



# ТЕХНОСФЕРА

Лекция 10  
Линейные модели  
алгоритмическая перспектива

Николай Анохин

25 апреля 2016 г.

# План занятия

Линейные модели

SVM

Функции ядра

SGD

## Постановка задачи

**Дано.** Признаковые описания  $N$  объектов  $\mathbf{x} = (x_1, \dots, x_m) \in \mathcal{X}$ , образующие тренировочный набор данных  $X$ , и значения целевой переменной  $y = f(\mathbf{x}) \in \mathcal{Y}$  для каждого объекта из  $X$ .

**Найти.** Для семейства параметрических функций

$$H = \{h(\mathbf{x}, \theta) = y : \mathcal{X} \times \Theta \rightarrow \mathcal{Y}\},$$

найти значение вектора параметров  $\theta^*$ , такое что  $h^*(\mathbf{x}) = h(\mathbf{x}, \theta^*)$  наилучшим образом приближает целевую функцию.

$\mathcal{Y} \in \{C_1, C_2, \dots, C_K\}$  – задача классификации

$\mathcal{Y} \in [a, b] \subset \mathcal{R}$  – задача регрессии

# Обобщенная линейная модель / GLM

$$y(\mathbf{x}, \mathbf{w}) \sim pdf \left[ f(\mathbf{w}^\top \phi(\mathbf{x})) \right],$$

- ▶  $\phi_n(\mathbf{x})$  – базисные функции
- ▶  $f(a)$  – функция активации
- ▶  $pdf$  – распределение из экспоненциального семейства

# Обобщенные линейные модели



# Линейные модели

Рассматривается случай 2 классов

Функция принятия решения

$$y(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + w_0$$

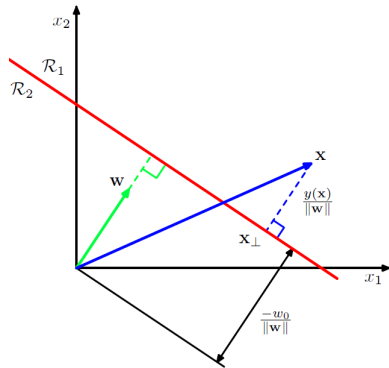
Регионы принятия решения

$$R_1 = \{\mathbf{x} : y(\mathbf{x}) > 0\}$$

$$R_2 = \{\mathbf{x} : y(\mathbf{x}) < 0\}$$

Задача

найти параметры модели  $\mathbf{w}$ ,  $w_0$



# Линейные модели: наблюдения

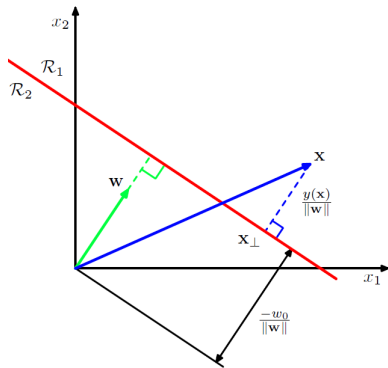
Разделяющая поверхность

$$\mathcal{D} = \{\mathbf{x} : \mathbf{w}^\top \mathbf{x} + w_0 = 0\}$$

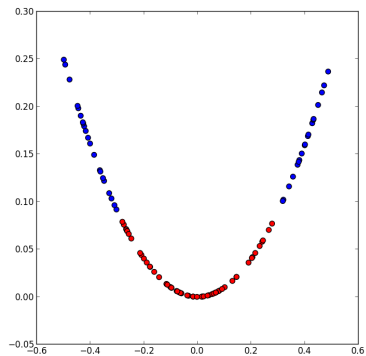
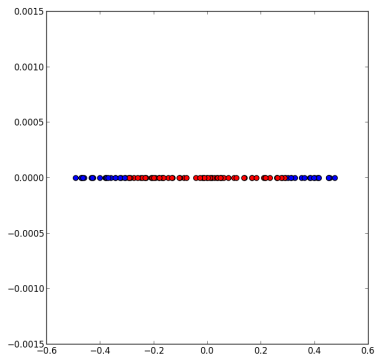
1.  $\mathbf{w}$  – нормаль к  $\mathcal{D}$
2.  $d = -\frac{w_0}{\|\mathbf{w}\|}$  – расстояние от центра координат до  $\mathcal{D}$
3.  $r(\mathbf{x}) = \frac{y(\mathbf{x})}{\|\mathbf{w}\|}$  – расстояние от  $\mathcal{D}$  до  $\mathbf{x}$

