Разбор летучки

Лекция 6

Линейные методы классификации

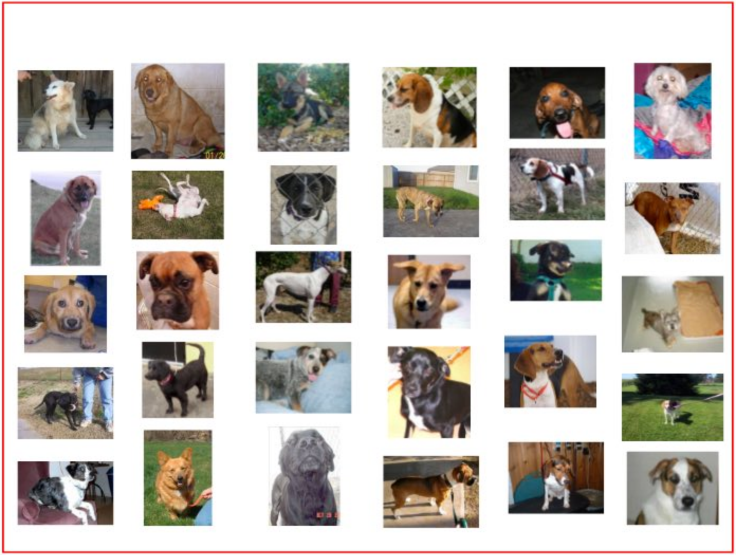
Екатерина Тузова

Мотивирующий пример

Пример

Пример

Пример



Постановка задачи

$X = \mathbb{R}^n$, $Y = \{-1, +1\}$
$X^l = (x_i, y_i)_{i=1}^l$ – обучающая выборка

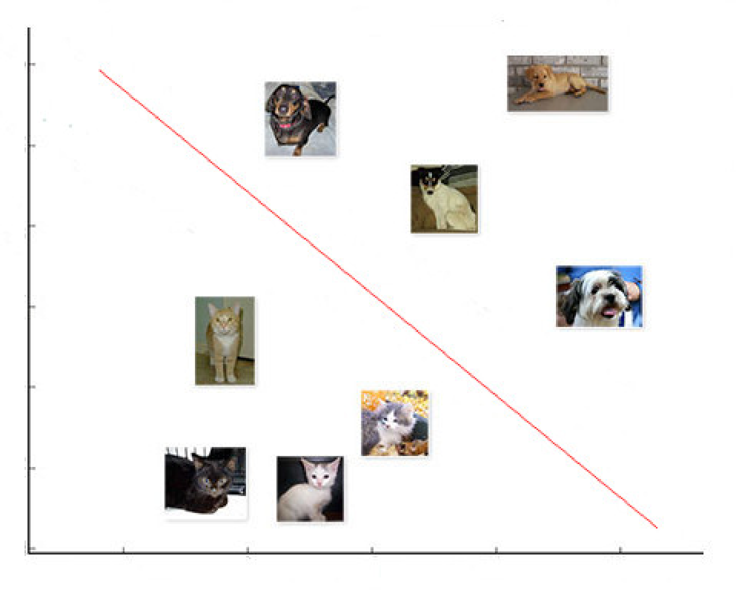
Задача: Построить алгоритм $a\colon X \to Y$, способный классифицировать произвольный объект $x \in X$.

Перцептрон

Идея: $(n-1)$-мерную гиперплоскость, которая разделяет данные.

decision surface, decision boundary

Пример



Модель McCulloch-Pitts

$X = \mathbb{R}^n$, $Y = \{-1, +1\}$

Модель McCulloch-Pitts

$X = \mathbb{R}^n$, $Y = \{-1, +1\}$

$$a(x, w) = \sigma(\sum_{j=1}^{n} w_j x^j - w_0) = \sigma(\langle \mathbf{w}, \mathbf{x} \rangle)$$

x^j — признаки объекта, $j = 1, \ldots, n$

$w_j \in R$ – веса признаков

$\sigma(s)$ – функция активации (например, sign)

