РЕФЕРАТ

Выпускная квалификационная работа содержит 54 страницы, 10 рисунков, 10 таблиц, 11 использованных источников.

ЗАДАЧА КЛАССИФИКАЦИИ, МАШИННОЕ ОБУЧЕНИЕ, АНАЛИЗ ДАННЫХ, ТЕОРИЯ ГРАФОВ, СЛУЧАЙНЫЙ ЛЕС, МЕТОД K-БЛИЖАЙШИХ СОСЕДЕЙ, ПРОЕКТИРОВАНИЕ ПРИЗНАКОВ, МЕТРИКИ КАЧЕСТВА.

Выпускная квалификационная работа посвящена анализу и разработке модели, которая способна предсказать потенциально мошеннические транзакции. Проектируется и разрабатывается прикладная программа для визуализации графов, на основе которой предполагается выявление дополнительных признаков. Проводится анализ, проектирование и отбор признаков. Проводятся различные способы улучшения качества работы спроектированной модели.

Реализованы модели случайного леса и k-ближайших соседей. Обучение модели и преобразование данных спроектированы и реализованы в виде отдельной программы. Полученная модель имеет высокий процент верных предсказаний.

Ставится задача построить алгоритм $a(x): X \to Y$, способный классифицировать произвольный объект $x \in X$.

1.2.1 Нормализация входных данных

Зачастую, чтобы достичь адекватности работы модели, необходимо нормализовать (масштабировать) входные данные. Как будет видно далее работоспособность некоторых моделей зависит от расстояния между объектами, вследствие чего возникает необходимость проведения данной процедуры. Проблема заключается в разных измерениях признаков. Например, если рассматривать погоду, то такие ее признаки как температура, давление, скорость ветра и т.д. измеряются в различных физических величинах, а их числовые значения могут на порядки отличаться.

Нормализовать данные можно разными способами, вот два основных.

Минимаксная нормализация:

$$x_i = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}. \tag{1.1}$$

Данный метод осуществляет переход от абсолютных значений признаков к относительным. Новые переменные будут принимать значения в диапазоне от 0 до 1.

Z-нормализация (масштабирование):

$$x_i = \frac{x_i - \bar{x}}{s}, \tag{1.2}$$

где $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i -$ выборочное среднее, $s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$ – выборочное среднеквадратичное отклонение.

Поскольку не все признаки имеют количественные значения, может применяться создание фиктивных переменных (dummy coding). В этом случае категориальные признаки заменяются бинарными. Например, заменяется признак «пол» на два новых: «пол мужской», «пол женский» со значения 0 и 1.

процедуру нормализации, описанную в разделе 1.2.1. Чем больше значение этой функции, тем менее схожими являются два объекта x, x'.

Для произвольного объекта u расположим объекты обучающей выборки x_i в порядке возрастания расстояний до u: $\rho(u, x_{1;u}) \le \rho(u, x_{2;u}) \le \ldots \rho(u, x_{m;u})$, где $x_{i;u}$ обозначает объект обучающей выборки, который является i-ым соседом объекта u. Аналогичное обозначение введём и для ответа на i-ом соседе: $y_{i;u}$. Таким образом, произвольный объект u порождает свою перенумерацию выборки. В наиболее общем виде алгоритм ближайших соседей выглядит так:

$$a(u) = \arg\max_{y \in Y} \sum_{i=1}^{m} [y(x_{i;u}) = y] w(i,u), \qquad (1.3)$$

где $w(i,u)$ – заданная весовая функция, которая оценивает степень важности i-го соседа для классификации объекта u. Эта функция неотрицательна и не возрастает по i.

Различно задавая весовую функцию, получаются варианты метода ближайших соседей.

- $w(i,u) = [i = 1]$ – простейший метод ближайшего соседа;
- $w(i,u) = [i \le k]$ – метод k ближайших соседей;
- $w(i,u) = [i \le k] q^i$ – метод k экспоненциально взвешенных ближайших соседей, где предполагается $q < 1$ (обычно используется в случае 3-х и более классов).

