Занятие 2
Линейные методы регрессии. Часть 1.

Елена Кантонистова

elena.kantonistova@yandex.ru

ВШЭ, 2021

ЛИНЕЙНАЯ РЕГРЕССИЯ

Пример (напоминание):

Предположим, что мы хотим предсказать *стоимость дома* y по его *площади* (x_1) и *количеству комнат* (x_2).

Линейная модель для предсказания стоимости:

$$\boldsymbol{a}(\boldsymbol{x}) = \boldsymbol{w_0} + \boldsymbol{w_1}\boldsymbol{x_1} + \boldsymbol{w_2}\boldsymbol{x_2},$$

где w_0, w_1, w_2 -

параметры модели (*веса*).

ЛИНЕЙНАЯ РЕГРЕССИЯ

Пример (напоминание):

Предположим, что мы хотим предсказать *стоимость дома* y по его *площади* (x_1) и *количеству комнат* (x_2).

Линейная модель для предсказания стоимости:

$$\boldsymbol{a(x) = w_0 + w_1 x_1 + w_2 x_2},$$

где w_0, w_1, w_2 -

параметры модели (*веса*).

Общий вид (линейная регрессия):

$$\boxed{\boldsymbol{a(x) = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n},}$$

где $x_1, \ldots, x_n$ - признаки объекта x.

ЛИНЕЙНАЯ РЕГРЕССИЯ

Линейная регрессия:

$$\boxed{a(x) = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n}$$

ЛИНЕЙНАЯ РЕГРЕССИЯ

Линейная регрессия:

$$a(x) = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n$$

- сокращенная запись:

$$a(x) = w_0 + \sum_{j=1}^{n} w_j x_j$$

ЛИНЕЙНАЯ РЕГРЕССИЯ

Линейная регрессия:

$$a(x) = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n$$

- сокращенная запись:

$$a(x) = w_0 + \sum_{j=1}^{n} w_j x_j$$

- запись через скалярное произведение (с добавлением признака $x_0 = 1$):

$$a(x) = w_0 \cdot 1 + \sum_{j=1}^{n} w_j x_j = \sum_{j=0}^{n} w_j x_j = (w, x)$$

ЛИНЕЙНАЯ РЕГРЕССИЯ

Линейная регрессия:

$$\boxed{a(x) = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n}$$

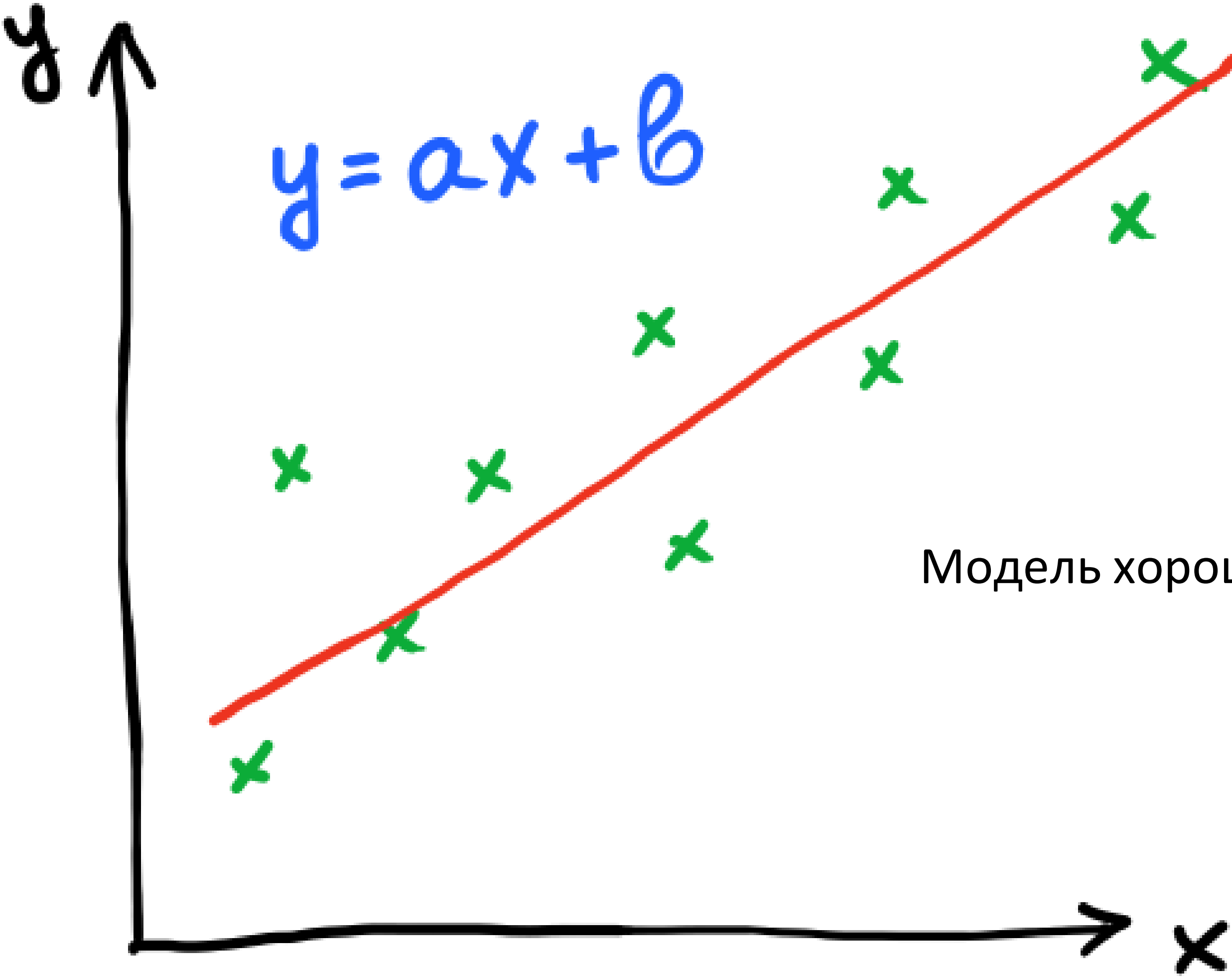
- сокращенная запись:

$$a(x) = w_0 + \sum_{j=1}^{n} w_j x_j$$

- запись через скалярное произведение (с добавлением признака $x_0 = 1$):

