{dz}\lambda(z)$ находим из системы линейных уравнений

$$\begin{cases} u^* \cdot \dfrac{\partial}{\partial z} n_i + \dfrac{\partial}{\partial z} m_i + n_i \left[\dfrac{d}{dz} u^*\right] = 0, \quad i \in I_h \\ \dfrac{\partial}{\partial z} a - \sum\limits_{j \in I_h} \left(\lambda_j \dfrac{\partial}{\partial z} n_j + n_j \cdot \left[\dfrac{d}{dz}\lambda_j\right]\right) = 0 \\ \left[\dfrac{d}{dz}\lambda_i\right] = 0, \quad i \notin I_h \end{cases}. \tag{14}$$

В матричной форме это выглядит так

$$\begin{pmatrix} N & 0 \\ 0 & N^T \end{pmatrix} \begin{pmatrix} \dfrac{du}{dz} \\ \dfrac{d\lambda}{dz} \end{pmatrix} = \begin{pmatrix} -\dfrac{\partial h}{\partial z} \\ \dfrac{\partial}{\partial z}\left(a - N^T\lambda\right) \end{pmatrix}. \tag{15}$$

Отсюда

$$\begin{aligned} &\frac{du}{dz} = N^{-1}\left(-\frac{\partial h}{\partial z}\right) \\ &\frac{d\lambda}{dz} = \left(N^T\right)^{-1} \cdot \frac{\partial}{\partial z}\left(a - N^T\lambda\right) = \left(N^{-1}\right)^T \cdot \frac{\partial}{\partial z}\left(a - N^T\lambda\right) \end{aligned}. \tag{16}$$

1.4. Метод поиска корня системы уравнений (5). Задача (5) – (6) имеет вид

$$\begin{cases} \lambda_i g_i(\lambda) = 0 \\ g_i(\lambda) \le 0 \\ \lambda_i \geqslant 0 \\ \lambda \neq 0 \end{cases}, \quad 1 \le i \le d(h), \tag{17}$$

где

$$g_i(\lambda) \equiv \left\langle n_i, c'\left(a - \sum_{1 \le j \le d(h)} \lambda_j n_j\right)\right\rangle + m_i. \tag{18}$$

Рассмотрим систему

$$\begin{cases} \tilde{g}_i(\lambda) = 0 \\ \lambda_i \geqslant 0 \end{cases}, \tag{19}$$

где

$$\tilde{g}_i(\lambda) \equiv \left(\lambda_i + \max(g_i(\lambda), 0)\right) g_i(\lambda) \equiv \lambda_i g_i(\lambda) + \max(g_i(\lambda), 0)^2, \tag{20}$$

Покажем, что система (19) эквивалентна системе (17).

$$\begin{cases} \left(\lambda_i + \max(g_i(\lambda), 0)\right) g_i(\lambda) = 0 \\ \lambda_i \geqslant 0 \end{cases} \Leftrightarrow \begin{cases} \left[\begin{array}{l} \lambda_i + \max(g_i(\lambda), 0) = 0 \\ g_i(\lambda) = 0 \end{array}\right. \\ \lambda_i \geqslant 0 \end{cases} \Leftrightarrow$$

$$\Leftrightarrow \begin{cases} \left[\begin{array}{l} \begin{cases} \lambda_i = 0 \\ \max(g_i(\lambda), 0) = 0 \end{cases} \\ g_i(\lambda) = 0 \end{array}\right. \\ \lambda_i \geqslant 0 \end{cases} \Leftrightarrow \begin{cases} \left[\begin{array}{l} \begin{cases} \lambda_i = 0 \\ g_i(\lambda) \leq 0 \end{cases} \\ g_i(\lambda) = 0 \end{array}\right. \\ \lambda_i \geqslant 0 \end{cases} \Leftrightarrow \begin{cases} \lambda_i g_i(\lambda) = 0 \\ g_i(\lambda) \leq 0 \\ \lambda_i \geqslant 0 \end{cases}$$

Решать уравнение $\tilde{g}_i(\lambda) = 0$ можно известными численными методами (например, методом Ньютона), автор предлагает использовать метод продолжения по параметру и для этого уравнения, т.к. он предъявляет не такие жесткие требования к выбору начального приближения. Для использования этих методов необходимо найти производную функции $\tilde{g}_i(\lambda)$

$$\frac{d}{d\lambda_i}\tilde{g}_i(\lambda) \equiv g_i(\lambda) + (\lambda_i + 2\max(g_i(\lambda), 0))\,\frac{dg_i(\lambda)}{d\lambda_i},$$

где

$$\frac{dg_i(\lambda)}{d\lambda_j} = -\left\langle n_i, c''\left(a - \sum_{1\leq j\leq d(h)} \lambda_j n_j\right) n_j \right\rangle = -N \cdot c'' \cdot N^T.$$

1.5. Алгоритм проверки принадлежности точки компакту. Следующий известный алгоритм принадлежности точки компакту ($\bar{u}^* \in U$) излагается в курсе лекций Ю.Н.Киселева.

Введем функцию

$$V(q) = \frac{1}{2}\,\|q\|^2 + c(q) - \langle \bar{u}^*, q\rangle\,., \tag{21}$$

где $c(q)$ — опорная функция множества U .

Свойства функции (21):

1. функция $V(q)$ строго выпукла;
2. имеет место предельное соотношение

$$V(q) \to +\infty \qquad \|q\| \to \infty;$$

3. существует единственный минимизатор

$$q_\# \equiv \arg\min_q V(q),$$

решающий задачу минимизации

$$V(q) \to \min_{q\in\mathbb{R}^n};$$

4. минимизатор принадлежит множеству уровня

$$q_\# \in \{q \in \mathbb{R}^n : V(q) \leq 0\},$$

причем

$$V_\# \equiv \min_{q\in\mathbb{R}^n} V(q) = V(q_\#) \leq 0.$$

Теорема.(критерий принадлежности точки $\bar{u}^*$ множеству U)

$$\bar{u}^* \in U \Leftrightarrow m_0 \geqslant 0,$$

$$\bar{u}^* \notin U \Leftrightarrow m_0 < 0,$$

где

$$m_0 \equiv \min_{\psi:\|\psi\|=1} (c(\psi) - (\bar{u}^*, \psi)).$$

Необходимо определить, существует ли $q : V(q) < 0$. Так как функция $V(q)$ выпуклая, с помощью какого-либо численного метода построим последовательность $\{q_n\} \to q_\# \equiv \arg\min_q V(q)$. Если на некотором шаге $V(q_n) < 0$, точка $\bar{u}^*$ не принадлежит множеству. Автор предлагает использовать градиентный метод, для которого необходимы производные функции $V(q)$

$$V'(q) = q + c'(q) - \bar{u}^*,$$
$$V''(q) = E + c''(q).$$

2. Аффинные задачи с фазовыми ограничениями. Пусть в задаче (1) присутствуют фазовые ограничения

$$P_s(x,t) \le 0, \quad 1 \le i \le d(P). \tag{22}$$

Будем требовать, чтобы функции $P_s(x,t)$ были непрерывно дифференцируемы по совокупности (x,t).

Продифференцировав активные фазовые ограничения вдоль траектории системы, получим

$$\left\langle \frac{\partial P_s(x,t)}{\partial x}, f(x,u,t) \right\rangle + \frac{\partial P_s(x,t)}{\partial t} \le 0, \quad s \in I_P, \tag{23}$$

где I_P - множество активных индексов фазовых ограничений.

Таким образом, на промежутках постоянства множества активных индексов I_P задачу с фазовыми ограничениями заменить эквивалентной задачей со смешанными ограничениями и применять к ней алгоритм предложенный выше. Из-за неизбежно возникающих погрешностей численного интегрирования фазовая траектория будет сходить с фазовых ограничений, для устранения этой проблемы целесообразно производить проектирование на поверхность активных фазовых ограничений.

Опишем алгоритм смены множества активных индексов фазовых ограничений. Допустим, на некотором шаге численного интегрирования динамической системы мы имеем множество активных индексов I_P. Смена множества активных индексов происходит, если на очередном шаге, находя максимизатор, исходя из предположения, что множество I_P остается прежним, мы тем самым вывели траекторию из фазовых ограничений (случай 1), или некоторые из активных фазовых ограничений перестали быть активными (случай 2).

В первом случае, шаг интегрирования отклоняется, и размер шага выбирается как можно большим при удовлетворении фазовых ограничений. Таким образом, некоторые фазовые ограничения переходят в разряд активных. Второй случай возникает, если некоторые смешанные ограничения, соответствующие активным фазовым ограничениям оказались неактивными. Такой шаг интегрирования принимается, и такие фазовые ограничения перестают быть активными.

При использовании предложенного подхода на протяжении всего процесса решения задачи траектория удерживается в фазовых ограничениях. Если же на некотором этапе удовлетворение смешанных ограничений невозможно, процесс решения останавливается.

Замечание.

Во многих задачах существует область начальных значений (x_0, ψ_0), начиная из которой, траекторию будет невозможно удержать в фазовых ограничениях описанным выше способом. Во избежание остановки вычислений в таких случаях можно предложить на начальных итерациях решения:

1. Отказаться от удовлетворения фазовых ограничений в точках, где это невозможно;
2. Ввести в систему, решаемую методом продолжения по параметру, уравнение, которое удовлетворяется только при условии удовлетворения фазовых ограничений (возможно добавление штрафного члена к существующему уравнению).

Таким образом, будет создано достаточно хорошее начальное приближение, для которого фазовые ограничения будут удовлетворяться описанным выше способом.

Приведенные алгоритмы реализованы в разрабатываемом автором программном комплексе Система Optimus [7]. Использование их внутри метода продолжения решения по параметру [8] позволило решать широкий класс сложных задач, представляющих практический интерес.

СПИСОК ЛИТЕРАТУРЫ

1. *Аввакумов С.Н., Киселев Ю.Н., Орлов М.В.* Методы решения задач оптимального управления на основе принципа максимума Понтрягина // Труды Математического Института им. В.А. Стеклова РАН. 1995. Т.211. 3–31.
2. *Арутюнов А.В.* К теории принципа максимума в задачах оптимального управления с фазовыми ограничениями // ДАН СССР. 1989. Т.304, №1. 11–14.
3. *Асеев С.М.* К теории необходимых условий оптимальности для задач с фазовыми ограничениями // Труды математического Института им. В.А. Стеклова РАН. 1998. Т.220. 35–44.
4. *Васильев Ф.П.* Методы оптимизации. М.: Факториал пресс, 2002.
5. *Дмитрук А.В., Милютин А.А., Осмоловский Н.П.* Принцип максимума в оптимальном управлении. М.: Изд-во механико-математического факультета МГУ, 2004.
6. *Дубовицкий А.Я., Милютин А.А.* Задачи на экстремум при наличии ограничений // Журнал вычислительной математики и математической физики. 1965. 5, №3. 395–453.
7. *Жулин С.С.* Численное решение задач оптимального управления с помощью сисемы Optimus // Проблемы динамического управления: Сборник научных трудов ВМиК МГУ им. М.В.Ломоносова /под ред. Ю.С. Осипова, А.В. Кряжимского, — М.: Издательский отдел факультета ВМиК МГУ. 2005. Вып. 1, 158–165.
8. *Жулин С.С.* Применение метода продолжения по параметру для решения сложных задач оптимального управления // Сборник статей молодых ученых факультета ВМиК МГУ, — М.: Издательский отдел факультета ВМиК МГУ. 2006. Вып. 3. 65–76.
9. *Исаев В.К., Сонин В.В.* Об одной модификации метода Ньютона численного решения краевых задач // Журнал вычислительной математики и математической физики. 1963. 3, №6. 1114–1116.
10. *Киселев Ю.Н.* Оптимальное управление. М.: Изд-во МГУ, 1988.
11. *Моисеев Н.Н.* Элементы теории оптимальных систем. М.: Наука, 1975.
12. *Федоренко Р.П.* Приближенное решение задач оптимального управления. М.: Наука, 1978.
13. *Allgower E.L., Georg K.* Introduction to numerical continuation methods. Berlin, New York: Springer-Verlag, 1990. (Reprinted as volume 45 of Classics in Applied Mathematics. SIAM, Philadelphia, PA, 2003).
14. *Avvakumov S.N., Kiselev Yu.N.* Boundary value problem for ordinary differential equations with applications to optimal control. // Spectral and Evolution Problems. 2000. Vol 10. Proceeding of the Tenth Crimean Autumn mathematical School – Symposium, Simferopol, Ukraine.
15. *Hartl R.F., Sethi S.P., Vicson R.G.* A survey of the maximum principles for optimal control problems with state constraints // SIAM Review. 1995. 37, №2. 181–218.

УДК 519.6

О ДВУХ МОДЕЛЯХ ПОПУЛЯЦИОННОЙ ДИНАМИКИ Т-ЛИМФОЦИТОВ

© 2007 г. О. А. Казакова

grim@list.ru

Кафедра вычислительных технологий и моделирования

1. Введение.

Эта работа относится к одному из разделов математической биологии и биоинформатики — математической иммунологии. Традиционный круг задач в данной области включает построение и исследование моделей иммунного ответа и иммунорегуляции, иммунной защиты организма при инфекционных заболеваниях [1]. Сложность изучения иммунной системы связана с необходимостью одновременного описания большого количества взаимодействующих элементов, с распределенным и адаптивным характером происходящих в ней процессов. Универсальные модели функционирования иммунной системы в настоящее время отсутствуют.

В последние годы в связи с увеличением средней продолжительности жизни населения европейских стран и пониманием важной роли иммунной системы в обеспечении здорового долголетия растет интерес к изучению взаимосвязей иммунного статуса с наблюдаемыми демографическими, эпидемиологическими и экологическими изменениями [12,13]. Для управления состоянием здоровья популяции необходимо знание *механизмов возрастных изменений* иммунной системы на индивидуальном уровне. Это привело к началу моделирования процессов долговременной адаптации системы иммунной защиты [4,5,9].

2. Иммунная система человека.

Основная функция иммунной системы заключается в поддержании целостности внутренней среды организма путем распознавания и нейтрализации генетически чужеродных факторов (*антигенов*). Иммунная система представлена двумя типами антиген-специфических клеток (детекторов антигенных структур): *Т-* и *В-лимфоцитами*. Каждый лимфоцит способен взаимодействовать лишь с одним антигеном. Такая способность имеет название *специфичности*. Лимфоциты одной специфичности образуют *клон*. Основной тип иммунной реакции — *иммунный ответ* — заключается в каскадном размножении клеток одного или нескольких клонов с последующей нейтрализацией антигена. В организме человека содержится около 10^{12} лимфоцитов, а общее количество клонов оценивается величиной порядка 10^6–10^7.

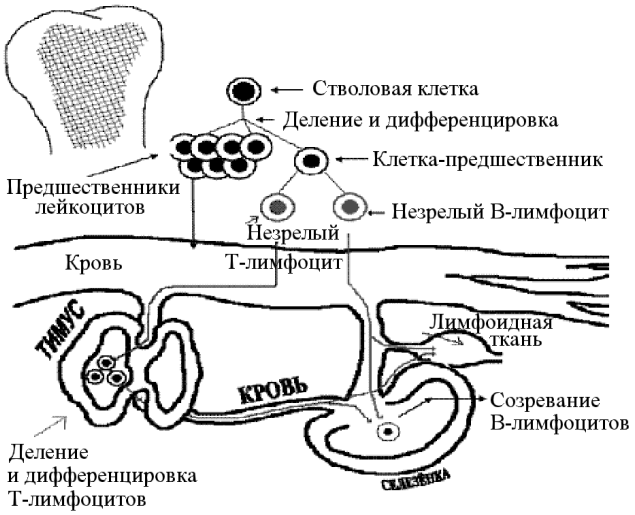


Рис. 1. Схема процесса развития Т- и В-лимфоцитов

Общая схема процесса развития лимфоцитов представлена на рис. 1. Из находящихся в костном мозге *полипотентных стволовых кроветворных клеток* в результате нескольких циклов деления и дифференцировки образуются предшественники В- и Т-лимфоцитов. Пре-В-лимфоциты завершают развитие и проходят “отбор” в специальном лимфоидном органе — селезенке, а пре-Т-лимфоциты — в тимусе. В процессе отбора около 95–97% попавших в тимус пре-Т-лимфоцитов погибают, а остальные клетки после нескольких циклов деления и дифференцировки выходят из тимуса в интактную периферическую лимфоидную ткань (ИПЛТ) в

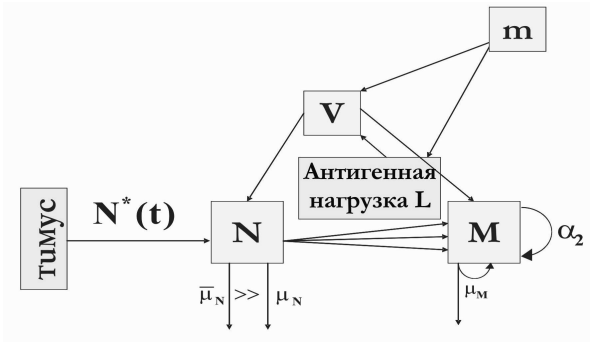


Рис. 2. Блок-схема модели Романюхи и Яшина [5]

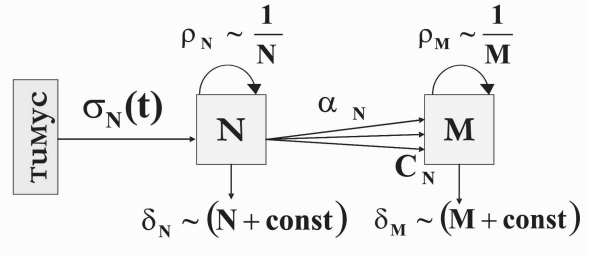


Рис. 3. Блок-схема модели Де Бура [9]

виде функционально зрелых *наивных* Т-лимфоцитов. При встрече с антигеном такие клетки участвуют в иммунном ответе. По завершении иммунного ответа большинство клеток клона погибает в результате *апоптоза* (программируемой клеточной гибели), а остальная часть (около 5%) формирует популяцию долгоживущих *клеток памяти*.

Упрощенно, В-лимфоциты обеспечивают защиту против бактериальных, а Т-лимфоциты — против вирусных антигенов. Кроме того, Т-лимфоциты участвуют в В-клеточном иммунном ответе. При этом наиболее выраженные изменения с возрастом наблюдаются именно в Т-системе иммунитета. Несмотря на большое количество проведенных экспериментальных и клинических исследований, изучение механизмов возрастной изменчивости Т-системы иммунитета остается сложной до конца не решенной задачей [7,13].

3. Модели популяционной динамики Т-лимфоцитов.

Состояние Т-системы иммунитета зависит от многих факторов. Ряд ее характеристик невозможно или крайне трудно определить непосредственно. Это привело к разработке ряда математических моделей функционирования Т-системы иммунитета в норме [4,5,9,10] и при заболеваниях [8,11,15]. Цель работы заключалась в сопоставлении двух таких моделей — модифицированной модели Романюхи и Яшина [5] и модели Де Бура [9].

В основу математического описания динамики возрастных изменений Т-системы иммунитета в обеих моделях положена *теломерная гипотеза старения*. В соответствии с этой гипотезой помимо процессов рождения–гибели и размножения клеток в моделях описан механизм снижения с возрастом *репликативного потенциала*, т. е. способности клеток к делению. Репликативный потенциал характеризуется длиной *теломер* — концевых участков хромосом, имеющих периодическую структуру $((\mathrm{TTAGGG})_n)$. У наивных Т-лимфоцитов репликативный потенциал максимален и составляет около 50–70 делений, что соответствует длине теломер порядка 9–10 Кб пар оснований (п. о.). При каждом делении теломеры Т-лимфоцита укорачиваются на определенную длину. Существует критическая длина теломер (около 4–5 Кб п. о.), называемая *пределом Хейфлика*, при которой клетка уже не способна к делению.

В некоторых типах клеток (например, в В-лимфоцитах) активен фермент *теломераза*, восстанавливающий укорачиваемые при делении теломеры. В Т-лимфоцитах активность теломеразы незначительна, поэтому Т-система иммунитета стареет быстрее. Другая причина этого заключается в раннем (в возрасте одного года!) начале атрофии тимуса.

Второе предположение, лежащее в основе рассматриваемых моделей, — о наличии гомеостатического механизма поддержания популяции Т-лимфоцитов — используется явно в модели Романюхи и Яшина и неявно — в модели Де Бура. Оно заключается в том, что, независимо от антигенной нагрузки, при уменьшении численности популяции Т-лимфоцитов ниже "должных" значений происходит их размножение с целью нормализации данного показателя.

4. Модифицированная модель Романюхи и Яшина.

Блок-схема модифицированной модели возрастных изменений популяции периферических Т-лимфоцитов [5] показана на рис. 2. Модель имеет вид системы 8 нелинейных обыкновенных

дифференциальных уравнений:

$$
\begin{aligned}
&\frac{dN^*}{dt} = -k_T N^*,\\
&\frac{dN}{dt} = \frac{N^*}{V} - \alpha_1 \frac{L}{V} N - \mu_N N - \frac{dV}{dt}\frac{N}{V},\\
&\frac{dM}{dt} = \rho_1\alpha_1 \frac{L}{V} N + \rho_2\alpha_2 \frac{L}{V} M + \mu_M (C^* - N - M) - \frac{dV}{dt}\frac{M}{V},\\
&\frac{dP^*}{dt} = -(\frac{\bar{k}_P}{m}\frac{dm}{dt} + k_P)P^*,\\
&\frac{dP_N}{dt} = (P^* - P_N)\frac{N^*}{NV},\\
&\frac{dP_M}{dt} = \rho_1\alpha_1 (P_N - P_M - \lambda_N)\frac{L}{V}\frac{N}{M} - (\rho_2 + 1)\alpha_2\lambda_M \frac{L}{V},\\
&\frac{dV}{dt} = \alpha_3 \frac{L}{V}\frac{dm}{dt} - k_V V,\\
&\frac{dm}{dt} = \alpha_4 m^{3/4} - k_m m.
\end{aligned}
\tag{1}
$$

Первое уравнение описывает снижение с возрастом скорости притока наивных клеток N^* из тимуса. Оно соответствует экспериментальным данным для возраста старше одного года.

Второе уравнение описывает динамику концентрации наивных Т-лимфоцитов (N) в ИПЛТ. Первое слагаемое в правой части характеризует приток клеток из тимуса, второе — убыль в результате антигенной стимуляции, третье — естественную гибель наивных Т-лимфоцитов, а четвертое — изменение концентрации клеток при изменении объема ИПЛТ (V).

Третье уравнение описывает динамику концентрации Т-лимфоцитов памяти в ИПЛТ (M). Первое слагаемое справа описывает приток клеток в результате деления наивных Т-лимфоцитов (ρ_1 — коэффициент размножения наивных клеток), второе слагаемое — увеличение концентрации клеток памяти за счет их повторной антигенной стимуляции (ρ_2 — коэффициент размножения клеток памяти). Третье слагаемое описывает механизм поддержания постоянной концентрации лимфоцитов в ИПЛТ (C^*) за счет ускоренной гибели избыточных клеток памяти, последнее слагаемое учитывает изменение объема ИПЛТ.

Четвертое уравнение описывает скорость снижения длины теломер P^* наивных Т-лимфоцитов, выходящих из тимуса. Предполагается, что эта скорость определяется длиной теломер стволовых клеток, зависящей от темпа обновления соматических клеток, и от дополнительной нагрузки на популяцию стволовых клеток, пропорциональной относительной скорости изменения массы тела.

Пятое уравнение описывает динамику длины теломер наивных Т-лимфоцитов (P_N). Величина P_N уменьшается за счет притока клеток из тимуса. Скорость изменения пропорциональна разности длины теломер этих клеток и относительной скорости притока новых клеток.

Шестое уравнение описывает динамику длины теломер клеток памяти (P_M). Величина P_M растет за счет длины теломер вновь образованных клеток памяти (первое слагаемое) и уменьшается при повторной антигенной стимуляции клеток памяти (второе слагаемое).

Седьмое уравнение описывает динамику изменения объема ИПЛТ. Предполагается, что темп увеличения объема ИПЛТ в процессе роста организма пропорционален удельной антигенной нагрузке $L(t)/V(t)$ и скорости изменения массы тела m. Второе слагаемое описывает медленное уменьшение объема ИПЛТ в старших возрастных группах.

Восьмое уравнение описывает динамику изменения массы тела m. Параметры α_4 и k_m в правой части задаются формулами $\alpha_4 = B_0 m_c/E_c$, $k_m = B_c/E_c$, где m_c — средняя масса клетки, E_c — энергия, необходимая для синтеза одной клетки, B_c — средняя мощность клеточного метаболизма, а B_0 — нормировочный коэффициент, характеризующий взаимосвязь между основным обменом B и массой тела m [16]: $B = B_0 m^{3/4}$. Это эмпирическое соотношение имеет название *закона трех четвертей* и было установлено экспериментально

М. Клейбером в 1932 году. В исходной работе [4] использовалось допущение о постоянстве антигенной нагрузки L. Однако в общем случае величину L можно считать пропорциональной основному обмену (B): $L = k_L B$, где k_L — постоянный параметр. Отсюда следует оценка зависимости антигенной нагрузки от массы тела [5]: $L = \alpha_5 m^{3/4}$, где $\alpha_5 = k_L B_0$.

Верификация модифицированной модели Романюхи и Яшина в работе [5] проводилась с использованием двух подходов. Сначала на основе анализа данных была построена *обобщенная картина* возрастной динамики Т-системы иммунитета, т.е. согласованное количественное описание типичного варианта рассматриваемого процесса в терминах зависимых переменных модели [2]. Начальные оценки параметров были уточнены по данным обобщенной картины при помощи аналога метода наименьших квадратов. Окончательный выбор значений параметров модели проводился с использованием принципа минимума диссипации энергии [3,14], основанного на формализации понятия биологической приспособленности [6].

5. Уравнения модели Де Бура.

Система уравнений модели Де Бура [9] (см. блок-схему на рис. 3) имеет вид

$$\begin{aligned}
\frac{dN}{dt} &= \sigma_N(t) + \rho_N(N)N - \delta_N(N)N - \alpha_N N,\\
\frac{dM}{dt} &= \sigma_M(N) + \rho_M(M)M - \delta_M(M)M,\\
\frac{d\mu_P}{dt} &= k,\\
\frac{d\mu_N}{dt} &= 2\rho_N(N) - (\sigma_N(t)/N)(\mu_N - \mu_P - K_P),\\
\frac{d\mu_M}{dt} &= 2\rho_M(M) - (\sigma_M(N)/M)(\mu_M - \mu_N - K_N).
\end{aligned} \tag{2}$$

Первое уравнение описывает динамику общей численности наивных Т-лимфоцитов в ИПЛТ (N). Первое слагаемое в правой части характеризует приток клеток из тимуса со скоростью $\sigma_N(t) = \sigma_N^0 e^{-\nu t}$, второе — гомеостатическое размножение со скоростью $\rho_N(N)N = \rho_N N/(1 + N/h_N)$. Третье слагаемое описывает естественную гибель наивных Т-лимфоцитов с параметром $\delta_N(N) = \delta(N) + \epsilon_N N$, четвертое — убыль в результате антигенной стимуляции.

Второе уравнение описывает динамику численности Т-лимфоцитов памяти в ИПЛТ (M). Первое слагаемое справа характеризует приток клеток памяти в результате размножения наивных клеток ($\sigma_M(N) = C_N \alpha_N N(t-\tau)$), где τ — период запаздывания, соответствующий длительности иммунного ответа. Так как интересующий нас интервал моделирования много больше τ, будем считать, что $N(t-\tau) \approx N(t)$. Второе слагаемое описывает размножение клеток памяти за счет повторной антигенной стимуляции ($\rho_M(M) = \rho_M/(1 + M/h_M)$), третье слагаемое — естественную гибель Т-лимфоцитов памяти ($\delta_M(M) = \delta(M) + \epsilon_M M$).

Третье уравнение описывает скорость увеличения среднего индекса деления μ_P клеток-предшественников. Предполагается, что эта скорость постоянна.

Четвертое уравнение описывает динамику среднего индекса деления наивных Т-лимфоцитов (μ_N). Величина μ_N растет при увеличении скорости размножения Т-лимфоцитов и уменьшается в результате притока новых клеток из тимуса.

Аналогичный вид имеет уравнение для динамики среднего индекса деления Т-лимфоцитов памяти (μ_M).

6. Модифицированная модель Де Бура.

Для удобства сопоставления рассматриваемых математических моделей будем считать в модели Де Бура "функцию источника" $\sigma_N(t) = \sigma_N^0 e^{-\nu t}$ зависимой переменной, вместо индексов деления μ_P, μ_N и μ_M рассмотрим длины теломер P^*, P_N и P_M. $P^*(t) = P_0^* - b_P \mu_P(t)$, где P_0^* — начальная длина теломер клеток-предшественников, $b_P = \frac{\lambda_P}{\rho_P}$ (λ_P — количество пар оснований, на которое уменьшается длина теломер за одно деление, ρ_P — частота деления, которую будем считать постоянной); $P_N(t) = P_N^0 - \frac{\lambda_N}{\rho_N}\mu_N(t)$ — средняя длина теломер наивных Т-лимфоцитов, где P_N^0 — начальная длина теломер наивных Т-лимфоцитов,

λ_N — среднее уменьшение длины теломер за одно деление, ρ_N — частота деления клеток; $P_M(t) = P_M^0 - \frac{\lambda_M}{\rho_M}\mu_M(t)$, где P_M^0 — начальная длина теломер клеток памяти, λ_M — среднее уменьшение длины теломер в результате деления, ρ_M — интенсивность деления Т-лимфоцитов памяти. Зависимость переменной P^* от времени будем считать экспоненциальной (на качество оценки параметров это практически не влияет).

В результате система уравнений модели приобретает вид

$$
\begin{aligned}
&\frac{d\sigma_N}{dt} = -\nu\sigma_N,\\
&\frac{dN}{dt} = \sigma_N(t) + \rho_N(N)N - \delta_N(N)N - \alpha_N N,\\
&\frac{dM}{dt} = C_N\alpha_N N_{t-\tau_N} + \rho_M(M)M - \delta_M(M)M,\\
&\frac{dP^*}{dt} = -b_P k\frac{P^*}{P_*^{avg}},\\
&\frac{dP_N}{dt} = -\lambda_N\left(2 - \frac{\sigma_N}{N}\left(\frac{1}{\lambda_N}(P_N^0 - P_N) - \frac{1}{b_P\rho_N(N)}(P_0^* - P^*) - \frac{K_P}{\rho_N(N)}\right) + \right.\\
&\left. + \frac{1}{(h_N + N)\lambda_N}(P_N^0 - P_N)\frac{dN}{dt}\right),\\
&\frac{dP_M}{dt} = -\lambda_M\left(2 - \frac{\sigma_M}{M}\left(\frac{1}{\lambda_M}(P_M^0 - P_M) - \frac{\rho_N(N)}{\lambda_N\rho_M(M)}(P_N^0 - P_N) - \frac{K_N}{\rho_M(M)}\right) + \right.\\
&\left. + \frac{1}{(h_M + M)\lambda_M}(P_M^0 - P_M)\frac{dM}{dt}\right).
\end{aligned}
\tag{3}
$$

7. Оценка параметров.

По аналогии с работой [5] проведено уточнение параметров модифицированной модели Де Бура по данным обобщенной картины. Для характеристики соответствия решений системы уравнений модели и данных использовался функционал вида

$$F = \sum_{i=1}^{3}\sum_{j}\left(\lg\left(\frac{x_i(t_j)}{X_i^j}\right)\right)^2,$$

где $x_i(t_j)$ — значение i-й компоненты решения в момент t_j, а X_i^j — данные обобщенной картины для i-й переменной в момент t_j. Сумма берется по тем i и j, для которых определено значение X_i^j.

Результаты уточнения параметров модели показаны на рис. 4–6. Светлыми кружками представлены данные, пунктиром и сплошными линиями — графики решений до и после настройки параметров.

8. Энергетическая цена взаимодействия с антигенами.

Общая двухкомпонентная структура энергетической цены (э. ц.) E взаимодействия с антигенами имеет вид

$$E = E_f + E_l,$$

где E_f — затраты энергии на поддержание популяции периферических Т-лимфоцитов, E_l — потери, связанные с воздействием инфекционной антигенной нагрузки [5]. В свою очередь,

$$E_f = E_f^m + E_f^t + E_f^d,$$

где E_f^m и E_f^d — расходы на поддержание и размножение наивных Т-лимфоцитов и клеток памяти в ИПЛТ, а E_f^t — расходы на образование наивных Т-лимфоцитов в тимусе. Величина E_l представима в виде

$$E_l = E_l^n + E_l^e,$$

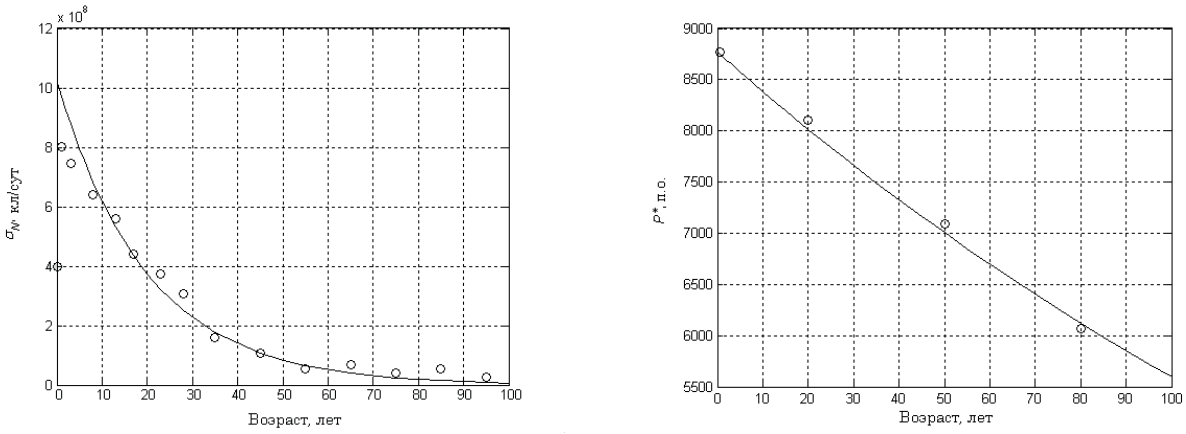


Рис. 4. Модель Де Бура. Динамика скорости притока наивных Т-лимфоцитов из тимуса (слева) и длины теломер клеток-предшественников (справа). Сплошная линия соответствует уточненным, а пунктирная — исходным значениям параметров.

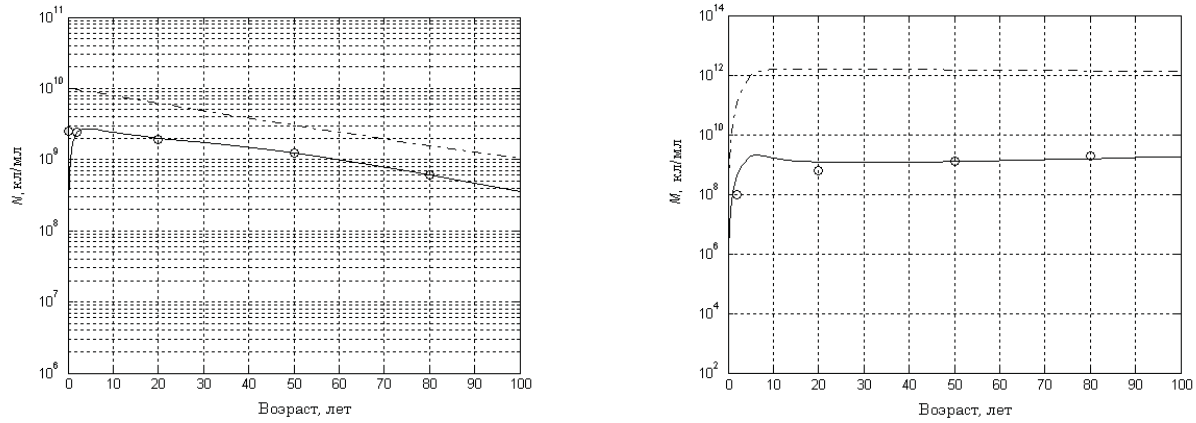


Рис. 5. Модель Де Бура. Динамика концентрации наивных Т-лимфоцитов (слева) и клеток памяти (справа).

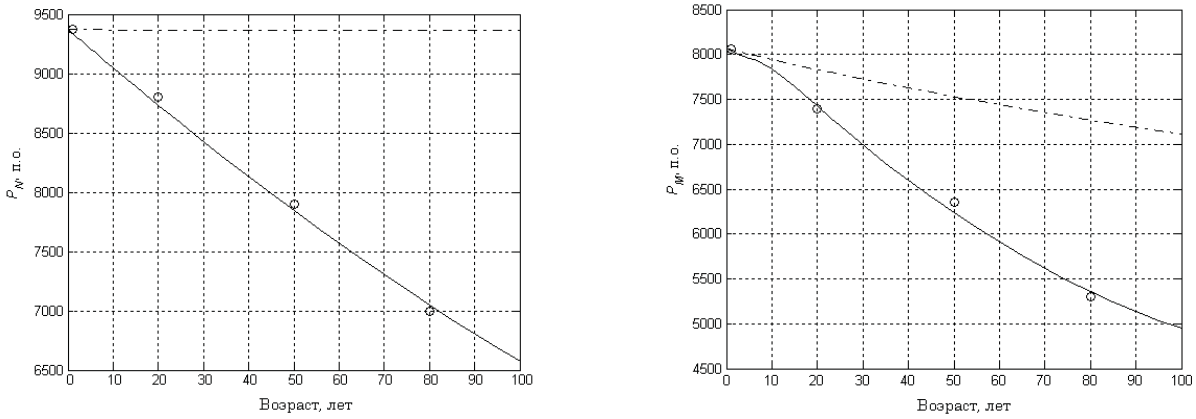


Рис. 6. Модель Де Бура. Динамика длины теломер наивных Т-лимфоцитов (слева) и теломер клеток памяти (справа).

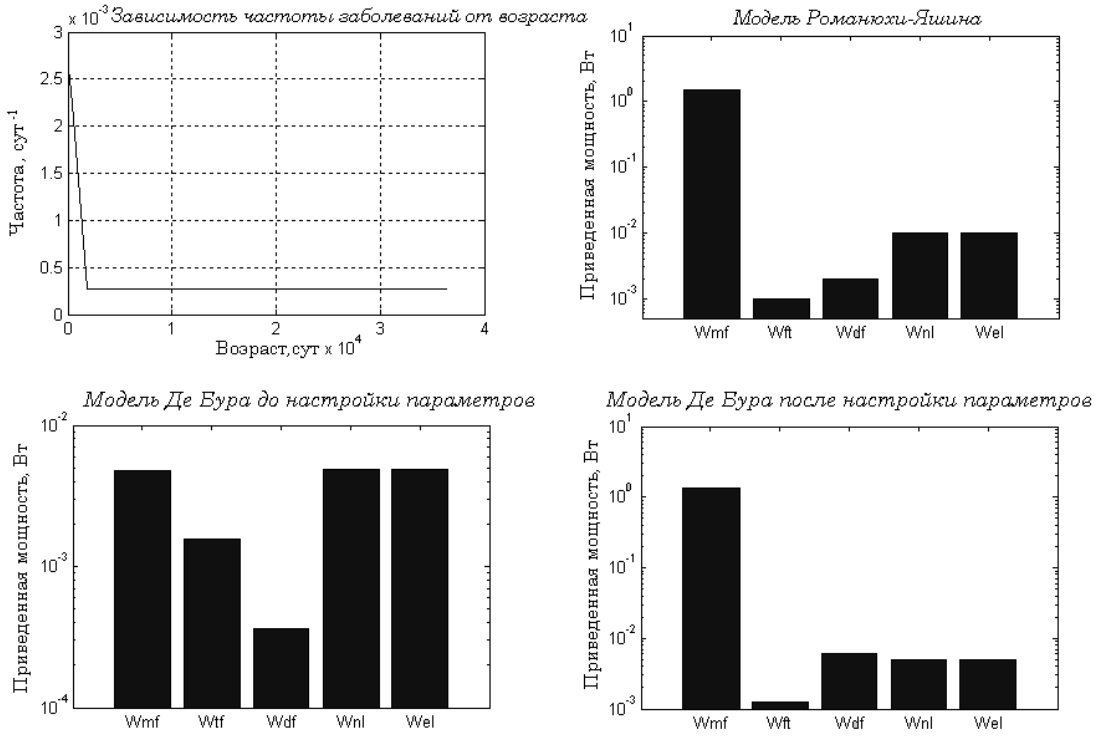


Рис. 7. Слева вверху — эмпирическая функция частоты инфекционной заболеваемости от возраста, справа — соотношения компонент расходов энергии на взаимодействие с антигенами для модели Романюхи и Яшина. Внизу показаны соответствующие данные для модели Де Бура до и после настройки параметров (результаты представлены в терминах мощностей).

где E_l^n и E_l^e — потери в результате воздействия новых и уже знакомых антигенов.

По аналогии с работой [5], для модели Де Бура эти величины можно записать в виде

$$E_f^m = \beta_1 \int_0^T (M+N)dt; \quad E_f^t = \beta_2 \int_0^T k_1\sigma_N dt; \quad E_f^d = \beta_2 \int_0^T (\rho_N N + \rho_M M + k_2 C_N \alpha_N N)dt; \quad (4)$$

$$E_l^n = \int_0^T E_0 \frac{\overline{N}}{N}\frac{\rho_N}{\overline{\rho_N}} f(t)dt; \quad E_l^e = \int_0^T E_1 \frac{\overline{M}}{M}\frac{\rho_M}{\overline{\rho_M}} \left(1 - f(t)\right) dt, \quad (5)$$

где T — время жизни индивида, $f(t)$ — частота заболеваемости в возрасте t, E_0 — приведенная э. ц. острого инфекционного заболевания в момент рождения, а E_1 — приведенная э. ц. взаимодействия с уже знакомыми антигенами: $E_1 = \int_0^T E_0 f(t)dt / \int_0^T (1 - f(t))\, dt$.

Относительный вклад компонент E_f^m, E_f^t, E_f^d, E_l^n, E_l^e в общую э. ц. взаимодействия с антигенами показан на рис. 7. После настройки параметров модифицированной системы уравнений модели Де Бура соответствующие компоненты э. ц. для рассматриваемых моделей оказались близки по величине. При этом основная часть расходов энергии была связана с поддержанием функции, а не размножением клеток, что согласуется с построенными в [5] независимыми оценками энергетики иммунной защиты по экспериментальным данным.

9. Обсуждение и выводы.

Существует ряд моделей, описывающих механизмы возрастных изменений Т-системы иммунитета. Две из них — модель А.А. Романюхи и А.И. Яшина [5] и модель Р. Де Бура [9] — рассмотрены и сопоставлены в настоящей работе. Обе модели основаны на теломерной гипотезе старения и предположении о наличии гомеостатического механизма поддержания популяции

периферических Т-лимфоцитов. Вместе с тем, ряд исходных предположений и приемов их математического описания в указанных моделях различен. Модель [5] включает зависимости от возраста массы тела и объема ИПЛТ, в явном виде в ней рассмотрена величина антигенной нагрузки L. В отличие от [5], в модели [9] предполагается наличие механизма гомеостатического размножения наивных Т-лимфоцитов без последующей дифференцировки их в клетки памяти. Константы скоростей антиген-зависимого размножения и гибели клеток в модели [9] зависят от концентрации, а в модели [5] они считаются постоянными. В модели [9] условие гомеостаза выражено в более мягкой форме.

Как и в [5], параметры модифицированной модели Де Бура [9] уточнены по данным обобщенной картины нормальных возрастных изменений Т-системы иммунитета, и для этой модели построен функционал энергетической цены взаимодействия с антигенами (характеристика эффективности иммунной системы, обратно пропорциональная приспособленности). Несмотря на перечисленные различия в структуре моделей, соответствие оценок параметров подтверждается близостью решений систем уравнений моделей и компонент энергетической цены взаимодействия с антигенами (рис. 7).

Результаты динамических обследований детей с первичными иммунодефицитами и ВИЧ-инфекцией свидетельствуют о значительном снижении чувствительности иммунной системы при увеличении антигенной нагрузки и о замедлении роста и развития организма. Это свидетельствует о необходимости учета в моделях долговременной адаптации иммунной защиты возможности изменения параметров моделей. Оценка применимости рассмотренных моделей в нестационарных условиях внешней антигенной среды требует дальнейших исследований.

Работа выполнена при частичной поддержке РФФИ (проект 07-01-00577).