Положим  $x_0 \equiv 1$ , получим модель

$$y(\tilde{\mathbf{x}}) = \tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}$$



# Мотивация





# Обобщенные линейные модели

Линейная модель

$$y(\mathbf{x}) = w_0 + \sum w_i x_i$$

Квадратичная модель

$$y(\mathbf{x}) = w_0 + \sum w_i x_i + \sum \sum w_{ij} x_i x_j$$

Обобщенная линейная модель

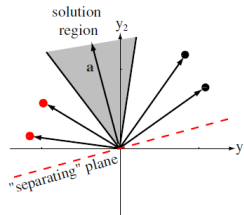
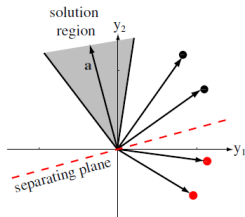
$$y(\mathbf{x}) = \sum w_i \phi_i(\mathbf{x})$$

# Случай линейно разделимых классов

Обобщенная линейная модель (внимание: переобозначения!)

$$g(\mathbf{x}) = \sum w_i \phi_i(\mathbf{x}) \sim \mathbf{w}^T \mathbf{x}$$

Дана обучающая выборка  $D = (X, Y)$



## Идея

Преобразовать объекты второго класса в обратные им и решать задачу оптимизации в области  $\mathbf{w}^T \mathbf{x}_i > 0, \forall i$

# Задача оптимизации

## Задача

Минимизируем критерий  $J(\mathbf{w})$  при условиях  $\mathbf{w}^T \mathbf{x}_i > 0, \forall i$

Пусть  $\mathcal{H}$  – множество неправильно проклассифицированных объектов

- ▶  $J_e(\mathbf{w}) = \sum_{\mathbf{x} \in \mathcal{H}} 1$
- ▶  $J_p(\mathbf{w}) = \sum_{\mathbf{x} \in \mathcal{H}} -\mathbf{w}^T \mathbf{x}$
- ▶  $J_q(\mathbf{w}) = \sum_{\mathbf{x} \in \mathcal{H}} (\mathbf{w}^T \mathbf{x})^2$
- ▶  $J_r(\mathbf{w}) = \sum_{\mathbf{x} \in \mathcal{H}} \frac{(\mathbf{w}^T \mathbf{x} - b)^2}{\|\mathbf{x}\|}$

Улучшение: добавить отступы

# Логистическая регрессия

функция правдоподобия (кросс-энтропия)

$$J_c(\mathbf{w}) = - \sum_{n=1}^N y_n \log \sigma(\mathbf{w}^T \mathbf{x}) + (1 - y_n) \log(1 - \sigma(\mathbf{w}^T \mathbf{x})) \rightarrow \min_{\mathbf{w}}$$

Градиент

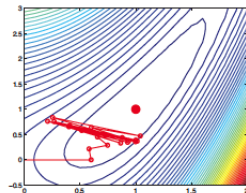
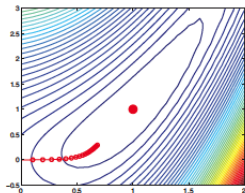
$$\nabla J_c(\mathbf{w}) = \sum_{n=1}^N (p(y = 1 | \phi_n) - y_n) \phi_n$$

Гессиан

$$\nabla^2 J_c(\mathbf{w}) = \sum_{n=1}^N p(y = 1 | \phi_n) (1 - p(y = 1 | \phi_n)) \phi_n \phi_n^T$$

## Градиентный спуск

```
1 function gd(grad, a0, epsilon):  
2     initialise eta(k)  
3     k = 0  
4     a = a0  
5     do:  
6         k = k + 1  
7         a = a - eta(k) grad(a)  
8     until eta(k) grad(a) < epsilon  
9     return a
```



