Методы автоматической классификации текстов

Article *in* Международный журнал Программные продукты и системы · March 2017

DOI: 10.15827/0236-235X.117.085-099

1 author:

Tatiana Batura
A.P. Ershov Institute of Informatics Systems
35 PUBLICATIONS **57** CITATIONS

УДК 004.048
DOI: 10.15827/0236-235X.030.1.085-099
Дата подачи статьи: 19.07.16
2017. Т. 30. № 1. С. 85–99

МЕТОДЫ АВТОМАТИЧЕСКОЙ КЛАССИФИКАЦИИ ТЕКСТОВ

Т.В. Батура, к.ф.-м.н., ведущий научный сотрудник, tatiana.v.batura@gmail.com (Новосибирский государственный университет, ул. Пирогова, 2, г. Новосибирск, 630090, Россия); старший научный сотрудник (Институт систем информатики им. А.П. Ершова СО РАН, просп. Лаврентьева, 6, г. Новосибирск, 630090, Россия)

Классификация текстов является одной из основных задач компьютерной лингвистики, поскольку к ней сводится ряд других задач: определение тематической принадлежности текстов, автора текста, эмоциональной окраски высказываний и др. Для обеспечения информационной и общественной безопасности большое значение имеет анализ в телекоммуникационных сетях контента, содержащего противоправную информацию (в том числе данные, связанные с терроризмом, наркоторговлей, подготовкой протестных движений или массовых беспорядков).

Данная статья представляет собой обзор методов классификации текстов, целями которого являются сравнение современных методов решения задачи классификации текстов, обнаружение тенденций развития данного направления, а также выбор наилучших алгоритмов для применения в исследовательских и коммерческих задачах.

Широко известный современный подход к классификации основывается на методах машинного обучения. В данной статье описываются наиболее распространенные алгоритмы построения классификаторов, проводимые с ними эксперименты и результаты этих экспериментов. Обзор подготовлен на основе выполненных за 2011–2016 гг. научных работ, находящихся в открытом доступе в сети Интернет и опубликованных в авторитетных журналах или в трудах международных конференций, высоко оцениваемых научным сообществом.

В статье произведены анализ и сравнение качества работы различных методов классификации по таким характеристикам, как точность, полнота, время работы алгоритма, возможность работы алгоритма в инкрементном режиме, количество предварительной информации, необходимой для классификации, независимость от языка.

Ключевые слова: *классификация текстов, анализ текстовой информации, обработка данных, машинное обучение, нейронные сети, качество классификации.*

Прогресс в области микроэлектроники и информационных технологий обусловил широкое распространение обработки в реальном времени больших потоков данных. Например, многие простые операции повседневной жизни, такие как использование кредитной карты или телефона, требуют автоматизированного создания, анализа и обработки различных данных. Поскольку эти операции часто выполняются большим числом участников, необходимы распределенные и массовые потоки данных. Точно так же социальные сети содержат большое количество специфических сетевых и текстовых потоков данных. Поэтому актуальна проблема создания моделей и алгоритмов, позволяющих эффективно обрабатывать большие потоки данных, особенно в условиях ограниченных временных и других ресурсов.

Для обеспечения информационной и общественной безопасности важное значение имеет анализ в телекоммуникационных сетях контента, содержащего противоправную информацию (в том числе данных, связанных с терроризмом, наркоторговлей, сетевым экстремизмом, подготовкой протестных движений или массовых беспорядков).

Целями данного обзора являются сравнение современных методов решения задачи классификации текстов, обнаружение тенденций развития данного направления, а также выбор наилучших алгоритмов для применения в исследовательских и коммерческих задачах.

Методы классификации текстов лежат на стыке двух областей – информационного поиска и машинного обучения. Их сходство состоит в способах представления самих документов и способах оценки качества алгоритмов. На сегодняшний день разработано большое количество методов и их различных вариаций для классификации текстов. Каждая группа методов имеет свои преимущества и недостатки, области применения, особенности и ограничения.

Особый интерес представляет случай, когда данные поступают в виде потока, например в телекоммуникационных сетях. Определенные трудности возникают из-за того, что обучение модели всегда основывается на совокупности свойств набора документов. Эти совокупные свойства могут изменяться с течением времени, и при построении потокового классификатора необходимо учитывать возможные изменения исходного распределения данных [1]. Желательно, чтобы выбранный метод мог поддерживать инкрементное обучение, то есть чтобы классификатор обучался на каждом отдельно взятом образце в режиме реального времени. При инкрементном обучении обучающие примеры поступают последовательно в процессе работы алгоритма, так что классификатор должен постоянно корректировать результаты обучения и дообучаться. При неинкрементном обучении вся обучающая выборка предоставляется сразу полностью. Ясно, что в случае инкрементного обучения поведение классификатора в процессе работы меняется, что уменьшает его предсказуемость и может осложнить настройку системы. В то же время инкрементное обучение делает систему гораздо более гибкой, адаптируемой к изменяющимся условиям.

Особенности процесса классификации в потоке связаны еще с тем, что не всегда удается контролировать скорость поступления данных. Некоторые классы документов могут встречаться в потоке только время от времени. Обнаружить этот редкий класс бывает непросто, и классификация текстов в таких случаях становится чрезвычайно сложной задачей.

Сравнение методов построения классификаторов является довольно сложной задачей по причине того, что разные входные данные могут приводить к различным результатам. Поэтому необходимо осуществить их программную реализацию и вычисление эффективности на одинаковых наборах документов для обучения и тестирования.

Формальная постановка задачи классификации текстов

Следует отличать классификацию от кластеризации. При классификации документов категории определены заранее, при кластеризации они не заданы и даже информация об их количестве может отсутствовать.

Формально постановку задачи классификации можно записать следующим образом.

Имеются множество документов $D = \{d_1, \ldots, d_{|D|}\}$ и множество возможных категорий (классов) $C = \{c_1, \ldots, c_{|C|}\}$. Неизвестная целевая функция Ф: $D \times C \to \{0, 1\}$ задается формулой

$$\Phi\left(d_j, c_i\right) = \begin{cases} 0, \text{ если } d_j \notin c_i, \\ 1, \text{ если } d_j \in c_i. \end{cases} \quad (1)$$

Требуется построить классификатор Ф′, максимально близкий к Ф.

В такой постановке задачи следует отметить, что о категориях и документах нет никакой дополнительной информации, кроме той, которую можно извлечь из самого документа.

Если классификатор выдает точный ответ:

$$\Phi': D \times C \to \{0, 1\}, \quad (2)$$

то классификация называется точной.

Если классификатор определяет степень подобия (Categorization Status Value) документа:

$$CSV: D \to [0, 1], \quad (3)$$

то классификация называется пороговой.

В общем случае процесс обучения с учителем (обучение по прецедентам, supervised learning) заключается в следующем. Системе предъявляется набор примеров, связанных с какой-либо заранее неизвестной закономерностью. Этот набор иногда называют обучающей выборкой L. Ее используют для обучения классификатора и определения значения его параметров, при которых классификатор выдает лучший результат. Далее в системе вырабатываются решающие правила, с помощью которых происходит разделение множества примеров на заданные классы. Качество разделения проверяется тестовой выборкой примеров T. При этом необходимо, чтобы выполнялись условия

$$L \cap T = \varnothing, \quad (4)$$

$$\Omega = L \cup T \subset C \times D. \quad (5)$$

Для множества примеров Ω известны значения целевой функции Ф.

Если в задаче каждому документу $d \in D$ может соответствовать только одна категория $c \in C$, то имеет место однозначная классификация, а если произвольное количество категорий, то многозначная классификация.

Частным случаем однозначной классификации является бинарная классификация, когда коллекцию документов нужно разбить на две непересекающиеся категории. Например, задача определения тональности высказываний (положительная или отрицательная окраска) или задача обнаружения спама (является сообщение спамом или нет) решается при помощи бинарного классификатора.

Решение задачи классификации состоит из четырех последовательных этапов:

– предобработка и индексация документов;
– уменьшение размерности пространства признаков;
– построение и обучение классификатора с помощью методов машинного обучения;
– оценка качества классификации.

При выборе конкретного алгоритма классификации следует учитывать особенности каждого из них. По-прежнему остается нерешенным вопрос определения набора классифицирующих признаков, их количества и способов вычисления весов. В алгоритмах глубокого обучения точность классификации сильно зависит от наличия обучающей выборки подходящего размера. Подготовка такой выборки – очень трудоемкий процесс. До сих пор остается также открытой проблема подбора параметров некоторых алгоритмов на этапе обучения.

Далее подробно рассмотрен каждый из этапов, описаны различные алгоритмы построения классификаторов, проводимые с ними эксперименты и результаты этих экспериментов.