СПИСОК ЛИТЕРАТУРЫ

1. *Марчук Г.И.* Математические модели в иммунологии. Вычислительные методы и эксперименты. М.: Наука, 1991. 304 с.
2. *Марчук Г.И., Романюха А.А., Бочаров Г.А.* Математическое моделирование противовирусного иммунного ответа при вирусном гепатите // В сб. Математические проблемы кибернетики. М.: Наука, 1989. С. 5–70.
3. *Романюха А.А.* Энергетическая цена противоинфекционной защиты организма. Эволюционный подход к анализу данных и моделированию // Тез. докладов. Второй Сибирский конгресс по прикладной и индустриальной математике (ИНПРИМ-96). Новосибирск, 1996. С. 44.
4. *Романюха А.А., Яшин А.И.* Математическая модель возрастных изменений в популяции периферических Т-лимфоцитов // Успехи геронтологии. 2001. Вып. 8. С. 58–69.
5. *Руднев С.Г., Романюха А.А., Яшин А.И.* Факторы, влияющие на развитие Т-системы иммунитета, и задача оптимального распределения ресурсов // Матем. моделирование. 2007 (принято в печать).
6. *Ханин М.А., Дорфман Н.Л., Бухаров И.Б., Левадный В.Г.* Экстремальные принципы в биологии и физиологии. М.: Наука, 1978. 256 с.
7. *Ярилин А.А.* Гомеостатические процессы в иммунной системе. Контроль численности Т-лимфоцитов // Иммунология. 2004. № 5. С. 312–320.
8. *Cohen Stuart J.W., Hazenberg M.D., Hamann D. et al.* The dominant source of CD4 and CD8 T-cell activation in HIV infection is antigenic stimulation // J. Acquir. Immune Defic. Syndr. 2000. V. 25. P. 203–211.
9. *De Boer R.J.* Mathematical models of human T-cell population kinetics // Neth. J. Med. 2002. V. 60. № 7. P. 36–43.
10. *De Boer R.J., Noest A.J.* T cell renewal rates, telomerase, and telomere length shortening // J. Immunol. 1998. V. 160. P. 5832–5837.
11. *Hazenberg M.D., Otto S.A., Van Rossum A.M. et al.* Establishment of the CD4 T-cell pool in healthy children and untreated children infected with HIV-1 // Blood. 2004. V. 104. P. 3513–3519.
12. *McDade T.W.* Life history theory and the immune system: steps toward a human ecological immunology // Yrbk Phys. Anthropol. 2003. V. 46. P. 100–125.
13. *Mocchegiani E., Fülop T., Friguet B., Marcellini F.* Inflammatory/immune responses in ageing: relevant factors and putative agents for intervention // Mech. Ageing Dev. 2006. V. 127. P. 515–516.
14. *Romanyukha A.A., Rudnev S.G., Sidorov I.A.* Energy cost of infection burden: an approach to understanding the dynamics of host-pathogen interactions // J. Theor. Biol. 2006. V. 241. P. 1–13.

15. *Sannikova T.E., Rudnev S.G., Romanyukha A.A., Yashin A.I.* Immune system aging may be affected by HIV infection: mathematical model of immunosenescence // Russ. J. Numer. Anal. Math. Modelling. 2004. V. 19. P. 315–329.
16. *West J.B., Brown J.H.* The origin of allometric scaling laws in biology from genomes to ecosystems: towards a quantitative unifying theory of biological structure and organization // J. Exp. Biol. 2005. V. 208. P. 1575–1592.

СБОРНИК СТАТЕЙ МОЛОДЫХ УЧЕНЫХ факультета ВМиК МГУ, 2007, выпуск № 4, с. 64–72

РАЗРАБОТКА И АНАЛИЗ СХОДИМОСТИ СПЕЦИАЛЬНЫХ ПРОЦЕДУР СИНТЕЗА МЕТРИК В ЗАДАЧЕ ПОСТРОЕНИЯ ВЗАИМОСОГЛАСОВАННЫХ МЕР БЛИЗОСТИ КЛИЕНТОВ И ТОВАРОВ АССОРТИМЕНТА

© 2007 г. А. Л. Кочеихин

kachock@yandex.ru

Кафедра Математических методов прогнозирования

Научный руководитель: **профессор, чл.-корр. РАН К. В. Рудаков**

1. Постановка задачи.

Одной из математических задач, решаемых в рамках CRM (Customer Relationship Management), является задача построения взаимосогласованных мер сходства клиентов и ассортимента. С помощью таких мер можно разделить клиентов на группы в соответствии с их предпочтениями, а также товары в соответствии с их популярностью у определённых групп клиентов.

Рассмотрим формальную математическую постановку задачи. Пусть есть два множества: $A = \{A_1, ..., A_n\}$ - клиенты, $B = \{B_1, ..., B_m\}$ - товары ассортимента. Необходимо построить метрики на этих множествах, исходя из следующих соображений: клиенты считаются близкими, если они приобретают схожие товары, а товары считаются близкими, если их приобретают схожие клиенты. Это приводит к тому, что на множествах A и B возникают пары взаимосогласованных метрик.

Исходными данными являются объёмы закупок клиентами товаров из ассортимента, которые отражаются в матрице объёмов закупок:

$$V = \begin{pmatrix} v_{11} & \dots & v_{1m} \\ \vdots & \vdots & \vdots \\ v_{n1} & \dots & v_{nm} \end{pmatrix}$$

Здесь v_{ij} - объём закупок i-м клиентом j-го товара.

Метрики на множестве клиентов и товаров могут быть полностью определены с помощью матриц:

$$\rho = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \vdots & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix}$$

$$r = \begin{pmatrix} b_{11} & \dots & b_{1m} \\ \vdots & \vdots & \vdots \\ b_{m1} & \dots & b_{mm} \end{pmatrix}$$

a_{ij} - расстояние между i-м и j-м клиентом, а b_{kl} - между k-м и l-м товаром. Соответственно, чем меньше элементы матриц, тем более близкими считаются соответствующие клиенты и товары.

Здесь необходимо сделать уточнение: на самом деле, речь идёт не о метриках, а о полуметриках.

Определение 1. Метрикой на множестве X называется отображение $m : X \times X \to \mathbb{R}$, подчинённое следующим аксиомам:

1. $\forall \quad x, y \in X \, m(x, y) \geq 0$;

2. $m(x,y) = 0 \Leftrightarrow x \equiv y$;
3. $\forall \quad x,y \in X\, m(x,y) = m(y,x)$;
4. $\forall \quad x,y,z \in X\, m(x,y) \le m(x,z) + m(y,z)$.

Полуметрики отличаются от метрик тем, что они подчиняются только первой и третьей аксиомам, т.е. нулевое расстояние может быть и между несовпадающими объектами и не требуется выполнения неравенства треугольника.

В нашей терминологии, a_{ij} может быть равно нулю, даже если $i \neq j$, а также не обязательно $a_{ij} \le a_{it} + a_{tj}$ для любых индексов i,j,t. В дальнейшем мы всё же будем использовать понятие метрики, учитывая данное уточнение.

Введём ещё одно ограничение на матрицы. Будем рассматривать нормированные матрицы, т.е. такие, для которых

$$||\rho|| = \max_{i,j=1...n} |a_{ij}| \le 1 \tag{1}$$

$$||r|| = \max_{i,j=1...m} |b_{ij}| \le 1 \tag{2}$$

Таким образом, $\forall i,j = 1...n \quad 0 \le a_{ij} \le 1$, и $\forall k,l = 1...m \quad 0 \le b_{kl} \le 1$. Таким образом, наиболее удалённые друг от друга объекты находятся на единичном расстоянии.

Обозначим пространство матриц вида $A = (a_{ij})_{n\times n} : \quad 0 \le a_{ij} \le 1$ через $\mathbb{R}_1^{n^2}$, а пространство матриц вида $B = (b_{ij})_{m\times m} : \quad 0 \le b_{kl} \le 1$ через $\mathbb{R}_1^{m^2}$.

Итак, задача состоит в том, чтобы построить пару взаимосогласованных метрик ρ и r на основе матрицы объёмов закупок.

2. Построение метрик.

Предлагается следующий итерационный способ построения метрик. Сначала строится метрика ρ_0 на множестве клиентов A. Её мы будем рассматривать в качестве начального приближения. Нам подходит любая симметричная матрица, элементы которой неотрицательны и не превосходят единицу и на диагонали которой стоят нули. В этом случае выполняются первая и третья аксиомы метрики. Выберем в качестве ρ_0 следующую матрицу:

$$\rho_0 = \begin{pmatrix} 0 & 1 & \dots & 1 \\ 1 & 0 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 0 \end{pmatrix}$$

размера $n \times n$.

Имея метрику ρ_0, построим метрику r_0 на множестве B товаров. Зная теперь r_0 построим метрику ρ_1 на множестве клиентов и т.д. В итоге, возникает следующая цепочка:

$$\rho_0 \quad \longrightarrow \quad r_0 \quad \longrightarrow \quad \rho_1 \quad \longrightarrow \quad r_1 \quad \longrightarrow \quad \rho_2 \quad \longrightarrow \quad \dots \tag{3}$$

По сути, это процесс построения матриц r_s по известным ρ_s и ρ_{s+1} по известным r_s для $s = 0,1,2...$ с помощью определённых преобразований R_1 и R_2:

$$R_1 : \mathbb{R}_1^{n^2} \to \mathbb{R}_1^{m^2} : \; r_s = R_1(\rho_s),\; s = 0,1,2...$$

$$R_2 : \mathbb{R}_1^{m^2} \to \mathbb{R}_1^{n^2} : \; \rho_{s+1} = R_2(r_s),\; s = 0,1,2...$$

Зная эти преобразования, можно определить две функции на множестве матриц $\mathbb{R}_1^{n^2}$ для пересчёта ρ_{s+1} по ρ_s и на множестве $\mathbb{R}_1^{m^2}$ для пересчёта r_{s+1} по r_s для $s = 0,1,2...$:

$$f : \mathbb{R}_1^{n^2} \to \mathbb{R}_1^{n^2} : \rho_{s+1} = f(\rho_s)$$

$$g : \mathbb{R}_1^{m^2} \to \mathbb{R}_1^{m^2} : r_{s+1} = g(r_s)$$

Таким образом, итерационный процесс (3) распадается на два процесса:

$$\rho_0 \xrightarrow{f} \rho_1 \xrightarrow{f} \rho_2 \xrightarrow{f} \rho_3 \quad \dots \tag{4}$$

$$r_0 \xrightarrow{g} r_1 \xrightarrow{g} r_2 \xrightarrow{g} r_3 \quad \dots \tag{5}$$

В случае сходимости результатами каждого из данных итерационных процессов будут соответствующие матрицы расстояний между клиентами и товарами ассортимента.

Теперь необходимо получить функции f и g, а также исследовать вопрос об условиях сходимости процессов (4) и (5).

Первый вопрос решается легко: зная преобразования R_1 и R_2 из процесса (3) функции f и g можно определить следующим образом:

$$\rho_{s+1} = f(\rho_s) = R_2(R_1(\rho_s))$$

$$r_{s+1} = g(r_s) = R_1(R_2(r_s))$$

Т.е., $f = R_2 \circ R_1$, а $g = R_1 \circ R_2$.

Таким образом, необходимо построить преобразование R_1, с помощью которого, зная элементы a_{ij}^s матрицы ρ_s, можно найти элементы b_{ij}^s матрицы r_s, и преобразование R_2, с помощью которого, зная элементы b_{ij}^s матрицы r_s, можно найти элементы a_{ij}^{s+1} матрицы ρ_{s+1}.

Предлагается следующая формула, описывающая преобразование R_1:

$$a_{ij}^{s+1} = \frac{\sum\limits_{k,l=1}^{m} b_{kl}^s v_{ik} v_{jl}}{1 + \sum\limits_{k,l=1}^{m} v_{ik} v_{jl}}, \quad i,j = 1...n,\ i \neq j,\ s = 0,1,2... \tag{6}$$

$$a_{ij}^{s+1} = 0, \quad i,j = 1...n,\ i = j,\ s = 0,1,2...$$

Чтобы знаменатель не обращался в нуль, к соответствующей сумме прибавляется единица.

Аналогичным образом предлагается формула для вычисления расстояния b_{kl} между k-м и l-м товаром, описывающая преобразование R_2:

$$b_{kl}^s = \frac{\sum\limits_{i,j=1}^{n} a_{ij}^s v_{ik} v_{jl}}{1 + \sum\limits_{i,j=1}^{n} v_{ik} v_{jl}}, \quad k,l = 1...m,\ k \neq l,\ s = 0,1,2... \tag{7}$$

$$b_{kl}^s = 0, \quad k,l = 1...m,\ k = l,\ s = 0,1,2...$$

Замечание. Из приведённых формул видно, что при $0 \le a_{ij}^s \le 1$ и $0 \le b_{kl}^s \le 1$ верно, что $0 \le a_{ij}^{s+1} \le 1$ и $0 \le b_{kl}^{s+1} \le 1$, $s = 0,1,2...$ По этой причине и были введены ограничения (1) и (2) на матрицы ρ и r.

Эти формулы в полной мере соответствуют постановке задачи: клиенты считаются близкими, если они приобретают схожие товары, а товары считаются близкими, если их приобретают схожие клиенты.

3. Сходимость итерационного процесса синтеза метрик.

Напомним, что для построения метрик на множестве клиентов и товаров используются два итерационных процесса:

$$\rho_0 \xrightarrow{f} \rho_1 \xrightarrow{f} \rho_2 \xrightarrow{f} \rho_3 \quad \dots \tag{4}$$

$$r_0 \xrightarrow{g} r_1 \xrightarrow{g} r_2 \xrightarrow{g} r_3 \quad \dots \tag{5}$$

Функции f и g определяются следующим образом:

$$f : \mathbb{R}_1^{n^2} \to \mathbb{R}_1^{n^2} : \rho_{s+1} = f(\rho_s)$$

$$f = R_2 \circ R_1$$

$$g : \mathbb{R}_1^{m^2} \to \mathbb{R}_1^{m^2} : r_{s+1} = g(r_s)$$

$$g = R_1 \circ R_2$$

Преобразования R_1 и R_2 описываются формулами (6) и (7).

Таким образом, выражения для фнукций f и g имеют следующий вид:

$$f : a_{ij}^{s+1} = \frac{\sum\limits_{k,l=1}^{m} b_{kl}^s v_{ik} v_{jl}}{1 + \sum\limits_{k,l=1}^{m} v_{ik} v_{jl}} = \{\text{учитывая (7)}\} = \frac{\sum\limits_{k,l=1}^{m} \left\{ v_{ik} v_{jl} \cdot \left(\frac{\sum\limits_{p,q=1}^{n} a_{pq}^s v_{pk} v_{ql}}{1 + \sum\limits_{p,q=1}^{n} v_{pk} v_{ql}} \right) \right\}}{1 + \sum\limits_{k,l=1}^{m} v_{ik} v_{jl}}, \; s = 0, 1, 2... \tag{8}$$

$$g : b_{kl}^s = \frac{\sum\limits_{i,j=1}^{n} a_{ij}^s v_{ik} v_{jl}}{1 + \sum\limits_{i,j=1}^{n} v_{ik} v_{jl}} = \{\text{учитывая (8)}\} = \frac{\sum\limits_{i,j=1}^{n} \left\{ v_{ik} v_{jl} \cdot \left(\frac{\sum\limits_{p,q=1}^{m} b_{pq}^s v_{ip} v_{jq}}{1 + \sum\limits_{p,q=1}^{m} v_{ip} v_{jq}} \right) \right\}}{1 + \sum\limits_{i,j=1}^{n} v_{ik} v_{jl}}, \; s = 1, 2... \tag{9}$$

$$b_{kl}^0 = \frac{\sum\limits_{i,j=1}^{n} a_{ij}^0 v_{ik} v_{jl}}{1 + \sum\limits_{i,j=1}^{n} v_{ik} v_{jl}} \tag{10}$$

Для исследования вопроса о сходимости итерационных процессов синтеза метрик воспользуемся принципом сжимающих отображений.

Определение 2. Пусть R - метрическое пространство. Отображение $F : R \to R$ называется сжимающим, если существует такое число $\lambda < 1$, что для любых $x, y \in R$ выполняется:

$$\mu(F(x), F(y)) \leq \lambda \mu(x, y)$$

Здесь μ - метрика пространства R.

Определение 3. Точка x называется неподвижной точкой отображения F, если $F(x) = x$.

Теорема (принцип сжимающих отображений). *Всякое сжимающее отображение, определённое в полном метрическом пространстве R, имеет одну и только одну неподвижную точку.*

Применим данную теорему к итерационным процессам (4) и (5).
Рассмотрим процесс (4):

$$f : \mathbb{R}_1^{n^2} \to \mathbb{R}_1^{n^2} : \rho_{s+1} = f(\rho_s)$$

Если мы покажем, что пространство $\mathbb{R}_1^{n^2}$ является полным метрическим пространством и покажем, что f является сжимающим отображением на $\mathbb{R}_1^{n^2}$, то, согласно принципу сжимающих отображений, $\exists \rho_N : f(\rho_N) = \rho_N$, т.е. в итерационном процессе (4) наступит стабилизация, а матрица ρ_N будет искомой матрицей расстояний между клиентами.
Рассмотрим процесс (5):

$$g : \mathbb{R}_1^{m^2} \to \mathbb{R}_1^{m^2} : r_{s+1} = g(r_s)$$

Если мы покажем, что пространство $\mathbb{R}_1^{m^2}$ является полным метрическим пространством и покажем, что g является сжимающим отображением на $\mathbb{R}_1^{m^2}$, то, согласно принципу сжимающих отображений, $\exists r_M : g(r_M) = r_M$, т.е. в итерационном процессе (5) наступит стабилизация, а матрица r_M будет искомой матрицей расстояний между товарами ассортимента.

Начнём с доказательства полноты $\mathbb{R}_1^{n^2}$ и $\mathbb{R}_1^{m^2}$.

Определение 4. Последовательность точек $\{x_n\}$ метрического пространства R называется фундаментальной, если $\forall \varepsilon > 0 \; \exists N_\varepsilon > 0 : \forall n^{'} > N_\varepsilon, \, n^{''} > N_\varepsilon \;\; \mu(x_{n'}, x_{n''}) < \varepsilon$. Здесь μ - метрика пространства R.

Определение 5. Если в пространстве R любая фундаментальная последовательность сходится к элементу этого пространства, то пространство называется полным.

Пространства $\mathbb{R}_1^{n^2}$ и $\mathbb{R}_1^{m^2}$ являются замкнутыми подпространствами пространств $\mathbb{R}^{n^2}$ и $\mathbb{R}^{m^2}$ действительных матриц размера $n \times n$ и $m \times m$ соответственно. Из функционального анализа известно, что замкнутое подпространство полного метрического пространства само является полным метрическим пространством, поэтому, если мы докажем полноту $\mathbb{R}^{n^2}$ и $\mathbb{R}^{m^2}$, то отсюда будет следовать полнота $\mathbb{R}_1^{n^2}$ и $\mathbb{R}_1^{m^2}$.

Покажем полноту $\mathbb{R}^{n^2}$:
Пусть $\{X^{(p)}\}$ - фундаментальная последовательность точек из $\mathbb{R}^{n^2}$.
Тогда:

$$\forall \varepsilon > 0 \quad \exists N(\varepsilon) > 0 : \quad \forall p, q > N(\varepsilon) \quad \mu(X^{(p)}, X^{(q)}) < \varepsilon$$

В качестве метрики μ пространства $\mathbb{R}^{n^2}$ возьмём следующую:

$$\forall X = (x_{ij})_{n\times n}, Y = (y_{ij})_{n\times n} \in \mathbb{R}^{n^2} \quad \text{положим} \quad \mu(X,Y) = \max_{i,j=1\ldots n} |x_{ij} - y_{ij}|$$

Тогда:

$$\mu(X^{(p)}, X^{(q)}) = \max_{i,j=1\ldots n} |x_{ij}^{(p)} - x_{ij}^{(q)}| < \varepsilon \quad \forall p, q > N(\varepsilon) \quad \Rightarrow$$

$$\Rightarrow \forall i, j = 1\ldots n \quad |x_{ij}^{(p)} - x_{ij}^{(q)}| < \varepsilon \quad \forall p, q > N(\varepsilon)$$

Получаем, что последовательность $x_{ij}^{(p)}$, состоящая из действительных чисел, является фундаментальной. Из курса математического анализа известна полнота пространства действительных чисел $\mathbb{R}$, поэтому последовательность $x_{ij}^{(p)}$ сходится и $\exists\, x_{ij} = \lim\limits_{p\to\infty} x_{ij}^{(p)}$. Положим $X = (x_{ij})_{n\times n}$, тогда очевидно, что $\lim\limits_{p\to\infty} X^{(p)} = X$, т.е. фундаментальная последовательность $\{X^{(p)}\}$ сходится, откуда, в силу произвольности выбора $\{X^{(p)}\}$, следует полнота пространства $\mathbb{R}^{n^2}$.

Полнота $\mathbb{R}^{m^2}$ доказывается полностью аналогично.

Таким образом, доказана полнота $\mathbb{R}_1^{n^2}$ и $\mathbb{R}_1^{m^2}$.

Теперь рассмотрим функции f и g.

1. Согласно определению, f — сжимающее отображение на $\mathbb{R}_1^{n^2}$, если существует такое число $\lambda < 1$, что для любых $A', A'' \in \mathbb{R}_1^{n^2}$ выполняется:

$$\mu(f(A'), f(A'')) \le \lambda\mu(A', A'') \tag{11}$$

Положим $f(A') = \widetilde{A}' = (\widetilde{a}'_{ij})_{n\times n}$, $f(A'') = \widetilde{A}'' = (\widetilde{a}''_{ij})_{n\times n}$, тогда, учитывая вид метрики μ, условие (11) примет следующий вид:

$$\max_{i,j=1\dots n} |\widetilde{a}'_{ij} - \widetilde{a}''_{ij}| \le \lambda \cdot \max_{i,j=1\dots n} |a'_{ij} - a''_{ij}| \tag{12}$$

Учитывая выражение для f (8), можно записать следующие формулы пересчёта:

$$\widetilde{a}'_{ij} = \frac{\sum\limits_{k,l=1}^{m} \left\{ v_{ik}v_{jl} \cdot \left(\frac{\sum\limits_{p,q=1}^{n} a'_{pq}v_{pk}v_{ql}}{1+\sum\limits_{p,q=1}^{n} v_{pk}v_{ql}} \right) \right\}}{1 + \sum\limits_{k,l=1}^{m} v_{ik}v_{jl}} \tag{13}$$

$$\widetilde{a}''_{ij} = \frac{\sum\limits_{k,l=1}^{m} \left\{ v_{ik}v_{jl} \cdot \left(\frac{\sum\limits_{p,q=1}^{n} a''_{pq}v_{pk}v_{ql}}{1+\sum\limits_{p,q=1}^{n} v_{pk}v_{ql}} \right) \right\}}{1 + \sum\limits_{k,l=1}^{m} v_{ik}v_{jl}} \tag{14}$$

Условие (12) при этом имеет вид:

$$\max_{i,j=1\dots n} \left| \frac{\sum\limits_{k,l=1}^{m} \left\{ v_{ik}v_{jl} \cdot \left(\frac{\sum\limits_{p,q=1}^{n} a'_{pq}v_{pk}v_{ql}}{1+\sum\limits_{p,q=1}^{n} v_{pk}v_{ql}} \right) \right\}}{1 + \sum\limits_{k,l=1}^{m} v_{ik}v_{jl}} - \frac{\sum\limits_{k,l=1}^{m} \left\{ v_{ik}v_{jl} \cdot \left(\frac{\sum\limits_{p,q=1}^{n} a''_{pq}v_{pk}v_{ql}}{1+\sum\limits_{p,q=1}^{n} v_{pk}v_{ql}} \right) \right\}}{1 + \sum\limits_{k,l=1}^{m} v_{ik}v_{jl}} \right| =$$

$$= \max_{i,j=1\dots n} \left| \frac{\sum\limits_{k,l=1}^{m} \left\{ v_{ik}v_{jl} \cdot \left(\frac{\sum\limits_{p,q=1}^{n} v_{pk}v_{ql}(a'_{pq}-a''_{pq})}{1+\sum\limits_{p,q=1}^{n} v_{pk}v_{ql}} \right) \right\}}{1 + \sum\limits_{k,l=1}^{m} v_{ik}v_{jl}} \right| \le \lambda \cdot \max_{i,j=1\dots n} |a'_{ij} - a''_{ij}| \tag{15}$$

Очевидно, что:

$$\max_{i,j=1\dots n}\left|\frac{\sum\limits_{k,l=1}^{m}\left\{v_{ik}v_{jl}\cdot\left(\frac{\sum\limits_{p,q=1}^{n}v_{pk}v_{ql}(a'_{pq}-a''_{pq})}{1+\sum\limits_{p,q=1}^{n}v_{pk}v_{ql}}\right)\right\}}{1+\sum\limits_{k,l=1}^{m}v_{ik}v_{jl}}\right|\le\max_{i,j=1\dots n}\left\{\frac{\sum\limits_{k,l=1}^{m}\left\{v_{ik}v_{jl}\cdot\left(\frac{\sum\limits_{p,q=1}^{n}v_{pk}v_{ql}|a'_{pq}-a''_{pq}|}{1+\sum\limits_{p,q=1}^{n}v_{pk}v_{ql}}\right)\right\}}{1+\sum\limits_{k,l=1}^{m}v_{ik}v_{jl}}\right\}$$

Поэтому (15) заведомо выполняется при выполнении неравенства:

$$\max_{i,j=1\dots n}\left|\frac{\sum\limits_{k,l=1}^{m}\left\{v_{ik}v_{jl}\cdot\left(\frac{\sum\limits_{p,q=1}^{n}v_{pk}v_{ql}|a'_{pq}-a''_{pq}|}{1+\sum\limits_{p,q=1}^{n}v_{pk}v_{ql}}\right)\right\}}{1+\sum\limits_{k,l=1}^{m}v_{ik}v_{jl}}\right|\le\lambda\cdot\max_{i,j=1\dots n}|a'_{ij}-a''_{ij}| \tag{16}$$

Введём обозначения:

$$\frac{\sum\limits_{p,q=1}^{n}\left\{v_{pk}v_{ql}|a'_{pq}-a''_{pq}|\right\}}{1+\sum\limits_{p,q=1}^{n}v_{pk}v_{ql}}=M_{kl}$$

$$\frac{\sum\limits_{k,l=1}^{m}\{v_{ik}v_{jl}\cdot M_{kl}\}}{1+\sum\limits_{k,l=1}^{m}v_{ik}v_{jl}}=T_{ij}$$

Тогда неравенство (16) примет вид:

$$\max_{i,j=1\dots n}T_{ij}\le\lambda\cdot\max_{i,j=1\dots n}|a'_{ij}-a''_{ij}| \tag{17}$$

Имеем:

$$T_{ij}\le\frac{\sum\limits_{k,l=1}^{m}\left\{v_{ik}v_{jl}\cdot\max\limits_{k,l=1\dots m}M_{kl}\right\}}{1+\sum\limits_{k,l=1}^{m}v_{ik}v_{jl}} \tag{18}$$

$$M_{kl}\le\frac{\sum\limits_{p,q=1}^{n}\left\{v_{pk}v_{ql}\max\limits_{p,q=1\dots n}|a'_{pq}-a''_{pq}|\right\}}{1+\sum\limits_{p,q=1}^{n}v_{pk}v_{ql}}=\frac{C\cdot\sum\limits_{p,q=1}^{n}v_{pk}v_{ql}}{1+\sum\limits_{p,q=1}^{n}v_{pk}v_{ql}}\quad,\ \text{где}\ C=\max_{p,q=1\dots n}|a'_{pq}-a''_{pq}|$$

Рассмотрим функцию $h(x) = \frac{cx}{1+x}$ при $x > 0, c \geq 0$. Производная $h'(x) = \frac{c}{(1+x)^2} \geq 0$ при $x > 0$, таким образом, при $x > 0$ функция возрастает.

$M_{kl} = h\left(\sum\limits_{p,q=1}^{n} v_{pk}v_{ql}\right)$ при $c = C$, поэтому

$$\max_{k,l=1\ldots m} M_{kl} = \frac{C \cdot C_1}{1 + C_1} = K_1 \quad , \text{ где } C_1 = \max_{k,l=1\ldots m}\left\{\sum_{p,q=1}^{n} v_{pk}v_{ql}\right\} \tag{19}$$

Учитывая (19), неравенство (18) принимает вид:

$$T_{ij} \leq \frac{K_1 \cdot \sum\limits_{k,l=1}^{m} v_{ik}v_{jl}}{1 + \sum\limits_{k,l=1}^{m} v_{ik}v_{jl}}$$

$T_{ij} = h\left(\sum\limits_{k,l=1}^{m} v_{ik}v_{jl}\right)$ при $c = K_1$, поэтому

$$\max_{i,j=1\ldots n} T_{ij} = \frac{K_1 \cdot C_2}{1 + C_2} \quad , \text{ где } C_2 = \max_{i,j=1\ldots n}\left\{\sum_{k,l=1}^{m} v_{ik}v_{jl}\right\}$$

Таким образом, неравенство (17) равносильно следующему:

$$\frac{K_1 \cdot C_2}{1 + C_2} \leq \lambda C \quad \Longleftrightarrow \quad \frac{\frac{CC_1}{1+C_1} \cdot C_2}{1 + C_2} \leq \lambda C \quad \Longleftrightarrow$$

$$\Longleftrightarrow \frac{C_1 C_2}{(1 + C_1)(1 + C_2)} \leq \lambda \tag{20}$$

Т.к. $\frac{C_1C_2}{(1+C_1)(1+C_2)} \leq 1$, то в качестве λ можно взять любое число из отрезка $\left[\frac{C_1C_2}{(1+C_1)(1+C_2)} \quad ; 1\right)$:

$$\lambda \in \left[\frac{C_1 C_2}{(1 + C_1)(1 + C_2)} \quad ; 1\right) \tag{21}$$

Таким образом, существует такое число $\lambda < 1$, что для любых $A', A'' \in \mathbb{R}_1^{n^2}$ выполняется:

$$\mu(f(A'), f(A'')) \leq \lambda\mu(A', A'')$$

Поэтому функция f, определённая в (8), является сжимающим отображением на $\mathbb{R}_1^{n^2}$.

2. Аналогично, согласно определению, g — сжимающее отображение на $\mathbb{R}_1^{m^2}$, если существует такое число $\lambda < 1$, что для любых $B', B'' \in \mathbb{R}_1^{m^2}$ выполняется:

$$\mu(g(B'), g(B'')) \leq \lambda\mu(B', B'') \tag{22}$$

Положим $g(B^{'}) = \widetilde{B}^{'} = (\widetilde{b}^{'}_{kl})_{m\times m}$, $g(B^{''}) = \widetilde{B}^{''} = (\widetilde{b}^{''}_{kl})_{m\times m}$, тогда, учитывая вид метрики μ, условие (22) примет следующий вид:

$$\max_{k,l=1\ldots m} |\widetilde{b}^{'}_{kl} - \widetilde{b}^{''}_{kl}| \le \lambda \cdot \max_{k,l=1\ldots m} |b^{'}_{kl} - b^{''}_{kl}| \tag{23}$$

Учитывая выражение для g (9), можно записать следующие формулы пересчёта:

$$\widetilde{b}^{'}_{kl} = \frac{\sum\limits_{i,j=1}^{n} \left\{ v_{ik}v_{jl} \cdot \left(\frac{\sum\limits_{p,q=1}^{m} b^{'}_{pq}v_{ip}v_{jq}}{1+\sum\limits_{p,q=1}^{m} v_{ip}v_{jq}} \right) \right\}}{1 + \sum\limits_{i,j=1}^{n} v_{ik}v_{jl}} \tag{24}$$

$$\widetilde{b}^{''}_{ij} = \frac{\sum\limits_{i,j=1}^{n} \left\{ v_{ik}v_{jl} \cdot \left(\frac{\sum\limits_{p,q=1}^{m} b^{''}_{pq}v_{ip}v_{jq}}{1+\sum\limits_{p,q=1}^{m} v_{ip}v_{jq}} \right) \right\}}{1 + \sum\limits_{i,j=1}^{n} v_{ik}v_{jl}} \tag{25}$$

Доказательство выполнения условия (23) полностью аналогично доказательству выполнения условия (12) в пункте 1.

В итоге, получаем, что функции f и g, определяемые выражениями (8), (9) и (10) являются сжимающими отображениями на $\mathbb{R}_1^{n^2}$ и $\mathbb{R}_2^{m^2}$ соответственно, поэтому итерационные процессы синтеза метрик (4) и (5) сходятся и $\exists \rho_N : f(\rho_N) = \rho_N, \quad \exists r_M : g(r_M) = r_M$.

ρ_N и r_M — матрицы, задающие искомые взаимосогласованные метрики на множестве клиентов и товаров ассортимента.

СПИСОК ЛИТЕРАТУРЫ

1. *Воронцов К.В., Рудаков К.В., Лексин В.А.* Технология выявления взаимосогласованных структур сходства пользователей и ресурсов.
2. *Колмогоров А.Н., Фомин С.В.* Элементы теории функций и функционального анализа: Учебник для вузов. — 6-е изд., испр. — М.: Наука. Гл. ред. физ.-мат. лит., 1989.
3. *Ходак Е.Е.* О пользе CRM, или клиент — это самое главное достояние, бесценное сокровище каждой компании.

СБОРНИК СТАТЕЙ МОЛОДЫХ УЧЕНЫХ факультета ВМиК МГУ, 2007, выпуск № 4, с. 73–81

УДК 517.95

НОВЫЙ МЕТОД ОБУЧЕНИЯ БАЙЕСОВСКОЙ ЛОГИСТИЧЕСКОЙ РЕГРЕССИИ С ИСПОЛЬЗОВАНИЕМ ЛАПЛАСОВСКОГО РЕГУЛЯРИЗАТОРА

© 2007 г. О. В. Курчин

4education@mail.ru

Кафедра Математических методов прогнозирования

Введение.

В данной статье рассматривается новый метод обучения Байесовской разреженной логистической регрессии с использованием лапласовского априорного распределения. В основе метода лежит разреженная логистическая регрессия (SLogReg) - подход, предложенный (Shevade & Keerthi, 2003), улучшенная проведением усреднения по априорному распределению, способом описанным (Cawley и Talbot, 2006). Суть самого подхода SLogReg состоит в использовании регуляризатора на основе нормы L1 (Tikhonov & Arsenin, 1977), возникающего из предположения о лапласовском априорном распределении параметров модели (Williams, 1995), используемом для определения разреженного подмножества наиболее значимых признаков. При этом получаемые обобщающая способность классификатора и степень разреженности значительным образом зависят от значения параметра регуляризатора, который вообще говоря неизвестен и потому, для достижения оптимального результата, доложен быть тщательным образом подобран. Подбор данного параметра зачастую происходит путем длительного поиска с помощью процедуры скользящего контроля. В качестве альтернативы этому способу, может быть использован Байесовский подход, заключающийся в данном случае в усреднении по параметру регуляризатора с помощью несобственного распределения Джеффри, как это описано у (Buntine & Weigend, 1991). Полученный классификатор без параметров (BLogReg) значительно проще в настройке и использовании при сравнительном с SLogReg качестве работы, и при этом на два-три порядка быстрее. Выигрыш в скорости работы достигается за счет того, что больше не требуется проведение затратного с вычислительной точки зрения выбора оптимальной модели путем подбора соответствующего значения параметра регуляризатора. Существующий метод настройки данного классификатора использует в качестве метода оптимизации покоординатный спуск, тогда как в новом разработанном методе предлагается провести аппроксимацию критерия качества его непрерывным аналогом, и затем воспользоваться методом оптимизации второго порядка. Будучи сравнимым по точности с существующим методом (Cawley и Talbot,2006), разработанный метод работает значительно быстрее.

В первой части данной статьи будет описан подход, используемый для обучения разреженной логистической регрессии, во второй будет показано, как проводится усреднение по параметру регуляризатора, в третьей будут описаны существующий (Cawley и Talbot,2006) и разработанный методы обучения Байесовской логистической регрессии с использованием лапласовского регуляризатора и в четвертой - полученные результаты в сравнении с существующими методами-аналогами.

1. Разреженная логистическая регрессия.

Рассмотрим задачу распознавания, в которой требуется построить решающее правило, позволяющее разделять объекты принадлежащие к одному из двух разных классов, на основе имеющегося множества из n прецедентов (обучающей выборки), $\mathcal{D} = (\mathcal{T}, \mathcal{X}) = \{(t_i, \vec{x}_i)\}_{i=1}^n, \vec{x}_i \in \mathbb{R}^d, t_i \in \{-1, +1\}$. Логистическая регрессия является одним из классических подходов к решению данной задачи. Его суть заключается в определении апостериорной вероятности принадлежности объекта к тому или иному классу с помощью линейной комбинации признаков объекта,

$$p(1|\vec{x}) = \frac{1}{1 + \exp\{-y(\vec{x})\}}$$

где

$$y(\vec{x}_i) = \sum_{j=1}^{d} w_j x_{ij} + w_0.$$

Параметры модели логистической регрессии, $\vec{w} = (w_0, w_1, \ldots, w_d)$, могут быть найдены путем максимизации *правдоподобия* обучающей выборки, или, что тоже самое, минимизации минус логарифма правдоподобия. Предполагая, что $\mathcal{D}$ представляет из себя множество независимых одинаково распределенных (н.о.р.) объектов выборки из распределения Бернулли , минус логарифм правдоподобия принимает вид,

$$L(\mathcal{T}|\mathcal{X}, \vec{w}) = \sum_{i=1}^{n} \log\left\{1 + \exp\left(-t_i y(\vec{x}_i)\right)\right\}.$$

Основным недостатком является тот факт, что вообще говоря, ни один из параметров полученной модели $\vec{w}$ не становится равен нулю. Мы же, в идеале, хотим получить модель, использующую небольшое число наиболее информативных признаков, при этом ожидается, что оставшиеся признаки будут из нее исключены (разреженная модель). Для получения такого результата можно использовать регуляризатор, как дополнительное слагаемое к минус логарифму правдоподобия (Williams, 1995), являющийся по своей сути отражением нашего предположения о лапласовском априорном распределении над $\vec{w}$. Это дает нам следующий вид критерия оптимизации,

$$F = L(\mathcal{T}|\mathcal{X}, \vec{w}) + \lambda R(\vec{w}) \quad \text{,где} \quad R(\vec{w}) = \sum_{i=1}^{d} |w_i| \tag{1}$$

а λ - параметр регуляризации, контролирующий компромисс между точностью и разреженностью итоговой модели. Стоит отметить, что обычно параметр сдвига w_0 остается нерегуляризованным. В точке минимума F, частные производные F по параметрам модели обращаются в ноль, то есть

$$\left|\frac{\partial L(\mathcal{T}|\mathcal{X}, \vec{w})}{\partial w_i}\right| = \lambda \quad \text{,при } |w_i| > 0$$
$$\left|\frac{\partial L(\mathcal{T}|\mathcal{X}, \vec{w})}{\partial w_i}\right| < \lambda \quad \text{,при } |w_i| = 0.$$

Это означает, что если отношение приращения минус логарифма правдоподобия к сдвигу параметра модели , w_i, становится меньше λ, то этот параметр обращается точно в ноль и соответствующий признак можно исключить из модели. Основными недостатками данного подхода являются невозможность проведения непрерывной оптимизации из-за того, что частные производные критерия разрывны, а также необходимость реализации большого количества итераций скользящего контроля для определения подходящего значения параметра регуляризации λ. Ниже будет рассмотрено одно из возможных решений обеих данных проблем с помощью усреднения параметра регуляризации λ и аппроксимации критерия гладкой функцией.

2. Байесовская регуляризация.

В этой части будет показано каким именно образом проводится усреднение по параметру регуляризации, в соответствии с подходом, предложенным (Buntine & Weigend, 1991) и (Williams, 1995).

Минимизация (1) имеет вполне очевидную интерпретацию в рамках Байесовского подхода. Если записать апостериорное распределение для параметров модели $\vec{w}$ следующим образом

$$p(\vec{w}|\mathcal{D}, \lambda) \propto p(\mathcal{D}|\vec{w}) p(\vec{w}|\lambda).$$

тогда F - минус логарифм правдоподобия - является апостериорной плотностью распределения с точностью до аддитивной константы, а априорное распределение для $\vec{w}$ может быть представлено в виде произведения одномерных плотностей распределения Лапласа:

$$p(\vec{w}|\lambda) = \left(\frac{\lambda}{2}\right)^N \exp\{-\lambda R(\vec{w})\} = \prod_{i=1}^{N} \frac{\lambda}{2} \exp\{-\lambda|w_i|\}, \tag{2}$$

где N количество активных (ненулевых) параметров модели. Поскольку значение параметра регуляризации λ не известно, то предлагается провести усреднение по нему, используя подход (Buntine, 1991; Williams, 1995),

$$p(\vec{w}) = \int p(\vec{w}|\lambda)p(\lambda)d\lambda.$$

Поскольку λ по сути своей является параметром масштаба, то наиболее подходящим для него будет несобственное априорное распределение Джеффри, $p(\lambda) \propto 1/\lambda$, дающее равномерное распределение на всей числовой прямой для $\log\lambda$. Подставляя данное распределение в (2), с учетом того, что λ строго положительно,имеем

$$p(\vec{w}) = \frac{1}{2^N} \int_0^\infty \lambda^{N-1} \exp\{-\lambda R(\vec{w})\}d\lambda.$$

Используя Гамма интеграл, $\int_0^\infty x^{\nu-1}e^{-\mu x}dx = \frac{\Gamma(\nu)}{\mu^\nu}$, получим

$$p(\vec{w}) = \frac{1}{2^N}\frac{\Gamma(N)}{R(\vec{w})^N} \quad \Longrightarrow \quad -\log p(\vec{w}) \propto N\log R(\vec{w}),$$

что дает нам следующий оптимизационный критерий для разреженной Байесовской логистической регрессии с лапласовским регуляризатором ($N\log R(\vec{w})$ предполагается равным 0, при $N = 0$),

$$Q = L(\mathcal{T}|\mathcal{X}, \vec{w}) + N\log R(\vec{w}), \tag{3}$$

при этом параметр регуляризации отсутствует. Для более точного теоретического обоснования см. (Williams, 1995).

3.Минимизация Байесовского оптимизационного критерия

Четкий подход к оптимизации

Чтобы получить процедуру оптимизации Байесовского критерия (3) для начала рассмотрим случай с заранее известным значением параметра регуляризации (без его усреднения) (1). Эффективный алгоритм, предложенный (Shevade & Keerthi, 2003) позволяет найти минимум критерия (1) оптимизируя его последовательно по каждому параметру с помощью метода Ньютона. При этом, из-за разрывной первой производной, следует отдельно рассматривать случай перехода через ноль. Для этого производится определение границ для параметра, w_i, как верхней так и нижней (H и L соответственно) таких, что полученный интервал не содержит 0, возможно за исключением границы. Эти границы могут быть вычислены с помощью значения градиента F по w_i в текущем значении и в окрестности нуля (слева и справа), как показано в Таблице 1.

Таблица 1. Случаи, которые необходимо учесть при проведении оптимизации F по w_i чтобы обойти возникновение разрывов у частной производной.

Случай	w_i	$\frac{\partial F}{\partial w_i}\Big\vert_{w_i}$	$\frac{\partial F}{\partial w_i}\Big\vert_{0^-}$	$\frac{\partial F}{\partial w_i}\Big\vert_{0^+}$	L	H
1	0	—	<0	<0	0	$+\infty$
2	0	—	>0	>0	0	$-\infty$
3	<0	>0	—	—	$-\infty$	w_i
4	>0	<0	—	—	w_i	$+\infty$
5	<0	<0	>0	—	w_i	0
6	>0	>0	—	<0	0	w_i
7	<0	<0	—	>0	0	$+\infty$
8	>0	>0	<0	—	$-\infty$	0
9	<0	<0	≤ 0	≥ 0	0	0
10	>0	>0	≤ 0	≥ 0	0	0

На каждой итерации процедуры минимизации следует выбрать параметр, по которому в данный момент будет проводиться оптимизация, при этом наиболее предпочтительными являются параметры с наибольшим значением градиента. Для увеличения общей производительности сначала учитываются только активные параметры (значения которых не равны нулю), а неактивные параметры учитываются только если не осталось активных с ненулевым градиентом. Вообще говоря, данная итерационная процедура не позволяет получить градиент *четко* равный нулю, поэтому на практике для оптимизации используются только те параметры, значения градиента которых превышают заранее установленный порог τ. Алгоритм прекращает свою работу, когда не остается ни одного такого параметра. Для детального изучения данного алгоритма см. (Shevade & Keerthi, 2003). В указанной работе подробно показано, что функционал качества для разреженной логистической регрессии с лапласовским регуляризатором может быть итерационно минимизирован с помощью изменения на каждом шаге всего лишь одного параметра. Стоит отметить, что сам регуляризатор не является всюду гладким, поскольку его первая производная является разрывной в $w_i = 0,\ \forall i \in \{1, 2, \ldots, N\}$, хотя в других точках он все-такие остается гладким. Эти свойства вытекают из вида его первой и второй производных,

$$\frac{\partial}{\partial w_i} \log R(\vec{w}) = \frac{w_i}{|w_i|} \frac{1}{R(\vec{w})}$$
$$\frac{\partial^2}{\partial w_i^2} \log R(\vec{w}) = -\frac{1}{R(\vec{w})^2}.$$

Таким образом наш критерий, использующий регуляризатор, (3) может быть минимизирован небольшой модификацией существующего алгоритма обучения для разреженной логистической регрессии. Дифференцируя исходный и модифицированный критерии (1,3), получим

$$\nabla F = \nabla L(\mathcal{T}|\mathcal{X}, \vec{w}) + \lambda \nabla R(\vec{w})$$
$$\nabla Q = \nabla L(\mathcal{T}|\mathcal{X}, \vec{w}) + \tilde{\lambda} \nabla R(\vec{w})$$

где

$$1/\tilde{\lambda} = \frac{1}{N} \sum_{i=1}^{N} |w_i|. \tag{4}$$

С точки зрения градиента, минимизация Q по сути эквивалентна минимизации F, в предположении, что параметр регуляризации, λ, постоянно пересчитывается в соответствии с (4) после каждого изменения вектора параметров модели, $\vec{w}$ (Williams, 1995).