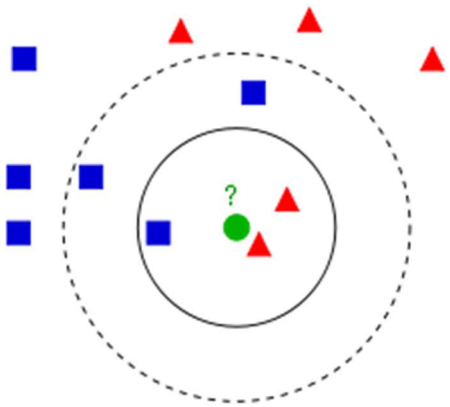

Рис. 1.6. Метод k ближайших соседей

4. Если вычисленное значение равно нулю, то алгоритм переходит в левую дочернюю, в ином случае в правую. Для новой вершины выполняется шаг 2.

Такой алгоритм называется бинарным решающим деревом.

На практике в большинстве случаев используются одномерные предикаты β_v, которые сравнивают значение одного из признаков с порогом: $\beta_v(x; j, t) = [x_j < t]$.

Объект x доходит до вершины *v* тогда и только тогда, когда выполняется конъюнкция $K_v(x)$, составленная из всех предикатов, приписанных внутренним вершинам дерева на пути от корня v_0 до вершины *v*.

Пусть T — множество всех терминальных вершин дерева. Множества объектов $\Omega_v = \{x \in X : K_v(x) = 1\}$, выделяемых терминальными конъюнкциями $v \in T$, попарно не пересекаются, а их объединение совпадает со всем пространством *X*. Данное утверждение легко доказывается индукцией по числу вершин дерева. Отсюда следует, что решающее дерево никогда не отказывается от классификации. А также, что алгоритм классификации $a(x) : X \to Y$, реализуемый бинарным решающим деревом, можно представить в виде простого голосования конъюнкций:

$$a(x) = \arg\max_{y \in Y} \sum_{v \in T} [c_v = y] K_v(x), \qquad (1.4)$$

причем для любого $x \in X$ одно и только одно слагаемое во всех этих суммах равно единице.

Можно увидеть, что для любой выборки можно реализовать решающее дерево, которое не допустит на ней ни одной ошибки. Даже с простыми одномерными предикатами можно сформировать дерево, в каждом листе которого находится ровно по одному объекту выборки. Вероятнее всего, такое дерево будет переобученным и не сможет показать хорошее качество классификации на новых данных.

- критерием останова.

Далее подробнее рассматривается каждый пункт из вышеперечисленного списка.

[8] При построении дерева необходимо задать функционал качества, на основе которого будет осуществляться разбиение выборки на каждом шаге. Пусть R_m – множество объектов, которые попали в вершину, разбиваемую на текущем шаге, а R_l и R_r – подмножества R_m, состоящие из объектов, попадающих при заданном предикате в левое и правое поддерево соответственно. Будет использован функционал следующего вида:

$$Q(R_m, j, s) = H(R_m) - \frac{|R_l|}{|R_m|} H(R_l) - \frac{|R_r|}{|R_m|} H(R_r), \qquad (1.5)$$

где $H(R)$ есть критерий информативности (impurity criterion), который оценивает качество распределения целевой переменной среди объектов множества R. Чем меньше разнообразие целевой переменной, тем меньше должно быть значение критерия информативности — и, соответственно, ставится задача минимизировать его значение. Функционал качества $Q(R_m, j, s)$ при этом необходимо максимизировать.

Как уже обсуждалось выше, в каждом листе дерево будет выдавать константу — вещественное число, вероятность или класс. Исходя из этого, можно предложить оценивать качество множества объектов R тем, насколько хорошо их целевые переменные предсказываются константой (при оптимальном выборе этой константы):

$$H(R) = \min_{c \in Y} \frac{1}{|R|} \sum_{(x_i, y_i) \in R} L(y_i, c) \qquad (1.6)$$

где $L(y_i, c)$ — некоторая функция потерь. Далее рассматривается, какие именно критерии информативности часто используют в задачах классификации.

1.2.3.2 Критерии информативности

Через p_k обозначается доля объектов класса k ($k \in \{1,...,K\}$), попавших в вершину R:

$$p_k = \frac{1}{|R|} \sum_{(x_i,y_i)\in R} [y_i = k], \tag{1.7}$$

Через k_* обозначается класс, представителей которого нашлось больше остальных среди объектов, попавших в данную вершину:

$$k_* = \arg\max_k p_k . \tag{1.8}$$

В пример приводятся два критерия информативности.

