Линейная регрессия

Елена Кантонистова

elena.kantonistova@yandex.ru

ВШЭ, 2023

ПЛАН ЗАНЯТИЯ

- Модификации градиентного спуска
- Переобучение и регуляризация

МОДИФИКАЦИИ ГРАДИЕНТНОГО СПУСКА

МЕТОД ГРАДИЕНТНОГО СПУСКА

На каждом шаге (на каждой итерации метода) движемся в сторону антиградиента функции потери!

Метод градиентного спуска можно записать в векторном виде:

- Инициализируем веса $\boldsymbol{w}^{(0)}$.
- На каждом следующем шаге обновляем веса по формуле:

$$\boldsymbol{w}^{(\boldsymbol{k})} = \boldsymbol{w}^{(\boldsymbol{k}-\mathbf{1})} - \boldsymbol{\nabla Q}(\boldsymbol{w}^{(\boldsymbol{k}-\mathbf{1})})$$

В формулу обычно добавляют параметр $\boldsymbol{\eta}$ **–** величина градиентного шага (learning rate). Он отвечает за скорость движения в сторону антиградиента:

$$\boldsymbol{w}^{(\boldsymbol{k})} = \boldsymbol{w}^{(\boldsymbol{k}-\mathbf{1})} - \boldsymbol{\eta\nabla Q}(\boldsymbol{w}^{(\boldsymbol{k}-\mathbf{1})})$$

ГРАДИЕНТНЫЙ СПУСК

Градиент функции Q вычисляется как сумма градиентов функции потерь $q_i(w)$ по всем объектам:

$$\nabla Q(w) = \sum_{i=1}^{l} \nabla q_i(w)$$

Градиентный спуск:

$$w^{(k)} = w^{(k-1)} - \eta \sum_{i=1}^{l} \nabla q_i(w^{(k-1)})$$

Скорость сходимости: $\boldsymbol{Q\left(w^{(k)}\right) - Q(w^{*}) = O(\frac{1}{k})}$

МЕТОДЫ ОЦЕНИВАНИЯ ГРАДИЕНТА: SGD

Stochastic gradient descent (SGD):

- на каждом шаге выбираем ***один случайный объект*** и сдвигаемся в сторону антиградиента по этому объекту:

$$w^{(k)} = w^{(k-1)} - \eta_k \cdot \boldsymbol{\nabla q_{i_k}\left(w^{(k-1)}\right)}$$

Скорость сходимости: $\mathrm{E}\left[\boldsymbol{Q}\left(\boldsymbol{w}^{(k)}\right) - \boldsymbol{Q}(\boldsymbol{w}^*)\right] = \boldsymbol{O}\left(\frac{\mathbf{1}}{\sqrt{\boldsymbol{k}}}\right)$

МЕТОДЫ ОЦЕНИВАНИЯ ГРАДИЕНТА: SGD

Stochastic gradient descent (SGD):

- на каждом шаге выбираем один случайный объект и сдвигаемся в сторону антиградиента по этому объекту:

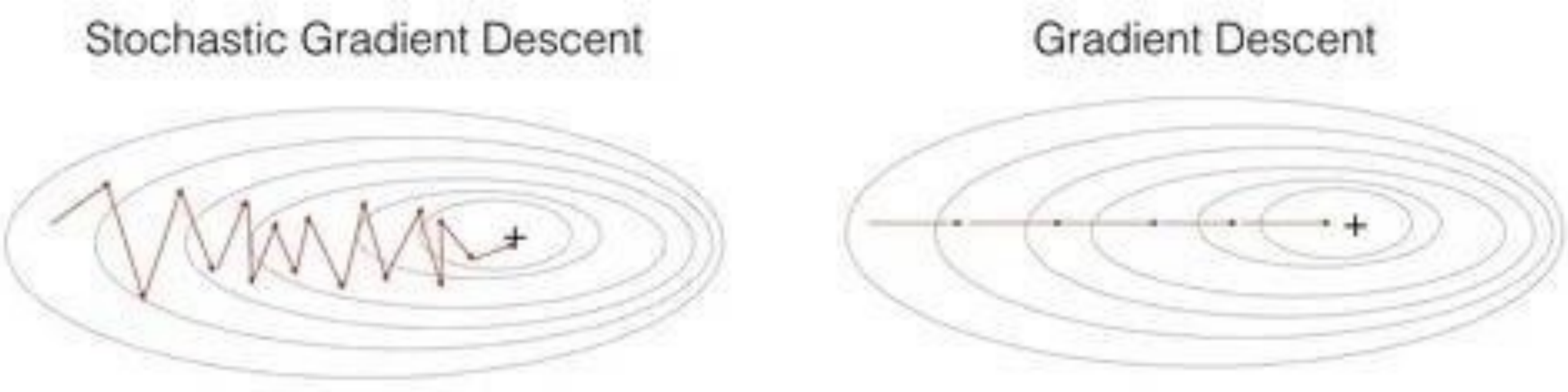
$$w^{(k)} = w^{(k-1)} - \eta_k \cdot \nabla q_{i_k}\left(w^{(k-1)}\right)$$

