МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра теории функций и стохастического анализа

АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧ КЛАССИФИКАЦИИ И ИХ ПРИЛОЖЕНИЯ

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

Студентки 4 курса 452 группы
направления 38.03.05 — Бизнес-информатика
механико-математического факультета
Илясовой Светланы Юрьевны

Научный руководитель
заф.каф. ТФиСа, д. ф.-м. н., доцент _____ С. П. Сидоров

Заведующий кафедрой
д. ф.-м. н., доцент _____ С. П. Сидоров

Саратов 2019

ВВЕДЕНИЕ

Актуальность: Во многих отраслях экономической деятельности возникают задачи классификации, решающиеся на основе имеющихся многочисленных данных с применением информационных технологий. В работе рассматриваются некоторые методы решения задач классификации с применением языка программирования R.

В настоящее время бизнес-структурами, финансовыми организациями и предприятиями накоплено огромное количество информации, в том числе и кредитных историй, основываясь на которых можно делать прогнозы на будущее, с помощью алгоритмов классификации. Использование компьютерных технологий облегчает эту задачу.

Целью бакалаврской работы является исследование и программная реализация алгоритмов, решающих задачу классификации, и их сравнительный анализ на реальных данных.

Объект исследования — задачи классификации.

Предмет исследования — методы и алгоритмы решения задач классификации.

Для достижения поставленной цели в работе необходимо решить следующие задачи:

- определить основные понятия, необходимые для анализа методов классификации;
- изучить следующие методы классификации: дерева решений, случайный лес, метод опорных векторов;
- изучить язык программирования R;
- провести анализ трёх наборов данных каждым из методов;
- рассчитать точность работы каждого из методов;
- провести сравнительный анализ эффективности методов с дальнейшим выбором наиболее подходящего.

Практическая значимость проводимого исследования состоит в том, что проведённый анализ поможет в дальнейшем использовать самый удачный метод классификации для наиболее точного прогнозирования кредитных ситуаций и минимизации кредитных рисков.

Работа состоит из введения, четырёх разделов, заключения, списка используемых источников, содержащего 27 наименований и одного приложения. В работу включено 7 рисунков и 25 таблиц. Общий объём работы составляет 59 страниц.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Работа включает в себя следующие разделы:

1. Количественный анализ кредитных рисков.
2. Деревья классификации.
3. Метод опорных векторов.
4. Реализация приложения для решения задач классификации.

Во введении обосновывается актуальность темы работы, формулируется цель работы и решаемые задачи, отмечается практическая значимость полученных результатов.

В первом разделе приводятся основные понятия количественного анализа рисков, а также понятия и содержание скоринговой модели.

Слово «скоринг» происходит от английского «scoring», что означает подсчет очков в игре. Скоринговая система основывается именно на подсчете очков.

Основная задача скоринга — минимизация риска при рассмотрении заявок на выдачу кредита и сокращение времени принятия решений по выдаче кредитов. Но скоринг не только дает ответ на вопрос, сможет ли клиент выплатить кредит или нет, но также позволяет оценить степень финансовой надежности и обязательности клиента.

В основе скоринговых систем лежит метод дедукции, а именно предположение, что люди со схожими социальными показателями ведут себя схоже (одинаково) в одних и тех же ситуациях. На этой концепции строятся многие статистические модели, применяемые при видении бизнеса.

Количественный анализ рисков дает возможность численно определить размеры рисков. Результатом количественного анализа является точная количественная метрика или иной математический показатель, который корректен для сравнения с аналогичными оценками.

К количественному анализу прибегают, чтобы определить, как наиболее существенные факторы риска могут повлиять на эффективность проекта. Здесь анализируются риски, имеющие высокие и умеренные ранги.

Задача количественного состоит в численном измерении влияния изменений рискованных факторов проекта на поведение критериев эффективности проекта.

Наиболее распространенным методом количественного анализа является анализ дерева решений, который мы и будем использовать в нашей работе.

В третьем разделе рассматриваются деревья классификации, включающие в себя метод деревьев решений и метод случайного леса, а также основные понятия и алгоритмы их построения.

