**Московский государственный университет имени М.В. Ломоносова**

**Факультет вычислительной математики и кибернетики**

**Кафедра математических методов прогнозирования**

Толстихин И. О.

# Применение логических алгоритмов классификации в задаче прогнозирования оттока клиентов

Отчет о проделанной работе

**Москва**
**2008**

# Содержание

# 1 Введение

## 1.1 История вопроса

Привлечение и удержание клиентов — чрезвычайно важная задача для любых коммерческих фирм. Особенно актуален этот вопрос в условиях жёсткой конкуренции. Примером может служить телекоммуникационная отрасль, предоставляющая долгосрочные услуги своим клиентам.

На рынке мобильных операторов последнее время наблюдается большой рост числа новых компаний. Это усугубляет положение более зрелых операторов со сложившейся репутацией. Главной проблемой крупных компаний становится отток клиентов, связанный с богатым выбором, представленным на рынке. Основным приоритетным направлением многих поставщиков телекоммуникационных услуг становится именно удержание клиентов.

В настоящее время крупным компаниям отток клиентов ежегодно обходится в миллиарды долларов. Также потеря своих 15 процентов рынка, как показали специалисты, грозит банкротством фирмы. Зачастую, гораздо дешевле, оказывается, проводить ряд акций по удержанию клиентов — рекламы, новые тарифы, персональные услуги. Для этого компания должна определять, какие из клиентов, приносящих большую прибыль, скорее всего откажутся от услуг.

## 1.2 Прогнозирование оттока клиентов

Основная цель анализа оттока клиентов состоит в создании списка контрактов, которые с большой вероятностью в ближайшем будущем будут прерваны. Существуют разные подходы к анализу оттока клиентов. Большинство из них основано на интеллектуальном анализе данных.

Для выявления «опасных» контрактов используются базы данных о клиентах, накопленные компаниями. На основе информации об уже ушедших клиентах и о клиентах, которые продолжают пользоваться услугами компании, требуется построить алгоритм, способный в будущем определять, склонен абонент к уходу, или нет. Таким образом, мы имеем дело с задачей *классификации*.

К алгоритму есть несколько важных требований:

- Его результаты должны хорошо интерпретироваться.
- Данных, как правило, очень много, поэтому он должен эффективно обучаться на сверхбольших выборках.
- Качество классификации должно контролироваться на стадии обучения.

Для прогнозирования ухода клиентов используются различные математические модели, среди них — логистическая регрессия, нейронные сети, деревья решений, бустинг, генетические алгоритмы.

Поскольку компании имеют ограниченные возможности по связям с клиентами, важной задачей также является определение *апостериорной вероятности классификации*. Зная список наиболее вероятных отказчиков, компания может построить оптимальную стратегию проведения акций.

# 2 Логические алгоритмы классификации

## 2.1 Описание данных и постановка задачи

В качестве начальных данных выступают клиенты. Точнее — их контракты с фирмой. У нас есть множество клиентов (*объектов*), мощностью $l : (x_i)_{i=1}^l$. Также имеется $n$ различных признаков, описывающих клиентов. Таким образом, клиента можно отождествлять с вектором в $n$-мерном пространстве: $x_i = (x_i^1, x_i^2, \ldots, x_i^n)$, где $x_i^k$ — $k$-тый признак объекта.

Каждый объект принадлежит одному из двух классов $Y$ — «churn» (клиент, отказавшийся от услуг фирмы) и «nonchurn» (клиент, продолжающий пользоваться услугами компании). То есть $|Y| = 2$. Обозначим их $\{1, -1\}$ соответственно. Пары «объект - класс» $(x_i, y_i)$ называются *прецедентами*. Совокупность прецедентов $X^l = (x_i, y_i)_{i=1}^l$ называется *обучающей выборкой*.

**Задача**. На основе обучающей выборки требовалось научиться классифицировать объекты, не входящие в неё.

Как правило, признаки в задаче прогнозирования оттока клиентов делят на 5 групп:

- **демографические данные о клиенте** — географическое местоположение клиента, а также данные о населении этого региона
- **биллинговые данные** — вся информация, связанная с финансовым аспектом контракта (среднемесячное пополнение счёта, плата за роуминг. . . )
- **данные о контракте** — информация о базовом контракте, о подключённых услугах, о рэйтинге данного тарифа. . .
- **данные об использовании** — информация о звонках (количество , продолжительность и характеристики использованных минут времени. . . )
- **данные о событиях** — информация об изменении тарифных планов, переключения услуг. . .

## 2.2 Определения и обозначения

### Логическая закономерность, информативность предиката

Пусть у нас есть предикат $\varphi \colon X \to \{0, 1\}$. Предикат *покрывает* объект $x$, если $\varphi(x) = 1$. Предикат называют *закономерностью*, если он выделяет достаточно много объектов одного класса и почти не выделяет объекты других классов.

Определим понятие *закономерноси* более строго.

