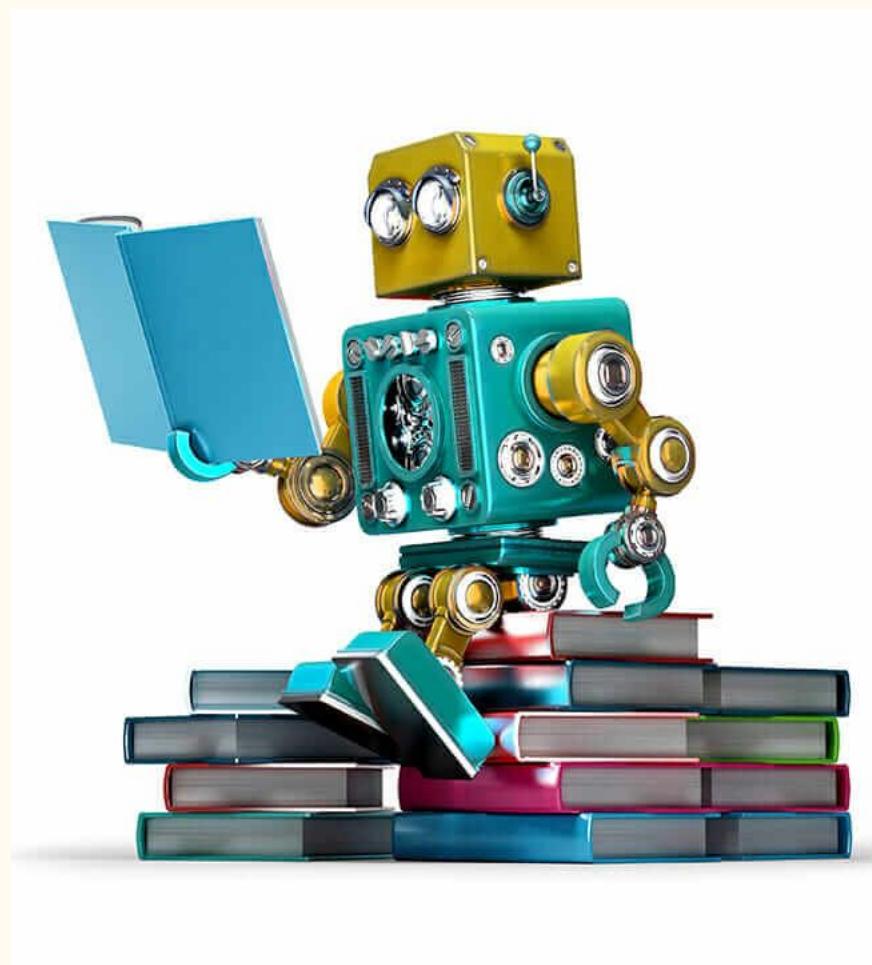


Lecture 2

Bulat Ibragimov
Yandex Research

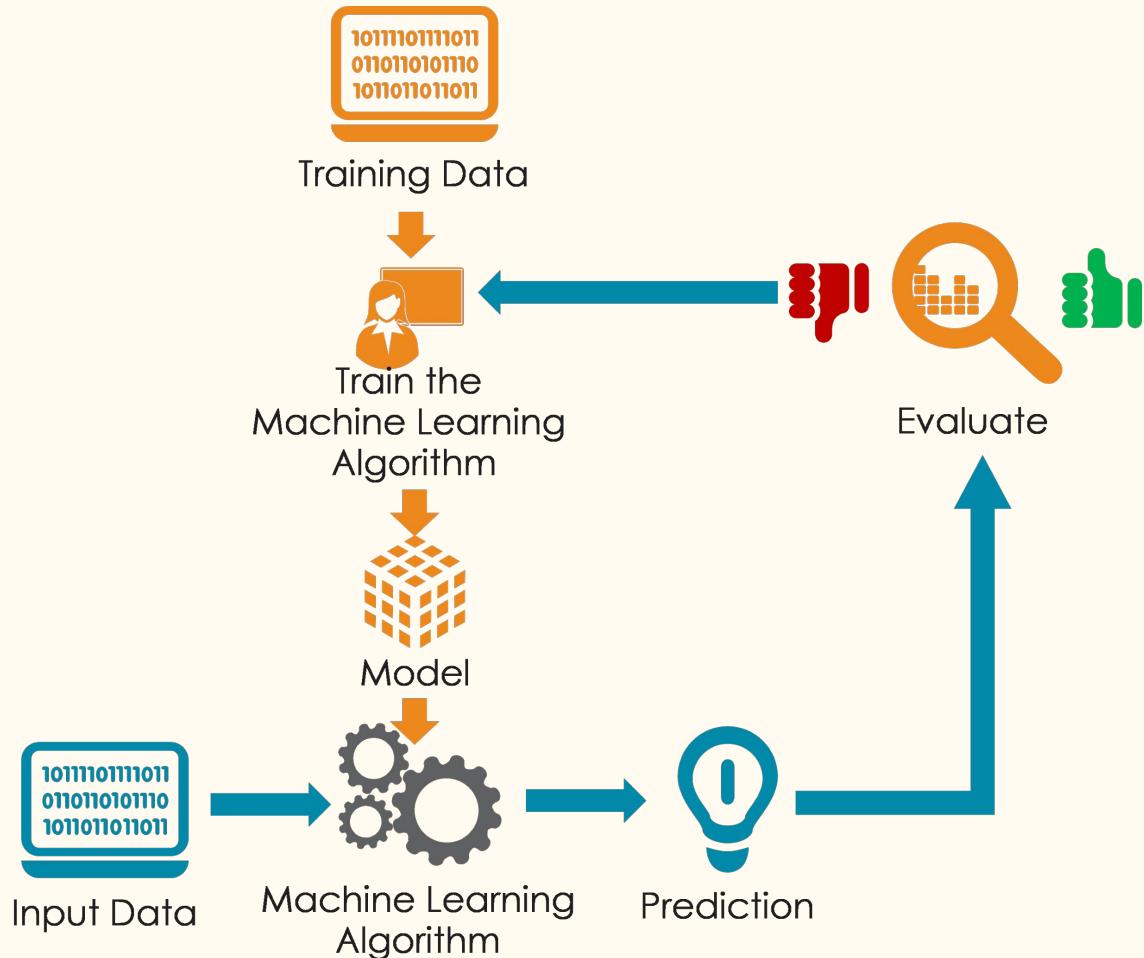
Plan

1. ML reminder
2. How to fit models
3. Linear regression
 - a. MSE solution
 - b. Ridge regression
 - c. Lasso regression
4. Bayesian approach
5. Linear classification
 - a. Logistic regression
 - b. SVM



Reminder

100
010



X – множество **объектов**

Y – множество **допустимых ответов**

y^* – целевая функция, $y^*: X \rightarrow Y$, $y_i = y^*(x_i)$ известны только на **конечном** подмножестве объектов x_1, \dots, x_m из X

Пары (x_i, y_i) – прецеденты

Совокупность пар таких пар при i из $1, \dots, m$ – **обучающая выборка** (X_{train})

a – **решающая функция** (алгоритм), которая любому объекту из X ставит в соответствие допустимый ответ из Y и приближает целевую функцию y^*

X_{test} – **выборка прецедентов** для тестирования построенного алгоритма a

Для решения задачи обучения по прецедентам в первую очередь фиксируется восстанавливаемой зависимости.

По выборке X_{train} построить решающую функцию (*decisionfunction*)
 $a : X \rightarrow Y$, которая приближает целевую функцию y^* , причём не
только на объектах ***обучающей выборки, но и на всём***
множестве X .

Решающая функция a должна быть вычислимой.

Как строится функция а?

