Министерство образования и науки Российской Федерации
Московский физико-технический институт (государственный университет)
Факультет управления и прикладной математики
Вычислительный центр им. А. А. Дородницына РАН
Кафедра «Интеллектуальные системы»

Романенко Александр Александрович

Категоризация текстов на основе монотонного классификатора ближайшего соседа

010656 — Математические и информационные технологии

Выпускная квалификационная работа бакалавра

Научный руководитель:
д. ф.-м. н. Воронцов Константин Вячеславович

Москва
2012

Содержание

Аннотация

В работе рассматривается задача категоризации текстовых коллекций. Для ее решения предлагается использовать метод монотонного ближайшего соседа, метод ближайшего соседа и алгоритм бустинга. В работе показано, как адаптировать монотонный метод ближайшего соседа для решения задачи категоризации. Также для улучшения категоризации предлагается использовать методы отбора признаков. В вычислительном эксперименте приводятся результаты работы предложенных методов и алгоритмов категоризации на коллекции 20NewsGroups.

Ключевые слова: *категоризация текстов, алгоритмы классификации, tf-idf, 20NewsGroups, Information Retrieval.*

1 Введение

Задача категоризации (каталогизации, рубрикации) документов, то есть отнесение документа к одной или нескольким темам, является весьма актуальной в связи с ростом объема доступной полнотекстовой информации. Эта задача имеет важные приложения в реальной жизни. Например, новые художественные произведения обычно разделяют по жанрам, а научные статьи часто разделяют по тематике. Еще одно широко распространенное приложение — фильтрация спама, где e-mail сообщения разделяют на две категории: «спам» и «не спам».

Решать задачу катологизации можно с помощью экспертов, которые на основе своего личного опыта будут относить прочитанный документ к наиболее подходящим темам и подтемам. Однако это способ решения очень не эффективен по времени. Также решать задачу категоризации текстов можно автоматически на основе статистических и метрических алгоритмов классификации, используемых в машинном обучении. Всесторонний обзор методов категоризации текстов и их результатов приведен в [2].

Для классификации документов из статистических алгоритмов часто применяют наивный байесовский классификатор [7, 8, 9]. Также хорошо показали себя методы, основывающиеся на алгоритмах тематичекого моделирования LDA и PLSA [10, 11, 12].

Среди метрических алгоритмов для решения задачи категоризации чаще всего применяются метод k ближайших соседей [14, 6], метод Роккио [15] и машина опорных векторов SVM [14, 16]. Для того чтобы применять к текстовым коллекциям алгоритмы машинного обучения, документы обычно представляют в виде вектора действительных чисел [2, 3, 4, 5]. Для того чтобы повысить эффективность применяемых алгоритмов используют методы отбора признаков. Основные способы отбора признаков, используемые в задачах категоризации описаны в [6, 13].

Монотонный метод ближайшего соседа был предложен в [22, 20] как способ построения корректирующей операции для агрегирования алгоритмов классификации в алгебраическом подходе [23]. В [21] монотонный метод ближайшего соседа был успешно применён для решения задачи ранжирования в информационном поиске.

В данной работе монотонный метод ближайшего соседа (monNN) рассматривается не как корректирующая операция над другими алгоритмами, а как самостоятельный алгоритм классификации. Его применение оправдано в тех задачах, где

имеется априорная информация о монотонности функции классификации по признакам: чем больше значение признака, тем чаще объекты относятся к классу 1, а не к классу 0. Такое ограничение естественно возникает в задачах категоризации текстов, в силу того, что каждая категория характеризуется более частым употреблением относительно небольшого подмножества ключевых слов. Объектами в этой задаче являются текстовые документы; признаки могут быть бинарными (слово есть/нет в документе), целочисленными (число вхождений) или вещественными (*tf-idf*). Чтобы алгоритм классификации был монотонным предлагается использовать алгоритмы отбора признаков и методы монотонизации выборки. В случае иерархической категоризации для каждой пары «категория–подкатегория» предлагается решать задачу бинарной классификации, где класс 1 – это документы выбранной подкатегории, класс 0 – все остальные документы.

Также в работе строятся алгоритмы категоризации текстов на основе метода ближайшего соседа (1NN) и на основе бустинга. Предлагается сравнить результаты работы рассматриваемых алгоритмов на коллекции документов 20NewsGroups [17].

2 Задача категоризации текстовых документов

Введем некоторые обозначения. Пусть D — множество (коллекция) текстовых документов, W — множество (словарь) всех употребляемых в них слов, C — множество категорий документов, зафиксированное заранее. Каждый документ $d \in D$ представляет собой последовательность слов $(w_1, \ldots, w_{n_d})$ из словаря W, где n_d — длина документа в словах. Одно и то же слово может повторяться в документе много раз.

Задача категоризации — это задача присвоения булева значения каждой паре $\{d, c\} \in D \times C$. Булево значение 1 означает, что документ d относится к категории c, в то время как значение 0 означает обратное. Более формально, задача категоризации — это задача восстановления неизвестной целевой функции Φ :

$$\Phi : D \times C \to \{1, 0\}.$$

Сформулируем два естественных предположения, которых обычно придерживаются, решая задачу категоризации.

1. Все категории — это только символьные метки, и никакого дополнительного смысла их значения не имеют.

2. При решении задачи категоризации нет никаких дополнительных источников данных, кроме текста документа. В частности, нет файлов с метаданными документов (дата публикации, тип документа, и т. д.).

В [2] описываются различные варианты постановок задачи категоризации, например каждый документ может быть отнесен только к одной категории (одноклассовая катологизация, *single-label categorization*) или документ может быть отнесен к нескольким темам сразу (многоклассовая категоризация, *multi-label categorization*).

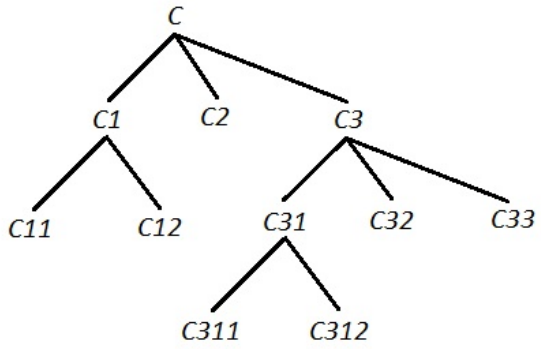


Рис. 1: Пример дерева категорий.

В задачах иерархической категоризации текстов каждый документ может быть отнесен к категориям, к подкатегориям, к подкатегориям подкатегорий и так далее. Таким образом можно говорить о «дереве» категорий. На рис. 1 изображен пример дерева категорий. Из него следует, что, например, «корневая» категория C делится три подкатегории $C1, C2, C3$, которые в свою очередь имеют или не имеют подкатегорий. Для задачи иерархической категоризации можно сделать различные варианты постановок:

- Каждый документ относится всегда только к одной из дочерних подтем. Это означает, что документ будет отнесён в итоге только к одному листу.

- Каждый документ может относится к нескольким дочерним подтемам. В итоге документ может дойти до листьев несколькими путями.

- Документ может не относится ни к одной из дочерних подтем. В итоге документ может не дойти до листьев и останется в теме.

2.1 Сведение задачи категоризации к задаче классификации

Задачу неиерархической категоризации можно рассматривать как задачу многоклассовой классификации, для которой множество классов — это множество категорий C, множество объектов — множество документов D, множество прецедентов — это заранее известное множество пар $\{d, c\}$, где $d \in D, c \in C$.

Задачу иерархической категоризации можно рассматривать как серию задач неиерархической категоризации, то есть как серию задач многоклассовой классификации. Например, для дерева категорий, изображенного на рис. 1, сначала для всего множества документов можно решать задачу категоризации с категориями $C1, C2, C3$. Затем для множества документов категории $C1$, решать задачу категоризации с категориями $C11, C12$, и т. д.

