УДК 004.853:519.2

М.В. Дидковская, канд. техн. наук, А.Ю. Гоголев

Сравнительный анализ алгоритмов кластеризации

Представлена математическая постановка задачи категоризации товаров, выделены следующиее этапы ее решения: индексация, классификация и оценка эффективности.

Экспериментальное исследование классификаторов (наивного байесовского классификатора, метода опорных векторов и деревьев принятия решения) показало, что для решения задачи категоризации наиболее эффективным является метод опорных векторов.

The mathematical formulation of the problem of goods' categorization is represented, the following stages of its decision are pointed out: indexing, classification and evaluation.

Experimental study of classifiers (naive Bayes classifier, SVM method and decision tree) has shown that SVM method is the most effective to solve the problem of categorization.

Введение

Одной из главных особенностей современного общества, является постоянный рост темпов производства информации. Это приводит к непропорциональному росту "информационного шума" ввиду слабой структурированности информации, несоответствии формально релевантной информации действительным потребностям и ее многократному дублированию.

Особую проблему это составляет для интернет-магазинов. Основным фактором влияющим на покупательную способность посетителей интернет-магазина, является простая и понятная навигация. Если покупатель оказывается не в состоянии быстро найти требуемый ему товар и необходимую информацию, то интернет-магазин теряет такого клиента. Базы каталогов интернет-магазинов содержат около полутора миллиона товаров различных фирм и их цены. Цены постоянно меняются, часть товаров перестает продаваться, или напротив – в продаже появляются новые товары. По этой причине интернет-каталог вынужден

постоянно обновлять базу своих товаров. А при каждом обновлении базы необходимо категоризировать товары, так как правильно сформированные рубрики каталога позволяют пользователю интуитивно находить нужный ему товар.

На сегодняшний день нет единого подхода к решению задачи категоризации товаров [1]. Одним из ее этапов является построение классификатора. Существует ряд алгоритмов классификации [2] и требуется выбрать наиболее эффективный в рамках поставленной задачи.

Таким образом, целями данной работы является формулировка математической постановки задачи категоризации и сравнительный анализ алгоритмов кластеризации, используемых для ее решения.

1. Математическая постановка задачи категоризации

Пусть D – множество товаров, а $C = \{c_1, \dots, c_{|C|}\}$ – множество заданных категорий. Для того чтобы категоризировать товар нужно поставить в соответствие каждой паре $\langle d_j, c_i \rangle \in D \times C$ булевое значение. Если для пары $\langle d_j, c_i \rangle$ задано значение True, то товар d_j принадлежит категории c_i, если же задано значение False, то d_j не принадлежит категории c_i. Таким образом, классификатор можно представить как функцию $F: D \times C \to \{T, F\}$. На практике, задача сводится к нахождению функции $\widetilde{F}$, аппроксимирующей функцию F, так как в исходных данных (товарах) присутствует доля шума (часть товаров не содержит в себе семантики).

Для построения классификатора возьмем подмножество товаров $\Omega = \{d_1, \dots, d_{|\Omega|}\} \subset D$ и множество категорий C, при этом все элементы Ω уже распределены по категориям. Одним из простейших подходов к проверке эффективности работы классификатора, является разбиение множества Ω на две части $\Omega = Tr \cup Te$ (не обязательно равной мощности), причем $Tr \cap Te = \emptyset$. Tr называется тренировочным множеством. Товары из множества Tr будут использованы для построения (обучения) классификатора. Te называется тестовым множеством. После построения классификатора элементы множества Te будут категоризированы. Сравнивая $\widetilde{F}(d_j)$ (результат выданный классификатором для каждого

товара) и истинное значение $F(d_j)$ для каждого $d_j \in Te$, можно оценить эффективность построенной модели классификатора.

