МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

Кафедра дискретной математики и алгоритмики

САТАНЕВСКИЙ
Владислав Валерьевич

СЕНТИМЕНТ-АНАЛИЗ ТЕКСТОВ СОЦИАЛЬНЫХ СЕТЕЙ

Отчет по практике

Специальность 1-31 81 09 «Алгоритмы и системы обработки больших объемов информации»

Руководитель практики от кафедры
Соболевская Елена Павловна
доцент, кандидат физико-математических наук

Минск, 2017

Отчет по практике 38 с., 3 таблицы, 11 источников.

СЕНТИМЕНТ-АНАЛИЗ, СОЦИАЛЬНЫЕ СЕТИ, АНАЛИЗ ТЕКСТА, МАШИННОЕ ОБУЧЕНИЕ.

Объектом исследования является задача сентимент-анализа текстов из социальных сетей.

Целью работы является изучение различных методов решения задачи сентимент-анализа. Построение моделей сентимент-анализа для текстов из социальных сетей. Применение выбранных моделей к реальным данным. Оценка результатов.

В результате исследованы методы сентимент-анализа текстов социальных сетей. Реализованы изученные методы. Проведен сравнительный анализ методов сентимент-анализа.

Методы исследования: машинное обучение.

Область применения: сентимент-анализ.

Abstract

Practice report 38 p, 3 tables, 11 sources.

SENTIMENT ANALYSIS, SOCIAL NETWORKS, TEXT ANALYSIS, MACHINE LEARNING.

Object of research: social networks texts sentiment analysis.

Goal of research: study sentiment analysis methods, building sentiment analysis models for social networks texts, apply built models to the real data, and evaluate the results.

As the result of the current research sentiment analysis methods were studied. Studied methods were implemented. Comparative analysis of built methods was made.

Research methods: machine learning.

Applications: sentiment analysis.

Рэферат

Справаздача па практыцы 38 ст., 3 табліцы, 11 крыніц.

СЕНТЫМЕНТ-АНАЛІЗ, САЦЫЯЯЛЬНЫЯ СЕТКІ, АНАЛІЗ ТЭКСТУ, МАШЫННАЕ НАВУЧАННЕ.

Аб’ектам даследвання з’яўляецца праблема сентымент-аналізу тэкстау саціяльных сетак.

Мэтай работы з’яўляецца вывучэнне розных метадаў рашэння задачы сентымент-аналіза. Пабудова мадэлей сентымен аналізу дзеля тэкстаў с сацыяльных сетак. Скарыстанне выбраных мадэлей да рэальных даных. Аценка вынікаў.

У выніку даследаваны метады сентымент-аналізу текстаў сацыяльных сетак. Рэалізаваны даследаваныя метады. Праведзены параўнальны аналіз метадаў сентымент-аналіза.

Метады даследавання: машыннае навучанне.

Галіна скарыстання: сентымент-аналіз.

Оглавление

Введение

Ключевым моментом в ведении любого бизнеса является исследование целевой аудитории. Это позволяет лучше узнать потребителей, сфокусировать свои усилия на удовлетворение их нужд. При оценке отношения потребителей к сложным продуктам, таким как образовательные услуги, туристические услуги, автомобили и пр., зачастую более важным является не столько рациональное, сколько эмоциональное восприятие товара. В качестве примера можно привести такие категории, как "престиж вуза" или "элитарность образования", которые в большей степени отражают субъективную (эмоциональную) составляющую оценки, нежели объективную (рациональную).

Современным инструментом оценки эмоционального восприятия продукта является сентимент-анализ. Основной задачей данного подхода является определение субъективного восприятия продукта на основе семантического разбора текста. Конечные методики, используемые в рамках сентимент-анализа, могут варироваться от сравнительно простого словарного разбора предложений до сложных эвристических алгоритмов.

Развитие глобальной сети Интернет и сервисов, которые в ней предоставляются, дает неограниченные возможности для сбора маркетинговых данных о выбранной предметной области. В частности, микроблоги - сервисы, позволяющие публиковать в сети короткие сообщения с помощью как стационарных, так и мобильных устройств, являются наиболее популярным способом агрегирования текстовой информации.

В данной работе рассматриваются методы сентимент-анализа текстов и их применимость для анализа текстов социальных сетей.

ГЛАВА 1. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1.1 Постановка задачи

В данной работе рассматривается задача сентимент-анализа. Под сентимент-анализом понимается процесс определения и классификации мнений, выраженных в тексте.

В данной работе мы рассматриваем случай, когда тексты были написаны пользователями в социальных сетях, на примере социальной сети Twitter.

1.2 Актуальность задачи

Задача сентимент-анализа известна уже достаточно давно и хорошо изучена. Есть множество различных алгоритмов её решения, построенные на различных принципах и использующие для обучения различные наборы данных. Однако, большинство алгоритмов строились таким образом, чтобы определять эмоциональную окрашенность текстов, написанных на достаточно чистом языке, в котором относительно мало опечаток, грамматических и пунктуационных ошибок, сленга.

Например, алгоритмы, предназначенные для распознавания эмоционального окраса отзывов на товары, зачастую полагаются на то, что текст написан относительно хорошим языком, ведь зачастую пользователи стараются максимально понятно и развернуто объяснить преимущества и недостатки товаров, допускают мало ошибок. К сожалению, тексты социальных сетей не обладают такими приятными свойствами: часто пользователи быстро пишут относительно короткие сообщения и не сильно обеспокоены качеством языка.

Однако, таких сообщений заметно больше, чем отзывов на товары, при этом пользователи пишут их более открыто и честно. В социальных сетях меньше ангажированных сообщения, а также содержат мнения на многие темы, для выражения мнений по которым нет специализированных ресурсов.

Именно поэтому анализ данных социальных сетей представляет сильный интерес. Например, продавцам товаров интересно, что думают об их продуктах, о чем отзываются хорошо, а о чем плохо. Политикам интересна реакция людей на их решения или выступления.

1.3 Основные известные подходы к решению задачи сентимент-анализа

1.3.1 Подходы на основе словарей

Данный класс методов основан на построении окраски текста на основе заранее составленных словарей, состоящих из слов, выражающих мнение, а также их окраски (например “хороший”, “крутой”, “ужасный”). Также, каждому слову может быть сопоставлена сила окраски (например, у слова “великолепный” более сильный положительный эмоциональный окрас, чем у “неплохой”). Далее, решение об эмоциональной окраске текста принимается на основе вхождения слов из словаря в текст. Например, если среди эмоционально окрашенных слов текста больше положительных, будем считать текст положительно окрашенным, а если наоборот, отрицательно окрашенным, и нейтральным, если количество положительных и отрицательных слов одинаково.

В подходах такого типа важно, чтобы составленный словарь был высокого качества, чего трудно добиться, составляя его вручную, поэтому это часто делают автоматическим способом. Достаточно популярными являются следующие методы:

1) На основе готовых данных о соотношениях слов. Например, взяв слово “хороший” и посмотрев в готовом корпусе (например, в специальном тезаурусе или WordNet) его синонимы, можно получить положительно окрашенные слова. Так можно делать несколько итераций с несколькими начальными словами, на каждой последующей итерации пополняя текущий словарь найденными синонимами. Видно, что метод достаточно точный, но требует наличие хорошего тезауруса, которого может не быть для искомого языка или лексики (например, для лексики из социальных сетей).
2) На основе конструкций вида “A и B”, “A, но B” и так далее. По этим конструкциям видно, что если известна тональность одного из слов, можно достаточно точно определить тональность и второго слова. Поэтому, имея тональность некоторых слов, можно извлечь из большого корпуса текстов конструкции заданного вида и найти тональность уже для новых слов. Данную операцию можно провести несколько раз. Недостаток метода в том, что если на каком-то шаге появятся ошибки, на последующих шагах их количество будет сильно расти.