Для этого введём понятия «позитивных» объектов и «негативных» объектов. Допустим, $P_c$ - число объектов выборки $X^l$, принадлежащих классу $c$. $N_c$ — число объектов других классов ($N_c = l - P_c$). *Позитивными* называются объекты, выделенные предикатом $\varphi$ из числа $P_c$ объектов класса $c$. Все остальные объекты, выделенные предикатом, называются *негативными*.

Пусть предикат $\varphi$ выделил $p_c(\varphi)$ позитивных объектов и $n_c(\varphi)$ негативных объектов. Введём ещё две величины:

- Доля негативных среди всех выделенных объектов:

$$E_c\left(\varphi, X^l\right) = \frac{n_c}{n_c + p_c}$$

- Доля выделенных позитивных объектов:

$$D_c\left(\varphi, X^l\right) = \frac{p_c}{l}$$

**Определение 2.1** *Предикат $\varphi$ будем называть логической $\varepsilon, \delta$-закономерностью для класса $c \in Y$, если $E_c\left(\varphi, X^l\right) \leq \varepsilon$ и $D_c\left(\varphi, X^l\right) \geq \delta$ для заданного достаточно малого $\varepsilon$ и достаточно большого $\delta$* $(\varepsilon, \delta \in [0, 1])$.

Если множество объектов $X$ — *вероятностное пространство*, выборка $X^l$ *случайная, независимая и одинаково распределённая*, а $\phi(x)$ и $y(x)$ - случайные величины, то можно проверить гипотезу о независимости событий $\{\varphi(x_i) = 1\}$ и $\{y(x_i) = c\}$. При условии справедливости гипотезы, вероятность одновременного наблюдения числа позитивных объектов, равного $p_c$, и негавтивных объектов, равного $n_c$, подчиняется *гипергеометрическому распределению*:

$$h\begin{pmatrix} p & n \\ P & N \end{pmatrix} = \frac{C_P^p C_N^n}{C_{P+N}^{p+n}}, \qquad 0 \leq p \leq P, \quad 0 \leq n \leq N \tag{2.1}$$

Если эта вероятность мала, а наблюдение $(p, n)$ всё равно реализовалось, то гипотеза о независимости отвергается.

Таким образом, мы можем ввести критерий информативности предиката $\varphi$ относительно конкретного класса.

**Определение 2.2** *Информативность предиката $\varphi(x)$ относительно класса $c \in Y$ по выборке $X^l$ есть*

$$I_c\left(\varphi, X^l\right) = -ln\, h\begin{pmatrix} p_c(\varphi) & n_c(\varphi) \\ P_c & N_c \end{pmatrix} \tag{2.2}$$

*Если $I_c\left(\varphi, X^l\right)$ больше наперёд заданного достаточно большого числа $I_0$, то предикат $\varphi(x)$ будем называть статистической закономерностью для класса $c$.*

*Логическим алгоритмом классификации* будем называть композицию закономерностей.

## 2.3 Алгоритм синтеза логических закономерностей

Логические закономерности должны иметь простой вид. Хороший вариант - представлять закономерность в виде конъюнкции предикатов, имеющих простую структуру. Договоримся называть число предикатов, входящих в состав конъюнкции, *рангом* этой конъюнкции.

Синтез логических закономерностей проводился в два этапа. Вначале проводилась *бинаризация исходных данных*. Затем на основе полученных элементарных предикатов строился список информативных конъюнкций. Опишем подробно оба этапа.

### 2.3.1 Бинаризация данных

Рассмотрим произвольный признак из нашего признакого пространства $f\colon X \to D_f$. Этот признак порождает семейство предикатов — индикаторов попадания признака в определённые подмножества $D_f$. Если $f$ — номинальный или порядковый признак, то как определять подмножества понятно (так как в этом случае $f$ принимает фиксированное конечное число значений). В этом случае предикаты имеют вид:

$$\beta(x) = [f(x) = d], \quad d \in D_f$$

в случае номинального признака и

$$\beta(x) = [f(x) \le d],\ \beta(x) = [d \le f(x)] \quad d \in D_f \tag{2.3}$$

в случае порядкового признака. (Предикаты вида ( 2.3) договоримся называть *элементарными*).

Если же $f$ —количественный признак (то есть $D_f = \mathbb{R}$), то возникает вопрос, как строить предикаты аналогичным образом. Для этого множество значений количественного признака $f$ на обучающей выборке $X^l$ делилось на информативные зоны с помощью *жадного алгоритма слияния зон.*

**«Жадный алгоритм слияния зон»**. Идея алгоритма состоит в том, чтобы разбить множество значений количественного признака $f$ на зоны. Это делается, чтобы определить пороги для формирования предикатов вида ( 2.3). При этом алгоритм стремится минимизировать потерю информации.