Задачу многоклассовой классификации можно решать с помощью серии задач бинарной классификации. Например, при классификации документов, относящихся к классу $C3$ на классы $C31, C32$ и $C33$ можно решить три задачи двухклассовой классификации: с классами $C31$ и $\overline{C31}$, с классами $C32$ и $\overline{C32}$ и с классами $C33$ и $\overline{C33}$.

Таким образом, мы показали,что задача категоризации сводится к серии задач двухклассовой классификации. Поэтому в дальнейшем будем рассматривать методы решения задач двухклассовой классификации со спецификой, свойственной задачам категоризации.

2.2 Предварительная обработка текстовой коллекции

Перед тем как использовать документы в задаче информационного поиска, они, как правило, подвергаются предобработке.

Лемматизация и стемминг. При построении классификатора текстов нет смысла различать формы (склонения, спряжения) одного и того же слова. Это приведёт к неоправданному разрастанию словаря, дроблению статистики, увеличению ресурсоёмкости и снижению качества модели.

Лемматизация — это приведение каждого слова в документе к его нормальной форме. В русском языке нормальными формами считаются: для существительных — именительный падеж, единственное число; для прилагательных — именительный падеж, единственное число, мужской род; для глаголов, причастий, деепри-

частий — глагол в инфинитиве. Существуют специальные программы — *лемматизаторы*, обычно основанные на явном хранении грамматического словаря со всеми формами слов. Недостатком лемматизации является трудоёмкость составления словарей, и, как следствие, их неполнота, особенно по части специальной терминологии и неологизмов, которые как раз и представляют наибольший интерес для тематического моделирования.

Стемминг — это более простая технология, которая состоит в отбрасывании изменяемых частей слов, главным образом, окончаний. Она не требует хранения словаря всех слов и основана на правилах морфологии языка. Недостатком стемминга является большее число ошибок. Стемминг лучше подходит для английского языка, но хуже для русского.

Отбрасывание стоп-слов. Слова, встречающиеся во многих текстах различной тематики, бесполезны при классификации, и могут быть отброшены. К ним относятся предлоги, союзы, числительные, местоимения, некоторые глаголы, прилагательные и наречия. Число таких слов обычно варьируется в пределах нескольких сотен. Их отбрасывание почти не влияет на длину словаря, но может приводить к заметному сокращению длины некоторых текстов.

Отбрасывание редких слов. Слова, встречающиеся в длинном тексте слишком редко, например, только один раз, также можно отбрасывать, полагая, что данное слово не имеет принципиального значения в данном тексте.

Выделение ключевых фраз. При обработке коллекций научных, юридических или других специальных текстов вместо отдельных слов выделяют ключевые фразы — словосочетания, являющиеся устойчивыми оборотами или терминами в данной предметной области. Это отдельная и довольно сложная задача, которая может решаться методами машинного обучения с привлечением экспертов для формирования обучающих выборок и контроля качества автоматического выделения терминов [1].

Далее будем полагать, что словарь W получен в результате предварительной обработки всех документов коллекции D и может содержать как отдельные слова, так и ключевые фразы. Элементы словаря $t \in W$ будем называть «терминами».

2.3 Векторное представление документа

Рассмотрим один из наиболее распространенных методов представления документа в задачах связанных с информационным поиском — векторную модель документа. В ней каждый документ рассматривается как вектор, состоящий из действительных чисел. Благодаря этому для классификации документов можно пользоваться методами, которые оперируют векторами действительных чисел.

Поставим в соответствие каждому документу вектор длины размера словаря $|W|$:

$$\mathbf{d} = \begin{pmatrix} w_{t_1,d} \\ \vdots \\ w_{t_j,d} \\ \vdots \\ w_{t_{|W|},d} \end{pmatrix}, \tag{2.1}$$

где $w_{t_j,d}$ — вес термина — некоторое число, зависящее от числа вхождений термина $t_j \in W$ в d и характеризующее «важность» термина t_j для понимания, к какому классу относится текст d. Сопоставление терминам документа весов называется взвешиванием, а правила этого сопоставления — схемами взвешивания.

Рассмотрим, некоторые схемы взвешивания.

Схема взвешивания *tf*. Присвоим каждому термину, встретившемуся в документе, вес, зависящий от количества появлений этого термина в данном документе. Если положить вес термина равным количеству вхождений этого термина t в документ d, то получим схему взвешивания, которая называется *частотой термина* и обозначается как $tf_{t,d}$, где индекс t обозначает термин, а индекс d — документ.

Частота в коллекции *cf* и обратная документная частота *idf*. Такой подсчет частоты терминов имеет серьезный недостаток: все термины считаются одинаково важными. Рассмотрим небольшой пример. Коллекция текстов об автомобильной промышленности содержит слово «автомобиль» практически в каждом документе. И хотя этот термин будет иметь высокую частотную характеристику, при рубрикации этой коллекции он не будет очень важен. Чтобы устранить этот недостаток, можно посчитать для каждого термина его *частоту в коллекции cf*, то есть общее количество вхождений термина во все документы коллекции, и снизить веса tf у терминов с высокой частотой в коллекции.

Также для устранения этого недостатка используют *документную частоту df*, представляющую собой количество документов в коллекции, содерщащих термин t.

Пусть N — число документов в коллекции D. Определим *обратную документную частоту* термина t следующим образом:

$$idf_t = \log \frac{N}{df_t}.$$

Таким образом, обратная документная частота редко встречающегося термина является большой, в то время как для часто встречающегося термина она не велика.

Схема взвешивания *tf-idf*. Комбинируя частоту термина в документе *tf* и обратную документную частоту, получают схему взвешивания *tf-idf*:

$$tf\text{-}idf_{t,d} = tf_{t,d} \times idf_t.$$

Вес *tf-idf* термина t в документе d обладает следующими свойствами:

1. Он достигает максимального значения, если термин t встречается много раз в небольшом количестве документов (тем самым усиливая их отличие от других документов).

2. Он уменьшается, если термин встречается в каком-то документе лишь несколько раз или во многих документах.

3. Он достигает минимального значения, если термин встречается практически во всех документах.

Стоит отметить, что в случае, если термин не встречается в документе, то его *tf-idf* вес в этом документе равен 0.

Сублинейное маштабирование *tf*. Кажется маловероятным, что двадцать вхождений термина в документ в самом деле в двадцать раз важнее одного вхождения. По этой причине получила распространение следующая модификация веса *tf*:

$$wf_{t,d} = \begin{cases} 1 + \log tf_{t,d}, & \text{если } tf > 0; \\ 0, & \text{в противном случае.} \end{cases}$$

Тогда вместо схемы *tf-idf* можно рассмотреть схему *wf-idf*:

$$wf\text{-}idf_{t,d} = wf_{t,d} \times idf_t.$$

Нормировка *tf* на максимальный *tf* в документе. Для каждого документа d введем обозначение $tf_{max}(d) = \max_{\tau \in d} tf_{\tau,d}$, где индекс τ пробегает по всем терминам документа d. Нормированную частоту термина t в документе d можно определить по формуле

$$ntf_{t,d} = a + (1-a)\frac{tf_{t,d}}{tf_{max}(d)}.$$

Здесь a — величина, лежащая между нулем и единицей (как правило, ее полагают равной 0,4). Она называется сглаживающим коэффициентом. Нормировка частоты термина по максимуму предназначена для того, чтобы избежать следующей аномалии: в более длинных документах наблюдаются более высокие частоты терминов просто потому, что в более длинных документах чаще содержатся повторяющиеся слова.

Аналогично величине *tf-idf* получают величину *ntf-idf*.

Однако нормировка величины *tf* по максимуму имеет следующие недостатки:

1. Данный метод является неустойчивым: изменения списка стоп-слов может резко изменить веса терминов.

2. Документ может содержать аномальный термин с необычно высокой частотой встречаемоти, который не отражает содержание документа.

2.4 Способы измерения качества классификации

Пусть каждый документ $d \in D$ относится к классу $y_d \in Y$, алгоритм классификации $a : \mathbb{R}^{|W|} \to Y$ относит документ d к классу a_d. В таблице 1 введены обозначения для количеств документов в подмножествах, на которые разбивается множество D после классификации бинарным классификатором $a : \mathbb{R}^{|W|} \to \{y, \bar{y}\}$. В задачах информационного поиска и категоризации текстов качество классификации чаще всего измеряют в терминах точности и полноты [2].

