СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ
SIBERIAN FEDERAL UNIVERSITY

ГОД ОБРАЗОВАНИЯ В СОДРУЖЕСТВЕ НЕЗАВИСИМЫХ ГОСУДАРСТВ

ПРОСПЕКТ СВОБОДНЫЙ-2016

МЕЖДУНАРОДНАЯ КОНФЕРЕНЦИЯ СТУДЕНТОВ,
АСПИРАНТОВ И МОЛОДЫХ УЧЁНЫХ

ЭЛЕКТРОННЫЙ СБОРНИК МАТЕРИАЛОВ МЕЖДУНАРОДНОЙ КОНФЕРЕНЦИИ СТУДЕНТОВ, АСПИРАНТОВ И МОЛОДЫХ УЧЁНЫХ **«ПРОСПЕКТ СВОБОДНЫЙ-2016»**, ПОСВЯЩЁННОЙ ГОДУ ОБРАЗОВАНИЯ В СОДРУЖЕСТВЕ НЕЗАВИСИМЫХ ГОСУДАРСТВ

КРАСНОЯРСК, СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ
15-25 АПРЕЛЯ 2016 Г.

Министерство образования и науки Российской федерации
ФГАОУ ВПО «Сибирский федеральный университет»

Сборник материалов Международной конференции студентов, аспирантов и молодых учёных «Проспект Свободный-2016», посвящённой Году образования в Содружестве Независимых Государств

Красноярск, Сибирский федеральный университет, 15-25 апреля 2016 г.

Красноярск, 2016

«Математика, информатика: теория вероятностей, математическая статистика и финансово-актуарная математика»

УДК 519.213

ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ РЕШЕНИЯ ЗАДАЧИ СЕТ-РЕГРЕССИИ

Атабеков М.
научный руководитель канд. физ.-мат. наук Семенова Д.В.
Сибирский федеральный университет

Для определения зависимости между двумя случайными множествами событий используется сет-регрессия, которая устанавливает вид средней функциональной зависимости между рассматриваемыми случайными множествами событий. Для нахождения регрессионных отображений используется подход, предложенный в работах [1-3]. Суть подхода заключается в нахождении регрессионного уравнения в виде условных сет-средних случайных множеств событий. В работе описывается разработанный комплекс проблемно-ориентированных программ SetRegression, направленных на решение задачи сет-регрессии.

Пусть $(\Omega, \mathsf{F}, \mathbf{P})$ — вероятностное пространство, (U, A) — измеримое пространство, где U — произвольное множество, а A — некоторая σ-алгебра его подмножества. Будем говорить, что функция $K = K(\omega)$, определенная на Ω и принимающая значения в U, есть F/A — измеримая функция, или случайный элемент (со значениями в U), есть для любого $A \in \mathsf{A}$ верно $\{\omega : K(\omega) \in \mathsf{A}\} \in \mathsf{F}$.

Случайное множество событий K на конечном множестве событий $\mathsf{X} \subset \mathsf{F}$ определяется как отображение $K : \Omega \to 2^{\mathsf{X}}$, измеряемое относительно пары алгебр $\left(\mathsf{F}/2^{2^{\mathsf{X}}}\right)$ такой, что $\mathbf{P}(X) = \mathbf{P}(K^{-1}(X))$. Вероятностное распределение случайного множества событий K, определенного над множеством избранных событий X можно представить несколькими эквивалентными распределениями вероятностей, порожденными этим множеством событий:

– вероятностное распределение I-го рода случайного множества K на X — это набор $\{p(X) = \mathbf{P}(K = X), X \subseteq \mathrm{X}\}$ из 2^{X} вероятностей вида

$$p(X) = \mathbf{P}\left(\left(\bigcap_{x \in X} x\right) \bigcap \left(\bigcap_{x \in X^c} x^c\right)\right); \qquad (1)$$

– вероятностное распределение II-го рода случайного множества событий K на X — это набор $\{p_X = \mathbf{P}(X \subseteq K), X \subseteq \mathrm{X}\}$ из 2^{X} вероятностей вида

$$p_X = \mathbf{P}\left(\bigcap_{x \in X} x\right). \qquad (2)$$

Вероятностные распределения I-го и II-го рода связаны формулами обращения Мёбиуса:

$$p_X = \sum_{X \subseteq Y} p(Y), \qquad p(X) = \sum_{X \subseteq Y} (-1)^{|Y|-|X|} p_Y. \qquad (3)$$

В [1, 2] введены следующие средние характеристики случайного множества событий и доказаны их экстремальные свойств

– сет-ожидание случайного множества событий K есть множество

$$\square K = \{x : p_x \geq h\}, \qquad (4)$$

где $h : \min_{\alpha \in [0,1]} \left\| \square K \right| - E|K| \|$;

– сет-квантиль порядка $\alpha \in [0,1]$ случайного множества событий K определяется как множество

$$\mathbf{Q}_\alpha K = \{ x : p_x \geq \alpha \}; \quad (5)$$

– сет-медиана случайного множества событий K определяется как сет-квантиль порядка $\alpha = \frac{1}{2}$, то есть

$$\mathbf{Med}K = \left\{ x : p_x \geq \frac{1}{2} \right\}. \quad (6)$$

– сет-мода случайного множества событий K есть множество

$$\mathbf{Mod}K = \max_{X \subseteq X} \mathbf{P}(K = X). \quad (7)$$

Рассмотрим постановку задачи сет-регрессии, используемую в работах [3, 4].

Задача сет-регрессии. *Пусть известны*

– *случайное множество событий* K *на* X;

– *случайное множество событий* L *на* Y;

– $\{p(X,Y) \mid X \subseteq \mathsf{X}, Y \subseteq \mathsf{Y}\}$ — *совместное распределение случайных множеств событий* K *и* L *на* $\mathsf{Z} = \mathsf{X} + \mathsf{Y}$, $\mathsf{X} \cap \mathsf{Y} = \varnothing$, *где* $p(X,Y) = \mathbf{P}(K = X, L = Y)$;

– *и пусть* $\rho : 2^{\mathsf{Y}} \to 2^{\mathsf{Y}}$ — *некоторая метрика в* Y.

Найти регрессионное отображение $\varphi : 2^X \to 2^Y$, *которое доставляет минимум функционалу* $\mathbf{E}\rho(L, \varphi(K))$, *где* $\mathbf{E}$ — *оператор математического ожидания.*

Таким образом, если дано совместное распределение двух случайных множеств событий K и L, значения которых содержаться в конечных множествах X и Y соответственно, то регрессией L на K называется любой оператор φ приближенно представляющий статистическую зависимость L от K. В работах [1-3] с использованием экстремальных свойств средних характеристик случайного множества событий доказано, что в зависимости от выбранной метрики ρ решение задачи сет-регрессии можно искать в виде:

– условного сет-ожидания $\square L \mid K$;

– условного сет-квантиля порядка α: $\mathrm{Q}_\alpha \left(L = Y \middle| K = X \right) = \{ y : p_{y|X} \geq \alpha \}$;

– условной сет-медианы $\mathrm{Med}\left(L = Y \middle| K = X \right) = \left\{ y : p_{y|X} \geq \frac{1}{2} \right\}$;

– условной сет-моды $\mathbf{Mod}\left(L \middle| K \right) = \max_{X \subseteq \mathsf{X}} \mathrm{P}(L = Y \mid_{K=X})$;

Автором разработан комплекс проблемно-ориентированных программ и алгоритмов моделирования регрессионных отображений на основе представленных выше теоретических результатов. Программный комплекс SetRegression разработан в среде программирования Pascal на языке программирования Delphi версии RAD Studio 10. Комплекс включает в себя программное приложение для решения следующих задач:

– построение булеана множества $\mathsf{Z} = \{ z_1, \ldots, z_n \}$, которое предполагает построение комбинаторных объектов – подмножеств конечного множества, заданной мощности $n = |\mathsf{Z}|$. Заметим, что множество Z имеет в точности 2^n различных подмножеств. Каждое из подмножеств $A \in \mathsf{Z}$ однозначно задается битовой шкалой $(a_1, \ldots, a_n)$, т.е. двоичным вектором длины n. Значит, построение всех различных

подмножеств множества $Z = \{z_1, \ldots, z_n\}$ можно свести к генерации всех n-разрядных двоичных векторов. В программном комплексе решение данной задачи реализовано с помощью алгоритма построения нерекурсивного кода Грея [6];

– генерирование/ввод совместного распределения I-го рода случайного множества событий, определенного на Z;

– переход между вероятностными распределениями случайного множества событий I-го и II-го рода по формулам обращения Мёбиуса (3);

– построение заданного пользователем разбиения множества Z на два непересекающихся подмножества X и Y ($Z = X + Y$, $X \cap Y = \varnothing$) и построение булеанов X и Y;

– матричное представление совместного распределения $\{p(X,Y) \mid X \subseteq X, Y \subseteq Y\}$ случайных множеств событий K и L, заданных соответственно на множествах X и Y, где $p(X,Y) = \mathbf{P}(K = X, L = Y)$;

– построение условных распределений случайного множества событий L при условии, что случайное множество K приняло значение $X \subseteq X : \{p(Y|X) | Y \subseteq Y\}$, где $p(Y|X) = \mathbf{P}(L = Y|_{K=X})$;

– для заданного случайного множества событий нахождение его сет-средних по формулам (4) – (7);

– нахождение сет-регрессии и построение графика функции сет-регрессии.

На рисунке 1 представлены результаты работы программногокомплекса SetRegression.

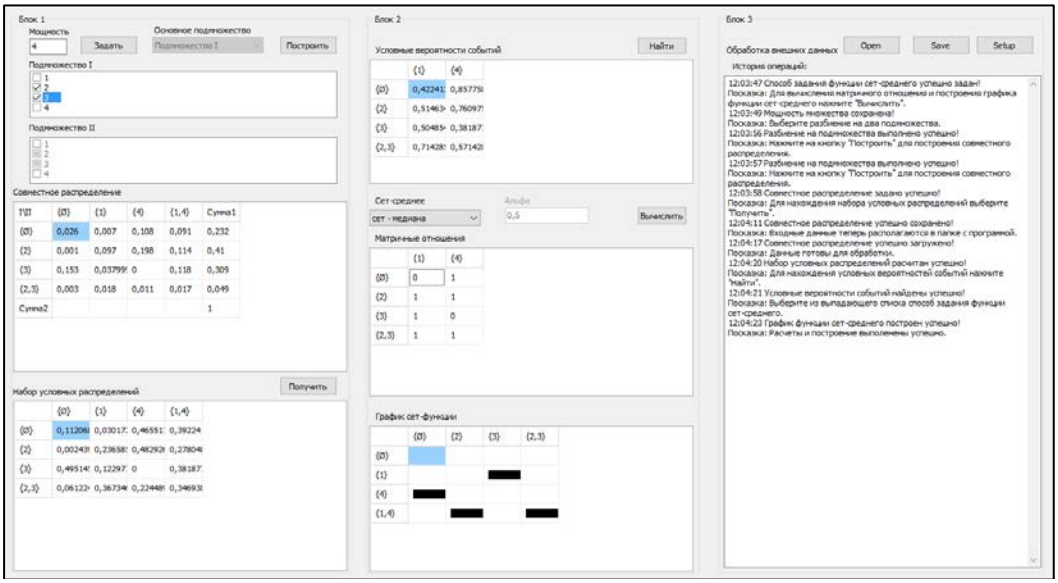

Рис. 1 – Пример работы программного комплекса SetRegression

Разработанный комплекс программ SetRegression позволяет проводить построение, исследование и анализ сет-регрессий случайных множеств событий. Данный комплекс может быть использован в научной и образовательной деятельности, а также для исследования функционирования реальных экономических систем [2-4].

Список литературы

1. Воробьев О.Ю. Эвентология. Красноярск: Сибирский федеральный университет, 2007
2. Воробьев О.Ю., Фомин А.Ю. Сет-регрессионный анализ зависимостей событий в статистических системах. Красноярск: ИВМ СО РАН, КрасГУ, 2004.
3. Тарасова, О. Ю. Сеточные и регрессионные алгоритмы аппроксимации сложных систем событий: автореф. дис. на соиск. учен. степ. канд. физ.-мат. н. Красноярск: СФУ, 2007.
4. Семенова Д.В., Иванова А.И. Нечеткая и сет-регрессионная модели распределения потребительских предпочтений между фирмами // Труды XIII международной ФАМЭМС'2014 конференции (под ред. О.Ю. Воробьёва). — Красноярск: СФУ, НИИППБ 2014. – с.164-170
5. Иванова А.И., Атабеков М.А. О задаче сет-регрессии // Труды XIV конференции по финансово-актуарной математике и эвентологии многомерной статистики, Красноярск, 2015 – С. 285-292.
6. Быкова, В. В. Комбинаторные алгоритмы: множества, графы, коды: учеб. пособие. Красноярск: Сиб. федер. ун-т, 2015.

УДК 519.23

ИССЛЕДОВАНИЕ ВЛИЯНИЯ ПОКАЗАТЕЛЕЙ ЗДОРОВЬЯ ПАЦИЕНТА НА ИСХОД ОПЕРАЦИИ И ЕГО ПРОГНОЗИРОВАНИЕ

Горохова Е.А.
научный руководитель канд. физ.-мат. наук, доц. Баранова И.В.
Сибирский федеральный университет

В работе проводится исследование влияния показателей здоровья пациентов на исход операции, а также выполняется его прогнозирование. Целью исследования является решение следующих задач: выявление связи между двумя или более показателями из выборки; определение типа этой связи и ее силы; осуществление прогноза на основе этой связи. Для решения этих задач в работе используются методы корреляционного и регрессионного анализа.

Корреляционный анализ — метод обработки статистических данных, с помощью которого измеряется теснота связи между двумя или более переменными. Корреляционный анализ тесно связан с регрессионным анализом (также часто встречается термин «*корреляционно-регрессионный анализ*», который является более общим статистическим понятием), с его помощью определяют необходимость включения тех или иных факторов в уравнение множественной регрессии, а также оценивают полученное уравнение регрессии на соответствие выявленным связям (используя коэффициент детерминации).

Корреляция представляет собой статистический метод, позволяющий определить, существует ли зависимость между переменными и насколько она сильна.

Регрессия – статистический метод, который применяется для описания характера связи между переменными (положительная или отрицательная, линейная или нелинейная зависимость).

Описание статистических данных

В работе использовалась статистика проведения операций, предоставленная медицинским учреждением, в которой содержатся данные о 166 пациентах. Данные разнесены в две большие группы: в первой группе перечислены случаи, в которых операции прошли с осложнениями, а во второй группе перечислены случаи, в которых операция прошла успешно, без осложнений.

В представленной статистике для каждой операции (оперируемого пациента) содержится 181 показателей, в том числе показатели здоровья пациента, личные данные и показатели медицинского обслуживания. Перечислим некоторые из них: ФИО, дата поступления, группа крови и её резус, артериальное давление при поступлении, возраст, рос, вес, пристрастия к вредным привычкам, наличие некоторых заболеваний, квалификация врача. На основании статистики было проведено исследование зависимости исхода операции от параметров и зависимость параметров друг от друга.

Задача выявления зависимости и характера связи

Как уже было сказано, для решения этих задач в работе используются методы корреляционного и регрессионного анализа.

Корреляция (корреляционная зависимость) — статистическая взаимосвязь двух или нескольких случайных величин (либо величин, которые можно с некоторой допустимой степенью точности считать таковыми). При этом изменения значений одной или нескольких из этих величин сопутствуют систематическому изменению значений другой или других величин. Математической мерой корреляции двух

случайных величин служит *корреляционное отношение*, либо коэффициент корреляции.

Пусть **X** и **Y** – две нормально распределенные случайные величины с математическим ожиданием μ_x и μ_y и ненулевыми дисперсиями σ_x^2 и σ_y^2. Рассмотрим нормированные случайные величины X^* и Y^*, определенные формулами:

$$X^* = \frac{X - \mu_x}{\sigma}, \qquad Y^* = \frac{Y - \mu_y}{\sigma}.$$

Корреляция этих случайных величин обозначается ρ(X,Y) и имеет вид:

$$\rho(X,Y) = Cov(X^*, Y^*) = \frac{Cov(X^*, Y^*)}{\sigma_x \sigma_y}.$$

Здесь $Cov(X,Y) = E((X - \mu_x)(Y - \mu_y)) = E(XY) - \mu_x \mu_y$ - ***ковариация,*** мера линейной зависимости двух случайных величин.

Если ковариация положительна, то с ростом значений одной случайной величины, значения второй имеют тенденцию возрастать, а если отрицательна — то убывать.

Пусть имеются пары наблюдаемых значений для параметров x и y – (x_1, y_1), (x_2, y_2), ..., (x_i, y_i), ..., (x_n, y_n).

Выборочной оценкой коэффициента корреляции ρ является ***коэффициент корреляции Пирсона***:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}},$$

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \quad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i,$$

где n – объем выборки.

Коэффициент корреляции измеряет силу и направление связи между двумя переменными, изменяется на отрезке от –1 до +1.

Если между переменными существует сильная положительная связь, то значение r будет близко к +1. Если между переменными существует сильная отрицательная связь, то значение r будет близко к –1.

Когда между переменными нет линейной связи или она очень слабая, значение r будет близко к нулю. Эта ситуация отсутствия статистической взаимосвязи может встречаться, например, для независимых случайных величин.

Коэффициент Пирсона используется для работы с числовыми параметрами, для нечисловых параметров используют коэффициент Спирмена и коэффициент детерминации.

В работе рассматриваются только числовые параметры.

Коэффициент корреляции Спирмена

Корреляция Спирмена является ранговой, то есть для оценки силы связи используются не численные значения, а соответствующие им ранги.

Рассмотрим последовательность рангов R_i (величин x_i) и R_j^* (величину y_j).

Находим разность рангов $d_i = R_i - R_j^*$, соответствующую паре (x_i, y_i). Коэффициент корреляции Спирмена определяется формулой:

$$\rho = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)}$$

Коэффициент детерминации

Коэффициент детерминации – это мера вариации зависимой переменной, которая определяется линией регрессии и независимой переменной.

$$r^2 = \frac{\sum\left(y' - \bar{y}\right)^2}{\sum\left(y - \bar{y}\right)^2}.$$

Коэффициент детерминации принимает значения от -1 до +1.

Регрессионный анализ

Как говорилось ранее, регрессия описывает характер связи между переменными.

Существует линейная и множественная регрессия.

Линейная регрессия

Линейный регрессионный анализ исходит из наличия зависимости $y = \alpha + \beta x$, где α и β − неизвестные коэффициенты регрессии. Выборочные оценки α и β в дальнейшем будем обозначать a и b, соответственно.

Оценки коэффициентов регрессии вычисляются по формулам

$$b = \frac{\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}$$

$$a = \frac{\sum_{i=1}^{n} y_i - b\sum_{i=1}^{n} x_i}{n}$$

Множественная линейная регрессия

Если случайная величина может зависеть одновременно от двух и более переменных, возникает задача оценки и анализа множественной регрессии $y=f(x_1, x_2, .., x_k)$, где x_j (j=1, 2, ..., k) − независимые переменные.

Рассмотрим в качестве примера регрессию $y = \alpha + \beta_1 x_1 - \beta_2 x_2$ для случая двух независимых переменных x_1 и x_2. Для большего числа переменных оценки могут быть получены по аналогии. Для нашего случая оценки коэффициентов β_1, β_2 и α имеют вид:

$$b_1 = \frac{S(x_2^2)S(x_1,y) - S(x_1x_2)S(x_2y)}{S(x_1^2)S(x_2^2) - [S(x_1x_2)]^2};$$

$$b_2 = \frac{S(x_1^2)S(x_2y) - S(x_1x_2)S(x_1y)}{S(x_1^2)S(x_2^2) - [S(x_1x_2)]^2};$$

$$a = \bar{y} - b_1x_1 - b_2x_2;$$

$$S(x_j^2) = \sum_{i=1}^{n}(x_{ji} - \bar{x}_i)^2 \quad , j=1,2;$$

$$S(x_jy) = \sum_{i=1}^{n}(x_{ji} - \bar{x}_i)(y_i - \bar{y}) \, , j=1,2;$$

$$S(x_1x_2) = \sum_{i=1}^{n}(x_{1i} - \bar{x}_1)(x_{2i} - \bar{x}_2);$$

$$S(y^2) = \sum_{i=1}^{n}(y_i - \bar{y})^2 .$$

После нахождения уравнения регрессии можно строить прогнозы: для значения независимой переменой x определить значение зависимой переменной y. Необходимо оценить надежность прогноза: для этого требуется найти значение коэффициента детерминации, стандартную ошибку прогноза и интервал предсказания.

Стандартная ошибка оценки – это стандартное отклонение наблюдаемых значений y

от предсказываемых значений y', где $y' = ax + b$:

$$s_{est} = \sqrt{\frac{\sum y^2 - b\sum y - a\sum xy}{n-2}} .$$

Интервал предсказания имеет вид: $y' - E < y < y' + E$, где

$$E = t_{\frac{\alpha}{2}} \cdot s_{est} \cdot \sqrt{1 + \frac{1}{n} + \frac{n(x-\bar{x})^2}{n\sum x^2 - \left(\sum x\right)^2}} .$$

Практическая задача анализа и прогнозирования

В данной работе была рассмотрена задача исследования и прогнозирования исхода операции. Были вычислены коэффициенты корреляции и построены линии линейной и множественной регрессии. Проведено исследование влияния показателей здоровья пациентов на исход операции. Выявлена связь между двумя или более показателями из выборки, и осуществлен прогноз исхода операции на основе этой связи. Также выполнена оценка достоверности результатов исследования.

Список литературы

1. Кобзарь А. И. Прикладная математическая статистика. Для инженеров и научных работников. М.: ФИЗМАТЛИТ, 2006. 816 с.

2. Вентцель Е.С. Теория вероятностей. Наука, 1969. 576 с.

3. Кендалл М.Дж., Стьюарт А. Статистические выводы и связи. М.: Наука. 1973. 899 с.

4. Феллер В. Введение в теорию вероятностей и ее приложения. В 2-х томах; пер. с англ. М.: Мир, 1984. Т.1: 528 с. Т2: 752с.

УДК 519.7

ФОРМИРОВАНИЕ И ВЕДЕНИЕ СЛОВАРЕЙ В КОРПУСЕ ТУВИНСКОГО ЯЗЫКА

Дагбажык А.С.
научный руководитель д-р физ.-мат. наук Быкова В.В.
Сибирский федеральный университет

В настоящее время активно создаются корпусы естественных языков с помощью современных информационных технологий и методов математического моделирования. Под корпусом понимается информационно-справочная система, основанная на собрании оцифрованных текстов. Корпус включает в себя различные типы письменных и устных текстов, представленных в данном языке, различные типы словарей, а также разметку – информацию о свойствах текстов. Разметка отличает корпус от электронных библиотек текстов [2]. Традиционно в корпусах используются следующие типы разметки: метатекстовая, морфологическая, синтаксическая, семантическая и др. Чем богаче и разнообразнее разметка, тем выше научная и учебная ценность корпуса. На основании корпусов решаются многие филологические и лингвистические задачи. Для многих языков народов Российской Федерации, в том числе для тюркских языков, создаются национальные корпусы. Работа над формированием Национального корпуса тувинского языка ведется преподавателями, аспирантами и студентами Тувинского государственного и Сибирского федерального университетов[1, 2].