Скорость сходимости: $\mathrm{E}[\boldsymbol{Q}\left(\boldsymbol{w}^{(k)}\right) - \boldsymbol{Q}(\boldsymbol{w}^{*})] = \boldsymbol{O}(\frac{\mathbf{1}}{\sqrt{\boldsymbol{k}}})$

+ Менее трудоемкий метод

- Медленнее сходится

СТОХАСТИЧЕСКИЙ ГРАДИЕНТНЫЙ СПУСК



Если функция $Q(w)$ выпуклая и гладкая, а также имеет минимум в точке w^*, то метод стохастического градиентного спуска при аккуратно подобранном η через некоторое число шагов гарантированно попадет в малую окрестность точки w^*. Однако, сходится метод медленнее, чем обычный градиентный спуск

MINI-BATCH GRADIENT DESCENT

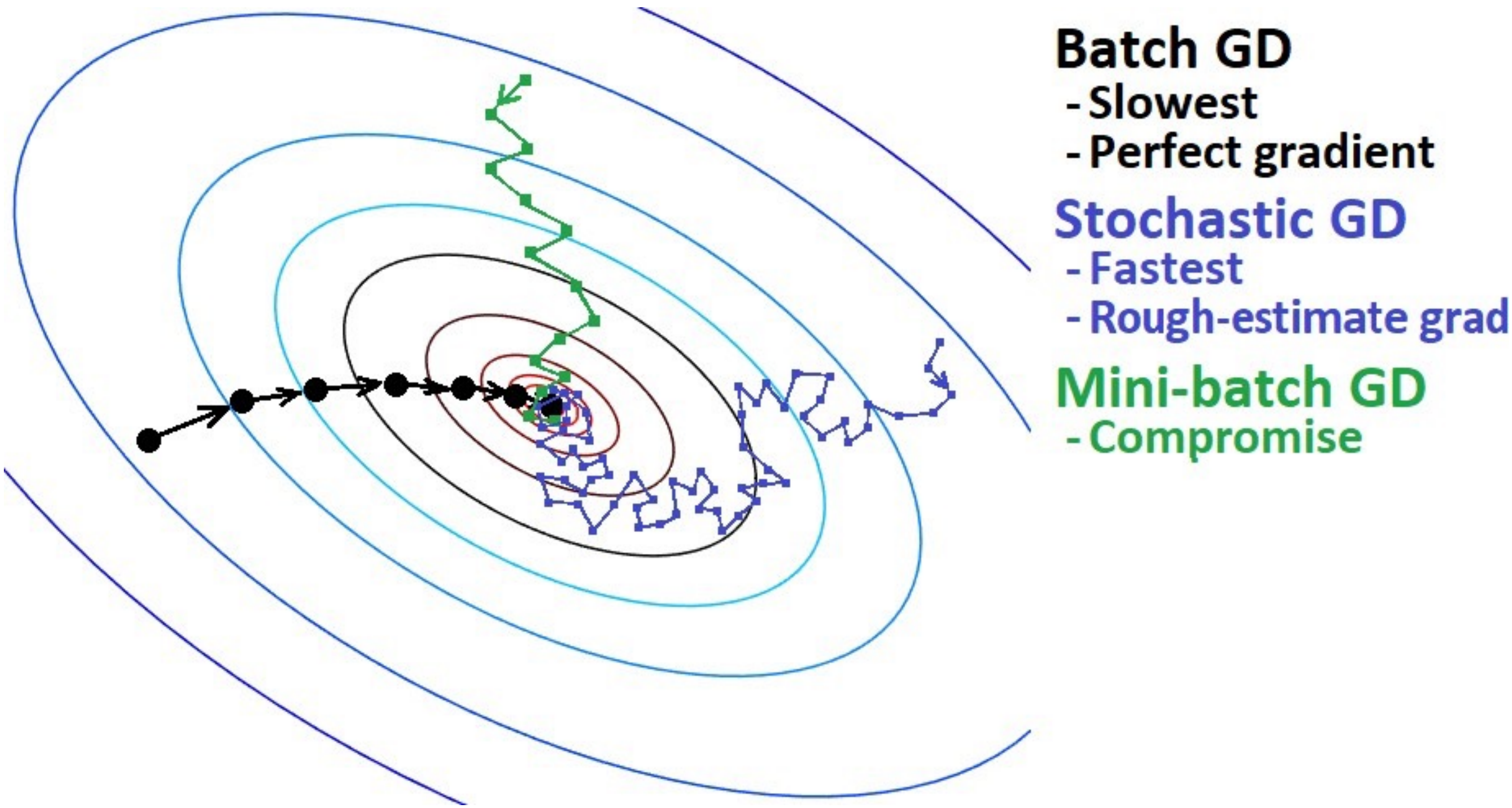
Промежуточное решение между классическим градиентным спуском и стохастическим вариантом.

- Выбираем *batch size* (например, 32, 64 и т.д.). Разбиваем все объекты на группы размера *batch size*.
- На i-й итерации градиентного спуска вычисляем $\nabla Q(w)$ только по объектам i-го батча:

$$w^{(k)} = w^{(k-1)} - \eta \cdot \boldsymbol{\nabla Q_i(w^{(k-1)})}$$

где $\nabla Q_i(w^{(k-1)})$ - градиент функции потерь, вычисленный по объектам из i-го батча.