1.3.2 Подходы на основе обучения с учителем

Понятно, что классификация тональности текста можем быть рассмотрена как задача обучения с учителем с несколькими классами (например, “положительный”, “отрицательный” и “нейтральный”). В различных исследованиях в качестве данных для обучения и тестирования алгоритмов берут отзывы на различные товары, так как собирать такие данные достаточно просто, при этом часто такие отзывы содержат пользовательскую оценку, по которой можно понять полярность написанного отзыва. Например, если пользователь выставляет количество звезд от 1 до 5, можно считать, что отзывы с 4-5 звездами являются положительными, с 1-2 звездами являются отрицательными, а с 3 звездами – нейтральными. Далее, в такой задаче может быть применен любой алгоритм для классификации текста (например, наивный байесовский классификатор или машины опорных векторов). Пэнг в своей работе [4] использовал этот подход для классификации обзоров фильмов на 2 класса, положительного и отрицательного. Он показал, что использование юниграмм (мешка слов) в качестве признаков работает хорошо при использовании наивного байесовского классификатора или машин опорных векторов. В последующих исследованиях использовали больше различных признаков и алгоритмов обучения с учителем. Следующие признаки были использованы:

1) Термы и их частоты: отдельные слова и комбинации (n-граммы) и их частоты. Иногда рассматриваются позиции слов. Также, схема TF-IDF из информационного поиска может быть использована для взвешивания различных признаков.
2) Части речи: Многими исследователями было показано, что прилагательные являются важным мнений. Поэтому, прилагательные рассматривались как отдельные признаки.
3) Слова и фразы, означающие мнение. Это такие слова, которые обычно используют для выражения положительного или отрицательного мнения. Например, “красивый”, “замечательный”, “хороший” и “великолепный” являются положительными словами, означающими мнение. При этом “плохой”, “убогий” и “ужасный” являются негативными словами, выражающими мнение. Кроме прилагательных и наречий, в качестве таких слов могут выступать даже существительные и глаголы (например “мусор”, “барахло”, “люблю”, “ненавижу”). Кроме одиночных слов, также могут быть различные фразы (например, “обещать золотые горы”).

Понятно, что выделение таких слов и фраз способствует улучшению анализа.

4) Отрицание. Понятно, что такие слова важны, так как из присутствие часто меняет ориентацию мнения. Например “мне не нравится эта камера” – негативный отзыв несмотря на то, что содержит “нравится”. Однако, не всегда упоминание слов отрицания на самом деле является отрицанием. Например, “не только красивый, но и практичный” содержит “не”, не меняет полярность текста.
5) Синтаксические зависимости. Это признаки, построенные на основе полученного из текста дерева разбора.
6) Семантическое представление слов. Представляет собой отображение слов в вектор из многомерного пространства таким образом

1.3.2 Подходы на основе обучения без учителя.

Нетрудно понять, что слова и фразы, выражающие мнение это одни из самых важных индикаторов для классификации полярности. Поэтому, использование обучения без учителя на основе таких слов и фраз может быть вполне естественным. Например, метод в [5] использует известные слова, выражающие мнение, для классификации, в то время как [6] определяет некоторые фразы, которые вероятно выражают мнение. Авторы [6] предлагают алгоритм, состоящий из трех шагов:

1) Извлекаются фразы, содержащие прилагательные и наречия, так как эти части речи являются хорошими индикаторами мнения. Однако, несмотря на то, что прилагательное может говорить о мнении, его самого может быть недостаточно для определения полярности. Например, слово “непредсказуемые” может иметь разную полярность в зависимости от контекста. Например, в обзоре автомобиля “непредсказуемое управление” имеет отрицательную полярность, в то время как в обзоре фильма “непредсказуемый сюжет” имеет положительную полярность. Поэтому, алгоритм извлекает слово вместе с контекстом. Это делается в случаях, когда части речи фразы удовлетворяют специальному паттерну.
2) Оценивается ориентация фразы, используя точечную взаимную информацию, посчитанную по формуле: $PMI(term_1, term_2) = \log_2 \frac{P(term_1 \wedge term_2)}{P(term_1) * P(term_2)}$. Здесь $P(term_1 \wedge term_2)$ – вероятность того, что оба терма встречаются одновременно, а $P(term)$ – вероятность того, что встречается один терм. Отношение $P(term_1 \wedge term_2)$ и $P(term_1) * P(term_2)$ показывает степень

статистической зависимости термов. Логарифм этого отношения это объем информации, который мы получаем о наличие первого слова, если мы наблюдаем второе и наоборот. Полярность фразы вычисляется на основе его ассоциации с положительным словом “excellent” и отрицательным словом “poor” по формуле $SO(phrase) = PMI(phrase, \text{"excellent"}) - PMI(phrase, \text{"poor"})$. Для оценки вероятностей, необходимых для подсчета PMI используя количество найденных поисковой системой документов. В качестве поисковой системы авторы использовали AltaVista, так как она поддерживает оператор NEAR в поисковых запросах.

3) Для данного отзыва алгоритм вычисляет среднее значение SO среди всех фраз в отзыве и классифицирует его как положительное, если среднее SO положительное и отрицательное в обратном случае. Точность алгоритма зависит от доменной области и варьируется от 66% для отзывов на фильмы до 84% для отзывов на автомобили.

1.3 Основные методы отбора признаков

Отбор признаков является одним из важных этапов решения задач машинного обучения. Поэтому, придумано достаточно много методов отбора признаков. Далее, приведем некоторые основные подходы, использующиеся при отборе признаков.

1.3.1 Подход, основанный на статистических методах

Суть таких методов заключается в статистической оценке значимости признаков. Как правило, признаки рассматриваются по отдельности и для каждого признака независимо строится мера его значимости (например, с помощью теста хи-квадрат независимости признака и скрытой переменной). Далее, выбираются признаки, мера значимости которых выше некоторого порога, который зачастую является параметром алгоритма. Преимуществом такого подхода является его относительная простота и зачастую небольшая вычислительная сложность. Однако основным недостатком является то, что мы перебираем каждый признак по отдельности и не учитываем признаки в совокупности.

1.3.2 Определение значимости признаков исходя из параметров обученной модели

Суть таких методов заключается в том, что параметры обученных моделей машинного обучения могут быть использованы для определения значимости

признаков. Например, в логистической регрессии обученный вектор параметров è уже даёт веса признаков по отношению к скрытой переменной. Например, высокое положительное или низкое отрицательное значение è$_i$ говорит о том, что значение признака x_i сильно влияет на предсказание модели, а, следовательно, значимо для неё. В методах, основанных на деревьях решений, ясно, что если признак, встречающийся в дереве, более значим, чем признак, в дереве не встречающийся. На основе этой информации, а также информации о том, как именно разделяет выборку значение признака, можно оценить его значимость для модели. В отличие от предыдущего подхода, признаки исследуются уже в совокупности (ведь именно совокупность признаков влияет на модель). Однако недостатком является более сложная интерпретируемость результатов, а также их зависимость от конкретной модели.

