Классификатор веб-страниц на базе SVM-Multiclass

© Р.Ф. Кузнецов

Балтийский Государственный
Технический Университет
ruslkuznetsov@gmail.com

Аннотация

Статья посвящена задаче классификации веб-страниц в рамках семинара РОМИП'2006. Рассмотрены вопросы рубрикации веб-страниц с использованием обучающей выборки.

1. Введение

Целью моего участия в семинаре РОМИП'2006 являлось желание опробовать свои силы в задаче классификации веб-страниц на коллекции, предложенной семинаром, и сравнить свои результаты с результатами других участников. Кроме того, меня интересовала возможность оценки влияния предложенного мной метода выделения значимой части веб-страниц на качество классификации. Непредвзятость асессоров гарантировалась методологией оценки и явилась дополнительным стимулом для работы над выбранной задачей.

Автоматическая классификация (категоризация, рубрикация) текста является одной из областей информационного поиска, к которой за последние годы проявился значительный интерес. Появление большого числа работ, освещающих эту тему, связано в первую очередь с быстрым ростом цифровых коллекций документов (в основном в Интернете и локальных сетях организаций) нуждающихся в классификации.

Все существующие методы автоматической классификации текста можно разбить на два класса – *методы, основанные на знаниях* (или иначе «инженерный подход») и *методы машинного обучения*.

В настоящее время, в связи с ростом возможностей компьютеров, наибольшей популярностью пользуется второй класс методов.

Однако при применении в задачах классификации методов машинного обучения возникают некоторые проблемы. Вот лишь некоторые из них:

Загрязненность обучающего множества. Это главная проблема, с которой приходится сталкиваться исследователям при построении автоматического классификатора на базе методов машинного обучения. Для обучения обычно используются распределенные по темам наборы документов, рубрикацией которых занимается группа экспертов. Вследствие того, что эксперты не всегда приходят к единому мнению относительно правильности отнесения документа к той или иной теме, возникает проблема - противоречивость интерпретации оценщиков, что в свою очередь, неминуемо приводит к противоречивости обучаемого множества [3]. Например, не всегда очевидно, куда нужно отнести сайт, посвященный увлечению Владимира Путина вольной борьбой. Есть несколько возможных вариантов. Первый – в рубрику посвященную политике, второй – в рубрику посвященную спорту, быть может, одновременно в ту и другую, а при определенных обстоятельствах ни в одну из них.

Не достаточно изучены подходы к построению иерархических каталогов. Хотя существует множество исследований, посвященных автоматической классификации текстов, в основном они предлагают «плоскую» классификацию. Отличительной особенностью такого подхода является отсутствие иерархии в рубрикаторе. Часто это оправдано, но многие задачи не возможно решить с помощью «плоской» классификации. В тех же работах, где предлагается подход к решению этой проблемы, в основном, в качестве решения используется бинарная классификация (для каждого листа иерархического каталога определяется, принадлежит ли ему данный сайт или нет), что при значительном числе рубрик приводит к неоправданной трате времени при построения классификатора. К тому же, при таком подходе, лишь малая часть документов принадлежит исследуемой рубрике, в то время как все остальные документы обучающей выборки к ней не принадлежат. Подобный перекос обучающей выборки приводит к тому, что системы автоматической классификации теряют свою эффективность.

Не достаточно изучены подходы к извлечению значимой для классификации информации из HTML-документов. Вследствие того, что основные методы автоматической классификации текста разрабатывались как универсальные, ощущается недостаток работ специализирующихся на классификации веб-страниц и сайтов. Ат-

рибуты HTML-страниц (такие как заголовок, размер и цвет шрифта, наличие ключевых слов в адресе страницы и пр.) могут дать множество значимой информации о принадлежности веб-страницы к определенной теме.

2. Обзор существующих методов

В общем случае построение автоматического классификатора веб-страниц на базе обучающего множества можно разделить на следующие этапы:

- извлечение текстовой информации из HTML-страниц представленных в выборке;
- приведение текстовой информации к виду пригодному для использования в методах машинного обучения;
- применение выбранного метода машинного обучения к обучающему множеству (построение модели классификатора);
- классификация страниц тестового множества;

Рассмотрим эти этапы подробнее.

2.1 Извлечение текстовой информации из HTML-страниц

Особенностью представления документов в сети Интернет является наличие на странице, помимо самого текста (содержательной части веб-документа) определяющего предмет страницы, большого количества вспомогательных элементов (навигационной части web-документа) призванных обеспечить навигацию по страницам сайта. Часто эти элементы не имеют прямого отношения к теме страницы и поэтому могут отрицательно влиять на качество информационного поиска [7, 5, 6] вообще и классификации [7] в частности.