Описание методов классификации

На рисунке 1 представлена общая схема процесса классификации. Рассмотрим каждый из его этапов.

Предобработка и индексация документов. Предварительная обработка текста включает в себя токенизацию, удаление функциональных слов (семантически нейтральных слов, таких как союзы, предлоги, артикли и пр.). Далее осуществляется морфологический анализ (производятся разметка по частям речи и стемматизация). Это позволяет значительно сократить размерность пространства. В результате в качестве признаков документа выступают все значимые слова, встречающиеся в документе.

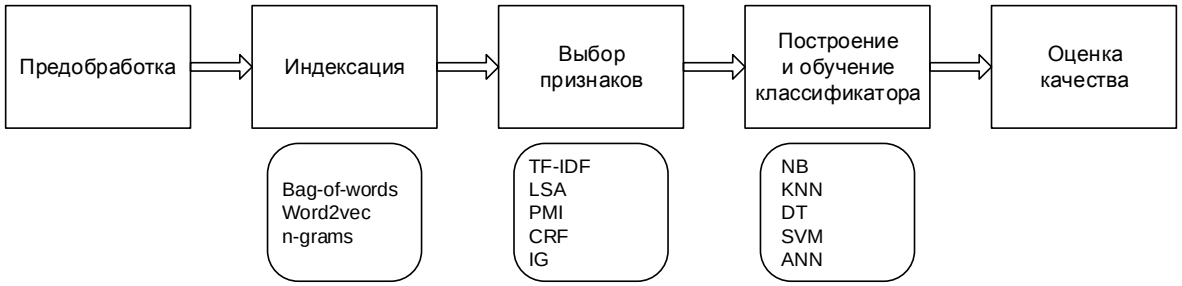

Рис. 1. Этапы процесса автоматической классификации текстов

Fig. 1. The steps of text automatic classification process

Индексация документов – это построение некоторой числовой модели текста, которая переводит текст в удобное для дальнейшей обработки представление.

Например, модель «мешка слов» (bag-of-words) позволяет представить документ в виде многомерного вектора слов и их весов в документе [2]. Другими словами, каждый документ – это вектор в многомерном пространстве, координаты которого соответствуют номерам слов, а значения координат – значениям весов.

Другая распространенная модель индексации – Word2vec [3]. Она представляет каждое слово в виде вектора, который содержит информацию о контекстных (сопутствующих) словах.

Еще одна модель индексации основана на учете *n*-грамм [2], то есть последовательностей из соседних символов.

Очевидно, что для обучающих и тестовых документов должен применяться один и тот же метод индексации.

Уменьшение размерности пространства признаков. Вычислительная сложность различных методов классификации напрямую зависит от размерности пространства признаков. Поэтому для эффективной работы классификатора часто прибегают к сокращению числа используемых признаков (терминов).

За счет уменьшения размерности пространства терминов можно снизить эффект переобучения – явление, при котором классификатор ориентируется на случайные или ошибочные характеристики обучающих данных, а не на важные и значимые. Переобученный классификатор хорошо работает на тех экземплярах, на которых он обучался, и значительно хуже на тестовых данных. Чтобы избежать переобучения, количество обучающих примеров должно быть соразмерно числу используемых терминов. В некоторых случаях сокращение размерности пространства признаков в 10 раз (и даже в 100) может приводить лишь к незначительному ухудшению работы классификатора.

Существуют несколько способов определения веса признаков документа. Наиболее распространенный – вычисление функции TF-IDF [2, 4, 5]. Его основная идея состоит в том, чтобы больший вес получали слова с высокой частотой в пределах конкретного документа и с низкой частотой употреблений в других документах.

Вычисляется частота термина TF (term frequency) – оценка важности слова в пределах одного документа d по формуле

$$TF = n_{t,d} / n_d, \quad (6)$$

где $n_{t,d}$ – количество употреблений слова t в документе d; n_d – общее число слов в документе d.

Обратная частота документа IDF (inverse document frequency) – инверсия частоты, с которой слово встречается в документах коллекции. IDF уменьшает вес общеупотребительных слов по формуле

$$IDF = \log(|D| / D_t), \quad (7)$$

где $|D|$ – общее количество документов в коллекции; D_t – количество всех документов, в которых встречается слово t.

Итоговый вес термина в документе относительно всей коллекции документов вычисляется по формуле

$$V_{t,d} = TF \cdot IDF. \quad (8)$$

Следует отметить, что по формуле (8) оценивается значимость термина только с точки зрения частоты вхождения в документ, без учета порядка следования терминов в документе и их лексической сочетаемости.

Для уменьшения размерности пространства терминов также применяют латентно-семантический анализ (LSA), использующий сингулярное разложение матриц [3, 6], поточечную взаимную информацию (PMI) [6, 7] (разновидность ассоциативной меры), условные случайные поля (CRF) [8] (обобщение скрытой марковской модели). Встречаются исследования [4, 9], в которых применяются статистические критерии и относительная энтропия для вероятностных распределений, называемая коэффициентом усиления информации, или дивергенцией Кульбака–Лейблера.

Построение и обучение классификатора с помощью методов машинного обучения. Можно выделить следующие методы классификации:

- вероятностные (например NB [4, 6]);
- метрические (например KNN [9]);
- логические (например DT [6, 10]);

– линейные (например SVM [4, 5, 6, 9]; логистическая регрессия [2, 8, 10]);

– методы на основе искусственных нейронных сетей (например FFBP [4, 10], RNN [8], DAN2 [9], CNN [2]).

Далее обобщенно описываются эти методы, указываются преимущества и недостатки каждого из них.

Метод Байеса (Naive Bayes, NB) относится к вероятностным методам классификации.

Пусть $P(c_i|d)$ – вероятность того, что документ, представленный вектором $d = (t_1, \ldots, t_n)$, соответствует категории c_i для $i = 1, \ldots, |C|$. Задача классификатора заключается в том, чтобы подобрать такие значения c_i и d, при которых значение вероятности $P(c_i|d)$ будет максимальным:

$$CSV(d) = \arg\max_{c_i \in C} P(c_i \mid d). \qquad (9)$$

Для вычисления значений $P(c_i|d)$ пользуются теоремой Байеса:

$$P(c_i \mid d) = \frac{P(c_i)P(d \mid c_i)}{P(d)}, \qquad (10)$$

где $P(c_i)$ – априорная вероятность того, что документ отнесен к категории c_i; $P(d \mid c_i)$ – вероятность найти документ, представленный вектором $d = (t_1, \ldots, t_n)$, в категории c_i; $P(d)$ – вероятность того, что произвольно взятый документ можно представить в виде вектора признаков $d = (t_1, \ldots, t_n)$.

По сути $P(c_i)$ является отношением количества документов из обучающей выборки L, отнесенных в категорию c_i, к количеству всех документов из L.

$P(d)$ не зависит от категории c_i, а значения $t_1, \ldots, t_n$ заданы заранее, поэтому знаменатель – это константа, не влияющая на выбор наибольшего из значений $P(c_i|d)$.

Вычисление $P(d \mid c_i)$ затруднительно из-за большого количества признаков $t_1, \ldots, t_n$, поэтому делают «наивное» предположение о том, что любые две координаты, рассматриваемые как случайные величины, статистически не зависят друг от друга. Тогда можно воспользоваться формулой

$$P(d \mid c_i) = \prod_{k=1}^{n} P(t_k \mid c_i). \qquad (11)$$

Далее все вероятности подсчитываются по методу максимального правдоподобия.

Преимущества метода:

– высокая скорость работы;

– поддержка инкрементного обучения;

– относительно простая программная реализация алгоритма;

– легкая интерпретируемость результатов работы алгоритма.

Недостатки метода: относительно низкое качество классификации и неспособность учитывать зависимость результата классификации от сочетания признаков.

Метод k ближайших соседей (k Nearest Neighbors, KNN) относится к метрическим методам классификации. Чтобы найти категорию, соответствующую документу d, классификатор сравнивает d со всеми документами из обучающей выборки L, то есть для каждого $d_z \in L$ вычисляется расстояние $\rho(d_z, d)$. Далее из обучающей выборки выбираются k документов, ближайших к d. Согласно методу k ближайших соседей, документ d считается принадлежащим тому классу, который является наиболее распространенным среди соседей данного документа, то есть для каждого класса c_i вычисляется функция ранжирования:

$$CSV(d) = \sum_{d_z \in L_k(d)} \rho(d_z, d) \cdot \Phi(d_z, c_i), \qquad (12)$$

где $L_k(d)$ – ближайшие k документов из L к d; $\Phi(d_z, c_i)$ – известные величины, уже расклассифицированные по категориям документы обучающей выборки.