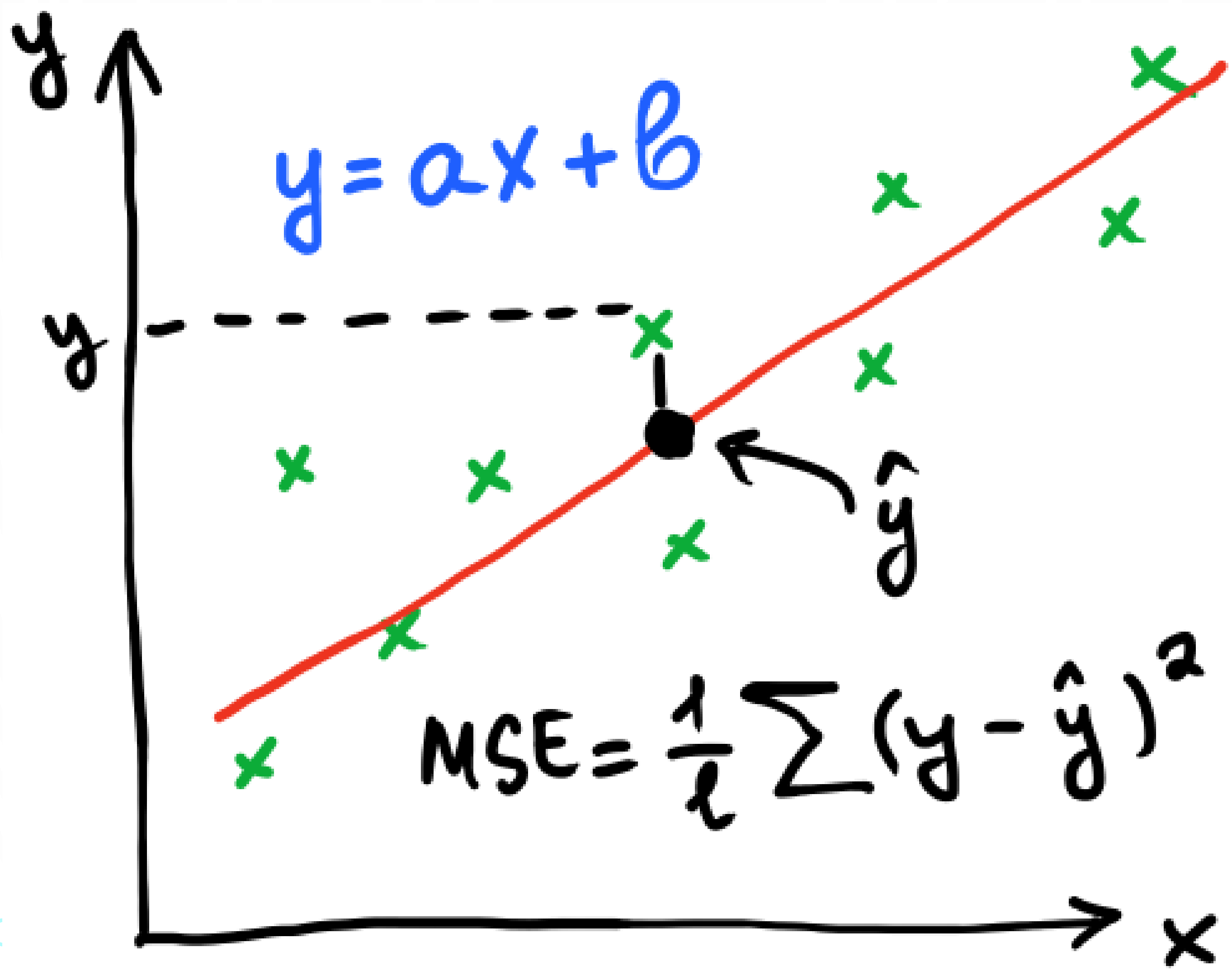
$$a(x) = w_0 \cdot 1 + \sum_{j=1}^{n} w_j x_J = \sum_{j=0}^{n} w_j x_j = (w, x) \leftrightarrow \boxed{a(x) = (w, x)}$$

ОБУЧЕНИЕ РЕГРЕССИИ



Модель хорошая?

ОБУЧЕНИЕ РЕГРЕССИИ



ЛИНЕЙНАЯ РЕГРЕССИЯ

Линейная регрессия:

$$a(x) = w_0 + \sum_{j=1}^{n} w_j x_j = (w, x)$$

Обучение линейной регрессии - минимизация среднеквадратичной ошибки:

$$\mathrm{Q}(a, X) = \frac{1}{l}\sum_{i=1}^{l}(a(x_i) - y_i)^2 = \frac{1}{l}\sum_{i=1}^{l}((w, x_i) - y_i)^2 \to \min_{w}$$

(здесь l – количество объектов)

Аналитическое решение задачи метода наименьших квадратов (МНК)

Задача обучения линейной регрессии (в матричной форме):

$$\frac{1}{\ell}\|Xw - y\|^2 \to \min_w$$

Точное (аналитическое) решение:

$$\boxed{w = (X^T X)^{-1} X^T y}$$

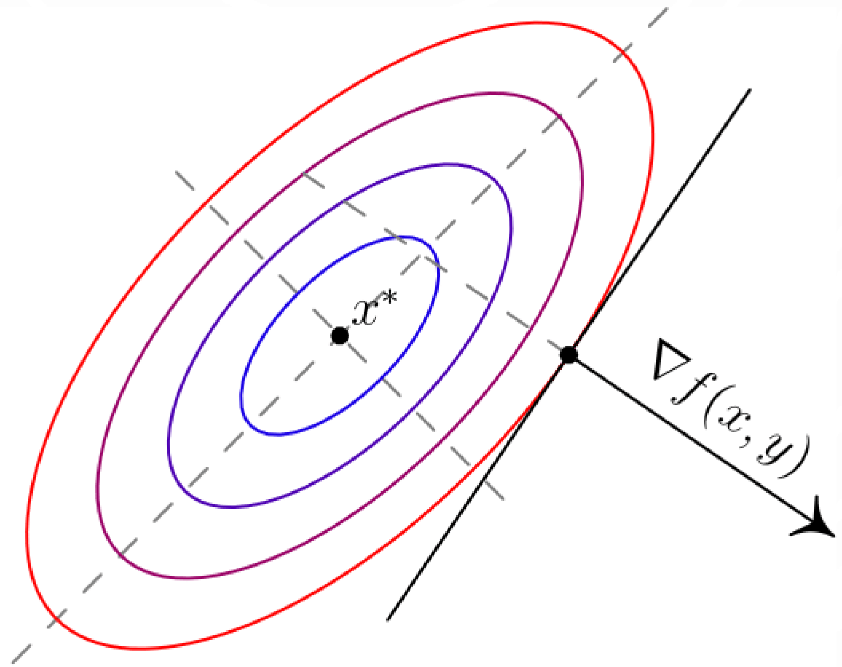
Недостатки аналитической формулы

- Обращение матрицы – сложная операция ($O(N^3)$ от числа признаков)
- Матрица $X^T X$ может быть вырожденной или плохо обусловленной
- Если заменить среднеквадратичный функционал ошибки на другой, то скорее всего не найдем аналитическое решение

ТЕОРЕМА О ГРАДИЕНТЕ

Теорема. Градиент – это вектор, в направлении которого функция быстрее всего растёт.