1.3.3 Методы отбора признаков, встраиваемые в модели

Это методы, которые устроены таким образом, что незначимые признаки исключаются из модели и реально ей не используются. Например, таким свойством обладает регуляризация L1 в линейных моделях. Плюсом является удобство, ведь модель “самостоятельно определяет”, какие признаки ей нужны, без отдельной стадии обработки данных. Недостатком модели, как и в предыдущем случае, является зависимость результатов от используемого алгоритма машинного обучения. Также, не все модели позволяют встраивание в них отбора признаков.

1.3.4 Определение множества значимых признаков на основе качества модели, использующей множество признаков

Суть таких методов заключается в том, что мы можем перебирать различные подмножества признаков (конкретный способ перебора и определяет алгоритм), и для каждого перебираемого подмножества тестировать модель, обученную на выбранном подмножестве признаков (это можно делать посредством скользящего контроля). Метрикой качества при этом может выступать функция, зависящая от качества модели и отобранных признаков (в более простом случае, количества признаков). В качестве результата выступает множество признаков, на котором значение метрики качества максимально. Преимуществом метода является применимость к любым моделям, а также тот факт, что правильная метрика качества отражает результат, который мы в итоге ожидаем от модели (высокая точность и, например, небольшое число признаков). Недостатком является вычислительная сложность (необходимо много раз обучать модель), а также невозможность перебирать все варианты (так как их экспоненциальное число от

количества признаков), что вынуждает придумывать нетривиальные алгоритмы перебора множества признаков.

1.7 Оценивание качества алгоритма методом скользящего контроля

Для оценки качества алгоритмов машинного обучения наиболее часто используются методы скользящего контроля. Остановимся на них поподробнее. Последующие материалы использованы из описания скользящего контроля в статье [13].

1.7.1 Общее описание

Скользящий контроль или кросс-валидация (cross-validation, CV) — процедура эмпирического оценивания обобщающей способности алгоритмов.

Фиксируется некоторое множество разбиений исходной выборки на две подвыборки: *обучающую* и *контрольную*. Для каждого разбиения выполняется настройка алгоритма по обучающей подвыборке, затем оценивается его средняя ошибка на объектах контрольной подвыборки. *Оценкой скользящего контроля* называется средняя по всем разбиениям величина ошибки на контрольных подвыборках.

Если выборка независима, то средняя ошибка *скользящего контроля* даёт несмещённую оценку вероятности ошибки. Это выгодно отличает её от средней ошибки на обучающей выборке, которая может оказаться смещённой (оптимистически заниженной) оценкой вероятности ошибки, что связано с явлением переобучения.

Скользящий контроль является стандартной методикой тестирования и сравнения алгоритмов классификации, регрессии и прогнозирования.

Рассматривается задача обучения с учителем.

Пусть X — множество описаний объектов, Y — множество допустимых ответов.

Задана конечная выборка прецедентов $X^L = (x_i, y_i)_{i=1}^L \subset X \times Y$.

Задан алгоритм обучения — отображение ì, которое произвольной конечной выборке прецедентов X^m ставит в соответствие функцию (алгоритм) $a: X \to Y$.

Качество алгоритма a оценивается по произвольной выборке прецедентов X^m с помощью *функционала качества* $Q(a, X^m)$. Для процедуры

скользящего контроля не важно, как именно вычисляется этот функционал. Как правило, он аддитивен по объектам выборки: $Q(a, X^m) = \frac{1}{m}\sum_{x_i \in X^m} L(a(x_i), y_i)$

где $L(a(x_i), y_i)$ — неотрицательная функция потерь, возвращающая величину ошибки ответа алгоритма $a(x_i)$ при правильном ответе y_i.

1.7.2 Процедура скользящего контроля

Выборка X^L разбивается N различными способами на две непересекающиеся подвыборки: $X^L = X_n^m \cup X_n^k$, где X_n^m — обучающая подвыборка длины m, X_n^k — контрольная подвыборка длины $k = L - m$, $n = 1, \dots, N$ — номер разбиения.

Для каждого разбиения n строится алгоритм $a_n = a(X_n^m)$ и вычисляется значение функционала качества $Q_n = Q(a_n, X_n^k)$. Среднее арифметическое значений Q по всем разбиениям называется *оценкой скользящего контроля*:

$$CV(a, X^L) = \frac{1}{N}\sum_{n=1}^{N} Q(a(X_n^m), X_n^k)$$

Различные варианты скользящего контроля отличаются видами функционала качества и способами разбиения выборки.

1.7.3 Доверительное оценивание

Кроме среднего значения качества на контроле строят также доверительные интервалы.

Непараметрическая оценка доверительного интервала. Строится вариационный ряд значений $Q_n = Q(a_n, X_n^k), n = 1, \dots, N$:

$$Q^{(1)} \le Q^{(2)} \le \dots \le Q^{(n)}$$

*Утверждение 1***.** Если разбиения осуществлялись случайно, независимо и равновероятно, то значение случайной величины $Q(a(X^m), X^k)$ не превосходит $Q^{(N-t+1)}$ с вероятностью $p = \frac{t}{N+1}$.

Следствие 1. Значение случайной величины $Q(a(X^m), X^k)$ не превосходит $Q^{(N)}$ с вероятностью $p = 1/(N+1)$. В частности, для получения верхней оценки с надёжностью 95% достаточно взять $N = 20$ разбиений.

Утверждение 2. Если разбиения осуществлялись случайно, независимо и равновероятно, то с вероятностью $p = 2t/(N+1)$ значение случайной

величины $Q(a(X^m), X^k)$ не выходит за границы доверительного интервала $\left[Q^{(t)}, Q^{N-t+1}\right]$.

Следствие 2. Значение случайной величины $Q(a(X^m), X^k)$ не выходит за границы вариационного ряда $\left[Q^{(1)}, Q^{(N)}\right]$ с вероятностью $p = \frac{2}{N+1}$.

В частности, для получения двусторонней оценки с надёжностью 95% достаточно взять $N = 40$ разбиений.

Параметрические оценки доверительного интервала основаны на априорном предположении о виде распределения случайной величины $Q(a(X^m), X^k)$. Если априорные предположения не выполняются, доверительный интервал может оказаться сильно смещённым. В частности, если предположения о нормальности распределения не выполнены, то нельзя пользоваться стандартным «правилом двух сигм» или «трёх сигм». Джон Лангфорд в своей диссертации указывает на распространённую ошибку, когда правило двух сигм применяется к функционалу частоты ошибок, имеющему на самом деле биномиальное распределение. Однако биномиальным распределением в общем случае тоже пользоваться нельзя, поскольку в результате обучения по случайным подвыборкам X^m вероятность ошибки алгоритма $a(X^m)$ оказывается случайной величиной. Следовательно, случайная величина $Q(a(X^m), X^k)$ описывается не биномиальным распределением, а (неизвестной) смесью биномиальных распределений. Аппроксимация смеси биномиальным распределением может приводить к ошибочному сужению доверительного интервала. Приведённые выше непараметрические оценки лишены этого недостатка.

1.7.4 Стратификация

Стратификация выборки — это способ уменьшить разброс (дисперсию) оценок скользящего контроля, в результате чего получаются более узкие доверительные интервалы и более точные (tight) верхние оценки.

Стратификация заключается в том, чтобы заранее поделить выборку на части (страты), и при разбиении на обучение длины m и контроль длины k гарантировать, что каждая страта будет поделена между обучением и контролем в той же пропорции $m{:}\, k$.

Стратификация классов в задачах классификации означает, что каждый класс делится между обучением и контролем в пропорции $m{:}\, k$.