Основываясь на этих догадках можно сделать предположение, что удаление или уменьшение веса навигационной части может оказать положительное влияние при решении задачи классификации веб-страниц.

Методы, применяемые для извлечение текстовой информации из HTML-страниц, можно разделить на два основных типа:

1. методы, основанные на выделении повторяющихся для всех (или части) страниц сайта фрагментов информации [1, 2, 4].
2. методы, основанные на анализе dom-деревьев страниц сайта [7, 5].

Методы, основанные на выделении повторяющихся фрагментов страниц одного сайта, показывают более высокие результаты и являются более универсальными, чем методы, основанные на анализе dom-деревьев. Однако для их работы необходима информация обо всех страницах сайта (или хотя бы части из них). Это не всегда возможно и при реализации алгоритма увеличивает время его работы, в связи с необходимостью анализа большого количества страниц.

2.2 Приведение текстовой информации к виду пригодному для использования в методах машинного обучения

После того, как из HTML-страницы извлечена текстовая информация, наступает этап ее приведения к виду, пригодному для использования в методах машинного обучения.

Большинство методов автоматической классификации текстов основаны на том предположении, что тексты каждой тематической рубрики содержат в себе некие отличительные признаки и наличие или отсутствие этих признаков говорит о принадлежности текста к той или иной рубрике. Одной из самых простых и получивших наибольшее распространение моделей представления документа, стала модель «bag-of-words». В ней текст страницы представлен как набор терминов, совокупность которых и определяет смысл документа, а соответственно и его тему.

Методы машинного обучения не могут использовать сами отличительные признаки (например, слова) в качестве непосредственного объекта для анализа, так как рассчитаны на работу с числовыми данными. Все они в качестве объектов анализа используют сжатое представление текста в виде вектора термов (или, как еще говорят, вектора признаков).

Наиболее исследованным и распространенным решением, при определении величин входящих в вектор термов, является подход, основанный на предположении, что документы, принадлежащие одной рубрике, имеют близкие распределения относительных частот слов входящих в текст.

Таким образом, при определении вектора термов для документа (размерность которого равна числу различных термов из всего массива), каждому слову из лексики коллекции ставится в соответствие координата в пространстве признаков. Она пропорциональна частоте слова в данном документе.

Для определения этих координат, в случае весового представления текста, часто используют стандартную *tfidf* функцию [8], которая определяется как

$$tfidf(t_t, d_j) = tf(t_t, d_j) * idf(t_t),$$

где *tf – частота терма (term frequency)* t_t в документе d_j, *idf - инверсная частота терма (inverse document frequency)*:

$$idf(t_t) = \log \frac{|D|}{df_t}$$

В этой формуле $|D|$ - количество документов содержащихся в обучающей коллекции, df_t - количество документов содержащих терм t_t.

Но кроме этого, нам необходимо учесть различную длину документов коллекции. Поэтому, обычно, веса, полученные с помощью функции *tfidf*, подвергают нормализации [9]

$$\omega_{tj} = \frac{tfidf(t_t, d_j)}{\sqrt{\sum_{s=1}^{|T|} (tfidf(t_s, d_j))^2}}$$

В этой формуле вес *t*-го терма в *j*-ом документе ω_{tj} рассчитывается исходя из того, чтобы сумма квадратов весов каждого документа была равна 1.

Размерность пространства при построении вектора признаков, равна числу различных термов содержащихся в обучающей коллекции. Многие алгоритмы классификации очень чувствительны к времени вычисления, которое часто является функцией от длины вектора, представляющего документ, поэтому необходимо стараться уменьшить размерность пространства признаков.

Первый и наиболее популярный подход для решения этой задачи – удаление из вектора признаков так называемых стоп-слов. *Стоп-слова* – это наиболее распространенные (встречающиеся в большинстве документов) и чаще всего высокоранговые слова, которые не несут высокой смысловой нагрузки и обычно используются для связи слов в предложении.

Одними из самых распространенных подходов, к уменьшению размерности пространства признаков, являются стемминг [10], алгоритмы, основанные на правилах словообразования [11], а также их комбинации. Они используют предположение, что приведение всех встречающихся словоформ к одной форме может положительно сказаться на результате классификации.