Преимущества метода:

– возможность обновления обучающей выборки без переобучения классификатора;

– устойчивость алгоритма к аномальным выбросам в исходных данных;

– относительно простая программная реализация алгоритма;

– легкая интерпретируемость результатов работы алгоритма;

– хорошее обучение в случае с линейно неразделимыми выборками.

Недостатки метода:

– репрезентативность набора данных, используемого для алгоритма;

– высокая зависимость результатов классификации от выбранной метрики;

– большая длительность работы из-за необходимости полного перебора обучающей выборки;

– невозможность решения задач большой размерности по количеству классов и документов.

Метод деревьев решений (Decision Trees, DT) относится к логическим методам классификации.

Деревом решений называют ациклический граф, по которому производится классификация объектов (в нашем случае текстовых документов), описанных набором признаков. Каждый узел дерева содержит условие ветвления по одному из признаков. У каждого узла столько ветвлений, сколько значений имеет выбранный признак. В процессе классификации осуществляются последовательные переходы от одного узла к другому в соответствии со значениями признаков объекта. Классификация считается завершенной, когда достигнут один из листьев (конечных узлов) дерева. Значение этого листа определит класс, которому принадлежит рассматриваемый объект. На практике обычно используют бинарные деревья решений, в которых принятие решения перехода по ребрам осуществляется простой проверкой наличия признака в документе. Если значение признака

меньше определенного значения, выбирается одна ветвь, если больше или равно, другая.

В отличие от остальных подходов, представленных ранее, подход, использующий деревья решений, относится к символьным (то есть нечисловым) алгоритмам.

Алгоритм построения бинарного дерева решений состоит из следующих шагов.

Создается первый узел дерева, в который входят все документы, представленные всеми имеющимися признаками. Размер вектора признаков для каждого документа равен n, так как $d = (t_1, \ldots, t_n)$.

Для текущего узла дерева выбираются наиболее подходящий признак t_k и его наилучшее пограничное значение v_k.

На основе пограничного значения выбранного признака производится разделение обучающей выборки на две части. Далее выбранный признак не включается в описание фрагментов в этих частях, то есть фрагменты в частях представляются вектором с размерностью $n - 1$.

Образовавшиеся подмножества обрабатываются аналогично до тех пор, пока в каждом из них не останутся документы только одного класса или признаки для различения документов.

Когда говорят о выборе наиболее подходящего признака, как правило, подразумевают частотный признак, то есть любой признак текста, допускающий возможность нахождения частоты его появления в тексте. Лучшим для разделения является признак, дающий максимальную на данном шаге информацию о категориях. Таким признаком для текста может являться, например, ключевое слово. С этой точки зрения любой частотный признак можно считать переменной. Тогда выбор между двумя наиболее подходящими признаками сводится к оценке степени связанности двух переменных. Поэтому для выбора подходящего признака на практике применяют различные критерии проверки гипотез, то есть критерии количественной оценки степени связанности двух переменных, поставленных во взаимное соответствие, где 0 соответствует полной независимости переменных, а 1 – их максимальной зависимости.

Для исследования связи между двумя переменными удобно использовать представление совместного распределения этих переменных в виде таблицы сопряженности (факторной таблицы, или матрицы частот появления признаков). Она является наиболее универсальным средством изучения статистических связей, так как в ней могут быть представлены переменные с любым уровнем измерения. Таблицы сопряженности часто используются для проверки гипотезы о наличии связи между двумя признаками при помощи различных статистических критериев: критерия Фишера (точного теста Фишера), критерия согласия Пирсона (критерия хи-квадрат), критерия Крамера, критерия Стьюдента (t-критерия Стьюдента) и пр.

Преимущества метода:

– относительно простая программная реализация алгоритма;

– легкая интерпретируемость результатов работы алгоритма.

Недостатки метода: неустойчивость алгоритма по отношению к выбросам в исходных данных и большой объем данных для получения точных результатов.

Метод опорных векторов (Support Vector Machine, SVM) является линейным методом классификации. В настоящее время этот метод считается одним из лучших. Рассмотрим множество документов, которые необходимо расклассифицировать. Сопоставим ему множество точек в пространстве размерности $|D|$.

Выборку точек называют линейно разделимой, если принадлежащие разным классам точки можно разделить с помощью гиперплоскости (в двухмерном случае гиперплоскостью является прямая линия). Очевидный способ решения задачи в таком случае – провести прямую так, чтобы по одну сторону от нее лежали все точки одного класса, а по другую – все точки другого класса. Тогда для классификации неизвестных точек достаточно будет посмотреть, с какой стороны прямой они окажутся.

В общем случае можно провести бесконечное множество гиперплоскостей (прямых), удовлетворяющих нашему условию. Ясно, что лучше всего выбрать прямую, максимально удаленную от имеющихся точек. В методе опорных векторов расстоянием между прямой и множеством точек считается расстояние между прямой и ближайшей к ней точкой из множества. Именно такое расстояние и максимизируется в данном методе. Гиперплоскость, максимизирующая расстояние до двух параллельных гиперплоскостей, называется разделяющей (на рисунке 2 обозначена буквой L). Ближайшие к параллельным гиперплоскостям точки называются опорными векторами (рис. 2), через них проходят пунктирные линии. Другими словами, алгоритм работает в предположении, что, чем больше разница или расстояние между этими параллельными гиперплоскостями, тем меньше будет средняя ошибка классификатора, так как максимизация зазора между классами способствует более уверенной классификации.

На практике структура данных зачастую бывает неизвестна и очень редко удается построить разделяющую гиперплоскость, а значит, невозможно гарантировать линейную разделимость выборки. Могут существовать такие документы, которые алгоритм отнесет к одному классу, а в действительности они должны относиться к противоположному. Такие данные называются выбросами, они создают погрешность метода, поэтому было бы лучше их игнорировать. В этом заключается суть проблемы линейной неразделимости.

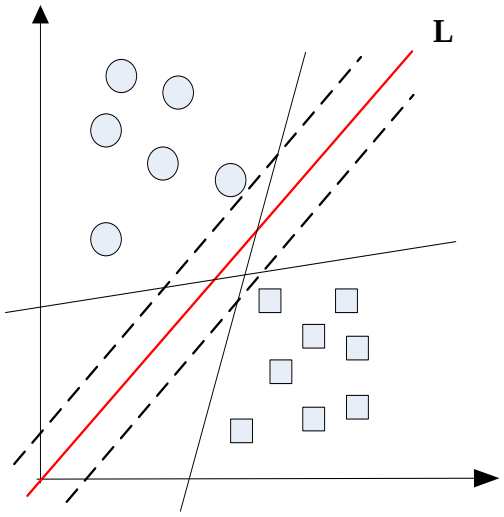

Рис. 2. Разделяющая гиперплоскость в методе опорных векторов

Fig. 2. A separating hyperplane in the Support Vector Machines

Выборку называют линейно неразделимой, если точки, принадлежащие разным классам, нельзя разделить с помощью гиперплоскости. Когда такой разделяющей гиперплоскости не существует, необходимо перейти от исходного пространства признаков документов к новому, в котором обучающая выборка окажется линейно разделимой. Для этого каждое скалярное произведение необходимо заменить на некоторую функцию, отвечающую определенным требованиям. Например, можно назначать некий штраф за каждый неверно расклассифицированный документ. Эту функцию называют ядром. Замена скалярного произведения функцией-ядром позволяет перейти к другому пространству признаков, где данные уже будут разделимы.

В случае линейной неразделимости проблема поиска оптимальной разделяющей гиперплоскости сводится к задаче, эквивалентной поиску седловой точки функции Лагранжа с условиями дополняющей нежесткости. Полученная система уравнений решается методами квадратичного программирования. Это уже чисто вычислительная задача.

Этот вариант алгоритма называют алгоритмом с мягким зазором (soft-margin SVM), тогда как в линейно разделимом случае говорят о жестком зазоре (hard-margin SVM).

Преимущества метода:

– один из наиболее качественных методов;

– возможность работы с небольшим набором данных для обучения;

– сводимость к задаче выпуклой оптимизации, имеющей единственное решение.

Недостатки метода: сложная интерпретируемость параметров алгоритма и неустойчивость по отношению к выбросам в исходных данных.

Логистическая регрессия (logit model, logistic regression) является линейным методом классификации. Этот метод используется для предсказания вероятности возникновения некоторого события по значениям множества признаков. Для этого вводятся так называемая зависимая переменная y, которая может принимать лишь одно из двух значений – как правило, это числа 0 (событие не произошло) и 1 (событие произошло), и множество независимых переменных (также называемых признаками, предикторами или регрессорами) – вещественных $x_1, \ldots, x_n$, на основе значений которых требуется вычислить вероятность принятия того или иного значения зависимой переменной. В случае классификации документов роль зависимой переменной выполняет категория c_i, а роль независимых переменных – набор документов $d_1, \ldots, d_n$.

