САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

КАФЕДРА ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

Зворыкин Егор Артемович

Выпускная квалификационная работа бакалавра

Классификация документов посредством кластеризации графа ссылок между ними

Направление 010400

Прикладная математика и информатика

Научный руководитель,

старший преподаватель

Мишенин А. Н.

Санкт-Петербург

2016

Содержание

Введение

С каждым годом все больше устройств получают доступ к сети Интернет, а значит, все больше пользователей делится друг с другом информацией. Данные становятся менее структурированными, а потребность в их анализе растёт. Исследования в этой области связаны на сегодняшний день с информационным поиском. Одной из актуальных задач информационного поиска является классификации документов.

Существует множество способов находить скрытые структуры в данных, одним из них является анализ графа ссылок между документами. Во многих случаях классификация коллекции документов может быть сведена к исследованию графа, порожденного ею.

Мы рассмотрим случай классификации статей на сайте Википедия, используем для решения этой задачи анализ графа ссылок, а также сравним, полученные результаты с текстовой классификацией.

Глава 1. Обзор традиционных алгоритмов текстовой классификации

1.1. Подготовка данных

Все методы машинного обучения направлены на работу с числовыми векторами, соответственно, текстовые документы необходимо тем или иным образом привести к виду, пригодному для применения алгоритмов. Существует два основных подхода к решению этой проблемы.

Мешок слов

При данном подходе появление каждого слова в тексте предполагается независимым, что является значительным упрощением текстовой модели, однако на практике зачастую этого бывает достаточно для задач обработки текстов. Из этого предположения вытекает, что документ может быть представлен, как набор слов, входящих в него в совокупности с их частотами. Для получения вектора проводится предобработка документов и составляется словарь уникальных слов. Затем для каждого документа по словарю составляется вектор, размерность которого равна длине словаря, а компонентами являются частоты вхождения, соответствующего слова в данный документ.

Такую модель представления текстов иногда называют униграммной. Она может быть расширена, чтобы включать словосочетания из пары или больше слов. Такие модели называют N-граммными. В этом случае составляется словарь не только из одиночных слов, но также из всех сочетаний N слов – N-грамм. Большинство N-грамм встречается один раз, и поэтому они могут быть выброшены из словаря. Обработка N-грамм является очень затратной и редко используется для N больше двух. Элементы словаря в таком случае называют терминами, объединяя понятия слова и N-граммы. Договоримся впредь всегда называть элементы словаря терминами.

Одним из недостатков модели мешка слов является существование в естественных языках часто встречающихся слов, появление которых не зависит от темы текста. Это, например, предлоги, союзы, артикли. Такие слова называют стоп-словами, их часто вручную исключают из рассмотрения. Однако это не всегда представляется возможным. Кроме того, существуют слова, которые могут встречаться в документах очень редко, но являющиеся специфичными для какой-то темы, кажется логичным оценивать вклад таких слов выше. Так появился второй подход к представлению документов.

TF-IDF

Для решения этих проблем вводится понятие документная частота термина[1], df_t, – количество документов в коллекции, содержащих данный термин. TF-IDF расшифровывается, как term frequency – inverse document frequency, и является произведением частоты термина, $tf_{t,d}$, и обратной документной частоты, idf_t. Обратная документная частота определяется для каждого термина, как $\log\left(\frac{M}{df_t}\right)$, где M – число документов в коллекции. Таким образом, термины, часто встречающиеся в небольшом числе документов, получают наибольший вес.

1.2. Алгоритм k-means

Постановка задачи кластеризации

Кластерный анализ относится к обучению без учителя – это класс алгоритмов машинного обучения, где система выполняет поставленную экспериментатором задачу без предоставленного обучающего множества. Обучающее множество – это множество объектов, той же природы, что и исследуемые, но ответ на поставленную задачу для которых уже есть.

Задача кластерного анализа ставится следующим образом. Имеется набор объектов $X^m = \{x_i\}_{i=1}^m$ из некоторого множества объектов X, на котором задана функция расстояния: $\rho(x, x')$, также имеется множество меток кластеров Y. Необходимо сопоставить каждому объекту метку кластера таким образом, чтобы объекты одного кластера были близки по метрике, а объекты разных кластеров находились далеко друг от друга. Число кластеров может быть известно или находиться в процессе обучения. Применительно к коллекции документов задача кластеризации – это задача разбиения документов по темам.