Быстрый нечеткий подход к оптимизации

Как было отмечено выше, из-за разрывности Байесовского критерия (3) мы не можем напрямую воспользоваться эффективными методами оптимизации, такими как метод Ньютона

второго порядка. Тем не менее мы можем заменить разрывную функцию $N \log R(\vec{w})$ ее непрерывной аппроксимацией. Напомним, что N это число ненулевых параметров модели, то есть каждый раз, обнуляя параметр модели, мы должны уменьшить N на один.

Теперь введем следующую гладкую аппроксимацию для N

$$\hat{N} = n - \sum_{i=1}^{n} \exp\left(-\frac{w_i^2}{2\sigma^2}\right),$$

где $\sigma > 0$ - некоторый положительный коэффициент нечеткости. Несложно видеть, что коэффициент $\hat{N}$ равен 0, если все параметры модели равны 0 и равен N, если все N параметров являются достаточно большими числами, а остальные равны нулю.

Несложно видеть, что минимум Байесовского критерия (3) лежит в том же гипероктанте $\mathcal{H}_{ML}$, что и максимум правдоподобия $\vec{w}_{ML}$. Таким образом, мы можем использовать условный оптимизационный метод Ньютона для нахождения минимума критерия в гипероктанте $\mathcal{H}_{ML}$. Отметим, что $\hat{N} \log R(\vec{w})$ является всюду гладкой функцией в гипероктанте $\mathcal{H}_{ML}$ за исключением точки $\vec{w} = 0$. Рассмотрим область

$$\mathcal{M} = \{\vec{w} \in \mathcal{H}_{ML}, |w_i| \geq \varepsilon, \forall i = 1, \ldots, n\}.$$

Функция $\hat{N} \log R(\vec{w})$ всюду гладкая в $\mathcal{M}$, поэтому мы можем использовать метод Ньютона для оптимизации аппроксимированного Байесовского критерия

$$\hat{Q} = L(\mathcal{T}|\mathcal{X}, \vec{w}) + \hat{N} \log R(\vec{w}). \tag{5}$$

При этом, как только достигается какая-либо из границ, то есть $w_i = \varepsilon$, соответствующий вес приравниваем 0. Следующее выражение устанавливает связь между σ и ε

$$\lim_{\varepsilon \to +0} \left(C_0 - \exp\left(-\frac{\varepsilon^2}{2\sigma^2}\right)\right) \log(C_1 + \varepsilon) = C_0 \log C_1, \tag{6}$$

$$\forall C_0 > 0, \quad C_1 > 0.$$

На Рисунке 1 изображено поведение при изменении одного из параметров четкого критерия

$$C_0 \log(C_1 + w_i)$$

и его нечеткого приближения

$$\left(C_0 - \exp\left(-\frac{\varepsilon^2}{2\sigma^2}\right)\right) \log(C_1 + \varepsilon)$$

с $C_0 = 2$, $C_1 = 0.03$, $\sigma = 0.3$. При этом аппроксимирующий регуляризатор является непрерывным для всех значений w_i и гладким для $w_i > 0$ (так же как и для $w_i < 0$).

В случае, когда все веса обращаются в ноль, предполагается, что $\hat{N} \log R(\vec{w})$ также обращается в ноль. Чтобы гарантировать это, потребуем чтобы предел нашего аппроксимирующего регуляризатора был равен нулю(6) при $C_0 = 0$ и $C_1 = 0$, то есть

$$\lim_{\varepsilon \to +0} -\exp\left(-\frac{\varepsilon^2}{2\sigma^2}\right) \log \varepsilon = \frac{\varepsilon^2}{2\sigma^2} \log \varepsilon = 0.$$

Данное равенство выполняется, когда $\sigma = \varepsilon^k$ при $k < 1$. Положим ε достаточно маленьким числом, а σ положим равным $\sqrt{\varepsilon}$ - это даст нам корректную аппроксимацию Байесовского критерия (3) его гладким аналогом (5) во всем гипероктанте $\mathcal{H}_{ML}$. В качестве начального приближения может быть выбрана точка максимума нерегуляризоавнного правдоподобия $\vec{w}_{ML}$.

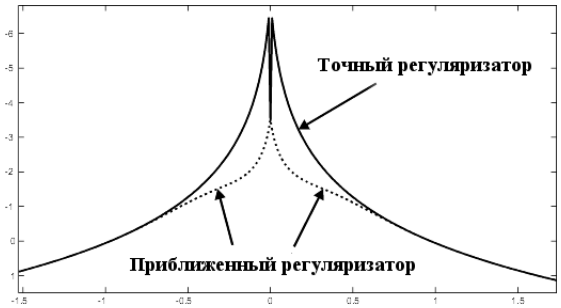


Рис. 8. Поведение четкого и аппроксимирующего регуляризатора при изменении одного из параметров.

4. Результаты

В данной части будет проведено сравнение качества работы двух процедур обучения (BLogReg и FuzzyBLogReg), используемых для предложенной логистической регрессии, метода релевантных векторов (Tipping, 2001) с базисными функцями $\phi_i(\vec{x}) = \langle \vec{x}_i, \vec{x} \rangle$ (как альтернативы применения Байесовского подхода к логистической регресии) и метода опорных векторов (Guyon et al., 1992) с линейной ядровой функцией и параметром $C = 1$. Для всех методов проводилось сравнение количества ошибок. Кроме того, для BLogReg и FuzzyBLogReg проводилось сравнение времени работы и получаемой разреженности решения. Данные для эксперимента были взяты их репозитория UCI. Для каждого набора данных номинальные признаки были заменены на бинарные, неизвестные значения признаков были заменены соответствующими средними значениями, после чего объекты были нормированы таким образом, что для каждого признака его среднее равнялось нулю, а отклонение - единице. Для определения количества ошибок, времени работы и разреженности решения при работе на каждом наборе данных использовался 5x2-скользящий контроль. Данный подход общепризнанно является одним из наиболее достоверных для определения характеристик классификатора (Dietterich, 1998). Результаты эксперимента представлены в Таблицах 2, 3 и 4. Ранг вычислялся стандартным образом: для каждого набора данных определялось место классификатора(от 1 до 4); затем места суммировались.

Таблица 2. Процент ошибок со стандартным отклонением.

Данные	BLogReg	FuzzyBLogReg	LinearSVM	LinearRVM
Bupa liver disorders	33.28 ± 2.60	32.64 ± 3.33	31.59 ± 0.58	32.35 ± 0.73
German credit numeric	25.12 ± 0.97	24.60 ± 0.69	24.92 ± 1.11	24.76 ± 0.22
Heart	18.96 ± 0.66	18.81 ± 0.99	18.81 ± 0.99	20.07 ± 1.49
Australian	15.07 ± 0.91	15.19 ± 0.84	16.17 ± 0.65	15.57 ± 0.22
Pima-indians-diabetes	23.10 ± 0.96	23.10 ± 0.96	23.39 ± 0.80	23.67 ± 0.50
Thyroid-sick.test	4.28 ± 0.35	4.07 ± 0.30	3.97 ± 0.57	3.91 ± 0.38
Wisconsin Diagnostic Breast Cancer	2.99 ± 0.48	2.78 ± 0.15	3.30 ± 0.45	3.23 ± 0.72
Wisconsin Prognostic Breast Cancer	22.73 ± 1.13	27.98 ± 1.81	22.53 ± 1.81	22.63 ± 0.90
Ранг	**22.50**	**17.00**	**19.50**	**21.00**

Таблица 3. Время обучения со стандартным отклонением (в секундах).

Данные	BLogReg	FuzzyBLogReg
Bupa liver disorders	0.67 ± 0.40	0.26 ± 0.11
German credit numeric	1.44 ± 0.16	0.33 ± 0.06
Heart	0.68 ± 0.05	0.34 ± 0.04
Australian	2.41 ± 0.31	0.57 ± 0.15
Pima-indians-diabetes	0.33 ± 0.05	0.14 ± 0.02
Thyroid-sick.test	4.14 ± 0.65	3.13 ± 1.55
Wisconsin Diagnostic Breast Cancer	3.31 ± 1.06	3.29 ± 0.79
Wisconsin Prognostic Breast Cancer	3.78 ± 0.81	0.51 ± 0.13

Таблица 4. Разреженность со стандартным отклонением

Данные	#Признаков	BLogReg	FuzzyBLogReg
Bupa liver disorders	6	5.30 ± 0.27	5.10 ± 0.22
German credit numeric	24	18.50 ± 1.06	17.20 ± 0.45
Heart	20	11.10 ± 0.89	8.90 ± 0.89
Australian	38	19.60 ± 1.47	17.30 ± 1.48
Pima-indians-diabetes	8	6.90 ± 0.42	6.70 ± 0.27
Thyroid-sick.test	31	11.60 ± 0.42	11.30 ± 0.67
Wisconsin Diagnostic Breast Cancer	30	9.80 ± 0.27	10.30 ± 0.67
Wisconsin Prognostic Breast Cancer	33	7.00 ± 0.00	11.30 ± 5.14

Полученные результаты позволяют сделать следующие выводы. С точки зрения качества распознавания все четыре алгоритма показывают сравнимые результаты. Это означает, что BLogReg и FuzzyBLogReg выходят в окрестность одной и той же точки $\vec{w}_{MP}$. Как следствие, оба алгоритма показывают практически одинаковую разреженность. При этом, FuzzyBLogReg оказался заметно быстрее, чем BLogReg во всех тестах. Однако, стоит отметить, что во всех наборах данных количество объектов было значительно больше количества признаков (то есть по сути - количества степеней свободы).

В следующем эксперименте был изучен случай, когда количество объектов сравнимо с количеством признаков. У объектов данного набора данных 111 признаков. Поскольку всего объектов 8124, то набор был разбит на подмножества по 200 объектов случайным образом с сохранением априорного распределения объектов по классам (наборы от 1 до 5). Полученные результаты приведены в Таблицах 5, 6 и 7.

Таблица 5. Процент ошибок со стандартным отклонением.

Данные	BLogReg	FuzzyBLogReg	RVM	LinearSVM
Agaricus-lepiota (mushroom) 1	1.48 ± 0.30	1.09 ± 0.14	2.62 ± 0.51	1.28 ± 0.21
Agaricus-lepiota (mushroom) 2	1.73 ± 0.00	1.09 ± 0.14	2.81 ± 0.62	0.99 ± 0.00
Agaricus-lepiota (mushroom) 3	1.53 ± 0.21	1.33 ± 0.37	3.46 ± 0.68	1.33 ± 0.37
Agaricus-lepiota (mushroom) 4	2.12 ± 0.62	1.33 ± 0.77	4.94 ± 2.93	1.68 ± 0.88
Agaricus-lepiota (mushroom) 5	1.88 ± 1.01	1.33 ± 1.11	4.59 ± 0.81	1.33 ± 1.11
Ранг	**15.00**	**7.00**	**20.00**	**8.00**

Таблица 6. Время обучения со стандартным отклонением (в секундах).

Данные	BLogReg	FuzzyBLogReg
Agaricus-lepiota (mushroom) 1	2.48 ± 0.40	0.15 ± 0.05
Agaricus-lepiota (mushroom) 2	2.58 ± 0.36	0.13 ± 0.01
Agaricus-lepiota (mushroom) 3	2.64 ± 0.46	0.13 ± 0.01
Agaricus-lepiota (mushroom) 4	2.73 ± 0.48	0.14 ± 0.02
Agaricus-lepiota (mushroom) 5	2.77 ± 0.42	0.12 ± 0.01

Таблица 7. Разреженность со стандартным отклонением

Данные	#Признаков	BLogReg	FuzzyBLogReg
Agaricus-lepiota (mushroom) 1	111	17.60 ± 0.42	110.40 ± 0.55
Agaricus-lepiota (mushroom) 2	111	17.90 ± 0.22	111.00 ± 0.00
Agaricus-lepiota (mushroom) 3	111	16.20 ± 0.76	107.60 ± 3.34
Agaricus-lepiota (mushroom) 4	111	17.60 ± 1.29	107.30 ± 3.07
Agaricus-lepiota (mushroom) 5	111	16.70 ± 1.25	109.40 ± 1.67

Как и в предыдущем эксперименте, оба алгоритма показали сравнимое качество работы, но, нечеткий вариант BLogReg дает уже значительно меньшую разреженность по сравнению с четким BLogReg. Такой результат возникает из-за того, что ни Байесовский критерий Q, ни его аппроксимация $\hat{Q}$ больше не являются унимодальными функциями (тогда как критерий F являлся). В случае, когда количество объектов значительно превышает количество признаков, оба метода выходят в точки, очень близкие друг к другу, тогда как с ростом количества признаков количество локальных экстремумов возрастает и нечеткий метод становится зависимым от выбора начальной точки приближения. В частности, начало оптимизации из точки $\vec{w}_{ML}$ дает точное, но не слишком разреженное решение. Поэтому в случаях, когда разреженность решения является критичным показателем работы алгоритма, предпочтение следует отдать четкому варианту BLogReg.

СПИСОК ЛИТЕРАТУРЫ

1. *Ambroise, C. and McLachlan, G. J.,* 2002, Selection bias in gene extraction on the basis of microarray gene-expression data. Proceedings of the National Academy of Sciences, 6562–6566.
2. *Alon, U. and Barkai, N. and Notterman, D. A. and Gish, K. and Ybarra, S. and Mack, D. and Levine, A. J.,* 1999, Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proceedings of the National Academy of Sciences, 6745–6750.
3. *Berger, J. O.,* 1985, Statistical decision theory and Bayesian Analysis.Springer Series in Statistics.
4. *Bishop C. M.,* 1995, Neural Networks for Pattern Recognition. Oxford University Press.
5. *Buntine, W. L. and Weigend, A. S.,* 1991. Bayesian back-propagation. Complex Systems, 603–643.
6. *G. C. Cawley and N. L. C. Talbot,* 2006, Gene selection in cancer classification using sparse logistic regression with Bayesian regularization. Bioinformatics,2348-2355.
7. *T. G. Dietterich,* 1998, Approximate statistical tests for comparing supervised classification learning algorithms.Neural Computation, 1895-1924.
8. *Faul, A. C. and Tipping, M. E.,* 2002, Analysis of sparse Bayesian learning.Advances in Neural Information Processing Systems,383–389.
9. *Figueiredo, M.,* 2003,Adaptive sparseness for supervised learning.IEEE Transactions on Pattern Analysis and Machine Intelligence,1150–1159.

10. *Guyon, I. M. and Boser, B. E. and Vapnik, V. N.,* 1992, A training algorithm for optimal margin classifiers. Fifth Annual Workshop on Computational Learning Theory.

11. *Keerthi, S. S. and Duan, K. and Shevade, S. K. and Poo, A. N.,* 2002, A Fast Dual Algorithm for Kernel Logistic Regression. Proceedings of the International Conference on Machine Learning.

12. *MacKay, D. J. C.,* 1992, A practical Bayesian framework for backprop networks. Neural Computation,448-472.

13. *MacKay, D. J. C.,* 1994, Hyperparameters : optimise or integrate out?. Maximum Entropy and Bayesian Methods.

14. *Shevade, S. K. and Keerthi, S. S.,* 2002, A simple and efficient algorithm for gene selection using sparse logistic regression. Bioinformatics,2246–2253.

15. *Tikhonov, A. N. and Arsenin, V. Y.,* 1977, Solutions of ill-posed problems.

16. *Tipping, M. E.,* 2001, Sparse Bayesian learning and the Relevance Vector Machine. Journal of Machine Learning Research, 211–244.

17. *Williams, P. M.,* 1995, Bayesian regularization and pruning using a Laplace prior. Neural Computation, 117–143.

СБОРНИК СТАТЕЙ МОЛОДЫХ УЧЕНЫХ факультета ВМиК МГУ, 2007, выпуск № 4, с. 82–89

УДК 517.956.2

ОБ ОДНОМ ПОДХОДЕ К ПОИСКУ БЫСТРЫХ АЛГОРИТМОВ УМНОЖЕНИЯ МАТРИЦ

© 2007 г. В. Б. Ларионов

Tesla-Coil@yandex.ru

Кафедра Математической кибернетики

Основные понятия. Одной из интересных и тяжёлых задач является задача умножения матриц. Её история начинается с замечательного результата Штрассена. Он показал в [5], что две квадратные матрицы размера 2 можно умножить, используя только 7 умножений линейных комбинаций элементов матриц, что повлекло за собой асимптотическую оценку $n^{\log_2 7}$ на количество умножений. Этот результат неоднократно улучшался (историю этого можно посмотреть в [8] и остановился на $O(n^{2.38})$ в работе [4] более пятнадцати лет назад. С этой задачей тесно связаны другие, не менее интересные задачи, такие, как нахождение определителя и обращение матриц ([8]). Отметим, что до сих пор задача не решена для квадратных матриц размера 3. Нижняя оценка в этом случае равна 18 ([9]), а верхняя — 23 ([6]) умножениям.

В [2] показано, что для задачи умножения матриц может быть использована идея "расширения модели".

Определение 1. Алгеброй называется линейное пространство с заданной на нём операцией умножения, линейной по обоим сомножителям.

Определение 2. Алгебра P называется алгеброй с простым умножением, если существует базис $e_1, e_2, \ldots, e_k$ в P и подстановка σ порядка k такие, что $e_i e_j = 0$ при $j \neq \sigma(i)$.

Нашей основной задачей будет поиск такой алгебры P над полем комплексных чисел, которая содержит в себе подалгебру M, изоморфную алгебре матриц (далее везде будем использовать эти буквы для обозначения соответствующих алгебр). Очевидно, это будет означать, что мы можем умножать матрицы, используя столько умножений, сколько ненулевых произведений базисных векторов в алгебре. Далее будем рассматривать только алгебры с простым умножением, поскольку по любой из них можно построить алгебру, обладающую этим свойством, увеличив число умножений не более, чем в два раза (нас интересует только асимптотика).

Зададим алгебру с простым умножением, как в [1], тремя матрицами A, C_σ и C_α. Напомним, что матрица C_σ задаёт перестановку σ из определения 2, в i-й строке матрицы A покоординатно записаны произведения $e_{\alpha(i)}e_{\sigma(\alpha(i))}$ (везде далее будем полагать, что $\{e_i\}_{i=1}^k$ – базис из определения алгебры с простым умножением), где α – перестановка, задаваемая матрицей C_α. Согласно [7], мы можем считать, что линейная оболочка всех векторов алгебры вида $e_i e_{\sigma(i)}$ есть в точности M. Откуда

$$\operatorname{rang} A = n^2$$

В этом случае, как показано в [1], для каждого $f = \sum_{i=1}^{k} f_i e_i \in P$ мы можем ввести операторы умножения векторов в алгебре слева и справа: H_f^L и H_f^R соответственно, имеющие вид

$$H_f^R = C_\sigma^{-1} f^{diag} C_\alpha^{-1} A, \tag{1}$$

$$H_f^L = C_\sigma f^{diag} C_\sigma^{-1} C_\alpha^{-1} A, \tag{2}$$

где через f^{diag} мы будем обозначать диагональную матрицу с элементами $f_{jj}^{diag} = f_j$.

Смысл этих операторов виден из следующих соотношений:

$$f \cdot g = R_f H_g^L = R_g H_f^R, \tag{3}$$

где f, g – произвольные вектора из алгебры, R_f – вектор-строка размера k с элементами f_i. Доказательство этих формул можно найти в [1]. Иногда мы будем обозначать введённые операторы H^L и H^R, когда конкретный вектор f для нас не будет иметь значения.

Далее будем использовать обозначения: $\dim M = n^2$, $\dim P = k$. Тот факт, что вектору $f \in M$ соответствует квадратная матрица размера n M_f, будем обозначать $f \sim M_f$.

В [2] показано, что при $n = 2$ существует в точности 3 неизоморфные 7-мерные алгебры, обладающие указанными выше свойствами, содержащие подалгебру M.

Также результаты о существовании быстрых билинейных алгоритмов для умножения комплексных чисел и кватернионов могут быть представлены как вложение этих алгебр в алгебры с простым умножением [10].

Свойства операторов H^R и H^L. В [1] показано, что алгебра P и указанные операторы при условии

$$M \in P \tag{4}$$

обладают рядом интересных свойств. Исходя из них, мы попытаемся найти необходимые для (4) условия.

Напомним некоторые из свойств, доказанных в [1]:

1. Требование единичной матрицы:

$$f \sim E \Leftrightarrow AH_f^R = AH_f^L = A; \tag{5}$$

2. Требование ассоциативности:

$$AH_f^R H_h^L = AH_h^L H_f^R, \quad \forall f, h \in M; \tag{6}$$

3. Требование замкнутости относительно умножения:

$$R_f H_h^L = R_g \Leftrightarrow R_h H_f^R = R_g \Leftrightarrow AH_h^R H_f^R = AH_g^R \Leftrightarrow AH_f^L H_h^L = AH_g^L, \quad \forall f, h, g \in M; \tag{7}$$

Докажем теперь некоторые более сложные и важные свойства исследуемых операторов.

Пусть P – алгебра с простым умножением и выполнено указанное выше свойство для линейной оболочки произведений базисных векторов, а также (4). Будем обозначать это условие (*) - оно нам неоднократно понадобится.

Лемма 1. *Пусть выполнено (*), f – произвольный вектор из M, $f \sim M_f$, тогда:*

1. $\operatorname{rang} M_f = \frac{1}{n} \operatorname{rang} AH_f^L = \frac{1}{n} \operatorname{rang} AH_f^R$;
2. *λ – собственное значение матрицы M_f геометрической кратности ([3]) s* $\Leftrightarrow \operatorname{rang}(AH_f^L - \lambda A) = \operatorname{rang}(AH_f^R - \lambda A) = n^2 - ns$.

Доказательство. Краткое доказательство этой леммы было дано в [1]. Здесь мы дадим развёрнутое доказательство.

1. Пусть $\operatorname{rang} M_f = s$. По известной теореме из линейной алгебры ([3]) получаем, что $\dim \ker M_f = n - s$. Пусть векторы-строки $v_1, v_2, \ldots, v_{n-s}$ образуют базис ядра матрицы M_f слева. Построим ядро для оператора H_f^L на подпространстве M. Пусть V_{ij} – матрица размера n на n, у которой в i-й строке записан вектор v_j, остальные строки нулевые. Очевидно, что $V_{ij} M_f = 0$ для любых i, j. Из линейной независимости векторов v_j следует, что все $n(n-s)$ матриц V_{ij} линейно независимы. Этим матрицам отвечают какие-то вектора $w_{ij} \in M$, которые, в силу невырожденности отображения также линейно независимы.

   Покажем, что построенная система векторов w_{ij} образует базис ядра оператора H_f^L. В самом деле, если это не так, то существует вектор $w_0 \in M$ такой, что $w_0 H_f^L = 0$ и

он не выражается линейно через систему w_{ij}. Тогда найдётся отвечающая ему матрица размера n на n M_0 такая, что $M_0M_f = 0$ и линейно независимая с матрицами V_{ij}. Получаем противоречие с определением векторов v_i.

Итак, оператор H_f^L имеет дефект на подпространстве M равный $n(n-s)$. Заметим, что матрица A, рассматриваемая как оператор, отображает пространство P в M, причём $\operatorname{Im} A = M$. Откуда

$$\operatorname{rang} AH_f^L = n^2 - n(n-s) = ns.$$

Повторив те же рассуждения для базиса ядра M_f справа, мы получим аналогичное утверждение для оператора H_f^R. Первый пункт доказан.

2. Будем обозначать вектор из M, отвечающий единичной матрице буквой e. По условию:

$$\operatorname{rang}(M_f - \lambda E) = n - s.$$

Воспользуемся уже доказанным первым пунктом леммы:

$$\operatorname{rang}(AH_f^L - \lambda AH_e^L) = \operatorname{rang}(AH_f^R - \lambda AH_e^R) = n(n-s).$$

Здесь мы учли линейность отображения матриц в алгебру. Воспользовавшись (5), получаем требуемое.

Лемма доказана.

Будем обозначать через $\varphi_f^L(\lambda)$, $\varphi_f^R(\lambda)$ и $\varphi_{M_f}(\lambda)$ характеристические многочлены ([3]) операторов H_f^L, H_f^R и матрицы M_f соответственно.

Лемма 2. *Пусть выполнено (*) и $f \sim M_f$, где f – произвольный вектор из M. Тогда каждой клетке Жордановой формы матрицы M_f с ненулевым собственным значением отвечает n таких же клеток операторов H_f^L и H_f^R.*

Доказательство. Пусть $\lambda_i \neq 0$ – корень характеристического многочлена матрицы M_f, J_{λ_i} – Жорданова клетка, соответствующая этому собственному значению, $v_1, \ldots, v_s$ – векторы-строки, $w_1, \ldots, w_n$ – векторы-столбцы Жорданова базиса, соответствующие этой клетке (см. [3]). Составим матрицы M_{ij}^L и M_{ij}^R размера $n * n$ следующим образом: в M_{ij}^L на i-й строке записан вектор v_j, остальные нули, а в M_{ij}^R в i-м столбце записан вектор w_j, остальные нули.

Возьмём в качестве Жорданова базиса оператора H_f^L образы матриц M_{ij}^L при отображении в алгебру и базис ядра самого H_f^L, а у H_f^R – образы матриц M_{ij}^R и векторы базиса ядра H_f^R. Легко понять, что при должной расстановке строк Жордановых форм этих операторов, мы получим требуемое.

Лемма доказана.

Следствие.

$$\varphi_f^L(\lambda) = \varphi_f^R(\lambda) = \left[\varphi_{M_f}(\lambda)\right]^n \lambda^{k-n^2}. \tag{8}$$

Следствие. Ещё одним необходимым для (4) условием является равенство спектров операторов H_f^L и H_f^R для всех $f \in M$.

Замечание. Пока мы доказали лишь существование такого базиса, построить его мы сможем несколько позже.

Замечание. Отметим, что, несмотря на указанное выше равенство спектров, геометрические кратности корня $\lambda = 0$ характеристических многочленов наших операторов совпадать не обязаны (хотя у Штрассена это имеет место). В Жордановой форме этих операторов на векторах, не относящихся к M расположена большая клетка с нулями на диагонали и неизвестным в общем случае числом единиц над диагональю. С этой клеткой соединяются другие с нулями на диагонали в случае вырожденных M_f. В этом случае дефект операторов H_f^L и H_f^R не обязан увеличиться, ядро может просто сдвинуться на подпространство M (в алгебре из [2] получен пример такой матрицы размера 2 на 2).

Итак, мы установили довольно тесную связь между матрицей M_f и операторами H_f^L, H_f^R.

Мы можем получить ещё одно необходимое для (4) условие. Достаточно вспомнить, что операторы H^L и H^R содержательно толкуются как умножение в алгебре слева и справа. Из линейной алгебры известно, что между собственными векторами справа и слева существует связь. В самом деле,

$$M = XJX^{-1}, \quad \forall M^{n*n}, \ |X| \neq 0,$$

J – Жорданова форма матрицы M. Очевидно, что в строках матрицы X^{-1} стоят собственные вектора-строки матрицы M, а в столбцах X – собственные вектора-столбцы матрицы M. Из того, что ортогональность сохраняется при переходе в алгебру, получаем, что доказана следующая

Лемма 3.

Пусть выполнено (*), f – произвольный вектор из M, $\{f_1^L, f_2^L, \ldots, f_h^L\}$ – любой базис из собственных векторов всех собственных подпространств оператора H_f^L. Тогда существует аналогичный базис H_f^R $\{f_1^R, f_2^R, \ldots, f_h^R\}$ (и его можно построить), обладающий следующим свойством: произведение любых веторов $f_i^L * f_j^R$ (в любом порядке), отвечающих разным собственным значениям равно нулю.

Подведём итог наших рассуждений в следующей теореме.

Теорема 1.

Необходимыми для (4), где P – алгебра с простым умножением определяется тройкой матриц A C_1 и C_2, линейная комбинация векторов-строк матрицы A есть в точности M, условиями являются:

1. *Матрицы C_1 и C_2 задают перестановки,* $\operatorname{rang} A = n^2$;
2. *(5), (6), (7);*
3. *(8), $\forall f \in M$, причём для любых, наперёд заданных, собственных чисел можно найти вектор в M, операторы которого ими обладают;*
4. *Жордановы формы операторов H_f^L и H_f^R для любого $f \in M$ имеют по n одинаковых клеток, соответствующих подпространству M;*
5. *Условия леммы 3.*

Алгоритм, проверяющий алгебру P на наличие в ней подалгебры M.

Теорема 2. *Существует алгоритм, проверяющей по алгебре P, заданной тремя матрицами A, C_σ и C_α, как описано выше, наличие в ней подалгебры M, изоморфной алгебре матриц. В случае положительного ответа алгоритм выдаёт отображение алгебры матриц в алгебру P.*

Доказательство. Найдём в подалгебре M (напомним, что это линейная оболочка векторов, записанных в матрице A) вектор, каждое собственное число оператора умножения которого, кроме быть может нулевого, имеет кратность не более, чем n, и на M все собственные числа ненулевые. Для этого найдём подпространства R_i ($i = 1, \ldots, n$) такие, что

$$R_1 \oplus R_2 \oplus \ldots \oplus R_n = M, \quad \dim R_i = n, \tag{9}$$

кроме того, подпространство R_i при любом отображении матриц в алгебру будет отвечать матрицам $n*n$, у которых в каждой строке записан вектор, коллинеарный некоторому вектору r_i (ясно, что при этом $\{r_i\}_{i=1}^n$ – базис пространства R^n).

Если мы найдём указанное разбиение, то можем построить искомый вектор следующим образом. Пусть $m_1, \ldots, m_{n^2}$ – базис M, $f = f_1 m_1 + \ldots + f_{n^2} m_{n^2}$ – искомый вектор, его координаты в указанном базисе – неизвестные переменные, $\lambda_1, \ldots, \lambda_n$ – различные числа, каждое из которых отлично от нуля. Запишем оператор умножения слева H_f^L для f по

формуле (2). В каждой клетке матрицы H_f^L будет записано линейное по $f_1, \ldots, f_k$ выражение. Составим систему:

$$\begin{aligned} R_1 H_f^L &= \lambda_1 R_1, \\ &\cdots \\ R_n H_f^L &= \lambda_n R_n. \end{aligned} \tag{10}$$

(Здесь каждое уравнение содержит в себе n уравнений, записанных для векторов базиса R_i). Это линейная по $f_1, \ldots, f_{n^2}$ система. Если $M \in P$, то она имеет решение. Найдём любое из них и получим f.

Построим теперь разбиение (9). Запишем операторы $H_{m_1}^L, \ldots, H_{m_{n^2}}^L$. Пусть $\lambda_1^i, \ldots, \lambda_{k_i}^i$ – собственные числа оператора $H_{m_i}^L$. Каждому λ_j^i сопоставим подпространство H_j^i – линейную оболочку, натянутую на вектора-строки Жорданова базиса оператора $H_{m_i}^L$, отвечающие клеткам с собственным значением λ_j^i и являющиеся собственными векторами оператора $H_{m_i}^L$. Очевидно, это алгоритмически легко осуществимо. Если у одного из элементов полученного множества подпространств размерность будет меньше, чем n, то алгоритм выдаёт отрицательный ответ.

Покажем, что, если среди подпространств H_j^i найдётся хотя бы одно размерности n, то мы можем построить искомое разбиение. В самом деле, подействуем на H_j^i всеми операторами $H_{m_l}^L$. Докажем, что в случае $M \in P$ из n^2 полученных в результате этой операции подпространств можно выбрать n таких, которые будут образовывать искомое разбиение. Вернёмся в алгебру матриц. Пусть $m_i \sim M_i$, H_0 – подпространство матриц ранга 1, образующих подпространство H_j^i при отображении в алгебру P. Это будет подпространство матриц, построенное на одном векторе-строке r_0. Последний факт следует из того, что мы нашли матричный оператор, у которого какому-то отличному от других собственному значению соответствует собственное подпространство H_0, и доказательства леммы 2. Поскольку $\{M_i\}_{i=1}^{n^2}$ – базис пространства матриц размера n на n, то произвольная матрица из этого пространства имеет вид:

$$M_a = \sum_{i=1}^{n^2} a_i M_i,$$

откуда

$$H_0 M_a = \sum_{i=1}^{n^2} a_i H_0 M_i,$$

где $H_0 M_i = \{M' * M_i : M' \in H_0\}$ - это будет подпространство матриц ранга 1, построенных на каком-то векторе-строке r_i. Из последнего соотношения следует, что образы при отображении в алгебру P n подпространств $H_0 M_i$ таких, что соответствующие n векторов r_i образуют базис R^n и будут образовывать искомое разбиение M (9).

В случае, если все подпространства H_j^i имеют размерность больше n, рассмотрим такой номер j, что оператор $H_{m_j}^L$ имеет хотя бы два различных собственных значения (предположим вначале, что такой найдётся) и среди всех таких операторов обладает подпространством минимальной размерности, отвечающем какому-либо собственному значению. Обозначим указанное подпространство H' и покажем, что из него можно выделить подпространство размерности n, которому в случае $M \in P$ будут отвечать матрицы ранга 1, построенные на одном векторе-строке. Пусть $\dim H' = nl$. Этот факт следует из того, что вектора Жорданова базиса оператора $H_{m_j}^L$ на M, соответствующие некоторому собственному значению, отвечают матрицам, построенным на векторах-строках из Жорданова базиса m_j, соответствующих тому жу собственному значению (доказательство леммы 2). Из того, что $m_1, \ldots, m_{n^2}$ – базис M, следует, что $1 < l < n$. Пусть $\{e_i\}_{i=1}^k$ – базис P, обладающий следующими свойствами: первые nl векторов образуют базис H', первые n^2 векторов образуют базис M. В качестве такого базиса возьмём Жорданов базис оператора $H_{m_j}^L$ на M. Из изложенного выше факта

следует, что вектора $e_{nl+1}, \ldots, e_{n^2}$ соответствуют матрицам, построенным на $n-l$ векторах, которые вместе с l векторами, образующими подпространство матриц, соответствующее H',образуют базис R^n. Пусть оператор $H'^L_{m_i}$ ($i = 1, \ldots, n^2$) отображает вектора $e_1, \ldots, e_{nl}$ в соответственно проекции векторов $e_1 H^L_{m_i}, \ldots, e_{nl} H^L_{m_i}$ на H', а $e_{nl+1}, \ldots, e_{n^2}$ – в нули. Для построения этих операторов воспользуемся системой, подобной (10):

$$e_1 H'^L_{g_i} = pr_{H'}(e_1 H^L_{m_i}),$$

$$\ldots$$

$$e_{nl} H'^L_{g_i} = pr_{H'}(e_{nl} H^L_{m_i}),$$

$$e_{nl+1} H'^L_{g_i} = 0,$$

$$\ldots$$

$$e_{n^2} H'^L_{g_i} = 0,$$

где $g_i = \sum\limits_{j=1}^{n^2} b_{ij} m_j$, $pr_{H'}$ обозначает проектирование на H'. То есть для каждого $i = 1, \ldots, n^2$ получаем систему уравнений с n^2 неизвестными $b_{i1}, \ldots, b_{in^2}$. Эта система в случае $M \in P$ для каждого i будет иметь хотя бы одно решение, поскольку подпространству H' отвечают матрицы, у которых в каждой строке записана линейная комбинация l линейно независимых векторов-строк. А мы строим операторы $H'^L_{g_i}$, которым будут соответствовать матрицы, отображающие указанные l векторов-строк в фиксированные для каждого i другие l векторов, а остальные – в нуль. Очевидно, для каждого i такая матрица найдётся, а значит и найдётся оператор $H'^L_{g_i}$ (здесь важно свойство базиса $\{e_i\}_{i=1}^k$, что при проектировании зануляются координаты при векторах $e_{nl+1}, \ldots, e_{n^2}$, которым при любом отображении матриц в алгебру соответствуют матрицы, построенные на векторах, линейно независимых с теми, на которых построены матрицы, соответствующие подпространству H'), так как по построению системы этот оператор записан для вектора $g_i \in M$.

Пусть теперь каждый оператор $H^L_{m_i}$ обладает на M только одним собственным значением. Возьмём любой из них и заменим вектор, для которого он записан на его сумму с произвольным вектором из Жорданова базиса этого оператора на M, не являющегося собственным. Несложно проверить, что оператор умножения полученного вектора будет иметь хотя бы два различных собственных значения.

Из построенной системы операторов $\{H'^L_{g_i}\}_{i=1}^{n^2}$ выберем nl линейно независимых (все эти операторы по построению вместо M работают на H' размерности nl) – пусть это будут nl первых операторов. И повторим теперь всё с начала с заменой M на H' и ситемы $\{H^L_{m_i}\}_{i=1}^{n^2}$ на $\{H'^L_{g_i}\}_{i=1}^{nk}$ (то есть снова найдём собственные вектора операторов, отвечающие им подпространства и так далее), поскольку для осуществления шага нам достаточно знать некоторое подпространство подалгебры M и базис из линейных операторов, действующих на нём. Заметим, что по построению

$$\dim H' \leqslant \frac{\dim M}{2}.$$

Таким образом процесс измельчения будет происходить довольно быстро. И когда мы остановимся, у нас будет подпрстранство $\overline{H}$ из M, которому будут соответствовать матрицы ранга 1, построенные на одном и том же векторе-строке, а значит, и разбиение (9).

Везде выше мы предполагали, что $M \in P$. Если на каком-то этапе алгоритма что-нибудь пойдёт не так, то мы остановимся, выдав отрицательный ответ.

Сопоставим вектору f любую матрицу M_f с собственными значениями $\lambda_1, \ldots, \lambda_n$. Для простоты будем считать, что M_f – диагональная матрица. Пусть $v_1, \ldots, v_n$ – собственные векторы-строки M_f, а $w_1, \ldots, w_n$ – собственные векторы-столбцы M_f. Обозначим

$$R_{ij} = \{M \in R^{n*n} : \ M = cw_i v_j, \ \forall c\}.$$

В нашем случае R_{ij} – это одномерное подпространство матриц, у которых в i-й строке j-ом столбце записано произвольное число, остальные элементы равны нулю. Построим операторы H_f^L и H_f^R, найдём их собственные подпространства $L_1, \ldots, L_n$ и $R_1, \ldots, R_n$ соответственно. Пусть

$$L_{ij} = L_i \cap R_j.$$

По лемме 2 $\dim L_{ij} = 1$. Очевидно, все эти подпространства линейно независимы и образуют базис M. Вычислим их.

Ясно, что при искомом отображении матриц в алгебру, R_{ij} должно переходить в L_{ij}. Мы можем восстановить соответствие только для R_{ii}. Возьмём любой вектор из L_{ii}, найдём собственное значение его оператора умножения (оно будет единственным кратности n) и сопоставим этому вектору матрицу, где в позиции (i,i) стоит это собственное значение, остальные – нули. Будем обозначать указанное соответствие:

$$f_{ii} \sim M_{ii},$$

где $f_{ii} \in L_{ii}$, $M_{ii} \in R_{ii}$.

Неопределённость присутствующая при доопределении искомого отображения характеризуется множеством

$$X = \{Y \in R^{n*n} : Y M_f Y^{-1} = M_f\}.$$

Поскольку у M_f все собственные значения разные, все такие матрицы Y имеют те же собственные векторы слева и справа, что и M_f, могут отличаться только собственными значениями. Очевидно, что

$$Y R_{ij} Y^{-1} = R_{ij}.$$

Сопоставим произвольным образом подпространства R_{12} и L_{12} (аналогично $f_{12} \sim M_{12}$). Несложно проверить (с учётом сказанного выше), что все матрицы Y, удовлетворяющие

$$Y M_f Y^{-1} = M_f,$$

$$Y M_{12} Y^{-1} = M_{12}$$

это матрицы множества X, у которых первые два собственные значения равны. Все они произвольным образом отображают подпространства $R_{i-1,i}$ $(i = 3, \ldots, n)$ в себя. Поэтому мы имеем право произвольно сопоставить $f_{23} \sim M_{23}$ и так далее до $f_{n-1,n} \sim M_{n-1,n}$. Среди матриц, осуществляющих автоморфизм, останутся только коллинеарные единичной, которые оставляют неподвижной любую матрицу, поэтому мы задали отображение однозначно. Получим его в явном виде.

Не ограничивая общности, будем считать, что обозначенные выше матрицы M_{ij} – это матрицы с единицей в позиции (i,j). Очевидно, что $f_{ik} * f_{kj} \sim M_{ij}$, откуда мы восстановим и проверим все соотношения $f_{ij} \sim M_{ij}$, где $i \leq j$. Для воосстановления остальных f_{ij} воспользуемся условием

$$f_{ij} * f_{ji} = f_{ii}.$$

Этого нам будет достаточно, поскольку мы знаем подпространство, соответствующее при переходе в алгебру одномерному подпространству матриц cM_{ij}, и нам нужно определить лишь константу.

Проверим не рассмотренные ранее произведения базисных элементов:

$$f_{ij} * f_{kl} = \delta_{jk} f_{il}.$$

Если всё выполняется, то M изоморфна подалгебре матриц размера n на n, и искомое отображение построено.

То, что переход к другому базису в алгебре матриц является автоморфизмом, гарантирует нам правильность отрицательного ответа.

Теорема доказана.

Следствие. Существует алгоритм для построения описанного в лемме 2 базиса.

СПИСОК ЛИТЕРАТУРЫ

1. *Алексеев В.Б., Ларионов В.Б.* О расширениях с простым умножением для алгебры матриц. // Труды VII международной конференции "Дискретные модели в теории управляющих систем М. 2006 С. 17–22.
2. *Алексеев В.Б.* Минимальные расширения с простым умножением для алгебры матриц второго порядка. // Дискретная математика, 1997, т.9, № 1. С. 71–82.
3. *Маркус М., Минк Х.* Обзор по теории матриц и матричных неравенст. //М. Едиториал УРСС. 1994. 232 с.
4. *Coppersmith D., Winograd S.* Matrix multiplication via arithmetic progression. //J. Symb. Comp., 1990, 9, 251–280.
5. *Strassen V.* Gaussian elimination is not optimal. //Numer. Math., 1969, v.13, 454–456.
6. *Laderman J.D.* A noncommutative algorithm for multiplying 3*3 matrices using 23 multiplications. //Bull. Amer. Math. Soc.,1976, v.82, т.9, № 1,126–128.
7. *Плукас М.* О некоторых свойствах алгебр с простым умножением, содержащих ассоциативные подалгебры. //Дискретная математика, 1997, № 2, С. 79–90.
8. *Алексеев В.Б.* Сложность умножения матриц. Обзор. //кибернетический сборник М.: Мир, 1988, № 25, С. 189–236.
9. *Blaser M* Algebras of minimal rank over arbitrary fields. //Schriftenreihe der Institute fur Informatik/Mathematik Serie A, 2002.
10. *Алексеев В.Б.* О некоторых алгебрах, связанных с быстрыми алгоритмами. //Дискретная математика, 1996, т.8 № 1, С. 52–64.

СБОРНИК СТАТЕЙ МОЛОДЫХ УЧЕНЫХ факультета ВМиК МГУ, 2007, выпуск № 4, с. 90–98

УДК 519.68

ОБ ИНТЕГРАЦИИ SEMANTIC WEB С POSTGRESQL

© 2007 г. Д. В. Левшин

levshin@nicevt.ru

Кафедра Системного Программирования

Введение. Автором идеи Semantic Web считается Тим Бернерс-Ли. История концепции уходит корнями в середину 90-х годов XX века, первые детализированные публикации относятся к 1998 году. С 1999 года проект Semantic Web развивается под эгидой Консорциума Всемирной паутины (W3C). Началось внедрение этой концепции многими крупными компаниями и корпорациями (например, в Oracle 10g или в TopBraid Composer от IBM). Кроме того, Semantic Web активно пропагандируется и внедряется многими проектами с открытым исходным кодом.

Сейчас значительная часть содержания Всемирной сети (World Wide Web) предназначена для чтения человеком, а не для осмысленного манипулирования им с помощью компьютерных программ. Компьютер способен умело разобраться в разметке и произвести простейшую обработку веб-страницы (например, выделить заголовок, ссылку на другую страницу) но, вообще говоря, у компьютера нет надёжного способа обрабатывать семантику документа. Главным же свойством Semantic Web является то, что ее содержимое будет предназначено не только для человека, но и для машинной обработки.

Semantic Web, например, позволит поисковым системам получать ответы на более сложные запросы. Так, сегодняшние поисковые системы выполняют поиск по ключевым словам и часто выдают множество совершенно не относящихся к запросу результатов, заставляя пользователя вручную отбирать материал, что может потребовать у него длительное время. Например, если вы ввели для поиска слово «cook», то компьютеру совершенно непонятно, имеется ли в виду повар, информация о рецептах приготовления пищи, какое-то место, человек или компания или ещё что-либо, в чьём имени или названии встречается слово «cook». Возможность указания требуемого смысла слова (семантического содержания) позволило бы поисковым системам выдавать результаты, наиболее приближенные к требуемым.

При этом Semantic Web — это не какая-то отдельная сеть, а расширение уже существующей такое, что в ней информация снабжена точно определённым смыслом, позволяющим человеку и машине успешно взаимодействовать. Semantic Web сохраняет такое свойство Всемирной сети, как её универсальность. Кроме того, подобно Интернету, Semantic Web будет максимально децентрализован, что предоставляет определенные возможности и свободу пользователям, но приводит и к некоторым сложностям, связанным с согласованностью информации.

