УДК 681.301

ЗАЙЦЕВ В.Г.,
ЛАН ЧУНЬЛИНЬ

СПОСОБЫ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ КЛАССИФИКАЦИИ ДОКУМЕНТОВ ДЛЯ КОНЕЧНОГО МНОЖЕСТВА ЯЗЫКОВ

В работе рассматриваются две основные проблемы классификации документов для конечного множества языков: анализ текста в обработке естественного языка; нахождение наилучшего алгоритма машинного обучения. Описывается предлагаемый эффективный метод машинного обучения классификации текстов для конечного множества языков.

The paper analysis on two major problems for the classification documents in multilingual environment: analysis of the text in natural language processing; choose the efficient algorithm methods for machine learning. The efficient algorithm methods for machine learning are described.

Введение

Задача разработки информационных систем, таких как интеллектуальные системы документооборота является одной из самых актуальных на сегодняшний день. Она рассматривается в контексте создания хранилищ данных и их систематизации с целью облегчения поиска необходимой информации. Современные информационные системы должны быть способны решать весь комплекс задач, связанных с управлением потоком входящих данных – автоматическую классификацию текстов для конечного множества языков. Действует целый ряд систем классификации больших объемов текстовой информации, в основе которых лежат технологии компьютерной лингвистики и алгоритмов распознавания образов. Обычные системы автоматической классификации текстов решают задачу автоматической классификации текстов на одном языке. Повышение эффективности классификации документов можно достичь изменением методов машинного обучения. Для конечного множества языков, это не только зависит от метода машинного обучения системы, а также зависит от обработки естественного языка (Natural Language Processing, NLP). Обработка естественного языка – общее направление искусственного интеллекта и математической лингвистики. Оно изучает проблемы компьютерного анализа и синтеза естественных языков. Применительно к искусственному интеллекту анализ означает понимание языка, а синтез – генерацию грамотного текста. Решение этих проблем будет означать создание более удобной формы взаимодействия компьютера и человека[1].

Таким образом, повышение эффективности автоматической классификации документов для конечного множества языков связано с нахождением наилучшего алгоритма машинного обучения и наилучшую обработку естественного языка с целью анализа исходных данных.

Анализ текста – одна из самых главных задач обработка естественного языка

Распознавание слов и сегментация фраз на слова являются важными шагами во многих приложениях обработки естественного языка. В западных и европейских языках слова в предложениях разделяются пробелами. Поэтому слова довольно просто определяются как человеком так и компьютером. В восточных языках, таких как китайский, корейский и японский языки, проблемы распознавания слов и сегментации становятся более сложными. Слово может являться одним слогом либо комбинацией слогов, расположенных вместе в предложении. Проблемы распознавания китайских слов и сегментации китайских предложений на слова не могут быть полностью решены из-за следующих причин: не существует алгоритма, который сегментирует китайскиое предложение на слова точно в соответствии с его смыслом, если предложение считается изолированным. Отсутствие алгоритма сегментации на слова, который работал бы в предложении, объясняется тем, что каждое слово может быть частью разных слов. Его смысл не может быть определен без контекста.

В контексте автоматической обработки текстовой информации наиболее очевидной является проблема делимитации слова в китайском языке. Технически сложность определения границ слова заключается в том, что в китайской системе письменности не принято отделять слова пробелами. Например, в упомянутых европейских языках проблема определения слова ограничивается нахождением его физических границ, и, как правило, все системы автоматического анализа письменного текста в качестве базовой единицы принимают именно графическое слово – последовательность знаков алфавита, отделённых друг от друга пробелами или знаками препинания. Проблема делимитации слова в китайском языке осложняется также и тем, что слова в китайском языке почти лишены формальных признаков морфологического уровня. В словообразовании доминирует корнесложение, флективные формы встречаются нерегулярно, словоизменение почти отсутствует, что в совокупности делает невозможным выделение хотя бы основных лексических единиц путём определения их границ по каким-либо морфологическим маркерам. Многие исследователи пытаются решить проблему так называемой сегментации китайского текста (разбиение текста на слова). Для этого используются три основных способа: словарный (обычно с применением алгоритма максимального соответствия), статистический и комбинированный, сочетающий в себе оба предыдущих. По сообщениям авторов им удаётся правильно сегментировать текст на 95-99 процентов, это проблема почти решена[2].