Для улучшения обобщающей способности алгоритма, то есть для уменьшения эффекта переобучения, на практике часто рассматривается логистическая регрессия с регуляризацией. Регуляризация заключается в том, что вектор параметров θ рассматривается как случайный вектор с некоторой заданной априорной плотностью распределения $p(\theta)$. Для обучения модели вместо метода наибольшего правдоподобия при этом используется метод максимизации апостериорной оценки, то есть должны быть найдены параметры θ, максимизирующие величину: $\prod_{i=1}^{n} P\{c_i \mid d_i, \theta\} \cdot p(\theta)$. (13)

Мультиномиальная логистическая регрессия – это общий случай модели логистической регрессии, в которой зависимая переменная имеет более двух категорий. В модели мультиномиальной логистической регрессии для каждой категории зависимой переменной строится уравнение бинарной логистической регрессии. При этом одна из категорий зависимой переменной становится опорной, а все другие категории сравниваются с ней. Уравнение мультиномиальной логистической регрессии прогнозирует вероятность принадлежности к каждой категории зависимой переменной по значениям независимых переменных.

Вообще говоря, логистическую регрессию можно представить в виде однослойной нейронной сети с сигмоидальной функцией активации, веса которой – коэффициенты логистической регрессии, а вес поляризации – константа регрессионного уравнения:

$P\{y = 1 \mid x\} = f(z)$. (14)

Преимущества метода:

– является одним из наиболее качественных;

– поддерживает инкрементное обучение;

– имеет относительно простую программную реализацию алгоритма.

Недостатки метода: сложная интерпретируемость параметров алгоритма и неустойчивость по отношению к выбросам в исходных данных.

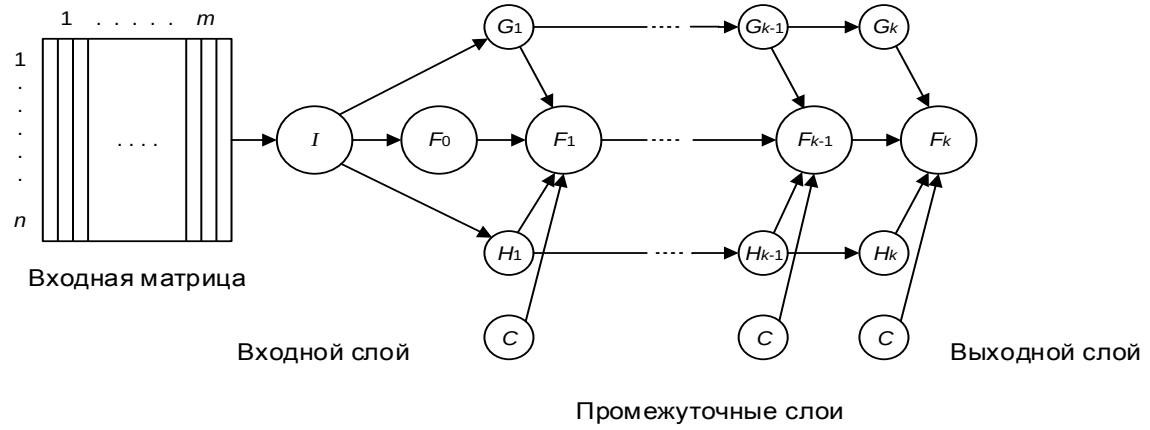

Рис. 3. Обобщенная схема нейронной сети с динамической архитектурой

Fig. 3. A general diagram of a dynamic architecture neural network

Методы на основе искусственных нейронных сетей. Существует большое количество разновидностей нейронных сетей, основные из них – сети прямого распространения, рекуррентные сети, радиально-базисные функции и самоорганизующиеся карты. Настройка весов может быть фиксированной или динамической.

В классических нейронных сетях прямого распространения (Feed Forward Back Propagation, FFBP) присутствуют входной слой, выходной слой и промежуточные слои: сигнал идет последовательно от входного слоя нейронов по промежуточным слоям к выходному. Примером такой структуры является многослойный перцептрон.

Для классификации документа d_i при помощи нейронной сети прямого распространения веса признаков документа подаются на соответствующие входы сети. Активация распространяется по сети; значения, получившиеся на выходах, и есть результат классификации. Стандартный метод обучения такой сети – метод обратного распространения ошибки. Суть его в следующем: если на одном из выходов для одного из обучающих документов получен неправильный ответ, то ошибка распространяется обратно по сети и веса ребер меняются так, чтобы уменьшить ошибку.

Количество промежуточных слоев нейронной сети может быть не задано заранее, такую архитектуру называют динамической. В этом случае слои последовательно динамически генерируются до тех пор, пока не будет достигнут нужный уровень точности.

Обобщенная схема DAN2 приведена на рисунке 3, взятом из статьи [9]. Каждый элемент F_k представляет собой функцию, которая содержит текущий элемент накопленных знаний (Current Accumulated Knowledge Element), полученный на предыдущем шаге обучения сети. C обозначают константы. Вершины G_k и H_k представляют собой текущие остаточные нелинейные компоненты процесса по передаточной функции взвешенной и нормализованной суммы входных переменных (Current Residual Nonlinear Element).

Сверточная нейронная сеть – однонаправленная многослойная сеть с применением операции свертки, при которой каждый фрагмент входных данных умножается на матрицу (ядро) свертки поэлементно, а результат суммируется и записывается в аналогичную позицию выходных данных.

Обобщенная схема CNN приведена на рисунке 4, взятом из статьи [2].

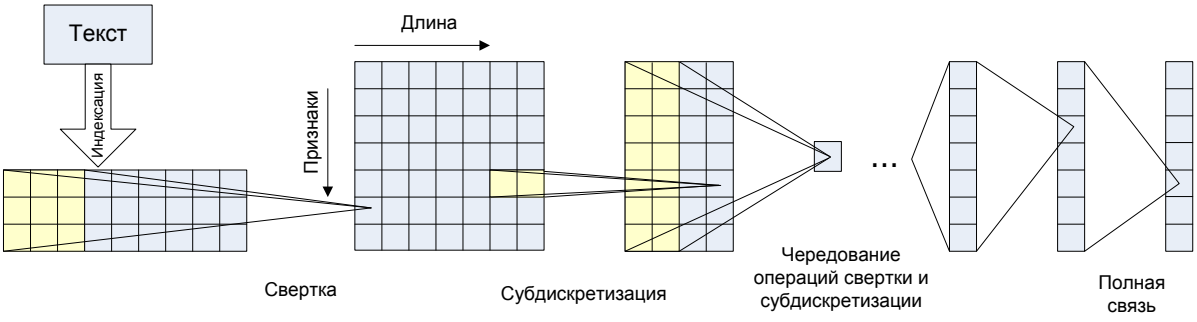

Рис. 4. Обобщенная схема сверточной нейронной сети

Fig. 4. A general diagram of a convolution neural network

Рекуррентная нейронная сеть получается из многослойного перцептрона введением обратных связей. Одна из широко распространенных разновидностей рекуррентных нейронных сетей – сеть Элмана – изображена на рисунке 5 [8]. В ней обратные связи идут не от выхода сети, а от выходов внутренних нейронов. Это позволяет учесть предысторию наблюдаемых процессов и накопить информацию для выработки правильной стратегии обучения. Главной особенностью рекуррентных нейронных сетей является запоминание последовательностей.

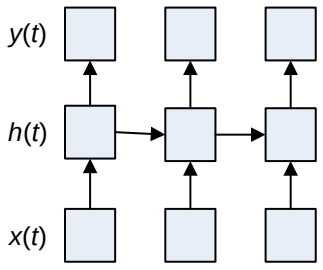

Рис. 5. Нейронная сеть Элмана (разновидность рекуррентной сети)

Fig. 5. Elman neural network (variety of a recurrent network)

Скрытый слой $h(t)$ в период времени t вычисляется путем преобразования текущего входного слоя $x(t)$ и предыдущего скрытого слоя $h(t-1)$. Далее из скрытого слоя $h(t)$ результат поступает на выходной слой $y(t)$.

Преимущества метода:

– имеет очень высокое качество алгоритма при удачном подборе параметров;

– является универсальным аппроксиматором непрерывных функций;

– поддерживает инкрементное обучение.

Недостатки метода:

– вероятность возможной расходимости или медленной сходимости, поскольку для настройки сети используются градиентные методы;

– необходимость очень большого объема данных для обучения, чтобы достичь высокой точности;

– низкая скорость обучения;

– сложная интерпретируемость параметров алгоритма.