Словари – важная часть корпуса. Различают несколько типов словарей. Словари в корпусах, как правило, многофункциональны. Корпусный словарь содержит всю лингвистическую информацию о каждом слове. В корпусах используются следующие основные типы словарей: диалектные, грамматические, орфографические, словообразовательные, переводные.

Диалектные (областные) словари – разновидность толковых словарей, описывающих лексику одного или группы говоров (диалектов). Диалект (от греч. dialektos – разговор, говор, наречие) – разновидность данного языка, употребляемая в качестве средства общения лицами, связанными тесной территориальной, социальной или профессиональной общностью.

Грамматические словари – это словари, которые содержат сведения о морфологических и синтаксических свойствах слова. Морфология (от др.-греч. – «форма» и «учение») – раздел грамматики, изучающий части речи, их категории и формы слов. Синтаксис (от др.-греч. – «построение, порядок, составление») – раздел лингвистики, изучающий строение словосочетаний и предложений.

Орфоэпические словари – словари, отражающие правила литературного произношения. Орфография – раздел лингвистики, изучающий правильность написания слова при письме. Орфоэпия (от др.-греч – «правильный» и греч. – «речь») – раздел фонетики, занимающийся нормами произношения, их обоснованием и установлением**.**

Словообразовательные словари (деривационные словари) – словари, показывающие членение слов на составляющие их морфемы, словообразовательную структуру слова, а также совокупность слов (словообразовательное гнездо) с данной морфемой – корневой или аффиксальной. Слова в словообразовательных словарях приводятся с расчленением на морфемы и с ударением.

Переводные словари – словари, содержащие сопоставление слова одного языка их переводным эквивалентам на другом языке (или на нескольких других языках, в таком случае переводной словарь является многоязычным). Переводные словари условно разделяют на две большие группы:

- общелексические переводные словари. Переводят общую лексику с одного языка на другой или на несколько иностранных языков;
- научные, научно-технические и технические переводные словари. Включают в себя специальные термины по основным отраслям науки и техники.

Несмотря на то, что существуют различные типы словарей, в их структуре можно выделить составные части, присутствующие под разными названиями практически во всех словарях. К таким элементам относятся [3]:

- введение или предисловие (*Introduction* или *Preface*);
- раздел «Как пользоваться словарем» (*User's Mannual* или *Guide to the Dictionary*);
- ключ к системе транскрипции, применяемой в словаре (*Transcription Codes* или *Keys to the Transcription*);
- список сокращений, используемых в словаре, и их объяснения (*Contractions* или *Contraction Codes* / *Keys*);
- корпус словаря (*Corpus* или *The Body of the Dictionary*), то есть основной список слов, представленных их словарными статьями;
- дополнительный материал, то есть различные приложения.

Корпус словаря составляют словарные статьи. Ступиным Л.П. в англо-русских переводных словарях выделены следующие части словарной статьи: *entry word* / *catch word* / *head word* – заглавное слово; *sense* / *meaning of the word* – значение слова; *definition* – толкование, определение; *verbal illustration* / *quotation* – цитата, иллюстрация; *reference* – отсылка; *label* – метка; *status label* – метка о временной или территориальной ограниченности употребления слова; *regional label* – метка о территориальной употребительности слова; *functional label* – метка о принадлежности слова к части речи; *subject label* – метка о принадлежности слова к определенной области знаний [3].

При разработке словаря для корпуса тувинского языка взята за основу структура словарной статьи Ступина Л.П. Словарная статья в корпусе тувинского языка имеет следующую структуру:

- заглавное слово;
- перевод (на русский язык, на английский язык и на другие языки);
- транскрипция;
- звучание заглавного слова;
- метка о морфологических признаках (часть речи, число, падеж, склонение, спряжение);
- значение слова;
- этимологическая справка;
- метка о принадлежности к аббревиатурам;
- метка о наличии синонима, омонима и антонима;
- дополнительная информация о слове.

База данных корпусного словаря включает следующие таблицы: main – основная таблица с заголовочным словом; rus, eng, ger – таблицы с переводом заголовочного слова на различные языки; morphology – таблица с морфологическими данными. Структура этих таблиц представлена на рис.1 – 4. База данных реализована в Microsoft Office Access. Для работы с корпусным словарем реализованы следующие функции: добавление новой статьи (рис. 5); редактирование статьи (рис. 6); удаление статьи (рис. 7); поиск словарной статьи с транскрипцией (рис. 8); формирование и визуализация морфологических признаков заглавного слова (рис. 9).

main

Имя поля	Тип данных	Описание (необ
entry_id	Счетчик	идентификатор для заголовочного слова
article	Короткий текст	заголовочное слово
mean	Длинный текст	значение слова
transcription	Короткий текст	транскрипция слова
example	Длинный текст	примеры применения слова в предложениях и в речи
speech	Числовой	помета о принадлежности части речи

Рисунок 1 – Структура таблицы main

rus

Имя поля	Тип данных	Опи
rus_id	Счетчик	идентификатор для слова на русском
russian	Короткий текст	слово на русском языке
entry_id	Числовой	используется для связи с таблицей main

Рисунок 2 – Структура таблицы rus

eng

Имя поля	Тип данных	Опис
eng_id	Счетчик	идентификатор для слова на английском
english	Короткий текст	слово на английском языке
entry_id	Числовой	используется для связи с таблицей main

Рисунок 3 – Структура таблицы eng

morphology

Имя поля	Тип данных	Опи
entry_id	Числовой	используется для связи с таблицей main
case_1	Короткий текст	падеж 1
case_2	Короткий текст	падеж 2
case_3	Короткий текст	падеж 3
case_4	Короткий текст	падеж 4
case_5	Короткий текст	падеж 5
case_6	Короткий текст	падеж 6
case_7	Короткий текст	падеж 7
case_8	Короткий текст	падеж 8
case_9	Короткий текст	падеж 9

Рисунок 4 – Структура таблицы morphology

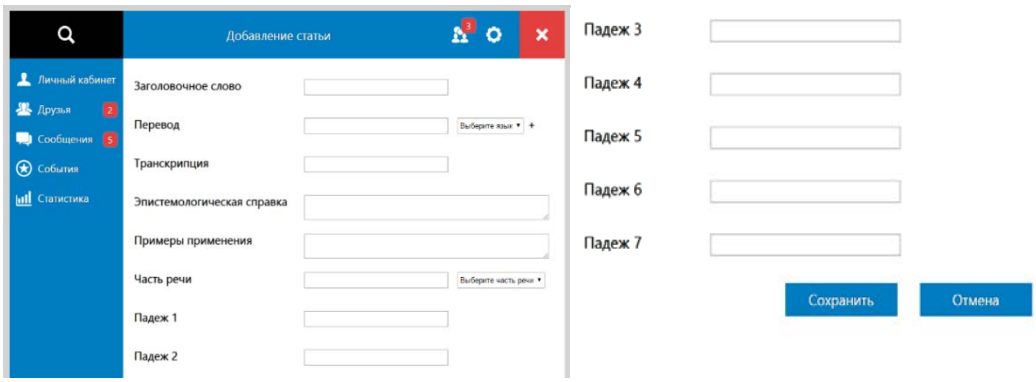

Рисунок 5 – Интерфейс для добавления новой словарной статьи

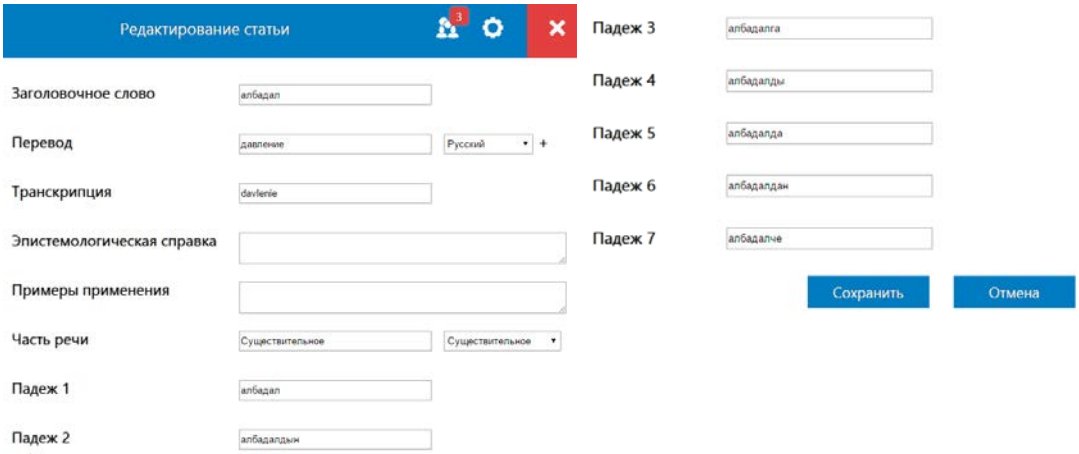

Рисунок 6 – Интерфейс для редактирования статьи

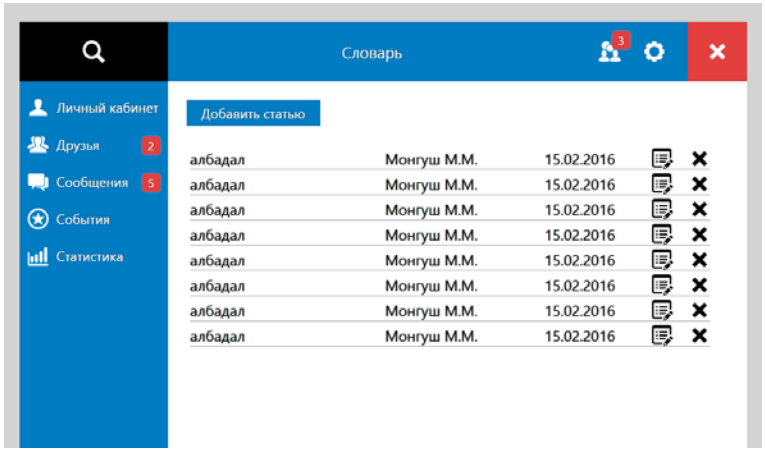

Рисунок 7 – Интерфейс для удаления статьи

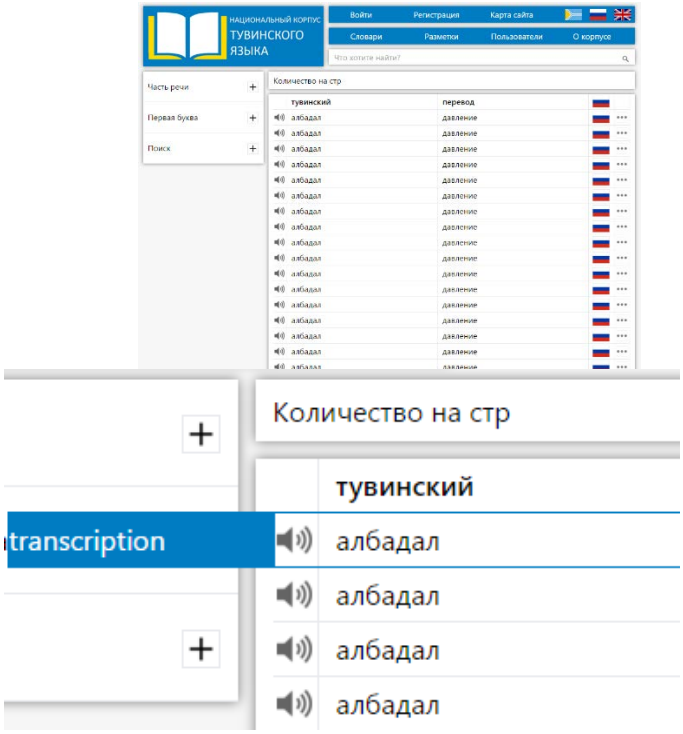

Рисунок 8 – Поиск словарной статьи с транскрипцией

Перевод		
аргументация	барымдаазын кергузер	барымдаалаар

Грамматические признаки

Род		
Мужской (masculinum)	Женский (femininum)	Средний (neutrum)
Число		
Единственное (Singular)	Множественное (Plural)	Собирательное (Multiplex)
Лицо		
Первое	Второе	Третье
Падеж		
Именительный		
Родительный		
Дательный		
Винительный		
Творительный		
Предложный		

Морфологические признаки

Часть речи
Имя существительное
Имя прилагательное
Имя числительное
Наречие
Местоимение
Глагол
Причастие
Деепричастие

Рисунок 9 – Морфологические признаки заглавного слова

Представленная выше структура словарной статьи и ее реализация в Microsoft Office Access позволяет рассматривать корпусный словарь как мультиструктурную организацию со сложным иерархическим строением, важнейшим корневым компонентом которого является словарная статья. Словарная статья является единицей корпусного словаря, которую можно формировать в различных аспектах и использовать для раз-

личных назначений: изучение тувинского языка с точки зрения написания, произношения, толкование, перевода на другие языки; формирование морфологической, синтаксической и семантической разметки текстов, входящих в корпус; играть роль тезауруса при анализе текстов; для организации поиска в корпусе.

В дальнейшем предполагается выполнение работ по формированию корпусного двуязычного словаря (тувинско-русского и русско-тувинского) с заполнением основных полей словарных статей преподавателями, аспирантами и студентами Тувинского государственного и Сибирского федерального университетов.

Список литературы

1. Бавуу-Сюрюн М. В. Тувинский язык на современном этапе [электронный ресурс]. Режим доступа: http://www.tuva.asia/journal/issue_7/2158-bavyy-suyruyn-mv.html

2. Салчак А. Я., Байыр-оол А. В. Электронный корпус тувинского языка: состояние, проблемы // Мир науки, культуры, образование, 2013. № 6. С. 408–409.

3. Ссорина М. С. Словарь как мультиструктурная организация // Ярославский педагогический вестник – 2011 – № 1. – Том 1 (Гуманитарные науки).

УДК 519.72

ИССЛЕДОВАНИЕ КРИПТОГРАФИЧЕСКИХ СИСТЕМ РЮКЗАЧНОГО ТИПА

Додонова М.М.
научный руководитель канд. физ.-мат. наук, доц. Баранова И.В.
Сибирский федеральный университет

Одной из наиболее актуальных и практически востребованных задач современной прикладной математики и информационных технологий является задача защиты информации.

Разработкой методов преобразования (*шифрования*) информации с целью ее защиты от незаконных пользователей занимается ***криптография***. Криптографическое преобразование – преобразование информации, основанное на некотором алгоритме, зависящем от изменяемого параметра (обычно называемого *секретным ключом*), и обладающее свойством невозможности восстановления исходной информации по преобразованной, без знания действующего ключа.

Практическое применение криптографии стало неотъемлемой частью жизни современного общества — её используют в таких отраслях как электронная коммерция, электронный документооборот (включая цифровые подписи), электронное голосование, телекоммуникации и других. Основным достоинством криптографических методов является то, что они обеспечивают высокую гарантированную стойкость защиты, которую можно рассчитать и выразить в числовой форме (средним числом операций или временем, необходимым для раскрытия зашифрованной информации или вычисления ключей).

Для современной криптографии характерно использование открытых алгоритмов шифрования, предполагающих использование вычислительных средств. Все методы можно подразделить на два класса: шифрование с симметричными ключами и с открытыми (асимметричными) ключами.

Криптографические системы с открытым ключом

Криптографическая система с открытым ключом (или ***асимметричное шифрование***) — система шифрования, при которой открытый ключ передаётся по открытому каналу и используется для шифрования сообщения. Для расшифровки сообщения используется секретный ключ. Криптографические системы с открытым ключом в настоящее время широко применяются в различных сетевых протоколах и стандартах цифровой подписи.

Наиболее распространенными в настоящее время являются криптосистемы, основанные на задаче факторизации (***RSA***). Но постоянное изменение параметров подобных систем приводит к определенным сложностям при их использовании. Поэтому большую актуальность приобрело направление разработки методов построения асимметричных криптосистем, не использующих факторизацию.

Одними из наиболее перспективных криптографических систем защиты информации являются ***рюкзачные системы защиты информации*** (***РСЗИ***). В их основе лежит известная задача о ранце или рюкзаке.

Задача о ранце (рюкзаке) – одна из NP-трудных задач комбинаторной оптимизации. Своё название задача получила от оптимизационной задачи укладки как можно большего числа ценных вещей в рюкзак при условии, что общий объём (или вес) всех предметов, способных поместиться в рюкзак, ограничен.

На ее основе был разработан ряд криптосистем, отличающихся простотой реализации. Криптосистема Меркла-Хеллмана представляет собой первый основанный

на задаче о ранце алгоритм для обобщённого шифрования с открытым ключом. В дальнейшем было предложено множество модификаций криптосистемы Меркла-Хеллмана и совершенно новых криптосистем на основе задачи о ранце (рюкзак Грэм-Шамира, Гудмана-Макколи, Накаше-Штерна и другие). Но в ходе их анализа были выявлены существенные недостатки, делающие эти системы уязвимыми для различного вида криптографических атак. В настоящее время системы этого класса не получили широкого распространения, но ведется работа по их модификации, которая позволит улучшить их надежность. Одним из способов такой модификации является использование специальных порождающих множеств в конечных группах для создания рюкзачной схемы.

Рюкзачные криптосистемы

Пусть имеется рюкзачный вектор $A = (a_1, a_2 ... a_n)$, представляющий собой упорядоченный набор из n различных натуральных чисел a_i, $n \geq 3$. ***Входом задачи о рюкзаке*** называем пару (A, m), где A — рюкзачный вектор, а m — натуральное число. Требуется найти вектор $X = (x_1, .., x_n)$, для которого верно

$$\sum_{j=1}^{n} a_j x_j = m, \qquad (1)$$

$$\text{где } x_j \in \{0,1\},\ j = (1, ..., n).$$

В общем случае для данной задачи нет эффективного алгоритма решения. Приходится применять полный перебор для нахождения решения или доказательства его отсутствия. Кроме того, в общей постановке задача о рюкзаке может иметь несколько различных решений. Но если рюкзак является *сверхрастущим*, то решение в случае его существования единственно и существует эффективный алгоритм его нахождения.

Рюкзачный вектор $A = (a_1, a_2 ... a_n)$ называется ***сверхрастущим***, если $a_j > \sum_{i=1}^{j-1} a_i$ для всех $2 \leq j \leq n$ [2].

Криптосистема Меркла-Хеллмана

Генерация ключей:

1. Выбирается некоторый сверхрастущий рюкзак.
2. Выбирается число k, где $k > \sum_{i=1}^{n} a_i$.
3. Выбирается число c, взаимно простое с k.
4. Формируется рюкзак-ловушка $B = (b_1, b_2 ... b_n) = c(a_1, a_2 ... a_n) (\text{mod } k)$, который и является открытым ключом.
5. Числа c и k являются секретными ключами.

Алгоритм шифрования:

1. Открытый текст представляется в виде двоичной последовательности.
2. Последовательность разбивается на блоки длины n.
3. Каждый блок (x1,x2,…xn) заменяется на число m, вычисленное по правилу

$$\sum_{j=1}^{n} a_j x_j = m.$$

Алгоритм дешифровки:

1. Находится исходный сверхрастущий рюкзак:

$A = (a_1, a_2 ... a_n) = c^{-1}(b_1, b_2 ... b_n)(\text{mod } k)$.

2. Для каждого элемента m шифра текста вычисляется элемент $m' = c^{-1}m$.

3. Для вычисленного m' решается задача о рюкзаке для $A = (a_1, a_2 ... a_n)$ и находится блок открытого текста $X = (x_1, .., x_n)$.

Как уже было сказано ранее, в ситуации сверхрастущего рюкзака нет необходимости полного перебора – достаточно выполнить просмотр вектора *A* один раз справа налево (начиная с самого большого элемента). Для заданного *k* (размера рюкзака) мы сначала проверяем $k \geq a$. Если ответом будет «нет», a_n не может входить в искомую сумму. Если же ответом будет «да», то a_n *должно* входить в сумму. Это следует из того факта, что все оставшиеся a_i -е в сумме дают число , меньшее *k*.

В криптосистеме Грэма – Шамира также используется сверхрастущий рюкзак. Но маскировка его под рюкзак общего вида производится не с помощью приведения по модулю, а с использованием вектора случайного шума. В системе Мори – Касахары используется мультипликативный способ шифрования и формирования секретного ключа. Некоторые из рюкзачных криптосистем до сих пор считаются стойкими, например, модель, предложенная в 1988 году Б. Шором и Р. Ривестом. Эта криптосистема не использует модульное умножение для того, чтобы скрыть рюкзачный вектор. Вместо этого использовалась арифметика конечных полей Галуа.

В работе был рассмотрен стандартный алгоритм построения сверхрастущего вектора на основе вектора $(1,2,2^2,...,2^{n-1})$ (степени двойки). А также предложены собственные алгоритмы построения сверхрастущих векторов.

Алгоритм построения сверхрастущего вектора с фиксированным шагом

Если a_1 – случайно выбранное число, $1 \leq a_1 \leq W$, и $a_j > h_{j-1} + \sum_{i=1}^{j-1} a_i$, где $1 \leq h_{j-1} \leq H$, $j = 2,...,n$, то вектор $A = (a_1, a_2 ... a_n)$ является сверхрастущим.

Алгоритм со случайно выбранным шагом

Если a_1 – случайно выбранное число, $1 \leq a_1 \leq W$, и $a_j > h + \sum_{i=1}^{j-1} a_i$, где $0 < h \leq H\,(h = const)$, $j = 2,...,n$, то вектор $A = (a_1, a_2 ... a_n)$ является сверхрастущим.

Проведено исследование зависимости скорости шифровки и расшифровки алгоритмов в зависимости от способа построения сверхрастущего вектора.

Математические модели нестандартных рюкзачных систем защиты информации

Также в работе рассматривались более сложные рюкзачные модели защиты информации. Общей их особенностью является то, что в отличие от стандартного случая, в них допускаются повторения компонент. Сообщение в этой модели сообщение разбивается на буквенные блоки, вместо которых далее рассматриваются их числовые эквиваленты $v \in Z_M$. Параметры и преобразования простейшей схемы на основе нестандартного рюкзака можно представить следующим образом:

Параметры системы: основание системы, размер рюкзачных векторов и вектора ограничений на количество повторений элементов рюкзака:

$$M \in \mathrm{N},\ n \in \mathrm{N},\ m = (m_1, \ldots, m_n).$$

Закрытый ключ: элемент для модульного умножения и сверхрастущий рюкзачный вектор:

$$A = (a_1, a_2 ... a_n),\ a_i \in Z_M,\ a_i > \sum_{j=1}^{i-1} m_j a_j.$$

Открытый ключ:

$$d \in Z_M,\ ed = 1 (\mathrm{mod}\ M),\ B = (b_1, b_2 ... b_n),\ b_i = ea_i (\mathrm{mod}\ M).$$

Прямое преобразование: для входа (A, m) вычисляется спектр X:

$$X = (x_1, .., x_n),\ \sum_{i=1}^{n} a_i x_i = m,\ \ x_i = \overline{1, m_i}.$$

Обратное преобразование: спектр входа X применяется к рюкзаку B, затем выполняется обратное модульное умножение.

В работе был реализован данный алгоритм, а также проведена небольшая модификация рюкзачной задачи.