Стратификация по вещественному признаку. Объекты выборки сортируются согласно некоторому критерию, например, по возрастанию одного из признаков. Затем выборка разбивается на k последовательных страт одинаковой (с точностью до 1) длины. При формировании контрольных выборок из каждой страты выбирается по одному объекту, либо с заданным порядковым номером внутри страты, либо случайным образом.

1.7.5 Разновидности скользящего контроля

Возможны различные варианты скользящего контроля, отличающиеся способами разбиения выборки.

Полный скользящий контроль (complete CV).

Оценка скользящего контроля строится по всем $N = C_L^k$ разбиениям. В зависимости от k (длины обучающей выборки) различают:

- Частный случай при $k = 1$ — контроль по отдельным объектам (leave-one-out CV);

Было показано, что контроль по отдельным объектом является асимптотически оптимальным при некоторых условиях..

- Общий случай при $k > 2$. Здесь число разбиений $N = C_L^k$ становится слишком большим даже при сравнительно малых значениях k, что затрудняет практическое применение данного метода. Для этого случая *полный скользящий контроль* используется либо в теоретических исследованиях, либо в тех редких ситуациях, когда для него удаётся вывести эффективную вычислительную формулу. Например, такая формула известна для метода *k* ближайших соседей, что позволяет эффективно выбирать параметр k. На практике чаще применяются другие разновидности *скользящего контроля*.

Случайные разбиения

Разбиения $n = 1, \dots, N$ выбираются случайно, независимо и равновероятно из множества всех C_L^k разбиений. Именно для этого случая справедливы приведённые выше оценки доверительных интервалов. На практике эти оценки, как правило, без изменений переносится и на другие способы разбиения выборки.

Контроль на отложенных данных (hold-out CV)

Оценка скользящего контроля строится по одному случайному разбиению, $N = 1$.

Этот способ имеет существенные недостатки:

1. Приходится слишком много объектов оставлять в контрольной подвыборке. Уменьшение длины обучающей подвыборки приводит к смещённой (пессимистически завышенной) оценке вероятности ошибки.

2. Оценка существенно зависит от разбиения, тогда как желательно, чтобы она характеризовала только алгоритм обучения.

3. Оценка имеет высокую дисперсию, которая может быть уменьшена путём усреднения по разбиениям.

Следует различать скользящий контроль по отложенным данным и контроль по тестовой выборке. Если во втором случае оценивается вероятность ошибки для классификатора, построенного по обучающей подвыборке, то в первом случае - для классификатора, построенного по полной выборке (то есть доля ошибок вычисляется не для того классификатора, который выдается в качестве результата решения задачи).

Контроль по отдельным объектам (leave-one-out CV)

Является частным случаем полного скользящего контроля при $k = 1$, соотвественно, $N = L$. Это, пожалуй, самый распространённый вариант скользящего контроля.

Преимущества LOO в том, что каждый объект ровно один раз участвует в контроле, а длина обучающих подвыборок лишь на единицу меньше длины полной выборки.

Недостатком LOO является большая ресурсоёмкость, так как обучаться приходится L раз. Некоторые методы обучения позволяют достаточно быстро перенастраивать внутренние параметры алгоритма при замене одного обучающего объекта другим. В этих случаях вычисление LOO удаётся заметно ускорить.

Контроль по q блокам (q-fold CV)

Выборка случайным образом разбивается на q непересекающихся блоков одинаковой (или почти одинаковой) длины $k_1, \dots k_q$: $X^L = X_1^{k_1} \cup \dots \cup X_q^{k_q}$

$k_1 + \dots + k_q = L$. Каждый блок по очереди становится контрольной подвыборкой, при этом обучение производится по остальным $q - 1$ блокам. Критерий определяется как средняя ошибка на контрольной подвыборке: $CV(a, X^L) = \frac{1}{q}\sum_{n=1}^{q} Q(a(X^L \backslash X_n^{k_n}), X_n^{k_n})$

Это компромисс между LOO, hold-out и случайными разбиениями. С одной стороны, обучение производится только q раз вместо L. С другой стороны, длина обучающих подвыборок, равная $L(q-1)/N$ с точностью до округления, не сильно отличается от длины полной выборки L. Обычно выборку разбивают случайным образом на 10 или 20 блоков.

Контроль по r × q блокам (r × q-fold CV)

Контроль по q блокам (q-fold CV) повторяется r раз. Каждый раз выборка случайным образом разбивается на q непересекающихся блоков. Этот способ наследует все преимущества q-fold CV, при этом появляется дополнительная возможность увеличивать число разбиений.

Данный вариант скользящего контроля, со стратификацией классов, является стандартной методикой тестирования и сравнения алгоритмов классификации. В частности, он применяется в системах WEKA и «Полигон алгоритмов».

1.8 Метрики качества в задаче классификации

Так как в результате мы будем строить модель, позволяющую для нового объекта предсказывать устойчивость к препарату, то есть, решать задачу классификации, а при тестировании алгоритма необходимо выбрать метрику качества, рассмотрим основные метрики, используемые в задаче классификации. Рассмотренное ниже описание составлено с помощью статьи [14].

1.8.1 Правильность

Наверно, самая простая метрика, которая определяется как доля объектов, которые алгоритм классифицировал верно, к числу всех объектов.

$$Accuracy = \frac{P}{N}$$

Где P – число верно классифицированных объектов, N – общее число объектов.

1.8.2 Точность и полнота

Точность (precision) и полнота (recall) являются метриками которые используются при оценке большей части алгоритмов извлечения информации. Иногда они используются сами по себе, иногда в качестве базиса для производных метрик, таких как F-мера или $R - Precision$. Суть точности и полноты очень проста.

Точность системы в пределах класса – это доля документов действительно принадлежащих данному классу относительно всех документов которые система отнесла к этому классу. Полнота системы – это доля найденных классификатором документов принадлежащих классу относительно всех документов этого класса в тестовой выборке.

Эти значения легко рассчитать на основании таблицы контингентности, которая составляется для каждого класса отдельно. Таблица контингентности представлена таблицей 1.

		Экспертная оценка	
		Положительная	Отрицательная
Оценка система	Положительная	TP	FP
	Отрицательная	FN	TN

Таблица 1 - таблица контингентности

В таблице содержится информация сколько раз система приняла верное и сколько раз неверное решение по документам заданного класса. А именно:

- TP — истино-положительное решение;
- TN — истино-отрицательное решение;
- FP — ложно-положительное решение;
- FN — ложно-отрицательное решение.

Тогда, точность и полнота определяются следующим образом:

$$Precision = \frac{TP}{TP + FP},$$
$$Recall = \frac{TP}{TP + FN}$$

1.8.3 Матрица неточностей

На практике значения точности и полноты гораздо более удобней рассчитывать с использованием матрицы неточностей (confusion matrix). В случае если количество классов относительно невелико (не более 100-150 классов), этот подход позволяет довольно наглядно представить результаты работы классификатора.

Матрица неточностей – это матрица размера $N \times N$, где N — это количество классов. Столбцы этой матрицы резервируются за экспертными решениями, а строки за решениями классификатора. Когда мы классифицируем документ из тестовой выборки мы инкрементируем число стоящее на пересечении строки класса который вернул классификатор и столбца класса к которому действительно относится документ.