Таблица 1: Разбиение множества документов D после классификации алгоритмом $a : \mathbb{R}^{|W|} \to \{y, \bar{y}\}$.

	$\{d \in D : y_d = y\}$	$\{d \in D : y_d \neq y\}$
$\{d \in D : a_d = y\}$	$tp = \lvert\{d \in D : y_d = y \wedge a_d = y\}\rvert$	$fp = \lvert\{d \in D : y_d \neq y \wedge a_d = y\}\rvert$
$\{d \in D : a_d \neq y\}$	$fn = \lvert\{d \in D : y_d = y \wedge a_d \neq y\}\rvert$	$tn = \lvert\{d \in D : y_d \neq y \wedge a_d \neq y\}\rvert$

Определение 1. Точностью (precision) относительно класса $y \in Y$ называется доля правильно классифицированных документов, среди всех документов, отнесенных алгоритмом a к классу y :

$$P_y(a) = \frac{tp}{tp + fp}.$$

Определение 2. Полнотой (recall) относительно класса $y \in Y$ называется доля правильно классифицированных документов, среди всех документов класса y :

$$R_y(a) = \frac{tp}{tp + fn}.$$

Чем больше значения точности и полноты, тем выше качество классификации. В качестве агрегированного показателя, объединяющего точность P и полноту R, принято использовать F_1-меру :

$$F_1 = \frac{2PR}{P + R}.$$

Также часто встречаемая мера качества классификации — правильность (*accuracy*):

$$Acc = \frac{tp + tn}{tp + tn + fp + fn}.$$

Однако эту меру стоит применять только в случае симметричных данных, то есть если количество документов в множествах $\{d \in D : y_d = y\}$ и $\{d \in D : y_d \neq y\}$ примерно равны. В случае сильно несимметричных данных:

$$|\{d \in D : y_d = y\}| \ll |\{d \in D : y_d \neq y\}|,$$

система может относить все документы не к классу y и правильность будет тем больше, чем несимметричнее данные. В задачах многоклассовой классификации данные несимметричны, поэтому ориентироваться на эту меру не стоит.

Обрабатывая коллекцию документов с помощью нескольких бинарных классификаторов, мы часто хотим вычислить единственный агрегированный показатель, объединяющий показатели отдельных классификаторов. Для этого есть два метода. При *макроусреднении* вычисляется обычное среднее значение по классам:

$$P_{macro} = \frac{1}{|Y|} \sum_{y \in Y} \frac{tp_y}{tp_y + fp_y}; R_{macro} = \frac{1}{|Y|} \sum_{y \in Y} \frac{tp_y}{tp_y + fn_y}.$$

При *микроусреднении* объединяются все решения на уровне документов по всем классам, а затем вычисляется мера эффективности по объединенным данным:

$$P_{micro} = \frac{\sum_{y \in Y} tp_y}{\sum_{y \in Y}(tp_y + fp_y)}; R_{micro} = \frac{\sum_{y \in Y} tp_y}{\sum_{y \in Y}(tp_y + fn_y)}.$$

Разница между этими методами может быть велика. Макроусреднение приписывает решениям классификатора для каждого класса одинаковые веса, в то время как микроусреднение приписывает одинаковые веса решениям классификатора на каждом документе, то есть классы с большим количеством документов и решений по ним вносят больший вклад в микроусреднение. Так как мера F_1 определяется в основном числом tp, то при микроусреднении большие классы доминируют над малыми. Следовательно, результаты применения микроусреднения на самом деле оценивают качество системы на крупных классах из текстовой коллекции. Чтобы правильнее оценить качество классификации на малых классах, следует применять макроусреднение.

3 Алгоритмы категоризации текстов

3.1 Методы отбора признаков

Как отмечалось выше, задача категориации сводится к серии задач двухклассовой классификации с классами c и $\overline{c}$, где c — класс документов категории c из заданного множества категорий, а $\overline{c}$ — класс документов, не относящихся к категории c. Как правило в задачах категоризации текстов размер словаря — это десятки тысяч слов. Ясно, что при решении задачи двухклассовой классификации большое количество слов из словаря будут являться либо шумовыми, либо неинформативными признаками. Для того, чтобы повысить качество классификации делают отбор информативных признаков. Рассмотрим некоторые методы отбора признаков, свойственные задачам категоризации.

Жадный отбор признаков на основе бустинга. Бустинг (англ. *boosting*) — это процедура последовательного построения композиции алгоритмов машинного обучения, когда каждый следующий алгоритм стремится компенсировать недостатки композиции всех предыдущих алгоритмов.

Рассмотрим задачу классификации на два класса, $\mathbf{Y} = \{1, -1\}$, 1 — соответствует классу c, -1 — классу $\overline{c}$. Базовый алгоритм $b_t(d)$ возвращают некоторое вещественное значение, положительное значение означает отнесение документа d к

классу 1, отрицательное — классу -1, 0 — отказ от классификации. Чем дальше возвращаемое значение лежит от нуля, тем надежнее алгоритм классифицирует документ.

Рассмотрим алгоритмическую композицию

$$a(d) = \sum_{t=1}^{T} \alpha_t b_t(d), \ d \in D, \text{ где } T \text{ — число используемых базовых алгоритмов.}$$

Определим функционал качества композиции как число ошибок, допускаемых ею на обучающей выборке длины l:

$$Q_T = \sum_{i=1}^{l} \left[y_i \sum_{t=1}^{T} \alpha_t b_t(d_i) < 0 \right].$$

При построении копмозиционного алгоритма с помощью бустинга обычно пользуются следующими эвристиками:

1. При добавлении в композицию слагаемоего $\alpha_t b_t(d)$ оптимизируется только базовый алгоритм b_t и коэффициент α_t при нем, а все предыдущие слагаемые $\alpha_1 b_1(d), \alpha_2 b_2(d), \ldots, \alpha_{t-1} b_{t-1}(d)$ полагаются фиксированными.

2. Пороговая функция потерь аппроксимируется непрерывно дифференцируемой оценкой сверху.

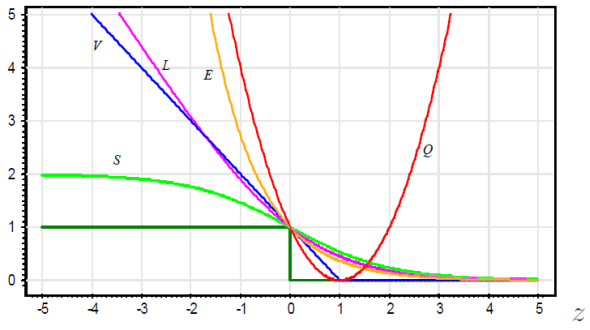


Рис. 2: Гладкие верхние аппроксимации пороговой функции потерь $[z < 0]$.

На рис. 2 показаны различные функции, используемые для оценивания сверху пороговой функции потерь:

- $S(z) = 2(1 + e^z)^{-1}$ — сигмоидная;
- $L(z) = \log_2(1 + e^{-z})$ — логистическая;

- $V(z) = (1-z)_+$ — кусочно-линейная;
- $E(z) = e^{-z}$ — экспоненциальная;
- $Q(z) = (1-z)^2$ — квадратичная.

Заменяя пороговую функцию потерь ее верхней оценкой $\mathcal{L}$, получим

$$Q_T \leqslant \widetilde{Q}_T = \sum_{i=1}^{l} \mathcal{L}\left(y_i \sum_{t=1}^{T} \alpha_t b_t(d_i)\right) = \sum_{i=1}^{l} \mathcal{L}\left(M_{T-1}(d_i) + \alpha_t b_t(d_i)\right),$$

где величина $M_T(d_i) = y_i \sum_{t=1}^{T} \alpha_t b_t(d_i)$ называется отступом документа d_i.