# Метод Ньютона

$$J(\mathbf{a}) \approx J(\mathbf{a}_k) + \nabla J(\mathbf{a}_k)^T (\mathbf{a} - \mathbf{a}_k) + \frac{1}{2} (\mathbf{a} - \mathbf{a}_k)^T \nabla^2 J(\mathbf{a}_k) (\mathbf{a} - \mathbf{a}_k) \rightarrow \min_{\mathbf{a}}$$
$$\mathbf{a} = \mathbf{a}_k - \nabla^2 J(\mathbf{a}_k)^{-1} \nabla J(\mathbf{a}_k)$$

```
1 function newton(grad, hessian, a0, epsilon):
2     initialise eta(k)
3     k = 0
4     a = a0
5     do:
6         k = k + 1
7         g = grad(a)
8         H = hessian(a)
9         d = solve(H * d = -g) # find d = - inv(H) * g
10        a = a + eta(k) d
11    until convergence
12    return a
```

BFGS – использовать приближение  $\nabla^2 J(\mathbf{a}_k)$  или  $\nabla^2 J(\mathbf{a}_k)^{-1}$

# Iterative Reweighted Least Squares

Градиент и Гессиан логистической регрессии в матричной форме

$$\nabla J_c(\mathbf{w}) = X^T(\sigma - Y)$$

$$\nabla^2 J_c(\mathbf{w}) = X^T S X = X^T \text{diag}\{\sigma_n(1 - \sigma_n)\} X$$

Обновление весов

$$\mathbf{w}_{k+1} = \mathbf{w}_k - (X^T S_k X)^{-1} X^T S_k \mathbf{z}_k,$$

$$\mathbf{z}_k = X \mathbf{w}_k + S_k^{-1}(Y - \sigma_k)$$

Минимизация

$$\sum_{n=1}^N S_{kn}(z_{kn} - \mathbf{w}^T x_n)^2$$

## Случай линейно неразделимых классов

- ▶ Использовать  $\eta(k) \rightarrow 0$  при  $k \rightarrow \infty$
- ▶ Линейное программирование
- ▶ Подобрать хитрый критерий оптимизации



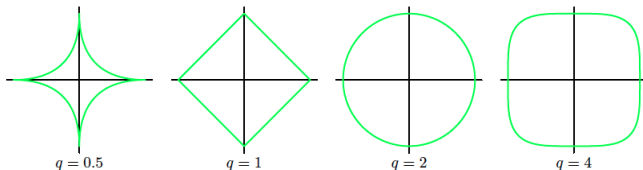
# Снова переобучение

Оптимизируем критерий с регуляризацией

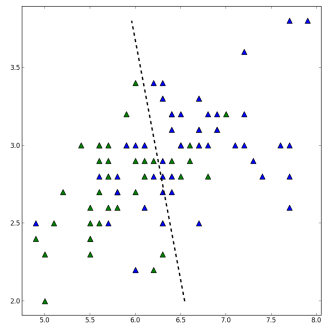
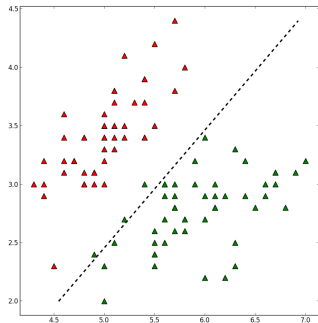
$$J_1(a) = J(a) + \lambda J_R(a)$$

$\lambda$  – коэффициент регуляризации

$$J_R(a) = \sum |a_j|^q$$



# Перцептрон: результаты



SVM



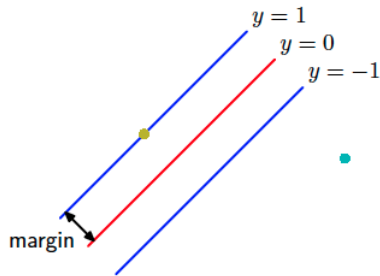
# Максимальный зазор

Margin – наименьшее расстояние между РП и обучающим объектом.

$$\begin{aligned}d_j &= \frac{|y(\mathbf{x}_j)|}{\|\mathbf{w}\|} = \frac{t_j y(\mathbf{x}_j)}{\|\mathbf{w}\|} = \\&= \frac{t_j(\mathbf{w}^\top \phi(\mathbf{x}_j) + b)}{\|\mathbf{w}\|}\end{aligned}$$

Оптимальная РП

$$\arg \max_{\mathbf{w}, b} \left[ \frac{1}{\|\mathbf{w}\|} \min_j t_j(\mathbf{w}^\top \phi(\mathbf{x}_j) + b) \right]$$