1.8.4 F-мера

Понятно что чем выше точность и полнота, тем лучше. Но в реальной жизни максимальная точность и полнота не достижимы одновременно и приходится искать некий баланс. Поэтому, хотелось бы иметь некую метрику которая объединяла бы в себе информацию о точности и полноте нашего алгоритма. В этом случае нам будет проще принимать решение о том какую реализацию запускать в production (у кого больше тот и круче). Именно такой метрикой является F-мера.

F-мера представляет собой гармоническое среднее между точностью и полнотой. Она стремится к нулю, если точность или полнота стремится к нулю.

$$F = 2\frac{Precision \times Recall}{Precision + Recall}$$

Данная формула придает одинаковый вес точности и полноте, поэтому F-мера будет падать одинаково при уменьшении и точности и полноты. Можно рассчитать F-меру придав различный вес точности и полноте, если вы осознанно отдаете приоритет одной из этих метрик при разработке алгоритма.

$$F = (\beta^2 + 1)\frac{Precision \times Recall}{\beta^2 Precision + Recall}$$

где в принимает значения в диапазоне $0 < в < 1$ если вы хотите отдать приоритет точности, а при $в > 1$ приоритет отдается полноте. При $в = 1$ формула сводится к предыдущей и вы получаете сбалансированную F-меру (также ее называют F_1).

F-мера является хорошим кандидатом на формальную метрику оценки качества классификатора. Она сводит к одному числу две других основополагающих метрики: точность и полноту.

1.8 Эффект переобучения

К сожалению, в имеющемся в нашем распоряжении наборе данных (описанном в разделе 4.1), лишь небольшое число объектов. В таком случае велик риск ситуации, называемой переобучением. Поэтому, рассмотрим эффект переобучения более подробно. Приведенное ниже описание выполнено с помощью статьи [15].

1.8.1 Понятие переобучения

Переобучение (*переподгонка*, *пере-* в значении «слишком», англ. *overfitting*) в машинном обучении и статистике — явление, когда построенная модель хорошо объясняет примеры из обучающей выборки, но относительно плохо работает на примерах, не участвовавших в обучении (на примерах из тестовой выборки).

Это связано с тем, что при построении модели («в процессе обучения») в обучающей выборке обнаруживаются некоторые случайные закономерности, которые отсутствуют в генеральной совокупности.

Даже тогда, когда обученная модель не имеет чрезмерного количества параметров, можно ожидать, что эффективность её на новых данных будет ниже, чем на данных, использовавшихся для обучения. В частности, значение коэффициента детерминации будет сокращаться по сравнению с исходными данными обучения.

1.8.2 Способы борьбы с переобучением

Способы борьбы с переобучением зависят от метода моделирования и способа построения модели. Например, если строится дерево принятия решений, то можно обрезать некоторые его ветки в процессе построения.

Для того чтобы избежать чрезмерной подгонки, необходимо использовать дополнительные методы, например:

- перекрёстная проверка,
- регуляризация ,
- ранняя остановка,
- вербализация нейронных сетей,
- априорная вероятность,
- байесовское сравнение моделей (англ. *bayesian model comparison*),

которые могут указать, когда дальнейшее обучение больше не ведёт к улучшению оценок параметров. В основе этих методов лежит явное ограничение на сложность моделей, или проверка способности модели к обобщению путём оценки её эффективности на множестве данных, не использовавшихся для обучения и считающихся приближением к реальным данным, к которым модель будет применяться.

1.9 Оптимизация гиперпараметров

Зачастую, алгоритм машинного обучения имеет набор параметров (который называется гиперпараметрами), от правильности задания которого зависит качество результирующей модели. Поэтому, для того, чтобы оценить, какой метод машинного обучения лучше, гиперпараметры подбираются достаточно близко к оптимальным (в той мере, в которой это удалось), и далее сравнивают алгоритмы, инициализированные найденными наилучшими параметрами. Поэтому, для того, чтобы сравнить алгоритмы верно, необходимо уметь правильно выбирать гиперпараметры. Поэтому, рассмотрим основные методы, позволяющие выбрать гиперпараметры близко к оптимальным значениям. На самом деле, задача поиска гиперпараметров, близких к оптимальным можно рассматривать как задачу оптимизации. Однако, к сожалению, большинство оптимизационных методов накладывают различные (причем достаточно жесткие) ограничения, которые не выполняются в случае оптимизации гиперпараметров. Поэтому, методы оптимизации гипермараметров рассматривают целевую функцию как черный ящик, то есть, не накладывают на нее серьезных ограничений (как, например, выпуклость, унимодулярность, гладкость). К сожалению, это существенно сказывается на теоретических и практических свойствах таких алгоритмов. Самое важное, что стоит отметить, так это то, что такие алгоритмы не гарантируют, что найденный результат будет оптимальным с некоторой точностью. Однако на практике оказывается, что результат хороших алгоритмов оказывается приемлемым и в большинстве случаев не уступает по качеству ручному поиску параметров, несмотря на то, что ручной поиск обладает тем важным преимуществом, что при

подборе параметров исследователь уже имеет опыт решения различных задач, поэтому имеет примерное представление о том, как подобрать параметры правильно.

1.9.1 Поиск по сетке

Наиболее традиционным способом оптимизации гиперпараметров довольно долгое время был и все еще является поиск по сетке. Поиск по сетке представляет собой исчерпывающий поиск среди параметров, заданных на некоторой сетке (вообще говоря, не равномерной). Среди всех рассматриваемых значений гиперпараметров выбирается то значение, на котором метрика качества максимальна. Метрика качества измеряется с помощью подходящего алгоритма тестирования модели (например, с помощью скользящего контроля). Так как, вообще говоря, полный исчерпывающий поиск всех возможных значений параметров выполнить не всегда возможно (например, потому что возможных значений гиперпараметров бесконечное число), сетку, на которой параметры будут перебираться, необходимо задать заранее.

Например, рассмотрим задачу подбора гиперпараметров градиентного бустинга над решающими деревьями. Одними из основных параметров являются скорость обучения ($learning\_rate$) и количество деревьев ($n\_estimators$). Ограничим возможные значения параметров следующей сеткой:

$$learning\_rate \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$$

$$n\_estimators \in \{10, 20, 50, 100, 200, 500, 1000\}$$

Всего в нашей сетке будет 70 различных вариантов гиперпараметров. Далее, поиск по сетке представляет собой перебор всех этих 70 параметров с последующей оценкой каждого из них (например, с помощью скользящего контроля).

1.9.2 Случайный поиск

Случайный поиск, как и поиск по сетке, представляет собой перебор некоторого множества вариантов гиперпараметров, только, в отличие от поиска по сетке, эти варианты генерируются из некоторого распределения, заданного на вход алгоритму. В простейшем случае, распределение может быть равномерным (равновероятным) на сетке. Так как в случае дискретных распределений есть ненулевая вероятность того, что один и тот же вариант гиперпараметров будет сгенерирован дважды. Чтобы этого избежать, повторно встречающиеся варианты

пропускаются. В отличие от поиска по сетке, случайный поиск обладает рядом преимуществ:

1. Количество перебираемых вариантов – один из параметров алгоритма. Можно выбрать его в зависимости от имеющихся вычислительных ресурсов.

2. Зачастую оказывается, что некоторые параметры модели влияют на её качество незначительно (или не влияют вовсе). В таком случае, поиск по сетке будет работать в несколько раз дольше по сравнению с аналогичным поиском, при котором незначимый параметр зафиксирован (то есть, не перебирается), при этом, результаты в двух случаях будут близки. Случайный поиск при этом будет работать в двух вышеописанных случаях с одинаковой производительностью, при этом выдаст аналогичный результат.