Деревья решений являются одним из самых популярных подходов к решению задач интеллектуального анализа данных. Они создают иерархическую структуру классифицирующих правил типа «если..., то...», имеющую вид дерева. Для того чтобы решить, к какому классу отнести некоторый объект или ситуацию, требуется ответить на вопросы, стоящие в узлах этого дерева, начиная с его корня. Для бинарных деревьев вопросы имеют вид «значение параметра A больше x?». Если ответ положительный, осуществляется переход к правому узлу следующего уровня, если отрицательный — то к левому узлу; затем снова следует вопрос, связанный с соответствующим узлом.

Дерево решений можно определить, как структуру, которая состоит из:

— узлов — листьев, каждый из которых представляет определенный класс;

— узлов принятия решений, специфицирующих определенные тестовые процедуры, которые должны быть выполнены по отношению к одному из значений атрибутов; из узла принятия решений выходят ветви, количество которых соответствует количеству возможных исходов тестирующей процедуры.

Более формально дерево можно определить, как конечное множество T, состоящее из одного или множества узлов, таких, что

— имеется один специально обозначенный узел, называемый корнем данного дерева;

— остальные узлы (исключая корень) содержатся в $m \geqslant 1$ попарно непере-

секающихся множествах $T_1, \ldots, T_m$, каждое из которых в свою очередь является деревом. Деревья $T_1, \ldots, T_m$ называют поддеревьями данного корня.

Из данного определения следует, что каждый узел дерева является корнем некоторого поддерева, которое содержится в этом дереве. Число поддеревьев данного узла называется степенью этого узла. Узел с нулевой степенью называется листом. Уровень узла по отношению к дереву T определяется следующим образом: говорят, что корень имеет уровень 1, а другие узлы имеют уровень на единицу выше их уровня относительно содержащего их поддерева T_j этого корня.

Если в дереве существует относительный порядок поддеревьев $T_1, \ldots, T_m$, то говорят, что дерево является упорядоченным; в случае, когда в упорядоченном дереве $m \geqslant 2$, имеет смысл называть T_2 «вторым поддеревом» данного корня и т.д. Если два дерева, отличающиеся друг от друга только относительным порядком узлов поддеревьев, не считать различными, то в этом случае говорят, что дерево является ориентированным, поскольку здесь имеет значение только относительная ориентация узла, а не их порядок.

Метод построения деревьев решений был впервые предложен Р. Куинленом (R. Quinlan) в 1993. Этот метод используется в одном из лучших алгоритмов построения деревьев решений — C4.5.

Обязательные требования к структуре данных и к самим данным, при выполнении которых алгоритм C4.5 будет работоспособен:

1. Описание атрибутов. Данные, необходимые для работы алгоритма, должны быть представлены в виде плоской таблицы. Вся информация об объектах (далее примеры) из предметной области должна описываться в виде конечного набора признаков (далее атрибуты). Каждый атрибут должен иметь дискретное или числовое значение. Сами атрибуты не должны меняться от примера к примеру. Количество атрибутов должно быть фиксированным для всех примеров.
2. Определенные классы. Каждый пример должен быть ассоциирован с конкретным классом, то есть один из атрибутов должен быть выбран в качестве метки класса.
3. Дискретные классы. Классы должны быть дискретными, то есть иметь

конечное число значений. Каждый пример должен однозначно относиться к конкретному классу. Случаи, когда примеры принадлежат к классу с вероятностными оценками, исключаются. Количество классов должно быть значительно меньше количества примеров.

Алгоритм построения дерева решений: Пусть задано множество примеров T, где каждый элемент этого множества описывается m атрибутами. Количество примеров в множестве T будем называть мощностью этого множества и будем обозначать $|T|$. Пусть метка класса принимает следующие значения $C_1, \ldots, C_k$.