В начале своей работы алгоритм сортирует значения признака $f$ на обучающей выборке $X^l$ по возрастанию: $f(x_{i_1}) \le f(x_{i_2}) \le \cdots \le f(x_{i_l})$. Затем разбивает множество значений признака на маленькие зоны, расставляя пороги $d_i$ ровно посередине между соседними точками $f(x_{i_k})$ и $f(x_{i_{k+1}})$, если $y(x_{i_k}) \ne y(x_{i_{k+1}})$. В результате, если было расставлено $r$ порогов, то мы получили $r+1$ элементарных предикатов:

$\xi_0(x) = [f(x) \le d_1]$,

$\xi_m(x) = [d_m \le f(x) \le d_{m+1}],\ m = 1 \dots r$,

$\xi_r(x) = [d_r \le f(x)]$.

Легко показать, что расстановка порогов между элементами одного класса ведёт к уменьшению информативности зоны.

Далее алгоритм ищет «сгустки» порогов и сливает их в одну точку так, чтобы чередование классов в соседних зонах не нарушалось. Этот шаг является эвристикой и направлен на сокращение начального числа элементарных предикатов.

После этих подготовительных этапов алгоритм начинает укрупнять зоны путём слияния троек соседних зон. Этот процесс происходит путём многократного последовательного прохода по всем порогам. Зоны сливаются до тех пор, пока статистическая информативность какой-либо из слитых зон $\xi_{k-1} \vee \xi_k \vee \xi_{k+1}$ превышает информативности исходных зон $\xi_{k-1}, \xi_k$ или $\xi_{k+1}$ больше чем на $\delta I$. $\delta I$ — параметр алгоритма. Альтернативным условием останова является достижение заранее заданного числа зон.

Для ускорения алгоритма, на каждом проходе сливается сразу несколько троек зон по следующему принципу: после слияния одной тройки мы отступаем на несколько зон (на 3 зоны), чтобы предотвратить «схлопывание» всех зон с первой.

Сливаются именно тройки соседних зон, для того чтобы сохранялся порядок чередования классов.

С учётом применя эвристик для сокращения перебора, сложность алгоритма равна $O(\ell)$.

*Замечание по реализации.* Удобно постоянно хранить вектор информативностей зон, чтобы на каждом шаге не пересчитывать их. При слиянии зон, информативности слитых зон удаляются из вектора, на их место вставляется информативность итоговой зоны.

### 2.3.2 Генерация закономерностей в форме конъюнкций

В результате жадного алгоритма мы получаем множество пороговых значений признака $f$ $(d_1, \ldots, d_{t_f})$. Каждому порогу соответствует два элементарных предиката $\xi_{i_1}(x) = [f(x) \leq d_i]$ и $\xi_{i_2}(x) = [d_i \leq f(x)]$.

Повторяя эту процедуру для заданных признаков, мы получим некоторое множество элементарных предикатов $\mathfrak{B}$.

Закономерности искались в виде конъюнкций. Для построения конъюнкций применялся алгоритм ТЭМП.

**«Алгоритм ТЭМП синтеза конъюнкций»**.

Алгоритм строит список «хороших» конъюнкций рангом не более заданного числа $R$.

В реализации алгоритма ТЭМП для сокращения полного перебора конъюнкций применялся ряд эвристик.

Синтез конъюнкций начинается с формирования списка конъюнкций ранга 1. Из множества $\mathfrak{B}$ выбирается $T$ самых информативных предикатов и заносятся в список $L$. Затем начинается наращивание конъюнкций. Алгоритм основан на обходе дерева конъюнкций в ширину. На $r$-м шаге на основе конъюнкций ранга $r$, входящих в $L$, наращиваются конъюнкции ранга $r+1$. Для этого к каждой конъюнкции списка ранга $r$ поочерёдно добавляется предикат из множества $\mathfrak{B}$, не входящий в неё. При оценке качества конъюнкции используется два критерия: информативность конъюнкции $I_c\left(\varphi, X^l\right)$ и доля её ошибок $E_c\left(\varphi, X^l\right)$. Если информативность больше заданного порога $I_0$, а доля ошибок меньше порога $E_{min}$, то нарощенная конъюнкция заносится в список $L$. Конъюнкция попадает в список только если её информативность превышает инфрмативность худшей конъюнкции в списке. При этом плохая конъюнкция выкидывается.

Этот процесс продолжается до тех пор, пока алгоритм не попадёт на $R+1$-й шаг, или пока на $r$-м шаге он не сможет найти в списке конъюнкцию ранга $r$.

Стоит отметить зависимость работы алгоритма от параметра $T$. Если взять $T$ равное единице, ТЭМП по сути превратится в жадный синтез конъюнкции — на каждом шаге алгоритм будет искать терм, который максимизирует прирост информативности конъюнкции. Этот предикат и добавится к конъюнкции.

Если $T$ положить равным $\infty$, то ТЭМП произведёт полный перебор конъюнкций.

*Замечание по реализации.* При наращивании стоит хранить список конъюнкций отсортированным по уменьшению информативности. Это позволит избежать поиска минимума информативности конъюнкций списка при каждой попытке добавления правила в список.

