**Наивный байесовский классификатор документов в Excel**

Самое распространенное применение *наивного Байеса* — классификация документов. Является ли это электронное письмо спамом или наоборот, долгожданной новостью? Эта запись в Twitter — благодушная или сердитая? Нужно ли передавать этот перехваченный звонок по сотовому для дальнейшего исследования федеральным агентам? Вы предоставляете «данные для обучения», например, классифицированные примеры документов, обучающему алгоритму, который в дальнейшем сможет «разбить» новые документы на те же категории, используя имеющиеся знания.[[1]](#footnote-1)

Самый распространенный подход к классификации документов — это использование модели *набор слов* в сочетании с наивным байесовским классификатором. Модель *набор слов* воспринимает документ как беспорядочное множество слов. «Джонни съел сыр» для него то же самое, что «сыр съел Джонни» — и то и другое состоит из одного и того же множества слов: {«Джонни», «съел», «сыр»}.

*Небольшое введение в теорию вероятностей.* Выражение р() используется для обозначения вероятности. Например, р(А) = 0*,*2 означает, что событие А произойдет с вероятностью 20%. Выражения, типа р(А|B) используются для обозначения условных вероятностей. Например, р(А|B) = 0,3 означает, что вероятность события А, при условии, что случилось событие В, составляет 30%. Совместная вероятность р(А, B) используется для обозначения вероятности того, что события А и В произойдут одновременно. Если события А и В независимы, то р(А, B) = р(А) \* р(B). Если события А и В зависимы, то р(А, B) = р(А) \* р(B|A).

Для удобства манипулирования условными вероятностями Томас Байес доказал теорему (подробнее см. [Идеи Байеса для менеджеров](http://baguzin.ru/wp/?p=6355)):



В качестве примера мы изучаем твиты о сервисе для отправки электронных писем – [Mandrill.com](http://mandrill.com/). При поиске по ключевому слову – *mandrill* – помимо полезных, появляются также ссылки, не имеющие отношения к делу. Наша задача – отфильтровать релевантные твиты. Допустим, ранее мы накопили базу, включающую 300 твитов: 150 – о приложении Mandrill.com, и 150 – других.

Каждый твит мы разбиваем на отдельные слова (называемые жетонами — *token*). Нам важны две вероятности:

р (приложение | слово1, слово2 ...)
р (другое | слово1, слово2, ...)

Это вероятность того, что твит либо о приложении, либо о чем-то другом, при том, что мы обнаруживаем слова: «слово1», «слово2» и т.д.

Если

(1) р (приложение | слово1, слово2, ...) > р (другое | слово1, слово2, ...)

то данный твит — о Mandrill.com. Но как же вычислить эти вероятности? Первый шаг — использование теоремы Байеса, которая позволяет переписать условную вероятность приложения как:

$$\left(2\right) р \left(прилож. \right|слово1, слово2,…)=\frac{р \left(прилож.\right)\*р (слово1, слово2, ... | прилож.) }{р (слово1, слово2, ...)}$$

Точно так же

$$\left(3\right) р \left(другое \right|слово1, слово2,…)=\frac{р \left(другое\right)\*р (слово1, слово2, ... | другое) }{р (слово1, слово2, ...)}$$

Подставив (2) и (3) в (1) и умножив обе части на *р(слово1, слово2, ...)*, получим условие (1) в виде:

(4) р(прилож.) \* р(слово1,слово2,...| прилож.) > р(другое) \* р(слово1,слово2,...| другое)

Применяемое для анализа [правило апостериорного максимума](https://ru.wikipedia.org/wiki/%D0%9E%D1%86%D0%B5%D0%BD%D0%BA%D0%B0_%D0%B0%D0%BF%D0%BE%D1%81%D1%82%D0%B5%D1%80%D0%B8%D0%BE%D1%80%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BC%D0%B0%D0%BA%D1%81%D0%B8%D0%BC%D1%83%D0%BC%D0%B0) (МАР) позволяет, во-первых, не обращать внимание на различие значений р (прилож.) и р (другое), а во-вторых, считать вероятность вхождения слов в твит независимым (хотя это и не так), и заменить:

р(слово1,слово2,...| прилож.) –> р(слово1| прилож.) \* р(слово2| прилож.) \* …
р(слово1,слово2,...|другое) –> р(слово1|другое) \* р(слово2|другое) \* …

В окончательном виде мы будем сравнивать две величины:

(5) р(слово1| прилож.) \* р(слово2| прилож.) \* … > р(слово1|другое) \* р(слово2|другое) \* …

Предположение о независимости позволяет разбить совместную условную вероятность набора слов при известном классе на вероятности нахождения каждого слова в данном классе. Считая слова независимыми, мы вносим в алгоритм МАР множество ошибок, но, в конце концов, они не влияют на правильность выбора между набором, относящимся к приложению и другими твитам.

