Введение в ML, часть 2

Алексей Шаграев

Перед тем, как начать

- Есть ли какие-нибудь вопросы? ☺
- Код по-прежнему лежит на гитхабе: https://github.com/ashagraev/ml_basics
- Проверьте, работает ли у вас text.py

TF-IDF

- Максимальные по частоте слова в любой теме неинформативны
- Критерий TD-IDF:
  - TF = частота терма в документе, от term frequency
  - IDF = inverse document frequency, обычно вычисляется так:

$$idf(w) = \log \frac{|D|}{|\{d \in D | w \in d\}|}$$

  - То есть, IDF для конкретного слова – логарифм отношения общего числа документов к числу документов, в которых есть это слово

Задача №7

- Загрузите документы из тематик:
  - alt.atheism
  - rec.sport.baseball
  - rec.autos
  - sci.space
  - talk.politics.guns
  - comp.windows.x
  - sci.med
- Распечатайте топ-50 слов по TF-IDF из тематик atheism и autos
- Для вычисления IDF используйте глобальную статистику
- Для вычисления TF используйте документы из темы

Задача №8

- Постройте словарь с TD-IDF слов в тематиках alt.atheism и rec.autos
- Вычислите средние значения TD-IDF из полученных словарей для слов из нескольких документов из тематик alt.atheism и rec.autos
- Подробнее:
  - tfidfAutos – словарь, в котором IDF определяется по общей статистике слов, а tf – по количеству вхождений слова в документы тематики autos
  - tfidfAtheism – словарь, в котором IDF определяется по общей статистике слов, а tf – по количеству вхождений слова в документы тематики atheism
  - Каждый документ – это последовательность слов. Напишите функцию, которая принимает документ и словарь значений TF-IDF, а возвращает среднее значение TF-IDF из словаря на словах из этого документа

Метрики качества

- Бинарный классификатор предсказывает для документа один из двух классов
- При этом документ сам относится к одному из классов
- Тогда есть четыре варианта:
  - True positive: classifier(d) = 1, target(d) = 1
  - False positive: classifier(d) = 1, target(d) = 0
  - True negative: classifier(d) = 0, target(d) = 0
  - False negative: classifier(d) = 0, target(d) = 1

Метрики качества

	target(d) = 1	target(d) = 0
classifier(d) = 1	True Positive	False Positive
classifier(d) = 0	False Negative	True Negative

Метрики качества

	target(d) = 1	target(d) = 0
classifier(d) = 1	True Positive	False Positive
classifier(d) = 0	False Negative	True Negative

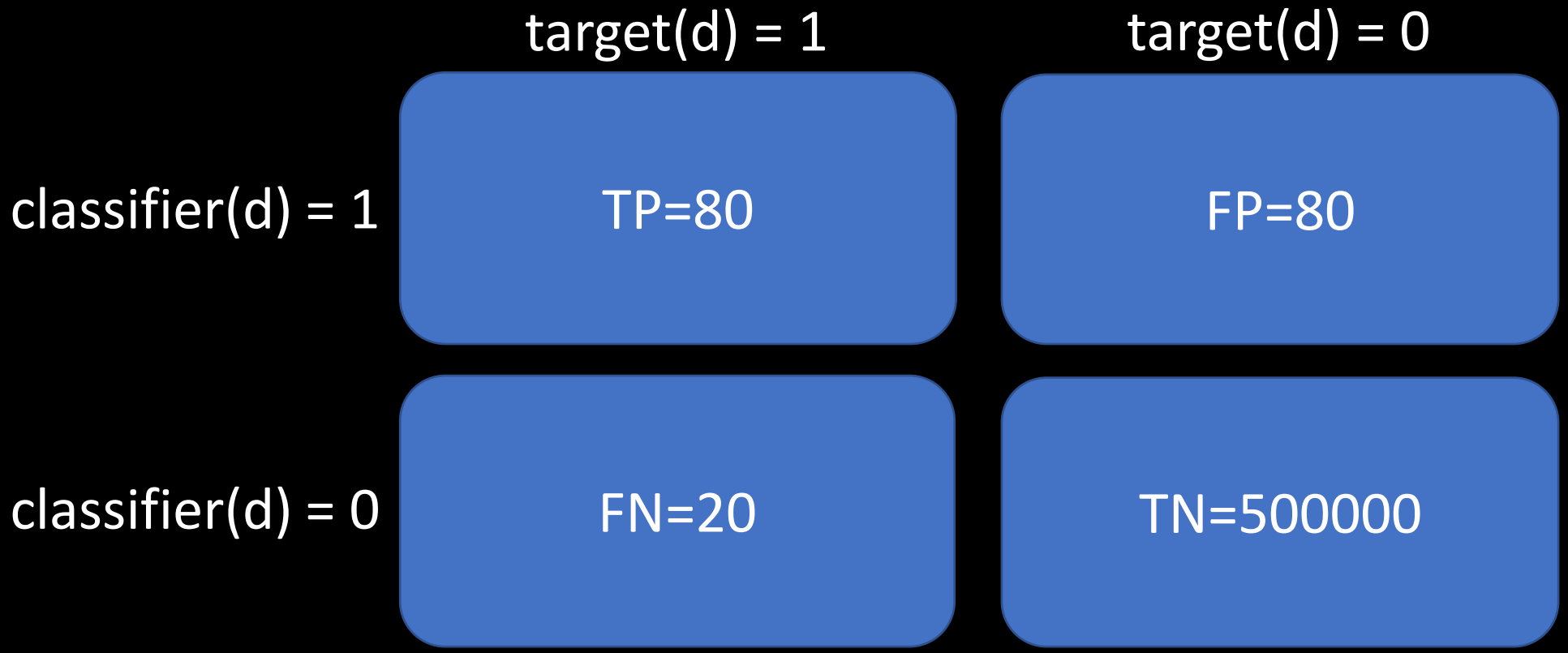
$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN}$$

