УДК 004.942

Классификация пользователей по данным из сети интернет

Пономарев И.В., Черкасов С.В.
Алтайский государственный университет, г. Барнаул
cherkasov.475@yandex.ru, igorpon@mail.ru

Аннотация

Проведен анализ способов сбора информации о пользователях на различных площадках в сети интернет. Рассмотрен способ извлечения информации из социальной сети "ВКонтакте". Для создания информационной базы исследования было выбрано наиболее информативный, на наш взгляд, раздел – список групп, в которых состоит пользователь. В процессе исследования был разработан алгоритм разбора текста до уровня понимания компьютером. С помощь наивного байесовского классификатора реализована классификация социального положения пользователя. Этот же алгоритм без каких-либо изменений можно адаптировать к классификации интересов пользователя.

Ключевые слова: классификация, соцсети, пользователи, анализ данных, сбор информации, байесовский классификатор.

1. Введение

Сбор и анализ информации о пользователе в данное время является одной из самых актуальных проблем для множества сфер деятельности. Нами решено рассмотреть конкретное направление – реклама, где невероятно важно знать о человеке какую-либо информацию. Ведь, если мы сможем узнать интересы пользователя, мы можем предоставить для него целевую рекламу, что, несомненно, увеличит шансы покупки товара именно у нас.

Анализ данных в этой области заключает в себе два аспекта: скорость и точность. Но так как область применения нашего алгоритма веб-сайты, то показ таргетированной (целевой) рекламы должен быть предоставлен сразу как человек попал на сайт. Из чего следует, что основным направлением является скорость определения категории пользователя.

2. Выбор источника информации

В современном мире существует множество различных способов откуда получить информацию. Их условно можно разделить на два типа: ручные и автоматические.

Ручные представляют собой методы, где пользователь сам указывает о себе информацию, например, анкетирование, заполнение профиля при регистрации на сайте и т.д. Плюсом этого метода является высокая точность, ведь пользователь почти всегда указывает верную информацию о себе. С другой стороны, пользователь не всегда захочет заполнять большой объем информации, и мы не получим никакой информации о нем, что, в свою очередь, является минусом данного метода. К минусам можно также отнести время, затраченное на заполнение пользователем анкеты, а скорость нам очень важна.

Автоматические – это методы, где принадлежность к классу пользователя вычисляется алгоритмами на основе предыдущих действий пользователей или информации, которую

они заполняли ранее. К плюсам можем отнести скорость определения класса человека, что, в свою очередь, удовлетворяет нашим условиям. Минусом является то, что мы делаем вывод на информации из прошлого. Значит наш результат может быть уже не актуальным для пользователя.

В нашем случае, был сделан выбор в пользу автоматических методов. Два самых надежных источника данных о пользователе являются его социальные сети и поисковые запросы. Поисковые запросы для нашей задачи подходят лучше всего потому, что все всегда что-то ищут в интернете, а во-вторых, почти всегда мы можем определить самые актуальные запросы человека и по ним сформировать мнение о нем. Большой проблемой данного метода является доступность к базе запросов человека. Эту информацию может получить только владелец сайта-поисковика, т.е такие копании как Google, Яндекс, Майл.ру и другие. Однако, есть возможность собирать и анализировать открытую информацию, например, данные пользователей из социальных сетей. Информация там всегда в открытом доступе и актуальность на хорошем уровне. Для определенности выберем социальную сеть в “ВКонтакте”.

Нам предоставляется 4 направления откуда брать информацию (на личной страничке пользователя): фотографии, записи репостов на стене пользователя, личные сообщения, список групп, на которые подписан пользователь, друзья пользователя.

Фотографии сложно технически обрабатывать и классифицировать, к тому же многие пользователи не публикуют свои фотографии на странице (около 9-10%), а используют фото знаменитостей, животных, мультгероев, которые могут запутать алгоритм. При использовании записей на стене появилась та же проблема, они присутствуют не у многих.

Друзья пользователя также не подойдут, так как для того, чтобы с нужной точностью определить кем является пользователь, нужно знать кем являются сами друзья, что, в свою очередь, вызывает рекурсивный процесс.

Личные сообщения для сторонних пользователей закрыты и даже не рассматривались нами как способ в моем случае.

Методом исключения был выбран поток с данных из групп, на которые подписан человек. Получение названий этих групп было реализовано с помощью официального API Vk для разработчиков [1]. На выходе мы получаем список названий сообществ, на которые подписан пользователь.

3. Подготовка текста для анализа

Для дальнейшей работы нужно привести текст к более понятному для компьютера, приведения его к “машинному языку” [2]. Сделаем это с использованием следующего алгоритма:

1. Разбиение на токены. Токены – это элементарные частицы текста, т.е. слова. Для выполнение этой задачи нужно создать массив, в каждый элемент которого записывать все слова из исходного списка по порядку, исключая знаки препинания, специальные символы и пробелы. На выходе получаем один большой массив со всеми словами из списка.