Оценка качества классификации

Для обучения и оценки качества классификации, как уже отмечалось ранее, требуются обучающая и тестовая выборки: $\Omega = L \cup T$. Прежде всего нужно выбрать обучающую и тестовую выборки, далее по обучающей выборке найти оптимальные признаки, а потом проверять качество на тестовой. Если сначала найти оптимальные признаки по всей выборке, а потом оценивать качество алгоритма, то отобранные признаки уже оптимизируют качество, поэтому оценка будет слишком оптимистичной. Чтобы оценка качества классификатора была объективной, необходимо правильно выбрать соотношение объемов этих выборок. Если взять очень маленькую обучающую выборку, оценка качества будет слишком пессимистичной. Если тестовая выборка будет маленькая, оценка окажется неточной. Как правило, обучающую и тестовую выборки берут исходя из соотношения 70/30.

Однако есть более объективный способ оценки качества классификатора – кросс-валидация. Суть ее состоит в следующем: все множество Ω разбивается на k частей, каждая из них по очереди выступает как тестовая. Здесь важно сделать оптимальный выбор k. Обычно предпочитают брать $k = 5$ или $k = 10$. Главный недостаток такого способа оценки – большие трудозатраты.

Основным критерием при оценке качества классификации является комбинация точности и полноты.

Точность (precision) классификации в пределах класса – это доля найденных классификатором документов, действительно принадлежащих данному классу, относительно всех документов, которые система отнесла к этому классу.

Полнота (recall) классификации – это доля найденных классификатором документов, действительно принадлежащих классу, относительно всех документов этого класса в тестовой выборке.

Оценка качества работы классификатора производится на тестовой выборке. Вместе с тем работу системы оценивает эксперт (см. табл. 1).

Таблица 1

Оценка качества работы классификатора

Table 1

Classification qualitaty assessment

Класс c_i		**Экспертная оценка**	
		Положительная	**Отрицательная**
Оценка системы	Положительная	*TP*	*FP*
	Отрицательная	*FN*	*TN*

В таблице приняты следующие условные обозначения: *TP* – истинно положительное решение; *TN* – истинно отрицательное решение; *FP* – ложно положительное решение; *FN* – ложно отрицательное решение.

Согласно определению, точность вычисляется следующим образом:

$$p = TP / (TP + FP). \quad (15)$$

Полнота вычисляется по формуле

$$r = TP / (TP + FN). \quad (16)$$

F-мера – характеристика качества работы алгоритма, которая объединяет в себе информацию о точности и полноте:

$$F_\beta = \frac{(\beta^2 + 1) pr}{\beta^2 \cdot p + r}, \quad (17)$$

где $0 \le \beta < \infty$.

При $0 \le \beta < 1$ большее значение имеет точность.

При $\beta = 1$ точность и полнота равноправны, тогда $F_\beta = 2pr / (p + r)$.

При $1 < \beta < \infty$ большее значение имеет полнота.

Часто можно встретить другую формулу для вычисления точности (*accuracy*). Эту величину иногда называют правильностью или аккуратностью метода:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \quad (18)$$

В некоторых случаях удобнее от долей перейти к процентам, умножив полученную величину на 100.

Иногда для сравнения алгоритмов классификации используют специфические характеристики, такие как точка безубыточности, или сбалансированная точность.

Точка безубыточности (break even point, BEP) – величина, заимствованная из экономики, отражающая объем производства и реализации продукции, при котором расходы будут компенсированы доходами, а при производстве и реализации каждой последующей единицы продукции предприятие начинает получать прибыль. В контексте рассматриваемой задачи точка безубыточности используется как мера качества классификации. Точка безубыточности наравне с F-мерой является сбалансированной характеристикой точности и полноты. Более подробное пояснение можно найти в [11].

Под быстродействием классификатора понимается время, затрачиваемое на отнесение документа к одному из классов. Применительно к задачам классификации текстов быстродействие измеряется как процессорное время (в секундах) или как количество вычислительных операций, необходимое для классификации. Измерение производят на обучающей выборке для оценки скорости процесса обучения и отдельно на тестовой выборке. Следует отметить, что высокие затраты при обучении в дальнейшем оправдываются за счет многократного использования настроенного классификатора.

Ясно, что увеличение точности классификации обычно приводит к снижению быстродействия из-за усложнения решающего правила, используемого в алгоритме классификации, а увеличение быстродействия сопровождается понижением точности из-за упрощения работы классификатора.

Эксперименты по сравнению методов

В работе [9] предложен алгоритм классификации на основе нейронных сетей с динамической архитектурой DAN2. Этот вид сетей был выбран на основании экспериментального сравнения DAN2 с обычными нейронными сетями прямого распространения (FFBP) и рекуррентными нейронными сетями (RNN). Для сравнения качества классификации в [9] были рассмотрены DAN2, KNN и SVM.

Классификация проводилась на широко известной коллекции данных Reuters-21578 (http://www.daviddlewis.com/resources/testcollections/reuters21578/), собранной и размеченной в 2004 году Д. Льюисом. Коллекция содержит 21 578 документов из ленты новостей Reuters. Обучающая выборка состоит из 9 603 документов. Тестовая выборка включает в себя 3 299 документов. В эксперименте разбиение производилось на десять наиболее часто встречающихся категорий, связанных с экономикой (нефть, пшеница, кукуруза, торговля, деньги и пр.). Авторы пришли к выводу, что оптимальное количество признаков для каждого из методов – около 2 200. Сравнение качества алгоритмов осуществлялось при помощи точности, полноты, точки безубыточности и F-меры. Установлено, что в зависимости от категории BEP для DAN2 – это 83,17–99,23 %, для KNN – 74,00–97,30 %, для SVM – 75,00–98,50 %. Точность DAN2 для разных классов варьируется от 97,56 до 100 %, полнота – 68,52–99,66 %, F-мера – 80,63–99,23 %.

Время, потраченное на обучение классификатора с использованием DAN2, варьируется в зависимости от категории – 4,078–410,2969 с, время работы уже обученного классификатора на тестовой выборке для выбранных десяти категорий составляет 0,0081–9,5859 с. Для экспериментов использовался многоядерный сервер со следующей конфигурацией: 2 Intel Quad Core Xeon @ 3,2 GHz, 16 GB of RAM, Adaptec Raid Controller with 4 SAS hard drives in RAID 1/0 configurations. Операционная система SuSE Linux Enterprise Server (SLES, 11) 64-bit. Еще пробовали VMWare Server, OpenMPI.

На основе полученных экспериментальных данных можно прийти к выводу, что DAN2 опережает KNN для всех десяти категорий и опережает SVM для девяти из десяти категорий. Вместе с тем следует отметить, что применение нейронных сетей сильно замедляет работу классификатора на этапе обучения.

В работе [4] утверждается, что формально SVM и нейронная сеть прямого распространения (FFNN) имеют похожую структуру, так как выходная функция может быть представлена в виде линейной комбинации простых функций, то есть

$$f(x) = b + \sum_{k=1}^{M} \lambda_k h(w_k, x). \quad (19)$$

В таком случае количество скрытых нейронов является долей числа опорных векторов (табл. 2). Более подробное описание используемой в таблице общепринятой терминологии можно найти, например, в [12].

Кроме того, SVM используется в задаче выпуклой оптимизации, которая всегда позволяет найти глобальный минимум и единственное решение, в то время как FFNN тренируется при помощи метода градиентного спуска, который не всегда сходится к оптимальному (глобальному) решению. В статье [4] предложены техники для минимизации

случая локальной сходимости, а также показано, что масштабированный метод сопряженных градиентов не сходится реже, чем традиционный метод сопряженных градиентов или метод обратного распространения с использованием градиентного спуска.

Таблица 2

Соотношение элементов SVM и FFNN

Table 2

Correlation of SVM and FFNN

Элемент метода	SVM	FFNN
M	Количество опорных векторов	Количество вершин в скрытом слое
h	Функция-ядро	Функция активации
$\{w_k\}_{k=1}^{M}$	Опорные векторы	Веса скрытого слоя
$\{\lambda_k\}_{k=1}^{M}$	Коэффициенты задачи выпуклой оптимизации	Веса выходного слоя

В подтверждение своих наблюдений авторы приводят описание эксперимента по разделению отзывов о фильмах, книгах, GPS и фотоаппаратах на положительные и отрицательные с использованием методов SVM, NB и ANN. В методе SVM в качестве обычного нелинейного ядра была взята радиальная базисная функция. В методе с искусственными нейронными сетями было отдано предпочтение прямоточной нейронной сети (однонаправленной сети с одним скрытым слоем). Для обучения нейронной сети использовался алгоритм обратного распространения ошибки (Backpropagation). Чтобы ускорить процесс обучения и сократить риск переобучения, применялась технология «ранней остановки».

Результаты классификации сравнивались на сбалансированных и несбалансированных данных. Данные для категории «фильмы» были взяты из популярной, часто цитируемой базы Movie Review Data (http://www.cs.cornell.edu/people/pabo/movie-review-data/), для остальных категорий авторы собирали коллекции самостоятельно с сайта Amazon (http://www.amazon.com/) по 2 000 отзывов для каждого класса.