Метрики качества: пример

	target(d) = 1	target(d) = 0
classifier(d) = 1	TP=80	FP=80
classifier(d) = 0	FN=20	TN=500000

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

Метрики качества: пример



$Precision = 0.50$ $Recall = 0.80$

Задача №9

- Будем считать, что все документы – либо из темы alt.atheism, либо из темы rec.autos
- Будем считать, что target(d) = 1, если он из темы alt.atheism
- Возьмите значения tfidf из темы alt.atheism
- Постройте массив пар
  - Первый компонент пары – среднее значение tfidf для слов документа d
  - Второй компонент пары – target(d)
  - Отсортируйте массив по убыванию
- Вычислите precision и recall, если классификатор предсказывает класс 1 для первых 10% документов из этого массива

Задача №10

- Вычислите значения precision и recall для случаев, когда классификатор предсказывает класс 1 для первых 10%, 20%, …, 90% документов
- Постройте диаграмму рассеяния для значений precision и recall

Ранжирование

- Пусть есть n игроков, каждый из которых обладает некоторой «силой»
- Силу выражают числовые «рейтинги»
- Скажем, что игроки играют попарно, и, если рейтинги равны r1 и r2, то первый побеждает с вероятностью

$$P_1 = \frac{1}{1 + 10^{\frac{r_2 - r_1}{400}}}$$

Задача №10

- Скажем, что игроки играют попарно, и, если рейтинги равны r1 и r2, то первый побеждает с вероятностью

$$P_1 = \frac{1}{1 + 10^{\frac{r_2 - r_1}{400}}}$$

- Реализуйте функцию, которая вычисляет вероятность победы игроков в зависимости от их рейтингов
- Чему равна вероятности победы, если рейтинги игроков совпадают?
- А если у одного из них рейтинг на 400 выше?

Задача №11

- Пусть есть 5 игроков с рейтингами 2500, 2200, 1900, 1600, 1300
- Реализуйте функцию, которая разыгрывает один круговой турнир: каждый играет с каждым один раз
- Разыграйте 1000 турниров
- Вычислите для каждого игрока, какое общее число побед он одержал
- Как проверить, что всё работает корректно?

Обновление рейтингов

- Сила игроков не является постоянной и её нужно обновлять по результатам игр
- Пусть играют два игрока с рейтингами r1 и r2, первый должен выиграть с вероятностью p1
- По факту он получит либо одно очко (победа), либо ноль очков
- Обозначим его число очков через S
- Тогда обновим его рейтинг, прибавив к нему величину

$$k \cdot (S - p_1)$$

- k – «скорость изменения рейтингов»

Задача №12

- Тогда обновим его рейтинг, прибавив к нему величину

$$k \cdot (S - p_1)$$

- k – «скорость изменения рейтингов»
- Реализуйте функцию, которая принимает рейтинги игроков, результат игры и возвращает обновленные рейтинги
- Будем считать, что k = 10

Задача №13

- Сохраните результаты игр для игроков с рейтингами 2500, 2200, 1900, 1600, 1300
- Создайте набор из новых пяти игроков со стартовыми рейтингами, равными 1600
- Будем считать, что k = 10
- Обновляйте для новых игроков рейтинги согласно сохранённым результатам игр
- Какие рейтинги получились в результате?
- А если k = 1?
- А если k = 100?

Задача №14

- Сохраните результаты игр для игроков с рейтингами 2500, 2200, 1900, 1600, 1300
- Создайте набор из новых пяти игроков со стартовыми рейтингами, равными 1600
- Будем считать, что k = 10
- Обновляйте для новых игроков рейтинги согласно сохранённым результатам игр, но при этом не разыгрывайте игры, если номера игроков отличаются более чем на 1 (т.е. первый играет только со вторым, второй только с третьим и т.д.)
- Какие рейтинги получились в результате?
- Как изменилось общее количество игр?

Что мы сделали на самом деле

- Рейтинги по результатам попарных игр – ранжирование из попарных оценок
- Обновление рейтингов – stochastic gradient descent
- Обновление рейтингов с игнорированием части данных
  - Рекомендации и коллаборативная фильтрация
  - Dropout

Спасибо!

https://github.com/ashagraev/ml_basics

https://habr.com/users/ashagraev/posts/

https://t.me/insilicio

https://www.facebook.com/ashagraev

https://vk.com/shagraev