К построенным таким образом списку применяется ещё два процесса — *стабилизация* и *редукция*.

**Стабилизация** заключается в следующем. Совершается попытка улучшить ка-

чество конъюнкции поочередной заменой или удалением всех элементарных предикатов, входящих в её состав. При этом замена осуществляется только на элементарные предикаты, построенные по тому же признаку.

Этот процесс позволяет найти локальный максимум информативности конъюнкции с заданным набором признаков. Его достоинство в том, что после стабилизации при попытке «подвигать» вручную пороги конъюнкции ничего хорошего не выйдет — информативность конъюнкции будет только падать.

*Информативности в стабилизации считаются по обучающей выборке.*

**Редукция** - попытка упростить конъюнкцию путём отброса элементарных предикатов. Идея та же, что и при стабилизации. Но теперь термы только удаляются.

Важным отличием редукции от стабилизации является тот факт, что *информативности конъюнкций при стабилизации считаются по контрольной выборке, заранее выделенной для этого процесса.*

Применение редукции увеличивает качество логического правила. При отбросе терма конъюнкция начинает выделять больше объектов, а значит повышается общность этого правила.

# 3 Обучение логических алгоритмов классификации по сверхбольшим выборкам

Как уже упомяналось, обучающие выборки в задачах прогнозирования оттока клиентов очень большие. В этом есть свои достоинства. Имея огромную выборку, мы можем получать большое количество подвыборок тоже немаленьких размеров. А эти выборки можно уже использовать по-разному: выделять большую выборку в качестве контрольной, использовать выборки для обучения алгоритма и так далее.

Сильный упор при построении алгоритма построения классификатора делался именно на наличии большого запаса обучающих данных.

### Схема решения задачи

Предлагается решать задачу классификации в три этапа. Обучающая выборка делится на 3 части, и каждый из этапов проводится по соответствующей подвыборке.

- Синтез закономерностей-конъюнкций по первой подвыборке.
- Отбор покрывающего набора закономерностей из числа построенных по второй подвыборке.
- Оценивание апостериорных вероятностей классов по третьей (контрольной) подвыборке.

Классификатор предполагается осуществлять в виде простого голосования правил.

## 3.1 Синтез закономерностей

Конъюнкции ищутся на выборке $X^{l_1}$ достаточно малого размера с помощью упомянутого в прошлом разделе алгоритма (жадный алгоритм + ТЭМП). Правила строятся отдельно для каждого из двух классов. Признаки, по которым строятся

конъюнкции, отбираются заранее. При этом параметры алгоритма подбираются таким образом, чтобы выделить избыточное число конъюнкций. То есть $I_0$ в ТЭМП занижается, а $E_{min}$ завышается.

Это делается для того, чтобы набрать как можно больше правил. Чем больше правил — тем больше шанс встретить действительно хорошую закономерность среди них.

## 3.2 Построение покрывающего набора правил

На прошлом шаге мы получили набор конъюнкций. На этом этапе нам нужно выбрать из них такой набор, чтобы голосавание, построенное по найденным закономерностям, имело хорошую обобщающую способность.

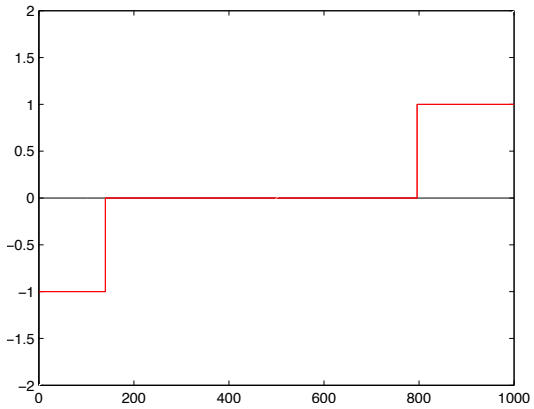
Опишем для начала общую структуру алгоритма простого голосования.

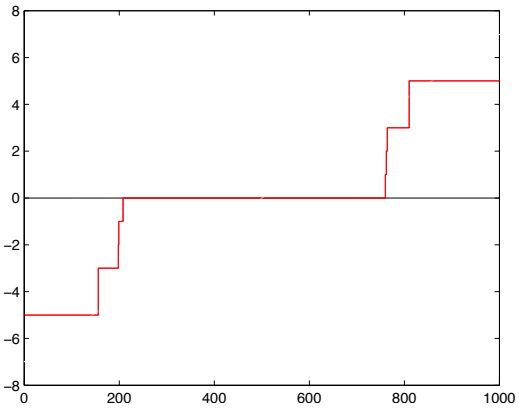
$$a(x) = sign\left(\sum_{i=1}^{T_+}\varphi_i^+(x) - \sum_{i=1}^{T_-}\varphi_i^-(x)\right) \tag{3.1}$$

Здесь $T_+$ и $T_-$ — число правил классов $+1$ и $-1$ соответственно. $\varphi_i^+(x)$ и $\varphi_i^-(x)$ — сами правила. Объект $x$ присваевается тому классу, за которого проголосовало больше правил.

