

# Vowpal Wabbit

Евгений Соколов  
sokolov.evg@gmail.com

30 Октября 2015

- очень качественная реализация стохастического градиентного спуска для линейных моделей
- считывает с диска по одному объекту и делает шаг только по нему, нет необходимости хранить выборку в памяти
- может быть запущен на кластере
- нормализация признаков, взвешивание объектов, адаптивный градиентный шаг
- матричные разложения, тематическое моделирование, активное обучение, обучение с подкреплением
- разнообразие методов оптимизации: сопряженные градиенты, квазиньютоновские методы (L-BFGS)

Одна строка — один объект:

```
123 10 | 1:0.43 5:2.1 age:20 some raw text here
```

- 123 — целевая переменная
- 10 — вес объекта (можно не указывать, по умолчанию 1)
- name:value — описание признака
  - если name — строка, то она хэшируется (см. Hashing Trick)
  - по умолчанию value=1
  - если признак не описан для данного объекта, то он считается равным нулю

Признаки можно разделять на группы:

```
123 10 |integer 1:0.43 5:2.1 age:20 |text some raw  
text here age:120
```

- integer и text — два пространства признаков
- в обоих пространствах есть признак age, так можно

Пусть выборка записана в файле `train.txt`.

Обучение:

```
vw -d train.txt --passes 10 -c -f model.vw
```

- `-d filename` — имя входного файла
- `--passes n` — количество проходов по выборке
- `-c` — включает кэширование, позволяет ускорить все проходы после первого
- `-f filename` — имя файла с моделью

Как получить прогноз:

```
vw -d test.txt -i model.vw -t -p predictions.txt
```

- `-d filename` — имя входного файла
- `-i filename` — имя файла с моделью
- `-t` — режим применения существующей модели
- `-p filename` — имя файла с прогнозами

Поддерживаемые функционалы (`--loss_function`):

- squared

$$\frac{1}{2}(y - a(x))^2$$

- classic — quadratic без перевзвешивания объектов
- quantile

$$\tau(a(x) - y)[y \leq a(x)] + (1 - \tau)(y - a(x))[y \geq a(x)]$$

- logistic

$$\log(1 + \exp(-ya(x)))$$

- hinge

$$\max(0, 1 - ya(x))$$

К функционалу можно добавить регуляризаторы:

- `--l1 coef`
- `--l2 coef`

Можно обучать SVM:

```
vw -d train.txt -f svm.vw --loss_function hinge --l2  
0.1
```



Градиентный шаг:

$$w^{(t+1)} = w^{(t)} - \alpha_t \nabla Q(x_{i_t}).$$

Как выбирать  $\alpha_t$ ?

$$\alpha_t = s \left( \frac{i}{i+t} \right)^p,$$

где

- -l s
- --initial\_t i
- --power\_t p

Эти параметры сильно влияют на качество!

- `-b n`: логарифм количества возможных значений хэш-функции для hashing trick
- `-q ab`: генерирует все парные признаки, где первый признак берется из пространств с именем « $a^*$ », второй — из « $b^*$ »
- `--cubic abc`: тройки признаков
- `--ngram an`: генерирует  $n$ -граммы для пространств « $a^*$ »
- `--skips ak`: разрешает делать пропуски длины  $k$  в  $n$ -граммах пространств « $a^*$ »

Читайте документацию!

- `--holdout_after`: ранний останов, использует отложенную выборку
- `--bfgs`: квазиньютоновская оптимизация, должна работать лучше
- `--ksvm`: SVM с kernel trick