Рассмотрим функцию потерь $\mathcal{L}$ как функцию параметра α_T,

$$\lambda(\alpha_T) = \mathcal{L}(M_{T-1}(d_i) + y_i \alpha_T b_T(d_i)),$$

и линеаризуем ее в окрестности значения $\alpha_T = 0$, разложив в ряд Тейлора и отбросив старшие члены, начиная с квадратичного: $\lambda(\alpha_T) \approx \lambda(0) + \alpha_T \lambda'(0)$. Это приведет к линеаризации функционала $\widetilde{Q}_T$ по параметру α_T :

$$\widetilde{Q}_T \approx \sum_{i=1}^{l} \mathcal{L}(M_{T-1}(d_i)) - \alpha_T \sum_{i=1}^{l} \underbrace{-\mathcal{L}'(M_{T-1}(d_i))}_{w_i} y_i b_T(d_i),$$

где w_i — веса объектов. Если параметр α_T фиксирован, то для минимизации $\widetilde{Q}_T$ необходимо строить базовый алгоритм b_T, исходя из принципа явной максимизации отступов:

$$\sum_{i=1}^{l} w_i y_i b(d_i) \to \max_b.$$

После того, как b_T построен, параметр α_T определяется путем одномерной минимизации функционала $\widetilde{Q}_T$.

В данной работе в качестве верхней оценки пороговой функции потерь использовалась логистическая функция $L(M) = \log_2(1+e^{-M})$, а каждый базовый алгоритм b_i соответствовал компоненте вектора в векторном представлении документа, т. е. слову словаря:

$$b_j(d) = \textit{tf-idf}_{t_j,d} - u_j, \text{ где } j = 1, \ldots, |W|$$

Параметр u_j вычисляется по обучающей выборке:

$$u_j = \textit{tf-idf}_{t_j,1} - \textit{tf-idf}_{t_j,-1}.$$

Здесь $\textit{tf-idf}_{t_j,1}$ — среднее значение *tf-idf* меры j-го термина словаря по всем документам, принадлежащим классу 1, $\textit{tf-idf}_{t_j,-1}$ — среднее значение *tf-idf* меры j-го термина словаря по всем документам, принадлежащим классу -1.

При таком задании базовых алгоритмов жадный выбор базового алгоритма будет означать отбор термина, на котором строится этот базовый алгоритм. Таким образом, процедура отбора признаков с помощью бустинга по обучающей выборке длины l выглядит так:

1. инициализировать отступы: $M_i := 0, i = 1, \ldots, l$;
2. инициализировать параметры u_j базовых алгоритмов b_j :
$$u_j := \textit{tf-idf}_{t_j,1} - \textit{tf-idf}_{t_j,-1},\ j = 1 \ldots, |W|;$$
3. для всех $t = 1, \ldots, T$, где T — требуемое количество признаков:
   - 3.1. $b_t := \arg\max_b \sum_{i=1}^{l} w_i y_i b(d_i)$, где $w_i = -\mathcal{L}'(M_i), i = 1, \ldots, l$
   - 3.2. $\alpha_t := \arg\min_{\alpha>0} \sum_{i=1}^{l} \mathcal{L}(M_i + \alpha b_t(d_i) y_i)$;
   - 3.3. пересчитать отступы: $M_i := M_i + \alpha_t b_t(d_i) y_i, i = 1, \ldots, l$;

Стоит отметить, что при отборе признаков с помощью бустинга, на отобранных признаках строится алгоритм, который также можно использовать для решения задачи классификации.

Отбор признаков с помощью взаимной информации. Этот метод отбора признаков широко распространен при решении задач категоризации. Метод основан на подсчете величины $MI(U, C)$ — ожидаемой взаимной информации о термине t и классе c (англ. *Mutual Information*). Этот показатель измеряет количество информации о принадлежности к классу c, которую несет наличие или отсутствие термина.

$$MI(U, C) = \sum_{i \in \{1,0\}} \sum_{j \in \{1,0\}} P(U = i, C = j) \log_2 \frac{P(U = i, C = j)}{P(U = i) P(C = j)}.$$

Здесь U и C — бернулевские случайные величины: $U = 1$, если документ содержит термин t, $U = 0$ — иначе; $C = 1$, если документ из класса c, $C = 0$ — иначе.

При использовании оценок максимального правдоподобия выражение для MI перепишется в виде:

$$MI(U, C) = \frac{N_{11}}{N} \log_2 \frac{N N_{11}}{N_{1.} N_{.1}} + \frac{N_{01}}{N} \log_2 \frac{N N_{01}}{N_{0.} N_{.1}} + \frac{N_{10}}{N} \log_2 \frac{N N_{10}}{N_{1.} N_{.0}} + \frac{N_{00}}{N} \log_2 \frac{N N_{00}}{N_{0.} N_{.0}}.$$

Здесь N — количество документов, а индексы соответствуют значениям переменных U и C. Например, N_{10} — количество документов, содержащих термин t (т. е. $U = 1$) и не принадлежащих классу c, то есть $C = 0$. Число $N_{1.}$ — это общее количество документов, содержащих термин t, $N = N_{11} + N_{10} + N_{01} + N_{00}$ — общее число документов.

Мера взаимной информации оценивает, сколько информации о классе — в теоретико-информационном смысле — содержит термин. Если распределение термина в классе совпадает с распределением термина во всей коллекции, то $MI(U, C) = 0$. Мера взаимной информации достигает своего максимума, если термин является идеальным индикатором для класса, т. е. если термин присутствует в документе тогда и только тогда, когда документ принадлежит классу.

Таким образом, при решении задачи классификации на классы c и $\overline{c}$ отбор признаков на основе MI можно произвести следующим образом:

1. Для каждого термина словаря подсчитать количество взаимной информации MI о термине t и классе c.

2. Отсортировать весь словарь по убыванию значения MI.

3. Принять за информативные признаки требуемое число первых слов в полученном списке.

3.2 Метод ближайшего соседа

Рассмотрим задачу классификации с множеством объектов $\mathbb{X}$ и множеством классов $\mathbb{Y} = \{1, 0\}$. Пусть $y\colon \mathbb{X} \to \mathbb{Y}$ — функция правильной классификации (целевая зависимость), $a\colon \mathbb{X} \to \mathbb{Y}$ — алгоритм классификации, а на множестве $\mathbb{X} = \{x_1, \ldots, x_L\}$ определена функция расстояния $\rho(x, x')$.

Относительно каждого объекта $x_i \in \mathbb{X}$ расположим все остальные $L - 1$ объектов в порядке возрастания расстояния до x_i, пронумеровав их двойными индексами: $x_i = x_{i0},\ x_{i1},\ x_{i2}, \ldots, x_{i,L-1}$. Таким образом,

$$0 = \rho(x_i, x_{i0}) \leqslant \rho(x_i, x_{i1}) \leqslant \cdots \leqslant \rho(x_i, x_{i,L-1}). \tag{3.1}$$

Метод ближайшего соседа (англ. *nearest neighbor, NN*) — это метод обучения μ, который запоминает обучающую выборку $X \subset \mathbb{X}$ и строит алгоритм $a = \mu X$, отно-