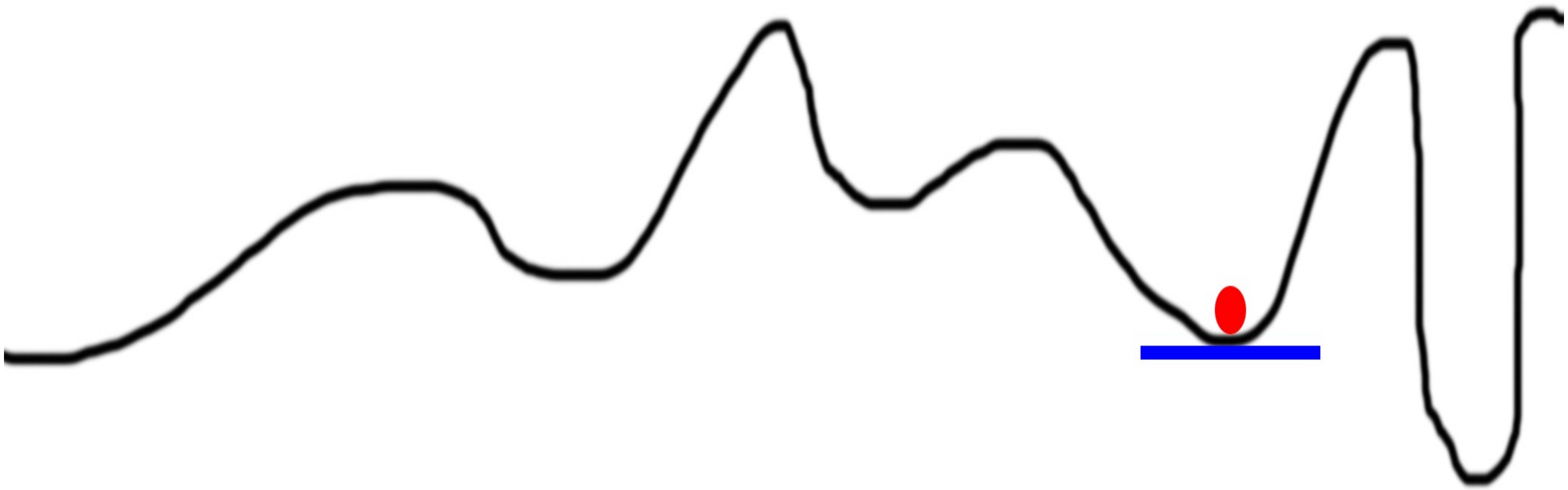
ВАРИАНТЫ ГРАДИЕНТНОГО СПУСКА



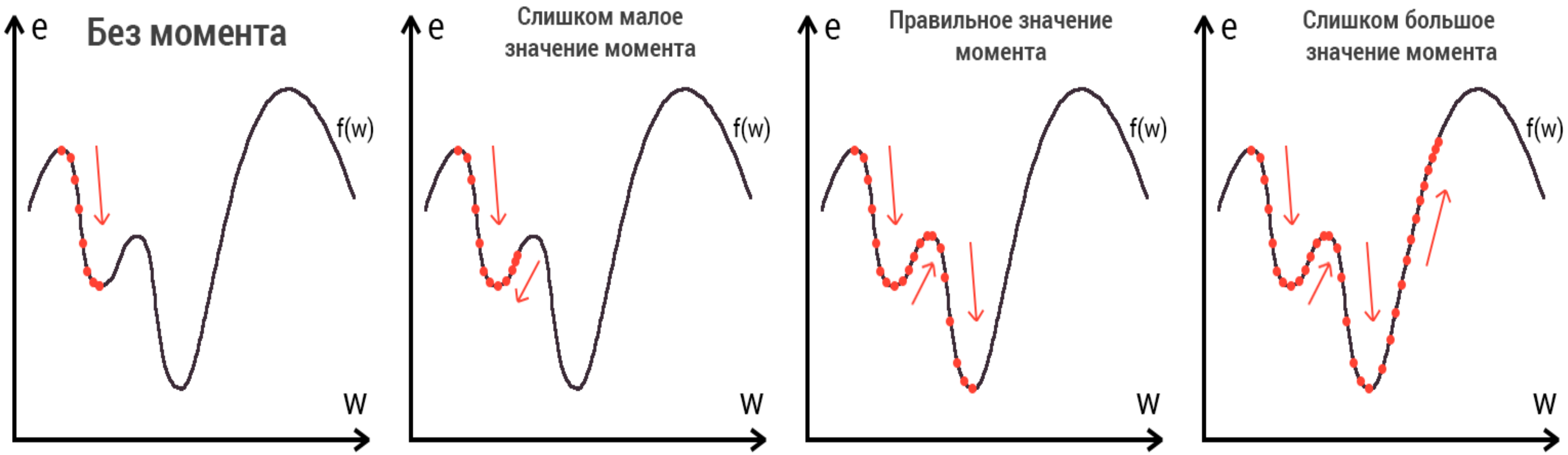
ПРОБЛЕМЫ ГРАДИЕНТНОГО СПУСКА

- Медленно сходится
- Застревает в локальных минимумах

ПРОБЛЕМА ЗАСТРЕВАНИЯ В LOCMIN



MOMENTUM


Без момента
Слишком малое значение момента
Правильное значение момента
Слишком большое значение момента
e
f(w)
w