Модель McCulloch-Pitts

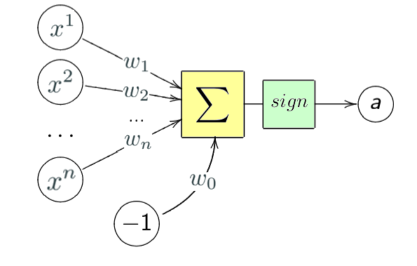
$X = \mathbb{R}^n$, $Y = \{-1, +1\}$

$$a(x, w) = \sigma(\sum_{j=1}^{n} w_j x^j - w_0) = \sigma(\langle \mathbf{w}, \mathbf{x} \rangle)$$

x^j — признаки объекта, $j = 1, \dots, n$

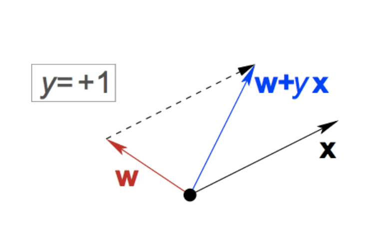
$w_j \in R$ – веса признаков

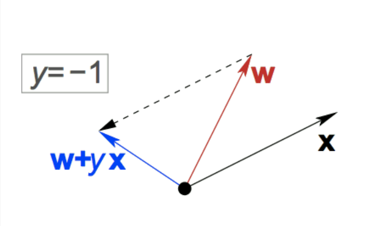
$\sigma(s)$ – функция активации (например, sign)

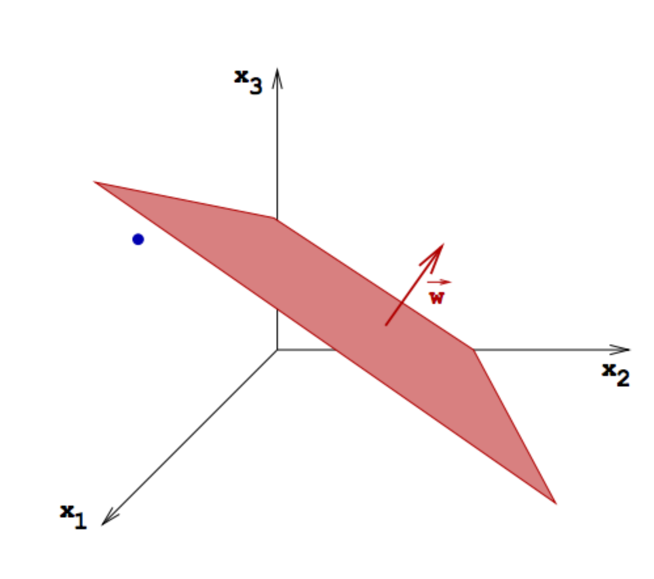


Как подобрать веса w?