В данной работе предлагается использовать более сложный подход проверки эффективности, который называется кросс-валидация. При таком подходе Ω разбивается на k равных частей $\Omega = Te_1 \cup \ldots \cup Te_k$, после чего строится k классификаторов $\widetilde{F}_i\ (i = 1..k)$. Для построения $\widetilde{F}_i$ в качестве тренировочного множества используется $\Omega \backslash Te_i$, и Te_i используется для тестирования. Результирующая оценка эффективности классификатора, получается как усреднение оценок эффективностей $\widetilde{F}_i\ (i = 1..k)$.

2. Организация процесса категоризации

В процессе категоризации товара можно выделить три этапа:

1) Индексация товаров;
2) Построение классификатора;
3) Оценка эффективности классификатора.

Рассмотрим эти этапы подробнее.

2.1. Индексация товаров

Товары из множестве Ω представляют собой строки и для того, чтобы применить к ним математический аппарат необходимо осуществить предварительное преобразование. На данном этапе строки с названиями товаров преобразовываются в вектора с бинарными координатами. Для осуществления этого преобразования сначала необходимо построить множество термов $Ter = \{t_1, \ldots, t_{|Ter|}\}$, которые будут играть роль координат. Множество термов строится следующим образом. Последовательно обрабатывая каждый товар из Ω из него извлекаются термы на основе заданного регулярного выражения. Например, если регулярное выражение имеет вид: */[a-zA-Z]+[a-zA-Z0-9]{3,20}/*, то согласно с ним товар «ipod shuffle 512 mb» породит термы «ipod» и «shuffle».

После того как сформировано множество термов Ter, каждому товару d_j ставится в соответствие вектор вида $\overrightarrow{d_j} = (w_{1j}, \ldots, w_{ij}, \ldots, w_{|Ter|j})$, где

$$w_i = \begin{cases} 1, & \text{если } t_i \text{ является подстрокой } d_j \\ 0, & \text{если } t_i \text{ не является подстрокой } d_j \end{cases}$$

После преобразования товаров в численную форму, перейдем к построению классификаторов.

2.2. Построение классификатора

Существует множество алгоритмов классификации В данной работе исследуются три из них: наивный байесовский классификатор, дерево принятия решений C4.5 и метод опорных векторов. Рассмотрим их детальнее.

2.2.1. Наивный Байесовский классификатор (NB)

Наивный Байесовский классификатор относится к классу вероятностных классификаторов и основан на теореме Байеса [3]:

$$P(c_i|\overrightarrow{d_j}) = \frac{P(c_i)P(\overrightarrow{d_j}|c_i)}{P(\overrightarrow{d_j})}$$

$P(\overrightarrow{d_j})$ это вероятность того что случайно выбранный товар будет представлен как вектор $\overrightarrow{d_j}$. А $P(c_i)$ это вероятность того случайно выбранный товар будет принадлежать категории c_i. Цель – найти c_h: $P\left(c_h|\overrightarrow{d_j}\right) > P\left(c_i|\overrightarrow{d_j}\right)\ \forall i \neq h$, т.е. категорию для которой вероятность $P(c_i|\overrightarrow{d_j})$ будет максимальна, именно к ней и будет отнесен товар d_j.

Чтобы оценить $P(\overrightarrow{d_j}|c_i)$ делается предположение, что любые две координаты вектора $\overrightarrow{d_j}$ не зависят друг от друга (в этом и проявляется наивность классификатора). Тогда $P(\overrightarrow{d_j}|c_i)$ можно представить следующим образом:

$$P(\overrightarrow{d_j}|c_i) = \prod_{k=1}^{|Ter|} P(w_{kj}|c_i)$$

Теперь нужно оценить $P(w_{kj}|c_i)$. Если обозначить $p_{ki} = P(w_k = 1|c_i)$, то можно записать:

$$P(w_{kj}|c_i) = p_{ki}^{w_{kj}}(1 - p_{ki})^{1-w_{kj}}$$

Неизвестными остаются $P(c_i)$, $P(\overrightarrow{d_j})$и p_{ki}. Численная оценка p_{ki} и $P(c_i)$ дается на основе количества документов и термов в категориях множества Tr. Так, если обозначить количество товаров из Tr в категории c_i как