Введём понятие *отступ* алгоритма $a$ на объекте $x_i$ обучающей выборки. Обозначим то, что стоит в скобках в (3.1), $\Gamma(x)$. Тогда отступ (margin) $M(x_i) = \Gamma(x_i)y(x_i)$. Очевидно, что отступ на объекте отрицательный тогда и только тогда, когда алгоритм $a$ допускает ошибку на этом объекте. Причём абсолютная величина отступа в некоторой степени отображает «качество» классификации объекта. Грубо говоря, если отступ отрицательный и имеет большой модуль, то это означает, что «алгоритму далеко» до правильного ответа на данном объекте.

Ниже представлены примеры отступа объектов обучающей выборки при условии, что в голосовании учавствует только одно правило (слева) и 5 правил (справа). *(Далее всюду выборка отсортирована и перенумерована по возрастанию отступа).*




На рисунке представлены результаты правил, построенных по подвыборке длинной 1000 по первым 10 признакам признакового пространства реальной задачи (описание реальной задачи в разделе «Эксперименты»). Все закономерности строились только для класса $\{+1\}$.

Теперь допустим, что мы уже отобрали $n-1$ правил. Требуется выбрать следующее $n$-е правило. Для этого мы будем искать правило $\varphi_n(x)$, которое максимизирует функционал:

$$\sum_{i=1}^{l} w_i \left(M_n(x_i) - M_{n-1}(x_i)\right)) \to \max \tag{3.2}$$

Здесь $w_i$ - вес $i$-го объекта. $M_{n-1}(x_i)$ — отступ объекта на имеющемся наборе из $n-1$ правил. $M_n(x_i)$ — отступ, пересчитаный после добавления в список голосующих правил $\varphi_n(x)$. Функционал будем вычислять по второй выборке большего размера.

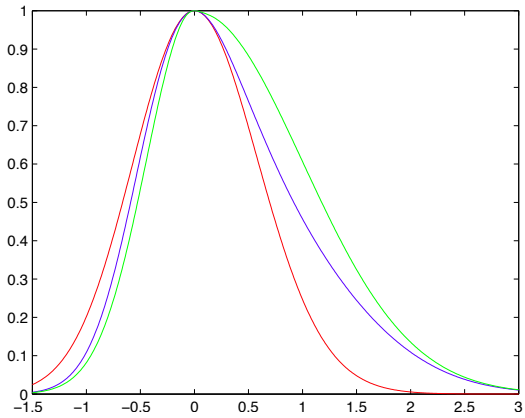
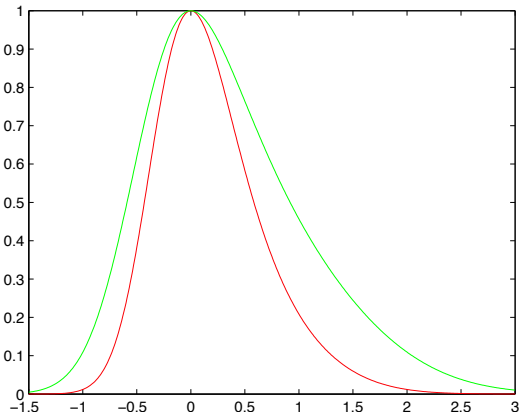
Весовая функция $w_i$ ищется в виде функции от отступа: $w_i = w(M(x_i))$. Веса вводятся для того, чтобы штрафовать объекты, покрытые большим числом правил по сравнению с другими объектами. Нам нужно отбирать правила так, чтобы объекты были покрыты примерно одинаковым числом правил. Это уменьшает вероятность ошибки алгоритма, исходя из неравенства Чебышёва.

В случае классификации с двумя классами можно установить соотвтетствие между величиной отступа и числом правил, выделяющих данный объект. Если у объекта достаточно большой по модулю отступ, то, скорее всего, его покрывает большое число правил. Такие объекты мы будем брать с меньшими весами. Если же отступ объекта близок к нулю, тут возможно два варианта. Первый — его просто выделяет малое число правил. Второй — этот объект плохо классифицируется данным набором правил. Значит, нам нужно правило, которое с одной стороны выделяет этот объект, с другой — увеличивает его отступ. Такие объекты мы будем брать с большими весами.