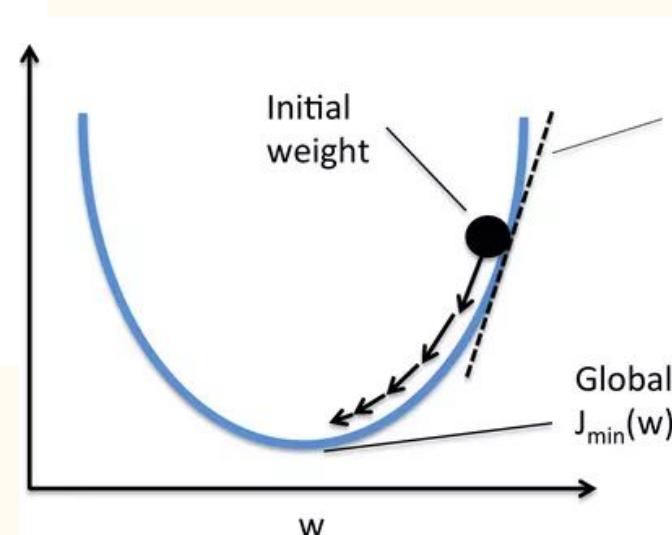
Обучающая выборка — выборка, по которой производится настройка (оптимизация параметров) модели зависимости.

Тестовая выборка — выборка, по которой оценивается качество построенной модели.

Функционал качества (обучение с учителем) — определяется как средняя ошибка ответов, выданных алгоритмом, по всем объектам выборки.

$$L(\hat{y}, y) = I(\hat{y} \neq y),$$

$$\text{logloss} = -\frac{1}{l} \cdot \sum_{i=1}^l (y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i))$$



Метод градиентного спуска

Идея метода

Основная идея метода заключается в том, чтобы осуществлять оптимизацию в направлении наискорейшего спуска, а это направление задаётся антиградиентом $-\nabla f$:

$$x^{[k+1]} = x^{[k]} - \lambda^{[k]} \nabla f(x^{[k]})$$

где $\lambda^{[k]}$ выбирается

- постоянной, в этом случае метод может расходиться;
- дробным шагом, т.е. длина шага в процессе спуска делится на некое число;
- наискорейшим спуском:

$$\lambda^{[k]} = \arg \min_{\lambda} f(x^{[k]} - \lambda \nabla f(x^{[k]}))$$

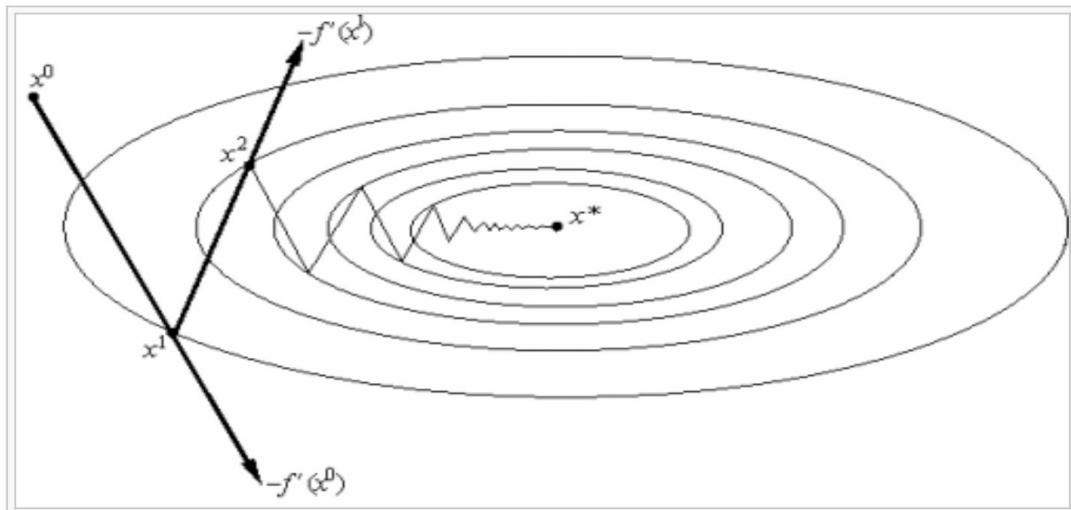
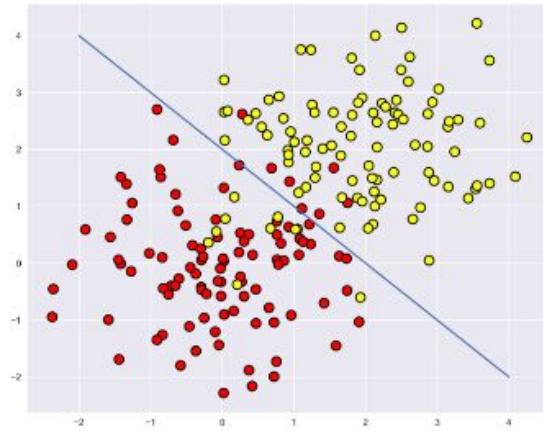
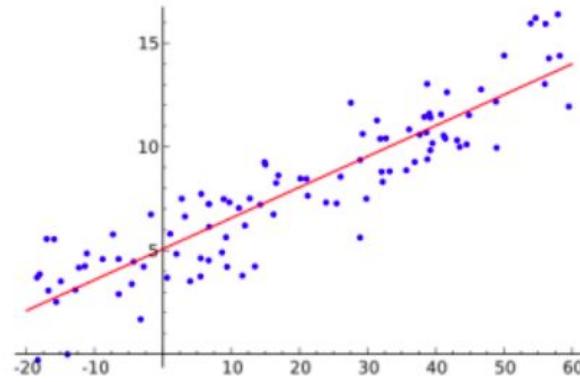