Принцип работы алгоритма k-means

Для работы алгоритма необходимо указать число кластеров k. Затем в пространстве объектов случайным образом выбирается k объектов, называемых центроидами. Дальше начинается итерационный процесс: для каждого объекта из множества X^m высчитывается расстояние до каждого центроида, и объекту приписывается метка того кластера, для которого расстояние наименьшее. После этого шага, получив для каждого объекта метку класса, высчитываем новые положения для центроидов, как среднее значение всех объектов, приписанных одному кластеру. Алгоритм завершается, когда для каждого объекта кластер остается тем же, что на предыдущем шаге.

Конечная цель алгоритма – минимизировать целевую функцию – сумму квадратов разностей между объектами кластеров и их центроидами. Хотя сходимость алгоритма гарантирована, иногда некоторые начальные положения кластеров могут привести к неоптимальному решению [2]. Чтобы избежать этого, алгоритм запускается несколько раз из разных начальных положений, а в качестве ответа берется то разбиение, для которого значение целевой функции является наименьшим. В [3] также предложен метод инициализации кластеров, ускоряющий сходимость.

Применение

Алгоритм k-means можно использовать, как начальный этап, в задаче классификации. Полученные с помощью него центроиды можно использовать, как обучающие примеры для алгоритма k ближайших соседей, который уже решает задачу классификации.

1.3. Алгоритм k ближайших соседей

Постановка задачи классификации

Задача классификации отличается от задачи кластеризации наличием набора объектов с известными классами – обучающего множества. Формально пусть есть некоторая выборка объектов генеральной совокупности с известными метками классов $X^l = (x_i, y_i)_{i=1}^l \subset X \times Y$. Требуется найти классификационное правило $a : X \to Y$.

Описание работы

Алгоритм k ближайших соседей является метрическим методом, а для его работы достаточно определить функцию расстояния или аналогично функцию сходства между объектами и иметь набор классифицированных объектов. Фактические алгоритм никак не обучается, а все вычисления производятся непосредственно во время классификации. Для любого объекта вычисляется расстояние до всех объектов обучающей выборки (или сходство), из них выбираются k наиболее близких, затем объекту присваивается класс, наиболее часто встречающийся среди его соседей.

Применение

В контексте нашей задачи, алгоритм k ближайших соседей только после предварительной подготовки, так как векторы документов имеют очень большую размерность. С увеличением размерности значения компонент векторов все меньше влияют на расстояние между ними, соответственно для векторов такой большой размерности, как у векторов документов, все расстояния между объектами будут почти одинаковыми. Этот эффект известен, как проклятие размерности.

1.4. Метод опорных векторов (SVM)

Метод опорных векторов – это целый класс методов машинного обучения, основанный на разделении данных линейной поверхностью. Впервые был предложен в [4]. Он может применяться к разным задачам, но в первую очередь к задаче классификации.

Метод опорных векторов относится к классу линейных бинарных классификаторов и предполагает линейную разделимость данных. Как и все линейные классификаторы, он строит линейную разделительную поверхность в пространстве векторов таким образом, что все объекты одного класса лежат по одну сторону от разделяющей поверхности. В общем случае он имеет следующий вид:

$$a(x) = sign(\langle \vec{w}, \vec{x} \rangle)$$

И обучение состоит в нахождении коэффициентов $w = \{w_0, w_1, \ldots w_n\}$, которые и определяют разделяющую поверхность. Если существует хоть одна такая разделяющая поверхность, то их существует бесконечное множество. Метод опорных векторов выбирает в некотором смысле оптимальную поверхность, одинаково отдаленную от объектов обоих классов. Однако линейная разделимость далеко не всегда присутствует, поэтому наиболее популярной разновидностью метода является soft-margin SVM. Она позволяет некоторым объектам обучающей выборки быть неправильно классифицированными.

Также для классификации нелинейных данных есть методы для перехода в пространство более высокой размерности, где они предположительно станут линейно разделимы (kernel trick).