Весовую функцию ($W(M)$) предлагается представить в виде «шапочки» с пиком в нуле. При этом у алгоритма есть допустимая доля ошибок — таким образом, «подъём» крайней левой части отступа нам не так интересен, как ,по крайней мере, фиксация крайней правой. Этим обусловлена ассиметрия весовой функции и её перевес вправо. *(Перечисленные требования являются эвристическими).*

Встаёт вопрос, как аналитически записать функцию ($W(M)$)? Вариантов, удовлетворяющих вышеперечисленным требованиям, очень много. Один из них — положить в основу гауссиан с центром в нуле. Дальше можно правую половину утяжелить путём умножения на какую-нибудь монотонную функцию, например на $e^{\alpha x}$. *(Домножение на $e^{\alpha x}$ сместит центр вправо. Чтобы избежать смещение центра, нужно подбирать функцию, у которой в нуле производная равна нулю).* Другой вариант - «слепить» две половины разных гауссианов. Этого можно добиться, если коэффициент перед $x^2$ в экспоненте гауссиана домножить на монотонную функцию от икса. Причём, если она будет убывать, то мы получим нужную нам картину, когда слева уклон круче, чем справа. В качестве такой функции можно взять, например, гиперболический тангенс.

В обоих перечисленных случаях параметры функции отвечают за «ширину» графика функции. Это позволяет нам достаточно эффективно контролировать интересующий нас интервал объектов.

На рисунках представлены графики возможных вариантов весовой функции (без масштабирования по горизонтальной оси):

$$f(x) = \exp(-x^2(a + b(1 - \tanh(cx))))$$

$a$,$b$ и $c$ — параметры функции. На левом рисунке представлены различные варианты параметра $c$. Грубо говоря, параметр $c$ отвечает за «скорость переключения» с одного гауссиана на другой. На правом рисунке — вариация параметров $a$ и $b$. Они отвечают за дисперсии гауссианов.

На данном этапе весовая функция берётся независимой от числа правил в наборе.

После определения $W(M)$ для выбора следующего правила нам остаётся найти $\varphi(x)$, максимизирующий функционал (3.2). Процесс пополнения набора повторяется, пока объекты не будут покрыты заданным количеством закономерностей, или пока доля ошибок не опустится ниже заранее выбранного значения.

## 3.3 Оценивание апостериорной вероятности классов

Последним этапом является оценивание апостериорных вероятностей классов по третьей большой выборке. Прадлегается два варианта оценивания. Первый вариант — простое оценивание апостериорной вероятности по частотам классов, покрываемых закономерностями на третьей контрольной подвыборке. Второй вариант основан на построении нечётких правил.

### 3.3.1 Оценивание по частотам классов на контрольной подвыборке

Самым простым методом оценивание апостериорной вероятности классов является использование частот классов на контрольной подвыборке.

На прошлом шаге мы построили покрывающий набор правил. Пусть на вход нашего алгоритма подан объект $x$. Мы смотрим, какими правилами выделен объект $x$:

$$\left\{\varphi_i(x), i = 1 \dots k \;\middle|\; \varphi_i(x) = 1, \forall i\right\}$$

Для каждого правила $\varphi_i(x)$ мы считаем, какое число объектов какого класса оно выделяет на контрольной подвыборке: $N_i^c$, $c \in Y$. Апостериорные вероятности принадлежности объекта $x$ классам предлагается оценивать следующим образом:

$$\hat{\mathsf{P}}(c \mid x) = \frac{\sum_{i=1}^{k} N_i^c}{\sum_{i=1}^{k} (N_i^+ + N_i^-)}$$

### 3.3.2 Оценивание с помощью построения нечётких правил

Этот способ оценивания основан на предположении о том, что вероятности $\mathsf{P}(y|x)$, $y \in Y$, независимы по признакам, входящим в состав правила $\varphi^y(x)$.

Для каждого правила $\varphi^y(x)$ будем пытаться оценивать вероятность $\mathsf{P}(y|x, \varphi^y(x) = 1)$. Допустим, в правило входит $k$ признаков. Это означает, что, фактически, правилу на вход подаётся не $n$-мерный объект, а $k$-мерная случайная величина, составленная из $k$ значений заданных признаков этого объекта. Предлагается выразить вероятность $\mathsf{P}(y|x, \varphi^y(x) = 1)$ при k-мерной случайной величине $x$ в виде произведения $k$ штук вероятностей $\mathsf{P}(y|x, \varphi_i^y(x) = 1)$ для одномерных случайных величин, где $\{\varphi_i^y(x)\}_{i=1}^k$ — элементарные предикаты, входящие в состав правила. Апостериорные вероятности для отдельных элементарных предикатов, в свою очередь, предлагается представлять в виде одномерных сигмоидных функций

$$p(z) = \frac{1}{1 + \exp(-az + b)},$$

где в данном случае $z$ — значение признака, образующего соответствующий элементарный предикат. $a$ и $b$ - параметры сигмоиды.

Параметры сигмоидных функций для каждого правила будут настраиваться отдельно по внешнему критерию. В данном случае будет максимизироваться функция правдоподобия на объектах третьей контрольной выборки, выделенных данным правилом.

