Открытый курс по машинному обучению

Автор материала: программист-исследователь Mail.ru Group, старший преподаватель Факультета Компьютерных Наук ВШЭ Юрий Кашницкий

Домашнее задание № 8. Часть 2

Vowpal Wabbit в задаче классификации тегов вопросов на Stackoverflow

План 2 части домашнего задания

2.1. Введение
2.2. Описание данных
2.3. Предобработка данных
2.4. Обучение и проверка моделей
2.5. Заключение

2.1. Введение

В этом задании вы будете делать примерно то же, что я каждую неделю – в Mail.ru Group: обучать модели на выборке в несколько гигабайт. Задание можно выполнить и на Windows с Python, но я рекомендую поработать под *NIX-системой (например, через Docker) и активно использовать язык bash. Немного снобизма (простите, но правда): если вы захотите работать в лучших компаниях мира в области ML, вам все равно понадобится опыт работы с bash под UNIX.

Веб-форма для ответов.

Для выполнения задания понадобится установленный Vowpal Wabbit (уже есть в докер-контейнере курса, см. инструкцию в README репозитория нашего курса) и примерно 70 Гб дискового пространства. Я тестировал решение не на каком-то суперкомпе, а на Macbook Pro 2015 (8 ядер, 16 Гб памяти), и самая тяжеловесная модель обучалась около 12 минут, так что задание реально выполнить и с простым железом. Но если вы планируете когда-либо арендовать сервера Amazon, можно попробовать это сделать уже сейчас.

Материалы в помощь:

- интерактивный тьюториал CodeAcademy по утилитам командной строки UNIX (примерно на час-полтора)
- статья про то, как арендовать на Amazon машину (еще раз: это не обязательно для выполнения задания, но будет хорошим опытом, если вы это делаете впервые)

2.2. Описание данных

Имеются 10 Гб вопросов со StackOverflow – скачайте эту выборку.

Формат данных простой:

текст вопроса (слова через пробел) TAB *теги вопроса* (через пробел)

Здесь TAB – это символ табуляции. Пример первой записи в выборке:

In [6]:
```
!head -1 stackoverflow.10kk.tsv
```

```
 is there a way to apply a background color through css at the tr level i can
apply it at the td level like this my td background color e8e8e8 background e
8e8e8 however the background color doesn t seem to get applied when i attempt
to apply the background color at the tr level like this my tr background colo
r e8e8e8 background e8e8e8 is there a css trick to making this work or does c
ss not natively support this for some reason    css css3 css-selectors
```

In [7]:
```
!ls
```

```
hw8_part1_komarov.ipynb

hw8_part2_vw_stackoverflow_tags_10mln.ipynb

stackoverflow.10kk.tsv

stackoverflow.10kk.tsv.gz

topic8_sgd_hashing_vowpal_wabbit.ipynb
```

Здесь у нас текст вопроса, затем табуляция и теги вопроса: *css, css3* и *css-selectors*. Всего

в выборке таких вопросов 10 миллионов.

In [8]:
```
%%time
!wc -l stackoverflow.10kk.tsv
```

```
10000000 stackoverflow.10kk.tsv
CPU times: user 1.04 s, sys: 240 ms, total: 1.28 s
Wall time: 48.9 s
```

Обратите внимание на то, что такие данные я уже не хочу загружать в оперативную память и, пока можно, буду пользоваться эффективными утилитами UNIX – head, tail, wc, cat, cut и прочими.

2.3. Предобработка данных

Давайте выберем в наших данных все вопросы с тегами *javascript, java, python, ruby, php, c++, c#, go, scala* и *swift* и подготовим обучающую выборку в формате Vowpal Wabbit. Будем решать задачу 10-классовой классификации вопросов по перечисленным тегам.

Вообще, как мы видим, у каждого вопроса может быть несколько тегов, но мы упростим себе задачу и будем у каждого вопроса выбирать один из перечисленных тегов либо игнорировать вопрос, если таковых тегов нет. Но вообще VW поддерживает multilabel classification (аргумент --multilabel_oaa).