3. Иногда бывают ситуации, когда близкие к оптимальным параметры нужно получить до того, как алгоритм отработал до конца (например, чтобы “прикинуть”, на какое качество можно рассчитывать, или в случае сбоя системы). В этом случае, даже результат промежуточного шага будет относительно неплохим, в отличие от результата поиска по сетке. Это связано с тем, при случайном выборе варианта гиперпараметров многие параметры к некоторому шагу примут довольно разнообразное число значений, в отличие от ситуации с поиском по сетке, при котором некоторые параметры могли принимать фиксированное значение во всех перебранных вариантах.

1.9.3 Выбор “равномерного” множества вариантов гиперпараметров

Если быть точным, то это не метод, а целый класс методов. Основной принцип подобных алгоритмов заключается в том, что вбираются некоторые варианты гиперпараметров так, чтобы, аппроксимация метрики качества была как можно лучше. Например, таковым является метод «латинский гиперкуб».

1.9.4 Байесовская оптимизация

Методы, основанные на байесовской оптимизации являются адаптивными, отличие которых от всех рассмотренных ранее методов заключается в том, что выбор вариантов гиперпараметров для проверки осуществляется с использованием результатов предыдущих проверок.

В данном случае мы решаем задачу максимизации метрики качества. При байесовском подходе искомая метрика качества рассматривается как случайная функция, для которой задано некоторое априорное распределение. Далее, при

вычислении значения этой функции в некоторой точке, вычисляется апостериорное распределение этой случайной функции, которое в дальнейшем используется в качестве априорного. При выборе следующей точки используют некоторый критерий и априорное распределение на значение рассматриваемой случайной функции (самым распространенной метрикой в этом случае является “матожидание улучшения качества”).

В отличие от случайного поиска, использование результатов предыдущих вычислений позволяет существенно ускорить сходимость. Правда, недостатком метода является вычислительно более сложный выбор следующей точки, что не подходит в случаях, когда метрика качества вычисляется очень быстро (в таком случае лучше воспользоваться случайным поиском).

ГЛАВА 2. ОТКРЫТЫЕ НАБОРЫ ДАННЫХ

2.1 Large Movie Review Dataset

Этот набор данных предоставлен исследователями из стэндфордского университета [7], содержащий 50000 отзывов на фильмы с сайта imdb.com. Так как количество отзывов на фильм может быть очень разным, при сборе данных ограничивались 30 отзывами на один фильм. Набор данных содержит одинаковое количество положительных и отрицательных отзывов. Негативным считается отзыв, имеющий оценку ≤ 4 из 10, а положительный отзыв имеет оценку ≥ 7. Нейтральные отзывы не включены в набор, то есть, оценка дается только сильно эмоционально окрашенным отзывам.

Данный набор достаточно популярен при исследовании сентимент-анализа и статья, в которой данный набор был описан, является широко цитируемой. Однако, отзывы на фильмы зачастую описаны “хорошим” языком с лексикой, сильно отличающейся от лексики социальных сетей.

2.2 Multi-Domain Sentiment Dataset

Данный набор данных был собран исследователями из университета Джона Хопкинса [8]. Набор данных состоит из отзывов на четыре разных типа продуктов: кники, DVD, электроника и кухонные принадлежности. Каждый отзыв содержит рейтинг (от 0 до 5), имя пользователя, оставившего отзыв, наименование продукта, название, дату и текст отзыва. Отзывы с рейтингом > 3 были отмечены как положительные, а отзывы с рейтингом < 3 отмечены как отрицательные, а все остальные отзывы были пропущены, потому что их полярность двусмысленна. Набор содержит 1000 положительных и отрицательных отзывов для каждой группы товаров. Также набор данных включает некоторое количество (от 3685 до 5945) неразмеченных отзывов.

Как и предыдущий набор данных, этот набор содержит отзывы на различные товары и лексически сильно отличается от текстов социальных сетей.

2.3 Sentiment Labeled Sentences Data Set

Данный набор данных собран исследователями из Школы Информации и Компьютерных наук Брена [9] и состоит их отзывов на трех различных площадок. В частности, отзывы на аксессуары для сотовых телефонов с Amazon, отзывы на фильмы с IMDB, а также отзывы на рестораны из Yelp.

2.4 Наборы, предоставленные компанией CrowdFlower

Компания CrowdFlower опубликовала множество наборов данных в открытый доступ [10], среди которых есть несколько наборов данных предназначенных для сентимент-анализа.

2.4.1 Airline Twitter sentiment

Набор данных был собран с целью сентимент-анализа сообщений в сети Twitter о крупных авиалиний США с целью определения недостатков их сервиса. Данные были собраны с февраля 2015 года и были размечены на 3 класса (положительные, отрицательные и нейтральные), а также содержали причины негативных отзывов (например, “задержки рейсов” или “грубое обслуживание”).

2.4.2 Apple Computers Twitter sentiment

Данные были собраны с целью анализа сообщений в Twitter о компании Apple. Выбирались твиты, содержащие тэги вида #AAPL, @apple. Твиты были размечены как положительные, отрицательные или нейтральные. Также данные содержат категорию сообщения (например, “состояние акций”, “новый продукт”, “сервис в Apple Store” и тд).

2.4.3 Coachella 2015 Twitter sentiment.

Данные были собраны с целью сентимент-анализа сообщений о крупном музыкальном фестивале Coachella для определения того, фанаты каких артистов были удовлетворены фестивалем больше всего а также для оценки всего фестиваля в целом.

2.4.4 New England Patriots Deflategate sentiment

Перед супербоулом (финальной игрой за звание чемпиона Национальной футбольной лиги США) был скандал о том, что мячи были сдуты и команда Patriots жульничала. Данные были собраны во время данного скандала с целью уловить настроение общественности по поводу данного скандала и матча.

2.4.5 Sentiment Analysis: Emotion in Text

Были собраны данные, содержащие смайлы, и размечены на 13 категорий в зависимости от эмоции (например, “счастье”, “печаль”, “злость”) с целью анализа качества работы логистической регрессии в задаче сентимент-анализа.

2.4.6 First GOP debate sentiment analysis

Было собрано несколько десятков тысяч твитов о политических дебатах в Августе 2015 года в штате Огайо с целью анализа дебатов. Была размечена полярность, какой кандидат был отмечен в твите а также категория сообщения.

2.4.7 Progressive issues sentiment analysis

Были собраны некоторые данные о некоторых общественных темах (например, аборты или феминизм). Твиты были классифицированы на классы “за”, “против” и “нейтрально” относительно вопроса. Также каждому твиту соответствует метка того, объективно или субъективно мнение автора.

2.4.8 Twitter sentiment analysis: Self-driving cars

Были собраны твиты о беспилотных автомобилях. Каждому твиту сопоставлен анализ полярности по пятибальной шкале, а также твиту была поставлена специальная метка в случае, если он не о беспилотных автомобилях.

2.5 Автоматический сбор данных

Заметим, что в целом больше исследований алгоритмов сентимент-анализа проводилось на различных отзывах, написанных “хорошим” языком, были собраны и автоматически размечены большие объемы данных для этой задачи. К сожалению, размеченных данных для сентимент-анализа текстов социальных сетей гораздо меньше. Однако количество неразмеченных данных в социальных сетях заметно больше и этим можно пользоваться, например, для автоматического сбора данных. Самым популярным способом сбором данных в литературе является сбор данных по смайлам: сообщение считается положительно окрашенным, если содержит только положительные смайлы (например, “:)”, “:-)”) и наоборот, отрицательно окрашенным, если содержит только отрицательные смайлы (например, “:(”, “:-(”).