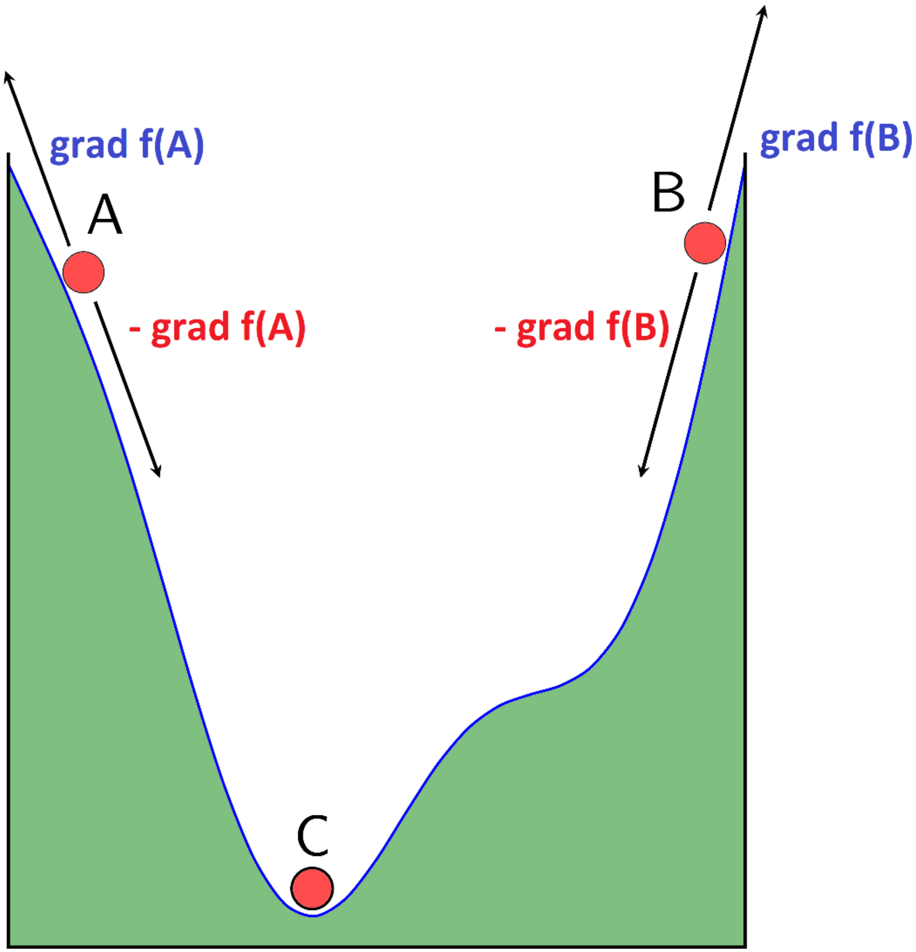
Антиградиент (вектор, противоположный градиенту) – вектор, в направлении которого функция быстрее всего убывает.



$$f(x,y) = \frac{3}{2}(x-y)^2 + \frac{1}{3}(x+y)^2$$

ТЕОРЕМА О ГРАДИЕНТЕ

Аантиградиент (вектор, противоположный градиенту) – вектор, в направлении которого функция быстрее всего убывает.



Метод градиентного спуска

- Наша задача при обучении модели – найти такие веса $\boldsymbol{w}$, на которых достигается **минимум функции ошибки**.

Метод градиентного спуска

- Наша задача при обучении модели – найти такие веса $\boldsymbol{w}$, на которых достигается минимум функции ошибки.
- В простейшем случае, если ошибка среднеквадратичная, то её график – это парабола.

Метод градиентного спуска

- Наша задача при обучении модели – найти такие веса $\boldsymbol{w}$, на которых достигается минимум функции ошибки.
- В простейшем случае, если ошибка среднеквадратичная, то её график – это парабола.
- Идея метода градиентного спуска:

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

То есть на каждом шаге движемся в направлении уменьшения ошибки.

Метод градиентного спуска

- Наша задача при обучении модели – найти такие веса $\boldsymbol{w}$, на которых достигается минимум функции ошибки.
- В простейшем случае, если ошибка среднеквадратичная, то её график – это парабола.
- Идея метода градиентного спуска:

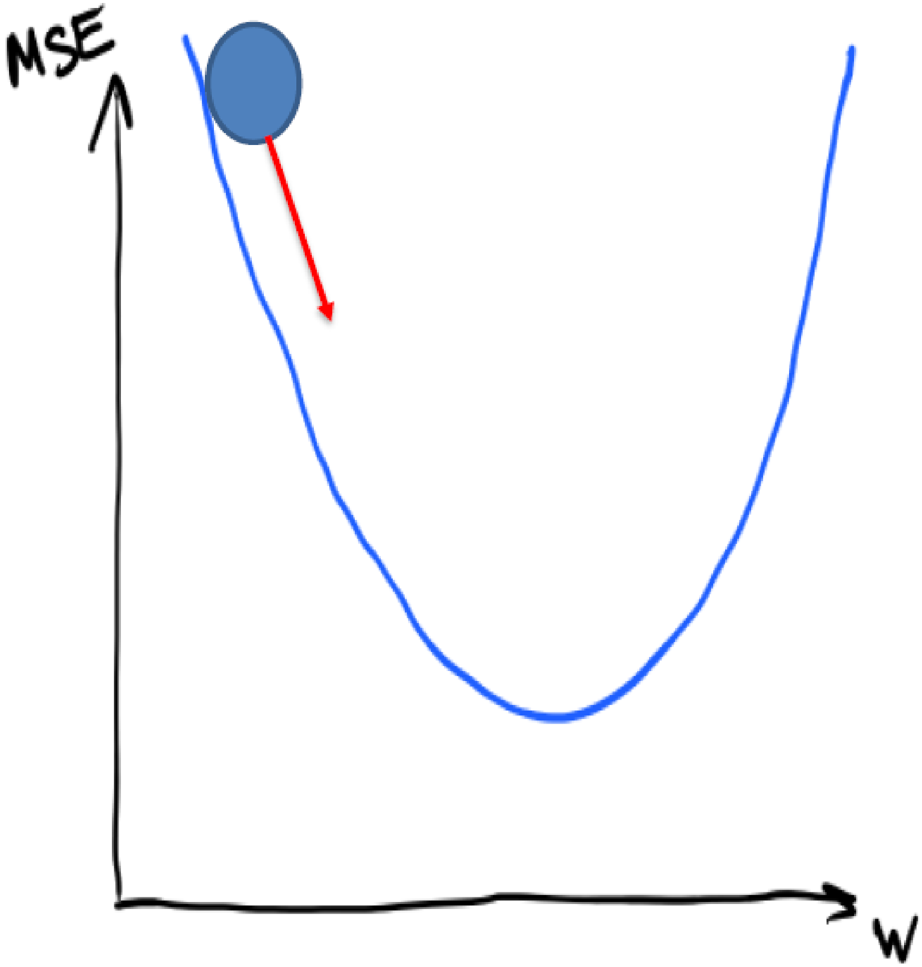
На каждом шаге (на каждой итерации метода) движемся в сторону антиградиента функции потерь!

То есть на каждом шаге движемся в направлении уменьшения ошибки.

Вектор градиента функции потерь обозначают $\boldsymbol{grad\ Q}$ *или* $\boldsymbol{\nabla Q}$.