Применение для текстовой классификации

Векторы документов обладают свойствами, которые делают применение методов опорных векторов для них весьма эффективными. Во-

первых, как уже было сказано, они имеют высокую размерность, методы опорных векторов в свою очередь не склонны к переобучению, благодаря регуляризации, Во-вторых, большинство признаков играет определенную роль, следовательно, использование жесткого отбора признаков понесет за собой потерю информации. В-третьих, для каждого документа лишь малая часть компонент не равна нулю, метод опорных векторов, согласно [5], подходит для решения таких задач. И наконец, чаще всего в задачах классификации документов присутствует линейная разделимость.

1.5. Латентное размещение Дирихле (LDA)

Латентный семантический анализ

Более глубокий подход к моделированию текстов состоит в исследовании связей между появлениями тем и связанных с ними слов. Модель латентного семантического анализа (LSA) использует сингулярное разложение матрицы термин-документ, содержащей частоты слов, взвешенные затем особым образом, схожим с tf-idf образом. Затем производится снижение размерности, согласно свойствам сингулярного разложения, полученное преобразование используется для отображения терминов и документов в одно семантическое пространство заданной размерности, где полученные векторы могут сравниваться между собой.

Большой проблемой этого метода является вероятностная и статистическая необоснованность. Например, эта модель предполагает нормальное распределение слов и документов, в то время как, наблюдается скорее распределение Пуассона. Вычисления, основанные на этой модели, сильно затрудняются с повышением числа документов и терминов. Это привело к разработке модели вероятностного латентного семантического анализа, которая основана на смешанном разложении и использует модель скрытых переменных[6]

$$P(d,w) = P(d)P(w \mid d)$$

$$P(w \mid d) = \sum_{z \in Z} P(w \mid z)P(z \mid d),$$

где z является скрытой переменной, соответствующей теме.

Хоть этот подход и хорошо обоснован с точки зрения вероятностного моделирования слов, но генеративная модель документов в нем не сформирована, это приводит нас к Латентному размещению Дирихле. [7].

Генеративная модель LDA

Будем использовать следующие обозначения:

$$\vec{w} = \{w_{mn}\}_{m=1,n=1}^{M,N} = \{w_i\}_{i=1}^{W} \text{ – вектор слов коллекции}$$

$$\vec{z} = \{z_{mn}\} = \{z_i\} \text{ – вектор скрытых переменных}$$

Скрытые переменные отвечают темам, соответствующим слову в документе.

LDA является смешанной моделью, то есть использует выпуклую оболочку компонентных распределений для моделирования наблюдений[8]. Такие модели генерируют слова следующим образом.

$$p(w=t) = \sum_k p(w=t \mid z=k)\, p(z=k),$$

где $\sum_k p(z=k) = 1$ – коэффициенты смеси.

Однако LDA развивает идею дальше, делая распределение тем, зависимым от документа. Вывод модели сводится к нахождению набора распределений $p(t \mid z=k) = \vec{\phi}_k$ – слов для каждой темы, и $p(z \mid d=m) = \vec{\theta}_m$ – тем, для каждого документа. А оценки наборов параметров $\Phi = \{\vec{\phi}_k\}_{k=1}^{K}$ и $\Theta = \{\vec{\theta}_m\}_{m=1}^{M}$ служат латентно-семантическим представлением документов.

Для вывода этих параметров необходимо сначала найти выражения для скрытых переменных z_{mn} при данных наблюдениях, то есть

$$p(\vec{z} \mid \vec{w}) = \frac{p(\vec{z}, \vec{w})}{p(\vec{w})} = \frac{\prod_{i=1}^{W} p(z_i, w_i)}{\prod_{i=1}^{W} \sum_{k=1}^{K} p(z_i = k, w_i)}$$

Из-за сложности вычисления знаменателя точный вывод невозможен. Поэтому используются алгоритмы для приближенного вывода, например семплирование Гиббса.