Данный способ обладает рядом недостатков. Во-первых, полученный набор данных будет смещенным, ведь это сообщения, в которых эмоциональный окрас был выражен явно, поэтому скорей всего получившийся набор состоит из более простых примеров. Во-вторых, не всегда положительный смайл означает положительный настрой автора (например, если присутствует сарказм или пользователь просто любит ставить положительные смайлы не придавая им сильного смысла). И в третьих, не понятно, как собрать примеры с нейтральным окрасом, ведь если пользователь не поставил никакого смайла, далеко не факт, что его сообщение никак не окрашено.

Несмотря на вышеописанные недостатки, этот способ все же довольно полезен, так как позволяет собрать достаточно большой набор данных, который можно использовать не напрямую. Например, для составления словарей с положительными и отрицательными словами или обучения представления слов (word embedding), которые можно использовать в дальнейшем.

3.ПОДХОДЫ К РЕШЕНИЮ ЗАДАЧИ

Исследуем качество различных подходов к решению задачи сентимент-анализа текстов социальных сетей. В качестве данных для проверки будем использовать объединение наборов данных из пункта 2.4, так как, в отличие от других найденных данных, они состоят из размеченных текстов twitter, что наиболее близко к исследуемой тематике социальных текстов. При этом мы будем решать задачу бинарной классификации, то есть, каждое из сообщений должно быть классифицировано как положительное или отрицательного (“нейтральных” сообщений нет), так как при такой разметке можно будет использовать все данные наборы.

Некоторые статистики исходного набора можно видеть в таблице 1.

Количество объектов	45446
Количество положительных объектов	17094
Количество отрицательных объектов	28352
Средняя длина текста (в символах)	100.64
Среднее число слов	15.79

Таблица 1. Статистики набора данных

Положительные	**Отрицательные**
Mom says I have to get a new phone IMMEDIATELY....off to T-Mobile. she paying....	my computer at work is not working...booo... need a new one! TGIF!
third date went well....moving on to fourth!!!!!!	i was too slow to get $1 Up tix
@DirtyRose17 told you, you would sweep haha :-p	now aches & chills have joined the party. Why couldn't the cold just go away? It had to get worse?
Goodmorning	now the pic wont load up on twitter
I cheer myself up when I'm down by listening to my playlist called, Genius: Ballads and Cellos. I love my iPod and my taste of music.	@iggigg too busy to see me in London this evening. What is a boy to do?
Yayz! Today's is the day I call and see if I got that job! x3 I ish so happy... even though I slept horribly... So I'm kinda drained...	@tahninial just called me a cheeseburglar. He made me sad
couldn't resist ? http://blip.fm/~5z7v3	Didn't bring connector for camera on trip...saving

	new photos will have to wait another week..
had a cool lil night. Now at Berrie's about to eat pizza waitin for @NOEL4PRESIDENT	FML.. today sucks.. i just hope the dance will bring my soul up.. i pray, but im still sad..i hateee todayyyyyy!!! >;(
just watched the movie Wanted... it was pretty darn good.	Flash lost my frisby on a roof. Sad days LOL
leaving florida want to live there forever! Texan*Girl	Hate this song ? http://blip.fm/~5jg6f

Таблица 2 – примеры положительных и отрицательных примеров

3.1 VADER

VADER представляет собой систему на основе правил, предназначенную для сентимент-анализа текстов социальных сетей. При проектировании системы основными требованиями были:

1) Хорошие результаты на текстах из социальных сетей.
2) Система не должна требовать данные для обучения.
3) Вычислительная производительность, достаточная для обработки большого потока данных онлайн.
4) Принятые компромиссы, необходимые для быстрой работы системы, не должны сильно ухудшить качество системы.

Для создания такой системы было сделано следующее:

1) Подготовлен качественный лексикон, содержащий полярность и яркость окрашенных часто упоминаемых в социальных сетях слов (однако подобранный лексикон может быть использован и в других областях).
2) Найдены довольно общие грамматические и синтаксические правила, позволяющие достаточно точно и обобщаемо определять полярность текста.

Реализация данного метода включена в NLTK, самую известную на данный момент платформу для анализа текста на естественном языке.

3.2 Наивный байесовский классификатор

Наивный байесовский классификатор является достаточно простым и опирается на предположения, которые очень редко выполняются. Но, несмотря на это на практике он часто показывает очень хорошие результаты в задаче классификации текста. Рассмотрим его поподробнее.

Наивный байесовский классификатор строится на теореме Байеса:

$$P(c|d) = \frac{P(d|c) \cdot P(c)}{P(d)}$$

Здесь $P(c|d)$ – вероятность того, что документ d принадлежит классу c, $P(d|c)$ – вероятность появления документа d при условии, что он принадлежит классу c, $P(c)$ – априорная вероятность того, что встретится документ класса c, а $P(d)$ – априорная вероятность того, что встретится документ d.

Данная формула позволяет нам получить вероятности всех классов документа при условии, что мы определили вероятности документа при условии класса.

Однако, задание $P(d|c)$ в общем случае – сложная задача, поэтому для упрощения используют байесовское предположение, а именно: $P(d|c) = P(w_1|c) \cdot P(w_2|c) \cdot \ldots \cdot P(w_n|c)$, где $P(w_i|c)$ – вероятность встретить слово w_i в документе класса c.

На самом деле, данное предположение вряд ли верно в тексте на естественном языке, например потому, что часто имея слово нетрудно предугадать его контекст и наоборот, то есть, порядок слов имеет значение. Здесь же он опущен. Плюс, понятно, что слова в документах возникают не независимо.

Однако на практике оказывается, что наивный байесовский классификатор работает хорошо в задачах классификации текста. При этом предсказание вероятности принадлежности классу обычно работает очень плохо, так как каждый объект относится к классу с вероятностью либо очень близкой к нулю либо очень близкой к единице.

Для оценки $P(w_i|c)$ используют оценку максимального правдоподобия со сглаживанием Лапласа: $P(w_i|c) = \frac{W_{ic}+\alpha}{\sum_j W_{jc}+\alpha|V|}$, где $\mathrm{W_{ic}}$ – количество раз, которое слово w_i встречается в документах класса c. При этом сглаживание необходимо, так как оценка максимального правдоподобия является точной только при большом размере выборки. На практике при анализе текста часто возникает ситуация, в которой многие слова встречаются очень редко (например, вполне нормально, когда половина слов встречаются лишь раз), поэтому оценка вероятности методом максимального правдоподобия приводит к очень неточной классификации. А

сглаживание в данном случае приводит к оценке требуемой вероятности в байесовской модели, где искомая вероятность является случайной величиной, имеющей бета-распределение. Байесовские оценки являются достаточно устойчивыми и работают адекватно даже в случаях, когда данных очень мало.

Оценить $P(c)$ можно по формуле $P(c) = \frac{D_c}{D}$, где D_c – число документов класса c, а D – общее число документов. При этом видно, что для того, чтобы найти класс с максимальным $P(c|d)$ не обязательно знать $P(d)$, ведь оно участвует только в знаменателе формулы и не влияет на то, у какого класса вероятность $P(c|d)$ больше, поэтому оценивать $P(d)$ нет необходимости.

3.3 Логистическая регрессия

Для описания логистической регрессии воспользуемся материалом из [8].