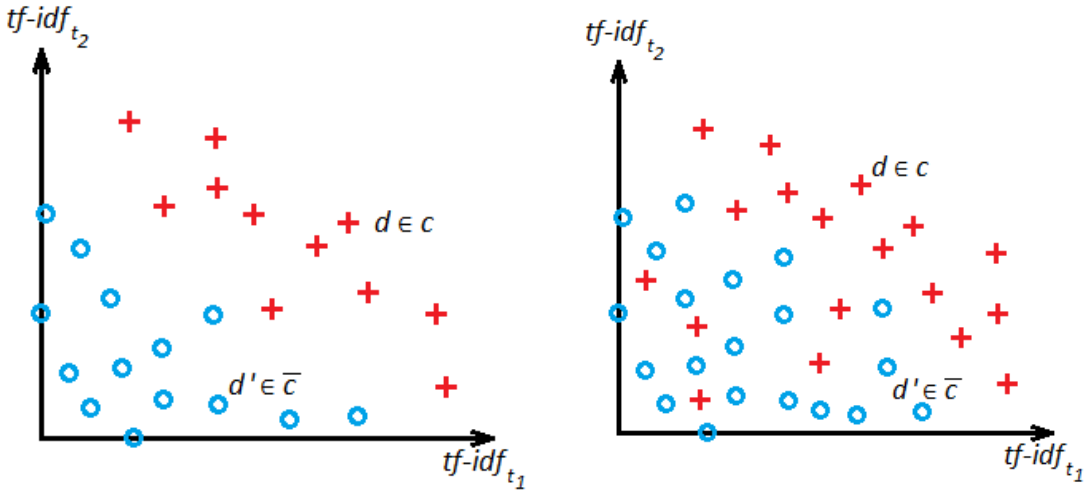
сящий произвольный объект $x \in \mathbb{X}$ к классу его ближайшего обучающего объекта:

$$a(x) = y\big(\arg\min_{x' \in X} \rho(x, x')\big).$$

3.3 Монотонный метод ближайшего соседа

Рассмотрим задачу классификации с множеством объектов $\mathbb{X}$ и множеством классов $\mathbb{Y} = \{1, 0\}$. Теперь предположим, что множество $\mathbb{X}$ частично упорядочено, неизвестная целевая зависимость y монотонна и множество алгоритмов A есть множество всех монотонных функций $a\colon \mathbb{X} \to \mathbb{Y}$ (то есть из $x \leqslant x'$ следует $a(x) \leqslant a(x')$ для всех $x, x' \in \mathbb{X}$).

Заметим, что при решении задачи категоризации текстов предположение о монотонности неизвестной целевой зависимости y вполне естественно. Теоретически, рассматривая задачу классификации с множеством классов $\mathbb{Y} = \{c, \bar{c}\}$, после отбора признаков объекты класса c лежат «выше» объектов класса $\bar{c}$, т. е. выборка монотонная, рис. 3 (а). Конечно, на реальных данных свойство монотонности выборки выполняется очень редко, рис 3 (б). Поэтому, перед тем как применять монотонные алгоритмы классификации, следует «монотонизовать» выборку. О методах монотонизации выборки будет рассказано ниже.



(a) Монотонная выборка

(b) Немонотонная выборка

Рис. 3: Примеры выборок в задаче классификации документов на классы $\mathbb{Y} = \{c, \bar{c}\}$ в двумерном случае, т. е. когда метод отбора признаков оставляет два признака.

Определение 3. Верхней и нижней областью объекта $x_i \in \mathbb{X}$ называются множе-

ства

$$M_i^1 = \{x \in \mathbb{X} : x_i \leqslant x\},$$

$$M_i^0 = \{x \in \mathbb{X} : x \leqslant x_i\}.$$

Определим для произвольной пары объектов $x, u \in \mathbb{X}$ расстояние $r(x,u)$ между нижней областью объекта x и верхней областью объекта u. Потребуем, чтобы функция r обладала следующими свойствами:

1) $r(x,u) = 0$ тогда и только тогда, когда $x \geqslant u$;

2) $r(x,u)$ не возрастает по x не убывает по u.

Функцию $r(x,u)$ можно интерпретировать также и как расстояние от объекта x до верхней области u, и как расстояние от объекта u до нижней области x.

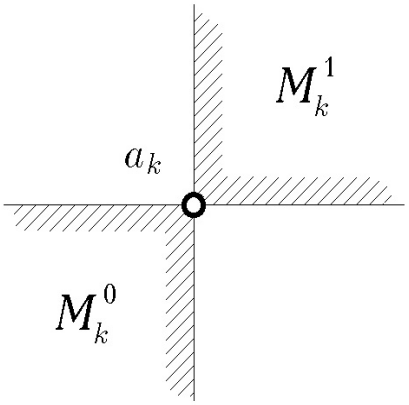


Рис. 4: Верхняя и нижняя область объекта a_k.

Рассмотрим алгоритм ближайшего соседа $a = \mu X$,

$$a(x) = y\big(\arg\min_{x' \in X} \rho(x,x')\big), \tag{3.2}$$

определив функцию расстояния $\rho(x,x')$ от классифицируемого объекта x до объекта обучающей выборки $x' \in X$ как расстояние до его нижней области, если $y(x') = 0$, и до его верхней области, если $y(x') = 1$:

$$\rho(x,x') = \begin{cases} (1-\lambda) r(x',x), & y(x') = 0, \\ \lambda r(x,x'), & y(x') = 1, \end{cases} \tag{3.3}$$

где $\lambda \in (0,1)$ определяет положение разделяющей поверхности внутри зазора между классами:

при $\lambda \to 0$ разделяющая поверхность проходит по нижней границе зазора, и все объекты из зазора относятся к классу 1;

при $\lambda \to 1$ разделяющая поверхность проходит по верхней границе зазора, и все объекты из зазора относятся к классу 0;

при $\lambda = \frac{1}{2}$ разделяющая поверхность проходит посередине зазора, рис. 5.

Для монотонного классификатора ближайшего соседа (3.2) справедлива теорема:

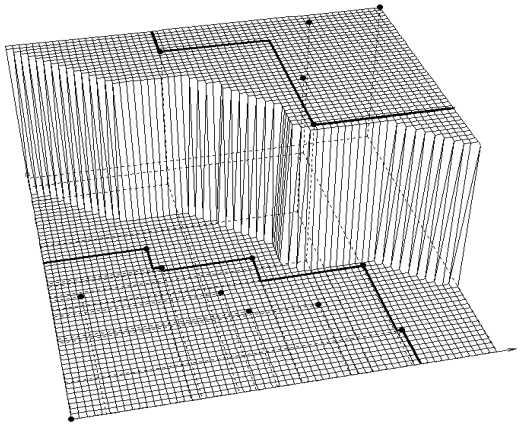


Рис. 5: Монотонная функция, проходящиая через точки монотонной обучающей выборки в $\mathbb{R}^2$. Ступенчатыми линиями показаны нижняя и верхняя границы зазора.

Теорема 1. Пусть выборка $\mathbb{X}$ монотонна, функция $r(x,u)$ удовлетворяет свойствам 1) и 2). Тогда функция $a(x)$ монотонно не убывает по x и проходит через точки обучающей выборки: $a(x) = y(x)$ для всех $x \in X$.

Доказательство можно найти в [20].

Расстояния до верхних и нижних областей. Рассмотрим один естественный способ определить функцию r, предложенный в [20] для построения монотонных корректирующих операций. Пусть объекты задаются n-мерными числовыми векторами, $\mathbb{X} \subset \mathbb{R}^n$, $x = (x^1, \ldots, x^n)$, $u = (u^1, \ldots, u^n)$. Положим

$$r(x,u) = \varphi\left((u^1 - x^1)_+, \ldots, (u^n - x^n)_+\right),$$

где индекс «+» обозначает операцию срезки: $z_+ = z \cdot [z \geqslant 0]$; функция $\varphi(z^1, \ldots, z^n)$ не убывает на всей области определения $[0, +\infty)^n$ и принимает нулевое значение $\varphi(z^1, \ldots, z^n) = 0$ тогда и только тогда, когда $z^1 = \cdots = z^n = 0$.

В качестве функции φ подходят: максимум $\max(z^1, \ldots, z^n)$, сумма $z^1 + \cdots + z^n$, p-норма $\left((z^1)^p + \cdots + (z^n)^p\right)^{\frac{1}{p}}$, число ненулевых аргументов $\#\{i\colon z^i > 0\}$. Произведение $z^1 \cdots z^n$ и минимум $\min(z^1, \ldots, z^n)$ не подходят, так как они принимают нулевые значения не только в точке $(0, \ldots, 0)$.

В ходе вычислительного эксперимента в качестве функции $r(x,u)$ использовалась следующая функция:

$$r(x,u) = \sqrt{((u^1 - x^1)_+)^2 +, \ldots, +((u^n - x^n)_+)^2}. \tag{3.4}$$

Варианты монотонизации выборки. Введем несколько определений.