МЕТОД МОМЕНТОВ (MOMENTUM)

Вектор инерции (*усреднение градиента по предыдущим шагам*):

$$h_0 = 0$$

$$h_k = \boldsymbol{\alpha h_{k-1}} + \eta_k \nabla Q\left(w^{(k-1)}\right)$$

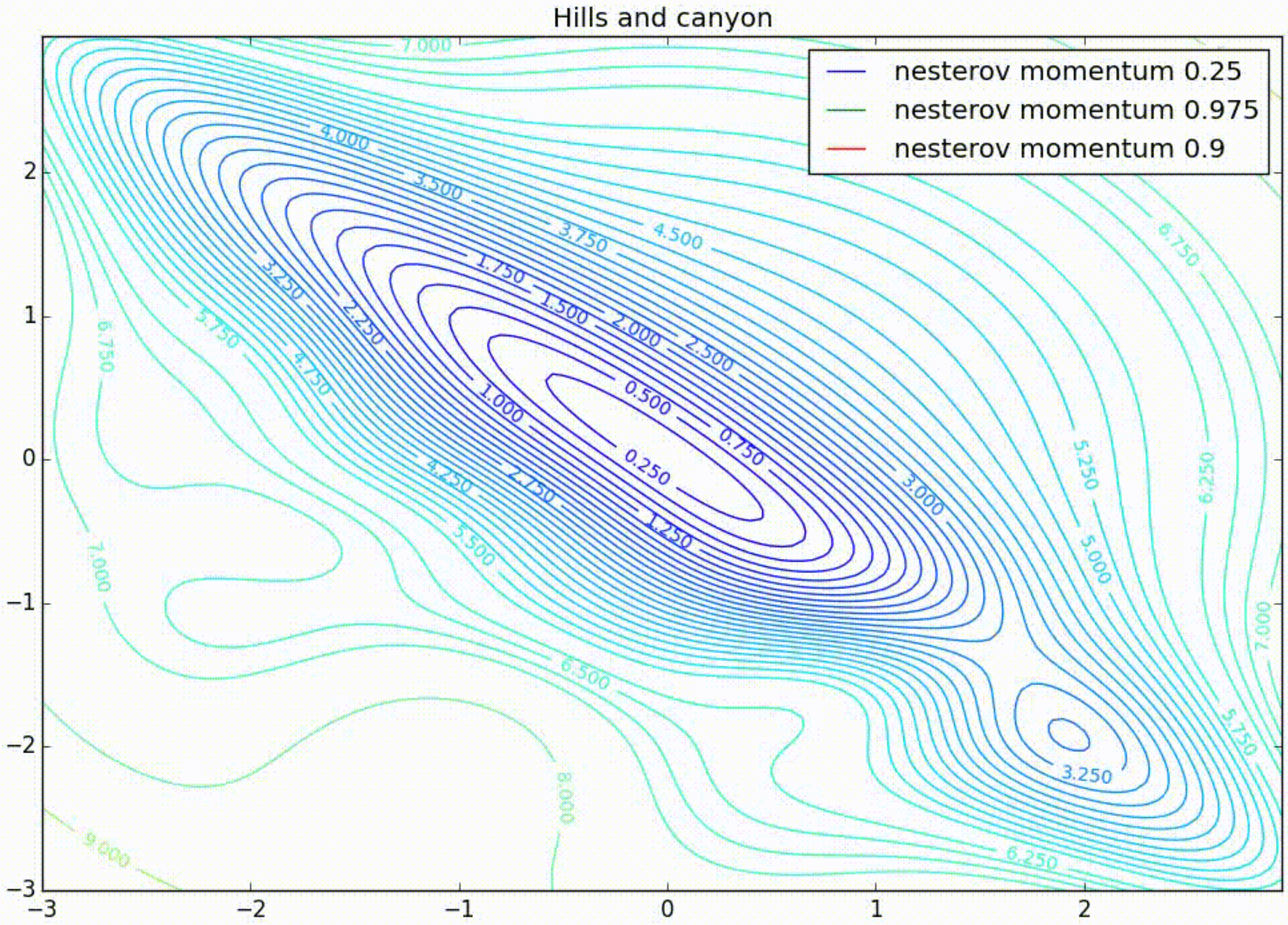
Формула метода моментов:

$$w^{(k)} = w^{(k-1)} - h_k$$

Подробнее:

$$w^{(k)} = w^{(k-1)} - \eta_k \nabla Q\left(w^{(k-1)}\right) - \boldsymbol{\alpha h_{k-1}}$$

MOMENTUM



Hills and canyon
nesterov momentum 0.25
nesterov momentum 0.975
nesterov momentum 0.9

ADAGRAD (ADAPTIVE GRADIENT)

Сумма квадратов обновлений:

$$g_{k-1,j} = (\nabla Q(w^{(k-1)}))_j^2$$

Формулы метода AdaGrad:

- $G_{k,j} = G_{k-1,j} + g_{k-1.j} = G_{k-1,j} + (\nabla Q(w^{(k-1)}))_j^2$
- $\omega_j^{(k)} = \omega_j^{k-1} - \frac{\boldsymbol{\eta}}{\sqrt{\boldsymbol{G_{k,j}+\varepsilon}}} \cdot \left(\nabla Q(w^{(k-1)})\right)_j$

Этот метод использует адаптивный шаг обучения – тем самым мы регулируем скорость сходимости метода.