Основные форматы Semantic Web были разработаны в академических кругах и достаточно трудны для понимания широкого круга пользователей. Поэтому для дальнейшего развития идеи Semantic Web необходимо появление различных приложений, способных упростить использование возможностей Semantic Web. В данной работе предлагается алгоритм интеграции Semantic Web со свободно распространяемой СУБД PostgreSQL, разработанной в университете Беркли.

1. Основные форматы Semantic Web.

1.1. RDF (Resource Description Framework). Для функционирования Semantic Web необходима возможность определения хранилищ данных и правил для выполнения логического вывода, которые могут осуществляться децентрализовано различными пользователями. Для этого в феврале 2004 г. в качестве стандарта W3C для создания понятного компьютеру описания ресурсов в Semantic Web был утвержден формат RDF (Resource Description Framework, система описания ресурсов).

При определении языка задания метаданных в Semantic Web требовалось решить две следующие проблемы: он должен был позволять определять выражения произвольной сложности,

но в то же время эти выражения должны иметь форму достаточно простую для понимания машиной. Поэтому в RDF все утверждения описывается в виде триплетов (троек) вида «предмет (субъект) – свойство (предикат) – значение свойства (объект)». Однако определения жесткой структуры утверждений недостаточно для машинного понимания документов: компьютер должен однозначно понимать, что означают субъект, свойство и объект утверждения, чтобы не возникало ситуаций, когда при передаче между двумя машинами RDF документа, каждая из них понимала утверждения в этом документе по-своему. Поэтому, исходя из того, что вся информация является распределенной, нужно было определить некоторый унифицированный способ для представления ресурсов. Для этого в качестве уникальных идентификаторов были выбраны– URI (Uniform Resource Identifier). Кроме того, такой подход помимо однозначности понимания предоставляет дополнительную возможность записывать утверждения об субъектах, которые сами являются свойствами.

RDF – это система описания ресурсов, при этом стоит понимать, что описываемыми ресурсами могут быть не только сетевые адреса. Ресурсом является любая (физическая или абстрактная) сущность, имеющая уникальный идентификатор (URI).

В каждом триплете субъект и предикат являются ресурсами, а объект может быть или ресурсом, или некоторым литералом в формате UNICODE. Существует несколько представлений RDF: представление в виде триплетов (N-triples) удобно для начального изучения RDF, представление в виде графов достаточно наглядно, RDF/XML используется для передачи в сети.

1.2. RDF Schema (RDFS). Модель RDF включает в себя только идею написания утверждений в виде триплетов, использования для этих целей URI, а также словарь rdf:. Словарь rdf: является каркасом при написании RDF документов, позволяет формулировать простейшие утверждения, определять списки, контейнеры ресурсов, использовать типизированные литералы, использование пустых узлов для определения структурированных ресурсов, предоставляет возможность написания утверждений об утверждениях (RDF реификация). Однако пользователю может потребоваться написать свой собственный словарь, который он хочет использовать для своих целей. RDF не предоставляет возможностей определения специфичных для приложений классов и свойств. Эту возможность предоставляет расширение RDF – RDF Schema (RDFS).

RDFS предоставляет возможность определения иерархии классов, указания того, какой тип должны иметь субъект и объект для данного свойства. Заметим, что средства RDFS не определяют ограничивающие рамки для информации, а, наоборот, предоставляют дополнительную информацию об описываемых ресурсах (хотя конкретное приложение может воспринимать описания, полученные при помощи RDFS, как ограничения). Например, если, используя RDFS, мы определили, что некоторый класс X является подклассом класса Y (X rdfs:subClassOf Y), а затем при описании некоторого ресурса r указали, что он принадлежит классу X (r rdf:type X), то система, «понимающая» RDFS, может сделать вывод, что ресурс r также принадлежит классу Y (r rdf:type Y), то есть была получена дополнительная информация о ресурсе r.

Отметим, что RDFS (как и форматы описанные ниже) является словарем, который расширяет RDF. Поэтому приложение, разбирающее RDF документ, сможет разобрать и документы в формате RDFS, но без «понимания» смысла, приписанного к ресурсам из данного словаря. Для полноценной работы с документами в формате RDFS и других в приложениях должно быть реализовано «понимание» семантики этих форматов.

1.3. OWL (Web Ontology Language). RDFS предоставляет базовые возможности для описания словарей, но также очень полезными могут быть дополнительные возможности. Например, в RDF Schema нет следующих возможностей: ограничение количества значений свойств, определение транзитивности свойств, указание того, что два класса с различными URI представляют один класс, указание того, что два экземпляра класса с различными URI представляют один экземпляр класса, возможность описания классов с использованием операций над множествами (объединение, пересечение). Отметим, что возможность указания эквивалентности двух ресурсов с различными идентификаторами является важной исходя из

того, что ресурсы определяются децентрализовано различными пользователями. Эти и некоторые другие возможности предоставляет язык веб-онтолгий OWL, который также определен в форме словаря, расширяющего RDF и RDFS.

OWL предоставляет значительно более выразительные возможности, и реализация всех его особенностей может означать, что система, занимающаяся логическим выводом новых утверждений на основе имеющихся, может не давать результат за требуемое время. Кроме того, пользователю может и не требоваться всей выразительной мощности, предоставляемой средствами OWL. Поэтому было предложено три различных по выразительности диалекта OWL: 1) OWL Lite поддерживает тех пользователей, которые нуждаются, прежде всего, в классификационной иерархии и простых ограничениях. Реализация OWL Lite проще, чем двух других диалектов. 2) OWL DL поддерживает тех пользователей, которые хотят максимальной выразительности при сохранении полноты вычислений (все заключения гарантировано будут вычисляемыми) и разрешимости (все вычисления завершатся в определенное время). OWL DL включает все языковые конструкции OWL, но они могут использоваться только согласно определенным ограничениям. OWL DL так назван из-за его соответствия дескриптивной логике (Description Logic). 3) OWL Full предназначается для пользователей, которые хотят максимальную выразительность и синтаксическую свободу RDF без вычислительных гарантий.

Отметим, что различия между OWL Lite и OWL DL не столь большие и в большинстве приложений, работающих с OWL, поддерживается именно OWL DL. В то же время реализация OWL Full кажется маловероятной.

1.4. SWRL (Semantic Web Rule Language). Semantic Web Rule Language (SWRL) основан на комбинации OWL DL и Rule Markup Language (RuleML) с ограничением унарности / бинарности Datalog правил. Этот формат является расширением OWL, добавляя возможность определения Хорно-подобных правил. Предлагаемые правила имеют вид импликации между предпосылкой (телом) и следствием (заголовком). Смысл правил можно определить следующим образом: всегда, когда выполняется условие, определенное в предпосылке, то должно выполняться и условие, определенное в следствии.

И предпосылка, и следствие могут состоять из нуля или более атомов. Пустая предпосылка воспринимается как всегда истинная (то есть удовлетворяется в любой интерпретации), при этом следствие тоже должно удовлетворяться в любой интерпретации. Пустое следствие воспринимается как всегда ложное (то есть не удовлетворяется ни в одной интерпретации), при этом и предпосылка не удовлетворяется ни в одной интерпретации. Если предпосылка или следствие состоят из нескольких атомов, то они воспринимаются как конъюнкция этих атомов. Отметим, что правила со следствиями, состоящими из конъюнкции атомов, можно преобразовать в несколько правил, имеющих следствия, состоящие из одного атома. Атомы в этих правилах имеют вид C(x), P(x,y), sameAs(x,y), differentFrom(x,y), где C – OWL описание, P – OWL свойство, x, y – либо переменные (на их место можно подставить произвольный OWL идентификатор или литерал), OWL идентификатор или литерал (там, где это позволяет синтаксис OWL).

Имея возможности, предоставленные RDF, RDF Schema, OWL и SWRL можно писать документы, содержащие достаточно выразительные утверждения. Однако при реализации OWL Full и SWRL возникают проблемы с тем, что запросы к информации, написанной с их помощью, могут не разрешаться за требуемое время, или вычисления могут вообще не заканчиваться. Отметим, что для написания запросов предлагаются различные форматы (RDQL, RDFQL, RQL, SquishQL, RSQL), но наиболее распространенным на данный момент является SparQL.

2. Основные возможности PostgreSQL. PostgreSQL - это свободно распространяемая объектно-реляционная система управления базами данных, разработанная в Беркли.
PostgreSQL считается наиболее развитой из открытых СУБД в мире и является реальной альтернативой коммерческим базам данных.

Основные возможности, предоставляемые PostgreSQL, следующие:

1. Надежность PostgreSQL обеспечивается следующими возможностями: полное соответ-

ствие принципам ACID; использование многоверсионности для поддержания согласованности данных в конкурентных условиях; использование общепринятого механизма протоколирования транзакций Write Ahead Logging (WAL), который позволяет восстановить систему после возможных сбоев; Использование систем репликации, поддержание целостности данных на уровне схемы (использование внешних ключей и ограничений) и др.

2. Производительность PostgreSQL основывается на использовании индексов (B-tree, R-tree, Hash и GiST), интеллектуальном планировщике запросов, системы блокировок, системе управления буферами памяти и кэширования, масштабируемости при конкурентной работе.

3. Расширяемость PostgreSQL означает, что пользователь может настраивать систему путем определения новых функций, агрегатов, типов, языков, индексов и операторов.

4. Поддержка SQL. Стоит отметить, что PostgreSQL поддерживает высокий уровень соответствия ANSI SQL 92, ANSI SQL 99 и ANSI SQL 2003. В частности, PostgreSQL поддерживает правила (Rules), представления (Views) и триггеры.

5. Богатый набор типов данных и встроенных функций и операций для работы с ними.

6. Поддержка 25 различных наборов символов, включая ASCII и UNICODE.

7. Интерфейсы в PostgreSQL реализованы для доступа к базе данных из ряда языков (C,C++,C#,python,Perl,ruby,PHP,Lisp и другие) и методов доступа к данным (JDBC, ODBC). В частности, для доступа к PostgreSQL из программы, написанной на C, можно воспользоваться библиотекой libpq.

8. Возможность использования процедурных языков.

9. Обеспечение безопасности данных.

Стоит подробнее остановиться на некоторых возможностях PostgreSQL, которые особенно важны для интеграции с Semantic Web.

2.1. Триггеры. Триггер определяет, что база данных автоматически выполняет определенную функцию, когда выполняется операция некоторого типа. Триггер может быть определен для выполнения до (в этом случае сначала вызывается функция триггера, а затем выполняется вызвавшее триггер событие) или после (наоборот, сначала выполняется вызывающее триггер событие, а затем выполняется функция триггера) INSERT, UPDATE или DELETE операции, для каждого кортежа (в этом случае функция триггера будет вызываться для каждой вставляемой, обновляемой или удаляемой строки) или для всего SQL выражения (функция триггера вызывается один раз). Когда выполняется событие триггера, функция триггера выполняется в соответствующее время для обработки этого события.

2.2. Система правил и представления. Использование правил некоторым образом похоже на применение триггеров. В обоих случаях определяется некоторое событие (INSERT, UPDATE или DELETE, но правила могут вызываться и на SELECT) и действие которое выполняется при выполнении этого события. Но между ними есть множество различий. Во-первых, у правил можно определять помимо событий еще условия выполнения правила при помощи конструкции WHERE (кроме правил на SELECT). Во-вторых, при определении триггера выполняемым действием является вызов определенной функции, а у правил - SQL команды или NOTHING (не делать ничего). Но главным отличием между триггерами и правилами заключается в способе их выполнения.

Вызов триггеров происходит уже на этапе выполнения запроса. Система правил изменяет сам запрос, после чего выполняется уже измененный запрос. Система правил расположена

между парсером и планировщиком, она получает на вход дерево запроса (внутреннее представление SQL выражения, которое строится для выполнения оптимизации его выполнения), построенное парсером, и определенные пользователем правила в виде деревьев запросов с некоторыми дополнительными условиями и создает ноль или более деревьев запросов. Ноль деревьев запросов может быть получено, если вместо данного дерева запроса (если правило было определено как INSTEAD) подставляем правило, у которого действие определено как NOTHING. Более одного дерева запроса может получиться, когда к текущему дереву запросов (пусть оно имеет некоторое условие cond) применяется правило с некоторым условием rule_cond, при этом будет получено два дерева: одно будет иметь условие cond and not rule_cond и в него не будет подставлено правило, второе будет иметь условие cond and rule_cond и в него будет подставлено правило. При подстановке правил в исходное дерево запросов система правил смотрит, на какое событие они определены, для какой таблицы, при каких условиях и другую информацию в определении правила. Результат работы системы правил - деревья запросов, которые могли быть получены и парсером для некоторых SQL выражений. Такой способ выполнения может быть очень полезен, например, для представлений.

Представления - это виртуальные таблицы, реальных экземпляров этих таблиц не существуют. Для выполнения запросов или модификации представлений используется система правил.

2.3. Процедурные языки. В PostgreSQL можно создавать функции, написанные на SQL, C, а также на процедурных языках. При выполнении функции, написанной на процедурном языке, сервер базы данных не имеет встроенного знания о том, как ее интерпретировать, для этого он вызывает специальный обработчик, который знает все детали этого языка. Обработчик может сам выполнять всю работу по лексическому и синтаксическому анализу и выполнению функции или служить "клеем"между PostgreSQL и существующей реализацией языка программирования. Обработчик сам является функцией на языке C, скомпилированной в разделяемый объект и загруженной в базу данных, как любая другая функция на C в PostgreSQL.

В настоящее время в PostgreSQL представлены четыре процедурных языка: PL/pgSQL, PL/Tcl (для написания функций на Tcl), PL/Perl (позволяет писать функции на Perl, но с некоторыми ограничениями) и PL/Python (для написания функций на Python). Остальные процедурные языки могут быть определены пользователем. Отметим, что PL/pgSQL добавляет к SQL контрольные структуры и прост в использовании. За исключением вычислительных функций для определенных пользователем типов и преобразований ввода/вывода, все, что может быть определено в C функциях, может быть сделано и с PL/pgSQL.

2.4. GiST (Generalized Search Tree). Эффективный доступ к данным является одной из важнейших задач базы данных. Для больших баз данных, которые не помещаются в оперативную память, эффективность доступа к данным определяется, в основном, количеством обращений к диску, поэтому основной задачей СУБД является минимизация этих обращений. Обычно, это достигается использованием индекса, вспомогательной структуры данных, предназначенной для ускорения получения данных, удовлетворяющих определенным поисковым критериям. Индекс позволяет уменьшить количество дисковых операций необходимых для считывания данных с диска.

В PostgreSQL, кроме стандартных методов индексирования B-tree, R-tree и Hash, реализован метод индексирования GiST (Generalized Search Tree, Обобщенное поисковое дерево), предложенный профессором Беркли Джозефом Хеллерстейном. Преимущество GiST заключается в том, что с его помощью реализуется не только расширяемость типов (то есть возможность использования индексов для определенных пользователем типов данных), но и расширяемый набор запросов. Реализация расширяемого набора запросов является важной особенностью, так как позволяет поддерживать запросы естественные для этих типов. Достигается это благодаря тому, что при определении GiST структуры можно использовать произвольные предикаты, а не только операции сравнения, как у остальных индексных структур. С использованием GiST определение новых типов данных с поддержкой индексирования и естественных запросов становится достаточно простой задачей.

3. Предлагаемый алгоритм интеграции. Проведенный анализ спецификаций форматов Semantic Web показал, что базовым форматом является RDF, а все остальные форматы, представляющие большую семантическую выразительность, расширяют его. Система, способная распознавать RDF документы, сможет разбирать и документы в остальных форматах Semantic Web. Поэтому для интеграции Semantic Web необходимо наличие некоторого средства для разбора RDF документов. Формат RDF имеет несколько нотаций (например, представление в виде триплетов, графическое представление). Однако для передачи данных в сети используется XML нотация, получившая широкое распространение. Следовательно, необходимо, чтобы средство для разбора обрабатывало именно RDF/XML документы. Поэтому для интеграции предлагается разработка модуля, который бы подсоединялся к базе данных, выполнял разбор указанного ему RDF/XML документа, строил по документу соответствующие триплеты и сохранял их в базе данных.

Для хранения информации о ресурсах и литералах в базе данных определяется таблица pg_rdf_values, в которой устанавливается взаимнооднозначное соответствие между ресурсами (литералами) и целочисленными идентификаторами, а также указывается тип ресурса.

```
CREATE TABLE pg_rdf_values (
value_id int PRIMARY KEY,
text_value text NOT NULL,
value_type int NOT NULL,
literal_type oid
);
```

Для типизированных литералов столбец literal_type указывает на строку в таблице pg_rdf_type, хранящей информацию о типах литералов.

```
CREATE TABLE pg_rdf_type (
typname text PRIMARY KEY,
typinput regproc,
typoutput regproc
) WITH oids;
```

Здесь typname – имя типа, typinput, typoutput определяют функции для преобразования строкового представления в значения требуемого типа. Отметим, что в pg_rdf_values каждый ресурс и литерал хранится один раз. Однако возможна следующая ситуация, что в документе встретились литералы «1» и «+1» целочисленного типа. Если их не преобразовывать, то они будут занимать две строки в таблице, сравнение литералов на равенство будет давать неверный результат. Поэтому при сохранении в базу данных модуль разбора для каждого типизированного литерала применяет функцию, указанную в таблице pg_rdf_type, которая преобразует текстовое представление литерала в его тип данных (в нашем примере в целое число 1), а затем полученное значение второй функцией преобразуется в строковое представление («1»), которое и сохраняется в таблицу pg_rdf_values. Отметим, что для сохранения литерала или ресурса можно использовать правило, триггер или вызывать функцию, которые будут гарантировать отсутствие дубликатов.

Для хранения троек, построенных модулем разбора предлагается использовать таблицу pg_rdf_triples, определенную следующим образом:

```
CREATE TABLE pg_rdf_triples (
subject_id int REFERENCES pg_rdf_values(value_id),
property_id int REFERENCES pg_rdf_values(value_id),
object_id int REFERENCES pg_rdf_values(value_id),
isinferenced smallint DEFAULT 0
) WITH oids;
```

Первые 3 атрибута используются для хранения субъекта, объекта и свойства, соответственно. Последний атрибут указывает, был ли получен данный триплет непосредственно (путем разбора) из какого-либо документа или путем логического вывода (на основании полученных данных и построенных правил). Заметим, что использование целочисленных идентификаторов вместо текстового представления ресурсов и литералов для хранения троек может значительно сократить используемую память, так как ресурсы, определенные в словарях RDF, RDFS,

OWL и SWRL могут встречаться очень часто.

Информация о типе ресурса хранится в таблице pg_rdf_valtypes

```
CREATE TABLE pg_rdf_valtypes (
value_id int PRIMARY KEY,
value_type varchar(8),
type_description text
);
```

Стоит отметить, что представление данных в виде троек чисел становится совершенно непонятным для пользователя, поэтому было определено представление для наглядного отображения этих данных пользователю – pg_rdf_view.

Для выполнения запросов к хранящейся информации наиболее удобным является также представление в виде троек. Предлагается задавать переменные, которые требуется найти в результате запроса, в виде идентификаторов, в начале которых стоит символ «?», остальные идентификаторы воспринимаются как URI или литералы. Примером является запрос следующего вида:

'(?a ex:hasFather ?b) (?b ex:married ex:Ann)'

В результате запроса должно быть получена таблица со столбцами a, b, смысл которых "a имеет отца b, a b женат на Ann".

Для осуществления данных запросов в PostgreSQL можно использовать табличные функции. Так как требуется преобразование из данного текстового представления в запрос на SQL, можно использовать табличную функцию на PL/Perl (Perl удобен для обработки текстов) RDF_QUERY, которая разобрав образец в запросе затем будет вызывать необходимые функции для выполнения запроса. Запросы с использованием RDF_QUERY имеют следующий вид:

```
SELECT * FROM RDF_QUERY('(?a hasFather ?b) (?b married Ann)') AS res (a text, b text);
```

Указанных таблиц, триггеров, правил и функций было бы достаточно для работы с RDF документами. Однако для того чтобы работать со всеми форматами Semantic Web, необходимо реализовать также семантику, заключенную в них. Осуществление логического вывода, основанного на сохраненных триплетах и имеющейся семантике, предлагается выполнять за счет средств самой СУБД: правил и триггеров. Для этого изменяем модуль разбора так, чтобы в зависимости от типа документа, он по-разному производил разбор. Для OWL и SWRL не сохранять все полученные тройки, а сразу автоматически генерировать правила в базе данных, которые будут осуществлять логический вывод.

Правила выполняются, когда происходит некоторое событие. Поэтому для осуществления логического вывода при выполнении запроса предлагается использовать специальную таблицу pg_rdf_tasks, в которую заносились бы образцы, по которым нужно активизировать правило, состоящие из идентификаторов предиката и двух аргументов. При этом аргументы могут принимать значение NULL, которое будет означать, что данный аргумент является переменной, а не литералом. Свойство не может принимать значение NULL, поэтому и при задании запроса в RDF_QUERY свойства должны быть определенными ресурсами, а не переменными.

Рассмотрим следующий пример: Пусть в базе данных сохранено правило
P(?x,?y) & P(?y,?z) $\rightarrow$ R(?x,?z) и имеется запрос ?R(?x,a). Тогда для данного запроса в pg_rdf_tasks нужно занести строку со значением идентификатора, соответствующего R, NULL для первого аргумента и идентификатора, соответствующего a, для второго аргумента. Однако для свойства P тоже могут быть определены правила, которые необходимо активизировать.

Чтобы решить эту проблему, используются две дополнительные таблицы pg_rdf_old и pg_rdf_new с такими же столбцами, как у pg_rdf_tasks. Модуль разбора для каждого SWRL правила и OWL правил генерирует дополнительны правила в базе данных, которые позволяют определить те свойства, которые надо занести в pg_rdf_tasks. Для примера, приведенного выше, нужно определить правило, которое при вставке в pg_rdf_old строки, содержащей свойство R, в pg_rdf_new будет вставлено две строки со свойством P и соответствующими значениями аргументов. На pg_rdf_new определен триггер, который проверяет, есть ли уже в pg_rdf_old такой кортеж; если нет, то вставляет его в pg_rdf_old, иначе не делает ничего. Кроме того, триггер может для каждой новой тройки с некоторым предикатом P, у которой

один аргумент (например, первый) является литералом или ресурсом, то есть не равен NULL, определить количество кортежей в pg_rdf_old с тем же предикатом и таким же значением второго аргумента (NULL, если у нового кортежа NULL), но разными значениями первого аргумента; если это количество больше некоторой заданной константы, то все такие кортежи из pg_rdf_old удаляются и вставляется кортеж со свойством P, тем же значением второго аргумента и NULL вместо первого аргумента. Также отметим, что триггер должен реализовывать «поглощение» переменными определенных значений, то есть, например, если в pg_rdf_old имеется строка со свойством P, у которой первый аргумент равен NULL, то при попытке вставки такого же кортежа, но со значением первого аргумента равного некоторому числу, эта вставка будет запрещена. Данный алгоритм определения активизированных завершится, так как количество различных свойств в базе данных конечно. Так же триггер гарантирует, что для каждого значения свойства в pg_rdf_old будут вставлены кортежи, число которых не больше некоторого числа.

Исходя из сказанного выше для определения тех свойств, которые «активизированы» в результате данного запроса, при выполнении запроса достаточно будет занести тройки со значениями свойств и аргументов, соответствующими запросу, все остальное автоматически выполнят правила и триггеры, определенные на pg_rdf_old и pg_rdf_new. После чего выполняется цикл, в котором каждый кортеж из pg_rdf_old вставляется в pg_rdf_tasks. Так как на pg_rdf_tasks определены правила, сгенерированные модулем разбора, будет выполняться логический вывод новых триплетов.

Для выполнения логического вывода определяется таблица pg_rdf_inferenced:

```
CREATE TABLE pg_rdf_inferenced (
subject_id int,
property_id int,
object_id int,
mark int DEFAULT 0
);
```

В этой таблице хранятся выведенные триплеты в виде троек целых чисел, а также хранится служебный атрибут mark для осуществления рекурсии при логическом выводе.

Правила генерируются модулем разбора так, чтобы новые кортежи вставлялись в эту таблицу. На таблице определяется триггер, который проверяет, чтобы вставлялись только новые кортежи, а уже имеющиеся игнорировались. После выполнения каждой итерации цикла происходит проверка, были ли получены новые кортежи (для этого проверяется атрибут mark). Если есть, то значение атрибута mark у новых кортежей меняется, и выполняется новая итерация цикла. Иначе цикл завершается, и выдается результат.

Заключение. Использование Semantic Web может принести пользователям WWW новые возможности, например, интеллектуальный поиск не только по ключевым словам, но и исходя из смысла вопроса. Но для того, чтобы Semantic Web получил широкое распространение необходимо, чтобы стали доступны приложения для создания, хранения и запросов к данным Semantic Web. Некоторые крупные компании и корпорации уже уделяют внимание к таким приложениям. Появляются и проекты с открытым кодом, пропагандирующие Semantic Web. В данной работе предлагаются методы интеграции PostgreSQL с Semantic Web, основанные на тех широких возможностях, которые предлагает СУБД.

СПИСОК ЛИТЕРАТУРЫ

1. *Tim Berners-Lee, James Hendler, Ora Lassila* The Semantic Web // Scientific American, 2001
2. *Frank Manola, Eric Miller, eds.* RDF Primer W3C Recommendation 10 February [HTML] (http://www.w3.org/TR/2004/REC-rdf-primer-20040210/)
3. *Michael K. Smith, Chris Welty, Deborah L. McGuinness* OWL Web Ontology Language Guide W3C Recommendation 10 February 2004 [HTML] (http://www.w3.org/TR/2004/REC-owl-guide-20040210/)
4. *Harold Boley, Benjamin Grosof, Said Tabet* RuleML Tutorial Draft, 13 May 2005 [HTML] (http://www.ruleml.org/papers/tutorial-ruleml-20050513.html)
5. *Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, Mike Dean* SWRL: A Semantic Web Rule Language Combining OWL and RuleML W3C Member Submission 21 May 2004 [HTML] (http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/)
6. *Eric Prud'hommeaux, Andy Seaborne* SPARQL Query Language for RDF W3C Working Draft 4 October 2006 [HTML] (http://www.w3.org/TR/rdf-sparql-query/)
7. *Бартунов О.* Что такое PostgreSQL? [HTML] (http://www.sai.msu.su/ megera/postgres/talks/what_is_postgresql.html)
8. PostgreSQL 8.2.3 Documentation [HTML] (http://www.postgresql.org/docs/8.2/static/index.html)
9. *M. Stonebraker, A. Jhingran, J. Goh, and S. Potamianos* On Rules, Procedures, Caching and Views in Database Systems, Proc. ACM-SIGMOD Conference on Management of Data, June 1990.
10. *Бартунов О., Сигаев Ф.* Написание расширений для PostgreSQL с использованием GiST [HTML] (http://www.sai.msu.su/ megera/postgres/talks/gist_tutorial.html)

СБОРНИК СТАТЕЙ МОЛОДЫХ УЧЕНЫХ факультета ВМиК МГУ, 2007, выпуск № 4, с. 99–106

УДК 519.834

КОАЛИЦИОННО-УСТОЙЧИВОЕ РАВНОВЕСИЕ УГРОЗ И КОНТРУГРОЗ. МАТРИЧНЫЙ СЛУЧАЙ

© 2007 г. З. С. Мальсагов

mals-zs@mail.ru

Кафедра Оптимального управления

1. Постановка задачи

Рассмотрим коалиционную динамическую линейно-квадратичную игру трех лиц

$$\langle \{1,2,3\}, \Sigma, \{\mathfrak{U}_i\}_{i=1,2,3}, \{J_i(u)\}_{i=1,2,3} \rangle \tag{1}$$

здесь $\{1,2,3\}$ — порядковые номера игроков, Σ — управляемая система, $U_i \in \mathfrak{U}_i$ — позиционная стратегия i-го игрока , за счёт выбора каждым игроком своей стратегии формируется ситуация $U = (U_1, U_2, U_3)^T$, $U_i \in \mathfrak{U}_i$, на множестве всех ситуаций определены функции выигрыша игроков $\{J_i(u)\}_{i=1,2,3}$. Предполагается, что фиксирован момент окончания игры: $\vartheta > 0$.

Дадим более подробное пояснение элементам (1):

Управляемая система описывается системой неоднородных нестационарных линейных уравнений:

$$\dot{x} = A(t)x + \sum_{i=1}^{3} u_i(t,x) + a(t), \quad , x(t_0) = x_0, \tag{2}$$

где $x \in \mathbb{R}^n$ — фазовый вектор, $A(t) \in C_{n\times n}[0,\vartheta]$ — матрица с непрерывными элементами, $u_i(t,x)$ — позиционное управление i-го игрока (определено ниже), $a(t) \in C_n[0,\vartheta]$ — напрерывная вектор-функция, а $(t_0,x_0) \in [0,\vartheta) \times \mathbb{R}^n$ — начальная позиция, которую мы не будем предполагать априори зафиксированной (так как игра рассматривается в позиционных стратегиях).

Множество позиционных стратегий i-го игрока имеет следующее формальное определение:

$$\mathfrak{U}_i = \{U_i \div u_i(t,x) | u_i(t,x) = P_i(t)x + p_i(t),\ \forall P_i \in C_{n\times n}[t_0,\vartheta], p_i \in C_n[t_0,\vartheta].\}, \quad i = 1,2,3. \tag{3}$$

Для выбора своей стратегии каждому игроку необходимо выбрать матрицу с непрерывными элементами $P_i(t)$ и непрерывную вектор-функцию $p_i(t)$. Как будет показано ниже, такого класса стратегий достаточно для построения подходящим образом определённого решения игры. Заметим также, что при подстановке стратегий игроков в управляемую систему получается векторное линейное неоднородное уравнение с непрерыными коэффициентами; согласно [?], такое уравнение всегда имеет единственное непрерывное решение, продолжимое на $[0,\vartheta]$. Подставив это решение $x(t)$ в позиционные стратегии игроков, можно получить *реализацию* стратегии для каждого игрока: $u_i[t] = P_i(t)x(t) + p_i(t)$, $i = 1,2,3$, тогда $u[t] = (u_1[t], u_2[t], u_3[t])$.

Функция выигрыша i-го игрока задаётся линейно-квадратичным интегрально-терминальным функционалом:

$$J_i[u] = x^T(\vartheta)C^{(i)}x(\vartheta) + x^T(\vartheta)c^{(i)} +$$
$$+ \int_{t_0}^{\vartheta} \{u^T[t]D^{(i)}(t)u[t] + 2u^T[t]d^{(i)}(t) + 2x^T(t)K^{(i)}(t)u[t] + x^T(t)G^{(i)}(t)x(t) + 2x^T(t)g^{(i)}(t)\}dt =$$
$$= x^T(\vartheta)C^{(i)}x(\vartheta) + x^T(\vartheta)c^{(i)} +$$
$$+ \int_{t_0}^{\vartheta} \{\sum_{l,\,j=1}^{3} u_l^T[t]D_{jl}^{(i)}(t)u_j[t] + 2\sum_{j=1}^{3} u_j^T[t]d_j^{(i)}(t) + 2\sum_{j=1}^{3} x^T(t)M_j^{(i)}(t)u_j[t] +$$
$$+ x^T(t)G^{(i)}(t)x(t) + 2x^T(t)g^{(i)}(t)\}dt, \quad i = 1,2,3. \quad (4)$$

$$C \in \mathbb{R}^{n\times n},\ c \in \mathbb{R}^n,\ D_{jl}^{(i)}(\cdot), G^{(i)}(\cdot), M_j^{(i)}(\cdot) \in \mathrm{C}_{n\times n}[t_0,t_1],\ d_j^{(i)}(\cdot), g(\cdot) \in \mathrm{C}_n[t_0,t_1],\ i,j,l = 1,2,3.$$

где $C^{(i)} \in \mathbb{R}^{n\times n}$ — постоянная симметричная матрица, $c^{(i)} \in \mathbb{R}^n$ — постоянный вектор, $D^{(i)}(t) \in C_{3n\times 3n}[0,\vartheta]$ — непрерывная блочная симметричная матрица порядка $3n \times 3n$, её элементы — матрицы порядка $n \times n$, $D_{jk}^{(i)}(t)$; $M^{(i)}(t) \in C_{n\times 3n}[0,\vartheta]$ — непрерывная блочная симметричная матрица порядка $n \times 3n$, её элементы — матрицы порядка $n \times n$, $M_j^{(i)}(t)$; $d^{(i)}$ — "блочный" вектор порядка $3n$, $d_j^{(i)}(t)$ — его компоненты, вектора размерности n; верхний индекс означает порядковый номер игрока, а нижние — номер элемента блочной матрицы.

Игроки не могут обмениваться долями выигрышей, но могут обмениваться информацией. Игра происходит следующим образом: сначала за счёт переговоров между игроками скадывается одна из следующих коалиционных структур:

$$\mathfrak{K} = \{\ \{\{1,2\},\{3\}\},\ \{\{1,3\},\{2\}\},\ \{\{2,3\},\{1\}\}\ \}, \quad (5)$$

а затем игра происходит уже как бескоалиционная игра двух лиц с векторными выигрышами, где в качестве игроков выступают сложившиеся коалиции.

В статье исследуются возможные подходы к понятию равновесия для игры (1), учитывающие не только наличие исходной коалиционной структуры, но и факт возможного её изменения в ходе игры. Ниже также будут получены достаточные условия существования подходящим образом определённого равновесия в игре (1).

2. Вспомогательные сведения Определение 1. Вектор a размерности n больше по Парето $(>_p)$ вектора b той же размерности, если

$$\forall i \in 1..n \quad a_i \geq b_i,$$
$$\exists j \in 1..n \quad a_j > b_j,$$

иными словами, если по крайней мере одна компонента a больше соответствующей компоненты b, а остальные компоненты a не меньше соответствующих компонент b.

Рассмотрим бескоалиционную игру двух лиц с векторными функциями выигрыша

$$\langle\{1,2\}, \{\mathfrak{V}_i\}_{i=1,2}, \{\{f_i(v)\}_{i=1,\ldots,k}, \{g_j(v)\}_{j=1,\ldots,l}\}\rangle. \quad (6)$$

здесь $\{1,2\}$ — порядковые номера игроков, $\{\mathfrak{V}_i\}_{i=1,2}$ — множества стратегий игроков, на множестве ситуаций $v = (v_1, v_2) \in \mathfrak{V}_1 \times \mathfrak{V}_2$ определены векторные функции выигрыша $f(v)$, $g(v)$, размерности k и l соответственно, которые игроки стремяться максимизировать за счет выбора своих стратегий.

Определение 2. Ситуацию $(v_1^p, v_2^p) \in \mathfrak{V}_1 \times \mathfrak{V}_2$ назовём Парето-максимальной в игре (3), если не существует такой пары стратегий $(v_1, v_2) \in \mathfrak{V}_1 \times \mathfrak{V}_2$, что вектор $(f(v_1, v_2),\ g(v_1, v_2))$ размерности $k + l$ больше по Парето вектора $(f(v_1^p, v_2^p), g(v_1^p, v_2^p))$.

Будем говорить, что первый игрок обладает угрозой на ситуацию (v_1, v_2) в игре (3), если $\exists v_1^t \in \mathfrak{V}_1$ такая, что

$$f(v_1^t, v_2) >_p f(v_1, v_2).$$

Будем говорить, что второй игрок обладает контругрозой на угрозу v_1^t первого игрока на ситуацию (v_1, v_2) в игре (3), если $\exists v_2^c \in \mathfrak{V}_2$ такая, что

$$\begin{cases} g(v_1^t, v_2^c) >_p g(v_1^t, v_2), \\ f(v_1^t, v_2^c) <_p f(v_1, v_2). \end{cases}$$

Аналогично определяется угроза второго игрока и контругроза первого.

Определение 3. Ситуация $(v_1^*, v_2^*) \in (\mathfrak{V}_1 \times \mathfrak{V}_2)$ называется паретовским равновесием угроз и контругроз в игре (3), если

а) она Парето-максимальна в игре (3);

б) на любую угрозу любого игрока на ситуацию (v_1^*, v_2^*) у оставшегося имеется контругроза.

Л е м м а 1. [**?**] Для того, чтобы ситуация $v^p \in \mathfrak{V}_1 \times \mathfrak{V}_2$ была максимальной по Парето в игре (3) достаточно существования таких постоянных $\alpha_i \in (0,1), i = (1, ..., k), \beta_j \in (0,1), j = (1, ..., l), \sum_{i=1}^k \alpha_i + \sum_{j=1}^l \beta_j = 1,$ что выполнено равенство

$$v^p = \underset{v \in \mathfrak{V}_1 \times \mathfrak{V}_2}{\operatorname{argmax}} \left(\sum_{i=1}^k \alpha_i f_i(v) + \sum_{j=1}^l \beta_j g_j(v) \right),$$

если $\mathfrak{V}_1 \times \mathfrak{V}_2$ — выпукло, а $f_i(v),\ i = (1, ..., k),\ g_j(v),\ j = (1, ..., l)$ — вогнуты по v, то данное условие становится и необходимым ■.

3. Понятие равновесия

Следуя [**?**], предложим понятие коалиционно-устойчивого равновесия угроз и контругроз в игре (1). В дополнение к ранее введённым понятиям угрозы и контругрозы для игры с векторными выигрышами, что эквивалентно случаю постоянной коалиционной структуры, введем понятия угрозы и контругрозы, учитывающие возможность образования других коалиций.

Рассмотрим угрозу на коалиционную структуру игры: в этом случае цель "угрожающей" коалиции — переманить на свою сторону игрока из другой коалиции. Зафиксируем в игре (1) коалиционную структуру K_1, состоящая из двух коалиций $k_1 = \{1, 2\}$, и $k_2 = \{3\}$. Будем считать, что коалиция k_2 обладает $\mathfrak{k}_i$-угрозой $(i = 1, 2)$ на коалицию k_1 из ситуации u^*, если

$$\begin{aligned} \exists u_{k_2}^T \in \mathfrak{U}_{k_2} : \quad & J_i(u_{k_1}^*, u_{k_2}^T) > J_i(u^*), \\ & J_3(u_{k_1}^*, u_{k_2}^T) > J_3(u^*). \end{aligned} \tag{7}$$

Заметим, что в соответствии с концепцией угроз и контругроз равновесная ситуация будет Парето-оптимальной, следовательно, в рассматриваемом в статье случае, коалиция k_1 не может быть заинтересована во включении в свой состав 3-го игрока — единственного в k_2.

В ответ на $\mathfrak{k}_i$-угрозу $u_{k_2}^T$ коалиции k_2 на коалицию k_1 из ситуации u^* у коалиции k_1 есть $\mathfrak{k}_i$-контругроза, если:

$$\begin{aligned} \exists u_{k_1 \setminus \{i\}}^C \in \mathfrak{U}_{k_1 \setminus \{i\}} : & \\ \forall l \in k_1, \quad & J_l(u_{k_1 \setminus \{i\}}^C, u_i^*, u_{k_2}^T) > J_l(u_{k_1}^*, u_{k_2}^T), \\ & J_3(u_{k_1 \setminus \{i\}}^*, u_i^*, u_{k_2}^T) < J_3(u^*). \end{aligned} \tag{8}$$

В данном случае контругроза осуществляется всеми игроками из k_1, кроме i-ого — того, которого пытаются переманить.

Для других коалиционных структур из $\mathfrak{K}$ понятия угрозы и контругрозы определяются аналогично. Иными словами, коалиционная $(\mathfrak{k}-)$ угроза означает попытку ряда игроков перейти от одной допустимой коалиционной стуктуры к другой из данной ситуации, а $\mathfrak{k}-$ контругроза других игроков пресекает такую попытку.

Приведём определение равновесной ситуации, учитывающее возможные изменения в коалиционной структуре игры. Пусть $K \in \mathfrak{K}$ — допустимая коалиционная структура в игре (1).

Определение 5. Ситуация $u^* = (u_1^*, u_2^*, u_3^*)^T$ называется $\mathfrak{k}$-равновесием угроз и контругроз в игре (1) при коалиционной структуре K, если:

а) (*Парето оптимальность.*) Она является Парето-оптимальной в игре (1).

б) (*Угрозы на ситуацию.*) На любую угрозу любой коалиции из K на u^* у оставшейся коалиции имеется контругроза.

в) (*Угрозы на коалиционную структуру.*) На любую $\mathfrak{k}_i$-угрозу коалиции k_l, $k_l \in K$ на коалицию $(k_j,\ i \in k_j,\ k_j \neq k_l,\ k_j \in \mathfrak{k})$ из ситуации u^*, у коалиции k_l имеется $\mathfrak{k}_i$-контругроза.

4. Условия существования

Введём следующие обозначения:

$$D^{(i)}_{k_1 k_1}(t) = \begin{pmatrix} D^{(i)}_{1\,1}(t) & D^{(i)}_{1\,2}(t) \\ D^{(i)}_{2\,1}(t) & D^{(i)}_{2\,2}(t) \end{pmatrix},\ D^{(i)}_{k_1 k_2}(t) = \begin{pmatrix} D^{(i)}_{1\,3}(t) \\ D^{(i)}_{2\,3}(t) \end{pmatrix},$$

$$d^{(i)}_{k_1}(t) = \begin{pmatrix} d^{(i)}_1(t) \\ d^{(i)}_2(t) \end{pmatrix},\ M^{(i)}_{k_1}(t) = \begin{pmatrix} M^{(i)}_1(t) \\ M^{(i)}_2(t) \end{pmatrix},$$

$$D^{(i)}_{k_2 k_1}(t) = \begin{pmatrix} D^{(i)}_{3\,1}(t) & D^{(i)}_{3\,2}(t) \end{pmatrix},\quad D^{(i)}_{k_2 k_2}(t) = \begin{pmatrix} D^{(i)}_{3\,3}(t) \end{pmatrix},$$

$$d^{(i)}_{k_2}(t) = \begin{pmatrix} d^{(i)}_3(t) \end{pmatrix},\quad M^{(i)}_{k_2}(t) = \begin{pmatrix} M^{(i)}_3(t) \end{pmatrix}.$$

Напомним, что матрицы $D^{(i)}(t)$ симметричны. Докажем следующую основную лемму в 4-х вариантах.

Л е м м а 2а. Если найдётся такая точка $t_1 \in (t_0, \vartheta)$, что в ней суммы элементов матриц $D^{(1)}_{k_1 k_1}$ и $D^{(2)}_{k_1 k_1}$ положительны, а сумма элементов матрицы $D^{(3)}_{k_1 k_1}$ отрицательна, то коалиция k_1 может при любой ситуации в игре только за счет выбора своей стратегии одновременно сделать:

а) свой выигрыш больше по Парето любого наперед заданного вектора,

б) выигрыш коалиции k_2 меньше любого наперед заданного числа.

Л е м м а 2б. Если найдётся такая точка $t_1 \in (t_0, \vartheta)$, что в ней сумма элементов матрицы $D^{(3)}_{k_2 k_2}$ положительна, а суммы элементов матриц $D^{(1)}_{k_2 k_2}$, $D^{(2)}_{k_2 k_2}$ отрицательны, то коалиция k_2 может при любой ситуации в игре только за счет выбора своей стратегии одновременно сделать:

а) свой выигрыш больше любого наперед заданного числа,

б) выигрыш коалиции k_1 меньше по Парето любого наперед заданного вектора.

Л е м м а 2в. Если найдётся такая точка $t_1 \in (t_0, \vartheta)$, что в ней сумма элементов матриц $D^{(1)}_{1\,1}$ и $D^{(2)}_{1\,1}$ положительна, а сумма элементов матрицы $D^{(3)}_{1\,1}$ отрицательна, то 1-ый игрок может при любой ситуации в игре только за счет выбора своей стратегии одновременно сделать:

а) свой выигрыш больше любого наперед заданного числа,

б) выигрыш 2-го игрока больше любого наперёд заданного числа.

в) выигрыш 3-го игрока меньше любого наперёд заданного числа.

Л е м м а 2г. Если найдётся такая точка $t_1 \in (t_0, \vartheta)$, что в ней сумма элементов матриц $D^{(1)}_{2\,2}$ и $D^{(2)}_{2\,2}$ положительна, а сумма элементов матриц $D^{(3)}_{1\,1}$ отрицательна, то 2-ой игрок может при любой ситуации в игре только за счет выбора своей стратегии одновременно сделать

а) свой выигрыш больше любого наперед заданного числа,

б) выигрыш 1-го игрока больше любого наперёд заданного числа.

в) выигрыш 3-го игрока меньше любого наперёд заданного числа.