Определение 4. Дефектом типа 1 называется объект обучающей выборки $x \in X$: $y(x) = 1$, в верхней области M^1 которого есть объект $x' \in X$ такой, что $y(x') = 0$.

Определение 5. Дефектом типа 0 называется объект обучающей выборки $x \in X$: $y(x) = 0$, в нижней области M^0 которого есть объект $x' \in X$ такой, что $y(x') = 1$.

Определение 6. Степенью дефекта $x \in X,\ y(x) = \alpha,\ \alpha = \{0, 1\}$ называется число объектов обучающей выборки $x' \in X : y(x') = 1 - \alpha$, лежащих в области M^α.

Для того, чтобы монотонизировать выборку, нужно удалить из выборки все дефектные объекты типа 0 и типа 1. В зависимости от того, в каком порядке удалять дефектные объекты, в итоге можно получить различные варианты монотонных выборок. Это происходит из-за того, что при удалении хотя бы одного дефекта могут измениться степени оставшихся дефектов. К тому же, если выборка будет существенно не монотонная, то время выполнения процедуры монотонизации при разных способах монотонизации может быть существенно разным.

В работе рассматриваются следующие методы монотонизации выборки:

1. После каждого пересчета степеней дефектов выбираются дефекты с максимальной степенью k. Если все выбранные дефекты одного типа, то они все удаляются из выборки. Если же среди выбранных объектов есть дефекты разных типов, то удаляются только дефекты типа 0. Затем производится пересчет степеней дефектов.

2. После каждого пересчета степеней дефектов для каждого типа выбираются дефекты с максимальной степенью для этого типа. Все выбранные дефекты удаляются из выборки. Затем производится пересчет степеней дефектов.

3. После каждого пересчета степеней дефектов среди дефектов типа 0 и типа 1 случайным образом выбирается по одному дефекту. Пусть степени выбранных дефектов равны k и m. Из выборки удаляются все дефекты типа 0 со степенью k и все дефекты типа 1 со степенью m. Затем производится пересчет степеней дефектов.

Метод 1) отбрасывает меньше объектов, чем методы 2) и 3). Однако время его работы больше. В методе 3) используется рандомизация, и работает этот метод быст-

рее оставшихся. Результаты применения этих метод для монотонизации реальных данных будут описаны в части «Вычислительный эксперимент».

Отбор эталонных объектов. После монотонизации выборки при использовании монотонного метода ближайшего соседа, можно произвести отбор эталонных объектов.

Определение 7. Объект обучающей выборки $x \in X : y(x) = \alpha,\ \alpha \in \{0, 1\}$ называется граничным, если в его области $M^{1-\alpha}$ нет ни одного объекта его класса:

$$\forall x' \in X : x' \in M^{1-\alpha} \Rightarrow y(x') = 1 - \alpha.$$

Ясно, что если при определении функции расстояния 3.3 использовать функцию 3.4, то ответ монотонного классификатора ближайшего соседа 3.2 не измениться, если от всей обучающей выборки оставить только граничные объекты. То есть граничные объекты и будут являться эталонами.

Алгоритм выбрасывания не граничных объектов выглядит следующим образом: Пока есть хотя бы один непросмотренный объект $x \in X$ выполнять:

1. Выбрать случайным образом непросмотренный объект $x,\ y(x) = \alpha,\ \alpha \in \{0, 1\}$
2. Удалить из обучающей выборки все объекты, лежащие в его области M^{α}, кроме самого x.
3. Пометить объект $x \in X$, как просмотренный.

Заметим, что введение в алгоритм отбора эталонов рандомизации позволяет ускорить его работу за счет удаления заведомо не граничных объектов.

4 Вычислительный эксперимент

4.1 Описание данных

Исторически сложилось, что алгоритмы категоризации тестируют на определенных текстовых коллекциях. К наиболее часто используемым коллекциям относятся такие коллекции как Reuters-21578, 20NewsGroups, OHSUMED.

Для тестирования предлагаемых алгоритмов использовалась текстовая коллекция 20NewGroups [17]. Это набор новостных Web-сообщений отсортированных по 20

различным категориям. Объем словаря W этой коллекции без неинформативных слов и без стоп-слов равен 8165 слов. Коллекция 20NewsGroups изначально разбита на обучающую и контрольную выборки. В обучающей выборке 11293 документа, в контрольной — 7528. В таблице 2 приведена информация о числе документов коллекции каждой категории в обучающей и контрольной выборках.

Таблица 2: Детализированная информация о категориях коллекции 20NewsGroups.

Category	Training	Test	Category	Training	Test
alt.atheism	480	319	rec.sport.hockey	600	399
comp.graphics	584	389	sci.crypt	595	396
comp.os.ms-windows.misc	572	394	sci.electronics	591	393
comp.sys.ibm.pc.hardware	590	392	sci.med	594	396
comp.sys.mac.hardware	578	385	sci.space	593	394
comp.windows.x	593	392	soc.religion.christian	598	398
misc.forsale	585	390	talk.politics.guns	545	364
rec.autos	594	395	talk.politics.mideast	564	376
rec.motorcycles	598	398	talk.politics.misc	465	310
rec.sport.baseball	597	397	talk.religion.misc	377	251

4.2 Результаты

Результаты экспериментов по отбору признаков. В ходе этих экспериментов по обучающей выборке для каждой категории коллекции 20NewsGroups на основе алгоритма бустинга и взаимной информации составлялся список слов, которые далее использовались как признаки объектов. Стоит отметить, что перед применением алгоритмов каждый документ обучающей выборки был представлен как вектор из значений *tf-idf* входящих в него слов.

В таблице 3 приведены отобранные указанными алгоритмами первые 15 слов для некоторых категорий коллекции 20NewsGroups. Из таблицы 3 интуитивно ясно, что отобранные слова соответствуют категориям, для которых они отбирались.

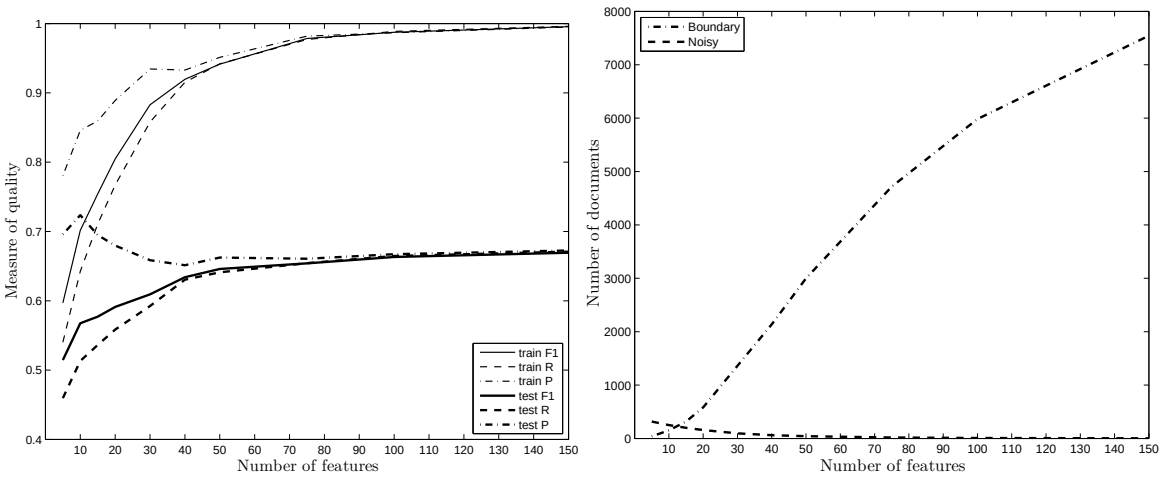
Таблица 3: Первые 15 признаков для некоторых категорий 20NewsGroups, отобранных с помощью алгоритма бустинга и взаимной информации.