Характеристиками для сравнения являлись точность (precision), полнота (recall), аккуратность (accuracy) и время в секундах. Сравнение осуществлялось с помощью 10-проходной кросс-валидации. Количество признаков в экспериментах варьировалось от 50 до 5 000. В среднем наилучшие результаты были получены при 500–1 000. Было замечено, что на 5 000 терминов для ANN время на обучение значительно увеличивается (с 3,70 с до 69,40 с), а время работы не меняется; для SVM, наоборот, время на обучение не меняется, но слишком большое количество признаков сильно сказывается на длительности работы.

На сбалансированных данных было проведено 28 тестов для четырех категорий. ANN показал лучший результат, чем SVM, в 13 тестах (t-тест с $p<0,05$); SVM превзошел ANN только в 2, хотя в целом разница в результатах не превысила 3 %.

Худшие результаты получены для класса «книги». Точность 0,88 % при 3 000 терминов для SVM; 0,86 % при 3 000 и 4 000 терминов для SVM. Полнота 0,8 при 1 000 терминов для ANN; 0,88 при 3 000 для SVM. Лучшая аккуратность (accuracy) 81,8 % при 1 000 терминов достигается для ANN.

Для остальных трех категорий результаты лучше. Для класса GPS лучшая точность 0,96 и лучшая полнота 0,99 достигаются методом NB, лучшая аккуратность (accuracy) 87,3 % – методом ANN. Для класса «фильмы» лучшая точность 0,95 достигается NB, на втором месте 0,87 – ANN, лучшая полнота 0,98 – NB, на втором месте 0,87 – ANN, лучшая аккуратность (accuracy) 86,5 % получена методом ANN. Для класса «фотоаппараты» лучшая точность 0,94 получена методом NB, лучшая полнота 0,96 – NB, лучшая аккуратность (accuracy) 90,3 % – ANN.

Время работы зависит от количества векторов для SVM и количества слоев для ANN. Наравне с ними рассматривался Байес. Он, бесспорно, быстрее всех (0,01–0,02 с) при любом количестве терминов для любого класса и в некоторых случаях, как ни странно, показывал лучшую точность. Время на обучение для класса «книги»: SVM – 0,22–1,5 с, ANN – 3,7–69,4 с в зависимости от количества признаков (50–5 000). Время на обучение для класса GPS: SVM – 0,2–1,3 с, ANN – 3,1–75,4 с. Время на обучение для класса «фильмы»: SVM – 0,27–5,6 с, ANN – 2,3–65,5 с. Время на обучение для класса «фотоаппараты»: SVM – 0,2–1,1 с, ANN – 4,5–77,2 с. К сожалению, в статье отсутствуют данные об аппаратном обеспечении, на котором проводилось исследование.

В работе [10] рассматриваются методы обнаружения ложных высказываний в текстах, когда люди намеренно говорят неправду, пытаясь обмануть. Авторы исследовали высказывания людей, вовлеченных в преступления на военных базах. Подозреваемые и свидетели описывали события своими словами. Сотрудники правоохранительных органов находили в архивных данных либо подтверждения, либо опровержения этим высказываниям. Таким образом оценивали истинность высказываний либо их ложность. Проанализировано 371 сообщение из специальных архивов о различных видах преступлений: дорожные нарушения, магазинные кражи, нападения и поджоги. Большой проблемой было собрать такую коллекцию данных, для которой можно установить истинность/ложность высказываний.

Для классификации первоначально экспертами был составлен перечень из 31 признака. Далее при помощи критерия хи-квадрат было выбрано 13 наиболее подходящих признаков: количество глаголов движения, личных местоимений, количество

слов с оттенком намерения, причины, с указанием времени, лексическое разнообразие и пр. Некоторые из отобранных признаков весьма специфичны и требуют составления семантических словарей.

В эксперименте проводилось сравнение многослойного перцептрона (MLP, разновидность FFNN), модификации деревьев решений (CART), логистической регрессии и ансамбля классификаторов. Для построения ансамбля классификаторов, как правило, используются два основных метода: бустинг (boosting) и бэггинг (bagging). При бустинге происходит последовательное обучение классификаторов. Например, первый классификатор обучается на всем наборе данных, второй – на выборке примеров, а третий – на наборе тех данных, в которых результаты первых двух классификаторов разошлись. Бэггинг использует параллельное обучение базовых классификаторов, то есть бэггинг является улучшающим объединением, а бустинг – улучшающим пересечением.

Для проверки использовалась 10-проходная кросс-валидация. Для метода MLP достигнута точность 73,46 %, для CART – 71,60 %, для логистической регрессии – 67,28 %. В результате был сделан вывод, что наиболее высокая точность, 74,07 %, достигается на ансамбле классификаторов. Преимуществом ансамблевых классификаторов является качество их работы на неравномерно распределенных данных. Эта особенность важна при потоковой обработке данных, когда некоторые редкие классы документов, появляющиеся и исчезающие в потоке, порой непросто обнаружить.

В работе [8] рассмотрено решение сразу двух задач: извлечение терминов и классификация для англоязычного и русскоязычного корпусов. Необходимо было определить эмоциональную окраску отзывов о ресторанах и автомобилях, то есть разделить отзывы на положительные и отрицательные, построив бинарный классификатор для каждой из категорий «рестораны» и «автомобили».

Для классификации использовались рекуррентные нейронные сети, в частности, нейронные сети Элмана (простые и двунаправленные BRNN) и LSTM. Для извлечения аспектных терминов в сравнении участвовали несколько методов: два вида многослойного персептрона (MLP), логистическая регрессия и условные случайные поля (CRF). Для условных случайных полей в качестве признаков использовались основы слов и принадлежность частям речи (проводилась процедура POS-tagging). Использовались две метрики для извлечения аспектных терминов: на основе точного количества и на основе пропорционального перекрытия.

Сравнение результатов классификации для английских текстов проводилось на недавно собранной коллекции SemEval-2014 ABSA Restaurants (http://metashare.ilsp.gr:8080/repository/search/?q=SemEval-2014+ABSA+Restaurants). Обучающая выборка состояла из 3 041 сообщения, тестовая – из 800. Для проверки качества методов на русских текстах была использована коллекция отзывов о ресторанах и автомобилях, собранная для проведения соревнований SentiRuEval-2015 в рамках конференции «Диалог». Для проверки полученных результатов применялась 7-проходная кросс-валидация.

Лучшие результаты как при извлечении терминов, так и при классификации отзывов показал метод LSTM. При извлечении терминов F-мера для LSTM составила 79,80 %. При классификации точность – 69,70 %. Метод LSTM для русскоязычных данных для класса «рестораны» показал точность 61,1 %, F-меру – 70,2 %. Для класса «автомобили» результаты хуже: точность – 58,0 %, F-мера – 62,4 %.

Проблема тематической классификации коротких текстовых сообщений (от нескольких слов до 2 предложений), например, смс-сообщений, комментариев к новостям, на форумах, в социальных сетях, рассматривается в [7]. Основная сложность, возникающая при решении этой проблемы, – определение набора признаков, по которым предстоит классифицировать. В качестве признаков классификации принято рассматривать слова и словосочетания, буквы и буквосочетания. Один из недостатков распространенных на сегодняшний день методов – привязка к конкретному естественному языку и опора на словари. В данной публикации уделяется внимание определению универсальных методов и дифференцирующих признаков для текстов на различных естественных языках.

В данной работе также описан алгоритм построения и обучения классификатора на основе метода взаимной информации (PMI). Применялась процедура POS-tagging, и для классификации были выбраны следующие признаки: *N* – существительные, *NA* – существительные и прилагательные, *NAV* – существительные, прилагательные, глаголы, *NNP* – существительные и именные группы, *VVP* – глаголы и глагольные группы, *Stem* – псевдоосновы словоупотреблений текста, полученные алгоритмами аналитического морфологического анализа (имеются в виду широко известные алгоритмы Портера, Ловинса, Пейса–Хаска и пр.).

Принадлежность текста к категории определяется наличием в нем признаков, релевантных данной категории и коррелирующих с признаками рассматриваемой категории, а также отсутствием нерелевантных признаков и признаков, не коррелирующих с признаками данной категории [7]. При таком подходе тексту можно сопоставить информационную матрицу I, элементы которой определяются как пара: $I_{ij} = \{\rho_{ij}, \varepsilon_{ij}\}$, где ρ_{ij} – коэффициент релевантности; ε_{ij} – коэффициент корреляции.