Вычислительные эксперименты

В работе были проведены вычислительные эксперименты, в ходе которых проводилось сравнение скорости работы алгоритмов шифровки и дешифровки со стороны кодировщика со скоростью дешифровки со стороны взломщика.

Заключение

В работе было проведено исследование рюкзачных систем защиты информации. Были изучены основные стандартные криптосистемы: Меркла–Хеллмана, Грэма–Шамира и Шора–Ривеста. Создано программное обеспечение, реализующее работу рассмотренных криптосистем по шифровке и дешифровке данных, а также линейный метод решения задачи о рюкзаке (для сверхрастущих векторов). Были предложены собственные алгоритмы построения сверхрастущих векторов. Проведен ряд вычислительных экспериментов по сравнению скорости работы алгоритмов и зависимости от размера рюкзака.

Список литературы

1. Кормен Т. Х, Лейзерсон Ч. И., Ривест Р.Л. Алгоритмы: построение и анализ: пер. с англ. под ред. А. Шеня. М.: Вильямс, 2005. 1296 с.

2. Пападимитриу, Х. Х., Стайглиц К. Комбинаторная оптимизация. Алгоритмы и сложность: пер. с англ. В. Б. Алексеева. М.: Мир, 1985. 512 с.

3. Саломаа, А. Криптография с открытым ключом. Springer-Verlag, 1990. с. 102—150.

4. В.О. Осипян, Ю.А. Карпенко, А.С. Жук, А.Х. Арутюнян. Диофантовы трудности атак на нестандартные рюкзачные системы защиты информации. Известия ЮФУ. Технические науки, 2014. С. 209–216.

5. Харин Ю.С., Берник В.И., Матвеев Г. В. Математические основы криптологии: учебное пособие. Минск: БГУ, 1999. С. 319.

УДК 519.23

МЕТОДЫ РЕШЕНИЯ ЗАДАЧ РАСПОЗНАВАНИЯ ИЗОБРАЖЕНИЙ

Докучаев Д.А.
научный руководитель канд. физ.-мат. наук, доц. Баранова И.В.
Сибирский федеральный университет

Задача распознавания изображений относится к широкому классу задач распознавания образов. Распознавание образов – раздел статистического анализа данных, объединяющий методы разбиения объектов по нескольким категориям или классам. Объекты называются ***образами***.

Прецедент – это образ, правильная классификация которого заранее известна. Классификация основывается на прецедентах.

Будем считать, что все объекты или явления разбиты на конечное число классов. Для каждого класса известно и изучено конечное число объектов – прецедентов. Задача распознавания образов состоит в том, чтобы отнести новый распознаваемый объект к какому-либо классу.

Распознавание образов является востребованным и успешно развивающимся направлением в современной теоретической информатике и анализе данных. Его методы имеют широкий спектр применений практически во всех областях человеческой деятельности, связанных с изучением объектов и процессов: медицине, биологии, химии, маркетинге, психологии, социологии, менеджменте и других.

Целью данной работы является исследование постановки задачи распознавания образов и основных методов распознавания, а также особенностей их работы в ситуации распознавания изображений.

Задача классификации и распознавания образов

Формулировка задачи распознавания образов выглядит следующим образом: имеется множество объектов $X = \{x_1, \dots, x_n\}$, где $x: x \in X$ – объект распознавания. Тогда

$g(x): X \to M,\ M = \{1, \dots, m\}$ – индикаторная функция, разбивающая пространство образов X на m непересекающихся классов. Индикаторная функция неизвестна.

Пусть F – это пространство признаков, тогда $f(x): F \to X$ функция, ставящая в соответствие каждому объекту $x_i \in X$ точку $f(x)$. Вектор $f(x)$ называется ***образом объекта*** $x \in X$. В пространстве образов признаков определены непересекающиеся множества точек $K_i \in F, i = \overline{1,m}$, соответствующих образам одного класса.

Функция $\hat{g}(f): F \to M$ называется решающим правилом, оно представляет собой оценку для $g(x)$ на основании $f(x)$, то есть $\hat{g}(f) = \hat{g}(f(x))$.

Пусть $f_j = f(x_j), j = \overline{1,N}$ – доступная информация о функциях $f(x)$ и $g(x)$, но сами эти функции неизвестны. Тогда (g_j, f_j) – есть множество прецедентов.

Требуется построить такое решающее правило $\hat{g}(f)$, чтобы распознавание и классификация проводилось с минимальным числом ошибок.

Как уже было сказано выше, каждый объект описывается набором своих характеристик, называемых *признаками*. Признаки могут быть следующих типов:

- *бинарный* признак: $F_i = \{0,1\}$,
- *номинальный (качественный)* признак: F_i — конечное множество;
- *порядковый* признак: F_i — конечное упорядоченное множество;

- *количественный* признак: $F_i = \Re$ — множество действительных чисел.

Самой распространенной ситуацией является кластеризация объектов, у которых все признаки являются количественными, т.е. когда $F = \{F_1, F_2, \ldots, F_n\} \in \Re^n$.

Для нашей задачи распознавания изображений в качестве множества объектов берутся изображения.

Решение задачи с помощью выделения вектора характеристик

Предполагается что, каждое изображение (объект) представляется в виде набора пикселей. Пусть $X = (x_1, \ldots, x_n)$– множество всех изображений, которые поступают на распознавание, m – заданное число классов, на которое требуется разбить множество X.

Требуется задать функцию расстояния на множестве изображений:

$$\boldsymbol{\rho(x_i, x_j): X \times X \to \Re_+}$$

и произвести классификацию по классам. В качестве меры близости изображений, представленных векторами значений признаков, обычно используется скалярное произведение векторов.

Далее представим каждое изображение в виде вектора:

$$\boldsymbol{x_i = \left(r(x_i, f_j), \ldots, r(x_i, f_j)\right)},$$

где $r(x_i, f_j)$– состояние, которое сейчас принимает пиксель $f_j \in F$ в изображении x_i.

Так же каждому классу изображений соответствует вектор весов той же размерности:

$$W_i = (w_1, \ldots, w_n).$$

Необходимость такого вектора возникает из-за того, что мы распознаем каждый раз новые изображения и добавляем их в наш класс. Далее они будут использовать для сравнения. Такое действие поможет улучшить качество распознавания.

Изначально полагаем что вектор весов единичный. При добавлении нового изображения количество элементов в классе увеличивается, а веса тем самым изменяются следующим образом

$$W_i = \left(\frac{w_1}{K_i}, \ldots, \frac{w_n}{K_i}\right) \quad i = \overline{1, m},$$

где K_i- количество элементов в i классе.

Затем вводится расстояние между файлами x_i и x_j как расстояние между векторами. Чаще всего в качестве функции расстояния $\rho(x_i, x_j)$ используется евклидово расстояние:

$$\rho(x_i, x_j) = \sqrt{\sum_{m=1}^{n} (r_i^m - r_j^m)^2 * w_i}.$$

Тогда, если два изображения похожи друг с другом, то расстояние между соответствующими им векторами будет наименьшим. Таким образом, если расстояние изображений x_i и x_j наименьшее, то файл x_j будет отнесен в класс, которому принадлежит x_i.

Практическая задача распознавания и классификации

В качестве множества распознаваемых объектов в работе использовались изображения с нарисованными на них цифрами. Имелась обучающая выборка изображений – прецеденты для каждого класса. Для каждой из 10 цифр в выборке присутствовало несколько изображений: чёткие изображения и изображения с

некоторыми внесенными шумами. Каждое из этих изображений было однозначным образом отнесено к одному из классов (конкретной цифре).

В работе было создано программное обеспечение, реализующее работу рассмотренного выше метода распознавания на изображениях. Программный модуль, выполняющий работу с помощью вектора характеристик, был реализован в среде Visual Studio 2012, на языке C++.

Для начала на вход программе подавались множество файлов из обучающей выборки. После обучения на вход давалось некоторое изображение, и задачей алгоритма было определить, какая это цифра и отнести ее в определенный класс. Для тестирования алгоритмов были поданы 10 файлов на распознание. Во всех этих файлах также были изображены эти же цифры, но уже с некоторыми шумами и неточностями.

На небольшом множестве объектов алгоритм абсолютно точно определил класс изображения. Однако при увеличении шумов и неточностей в изображении алгоритм начал допускать ошибки. При увеличении объема обучающей выборки алгоритм допускал меньше ошибок. Следует отметить, алгоритм также чувствителен к качеству обучающей выборки. При плохой обучающей выборке может возникнуть задача переобучения, для улучшения результата распознавания.

Заключение

В данной работе была рассмотрена задача распознавания образов и ее частный случай – распознавание изображений. Был приведен метод решения задачи распознавания. Разработано программное обеспечение, реализующее работу данного алгоритма. Решена практическая задача распознавания изображений, а также выполнен анализ полученных результатов.

Список литературы

1. Воронцов К. В. Лекции по алгоритмам кластеризации и многомерного шкалирования. М.: МГУ, 2007. 18 с.

2. Загоруйко Н.Г. Прикладные методы анализа данных и знаний. Новосибирск: ИМ СО РАН, 1999. 270 с.

3. Лепский А.Е., Броневич А.Г. Математические методы распознавания образов. Таганрог:ТТИ ЮФУ, 2009. 155с.

4. Местецкий Л.М. Математические методы распознавания образов. М.:МГУ ВМиК, 2004. 85с.

УДК 519.213

ПРЕПРОЦЕССИНГ ДАННЫХ ДЛЯ ЗАДАЧИ СЕТ-РЕГРЕССИИ

А. И. Иванова
научный руководитель канд. физ.-мат. наук, доц. Д. В. Семенова
Сибирский федеральный университет

Для исследования функциональной зависимости между случайными множествами событий в [1, 2, 10] было предложено искать решение задачи построения регрессионного отображения (функции сет-регрессии) в виде условных сет-средних случайных множеств событий. В [4-8] сет-регрессия использовалась для моделирования потребительских предпочтений и сегментации рынка. Входными данными для задачи является совместное распределение случайных множеств событий, которое было получено из имеющейся статистики. Однако чтобы статистически оценить распределение случайного множества N событий на основе наблюдений приходится сталкиваться с проблемой хронической недостаточности имеющегося количества наблюдений для надлежащей оценки всех 2^N вероятностей $p(X)$, образующее данное вероятностное распределение. Оптимизация данных, как элемент предобработки (препроцессинга), включает снижение размерности входных данных [11]. В [9] предложен рекуррентный метод построения вероятностных распределений случайных множеств событий с помощью аппарата ассоциативных функций. Основная идея рекуррентного метода [9] — выразить вероятности пересечений множества событий функционально через вероятности самих событий, что приводит к уменьшению числа параметров, необходимых для построения вероятностных распределений случайных множеств событий. Целью данной работы является получение аналитического вида функции сет-регрессии для известных вероятностных распределений случайных множеств событий, полученных рекуррентным методом.

Рассмотрим вероятностное пространство (Ω, F, $\mathbf{P}$). Пусть $\mathsf{X} \subset F$ конечное множество событий выбранных из алгебры F этого пространства. Обозначим $N = |\mathsf{X}|$.

Определение 1. Случайное множество событий K на конечном множестве событий $U \subset F$ определяется как отображение $K : \Omega \to 2^{\mathsf{X}}$, измеримое относительно пары алгебр $\left(F, 2^{2^{\mathsf{X}}}\right)$ в том смысле, что для всякого $X \in 2^{2^{\mathsf{X}}}$ существует прообраз $K^{-1}(X) \in F$, такой что $\mathbf{P}(X) = \mathbf{P}\left(K^{-1}(X)\right)$.

Случайное множество событий K, заданное на конечном множестве событий X, определяется своим дискретным вероятностным распределением. Если мощность рассматриваемого множества событий $|\mathsf{X}| = N < \infty$, то имеется 2^N видов вероятностных зависимостей между событиями этого множества, т.е. ровно столько, сколько у этого множества подмножеств.

Определение 2. Вероятностное распределение случайного множества событий K, заданного на конечном множестве избранных событий $\mathsf{X} \subset F$ есть это набор 2^N значений вероятностной меры $\mathbf{P}$ на событиях из 2^{X}. Как известно, такое распределение можно задать шестью эквивалентными способами [2]. В настоящей работе рассматриваются только два из них:

– вероятностное распределение I-го рода случайного множества событий K на X – это набор $\{p(X), X \subseteq \mathsf{X}\}$ из 2^N вероятностей вида

$$p(X)=\mathbf{P}(K=X)=\mathbf{P}\left(\left(\bigcap_{x\in X}x\right)\cap\left(\bigcap_{x\in X^C}x^c\right)\right),\ X^c=\mathsf{X}\setminus X,\ x^c=\Omega\setminus x.$$

– вероятностное распределение II-го рода случайного множества событий K на X – это набор из 2^N вероятностей вида $\{p_X, X\subseteq\mathsf{X}\}$, где $p_X=\mathbf{P}(K\supseteq X)=\mathbf{P}\left(\bigcap_{x\in X}x\right)$, которые удовлетворяют системе из 2^N неравенств Фреше-Хёффдинга

$$0\le\max\left\{0,1-\sum_{x\in X}(1-\mathbf{P}(x))\right\}\le p_X\le\min_{x\in X}\mathbf{P}(x)\le 1.$$

Вероятностные распределения I-го и II-го рода связаны взаимно-обратными формулами обращения Мёбиуса [1]:

$$p_X=\sum_{Y\in 2^{\mathsf{X}}:\,X\subseteq Y}p(Y),\quad p(X)=\sum_{Y\in 2^{\mathsf{X}}:\,X\subseteq Y}(-1)^{|Y|-|X|}p_Y,\quad \text{для всех } X\in 2^{\mathsf{X}}.$$

Определение 3. Ассоциативная функция $AF:[0,1]^2\to[0,1]$ определяется как двуместная функция, удовлетворяющая следующим свойствам [10].

A1. Граничные условия: $AF(a,0)=AF(0,1)=0,\ AF(a,1)=AF(1,a)=a,\ \forall a\in[0,1]$.

A2.Монотонность: $\forall a_1,a_2,b_1,b_2\in[0,1]$ таких, что $a_1\le a_2, b_1\le b_2$ справедливо $AF(a_1,b_1)\le AF(a_2,b_2)$.

A3. Коммутативность: $AF(a,b)=AF(b,a),\ \forall a,b\in[0,1]$.

A4. Ассоциативность: $AF(AF(a,b),c)=AF(a,AF(b,c)),\ \forall a,b,c\in[0,1]$.

A5. Условие Липшиц-непрерывности: $AF(c,b)-AF(a,b)\le c-a, a\le c$, $a,b,c\in[0,1]$.

Если дано совместное распределение двух случайных множеств событий K и L, значения которых содержаться в конечных множествах X и Y соответственно, то регрессией K на L называется любой оператор φ, приближенно представляющий статистическую зависимость K от L. Для определения зависимости между двумя случайными множествами событий предложено использовать сет-регрессию, которая устанавливает вид средней сет-функциональной зависимости между этими двумя случайными множествами событий [1, 2, 10].

Задача сет-регрессии [4]. Пусть известны: случайное множество событий K на X; случайное множество событий L на Y; $\{p(X,Y)\,|\,X\subseteq\mathsf{X},Y\subseteq\mathsf{Y}\}$ — совместное распределение случайных множеств событий K и L на $\mathsf{Z}=\mathsf{X}+\mathsf{Y}$, $\mathsf{X}\cap\mathsf{Y}=\varnothing$, где $p(X,Y)=\mathbf{P}(K=X,L=Y)$; $\rho: 2^{\mathsf{Y}}\to 2^{\mathsf{Y}}$ — некоторая метрика в Y. Необходимо найти сет-функцию $\varphi: 2^X\to 2^Y$, которое доставляет минимум функционалу $\mathbf{E}\rho(L,\varphi(K))$, где $\mathbf{E}$ — оператор математического ожидания.

Решение задачи сет-регрессии в данной постановке подробно рассмотрено в работах [4-8]. В работе исходное совместное распределение $\{p(X,Y)\,|\,X\subseteq\mathsf{X},Y\subseteq\mathsf{Y}\}$ будем задавать с помощью рекуррентного метода [10] , используя формулу

$$\mathbf{P}\left(\bigcap_{v\in Z}v\right)=AF\left(p_v,\mathbf{P}\left(\bigcap_{w\in Z\setminus\{v\}}w\right)\right),\ \forall Z\subseteq\mathsf{Z}$$

со следующими ассоциативными функциями:

1. $AF(a,b) = a \cdot b;$
2. $AF(a,b) = \min\{a,b\};$
3. $AF(a,b) = \max\{a+b-1,0\};$
4. $AF(a,b) = \ln\left(1+\frac{(e^{-\alpha a}-1)(e^{-\alpha b}-1)}{e^{-\alpha}-1}\right)^{-\frac{1}{\alpha}}$, $\alpha \neq 0$.

Преимущество предлагаемого подхода заключается в том, что для определения вероятностного распределения вместо полного набора $2^{|Z|}$ вероятностей достаточно знать $|Z|$ вероятностей событий и вид ассоциативной функции [10].

На рис. 1 приведено решение задачи сет-регрессии с известным совместным распределением. На рис. 2, 3 приведены примеры решения задачи сет-регрессии с применением рекуррентного метода.

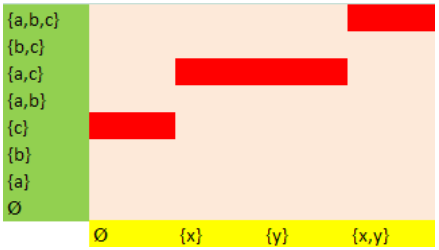

Рис. 1 Сет-регрессия в виде сет-квантиля с уровнем α=1/4, совместное распределение задано статистической оценкой.

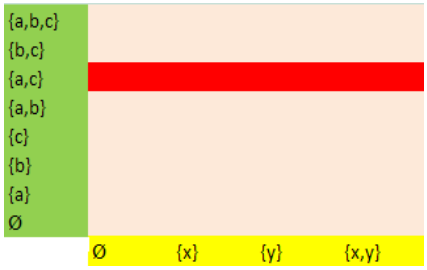

Рис. 2 Сет-регрессия в виде сет-квантиля с уровнем α=1/4, совместное распределение задано с помощью рекуррентного метода с ассоциативной функцией $AF(a,b) = a \cdot b$

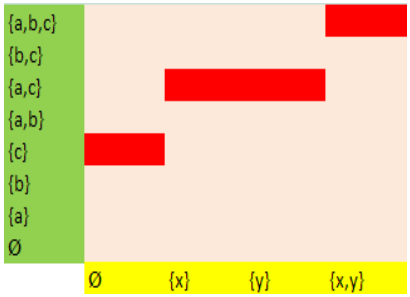

Рис. 3 Сет-регрессия в виде сет-квантиля с уровнем α=1/4, совместное распределение задано с помощью рекуррентного метода с ассоциативной функцией $AF(a,b) = \min\{a,b\}$

Список литературы

1. Воробьёв, О.Ю. Сет-регрессионный анализ зависимостей событий в статистических системах: учеб. пособие О.Ю. Воробьев, А.Ю. Фомин-Красноярск: ИВМ СО РАНб КрасГУ, 2004. 116 с.
2. Воробьёв, О.Ю. Эвентология. Красноярск: Сибирский федеральный университет, 2007. 435 с.
3. Иванова А.И. Суперпозиция сет-регрессий // Проспект Свободный-2015: материалы науч. Конф., посвященной 70-летию Великой Победы [Электронный ресурс] – Красноярск: Сиб. Федер. Ун-т, том «Математика, информатика: теория вероятностей, математическая статистика и финансово-актуарная математика», 2015. – с. 21-23.
4. Иванова А.И., Атабеков М.А. *О задаче сет-регрессии* // Труды XIV конференции по финансово-актуарной математике и эвентологии многомерной статистики, Красноярск, 2015 – С. 285-292.
5. Иванова, А.И. Эвентологическая регрессия случайных множеств событий в виде сет-средних //”Молодежь и наука”: сборник материалов X Юбилейной Всероссийской научно-технической конференции студентов, аспирантов и молодых ученых с международным участием, посвященной 80-летию образования Красноярского края, [Электронный ресурс], № заказа 1644/отв. ред. О. А. Краев — Красноярск: Сиб. федер. ун-т., 2014
6. Иванова, А.И. Эвентологические регрессии случайных множеств событий в виде условного сет-квантиля // Перспективы развития фундаментальных наук [Электронный ресурс]: труды XI Международной конференции студентов и молодых учёных. Россия, Томск, 22–25 апреля 2014 г. / под ред. Е.А. Вайтулевич. – Национальный Исследовательский Томский политехнический университет, 2014. — С. 597-599.
7. Семенова Д.В., Иванова А.И. Сравнение нечеткого и событийного подхода в задаче сегментации рынка // Новые информационные технологии в исследовании сложных структур: Материалы Десятой российской конференции с международным участием. – Томск: Издательский Дом Томского государственного университета, 2014 – с. 128
8. Семенова Д.В., Иванова А.И. Нечеткая и сет-регрессионная модели распределения потребительских предпочтений между фирмами. Труды XIII ФАМЭМС'2014 конференции (под ред. О.Ю. Воробьёва). --- Красноярск: СФУ, 2014 – С. 222-227.
9. Семенова Д.В., Лукьянова Н.А., Рекуррентное построение дискретных вероятностных распределений случайных множеств событий, ПДМ, 2014, номер 4, 47–58.
10. Тарасова, О. Ю. Сеточные и регрессионные алгоритмы аппроксимации сложных систем событий: автореф. дис. на соиск. учен. степ. канд. физ.-мат. н. по специальности 05.13.01 / Ольга Юрьевна Тарасова; Красноярск, 2007.
11. Han J., Kamber M. Data Mining: Concepts and Techniques. Elsevier Inc., 2006.

УДК 519.179

АТОМАРНОЕ РАЗЛОЖЕНИЕ ГРАФА И ЕГО ПРИМЕНЕНИЕ ДЛЯ ЗАДАЧИ О КЛИКЕ НА РАЗРЕЖЕННЫХ ГРАФАХ

Илларионов Р.Е.
научный руководитель д-р физ.-мат. наук Быкова В.В.
Сибирский федеральный университет

Рассматривается декомпозиционный подход, приводящий к ускорению работы классических алгоритмов решения оптимизационных задач на разреженных графах большой размерности. Разреженность графа выражается ограничением на древовидную ширину графа. Демонстрируется практическое применение данного подхода к задаче о клике, которая решается с помощью точного и приближенного алгоритмов.

Отличительными особенностями графов, возникающих в современных приложениях, являются их большая размерность и разреженность. Ускорение работы любого из известных алгоритмов решения оптимизационных задач на графах может быть достигнуто за счет предобработки исходного графа. Технология предобработки реализуется, как правило, в виде двухфазной процедуры. На первой фазе производится предобработка входного графа путём его редуцирования или декомпозиции. Выбор вида предобработки определяется особенностями решаемой задачи и структурой графа. На второй фазе обычно применяется известный алгоритм решения задачи к преобразованному графу и формируется решение для исходного графа. При этом требуется, чтобы предобработка существенно снижала размерность задачи, вычислительная сложность предобработки не превышала сложности решения задачи на исходном графе и обеспечивала корректность формирования решения задачи, исходя из решений, полученных для преобразованного графа.