Первым будет ошибка классификации. Индикатор ошибки рассматривается, как функция потерь:

$$H(R) = \min_{c\in Y} \frac{1}{|R|} \sum_{(x_i,y_i)\in R} [y_i \neq c], \tag{1.9}$$

Заметно, что оптимальным предсказанием тут будет наиболее популярный класс k_*. Это означает, что критерий будет равен следующей доле ошибок:

$$H(R) = \min_{c\in Y} \frac{1}{|R|} \sum_{(x_i,y_i)\in R} [y_i \neq k_*] = 1 - p_{k_*}, \tag{1.10}$$

Данный критерий считается довольно грубым, так как учитывает частоту p_{k_*} лишь одного класса.

Вторым будет Критерий Джини. Рассматривается ситуацию, когда в вершине выдается не один класс, а распределение по всем классам $c = (c_1,\ldots,c_n), \sum_{k=1}^{K} c_k = 1$. Качество такого распределения можно измерять с помощью критерия Бриера (Brier score):

$$H(R) = \min_{\sum_k c_k = 1} \frac{1}{|R|} \sum_{(x_i,y_i)\in R} \sum_{k=1}^{K} (c_k - [y_i = k])^2 , \tag{1.11}$$

Можно показать, что оптимальный вектор вероятностей состоит из долей классов p_k : $c_* = (p_1, \ldots, p_K)$.

Если подставить эти вероятности в исходный критерий информативности (1.6), провести ряд преобразований, то на выходе получается критерий Джини:

$$H(R = \sum_{k=1}^{K} p_k (1 - p_k). \qquad (1.12)$$

1.2.3.3 Критерии останова

Можно придумать большое количестве критериев останова. Ниже приведены некоторые ограничения и критерии:

- ограничение максимальной глубины дерева;
- ограничение минимального числа объектов в листе;
- ограничение максимального количества листьев в дереве;
- останов в случае, если все объекты в листе относятся к одному классу;
- требование, чтобы функционал качества при дроблении улучшался как минимум на s процентов.

С помощью грамотного выбора подобных критериев и их параметров можно существенно повлиять на качество дерева.

$$accuracy = \frac{TP + TN}{TP + TN + FN + FP}. \quad (1.13)$$

На практике такая метрика очень редко используется, потому что является бесполезной в задачах, где количество объектов одного класса сильно превалирует на другим.

Далее введем метрики для оценки работы модели на каждом из классов по отдельности. Первой будет точность (precision):

$$precision = \frac{TP}{TP + FP}, \quad (1.14)$$

Она показывает долю объектов, определенных классификатором как класс C_1, которые в действительности относятся к классу C_1.

Второй метрикой будет полнота (recall):

$$recall = \frac{TP}{TP + FN}, \quad (1.15)$$

она показывает, какую долю объектов класса C_1 из всех его объектов нашел классификатор.

Объединяет последние две метрики F_1-мера. Которая является средним гармоническим точности и полноты:

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (1.16)$$

Очевидно, что F_1-мера будет максимальной при максимальных значениях полноты и точности, и будет близка к нулю, если один из аргументов близок к нулю.

Таблица 2.1 Описание признаков

Название в *.csv* файле	**Смысл**	**Принимаемые значения**
step	Аналог времени (1 step = 1 час)	1–744 (30 дней)
type	тип транзакции	CASH-IN, CASH-OUT, DEBIT, PAYMENT, TRANSFER
amount	сумма перевода	Действительное число больше нуля
nameOrig	ID пользователя-отправителя	ID в формате *‘C165564’*
oldbalanceOrg	баланс отправителя до транзакции	Действительное неотрицательное число
newbalanceOrig	баланс отправителя после транзакции	Действительное неотрицательное число
nameDest	ID пользователя-получателя	ID в формате *‘C165564’* (магазин, если первая буква ID – *M)*
oldbalanceDest	баланс получателя до транзакции	Действительное неотрицательное число
newbalanceDest	баланс получателя после транзакции	Действительное неотрицательное число
isFraud	пометка о мошеннической транзакции	1 – мошенническая 0 – в противном случае

Пример строки в файле:

184,CASH_IN,90687.53,C594792269,14620285.23,14710972.76,C1485172763,166355.91,75668.37,0,0,16,1,1,0,0,0,0