Задача будет заключаться в построении иерархической классификационной модели в виде дерева из множества примеров T. Процесс построения дерева будет происходить сверху вниз. Сначала создается корень дерева, затем потомки корня и т.д. На первом шаге мы имеем пустое дерево (имеется только корень) и исходное множество T (ассоциированное с корнем). Требуется разбить исходное множество на подмножества. Это можно сделать, выбрав один из атрибутов в качестве проверки. Тогда в результате разбиения получаются n (по числу значений атрибута) подмножеств и, соответственно, создаются n потомков корня, каждому из которых поставлено в соответствие свое подмножество, полученное при разбиении множества T. Затем эта процедура рекурсивно применяется ко всем подмножествам (потомкам корня) и т.д.

Случайный лес Бреймана — это ансамбль деревьев решений, каждое из которых строится на основе бутстреп выборки из исходной обучающей выборки (баггинг), причем для расщепления вершин аналогично используется только доля случайно отбираемых признаков. Кроме того, строится полное дерево (без усечения). Классификация деревьев в ансамбле осуществляется большинством голосов.

Алгоритм индуктивного построения случайного леса может быть представлен в следующем виде:

1. Для $i = 1, 2, ..., B$ (здесь B — количество деревьев в ансамбле) выполнить:
   - Сформировать бутстреп выборку S размера l по исходной обучающей выборке $D = \{x_i, y_i\}_{i=1}^{l}$;

— По бутстреп выборке S индуцировать неусеченное дерево решений T_i с минимальным количеством наблюдений в терминальных вершинах равным n_{min}, рекурсивно следуя следующему подалгоритму:

а) из исходного набора n признаков случайно выбрать p признаков;

б) из p признаков выбрать признак, который обеспечивает наилучшее расщепление;

в) расщепить выборку, соответствующую обрабатываемой вершине, на две подвыборки;

2. В результате выполнения шага 1 получаем ансамбль деревьев решений $\{T_i\}_{i=1}^{B}$;
3. Предсказание новых наблюдений осуществлять следующим образом:

а) для регрессии:

$f_{rf}^{B}(x) = \frac{1}{B}\sum_{i=1}^{B} T_i(x)$;

б) для классификации:

пусть $\varpi_i(x) \in \{\varpi_1, \varpi_2, ...\varpi_c\}$ —

класс, предсказанный деревом решений Ti,

т. е. $Ti(x) = \varpi_i(x)$; тогда $\varpi_{rf}^{B}(x)$ —

класс, наиболее часто встречающийся в множестве $\{\varpi_b(x)\}_{i=1}^{B}$

В третьем разделе рассматривается метод опорных векторов.

Рассмотрим задачу классификации на два непересекающихся класса, в которой объекты описываются n-мерными вещественными векторами: $X = R^n$, $Y = \{-1, +1\}$.

Будем строить линейный пороговый классификатор:

$$a(x) = \text{sign}(\sum_{j=1}^{n} w_j x^j - w_0) = \text{sign}(\langle w, x\rangle - w_0),$$

где $x = (x^1, ..., x^n)$ — признаковое описание объекта x; вектор $w = (w^1, ..., w^n) \in R^n$ и скалярный порог $w_0 \in R$ являются параметрами алгоритма.

Уравнение $\langle w, x\rangle = w_0$ описывает гиперплоскость, разделяющую клас-

сы в пространстве R^n.

Предположим, что выборка линейно разделима, то есть существуют такие значения параметров w, w_0, при которых функционал числа ошибок

$$Q(w, w_0) = \sum_{j=1}^{l} [y_i(\langle w, x_i \rangle - w_0) < 0]$$

принимает нулевое значение. Но тогда разделяющая гиперплоскость не единственна. Идея метода заключается в том, чтобы разумным образом распорядиться этой свободой выбора. Потребуем, чтобы разделяющая гиперплоскость максимально далеко отстояла от ближайших к ней точек обоих классов.

Первоначально данный принцип классификации возник из эвристических соображений: вполне естественно полагать, что максимизация зазора (margin) между классами должна способствовать более уверенной классификации.