Бустинг				Взаимная информация			
№	rec.autos	comp.graphics	sci.med	№	rec.autos	comp.graphics	sci.med
1	car	graphic	msg	1	car	graphic	geb
2	auto	imag	doctor	2	automot	imag	medic
3	wagon	polygon	diseas	3	auto	file	gordon
4	oil	file	pitt	4	ford	gif	diseas
5	engin	tiff	inform	5	dealer	polygon	jxp
6	warn	cview	treatment	6	toyota	tiff	doctor
7	ford	pov	articl	7	dumbest	program	chastiti
8	test	surfac	seizur	8	engin	algorithm	dsl
9	dealer	mpeg	treat	9	mustang	pov	cadr
10	blah	curv	pain	10	sedan	format	intellect
11	brake	anim	test	11	callison	color	pitt
12	tek	mail	health	12	wheel	vga	patient
13	drive	map	scienc	13	drive	ftp	skeptic
14	list	speedstar	work	14	vehicl	cview	bank
15	honda	fractal	gordon	15	dodg	jpeg	diet

Зависимость качества категоризации от числа выбираемых признаков. В ходе этого эксперимента для каждой категории коллекции 20NewsGroups по обучающей выборке для различных значений числа отбираемых признаков строился монотонный классификатор ближайшего соседа. Для решения задачи категоризации документа d, решается последовательно 20 задач классификации с помощью построенных алгоритмов. Качество категоризации измерялось в терминах F_1-меры, точности P, полноты R и правильности Acc. Кроме того, для каждого значения числа признаков считалось усредненное по всем категориям число объектов, отброшенных при монотонизации выборки, и усредненное по всем категориям число эталонных объектов. Стоит отметить, что при построении монотонных классифкаторов использовался первый из описанных выше методов монотонизации.

На рис. 6, 7 приведены результаты описанного эксперимента. Стоит отметить,

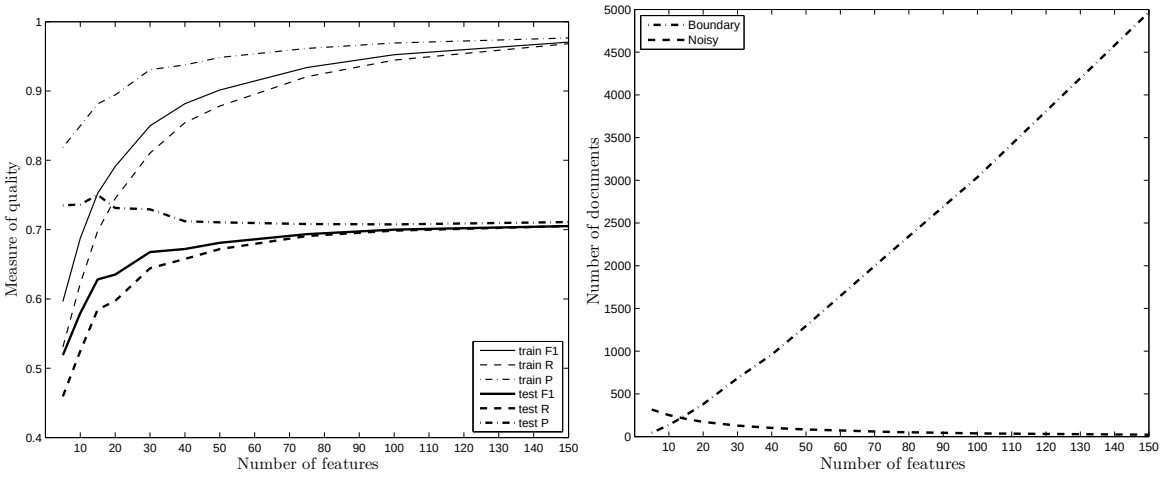
что на рис. 6 (a) и рис. 7 (a) приведены макроусредненные значения мер качества.



(a) Усредненное по всем категориям качество категоризации.

(b) Число шумовых и эталонных объектов.

Рис. 6: Зависимость результатов работы монотонного классификатора ближайшего соседа от числа признаков, отбираемых с помощью алгоритма бустинга.



(a) Усредненное по всем категориям качество категоризации.

(b) Усредненное по всем категориям число шумовых и эталонных объектов.

Рис. 7: Зависимость результатов работы монотонного классификатора ближайшего соседа от числа признаков, отбираемых с помощью взаимной информации.

Из рис. 6, 7 видно, что при любом методе отбора признаков, начиная с числа признаков приблизительно равного 100, значения мер качества на контрольной выборке практически не растут. Однако, алгоритм категоризации, использующий метод

отбора признаков на основе взаимной информации работает лучше и вычислительно эффективнее: при одном и том же числе признаков качество категоризации выше, а число эталонных объектов меньше. Требование малого количества эталонов для каждого классификатора является очень важным, так как в реальных системах, решающих задачи иерархической категоризации с разветвленным деревом категорий, строятся сотни классификаторов, и хранить огромный массив данных для каждого классификатора очень неэффективно.

Для того чтобы оценить насколько сильно ухудшается качество категоризации при сокращении числа эталонных объектов, был проведен эксперимент, в котором для фиксированного числа отобранных признаков из множества эталонных объектов случайным образом удаляются объекты. После каждого удаления объектов производилась оценка качества категоризации на обучающем множестве. Число признаков было выбрано равным 100, метод отбора признаков — бустинг. Результаты эксперимента показаны на рис. 8. Из него видно, что чем больше мы удаляем эталонов, тем меньше качество категоризации, причем скорость ухудшения качества растет с числом удаляемых эталонов. При этом из 6 и 8 понятно, что для того чтобы получить некоторое значение F_1-меры выгоднее уменьшить число признаков, чем удалять эталонные объекты предложенным способом, оставив число признаков неизменным.

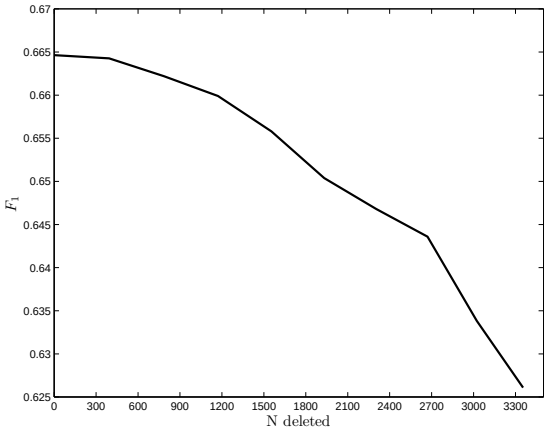


Рис. 8: Зависимость качества категоризации, от числа удаляемых эталонных объектов.

Покажем результаты категоризации для всех категорий коллекции 20NewsGroups при 100 признаках, отобранных с помощью взаимной информации. Результаты сведены в таблицу 4.

Таблица 4: Детализированные результаты категоризации коллекции 20NewsGroups с помощью monNN.

Category	F_1	P	R	Category	F_1	P	R
alt.atheism	0,62	0,65	0,59	rec.sport.hockey	0,90	0,89	0,91
comp.graphics	0,68	0,68	0,68	sci.crypt	0,82	0,81	0,83
comp.os.ms-windows.misc	0,65	0,66	0,64	sci.electronics	0,54	0,59	0,49
comp.sys.ibm.pc.hardware	0,58	0,58	0,59	sci.med	0,66	0,60	0,73
comp.sys.mac.hardware	0,63	0,67	0,59	sci.space	0,84	0,85	0,84
comp.windows.x	0,74	0,79	0,70	soc.religion.christian	0,72	0,67	0,77
misc.forsale	0,53	0,46	0,61	talk.politics.guns	0,72	0,64	0,83
rec.autos	0,78	0,84	0,73	talk.politics.mideast	0,84	0,85	0,83
rec.motorcycles	0,83	0,88	0,80	talk.politics.misc	0,58	0,64	0,53
rec.sport.baseball	0,85	0,86	0,84	talk.religion.misc	0,50	0,55	0,45

Результаты экспериментов по монотонизации выборки. В ходе этих экспериментов, строились алгоритмы, решающие задачу категоризации, состоящие из серии монотонных классификаторов ближайшего соседа, решающих задачи двухклассовой классификации. Число признаков, используемых при классификации равно 100. Разница между алгоритмами была лишь в методе монотонизации выборки. При каждом методе монотонизации измерялось качество работы построенного алгоритма на контрольной выборке.