$count(d, c_i)$, а количество вхождений терма w_k в товары категории c_i как $count(w_k, c_i)$, то:

$$p_{ki} = \frac{count(w_k, c_i)}{\sum_{v=1}^{|Ter|} count(w_v, c_i)}$$

$$P(c_i) = \frac{count(d, c_i)}{\sum_{v=1}^{|Tr|} count(d, c_v)}$$

Оценивать $P(\overrightarrow{d_j})$ не нужно, т.к. $P(\overrightarrow{d_j}) = const\ \forall c_i, i = 1, \dots, |Tr|$, а для принятия решения абсолютные оценки $P(c_i|\overrightarrow{d_j})$ для разных категорий не нужны; необходима только возможность их сравнить и найти максимум.

Зная оценки всех необходимых величин, можно найти наиболее подходящую категорию для каждого товара из тестового множества Te.

2.2.2. Дерево принятия решений C4.5

Алгоритм построения дерева C4.5 основан на алгоритме ID3 [4]. В основе алгоритма лежит понятие энтропии.

Построение дерева осуществляется сверху вниз. Выбирается w_{kj} (одна из координат $\overrightarrow{d_j}$) и Tr разбивается по ее значению на два подмножества $Tr = Tr_1 \cup Tr_2$, где $Tr_1 = \{d_j \in Tr | w_{jk} = 1\}$ и $Tr_2 = \{d_j \in Tr | w_{jk} = 0\}$. Затем данная процедура рекурсивно применяется ко всем подмножествам до тех пор, пока в узле не окажется множество товаров целиком принадлежащих одному из классов.

Рассмотрим, как выбирать координату w_{kj} на каждом шаге. Обозначим количество товаров из категории c_i в множестве Tr как $freq(c_i, Tr)$. Используя понятие энтропии, можно оценить среднее количество информации в битах, необходимое для определения класса товара из тренировочного множества Tr:

$$Info(Tr) = -\sum_{i=1}^{|C|} \frac{freq(c_i, Tr)}{|Tr|} \times log_2\left(\frac{freq(c_i, Tr)}{|Tr|}\right)$$

где $\frac{freq(c_i, Tr)}{|Tr|} = P(c_i)$ – вероятность того, что случайно выбранный товар будет принадлежать категории c_i.

После разбиения на подмножества по координате w_k эту же оценку можно записать в виде:

$$Info_x(Tr) = \sum_{i=1}^{2} \frac{|Tri|}{|Tr|} \times Info(Tr_i)$$

Для выбора координаты на каждом шаге рассчитывается $Gain(w_k)$, количество информации которую дает знание значения w_k:

$$Gain(w_k) = Info(M) - Info_x(M),$$

где M это множество которое необходимо разбить.

Выбирается координата с максимальным $Gain(w_k)$ и на ее основе осуществляется разбиение. Используя данный критерий, строится дерево принятия решений для поставленной задачи.

2.2.3 Метод опорных векторов (SVM)

Суть метода опрных векторов состоит в том, чтобы построить гиперплоскости, которые бы разделяли в пространстве $R^{|Ter|}$ точки (товары) на категории [5]. Для наглядности будем считать, что имеется всего две категории. В случае большего числа категорий, задачу классификации все равно можно свести к случаю двух категорий.

Будем считать, что Tr линейно разделимо. Разделяющая плоскость не единственна (Рис. 1). Требуется выбрать «наилучшую» из всех гиперплоскостей. Для этого потребуем, чтоб плоскость отстояла максимально далеко от «пограничных» точек обеих классов, т.е. чтоб полоса между категориями имела максимальную ширину. Уравнение описывающие гиперплоскость в общем виде имеет вид $\langle w, x\rangle = w_0$. Для построения классификатора SVM найдем параметры w и w_0.

Формализируем требования к гиперплоскости. Домножим w и w_0 на константу таким образом, что $\langle w, x\rangle - w_0 = y_j$ (где $y_j = \pm 1$, это индикатор принадлежности товара d_j к одной из категорий), для всех «пограничных» точек. Это можно сделать, т.к. все «пограничные» точки находятся от гиперплоскости на одинаковом расстоянии. Заметим, что сама гиперплоскость от этого не поменяется.