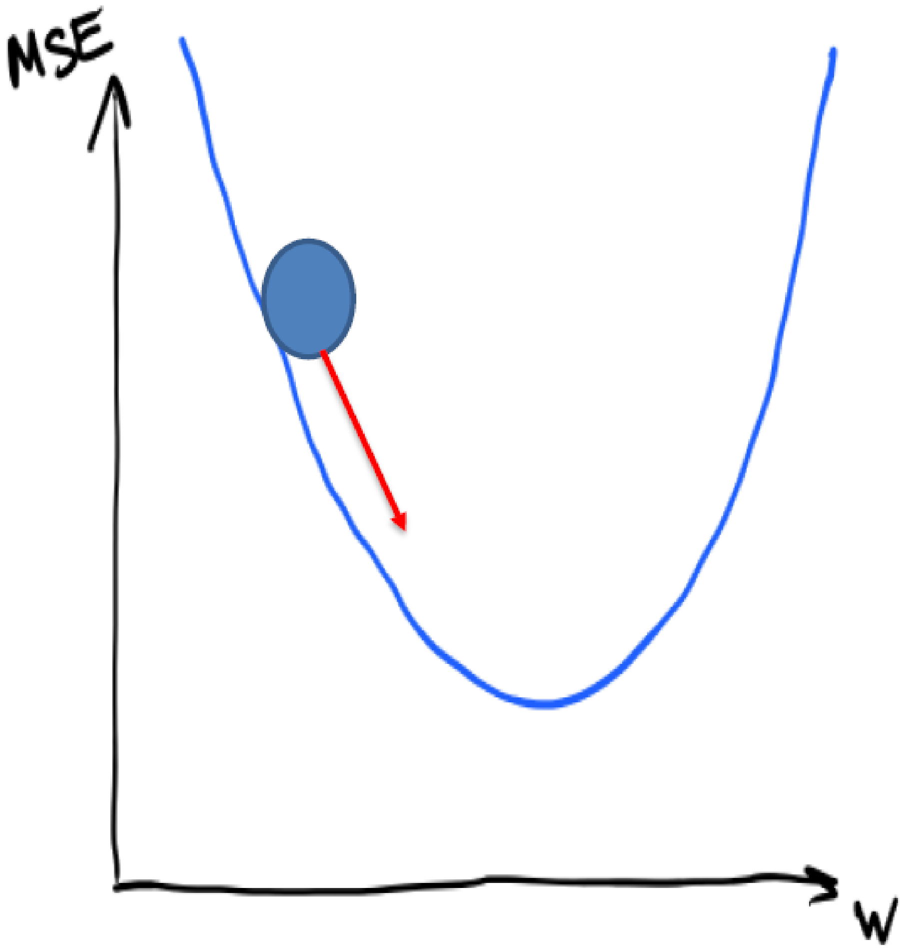
МЕТОД ГРАДИЕНТНОГО СПУСКА

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



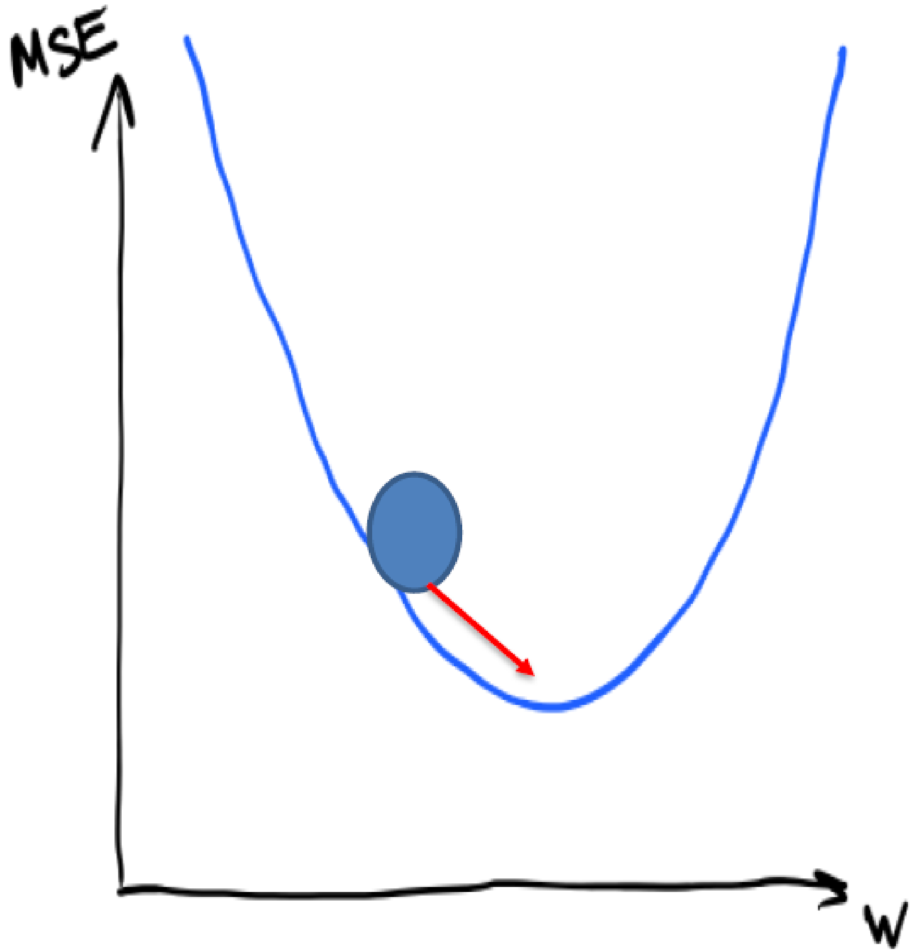
МЕТОД ГРАДИЕНТНОГО СПУСКА

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



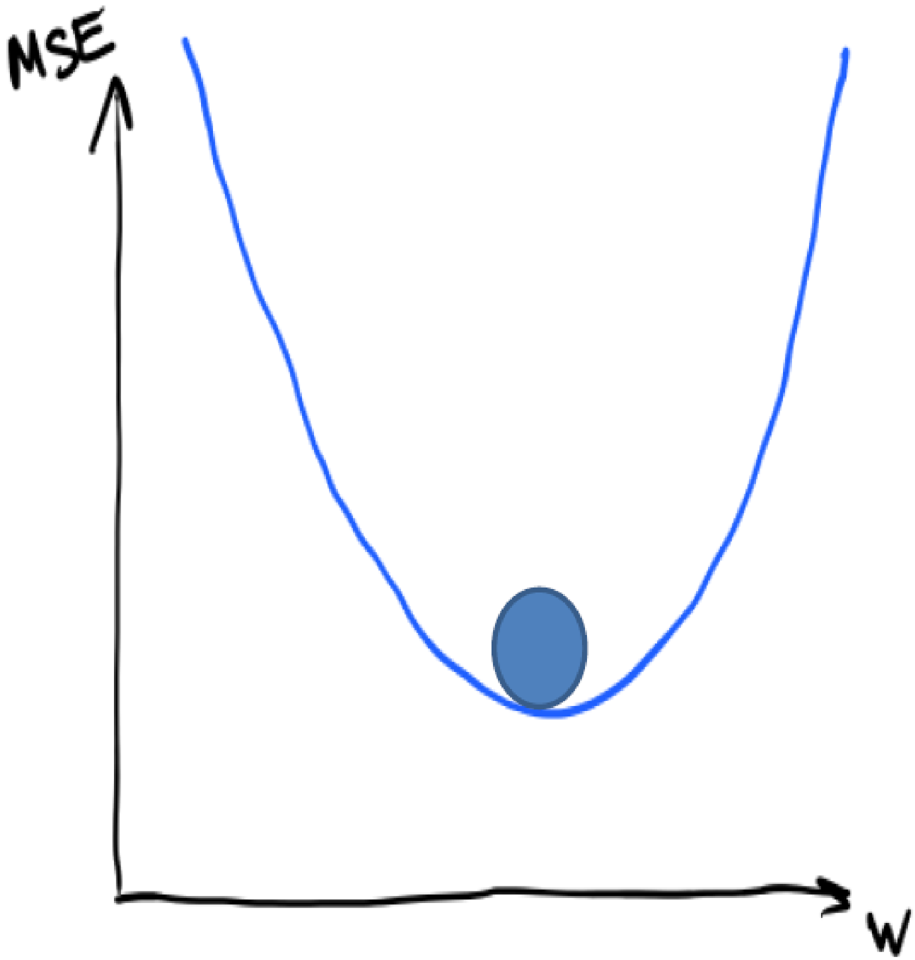
МЕТОД ГРАДИЕНТНОГО СПУСКА

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



МЕТОД ГРАДИЕНТНОГО СПУСКА

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



МЕТОД ГРАДИЕНТНОГО СПУСКА

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

Метод градиентного спуска:

- Инициализируем веса $w_0^{(0)}, w_1^{(0)}, w_2^{(0)}, \dots, w_n^{(0)}$.

МЕТОД ГРАДИЕНТНОГО СПУСКА

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

Метод градиентного спуска:

- Инициализируем веса $w_0^{(0)}, w_1^{(0)}, w_2^{(0)}, \dots, w_n^{(0)}$.
- На каждом следующем шаге обновляем веса, сдвигаясь в направлении антиградиента функции потерь Q:

$$w_0^{(k)} = w_0^{(k-1)} - \nabla Q\left(w_0^{(k-1)}\right),$$

МЕТОД ГРАДИЕНТНОГО СПУСКА

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

Метод градиентного спуска:

- Инициализируем веса $w_0^{(0)}, w_1^{(0)}, w_2^{(0)}, \dots, w_n^{(0)}$.