2.3 Построение модели классификатора

После построения вектора признаков анализируемых страниц мы приступаем к использованию методов машинного обучения для построения модели классификатора.

Существует большое количество методов машинного обучения используемых для задач классификации текстов. Они отличаются следующими характеристиками:

- скорость обучения и классификации
- качество классификации

Наибольшее распространение получили такие методы как метод k–ближайших соседей [12], метод опорных векторов [13], метод Байеса [14] и другие.

2.4 Классификация страниц тестового множества

С помощью построенной модели производится классификация страниц тестового множества. Этот этап является завершающим при построении классификатора.

Разнообразие алгоритмов, предложенных для автоматической классификации текстов, объясняется, в основном, не одинаковым качеством их работы на различных коллекциях документов. Для одного и того же алгоритма качество классификации зависит от свойств коллекции – ее размера, непротиворечивости, иерархичности, числа рубрик и т.д. Это, в свою очередь, приводит к необходимости настройки методов под обучающее множество (в нашем случае под коллекцию РОМИП).

3. Реализация классификатора веб-страниц на коллекции РОМИП

Коллекция веб-страниц РОМИП, состоит из обучающего и тестового множеств. Обучающее множество представлено 2 100 сайтами из каталога DMOZ.ORG (300 000 веб-страниц), тестовое 22 000 сайтами из каталога NAROD.RU (728 000+ веб-страниц, 7+ Gb).

Первым шагом было извлечение текстовой информации из HTML-страниц принадлежащих обоим множествам. Однако, как уже упоминалось выше, помимо использования всего текста, представленного на странице, было бы разумно попытаться выделить и использовать для задачи классификации только содержательную часть веб-документа, что могло бы положительно сказаться на качестве классификации.

Исходя из этого, асессорам были предоставлены два прогона. Их основное отличие заключалось в том, что второй прогон, в отличие от первого, использовал алгоритм выделения значимой части веб-страницы.

В качестве алгоритма был применен разработанный мною метод, использующий выделение предложений. Рассмотрим его подробнее.

3.1 Алгоритм выделения значимой части

Алгоритм основан на двух предположениях:

1. содержательная часть страницы обычно включает предложения;
2. вся информация, находящаяся в структурном блоке, включающем предложения, относится к содержательной;

Структурным блоком будем называть фрагмент текстового содержания страницы, обрамленный тегами, не позволяющими сохранить единство предложения (например, <p>,
, <table> и т.п.).

Если мы рассмотрим различные типы web-страниц, то придем к определенным выводам. В большинстве случаев предложения, как единица связной речи, является надежным признаком того, что текстовая информация относится к содержательной части страницы.

Далее, рассматривая структурные блоки, мы видим, что те из них, что включают предложения, почти всегда относятся к содержательной части web-страницы.

Однако отсутствие в блоке предложения не гарантирует того, что он относится к навигационной части. Например, рассмотрим заголовок веб-страницы. Обычно в конце заголовка знаков препинания не ставят. Тем не менее, в этом случае мы можем использовать следующую эвристику: часто заголовок страницы выделен относительного нижерасположенного текста (например, тегами <b>, <h1>, <h2>, <h3> и т.п. и/или размером шрифта) и не содержит ссылок. Эти особенности также будут использованы в анализируемом методе.

При рассмотрении алгоритма можно выделить следующие этапы:

1. представление HTML-документа как совокупность структурных блоков
2. поиск предложений в каждом из блоков
3. поиск заголовков для блоков, отнесенных к содержательным

Алгоритм разбивает HTML-документ на структурные блоки, используя следующие тэги - <p>,
, <td>, , <hr>, <form>.

Далее в каждом блоке ищется хотя бы одно предложение. Поиск предложений основан на знаках препинания, таких как точка, восклицательный и вопросительный знаки. Алгоритм ищет точку после слова. «Слово» перед точкой может заканчиваться как буквой, так и одним из знаков препинания (например, : ( ) {} [] ‘ ’ “ ” < >) или цифрой.

После того, как с помощью первых двух этапов работы алгоритма web-страница будет разделена на навигационную и содержательную части, происходит поиск содержательной информации среди блоков, не включающих предложения. Для блоков, приписанных к содержательной части, перед которыми расположены блоки из навигационной части, алгоритм ищет заголовки. Иными словами мы проверяем, является ли текстовая строка (приписанная к навигационной части и расположенная перед блоком, приписанным к содержательной части), заголовком.