Реализация декомпозиционного подхода, т. е. процесс выделения частей исходного графа, во многом зависит от структурных особенностей этого графа, в том числе от ограничений на параметры, характеризующие его разреженность. Один из параметров разреженности графа – ограничение на древовидную ширину графа [1, 2]. Для графов с ограниченной древовидной шириной используется разложение графа кликовыми минимальными сепараторами. Идея такого разложения была предложена Р. Тарьяном [3] как средство реализации принципа «разделяй и властвуй» для решения NP-трудных задач на графах, базирующихся на отношениях смежности вершин графа. Полученные в результате разложения части графа были названы атомами. Тарьяном было установлено, что разложение графа на атомы не разрушает клики этого графа, не порождает новых клик, и в этом смысле сохраняет его структуру.

В данной работе предлагаются алгоритмы и программы, реализующие идею Тарьяна. Демонстрируется эффективность разложения графа на атомы для NP-трудной задачи о клике, решаемой с помощью экспоненциального по времени работы алгоритма Уилфа и полиномиальной по времени жадной эвристики.

Задачу разложения графа на атомы (Decomposition of Graph into Atoms Problem, DGAP) можно сформулировать следующим образом.

Decomposition of Graph into Atoms Problem

Заданы: связный граф $G = (V, E)$ с ограниченной древовидной шириной $tw(G) \leq k$, а также множество всех его кликовых минимальных сепараторов $\Delta(G)$.

Требуется: найти для G множество атомов $\Omega(G) = \{G_1, G_2, ..., G_p\}$, где каждый атом G_i, $i = 1, \ldots, p$, определяется как максимальный связный подграф графа G, не имеющий кликовых минимальных сепараторов графа G.

Множество $\Omega(G)$ принято называть атомарным представлением (или разложением на атомы) графа G. Поясним понятия, использованные в формулировке DGAP.

Говорят, что множество $S \subseteq V$ разделяет несмежные вершины a и b графа $G = (V, E)$, если вершины a и b принадлежат различным компонентам связности графа $G(V \setminus S)$. Множество S при этом называют (a, b)-сепаратором, и минимальным (a, b)-сепаратором, если в нем нет собственного подмножества, являющегося (a, b)-сепаратором. Сепаратор S минимальный, если в G существует такая пара вершин a и b, что S является минимальным (a, b)-сепаратором. Множество $S \subseteq V$ считают кликовым минимальным сепаратором графа G, если S является в G одновременно кликой и минимальным сепаратором.

Древовидная ширина связного графа $G = (V, E)$ вычисляется через дерево декомпозиции, которое определяется как пара (M, T), задающая некоторое разбиение множества вершин и множества ребер графа G. При этом $M = \{X_i \subseteq V: i \in I\}$ – семейство подмножеств множества V, именуемых «мешками», а $T = (I, W)$ – помеченное дерево, каждому узлу которого соотносится некоторый «мешок» из M. Для всякого дерева декомпозиции $T = (I, W)$ семейство «мешков» M и множество ребер W непременно должны удовлетворять трем условиям: объединение всех «мешков» совпадает с множеством вершин V графа G; для всякого ребра графа G обязательно имеется хотя бы один «мешок», содержащий обе вершины этого ребра; для любой вершины графа G множество узлов дерева T, «мешки» которых содержат эту вершину, индуцирует связный подграф, являющийся поддеревом дерева T [2]. Всякое дерево декомпозиции графа G характеризуется шириной, значение которой вычисляется по формуле: $\max \{|X_i| - 1: \ i \in I\}$. Древовидная ширина графа G определяется как наименьшая ширина всех его деревьев декомпозиции и обозначается через $tw(G)$. Всегда $1 \le tw(G) \le n - 1$. Пусть k – некоторая заданная целая константа и $0 < k < n = |V|$. Если $tw(G) \le k$, то говорят, что граф $G = (V, E)$ обладает ограниченной (значением k) древовидной шириной. Чем меньше значение $tw(G)$, тем более разреженным является граф G, так как при $tw(G) \le k$ верна оценка $|E| \le kn - k(k + 1)/2$.

Доказано [3, 4], что задача DGAP полиномиально разрешима и имеет единственное решение. Последнее означает, что множество атомов $\Omega(G)$ уникально для G, если разложение осуществлять на основе $\Delta(G)$ – множества всех кликовых минимальных сепараторов графа G. Если в $\Delta(G)$ имеются минимальные сепараторы, не являющиеся кликами в G, то $\Omega(G)$ может зависеть как от состава $\Delta(G)$, так и от порядка просмотра сепараторов в процессе разложения. Следует заметить, что число минимальных сепараторов графа, как правило, больше числа его кликовых минимальных сепараторов, и только для хордовых графов имеет место равенство. В общем случае граф $G = (V, E)$ может иметь экспоненциальное число минимальных сепараторов относительно $n = |V|$, тогда как число кликовых минимальных сепараторов всегда не более $n - 2$ [1, 2].

Для определения уникального $\Omega(G)$ необходимо найти все кликовые минимальные сепараторы графа G. На сегодняшний день единственным эффективным методом нахождения всех кликовых минимальных сепараторов связного графа является извлечение их из минимальной триангуляции этого графа. Триангуляцией графа $G = (V, E)$ называется хордовый граф $H = (V, E')$, $E \subseteq E'$, который содержит G в качестве остовного подграфа. Триангуляция H минимальная, если для G не существует другой триангуляции, которая является собственным подграфом графа H. Любой граф G может быть преобразован в хордовый путем добавления в него дополнительных ребер (хорд). Для него всегда может быть построена некоторая минимальная (не

обязательно наименьшая) триангуляция H за полиномиальное время, при этом $tw(G) \le tw(H)$ [1, 2].

Решение DGAP сводится к выполнению следующих шагов.

Шаг 1. Нахождение для графа $G = (V, E)$ минимальной триангуляции $H = (V, E')$, $E \subseteq E'$.

Шаг 2. Поиск всех максимальных клик в триангуляции H.

Шаг 3. Построение дерева клик для триангуляции H.

Шаг 4. Установление множества минимальных сепараторов $\Delta(H)$ для триангуляции H. Вычисление для графа G множества кликовых минимальных сепараторов $\Delta(G)$ путем выделения из $\Delta(H)$ тех сепараторов, которые образуют клики в G, так что $\Delta(G) \subseteq \Delta(H)$.

Шаг 5. Формирование множества атомов $\Omega(G)$ на основе $\Delta(G)$.

Шаги 1 и 2 реализуются с помощью алгоритма MCS-M, который находит для G минимальную триангуляцию H и все максимальные клики в H, затрачивая на это время $O(n^3)$ [2]. На шаге 3 осуществляется построение дерева клик для связного хордового графа H, исходя из графа пересечения всех его максимальных клик, с применением классического алгоритма построения оптимального остова за время $O(n^2)$. Шаг 4, также как шаги 2 и 3, основан на свойствах хордовых графов [2]. Согласно этим свойствам минимальные сепараторы для H находятся через точки сочленения дерева клик, и их число не превышает $n - 2$. Кроме того, каждый минимальный сепаратор минимальной триангуляции H является минимальным сепаратором для входного графа G [4]. Значит, $\Delta(G) \subseteq \Delta(H)$. Поскольку граф H хордовый, любой сепаратор из $\Delta(H)$ образует клику в H, но необязательно клику в G. Между тем, всякий кликовый минимальный сепаратор графа G всегда в $\Delta(H)$, независимо от выбранной на шаге 1 минимальной триангуляции H. Возможен случай, когда $\Delta(G) = \varnothing$, например, это имеет место для полного графа. Непосредственное формирование $\Omega(G)$ сводится к многократному разделению графа G на части одним из $S \in \Delta(G)$, выделению компонент связности графа $G(V \setminus S)$ и копированию S в эти компоненты. Процесс разделения продолжается до тех пор, пока в полученных частях не будет кликовых минимальных сепараторов. В результате $1 \le |\Omega(G)| < n$. Время выполнения шагов 1–5 в целом составляет $O(n^3)$.

Изложенный декомпозиционный подход применим для ускорения алгоритмов решения полиномиально разрешимых и NP-трудных задач на графах, базирующихся на отношениях смежности вершин графа. Пару множеств $\Omega(G)$ и $\Delta(G)$ можно рассматривать также как форму представления разреженного графа большой размерности. Такое представление может быть предварительно создано и хранится во внешней памяти компьютера, а необходимые для обработки атомы последовательно загружаться в оперативную память. Основной недостаток атомарного представления графа: не все графы имеют кликовые минимальные сепараторы. Между тем, если граф отличен от полного, то в нем всегда имеются минимальные сепараторы. Если при разложении воспользоваться ими, уникальность $\Omega(G)$ не гарантируется, но при $tw(G) \le k$ размер частей графа, получаемых при разложении, составит $O(k)$. Будут сохранены также все клики входного графа.

Декомпозиционный подход был применен к NP-трудной задаче о наибольшей клике графа (Maximum-Clique-Problem, MCP). Для нахождения точного решения MCP использовался алгоритм Уилфа. Данный алгоритм является рекурсивным и находит решение за время $O(\text{poly}(n) \cdot 1{,}39^n)$ [5]. При двухфазной процедуре решения MCP фаза 1 (декомпозиция графа) выполняется за время $O(n^3)$, а фаза 2 (алгоритм Уилфа для каждого атома и связывание решений) – за время $O(n \cdot \text{poly}(k) \cdot 1{,}39^k)$. Таким образом, время работы двухфазной процедуры полиномиально зависит от $n = |V|$ и

экспоненциально от значения k, $tw(G) \leq k$. С помощью жадной эвристики можно найти приближенное решение МСР за время $O(n^2)$. С теоретической точки зрения такое же время потребуется и для двухфазной процедуры в этом случае.

Двухфазные алгоритмы были реализованы в виде программ и тестировались на случайных связных разреженных графах большой размерности. Вычислительные эксперименты подтвердили теоретические оценки времени работы этих алгоритмов и показали, что по сравнению с однофазным подходом (без предобработки) двухфазные алгоритмы работают значительно быстрее и порядок ускорения существенно зависит от размерности и разреженности входного графа (рис. 1).

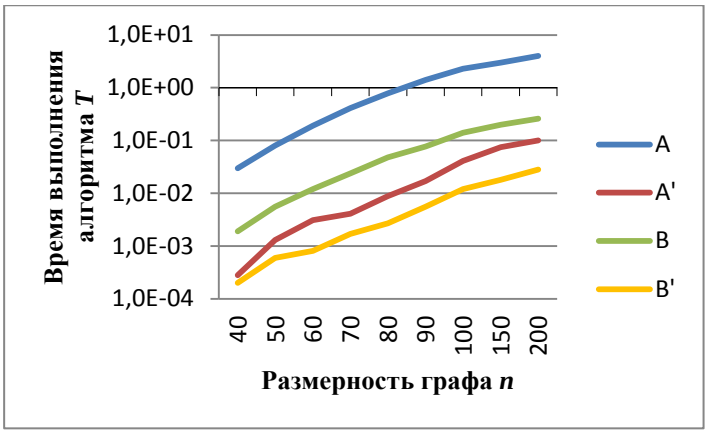

Рис. 1. Сравнение алгоритмов по времени работы:
A – алгоритм Уилфа; A′ – алгоритм Уилфа с предобработкой;
B – жадная эвристика; B′ – жадная эвристика с предобработкой

Список литературы

1. Быкова, В.В. Вычислительные аспекты древовидной ширины графа / В.В. Быкова // Прикладная дискретная математика. – 2011. – №3 (13). – С. 65–79.
2. Bodlaender, L. Discovering treewidth / L. Bodlaender // In Proc. of Conference SOFSEM. – V. 3381 of LNCS. – Springer, 2005. – P. 1–16.
3. Tarjan, R.E. Decomposition by clique separators / R.E. Tarjan // Discrete Mathematics. – 1985. – № 55. – P. 221–232.
4. Leimer, H.G. Optimal decomposition by clique separators / H.G. Leimer // Discrete Mathematics. – 1993. – № 113. – P. 99–123.
5. Wilf, H.S. Algorithms and complexity / H.S. Wilf // Taylor & Francis, 2002. – 219 p.

УДК 519.7

МАГИСТРАЛЬНАЯ МАРШРУТИЗАЦИЯ В БЕСПРОВОДНЫХ ОДНОРАНГОВЫХ СЕТЯХ НА ОСНОВЕ СВЯЗНОГО ДОМИНИРОВАНИЯ

Исхаков Р.Р.
научный руководитель д-р физ.-мат. наук Быкова В.В.
Сибирский федеральный университет

Беспроводная одноранговая сеть – это особый тип беспроводной сети, в которой сосредоточение мобильных хостов с беспроводными сетевыми интерфейсами может образовывать временную сеть без установленной инфраструктуры или централизованного администрирования. Если в беспроводной одноранговой сети участвуют только два хоста, расположенные близко друг к другу в зоне действия беспроводной сети, то никакого реального протокола маршрутизации не требуется. Однако, если два хоста, которые хотят общаться, находятся вне зон беспроводной передачи друг друга, то они могут общаться только тогда, когда другие хосты между ними готовы передавать пакеты для них.

К примеру, в сети, изображенной на рис. 1, мобильный хост *C* находится вне зоны действия беспроводного передатчика хоста *A* (отмеченного кругом вокруг *A*) и хост *A* также находится вне зоны действия беспроводного передатчика хоста *C*. Если *A* и *C* хотят обмениваться пакетами, они могут использовать хост *B* для передачи пакетов для них, так как *B* находится в пределах перекрытия зон *A* и *C*.

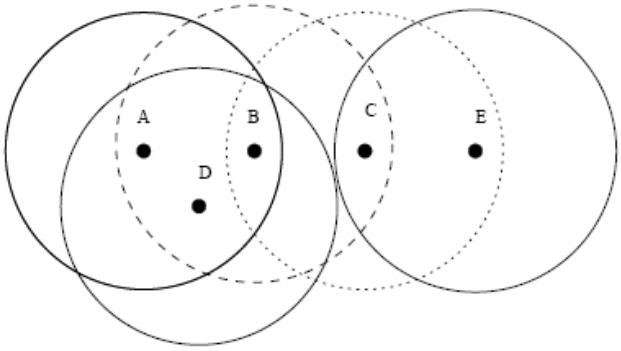

Рис. 1 – Беспроводная одноранговая сеть из пяти мобильных хостов

Беспроводную одноранговую сеть можно представить в виде неориентированного графа $G = (V, E)$, где V представляет собой множество вершин – беспроводных мобильных хостов, а E – множество ребер. Ребро между парами хостов $\{v, u\}$ обозначает, что оба хоста v и u находятся в пределах зон беспроводной передачи друг друга. Предположим, что все мобильные хосты являются гомогенными (или однородными), то есть их зоны беспроводной передачи одинаковы. Другими словами, если ребро $e = \{v, u\} \in E$, это означает, что хост u находится в пределах действия зоны хоста v, а хост v находится в пределах действия зоны хоста u. Граф на рис. 2 моделирует беспроводную одноранговую сеть, изображенную выше.

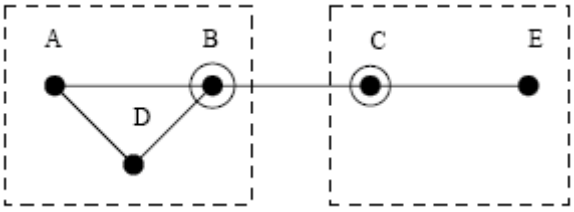

Рис. 2 – Граф беспроводной одноранговой сети, изображенной на рис. 1

Особенностями мобильных сетей являются низкая пропускная способность, мобильность хостов и низкая мощность. Беспроводные сети обеспечивают более низкую пропускную способность, чем проводные сети, поэтому сбор информации (во время формирования таблицы маршрутизации) является дорогостоящим. Мобильность хостов, которая вызывает топологические изменения базовой сети, также повышает непостоянство (изменчивость) сетевой информации. Кроме того, ограничение мощности приводит к тому, что пользователи часто отключают мобильное устройство в целях экономии потребляемой мощности. Традиционные протоколы маршрутизации, используемые в проводных сетях, не подходят для беспроводных одноранговых сетей. В среде мобильных хостов поиск новых и стабильных маршрутов после динамических изменений топологии сети может быть медленным, и этот процесс является дорогостоящим в силу низкой пропускной способности. Поэтому информация о маршрутизации должна быть локализована для быстрой адаптации к изменениям топологии сети.

Кластерная маршрутизация [19] является удобным способом маршрутизации в беспроводных одноранговых сетях. В беспроводной одноранговой сети хосты в пределах окрестности (физически близкие друг к другу) образуют кластер или клику, которая является полным подграфом. Каждый кластер имеет один или более шлюзовых хостов для связи остальных кластеров в сети. Шлюзовые хосты (из разных кластеров) обычно связаны между собой. На рис. 1 хосты *A, B* и *D* образуют один кластер, а хосты *C* и *E* образуют другой. Узлы *B* и *C* являются шлюзовыми хостами, соединенными между собой.

Следует заметить, что шлюзовые хосты беспроводной одноранговой сети образуют доминирующее множество в соответствующем модельном графе. Более того, это множество должно быть связным доминирующим множеством (Connected Dominating Set, CDS). Все подходы, использующие шлюзовые хосты для образования CDS, называются маршрутизацией на основе доминирующего множества или магистральной маршрутизацией. Главным преимуществом маршрутизации, основанной на CDS, является то, что она упрощает процесс маршрутизации, сводя его к маршрутизации в меньшей подсети, сформированной из CDS. Это означает, что для хранения информации о маршрутизации достаточно шлюзовых хостов. До тех пор, пока изменения в топологии сети не затрагивают эту подсеть, нет необходимости пересчитывать таблицы маршрутизации.

Напомним, что доминирующее множество графа $G = (V, E)$ представляет собой подмножество вершин $D \subseteq V$ такое, что каждая вершина из $V \setminus D$ является смежной по меньшей мере с одной вершиной из D. Связное доминирующее множество – это такое доминирующее множество D, которое порождает связный граф $G[D]$.

Известно, что задача нахождения наименьшего доминирующего множества, в том числе CDS, является NP-трудной для графов общего вида. Поэтому для её решения на практике чаще всего применяются приближенные или эвристические алгоритмы.

Наиболее известным из них является алгоритм маркировки. Данный алгоритм приписывает каждой вершине $v \in V$ связного неориентированного графа $G = (V, E)$ метку $m(v)$, которая равна T (*True*, вершина помечена) или F (*False*, вершина не помечена). Пусть $N(v) = \{u | \{v, u\} \in E\}$ – открытая окрестность вершины v в графе G. Тогда работу алгоритма маркировки можно описать следующим образом.

Алгоритм MarkerCDS

Вход: связный граф G.

Выход: связное доминирующее множество D заданного графа.

---

1. Изначально для каждой вершины назначается метка F.
2. Для каждой вершины $v \in V$ выполняется анализ её окрестности $N(v)$: если в $N(v)$ имеются хотя бы две несвязные вершины, то метке $m(v)$ присваивается значение T.

Множество вершин, получивших метку T, образует CDS. При этом данное множество не обязательно является минимальным, его мощность можно уменьшить с помощью следующих двух правил.

Правило 1. Если $u, v \in D$ и $N[v] \subseteq N[u]$, то изменить метку вершины v на F.

Правило 2. Предположим, что u и w являются двумя помеченными соседями помеченной вершины v в G'. Если $N(v) \subseteq N(u) \cup N(w)$ в G, то изменить метку вершины v на F.

В этих правилах $N[v] = N(v) \cup \{v\}$ – замкнутая окрестность вершины v. Данные правила реализуются за время $O(n^3)$. Алгоритм маркировки с учетом правил 1 и 2 имеет вычислительную сложность по времени $O(n^3)$.

Предлагаются две более быстрые по времени жадные эвристики для нахождения CDS: GreedCDS и GreedCDSModified. В алгоритме GreedCDS на каждом шаге предпочтение отдаётся той вершине, которая имеет наибольшую степень, то есть доминирует над наибольшим количеством смежных с ней вершин. Алгоритм работает по следующей схеме.

Алгоритм GreedCDS

Вход: связный граф G.

Выход: связное доминирующее множество D заданного графа.

---

1. Пусть D, P – пустые множества.
2. Найти в G вершину v максимальной степени.
3. Добавить найденную вершину в D и P.
4. Добавить в P все вершины из $N(v)$.
5. Найти вершину w максимальной степени среди всех вершин множества P, за исключением вершин, также принадлежащих D. Добавить эту вершину в D и P. Также добавить все вершины из $N(w)$ в P.
6. Выполнять шаги 3-5 до того момента, пока множество P не станет равным V.

По построению алгоритм GreedCDS всегда формирует CDS. Время работы данного алгоритма составляет $O(n^2)$.

Алгоритм GreedCDSModified является модификацией алгоритма GreedCDS. Данный алгоритм имеет ту же вычислительную сложность по времени, однако дает меньшее по мощности CDS.

Алгоритм GreedCDSModified

Вход: связный граф G.

Выход: связное доминирующее множество D заданного графа.

---

1. Пусть D, P – пустые множества.
2. Найти в G вершину v максимальной степени;

3. Добавить найденную вершину в D и P.
4. Добавить в P все вершины из $N(v)$.
5. Найти вершину w максимальной степени среди всех вершин множества P, за исключением вершин, также принадлежащих D. При этом степень вершины определяется специальным способом: при вычислении степени вершины не учитываются те смежные с ней вершины, которые уже находятся в P. Добавить найденную вершину w в D и P. Также добавить все вершины из $N(w)$ в P.
6. Выполнять шаги 3-5 до того момента, пока множество P не станет равным V.

С целью сравнения эффективности вышеуказанных алгоритмов была проведена серия вычислительных экспериментов для случайных графов. Для этого было сгенерировано 10 случайных графов с числом вершин 50, 100, 150, 200, 250, 300, 350, 400, 450, 500 соответственно. Результаты данной серии представлены в таблице 1.

Таблица 1 – Результаты серии экспериментов

Граф	Параметры графа: n – число вершин, m – число рёбер, δ – наименьшая степень, d – средняя степень, Δ – наибольшая степень	Результат (число доминирования / время нахождения решения, мс)		
		MarkerCDS	GreedCDS	GreedCDSModified
G_1	$n = 50$, $m = 1027$, $\delta = 37$, $d = 41$, $\Delta = 46$	3 / 218	2 / 8	2 / 8
G_2	$n = 100$, $m = 3609$, $\delta = 58$, $d = 72$, $\Delta = 83$	4 / 3437	3 / 35	3 / 36
G_3	$n = 150$, $m = 3829$, $\delta = 38$, $d = 51$, $\Delta = 67$	150 / 46588	9 / 69	6 / 44
G_4	$n = 200$, $m = 3980$, $\delta = 27$, $d = 39$, $\Delta = 55$	200 / 53639	17 / 304	10 / 153
G_5	$n = 250$, $m = 4575$, $\delta = 23$, $d = 36$, $\Delta = 52$	250 / 40850	22 / 206	13 / 122
G_6	$n = 300$, $m = 5138$, $\delta = 17$, $d = 34$, $\Delta = 51$	300 / 66024	35 / 353	18 / 209
G_7	$n = 350$, $m = 5276$, $\delta = 15$, $d = 30$, $\Delta = 47$	350 / 60074	44 / 469	22 / 278
G_8	$n = 400$, $m = 5770$, $\delta = 16$, $d = 28$, $\Delta = 43$	400 / 185929	88 / 3287	27 / 919
G_9	$n = 450$, $m = 6542$, $\delta = 11$, $d = 29$, $\Delta = 48$	450 / 221423	92 / 2090	30 / 731
G_{10}	$n = 500$, $m = 6703$, $\delta = 14$, $d = 26$, $\Delta = 39$	500 / 202258	80 / 1861	34 / 816

Из таблицы 1 видно, что алгоритм GreedCDSModified является наиболее эффективным с точки зрения времени работы и точности найденного решения.