2. Удаление стоп-слов. Стоп-слова-это слова, которые не несут никакой смысловой нагрузки и союзы. Мы составляем базу данных стоп-слов. Это можно сделать вручную, или найти готовую в интернете. Далее, мы проходим по полученному в предыдущем этапе массиву и ищем каждое слово в базе данных стоп-слов, если оно есть в базе, то мы его удаляем из массива. На выходе получаем массив из только значащих слов.

3. Стимминг. Стимминг – отсечение от слова окончаний и суффиксов, чтобы оставшаяся часть, называемая *stem*, была одинаковой для всех грамматических форм слова.

Например, коты-котик-кот → кот. Алгоритм стимминга достаточно сложный, так как требует большую базу знаний окончаний и суффиксов. Использовав алгоритм стимминга для каждого слова массива, мы получаем на выходе массив уже понятных для компьютера слов.

4. Формирование базы слов. На данном этапе мы записываем в нашу базу данных весь полученный массив, подсчитывая количество вхождений каждого слова в этом массиве. На выходе получаем базу слов по пользователю.

4. Выбор алгоритма классификации

Для проведения классификации был выбран наивный байесовский классификатор, так как он является одним из самых быстрых классификаторов [3].

По сути, он представляет собой вероятностную модель. Пусть задано множество наблюдений $x = (x_1, x_2, \ldots, x_n)$. Модель присваивает каждому наблюдению условную вероятность $p(C_k|x_1, x_2, \ldots, x_n)$, C_k – класс.

Используя теорему Байеса, можно записать:

$$p(C_k|x) = \frac{p(C_k) \cdot p(x|C_k)}{p(x)}.$$

В этой формуле интерес с точки зрения классификации представляет только числитель, поскольку знаменатель от метки классов не зависит и является константой. При условии, что признаки независимы, можно показать, что

$$p(C_k|x) = p(C_k) \prod_{i=1}^{n} p(x_i|C_k).$$

Тогда простой байесовский классификатор можно рассматривать как функцию, которая каждому выходному значению модели присваивает метку класса, т.е. $y = C_k$ следующим образом:

$$C_{opt} = argmax \left(p(y = c) \prod_{i=1}^{n} p(f_i|y = c) \right).$$

Для классификации рассчитываются вероятности попадания пользователя в тот или иной класс: выбирается тот класс, в котором вероятность больше. Если слово ни разу не встречается в базе данных, т.е. его вероятность становиться равна нулю, то общая вероятность попадания в класс пользователя также равна нулю. Во избежание этой проблемы, если слово не встречается, его вероятность указываем равной 0,0000001.

5. Реализация алгоритма и анализ результатов

Так как алгоритм будет использоваться на сайте, языком программирования был выбран PHP.

Было создано три веб-страницы: окно ввода ссылки на пользователя, окно вывода результатов, окно обучения алгоритма и создание базы данных. Для каждого действия была реализована отдельная функция, а также была создана база данных для хранения пользователей, а также таблица слов.

Обучение проходит следующем образом:

1. Вводиться ссылка на пользователя Вконтакте. Формат www.vk.com/id******.
2. Указывается принадлежность этого пользователя к классу.

3. Нажимается кнопка отправить, которая и запускает алгоритм записи данных в базу.

Данные брались с реальных пользователей вручную. Были выбраны 4 класса: школьник, студент, рабочий, пенсионер.

Во время заполнения данных проводились тестовые классификации. На 50 записях результаты практически не попадали в реальные данные. После достижения 350 записей алгоритм стал выдавать стабильно верные результаты.

Трудности у программы возникли в классах “рабочий” и “пенсионер”. Алгоритм путает эти два класса. Причина, скорее всего, кроется в том, что эти пользователи менее активны в социальных сетях и подписаны чаще всего на информационные сайты или группы. Очень интересным выводом можно сказать, что многие из юмористических групп показали явное распределение по возрастному фактору.

6. Выводы

Подводя итоги работы, программа в полной мере удовлетворяет условиям скорости и эффективности. Так как в качестве классов можно использовать абсолютно любой фактор. Алгоритм является универсальным и может использоваться в любой сфере. Для изменения классификации нужно:

1. Указать набор классов.
2.Произвести обучение алгоритма с этой классификацией.

Список литературы

1. Официальная документация Vk API. — URL: https://vk.com/dev/methods.

2. Обработка естественного языка в Node.js. — URL: https://medium.com/devschacht/natural-language-processing-for-node-js-da990c7dd886.

3. Domingos P., Pazzani M. On the optimality of the simple Bayesian classifier under zero-one loss // Machine Learning. — 1997. — no. 29. — P. 103–137.