ADAGRAD (ADAPTIVE GRADIENT)

Сумма квадратов обновлений:

$$g_{k-1,j} = (\nabla Q\left(w^{(k-1)}\right))_j^2$$

Формулы метода AdaGrad:

- $G_{k,j} = G_{k-1,j} + g_{k-1.j}$
- $\omega_j^{(k)} = \omega_j^{k-1} - \frac{\eta}{\sqrt{G_{k,j}+\varepsilon}} \cdot \left(\nabla Q\left(w^{(k-1)}\right)\right)_j$

+ Автоматическое затухание скорости обучения

- G_{kj} монотонно возрастают, поэтому шаги укорачиваются, и мы можем не успеть дойти до минимума

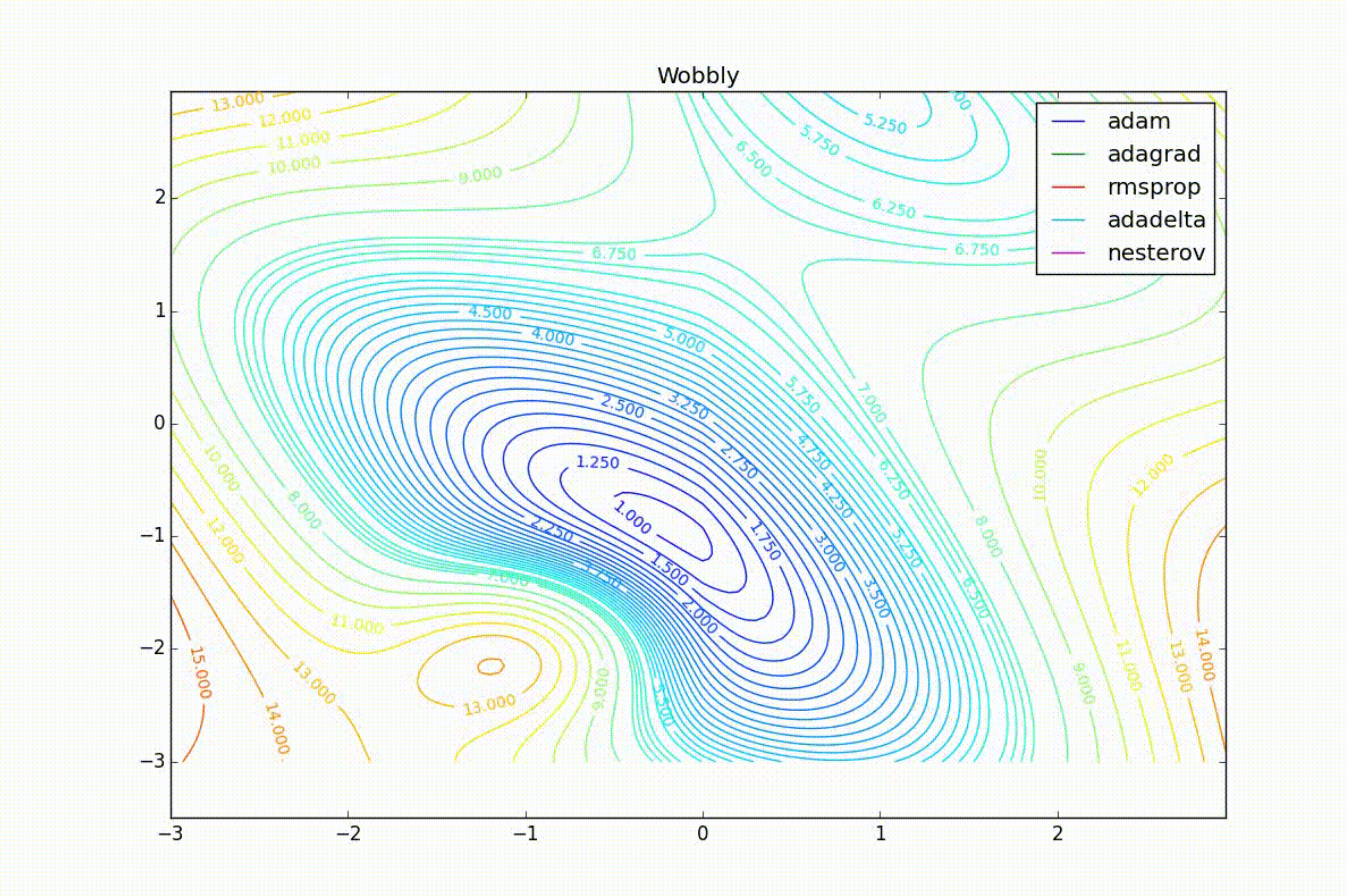
RMSPROP (ROOT MEAN SQUARE PROPAGATION)