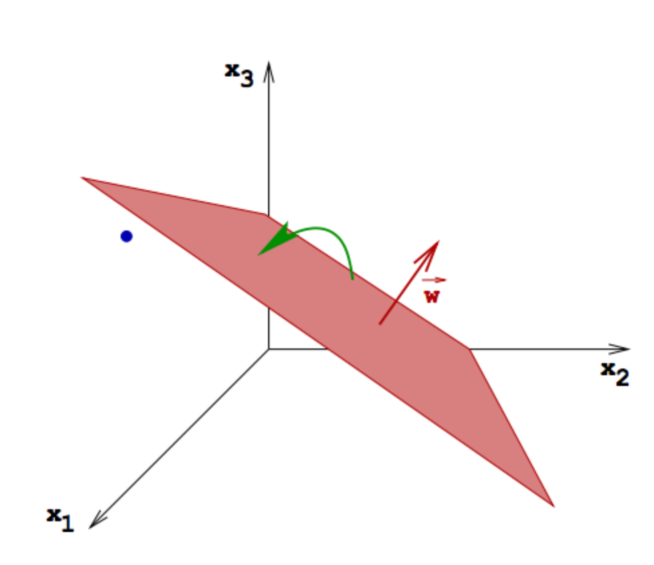
Идея



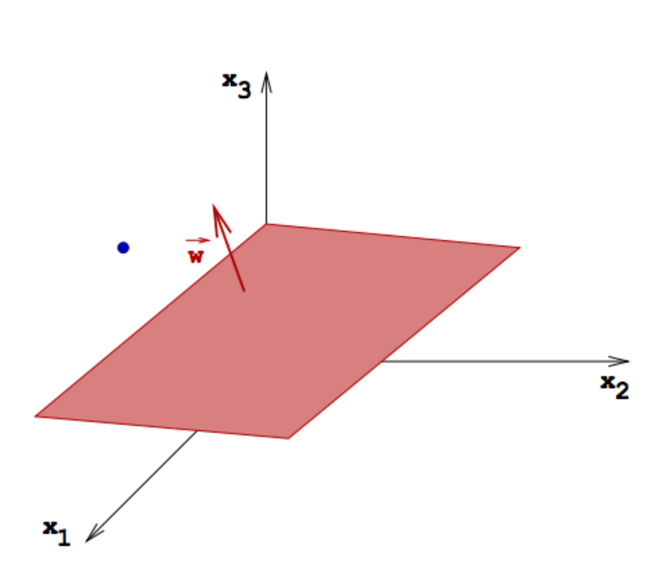




x_3
$\vec{w}$
x_2
x_1



x_3
$\vec{w}$
x_2
x_1



x_3
$\vec{w}$
x_2
x_1

Perceptron Learning Algorithm

```
function PLA(X^l)
    Инициализировать w_0, ..., w_n
    repeat[пока w изменяются]
        for i = 1, ..., l do
            if a(x_i) ≠ y_i then
                w = w + y_i x_i
```

Что делать в случае нескольких классов?

Несколько классов

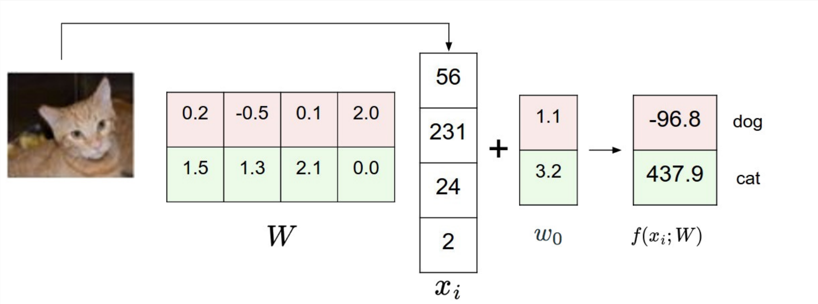
Идея: Для каждого класса заведём отдельный вектор $\mathbf{w_y}$.

Несколько классов

Идея: Для каждого класса заведём отдельный вектор $\mathbf{w_y}$.

$$a(x) = \arg\max_y \langle \mathbf{w_y}, \mathbf{x} \rangle$$

Пример



0.2
-0.5
0.1
2.0
1.5
1.3
2.1
0.0
W
56
231
24
2
x_i
+
1.1
3.2
w_0
-96.8
437.9
dog
cat
$f(x_i; W)$

Multiclass PLA

function $\text{MULTICLASS\_PLA}(X^l)$

Инициализировать $\mathbf{w_0}, \dots, \mathbf{w_n}$

repeat[пока $\mathbf{w}$ изменяются]

for $i = 1, \dots, l$ **do**

$a(\mathbf{x_i}) = \arg\max_y \langle \mathbf{w_y}, \mathbf{x_i} \rangle$

if $a(\mathbf{x_i}) \neq y_i$ **then**

$\mathbf{w_y} = \mathbf{w_y} - \mathbf{x_i}$

$\mathbf{w_{y^*}} = \mathbf{w_{y^*}} + \mathbf{x_i}$

Будем улучшать алгоритм

Алгоритм

function $\textsc{multiclass\_PLA}(X^l,\ \alpha)$

  Инициализировать $\mathbf{w_0}, \ldots, \mathbf{w_n}$

  repeat[пока $\mathbf{w}$ изменяются]

    for $i = 1, \ldots, l$ **do**

      $a(\mathbf{x_i}) = \arg\max\limits_{y} \langle \mathbf{w_y}, \mathbf{x_i} \rangle$

      if $a(\mathbf{x_i}) \neq y_i$ **then**

        $\mathbf{w_y} = \mathbf{w_y} - \alpha \mathbf{x_i}$

        $\mathbf{w_{y^*}} = \mathbf{w_{y^*}} + \alpha \mathbf{x_i}$

Наблюдения

+ Алгоритм гарантированно сходится, если обучающая выборка линейно разделима

Наблюдения

+ Алгоритм гарантированно сходится, если обучающая выборка линейно разделима

– Не работает в случае неразделимой выборки

Что делать, если данные не линейно разделимы?