При решении этой задачи мы сравниваем текст первого и второго блоков. Если текст первого выделен относительно с текста второго (отмечен такими тэгами как <b>, <h1>, <h2>, <h3> и т.п. и/или размером шрифта), то он признается заголовком и приписывается к содержательной части рассматриваемой web-страницы.

Целью использования этого алгоритма являлось желание оценить влияние его применения на качество классификации веб-страниц.

3.2 Дальнейшая работа по построению классификатора

Далее все страницы приводились к виду пригодному к использованию методами машинного обучения. Основной задачей на этом этапе было сокращение размерности пространства признаков. Все словоформы были приведены к их корню с помощью стемминга.

Помимо этого, из рассмотрения были удалены все термы, встречающиеся менее чем на 175 страницах коллекции (для того, чтобы размерность пространства признаков составила, приблизительно, 50000). Также не учитывались страницы, содержащие менее 5 термов.

В результате размерность пространства признаков составила 44430 для первого прогона и 39735 для второго.

Далее каждая страница была представлена в виде вектора термов. Для определения веса каждого терма на странице была использована формула описанная выше. Таким образом, все страницы кол-

лекции были подготовлены к использованию методами машинного обучения и встал вопрос выбора наиболее подходящего из них.

В результате сравнений различных методов машинного обучения чаще всего наилучшие результаты показывал метод опорных векторов [12]. И так как передо мной была поставлена задача, добиться высокого качества классификации, а время на ее реализацию было ограничено, я решил воспользоваться готовым приложением.

Этим приложением стал SVM-Multiclass [15]. Его преимуществом, перед ставшим фактически стандартом в приложениях метода опорных векторов SVM-Light, являлось то, что он специально разработан для задач мультитематической классификации.

В результате тестовых прогонов всплыла еще одна проблема. Размер обучающего множества составлял 300 000 веб-страниц, и использование всех из них в обучении приводило к чрезвычайно большим временным затратам. Кроме того, далеко не все страницы сайта могут относиться к теме, к которой данный сайт приписан, что вызывает загрязнение обучающей выборки. Действительно, редакторы Интернет-каталогов, в большинстве случаев, принимают решения о принадлежности сайта, основываясь лишь на нескольких его страницах. Наиболее часто это главная страница сайта и страницы ближайшие к ней.

Исходя из этого, было принято решение о сокращении выборки за счет того, что в нее были включены только страницы сайта, находящиеся на расстоянии не более одного клика от главной. В результате, объем обучающего множества значительно сократился.

Таким образом, асессорам были предложены два прогона. Их характеристики представлены в таблице 1.

	Использование метода извлечения значимой информации из веб-страниц	Размерность пространства признаков	Метод машинного обучения	Используемое приложение
1-й прогон	нет	44430	Метод опорных векторов	SVM-Multiclass
2-й прогон	да	39735	Метод опорных векторов	SVM-Multiclass

Таблица 1. Характеристики прогонов.

4. Анализ результатов

Результаты оценки прогонов асессорами представлены в таблице 2.

	1-й прогон		2-й прогон	
	AND	OR	AND	OR
F1 (micro average)	0.1322	0.1577	0.0011	7.75E-4
Recall	0.2542	0.0866	0.1684	3.50E-4
Precision (micro average)	0.1117	0.2133	9.35E-4	9.35E-4
Error	0.0045	0.0084	0.0042	0.0091
F1	0.1390	0.0891	0.0842	4.36E-4
Recall (micro average)	0.1620	0.1251	0.0016	6.62E-4
Accuracy	0.9954	0.9915	0.9957	0.9908
Precision	0.1507	0.1511	0.0839	5.78E-4

Таблица 2. Результаты прогонов согласно схеме B (учитывались только оцениваемые документы).

Не высокие значения суммарных оценок, представленных в таблице 2, вполне объяснимы. Дело в том, что для 6-ти из 24-х оцениваемых тем (для первого прогона), асессорами не было рассмотрено ни одного документа, а для 9 тем – не больше 15 документов. Таким образом, небольшое количество оцененных документов, для более чем половины тем, значительно повлияло на общие результаты.

Для остальных оцененных тем первого прогона, классификатор показал различные результаты. Например, для темы "Бизнес -> Электроника и электротехника" (141 оцененная страница) значение F1 (сильная релевантность) составило 0.0487, а для темы "Искусство->Кино" (133 оцененных страницы) значение F1 (сильная релевантность) составило 0.4017.

Рассматривая результаты второго прогона можно сделать следующий вывод: в том виде, в котором метод извлечения значимой информации из веб-страницы был применен, он негативно сказывается на качестве классификации. Только для трех тем значение F-меры было больше нуля.