- На каждом следующем шаге обновляем веса, сдвигаясь в направлении антиградиента функции потерь Q:

$$w_0^{(k)} = w_0^{(k-1)} - \nabla Q\left(w_0^{(k-1)}\right),$$

$$w_1^{(k)} = w_1^{(k-1)} - \nabla Q\left(w_1^{(k-1)}\right),$$

$$\dots$$

$$w_n^{(k)} = w_n^{(k-1)} - \nabla Q\left(w_n^{(k-1)}\right),$$

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:

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

МЕТОД ГРАДИЕНТНОГО СПУСКА

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

Метод градиентного спуска можно записать в векторном виде:

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

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

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

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

ВАРИАНТЫ ИНИЦИАЛИЗАЦИИ ВЕСОВ

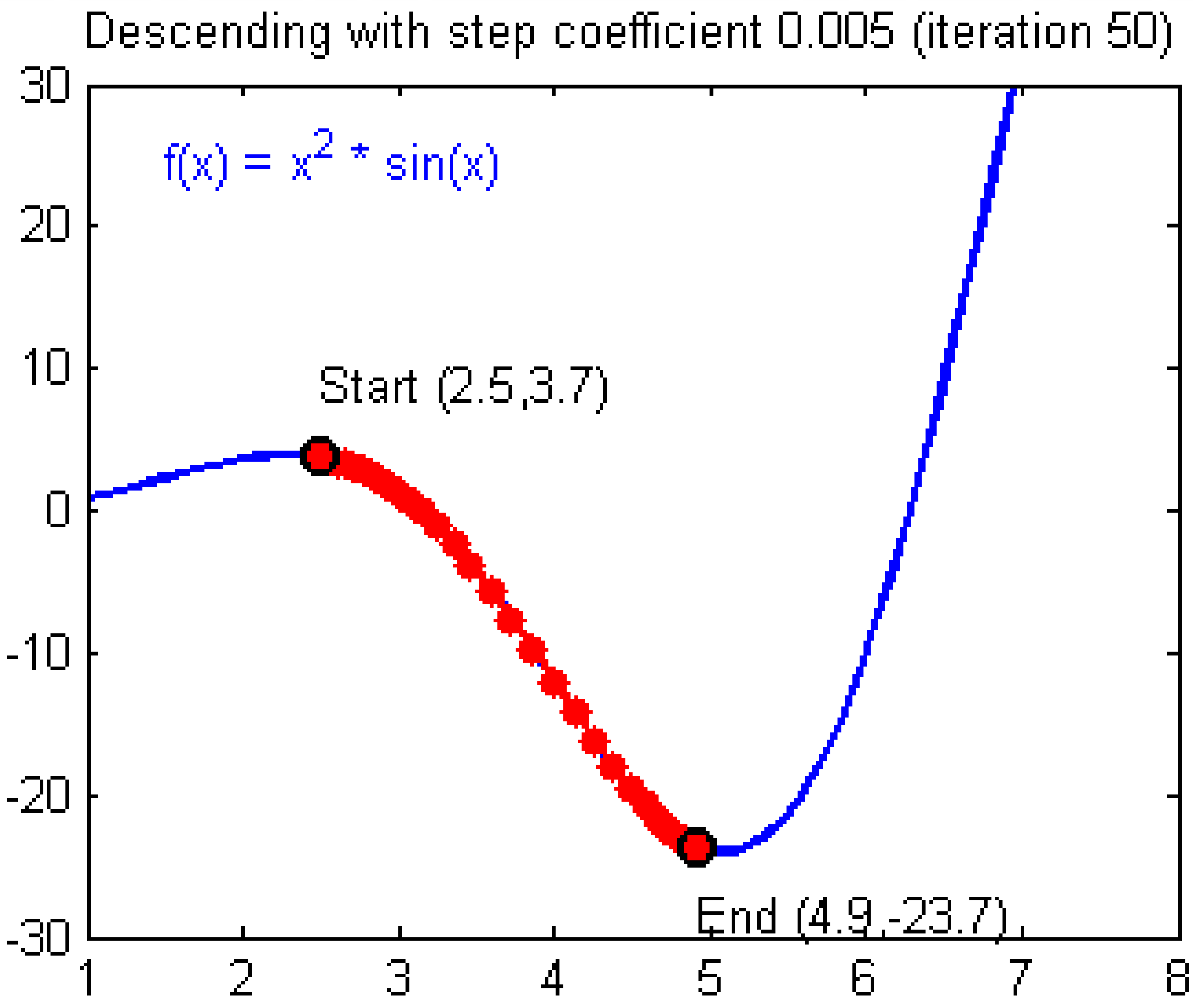
- $w_j = 0, j = 1, \dots, n$
- Небольшие случайные значения:
$$w_j := random(-\varepsilon, \varepsilon)$$
- Обучение по небольшой случайной подвыборке объектов
- Мультистарт: многократный запуск из разных случайных начальных приближений и выбор лучшего решения