Реализуйте в виде отдельного файла `preprocess.py` код для подготовки данных. Он должен отобрать строки, в которых есть перечисленные теги, и переписать их в отдельный файл в формат Vowpal Wabbit. Детали:

- скрипт должен работать с аргументами командной строки: с путями к файлам на входе и на выходе
- строки обрабатываются по одной (можно использовать tqdm для подсчета числа итераций)
- если табуляций в строке нет или их больше одной, считаем строку поврежденной и пропускаем
- в противном случае смотрим, сколько в строке тегов из списка *javascript, java, python, ruby, php, c++, c#, go, scala* и *swift*. Если ровно один, то записываем строку в выходной файл в формате VW: `label | text`, где `label` – число от 1 до 10 (1 - *javascript*, ... 10 – *swift*). Пропускаем те строки, где интересующих тегов больше или меньше одного
- из текста вопроса надо выкинуть двоеточия и вертикальные палки, если они есть – в VW это спецсимволы

In [9]:
```
import os
from tqdm import tqdm
from time import time
import numpy as np
from sklearn.metrics import accuracy_score
```

Должно получиться вот такое число строк – 4389054. Как видите, 10 Гб у меня обработались примерно за полторы минуты.

In [10]:
```
!python preprocess.py stackoverflow.10kk.tsv stackoverflow.vw
```

```
python: can't open file 'preprocess.py': [Errno 2] No such file or directory
```

Поделите выборку на обучающую, проверочную и тестовую части в равной пропорции - по

1463018 в каждый файл. Перемешивать не надо, первые 1463018 строк должны пойти в обучающую часть `stackoverflow_train.vw`, последние 1463018 – в тестовую `stackoverflow_test.vw`, оставшиеся – в проверочную `stackoverflow_valid.vw`.

Также сохраните векторы ответов для проверочной и тестовой выборки в отдельные файлы `stackoverflow_valid_labels.txt` и `stackoverflow_test_labels.txt`.

Тут вам помогут утилиты head, `tail`, `split`, `cat` и `cut`.

In [ ]:
```
''' ВАШ КОД ЗДЕСЬ '''
```

2.4. Обучение и проверка моделей

Обучите Vowpal Wabbit на выборке `stackoverflow_train.vw` 9 раз, перебирая параметры passes (1,3,5), ngram (1,2,3). Остальные параметры укажите следующие: bit_precision=28 и seed=17. Также скажите VW, что это 10-классовая задача.

Проверяйте долю правильных ответов на выборке `stackoverflow_valid.vw`. Выберите лучшую модель и проверьте качество на выборке `stackoverflow_test.vw`.

In [ ]:
```
''' ВАШ КОД ЗДЕСЬ '''
```

Вопрос 1. Какое сочетание параметров дает наибольшую долю правильных ответов на проверочной выборке `stackoverflow_valid.vw`?

- Биграммы и 3 прохода по выборке
- Триграммы и 5 проходов по выборке

- Биграммы и 1 проход по выборке
- Униграммы и 1 проход по выборке

Проверьте лучшую (по доле правильных ответов на валидации) модель на тестовой выборке.

In [ ]:
```
''' ВАШ КОД ЗДЕСЬ '''
```

Вопрос 2. Как соотносятся доли правильных ответов лучшей (по доле правильных ответов на валидации) модели на проверочной и на тестовой выборках? (здесь % – это процентный пункт, т.е., скажем, снижение с 50% до 40% – это на 10%, а не 20%).

- На тестовой ниже примерно на 2%
- На тестовой ниже примерно на 3%
- Результаты почти одинаковы – отличаются меньше чем на 0.5%

Обучите VW с параметрами, подобранными на проверочной выборке, теперь на объединении обучающей и проверочной выборок. Посчитайте долю правильных ответов на тестовой выборке.

In [ ]:
```
''' ВАШ КОД ЗДЕСЬ '''
```

Вопрос 3. На сколько процентных пунктов повысилась доля правильных ответов модели после обучения на вдвое большей выборке (обучающая `stackoverflow_train.vw` + проверочная `stackoverflow_valid.vw`) по сравнению с моделью, обученной только на `stackoverflow_train.vw`?

- 0.1%
- 0.4%
- 0.8%
- 1.2%

2.5. Заключение

В этом задании мы только познакомились с Vowpal Wabbit. Что еще можно попробовать:

- Классификация с несколькими ответами (multilabel classification, аргумент `multilabel_oaa`) – формат данных тут как раз под такую задачу
- Настройка параметров VW с hyperopt, авторы библиотеки утверждают, что качество должно сильно зависеть от параметров изменения шага градиентного спуска (`initial_t` и `power_t`). Также можно потестировать разные функции потерь – обучать логистическую регресиию или линейный SVM
- Познакомиться с факторизационными машинами и их реализацией в VW (аргумент `lrq`)