# Задача оптимизации

Расстояние от точки  $x_j$  до РП

$$d_j = \frac{t_j(\mathbf{w}^\top \phi(\mathbf{x}_j) + b)}{\|\mathbf{w}\|}$$

Для точки  $x_j$ , лежащей на минимальном расстоянии от РП положим

$$t_j(\mathbf{w}^\top \phi(\mathbf{x}_j) + b) = 1$$

## Задача оптимизации

$$\frac{1}{2} \|\mathbf{w}\|^2 \rightarrow \min_{\mathbf{w}, b}$$

при условиях

$$t_j(\mathbf{w}^\top \phi(\mathbf{x}_j) + b) \geq 1, \quad \forall j \in 1, \dots, N$$

Метод множителей Лагранжа  $\mathbf{a} = (a_1, \dots, a_N)^\top$ ,  $a_i \geq 0$ .

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{j=1}^N a_j [t_j (\mathbf{w}^\top \phi(\mathbf{x}_j) + b) - 1]$$

Дифференцируем по  $\mathbf{w}$  и  $b$

$$\mathbf{w} = \sum_{j=1}^N a_j t_j \phi(\mathbf{x}_j), \quad 0 = \sum_{j=1}^N a_j t_j$$

Подставляем  $\mathbf{w}$  и  $b$  в лагранжиан

# Сопряженная задача

## Сопряженная задача

$$\tilde{L}(\mathbf{a}) = \sum_{j=1}^N a_j - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N a_i a_j t_i t_j \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j) \rightarrow \max_{\mathbf{a}}$$

при условиях

$$a_j \geq 0, \quad \forall j \in 1, \dots, N$$

$$\sum_{j=1}^N a_j t_j = 0$$

## Наблюдения

- ▶  $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$  – неотрицательно-определенная функция
- ▶ лагранжиан  $\tilde{L}(\mathbf{a})$  – выпуклая и ограниченная сверху функция

# Классификация

Функция принятия решения

$$y(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}) + b = \sum_{j=1}^N a_j t_j \phi(\mathbf{x}_j)^\top \phi(\mathbf{x}) + b = \sum_{j=1}^N a_j t_j k(\mathbf{x}_j, \mathbf{x}) + b$$

Условия Karush-Kuhn-Tucker

$$\begin{aligned} a_j &\geq 0 \\ t_j y(\mathbf{x}_j) - 1 &\geq 0 \\ a_j \{t_j y(\mathbf{x}_j) - 1\} &= 0 \end{aligned}$$

Опорным векторам  $\mathbf{x}_j \in S$  соответствуют  $a_j > 0$

$$b = \frac{1}{N_s} \sum_{i \in S} \left( t_i - \sum_{j \in S} a_j t_j k(\mathbf{x}_i, \mathbf{x}_j) \right)$$



# Линейно-разделимый случай

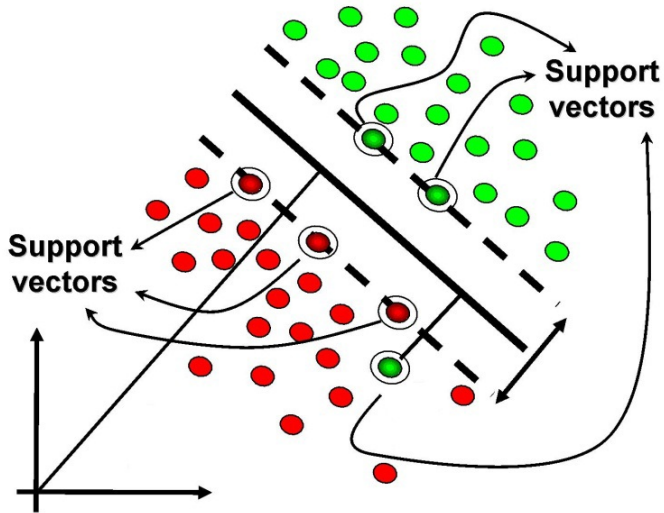
## Задача

Дана обучающая выборка

	$x_1$	$x_2$	$t$
$\mathbf{x}_1$	1	-2	1
$\mathbf{x}_2$	1	2	-1

Найти оптимальную разделяющую плоскость, используя сопряженную задачу оптимизации

## Линейно-неразделимый случай



# Смягчение ограничений

Переменные  $\xi_j \geq 0$  (slacks):