Критерии останова

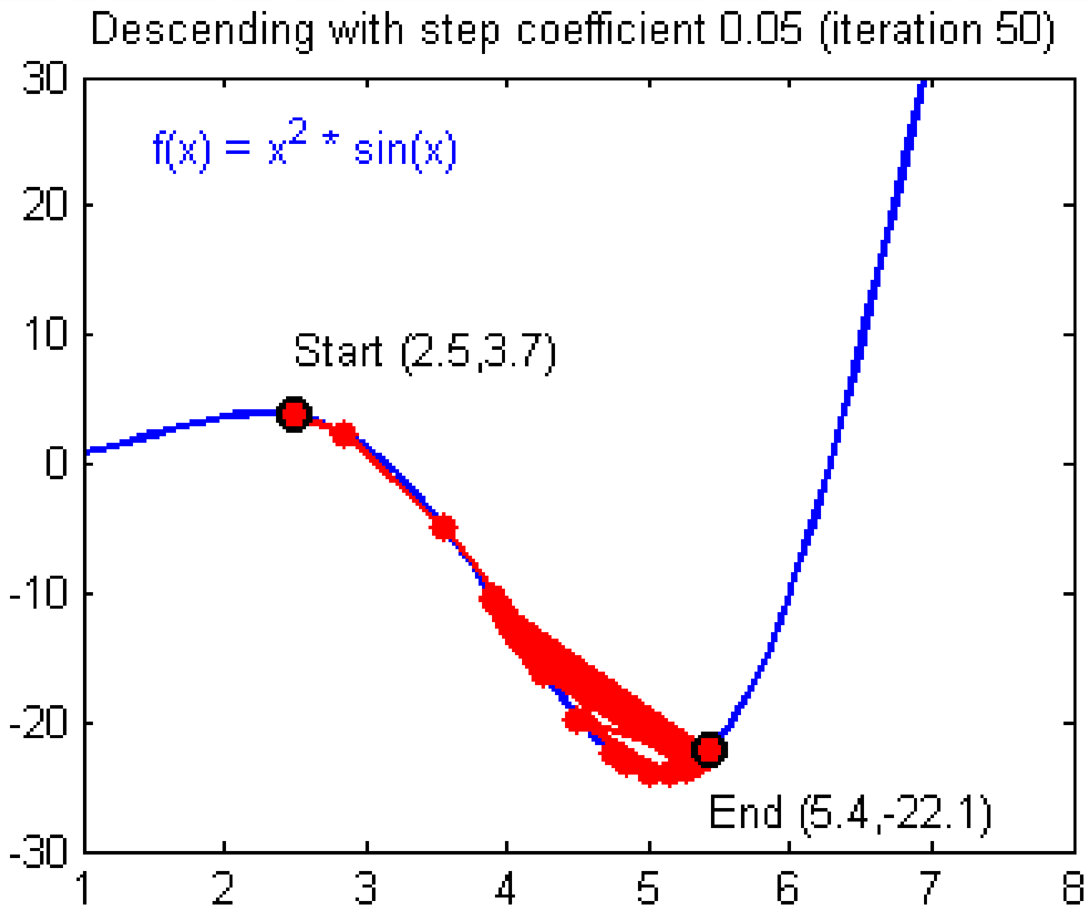
- $\left|Q\left(w^{(k)}\right) - Q\left(w^{(k-1)}\right)\right| < \varepsilon$

- $\|w^{(k)} - w^{(k-1)}\| < \varepsilon$

ГРАДИЕНТНЫЙ СПУСК



ПРОБЛЕМА ВЫБОРА ГРАДИЕНТНОГО ШАГА



ГРАДИЕНТНЫЙ ШАГ

В общем случае градиентный шаг может зависеть от номера итерации, тогда будем писать не η, а η_k.

- $\eta_k = c$
- $\eta_k = \frac{1}{k}$
- $\eta_k = \lambda \left(\frac{s_0}{s_0 + k} \right)^p$, λ, s_0, p - параметры

СТОХАСТИЧЕСКИЙ ГРАДИЕНТНЫЙ СПУСК

Stochastic gradient descent (SGD):

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

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

СТОХАСТИЧЕСКИЙ ГРАДИЕНТНЫЙ СПУСК

Stochastic gradient descent (SGD):

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

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

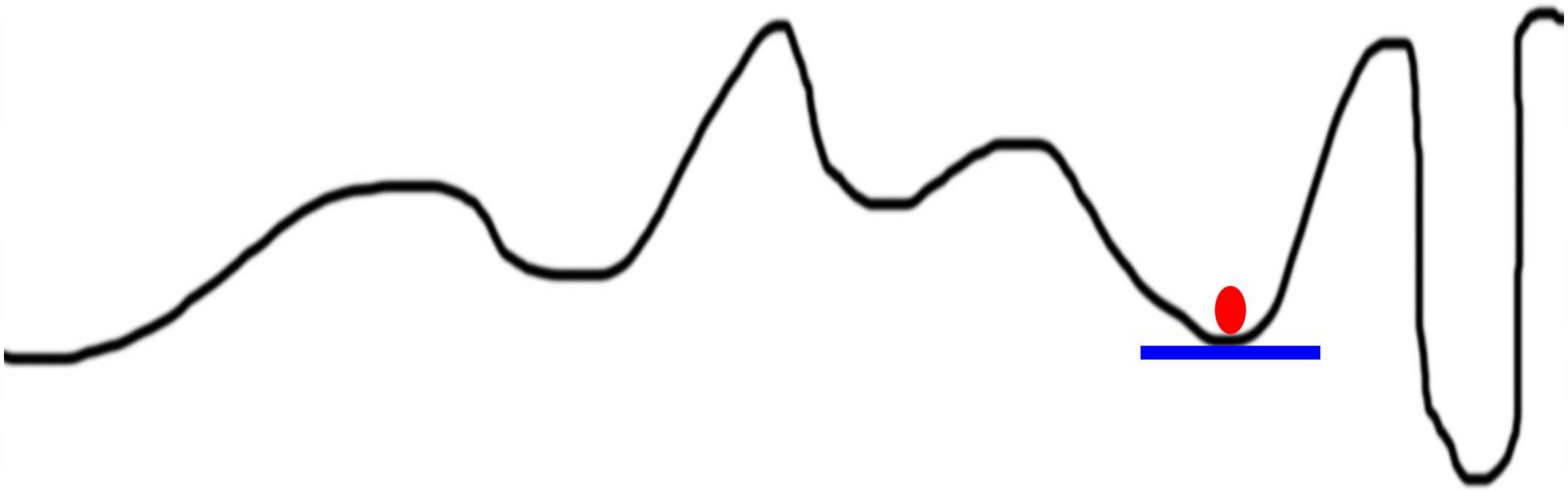
+ Менее трудоемкий метод

- Медленнее сходится

ПРОБЛЕМЫ ГРАДИЕНТНОГО СПУСКА

- Медленно сходится
- Застревает в локальных минимумах

ПРОБЛЕМА ЗАСТРЕВАНИЯ В LOCMIN



МЕТОД МОМЕНТОВ (MOMENTUM)