Метод реализует экспоненциальное затухание градиентов

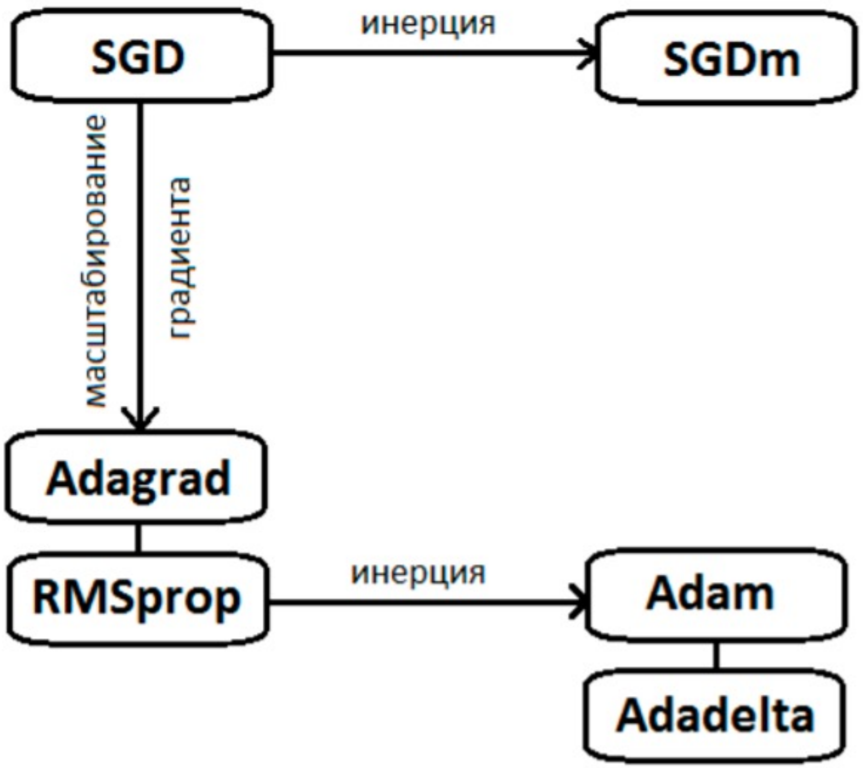
Формулы метода RMSprop (*усредненный по истории квадрат градиента*):

- $G_{k,j} = \boldsymbol{\alpha} \cdot G_{k-1,j} + (\mathbf{1} - \boldsymbol{\alpha}) \cdot g_{k-1,j}$
- $\omega_j^{(k)} = \omega_j^{k-1} - \frac{\eta}{\sqrt{G_{k,j} + \varepsilon}} \cdot \left(\nabla Q\left(w^{(k-1)}\right)\right)_j$

МОДИФИКАЦИИ ГРАДИЕНТНОГО СПУСКА



МОДИФИКАЦИИ SGD



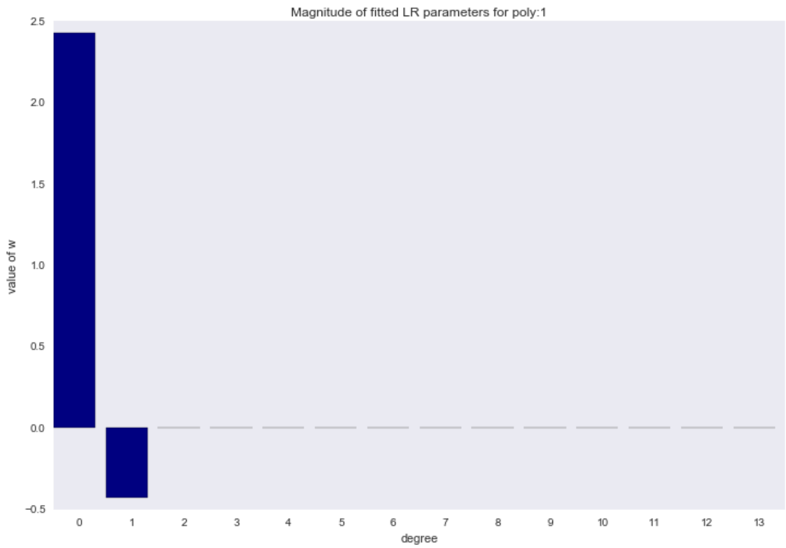
[ссылка на статью со схемой]