Доказательство. Проведём на примере леммы 3а. Зафиксируем стратегию третьего игрока $u_3^*[t] = P_3^*(t)x(t) + p_3^*(t)$, обозначим

$$\overline{u}(t,x) = (u_1(t,x), u_2(t,x)), \widetilde{u}(t,x) = u_1(t,x) + u_2(t,x),$$

$$\overline{A}(t) = A(t) + P_3^*(t), \overline{a}(t) = a(t) + p_3^*(t),$$

$$\overline{G}^{(i)}(t) = G^{(i)}(t) + (P_3^*(t))^T D^{(i)}_{k_2\,k_2}(t) P_3^*(t) + M^{(i)}_{k_2}(t) P_3^*(t) + (P_3^*(t))^T (M^{(i)}_{k_2}(t))^T,$$

$$\overline{g}(t) = g(t) + (p_3^*(t))^T d^{(i)}_{k_2} + M^{(i)}_{k_2}(t) p_3^*(t),$$

$$\overline{M}^{(i)}(t) = M^{(i)}_{k_1} + (P_3^*(t))^T D^{(i)}_{k_2\,k_1},$$

$$\overline{r}(t) = 2(p_3^*(t))^T d^{(i)}(t) + (p_3^*(t))^T D^{(i)}_{k_2\,k_2}(t) p_3^*(t).$$

$$\overline{D}^{(i)}(t) = D^{(i)}_{k_1\,k_1}(t),\ \overline{d}^{(i)}(t) = d^{(i)}_{k_1}(t)$$

Таким образом, мы приходим к следующей постановке задачи:

$$\begin{cases} \dot{x}(t) = \overline{A}(t)\,x(t) + \widetilde{u}(t,x) + \overline{a}(t), \\ x(t_0) = x_0; \\ J^{(i)}[u] = x(t_1)'Cx(t_1) + c'x(t_1) + \\ + \int_{t_0}^{t_1} \{\overline{u}[t]'\overline{D}(t)\overline{u}[t] + \overline{d}(t)'\overline{u}[t] + x(t)'\overline{G}(t)x(t) + \overline{g}(t)'x(t) + x(t)'\overline{M}(t)\overline{u}[t]\}dt,\ i = 1,2,3. \\ \overline{u}(t,x) = (Q^1(t)x, Q^2(t)x) + (q^1(t), q^2(t)) = Q(t)x + q(t), \\ \widetilde{u}(t,x) = (Q^1(t)x + Q^2(t)x) + (q^1(t) + q^2(t)). \end{cases} \tag{9}$$

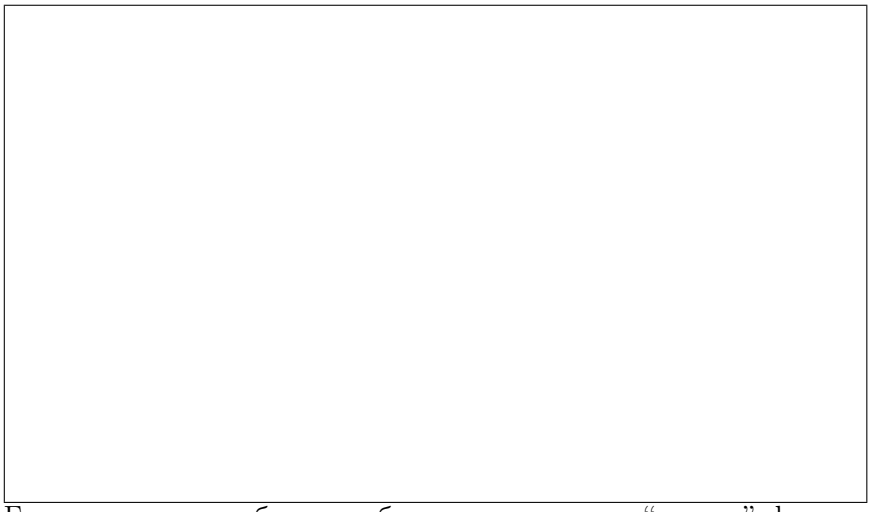
Причём первый и второй функционал необходимо сделать больше любого наперёд заданного числа, а третий — меньше. Без ограничения общности будем считать, что предположение леммы выполняется в точке $\frac{1}{2}(t_0 + \vartheta)$ — все дальнейшие рассуждения можно аналогично повторить и для любой другой точки интервала (t_0, ϑ). Сумма непрерывных функций непрерывна, и из положительности непрерывной функции в точке следует её положительность на некоторой отрезке, содержащем эту точку. Пусть $[t_1, t_2]$ — отрезок, на котором суммы элементов матриц $D^{(1)}_{k_1\,k_1}$ и $D^{(2)}_{k_1\,k_1}$ положительны, а сумма элементов матрицы $D^{(3)}_{k_1\,k_1}$ отрицательна; $t_1 < \frac{1}{2}(t_0 + \vartheta) < t_2$.

Мы проведём рассуждения только для первого функционала — всё то же самое будет верно и для второго, а для третьего измениться только знак неравенства в оценках; здесь существенно то, что для всех трёх функционалов используется одна и та же стратегия. Построим явно последовательность функций из $\mathfrak{U}$, которая удовлетворяет условиям леммы. Для этого необходимо указать непрерывные функции-матрицы $Q^1_m(t), Q^2_m(t)$ и вектор-функции $q^1_m(t), q^2_m(t)$. Положим $Q^1_m(t) = Q^2_m(t) = -\frac{1}{2}\overline{A}(t)\ \forall m > 0$. Функции $q^1_m(t), q^2_m(t)$ устроены чуть более сложно:

$$q^1_m(t) = q^2_m(t) = -\frac{1}{2}\overline{a}(t) + e v_m(t),$$

$$v_n(t) = \begin{cases} 0, & t < t_0 + \frac{t_1 - t_0}{2} - \frac{1}{2m} \\ 4m^2 t - 4m^2(t_0 + \frac{t_1 - t_0}{2} - \frac{1}{2m}), & t_0 + \frac{t_1 - t_0}{2} - \frac{1}{2m} \leqslant t < t_0 + \frac{t_1 - t_0}{2} \\ -4m^2 t + 4m^2(t_0 + \frac{t_1 - t_0}{2} + \frac{1}{2m}), & t_0 + \frac{t_1 - t_0}{2} < t \leqslant t_0 + \frac{t_1 - t_0}{2} + \frac{1}{2m} \\ 0, & t \leqslant t_0 + \frac{t_1 - t_0}{2} + \frac{1}{2m} \end{cases} \tag{10}$$

Здесь e — вектор из n единиц.
Например, для $t_0 = 0, t_1 = 1, m = 10$ функция $v_m(t)$ имеет следующий график:



Без ограничения общности будем считать, что "иголка" функции $v_m(t)$ укладывается в интервал $[t_1, t_2]$ для всех $m > 0$ (иначе просто будем считать не с нуля, а с первого m, для которого это выполнено). Непосредственным вычислением убеждаемся, что:

$$\int_{t_0}^{\vartheta} v_m(t)dt = 1$$

$$\int_{t_0}^{\vartheta} v_m^2(t)dt = \frac{4m}{3}$$

Далее, решая систему дифференциальных уравнений находим, что:

$$x_m(t) = x_0 + 2\int_{t_0}^{t} v_m(t)dt,\ x_m(t_1) = x_0 + 2.$$

Очевидно, что $\forall m > 0\ x_m(\cdot)$ ограничены (например, снизу x_0, а сверху x_0+2) и непрерывны. Далее оценим значение критерия качества.

$$J_1^{(1)}[u] = \int_{t_0}^{t_1} \{\overline{u}[t]'\overline{D}^{(1)}(t)\overline{u}[t] + 2\overline{d}^{(1)}(t)'\overline{u}[t] + 2x(t)'\overline{K}^{(1)}(t)\overline{u}[t]\}dt.$$

Очевидно, что существует такое постоянная $Z_1 \in \mathbb{R}$, что

$$J_1^{(1)}[u_m] + Z_1 < J^{(1)}[u_m]$$

для любых $m > 0$ (мы отбросили заведомо ограниченные слагаемые, зависящие только от $x(\cdot)$). Подставим $u_m[t] = Q_m(t)x(t) + m_n(t)$.

$$J_1^{(1)}[u_m] = \int_{t_0}^{\vartheta} \{(Q_m(t)x(t) + q_m(t))'\overline{D}^{(1)}(t)(Q_m(t)x(t) + q_m(t)) + \overline{d}^{(1)}(t)'(Q_m(t)x(t) + q_m(t)) + x(t)'\overline{M}^{(1)}(t)(Q_m(t)x(t) + q_m(t))\}dt.$$

$$J_1^{(1)}[u_m] = \int_{t_0}^{\vartheta} \{x^T(t)Q_n(t)\overline{D}^{(1)}(t)Q_n(t)x(t) + q_n^T(t)^T\overline{D}^{(1)}(t)Q_n(t)x(t) + x^T(t)Q_n(t)\overline{D}^{(1)}(t)q_n(t) + q_n^T(t)\overline{D}^{(1)}(t)q_n(t) + (\overline{d}^{(1)}(t))^TQ_n(t)x(t) + \overline{d}^{(1)}(t)'q_n(t) + x^T(t)\overline{M}^{(1)}(t)Q_n(t)x(t) + x^T(t)\overline{M}^{(1)}(t)q_n(t)\}dt.$$

Подставим $Q_n(t) = -\frac{1}{2}(\overline{A}(t), \overline{A}(t)) = -\widehat{A}(t),\ q_n(t) = -\frac{1}{2}(a(t), a(t)) + (e, e)v_n(t) = -\widehat{a}(t) + ev_n(t)$.

$$J_1^{(1)}[u_m] = \int_{t_0}^{t_1} \{x^T(t)\widehat{A}(t)\overline{D}^{(1)}(t)\widehat{A}(t)x(t) + \widehat{a}^T(t)D^{(1)}(t)\widehat{A}(t)x(t) + x^T(t)\widehat{A}(t)\overline{D}^{(1)}(t)\widehat{a}(t) + \widehat{a}^T(t)\overline{D}(t)\widehat{a}(t) - \overline{d}^T(t)\widehat{A}(t)x(t) - \overline{d}^T(t)\widehat{a}(t) - x^T(t)\overline{K}^{(1)}(t)\widehat{A}(t)x(t) - x^T(t)\overline{K}^{(1)}(t)\widehat{a}(t) - v_m(t)e^T\overline{D}^{(1)}(t)\widehat{A}(t)x(t) - v_m(t)x^T(t)\widehat{A}(t)\overline{D}^{(1)}(t)e + v_m(t)^2e^T\overline{D}^{(1)}(t)e - v_m(t)e^T\overline{D}^{(1)}(t)\widehat{a}(t) - v_m(t)\widehat{a}^T(t)\overline{D}^{(1)}(t)e + v_m(t)(\overline{d}^{(1)}(t))^Te + v_m(t)x^T(t)\overline{K}^{(1)}(t)e\}dt.$$

Опять же, отбросим заведомо ограниченные слагаемые:

$$J_2^{(1)}[u_m] = \int_{t_0}^{\vartheta} \{-v_m(t)e^T\overline{D}^{(1)}(t)\widehat{A}(t)x(t) - v_m(t)x^T(t)\widehat{A}(t)\overline{D}^{(1)}(t)e + v_m(t)^2e^T\overline{D}^{(1)}(t)e - v_m(t)e^T\overline{D}^{(1)}(t)\widehat{a}(t) - v_m(t)a^T(t)\overline{D}^{(1)}(t)e + v_m(t)(\overline{d}^{(1)}(t))^Te + v_m(t)x^T(t)\overline{K}^T(t)e\}dt.$$

Очевидно, что существует такое постоянная $Z_2 \in \mathbb{R}$, что

$$J_2[u_m] + Z_2 < J_1[u_m]$$

для любых $m > 0$. Далее, выделим все слагаемые, в которые $v_m(t)$ входит линейно. Воспользуемся теоремой о среднем значении ($\tau \in [t_0, \vartheta]$), после чего возьмём интеграл. Например:

$$\int_{t_0}^{\vartheta} -v_m(t)e^T\overline{D}^{(1)}(t)\widehat{A}(t)x(t)dt = -(e^T\overline{D}^{(1)}(\tau)\widehat{A}(\tau)x(\tau))\int_{t_0}^{\vartheta} v_m(t)dt = -(e^T\overline{D}^{(1)}(\tau)\widehat{A}(\tau)x(\tau)),$$

как видно, все такие слагаемые ограничены при $m \to +\infty$. Итак, получаем следующую оценку:

$$J_3^{(1)}[u_m] = \int_{t_0}^{\vartheta} \{v_m(t)^2e^T\overline{D}^{(1)}(t)e\}dt = \int_{t_1}^{t_2} \{v_m(t)^2e^T\overline{D}^{(1)}(t)e\}dt$$

$$J_3^{(1)}[u_m] + Z_3 < J_2^{(1)}[u_m].$$

Обозначим за $\mathcal{D}^{(1)}$ точную нижнюю грань по $t \in [t_1, t_2]$ суммы всех элементов матрицы $\overline{D}^{(1)}(t)$. По условию леммы она больше 0.

$$J_3^{(1)}[u_m] \geqslant \mathcal{D}^{(1)}\int_{t_0}^{t_1} v_m(t)^2dt = \mathcal{D}^{(1)}\frac{4m}{3}.$$

Сопоставляя полученные оценки получаем, что

$$J^{(1)}[u_m] > \mathcal{D}^{(1)}\frac{4m}{3} + Z,$$

где постоянная Z не зависит от m. Аналогично оцениваются и другие критерии. Что и требовалось доказать.

Остальные леммы доказываются аналогично.

Т е о р е м а 1. Пусть в игре (1) зафиксирована исходная коалиционная структура

$\mathfrak{k} = \{K_1, K_2\} = \{\{1,2\},\{3\}\}$ и нашлась такая точка $t \in (t_0, \vartheta)$, что в ней выполняются следующие ограничения на суммы элементов матриц, фигурирующих в функциях выигрыша игроков (здесь $\mathcal{D}_{jl}^{(i)}$ означает сумму элементов матрицы $D_{jl}^{(i)}(t)$, взятую в момент времени t):

$$\begin{array}{ll} \mathcal{D}_{11}^{(1)} > 0 & \mathcal{D}_{12}^{(1)} > 0 \\ \mathcal{D}_{21}^{(1)} > 0 & \mathcal{D}_{22}^{(1)} > 0 \end{array} \qquad \mathcal{D}_{33}^{(1)} < 0 \tag{11}$$

$$\begin{array}{ll} \mathcal{D}_{11}^{(2)} > 0 & \mathcal{D}_{12}^{(2)} > 0 \\ \mathcal{D}_{21}^{(2)} > 0 & \mathcal{D}_{22}^{(2)} > 0 \end{array} \qquad \mathcal{D}_{33}^{(2)} < 0 \tag{12}$$

$$\begin{array}{ll} \mathcal{D}_{11}^{(3)} < 0 & \mathcal{D}_{12}^{(3)} < 0 \\ \mathcal{D}_{21}^{(3)} < 0 & \mathcal{D}_{22}^{(3)} < 0 \end{array} \qquad \mathcal{D}_{33}^{(3)} > 0 \tag{13}$$

то любая Парето-максимальная ситуация будет $\mathfrak{k}$-равновесием угроз и контругроз в игре (1).
Доказательство. Достаточно сослаться на леммы 2 — при данных ограничениях с очевидностью выполняются их условия, следовательно у игроков при любой ситуации в игре будут все необходимые для определения 5 контругрозы.

Замечание. Если выполнены условия теоремы 1, которые очень легко проверяются, то для отыскания $\mathfrak{k}$-равновесия угроз и контругроз достаточно найти Парето-максимальную ситуацию в игре 1. Применение леммы 1 и подходящего варианта метода динамического программирования из [?] позволяет найти явный вид решения игры 1 — $\mathfrak{k}$-равновесия угроз и контругроз.

СПИСОК ЛИТЕРАТУРЫ

1. *Жуковский В. И.* Введение в дифференциальные игры при неопределенности. М.: МНИИПУ, 1997.
2. *Вайсборд Э. М., Жуковский В. И.* Введение в дифференциальные игры нескольких лиц и их приложения. М.: Сов. радио, 1980.
3. *Воробьёв Н. Н.* Основы теории игр. Бескоалиционные игры. М.:Наука, 1984.
4. *Подиновский В. В., Ногин В.Д.* Парето-оптимальные решения многокритериальных задач. М.:Наука, 1982.
5. *Васильев Ф. П.* Методы оптимизации. М.: Факториал Пресс, 2002.
6. *Жуковский В. И., Чикрий А.А.* Линейно-квадратичные дифференциальные игры. Киев: Наукова Думка, 1994.
7. *Жуковский В. И.* Кооперативные игры при неопределённости и их приложения. М.:Эдиториал УРСС, 1999.
8. *Понтрягин Л. С.* Обыкновенные дифференциальные уравнения М.: ГИФМЛ, 1961.

УДК 512.624.95

ОЦЕНКА СЛОЖНОСТИ ПРИМЕНЕНИЯ СИМВОЛЬНЫХ МЕТОДОВ В КРИПТОАНАЛИЗЕ АЛГОРИТМА ГОСТ 28147-89

© 2007 г. А. С. Мелузов

asmelouzov@mail.ru

Научный руководитель: СНС ИПИБ МГУ, к.ф.-м.н В. Н. Цыпышев

Кафедра Математической кибернетики

Введение. В статье описан способ применения символьных методов криптографического анализа к алгоритму ГОСТ 28147-89. Для этого приведены способы построения системы полиномиальных уравнений, описывающих работу алгоритма ГОСТ 28147-89, проведена оценка сложности и структуры получаемой системы полиномиальных уравнений, а также проведена оценка эффективности алгоритмов решения систем полиномиальных уравнений над конечными полями с использованием стандартных базисов (базисов Гребнера).

1. Постановка задачи. В настоящее время в целях криптоанализа широко используется следующий подход, целиком и полностью основанный на методах символьных вычислений в кольцах многочленов над различными алгебраическими структурами. А именно: с использованием различных методик функционирование криптосхемы описывается с помощью системы полиномиальных уравнений над какой-либо алгебраической структурой (как правило, над конечным полем) с тем, чтобы иметь возможность свести задачу криптоанализа к решению построенной системы полиномиальных уравнений в символьном виде. Как правило, этот шаг выполняется путем построения базиса Грёбнера соответствующего полиномиального идеала с использованием одного из широко известных алгоритмов, таких, как алгоритм Бухбергера, XL-метод, алгоритмы F_4 и F_5, принадлежащие Ж.-К. Фажере.

Необходимо применить данный подход к алгоритму блочного шифрования ГОСТ 28147-89 в режиме простой замены.

2. Алгоритм ГОСТ 28147-89. В режиме простой замены алгоритм ГОСТ 28147-89 работает следующим образом: открытые данные, подлежащие зашифрованию, разбивают на блоки по 64 бит в каждом. Блок открытого текста представляется в виде конкатенации двух блоков по 32 бита каждый:

$$T_0 = (a_1(0), \ldots, a_{32}(0), b_1(0), \ldots, b_{32}(0)) = a(0)||b(0), \tag{1}$$

где в последующем a_1 считается младшим,а a_{32} — старшим битом двоичной записи некоторого целого числа.

Объем ключа составляет 256 бит $(W_{256}, \ldots, W_1)$, которые разбиваются на восемь 32-х разрядных вектора

$$\begin{array}{c} X_0 = (W_{32}, \ldots, W_1), \\ X_1 = (W_{64}, \ldots, W_{33}) \\ \ldots\ldots \\ X_7 = (W_{256}, \ldots, W_{225}). \end{array} \tag{2}$$

Уравнения зашифрования в режиме простой замены имеют вид:

$$\left\{ \begin{array}{l} a(j) = \big(a(j-1) + X_{(j-1) \pmod 8}\big)\, KR \oplus b(j-1), \\ b(j) = a(j-1), j = \overline{1,24}, \\ a(j) = \big(a(j-1) + X_{(32-j)}\big)\, KR \oplus b(j-1), \\ b(j) = a(j-1), j = \overline{25,31}, \\ a(32) = a(31), \\ b(32) = (a(31) + X_0)\, KR \oplus b(31). \end{array} \right. \tag{3}$$

Здесь $+$ — операция сложения по модулю 2^{32} двух чисел, двоичным представлением которых являются операнды, причем результатом операции является двоичный вектор длины 32, являющийся двоичным представлением получившегося 32-х разрядного числа.

Преобразование K выглядит следующим образом: поступающтй на его вход 32-х разрядный вектор разбивается на 8 подвекторов длины 4, каждый из которых, в свою очередь, поступает на вход соответствующего *узла замены* $K_1, \ldots, K_8$, осуществляющего перестановку на множестве двоичных векторов длины 4. Восемь результатов перестановок путем конкатенации составляют результирующий вектор:

$$XK = (X_8 || \ldots || X_1)\, K = (K_8(X_8) || \ldots || K_1(X_1))\,. \tag{4}$$

R — операция циклического сдвига на одиннадцать шагов в сторону старших разрядов:

$$(y_{32}, \ldots, y_1)R = (y_{21}, y_{20}, \ldots, y_2, y_1, y_{32}, y_{31}, \ldots, y_{23}, y_{22}). \tag{5}$$

Операция $\oplus$ есть покоординатное суммирование 32-разрядных векторов по модулю 2.

3. Построение системы полиномиальных уравнений. Обратимся теперь к вопросу о построении системы полиномиальных уравнений, аппроксимирующей зашифрование в режиме простой замены согласно алгоритму ГОСТ 28147-89.

Очевидно, что для того, чтобы построить полиномиальную аппроксимацию полного алгоритма зашифрования, достаточно построить полиномиальную аппроксимацию для одного раунда. Полиномиальная аппроксимация для 32 раундов получается путем введения дополнительных переменных и дупликации системы.

3.1. Модульное сложение.Первый этап при построении системы полиномиальных уравнений, аппроксимирующей один раунд алгоритма ГОСТ 28147-89, состоит в том, чтобы выразить координаты двоичного представления суммы двух чисел по модулю 2^{32} как булевы функции от координат двоичных представлений складываемых чисел.

Это можно сделать, итеративно вычисляя функции переноса в старший разряд и, на их основе, координатные функции, согласно формулам [4,(1)]: для $\bar{r} = (r_0, \ldots, r_{n-1}), \bar{x} = (x_0, \ldots, x_{n-1}), \bar{a} = (a_0, \ldots, a_{n-1})$ при

$$\begin{array}{l} r_0 + r_1 \cdot 2 + \cdots + r_{n-1} \cdot 2^{n-1} = \\ (x_0 + \cdots + x_{n-1} \cdot 2^{n-1}) + (a_0 + \cdots + a_{n-1} \cdot 2^{n-1}) \pmod{2^n} \end{array} \tag{6}$$

имеют место равенства:

$$\begin{array}{ll} r_0 = x_0 \oplus a_0 \oplus c_0, & c_0 = 0, \\ r_i = x_i \oplus a_i \oplus c_i, & c_i = x_{i-1}a_{i-1} \oplus x_{i-1}c_{i-1} \oplus a_{i-1}, \\ i = \overline{1, n-1}. & \end{array} \tag{7}$$

Как видно из соотношений (6) и (7), i-я координатная функция суммы $\bar{x}+\bar{a}$ имеет степень $i+1$ и содержит $2^i + 1$ термов. Поэтому нахождение координатных функций суммы $\bar{x} + \bar{a}$ при значениях i, близких к 32, является трудоемкой вычислительной задачей, невыполнимой на обычном персональном компьютере.

3.2. Блоки замены. Второй этап построения системы полиномиальных уравнений, аппроксимирующей один раунд криптоалгоритма ГОСТ 28147-89, состоит в том, чтобы построить покоординатную аппроксимацию булевыми функциями узлов замены с последующим вычислением композиции полученной аппроксимации и найденных на первом этапе координатных функций модульного сложения. Можно сделать двумя способами.

Во-первых, естественный способ состоит в представлении каждого узла замены $K_i, i = \overline{1,8}$ как вектора $(f_{i,1}, \ldots, f_{i,4})$ из четырех булевых функций от четырех переменных.

Во-вторых, можно обратиться к методике, предложенной в статье [2, Раздел 3.2].

Каждая методика имеет свои плюсы и свои минусы. Первый способ предпочтителен по той причине, что в этом случае не приходится вводить новые служебные переменные, и, таким

образом, число переменных, входящих в уравнения системы, значительно ниже. Однако сами уравнения, как правило, имеют более высокую степень и состоят из большего числа термов. Второй способ имеет своим преимуществом то, что этот путь позволяет получить значительно большее число уравнений, причем фиксированной степени (а именно по 21 уравнению степени не выше 2 от входных переменных для каждого узла замены). Но, с другой стороны, при этом наблюдается рост числа служебных переменных, что усложняет решение системы.

Однако, при использовании второго способа, структура итоговой системы полиномиальных уравнений представляется более ясной, поэтому будем считать, что применяется именно этот способ.

3.3. Сдвиг и побитовое сложение. Последний этап построения системы полиномиальных уравнений, аппроксимирующей один раунд криптоалгоритма ГОСТ 28147-89, состоит в подстановке в полученные уравнения переменных, соответствующих результату работы данного раунда, просуммированных с переменными, описывающими результат работы раунда за два до текущего в соответствии с (3) (или с битами левой части открытого текста, если описываемый раунд — первый и правой части открытого текста, если второй) с учетом операции побитового сдвига, в соответствии с (3). Для этого выходы узлов замены были выражены через левую (правую) часть открытого текста (или результат работы раунда за два до текущего), а выходы — через результат работы описываемого раунда работы алгоритма (или зашифрованный текст, если описываемый раунд - последний или предпоследний). Если y_i — i-й бит выхода узлов замены, z_i — i-й бит результата работы текущего раунда текста, а l_i — i-й бит результата работы предыдущего раунда, то верно следующее соотношение:

$$y_i = z_{(i+11) \bmod 32} + l_{(i+11) \bmod 32}.$$

3.3. Построение итоговой системы. Таким образом, мы получим системы полиномиальных уравнений, описывающие каждый раунд алгоритма шифрования ГОСТ 28147-89. С помощью отождествления переменных, соответствующим одним и тем же значениям (например, результаты работы первого раунда будут складываться со значениями, полученными после побитового сдвига в третьем раунде, а также подаваться на вход регистра модульного сложения с ключом во втором раунде), получим систему полиномиальных уравнений, описывающую алгоритм шифрования ГОСТ 28147-89 целиком.

Итак, при анализе криптографического алгоритма ГОСТ 28147-89 было показано, что система полиномиальных уравнений, аппроксимирующая его работу в режиме простой замены зависит от 1248 переменных (из которых 256 — биты ключа, а 992 — вспомогательные переменные (промежуточные результаты работы на каждом раунде)), и состоит из 5376 уравнений, по 672 уравнения 7,15,23,31,39,47,55 и 63 степени.

4. Оценка сложности решения построенной системы.

После построения системы полиномиальных уравнений, аппроксимирующей работу алгоритма ГОСТ 28147-89, необходимо её решить. Решение СПУ будем осуществлять методом построения базиса Грёбнера.

Существует довольно много алгоритмов, строящих базис Грёбнера заданной системы уравнений, однако, самым быстрым на данный момент является алгоритм F_5, принадлежащий Ж.-К. Фажере.

В статье [1] подробно рассматриваются особенности применения этого алгоритма к полиномам над полем $GF(2)$.

Суть алгоритма состоит в построении последовательности матриц $\mathcal{M}_{d,m}$, столбцы которых соответствуют всевозможным мономам степени d, выстроенным в лексикографичеком порядке, а строки соответствуют всевозможным произведениям $t \times f_j$, где t — моном, такой что $deg(t \times f_j) = d$, а $1 \le j \le m$, $f_1, \ldots, f_m$ – полиномы, входящие в исходную систему.

Далее, над каждой такой матрицей производятся преобразования, приводящие её к треугольному виду. Поскольку матрица сильно разряжена, можно считать, что сложность таких преобразований будет равна $\mathcal{O}(k^2)$, где k – ранг матрицы $\mathcal{M}_{d,m}$.

Общая сложность алгоритма определяется сложностью вычислений линейной алгебры для наибольшей матрицы. В соответствии с [1] такой матрицей будет та, которая соответствует

полиномам степени D_{reg}. А D_{reg} – степень полурегулярности исходной системы. Термин "полурегулярная последовательность полиномов"вводится как обобщение понятия "регулярная последовательность полиномов"для полиномов над полем $GF(2)$.

Степень полурегулярности вычисляется как номер первого неположительного члена порождающего ряда последовательности полиномов (исходной системы). Сумма порождающего ряда равна $S_{m,n}(z) = (1+z)^n / \prod_{k=1}^{m}(1+z^{d_k})$, где n – число неизвестных в исходной системе, d_k – степень k-того полинома, $f_1, \ldots, f_m$ – полиномы, входящие в исходную систему.

В соответствии с описанной в [1] методикой, был построен порождающий ряд последовательности полиномов, описывающей работу алгоритма ГОСТ 28147-89 в режиме простой замены:

$$\frac{(1+z)^{1248}}{((1+z^7)(1+z^{15})(1+z^{23})(1+z^{31})(1+z^{39})(1+z^{47})(1+z^{55})(1+z^{63}))^{672}}$$

и найден его первый неположительный член, номер которого соответствует степени регулярности системы полиномиальных уравнений (последовательности полиномов). Степень полурегулярности для системы полиномиальных уравнений, описывающей работу криптоалгоритма ГОСТ 28147-89 в режиме простой замены $D_{reg} = 402$.

Следовательно, максимальный размер матрицы в алгоритме F_5 будет равен рангу матрицы $\mathcal{M}_{d,m}$ на шаге D_{reg} и равен числу столбцов этой матрицы, то есть

$$\binom{n}{D_{reg}} = \binom{1248}{402} \approx 2^{1126},$$

а сложность вычисления базиса Гребнера построенной системы будет равна 2^{2252}, при допущении, что коэффициент сложности вычислений линейной алгебры $\omega = 2$, по причине сильной разреженности матриц.

СПИСОК ЛИТЕРАТУРЫ

1. *Magali Bardet, Jean-Charles Faugère, Bruno Salvy.* Complexity of Gröebner basis computation for Semi-regular Overdetemined sequences over $\mathbb{F}_2$ with solutions in $\mathbb{F}_2$.
2. *N.T.Courtois, J.Pieprzyk* Crytoanalysis of block ciphers with overdefined systems of equations. // ASIACRYPT 2002, LNCS 2501, pp.267–287, 2002
3. *Jean-Charles Faugère* A new efficient algorithm for computation Gröebner bases without reduction to zero (F_5).
4. *A.Braeken, I.Semaev* The ANF of the Composition of Addition and Multiplication mod 2^n with a Boolean Function // Fast Software Encryption 2005, Proceedings, pp.115–127

СБОРНИК СТАТЕЙ МОЛОДЫХ УЧЕНЫХ факультета ВМиК МГУ, 2007, выпуск № 4, с. 111–116

УДК 004.492.3

ПРИМЕНЕНИЕ АЛГОРИТМА K-MEANS ДЛЯ РЕАЛИЗАЦИИ АНАЛИЗАТОРА СТАТИСТИЧЕСКОЙ СИСТЕМЫ ОБНАРУЖЕНИЯ АТАК

© 2007 г. Е. А. Наградов, А. И. Качалин, В. А. Костенко

nea@lvk.cs.msu.su

Кафедра Автоматизации Систем Вычислительных Комплексов.
Лаборатория Вычислительных Комплексов.

Введение.

Одним из распространенных подходов к реализации статистических систем обнаружения атак является подход обнаружения аномалий активности системы [1]. Для методов обнаружения атак, основанных на подходе обнаружения аномалий, характерна задача обработки обучающей выборки большого объема, в результате чего наиболее применимы алгоритмы обработки с линейной временной сложностью относительно размера обучающей выборки. В данной работе рассматривается применение методов кластерного анализа для решения задачи обнаружения атак, и, в частности, алгоритма k-means, обладающего линейной временной сложностью относительно размера обучающей выборки.

1. Задача обнаружения атак.

Будем рассматривать наблюдаемую систему как совокупность объектов системы и взаимодействий между объектами системы. Тогда под состоянием системы в определенный момент времени будем понимать вектор характеристик объектов системы и взаимодействий между объектами системы. Под нормальным состоянием системы будем понимать состояние системы в процессе штатного функционирования системы. Предполагается, что множество нормальных состояний системы может быть статистически отличимо от множества аномальных состояний, наблюдаемых в процессе осуществления атак на систему.

Определим атаку на наблюдаемую систему как воздействие, переводящее наблюдаемую систему из нормального состояния в аномальное. Тогда задача обнаружения атаки сводится к задаче определения принадлежности наблюдаемого состояния системы к множеству нормальных состояний системы.

Решение задачи обнаружения атак может быть разделено на два этапа [4]: этап обучения и этап обнаружения. Этап обучения заключается в построении модели нормального поведения системы на основании обучающей выборки, содержащей примеры нормальных состояний системы. Этап обнаружения заключается в проверке принадлежности наблюдаемых состояний системы к построенной модели нормального поведения.

В связи с тем, что раннее обнаружение атаки в реальной системе может помочь минимизировать потери от осуществленной атаки, дополнительно накладывается требование обнаружения атак в режиме реального времени.

Далее в работе предлагается метод обнаружения атак, основанный на использовании методов кластеризации, позволяющий выполнять обнаружение атак в режиме реального времени, а также приводятся результаты тестирования предложенного метода.

2. Этап обучения.

В данной работе применение методов кластерного анализа основано на использовании гипотезы компактности [5]: классам векторов (кластерам) соответствуют некоторые компактные множества в пространстве характеристик.

Идея предлагаемого метода опирается на предположение о том, что принадлежность анализируемого вектора к множеству нормальных состояний системы можно определить на основании характеристик модели нормального поведения системы.

В качестве модели нормального поведения будем использовать описание множества нормальных состояний системы посредством совокупности кластеров, выделенных при помощи одного или нескольких алгоритмов кластерного анализа. Тогда характеристики модели нормального поведения представляют собой совокупность характеристик выделенных кластеров. Таким образом, задача построения модели нормального поведения системы сводится к задаче кластеризации векторов обучающей выборки. Выбор конкретного представления кластера зависит от используемых алгоритмов кластеризации. Для того чтобы обеспечить достаточную точность описания множества нормального поведения, требуется обеспечить высокую степень покрытия множества нормального поведения примерами обучающей выборки. Следствием этого является необходимость анализа обучающей выборки большого объема, в связи с чем для кластеризации обучающей выборки в данной работе выбран алгоритм k-means. Алгоритм k-means выбран на основании того, что он обладает линейной временной сложностью относительно размера обучающей выборки и размерности входных векторов. В связи с тем, что алгоритм k-means использует представление кластеров в виде центроидов [3,6], характеристикой кластера в данной работе является центроид. Центроид кластера представляет собой среднее арифметическое векторов, принадлежащих кластеру.

Параметрами алгоритма k-means являются количество кластеров и начальное расположение кластеров, причем точность работы алгоритма сильно зависит от выбора значений параметров. Исходя из особенностей алгоритма k-means, была выбрана следующая последовательность шагов этапа обучения: определение количества кластеров и начального расположения кластеров, объединение ближайших кластеров и применение алгоритма k-means для минимизации ошибки кластеризации.

2.1. Определение количества кластеров и начального расположения кластеров. Для определения количества кластеров и начального расположения кластеров был разработан алгоритм, являющийся модификацией алгоритма BIRCH, предложенного в работе [9]. Идея алгоритма заключается в построении дерева кластеров, которое используется для быстрого поиска кластера, ближайшего к анализируемому вектору. Дерево кластеров представляет собой дерево, узлами которого являются кластеры, причем каждый узловой кластер представляет собой объединение кластеров-подузлов. Алгоритм является инкрементальным, т.е. осуществляет изменение дерева кластеров путем поочередного добавления векторов обучающей выборки в дерево кластеров.

Добавление вектора обучающей выборки к дереву кластеров осуществляется следующим образом:

1) определяется кластер-подузел, наиболее близко (по заданной мере близости) расположенный к добавляемому вектору,

2) выполняется добавление вектора к поддереву с корнем - выбранным подузлом,

3) в том случае, когда достигнут листовой кластер, производится добавление вектора к кластеру в том случае, когда соблюдаются ограничения на размер кластера (расстояние до центроида не превосходит заданное максимальное расстояние), иначе создается новый листовой кластер,

4) производится обновление центроидов всех кластеров-подузлов, которые были пройдены в процессе добавления,

5) в том случае, когда узловой кластер содержит чрезмерное количество подузлов, производится разбиение узлового кластера на несколько кластеров-подузлов.

По результатам тестирования различных мер, в качестве меры близости вектора и кластера в данной работе была выбрана сумма расстояний до центроида кластера до и после добавления вектора в кластер.

Разработанный алгоритм является итеративным, каждый шаг алгоритма состоит из трех этапов: добавление всех векторов обучающей выборки к дереву кластеров, проверку попадания векторов в листовые кластеры, и оптимизацию дерева.

Проверка попадания вектора в узловой кластер заключается в поиске ближайшего к рассматриваемому вектору листового кластера путем просмотра дерева (аналогично процессу добавления вектора) и определении расстояния до центроида листового кластера. В том случае,

когда расстояние меньше заданного максимального расстояния, фиксируется попадание вектора в кластер, иначе фиксируется промах. Шаг проверки позволяет выявить пустые кластеры - кластеры, в которые при проверке не попал ни один вектор обучающей выборки. Пустые кластеры возникают вследствие того, что процесс кластеризации является инкрементальным, и в зависимости от порядка обработки векторов в разных подузлах дерева могут образовываться близко расположенные листовые кластеры. Тогда на этапе тестирования может происходить перераспределение векторов по таким листовым кластерам.

Оптимизация дерева заключается в удалении пустых кластеров, кластеров с малым количеством векторов и кластеров с большим количеством промахов.

Результатом работы алгоритма является совокупность листовых кластеров после нескольких итераций алгоритма.

2.2. Объединение ближайших кластеров.

При тестировании эффективности работы описанного выше алгоритма была выявлена проблема, связанная с выделением большого количества близко расположенных кластеров. Кроме того, для обеспечения возможности обнаружения аномалий активности системы в режиме реального времени требуется ограничить максимально возможное количество кластеров в модели нормального поведения.

Для решения перечисленных проблем в данной работе применяется алгоритм, основанный на использовании иерархического метода кластеризации [2,3]. Алгоритм заключается в последовательном объединении наиболее близких (по заданной мере близости) кластеров. В данной работе в качестве меры близости кластеров было выбрано расстояние между центроидами кластеров. Алгоритм останавливается при достижении требуемого количества кластеров.

Результатом работы алгоритма является совокупность кластеров, количество которых не превышает максимально возможное количество кластеров, при котором обнаружение может выполняться в режиме реального времени.

2.3 Применение алгоритма k-means.

На основании результатов работы алгоритма объединения кластеров задаются начальные параметры алгоритма k-means. Алгоритм k-means является итеративным алгоритмом, минимизирующим ошибку кластеризации обучающей выборки при заданном количестве кластеров.

Каждый шаг алгоритма заключается в определении ближайшего кластера (на основании расстояния до центроида кластера) для каждого входного вектора, и вычислении нового расположения центроидов кластеров как среднего арифметического векторов, для которых кластер является ближайшим.

Результатом работы алгоритма является множество центроидов кластеров, составляющих модель нормального поведения системы.

3. Этап обнаружения.

Задача этапа обнаружения заключается в определении принадлежности вектора наблюдаемого состояния системы к множеству нормальных состояний системы. Согласно предположению о том, что принадлежность вектора к множеству нормальных состояний может быть определена на основании характеристик выделенных кластеров, для решения задачи был разработан алгоритм, основанный на алгоритме определения степени обособленности [7].

Идея разработанного алгоритма заключается в вычислении степени обособленности как суммы расстояний до заранее заданного количества ближайших векторов нормальных состояний системы. Но, в отличие от общего алгоритма, приведенного в [7], в качестве векторов нормального поведения в данной работе выбираются центроиды ближайших кластеров столько раз, сколько векторов содержится в кластере.

В качестве параметра алгоритма задается пороговое значение степени обособленности и требуемое количество ближайших векторов. На основании вычисленного значения степени обособленности определяется принадлежность к множеству нормальных состояний следующим образом: если значение степени обособленности ниже порогового значения, то входной вектор считается принадлежащим множеству нормальных состояний. В противном случае, наблюдаемое состояние системы считается аномальным.

4. Тестирование предложенного метода.

Обучение и тестирование анализатора, реализованного на основе предлагаемого метода, производилось на тестовом наборе данных KDD Cup 99 [8]. Целями тестирования являлись оценка эффективности реализованного анализатора и определение изменения показателей эффективности при добавлении в обучающую выборку векторов, соответствующих атакам.

Тестирование анализатора было разделено на два этапа. Этапы различались по содержанию обучающей выборки, на которой производилось обучение анализатора:

1) обучающая выборка не содержит векторов состояний системы, соответствующих атакам,

2) обучающая выборка содержит малое количество векторов, соответствующих атакам.

4.1. Тестирование анализатора на обучающей выборке, не содержащей векторов, соответствующих атакам.

Обучение производилось на обучающей выборке, содержащей 40 тысяч векторов, соответствующих нормальному поведению системы. Тестирование производилось при различных пороговых значениях степени обособленности в алгоритме этапа обнаружения на тестовой выборке, содержащей 8 тысяч векторов, включающих в себя 4 тысячи векторов, соответствующих нормальному поведению, и 4 тысячи векторов, соответствующих атакам различных классов.

Рис. 1. Показатели эффективности анализатора в зависимости от порогового значения степени обособленности в алгоритме этапа обнаружения.

Рис. 2. Соотношение между % обнаруженных атак и % ошибок второго рода.

Результаты тестирования приведены на рисунках 1 и 2. Точки графика на рисунке 2 соответствуют парам (процент обнаруженных атак, процент ошибок 2 рода) для различных пороговых значений степени обособленности в алгоритме этапа обнаружения.

Под точностью далее понимается процент правильно распознанных векторов тестовой выборки. Максимальная точность (96.35%) достигается при пороговом значении степени обособленности в алгоритме этапа обнаружения равном 6.0. При таком значении параметра ошибка первого рода (необнаруженные атаки) составила 3.48%, ошибка второго рода (ложные срабатывания) составила 3.66%.

4.2 Тестирование анализатора на обучающей выборке, содержащей вектора, соответствующие атакам.

При обучении системы обнаружения атак на реальной системе возможны ситуации, когда вектора, соответствующие атакам, попадают в обучающую выборку как примеры нормальных состояний системы. Задача данного этапа тестирования - определить изменение показателей эффективности предлагаемого метода при наличии в обучающей выборке векторов, соответствующих атакам.

Обучение производилось на обучающей выборке, содержащей 40 тысяч векторов, соответствующих нормальному поведению системы, и 1 тысячу векторов, соответствующих атакам различных классов. Тестирование анализатора проводилось при различных пороговых значениях степени обособленности в алгоритме этапа обнаружения на тестовой выборке, совпадающей с тестовой выборкой для первого этапа.

Рис. 3. Показатели эффективности анализатора в зависимости от порогового значения степени обособленности в алгоритме этапа обнаружения.

Рис. 4. Соотношение между % обнаруженных атак и % ошибок второго рода.

Результаты тестирования приведены на рисунках 3 и 4. Точки графика на рисунке 4 соответствуют парам (процент обнаруженных атак, процент ошибок 2 рода) для различных пороговых значений степени обособленности в алгоритме этапа обнаружения.

Максимальная точность (95.46%) достигается при пороговом значении степени обособленности в алгоритме этапа обнаружения равном 5.5. При таком значении параметра ошибка первого рода (необнаруженные атаки) составила 3.67%, ошибка второго рода (ложные срабатывания) составила 5.36%.

Заключение.

Тестирование предложенного метода обнаружения атак показало высокие показатели эффективности на обоих этапах. Однако предложенный метод является слабо устойчивым к наличию в обучающей выборке векторов, соответствующих атакам. Это характеризуется высокими значениями ошибки первого рода при малых значениях ошибки второго рода, что продемонстрировано на рисунке 4. Таким образом, для эффективного применения предложенного метода требуется обеспечить отсутствие в обучающей выборке векторов, соответствующих атакам.

Эффективность работы анализатора определяется значением параметров алгоритма этапа обнаружения. Оптимальное пороговое значение степени обособленности в алгоритме этапа обнаружения в проведенном тестировании выбиралось с целью максимизации количества правильно распознанных векторов тестовой выборки. Однако при создании промышленной системы обнаружения атак, значение параметра может быть определено с целью максимизации точности обнаружения атак определенного класса, либо с целью достижения определенных показателей ошибки первого и второго рода. Такое значение параметра может быть выбрано на основании тестирования системы обнаружения атак на способность обнаружения атак различных классов.

Тестирование показало, что предложенный метод является ограниченно применимым для решения задачи обнаружения атак. Достоинствами предложенного метода являются высокие показатели эффективности среди алгоритмов, использующих подход обнаружения аномалий активности системы, а также возможность адаптации метода для обеспечения обнаружения атак в режиме реального времени. Перечисленные достоинства позволяет применять метод в составе реальных систем обнаружения атак при условии использования альтернативных методов обнаружения атак (например, сигнатурных) в процессе сбора обучающей выборки.