Логистическая регрессия применяется для предсказания вероятности возникновения некоторого события по значениям множества признаков. Для этого вводится так называемая зависимая переменная y, которая принимает 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 $x_1, \dots x_n$, на основе значений которых требуется вычислить вероятность принятия того или иного значения зависимой переменной.

Делается предположение о том, что вероятность события $y = 1$ равняется:

$$P\{y = 1|x\} = f(z), \text{где z} = \theta_1 x_1 + \cdots + \theta_n x_n \quad (1)$$

Здесь x и θ — векторы-столбцы значений независимых переменных $x_1, \dots, x_n$ и параметров (коэффициентов регрессии) — вещественных чисел $\theta_1, \dots, \theta_n$, соответственно, а $f(z)$ — так называемая *логистическая функция* (иногда также называемая сигмоидом или логит-функцией): $f(z) = \frac{1}{1+ e^{-z}}$.

Для подбора параметров $\theta_1, \dots, \theta_n$ необходимо составить обучающую выборку, состоящую из наборов значений независимых переменных и соответствующих им значений зависимой переменной y. Формально, это множество пар $\left(x^{(1)}, y^{(1)}\right), \dots, (x^{(m)}, y^{(m)})$, где $x^i \in \mathbb{R}^n$ — вектор значений независимых переменных, а $y^{(i)} \in \{0,1\}$ — соответствующее им значение y. Каждая такая пара называется обучающим примером.

Обычно используется метод максимального правдоподобия, согласно которому выбираются параметры θ, максимизирующие значение функции правдоподобия на обучающей выборке.

Для улучшения обобщающей способности получающейся модели, то есть 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.

В качестве априорного распределения часто выступает многомерное нормальное распределение с нулевым средним и матрицей ковариации, соответствующее априорному убеждению о том, что все коэффициенты регрессии должны быть небольшими числами, идеально — многие малозначимые коэффициенты должны быть нулями. Подставив плотность этого априорного распределения в формулу выше и прологарифмировав, получим следующую оптимизационную задачу:

$$\sum_{i=1}^{m} \log P\{y^i | x^i, \theta\} - \lambda \|\theta\|_2 \quad \rightarrow max \tag{2}$$

где $\lambda = const/\sigma^2$ — параметр регуляризации. Этот метод известен как L2-регуляризованная логистическая регрессия, так как в целевую функцию входит L2-норма вектора параметров для регуляризации.

Если вместо L2-нормы использовать L1-норму, что эквивалентно использованию распределения Лапласа, как априорного, вместо нормального, то получится другой распространённый вариант метода — L1-регуляризованная логистическая регрессия:

$$\sum_{i=1}^{m} \log P\{y^i | x^i, \theta\} - \lambda \|\theta\|_1 \rightarrow \max \tag{3}$$

3.4 Метод опорных векторов

Метод опорных векторов (SVM – Support vector machines) использует гиперплоскость, чтобы классифицировать данные по 2 классам.

SVM позволяет спроецировать ваши данные в пространство большей размерности. Когда данные спроецированы, SVM определяет лучшую гиперплоскость, которая делит данные на 2 класса.

Например, если объекты двух классов линейно разделены в пространстве, можно построить разделяющую гиперплоскость. Далее, если добавляется новый объект, можно предсказать его класс, зная, по какую сторону от разделяющей гиперплоскости он находится. SVM самостоятельно определяет функцию гиперплоскости.

Если линейная функция плохо подходит для разделения объектов, SVM позволяет отобразить данные в пространство с более высокой размерностью и провести гиперплоскость уже в новом пространстве. Такой прием называется kernel trick.

При построении требуемой гиперплоскости SVM оптимизирует «отступ» (margin). Отступ гиперплоскости – это расстояние между гиперплоскостью и двумя ближайшими точками данных каждого класса.

Для поддержки не линейно разделимых данных SVM строит гиперплоскость, минимизируя следующую функцию потерь:

$$\frac{1}{n}\sum_{i}\max(0, 1 - y_i(\vec{w}\cdot\vec{x}_i - b)) + \lambda\|\vec{w}\|^2$$

Суть в том, что SVM пытается максимизировать отступ так, чтобы гиперплоскость находилась примерно на одинаковом расстоянии от объектов обоих классов – это снижает шанс ошибок классификации.

Гиперплоскость равноудалена от ближайших объектов двух классов. Эти объекты – примеры, которые называются опорными векторами (support vectors), потому что они поддерживают гиперплоскость.

Этот метод требует обучения. Чтобы показать SVM, что такое классы, используется набор данных, только после этого он оказывается способен классифицировать новые данные.

Есть множество реализаций SVM. Самые популярные – это scikit-learn, MATLAB и libsvm.

3.5 Тестирование и результаты

Оценивать различные методы будем посредством вычисления метрик качества AUC и accuracy используя скользящий контроль с 10 разбиениями. Для подбора гиперпараметров наивного байесовского классификатора и логистической регрессии воспользуемся методом поиска по сетке, так как в данных алгоритмах есть только один параметр (сила сглаживания), который на практике достаточно подобрать только с точностью до порядка, а метод является очень простым. В случае с VADER, перебор параметров не требуется, так как эта модель уже обучена разработчиками алгоритма.

Результаты представлены в следующей таблице:

	AUC	Accuracy
VADER	0.72	0.66
Наивный байесовский классификатор	0.85	0.78
Логистическая регрессия	0.88	0.81
Метод опорных векторов	-	0.81

Таблица 3 – результаты исследованных методов

Заключение

В ходе проведенной работы было сделано следующее:

1. Изучены имеющиеся алгоритмы и подходы сентимент-анализа, а также открытые размеченные наборы данных для обучения и тестирования.
2. Реализованы изученные алгоритмы сентимент-анализа.
3. Проведен сравнительный анализ имеющихся алгоритмов “общего” сентимент-анализа а также реализованных подходов. Показано, что качество алгоритмов, обученных для распознавания тональности текста социальных сетей заметно выше “общих” подходов.

Список использованной литературы

1. Miller, G., R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. Wordnet: an online lexical database. Oxford Univ. Press., 1990.
2. Mohammad, S., C. Dunne., and B. Dorr. Generating high-coverage semantic orientation lexicons from overly marked words and a thesaurus. In Proceeding of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2009, 2009.
3. Pang, B., L Lee, and S. Vaithynathan. Thumbs up?: sentiment classification using machine learning techniques. In Proceeding of Conference on Empirical Methods in Natural Language Processing (EMNLP-2002), 2002.
4. Логистическая регрессия [Электронный ресурс] / Википедия. – Режим доступа: https://ru.wikipedia.org/wiki/Логистическая_регрессия - Дата доступа: 23.05.2016
5. Taboada M., J. Brooke, M. Tofilouski, K. Voll, and M. Stede, Lexicon-based methods for sentiment analysis, Computational Intelligence, 2010.
6. Turney, P. Thumbs up or thymbs down?: semantic orientation applied to unsupervised classification of reviews. In Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL2002), 2002.
7. Mass A. et al, Learning Word Vectors for Sentiment Analysis. In Proceeding of the 49th Annual Meeting for Computational Linguists: Human Language Technologies, 2011.
8. Blitzer J, Dredze M, Pereira F. Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. Association of Computational Linguistics (ACL), 2007.
9. Kotzias et. al. From Group to Individual Labels using Deep Features. KDD 2017
10. https://www.crowdflower.com/data-for-everyone/
11. Hutto, C.J., Gilbert, E.E, VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text