Идея: На этапе предобработки скомбинировать признаки в полиномиальные признаки.

Обобщённый линейный классификатор

Постановка задачи

$X = \mathbb{R}^n$, $Y = \{-1, +1\}$
$X^l = (x_i, y_i)_{i=1}^l$ – обучающая выборка

Задача:
$(n-1)$-мерную гиперплоскость, которая разделяет данные как можно лучше.

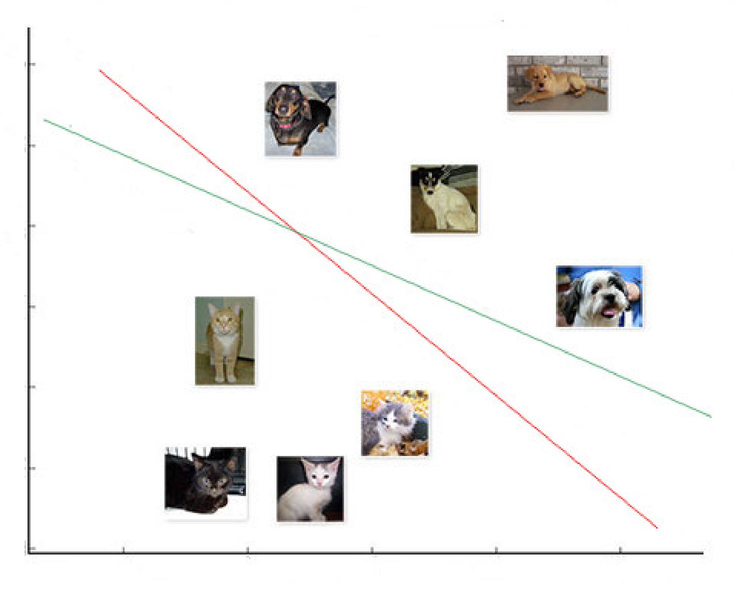
Постановка задачи

$X = \mathbb{R}^n$, $Y = \{-1, +1\}$
$X^l = (x_i, y_i)_{i=1}^l$ – обучающая выборка

Задача:
$(n-1)$-мерную гиперплоскость, которая разделяет данные как можно лучше.

Как можно лучше – это как?

Пример



Определение отступа

$g = \langle \mathbf{w}, \mathbf{x} \rangle = 0$ – разделяющая поверхность
$M_i(\mathbf{w}) = \langle \mathbf{w}, \mathbf{x_i} \rangle y_i$ – отступ объекта x_i
$M_i(\mathbf{w}) < 0 \Rightarrow$ алгоритм $a(\mathbf{x}, \mathbf{w})$ ошибается на x_i

Минимизация эмпирического риска

Число ошибок на обучающей выборке:

$$Q(\mathbf{w}) = \sum_{i=1}^{l} [M_i(\mathbf{w}) < 0] \to \min_{\mathbf{w}}$$

Минимизация эмпирического риска

Число ошибок на обучающей выборке:

$$Q(\mathbf{w}) = \sum_{i=1}^{l} [M_i(\mathbf{w}) < 0] \to \min_{\mathbf{w}}$$

– Индикаторную функцию сложно оптимизировать

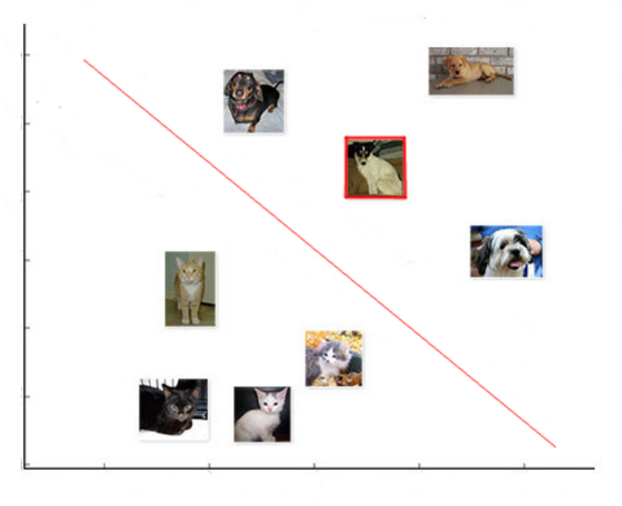
Минимизация эмпирического риска

Число ошибок на обучающей выборке:

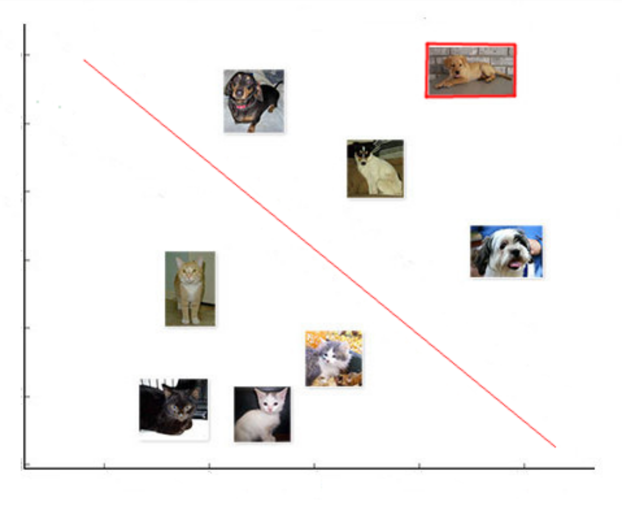
$$Q(\mathbf{w}) = \sum_{i=1}^{l} [M_i(\mathbf{w}) < 0] \to \min_{\mathbf{w}}$$

- Индикаторную функцию сложно оптимизировать
- Теряем информацию насколько i-й объект был надежен

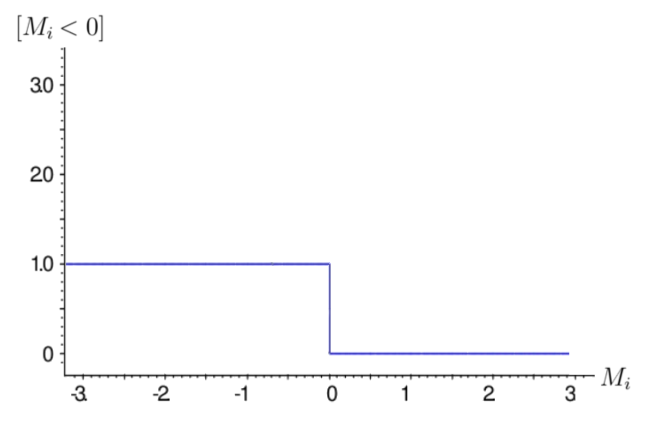
Отступ



Отступ



Функция $[M < 0]$

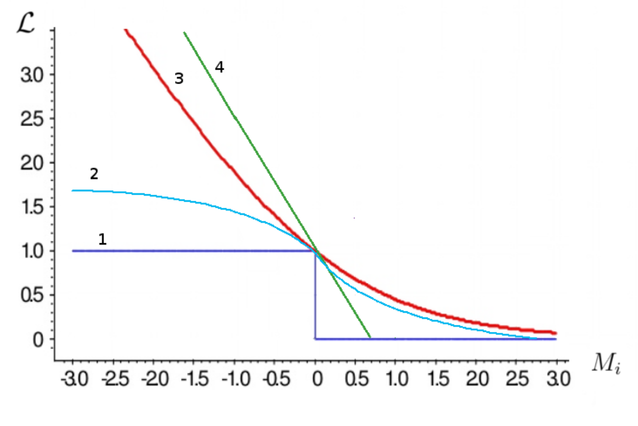


Минимизация эмпирического риска

$$Q(\mathbf{w}) = \sum_{i=1}^{l} [M_i(\mathbf{w}) < 0] \leq \sum_{i=1}^{l} \mathcal{L}(M_i(\mathbf{w})) \to \min_{\mathbf{w}}$$

$\mathcal{L}$ – функция потерь, невозрастающая, неотрицательная.
$\mathcal{L}$ должна мажорировать $[M_i(\mathbf{w}) < 0]$

Примеры $\mathcal{L}$



1. $[M_i(\mathbf{w}) < 0]$
2. $S(M) = 2(1 + e^M)^{-1}$ – сигмоидная
3. $L(M) = \log_2(1 + e^{-M})$ – логарифмическая
4. $V(M) = (1 - M)_+$ – кусочно-линейная

Что такое градиент?