В западных языках и европейских языках существует проблема стемминга. Стемминг (stemming) – это процесс нахождения основы слова для заданного исходного слова. Стеммер Портера (Stemmer Porter) – алгоритм стемминга. Впоследствии Мартин создал проект «Snowball» и, используя основную идею алгоритма, написал стеммеры для распространённых индоевропейских языков, в том числе для русского. Алгоритм не использует баз основ слов, а лишь, применяя последовательно ряд правил, отсекает окончания и суффиксы, основываясь на особенностях языка[3].

Результаты исследований показали, что существует хорошее программное обеспечение для обработки естественного языка: General Architecture for Text Engineering (GATE); LingPipe; Natural Language Toolkit (NLTK). Для решения проблемы делимитации слов китайского языка рекомендуют использовать ICTCLAS (Institute of Computing Technology, Chinese Lexical Analysis System)[4]. Для европейских языков для решения проблемы стемминга рекомендуют использовать Snowball[5].

Классификация документов

Классификация документов – одна из задач информатики, заключающаяся в отнесении документа к одной из нескольких категорий на основании содержания документа. Используются методы информационного поиска и машинного обучения[6].

Постановка задачи классификации

Определим задачу формально. Пусть задано некоторое конечное множество категорий $C = \{c_1,...,c_{|c|}\}$, конечное множество документов $D = \{d_1,...,d_{|D|}\}$ и некоторая вначале неизвестная целевая функция Φ, которая для каждой пары <документ, категория> определяет, соответствуют ли они друг другу: $\Phi: D \times C \rightarrow \{0,1\}$. Задача состоит в том, чтобы найти максимально близкую к функции Φ функцию Φ'. Функцию Φ' называют классификатором. Машинное обучение основывается на начальном множестве документов $\Omega = \{d_1,...,d_{|\Omega|}\} \subset D$. При этом, значение целевой функции Φ известно для каждой пары $< d_j, c_i > \in \Omega \times C$. Документы из Ω разделяют на два непересекающиеся множества:

"Учебное" множество $T_r = \{d_1,...,d_{|T_r|}\}$ и множество документов, с помощью которой создается классификатор Φ'. Φ' обучается индуктивно, основываясь на замеченных характеристиках этих документов.

"Тестовое" множество $T_e = \{d_{|T_r+1|},...,d_{|\Omega|}\}$ и множество документов, на котором тестируется эффективность построенного классификатора. Каждый "тестовый" документ подается на вход классификатора Φ', а затем сравнивается результат классификатора $\Phi'(d_j, c_i)$ с известным значением функции $\Phi(d_j, c_i)$. Классификатор считается тем эф-

фективнее, чем чаще эти значения совпадают.

Документ $d \in \Omega$, называется положительным или отрицательным примером для категории *c*, если значение функции *Ф(d,c)* равно *1* или *0*, соответственно[7].

Методы машинной классификации

Существует несколько наиболее известных методов классификации:

1. Метод Байеса.

Метод Байеса основан на анализе совместных распределений признаков документа и категорий [8]. Документу $D =< d_1, d_2, ..., d_n >$ сопоставляется наиболее вероятная апостериорная категория, определяемая по формуле:

$$c^* = \arg\max_{c \in C} P(c \mid x_1 = d_1, x_2 = d_2, ..., x_n = d_n) \quad (1)$$

Апостериорная вероятность принадлежности документа некоторой категории вычисляется по формуле Байеса, связывающей априорную вероятность с апостериорной:

$$P(c \mid x_1 = d_1, x_2 = d_2, ..., x_n = d_n) = \frac{P(x_1 = d_1, x_2 = d_2, ..., x_n = d_n \mid c) \cdot P(c)}{P(x_1 = d_1, x_2 = d_2, ..., x_n = d_n)} \quad (2)$$

Подставляя (2) в (1), получаем:

$$c^* = \arg\max_{c \in C} \frac{P(x_1 = d_1, x_2 = d_2, ..., x_n = d_n \mid c) \cdot P(c)}{P(x_1 = d_1, x_2 = d_2, ..., x_n = d_n)} \quad (3)$$