Заметим, что параметры линейного порогового классификатора определены с точностью до нормировки: алгоритм $a(x)$ не изменится, если w и w_0 одновременно умножить на одну и ту же положительную константу. Удобно выбрать эту константу таким образом, чтобы для всех пограничных (т. е. ближайших к разделяющей гиперплоскости) объектов x_i из X^l выполнялись условия

$$\langle w, x_i \rangle - w_0 = y_i.$$

Сделать это возможно, поскольку при оптимальном положении разделяющей гиперплоскости все пограничные объекты находятся от неё на одинаковом расстоянии. Остальные объекты находятся дальше. Таким образом, для всех $x_i \in X^l$

$$\langle w, x_i \rangle - w_0 = \begin{cases} \leq 1 & \text{если } y_i = -1; \\ \geq 1 & \text{если } y_i = +1; \end{cases} \tag{1}$$

Условие $-1 < \langle w, x_i \rangle - w_0 < 1$ задаёт полосу, разделяющую классы. Ни одна из точек обучающей выборки не может лежать внутри этой полосы. Границами полосы служат две параллельные гиперплоскости с направляющим вектором w. Точки, ближайшие к разделяющей гиперплоскости, лежат

в точности на границах полосы. При этом сама разделяющая гиперплоскость проходит ровно по середине полосы.

Чтобы разделяющая гиперплоскость как можно дальше отстояла от точек выборки, ширина полосы должна быть максимальной. Пусть x_- и x_+ — это две произвольные точки классов -1 и $+1$ соответственно, лежащие на границе полосы. Тогда ширина полосы есть

$$\langle (w_+ - x_-), \frac{w}{||w||} \rangle = \frac{\langle w, x_+ \rangle - \langle w, x_- \rangle}{||w||} = \frac{(w_0 + 1) - (w_0 - 1)}{||w||} = \frac{2}{||w||}.$$

Ширина полосы максимальна, когда норма вектора w минимальна.

Итак, в случае, когда выборка линейно разделима, достаточно простые геометрические соображения приводят к следующей задаче: требуется найти такие значения параметров w и w_0, при которых норма вектора w минимальна при условии (1). Это задача квадратичного программирования. Она будет подробно рассмотрена в следующем разделе. Затем будет сделано обобщение на тот случай, когда линейной разделимости нет.

Построение оптимальной разделяющей гиперплоскости сводится к минимизации квадратичной формы при l ограничениях-неравенствах вида (1) относительно $n + 1$ переменных w, w_0:

$$\begin{cases} \langle w, w \rangle \to \min; \\ y_i(\langle w, w \rangle - w_0), \quad i = 1, ..., l. \end{cases} \tag{2}$$

Чтобы обобщить SVM на случай линейной неразделимости, позволим алгоритму допускать ошибки на обучающих объектах, но при этом постараемся, чтобы ошибок было поменьше. Введём набор дополнительных переменных $\epsilon_i > 0$, характеризующих величину ошибки на объектах
$x_i, i = 1, ..., l$. Возьмём за отправную точку задачу (2); смягчим в ней ограничения-неравенства, и одновременно введём в минимизируемый функционал штраф

за суммарную ошибку:

$$\begin{cases} \frac{1}{2}\langle w, w\rangle + C\sum\limits_{i=1}^{l} \to \min_{w,w_0,\epsilon}; \\ y_i(\langle w, w\rangle - w_0) \le 1 - \epsilon_i, & i = 1, ..., l; \\ \epsilon_i \ge 0 & i = 1, ..., l. \end{cases} \tag{3}$$

К этой же оптимизационной задаче приводит ещё одна цепочка рассуждений. Вспомним, что в случае $Y = -1, +1$ отступом (margin) объекта x_i от границы классов называется величина $m_i = y_i(\langle w, x_i\rangle - w_0)$.

Алгоритм допускает ошибку на объекте x_i тогда и только тогда, когда отступ m_i отрицателен. Если $m_i \in (-1, +1)$, то объект x_i попадает внутрь разделяющей полосы. Если $m_i > 1$, то объект x_i классифицируется правильно, и находится на некотором удалении от разделяющей полосы.