СПИСОК ЛИТЕРАТУРЫ

1. *Sundaram A.* An Introduction to Intrusion Detection [HTML] (http://www.acm.org/crossroads/xrds2-4/intrus.html)
2. Кластерный анализ [HTML] (http://www.statsoft.ru/home/textbook/modules/stcluan.html)
3. *Jain A.K, Murty M.N, Flynn P.J.* Data Clustering A review // ACM Computing Surveys. 1999. 31. No. 3. [PDF] (http://www.cs.rutgers.edu/ mlittman/courses/lightai03/jain99data.pdf)
4. *Петровский М.И.* Применение методов интеллектуального анализа данных в задачах выявления компьютерных вторжений // Труды Второй Всероссийской научной конференции, Методы и средства обработки информации. М.: Изд-во факультета ВМиК МГУ, 2005. С. 158-168.
5. *Половикова О.Н.* Применение кластерного анализа для математического моделирования информационно-поисковых систем [HTML] (http://www.ict.nsc.ru/ws/YM2003/6285/)
6. *Luke B.T.* K-Means Clustering [HTML] (http://fconyx.ncifcrf.gov/ lukeb/kmeans.html)
7. *S. Harmeling, G. Dornhege, D. Tax, F. Meinecke, K.-R. Muller* From outliers to prototypes: ordering data // Neurocomputing, 2006 [PDF] (http://ida.first.fhg.de/ harmeli/ordering.pdf)
8. KDD Cup 1999 Data [HTML] (http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html)
9. *Tian Zhang, Raghu Ramakrishnan, Miron Livny* BIRCH: An Efficient Data Clustering Method for Very Large Databases // ACM SIGMOD International Conference on Management of Data. 1996. [PDF] (http://citeseer.ist.psu.edu/zhang96birch.html)

УДК 517.67

ПРИМЕНЕНИЕ АДАПТИВНЫХ МЕТОДОВ ДЛЯ РЕШЕНИЯ ОБРАТНЫХ ЗАДАЧ ДИАГНОСТИКИ ПЛАЗМЫ

© 2007 г. С. В. Носов

nosov.sergey@gmail.com

Кафедра Автоматизации научных исследований

Введение. Обработка и интерпретация данных, получаемых с помощью магнитной диагностики плазмы, является непростой задачей. Это связано как с огромным потоком данных, так и со сложностью восприятия информации, представленной в виде разнообразных колебательных процессов. В настоящей работе для решения проблемы интерпретации магнитных измерений предлагается новый подход, основанный на скрытых моделях Маркова. Скрытые модели Маркова хорошо зарекомендовали себя при решении задачи распознавания человеческой речи, а магнитные частотные характеристики часто похожи на речевые. Каждому разряду ставится в соответствие своя уникальная марковская модель, которая сдержит не более десяти состояний. Новый подход позволяет сократить объем информации о разряде с десятков гигабайт до нескольких килобайт. При этом появляется возможность найти вероятности перехода плазменного разряда от одного состояния к другому. Для каждого состояния определяются характерные визуальные изображения плазмы. Всё это существенно облегчает изучение динамики плазмы. Кроме того, предлагаемая техника является крайне полезной для создания системы навигации по базам данных колоссального объема, содержащих записи о магнитной диагностике плазмы.

При проведении разрядов на установках токамак все изменения магнитного поля плазмы фиксируются с помощью большого числа так называемых катушек Мирнова. По мнению специалистов, занимающихся обработкой результатов экспериментов, полученная информация используется в незначительной мере, и это при том, что объем информации для каждого разряда в оцифрованном виде занимает десятки гигабайт! В настоящей работе предлагается использовать данные магнитных измерений для индексирования разрядов, что позволит построить систему навигации по базе данных, хранящей информацию о всех плазменных разрядах. В итоге исследователь сможет избавится от многих трудоёмких рутинных операций и сосредоточиться на научной стороне дела, причём связанной не только с магнитной диагностикой.

Для того, чтобы провести индексирование, нужно существенным образом сократить объем данных без потери основной информации о характере жизни плазмы в каждом разряде, а также привести полученный сжатый набор данных к стандартизованной форме. Последнее требование особенно трудно выполнить из-за различной продолжительности плазменных разрядов. Для поставленных целей на наш взгляд наилучшим образом подходят скрытые модели Маркова (СММ), которые позволяют моделировать исходный сигнал некоторым случайным процессом, параметры которого могут быть оценены достаточно точно. Иными словами, СММ можно рассматривать как источник случайного сигнала с заданными характеристиками, соответствующими характеристикам моделируемого сигнала. Количество параметров, которые полностью описывают СММ, обычно имеют на несколько порядков меньший объем, чем исходные данные, поэтому они идеально подходят для целей индексирования.

После того, как построена скрытая модель Маркова, для удобства интерпретации эксперимента можно установить связь найденных состояний модели с другими параметрами разряда. Например, на установке MAST (Великобритания) каждый разряд также фиксируется высокоскоростной видеокамерой. Установив соответствие между характерными изображениями плазмы в каждом состоянии, можно визуализировать последовательность переходов в естественном для экспериментаторов виде.

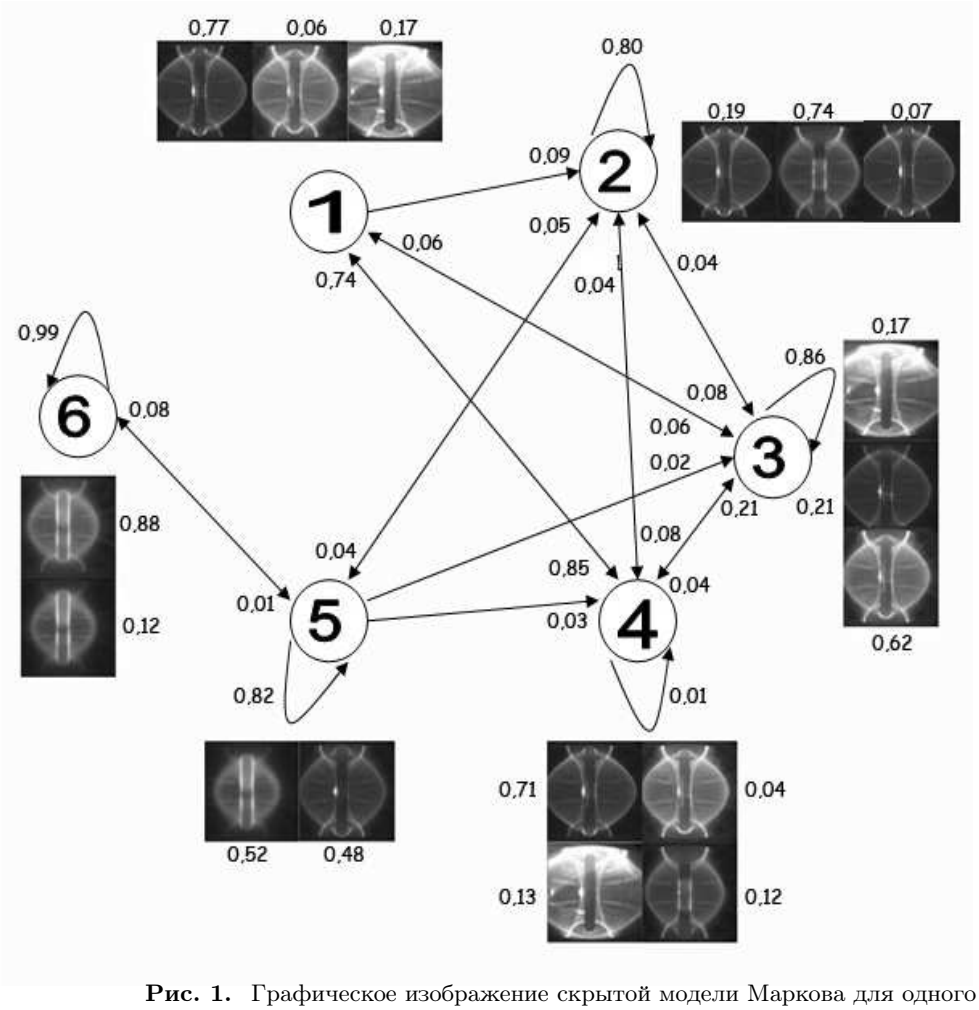


Рис. 1. Графическое изображение скрытой модели Маркова для одного разряда.

Настройка скрытых моделей Маркова. Каждая СММ характеризуется следующими параметрами:

1) Вероятностью выбора начального состояния.

2) Матрицей переходов из состояния в состояние.

3) Матрицей вероятностей наблюдения заданных признаков в каждом состоянии.

Для того, чтобы найти эти параметры, нужно сначала провести параметризацию исходного сигнала, т.е. заменить его последовательностью векторов из заданного множества, которое называют кодовой книгой. Для построения кодовой книги необходимо обработать все (или, по крайней мере, большую часть) данные, полученные в результате магнитных измерений; преобразовать их в наборы характерных признаков, и полученную совокупность признаков кластеризовать, т.е. разбить на классы. Набор векторов, состоящий из центроидов кластеров, как раз образует кодовую книгу. Для выделения характерных признаков каждый сигнал был разбит на набор перекрывающихся сегментов. Для снижения влияния граничных эффектов к данным в пределах каждого сегмента применена оконная функция Хемминга, после чего был вычислен кепстр. Набор из нескольких десятков кепстральных коэффициентов использовался в качестве вектора характерных признаков. Для классификации использовался модифицированный алгоритм К-средних.

Построение СММ проводилось следующим образом. Все начальные данные модели инициализировались с помощью датчика случайных чисел, после чего суммы элементов строк или столбцов матриц, которые должны соответствовать достоверным событиям, нормировались на единицу. Далее все параметры уточнялись поэтапно с помощью процедуры переоценки параметров Баума-Уэлша до достижения стационарных значений.

Полученные результаты. Обработка большого числа разрядов, проведенных на установке MAST, позволила установить, что скрытые модели Маркова, соответствующие этим разрядам, могут содержать не более чем 10 состояний! Это позволяет сократить объем информации, необходимой для осуществления индексирования, до нескольких килобайт. На рисунке приведено изображение СММ для одного разряда. Круги с цифрами внутри обозначают состояния, а линии со стрелками - возможные переходы; числа, стоящие рядом с линиями, соответствуют вероятности данного перехода. Рядом с каждым кружком приведены изображения плазмы, наиболее характерные для данных состояний. Числа на фотографиях соответствуют вероятности наблюдения данного состояния плазмы в соответствующем состоянии модели. Ясно, что столь наглядную и содержательную информацию о разряде практически невозможно получить вручную.

Заключение. В настоящей работе предложен новый способ обработки данных магнитных измерений плазмы, основанный на применении скрытых моделей Маркова. С помощью предложенной методики обработано большое число разрядов, проведенных на установке MAST. В результате объем информации о каждом разряде был сокращен с десятков гигабайт до нескольких килобайт. Для каждого состояния моделей были найдены характерные визуальные изображения разряда и вероятности перехода между ними, что позволяет интерпретировать процесс поведения плазмы наглядным образом. Начата работа над системой навигации по базе данных, реализующей такие функции, как, например, поиск наиболее близких или удалённых разрядов, разрядов с заданными состояниями и вероятностями перехода.

СПИСОК ЛИТЕРАТУРЫ

1. *Зайцев Ф. С.* Математическое моделирование эволюции тороидальной плазмы. М.: МАКС Пресс, 2005. 524 с.
2. *Лукьяница А. А.* Скрытые модели Маркова. http://leader.cs.msu.su/ luk/HMM_rus.html
3. *Rabiner L. R.* A Tutorial on Hidden Markov Models and Selected applications in speech recognition. // Proceeding of the IEEE, Feb 1989. Vol. 77. N 2.

УДК 517.5 (683.32)

ФОРМУЛЫ ДЛЯ ПРЕОБРАЗОВАНИЯ ФУНКЦИЙ В ПРОСТРАНСТВЕ КОЭФФИЦИЕНТОВ РАЗЛОЖЕНИЯ ПО БАЗИСУ ПОЛИНОМОВ ЧЕБЫШЕВА ВТОРОГО РОДА

© 2007 г. Д. А. Новикова, А. В. Поволоцкий

novikova_dari@bk.ru, arseny-povolotsky@yandex.ru

Кафедра Математических методов прогнозирования

Введение. Рассмотрим задачу аппроксимации функции(или сигнала) заданным базисом из полиномов дискретного или непрерывного аргумента. Другими словами, это можно назвать отображением пространства, на котором задана функция, в пространство коэффициентов разложения этой функции по базису.

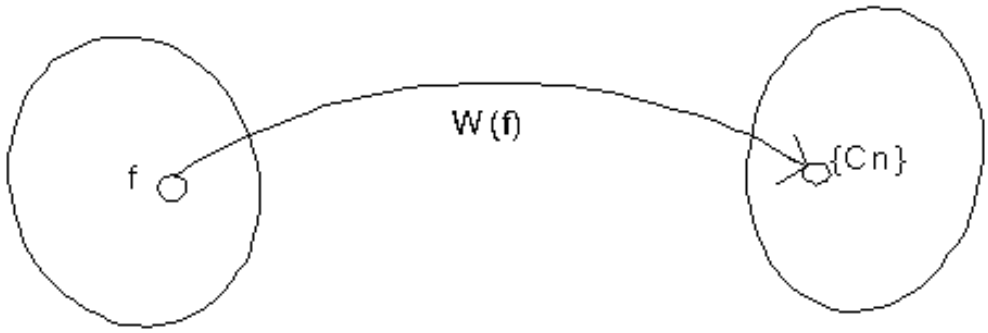


Рис. 1. Переход от пространства функций к пространству коэффициентов разложения

В процессе работы с функцией f возможны преобразования, меняющие ее, а также возможно взаимодействие с другими функциями пространства, на котором задана функция. При этом результирующей функции будет соответствовать другой вектор коэффициентов разложения.

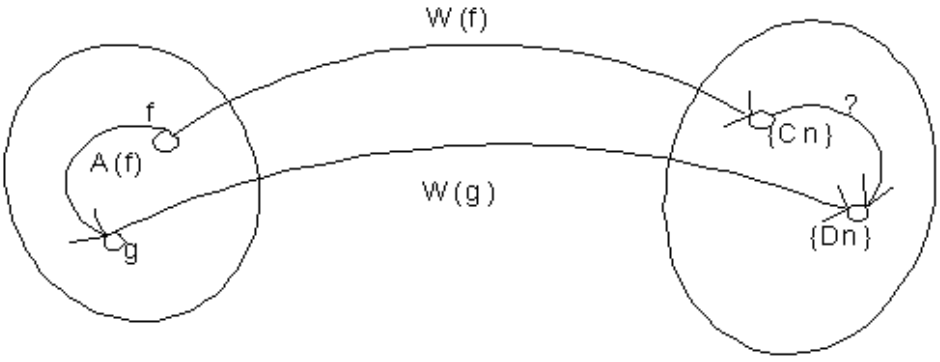


Рис. 2. Преобразования в пространстве функций и пространстве коэффициентов

В некоторых случаях удобнее перейти от пространства функций к пространству коэффициентов разложения и работать только в нем, не храня дополнительно информации о функции.

1. Постановка задачи. В работе рассматривается задача поиска алгоритмов преобразования коэффициентов разложения в пространстве коэффициентов разложения по базису полиномов, соответствующих некоторым известным операторам преобразований сигналов. Рассмотрим пространство функций L_2. В качестве базиса рассмотрим семейство полиномов Чебышева второго рода $\{U_i(x)\ i = 0 \dots m\}$:

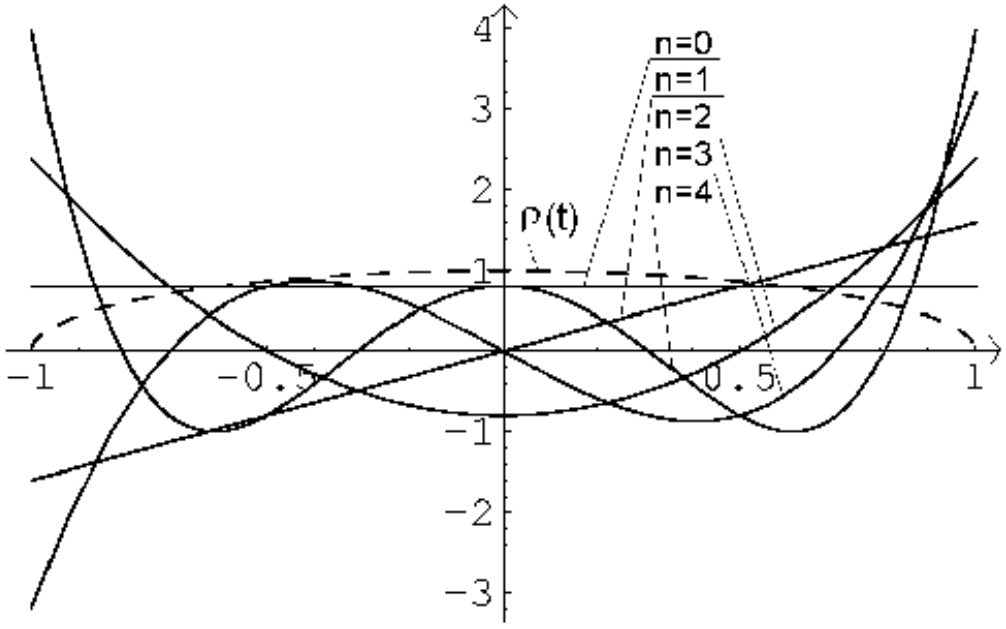


Рис. 3. Полиномы Чебышева второго рода $U_n(t)$, при $n = 0 \dots 4$

Рассмотрим сигнал или функцию $f_i = f(x_i)$ дискретного аргумента $x_i \in (a, b), i = 1...N$. *Спектром* f назовем вектор $\{C_n\} = \{C_0 \dots C_m\}$ коэффициентов разложения по базису $\{U_i(x)\}, i = 0 \dots m$. Рассмотрим линейный оператор A: $A(\alpha f + \beta g) = \alpha A(f) + \beta A(g)$.

Требуется разработать алгоритм, осуществляющий отображение: $\{C_n\} \to \{D_n\}$, соответствующий оператору $A : f \to A(f)$, где $\{C_n\}$ - спектр *f*, а $\{D_n\}$ - спектр *A(f)*.

Рассматриваются случаи операторов сложения, вычитания сигналов, умножения сигнала

на число, дифференцирования, интегрирования сигнала, умножения сигналов,умножения на линейный множитель.

2. Начальные сведения о семействе полиномов Чебышева второго рода.
1)Обозначение:

$$U_k(t), t \in (-1, 1).$$

2)Общий вид:

$$U_k(t) = \frac{sin[(k+1)arccos(t)]}{\sqrt{1-t^2}}.$$

3)Весовая функция:

$$\rho(t) = \sqrt{1-t^2}.$$

4)Соотношение ортогональности:

$$\int_{-1}^{1} U_n(t)U_m(t)\sqrt{1-t^2}dt = 0,\ m \neq n.$$

5)Рекуррентное соотношение:

$$U_{n+1}(t) = 2tU_n(t) - U_{n-1}(t).$$

6)Частные случаи:

$$U_0(t) = 1,\ U_1(t) = 2t,\ U_2(t) = 4t^2 - 1,\ ...$$

3. Спектральное разложение и восстановление сигнала по базису полиномов Чебышева второго рода. Коэффициенты разложения $\{C_n\} = \{C_0...C_m\}$ считаются по формуле:

$$C_k = \int_{-1}^{1} f(t)U_n(t)\rho(t)dt = \int_{-1}^{1} f(t)U_n(t)\sqrt{1-t^2}dt$$

Для подсчета коэффициентов используется метод квадратурных формул Гаусса, позволяющий достаточно точно подсчитать интеграл $\int_{-1}^{1} F(t)\sqrt{1-t^2}dt$ следующим способом:

$$\int_{-1}^{1} F(t)\sqrt{1-t^2}dt = \sum_{i=1}^{N} F(x_i)w_i,$$

где x_i - узлы сетки Гаусса, w_i - веса Гаусса, $i = 1 \dots N$.
Узлами сетки Гаусса являются корни полиномов Чебышева второго рода, веса Гаусса считаются по формуле: $w_i = \frac{C_N}{C_{N+1}} \frac{\|U_{N-1}\|^2}{U'_N(x_i)U_{N-1}(x_i)}$, где C_N, C_{N+1} -коэффициенты при старших степенях N-го и $N+1$-го полиномов, $\|U_n\|^2 = \int_{-1}^{1} U_n(t)U_n(t)dt$ -норма полинома n-ой степени.
Узлами сетки Гаусса являются точки:

$$x_k = cos\left(\frac{\pi k}{N+1}\right),\ k = 1 \dots N.$$

Подсчитаем веса Гаусса: из рекуррентного соотношения(п.2) видно, что $C_N = 2^N$.
$\|U_n\|^2 = \frac{\pi}{2}$ для каждого n. Для вычисления $U'_N(t)$ и $U_{N+1}(t)$ воспользуемся общим видом полиномов(п.2). Таким образом, для рассматриваемого семейства полиномов веса Гаусса имеют вид:

$$w_k = -\frac{\pi sin^3(\frac{\pi k}{N+1})}{(N+1)\, cos\pi k\, sin(\frac{\pi k}{N+1}\pi k)},\ k = 1 \dots N.$$

Спектральное разложение сигнала представляет собой вычисление его спектра.
Восстановление сигнала осуществляется следующим образом:

$$f(x) \approx \sum_{i=0}^{m} C_i U_i(x).$$

4. Метод каскадов и диффузий. Для реализации преобразований над вектором коэффициентов разложения, представленных рекуррентым соотношением, существует быстрый и удобно реализуемый на ЭВМ метод *каскадов и диффузий*[3]. Он работает с вектором, преобразуя его в соответствии с содержанием рекуррентного соотношения. Рассмотрим в качестве примера рекуррентное соотношение следующего вида:

$$A(P_n(x)) = \underbrace{\alpha_n P_{n-1}(x) + \beta_n P_n(x) + \gamma_n P_{n+1} \dots}_{\textit{диффузионная часть}} + \underbrace{a_n A(P_{n-1}(x)) + b_n A(P_{n-2}(x)) \dots}_{\textit{каскадная часть}}$$

На рисунке 3 показано, каким образом преобразуется вектор с применением каскада(процесса распространения элементов исходного вектора на элементы этого же вектора) и диффузии(процесса распространения элементов исходного вектора на элементы результирующего вектора):

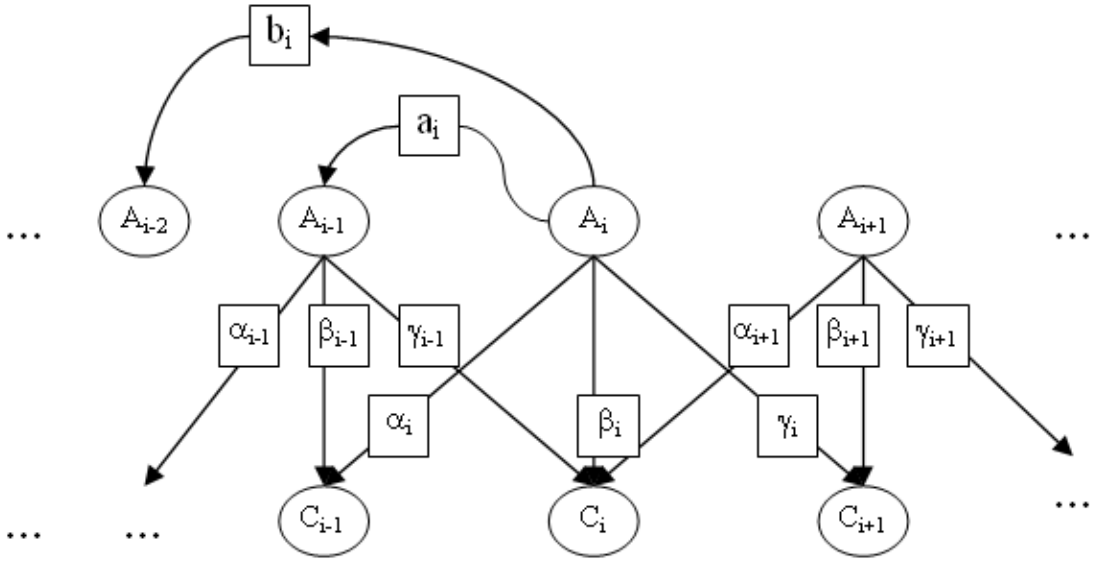


Рис. 4. Преобразование вектора методом каскадов и диффузий.

Такой метод вычисления элементов искомого вектора имеет линейную сложность и алгоритмически просто записывается.
Для выработки алгоритма преобразования спектра сигнала необходимо выяснить, как ведет себя каждый полином семейства при том или ином преобразовании, выписать рекуррентное соотношение и в соответствии с ним преобразовать элементы спектра.

Рассмотрим сигналы $f(t) = \sum\limits_{i=0}^{m} C_i U_i(t)$ $g(t) = \sum\limits_{j=0}^{m} B_j U_j(t)$, имеющие соответственно спектры $\{C_0 \dots C_m\}$ и $\{B_0 \dots B_m\}$.

4. Сложение, вычитание, умножение на число полиномов Чебышева второго рода. Преобразование коэффициентов.

Сумма, разность: $f(t) \pm g(t) = \sum\limits_{k=0}^{m} D_k U_k(t)$, $D_k = C_k \pm B_k$, $k = 0 \dots m$.
Преобразование коэффициентов разложения:

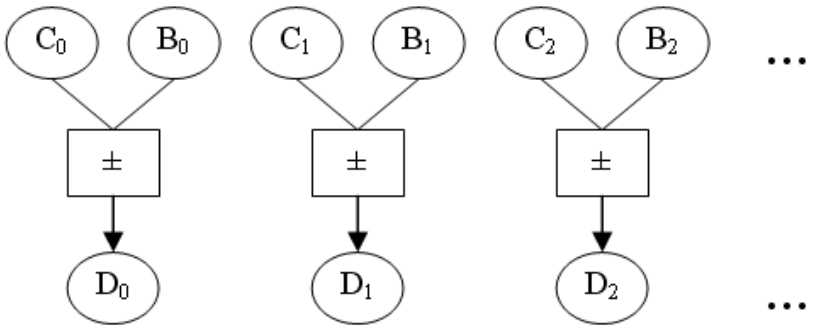


Рис. 5. Преобразование коэффициентов при сложении(вычитании) сигналов.

Умножение на число: $\alpha f(t) = \sum\limits_{k=0}^{m} D_k U_k(t)$, $D_k = \alpha C_k$, $k = 0 \dots m$.
Преобразование коэффициентов разложения:

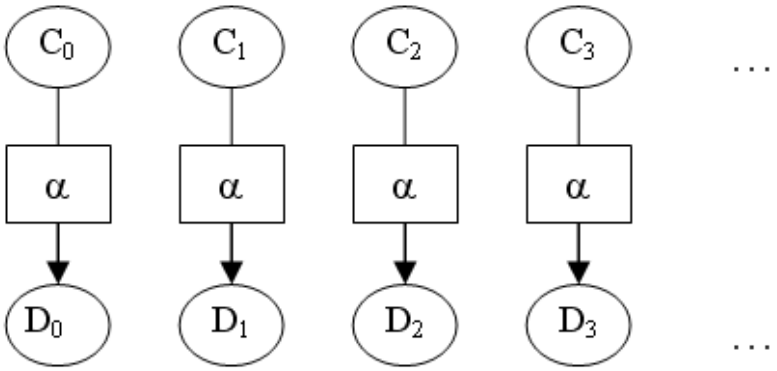


Рис. 6. Преобразование коэффициентов при умножении сигнала на число.

5. Дифференцирование полиномов Чебышева второго рода. Преобразование коэффициентов.

$$f'(t) = \left(\sum_{k=0}^{m} C_k U_k(t) \right)' = \sum_{k=0}^{m} D_k U'_k(t).$$

Для полиномов Чебышева второго рода было получено следующее рекуррентное соотношение:

$$U'_n(t) = 2nU_{n-1}(t) + U'_{i-2}(t)$$

Таким образом, преобразование коэффициентов разложения f для получения спектра $f'(t)$ схематично выглядит следующим образом:

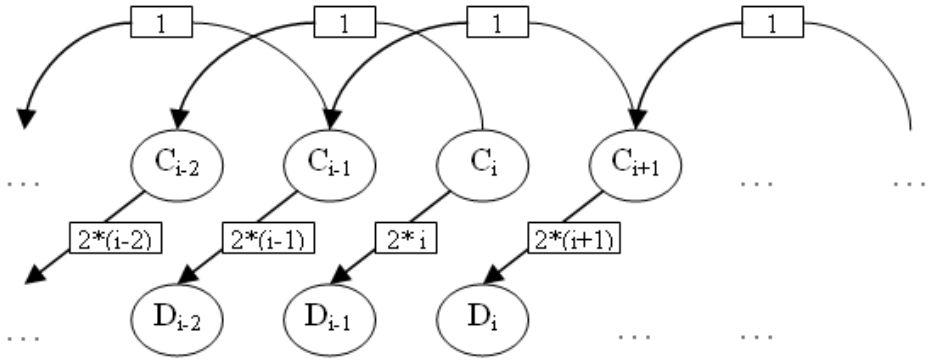


Рис. 7. Преобразование коэффициентов при дифференцировании сигнала.

Алгоритмически преобразование можно записать так: скопировать исходный вектор во временный $\{Z_k\}$, $k = 0 \dots m$. Для каждого $i \in \{m \dots 0\}$ выполнить:

$$Z_{i-2} := Z_{i-2} + Z_i;\ D_{i-1} := D_{i-1} + 2i\ Z_i$$

6. Интегрирование полиномов Чебышева второго рода. Преобразование коэффициентов.

`Определенный интеграл`:

$$\int_{-1}^{1} f(t)dt = \sum_{k=0}^{m} \int_{-1}^{1} C_k U_k(t)dt$$

Было выведено следующее соотношение:

$$\int_{-1}^{1} U_n(t)dt = \begin{cases} \frac{2}{n+1}, & \text{если } n \text{ четно} \\ 0, & \text{если } n \text{ нечетно} \end{cases}$$

Преобразование вектора коэффициентов:

$$\{C_0, C_1, ...C_m\} \to \sum_{k=0}^{\lfloor \frac{m}{2} \rfloor} \frac{2}{2k+1} C_{2k}$$

`Неопределенный интеграл`:

$$\int_{-1}^{x} f(t)dt = \sum_{k=0}^{m} \int_{-1}^{x} C_k U_k(t)dt$$

Замечание Для определенности константы интегрирования рассмотрим интеграл с пределами интегрирования -1 и x.

Было выведено следующее рекуррентное соотношение:

$$\int_{-1}^{x} U_n(t)dt = \frac{U_{n+1}(t) - U_{n-1}(t)}{2(n+1)}$$

Таким образом, преобразование коэффициентов разложения f для получения спектра $\int_{-1}^{x} f(t)dt$ схематично выглядит следующим образом:

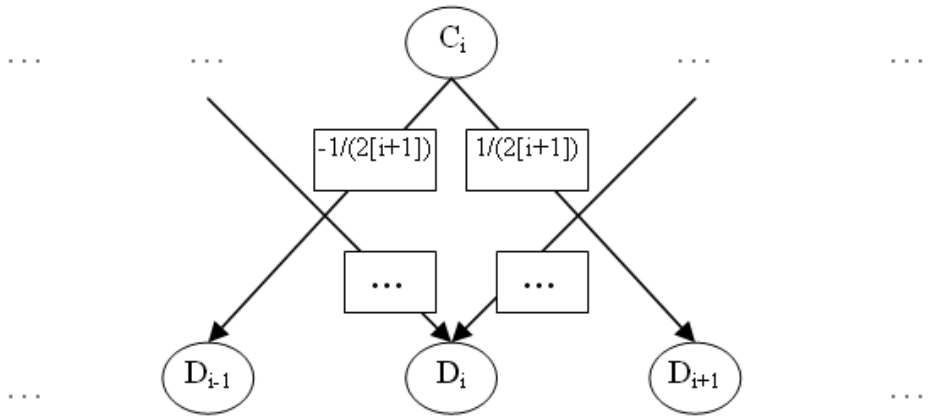


Рис. 8. Преобразование коэффициентов при интегрировании сигнала.

Алгоритмически преобразование коэффициентов можно сформулировать следующим образом: для каждого $i \in \{m \dots 0\}$ выполнить:

$$D_{i-1} := D_{i-1} - \frac{1}{2i+1}C_i;\ D_{i+1} := D_{i+1} + \frac{1}{2i+1}C_i$$

7. Умножение сигналов. Преобразование коэффициентов.

Рассмотрим $f(t) = \sum_{i=0}^{m_1} C_i U_i(t)$ и $g(t) = \sum_{j=0}^{m_2} B_j U_j(t)$.

$$f(t) * g(t) = \left[\sum_{i=0}^{m_1} C_i U_i(t)\right]\left[\sum_{j=0}^{m_2} B_j U_j(t)\right] = \sum_{i=0}^{m_1}\sum_{j=0}^{m_2} C_i B_j U_i(t)U_j(t) = \sum_{k=0}^{m_1+m_2} D_k U_k(t).$$

Для попарного произведения полиномов $U_i(t)U_j(t)$ выведено следующее рекуррентное соотношение:

$$U_i(t)U_j(t) = U_i(t)U_{j+1}(t) + U_i(t)U_{j-1}(t) - U_{i-1}(t)U_j(t).$$

Используя предложенное выше рекуррентное соотношение, можно применить метод каскадов и диффузий для вычисления искомого спектра. Без ограничения общности будем считать, что $m_2 \geqslant m_1$. Составим матрицу размеров $(m_1+1)*(m_1+m_2+1)$, отображающую коэффициенты перед каждым произведением вида $U_i(t)U_j(t)$, где $i \in \{0 \dots m_1\}$, $j \in \{0 \dots m_2\}$ после того, как сигналы были перемножены и все подобные слагаемые были приведены:

$$\begin{pmatrix} A_{00} & \cdots & A_{0m_1} & \cdots & A_{m_1+m_2} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ A_{m_1 0} & \cdots & A_{m_1 m_1} & \cdots & A_{m_1(m_1+m_2)} \end{pmatrix}$$

, где

$$A_{ij} = \begin{cases} C_i B_i, & \text{если } i = j,\ j \leqslant m_1 \\ C_i B_j + C_j B_i, & \text{если } i < j \leqslant m_1 \\ 0, & \text{иначе} \end{cases}$$

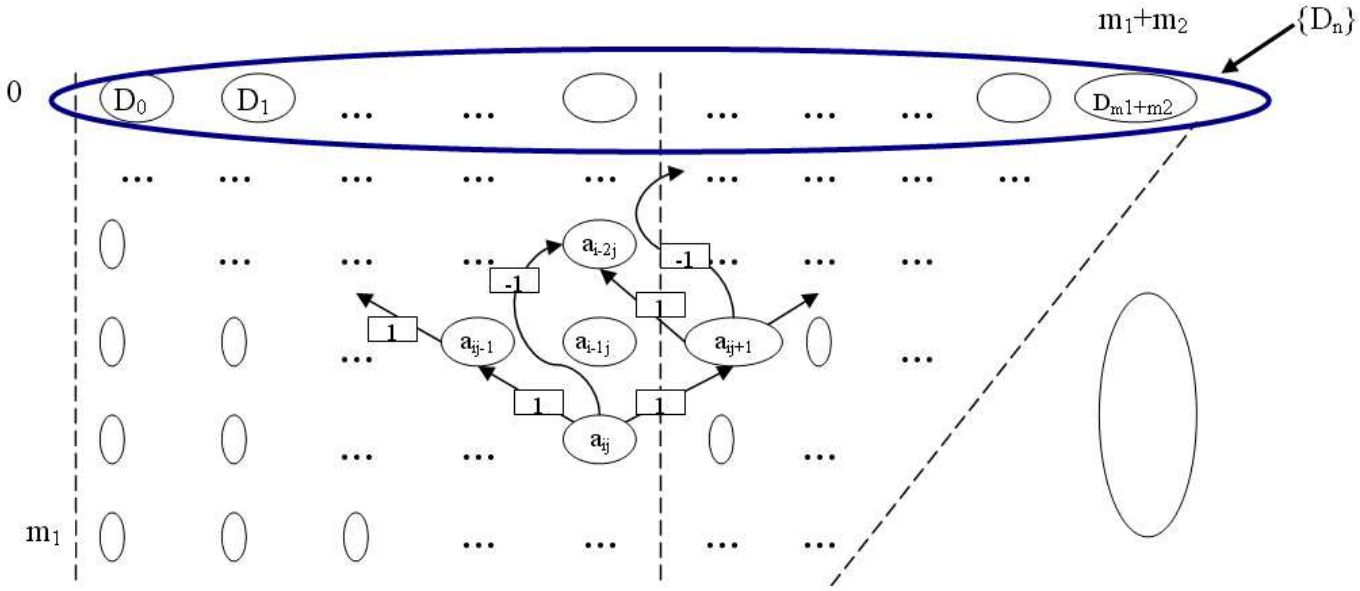


Рис. 9. Преобразование коэффициентов при умножении сигналов.

Будем преобразовывать каждую строку матрицы так, как это отражено в рекуррентном соотношении.

Преобразование матрицы коэффициентов схематично можно представить следующим образом:

Алгоритмически преобразование коэффициентов выглядит следующим образом: в каждой строке $i \in \{0 \dots m_1\}$ для каждого элемента выполнить:

$$A_{i(j-1)} := A_{i(j-1)} + A_{ij};\ A_{i(j+1)} := A_{i(j+1)} + A_{ij};\ A_{i(j-2)} := A_{i(j-2)} - A_{ij};$$

Первая строка преобразованной матрицы содержит коэффициенты перед произведениями вида $U_0(t)U_k(t)$. Из начальных сведений(п.2) видно, что $U_0(t) = 1$, следовательно, первая строка содержит коэффициенты искомого спектра произведения сигналов. Поэтому она и будет результирующим вектором.

8. Умножение сигнала на линейный множитель (at+b). Преобразование коэффициентов.

$$f(t)(at+b) = \sum_{k=0}^{m} C_k U_k(t)(at+b) = at \sum_{k=0}^{m} C_k U_k(t) + b \sum_{k=0}^{m} C_k U_k(t)$$

Умножение на линейный множитель сводится к двум частям: умножение на константу a и t, а также умножение на константу b. Умножение на константу описано выше(п.4). Используем выведенное рекуррентное соотношение:

$$tU_n(t) = \frac{U_{n+1}(t) - U_{n-1}(t)}{2}.$$

Преобразование коэффициентов схематично можно представить следующим образом:

Заключение. Рассмотрена задача разработки алгоритмов, осуществляющих преобразования над функциями из пространства L_2 и действующих в пространстве коэффициентов разложения по базису полиномов Чебышева второго рода.

Разработаны алгоритмы для осуществления операций сложения, вычитания, умножения на число, дифференцирования, интегрирования, умножения функций, умножения на линейный множитель. Комбинируя алгоритмы, можно реализовать различные операции над сигналами.

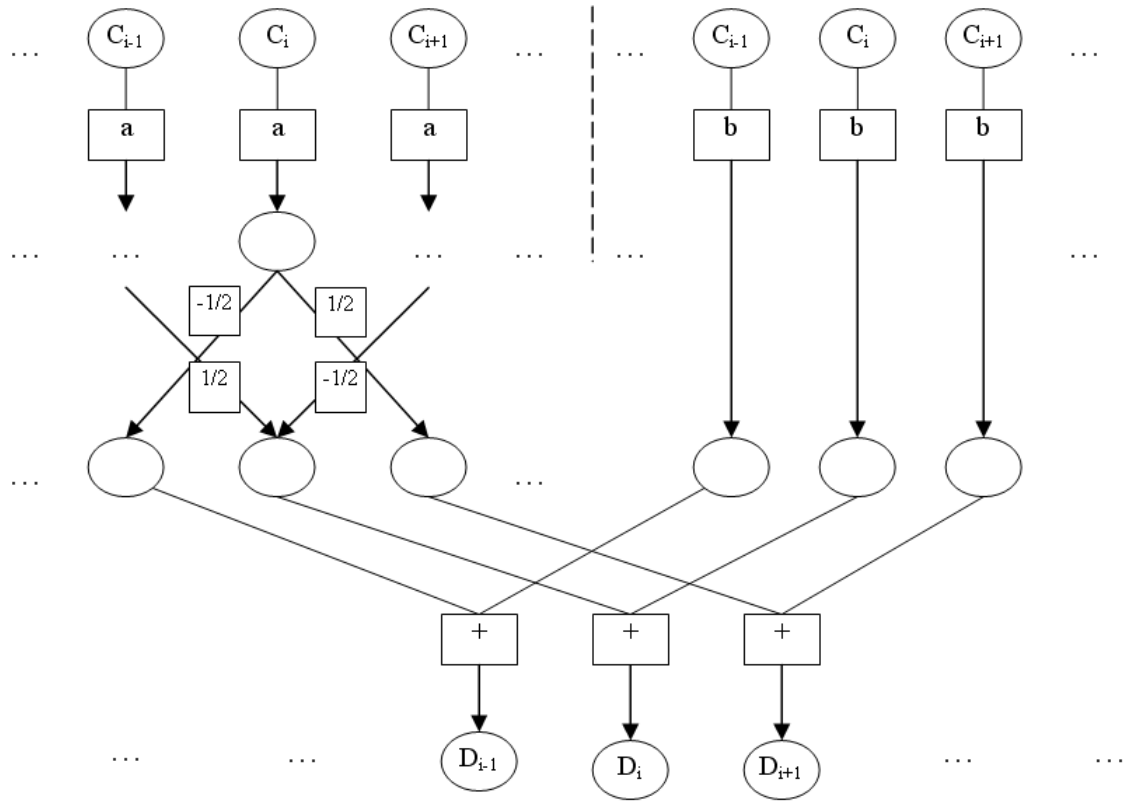


Рис. 10. Преобразование коэффициентов при умножении на линейный множитель.

Настоящая работа выполнена при частичной поддержке гранта РФФИ 06-01-08039. Спасибо за содействие Дедусу Ф.Ф., Панкратову А.Н., Тетуеву Р.К.

СПИСОК ЛИТЕРАТУРЫ

1. *Ф.Ф. Дедус, Л.И. Куликова, А.Н. Панкратов, Р.К. Тетуев.* Классические ортогональные базисы в задачах аналитического описания и обработки информационных сигналов - Москва, 2004.
2. *А.Ф.Никифоров ,В.Б. Уваров.* Специальные функции математической физики - Москва: Наука, 1978.
3. *П.К. Суетин.* Классические ортогональные многочлены - Москва: Наука, 1976.
4. *Р.К.Тетуев, Ф.Ф.Дедус.* Классические ортогональные полиномы. Применение в задачах обработки данных - Пущино: препринт ИМПБ, 2007.
5. W. H. Press and others Numerical Recipes// Cambridge University Press, 1992.

СБОРНИК СТАТЕЙ МОЛОДЫХ УЧЕНЫХ факультета ВМиК МГУ, 2007, выпуск № 4, с. 129–135

УДК 519.1

АСИМПТОТИКА ЛОГАРИФМА ЧИСЛА МНОЖЕСТВ, СВОБОДНЫХ ОТ ПРОИЗВЕДЕНИЙ, В ГРУППАХ

© 2007 г. Т. Г. Петросян

taron@nm.ru

Кафедра математической кибернетики

ВВЕДЕНИЕ

Пусть A подмножество элементов группы G. Если не существует тройки $(x, y, z) \in A^3$, такой, что $xy = z$, то говорят, что множество A *свободно от произведений*. Семейство всех множеств, свободных от произведений (МСП), группы G обозначим через $\mathrm{PF}(G)$. Множество $A \in \mathrm{PF}(G)$ называется *максимальным МСП* в G, если $|A| \geq |T|$ при любом $T \in \mathrm{PF}(G)$. Обозначим через $\lambda(G)$ размер максимального МСП группы G.

Н. Алон (Alon, N.) [4] доказал, что

$$|\mathrm{PF}(G)| \leq 2^{\left(1/2+o(1)\right)n},$$

где G – произвольная конечная группа порядка n.

Данная теорема была уточнена в работах [1] и [2] для конечных групп, содержащих нормальную подгруппу простого индекса.

[2, Петросян, теорема 1] *Для любой группы G порядка n с числом подгрупп индекса 2, равным t, $t \geq 1$,*

$$t \cdot 2^{n/2} - 2^{n(1+\varphi(n))/4} \leq |\mathrm{PF}(G)| \leq t \cdot 2^{n/2} + 2^{n(1/2-\epsilon)},$$

где $\epsilon > 0$, $\varphi(n) \to 0$ при $n \to \infty$ и второе неравенство справедливо, начиная с некоторого n.

[3, Петросян, теорема 5] *Пусть G конечная группа порядка n, а K – ее нормальная подгруппа наименьшего простого индекса. Пусть $|G : K| = p$, $p \geq 3$, где p – простое число. Тогда*

$$2^{\lceil \frac{p-1}{3} \rceil \frac{n}{p}} \leq |\mathrm{PF}(G)| \leq 2^{\frac{n}{2}\left(1-\frac{0{,}004}{p}\right)}.$$

В данной работе будет доказан следующий результат.

Теорема 1. *Пусть G – конечная группа порядка n и $\lambda(G) \geq |G|/3$. Тогда*[*]

$$\log |\mathrm{PF}(G)| = \lambda(G) + O\big(n(\log n)^{-1/45}\big).$$

Эта теорема обобщает [5, Green-Ruzsa, Th. 8], доказанную для абелевых групп, на случай произвольных конечных групп.

ВСПОМОГАТЕЛЬНЫЕ УТВЕРЖДЕНИЯ

Нам понадобится следующая теорема.

[1, Кострикин, гл. 8, п. 5, теорема 5] *Представления степени 1 конечной группы G над $\mathbb{C}$ находятся в биективном соответствии с неприводимыми представлениями факторгруппы G/G' (G' – коммутант группы G). Их число равно индексу $|G : G'|$.*

Из последней теоремы легко вывести следующее равенство.

Следствие 1. $\sum_{\gamma \in \Gamma} \gamma(x) = \frac{|G|}{|G'|}\delta_{x,e}$, *где $\delta_{x,e}$ - символ Кронекера.*

[*]Здесь и далее логарифмы берутся по основанию 2

Обозначим через $a_k(G)$ число подгрупп индекса k группы G.