Осталось решить две проблемы: что делать с редкими словами, и как победить исчезающе малые величины, появляющиеся при перемножении большого числа вероятностей, близких к нулю? Принято добавлять единицу к каждому значению (даже нулевому). Это называется дополнительным сглаживанием и часто используется для приспособления неведомых ранее слов к модели наборов слов. А вместо умножения используется сложение логарифмов. Например, у вас есть произведение: 0,2 \* 0,8. Прологарифмируйте его: ln(0,2 \* 0,8) = ln(0,2) + ln(0,8).

Итак, все объяснения даны, и можно перейти к Excel. На первых двух листах книги с примерами содержатся по 150 твитов, относящихся к приложению Mandrill.com (рис. 1) и к другим темам. Последовательно в оригинальном тексте твитов все буквы заменяются на строчные, а затем знаки препинания – на пробелы. Например, формула в ячейке Е2 =ПОДСТАВИТЬ(D2;"?";" ") – заменяет в тексте, содержащемся в ячейке D2, все знаки вопроса на пробелы.



Рис. 1. Удаление лишних знаков в базе твитов о приложении

Теперь нам необходимо сосчитать, сколько раз каждое слово используется в записях данной категории. Для этого нужно собрать все слова из твитов каждой базы в одном столбце. Предполагая, что каждый твит содержит не более 30 слов, и собираясь присвоить каждому жетону отдельную строку, вам понадобится 150 \* 30 = 4500 строк. Создайте новый лист, назовите его *Жетоны\_прил.* Назовите ячейку А1 *Твиты*. Скопируйте в буфер значения Н2:Н151 с листа *Приложение.* Выделите на листе *Жетоны\_прил.* область А2:А4501 и кликните *Вставить* –> *Специальная вставка* –> *значения* (рис. 2). Нажмите Ok. Обратите внимание: так как вы вставляете 150 твитов в 4500 строк, Excel повторяет все за вас. Это означает, что если вы выделите первое слово из первого твита в строке 2, этот самый твит повторится для выделения второго слова в строке 152, третьего — в 302 и т.д.



Рис. 2. Специальная вставка

Изучите формулы в столбцах В:D листа *Жетоны\_прил.*, чтобы понять механику последовательного извлечения жетонов из твита (рис. 3). Аналогично создайте лист *Жетоны\_др.* для базы твитов, не относящихся к приложению Mandrill.com.



Рис. 3. Фрагмент листа *Жетоны\_прил.,* на котором извлекаются жетоны из базы данных твитов, относящихся к приложению Mandrill.com

Теперь на базе листа *Жетоны\_прил.* следует создать сводную таблицу, которая подсчитает число вхождений каждого жетона. С помощью фильтра сводной таблицы исключите слова длиною до 4 символов, а также добавьте столбцы для подсчета логарифма частоты вхождения жетона (рис. 4). Повторите операцию для листа *Жетоны\_др.*



Рис. 4. Логарифмированные вероятности для жетонов, относящихся к приложению

Теперь, когда модель классификатора «обучена», настало время ее использовать. На листе *Тест* размещены 20 твитов, которые нужно классифицировать. Они также обработаны (как и на рис. 1). Поместим подготовленные твиты на лист *Классификация*. Выделите D2:D21 и выберите *ДАННЫЕ* –> *Текст по столбцам*. В появившемся окне выберите *С разделителями* и нажмите *Далее*. На втором шаге выберите знаки табуляции и пробела в качестве разделителей, а также *Считать последовательные разделители одним* (рис. 5). Ограничитель строк установите *(нет)*. Нажмите *Далее*. На последнем шаге *Формат данных столбца* установите *общий*. Нажмите *Готово*.



Рис. 5. Разделение тестовых твитов по столбцам

Процедура разбросает твиты по столбцам всего листа до столбца AI (рис. 6).



Рис. 6. Жетоны из тестовых твитов

Теперь с помощью функции ВПР извлечем данные о логарифмах вероятностей вхождения тестовых жетонов в два набора данных (приложение / другие). Сравним суммы, и сделаем вывод о принадлежности тестов к тому или иному классу (рис. 7). Выделены цветом твиты, разность логарифмов по которым менее 1. Подробнее с формулами можно ознакомиться на листе *Классификация.*



Рис. 7. Классифицированные тестовые твиты

Вот и все. Модель построена, предположения сделаны.

1. Написано по материалам книги Джона Формана [Много цифр: Анализ больших данных при помощи Excel](http://baguzin.ru/wp/?p=13802). – М.: Альпина Паблишер, 2016. – С. 101–128 [↑](#footnote-ref-1)