Семплирование Гиббса

Алгоритм используется для оценки совместного распределения множества случайных величин, когда точный его вид определить не удается, однако условные вероятности для каждой отдельной переменной известны. Например, для некоторого совместного распределения $p(\vec{x})$ алгоритм работает следующим образом:

1. Выбрать переменную x_i из вектора $\vec{x}$
2. Сгенерировать значения этой переменной из условного распределения $p(x_i \mid x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n)$

Метод относится к MCMC – Markov Chain Monte Carlo методам. Построенная выборка позволяет оценить распределение $p(\vec{x})$, благодаря свойствам полученной цепи Маркова. Проблема возникает при оценке сходимости метода, однако при текстовой классификации зачастую просто проводят заранее заданное число итераций (порядка нескольких тысяч).

Глава 2. Частичное обучение на графе

2.1. Постановка задачи частичного обучения

Пусть дано множество объектов $\tilde{X} = \{x_1, \dots x_p, x_{p+1}, x_N\} \subset X$ и множество меток классов $Y = \{y_i\}_{i=1}^{K}$, для простоты будем говорить о классах $1, \dots, K$. Для p объектов известно, к какому классу они принадлежат, ставится задача определить, к какому классу принадлежат остальные $x_{p+1}, \dots x_N$. Также возможна постановка, где требуется найти классификатор $a(x): X \to Y$ (индуктивное обучение).

Как правило, общее число объектов сильно превосходит число p. Это может быть обосновано тем, что проведение экспертной оценки для большего числа объектов невозможно, но получение данных относительно просто. Чтобы непомеченные данные были полезны при обучении, полезно сделать некоторые предположения об их природе. Предположение гладкости – объекты в признаковом пространстве находящиеся ближе друг к другу, более вероятно находятся в одном классе. Предположение о кластеризации частично следует из предыдущего – объекты одного класса, как правило, образуют кластеры. Предположение о подпространстве – данные располагаются в подпространстве признакового пространства меньшей размерности. Для разных предположений применяются свои алгоритмы или вариации алгоритмов.

2.2. Обучение на графе

Если данные естественным образом связаны с каким-то графом (социальные сети, веб-страницы), то это знание также можно использовать в частичном обучении. Также граф можно построить, основываясь на отношении подобия. Обозначим за $W = \{w_{ij}\}$ матрицу подобия объектов.

Тогда, если объекты представлены нормализованными векторами действительных чисел, то можно представить её, используя функцию Гаусса.

$$w_{ij} = \exp(\frac{-\left\| x_i - x_j \right\|^2}{\gamma})$$

Или можно использовать метод k ближайших соседей:

$$w_{ij} = \begin{cases} 1, \text{ если } x_j \text{ является одним из k ближайших соседей } x_i \\ 0, \text{ иначе} \end{cases}$$

Матрицу подобия будем предполагать симметричной $W^T = W$. Из несимметричной матрицы, полученной методом ближайших соседей можно построить симметричную следующим образом:

$$W' = \frac{W + W^T}{2}$$

Обозначим также диагональную матрицу D, с элементом диагонали d_i.

$$d_i = d_{ii} = \sum_{j=1}^{N} w_{ij}$$

Определим матрицу Y размерности $N \times K$.

$$Y_{ik} = \begin{cases} 1, \text{ если известно, что } x_i \text{ принадлежит классу } k \\ 0, \text{ иначе} \end{cases}$$

Колонки $Y_{\cdot k}$ этой матрицы будет называть функцией меток. Также определим матрицу F той же размерности и будем называть её колонки классифицирующими функциями. Задача обучения на графе состоит в нахождении классифицирующих функций, наиболее близких к функциям меток, и при этом достаточно гладких относительно матрицы подобия.

Есть два широко используемых метода оптимизации для нахождения классифицирующей функции: основанный на стандартном лапласиане и нормализованном.

$$\min_F \left\{ \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} \left\| F_{i\cdot} - F_{j\cdot} \right\|^2 + \mu \sum_{i=1}^{N} d_i \left\| F_{i\cdot} - Y_{i\cdot} \right\|^2 \right\}$$

$$\min_F \left\{ \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} \left\| \frac{F_{i\cdot}}{\sqrt{d_{ii}}} - \frac{F_{j\cdot}}{\sqrt{d_{jj}}} \right\|^2 + \mu \sum_{i=1}^{N} \left\| F_{i\cdot} - Y_{i\cdot} \right\|^2 \right\}$$

Здесь регуляризационный параметр μ отражает выбор между гладкостью классификационной функции и её близостью к функции меток.