Что касается требований к получаемому CDS, то оно должно в первую очередь обладать как можно меньшей мощностью, то есть содержать как можно меньшее количество вершин для эффективного покрытия сети минимальными ресурсами. Также очень важно, чтобы полученное CDS было отказоустойчивым, то есть при отбрасывании части рёбер (потери связи между отдельными хостами CDS) данное множество как можно дольше сохраняло свою связность. Иными словами, CDS должно быть минимально подвержено воздействию каких-либо сбоев связи в сети и её динамической топологии. Также необходимо, чтобы у полученного CDS был как

можно меньший диаметр, чтобы разброс шлюзовых хостов был минимальным и связь между ними происходила как можно проще, быстрее и эффективнее. Направление дальнейших исследований связано с учетом вышеуказанных требований к CDS.

Список литературы

1. Wu. J, H. Li. A Dominating-Set-Based Routing Scheme in Ad Hoc Wireless Networks. – Telecommunication Systems. – 2001. – V. 18 Issue 1. – P. 13 – 36.

2. Preetha K. Ga, Unnikrishnan A. Improving the Routing Performance of Mobile Ad hoc Networks Using Domination Set. – Procedia Computer Science. – 2015. – V. 46. – P. 1209 – 1215.

УДК 519.23

МЕТОДЫ ФИЛЬТРАЦИИ СПАМА

Кулистов А.В.
научный руководитель канд. физ.-мат. наук, доц. Баранова И.В.
Сибирский федеральный университет

В настоящее время электронная почта стала одним из наиболее распространенных средств связи, управления и бизнеса. Она достаточно совершенна в техническом отношении и представляет собой недорогую альтернативу привычным средствам связи. Поэтому разработка методов фильтрации потока электронной почты является одним из самых востребованных направлений в области защиты информации.

Спам (англ. *spam*) – массово распространяемые рекламные материалы или подобные коммерческие виды сообщений лицам, не выражавшим желания их получать.

В общепринятом значении термин в русском языке впервые стал употребляться применительно к рассылке электронных писем. Доля спама в мировом почтовом трафике составляет от 60% до 80%.

Целью работы является исследование и реализация основных алгоритмов решения задачи фильтрации спама. В работе рассматриваются следующие методы фильтрации спама: байесовский метод и классификация на основе «мешка слов».

Задача фильтрации спама

Фильтрация спама является разновидностью задачи классификации текстов, поэтому для решения нашей задачи будем использовать модель задачи классификации.

Постановка задачи классификации текстов

Пусть имеется $D = \{d_1, d_2, \ldots, d_m\}$ – множество текстовых документов с набором признаков $P = \{P_1, P_2, \ldots, P_n\}$ (т.е. каждый текстовый документ $d_i = (w_i^1, w_i^2, \ldots, w_i^n)$) и задана функция расстояния (метрика) между объектами $\rho = (d_i, d_j), d_i, d_j \in c.$

Задано множество классов $C = \{C_1, C_2, \ldots, C_k\}, k \le m,$ которое представляет собой разбиение множества объектов такое, что класс $C_i \cap C_j = \varnothing.$ В нашей задаче $k = 2$.

Функцией классификации называется функция $f : D \to Y$, которая любому объекту $d \in D$ ставит в однозначное соответствие номер кластера $y \in Y = \{1, \ldots, k\}, k \le m.$ Кластер $C_i = \{d \in D,\ f(d) = i\}.$

Задано конечное множество объектов $\tilde{D} \subset D$, для которых известно, к каким классам они относятся. Это множество называется ***обучающей выборкой***. Классовая принадлежность остальных объектов не известна.

Тогда постановку задачу классификации данных можно сформулировать следующим образом:

Требуется найти такую функцию f^*, чтобы

$$Q(f^*, C, \rho) = \min_f Q(f, C, \rho),$$

где $Q(f, C, \rho)$ – выбранный критерий качества классификации. В качестве критерия используется похожесть слов между собой.

Наиболее часто в задачах классификации или кластеризации текстовых данных используется байесовский классификатор.

Решение задачи фильтрации с помощью байесовского классификатора

Данная модель классификации базируется на понятии условной вероятности принадлежности документа *d* классу *c*.

В основе наивного байесовского классификатора лежит теорема Байеса:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}.$$

которая позволяет вычислить условную вероятность $P(A|B)$ того, что имело место событие A, если в результате эксперимента наблюдалось событие B, при известных вероятностях наступления событий A и B – $P(A|B)$ и $P(B)$, соответственно, и условной вероятности наступления события A при существующем B – $P(B|A)$.

Для данной модели документ – это вектор: $d = \{w_1,...,w_n\}$, где w_i - вес *i*-ого термина, а n – размер словаря выборки. Признаками документа считаем слова, содержащиеся в нем. Согласно теореме Байеса, вероятность принадлежности документа *d* к классу *c* вычисляется следующем образом:

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)},$$

где $P(c|d)$ – вероятность, что документ *d* принадлежит классу *c*, именно ее нам надо рассчитать; $P(d|c)$ – вероятность встретить документ *d* среди всех документов класса *c*; $P(c)$ – безусловная вероятность встретить документ класса *c* в корпусе документов; $P(d)$ – безусловная вероятность встретить документ *d* в корпусе документов.

Цель классификации состоит в том, чтобы понять к какому классу принадлежит документ.

Наиболее вероятным классом c^*, к которому принадлежит документ *d*, является тот класс, для которого условная вероятность принадлежности документа *d* классу с максимальна:

$$c^* = \max_c P(c/d).$$

Необходимо вычислить вероятность для всех классов и выбрать тот класс, для которого вероятность имеет максимальное значение. По теореме Байеса:

$$c^* = \max_c P(c/d)P(c).$$

Так как $d = \{w_1,...,w_n\}$, то $c^* = \max_c \frac{P(w_1,w_2,...w_m/c)P(c)}{P(w_1,w_2,...w_m/c)}$.

Далее делается существенное допущение, которое и объясняет, почему этот алгоритм называют наивным. Оно звучит следующим образом: знаменатель может быть опущен, так как для одного и того же документа *d* вероятность $P(d)$ будет одинаковой, а это значит, что ее можно не учитывать:

$$c^* = \max_c P(w_1, w_2, ... w_m/c)P(c).$$

Также в модели наивного байесовского классификатора предполагается, что все признаки $w_1, w_2, \dots, w_n$ документа *d* независимы друг от друга. Также вносится уточнение, что позиция термина в предложении не важна.

Таким образом, условную вероятность $P(w_1,...,w_n|c)$ для признаков $w_1, w_2, \dots, w_n$ можно представить в следующем виде:

$$P(w_1|c)*(w_2|c)*...*(w_n|c) = \prod_i P(w_i|c_j).$$

Таким образом, для нахождения наиболее вероятного класса для документа $d = \{w_1,...,w_n\}$ с помощью наивного байесовского классификатора, необходимо вычислить условные вероятности принадлежности документа d для каждого из представленных классов и выбрать класс, имеющий максимальную вероятность:

$$c_{NB} = \max_c \left[P(c_j) \prod_i P(w_i|c_j) \right].$$

Вероятность каждого класса можно оценить частотой встречаемости документов этого класса в обучающей выборке: $P(c_j)$ оценивается отношением количества документов класса j в обучающей выборке к общему количеству документов в выборке:

$$P(c) = \frac{D_c}{D},$$

где D_c – количество документов класса c, а D – общее количество документов в выборке.

Для оценки условных вероятностей встречаемости слов $\hat{P}(w_i|c_j)$ используется отношение количества терминов в классе c_j общему количеству терминов в этом классе.

Решение задачи с помощью модели «мешка слов»

Пусть $D = \{d_1, d_2, \ldots, d_m\}$ – множество всех текстовых документов, k – заданное число кластеров, на которое требуется разбить множество D.

Требуется задать функцию расстояния на множестве документов:

$$\rho(d_i, d_j): D \times D \to \Re_+$$

и провести классификацию текстовой коллекции.

Перед проведением кластеризации проводится предварительная обработка текстов – из них удаляются *стоп-слова* и слова, встречающиеся не более одного раза в тексте, как шумовые составляющие.

Стоп-слова − слова, знаки или символы, которые не несут самостоятельной смысловой нагрузки и просто игнорируются поисковыми системами при осуществлении ранжирования или индексации сайтов. К ним относят союзы и союзные слова, местоимения, предлоги, частицы, междометия, указательные слова, цифры, знаки препинания, вводные слова, ряд некоторых существительных, глаголов и наречий.

Далее каждому преобразованному документу ставится в соответствие следующий вектор:

$$\mathbf{d_i} = \left(n(d_i, \omega_1)...n(d_i, \omega_j)...n(d_i, \omega_W) \right),$$

где $n(d_i, \omega_j)$ – число вхождений слова $\omega_j \in W$ в текст d_i.

Затем вводится расстояние между документами d_i и d_j как расстояние между векторами. В работе в качестве функции расстояния $\rho(d_i, d_j)$ используется евклидово расстояние:

$$\rho(d_i, d_j) = \sqrt{\sum_{k=1}^{n} \left(d_i^k - d_j^k \right)^2}.$$

Если два текста близки друг другу, то расстояние между соответствующими им векторами будет наименьшим.

Практическая задача фильтрации спама

В работе было создано программное обеспечение, реализующее работу обоих рассмотренных методов фильтрации спама.

Имелась обучающая выборка писем, состоящая из обычных и спам-писем.

На вход программе подавалось новое письмо, и задачей алгоритма было установить, является это письмо спамом или нет. Для тестирования работы алгоритмов были выбраны три письма. Первое письмо являлось спамом, второе – нет. Третье письмо не являлось спамом, хотя и содержало некоторые слова, часто встречающиеся в спаме.

Число совпавших слов было мало, так как размер обучающей выборки был небольшим. В дальнейшем предполагается увеличить обучающую выборку, для выявления большего числа совпадений.

Заключение

В данной работе была рассмотрена одна из задач классификации данных – задача фильтрации спама. Были рассмотрены и реализованы два метода решения данной задачи. А также решена практическая задача фильтрации спама.

Список литературы

1. Воронцов К. В. Лекции по алгоритмам кластеризации и многомерного шкалирования. М.: МГУ, 2007. 18 с.

2. Миркин Б.Г. Методы кластер – анализа для поддержки принятия решений: обзор: препринт WP7.2011.03. М.: Изд. дом национального исследовательского университета «Высшая школа экономики», 2011. 88 с.

3. Загоруйко Н.Г. Прикладные методы анализа данных и знаний. Новосибирск: ИМ СО РАН, 1999. 270 с.

4. Дюран Б, Одел П. Кластерный анализ: под ред. А.Я. Боярского. М.: «Статистика», 1977. 128 с.

УДК 519.7

ИССЛЕДОВАНИЕ ТЕКСТОВ ТУВИНСКОГО ФОЛЬКЛОРА НА ОСНОВЕ АНАЛИЗА ФОРМАЛЬНЫХ ПОНЯТИЙ

Монгуш Ч.М.
научный руководитель д-р физ.-мат. наук Быкова В.В.
Сибирский федеральный университет

В настоящее время активно создаются корпусы естественных языков с помощью современных информационных технологий и методов математического моделирования. Под корпусом понимается информационно-справочная система, основанная на собрании оцифрованных текстов. Корпус включает в себя различные типы письменных и устных текстов, представленных в данном языке, а также разметку – информацию о свойствах этих текстов. Разметка отличает корпус от электронных библиотек текстов [5]. Традиционно в корпусах используются следующие типы разметки: метатекстовая, морфологическая, синтаксическая, семантическая и др. Чем богаче и разнообразнее разметка, тем выше научная и учебная ценность корпуса. На основании корпусов решаются многие филологические и лингвистические задачи. Для многих языков народов Российской Федерации, в том числе для тюркских языков, создаются национальные корпусы. Работа над формированием Национального корпуса тувинского языка ведется преподавателями, аспирантами и студентами Тувинского государственного и Сибирского федерального университетов [1, 5].

Важнейшей частью поискового аппарата всякого корпуса является метаразметка. Под метаразметкой понимается приписывание тексту параметров, характеризующих текст в целом. Информация, отражающая метаразметку, как правило, хорошо структурирована и допускает представление ее в виде матрицы «объект-признак», называемой контекстом. Наличие такой информации позволяет применять в корпусах математический аппарат теории распознавания образов и анализа формальных понятий (АФП).

АФП – современный математический метод анализа данных [3]. Основные идеи АФП были сформулированы в работах Р. Вилле и Б. Гантера. АФП позволяет выявлять и исследовать формальные понятия исходя из контекста. Данный метод нашел широкое применение в различных областях машинного обучения [2–4]. С помощью АФП и методов на его основе можно выделять наборы формальных понятий и решать задачи бинарной классификации по положительным и отрицательным прецедентам, возникающие при исследовании корпусов, включая следующие филологические задачи: принадлежность к определенному жанру, тематике и пространственно-временному периоду, исследование текста с целью установления автора (сказителя). Существуют различные алгоритмы классификации на основе АФП. К ним относятся: алгоритмы, основанные на использовании всей решетки понятий или некоторого ее подмножества, и алгоритмы, основанные на гипотезах. Достоинством этих алгоритмов является наглядное и удобное для изучения представления результатов в виде решеток. Основные недостатки указанных алгоритмов: экспоненциальная сложность построения решетки и высокая доля отказов от классификации.

В данной работе решается задача бинарной классификации на основе АФП применительно к произведениям тувинского фольклора, с целью определения принадлежности к жанру героического эпоса. Для классификации используется подход, основанный на гипотезах. Предлагаются алгоритмы и программы формирования формальных понятий для заданного контекста, построения решеток формальных понятий, выявления гипотез и реализации правил классификации.

Пусть задано конечное множество объектов G, разделенное на два класса G^+ и G^-, $G^+ \cap G^- = \varnothing$, $G^+ \cup G^- = G$. Такое разбиение определено с помощью некоторой обучающей выборки и целевого признака z. Элементы множеств G^+ и G^- называют положительными и отрицательными прецедентами соответственно. Все объекты из G описаны через конечное множество признаков M. Это описание задается объектно-признаковой таблицей T. Элементы таблицы T определяют степень выраженности определенного признака в данном объекте. В частном случае таблица T может быть представлена (0, 1)-матрицей, кодирующей наличие или отсутствие признаков. Пусть задан объект $x \notin G$, описанный также через множество признаков M. Требуется, исходя из таблицы T, для объекта x определить класс, к которому может быть отнесен объект x.

Решающее правило определения класса объектов должно обладать следующими свойствами: приводить к отказу от классификации, когда вопрос о принадлежности объекта x к тому или иному классу остается открытым; обладать свойством оптимальности по отношению к некоторому критерию, определяющему качество классификации. В качестве критерия качества классификации для филологических и лингвистических задач может служить экспертная оценка.

Формализуем задачу бинарной классификации по прецедентам в терминах АФП. Пусть I – отношение инцидентности между множествами G и M. Запись $g\ I\ m$ означает, что объект $g \in G$ обладает признаком $m \in M$. Очевидно, что I однозначно определяется матрицей T. Тройка $K = (G, M, I)$ называется формальным контекстом для множества объектов G.

Для произвольных подмножеств A и B, где $A \subseteq G$, $B \subseteq M$, введем два отображения φ: $2^G \to 2^M$ и ψ: $2^M \to 2^G$ такие, что $\varphi(A) = \{m \in M \mid \forall g \in A\ (g\ I\ m)\}$, $\psi(B) = \{g \in G \mid \forall\ m \in B\ (g\ I\ m)\}$, где 2^G – всевозможные подмножества множества G, 2^M – всевозможные подмножества множества M. Пусть $A' = \varphi(A)$ – множество признаков, которыми обладают все объекты из множества A и $B' = \psi(B)$ – множество объектов, которые обладают всеми признаками из множества B. Пара множеств (A, B), $A \subseteq G$, $B \subseteq M$, таких, что $A' = B$ и $B' = A$, называется формальным понятием контекста K. При этом A называется объемом, а B – содержанием формального понятия (A, B).

Если контекст $K = (G, M, I)$ представлен в виде (0, 1)-матрицы T, то формальному понятию (A, B) соответствует максимальная подматрица матрицы T, заполненная единицами. Это означает, что подмножество объектов B обладает всеми признаками из B, и наоборот всякий признак из B характерен для всех объектов из A. Далее полагаем, что T является (0, 1)-матрицей.

Пусть $A_1, A_2 \subseteq G$ и $B_1, B_2 \subseteq M$. Формальные понятия (A_1, B_1) и (A_2, B_2) связаны отношением частичного порядка $(A_1, B_1) \leq (A_2, B_2)$, если $A_1 \subseteq A_2$ (что эквивалентно $B_2 \subseteq B_1$). Частично упорядоченное по вложению объемов множество формальных понятий контекста K называется решеткой понятий контекста K и обозначается $\beta(K)$.

В терминах АФП элементы классов G^+ и G^- могут быть описаны с помощью двух контекстов по отношению к целевому признаку z: положительного контекста $K^+ = (G^+, M, I^+)$, отрицательного контекста $K^- = (G^-, M, I^-)$.

Определим положительные и отрицательные гипотезы для классификации по целевому признаку z. Гипотезой называется некоторый набор признаков, который присутствует в описании объектов одного класса и не присутствует в описании объектов из другого класса. Гипотезы предназначены для определения класса, к которому может быть отнесен объект $x \notin G$. Согласно методу АФП гипотезы извлекаются из решеток формальных понятий. Если содержание B^+ положительного понятия $(A^+, B^+) \in K^+$ не является подмножеством содержания какого-либо отрицательного понятия, то оно называется положительной гипотезой, иначе оно

называется фальсифицированной положительной гипотезой. Аналогично для отрицательных гипотез.

Пусть $M_x \subseteq M$ – множество признаков (содержание) объекта $x \notin G$. Тогда простейшее решающее правило бинарной классификации для объекта x можно сформулировать следующим образом: объект x относится к классу G^+, если множество M_x включает, хотя бы одну положительную гипотезу и не включает ни одной отрицательной гипотезы, в противном случае относится к классу G^-; отказ от классификации происходит, если M_x не содержит в качестве подмножеств ни положительных, ни отрицательных гипотез, и если M_x содержит как положительные, так и отрицательные гипотезы.

Для решения задачи бинарной классификации были разработаны алгоритмы и программы, осуществляющие для заданного контекста формирование формальных понятий, построение решеток формальных понятий, выявление гипотез и реализацию правил классификации по положительным и отрицательным прецедентам. В качестве исходных данных используются: положительный контекст $K^+ = (G^+, M, I^+)$ в виде (0, 1)-матрицы, отрицательный контекст $K^- = (G^-, M, I^-)$ в виде (0, 1)-матрицы, множество M_x – признаковое описание объекта $x \notin G$. Результатом работы программ являются: решетки положительных и отрицательных формальных понятий $\beta(K^+)$, $\beta(K^-)$, решение задачи классификации в виде указания класса принадлежности объекта x или отказа от классификации. Программы реализованы в визуальной среде разработки Delphi 9. Ниже приводится пример применения АФП и разработанных программ для решения отдельной филологической задачи – определение принадлежности произведения тувинского фольклора к жанру героического эпоса.

Рассмотрим множество из семи произведений тувинского фольклора, разбитое на два класса G^+ и G^- по целевому признаку z = «произведение относится (не относится) к жанру героического эпоса». Требуется для нового произведения x определить класс. В таблице 1 представлен результат операции шкалирования объектно-признаковой матрицы для восьми произведений тувинского фольклора [1].

Таблица 1 – Контекст K произведений тувинского фольклора

№	s_1	s_2	s_3	s_4	a_1	a_2	g_1	g_2	c_1	c_2	c_3	t_1	t_2	t_3	z
1			1		1		1		1			1			+
2	1					1	1		1				1		+
3		1			1		1			1				1	+
4			1		1		1			1		1			+
5		1			1		1			1		1			**?**
6				1		1		1			1		1		–
7				1	1		1		1				1		–
8				1		1		1			1			1	–

Для положительного контекста $K^+ = (G^+, M, I^+)$ имеем $G^+ = \{1, 2, 3, 4\}$, $M = \{s_1, s_2, s_3, s_4, a_1, a_2, g_1, g_2, c_1, c_2, c_3, t_1, t_2, t_3\}$, а соответствие I^+ представлено верхней частью таблицы 1. Для отрицательного контекста $K^- = (G^-, M, I^-)$ имеем $G^- = \{6, 7, 8\}$, а соответствие I^- представлено нижней частью таблицы 1.

Сначала определим содержание с одноэлементными объемами положительных формальных понятий: $\{1\}^+ = \{s_3, a_1, g_1, c_1, t_1\}$, $\{2\}^+ = \{s_1, a_2, g_1, c_1, t_2\}$, $\{3\}^+ = \{s_2, a_1, g_1, c_2, t_1\}$, $\{4\}^+ = \{s_3, a_1, g_1, c_2, t_1\}$. Затем, рассматривая всевозможные попарные пересечения содержаний построенных элементарных понятий, а также вложенность содержаний, получим новые положительные формальные понятия для K^+: $(\{1,2\}, \{g_1, c_1\})$, $(\{1,4\}, \{s_3, a1, g_1, t_1\})$, $(\{3,4\}, \{a_1, g_1, c_2\})$. Отношение частичного порядка между установленными положительными формальными понятиями

описываются решеткой $\beta(K^+)$, представленной на рисунке 1. Отрицательные формальные понятия и решетка $\beta(K^-)$ для контекста K^- строится аналогичным образом.