[7, Lubotzky-Segal, Corollary 1.7.5] *Пусть G произвольная конечная группа. Тогда*

$$a_k(G) \leq k^{2\log|G|}.$$

Следствие 2. *Число подгрупп конечной группы G не превосходит $2^{3\log^2|G|}$.*

[5, Green-Ruzsa, Lemma 13] *Пусть $x_1, \ldots, x_k$ являются действительными числами, такими, что $x_i \geq 1$ и $\sum x_i \leq K$. Пусть $\tau \geq e^{1/e}$. Тогда*

$$\prod_{i=1}^{k} \max(\tau, x_i) \leq \tau^K.$$

[5, Green-Ruzsa, Lemma 15] *Пусть ρ не превосходит некоторой положительной константы и n достаточно велико. Тогда число подмножеств мощности, не превосходящей ρn, некоторого n элементного множества не больше чем $2^{n\rho^{1/2}}$.*

Как обычно, обозначим через $\delta(\Gamma)$ минимальную степень графа Γ.

[6, Lev-Luczak-Schoen, Lemma 6] *Для любого графа $\Gamma = (V, E)$ средняя степень, которого не меньше $(1-\lambda)|V|$, существует подграф $\Gamma' = (V', E')$, такой, что:*

$$1)\ |V'| \geq (1-\sqrt{\lambda})|V|;$$

$$2)\ \delta(\Gamma') > (1-2\sqrt{\lambda})|V|.$$

Пусть A является подмножеством (не обязательно конечной) группы G и $K \in \mathbb{N}$. Обозначим через $Q_K(A)$ множество всех элементов $g \in G$, имеющих по крайней мере K различных представлений в виде $a_1 a_2^{-1}$ или $a_1^{-1} a_2$, где $a_1,\ a_2 \in A$. Для произвольного $X \subseteq G$ обозначим множество $X^{-1}X \cup XX^{-1}$ через $\tilde{X}$. Докажем обобщение [6, Lev-Luczak-Schoen, Proposition 1] с абелевых групп на случай произвольных конечных групп.

Лемма 1. *Пусть $A \subset G,\ K \in \mathbb{N}$ и*

$$|Q_K(A)| \leq 2|A| - 5\sqrt{K|\tilde{A}|}.$$

Тогда существует подмножество $T \subseteq A$, такое, что

$$|A \setminus T| \leq \sqrt{K|\tilde{A}|} \text{ и } \tilde{T} \subseteq Q_K(A).$$

Доказательство. Без ограничения общности можем считать, что $K \leq |A|$. В самом деле, если $K > |A|$, то $\sqrt{K|AA^{-1}|} > |A|$. Из последнего неравенства очевидным образом следует утверждение леммы.

Рассмотрим граф $\Gamma = (A, E)$ на множестве A, в котором $(a_1, a_2) \in E$ тогда и только тогда, когда $a_1^{-1}a_2$ или $a_1 a_2^{-1}$ принадлежит $Q_K(A)$. Ребра дополнения $\bar{\Gamma}$ соответствуют элементам $c \in \tilde{A} \setminus Q_K(A)$. Заметим, что двум элементам c и c^{-1} соответствует одно и то же ребро. Отсюда следует, что число ребер дополнения $\bar{\Gamma}$ не превосходит

$$\frac{1}{2}(K-1)|\tilde{A} \setminus Q_K(A)| \leq \frac{1}{2}(K-1)|\tilde{A}|.$$

Следовательно, число ребер в Γ не меньше

$$\binom{|A|}{2} - \frac{1}{2}(K-1)|\tilde{A}|,$$

а средняя степень Γ не меньше

$$|A| - 1 - (K-1)|\tilde{A}|/|A| \geq |A| - K|\tilde{A}|/|A| = |A|(1 - K|\tilde{A}|/|A|^2).$$

Из [6, Lemma 6] следует, что существует подграф $\Gamma' = (T, E')$, такой, что

$$|T| \geq |A|\Big(1 - \sqrt{K|\tilde{A}|/|A|^2}\Big) = |A| - \sqrt{K|\tilde{A}|},$$

и степень $d(t)$ любой вершины $t \in T$ больше чем

$$|A| - 2\sqrt{K|\tilde{A}|}.$$

Предположим, что $\tilde{T} \not\subseteq Q_K(A)$. Тогда существуют два элемента u и v из T, такие, что либо $u^{-1}v \notin Q_K(A)$, либо $uv^{-1} \notin Q_K(A)$. Допустим, что $uv^{-1} \notin Q_K(A)$. Следовательно, число представлений uv^{-1} в виде отношения двух элементов из A (либо $a_1^{-1}a_2$, либо $a_1a_2^{-1}$), не превосходит $K-1$. Отсюда

$$|\big(N(u)\big)^{-1}u \cap \big(N(v)\big)^{-1}v| \leq K - 1.$$

Поскольку множества $\big(N(u)\big)^{-1}u$ и $\big(N(v)\big)^{-1}v$ содержаться в $Q_K(A)$, то

$$|Q_K(A)| \geq |N(u)| + |N(v)| - (K-1) > 2|A| - 4\sqrt{K|\tilde{A}|} - K >$$

$$> 2|A| - 5\sqrt{K|\tilde{A}|}.$$

Случай, когда $uv^{-1} \in Q_K(A)$ и $u^{-1}v \notin Q_K(A)$ рассматривается аналогично. □

Следующая лемма является обобщением [5, Green-Ruzsa, Proposition 11].

Лемма 2. *Пусть $\varepsilon > 0$. Пусть число элементов множества $F \subseteq G$ не меньше чем $\big(\frac{1}{3} + \varepsilon\big)n$, и F содержит не более чем $\varepsilon^3 n^2/27$ троек Шура. Тогда существует МСП $S \subseteq F$, такое, что $|S| \geq |F| - \varepsilon n$.*

Доказательство. Введем обозначения: $N = \varepsilon^3 n^2/27$ и $K = \lceil N^{2/3} n^{-1/3} \rceil$. Заметим, что

$$N \geq |Q_K(F) \cap F| K, \tag{1}$$

иначе множество F порождало бы более чем N троек Шура. Следовательно,

$$\frac{N}{K} \geq |Q_K(F) \cap F| \geq |Q_K(F)| + |F| - n.$$

Поэтому

$$|Q_K(F)| \leq \frac{N}{K} - |F| + n \leq \frac{N}{K} + 2|F| - 3\varepsilon n.$$

Из того что

$$\frac{N}{K} + 5\sqrt{KN} < (Nn)^{1/3} + 5\sqrt{2}(Nn)^{1/3} < 9(Nn)^{1/3} = 3\varepsilon n$$

следует, что

$$|Q_K(F)| < 2|F| - 5\sqrt{Kn} \leq 2|F| - 5\sqrt{K|\tilde{F}|}.$$

Из леммы 1 следует существование подмножества F', такого, что $|F'| \geq |F| - \sqrt{Kn}$ и $F'(F')^{-1} \subseteq Q_K(F)$. Ясно, что множество $S = F' \setminus F'(F')^{-1}$ является свободным от произведений. Из (1) вытекает

$$|F' \cap F'(F')^{-1}| \leq |F \cap Q_K(F)| \leq \frac{N}{K}.$$

Следовательно,

$$|S| \geq |F'| - \frac{N}{K} \geq |F| - \left(\frac{N}{K} + \sqrt{Kn}\right) > |F| - 3(Nn)^{1/3} = |F| - \varepsilon n.$$

□

Следствие 3. *Пусть G – конечная группа и $F \subseteq G$ содержит δn^2 троек Шура. Тогда* $|F| \leq \left(\max\left(\frac{1}{3}, \mu(G)\right) + 3\delta^{1/3}\right)n$.

Лемма 3. *Пусть n достаточно большое и $L \leq n^{1/2}$. Тогда число подмножеств группы G, являющихся L-гранулярными, не превосходит $2^{2n/L}$.*

Доказательство. Из следствия 2 получаем, что число подгрупп группы G не превосходит $2^{3\log^2 n}$. Следовательно число множеств типа объединения смежных классов не превосходит $2^{3\log^2 n + n/L}$. Ясно, что число множеств типа объединения L-гранул не более чем $n2^{n/L}$. □

ОСНОВНАЯ ЛЕММА

Пусть Γ – группа характеров на G. Если $f : G \to \mathbb{R}$, то определим преобразование Фурье функции f относительно $\gamma \in \Gamma$ формулой

$$\hat{f}(\gamma) = \sum_{x \in G} f(x)\gamma(x).$$

Напомним понятие *гранулы*, введенное в [5, Green-Ruzsa]. Пусть $d \in G$ имеет порядок m. Рассмотрим факторгруппу G/G_d, где G_d – подгруппа G порожденная элементом d. Разобьем каждый смежный класс группы G на $\lfloor m/L \rfloor$ частей $\{x, xd, \ldots, xd^{L-1}\}$ и некоторый остаток из $m \pmod L$ элементов. Мы зафиксируем некоторое разбиение такого вида. Элемент разбиения, не являющийся остатком, будем называть *гранулой первого типа.* Другим типом гранул являются смежные классы по некоторой подгруппе H, $|H| \geq M$, группы G, назовем их *гранулами второго типа.* Множество называется *L-гранулярным типа прогресии*, если оно является объединением гранул первого типа. Если множество является объединением гранул второго типа, то оно называется *M-гранулярным типа объединения смежных классов.*

Через $C(x)$ обозначим характеристическую функцию множества C. Она равна 1, если $x \in C$, и 0 – в противном случае.

Следующая лемма является обобщением [5, Green-Ruzsa, Lemma 12] на случай некоммутативных групп.

Лемма 4. *Пусть $C \in \mathrm{PF}(G)$ и $\alpha \in (0, 1/8)$. Пусть натуральные числа L и M удовлетворяют неравенству*

$$n > M(3L/\alpha)^{2^{20}\alpha^{-8}}. \tag{2}$$

Тогда существует множество C', такое, что:

1. C' является либо L-гранулярной типа прогрессии, либо M-гранулярной типа объединения смежных классов;
2. $|C \setminus C'| \leq \alpha n$;
3. C' порождает не более чем αn^2 троек Шура.

Доказательство.
Рассмотрим функцию

$$g(\gamma) = \frac{1}{|P|} \sum_{b \in P} \gamma(b).$$

В зависимости от C множество P будет либо подгруппой $H \leq G$, либо прогрессией $\{d^{-(L-1)}, \ldots, d^{L-1}\}$.

Мы позднее выберем P таким образом, чтобы для всех $\gamma \in \Gamma$ выполняется следующее неравенство

$$|\hat{C}(\gamma)(1 - g(\gamma))| \leq \delta n, \tag{3}$$

где $\delta = 2^{-8}\alpha^4$.

Определим множество C'. Если $P = H$ является подгруппой, то через C' обозначим объединение тех левых смежных классов по подгруппе H, в которых содержится хотя бы $\alpha|H|$ элементов из C. Очевидно, что свойства (1) и (2) выполняются в этом случае. Пусть теперь $P = \{d^{-(L-1)}, \ldots, d^{L-1}\}$, где $\mathrm{ord}(d) = m$, $\mathrm{ord}(m) \geq 2L-1$. В этом случае в качестве C' возьмем объединение гранул, содержащих хотя бы $\alpha L/2$ элементов из C. Тогда множество $C \setminus C'$ содержит не более чем $\alpha L/2$ элементов из каждой гранулы, и не более чем L элементов из оставшихся n/m частей смежных классов. Если потребовать выполнения условия

$$m \geq \frac{2L}{\alpha}, \tag{4}$$

то верна оценка

$$|C \setminus C'| \leq \frac{\alpha n}{2} + \frac{nL}{m} \leq \alpha n.$$

Таким образом, первые два свойства леммы выполняются в случаях, когда P является прогрессией типа объединения смежных классов или же прогрессией типа объединения гранул.

Для того, чтобы доказать третье свойство, введем вспомогательную функцию $d(x) = |C \cap xP|/|P|$. Заметим, что $\hat{d}(\gamma) = \hat{C}(\gamma)g(\gamma)$. Поскольку множество $C \in \mathrm{PF}(G)$, верно равенство:

$$\sum_{xy=z} d(x)d(y)d(z) = \sum_{xy=z} \big(d(x)d(y)d(z) - C(x)C(y)C(z)\big). \tag{5}$$

Пользуясь следствием 1 ($\sum_{\gamma\in\Gamma} \gamma(x) = k\delta_{x,e}$, где $k = |G : G'|$) и равенством Парсеваля оценим правую часть равенства (5):

$$\begin{aligned}
&= n^{-1} \sum_{\gamma} \Big(|\hat{d}(\gamma)|^2 \hat{d}(\gamma) - |\hat{C}(\gamma)|^2 \hat{C}(\gamma) \Big) \\
&= n^{-1} \sum_{\gamma} |\hat{C}(\gamma)|^2 \hat{C}(\gamma) \big(1 - |g(\gamma)|^2 g(\gamma)\big) \\
&\leq n^{-1} \cdot \max_{\gamma} |\hat{C}(\gamma)||1 - |g(\gamma)|^2 g(\gamma)| \sum_{\gamma} |\hat{C}(\gamma)|^2 \\
&\leq |C| \cdot \max_{\gamma} |\hat{C}(\gamma)||1 - g(\gamma)^3| \leq 3|C| \cdot \max_{\gamma} |\hat{C}(\gamma)||1 - g(\gamma)|
\end{aligned}$$

Отсюда и из неравенства (3) следует, что

$$\sum_{xy=z} d(x)d(y)d(z) \leq 3\delta n^2. \tag{6}$$

Рассмотрим элемент $x \in C'$. Если P - подгруппа, то xP содержит по крайней мере $\alpha|P|$ элементов из C. Когда P является прогрессией, xP содержит гранулу множества C', в которой содержится x и, следовательно, еще $\alpha|P|/4$ элементов C. В обоих случаях $d(x)$ не меньше $\alpha/4$, поэтому $d(x) \geq \alpha C'(x)/4$ для всех $x \in G$. Отсюда и из (6) следует, что число троек Шура не превосходит

$$\sum_{xy=z} C'(x)C'(y)C'(z) \leq 2^8\alpha^{-3}\delta n^2 \leq \alpha n^2.$$

Итак, свойство 3 доказано.

Осталось показать, что существует множество P, такое, что выполняется неравенство (3). Так как $g(1) = 1$ и $g(\gamma) \in [-1, 1]$, то неравенство (3) выполняется для $\gamma = 1$ и тех γ, для которых $|\hat{C}(\gamma)| \le \delta n/2$. Пусть R, $|R| = k$, обозначает множество всех $\gamma \ne 1$, для которых $|\hat{C}(\gamma)| > \delta n/2$. Пусть Γ_1 является подгруппой, порожденной множеством R, а H является аннулятором Γ_1. Если $|H| \ge M$, то тогда мы берем в качестве P подгруппу H. Ясно, что в этом случае, неравенство (3) верно. Пусть $|H| < M$. Найдем элемент d, такой, чтобы множество $P = \{d^{-(L-1)}, \dots, d^{L-1}\}$ удовлетворяло неравенству (3). Рассмотрим произвольный характер $\gamma \in \Gamma$ и обозначим $\arg\gamma(d) = \beta \in [-\pi, \pi)$.

$$1 - g(\gamma) = \frac{2}{2L-1}\sum_{j=1}^{L-1}(1 - \cos j\beta) \le \frac{1}{2L-1}\sum_{j=1}^{L-1}(j\beta)^2 = \frac{L(L-1)}{6}\beta^2 \le \frac{(L\beta)^2}{6}.$$

Следовательно, если для любого $\gamma \in R$ выполняется

$$|\arg\gamma(d)| \le \frac{1}{L}\sqrt{\frac{6\delta n}{|\hat{C}(\gamma)|}}, \tag{7}$$

то неравенство (3) также будет выполняться. Для выполнения неравенства (4) потребуем, чтобы $d \notin H$ и усилим условие (7):

$$|\arg\gamma(d)| \le \frac{1}{L}\min\left(\alpha\pi, \sqrt{\frac{6\delta n}{|\hat{C}(\gamma)|}}\right). \tag{8}$$

Рассмотрим фактор-группу G/H. Ясно, что можно найти элемент $D \in G/H$, такой, что $D \ne H$ и $|\arg\gamma(D)| < \nu_\gamma$, при заданных положительных числах ν_γ, если

$$|G/H| > \prod_{\gamma\in R}\bigl(1 + \lfloor 2\pi/\nu_\gamma \rfloor\bigr). \tag{9}$$

При числах ν_γ, заданных правой частью неравенства (8), оценим сверху правую часть неравенства (9) через

$$\prod_{\gamma\in R}\left(1 + 2\pi L\max\Bigl(\frac{1}{\pi\alpha}, \sqrt{|\hat{C}(\gamma)|/6\delta n}\Bigr)\right) = \prod_{\gamma\in R}\left(1 + 2L\max\Bigl(\frac{1}{\alpha}, \frac{\pi}{\sqrt{6}}\sqrt{2^8|\hat{C}(\gamma)|/\alpha^4 n}\Bigr)\right)$$
$$\le \prod_{\gamma\in R}\left(1 + 2L\max\Bigl(\frac{1}{\alpha}, \frac{2^5}{\alpha^2}\sqrt{|\hat{C}(\gamma)|/n}\Bigr)\right) \le (3L)^k\prod_{\gamma\in R}\max\left(\frac{1}{\alpha}, \frac{2^5}{\alpha^2}\sqrt{|\hat{C}(\gamma)|/n}\right).$$

Пусть в [5, Green-Ruzsa, Lemma 13] $\tau = \alpha^{-1}$ и $x_\gamma = 2^{20}|\hat{C}(\gamma)|^2\alpha^{-8}n^{-2}$. Из равенства Парсеваля следует

$$\sum_{\gamma\in R}\frac{|\hat{C}(\gamma)|^2}{n^2} \le \frac{n|C|}{n^2}.$$

Поэтому в качестве K можно взять $2^{20}/\alpha^8$. Таким образом,

$$\prod_{\gamma\in R}\max\left(\frac{1}{\alpha}, \frac{2^5}{\alpha^2}\sqrt{\frac{|\hat{C}(\gamma)|}{n}}\right) \le \alpha^{-2^{20}/\alpha^8}.$$

Отсюда следует, что правая часть (9) не превосходит $(3L/\alpha)^{2^{20}/\alpha^8}$. С другой стороны, она не меньше чем n/M, поэтому из условия (2) следует выполнение (9). □

АСИМПТОТИКА ЛОГАРИФМА ЧИСЛА МСП

Пусть Γ – группа характеров на G. Если $f : G \to \mathbb{R}$, то определим преобразование Фурье функции f относительно $\gamma \in \Gamma$ формулой

$$\hat{f}(\gamma) = \sum_{x \in G} f(x)\gamma(x).$$

Через $C(x)$ обозначим характеристическую функцию множества C. Она равна 1, если $x \in C$, и 0 – в противном случае.

Теорема 2. *Пусть G – произвольная конечная группа. Существует семейство $\mathcal{F}$ почти МСП группы G со следующими свойствами:*

$$1) \log |\mathcal{F}| = o(n);$$

$$2) \textit{ Любое } A \in \mathrm{PF}(G) \textit{ содержится в некотором } F \in \mathcal{F}.$$

Доказательство.

Пусть $L = M = \log n$ и $\varepsilon = (\log n)^{-1/9}$. Ясно, что при достаточно большом n выполняется условие (2). Поэтому мы можем применить лемму 4 с данными значениями L, M и ε.

Зафиксируем для каждого множества $C \in \mathrm{PF}(G)$ множество C'. Пусть $\mathcal{F}$ является семейством множеств $C \cup C'$ для всех $C \in \mathrm{PF}(G)$. Тогда ясно, что выполняется второе свойство. Несложно показать, что свойство 3 также выполнено. Действительно, добавление нового элемента $x \in G$ к некоторому множеству B не создает более чем $3n$ новых троек Шура, поэтому любое $F \in \mathcal{F}$ содержит не более чем εn^2 троек Шура. Из лемм [5, Green-Ruzsa, Lemma 15] и [6, Lev-Luczak-Schoen, Lemma 6] следует, что

$$\log |\mathcal{F}| \le \frac{2n}{L} + \sqrt{\alpha} n \le n\alpha = n(\log n)^{-1/18}.$$

□

Доказательство теоремы 1. Рассмотрим семейство $\mathcal{F}$ почти МСП множеств построенных при доказательстве теоремы 2. Поскольку любое $F \in \mathcal{F}$ порождает не более чем $n^2(\log n)^{-1/9}$ троек Шура, то из следствия 3 вытекает, что число элементов каждого такого множества не превышает

$$\lambda(G) + O\big(n(\log n)^{-1/45}\big).$$

Отсюда следует

$$|\mathrm{PF}(G)| = 2^{\lambda(G) + O\left(n(\log n)^{-1/45}\right)},$$

так как $\log |\mathcal{F}| \le n(\log n)^{-1/18}$ и любое $C \in \mathrm{PF}(G)$ содержится в некотором $F \in \mathcal{F}$. □

Настоящая работа выполнена при поддержке РФФИ (проект 04-01-00359).

СПИСОК ЛИТЕРАТУРЫ

1. *А.И. Кострикин* Введение в алгебру. – М.: Изд-во "Наука 1977.
2. *Т.Г. Петросян* О числе множеств, свободных от произведений, в группах четного порядка, Дискретная математика, **17** (2005), No. 1, 89-101.
3. *Т.Г. Петросян* О числе множеств, свободных от произведений, Дискретная математика, **21** (2006), No. 1.
4. *Alon N.* Independent sets in regular graphs and sum-free subsets of finite groups, Israel Journal of Math., **73** (1991), 2, 247-256.
5. *Green, B.; Ruzsa, I.Z.* Sum-free sets in abelian groups, Israel J. Math. **147** (2005), 157-189.
6. *Lev, V.F.; Luczak, T.; Schoen, T.* Sum-free sets in abelian groups, Israel J. Math. **125** (2001), 347-367.
7. *Lubotzky, A.; Segal, D.* Subgroup Growth, Birkh ä user, Basel, 2003.

СБОРНИК СТАТЕЙ МОЛОДЫХ УЧЕНЫХ факультета ВМиК МГУ, 2007, выпуск № 4, с. 136–147

УДК 518.683.8

ОБ ОДНОМ ПОДХОДЕ К ПОСТРОЕНИЮ УНИВЕРСАЛЬНЫХ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

© 2007 г. А. В. Столяров

avst@cs.msu.ru

Кафедра Алгоритмических языков

Введение. Многообразие существующих языков программирования естественным образом приводит к возникновению вопроса о возможности построения *универсального* языка, т.е. такого языка программирования, который подходил бы для реализации любого проекта, не уступая при этом другим языкам ни по каким параметрам (т.е. не оставляя технических причин для предпочтения другого языка).

Иначе говоря, вопрос в том, можно ли создать такой язык программирования, чтобы для программиста, знающего этот язык и любое количество других языков программирования, при решении любой возникающей программистской задачи выбор языка программирования был бы очевиден и всегда оказывался в пользу универсального языка.

Языки программирования традиционно делились на высокоуровневые и низкоуровневые. Следует заметить, что смысл этих терминов с течением времени менялся. Если изначально под языком низкого уровня понимался исключительно язык ассемблера, а все остальные языки программирования считались языками высокого уровня, причем о сравнении относительной ≪высоты уровня≫ двух высокоуровневых языков речь не шла, то со временем концепция уровня языка стала более гибкой. С одной стороны, этому способствовало появление языка программирования C, который, не являясь языком ассемблера, позволял, тем не менее, описывать программу в терминах, близких к машинным. С другой стороны, с появлением всё большего количества языков программирования становилась более ясной недостаточная выразительность термина ≪язык высокого уровня≫, т.к. языки, являющиеся таковыми, оказывались очевидно по-разному удалены от машинного языка и возможностей машины; так, язык Pascal имеет с возможностями машины гораздо больше общего, чем, скажем, язык Prolog или язык Lisp. Это видно хотя бы из того факта, что динамические структуры данных в языках Lisp и Prolog входят в число первичных понятий языка, тогда как в языке Pascal их следует строить в явном виде; также языки Lisp и Prolog имеют механизмы автоматической сборки мусора, каковые в низкоуровневых терминах реализуются достаточно сложным образом; наконец, и сами модели вычисления, т.е. собственно работы программы, в этих языках не имеют ничего общего с низкоуровневыми вызовами функций и машинными командами, тогда как в языке Pascal достаточно очевидны правила, по которым конструкциям языка ставятся в соответствие фрагменты машинного кода.

Наконец, в последние 10–15 лет из программистской практики оказался почти полностью вытеснен язык ассемблера, вместо которого чаще всего используется язык C. Следует заметить, что в задачах, ранее решавшихся на ассемблере, обычно практически невозможно использовать большинство языков высокого уровня; C в этом смысле явно отличается от многих других языков.

Все это сделало термин ≪язык высокого уровня≫ неадекватным для классификации языков программирования. В результате концепция уровня языка трансформировалась из бинарной (низкий–высокий) в относительную (язык L_1 имеет уровень более высокий, нежели язык L_2).

Уровень языка является важным, а в некоторых случаях и определяющим критерием при выборе. Так, существуют проблемные области, в которых применение языков достаточно высокого уровня (в частности, языков, имеющих встроенный механизм сборки мусора) оказывается

невозможным. К таким проблемным областям относятся, в частности, создание систем реального времени, разработка ядер операционных систем, программирование микроконтроллеров и т.п.

В то же время существуют и такие проблемные области, в которых оказывается неоправданным (хотя и возможным) применение языков программирования сравнительно низкого уровня. К таковым относятся самые разнообразные области, от задач организации документооборота на предприятии до задач искусственного интеллекта.

При обсуждении достоинств и недостатков тех или иных языков программирования часто упоминаются *парадигмы программирования*. Отметим, что сам термин ≪парадигма программирования≫ нельзя считать устоявшимся; разные авторы вкладывают в этот термин весьма различный смысл.

Можно заметить, что парадигмы программирования обычно не являются принадлежностью конкретного языка. Рассмотрим, к примеру, парадигму функционального программирования, т.е. парадигму, в рамках которой программа воспринимается как набор взаимозависимых функций, не имеющих побочных эффектов, а выполнение программы представляется как вычисление некоторой функции при заданном значении аргументов.

Некоторые языки (например, Hope [3]) требуют работы именно в таких рамках. Выход за рамки функционального программирования в языке Hope невозможен.

Другие языки, такие как Lisp или Haskell, стимулируют применение функционального программирования, однако допускают и отступления от него в виде побочных эффектов функций, применения императивных конструкций и т.п.

Если рассмотреть такие языки, как C или Pascal, можно заметить, что эти языки *допускают* применение функционального программирования, поскольку, вообще говоря, никто не запрещает в этих языках писать функции без побочных эффектов.

Наконец, язык Фортран делает применение функционального программирования невозможным.

Вообще, некоторый язык программирования может находиться с определённой парадигмой в одном из следующих вариантов взаимоотношений:

1. язык **навязывает** примененение парадигмы. Программирование на данном языке без применения данной парадигмы категорически невозможно. Примеры: язык Smalltalk и объектно-ориентированное программирование; язык Fortran и присваивания.

2. язык **понуждает** к применению парадигмы. Программирование на данном языке без применения данной парадигмы возможно, но очень неудобно. Примеры: язык Lisp и рекурсия; язык Pascal и присваивания.

3. язык **поощряет** применение парадигмы. Программирование на данном языке без применения данной парадигмы возможно и достаточно удобно, однако при освоении данной парадигмы программист получает вознаграждение в виде резко возрастающего удобства работы. Примеры: язык C++ и механизм исключений; язык Lisp и функции высокого порядка (функционалы).

4. язык **поддерживает** применение парадигмы. Язык включает в себя специальные средства для применения данной парадигмы, расчитанные на программистов, привыкших к её применению, однако допускает и другие, в некоторых случаях более удобные варианты решения аналогичных задач. Примеры: язык C++ и макропроцессирование; язык Lisp и циклы; язык C и рекурсия.

5. язык **допускает** применение парадигмы. Язык не включает никакой специальной поддержки для данной парадигмы, однако при определенных навыках программист все еще может ее применять. Примеры: язык Pascal и функциональное программирование; язык C и обобщенное программирование; язык Pascal и объектно-ориентированное программирование.

6. язык **препятствует** применению парадигмы. Применение данной парадигмы в данном языке теоретически возможно, однако связано с затратами, делающими её применение неоправданным. Примеры: язык C и обработка исключений (возможно с помощью `setjmp()`/`longjmp()`, но требует серьезных трудозатрат); язык Pascal и виртуальные функции.

7. язык **запрещает** применение парадигмы. В данном языке недостаточно средств для применения данной парадигмы. Примеры: Fortran и функциональное программирование; Hope и императивное программирование; Bourne Shell и объектно-ориентированное программирование.

Следует особо отметить, что определяющим фактором при обсуждении языков программирования и парадигм оказывается не техническая сторона проблемы, а мышление программиста, т.е. психологическая составляющая.

Постановка задачи. Сформулируем коротко требования, без выполнения которых язык заведомо не может претендовать на статус универсального.

- Язык должен быть полностью компилируемым. Желательно, чтобы минимальный возможный для данного языка объем библиотеки времени выполнения был строго равен нулю (именно так обстоят дела для ассемблеров).

- Язык должен быть пригоден для низкоуровневого программирования. Это означает, что

  – не должно быть таких возможностей базового вычислителя (аппаратного обеспечения), которые программа на данном языке не могла бы использовать и

  – ядро языка должно включать в себя только такие средства, реализация которых очевидна и не вызывает сомнений; более сложные средства, допускающие разнообразные способы реализации, должны быть вынесены в библиотеку с тем, чтобы позволить программисту использовать собственную реализацию.

- Язык должен как минимум *допускать* (в смысле, приведённом в предыдущем параграфе) применение всех парадигм программирования, известных на момент создания языка; с другой стороны, язык не должен по возможности *навязывать* использование тех или иных парадигм.

- Язык должен включать средства генерации абстракций достаточно высокого уровня, чтобы удовлетворить запросы программистов-практиков (в идеале должна быть предоставлена возможность генерации абстракций сколь угодно высокого уровня).

Поясним сказанное. Интерпретируемое исполнение недопустимо при программировании микроконтроллеров, в системах реального времени и некоторых других областях. Если язык непригоден к низкоуровневому программированию, нам приходится исключить его из рассмотрения при реализации таких проектов, как ядра операционных систем, системы реального времени и т.п. Если язык включает средства, допускающие различные реализации, это практически заведомо означает, что в том или ином конкретном проекте реализация, предложенная разработчиками компилятора, окажется непригодной.

В случае, если язык не допускает применение какой-либо парадигмы программирования, возможно, что именно эта парадигма будет признана наиболее удобной при реализации конкретной задачи и язык окажется отвергнут.

Наконец, при ограниченных возможностях генерации абстракций высокого уровня реализация той или иной задачи на данном языке может оказаться заведомо многократно более трудоемкой, нежели на языке более высокого уровня.

Отметим, что в настоящее время автору не известен ни один язык, удовлетворяющий всем перечисленным требованиям одновременно. Из известных языков ближе всех к свойству универсальности подошел язык C++, однако включение в этот язык встроенных средств идентификации типов во время исполнения (RTTI, runtime type identification), множественного

наследования в общем виде, а также сложных и неочевидных по способу реализации средств обработки исключительных ситуаций сделали этот язык слишком высокоуровневым. Довершило дело введение в стандарт языка библиотеки шаблонных классов STL, которая современными программистами зачастую воспринимается как часть языка (хотя и не является таковой). В итоге разработчики операционных систем от языка C++ отвернулись; в современном сообществе преобладает восприятие C++ как одного из многих языков высокого уровня.

Метод непосредственной интеграции. Метод непосредственной интеграции впервые предложен в статье [1] в качестве подхода к интеграции разнородных языковых изобразительных средств в одном проекте. Суть метода в том, что синтаксические возможности базового языка (такого как C++) используются для построения выражений, синтаксически близких конструкциям языка альтернативного, такого как Lisp. В применении к языкам Lisp и C++ метод реализован в библиотеке InteLib. Библиотека позволяет, например, записывать такие выражения:

```
(L|DEFUN, ISOMORPHIC, (L|TREE1, TREE2),
  (L|COND,
    (L|(L|ATOM, TREE1), (L|ATOM, TREE2)),
      (L|(L|ATOM, TREE2), NIL),
      (L|T, (L|AND,
        (L|ISOMORPHIC,
          (L|CAR, TREE1),
          (L|CAR, TREE2)),
        (L|ISOMORPHIC,
          (L|CDR, TREE1),
          (L|CDR, TREE2)))))
).Evaluate();
```

причем семантика такого выражения полностью соответствует семантике аналогичного текста, записанного на языке Lisp:

```
(defun isomorphic (tree1 tree2)
 (cond
   ((atom tree1) (atom tree2))
   ((atom tree2) NIL)
   (t (and
     (isomorphic
       (car tree1)
       (car tree2))
     (isomorphic
       (cdr tree1)
       (cdr tree2))))))
```

Такая возможность достигается за счет переопределения символов стандартных операций (в данном случае – запятой и вертикальной черты). Важно подчеркнуть, что с точки зрения компилятора языка C++ вышеприведённый фрагмент представляет собой обычное арифметическое выражение.

Приведённый пример показывает, насколько существенно можно расширить изобразительную мощь языка программирования (в данном случае C++) за счет использования перегрузки символов стандартных операций.

Вместе с тем, C++ исходно не предназначался для подобных применений. Возможность перекрытия операции «запятая», по словам автора языка C++ Б. Страуструпа, в его исходные намерения не входила вовсе и никакого смысла в такой возможности он не видит [4].

Из сказанного можно сделать один чрезвычайно важный вывод: **если построить язык программирования низкого уровня, имеющий при этом средства для описания абстрактных типов данных и предоставляющий возможность перекрытия символов**

операций, то при определённом уровне гибкости этой возможности можно написать библиотеки, моделирующие вычислительные модели языков-носителей альтернативных парадигм программирования, в результате чего задача построения универсального языка программирования будет решена.

Ключевыми здесь являются требования о низкоуровневой сущности гипотетического языка и о высоком уровне гибкости средств построения абстракций высокого уровня.

Попытаемся сформулировать свойства языка, который предполагается создать в качестве универсального.

Наследие языка C. Прежде всего, сам по себе этот язык необходимо сделать языком низкого уровня, подобно ANSI C. Следует отметить, что даже язык C в той его версии, которая описывается стандартом C99, является языком чрезмерно высокого уровня. Так, например, C99 допускает следующий код:

```
void f(int n) {
    int a[n];
    int b[2*n];
    int c[3*n];
    /* ... */
}
```

Если при отсутствии возможности задания размерностей массивов подобным образом всегда можно было сказать, что любое имя локальной переменной представляет собой (с низкоуровневой точки зрения) ни что иное как константное смещение относительно базы стекового фрейма, то в приведённом выше примере имя `c` — это сложное адресное выражение, зависящее от переменной `n`, и способов реализации такого кода возможно несколько, причём найти среди них наиболее оптимальный не представляется возможным.

Представляется целесообразным во избежание появления лишних проблем сделать новый язык C-подобным, но лишь до определённого предела. В силу некоторых причин, которые будут рассмотрены позднее, совместимость с языком C на уровне синтаксиса невозможно сохранить без серьёзных компромиссов на пути достижения основных задач.

С другой стороны, задача создания стандартной библиотеки является крайне трудоёмкой и бессмысленной, поскольку для языка C она уже решена. В связи с этим представляется целесообразным, убрав в языке некоторые синтаксические несообразности языка C, тем не менее сохранить модель вызовов языка C, а также его систему типов; это позволит воспользоваться функциями стандартной библиотеки языка C.

Одним из недостатков синтаксиса языка C является чрезмерная перегруженность символа ≪запятая≫. Этот символ обозначает арифметическую операцию, однако он же используется для разделения формальных параметров в заголовках функций, фактических аргументов в вызовах функций, имён переменных в описаниях, элементов в сложных инициализаторах и констант в описаниях перечислимых типов. Если в каком-либо из перечисленных контекстов символ запятой требуется использовать для обозначения операции, приходится использовать лишние скобки.

Заметим, что во всех случаях кроме описания переменных запятую можно заменить на точку с запятой, сняв, таким образом, проблему. Что касается описаний вида

```
int a, b, c;
```

то их представляется целесообразным попросту запретить. Заметим, что действующий в языке C совершенно аналогичный запрет на описание нескольких формальных параметров одного типа в заголовке функции никому не мешает. В то же время, требование вместо вышеприведённого описания использовать

```
int a;
int b;
int c;
```

позволяет устранить ещё одну проблему, которую сторонники языка C попросту не замечают, но которая часто возникает у начинающих. Допустим, требуется описать два указателя на `int`. В языке C это можно сделать, например, так:

```
int *p, *q;
```

Для человека, плохо знакомого со спецификой языка C, такая конструкция выглядит нелогично, поскольку символ `*` имеет в данном случае отношение к типу, а не к переменной. Начинающие часто предпочитают записывать объявление указателя, ставя пробел после символа `*`, а не до него, как опытные программисты на C:

```
int* p;
```

Отметим, что язык C вполне допускает такое описание. Усложнив его, получим

```
int* p, q;
```

Для начинающего программиста естественно восприятие, при котором переменная `q` должна иметь тип «указатель на `int`», тогда как на самом деле эта переменная в данном случае оказывается типа `int`. Запрет описания нескольких переменных одного типа без указания типа решает эту проблему.

Язык и библиотеки. Одним из важнейших условий, необходимых для низкоуровнего программирования, является четкое отделение библиотеки (сколь угодно стандартной) от собственно языка. В частности, компилятор низкоуровневого языка не вправе ничего знать об именах (функций, переменных и т.п.), описываемых в библиотеке. В противном случае в ситуациях, когда использование (стандартной) библиотеки по тем или иным причинам нецелесообразно, язык и его компилятор оказываются неполноценны. Между тем, такие ситуации возникают гораздо чаще, чем можно подумать: стандартная библиотека того же языка C, несмотря на её логичность и высокое качество проектирования, тем не менее не используется при написании ядер операционных систем, при программировании всевозможных микроконтроллеров и т.п.

Отметим, что принцип разграничения языка и библиотеки часто нарушается даже для языка C: в частности, пресловутый стандарт C99 указывает, что операция `sizeof`, встроенная в язык, должна возвращать значение типа `size_t`, при том что этот тип в язык не встроен и должен описываться в библиотеке.

В качестве второго условия следует назвать следующее правило: всё, что *может быть* вытеснено из языка в библиотеку, *должно быть* вытеснено в библиотеку. В крайнем случае, если в языке необходим некий механизм, то лучше предусмотреть в языке не сам этот механизм, а средства, позволяющие его описать. Так, в языках высокого уровня обычно разрешается «складывать» (конкатенировать) строки с помощью операции `+` (это так, например, для языков Pascal, Java и т.п.). В этом плане язык C++ следует признать более удачным, т.к. он, не предоставляя подобных средств в самом языке, при этом позволяет переопределить операцию `+` для произвольных пользовательских типов, что, в свою очередь, позволяет описывать строки, которые можно «складывать», но не только их: с неменьшим успехом операция `+` используется для сложения комплексных чисел, матриц или, например, полиномов, если таковые описаны в виде классов.

Третье условие касается скорее не самого разрабатываемого языка, а предполагаемой политики стандартизации. Как показывает пример языка C++, стандартизация библиотеки в составе стандарта языка может нанести непоправимый урон культуре программирования на этом языке, а также сообществу программистов, использующих данный язык. Присвоение библиотеке STL статуса составной части стандарта C++ поставило другие библиотеки аналогичного назначения в негативные условия, и, кроме того, для подавляющего большинства программистов, использующих C++, введение STL в стандарт послужило сигналом к тому, что при работе на языке C++ следует *всегда* использовать STL; между тем, использование STL резко снижает читаемость кода, многократно усложняет отладку и сопровождение программ,

давая при этом весьма сомнительный выигрыш на стадии кодирования. Обучение начинающих программистов с использованием STL приводит к тому, что студенты, завершившие обучение, попросту не понимают принципиальных отличий C++ от других языков и не представляют программирование на C++ иначе как с использованием STL.

Вместе с тем, стандартизация интерфейсов библиотек имеет ряд несомненных достоинств, прежде всего — повышение переносимости программ.

В связи с этим представляется целесообразным проводить стандартизацию библиотек *без привязки к стандарту языка*, не давая при этом никаким библиотекам статуса «стандартной библиотеки языка». Это позволит, с одной стороны, использовать положительные стороны стандартизации интерфейсов библиотек, и, с другой стороны, избежать катастрофических последствий стандартизации одной конкретной библиотеки.

Вместе с языком можно (хотя и не обязательно) стандартизовать те и только те элементы библиотеки, которые не могут быть в силу тех или иных причин переносимым образом реализованы средствами самого языка. Примером таких элементов является интерфейс для доступа к системным вызовам операционной системы.

Арифметические выражения как основа синтаксической гибкости. Чтобы сделать возможным библиотечное моделирование альтернативных вычислительных моделей (парадигм), язык должен обладать определённой синтаксической гибкостью. В то же время необходимо сохранять эту гибкость в определённых рамках: попытки создания языков программирования с полностью программируемым синтаксисом в истории известны и к положительным результатам не приводили.

Как показывает пример проекта InteLib, в действительности для моделирования изобразительных возможностей многих альтернативных языков программирования достаточно выразительной мощности арифметических выражений, при условии, что в базовом языке предусмотрен достаточно широкий спектр арифметических операций и имеются возможности их переопределения для введённых пользователем типов.

Отметим, что слова «достаточно широкий» в предыдущем абзаце могут быть истолкованы весьма по-разному в зависимости от конкретной предметной области. Так, для моделирования выражений языка Lisp оказалось достаточно операций, имеющихся в языке C++; более сложный язык, такой как Haskell, столь удачно в имеющемся наборе операций представить не удастся.

Операция «пробел». Коль скоро символам инфиксных операций выделяется столь важная роль, целесообразно будет сделать набор таких операций возможно более широким. В частности, введение «операции пробел» (то есть соглашения, при котором два выражения произвольных типов, стоящие друг за другом и не разделённые знаком операции, считаются операндами бинарной операции, называемой «операция пробел») способно резко повысить выразительную мощность арифметических выражений.

В частности, при наличии операции «пробел» обозначить применение математического оператора к его аргументу можно будет способом, напоминающим традиционный для математики: `D f`.

Приоритет такой операции следует, видимо, сделать ниже, чем приоритеты всех имеющихся в языке арифметических операций. Действительно, выражение `a+b c+d` воспринимается скорее как два выражения сложения, записанные одно за другим, нежели как сумма из трёх элементов, вторым из которых является вызов операции «пробел», в особенности если вместо пробела использовать, скажем, символ перевода строки. То же самое можно сказать и обо всех остальных операциях, исключая разве что операцию «запятая». Вопрос о соотношении приоритетов операций «запятая» и «пробел» оставим пока открытым.

Также открытым оставим и вопрос об ассоциативности операции «пробел», то есть о том, какой из её аргументов вычисляется первым, левый или правый.

Сочетание операции «пробел» с операцией «вызов функции». Действие «вызов функции» в языках C и C++ является операцией, обозначаемой как выражение, первый

операнд которого представляет собой выражение типа ≪адрес функции≫ (имя функции является примером такого выражения), а за первым операндом следуют круглые скобки, воспринимаемые как символ операции; внутри круглых скобок перечисляются фактические параметры вызова.

Круглые скобки, таким образом, играют в языке две совершенно различные роли. С одной стороны, они используются для группировки подвыражений в выражениях с целью изменения порядка применения операций. С другой стороны, круглые скобки обозначают операцию вызова функции.

В языках C и C++ это не создаёт никаких проблем, поскольку синтаксически эти ситуации полностью разнесены: если перед круглой скобкой находится законченное выражение, то скобка обозначает вызов функции, тогда как если перед скобкой никакого выражения нет либо выражение не закончено и необходим ещё один операнд (то есть скобка находится в позиции, где ожидается выражение), то скобка воспринимается как группирующий символ.

При введении в язык операции ≪пробел≫ ситуация несколько изменяется. Для случая вызова функции от нуля аргументов проблем не возникает, поскольку при использовании скобок в качестве группирующих символов закрывающая скобка не может оказаться сразу за открывающей. При условии использования для разделения аргументов символа “`;`”, а не запятой (вообще говоря, попросту символа, не являющегося изображением какой-либо операции), проблем не возникает также и в случае вызова функции от двух и более аргументов (вызов отличается от группировки по наличию символа, разделяющего аргументы).

Что касается случая вызова функции от ровно одного параметра, то выражение вида `a(b)` оказывается имеющим два различных трактования: как операция вызова функции `a` от аргумента `b`, либо как вызов операции ≪пробел≫ для аргументов `a` и `b` (в этом случае аргумент `b` представляется заключённым в круглые скобки с целью группировки, что вполне имеет смысл, скажем, если `b` представляет собой, в свою очередь, выражение, с операцией, имеющей более низкий приоритет, чем вызов функции).

Возможно выбрать одно из этих толкований на основе контекстных условий и дополнительных соглашений. Подобно этому в языке C++ инициализация контруктором по умолчанию, применённая при описании переменной, оказалась бы неотличима от описания функции от нуля аргументов, если бы только для этого применялись общие синтаксические правила:

```
A a(25,26);// объект класса A,
           // конструктор от двух аргументов
A a2(77);  // объект класса A,
           // конструктор от одного аргумента
A a3();    // прототип функции от 0 аргументов,
           // которая возвращает объект класса A
A a4;      // объект класса A, использован
           // конструктор от нуля аргументов
```

Заметим, что описание отдельно стоящей переменной — это единственный случай, когда конструктор от нуля аргументов вызывается без круглых скобок. Так, в выражении

```
( A(27, 44) + A() )
```

имеет место создание двух анонимных объектов класса `A`, первый из которых создаётся конструктором от двух аргументов, второй — конструктором от нуля аргументов. Заметим, что скобки в данном случае на месте, а выражение

```
( A(27, 44) + A )
```

является синтаксической ошибкой.