Обобщенная оптимизационная задача

В [9] приведена формулировка оптимизационной задачи для классификационной функции, объединяющая в себе обе предыдущие.

$$\min_F \left\{ \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} \left\| d_{ii}^{\sigma-1} F_{i\cdot} - d_{jj}^{\sigma-1} F_{j\cdot} \right\|^2 + \mu \sum_{i=1}^{N} d_{ii}^{2\sigma-1} \left\| F_{i\cdot} - Y_{i\cdot} \right\|^2 \right\}$$

Отмечается, что при $\sigma = 0$, получается наиболее устойчивое к изменению регуляризационного параметра решение.

Решение задачи находится аналитически и дается формулой

$$F_{\cdot k} = \frac{\mu}{2+\mu} \left(I - \frac{2}{2+\mu} D^{-\sigma} W D^{\sigma-1} \right)^{-1} Y_{\cdot k},$$

где $k = 1, \ldots K$. Если же решение, соответствующее $\sigma = 0$ записать в транспонированной форме, и обозначить $\alpha = \frac{2}{2+\mu}$. Получим выражение, соответствующее PageRank классификации, описанной в [10].

$$F_{\cdot k}^T = (1-\alpha) Y_{\cdot k}^T (I - \alpha D^{-1} W)^{-1}$$

Что позволяет интерпретировать решения, основанные на лапласиане, с точки зрения случайного блуждания по графу. Решение, основанное на стандартном лапласиане F_{ik} равно, с точностью до постоянного множителя, ожидаемому числу посещений, помеченных классом k вершин, если случайное блуждание начинается в вершине i. А решение, данное PageRank методом, с точностью до постоянного множителя соответствует числу ожидаемых посещений вершины i, если случайное блуждание началось с вершины, взятой из равномерного распределения по всем помеченным классом k вершинам.

Глава 3. Проведение эксперимента

Имея коллекцию текстовых документов $D = \{d_m\}_{m=1}^{M}$, связанных ссылками, то есть образующими граф $G = \{D, E\}$, где $E = \{(d, d')\}_{d,d' \in D}$, необходимо разбить множество документов на непересекающиеся группы, согласно приведенным темам. Предполагается, что структура графа поможет решить поставленную задачу.

Целью эксперимента является сравнение традиционных алгоритмов текстовой классификации с алгоритмом частичного обучения на графе. В качестве набора данных используется небольшой подграф страниц Википедии из 909 вершин. Все статьи в наборе относятся к категории «Математика» и разделены экспертами на три группы: «Прикладная математика», «Анализ» и «Дискретная математика». Предлагается провести классификацию, используя небольшой набор обучающих данных.

Статьи Википедии являются отличным примером данных, где естественно полученный граф может предоставить полезную дополнительную информацию о связях между документами. Граф строится по наличию ссылок на страницах, однако ориентированность графа отбрасывается. Можно сказать, что делается предположение о том, что наличие ссылки с первой страницы на вторую одинаково влияет на принадлежность обеих страниц к какому-либо классу.

Сравнение качества алгоритмов

Для сравнения качества будем применять простую оценку качества классификации: долю верно классифицированных документов.

Сначала попробуем классифицировать tf-idf вектора документов, не учитывая структуру графа.

Метод опорных векторов

В качестве обучающей выборке выберем случайным образом небольшую часть документов 15 штук. Обучив классификатор на каждом из 10 таких наборов, получаем результат, приведенный на Рисунке 1. Видно, что качество сильно зависит от набора, что говорит о недостаточной длине обучающей выборки, тем не менее классификатор сильно превосходит случайное гадание.