Таблица 5: Результаты категоризации коллекции 20NewsGroups при использовании различных методов монотонизации.

Бустинг				Взаимная информация			
№ метода	F_1	P	R	№ метода	F_1	P	R
1	0,664	0,666	0,663	1	0,7	0,708	0,698
2	0,663	0,666	0,662	2	0,698	0,703	0,697
3	0,662	0,665	0,661	3	0,695	0,703	0,693

В таблице 5 приведены макроусредненные значения F_1-меры, точности P и полноты R при использовании методов монотонизации, описанных в пункте «Варианты монотонизации выборки». Из нее видно, что при использовании метода монотонизации 1) результаты самые лучшие, метода монотонизации 3) — самые худшие. Однако результаты отличаются не сильно. Также стоит отметить, что быстрее всего монотонизация выборки выполняется при использовании метода 3), медленнее всего — при использовании метода 1). Однако, при использовании эффективных структур данных разница в скорости методов монотонизации ничтожна.

Сравнение алгоритмов категоризации. В ходе этого эксперимента были построены алгоритмы категоризации текстов на основе метода ближайшего соседа 1NN, метода монотонного классификатора ближайшего соседа monNN и на основе бустинга. Для сравнения полученных алгоритмов использовалось одинаковое количество признаков — 100 признаков для каждого классификатора. В таблицах 6 и 7 приведены макроусредненные значения F_1-меры, точности P, полноты R и правильности Acc.

Таблица 6: Результаты сравнения алгоритмов категоризации (отбор признаков с помощью бустинга).

	F_1	P	R	Acc
Бустинг	0,638	0,662	0,640	0,964
monNN	**0,663**	**0,667**	**0,665**	**0,967**
1NN	0,615	0,659	0,596	0,96

Таблица 7: Результаты сравнения алгоритмов категоризации (отбор признаков с помощью взаимной информации).

	F_1	P	R	Acc
monNN	**0,7**	**0,708**	**0,698**	**0,97**
1NN	0,458	0,625	0,393	0,939

Стоит отметить, что согласно [18] для коллекции 20NewsGroups хорошим результатом категоризации считается макроусредненное значение F_1-меры равное 0,667. Согласно [19], где изучается зависимость качества алгоритма категоризации текстов на основе SVM алгоритма от количества признаков, максимальные значения макроусредненной F_1 меры при различных методах отбора признаков расположены в интервале $(0{,}65, 0{,}75)$. Таким образом, алгоритм категоризации на основе monNN решает задачу категоризации лучше, чем алгоритмы на основе бустинга и метода 1NN, причем даже на признаках, оптимизирующих алгоритм бустинга. Кроме того, получаемое качество категоризации можно считать хорошим.

5 Заключение

В данной работе монотонный метод ближайшего соседа рассматривается как самостоятельный алгоритм классификации и адаптируется для решения задачи категоризации текстов. Для этого производится корректировка функции расстояния, используемой в методе ближайшего соседа, и монотонизация выборки несколькими способами. В работе также сравниваются методы монотонизации выборки.

Полученный алгоритм категоризации сравнивается на реальных данных с алгоритмами, построенными на основе метода ближайшего соседа и на основе бустинга. Кроме того, в работе рассматриваются два метода отбора признаков, улучшающие качество категоризации.

В результате было установлено, что для используемой текстовой коллекции алгоритм категоризации на основе monNN работает лучше, чем алгоритмы на основе бустинга и метода 1NN.

Основные результаты работы:

1. для решения задачи категоризации адаптирован метод монотонного ближайшего соседа;
2. предложено два алгоритма отбора признаков;
3. предложено и проведено сравнение трех вариантов процедуры монотонизации выборки;
4. выполнена серия вычислительных экспериментов, иллюстрирующих работу рассмотренных методов и алгоритмов на реальной текстовой коллекции.

Список литературы

[1] *Лукашевич Н. В.* Тезаурусы в задачах информационного поиска – М.: Издательство МГУ имени М. В. Ломоносова, 2011.

[2] *Sebastiani, Fabrizio* Machine learning in automated text categorization // ACM Computing Surveys, 2002, vol. 34, pp. 1-47.

[3] *Luhn H. P.* A Statistical Approach to Mechanized Encoding and Searching of Literary Information // IBM Journal of Research and Development, 1957, vol. 1, pp. 309-317.

[4] *Luhn H. P.* The Automatic Creation of Literature Abstracts // IBM Journal of Research and Development, 1958, vol. 2.

[5] *Papineni K.* Why inverse document frequency? /В кн.:Second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies 2001. – Pittsburgh: Association for Computational Linguistics, 2001, pp. 1-8.

[6] *Manning C. D., Raghavan P., Schtze H.* Introduction to Information Retrieval. – Cambridge: Cambridge University Press, 2008.

[7] *Maron M. E., Kuhns J. L.* On Relevance, Probabilistic Indexing and Information Retrieval // Journal ACM, 1960, vol. 7, pp. 216-244.

[8] *Lewis D. D.* Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval. /В кн.: Proceedings of the 10th European Conference on Machine Learning. – London: Springer-Verlag, 1998.

[9] *Domingos P., Pazzani M.* On the Optimality of the Simple Bayesian Classifier under Zero-One Loss // Machine Learning, 1997, vol. 29, pp. 103-130.

[10] *Blei D. M., Jordan M. I.* Latent dirichlet allocation // Journal of Machine Learning Research, 2003, vol. 3, pp. 993-1022.

[11] *Daud A., Li J., Zhou L., Muhammad F.* Knowledge discovery through directed probabilistic topic models: a survey // Frontiers of Computer Science in China, 2010, vol. 4, pp. 280-301.

[12] *Hofmann T.* Probabilistic latent semantic indexing. /В кн.: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. – NY: ACM, 1999, pp. 50-57.

[13] *Yang Y, Pedersen.* Feature Selection in Statistical Learning of Text Categorization // In Proc. ICML, 1997.

[14] *Воронцов К. В.* Лекции по машинному обучению. http://www.machinelearning.ru

[15] *Rocchio J. J.* Relevance Feedback in Information Retrieval. – Salton, 1971.

[16] *Tsochantaridis I.,Joachims T., Hofmann T., Altun Y.* Large margin methods for structured and interdependent output variables // Journal of Machine Learning Research, 2005, vol. 6, pp. 1453-1484.

[17] *Lang K.* The 20 Newsgroups data set. http://people.csail.mit.edu/jrennie/20Newsgroups/

[18] *Rahmoun A., Elberrichi Z., Bentaalah M. A.* Using WordNet for Text Categorization // The International Arab Journal of Information Technology, 2008, pp. 16-24.

[19] *Forman G.*An extensive empirical study of feature selection metrics for text classification // Journal of Machine Learning Research, 2003, vol. 3, pp. 1289-1305.

[20] *Воронцов К. В.* Оптимизационные методы линейной и монотонной коррекции в алгебраическом подходе к проблеме распознавания // ЖВМ и МФ, 2000, vol. 40, pp. 166-176.

[21] *Spirin N. V.,Vorontsov K. V.* Learning to Rank with Nonlinear Monotonic Ensemble /В кн.: 10th International Workshop on Multiple Classidier Systems. Naples, Italy, June 15–17, 2011. – Lecture Notes in Computer Science. Springer-Verlag, 2011, pp. 16-25.

[22] *Rudakov K. V., Vorontsov K. V.* Methods of Optimization and Monotone Correction in the Algebraic Approach to the Recognition Problem // Doklady Mathematics, 1999, vol. 60.

[23] *Журавлёв Ю. И.* Об алгебраическом подходе к решению задач распознавания или классификации // Проблемы кибернетики, 1978, vol. 33, pp. 5-68.