Градиент

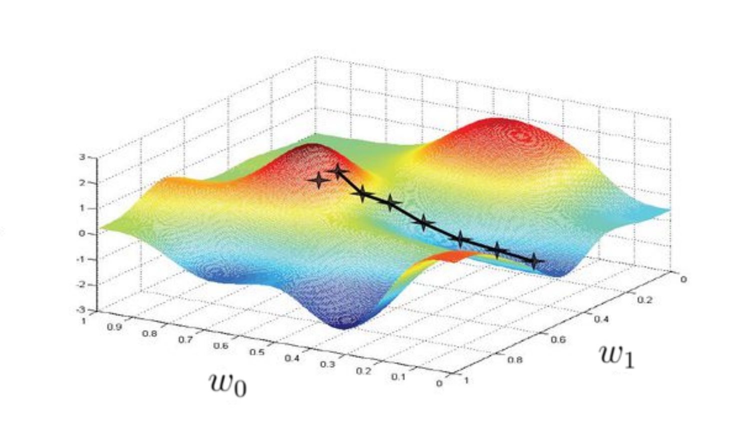
Метод градиентного спуска

function $\textsc{gradient\_descent}(X^l, \alpha)$
    Инициализировать $\mathbf{w}$
    repeat[пока $\mathbf{w}$ изменяются]
        $\mathbf{w} = \mathbf{w} - \alpha \bigtriangledown Q(\mathbf{w})$

$$\bigtriangledown Q(\mathbf{w}) = \left(\frac{\partial Q(\mathbf{w})}{\partial w_j}\right)_{j=0}^{n} = \sum_{i=1}^{l} \mathcal{L}'(\langle \mathbf{w}, \mathbf{x_i} \rangle y_i)\mathbf{x_i} y_i$$

α – градиентный шаг (темп обучения)

Градиентный спуск



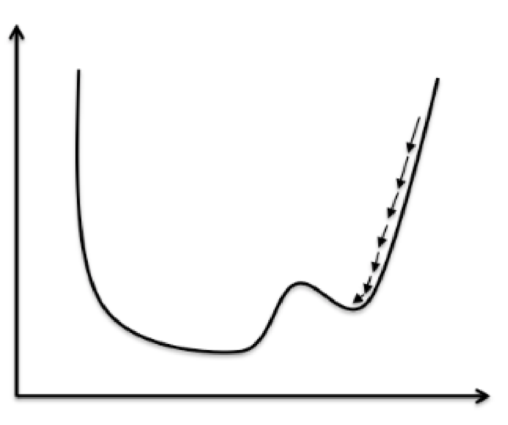
Метод градиентного спуска

Случай двух признаков.

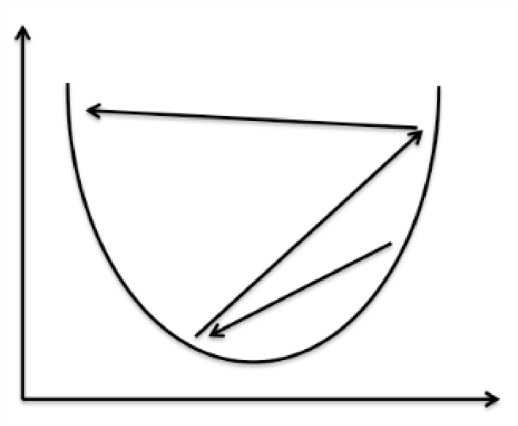
function $\textsc{gradient\_descent}(X^l, \alpha)$
    Инициализировать w_0, w_1
    repeat[пока w_0 и w_1 не стабилизируются]
        $tmp_0 = w_0 - \alpha \frac{\partial Q(w)}{\partial w_0}$
        $tmp_1 = w_1 - \alpha \frac{\partial Q(w)}{\partial w_1}$
        $w_0 = tmp_0$
        $w_1 = tmp_1$

Почему важно обновить w_0 и w_1 одновременно?