Для коэффициентов релевантности и корреляции справедливо следующее утверждение: большие значения этих коэффициентов соответствуют признакам, наиболее точно характеризующим

выбранный класс. Пороговые значения релевантности и корреляции служат параметрами, определяющими точность. В процессе классификации вычисляются коэффициенты релевантности и корреляции для каждого текста как суммы соответствующих коэффициентов для данного класса по всем вхождениям признаков. Документ считается отнесенным к тем категориям, для которых произошло превышение пороговых значений по обеим характеристикам: как по коэффициенту корреляции, так и по коэффициенту релевантности. Пороговые значения для каждой категории могут быть заданы пользователем или же рассчитаны автоматически по обучающей выборке.

В работе [7] представлены результаты экспериментов для метода NB и метода на основе PMI. Обучение классификаторов проводилось на созданных экспертами выборках текстов с сайтов из Интернета: для русского языка объемом 57,3 Мб, башкирского – 1,87 Мб, татарского – 2,68 Мб. В качестве классов условно были выбраны «наркотики», «насилие», «национализм», «отрицание традиционных ценностей», «порнография», «терроризм», «фашизм», «экстремизм».

Учет выбранных морфологических признаков оказывает различное влияние на качество классификации в зависимости от класса. Для некоторых классов (например «фашизм») могут оказывать положительное влияние существительные и именные группы, а на определение некоторых тематик (например «наркотики», «фашизм») отрицательное влияние оказывает учет глагольных групп.

Лучшие значения F-меры достигаются предложенным методом (на основе PMI, в качестве признаков – псевдоосновы) для классов: «жестокость» – 0,913, «отрицание традиционных ценностей» – 0,862, «наркотики» – 0,765. В итоге можно сделать вывод, что псевдоосновы, выделенные аналитическим алгоритмом морфологического анализа, могут считаться универсальными дифференцирующими признаками при классификации коротких текстовых сообщений.

В работе [2] предложен алгоритм классификации с применением сверточных нейронных сетей (CNN), проведено сравнение этого метода с методом на основе рекуррентных нейронных сетей (LSTM) и мультиномиальной логистической регрессией (logit model) в разных вариациях («мешок слов», TF-IDF, *n*-граммы, Word2vec). Тестирование сверточных нейронных сетей проводилось на данных, основанных на словах (word-based) и на символах (character-based).

Особенность метода CNN заключается в необходимости использования очень больших коллекций для обучения. Большинство открытых коллекций для классификации текстов (даже на английском языке) слишком малы, поэтому для эксперимента авторы самостоятельно собрали тексты с новостных сайтов, обзоры и отзывы пользователей, данные из DBPedia, которая содержит структурированные данные из википедии. В результате были получены коллекции данных на английском и китайском языках. Объемы собранных коллекций приведены в таблице 3: обучающие выборки содержали от 120 000 до 3 600 000 текстов, тестовые – от 7 600 до 650 000 текстов.

Таблица 3

Количество классов и объем обучающей и тестовой выборок для реализации метода CNN

Table 3

A number of classes, training and test set volume for CNN method implementation

Название коллекции	Количество классов	Обучающая выборка	Тестовая выборка
AG's News	4	120 000	7 600
Sogou News	5	450 000	60 000
DBPedia	14	560 000	70 000
Yelp Review Polarity	2	560 000	38 000
Yelp Review Full	5	650 000	50 000
Yahoo! Answers	10	1 400 000	60 000
Amazon Review Full	5	3 000 000	650 000
Amazon Review Polarity	2	3 600 000	400 000

Наименьшие погрешности измерения 1,31–7,64 % были достигнуты для моделей, использующих в качестве признаков *n*-граммы, а также сверточные нейронные сети (погрешность измерения 4,93–40,43 %) в зависимости от коллекции.

Самые плохие результаты показала модель с применением метода Word2vec. Это означает, что получивший широкое распространение метод представления слов Word2vec в виде векторов не дает преимуществ в задаче классификации текстов. Хотя авторам статьи [2] еще предстоит детально интерпретировать полученные результаты, а также продолжать эксперименты на коллекциях текстов для других языков, сейчас можно сделать вывод, что для задачи классификации текстов лучшим оказался символьный уровень, когда рассматриваются буквосочетания (без привязки к конкретному языку).

Результаты исследования

Классификация текстов является одной из основных задач компьютерной лингвистики, поскольку к ней сводится ряд других задач: определение тематической принадлежности текстов, автора текста, эмоциональной окраски высказываний и др.

Формально задачу классификации текстов можно описать следующим образом. Имеется мно-

жество документов и множество возможных категорий (классов). Требуется построить классификатор, относящий выбранный документ к одной из нескольких заранее определенных категорий на основании содержания документа. Наиболее распространенный современный подход к классификации основывается на методах машинного обучения. Согласно этим методам, набор правил или критерий принятия решения текстового классификатора вычисляется автоматически на основе обучающих данных. Обучающими данными являются образцы документов из каждого класса.

Решение задачи классификации состоит из четырех последовательных этапов: предобработка и индексация документов, уменьшение размерности пространства признаков, построение и обучение классификатора с помощью методов машинного обучения, оценка качества классификации.

Для предварительной обработки и индексации документа (то есть при построении некоторой числовой модели текста) обычно применяется одна из трех моделей: модель «мешка слов», Word2vec и модель, основанная на учете *n*-грамм. Для реализации первой и второй моделей необходимы дополнительные знания о морфологической и синтаксической структуре языка. Применение символьных *n*-грамм позволяет не накладывать ограничения на использование конкретного языка, поэтому в ряде случаев является предпочтительным.

Вычислительная сложность различных методов классификации напрямую зависит от размерности пространства признаков. За счет уменьшения размерности пространства терминов можно снизить эффект переобучения – явление, при котором классификатор ориентируется на случайные или ошибочные характеристики обучающих данных, а не на важные и значимые. Переобученный классификатор хорошо работает на тех экземплярах, на которых он обучался, и значительно хуже на тестовых данных. Чтобы избежать переобучения, количество обучающих примеров должно быть соразмерно числу используемых терминов, поэтому для эффективной работы классификатора часто прибегают к сокращению числа используемых признаков (терминов). Для уменьшения размерности пространства терминов применяют такие методы, как LSA, TF-IDF, PMI, CRF, IG. Наибольшее распространение из них получил метод TF-IDF.

Выводы

В статье были рассмотрены следующие наиболее распространенные методы построения и обучения классификатора: NB, KNN, SVM, DT, логистическая регрессия и алгоритмы глубокого обучения, основанные на искусственных нейронных сетях (FFBP, RNN, DAN2, CNN).

Для обучения и оценки качества классификации необходимо подготовить обучающую и тестовую выборки, далее по обучающей выборке найти оптимальные признаки, а затем проверять качество на тестовой выборке. Чтобы оценка качества классификатора была объективной, требуется правильно выбрать соотношение объемов этих выборок. Как правило, обучающую и тестовую выборки берут исходя из соотношения 70/30. Более объективным способом оценки качества классификатора является кросс-валидация.

Общепризнанными характеристиками качества работы классификатора являются точность, полнота и их комбинация (F-мера). На основе проведенного исследования можно сделать вывод, что наилучшее соотношение этих характеристик достигается при использовании методов SVM (точность 80–85 %, полнота 83–87 %) и CNN (точность 90–95 %, полнота 80–85 %). Помимо характеристик качества классификации, целесообразно учитывать также другие факторы: время работы алгоритма, возможность работы алгоритма в инкрементном режиме, количество предварительной информации, необходимой для классификации, независимость от языка. Скорость работы алгоритма NB одна из самых высоких, однако точность для различных экспериментов сильно варьируется (71–90 %). При потоковой обработке текстов классификация документов должна осуществляться одновременно с поступлением их из источника, поэтому предпочтение должно отдаваться инкрементным алгоритмам, таким как CNN или SVM.

В настоящее время по-прежнему остается нерешенным вопрос определения набора классифицирующих признаков, их количества и способов вычисления весов. При выборе определенного метода следует помнить, что при большом количестве признаков (около 5 000) время на обучение для нейронных сетей значительно увеличивается, а время работы не меняется; для SVM, наоборот, время на обучение не меняется, но слишком большое количество признаков сильно сказывается на длительности работы. Оптимальным является 500–1 000 признаков, в некоторых случаях – до 2 200 признаков. В качестве признаков удобно рассматривать частоты символьных *n*-грамм, чтобы не накладывать ограничения на использование конкретного языка.

В алгоритмах глубокого обучения точность классификации существенно зависит от наличия обучающей выборки подходящего размера. Подготовка такой выборки – очень трудоемкий процесс. Подбор параметров некоторых алгоритмов на этапе обучения до сих пор также остается открытой проблемой. Согласно результатам проведенного исследования, для обучения и тестирования классификатора с использованием метода SVM на русском языке нужна размеченная коллекция текстов объемом 1 000–2 000 текстов для достижения точности 80–85 %. Для обучения и тестирования классификатора с применением метода CNN необхо-

димо собрать и подготовить коллекцию текстов на русском языке объемом около 1 000 000 документов для достижения точности 90–95 %.