Следует отметить, что решение, применённое в C++, нарушает концептуальную целостность синтаксиса языка (т.к. вводит особый случай), обладает определённой неочевидностью и затрудняет восприятие синтаксиса, в особенности для начинающих программистов. В связи с этим для решения проблемы неоднозначности между вызовом функции от одного аргумента

и вызовом операции ≪пробел≫ предлагается несколько иное решение, основанное на общем правиле без особых случаев и обладающее дополнительными полезными свойствами.

Кортежи. Под кортежем будем понимать синтаксический конструкт языка, состоящий из нескольких выражений произвольных типов, разделённых символом ≪точка с запятой≫ и заключенных в круглые скобки, например:

```
(25; "a string"; x+y)
```

Выделим два особых случая, а именно, кортеж, состоящий из нуля элементов и обозначаемый `()`, а также кортеж из одного элемента, который положим семантически эквивалентным самому этому элементу (таким образом, произвольное выражение становится частным случаем кортежа, а именно, кортежем из одного элемента). Введём соглашение, что при записи кортежа из одного элемента скобки можно опустить; таким образом, например, выражения `25` и `(25)` останутся эквивалентными, как и до введения понятия кортежа.

Будем считать, что кортеж с указанием количества и типов параметров является, в свою очередь, типом данных с точки зрения профилей функций (то есть может, например, выступать в качестве параметра функции), но при этом не является, вообще говоря, структурой данных, то есть не может быть присвоен переменной (как не может и описываться переменная типа ≪кортеж≫). Вызов функции от кортежа из n элементов физически реализуется как вызов функции от n параметров.

Логично разрешить присваивание кортежа выражений кортежу переменных, например:

```
int x;
const char *p;
float f;
(x; p; f) = (25, "string", 3.7);
```

Это позволит пользоваться кортежами имён формальных параметров при описании и вызове функций от кортежей. Например:

```
void f((int a ; int b) ; int c)
{ /* ... */ }
                  /* ... */
f((2 ; 3) ; 4);
```

Наконец, определим понятие операции вызова функции как частный случай операции ≪пробел≫ от имени функции (или, если угодно, указателя на функцию) и кортежа параметров. Как следствие, вызов функции от одного параметра станет возможно записать без скобок, например, запись `sin(x)` окажется семантически эквивалентна записи `sin x`.

Подчеркнём ещё раз, что вызовы `f(1;2;3;4)` и `f((1;2);(3;4))` можно различать с точки зрения профилей функций (т.е. компилятор должен вызывать при этом разные функции), но с точки зрения реализации ничем, кроме адреса вызываемой функции, эти два вызова различаться не должны.

Введение новых символов инфиксных операций. Как показал пример библиотеки InteLib, имеющихся в языке символов инфиксных операций может и не хватить, либо может ощущаться их недостаток.

Известны языки программирования, в которых допускается введение новых (не предусмотренных в языке) символов инфиксных операций; так, это возможно в языке Prolog.

Вместе с тем, введение операции ≪пробел≫ даёт возможность моделировать новые символы инфиксных операций, вводя их как идентификаторы объектов, для которых переопределена операция ≪пробел≫: так, например, выражение `a in b` можно рассматривать не как операцию `in` от аргументов `a` и `b`, а как два вызова операции ≪пробел≫; в этом случае слово `in` должно быть именем переменной, имеющей некоторый пользовательский тип. В зависимости от ассоциативности операции ≪пробел≫ это будет, соответственно, выражение

```
operator space(a, operator space(in, b))
```

или

```
operator space(operator space(a, in), b)
```

Вместе с тем, такой способ моделирования новых инфиксных операций обладает сравнительно низкой гибкостью, т.к. ограничен фиксированным приоритетом и направлением ассоциативности операции ≪пробел≫. Кроме того, в качестве символов таких операций можно будет использовать исключительно идентификаторы языка.

Как уже говорилось, к гибкости синтаксиса языка следует относится с определённой осторожностью, поэтому следует оставить открытым вопрос о целесообразности введения в язык возможности определения новых символов операций, подобной имеющейся в языке Prolog. Вернуться к этому вопросу можно будет после того, как язык будет реализован и будет получен определённый опыт работы с ним.

Наследование. Наследование как таковое является важнейшим средством создания высокоуровневых абстракций, так что необходимость наличия наследования в языке как таковая несомненна. Вместе с тем, вопросы, связанные с множественным наследованием, оказываются достаточно дискуссионными; кроме того, механизм виртуальных функций оказывается достаточно нетривиален, чтобы его нельзя было считать средством низкого уровня.

В частности, реализация *множественного наследования* в общем виде, подобно тому, как это сделано в языке C++, несмотря на кажущуюся простоту, влечёт массу нетривиальных проблем.

Действительно, пока все базовые классы неполиморфны (то есть не содержат виртуальных функций) и не являются сами чьими-то наследниками, реализация множественного наследования остаётся достаточно простой. Единственная привносимая проблема — необходимость пересчёта численного значения указателя или ссылки при преобразованиях от типа ≪потомок≫ к типу ≪предок≫ (и обратно), если соответствующий предок не является первым в списке базовых классов.

Появление у базовых классов общих предков влечёт появление понятия виртуального класса и множества сомнительных ситуаций, как в случае, если один и тот же класс является виртуальным предком части базовых классов и невиртуальным предком другой части базовых классов.

Появление в двух и более базовых классах таблиц виртуальных функций приводит к резкому усложнению реализации самого механизма виртуальных функций: для каждой виртуальной функции теперь, помимо адреса, требуются ещё, как минимум, смещения для пересчёта указателя `this` и положения указателя на таблицу виртуальных методов, так что вместо одного поля (адреса функции) строка таблицы виртуальных методов оказывается состоящей из трёх полей.

В этой связи представляется разумным компромисс, принятый в языке Java: среди базовых классов может быть не более одного класса, не являющегося *интерфейсом* (то есть классом, состоящим исключительно из чисто виртуальных функций).

В то же время статические преобразования указателей сами по себе не сложны в реализации и вполне укладываются в представления о низкоуровневом программировании, так что (при условии успешного вытеснения механизмов виртуализации в библиотеку) в язык без ущерба его низкоуровневости можно внести и более общие виды множественного наследования. Конечно, при этом следует постулировать, что наследование от нескольких классов, имеющих общий базовый класс, *всегда* приводит к наличию в объекте соответствующего количества экземпляров этого базового класса, поскольку имеющийся в языке C++ механизм виртуальных базовых классов заведомо чрезмерно сложен для низкоуровневого языка.

Библиотечная поддержка виртуальных функций. Как отмечалось выше, механизм виртуальных функций (в особенности в ситуации множественного наследования) оказывается чрезмерно сложен, чтобы считаться средством низкого уровня.

С другой стороны, библиотечная реализация виртуальных вызовов практически неизбежно потянет за собой необходимость достаточно развитых макросредств.

Само по себе это, возможно, и не столь плохо. Дело в том, что большинство возражений против применения макропроцессора, высказываемых относительно языков C и C++, основаны на том, что макроимена не подчиняются обычным для языка соглашениям об именах. Вместе с тем, язык C++ вводит т.н. шаблоны, которые отличаются от макросов практически лишь в том, что являются полноценной частью языка, подчиняются всем законам локализации имён и поддерживают проверку типов параметров (ограничение на возможные типы параметров, присутствующее в C++, выглядит скорее реализаторским, чем концептуальным).

Можно привести и другие примеры макросистем, никоим образом не нарушающих концептуальную целостность языка и не создающих никаких проблем. Такова, например, система макросредств в языке Common Lisp.

Отметим еще один момент, непосредственно связанный с виртуальными вызовами. Вообще говоря, виртуальный вызов представляет собой действие совершенно иное, нежели обыкновенный вызов. Возможно, имеет смысл зарезервировать для этого действия отдельные синтаксические соглашения (например, ввести тот или иной символ операции).

При наличии в языке макроподобных механизмов, допускающих перегрузку по типам параметров, возможно также ввести и описание инфиксных операций не (или не только) в виде функций, как это сделано в C++, но и в виде макросов (так, в C++ пришлось вводить весьма нетривиальные правила переопределения операции `->` как одноместной; определение её как макроса окажется заведомо проще для восприятия, поскольку такой макрос можно сделать двуместным). Введя возможность описания инфиксной операции в виде макроса, мы получаем возможность реализовать в виде макросов всё связанное с механизмом виртуальных вызовов, вытеснив, таким образом, эти (заведомо высокоуровневые) средства в библиотеку.

Обработка исключений. Обработка исключительных ситуаций представляет собой крайне важный механизм, использование которого повышает читабельность и надёжность программного кода и существенно снижает трудоёмкость программирования. В особенности это средство полезно в сочетании с автоматическим вызовом деструкторов локальных объектов, как это сделано в языке C++.

С другой стороны, обработка исключений в том виде, в котором она присутствует в современных языках программирования (в том числе и в C++), является средством заведомо высокого уровня. Реализация исключений неочевидна и непрозрачна.

Проблема вытеснения механизмов обработки исключительных ситуаций из ядра языка в библиотеку достаточно интересна. Ключ к решению видится в выработке удобного и логичного интерфейса к явной манипуляции стековыми фреймами; отметим, что это средство должно быть частью языка, т.к. должно позволять, например, при принудительном уничтожении стекового фрейма отработать деструкторы уже сконструированных объектов. Таким образом, механизма, подобного библиотечным функциям `setjmp()` и `longjmp()`, для этого не достаточно.

Вполне возможно, что соответствующее средство языка может быть похожим на «альтернативное тело», подобное блокам `catch` в языке C++; вместо `try`-блока следует, по-видимому, предусмотреть конструкцию, семантика которой сводится к «пометке фрейма» (для такой пометки можно использовать, например, нетипизированный указатель), и операцию «раскрутки стека на один фрейм», которая возвращала бы для помеченных фреймов значение метки, а для фреймов, не помеченных меткой — нулевой указатель.

Наличие таких средств в языке позволит предоставить программисту выбор из нескольких библиотек, обслуживающих обработку исключительных ситуаций и предоставляющих сервис различного уровня сложности. Так, не во всяком проекте требуется возможность использования *произвольного* объекта в качестве «носителя» исключительной ситуации; в некоторых случаях можно обойтись простым кодом ошибки либо фиксированной структурой данных для хранения информации об ошибке, получив при этом выигрыш в эффективности.

Следует отметить, что автору известны случаи сознательного отказа от работы с исключениями, мотивируемые чрезмерной сложностью этого механизма в языке C++. Наличие про-

стой и прозрачной основы для такого механизма потенциально способно исправить ситуацию.

Заключение. Традиционное восприятие роли языка программирования в вычислительной системе, состоящей из аппаратуры, операционной системы и систем программирования, можно описать приблизительно следующим образом. Имеется некий конгломерат аппаратуры и операционной системы (либо даже попросту виртуальная машина), именуемый обычно *платформой* или *операционной средой*; в рамках этой среды имеется несколько *систем программирования*, каждая из которых построена вокруг того или иного языка программирования или даже некоторой конкретной его реализации. Среди имеющихся (доступных для данной платформы) систем программирования обычно для конкретного проекта выбирается какая-то одна (реже — несколько), причем выбор может быть обусловлен как поставленной задачей, так и личными предпочтениями разработчиков.

Разработка и реализация универсального языка программирования в соответствии с подходом, описываемом в настоящей статье, может несколько изменить представление о взаимоотношениях системы программирования и операционной платформы. Если сформулированные цели будут достигнуты, для заданной платформы может оказаться разумным поддерживать *один* (универсальный) язык программирования, снабженный при этом широкой коллекцией библиотек. Именно многообразие этих библиотек и займёт в этом случае нишу нынешнего многообразия языков программирования; от особенностей задачи и личных предпочтений программистов будет тогда зависеть не выбор языка, а выбор набора библиотек.

Необходимо признать, что две программы, написанные на одном и том же (универсальном) языке, но с использованием принципиально различных библиотек, могут различаться по стилю практически столь же сильно, как сейчас различаются программы, написанные на разных языках программирования. Тем не менее, использование именно одного языка с множеством библиотек вместо традиционного множества языков имеет как минимум одно важное достоинство: трудности интеграции между собой фрагментов программы, написанных в различном стиле, окажутся раз и навсегда сняты именно в силу использования одного языка.

СПИСОК ЛИТЕРАТУРЫ

1. *E. Bolshakova and A. Stolyarov.* Building functional techniques into an object-oriented system. // Knowledge-Based Software Engineering. Proceedings of the 4th JCKBSE, volume 62 of Frontiers in Artificial Intelligence and Applications, pages 101–106, Brno, Czech Republic, September 2000. IOS Press, Amsterdam.
2. *И. Г. Головин, А. В. Столяров.* Объектно-ориентированный подход к мультипарадигмальному программированию. // Вестник МГУ, сер. 15 (ВМиК), №1, 2002 г., стр. 46–50.
3. *A. J. Field and P. G. Harrison.* Functional Programming. Addison-Wesley, Reading, Massachusets, 1988. *Русский перевод: А.Филд, П.Харрисон. Функциональное программирование. М.:Мир, 1993.*
4. *B. Stroustrup.* The design and evolution of C++. Addison-Wesley, Reading, Massachusets, 1994. *Русский перевод: Бьерн Страуструп, Дизайн и эволюция языка C++, М.: ДМК, 2000.*

СБОРНИК СТАТЕЙ МОЛОДЫХ УЧЕНЫХ факультета ВМиК МГУ, 2007, выпуск № 4, с. 148–152

УДК 519.714; 519.68: 681.513.7

ОПРЕДЕЛЕНИЕ ОБЛАСТЕЙ КОМПЕТЕНТНОСТИ АЛГОРИТМОВ СИНТЕЗА КОМБИНАЦИОННО-ЛОГИЧЕСКИХ СХЕМ. ГИПОТЕЗА КОМПАКТНОСТИ.

© 2007 г. И. Н. Фатхутдинов

ild_ar@mail.ru

Кафедра математических методов прогнозирования

Введение. На сегодняшний день существуют значительное количество алгоритмов логического синтеза комбинационных логических схем реализованных в составе ряда САПР и основанных на тех или иных точных либо эвристических методах. Однако, в связи с резким ужесточением параметров вновь проектируемых схем (до 1 млрд. ключей на кристалле, размер ключа менее 50 нм) алгоритмы синтеза перестают удовлетворять предъявляемым требованиям по качеству получаемой схемы. Поэтому на повестке дня стоит разработка принципиально новых типов указанных алгоритмов т.н. алгоритмов синтеза нового поколения.

Данные алгоритмы должны осуществлять синтез отдельных частей сложной логической схемы исходя из дифференцированных критериев. Как известно, различные синтезирующие алгоритмы демонстрируют различную эффективность на различных типах схем. Кроме того, данная эффективность может определяться исходя из ряда критериев (площадь, занимаемая схемой на кристалле, общая задержка по критическому пути, потребляемая мощность и т.д.). Синтезирующая система должна быть "чувствительна"к специфическим особенностям поведенческого описания тех или иных участков схемы с учётом их взаимосвязей с другими участками и общего задаваемого критерия эффективности.

Первым шагом на пути создания САПР с синтезирующими алгоритмами нового типа может быть разработка программной системы, определяющей по предъявленному входному описанию схемы некоторые характеристики и, в зависимости от них, принимающая решение, какому из имеющихся алгоритмов "поручить"синтез схемы. При разработке описанной системы возникают задачи построения признакового пространства описаний схем и разбиения его на части, принадлежность к которым определяет выбор синтезирующего алгоритма. В теории распознавания образов указанные части называют областями компетентности алгоритмов, в нашем случае - алгоритмов синтеза комбинационных логических схем.

В настоящей работе описан опыт построения программной системы, определяющей области компетентности алгоритмов синтеза комбинационных схем.

1. Постановка задачи.Под Θ будем понимать множество всевозможных описаний входных объектов в некоторой стандартной форме, а под Λ — конечное множество алгоритмов обработки указанных объектов. Пусть $S(A, \delta_0)$ — функционал качества полученного выходного объекта в результате обработки входного объекта $\delta_0 \in \Theta$ алгоритмом $A \in \Lambda$. Требуется построить пространство и разбить его на области компетентности алгоритмов A (из Λ).

Определение 1. Областью компетентности алгоритма будем называть подмножество пространства описания объектов (комбинационных логических схем, в нашем случае), для которых данный алгоритм оптимален по выбранному критерию.

Иными словами, нам нужно определить $A_0 = Arg\,min\, S(A, \delta_0)$ для произвольного объекта δ_0 из пространства Θ, где минимум берётся по всем $A \in \Lambda$.

В нашем случае Θ есть множество систем частичных булевых функций (СЧБФ) в формате системы синтеза схем SIS (см. 5), Λ — совокупность реализованных в SIS алгоритмов синтеза СЧБФ в виде схем (19 алгоритмов), а в качестве $S(A, \delta)$ взята количественная оценка площади синтезированной схемы.

2. Гипотеза компактности. Понятно, что алгоритм классификации не может оперировать с описанием схем. Ему необходимы количественные описания каких-либо характеристик. И с ними уже алгоритм будет работать. В этой работе эти характеристики указывались для каждого объекта (схемы) независимо от других объектов, т.е. каждому объекту приписывался некий вектор свойств. Т.о. нам необходимо пространство схем Θ отобразить в пространство свойств Ω. При этом далеко не любое отображение нам будет подходить. Подойдут лишь те, в которых образы схем будут легко разделятся на классы алгоритмом классификации. Под классом понимаем подмножество свойств объектов попадающих в одну область компетентности одного из алгоритмов минимизации. Проблема создания пространства свойств не решена в полной мере. Она решена лишь на уровне предписаний и рекомендаций.

При этом после выбора признакового пространства (или пространства свойств) предполагают, что верна гипотеза компактности, которая определяется: объектам одного класса в пространстве признаков соответствуют компактные сгустки, а разные классы достаточно хорошо разделяются. Ниже будет дано ее формальное описание.

Назовем n признаков, входящих в информативное множество признаков X, описывающими, а номинальный $(n+1)$-й признак z, указывающий имя образа, целевым. Обозначим некоторое множество объектов через A, произвольный объект через q, а тот факт, что объекты множества A компактны (эквивалентны, похожи или близки друг другу) в пространстве n характеристик X - через C_A^X. Мера компактности может быть любой: она может характеризоваться средним расстоянием от центра тяжести до всех точек образа; средней длиной ребра полного графа или ребра кратчайшего незамкнутого пути, соединяющего точки одного образа; максимальным расстоянием между двумя точками образа и т.д. Например, компактными (эквивалентными) считаем два объекта, если все признаки одного объекта равны соответствующим признакам другого. Или: объекты компактны, если евклидово расстояние между векторами их признаков не превышает величину r.

Фактически гипотеза компактности равнозначна предположению о наличии закономерной связи между признаками X и z, и с учетом вышесказанного ее тестовый алгоритм может быть представлен следующим выражением:

$$[C_\Omega^{X,z} \;\&\; C_{\Omega,q}^X \;\Rightarrow\; C_{\Omega,q}^z] \tag{1}$$

Т.е. если объекты класса Ω компактны в пространстве (X,z) и объекты класса (Ω,q) компактны в пространстве описывающих свойств X, то объекты Ω и q будут компактными и в пространстве целевого признака z.

Такого рода формализация уже позволяет оценить на сколько хорошее признаковое пространство мы выбрали: на какой доле обучающей выборки одного класса выполнено соотношение (1). С помощью этого правила так же можно оценивать представительность выборки. Ясно, что решение задачи возможно, если набор имеющихся прецедентов некоторым образом отражает объекты, которые реально будут встречаться при классификации произвольного объекта. Это предположение в теории распознавания именуется гипотезой представительности. Неформально она задается: обучающей информации достаточно для восстановления свойств классов с достаточной степенью точности.

3. Метод Бонгарда. В 60-х годах прошлого столетия Бонгардом был предложен эвристический метод формирования признакого пространства, когда не совсем ясно, что следует брать за признаки, свойства. Суть метода заключается в следующем: сначала тем или иным способом сформировать некоторый набор произвольных первичных признаков – числовых характеристик объектов. Эти признаки, вообще говоря, не позволяют успешно решить задачу классификации (разбиение признакового пространства). Затем с помощью некоторых функциональных преобразований из первичных признаков строятся всевозможные выражения, называемые в дальнейшем вторичными признаками. Для первоначального построения нет необходимости пользоваться сложными функциональными преобразованиями и при разработке системы были использованы (как и у М.Бонгарда) простейшие арифметические операции ($+$, $-$, $*$, $/$, min, max)

4. Математическая модель. Пусть имеется N алгоритмов минимизации управляющих систем $A_1, A_2, \ldots, A_N$, $\Re_i$, $i = 1, \ldots, N$ — области компетентности алгоритмов. Назовем первичным признаком произвольную функцию от табличного задания СЧБФ.

Пусть $P_1(\delta), P_2(\delta) \ldots, P_M(\delta)$ — первичных признаков конкретной СЧБФ δ. Вторичным признаком частично заданной булевой функции δ i-го алгоритма назовем произвольную функцию

$$f_{i,j}(P_1, P_2, \ldots, P_M) = f_{i,j}^{+,-,\times,/,min,max}(P_1(\delta), P_2(\delta), \ldots, P_M(\delta)), \quad i = 1, \ldots, N, \qquad (2)$$

где j — номер признака, построенную с помощью арифметических операций $+, -, \times, /, min, max$ из первичных признаков.

Область компетентности каждого алгоритма минимизации будем задавать системой неравенств

$$\begin{cases} f_{i,1}(P_1, P_2, \ldots, P_M) \leqslant Const_{i,1} \\ \ldots \\ f_{i,c_i}(P_1, P_2, \ldots, P_M) \leqslant Const_{i,c_i} \end{cases},$$

где константы $Const_{i,1}, \ldots, Const_{i,c_i}$ и операция сравнения «больше» или «меньше» выбираются из условия минимизации функционала качества вторичного признака (с привязанной к нему операцией сравнения и константы):

$$Q(F_{i,c_j}) = \sum_{\delta \in \Omega} [F_{i,c_j} - t]^2(\delta) \to min,$$

где F_{i,c_j} — индикатор принадлежности объекта δ классу с номером i, строящийся из $f_{i,k}$, операций $>$, $<$ и $Const_{i,k}$.

Вообще говоря, любая функция (с ограничениями на существование производных нужного порядка) разложима в ряд Тейлора, что говорит о том, что класс всех функций $\{f^{+,-,\times,/,min,max}\}$ достаточно широк.

Введем набор индикаторных функций

$$k_i(\delta) = \begin{cases} 1, \text{ если выполнена система неравенств;} \\ -1, \text{ иначе,} \end{cases}$$

и функцию штрафа

$$U_i(\delta) = \begin{cases} 1, \text{ если } \delta \in \Re_i; \\ -1, \text{ если } \delta \notin \Re_i. \end{cases}$$

Рассмотрим функцию

$$Z_i(\delta) = \frac{1 - K_i(\delta) \cdot U_i(\delta)}{2}.$$

Получается, что

$$Z_i(\delta) = \begin{cases} 0, \text{ если задает предикат } \delta \in \Re_i; \\ 1, \text{ иначе.} \end{cases}$$

И, наконец, рассмотрим функционал качества

$$\Phi_i = \Sigma_{\delta \in \Omega} Z_i(\delta), \quad i = 1, \ldots, N. \qquad (3)$$

Теперь задача построения классификатора свелась к задаче минимизации функционала качества для каждого алгоритма по набору вторичных признаков:

$$\Phi_i \to min \qquad (4)$$

для всех $i = 1, \ldots, N$. Заметим сразу, что поскольку Ω — потенциально бесконечное множество, то программно подсчитать (3) не представляется возможным. Потому на практике приходится ограничиваться предположительно достаточно представительным фиксированным подмножеством множества $\Re_i$, составляющее множество Ω прецедентов.

5. Программная реализация. Необходимая для решения задачи прецедентная информация была получена с помощью системы SIS (A System for Sequential Circuit Synthesis, система синтеза логических схем) разработанной в Калифорнийском университете (Беркли, США) как интерактивная система для оптимизации и синтеза логических схем [5]. Система разработана для использования в Unix-системах. SIS совместима по файлам с рядом практических систем логического синтеза БИС. В программный пакет SIS входят стандартные алгоритмы синтеза (их количество 19), которые стали объектом исследования в данной работе

Прецедентная информация (набор СЧБФ) представлялась в виде матриц входа и выхода, строящихся из элементов 0 , 1 и $-$. Матрица входа задает интервал булевых переменных, а матрица выхода - соответствующую интервалу, строку значений набора функций. В качестве прецедентной информации были взяты математические схемы, поставляемые вместе с пакетом SIS, и схемы сгенерированные случайным образом, с использованием нормального, гамма, экспоненциального, бета, пуассоновского, Коши, биномиального распределений. Были схемы сгенерированные с помощью только одного распределения, а так же схемы, которые делились на блоки, каждый из которых сгенерирован с помощью своего распределения из списка, перечисленного выше. Заметим, что самым важным моментом на этом этапе разработки было получение представительной (наиболее общей) выборки.

Первичные признаки строились следующим образом. В матрице входа и выхода конкретной СЧБФ выделяется 5 подобластей: матрица, "разрезанная" на четыре равные части (4 подобласти) и подматрица, образующая прямоугольник, накрывающий центр матрицы. Для каждой подобласти определялось (называемое в дальнейшем первичным признаком СЧБФ):

1. Процентное соотношение количеств символов 0, 1 и $-$ относительно количества элементов в подобластях (3 признака).

2. Среднестатистическое положение (математическое ожидание) количеств символов 0, 1 и $-$ в строках и столбцах каждой подобласти, нормализованное по длине строки и столбца соответственно (3×2 признака).

3. Среднее отклонение от среднестатистического положения (дисперсия) количеств символов 0, 1 и $-$ в строках и столбцах каждой подобласти, нормализованное по длине строки и столбца соответственно (3×2 признака).

Итого 150 первичных признаков для каждого прецедента.

Далее описано, как проводился отбор вторичных признаков. Прецедентная выборка разбивается на три равномощных множества (для отсутствия переобучения методом скользящего контроля). Первое — для построения вторичных признаков и селекции из них наилучших. Второе — для ограничения переобучения. Третье — для оценки качества классификации.

По выбранным 15 первичным признакам строятся все возможные вторичные признаки вида (2), сложность которых (число используемых операций) не превышала четырех. Среди них, для каждого класса, оставляем только те, с помощью которых можно отделить несколько (10 штук) прецедентов этого класса от остальных. При этом прецеденты рассматриваются только из первого множества разбиения.

Далее итеративно во множестве выбранных вторичных признаков (изначально оно пустое) производится операция добавления/удаления признаков. Критерий проведения операции — уменьшение функционала качества (3) на прецедентах второго множества разбиения, т.е. решаем задачу (4).

Данный этап требует основных вычислительных затрат, поэтому нуждается в теоретических исследованиях, с помощью которых можно будет сократить перебор.

Качество алгоритма оценивалась методом скользящего контроля. Для этого из всего множества прецедентов выбирался один объект, который помещался в контрольную выборку, на остальных проходило обучение. После получения оптимального алгоритма классификации (с

учетом ограничений — длина вторичного признака не больше 5 и ещё некоторых), проводилась классификация выбранного контрольного объекта. Данная процедура проводилась для каждого объекта из прецедентной информации.

Полученная оценка количества ошибок при классификации (на материале обучения) предложенным алгоритмом составляет приблизительно 5%. В задачах распознавания такой сложности, как решаемая нами, трудно требовать большей точности. Применяя современные методы классификации можно улучшить качество распознавания. Однако это может привести к известному эффекту переобучения, когда классификатор, корректно работающий на схемах "близких" к прецедентам, ведёт себя непредсказуемо на всех остальных объектах.

6. Выводы.Созданная модельная программная система показала принципиальною возможность реализации указанного подхода при создании промышленных САПР СБИС. Обучение практической системы формирования признакового пространства и разбиения его на области компетентности следует проводить на описаниях реальных схем. Объём и, главное, представительность набора прецедентов существенным образом определяет качество решения задачи. Заметим, что выбор в качестве Θ описаний автоматных функций (и, соответственно, в качестве Λ – множества алгоритмов синтеза автоматов) и/или в качестве $S(A, \delta)$ других характеристик схемы (максимальная задержка, длина критического пути, потребляемая мощность) не приведёт к принципиальному изменению предложенного метода решения задачи.

Сейчас автором статьи ведется активная исследовательская работа в области минимизации перебора вторичных признаков, целью которой является ускорение процесса обучения алгоритма классификации; а так же в области гипотезы компактности, целью которой является определение совокупной информативности вторичных признаков обходя процесс обучения классификатора.

Работа выполнена в сотрудничестве и по заказу ЗАО "Интел А/О". Авторы благодарят А.М. Марченко и О.В. Маевского за оказанную помощь и консультации.

Теоретическая часть исследования выполнена при финансовой поддержке РФФИ (код проекта 04-01-00161).

СПИСОК ЛИТЕРАТУРЫ

1. *Бонгард М.М.* Проблема узнавания. М.: Наука, 1967.
2. *Ту Дж., Гонсалес Ф.* Распознавание образов. - М.: Наука.1978.
3. *Журавлев Ю.И.* Об алгебраических методах в задачах распознавания и классификации // Распознавание. Классификация. Прогноз. Математические методы и их применение. Вып. 1. - М.: Наука. 1989.
4. *Айзерман М.А., Браверман Э.М., Розоноэр Л.И.* Метод потенциальных функций в теории обучения машин. М.: Наука, 1970.
5. SIS: A System for Sequential Circuit Synthesis / Dep. of Electrical Engineering and Computer Science Univ. of California, Berkeley. 1992.

РЕФЕРАТЫ

И. С. Барская, С. И. Мухин, В. М. Чечеткин. МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ РАВНОВЕСНЫХ КОНФИГУРАЦИЙ ЗВЕЗД НЕБОЛЬШОЙ МАССЫ // *СБОРНИК СТАТЕЙ МОЛОДЫХ УЧЕНЫХ факультета ВМиК МГУ, 2007, выпуск № 4, С. 3–15.* Одной из актуальных, но недостаточно изученных и требующих дальнейшего исследования проблем математического моделирования эволюции звезд является построение равновесных конфигураций звезд небольшой массы. Такие конфигурации важны для моделирования газодинамических процессов на этапах образования нейтронных звезд или черных дыр, а также для моделирования коллапса и вспышек сверхновых. В данной работе нами показано, как строится модель звезды и находятся ее равновесные конфигурации. Объект рассматривается как самогравитирующее газовое облако и описывается с помощью физических параметров, распределение которых находится из системы уравнений газовой динамики. Путем проведения серии численных экспериментов на основе построенной трехмерной нестационарной газодинамической модели получены равновесные конфигурации звезды с массой порядка 1.868 массы солнца на различных пространственных сетках. Найденные равновесные конфигурации могут быть использованы в качестве начальных данных при численном трехмерном моделировании газодинамических процессов на поздних этапах эволюции звезд.

Библиография 19 работ.

В. В. Глазкова, В. А. Масляков, И. В. Машечкин, М. И. Петровский . ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА АНАЛИЗА И ФИЛЬТРАЦИИ ИНТЕРНЕТ ИНФОРМАЦИИ // *СБОРНИК СТАТЕЙ МОЛОДЫХ УЧЕНЫХ факультета ВМиК МГУ, 2007, выпуск № 4, С. 16–24.* В статье предлагается новый подход, который заключается в применении методов машинного обучения и интеллектуального анализа данных при построении системы анализа и фильтрации Интернет трафика. Такие методы позволят разрабатываемой системе легко адаптироваться к постоянно изменяющейся природе Интернет ресурсов и учитывать специфику анализа сетевого трафика для различных организаций и стран. Систему такого уровня можно применять для анализа и фильтрации Интернет трафика в локальных сетях любых государственных, коммерческих и общественных организаций. Основными конкурентными преимуществами разрабатываемой системы являются её адаптируемость и точность работы.

Библиография 21 работа.

И. А. Громов. ИНТЕРАКТИВНЫЕ МЕТОДЫ КОРРЕКЦИИ ПОЛУМЕТРИК // *СБОРНИК СТАТЕЙ МОЛОДЫХ УЧЕНЫХ факультета ВМиК МГУ, 2007, выпуск № 4, С. 25–38.* Работа посвящена исследованию свойств функций расстояния и разработке новых методов преобразования метрической информации, используемых в интеллектуальном анализе данных. Рассмотрена задача коррекции полуметрики при заданном экспертом значении $\rho(i, j)$, которое требуется сохранить; введен ряд функционалов различия метрик; описана трехэтапная схема построения методов коррекции полуметрик. Предложен ряд алгоритмов, не требующих рассмотрения в процессе коррекции всех троек объектов и имеющих квадратичную, а в специальном случае - линейную по количеству объектов, сложность. Корректность алгоритмов была доказана. Предложен алгоритм коррекции, предусматривающий возможность применения процедуры обучения с целью улучшения качества коррекции.

Библиография 8 работ.

Ф. М. Жданов. ОЦЕНКА ПАРАМЕТРОВ РАЗРЯДА ПЛАЗМЫ МЕТОДОМ ОПОРНЫХ ВЕКТОРОВ // *СБОРНИК СТАТЕЙ МОЛОДЫХ УЧЕНЫХ факультета ВМиК МГУ, 2007, выпуск № 4, С. 39–45.* Работа посвящена применению нового подхода для оценки параметров удержания термоядерной плазмы в установках токамак. Для проведения разряда плазмы важно определить, в какой моде окажется разряд - в H-mode или L-mode. В настоящей работе для решения этих задач предлагается использовать современный статистический подход теории data mining - метод опорных векторов.

Библиография 9 работ.

С. С. Жулин. ЧИСЛЕННОЕ РЕШЕНИЕ П-СИСТЕМ ДЛЯ ЗАДАЧ ОПТИМАЛЬНОГО УПРАВЛЕНИЯ С ФАЗОВЫМИ И СМЕШАННЫМИ ОГРАНИЧЕНИЯМИ // *СБОРНИК СТАТЕЙ МОЛОДЫХ УЧЕНЫХ факультета ВМиК МГУ, 2007, выпуск № 4, С. 46–54.* Настоящая статья является продолжением работы [8], здесь решаются проблемы, связанные с применением предложенного метода к классу задач

оптимального управления с фазовыми и смешанными ограничениями. Предлагается метод нахождения управления, максимизирующего функцию Гамильтона-Понтрягина, посредством численного решения нелинейного уравнения. Приводятся выкладки нахождения производных максимизатора и множителей Лагранжа для построения системы уравнений первой вариации по начальным данным. Это позволяет применить метод продолжения по параметру к поиску экстремали Понтрягина в указанном классе задач.

Библиография 15 работ.

О. А. Казакова. О ДВУХ МОДЕЛЯХ ПОПУЛЯЦИОННОЙ ДИНАМИКИ Т-ЛИМФОЦИТОВ // *СБОРНИК СТАТЕЙ МОЛОДЫХ УЧЕНЫХ факультета ВМиК МГУ, 2007, выпуск № 4, С. 55–63.*

Сопоставлены две модели популяционной динамики Т-лимфоцитов и показано их соответствие при описании данных, характеризующих нормальные возрастные изменения Т-системы иммунитета. Для этого использованы аналог метода наименьших квадратов и вариационный принцип минимума диссипации энергии. Предполагается, что применение последнего позволяет неявно учитывать механизмы наблюдаемых изменений чувствительности иммунной системы при разной антигенной нагрузке и в дальнейшем послужит основой для разработки нестационарных динамических моделей долговременной адаптации системы иммунной защиты.

Библиография 16 работ.

А. Л. Кочеихин. РАЗРАБОТКА И АНАЛИЗ СХОДИМОСТИ СПЕЦИАЛЬНЫХ ПРОЦЕДУР СИНТЕЗА МЕТРИК В ЗАДАЧЕ ПОСТРОЕНИЯ ВЗАИМОСОГЛАСОВАННЫХ МЕР БЛИЗОСТИ КЛИЕНТОВ И ТОВАРОВ АССОРТИМЕНТА // *СБОРНИК СТАТЕЙ МОЛОДЫХ УЧЕНЫХ факультета ВМиК МГУ, 2007, выпуск № 4, С. 64–72.*

В данной работе описывается задача построения взаимосогласованных мер близости клиентов и товаров ассортимента, описываются процессы построения соответствующих метрик, а также доказывается их сходимость на основе принципа сжимающих отображений.

Библиография 3 работы.

О. В. Курчин. НОВЫЙ МЕТОД ОБУЧЕНИЯ БАЙЕСОВСКОЙ ЛОГИСТИЧЕСКОЙ РЕГРЕССИИ С ИСПОЛЬЗОВАНИЕМ ЛАПЛАСОВСКОГО РЕГУЛЯРИЗАТОРА // *СБОРНИК СТАТЕЙ МОЛОДЫХ УЧЕНЫХ факультета ВМиК МГУ, 2007, выпуск № 4, С. 73–81.* В данной статье представлен новый метод для обучения Байесовской логистической регрессии с использованием лапласовского регуляризатора. Основным его достоинством является значительное сокращение время обучения классификатора, по сравнению с существующими, при сравнимом качестве распознавания. Помимо описания самого метода в статье также кратко изложено каким образом может быть использован Байесовский подход при решении задач распознавания, описаны наиболее известные из существующих методов обучения логистической регрессии, а также приведены результаты сравнительных экспериментов.

Библиография 17 работ.

В. Б. Ларионов. ОБ ОДНОМ ПОДХОДЕ К ПОИСКУ БЫСТРЫХ АЛГОРИТМОВ УМНОЖЕНИЯ МАТРИЦ // *СБОРНИК СТАТЕЙ МОЛОДЫХ УЧЕНЫХ факультета ВМиК МГУ, 2007, выпуск № 4, С. 82–89.*

В данной работе исследуется новый подход к задаче об умножении матриц. От алгебры M квадратных матриц размера n осуществляется переход в более широкую алгебру с простым умножением P. С помощью операторов умножения устанавливаются необходимые для $M \in P$ условия, и строится конструктивный алгоритм проверки данного включения.

Библиография 10 работ.

Д. В. Левшин. ОБ ИНТЕГРАЦИИ SEMANTIC WEB С POSTGRESQL // *СБОРНИК СТАТЕЙ МОЛОДЫХ УЧЕНЫХ факультета ВМиК МГУ, 2007, выпуск № 4, С. 90–98.* В данной работе рассматривается концепция Semantic Web, основные форматы - RDF, RDFS, OWL и SWRL. Показаны основные возможности СУБД PostgreSQL, которые могут быть особенно полезны для решения задачи интеграции с Semantic Web. Предложен метод интеграции данной СУБД с Semantic Web, основанный на использовании системы правил, триггеров, возможностей расширения СУБД.

Библиография 10 работ.

З. С. Мальсагов . КОАЛИЦИОННО-УСТОЙЧИВОЕ РАВНОВЕСИЕ УГРОЗ И КОНТРУГРОЗ. МАТРИЧНЫЙ СЛУЧАЙ // *СБОРНИК СТАТЕЙ МОЛОДЫХ УЧЕНЫХ факультета ВМиК МГУ, 2007, выпуск № 4, С. 99–106.* В статье исследуются возможные подходы к понятию равновесия для игры, учитывающие не только наличие исходной коалиционной структуры, но и факт возможного её изменения в ходе игры.

Библиография 8 работ.

А. С. Мелузов. ОЦЕНКА СЛОЖНОСТИ ПРИМЕНЕНИЯ СИМВОЛЬНЫХ МЕТОДОВ В КРИПТОАНАЛИЗЕ АЛГОРИТМА ГОСТ 28147-89 // *СБОРНИК СТАТЕЙ МОЛОДЫХ УЧЕНЫХ факультета ВМиК МГУ, 2007, выпуск № 4, С. 107–110.* В докладе описан способ применения символьных методов криптографического анализа к алгоритму ГОСТ 28147-89, приведены способы построения системы полиномиальных уравнений, описывающих работу алгоритма ГОСТ 28147-89, проведена оценка сложности и структуры получаемой системы полиномиальных уравнений, а также проведена оценка эффективности алгоритмов решения систем полиномиальных уравнений над конечными полями с использованием стандартных базисов (базисов Гребнера).

Библиография 4 работы.

Е. А. Наградов, А. И. Качалин, В. А. Костенко. ПРИМЕНЕНИЕ АЛГОРИТМА K-MEANS ДЛЯ РЕАЛИЗАЦИИ АНАЛИЗАТОРА СТАТИСТИЧЕСКОЙ СИСТЕМЫ ОБНАРУЖЕНИЯ АТАК // *СБОРНИК СТАТЕЙ МОЛОДЫХ УЧЕНЫХ факультета ВМиК МГУ, 2007, выпуск № 4, С. 111–116.*

Данная работа затрагивает вопросы применения кластерного анализа для реализации анализатора статистической системы обнаружения атак, использующей подход обнаружения аномалий. В работе предложен метод обнаружения атак, основанный на применении алгоритма k-means, позволяющий выполнять обнаружение атак в режиме реального времени, а также приводятся результаты тестирования предложенного метода.

Библиография 9 работ.

С. В. Носов. ПРИМЕНЕНИЕ АДАПТИВНЫХ МЕТОДОВ ДЛЯ РЕШЕНИЯ ОБРАТНЫХ ЗАДАЧ ДИАГНОСТИКИ ПЛАЗМЫ // *СБОРНИК СТАТЕЙ МОЛОДЫХ УЧЕНЫХ факультета ВМиК МГУ, 2007, выпуск № 4, С. 117–119.* Обработка и интерпретация данных, получаемых при магнитной диагностике плазмы, является непростой задачей. Это связано как огромным потоком данных, так и со сложностью восприятия информации, представленной в виде разнообразных колебательных процессов. В настоящей работе для решения проблемы интерпретации магнитных измерений предлагается новый подход, основанный на использовании скрытых моделей Маркова, которые хорошо зарекомендовали себя при решении задачи распознавания человеческой речи.

Библиография 3 работы.

Д. А. Новикова, А. В. Поволоцкий. ФОРМУЛЫ ДЛЯ ПРЕОБРАЗОВАНИЯ ФУНКЦИЙ В ПРОСТРАНСТВЕ КОЭФФИЦИЕНТОВ РАЗЛОЖЕНИЯ ПО БАЗИСУ ПОЛИНОМОВ ЧЕБЫШЕВА ВТОРОГО РОДА // *СБОРНИК СТАТЕЙ МОЛОДЫХ УЧЕНЫХ факультета ВМиК МГУ, 2007, выпуск № 4, С. 120–128.* В работе описаны основные свойства семейства полиномов Чебышева второго рода, а также приведена информация о спектральном разложении и восстановлении сигналов по данному базису. Описаны алгоритмы преобразования сигналов в пространстве коэффициентов разложения по базису полиномов Чебышева второго рода для некоторых линейных преобразований над сигналами.

Библиография 5 работы.

Т. Г. Петросян. АСИМПТОТИКА ЛОГАРИФМА ЧИСЛА МНОЖЕСТВ, СВОБОДНЫХ ОТ ПРОИЗВЕДЕНИЙ, В ГРУППАХ // *СБОРНИК СТАТЕЙ МОЛОДЫХ УЧЕНЫХ факультета ВМиК МГУ, 2007, выпуск № 4, С. 129–135.* Получена асимптотика логарифма числа множеств, свободных от произведений, в конечных группах, размер максимального МСП которых не меньше трети порядка группы.

Библиография 7 работ.

А. В. Столяров. ОБ ОДНОМ ПОДХОДЕ К ПОСТРОЕНИЮ УНИВЕРСАЛЬНЫХ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ // *СБОРНИК СТАТЕЙ МОЛОДЫХ УЧЕНЫХ факультета ВМиК МГУ, 2007, выпуск № 4, С. 136–147.* В статье рассматривается проблема построения универсального языка программирования. Приводится классификация языков программирования по (1) уровню абстрагирования и (2) поддерживаемым парадигмам программирования. Предлагается решение, при котором реализуется язык программирования низкого уровня (подобный языку C), снабженный развитыми средствами построения абстракций высокого уровня и соответствующими библиотеками.

Библиография 4 названия.

И. Н. Фатхутдинов. ОПРЕДЕЛЕНИЕ ОБЛАСТЕЙ КОМПЕТЕНТНОСТИ АЛГОРИТМОВ СИНТЕЗА КОМБИНАЦИОННО-ЛОГИЧЕСКИХ СХЕМ. ГИПОТЕЗА КОМПАКТНОСТИ // *СБОРНИК СТАТЕЙ МОЛОДЫХ УЧЕНЫХ факультета ВМиК МГУ, 2007, выпуск № 4, С. 148–152.*

В работе представлен удачный опыт разбиения множества комбинационно-логических схем на классы, каждый из которых оптимальнее всего обрабатывается одним из конечного числа алгоритмов оптимизации. Описан способ проверки гипотезы компактности классов в выбранном признаковом пространстве. Не выполнение гипотезы компактности гарантирует невозможность решения задачи в выбранном признаковом пространстве.

Библиография 5 работы.