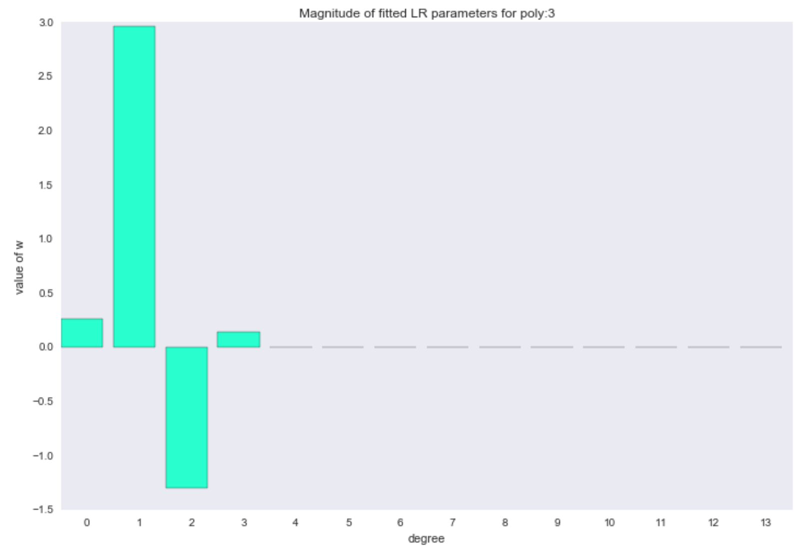
ПЕРЕОБУЧЕНИЕ И РЕГУЛЯРИЗАЦИЯ

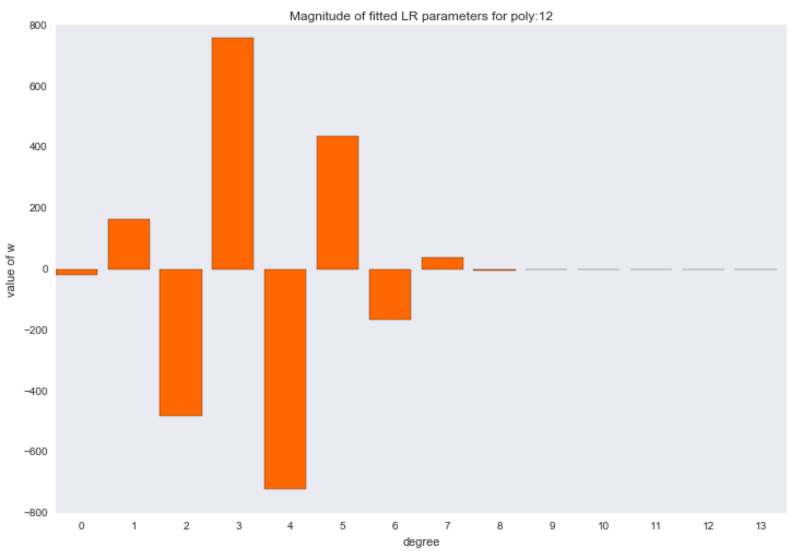
ПРИЗНАКИ ПЕРЕОБУЧЕННОЙ МОДЕЛИ

- Большая разница в качестве на тренировочных и тестовых данных (модель подгоняется под тренировочные данные и не может найти истинную зависимость)
- Большие значения параметров (весов) w_j модели

ПЕРЕОБУЧЕНИЕ: ПРИМЕР







МЕТОД БОРЬБЫ С ПЕРЕОБУЧЕНИЕМ: РЕГУЛЯРИЗАЦИЯ

Утверждение. Если в выборке есть линейно-зависимые признаки, то задача оптимизации $Q(w) \to min$ имеет бесконечное число решений.

- Большие значения параметров (весов) модели w – признак переобучения.

Решение проблемы – ***регуляризация***.

Будем минимизировать регуляризованный функционал ошибки:

$$\boldsymbol{Q_{alpha}(w) = Q(w) + \alpha \cdot R(w) \to \min_{w}} \quad ,$$

где $R(w)$ - регуляризатор.

РЕГУЛЯРИЗАЦИЯ

- **Регуляризация штрафует за слишком большие веса.**

Наиболее используемые регуляризаторы:

- L_2-регуляризатор: $R(w) = \left||w|\right|_2 = \sum_{i=1}^{d} w_i^2$
- L_1-регуляризатор: $R(w) = \left||w|\right|_1 = \sum_{i=1}^{d} |w_i|$

РЕГУЛЯРИЗАЦИЯ

- **Регуляризация штрафует за слишком большие веса.**

Наиболее используемые регуляризаторы:

- L_2-регуляризатор: $R(w) = \left||w|\right|_2 = \sum_{i=1}^{d} w_i^2$
- L_1-регуляризатор: $R(w) = \left||w|\right|_1 = \sum_{i=1}^{d} |w_i|$

Пример регуляризованного функционала:

$$Q(a(w), X) = \frac{1}{l}\sum_{i=1}^{l}((w, x_i) - y_i)^2 + \alpha \sum_{i=1}^{d} w_i^2 ,$$

где α – коэффициент регуляризации.

АНАЛИТИЧЕСКОЕ РЕШЕНИЕ ЗАДАЧИ МНК С L_2-РЕГУЛЯРИЗАТОРОМ

Задача оптимизации в матричном виде:

$$Q(w) = (y - Xw)^T(y - Xw) + \alpha w^T I w \to min \quad (*)$$

где I – единичная матрица.