Тогда, если точки $\overrightarrow{d_+}$ и $\overrightarrow{d_-}$ это две произвольные точки разных категорий, лежащие на границах разделительной полосы, то ширина полосы равна:

$$\langle\left(\overrightarrow{d_+} - \overrightarrow{d_-}\right), \frac{w}{\|w\|}\rangle = \frac{\langle w, \overrightarrow{d_+}\rangle - \langle w, \overrightarrow{d_-}\rangle}{\|w\|} = \frac{(w_0 + 1) - (w_0 - 1)}{\|w\|} = \frac{2}{\|w\|}$$

Следовательно, для максимизации ширины разделяющей полосы, необходимо минимизировать $\|w\|$.

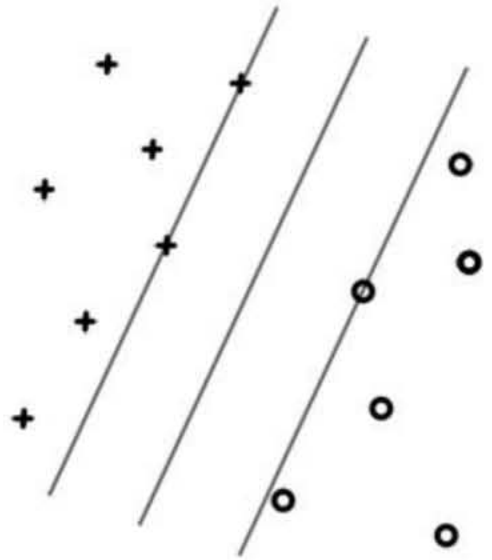

Рис. 1. Разделяющие плоскости

Таким образом, все свелось к минимизации квадратичной формы при $|Tr|$ ограничениях-неравенствах для каждой точки тренировочного множества:

$$\begin{cases} \langle w, w \rangle \to \min \\ y_i(\langle w, x_i \rangle - w_0) \geq 1, \ \ j = 1, \ldots, |Tr| \end{cases}$$

По теореме Куна-Таккера эта задача эквивалентна двойственной задачи поиска седловой точки функции Лагранжа:

$$\begin{cases} \mathcal{L}(w, w_0, \lambda) = \frac{1}{2}\langle w, w \rangle - \sum_{i=1}^{|Tr|} \lambda_j \left(y_j \left(\langle w, \overrightarrow{d_j} \rangle - w_0 \right) - 1 \right) \to \min\limits_{w, w_0} \max\limits_{\lambda} \\ \lambda_j \geq 0, \quad j = 1, \ldots, |Tr| \\ \lambda_j = 0, \text{либо } \langle w, \overrightarrow{d_j} \rangle - w_0 = y_j, \qquad j = 1, \ldots, |Tr| \end{cases}$$

Решая задачу, находим параметры классификатора.

2.3. Оценка эффективности классификаторов

Для оценки результатов используется экспериментальный подход. В данной работе предлагается провести сравнительный анализ работы классификаторов с помощью трех мер эффективности: точности (precision), полноты (recall), и F-меры.

Точность (π_i) определяется как $P\left(\tilde{F}(d_j, c_i) = T \middle| F(d_j, c_i) = T\right)$, т.е. вероятность того что случайный товар из категории c_i будет правильно классифицирован.

Полнота (ρ_i) – вероятность того, что документ отнесенный классификатором к категории c_i, действительно принадлежит к c_i, т.е. $P\left(F(d_j, c_i) = T \middle| \tilde{F}(d_j, c_i) = T\right)$.

F-мера – гармоническое среднее полноты и точности $F = \frac{2\rho\pi}{\rho+\pi}$.

Таблица 1 отображает решения классификатора и эксперта для категории c_i. Например, FP_i – это число товаров из тестового множества Te, которые классификатор отнес к c_i (**ДА**), хотя это не так (**НЕТ**). Аналогичный смысл имеют числа TP_i, FN_i, TN_i.