Маленький градиентный шаг



Большой градиентный шаг



Выбор величины шага

- $\alpha_t \to 0$
- Метод скорейшего градиентного спуска:
  $Q(w - \alpha \bigtriangledown Q(w)) \to \min\limits_{\alpha}$
- Пробные случайные шаги

В чем проблема?

Метод стохастического градиента

$$\mathbf{w} = \mathbf{w} - \alpha \sum_{i=1}^{l} \mathcal{L}'(\langle \mathbf{w}, \mathbf{x_i} \rangle y_i) \mathbf{x_i} y_i$$

Метод стохастического градиента

$$\mathbf{w} = \mathbf{w} - \alpha \sum_{i=1}^{l} \mathcal{L}'(\langle \mathbf{w}, \mathbf{x_i} \rangle y_i) \mathbf{x_i} y_i$$

Идея: Давайте брать (x_i, y_i) по одному и сразу обновлять вектор весов

Алгоритм

function STOCHASTIC_GRADIENT(X^l, α, η)

  Перемешать данные в X^l

  Инициализировать $\mathbf{w}$

  $Q(\mathbf{w}) = \sum\limits_{i=1}^{l} \mathcal{L}(\langle \mathbf{w}, \mathbf{x_i} \rangle y_i)$

  repeat[пока Q и/или w не стабилизируются]

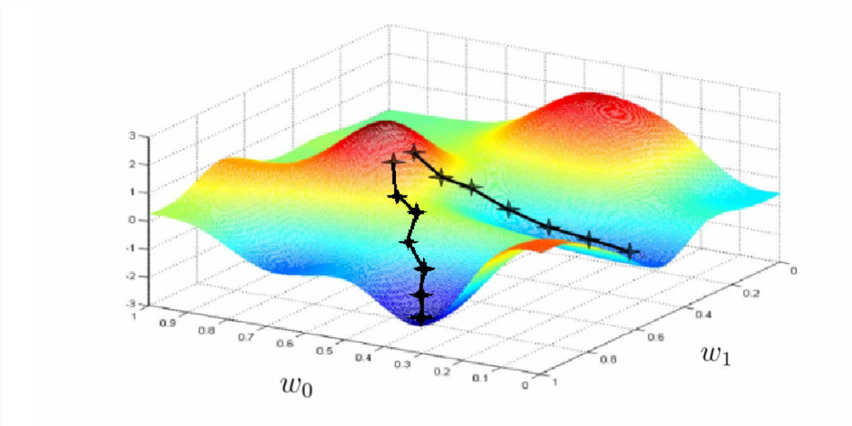
    Взять $\mathbf{x_i}$ из X^l

    Потеря: $\varepsilon_i = \mathcal{L}(\langle \mathbf{w}, \mathbf{x_i} \rangle y_i)$

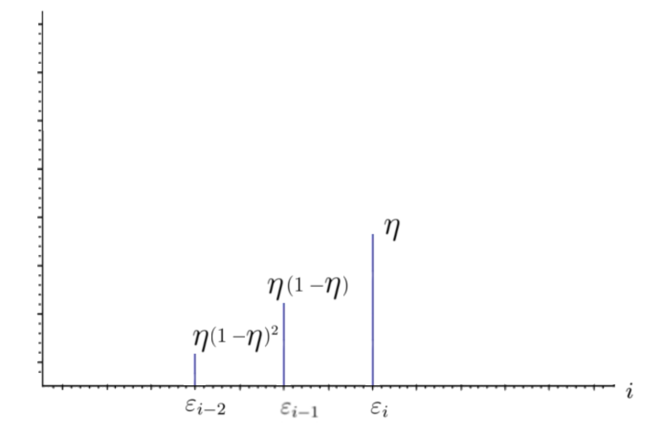
    Градиентный шаг: $\mathbf{w} = \mathbf{w} - \alpha \mathcal{L}'(\langle \mathbf{w}, \mathbf{x_i} \rangle y_i) \mathbf{x_i} y_i$