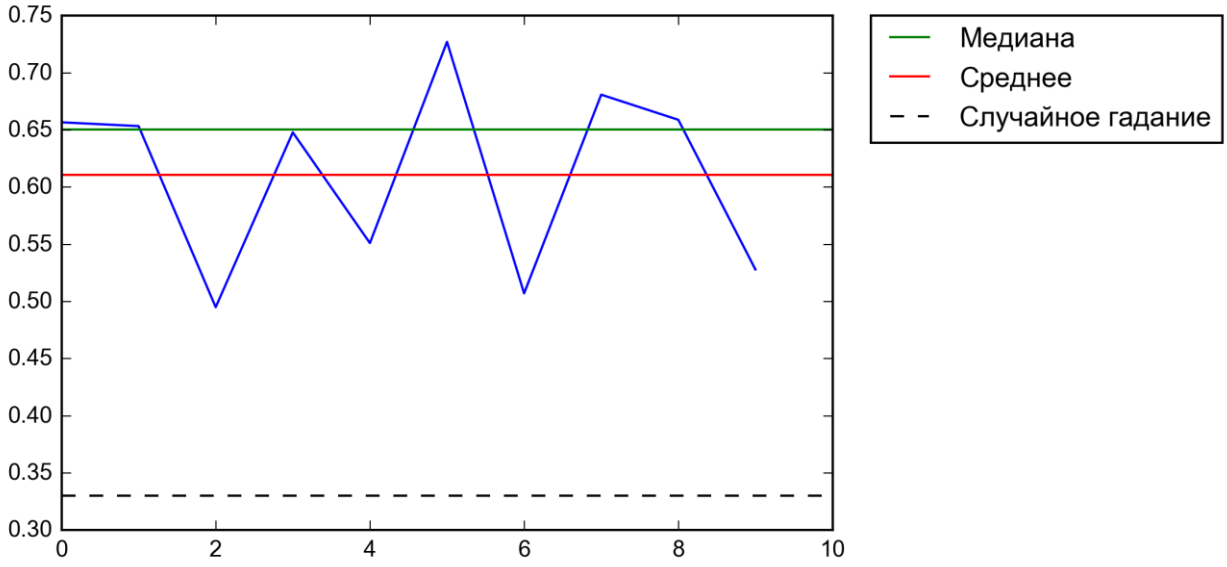


Рисунок 1: Доля верных ответов SVM

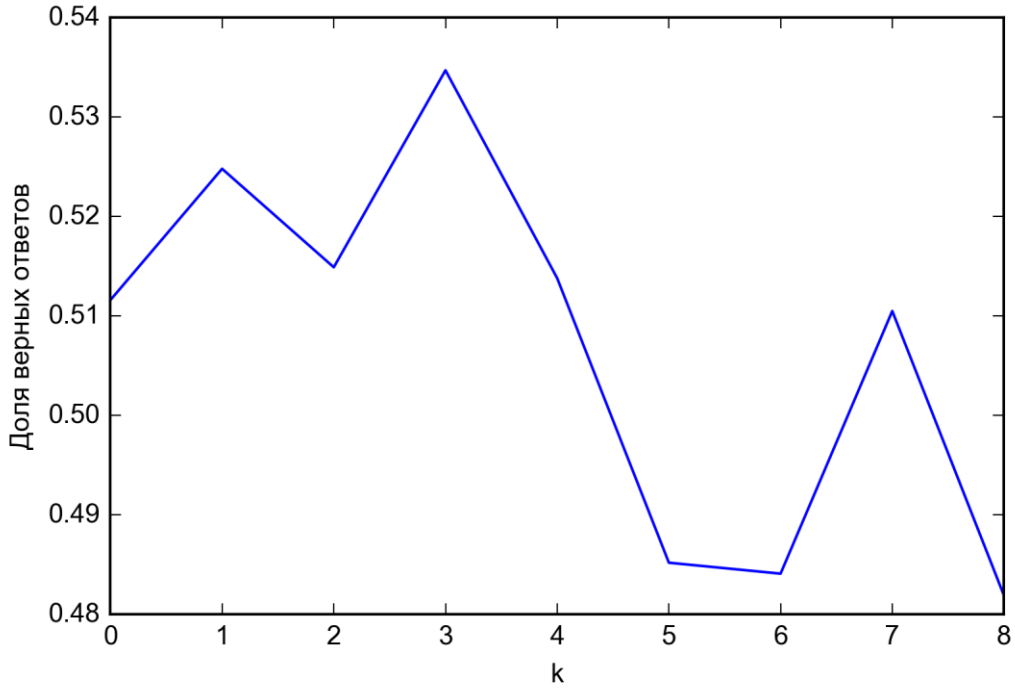
Попробуем также обучить метод опорных векторов на большем количестве документов, для этого просто объединим все 10 наборов. Как и следовало ожидать, доля верных ответов повышается и составляет около *0.834*.