Рис.1 Геометрическая интерпретация метода градиентного спуска с постоянным шагом.
На каждом шаге мы сдвигаемся по вектору антиградиента, "уменьшенному в λ раз".

Регрессия VS Классификация

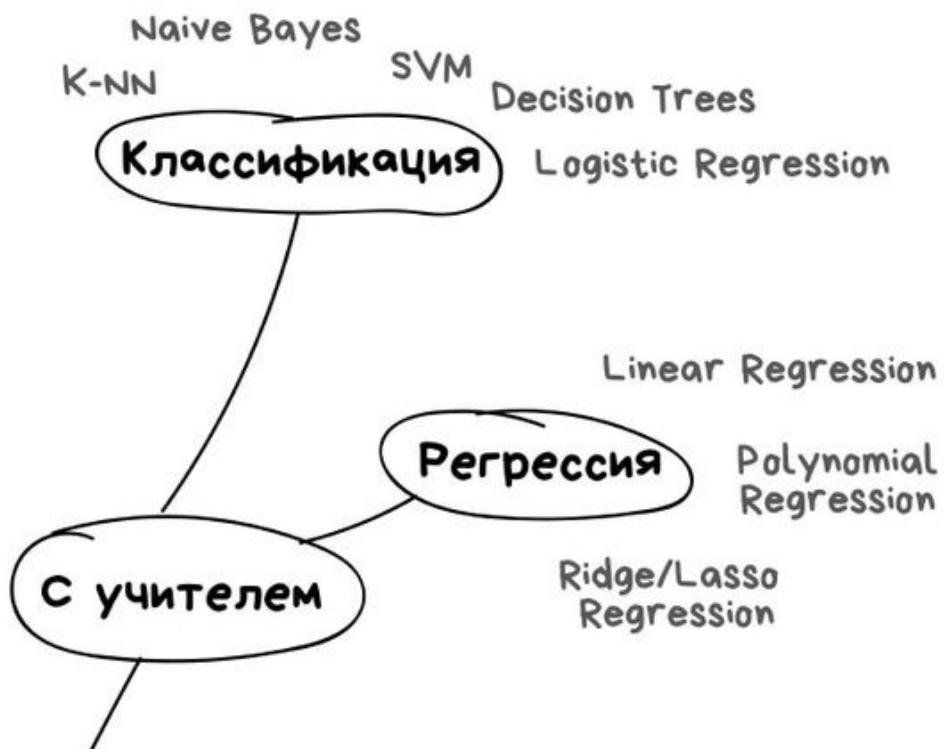


$$Y = \{1, \dots, N\}$$



$$Y \subseteq \mathbf{R}$$





Linear models

100
010

Модель принятия решений

Стоит ли сходить в кино?