Таблица 1. Соответствие решений классификатора и эксперта

Категория c_i		Решение эксперта	
		ДА	**НЕТ**
Решение классификатора	**ДА**	TP_i	FP_i
	НЕТ	FN_i	TN_i

Оценки точности и полноты можно представить в виде:

$\tilde{\pi}_i = \frac{TP_i}{TP_i+FP_i}$, $\quad \tilde{\rho}_i = \frac{TP_i}{TP_i+FN_i}$,

после чего можно подсчитать F-меру.

Для сравнения работы классификаторов проведем экспериментальное исследование.

3. Экспериментальное исследование работы классификаторов. Сравнительный анализ.

В качестве тестового множества взято 59416 товаров ($|\Omega| = 59416$), распределенных по 68-ми категориям ($|C| = 68$). Распределение является неравномерным. Самая маленькая категория содержит 78 товаров, а самая большая – 5000 товаров. Вектора представляющие товары имеют размерность 1300 ($|Ter| = 1300$). При кросс-валидации Ω разбивается на десять частей.

Результаты экспериментальных исследований представлены в таблице 2. По строкам расположены классификаторы (NB, C4.5, SVM), по столбцам – их параметры.

Таблица 2. Сравнение параметров классификаторов NB, C4.5, SVM

Название классификатора	% правильно классифицированных товаров	% неправильно классифицированных товаров	Время обучения классификатора (сек.)	Точность	Полнота	F-мера
NB	84.0794	15.9206	41.3	0.85	0.841	0.839
C4.5	~85.45	~14.55	~10*60*60	~0.863	~0.855	0.856
SVM	87.1255	12.8745	121.5	0.887	0.871	0.874

Как видно из таблицы, значение F-меры мало отличается для всех трех классификаторов. Значение F-меры достаточно высоко, а значит применение классификаторов к реальному каталогу целесообразно. Кроме F-меры важно оценить время обучения алгоритма. По этому показателю дерево принятия решений C4.5 значительно проигрывает двум другим алгоритмам. Быстрее всего оказался метод опорных векторов. Сопоставляя показатели быстродействия и F-меры можно сделать вывод, что метод опорных векторов эффективнее других двух алгоритмов справляется с поставленной задачей.

В дальнейшем рекомендуется провести исследование с использованием нейронных сетей в качестве классификаторов, а также значительно существенно увеличить изначальный массив данных Ω и категорий C, что позволит увеличить полноту данных, а следовательно получать более обобщенные результаты.

Выводы

1. Представлена математическая постановка задачи категоризации товаров и выделены основные этапы ее решения, а именно: индексация, классификация и оценка эффективности.
2. Проведено экспериментальное исследование работы наивного байесовского классификатора, метода опорных векторов и деревьев принятия решения, которое показало, что наиболее эффективным является метод опорных векторов.
3. Предложены рекомендации для дальнейшего развития работы, которые предполагают использование нейронных сетей в качестве классификатора, а также расширение тестового множества с целью получения более обощенных результатов исследования.

Литература

1. *Sebastiani F.* A tutorial on automated text categorisation. // Proceedings of the 1st Argentinian Symposium on Artificial Intelligence (ASAI'99). – Buenos Aires, 1999. – p. 7–34.

2. *Калниболотский Ю.М.*, Дидковская М.В. Креативная составляющая при решении интеллектуальных задач (структуризация проблемы) // Электроника и связь, №6 – Киев, 2008. – С. 63-71.

3. *Lewis D.D.* Naive (Bayes) at forty: The independence assumption in information retrieval.// Proceedings of ECML-98, 10th European Conference on Machine Learning. – Chemnitz, 1998. – p. 4–15.

4. *Apt´e C.*, Damerau F., Weiss S.. Automated learning of decision rules for text categorization.// ACM Transactions on Information Systems №12(3). –1994. – p. 233–251.

5. *Salton G.*, Wong A., Yang C.. A vector space model for automatic indexing.// *Communications of the ACM* №18(11). – 1975. – p. 613–620.