Существует ещё один подход к решению проблемы линейной неразделимости. Это переход от исходного пространства признаковых описаний объектов X к новому пространству H с помощью некоторого преобразования $\psi : X \to H$. Если пространство H имеет достаточно высокую размерность, то можно надеяться, что в нём выборка окажется линейно разделимой (легко показать, что если выборка X^l не противоречива, то всегда найдётся пространство размерности не более l, в котором она будет линейно разделима). Пространство H называют спрямляющим.

Постановка задачи, и сам алгоритм классификации зависят только от скалярных произведений объектов, но не от самих признаковых описаний. Это означает, что скалярное произведение $\langle x, x'\rangle$ можно формально заменить ядром $K(x, x')$. Поскольку ядро в общем случае нелинейно, такая замена приводит к существенному расширению множества реализуемых алгоритмов $a : X \to Y$.

Следующие правила порождения позволяют строить ядра в практических задачах.

1. Произвольное скалярное произведение $K(x, x') = \langle x, x'\rangle$ является ядром.
2. Константа $K(x, x') = 1$ является ядром.

3. Произведение ядер $K(x, x') = K1(x, x')K2(x, x')$ является ядром.
4. Для любой функции $\psi : X \to R$ произведение $K(x, x') = \psi(x)\psi(x')$ является ядром.
5. Линейная комбинация ядер с неотрицательными коэффициентами $K(x, x') = \alpha_1 K_1(x, x') + \alpha_2 K_2(x, x')$ является ядром.
6. Композиция произвольной функции $\phi : X \to X$ и произвольного ядра K_0 является ядром: $K(x, x') = K_0(\phi(x), \phi(x'))$.
7. Если $s : X \times X \to R$ — произвольная симметричная интегрируемая функция, то $K(x, x') = \int_X s(x, z)s(x', z)dz$ является ядром.
8. Функция вида $K(x, x') = k(x - x')$ является ядром тогда и только тогда, когда Фурье-образ $F[k](\omega) = (2\pi)^{\frac{n}{2}} \int_X e^{-i\langle\omega,x\rangle} k(x)dx$ неотрицателен.
9. Предел локально-равномерно сходящейся последовательности ядер является ядром.
10. Композиция произвольного ядра K_0 и произвольной функции $f : R \to R$, представимой в виде сходящегося степенного ряда с неотрицательными коэффициентами $K(x, x') = f(K_0(x, x')$, является ядром. В частности, функции $f(z) = e^z$ и $f(z) = \frac{1}{1-z}$ от ядра являются ядрами.

В четвёртом разделе рассматривается язык R, со всеми его достоинствами и недостатками, а также проводится рассмотрение наборов данных с описанием их атрибутов и реализация каждого описанного выше метода на конкретных выборках.

Язык R является одним из технологий анализа больших объемов данных Big data.

Изначально R был разработан сотрудниками статистического факультета Оклендского университета Россом Айхэкой и Робертом Джентлменом. А название языка пришло из первых букв имен создателей — R. Язык и его среда поддерживаются и развиваются организацией R Foundation.

Несмотря на то, что R в первую очередь предназначен для статистиков, он используется не только при обработки статистических данных.

В настоящее время существует множество пакетов и расширений языка, а также различных адаптаций. За двадцать пять лет своего существования, язык R фактически стал стандартом для статистическиз программ.

В R используется интерфейс командной строки. Однако также имеет в

доступе несколько графических интерфейсов пользователя. (Например пакет R Commander, RKWard, RStudio, Weka, Rapid Miner, KNIME)

Таким образом, в настоящее время язык R является одним из ведущих статистических инструментов в мире. Он активно применяется в генетике, молекулярной биологии и биоинформатике, науках об окружающей среде (экология, метеорология) и сельскохозяйственных дисциплинах. Также R все больше используется в обработке медицинских данных, вытесняя с рынка такие коммерческие пакеты, как SAS и SPSS.

В заключении представлены результаты бакалаврской работы.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ

1. Определены основные понятия, необходимые для понимания задач классификации.
2. Изучены алгоритмы методов классификации.
3. Изучен язык программирования R, используемый для реализации методов классификации на основе выборок.
4. Разработано приложение для реализации методов классификации.
5. Проведена количественная оценка точности каждого из методов, выбран наиболее точный метод.