Латентное размещение Дирихле и метод k ближайших соседей

LDA является базовым алгоритмом в анализе текстов. В своей постановке он решает задачу кластеризации. Но результат его работы, а именно, распределение каждого документа по темам, может быть использован, в качестве нового признакового пространства для документов. В нашем случае, однако, различия между классами, кажется, довольно

сложным определить, так как определяющим фактором являются скорее формулы и глубокие концепции.

Применяя LDA убеждаемся, что алгоритм не может уловить, нужные нам темы, поэтому точность остается в районе случайного гадания. Если же использовать его в качестве получения нового признакового пространства. А для результатов применить метод k ближайших соседей, получим немногим лучший результат.



Вероятно, проблема лежит в чрезмерной похожести заданных тем, и в данном случае формальный векторный подход лучше, чем семантический анализ. Другим фактором может являться недостаточное количество документов в коллекции для корректной работы LDA.

Теперь попробуем применить знания о связи между документами.

Частичное обучение на графе

Применим метод, полученный в [9]. Для этого необходимо найти матрицу подобия W. Зададим её сначала, как

$$w_{ij} = \begin{cases} 1, & \text{если есть ссылка между страницами} \\ 0, & \text{иначе} \end{cases}$$

Затем попробуем применить знания о самих текстах статей. Для этого посчитаем косинусную меру между векторами документов. Она определяется следующим образом

$$similarity(A,B) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

И будем использовать матрицу вида

$$w_{ij} = \begin{cases} similarity(i, j), & \text{если между i и j есть ссылка} \\ 0, & \text{иначе} \end{cases}$$

Результаты обучения на каждом из 10 наборов усредним, а также будем использовать разные значения регуляризационного параметра α.

Результаты представлены на Рисунке 2.