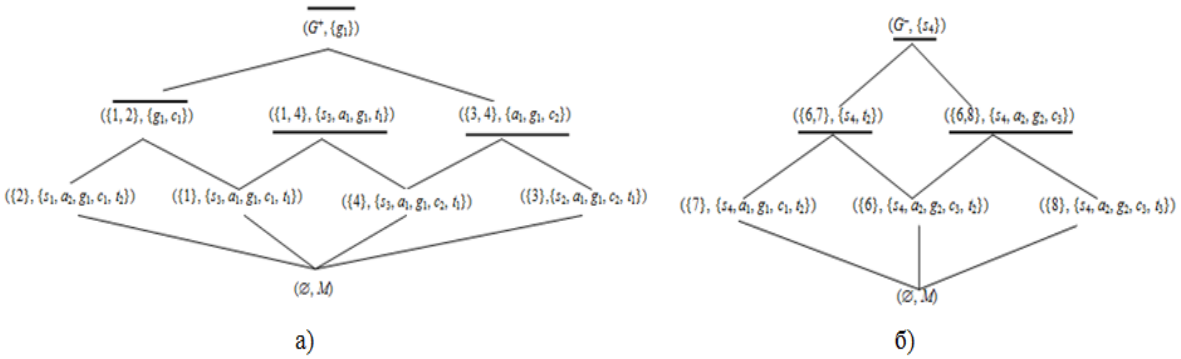

Рис. 1 – а) Решетка положительных формальных понятий $\beta(K^+)$. б) Решетка отрицательных формальных понятий $\beta(K^-)$

Для рассматриваемого целевого признака z положительными гипотезами являются $\{s_3, a_1, g_1, t_1\}$, $\{a_1, g_1, c_2\}$, они подчеркнуты в $\beta(K^+)$. Надчёркнуты в $\beta(K^+)$ элементы, соответствующие положительным фальсифицированным гипотезам: $\{g_1, c_1\}$, $\{g_1\}$. Отрицательными гипотезами являются $\{s_4, t_2\}$, $\{s_4, a_2, g_2, c_3\}$, $\{s_4\}$, они в $\beta(K^-)$ подчеркнуты. В соответствии с правилом бинарной классификации получаем: заданный объект x с содержанием $M_x = \{s_2, a_1, g_1, c_2, t_1\}$, будет отнесен классу G^+, т. е. к жанру героического эпоса, так как M_x включает положительную гипотезу $\{a_1, g_1, c_2\}$ и не содержит отрицательных гипотез. Если $M_x = \{s_3, a_1, g_1, t_1\}$, то объект x также будет отнесен к классу G^+.Однако если x обладает набором признаков $\{s_4, t_2\}$ или $\{s_4, a_2, g_2, c_3\}$, $\{s_4\}$, то этот объект будет отнесен к классу G^-. При $M_x = \{g_1, c_1\}$ или $M_x = \{g_1\}$ произойдет отказ от классификации. Экспертами установлено, что полученные результаты классификации, исходя из семантики признаков $M = \{s_1, s_2, s_3, s_4, a_1, a_2, g_1, g_2, c_1, c_2, c_3, t_1, t_2, t_3\}$, соответствуют действительности, т. е. являются филологически правильными.

В данной работе показано применение метода АФП для исследования текстов тувинского фольклора. Метод АФП сохраняет объектно-признаковое описание произведений, позволяет выделять произведения с общими признаками, строит отношение между ними и классифицирует неопределенные произведения. Поэтому данный метод подходит для решения филологических задач в рамках корпуса тувинского языка. Недостаток метода АФП – размер решетки формальных понятий экспоненциально зависит от числа объектов и признаков. Поэтому процесс выявления формальных понятий и построения решетки – это вычислительно сложная задача. Дальнейшие исследования будут направлены на решение указанных проблем, а также на разработку программы автоматического формирования контекста на основе корпуса тувинского языка.

Список литературы

1. Бавуу-Сюрюн М. В. Тувинский язык на современном этапе [электронный ресурс]. Режим доступа: http://www.tuva.asia/journal/issue_7/2158-bavyy-suyruyn-mv.html

2. Башеева А. О., Сатекбаева А. Ж. Использование решеток формальных понятий // Вестник ЕНУ им. Л. Н. Гумилева, 2012. № 4. С. 51 – 57.

3. Гуров С. И., Онищенко А. А. Классификация на основе АФП и бикластеризации: возможности подхода // Прикладная математика и информатика: труды факультета Вычислительной математики и кибернетики, 2011. Т. 38. С. 77–87.

4. Прокашева О. В. Повышение эффективности алгоритма классификации на основе анализа формальных понятий // Машинное обучение и анализ данных, 2013. Т.1. № 5. С. 552 – 558.

5. Салчак А. Я., Байыр-оол А. В. Электронный корпус тувинского языка: состояние, проблемы // Мир науки, культуры, образование, 2013. № 6. С. 408–409.

УКД 519.213

ПРИМЕНЕНИЕ БАЙЕСОВСКОЙ СЕТИ В ЗАДАЧЕ КРЕДИТНОГО СКОРИНГА

Огородникова А.В.
научный руководитель канд. физ.-мат. наук, доц. Семенова Д.В.
Сибирский федеральный университет

Вероятностный подход к представлению знаний состоит в том, что область знаний представляется совокупностью переменных и совместным распределением их вероятностей или его приближением. Это позволяет вычислить или оценить условное распределение вероятностей части переменных (гипотезы), когда некоторые из остальных (наблюдение) приняли определенное значение. На основании сведений о распределении вероятностей можно:

– определить ожидаемые значения ненаблюдаемых переменных и их совокупностей;

– подтвердить или отвергнуть гипотезы о событиях (о том, что переменные приняли совокупность значений);

– верифицировать (обнаружить противоречия или убедиться в их отсутствии) значения переменных, составивших наблюдение.

Графическая вероятностная модель — это вероятностная модель, в которой в виде графа представлены зависимости между случайными величинами. Байесовская сеть представляет случай графической модели с ориентированным ациклическим графом, при этом ориентированные ребра кодируют отношения вероятностной зависимости между переменными. Вершины графа соответствуют случайным переменным, а ребра — непосредственным вероятностным взаимосвязям между случайными величинами. Графические модели широко используются в теории вероятностей, статистике (особенно в Байесовской статистике), а также в машинном обучении. Так называемая наивная классификация или наивно-байесовский подход является наиболее простым вариантом метода, использующего байесовские сети. При этом подходе решаются задачи классификации, результатом работы метода являются так называемые "прозрачные" модели. Фактически наивный классификатор Байеса можно рассматривать как классификатор, основанный на байесовской сети, структура которой содержит связи от переменной класса к переменным атрибутам

Рассмотрим X — множество объектов, Y — конечное множество имён классов, множество $X \times Y$ является вероятностным пространством с плотностью распределения $p(x, y) = P(y)p(x \mid y)$. Вероятности появления объектов каждого из классов $P_y = P(y)$ называются априорными вероятностями классов. Плотности распределения $p_y(x) = p(x \mid y)$ называются функциями правдоподобия классов. Вероятностная постановка задачи классификации разделяется на две независимые подзадачи:

Задача 1. Имеется простая выборка $X^l = (x_i, y_i)_{i=1}^l$ из неизвестного распределения $p(x, y) = P_y p_y(x)$. Требуется построить эмпирические оценки [2] априорных вероятностей $\hat{P}_y$ и функций правдоподобия $P(s) \ni P(f)$ для каждого из классов $y \in Y$.

Задача 2. По известным плотностям распределения $p_y(x)$ и априорным вероятностям P_y всех классов $y \in Y$ построить алгоритм $a(x)$, минимизирующий вероятность ошибочной классификации.

Вторая задача решается относительно легко. Поскольку знание функций правдоподобия позволяет находить вероятности событий вида «$x \in W$ при условии, что x принадлежит классу y»:

$$P(W \mid y) = \int_W p_y(x)dx \ , \ W \subset X.$$

Рассмотрим произвольный алгоритм $a: X \to Y$. Он разбивает множество X на непересекающиеся области $A_y = \{x \in X \mid a(x) = y\}$, $y \in Y$. Вероятность того, что появится объект класса y и алгоритм a отнесёт его к классу s, равна $P_y P(A_s \mid y)$. Каждой паре $(y, s) \in Y \times Y$ поставим в соответствие величину потери λ_{ys} при отнесении объекта класса y к классу s. Обычно полагают $\lambda_{yy} = 0$, и $\lambda_{ys} > 0$ при $y \neq s$. Соотношения потерь на разных классах, как правило, известны заранее.

Теперь дадим определение функционала среднего риска. Функционалом среднего риска называется ожидаемая величина потери при классификации объектов алгоритмом a:

$$R(a) = \sum_{y \in Y} \sum_{s \in Y} \lambda_{ys} P_y P(A_s \mid y).$$

Если величина потерь одинакова для ошибок любого рода, $\lambda_{ys} = [y \neq s]$, то средний риск $R(a)$ совпадает с вероятностью ошибки алгоритма a.

Теорема [4] о достижении минимума среднего риска утверждает, что если известны априорные вероятности P_y и функции правдоподобия $p_y(x)$, то минимум среднего риска $R(a)$ достигается алгоритмом

$$a(x) = \arg \min_{s \in Y} \sum_{y \in Y} \lambda_{ys} P_y p_y(x) \ . \qquad 1)$$

Также часто можно полагать, что величина потери зависит только от истинной классификации объекта, но не от того, к какому классу он был ошибочно отнесён. В этом случае формула оптимального алгоритма упрощается. Известно [4], что если P_y и $p_y(x)$ известны, $\lambda_{yy} = 0$ и $\lambda_{ys} \equiv \lambda_y$ для всех $y, s \in Y$, то минимум среднего риска достигается алгоритмом:

$$a(x) = \arg \max_{y \in Y} \lambda_y P_y p_y(x). \qquad 2)$$

Таким образом, теоремы, изложенные выше, позволяют полностью найти вид решения задачи 2. Выражение (1) также называют байесовским решающим правилом. Минимальное значение среднего риска $R(a)$, достигаемое байесовским решающим

правилом, аналогично называют байесовским риском или байесовским уровнем ошибки.

В свою очередь, первая задача имеет множество решений, поскольку многие распределения $p(x, y)$ могли бы породить одну и ту же выборку X^l. Приходится привлекать различные предположения о плотностях, что и приводит к большому разнообразию байесовских методов: таких как наивный байесовский классификатор, непараметрические оценки плотности, а также, оценка при помощи случайных множеств событий.

Одна из типовых и применяемых задач классификации является задача оценивания заёмщиков, которая решается банками при выдаче кредитов. Объектами в данном случае являются заёмщики — физические или юридические лица, претендующие на получение кредита. В случае физических лиц признаковое описание состоит из анкеты, которую заполняет сам заёмщик, и, возможно, дополнительной информации, которую банк собирает о нём из собственных источников. Обучающая выборка составляется из заёмщиков с известной кредитной историей. В простейшем случае принятие решений сводится к классификации заёмщиков на два класса: «хороших» и «плохих». Кредиты выдаются только заёмщикам первого класса. В более сложном случае оценивается суммарное число баллов заёмщика, набранных по совокупности информативных признаков. Чем выше оценка, тем более надёжным считается заёмщик. Отсюда и название — кредитный скоринг. На стадии обучения производится синтез и отбор информативных признаков и определяется, сколько баллов назначать за каждый признак, чтобы риск принимаемых решений был минимален.

Приведем формулировку этой задачи в терминах теории вероятностей.

Пусть дано некоторое множество объектов $X = \{x_i\}, i = 1,...,N$ и множество ответов Y, такое что совокупность пар $(x_i, y_i)_{i=1}^N$ характеризует набор ответов x_i клиента на вопросы анкеты. Также дано множество признаков $f_j, j = 1,...,M$, таких что $P(x_i, f_j)$№0. Требуется найти алгоритм $a: X \to Y$, разбивающий множество X на непересекающиеся области $A_s = \{x \in X \mid a(x) = s\}$, где s — некоторая характеристика, такая, что $P(s)$З $P(f_j)$№$P(s, f_j)$.

Таким образом, зная априорные вероятности P_y и функции правдоподобия $p_y(x)$, решение данной задачи можно найти с помощью методов (1), (2).

Также, остается открытой проблема обучения Байесовских сетей. Основными методами обучения на сегодняшний день являются метод описания минимальной длины (ОМД) и метод Купера и Гершаковича (КГ) [1]. В данной работе мы будем использовать метод ОМД, поскольку метод КГ требует значительных вычислительных затрат на больших выборках.

Принцип ОМД в своей нестрогой и наиболее общей формулировке гласит: среди множества моделей следует выбрать ту, которая позволяет описать данные наиболее коротко без потери информации [1].

В общем виде задача ОМД выглядит следующим образом. Сначала задается множество обучающих данных $D = \{d_1, \ldots, d_n\}$, $d_i = \{x_i^{(1)}\ x_i^{(2)} \ldots x_i^{(N)}\}$ (нижний индекс – номер наблюдения, а верхний – номер переменной), n – количество наблюдений, каждое наблюдение состоит из N $(N \geq 2)$ переменных

$X^{(1)}, X^{(2)}, \ldots, X^{(N)}$, каждая $j-$я переменная $\left(j = 1, \ldots, N\right)$ имеет $A^{(j)} = \left\{0, 1, \ldots, \alpha^{(j)} - 1\right\}$ $\left(\alpha^{(j)} \geq 2\right)$ состояний, каждая структура $g \in G$ байесовской сети представляется N множествами предков $\left(\Pi^{(1)}, \ldots, \Pi^{(N)}\right)$, то есть для каждой вершины $j = 1, \ldots, N$, $\Pi^{(j)}$ – это множество родительских вершин, такое что $\Pi^{(j)} \subseteq \left\{X^{(1)}, \ldots, X^{(N)}\right\} \setminus \left\{X^{(j)}\right\}$ (вершина не может быть предком самой себе , то есть петли в графе отсутствуют). Тогда ОМД структуры $g \in G$ при заданной последовательности из n наблюдений $x^n = d_1 d_2 \ldots d_n$ вычисляется по формуле: $L(g, x^n) = H(g, x^n) + \left(k(g)/2\right) \cdot log\left(n\right)$, где $k\left(g\right)$ – количество независимых условных вероятностей в сетевой структуре g ,а $H\left(g, x^n\right)$ – эмпирическая энтропия.

$$H\left(g, x^n\right) = \sum_{j \in J} H\left(j, g, x^n\right) ; k\left(g\right) = \sum_{j \in J} k\left(j, g\right)$$

ОМД j-й вершины вычисляется по формуле:

$$L\left(j, g, x^n\right) = H\left(j, g, x^n\right) + \left(k\left(j, g\right)/2\right) \cdot log\left(n\right).$$

$k\left(j, g\right)$ – количество независимых условных вероятностей $j-$ й вершины:

$$k\left(j, g\right) = \left(\alpha^{(j)} - 1\right) \cdot \prod_{k \in \phi(j)} \alpha^k ,$$

где $\varphi\left(j\right) \subseteq \left\{1, \ldots, j-1, j+1, \ldots, N\right\}$ это такое множество что

$$\Pi^{(j)} = \{X^{(k)} : k \in \varphi^{(j)}\}.$$

Эмпирическая энтропия j-й вершины вычисляется по формуле:

$$H\left(j, g, x^n\right) = \sum_{s \in S(j,g)} \sum_{q \in A^{(j)}} -n[q, s, j, g] \cdot log \frac{n[q, s, j, g]}{n[s, j, g]} ,$$

где $n(s, j, g) = \sum_{i=1}^{n} I(\pi_i^{(j)} = s)$; $n[q, s, j, g] = \sum_{i=1}^{n} I(x_i = q, \pi_i^{(j)} = s)$, где $\pi^{(j)} = \Pi^{(j)}$ означает $X^{(k)} = x^{(k)}$, $\forall k \in \varphi^{(j)}$, функция $I\left(E\right) = 1$, когда предикат $E = true$, в противном случае $I\left(E\right) = 0$.

Таким образом, метод ОМД по сути является обращением задачи оптимального кодирования: из поиска оптимального представления данных извлекается модель источника.

В данной работе на базе кредитных историй имеющихся клиентов банка решается задача нахождения вероятности того, что произвольный клиент погасит просроченную задолженность. Для решения задачи 2 рассматривались функции правдоподобия из экспоненциального семейства

$$p(x \mid y) = \exp\left(c_y(d)\left\langle q_y, x\right\rangle + b_y(d, q_y) + d(x, d)\right),$$

где q_y ОЎ^N — параметр сдвига, d— параметр разброса, b_y, c_y, d— произвольные числовые функции независящие от y. При решении задачи 1 использовался EM-алгоритм и его модификации [4].

Список литературы

1. Grunwald P. A Tutorial Introduction to theMinimumDescription Length Principle. // Advances in Minimum Description Length: Theory and Applications MIT Press Cambridge, MA, USA, 2005, 80 p.

2. Вапник В. Н. Восстановление зависимостей по эмпирическим данным. — М.: Наука, 1979.

3. Вапник В. Н., Червоненкис А. Я. Теория распознавания образов. — М.: Наука, 1974.

4. Математические методы обучения по прецедентам (теория обучения машин) [электронный ресурс]. Режим доступа: http://www.machinelearning.ru/wiki/images/6/6d/Voron-ML-1.pdf

УДК 519.7

ГИБРИДНЫЙ АЛГОРИТМ ЗАМЕЩЕНИЯ СТРАНИЦ ДЛЯ ЗАДАЧИ КЭШИРОВАНИЯ

Петраков И. Е.
научный руководитель д-р физ.-мат. наук Быкова В. В.
Сибирский федеральный университет

Компьютер состоит из самых разных по принципам работы и назначению устройств, и его подсистемы и узлы работают с различной скоростью. Использование кэш-систем является одним из приемов, направленных на согласование работы разных по скорости устройств и повышение производительности вычислительных систем [5]. В зависимости от назначения кэш-системы могут значительно различаться. Однако основным принципом их реализации является использование двух типов памяти: основной памяти (большой по объему, но медленной) и кэш-памяти (быстрой, но ограниченной по объему) [3]. При проектировании кэш-системы требуется определить основные ее параметры: количество уровней кэша, степень ассоциативности, размер кэш-памяти, стратегия замещения (или политика вытеснения) страниц из кэша [2]. Задачу определения оптимальной стратегии замещения страниц называют задачей кэширования. Суть этой задачи для одноуровневой системы кэширования заключается в следующем (рис. 1). Основная память разбита на страницы равной длины. Страницы пронумерованы. Все страницы равны по важности и размеру. Кэш-память содержит ограниченное количество страниц. Первоначально кэш-память пуста. На очередном шаге поступает запрос на какую-то страницу. Если эта страница в кэш-памяти уже имеется, то достаточно сообщить ее номер расположения в кэше. Если запрашиваемой страницы нет в кэше, то требуется решить, в какое место кэша можно ее записать и сделать это, считав запрашиваемую страницу из основной памяти. Задача кэширования состоит в том, чтобы определить какую именно страницу удалить из кэша, чтобы освободить место для новой страницы. Алгоритмы решения этой задачи называются алгоритмами кэширования или стратегиями замещения.

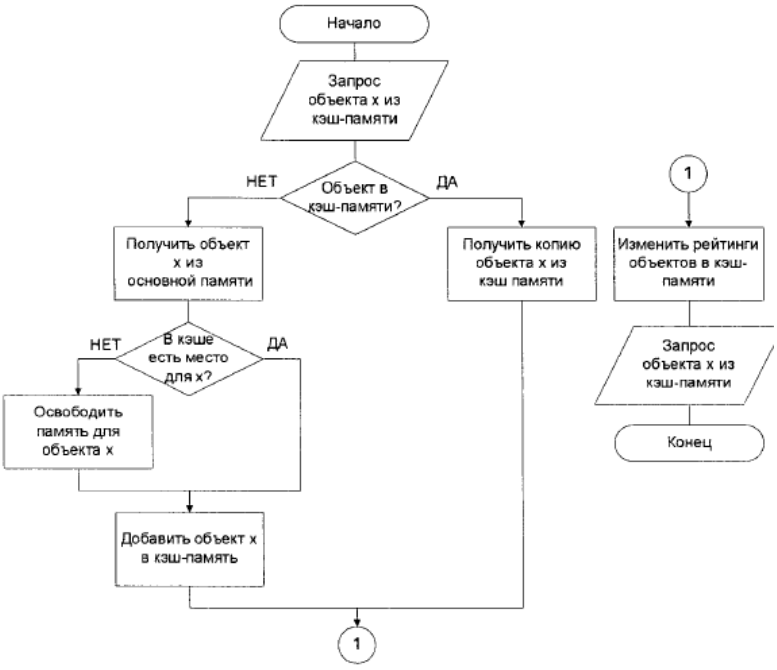

Рис. 1

Для исследования задачи кэширования целесообразно построить математическую модель стратегии замещения страниц. Для этого хорошо подходит автоматная модель. Предварительно приведем основные сведения из теории автоматов.

Конечным автоматом (в дальнейшем просто автоматом) называется система

$G = \{I, Q, V, \delta, \lambda\}$ (1)

где $I = \{a_1, \ldots, a_m\}$, $Q = \{q_1, \ldots, q_n\}$, $V = \{v_1, \ldots, v_k\}$ – конечные множества, δ: $Q \times I \rightarrow Q$ и λ: $Q \times I \rightarrow V$ отображения, определенные на этих множествах. Множество I называется входным алфавитом, V – выходным алфавитом, Q – алфавитом состояний, δ – функцией переходов, λ – функцией выходов. Такая модель автомата называется автоматом Мили. В случае, если функция выходов зависит только от состояний, автомат называется автоматом Мура. Существует теорема, доказывающая, что для любого автомата Мили существует эквивалентный ему автомат Мура. Благодаря этой теореме, можно определить автомат как $A = (I, Q, \delta, q_0)$ без указания выходного алфавита и функции выхода [4].

Опишем автоматную модель алгоритма кэширования. Введем следующие обозначения. Пусть $N = \{1,2...d\}$ – множество страниц в основной памяти; $\omega = (r_1, r_2, \ldots, r_T)$ – трасса запросов на доступ к объектам, r_t – объект, запрошенный от кэш-системы в момент времени t; $M = \{S \mid S \subseteq N, |S| < k\}$ – множество подмножеств множества объектов N, которые могут быть расположены в кэш-памяти размера k. Полагаем, что каждое подмножество S упорядочено, т.е. все входящие в S страницы упорядочены в соответствии с их рейтингами, установленными кэш-системой. Назовем S некоторым состоянием кэш-памяти, а M множеством всех возможных состояний кэш-памяти.

Рассмотрим следующую систему:

$A = \{N, M, q_0, g\}$ (2)

где M – множество состояний управления; q_0 – начальное состояние управления; g – функция перехода, определяющая новое состояние управления и новое состояние кэш-памяти, которое содержит запрошенный объект.

В терминах конечных автоматов для алгоритма кэширования входным алфавитом I является множество N, алфавитом состояний Q – множество M, а функция перехода δ – функция g, которая определяется выбранным алгоритмом кэширования. Таким образом, система (2) задает автоматную модель алгоритма кэширования.

Существует уже много разработанных алгоритмов кэширования со своими достоинствами и недостатками. Одним из способов улучшения эффективности работы кэш-системы является использование гибридных алгоритмов. Гибридным алгоритмом будем называть два или более алгоритмов кэширования, связанных некоторой функцией, определяющей выбор алгоритма при каждом запросе. Преимуществом гибридного алгоритма является то, что он позволяет исключить недостатки работы каждого из алгоритмов и использовать их достоинства в наиболее выгодных ситуациях.

Пусть гибридный алгоритм кэширования является комбинацией алгоритмов $A_1 = \{N, M_1, q_{01}, g_1\}$ и $A_2 = \{N, M_2, q_{02}, g_2\}$. Множества M_1 и M_2 – множества всех подмножеств множества N, а множество N одинаково для A_1 и A_2. Начальное состояние гибридного алгоритма определяется парой (q_{01}, q_{02}). Для перехода на каждом шаге выбирается одна из пары функций (g_1, g_2) исходя из параметров гибридизации. Таким образом, гибридный алгоритм определяется системой $(N, M, (q_{01}, q_{02}), (g_1, g_2), f)$, где f – функция выбора (g_1, g_2).

Исследуем гибридный алгоритм Hyb_LRU+LFU, являющийся комбинацией известных алгоритмов LRU и LFU [2]. Напомним стратегии замещения страниц, реализуемые алгоритмами LRU и LFU.