# 4 Эксперименты

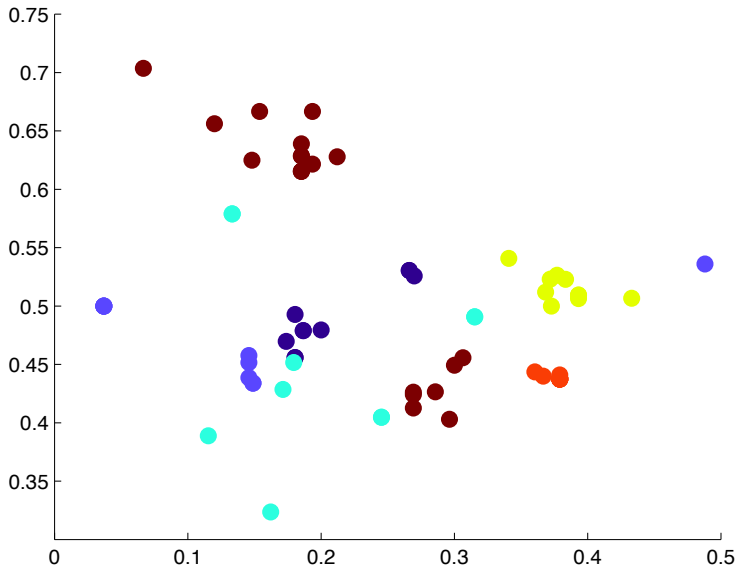



Рис. 1: Правила, построенные по подвыборке длиной 500

В экспериментах использовались данные, предоставленные компанией Teradata (CRM Center, Duke). Обучающая выборка состоит из 100,000 пользователей со 171 признаками. Teradata при формировании выборки применяла «over sampling» — число объектов класса $\{+1\}$ в обучающей выборке исскуственным образом увеличивалось. В результате соотношение объектов класса $\{+1\}$ к $\{-1\}$ составляет почти 50-50.

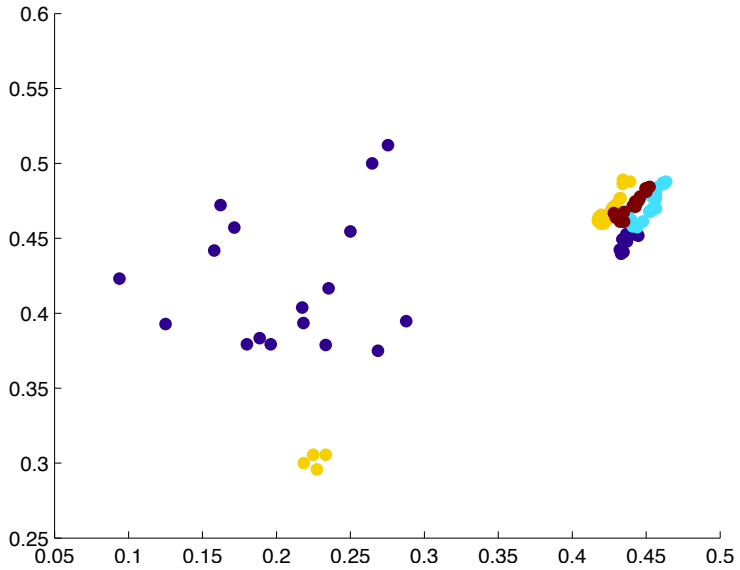



Рис. 2: Правила, построенные по подвыборке длиной 1000

(Точное число — 49,562 отказчиков и 50,438 лояльных клиентов). Данные использовались без нормировки. Никакой предварительной обработки не применялось.

На данном этапе эксперименты преследовали две основных цели. Первая — попытки построения правил по обучающей подвыборке с применением реализованного алгоритма синтеза закономерностей. Вторая — наблюдение поведения этих правил на контрольной подвыборке.

Правила генерировались на обучающих подвыборках длиной 500 и 1000. Подвыборки выбирались случайным образом из генеральной выборки без возвращения и без стратификации. Применялся упомянутый выше алгоритм синтеза правил.

На картинках точкам соответствуют правила. Причём по горизонтали — доля ошибок правила на обучении, по вертикали — на контроле.

При построении правил варьировались 4 параметра: порог информативности $\delta I$ жадного алгоритма слияния зон, порог информативность $I_0$ и порог доли ошибок $E_{min}$ алгоритма ТЭМП, а также длина списка T в алгоритме ТЭМП. Ранг конъюнкции в ТЭМП всюду брался равным 3-м.