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

$$h_0 = 0$$

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

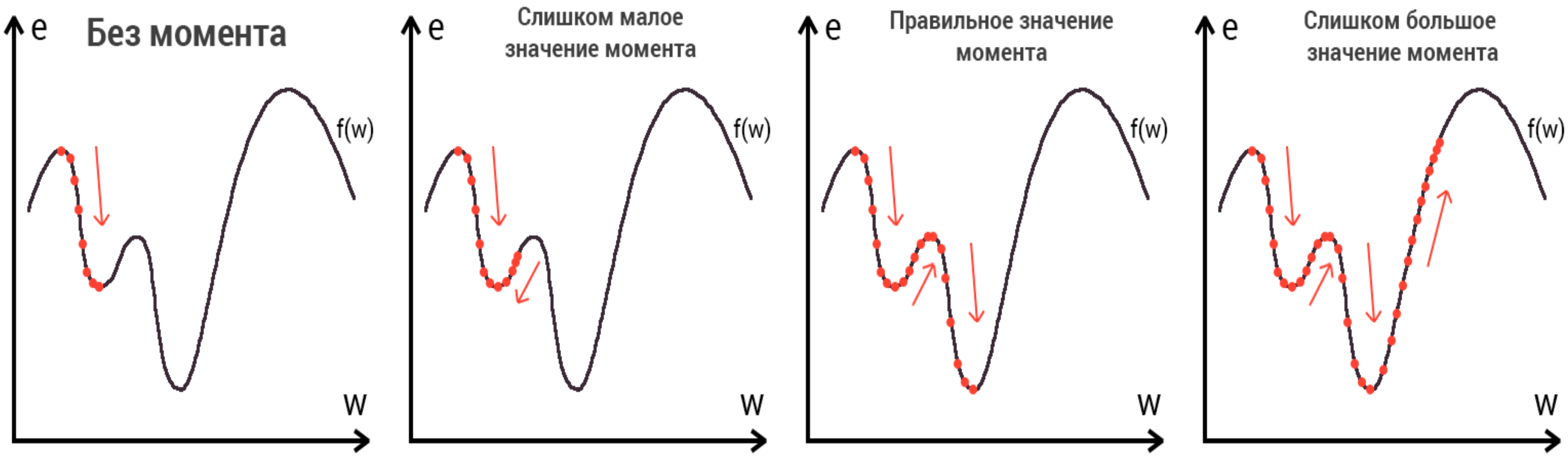
Формула метода моментов:

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

Подробнее:

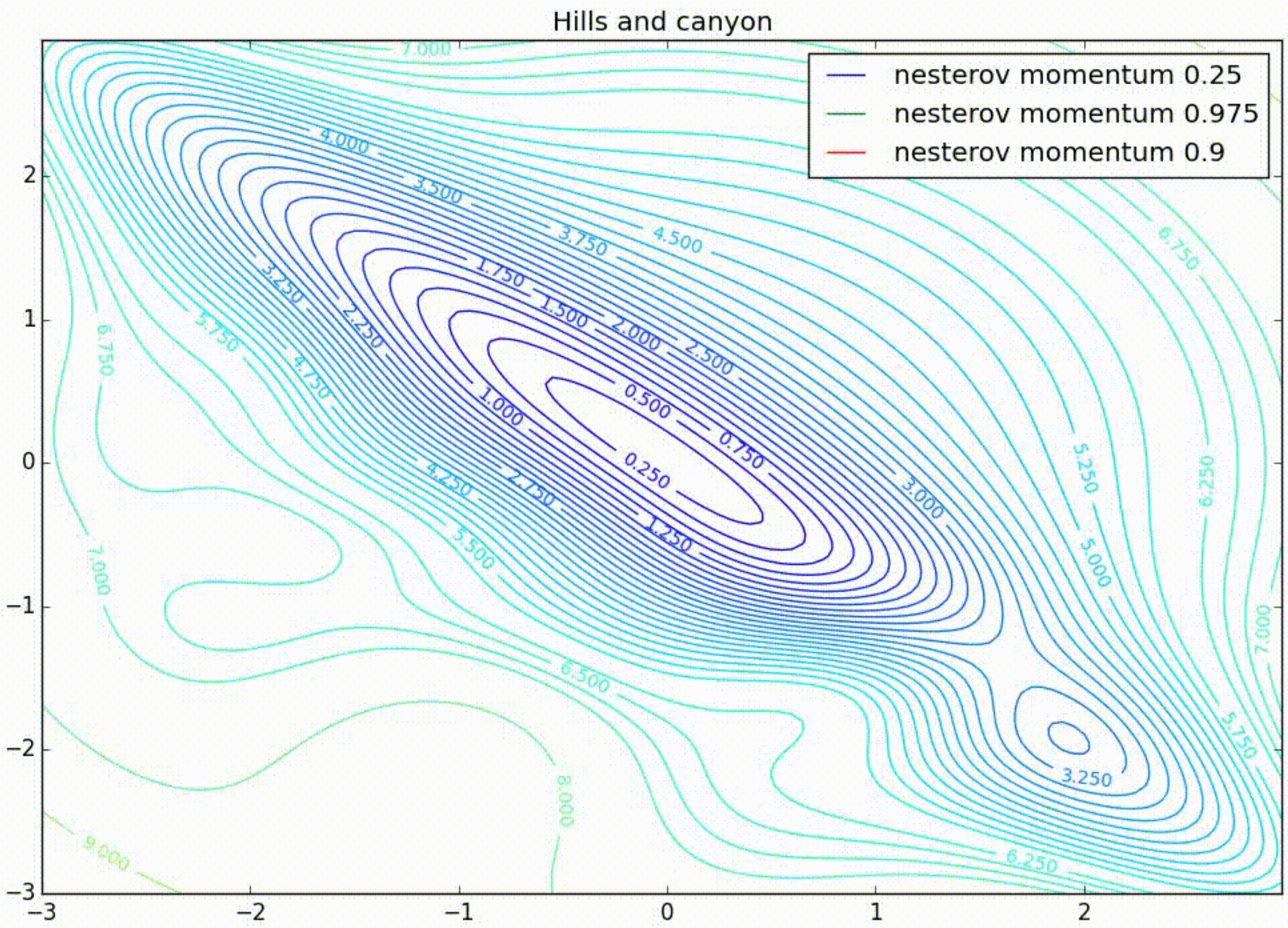
$$w^{(k)} = w^{(k-1)} - \eta_k \nabla Q\left(w^{(k-1)}\right) - \boldsymbol{\alpha h_{k-1}}$$