Least Recently Used, LRU. В этом алгоритме вытесняется страница, неиспользованная дольше всех. Алгоритм требует хранения времени последнего запроса к каждой странице, находящейся в кэше. Постоянное обновление этой информации и поиск страницы для вытеснения, может потребовать много времени.

Least Frequently Used, LFU. Алгоритм LFU подсчитывает, как часто используется страница. Те страницы, обращения к которым происходят реже всего, вытесняются в первую очередь.

Описание гибридного алгоритма Hyb_LRU+LFU представлено на рис. 2.

В качестве критерия эффективности гибридного алгоритма используем рейтинг кэш-попаданий (HR): отношение количества кэш-попаданий к общему количеству запросов. При обработке достаточно длинных последовательностей запросов, у алгоритма LFU возникает характерная проблема – проблема загрязнения кэша. Суть проблемы заключается в том, что популярные в недавнем времени объекты, которые набрали большие значения кэш-рейтинга, остаются в кэше на протяжении некоторого времени (возможно, значительного) после того, как популярность запросов к ним существенно снижается. Гибридизация, описанная выше, позволяет избавится от этой проблемы. При загрязнении кэша, эффективность алгоритма LFU начнет падать, что приведет к переходу на алгоритм LRU, который позволяет очистить кэш от страниц с высокой частотой запросов, но уже не используемые в текущем периоде.

Алгоритм Hyb_LRU+LFU (N, k, σ, Δ)

Вход: N, k, σ, Δ, где N – количество страниц внешней памяти, k – размер кэша, σ – трасса запросов, Δ – длина периода адаптации, алгоритм LRU (N, k, σ), алгоритм LFU (N, k, σ).

Выход: r – количество запросов к внешней памяти.

Пусть $\sigma_i = [\sigma_{i \cdot \Delta}, \ldots, \sigma_{(i+1) \cdot \Delta})$

$d \leftarrow len(\sigma)$

$n \leftarrow d / \Delta$

если LRU (N, k, σ_0) < LFU (N, k, σ_0) **тогда**

    $r \leftarrow r +$ LRU (N, k, σ_0)

иначе

    $r \leftarrow r +$ LFU (N, k, σ_0)

для i **с** 1 **до** n

    если LRU (N, k, σ_{i-1}) < LFU (N, k, σ_{i-1}) **тогда**

        $r \leftarrow r +$ LRU (N, k, σ_i)

    иначе

        $r \leftarrow r +$ LFU (N, k, σ_i)

Рис. 2

Алгоритмы LRU, LFU и Hyb_LRU+LFU были реализованы в виде программ, с помощью которых были выполнены вычислительные эксперименты. Было проведено две серии экспериментов. Алгоритмы LRU, LFU и гибридный алгоритм LRU+LFU сравнивались по рейтингу попаданий HR.

В таблице 1 представлены результаты первой серии экспериментов. В этой серии использовались специально построенные трассы для выявления «худшего случая» работы алгоритма LRU.

Таблица 1 – Рейтинг попаданий при «худшем случае» работы LRU

Длинна трассы	LRU	LFU	LRU+LFU	% работы LFU
1000	0.035	0.03	0.029	60%
2000	0.0265	0.027	0.0275	70%
5000	0.036	0.1402	0.0676	84%
10000	0.0368	0.1276	0.0781	88%
15000	0.0358667	0.147667	0.101	96%
25000	0.03452	0.23204	0.1022	95.2%
40000	0.0346	0.278575	0.1088	99%

В таблице 2 представлены результаты первой серии экспериментов. Эта серия проводилась с переменой длинной трассы. Целью данной серии экспериментов было показать несостоятельность работы алгоритма LFU без очистки кэша при его «загрязнении» с увеличением длины трассы.

Таблица 2 – Рейтинг попаданий при «худшем случае» работы LFU

Длинна трассы	LRU	LFU	LRU+LFU	% работы LFU
1000	0.291	0.104	0.287	0%
2000	0.412	0.1085	0.4195	80%
5000	0.5342	0.1266	0.5162	56%
10000	0.5615	0.1284	0.5577	66%
15000	0.5786	0.125733	0.570333	68%
25000	0.5796	0.12608	0.57372	61.6%
40000	0.590875	0.123375	0.588	65.5%

Список литературы

1 Аль-Згуль М. Б. Гибридные алгоритмы в системах кэширования объектов / М. Б. Аль-Згуль // Вестник ДГТУ. – 2008. – №4.

2 Болотов, П. Принципы работы кэш-памяти [Электронный ресурс]: Alasir / П. Болотов. Режим доступа: http://alasir.com/articles/cache_principles/index_rus.html.

3 Жуков, А. И. Адаптивные алгоритмы кэширования в информационных системах: дис. канд. техн. наук: 05.13.01 / А. И. Жуков - Ростов на Дону, 2012.

4 Кузнецов, О. П. Дискретная математика для инженера: учебник / О.П. Кузнецов – Москва: Лань, 2009.

5 Соболенко, Р. Кэширование данных: путь к повышению производительности [Электронный ресурс]: Компьютерная газета / Р. Соболенко // Компьютерная газета, 1997. – Режим доступа: http://www.nestor.minsk.by/kg/1997/38/kg73803.htm.

УДК 519.23

ИССЛЕДОВАНИЕ НЕИЕРАРХИЧЕСКИХ АЛГОРИТМОВ КЛАСТЕРИЗАЦИИ ДАННЫХ

Портнягин Н.П.
научный руководитель канд. физ.-мат. наук, доц. Баранова И.В.
Сибирский федеральный университет

Кластерный анализ [1], [2] представляет собой раздел статистического анализа данных, объединяющий методы разбиения (группировки) множества наблюдаемых объектов на сравнительно однородные группы, называемые ***кластерами***. Однородность кластеров означает, что объекты, отнесенные к одному кластеру, должны быть близки относительно выбранной метрики. Объекты из разных кластеров должны существенно отличаться. Задача кластеризации относится к статистической обработке, а также к широкому классу задач обучения без учителя [3].

Целью данной работы является изучение методов кластеризации данных, а также решение практической задачи кластеризации данных.

Постановка задачи кластеризации

Пусть $X = \{x_1, x_2, \ldots, x_m\}$ – множество объектов, заданных значениями в пространстве признаков $P = \{P_1, P_2, \ldots, P_n\}$ (т.е. каждый объект $x_i = (x_i^1, x_i^2, \ldots, x_i^n)$) и задана функция расстояния (метрика) между объектами $\rho(x_i, x_j)$, $x_i, x_j \in X$.

Функцией **кластеризации** называется функция $f : X \to Y$, которая любому объекту $x \in X$ ставит в однозначное соответствие номер $y \in Y = \{1, \ldots, k\}, k \le m.$

Множество кластеров $C = \{C_1, C_2, \ldots, C_k\}$, $k \le m$, представляет собой разбиение множества объектов X такое, что кластер $C_i = \{x \in X,\ f(x) = i\}$, $C_i \cap C_j = \varnothing$. Причем для C справедливо следующее: если $x_i, x_j \in C_i$, то $\rho(x_i, x_j) \to \min$. Если $x_i \in C_i, x_j \in C_j$, то $\rho(x_i, x_j) \to \max$.

Тогда постановку задачу кластеризации данных можно сформулировать следующим образом:

Требуется найти такую функцию кластеризации f^*, чтобы

$$Q(f^*, C, \rho) = \min_f Q(f, C, \rho),$$

где $Q(f, C, \rho)$ – выбранный критерий качества кластеризации.

Методы кластеризации данных

На сегодняшний день предложено значительное количество методов кластеризации данных. Условно их можно разделить на две группы: *иерархические и неиерархические.*

Суть ***иерархической кластеризации*** состоит в последовательном объединении меньших кластеров в большие, или разделении больших кластеров на меньшие (метод Варда, метод деревьев решений и т.п.).

Неиерархические методы кластеризации представляют собой *итеративные методы дробления исходной совокупности объектов.* В процессе деления новые кластеры формируются до тех пор, пока не будет выполнено правило остановки. Среди неиерархических методов принято выделять следующие группы методов:

Общепринятой классификации методов кластеризации не существует, но можно выделить ряд общепринятых групп подходов:

1. Вероятностный подход:
- k-средних (k-means),
- k-medians,
- EM-алгоритм,
- Алгоритмы семейства FOREL,

2. Подходы на основе систем искусственного интеллекта:
- метод нечеткой кластеризации C-средних (C-means),
- нейронные сети Кохонена,
- генетические алгоритмы,

3. Теоретико-графовый подход (графовые алгоритмы кластеризации),

4. Статистические методы:
- ансамбль кластеризаторов,
- алгоритмы семейства KRAB,
- метод просеивания.

Выбор метода кластеризации зависит от количества данных и от того, есть ли необходимость работать одновременно с несколькими типами данных. В работе рассматриваются два наиболее популярных алгоритма кластеризации данных: метод k-средних и Ланса-Уильямса.

Описание алгоритма k-средних

Одним из самых популярных методов кластеризации данных является метод k–средних. В отличие от иерархических методов, которые не требуют предварительных предположений относительно числа кластеров, для возможности использования неиерархических методов необходимо иметь гипотезу о наиболее вероятном количестве кластеров.

Число k в названии метода означает количество кластеров, на которое производится разбиение данных. Выбор числа k может базироваться на результатах предшествующих исследований, теоретических соображениях или интуиции.

Центроид – точка, представляющая собой центр масс точек кластера, т.е. покоординатные средние точек из кластеров:

$$\mu_j = (\mu_{j1,}, \mu_{j2}, ..., \mu_{jn}),$$

$$\mu_j = \sum_{x_j \in S} x_{jl}$$

где $l = 1,...,n$, $j = 1,...,k$.

Приведем описание алгоритма:

Пусть имеется множество точек данных $X = \{x_1, ..., x_m\}$, где $x_i = (x_i^1, x_i^2, ..., x_i^m) \in \Re^n$

Задается количество кластеров k, и на первом шаге производится задание центроидов μ_j, $j = 1,...,k$ – «центров масс» кластеров S_j, $j = 1,...,n$

Каждому кластеру соответствует один центр. Выбор начальных центроидов может осуществляться следующим образом:

- выбор k – наблюдений для максимизации начального расстояния;
- случайный выбор k – наблюдений;
- выбор первых k - наблюдений.

Кластеры . $S_j = \{\varnothing\}$, $j = 1,...,k$

1. Производится распределение объектов по кластерам. Точка $x_i, i = 1,..., n$ относится к ближайшему кластеру, т.е., $x_i \in S_j$ где

$$p(x_i, \mu_{j*}) = \min_{j=1,...,k} p(x_i, \mu_j)$$

В качестве метрики используется евклидова метрика.

В результате каждый объект назначен определенному кластеру.

2. Вычисляются новые центры кластеров μ_j, $j = 1,..., k$ как центры масс новых кластеров $S_j, j = 1,..., n$ полученных на предыдущем этапе.

3. Продолжать итерационный процесс вычисления центров и перераспределения объектов до тех пор, пока не выполнится одно из условий:

- кластерные центры μ_j, $j = 1,..., k$ стабилизировались (перестали изменяться);
- число итераций равно максимальному числу итераций (ограничение на число итераций).

Практическая задача кластеризации кинофильмов по некоторым критериям

В работе было создано программное обеспечение, реализующее работу методов k-средних и Ланса-Уильямса. Программный модуль был выполнен в среде Visual Studio 2013 на языке C++. С помощью реализованных алгоритмов была выполнена кластеризация ряда кинофильмов. Кластеризация осуществлялась по нескольким критериям: рейтингу кинокритиков, рейтингу пользователей и ряду показателей прибыли. Данные были взяты с сайта http://www.rottentomatoes.com.

Заключение

В работе было проведено исследование методов кластеризации данных. Создано программное обеспечение, реализующее работу методов k-средних и Ланса-Уильямса. Решена практическая задача кластеризации данных этими двумя методами и выполнено сравнение полученных результатов.

Список литературы

1. Воронцов К. В. Лекции по алгоритмам кластеризации и многомерного шкалирования. М.: МГУ, 2007. 18 с.

2. Миркин Б.Г. Методы кластер – анализа для поддержки принятия решений: обзор: препринт WP7.2011.03. М.: Изд. дом национального исследовательского университета «Высшая школа экономики», 2011. 88 с.

3. Загоруйко Н.Г. Прикладные методы анализа данных и знаний. Новосибирск: ИМ СО РАН, 1999. 270 с.

4. Дюран Б, Одел П. Кластерный анализ: под ред. А.Я. Боярского. М.: «Статистика», 1977. 128 с.

УДК 004.9

РАЗРАБОТКА ГЕНЕТИЧЕСКОГО АЛГОРИТМА ДЛЯ РЕШЕНИЯ ЗАДАЧИ КЛАССТЕРИЗАЦИИ ДАННЫХ

Семёнов М.Г.
научный руководитель канд. физ.-мат. наук, доц. Баранова И.В.
Сибирский федеральный университет

В настоящее время практически во всех областях человеческой деятельности существует настоятельная потребность в изучении статистических данных, описывающих поведение наблюдаемых объектов, событий, процессов или явлений. Одной из наиболее актуальных и практически востребованных задач анализа данных является задача разбиения объектов на сравнительно однородные группы, называемые кластерами. Данная задача называется задачей кластеризации данных. Также ее принято называть таксономией, автоматической классификацией, группировкой объектов или задачей обучения без учителя.

Кластер — группа однородных элементов, характеризуемых общим свойством. Однородность кластеров означает, что объекты, отнесенные к одному кластеру, должны быть близки относительно выбранной метрики. Объекты из разных кластеров должны существенно отличаться.

Целью данной работы является исследование генетических алгоритмов и методов кластеризации данных, а также разработка генетического алгоритма, предназначенного для решения задачи кластеризации данных.

Постановка задачи кластеризации

Пусть $X = \{x_1, x_2, \ldots, x_n\}$ – множество объектов, заданных значениями в пространстве признаков $P = \{P_1, P_2, \ldots, P_m\}$ (т.е. каждый объект $x_i = (x_i^1, x_i^2, \ldots, x_i^m)$) и задана функция расстояния (метрика) между объектами $\rho(x_i, x_j)$, $x_i, x_j \in X$.

Функцией кластеризации называется функция $f : X \to Y$, которая любому объекту $x \in X$ ставит в однозначное соответствие номер $y \in Y = \{1, \ldots, k\}, k \le m.$

Множество кластеров $C = \{C_1, C_2, \ldots, C_k\}$, $k \le m$, представляет собой разбиение множества объектов X такое, что кластер $C_i = \{x \in X,\ f(x) = i\}$, $C_i \cap C_j = \varnothing$. Причем для C справедливо следующее: если $x_i, x_j \in C_i$, то $\rho(x_i, x_j) \to \min$. Если $x_i \in C_i, x_j \in C_j$, то $\rho(x_i, x_j) \to \max$.

Тогда постановку задачу кластеризации данных можно сформулировать следующим образом:

Требуется найти такую функцию кластеризации f^*, чтобы

$$Q(f^*, C, \rho) = \min_f Q(f, C, \rho), \qquad (1)$$

где $Q(f, C, \rho)$ – выбранный критерий качества кластеризации.

Как уже было сказано выше, каждый объект описывается набором своих характеристик, называемых *признаками*. Признаки могут быть следующих типов:

- *бинарный* признак: $P_i = \{0, 1\}$,
- *номинальный (качественный)* признак: P_i — конечное множество;

- *порядковый* признак: P_i — конечное упорядоченное множество;
- *количественный* признак: $P_i = \Re$ — множество действительных чисел.

Самой распространенной ситуацией является кластеризация объектов, у которых все признаки являются количественными, т.е. когда $P = \{P_1, P_2, \ldots, P_n\} = \Re^n$ (каждый объект $x_i = \left(x_i^1, x_i^2, \ldots, x_i^n\right) \in \Re^n$). В работе мы будем рассматривать именно такую ситуацию.

Методы решения задачи кластеризации данных

На данный момент существует большое количество методов кластеризации данных. Их можно разделить на две группы: *иерархические и неиерархические.*

Суть иерархической кластеризации состоит в последовательном объединении меньших кластеров в большие, или разделении больших кластеров на меньшие (метод Варда, метод деревьев решений и т.п.).

Неиерархические методы кластеризации представляют собой *итеративные методы дробления исходной совокупности объектов.* В процессе деления новые кластеры формируются до тех пор, пока не будет выполнено правило остановки. Среди неиерархических методов принято выделять следующие группы методов: *вероятностный подход* (k-средних, к-медиан, EM-алгоритмы, алгоритмы семейства FOREL), *графовые алгоритмы кластеризации, подход на основе систем искусственного интеллекта* (метод нечеткой кластеризации С-средних, нейронные сети Кохонена, генетические алгоритмы).

Метод k-средних

В работе рассмотрен один из самых популярных алгоритмов кластеризации – алгоритм k-средних, относящийся к неиерархическому подходу. Также этот метод называют быстрым кластерным анализом.

В отличие от иерархических методов, которые не требуют предварительных предположений относительно числа кластеров, для возможности использования этого метода необходимо иметь гипотезу о наиболее вероятном количестве кластеров k.

Центроид — точка данных $c_j = (c_j^1, c_j^2, ..., c_j^n)$, представляющая собой центр масс точек кластера, т.е. покоординатное среднее точек из кластеров: $c_j = \sum_{x_j \in C} x_j^i$, $j = \overline{1,n}$, $i = \overline{1,k}$.

Алгоритм метода k-средних имеет следующий вид:

1) Выделяются некоторые образы из обучающей выборки – начальные центры кластеров $c_1^{(0)}, ..., c_k^{(0)}$ и полагается $m = 0$.

2) Вся обучающая выборка разбивается на k кластеров – получаются некоторые кластеры $X_1^{(m)}, ..., X_k^{(m)}$.

3) Вычисляются новые центры кластеров по формуле:

$$c_i^{(m+1)} = \frac{1}{|X_i^{(m)}|} \sum_{x \in X_i^{(m)}} x.$$

4) Проверяется выполнение условия остановки: $c_i^{(m+1)} = c_i^{(m)}$ для всех $m = \overline{1,k}$. В противном случае переход к пункту 2.

Алгоритм k-средних минимизирует функционал суммарной выборочной дисперсии $Q^{(3)}$ и сходится за конечное число шагов.

Генетические алгоритмы

Генетические алгоритмы – это адаптивные методы поиска, которые в последнее время используются для решения задач оптимизации. В них используются как аналог механизма генетического наследования, так и аналог естественного отбора. Впервые понятие генетического алгоритма было предложено в 1975 г. в работах Джона Холланда. Генетические алгоритмы основаны на принципах естественного отбора Ч. Дарвина. Они относятся к стохастическим методам. Эти алгоритмы успешно применяются в различных областях деятельности (экономике, физике, технических науках и т.д.). Основной идеей генетических алгоритмов является организация «борьбы за существование» и «естественного отбора» среди пробных решений задачи.

Перечислим основные операторы генетических алгоритмов:

Оператор селекции – отбирает популяцию для дальнейшего размножения. Заключается в том, что родителями могут стать только те особи, значение приспособленности которых не меньше пороговой величины, например, среднего значения приспособленности по популяции. Такой подход обеспечивает более быструю сходимость алгоритма.

Оператор скрещивания – служит для распространения хороших генов по популяции.

Оператор мутации – изменяет значение гена в хромосоме на любое другое возможное значение. Случайное изменение генов должно осуществляться с низкой вероятностью, обычно в пределах [0.001;0.01]. После процесса воспроизводства происходят мутации. Данный оператор необходим для «выбивания» популяции из локального экстремума и препятствует преждевременной сходимости. Это достигается за счет того, что изменяется случайно выбранный ген в хромосоме.

В работе предлагается генетический алгоритм для решения вышеприведенной задачи кластеризации данных (1) с детерминированным числом кластеров *k*.

Генетический алгоритм решения задачи кластеризации данных

1. Генерируется начальный массив векторов A из $l = \frac{n}{2}$ векторов вида $A_i = (a_{i1}, a_{i2}, \ldots, a_{im}),\ i = \overline{1,l}$, где a_{ij} – номер кластера, к которому принадлежит j-ая точка, $a_{ij} \in [1,k]$. Причем задается случайным образом.

2. Для каждого A_i считаются центры масс для каждого кластера $\vec{c}_{ij}$, по следующей формуле $\vec{c}_{ij} = \frac{\sum_{t=0}^{h} \vec{x}_{jt}}{h}$, где *h* - количество точек в кластере j, а x_{jt} - точка из этого кластера.

3. Радиусом j-ого кластера назовем $r_{ij} = \max_t \rho(c_{ij}, x_{jt})$, где $\rho(c_{ij}, x_{jt})$ - расстояние между точками в метрике Евклида, находится по следующей формуле $\rho(c_{ij}, x_{jt}) = \sqrt{\sum (c_{ijp} - x_{jtp})^2}$, $p = 1,2,\ldots,n$. Тогда коэффициентом приспособленности для i-ого вектора назовем $s_i = \max_j r_{ij}$, а сам вектор s - вектором приспособленности.

4. Производится новое массив векторов а $\tilde{A}$. Каждый новый вектор $\tilde{A}_i$ формируется из векторов и A_{l-i}, последующему правилу: $\tilde{A}_i = \left(a_{i1}, \ldots, a_{i\frac{n}{2}}, a_{l-i\frac{n}{2}+1}, \ldots, a_{l-im} \right)$. Считается вектор приспособленности $\tilde{s}$ для нового поколения.

5. Сравниваются векторы $\tilde{s}$ и s, если $\tilde{s}_i < s_j$ то A_j заменяется на $\tilde{A}_i$, а s_j на $\tilde{s}_i$.

6. Мутация производится изменением в каждом векторе A_i случайной точки a_{ij} на случайное число из отрезка $[1, k]$. Считается новый вектор приспособленности $\hat{s}$ для мутировавшего поколения $\hat{A}$.

7. Сравниваются векторы $\hat{s}$ и s, и если $\hat{s}_i < s_j$ то A_j заменяется на $\hat{A}_i$, а s_j на $\hat{s}_i$.

8. 4 - 7 повторяется до тех пор, пока меняется вектор s.

В работе было создано программное обеспечение, реализующее работу предложенного генетического алгоритма кластеризации, а также методов k-средних и FOREL.

Практическая задача кластеризации

В работе решалась практическая задача кластеризации данных - кластеризации моделей телефонов по показателям их функциональности. В качестве показателей функциональности рассматривались комплексная оценка надежности и возможностей телефона пользователями, стоимость и технические характеристики, такие как продолжительность работы аккумулятора и разрешение фотокамеры. В кластеризации участвовало 50 моделей телефонов. Кластеризация выполнялась с помощью предложенного генетического алгоритма, а также методов к-средних и FOREL. Было проведено сравнение полученных результатов кластеризации.

Заключение

В работе было проведено исследование генетических алгоритмов и ряда методов кластеризации данных. Был предложен и разработан генетический алгоритм решения задачи кластеризации. Создано программное обеспечение, реализующее работу предложенного алгоритма кластеризации, а также методов k-средних и FOREL. Решена практическая задача кластеризации данных этими тремя методами и выполнено сравнение полученных результатов.

Список литературы

1. Быкова В. В. Дискретная математика с использованием ЭВМ: учебное пособие. Красноярск: РИО КрасГУ, 2006. 200 с.