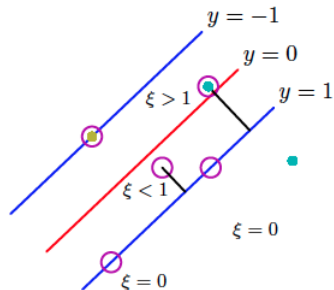
$$\xi_j = \begin{cases} 0, & \text{если } y(\mathbf{x}_j)t_j \geq 1 \\ |t_j - y(\mathbf{x}_j)|, & \text{иначе} \end{cases}$$

Задача оптимизации

$$C \sum_{j=1}^N \xi_j + \frac{1}{2} \|\mathbf{w}\|^2 \rightarrow \min_{\mathbf{w}, b}$$

при условиях

$$t_j y(\mathbf{x}_j) \geq 1 - \xi_j, \quad \xi_j \geq 0$$



# Сопряженная задача

## Сопряженная задача

$$\tilde{L}(\mathbf{a}) = \sum_{j=1}^N a_j - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N a_i a_j t_i t_j \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) \rightarrow \max_{\mathbf{a}}$$

при условиях

$$0 \leq a_j \leq C, \quad \forall j \in 1, \dots, N$$

$$\sum_{j=1}^N a_j t_j = 0$$

## Наблюдения

- ▶  $a_j = 0$  – правильно проклассифицированные объекты
- ▶  $a_j = C$  – опорные векторы внутри отступа
- ▶  $0 < a_j < C$  – опорные векторы на границе

# Классификация

Функция принятия решения

$$y(\mathbf{x}) = \sum_{j=1}^N a_j t_j k(\mathbf{x}_j, \mathbf{x}) + b$$

Константа  $b$

$$b = \frac{1}{N_{\mathcal{M}}} \sum_{i \in \mathcal{M}} \left( t_i - \sum_{j \in \mathcal{S}} a_j t_j k(\mathbf{x}_i, \mathbf{x}_j) \right)$$

# Задача регрессии

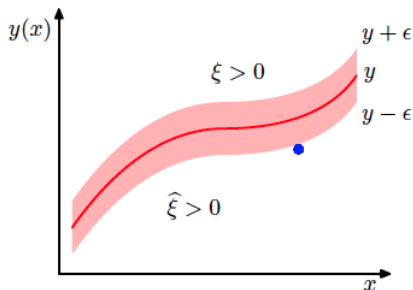
Переменные  $\xi_j \geq 0$ ,  $\hat{\xi}_j \geq 0$  (slacks):

$$t_j \leq y(\mathbf{x}_j) + \epsilon + \xi_n$$

$$t_j \geq y(\mathbf{x}_j) - \epsilon - \hat{\xi}_n$$

Задача оптимизации

$$C \sum_{j=1}^N (\hat{\xi}_j + \xi_j) + \frac{1}{2} \|\mathbf{w}\|^2 \rightarrow \min_{\mathbf{w}, b}$$



# Численные методы оптимизации

- ▶ Chunking (Vapnik, 1982)
- ▶ Decomposition (Osuna, 1996)
- ▶ Sequential Minimal Optimization (Platt, 1999)

## Функции ядра





# Функции ядра

$\phi(\mathbf{x})$  – функция преобразования  $\mathbf{x}$  из исходного пространства в спрямляющее пространство

Проблема: количество признаков может быть очень велико

## Идея Kernel Trick

В процессе тренировки и применения SVM исходные векторы  $\mathbf{x}$  используются только как аргументы в скалярном произведении  $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$ . Но в этом случае можно избежать вычисления  $\phi(\mathbf{x})$ !

# Теорема Мерсера

## Теорема

Функция  $k(\mathbf{x}, \mathbf{z})$  является ядром тогда и только тогда, когда она

- ▶ симметрична

$$k(\mathbf{x}, \mathbf{z}) = k(\mathbf{z}, \mathbf{x})$$

- ▶ неотрицательно определена

$$\int_{\mathbf{x} \in \mathbf{X}} \int_{\mathbf{z} \in \mathbf{X}} k(\mathbf{x}, \mathbf{z}) g(\mathbf{x}) g(\mathbf{z}) d\mathbf{x} d\mathbf{z} \geq 0, \quad \forall g(\mathbf{x}) : \mathbf{X} \rightarrow \mathbb{R}$$

## Задача

Пусть  $\mathbf{x} \in \mathbb{R}^2$ , а преобразование  $\phi(\mathbf{x})$

$$\phi(\mathbf{x}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, \sqrt{2}x_1x_2, x_2^2).$$

Проверить, что функция  $k(\mathbf{x}, \mathbf{z}) = (1 + \mathbf{x}^\top \mathbf{z})^2$  является функцией ядра для данного преобразования.