Следует отметить, что большинство упоминаемых в обзоре экспериментов проводилось на коллекциях англоязычных текстов. Довольно часто встречаются статьи с описанием исследований применительно к китайскому языку. Исследования по сравнению различных методов классификации для русскоязычных текстов проводятся в основном в контексте задачи сентимент-анализа, где рассматриваются два класса: положительный и отрицательный.

Создание общедоступной коллекции необходимого размера позволило бы российским исследователям активнее изучать проблемы автоматической обработки текстовой информации в целом и вместе с тем разрабатывать новые инструменты для решения прикладных задач в данной области.

Работа выполнена при финансовой поддержке Минобрнауки РФ (договор № 02.G25.31.0146) в рамках реализации Постановления Правительства РФ № 218.

Литература

1. Aggarwal C. Data classification: algorithms and applications. CRC Press, 2014, chap. 9, pp. 245–273.

2. Xiang Zhang, Junbo Zhao, Yann LeCun. Character-level convolutional networks for text classification. Proc. Neural Inform. Processing Systems Conf. (NIPS 2015). Montreal, Canada, 2015. URL: https://arxiv.org/abs/1509.01626 (дата обращения: 18.07.2016).

3. Ju R. et al. An Efficient Method for Document Categorization Based on Word2vec and Latent Semantic Analysis. 2015 IEEE Intern. Conf. on Comp. and Inform. Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing. Liverpool, UK, 2015, pp. 2276–2283.

4. Moraes R., Valiati J.F., and Gavião Neto W.P. Document-level sentiment classification: An empirical comparison between SVM and ANN. Expert Systems with Applications, 2013, no. 40, pp. 621–633.

5. Pontiki M Galanis D., Pavlopoulos J., Papageorgiou H., Androutsopoulos I., Manandhar S. SemEval-2014 Task 4: Aspect based sentiment analysis. The 8th Intern. Workshop on Semantic Evaluation (SemEval 2014). Dublin, Ireland. 2014, pp. 27–35.

6. Medhat W., Hassan A., Korashy H. Sentiment analysis algorithms and applications: a survey. Ain Shams Eng. Jour., 2014, no. 5, pp. 1093–1113.

7. Поляков И.В., Соколова Т.В., Чеповский А.А., Чеповский А.М. Проблема классификации текстов и дифференцирующие признаки // Вестн. НГУ. Сер.: Информационные технологии. 2015. Т. 13. Вып. 2. С. 55–63.

8. Tarasov D.S. Deep recurrent neural networks for multiple language aspect-based sen-timent analysis. Computational Linguistics and Intellectual Technologies. In Proc. Annual Intern. Conf. «Dialogue-2015». Moscow, 2015, vol. 2, iss. 14 (21), pp. 65–74.

9. Ghiassi M., Olschimke M., Moon B., Arnaudo P. Automated text classification using a dynamic artificial neural network model. Expert Syst. with Applications, 2012, no. 39, pp. 10967–10976.

10. Fuller C.M., Biros D.P. and Delen D. An investigation of data and text mining methods for real world deception detection. Expert Syst. with Applications, 2011, no. 38, pp. 8392–8398.

11. Yang Y. An evaluation of statistical approaches to text categorization. Information Retrieval Jour., 1999, vol. 1, iss. 1, pp. 69–90.

12. Haykin S. Neural networks: A comprehensive foundation (2nd ed.). Pearson Education, Singapore, 2001, 824 p.

Software & Systems
DOI: 10.15827/0236-235X.030.1.085-099

Received 19.07.16
2017, vol. 30, no. 1, pp. 85–99

AUTOMATIC TEXT CLASSIFICATION METHODS

T.V. Batura[1, 2], *Ph.D. (Physics and Mathematics), Leading Researcher, Senior Researcher, tatiana.v.batura@gmail.com*

[1] *Novosibirsk State University, Piroqov St. 2, Novosibirsk, 630090, Russian Federation*
[2] *A.P. Ershov Institute of Informatics Systems (IIS), Siberian Branch of the Russian Federationn Academy of Sciences, Lavrentev Av. 6, Novosibirsk, 630090, Russian Federation*

Abstract. Text classification is one of the main tasks of computer linguistics because it unites a number of other problems: theme identification, authorship identification, sentiment analysis, etc. Content analysis in telecommunication networks is of great importance to ensure information security and public safety. Texts may contain illegal information (including data related to terrorism, drug trafficking, organization of protest movements and mass riots). This article provides a survey of text classification methods. The purpose of this survey is to compare modern methods for solving the text classification problem, detect a trend direction, and select the best algorithm for using in research and commercial problems.

A well-known modern approach to text classification is based on machine learning methods. It should take into account the characteristics of each algorithm for selecting a particular classification method. This article describes the most popular algorithms, experiments carried out with them, and the results of these experiments. The survey was prepared on the basis of scientific publications which are publicly available on the Internet, made in the period of 2011–2016, and highly regarded by the scientific community.

The article contains an analysis and a comparison of different classification methods with the following characteristics: precision, recall, running time, the possibility of the algorithm in incremental mode, amount of preliminary information necessary for classification, language independence.

Keywords: text classification, analysis of text information, data mining, text mining, natural language processing, classification quality, machine learning, deep learning, neural networks.

Acknowledgements. *The work has been financially supported by Ministry of Education and Science of the Russian Federation (contract no. 02.G25.31.0146) as a part of RF Government Regulation no. 218 execution.*

References

1. Aggarwal C. *Data Classification: Algorithms and Applications*. CRC Press, 2014, pp. 245–273.
2. Zhang X., Zhao J., LeCun Y. Character-level Convolutional Networks for Text Classification. *Proc. of the Neural Information Processing Systems Conf. (NIPS 2015)*. Montreal, Canada, 2015. Available at: https://arxiv.org/abs/1509.01626 (accessed July 18, 2016).
3. Ju R. An Efficient Method for Document Categorization Based on Word2vec and Latent Semantic Analysis. *2015 IEEE Int. Conf. on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*. Liverpool, UK, 2015, pp. 2276–2283.
4. Moraes R., Valiati J.F., Gavião Neto W.P. Document-level sentiment classification: An empirical comparison between SVM and ANN. *Expert Systems with Applications*. 2013, no. 40, pp. 621–633.
5. Pontiki M., Galanis D., Pavlopoulos J., Papageorgiou H., Androutsopoulos I., Manandhar S. SemEval-2014 Task 4: Aspect based sentiment analysis. *Proc. 8th Int. Workshop on Semantic Evaluation (SemEval 2014)*. Dublin, Ireland, 2014, pp. 27–35.
6. Medhat W., Hassan A., Korashy H. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journ*. 2014, no. 5, pp. 1093–1113.
7. Polyakov I.V., Sokolova T.V., Chepovsky A.A., Chepovsky A.M. Text classification problem and features set. *Vestn. NGU. Ser.: Informatsionnye tekhnologii* [Novosibirsk State Univ. Journ. of Information Technologies]. 2015, vol. 13, iss. 2, pp. 55–63 (in Russ.).
8. Tarasov D.S. Deep Recurrent Neural Networks for Multiple Language Aspect-Based Sentiment Analysis. *Computational Linguistics and Intellectual Technologies: Proc. of Annual Int. Conf. "Dialogue-2015"*. Moscow, Russia, 2015, vol. 2, iss. 14 (21), pp. 65–74.
9. Ghiassi M., Olschimke M., Moon B., Arnaudo P. Automated text classification using a dynamic artificial neural network model. *Expert Systems with Applications*. 2012, no. 39, pp. 10967–10976.
10. Fuller C.M., Biros D.P. and Delen D. An investigation of data and text mining methods for real world deception detection. *Expert Systems with Applications*. 2011, no. 38, pp. 8392–8398.
11. Yang Y. An evaluation of statistical approaches to text categorization. *Information Retrieval Jour*. 1999, vol. 1, iss. 1, pp. 69–90.
12. Haykin S. *Neural networks: A comprehensive foundation*. 2nd ed., Pearson Education Publ., Singapore, 2001, 824 p.

Примеры библиографического описания статьи

1. Батура Т.В. Методы автоматической классификации текстов // Программные продукты и системы. 2017. Т. 30. № 1. С. 85–99; DOI: 10.15827/0236-235X.030.1.085-099.

2. Batura T.V. Automatic text classification methods. *Programmnye produkty i sistemy* [Software & Systems]. 2017, vol. 30, no. 1, pp. 85–99 (in Russ.); DOI: 10.15827/0236-235X.030.1.085-099.