Так как знаменатель не зависит от категории, его можно исключить:

$$c^* = \arg\max_{c \in C} P(x_1 = d_1, x_2 = d_2, ..., x_n = d_n \mid c) \cdot P(c) \quad (4)$$

Условные вероятности $P(x_1 = d_1, x_2 = d_2, ..., x_n = d_n \mid c)$ можно вычислить в предположении условной независимости переменных $x_1, x_2, ..., x_n$. В этом случае, формула для определения наиболее вероятной категории будет выглядеть следующим образом:

$$c^* = \arg\max_{c \in C} P(c) \cdot \prod_{i=1..n} P(x_i = d_i \mid c) \quad (5)$$

Для множества обучающих документов вероятности $P(x_i = d_i \mid c)$ вычисляются по формуле [9,10,11]:

$$P(x_i = d_i \mid c) = \frac{|\{D \in Ex \mid c \in Rub(D) \wedge D_i = d_i\}| + 1}{|\{D \in Ex \mid c \in Rub(D)\}|} \quad (6)$$

2. Метод опорных векторов SVM (Support Vector Machines).

Метод опорных векторов [12,13,14] разработан на основе принципа структурной минимизации риска – одновременного контроля количества ошибок классификации на множестве для обучения и «степени обобщения» обнаруженных зависимостей.

Нахождение оптимальной плоскости разделения множеств методом SVM сводится к решению оптимизационной задачи с линейными ограничениями типа равенств и неравенств [12]:

$$L_D(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \to \max, \quad 0 \le \alpha_i \le C, \sum_{i=1}^{l} \alpha_i y_i = 0 \quad (7)$$

Здесь $K(x_i, x_j)$ – функция ядра SVM, которая в простейшем случае равна евклидову скалярному произведению векторов x_i и x_j. Для решения задачи (7) предложены эффективные методы решения [15,16].

3. Метод ближайших соседей.

Метод ближайших соседей – простейший метрический классификатор, основанный на оценивании сходства объектов. Метод ближайшего соседа является, пожалуй, самым простым алгоритмом классификации. Классифицируемый объект *x* относится к тому классу y_i, которому принадлежит ближайший объект обучающей выборки x_i. Метод *k* ближайших соседей. Для повышения надёжности классификации объект относится к тому классу, которому принадлежит большинство из его соседей – *k* ближайших к нему объектов обучающей выборки x_i. В задачах с двумя классами число соседей берут нечётным, чтобы не возникало ситуаций неоднозначности, когда одинаковое число соседей принадлежат разным классам.

Основная формула: Пусть задана обучающая выборка пар «объект–ответ» $X^m = \{(x_1, y_1), ..., (x_m, y_m)\}$. Пусть на множестве объектов задана функция расстояния *ρ(x,x')*. Эта функция должна быть достаточно адекватной моделью сходства объектов. Чем больше значение этой функции, тем менее схожими являются два объекта *x,x'*. Для произвольного объекта *u* расположим объекты обучающей выборки x_i в порядке возрастания расстояний до *u*: $\rho(u, x_{1;u}) \le \rho(u, x_{2;u}) \le ... \le \rho(u, x_{m;u})$, где через $x_{i;u}$ обозначается тот объект обучающей выборки, который является *i*-м соседом объекта *u*. Аналогичное обозначение введём и для ответа на *i*-м соседе: $y_{i;u}$. Таким образом, произвольный объект *u* порождает свою перенумерацию выборки. В наиболее общем виде алгоритм ближайших соседей есть:

$$a(u) = \arg\max_{y \in Y} \sum_{i=1}^{m} [x_{i;u} = y] \omega(i,u) \qquad (8)$$

где $\omega(i,u)$ – заданная весовая функция, которая оценивает степень важности *i*-го соседа для классификации объекта *u*. Естественно полагать, что эта функция неотрицательна и не возрастает по *i*. По-разному задавая весовую функцию, можно получать различные варианты метода ближайших соседей, $\omega(i,u) = [i = 1]$ – простейший метод ближайшего соседа, $\omega(i,u) = [i \le k]$ – метод *k* ближайших соседей[17].