# Некоторые стандартные функции ядра

- ▶ Линейное ядро

$$k(\mathbf{x}, \mathbf{z}) = \mathbf{x}^\top \mathbf{z}$$

- ▶ Полиномиальное ядро степени  $d$

$$k(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^\top \mathbf{z} + r)^d$$

- ▶ Radial Basis Function

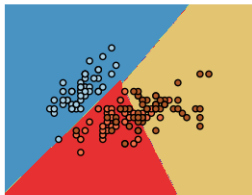
$$k(\mathbf{x}, \mathbf{z}) = e^{-\gamma \|\mathbf{x} - \mathbf{z}\|^2}$$

- ▶ Sigmoid

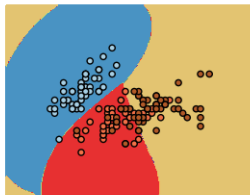
$$k(\mathbf{x}, \mathbf{z}) = \tanh(\gamma \mathbf{x}^\top \mathbf{z} + r)$$

# Опять ирисы

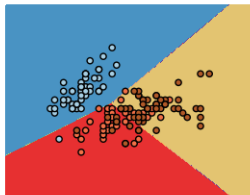
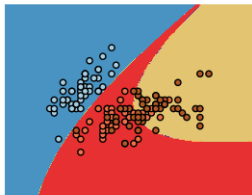
SVC with linear kernel



SVC with RBF kernel



SVC with polynomial (degree 3) kernel    LinearSVC (linear kernel)



# SGD

SGD  
stands for  
Stochastic Gradient  
Descent  
...  
by [allacronyms.com](https://allacronyms.com)



# Связь с линейными моделями

Задача оптимизации

$$C \sum_{j=1}^N \xi_j + \frac{1}{2} \|w\|^2 \sim \sum_{j=1}^N E(y(\mathbf{x}_j), t_j) + \lambda \|w\|^2 \rightarrow \min_{\mathbf{w}, b}$$

Hinge loss

$$E(y_j, t_j) = \begin{cases} 1 - y_j t_j, & \text{если } y_j t_j < 1 \\ 0, & \text{иначе} \end{cases}$$

# Stochastic Gradient Descent

Градиентный спуск

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \eta(k) \frac{1}{N} \sum_{n=1}^N \nabla_{\mathbf{w}} l(\mathbf{x}_n, \mathbf{w}, t_n)$$

Стохастический градиентный спуск

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \eta(k) \nabla_{\mathbf{w}} l(\mathbf{x}_k, \mathbf{w}, t_k)$$

Усредненный стохастический градиентный спуск  $\bar{\mathbf{w}}_k = \frac{1}{k} \sum_{j=1}^k \mathbf{w}_j$

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \eta(k) \nabla_{\mathbf{w}} l(\mathbf{x}_k, \mathbf{w}, t_k), \quad \bar{\mathbf{w}}_{k+1} = \frac{k}{k+1} \bar{\mathbf{w}}_k + \frac{1}{k+1} \mathbf{w}_{k+1}$$

Сходимость:  $\sum_k \eta_k^2 < \infty$ ,  $\sum_k \eta_k = \infty$

## SGD tips

- ▶ Использовать SGD, когда обучение модели занимает слишком много времени
- ▶ Перемешать тренировочную выборку
- ▶ Следить за training error и **validation error**
- ▶ Проверять, правильно ли вычисляется градиент

$$Q(z, w + \delta) \approx Q(z, w) + \delta g$$

- ▶ Подобрать  $\eta_0$  на небольшой выборке

$$\eta_k = \eta_0(1 + \eta_0 \lambda k)^{-1}, \quad \lambda - \text{параметр регуляризации}$$



## SVM – итоги

- + Нелинейная разделяющая поверхность
- + Глобальная оптимизация
- + Разреженное решение
- + Хорошая обобщающая способность
  - Не поддерживает  $p(C_k|\mathbf{x})$
  - Чувствительность к выбросам
  - Нет алгоритма выбора ядра
  - Медленное обучение

## Вопросы