Возможные причины низкого результата:

- некорректная работа метода (алгоритм совершает ошибки при разделении значимой и навигационной частей веб-страницы)

- метод, удаляя навигационную часть веб-страницы, значительно уменьшает размер документа, что не дает классификатору корректно его оценить
- часть навигационной информации (например, пункты меню) может быть полезна в задаче классификации

Однако, на мой взгляд, столь низкий результат не является доказательством бесполезности данного подхода в задачах классификации веб-страниц. Дальнейшее исследования, возможно, смогут изменить его влияние на качество классификации в лучшую сторону.

Возможные пути улучшения результата:

- уменьшить число ошибок, при выделении содержательной части веб-страницы, путем совершенствования метода
- при учете навигационной части веб-страницы использовать понижающие коэффициенты, а не удалять ее полностью

5. Выводы

Эксперименты в рамках семинара РОМИП позволили объективно оценить качество работы представленного классификатора, а также проверить гипотезу о том, что удаление навигационной части веб-страницы положительно влияет на качество рубрикации веб-страниц.

Классификатор, построенный на базе приложения SVM-Multiclass, доказал свою работоспособность. Однако, также, стала очевидна необходимость более тщательной подгонки предложенной модели под обучающую и тестовую выборку.

Классификатор, построенный с применением метода извлечения значимой информации из веб-страниц, показал низкие результаты. Это можно объяснить как недостаточно корректной работой метода, так и неоднозначным влиянием удаления подобной информации на качество классификации.

Литература

[1] *М.С. Агеев, И.В. Вершинников, Б.В. Добров.* Извлечение значимой информации из web-страниц для задач информационного поиска. Интернет-математика 2005. Сборник работ по программам научных стипендий Яндекса. Москва, 2005.

[2] *И. Некрестьянов, Е. Павлова.* Обнаружение структурного подобия HTML-документов. Труды четвертой всероссийской конференции RCDL'2002, 38-54, Дубна, Россия, 2002.
[3] *F.W. Lancaster.* Indexing and Abstracting in Theory and Practice. Second Edition. Champaign, IL: Graduate School of Library and Information Science; 1998
[4] *S. Gupta, G.E. Kaiser, P. Grimm, M. F. Chiang and J. Starren,* Automating Content Extraction of HTML Documents. World Wide Web Journal, January 2004.
[5] *C. H. Lee, M.Y. Kan and S. Lai.* Stylistic and Lexical Co-training for Web Block Classification. In the Proceedings of Workshop on Web Information and Data Management (WIDM '04), USA, 2004.
[6] *L. Yi, and B. Liu*, Web Page Cleaning for Web Mining through Feature Weighting. In the Proceedings of Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03), Acapulco, Mexico, August, 2003.
[7] *L. K. Shih and D. Karger*. Using URLs and table layout for web classification tasks. In Proceedings of the 13th International Conference on the World Wide Web, pages 193--202, New York, NY, 2004.
[8] *M. Kantrovtiz, B. Mohit and V. Mittal*. Stemming and its effects on TFIDF Ranking. In proceedings of ACM-SIGIR 2000 Athens, Greece.
[9] *F. Debole and F. Sebastiani*, Supervised term weighting for automated text categorization. In the Proceedings of SAC-03, 18th ACM Symposium on Applied Computing, Melbourne, US: ACM Press, New York, US, 2003, pp. 784--788.
[10] *D.A. Hull.* Stemming Algorithms – A Case Study for Detailed Evaluation, JASIS, 47(1): 70-84, 1996
[11] *M.F.Porter*. An algorithm for suffix stripping, *Program*, 14(3) :130-137, 1980.
[12] *Y. Yang and X. Liu.* A re-examination of text categorization methods. // Proc. of Int. ACM Conference on Research and Development in Information Retrieval (SIGIR-99), 1999 — pp. 42-49.
[13] *T. Joachims.* Text categorization with support vector machines: learning with many relevant features. In Proceedings of ECML-98, 10th European Conference on Machine Learning (Chemnitz, Germany, 1998), 137–142, 1998.
[14] *Charles Elkan.* Naive Bayesian Learning. Adapted from Technical Report No. CS97-557, Department of Computer Science and Engineering, University of California, San Diego, September 1997.
[15] *Thorsten Joachims.* SVM-Multiclass. http://www.cs.cornell.edu/People/tj/svm%5Flight/svm_multiclass.html