    Оценить $Q = (1 - \eta) Q + \eta \varepsilon_i$

Градиентный спуск

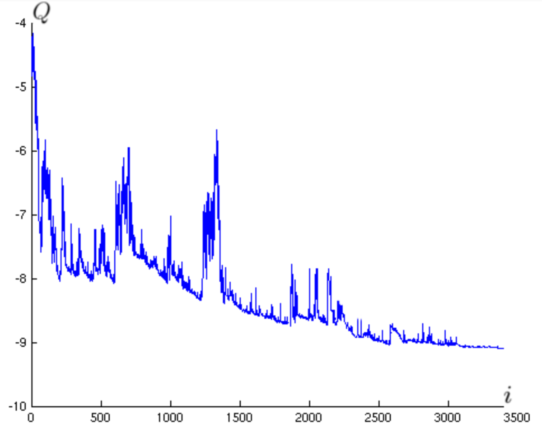


Учет ошибки ε_i в алгоритме

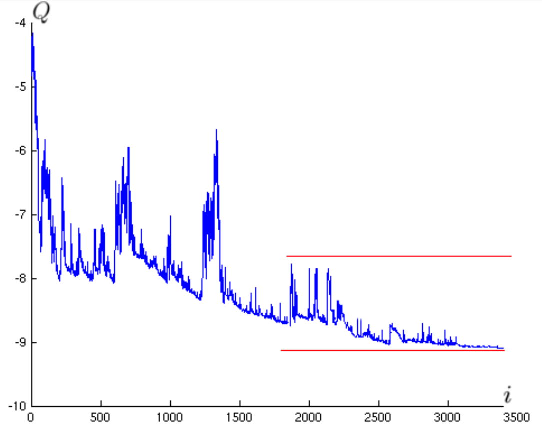


Что значит – "пока Q и/или w не стабилизируются"?

Зависимость Q от номера итерации



Зависимость Q от номера итерации



Актуальные вопросы

– Инициализация весов

Актуальные вопросы

- Инициализация весов
- Порядок предъявления объектов

Инициализация весов

Инициализация весов

- $w_j = 0$, j = 1, . . . , n
- $w_j = random(-\frac{1}{2n}, \frac{1}{2n})$ – небольшие случайные значения
- Обучение по небольшой случайной подвыборке объектов
- Многократный запуск из разных случайных начальных приближений

Порядок предъявления объектов

Порядок предъявления объектов

- Попеременно брать объекты из разных классов
- Чаще брать те объекты, на которых была допущена большая ошибка
- Вообще не брать "хорошие" объекты с $M_i > \mu_+$
- Вообще не брать выбросы с $M_i < \mu_-$

Плюсы и минусы

\+ Легко реализовать

Плюсы и минусы

+ Легко реализовать
+ Легко обобщить на разные g, $\mathcal{L}$

Плюсы и минусы

- \+ Легко реализовать
- \+ Легко обобщить на разные g, $\mathcal{L}$
- \+ Не обязательно брать все элементы выборки для обучения

Плюсы и минусы

+ Легко реализовать
+ Легко обобщить на разные g, $\mathcal{L}$
+ Не обязательно брать все элементы выборки для обучения

– Возможна медленная сходимость

Плюсы и минусы

+ Легко реализовать
+ Легко обобщить на разные g, $\mathcal{L}$
+ Не обязательно брать все элементы выборки для обучения

– Возможна медленная сходимость
– Застревание в локальных минимумах

Плюсы и минусы

+ Легко реализовать
+ Легко обобщить на разные g, $\mathcal{L}$
+ Не обязательно брать все элементы выборки для обучения

– Возможна медленная сходимость
– Застревание в локальных минимумах
– Подбор параметров

Плюсы и минусы

+ Легко реализовать
+ Легко обобщить на разные g, $\mathcal{L}$
+ Не обязательно брать все элементы выборки для обучения

– Возможна медленная сходимость
– Застревание в локальных минимумах
– Подбор параметров
– Проблема переобучения

Вопросы?

Что почитать по этой лекции

- Professor Yaser Abu-Mostafa MOOC
- Tom Mitchell "Machine Learning" Chapter 4

На следующей лекции

- Функционалы качества
- Неравенство Хефдинга
- Близость гипотез
- Неравенство Вапника-Червоненкиса
- Генерация модельных данных