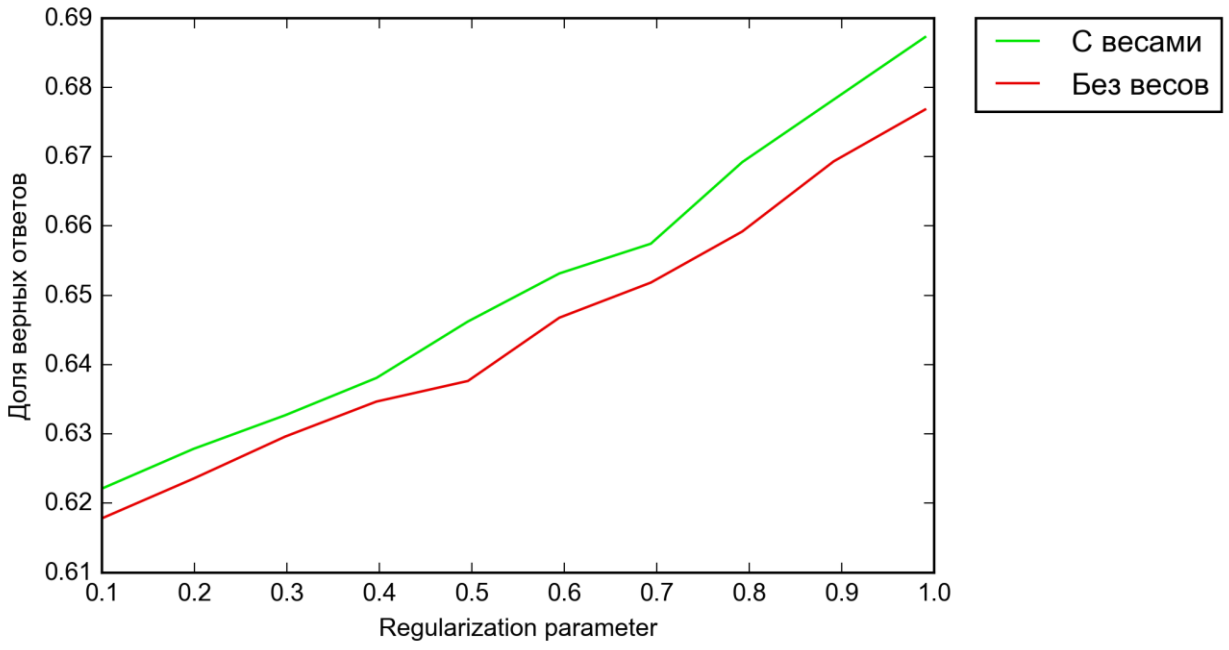


Рисунок 2: PageRank classification

Во-первых, замечаем, что введение весов положительно повлияло на качество, однако не существенно. Это свидетельствует о том, что информация о схожести документов не помогает различить между собой разные классы. Во-вторых, видим, что чем ближе регуляризационный параметр к 1, тем лучшее качество получаем. Это можно объяснить в терминах случайного блуждания, тогда $1-\alpha$ обозначает вероятность перейти с текущей страницы не по ссылке, а на любую другую случайным образом. Значит, выбирая параметр близким к 1, мы принуждаем случайное блуждание переходить только по ссылкам в документе. Учитывая структуру Википедии, вполне логично, что лишь переходя по ссылкам, мы можем обойти все страницы одного класса.

Выводы

В ходе эксперимента наилучший результат классификации показал метод опорных векторов, при обучении которого применялась выборка большего размера. На выборке меньшего размера лучший результат показал PageRank метод за счет применения непомеченных данных. Предположительно самый мощный метод анализа текстовых данных LDA, не проявил себя в связи со спецификой задачи.

Есть два пути дальнейшего исследования. Первый – попробовать объединить методы текстовой классификации и анализ ссылок. Такой метод описан, например, в [11]. Другой путь – это применение глубоких нейронных сетей. Они согласно исследованиям [12] позволяют находить абстракции более высокого уровня, такие как темы документа.

Заключение

Были рассмотрены различные традиционные методы классификации применительно к задаче классификации документов. Также был рассмотрен метод частичного обучения на графе. Проведен эксперимент с использованием выборки статей с сайта Википедия. В ходе эксперимента проведено сравнение качества работы алгоритмов обоих типов.

Список литературы.

1. Маннинг, К., Рагхаван, П., Шютце, Х. Введение в информационный поиск. М.: Изд. дом «Вильямс», 2011. 135 с.
2. MacQueen, J. Some methods for classification and analysis of multivariate observations // Proc. of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics, University of California Press, Berkeley, Calif., 1967. С. 281-297.
3. Arthur, D., Vassilvitskii S. How slow is the k-Means Method? // Proc. Symposium on Computational Geometry, С. 144-153.
4. Вапник, В. Н. Восстановление зависимостей по эмпирическим данным. — М.: Наука, 1979.
5. Joachims, T. Text Categorization with Support Vector Machines: Learning with many relevant features. // Machine Learning: ECML-98: 10th European Conference on Machine Learning Chemnitz, Germany, April 21--23, 1998 Proceedings.
6. Hofmann, T. Probabilistic Latent Semantic Analysis // Proc. 15th Conf. Uncertainty in Artificial Intelligence, pp. 289-296, 1999.
7. Blei, D., Ng, A., Jordan, M., Lafferty, J. Latent dirichlet allocation // Journal of Machine Learning Research, Vol. 3, 2003
8. Heinrich G., Parameter estimation for text analysis // Technical Report, 2004.
9. Avrachenkov, K., Gonçalves, P., Mishenin, A., Sokol, M. Generalized Optimization Framework for Graph-based Semi-supervised Learning // Computing Research Repository vol. abs/1110.4278, 2011
10. Avrachenkov, K., Dobrynin, V., Nemirovsky, D., Son Kim Pham, Smirnova, E. PageRank Based Clustering of Hypertext Document Collections // Proceedings of the 31-st annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR, 2008, С. 873-874.

11. Dietz, L., Bickel, S., Scheffer, T., Unsupervised Prediction of Citation influences // Proceedings of the 24-st International conference on machine learning, 2007.
12. Zhang, X., LeCun, Y., Text understanding from scratch. // Computing research repository vol. abs/1502.01710