На рисунке 1 изображена серия экспериментов при длине обучающей и контрольной подвыборок равной 500. Правила строились по признакам 0 — 60.

| **Рис. 1** | $\delta I$ | $I_0$ | $E_{min}$ | $T$ |
|---|---|---|---|---|
| синяя | -3 | -8 | 0.4 | 10 |
| голубая | -3 | -7 | 0.45 | 10 |
| бирюзовая | -3 | -7 | 0.40 | 10 |
| коричневая | -3 | -9 | 0.42 | 20 |
| жёлтая | -3 | -10 | 0.40 | 10 |
| красная | -3 | -8 | 0.40 | 10 |

На рисунке 2 изображена серия экспериментов при длине обучающей и контрольной подвыборок равной 1000. Правила строились по признакам 0 — 60.

| **Рис. 2** | $\delta I$ | $I_0$ | $E_{min}$ | $T$ |
|---|---|---|---|---|
| синяя | -2 | -9 | 0.42 | 30 |
| голубая | -3 | -7 | 0.45 | 20 |
| коричневая | -1 | -9 | 0.40 | 20 |
| жёлтая | -3 | -8 | 0.40 | 20 |

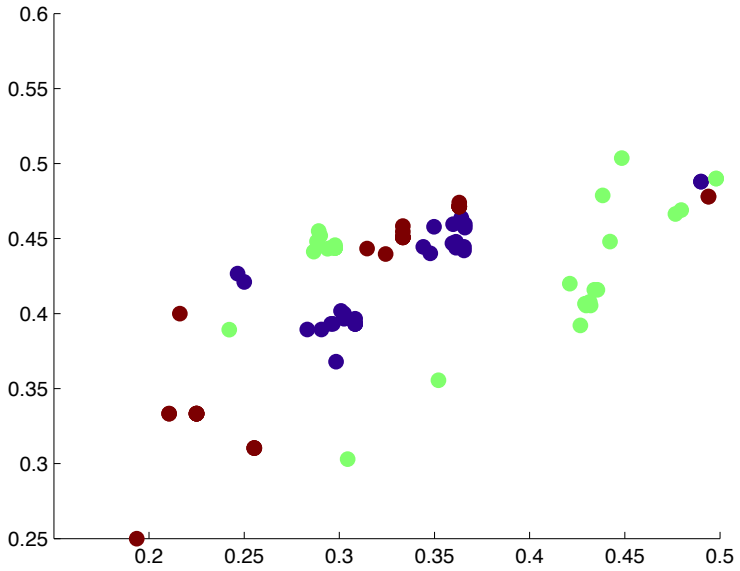



Рис. 3: Использование информативных признаков

На рисунке 3 изображена серия экспериментов при длине обучающей и контрольной подвыборок равной 1000. Признаки, по которым строились правила, были взяты в заключительном отчёте Salford Systems по конкурсу Teradata [4, с. 39]. Номинальные и порядковые признаки предварительно обрабатывались следующим образом: для $i$-го порядкового или номинального признака вводилось $K_i$ меток (целых чисел, начиная от 1), где $\{K_i =$ число различных значений данного признака $\}$. Затем каждое значение просто заменялось на соответствующую ему метку.

| **Рис. 3** | $\delta I$ | $I_0$ | $E_{min}$ | $T$ |
|---|---|---|---|---|
| синяя | -3 | -9 | 0.38 | 30 |
| зелёная | -3 | -7 | 0.40 | 30 |
| коричневая | -3 | -8 | 0.40 | 30 |

*Замечание.* При построении правил алгоритмом ТЭМП была сделана попытка штрафовать генерацию похожих правил. Это контролировалось путём проверки вектора $L$ при добавлении в него новой конъюнкции на наличие конъюнкций с таким же набором признаков.

# 5 Заключение

В работе был приведен обзор существующих методом логической классификации, способных работать с большими и сверхбольшими выборкаи данных. Упор делался на способах синтеза логических закономерностей, на основе которых в будущем строится решающее правило. Ряд методов был применен к выборкам реальных данных, содержащих пользовательскую статистику конкурса Teradata. Эксперименты показывают, что использованные методы синтеза закономерностей имеют ряд недостатков, с которыми в будущем предполагается бороться.

# Список литературы


[1] Воронцов К. В. Лекции по логическим алгоритмам классификации. —М., 2007.

[2] Крамер Г. Математические методы статистики. — М.: Мир, 1975.

[3] Au W., Chan C. C., Yao X. A Novel Evolutionary Data Mining Algorithm with Applications to Churn Prediction // *IEEE Transactions on Evolutionary Computation.* — 2003. —Vol. 7, No. 6. — Pp. 532-545.

[4] Cardell N. S., Golovnya M., Steinberg D. Churn Modeling for Mobile Telecommunications: : Winning the Duke/NCR Teradata Center for CRM Competition // Salford Systems. —2003.

[5] Hadden J., Tiwari A., Roy R., Ruta D. Churn Prediction: Does Technology Matter? // it International Journal of Intelligent Technology. — 2006. —Vol. 1, No. 2. — Pp. 104-110.

[6] Mozer M. C., Wolniewicz R., Grimes D. B., Johnson E., Kaushansky H. Predicting Subscriber Dissatisfaction and Improving Retention in the Wireless Telecommunications Industry // *IEEE Transactions on Neural Networks.* — 2001. — Vol. 11, No. 3. — Pp. 690-696.

[7] Sas Institute Inc. Прогнозирование Оттока Клиентов. // Sas Institute Whate Paper. — 1999.