Оценка качества классификации

С целью определения наилучшего метода были предложены следующие критерии: полнота; точность. Для оценки качества классификации используются метрики из информационного поиска:

Полнота (recall): отношение количества найденных документов из категории к общему количеству документов категории.

Точность (precision): Определяется как отношение числа релевантных документов, найденных ИПС (информационно-поисковые системы), к общему числу документов.

$$recall = \frac{|D_{rel} \cap D_{retr}|}{|D_{rel}|} \qquad (9)$$

$$precision = \frac{|D_{rel} \cap D_{retr}|}{|D_{retr}|} \qquad (10)$$

где D_{rel} – это множество релевантных документов в базе, а D_{retr} – множество документов, найденных системой[18].

Тестовые коллекции для классификации

Существует несколько наиболее известных коллекции для экспериментов.

Коллекции для английского языка: Reuters-21578 Text Categorization Collection (http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html); OHSUMED Test Collection (http://ir.ohsu.edu/ohsumed/); 20 Newsgroups Data (http://people.csail.mit.edu/jrennie/20Newsgroups).

Коллекции для китайского языка: SogouT (http://www.sogou.com/labs/dl/t.html).

Коллекции для русского языка: Тестовые коллекции РОМИП (Российский семинар по Оценке Методов Информационного Поиска) (http://romip.ru/ru/collections/index.html).

Результаты сравнение методов машинного обучения

В статьях [7,8,10,11] сравниваются различные методы машинного обучения. Проведенные эксперименты показали, что метод SVM имеет преимущество перед другими методами машинного обучения, поскольку обеспечивают самую высокую точность и полноту. Метод k ближайших соседей обеспечивает наилучшее время но самую низкую точность и полноту. Метод Байеса даёт средние временные характеристики, а также точность и низкую полноту. Поэтому можно рекомендовать для классификации использован SVM метод.

Выводы

Выбранный метод и результаты работы подтверждают возможность создания эффективной системы автоматической классификации документов для конечного множества языков по критерию принадлежности к определенной области знания, используя современные средства вычислительной техники.

Список литературы

1. http://ru.wikipedia.org/wiki/Обработка_естественного_языка.
2. Загибалов Т. Е. Автоматический анализ текстов на китайском языке. Проблема выбора базовой единицы. // Труды международной конференции "Диалог 2005".
3. http://ru.wikipedia.org/wiki/Стеммер_Портера.
4. http://www.ictclas.org/
5. http://snowball.tartarus.org/
6. http://ru.wikipedia.org/wiki/Классификация_документов.
7. Юрий Лифшиц. Курс "Алгоритмы для Интернета" РАН 2006.
8. Yang Y., Liu X. A re-examination of text categorization methods. // Proc. of Int. ACM Conference on Research and Development in Information Retrieval (SIGIR-99), 1999 — p. 42-49.
9. Joachims T. A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. // Proceedings of ICML-97, 14th International Conference on Machine Learning. — 1996. — p. 2-13.
10. Yang Y. An Evaluation of Statistical Approaches to Text Categorization. / Journal of Information Retrieval, 1999 — V.1 — p. 67--88.
11. Joachims T. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. // Proceedings of ECML-98, 10th European Conference on Machine Learning — 1998. — p. 6-18.
12. Vapnik V. The Nature of Statistical Learning Theory. — Springer-Verlag — New York, 1995. — p. 123-167.
13. Burges C.J.C. A tutorial on support vector machines for pattern recognition. // Data Mining and Knowledge Discovery, 1998. — p. 955-974,
14. Вапник В.Н. Восстановление зависимостей по эмпирическим данным. — М.: Наука, 1979. —с. 223-255.
15. Joachims T. Making Large-Scale SVM Learning Practical. Advances in Kernel Methods // Support Vector Learning, Burges C., Smola A. (ed.), — MIT-Press, 1999. —p. 5-12.
16. Joachims T. Estimating the Generalization Performance of a SVM Efficiently. // Proceedings of the International Conference on Machine Learning, — Morgan Kaufman, 2000. —p. 5-22.
17. http://www.machinelearning.ru/wiki/index.php?title=Метод_ближайших_соседей.
18. http://ru.wikipedia.org/wiki/Информационный_поиск.

Поступила в редакцию 17.12.2009