2. Кормен Т. Х, Лейзерсон Ч. И., Ривест Р.Л. Алгоритмы: построение и анализ: пер. с англ. под ред. А. Шеня. М.: Вильямс, 2005. 1296 с.

3. Панченко Т. В. Генетические алгоритмы. Астрахань: Издательский дом «Астраханский университет», 2007. 87 с.

4. Пападимитриу, Х. Х., Стайглиц К. Комбинаторная оптимизация. Алгоритмы и сложность: пер. с англ. В. Б. Алексеева. М.: Мир, 1985. 512 с.

УДК 519.178

ПРИЕМЫ УСКОРЕНИЯ АЛГОРИТМА ДЕЙКСТРЫ В ЗАДАЧАХ МАРШРУТИЗАЦИИ

Солдатенко А. А.
научный руководитель д-р физ.-мат. наук, Быкова В. В.
Сибирский федеральный университет

Задача поиска кратчайшего пути в графе (Shortest-Paths, SP) – одна из основных задач теории графов, с помощью которой моделируются стратегии передвижения в дорожных и беспроводных одноранговых сетях. Задача SP состоит в поиске кратчайшего пути в заданном графе между двумя его вершинами s и d (стартовой и целевой соответственно), в котором минимизируется сумма длин рёбер, составляющих (s, d)-путь [1, 2]. Традиционно SP формулируется следующим образом. Задан связный граф $G = (V, E)$ с неотрицательной вещественнозначной функцией весов $w: E \to R^+$, а также заданы стартовая вершина (или источник) $s \in V$ и целевая вершина (или сток) $d \in V$. Требуется найти в G кратчайший (s, d)-путь.

Чаще всего задача SP решается с помощью эффективного алгоритма Дейкстры, который осуществим за время $O(m + n \log n)$, где m и n – число ребер и вершин графа. Но даже при такой производительности данный алгоритм начинает работать неприемлемо долго на огромных графах. Типичным примером является поиск кратчайшего маршрута в дорожной сети, насчитывающей миллионы узлов. Чтобы справиться с этой проблемой используют различные приемы (техники) ускорения алгоритма Дейкстры. В данной работе представлен краткий обзор наиболее эффективных современных техник ускорения.

Приемы ускорения алгоритма Дейкстры подразделяют на приемы, использующие предобработку исходного графа, и приемы без предобработки. Вначале рассмотрим приемы ускорения без предобработки. К ним относят двунаправленный поиск, целенаправленный поиск или алгоритм A* [3].

При двунаправленном поиске запуск алгоритма Дейкстры выполняется одновременно из стартовой вершины s и целевой вершины d, в последнем случае используется реверсивный набор рёбер $E' = \{(u, v) \mid (v, u) \in E\}$. Как только будет найдена вершина, помеченная как пройденная для обоих направлений, кратчайший путь найден.

Целенаправленный поиск использует потенциальную функцию для определения всякой очередной вершины. Потенциальная функция $\pi(v)$ должна быть задана или вычислена для каждой вершины v графа $G = (V, E)$. Значение $\pi(v)$ определяет потенциал вершины v, как нижнюю оценку кратчайшего пути от вершины v до целевой вершины d. Таким образом, алгоритм Дейкстры выбирает на каждом шаге вершину с минимальной суммой $dist(v) + \pi(v)$, где $dist(v)$ – расстояние от стартовой вершины до вершины v, $\pi(v)$ – оценка пути от вершины v до целевой вершины d. Этот метод применим, если вершины графа задаются координатами точек на плоскости, а веса дуг графа удовлетворяют неравенству треугольника.

Приемы с предобработкой делят на несколько групп.

Первая группа – иерархические методы ускорения. Данная группа производит изменение входного графа при помощи ввода новых рёбер, которые представляют собой некоторые кратчайшие пути. Основные представители этой группы: многоуровневый подход, магистральный подход, Shortcuts [5, 6, 8].

Многоуровневый подход производит последовательную декомпозицию графа при помощи вершинных сепараторов $S_i \subset V$, $i = 0, \ldots, p$, которые называются шлюзами. При этом каждый i-й шлюз используется на i-ом уровне:

$S_0 = V \supseteq S_1 \supseteq \ldots \supseteq S_p$.

После выделения p уровней графа, производится добавление новых рёбер в графе. Выделяют три вида рёбер:

- восходящее ребро, идущее от вершины, не входящей в текущий шлюз, до вершины текущего шлюза;
- нисходящее ребро, идущее от вершины текущего шлюза, к вершине другого шлюза;
- ребро уровня, идущее между вершинами текущего шлюза.

Всем новым рёбрам присваивается вес равный длине кратчайшего пути между вершинами, которые оно соединяет. Данный подход очень эффективно снижает время выполнения запроса, однако требует много времени на предобработку.

Магистральный подход не требует вычисления дополнительной информации, такой как сепараторы. Идея строится на том, что существуют каноничные пути, которые содержатся во многих кратчайших путях. Примером таких путей являются высокоскоростные магистральные дороги. Для определения иерархии графа используется дерево кратчайших путей, для некоторой выборки (s, d)-путей. На основе полученных деревьев определяются каноничные пути. Следующий уровень иерархии – объединение всех узлов со степенью не меньше двух. Завершающим этапом является замена всех смежных между собой вершин степени два, одним ребром.

Главная идея техники Shortcuts заключается в замене ребер, соединяющих последовательность вершин степени два, одним ребром. Также, в некоторых случая, возможно, полное удаление ребра, если существует более краткий путь, что проверяют, наращивая дерево кратчайших путей. Однако, если ребро очень длинное, и для проверки потребовалось больше заданного числа итераций k, ребро сохраняют.

Вторая группа приемов ускорения связана с маркировкой вершин и рёбер графа. Присвоенные метки вершинам и ребрам графа служат некоторыми индикаторами во время выполнения запроса. В данную группу входят следующие приемы: достижимость (Reach), алгоритм Arc-flags.

Достижимость (Reach) – прием, основанный на присваивании каждой вершине меры достижимости. Для всякой вершины $v \in P$, делящей кратчайший путь P на две части, мера достижимость вычисляется по формуле:

$r_P(v) = \min(dist(P_{sv}), dist(P_{vd}))$.

Общая мера достижимости для вершины v определяется как величина

$r(v) = \max_P (r_P(v))$.

Таким образом, для вычисления мер достижимости требуется вычислить все кратчайшие пути в графе, либо использовать приближённые значения, например, евклидово расстояние. При выполнении запроса используется следующее условие: если

$r(u) < \min(dist(v) + w(v, u), LB(u, d))$,

где $LB(u, d)$ нижняя оценка расстояния от u до d, то вершина u не лежит на кратчайшем пути [6, 7].

Алгоритм Arc-flags всем ребрам входного графа также присваивает метки. Для этого граф разбивается на k примерно равных частей C_i. Далее каждому (u, v)-ребру графа G, присваивается k-битный вектор. Для каждого граничного (u, v)-ребра, т. е. такого, что

вершина $u \in C_i$, а вершина $v \notin C_i$, вычисляется дерево кратчайших путей с корнем в вершине u. Такая вершина u называется граничной вершиной части C_i. Всем рёбрам, входящим в это дерево, i-му биту вектора присваивается значение «истина». На фазе выполнения запроса алгоритм Дейкстры использует только те ребра, i-й бит которых имеет значение «истина», где i – номер части C_i, содержащей целевую вершину d [7].

Существует ещё один прием ускорения с предобработкой, который не относится ни к одной из двух представленных выше групп. Этот приём называется Landmarks. В данном случае вначале выбирается некоторое подмножество $L \subset V$ множества вершин графа $G = (V, E)$. Вершины, входящие в L, называются «ориентирами». Затем вычисляются и сохраняются длины кратчайших путей от ориентиров до всех вершин графа. Далее определяется потенциальная функция для всякого ориентира $l \in L$ по формуле:

$$\pi_l(v) = dist(l, v) - dist(l, d),$$

где v – некоторая вершина графа, а d – целевая вершина (s, d)-запроса. Потенциал вершины v вычисляется следующим образом:

$$\pi_L(v) = \max \{\pi_l(v): l \in L\}.$$

Прием Landmarks даёт значительное ускорение при хорошем расположении ориентиров, однако задача оптимального размещения ориентиров в вершинах графа является NP-трудной [5].

Перспективное направление развития приемов ускорения алгоритма Дейкстры является их комбинирование. Подбор комбинаций во многом зависит от особенностей решаемой задачи маршрутизации. Кроме того, актуально направление совершенствования этих приемов с целью учета динамических изменений топологии сети в течение времени [8].

Одной из стандартных комбинаций, является комбинирование приемов Shortcuts, Arc-flags, Landmarks, Reach, многоуровневого и магистральных подходов с двунаправленным поиском. Двунаправленный поиск не вносит дополнительных затрат и дает ускорение примерно в два раза. Однако двунаправленный поиск нельзя использовать в сетях с динамически изменяющейся топологией, т. е. в нестационарных сетях.

Комбинация SHARC (Shortcuts + Arc-flags) является эффективной, но затратной по времени работы. Такая комбинация может использоваться с некоторой доработкой на нестационарных сетях [4].

Наиболее популярной комбинацией является объединение трёх приемов: Landmarks, Reach и Shortcuts. Эта комбинация требует много дополнительной памяти, но увеличивает скорость выполнения запросов в несколько раз [6].

В современных приложениях часто приходится иметь дело с расширением задачи SP, когда вес каждой дуги графа является функцией от времени отправления из начальной вершины этой дуги. Такой граф называют нестационарной сетью, а саму задача –Time-Dependent Shortest-Path problem (TDSP). Например, TDSP возникает при передвижении на автомобиле в час пик по городским дорожным сетям в условиях пробок. Подобные ситуации считаются предсказуемыми и описываются через функции весов дуг от времени. Если SP полиномиально разрешимая задача, то TDSP для нестационарной сети общего вида без каких-либо ограничений на топологию сети и функции весов дуг NP-трудная задача [9]. В частном случае, когда нестационарная сеть удовлетворяет специальным требованиям (условию FIFO и неравенству треугольника), TDSP полиномиально разрешима и ее точное решение может быть найдено.

Для решения задачи TDSP используют классический алгоритм Дейкстры адаптированный к работе с ребрами, зависящими от времени. Основной техникой ускорения является Landmark [10]. Такой прием ускорения использован в задаче маршрутизации в нестационарных сетях. Его эффективность показана в работе [11]. Разработанный программный комплекс зарегистрирован в Объединенном Фонде Электронных Ресурсов «Наука и образование» [12].

Список литература

1. Емеличев В.А., Мельников О.И., Сарванов В.И., Тышкевич Р.И. Лекции по теории графов. М.: Книжный дом «Либроком», 2012. 392 с.
2. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы. Построение и анализ. Издательский дом «Вильямс», 2013. 1328 с.
3. Рассел С., Норвинг П. Искусственный интеллект: современный подход. Издательский дом «Вильямс», 2006. 1408 с.
4. Bauer R. & Delling D. SHARC: Fast and robust unidirectional routing. Journal of Experimental Algorithmics. 2009. Vol. 14. no. 4, pp. 1–29.
5. Goldberg A. & Harrelson C. Computing the shortest path: A* search meets graph theory. In SODA-2005, SIAM, 2005, pp. 156–165.
6. Goldberg A., Kaplan H. & Werneck R. Better landmarks within reach. In Proceedings International Workshop on Experimental Algorithms (WEA), Vol. 4525 of Lecture Notes in Computer Science. Springer, 2007, pp. 38–51.
7. Köhler E., Möhring R. H. & Schilling H. Fast point-to-point shortest path computations with arc-flags. 9th DIMACS Implementation Challenge, 2006, P. 27.
8. Wagner D. & Willhalm T. Speed-up techniques for shortest-path computations. In Proceedings STACS, 2007, pp. 23–36.
9. Sherali, H.D. The time-dependent shortest pair of disjoint paths problem: Complexity, models, and algorithms. Networks, 1998, № 31 (4), pp. 259–272.
10. Kaufman, D.E. Fastest paths in time-dependent networks for intelligent vehicle-highway systems application. Journal of Intelligent Transportation Systems, 1993, № 1 (1), pp. 1–11.
11. Быкова В.В., Солдатенко А.А. Адаптивное размещение ориентиров в задаче о кратчайшем пути для графов большой размерности. Программные продукты и системы, 2016, № 1 (113), С. 60–67.
12. Быкова В.В., Солдатенко А.А. Комплекс программ ALT_ADAPT для вычисления кратчайшего пути в графе большой размерности на основе алгоритма ALT и адаптивного размещения ориентиров: Свидетельство о регистрации электронного ресурса № 21097 от 27.07.2015 (ФГБНУ ИУО РАО, ОФЭРНиО).

УДК 339.13.017

МОДЕЛИРОВАНИЕ СТОИМОСТИ ЖИЛЬЯ В Г. КРАСНОЯРСКЕ ЗА 2013 ГОД

Юзаева А.Г., Савченко Л.М.
научный руководитель д-р физ.-мат. наук, проф. Сенашов С.И.
Сибирский государственный аэрокосмический университет имени академика М. Ф. Решетнева

Собраны и проанализированы данные по стоимости вторичной жилой недвижимости в г. Красноярске за 2013г. Количество объектов 2994. Каждая квартира характеризуется 13 параметрами: количество комнат, планировка, микрорайон, этаж, всего этажей в доме, материал стен, телефон, общая площадь, жилая площадь, площадь кухни, санузел, кухонная плита, наличие балкона или лоджии. На основе МНК построена адекватная линейная модель стоимости квартиры. В модель вошли 7 значимых факторов: общая площадь, количество комнат, микрорайон, материал стен, планировка, жилая площадь, санузел. При использовании полученной линейной модели наблюдается гетероскедастичность остатков. С помощью метода ОМНК, на основе линейной модели, построена новая модель с гомоскедастичными остатками. Построенная модель адекватно описывает опытные данные

Ранее, стоимость вторичного жилья за период 1998-2012 гг. в г.Красноярске изучалась в работах [1-4].

Рынок вторичного жилья 2013 года отличается от предыдущих годов тем, что на нем присутствовали интересные предложения кредитования, ипотеки от банков. Объем предложения квартир вырос на 24%, при этом жилье новой планировки стали продавать на 10% больше, в то время как в домах старого фонда предложение упало на 12%.

Предпочтения покупателей в 2013 году распределились так, что 33% проданных квартир - однокомнатные, по 27% - двух- и трехкомнатные, оставшиеся 13% квартир - с четырьмя и более комнатами.

Самыми востребованными квартирами у покупателей были однокомнатные квартиры ценою до 3 миллионов рублей. Такое жилье уже на протяжении многих лет не теряет своей конкурентоспособности, потому что является самым доступным вариантом жилья для молодых семей.

Разделение города по стоимости квадратного метра, на 2013 год, было таковым: самые дорогие квартиры в микрорайонах Центр, Академгородок, Взлетка, Копылова, Свободный. Цена за квадратный метр там варьируется в диапазоне от 68 000 до 120 000 рублей. В свою очередь, самое недорогое жилье типовой планировки можно приобрести в микрорайонах Черемушки, Энергетики, цена которых находится в диапазоне от 42 000 до 50 000 рублей на один квадратный метр.

Также на рынок вторичного жилья оказывает влияние большое количество строящихся домов в самом городе и его окрестностях, также активная реклама жилья в Дивногорске и Сосновоборске. Ещё одним конкурентом вторичного жилья начинают выступать поселки коттеджного типа в пригороде Красноярска. Они привлекают потенциальных покупателей низкими ценами, отсутствием пробок, красивой природой, чистым воздухом и отсутствием городского шума.

Цены на объекты недвижимости и их характеристики взяты из [5].

Предполагаем, что стоимость жилья описывается следующим линейным многофакторным уравнением регрессии

$$Z^i = a_0 + a_1 X_1^i + \ldots + a_{14} X_{14}^i, \quad (1)$$

Здесь Z^i - стоимость i квартиры, a_i - коэффициенты, определяемые по методу МНК, X_j^i - j фактор i квартиры.

С помощью метода наименьших квадратов было исследовано уравнение (1).

Оказалось, что она адекватно описывает стоимость квартир в городе Красноярске за 2013 г., поскольку $R^2 = 0{,}7$, а F – статистика равна 524,01 и превыша (2) табличное значение. Значимых факторов оказалось 7: общая площадь, количество комнат, микрорайон, материал стен, планировка, жилая площадь, санузел. Модель имеет вид:

$$Z_1^i = \underset{(122,9)}{985{,}67} - \underset{(34,23)}{308{,}15}\, X_1^i - \underset{(13,9)}{47{,}3}\, X_2^i - \underset{(5,98)}{55{,}7}\, X_3^i - \underset{(33,21)}{198{,}52}\, X_4^i + \underset{(1,58)}{65{,}6}\, X_5^i + \underset{(1,45)}{4{,}62}\, X_6^i + \underset{(41,}{81}$$

X_1^i - количество комнат в квартире;

X_2^i - планировка квартиры;

X_3^i - микрорайон, где находится квартира;

X_4^i - материал стен дома;

X_5^i - общая площадь квартиры;

X_6^i - жилая площадь квартиры;

X_7^i - санузел.

Качественный фактор X_2^i, планировка квартиры, заменяется значениями: индивидуальная планировка (это планировка квартиры в доме, построенном по индивидуальному проекту, обычно не бывает двух совершенно одинаковых квартир; отличается большой площадью, имеет кухню 10-15 кв.м., высокие потолки, большой коридор, одну или несколько лоджий) – 1, новая планировка (строительство началось в 1980-е годы, дома кирпичные и панельные, преимущественно девяти-, двенадцатиэтажные; основное отличие таких квартир - это кухня не менее 9 кв.м., комнаты изолированы, санузел раздельный, лоджия) – 2, ленинградка (строительство таких домов начало с 1970-ых годов, материал домов – панель или блок; лоджия в таких квартирах расположена с торца здания, а комнаты находятся по одной стороне вдоль длинного, широкого коридора) – 3, сталинка (квартиры в трех-пятиэтажных домах, строительство которых началось с 1950-ых годов, основные показатели таких квартир - высокие потолки и большая жилплощадь, комнаты изолированы, санузел раздельный.) – 4, улучшенная планировка (строительство таких домов началось в 1970-е годы, дома кирпичные и панельные пятиэтажки, позже появились и девятиэтажки) – 5, хрущевка (строительство домов началось в 1960-е, они панельные и кирпичные с тонкими стенами, сами квартиры имеют низкие потолки, маленькую кухню (не более 6 кв.м.), смежные комнаты, совмещенный санузел, узкий коридор) – 6, гостинка (тип жилого помещения, представляющего собой либо малометражную однокомнатную квартиру, либо комнату с кухонной нишей и санузлом) – 7, секционка (жилая площадь, где в одной секции 6 комнат, одна кухня, ванна и туалет) – 8, коммунальная (в коммунальной квартире живёт несколько семей или отдельных людей; они занимают одну или несколько комнат, вместе пользуются ванной, туалетом и кухней, а также коридором и прихожей) – 9, общежитие (это раздельные комнаты, расположенные вдоль коридора; санузлы, душевые и кухня – общие и находятся также в коридоре) – 10.

Вместо X_3^i, микрорайон, используются значения: академгородок – 1; центр, северный, копылова, железнодорожников, зеленая роща, солнечный, ж/д вокзал – 2;

взлетка, иннокентьевский, краевая больница – 3; студенческий городок, предмостная площадь, юбилейная, ветлужанка, ботанический – 4; космос, кецховели, красноармейская, БСМП, ГорДК, ДОК – 5; свободный, с/з октябрьский – 6; покровка, затон, цирк – 7; калинина, северо-западный, пашенный, емельяновский, мясокомбинат, козульский – 8; торговый центр, родина, ТЮЗ, спутник, океан, баджей – 9; кинотеатр Енисей, станция Енисей, Дивногорск, удачный – 10; первомайский, злобино – 11; крастец, уярский – 12; водников – 13; черемушки, энергетики, шинники, березовский, Сосновоборск – 14.

Вместо X_4^i, материал стен дома, используются: шлакоблок – 0, монолит – 1, кирпич – 2, панель – 3, дерево – 4.

Вместо X_7^i, тип санузла, используются значения: смежный – 1, раздельный – 2, два санузла – 3.

Под каждым коэффициентом указана его стандартная ошибка.

По тесту Голдфелда – Квандта, показано, что остатки $e^i = Y^i - Z^i$ для модели (2), с вероятностью 95% гетероскедастичны. Здесь Y^i - фактическая стоимость квартиры с номером i, а Z_1^i - ее стоимость, вычисленная по формуле (2).

На основе МНК построена регрессионная адекватная модель для остатков, которая имеет вид:

$$(e^i)^2 = \underset{(75,6)}{2772,6} + \underset{(6,01)}{54,49}\,X_1^i - \underset{(1,9)}{14,28}\,X_2^i - \underset{(4,47)}{4,78}\,X_3^i - \underset{(7,85)}{85,25}\,X_4^i + \underset{(0,01)}{0,23}\,X_5^i + \underset{(13,13)}{84,43}\,X_6^i + 1$$
$$- \underset{(0,02)}{0,15}\,X_8^i X_9^i + \underset{(0,07)}{0,83}\,X_{10}^i X_{11}^i \qquad (3)$$

Введены новые переменные, равные:

$$N^i = \frac{Z^i}{\sqrt{(e^i)^2}},\ M_1^i = \frac{1}{\sqrt{(e^i)^2}},\ M_j^i = \frac{X_{j+1}^i}{\sqrt{(e^i)^2}}, \qquad (4)$$

По методу ОМНК построена новая модель с гомоскедастичными остатками:

$$A^i = \underset{(10,15)}{-3,23}\,N^i + \underset{(572,17)}{2705.39}\,M_1^i + \underset{(9,28)}{23,19}\,M_2^i - \underset{(2,55)}{10,59}\,M_3^i - \underset{(0,61)}{3,9}\,M_4^i - \underset{(10,86)}{51,69}\,M_5^i + \qquad (5)$$
$$\underset{(0,02)}{0,29}\,M_6^i + \underset{(13,4)}{18,61}\,M_7^i + \underset{(0,74)}{2,13}\,M_8^i - \underset{(0,02)}{0,01}\,M_9^i + \underset{(0,1)}{0,48}\,M_{10}^i$$

Проведенное тестирование показало, что модель (5) с достаточной точностью позволяет рассчитывать стоимость квартиры в городе Красноярске за 2013 г.

Список литературы

1. Информационная система оценки стоимости квартир на вторичном рынке жилья как инструмент управления инвестициями./ С.И.Сенашов и [др.]/Вестник СибГАУ, 2009, в.4(25), часть 1, с.219-223.
2. С.И.Сенашов, Юферова Н.Ю., Грошак Е.В. Моделирование стоимости жилья в г. Красноярске. Красноярск, изд-во СибГТУ, 2007, 204с.
3. S.I.Senashov and at all. Cost estimation of information system of apartments at secondary housing markets as a management investment tool. Vestnik scientific journal of Siberian Aerospace University, 2009, 5(26) pp.154-157.

4. Актуальное моделирование недвижимости в Красноярске./ С.И.Сенашов и [др.]/Вестник СибГАУ, 2013, в.2(48), с.86-91
5. Филюшина Е.В.,Томаровская И.В. Квартиры Красноярска 2014 Свидетельство о государственной регистрации базы данных № 2014620939