Эта задача имеет аналитическое решение:

$$\boldsymbol{w = \left(X^T X + \alpha I\right)^{-1} X^T y}$$

- Матрица $X^T X + \alpha I$ всегда положительно определена, поэтому её можно обратить. Следовательно, задача (*) имеет единственное решение.

ПОЛЕЗНОЕ СВОЙСТВО L1-РЕГУЛЯРИЗАЦИИ

Все ли признаки в задаче нужны?

- Некоторые признаки могут не иметь отношения к задаче, т.е. они не нужны.
- Если есть ограничения на скорость получения предсказаний, то чем меньше признаков, тем быстрее
- Если признаков больше, чем объектов, то решение задачи будет неоднозначным.

Поэтому в таких случаях надо делать отбор признаков, то есть убирать некоторые признаки.

L_1-РЕГУЛЯРИЗАЦИЯ

Утверждение. В результате обучения модели с L_1-регуляризатором происходит зануление некоторых весов, т.е. отбор признаков.

Можно показать, что задачи

$$(1)\quad Q(w) + \alpha\left||w|\right|_1 \to \min_w$$

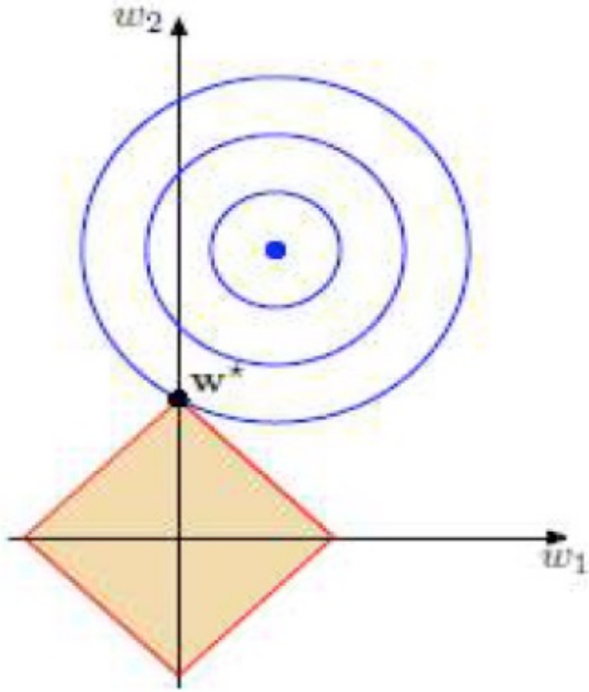
и

$$(2)\quad \begin{cases} Q(w) \to \min\limits_w \\ \left||w|\right|_1 \le C \end{cases}$$

эквивалентны.

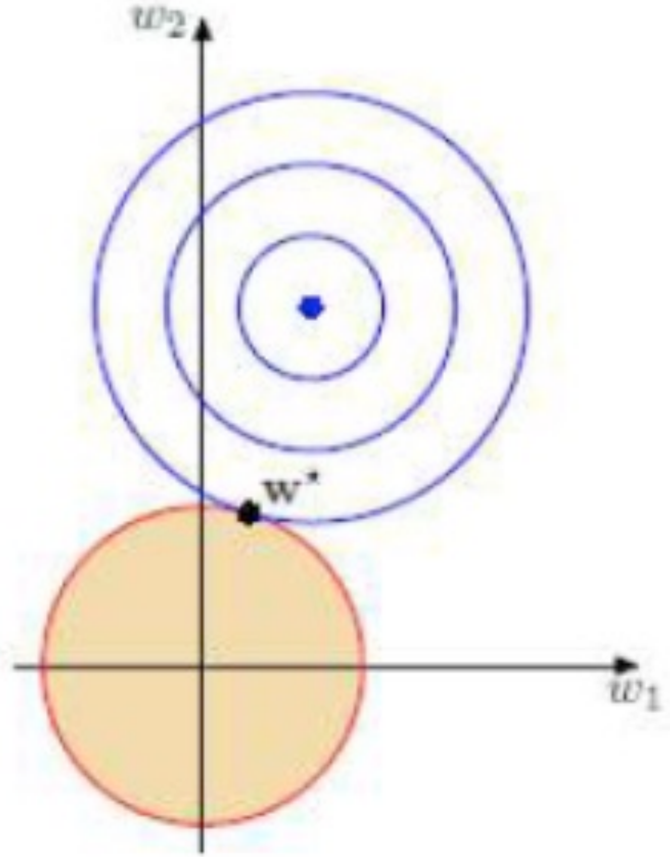
ОТБОР ПРИЗНАКОВ ПО L1-РЕГУЛЯРИЗАЦИИ

Нарисуем линии уровня $Q(w)$ и область $||w||_1 \leq C$:



Если признак незначимый, то соответствующий вес близок к 0. Отсюда получим, что в большинстве случаев решение нашей задачи попадает в вершину ромба, т.е. обнуляет незначимый признак.

L2-РЕГУЛЯРИЗАЦИЯ НЕ ОБНУЛЯЕТ ПРИЗНАКИ



РАЗРЕЖЕННЫЕ МОДЕЛИ

Модели, в которых часть весов равна 0, называются ***разреженными моделями***.

- L1-регуляризация зануляет часть весов, то есть делает модель разреженной.