MOMENTUM

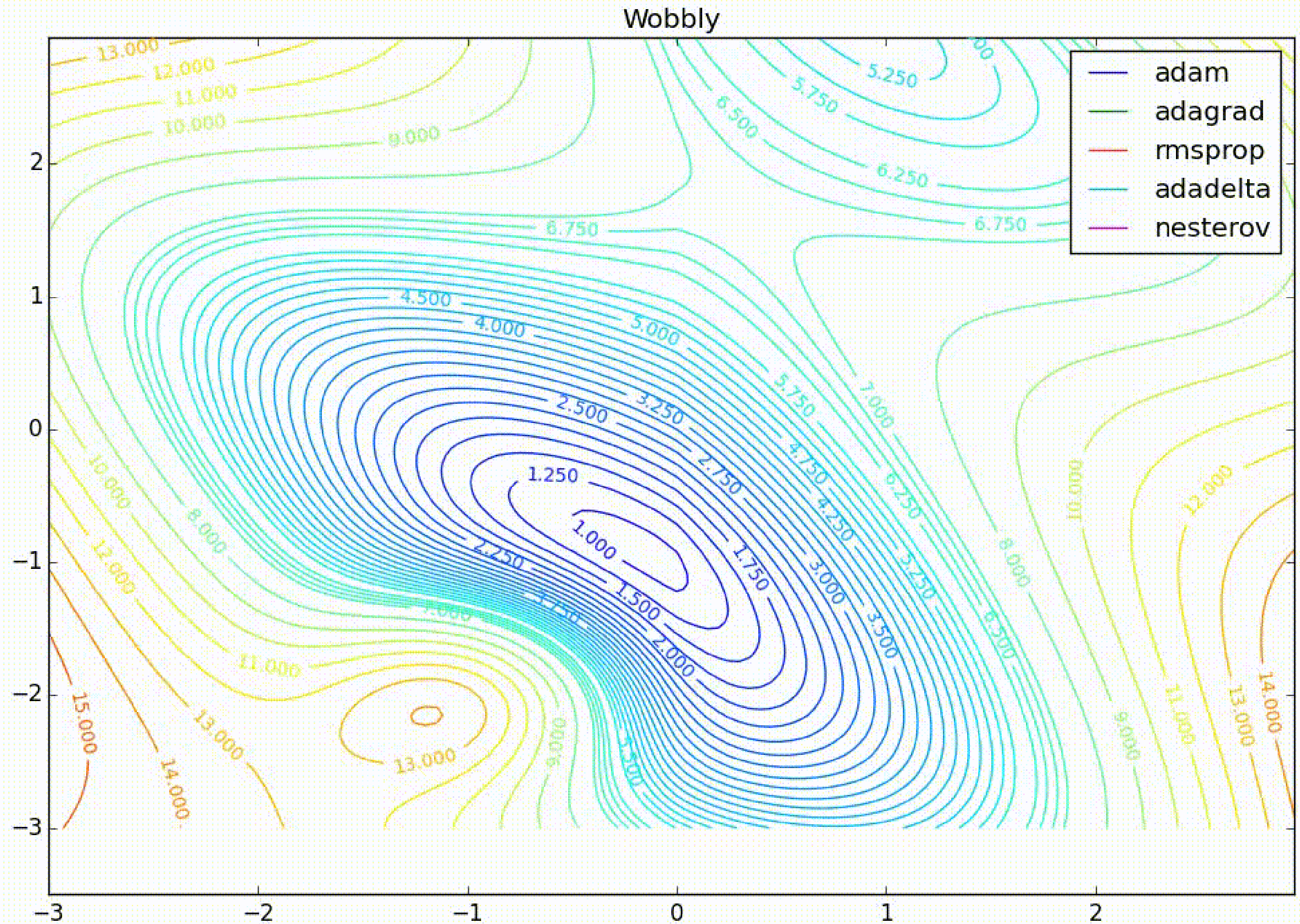


e
Без момента
f(w)
w
e
Слишком малое значение момента
f(w)
w
e
Правильное значение момента
f(w)
w
e
Слишком большое значение момента
f(w)
w

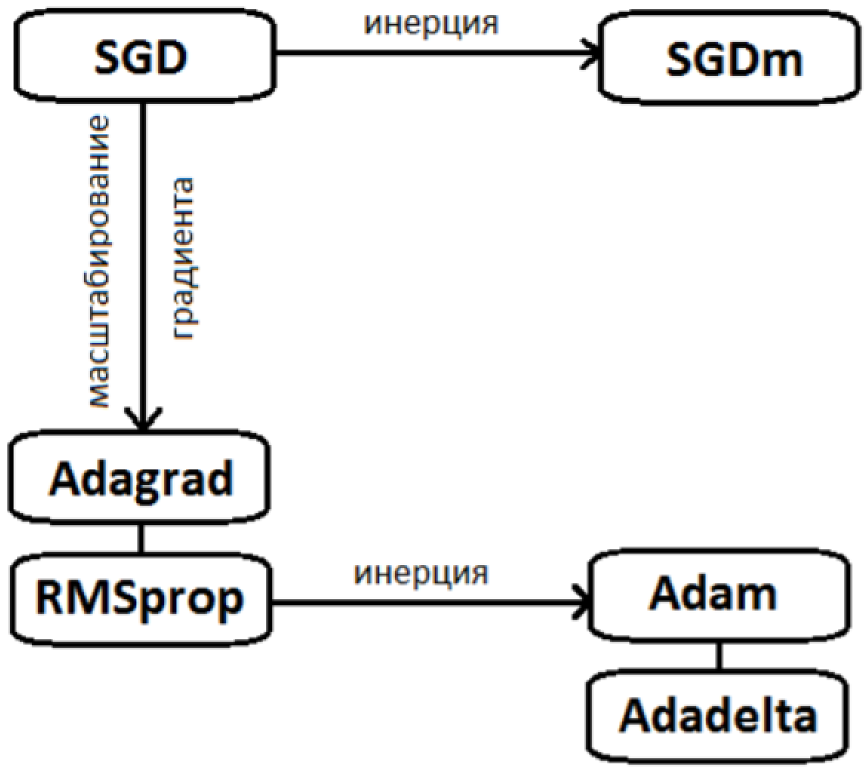
MOMENTUM


Hills and canyon
nesterov momentum 0.25
nesterov momentum 0.975
nesterov momentum 0.9
7.000
4.000
3.500
3.750
4.500
6.750
1.750
1.500
2.000
2.500
3.250
2.250
5.750
4.750
1.000
0.500
0.750
0.250
5.250
6.250
3.000
4.250
2.750
1.250
5.500
5.000
6.500
8.000
9.000
6.250
3.250
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2

МОДИФИКАЦИИ ГРАДИЕНТНОГО СПУСКА



МОДИФИКАЦИИ SGD



[ссылка на статью со схемой]