- Хорошая погода на улице +5
- Хорошая компания +10
- Плохие отзывы -7
- Есть важные дела -10
- Близко к дому +5
- Общее настроение +3

Сумма +6 — идём

Линейная модель формально

$$Y = X_1 * W_1 + X_2 * W_2 + \dots + X_k * W_k + b$$

Y - то, что хотим предсказать

X - признаки модели

b - bias (свободный член)

Линейная модель

На практике никогда не встречаются точные линейные модели, поэтому работаем в предположении, что

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

Diagram illustrating the components of a linear model:

- Dependent Variable (Y_i)
- Population Y intercept (β_0)
- Population Slope Coefficient (β_1)
- Independent Variable (X_i)
- Random Error term (ε_i)

The equation is divided into two main components:

- Linear component: $\beta_0 + \beta_1 X_i$
- Random Error component: ε_i

Linear regression

100
010

Линейная регрессия

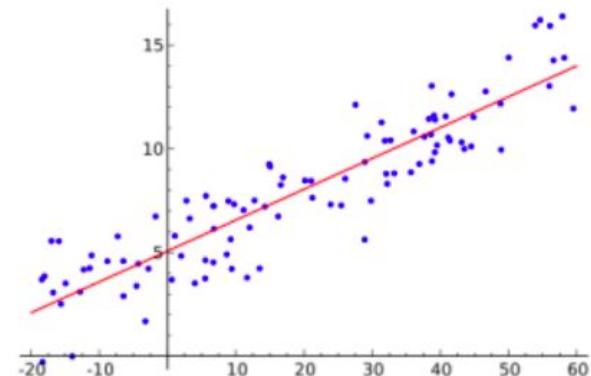


Линейная регрессия

$$\vec{y} = X\vec{w} + \epsilon,$$

где

- $\vec{y} \in \mathbb{R}^n$ – объясняемая (или целевая) переменная;
- w – вектор параметров модели (в машинном обучении эти параметры часто называют весами);
- X – матрица наблюдений и признаков размерности n строк на $m + 1$ столбцов (включая фиктивную единичную колонку слева) с полным рангом по столбцам: $\text{rank}(X) = m + 1$;
- ϵ – случайная переменная, соответствующая случайной, непрогнозируемой ошибке модели.



Линейная регрессия

Постановка задачи:

$$y = WX$$

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1n} & 1 \\ x_{21} & x_{22} & x_{23} & \dots & x_{2n} & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{d1} & x_{d2} & x_{d3} & \dots & x_{dn} & 1 \end{bmatrix}$$

Линейная регрессия

Постановка задачи:

$$y = WX$$

Минимизируем:

$$\|y - Xw\|^2 \rightarrow \min_w$$

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1n} & 1 \\ x_{21} & x_{22} & x_{23} & \dots & x_{2n} & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{d1} & x_{d2} & x_{d3} & \dots & x_{dn} & 1 \end{bmatrix}$$

Линейная регрессия

Постановка задачи:

$$y = WX$$

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1n} & 1 \\ x_{21} & x_{22} & x_{23} & \dots & x_{2n} & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{d1} & x_{d2} & x_{d3} & \dots & x_{dn} & 1 \end{bmatrix}$$

Минимизируем:

$$\|y - Xw\|^2 \rightarrow \min_w$$

Решение:

$$\hat{y} = Xw = X(X^T X)^{-1} X^T y$$

Какие могут возникнуть проблемы?

Линейная регрессия

Решение проблемы: Регуляризация

$$W_{ans} = (W^T W + \lambda I)^{-1} W^T y$$

I -- единичная матрица

Ridge regression

100
010

Регуляризация

Для борьбы с переобучением в оптимизационную задачу добавляется регуляризация:

$$Q(w) = \sum_{i=1}^l \mathcal{L}_i(w) + R(w) \rightarrow \min_w$$

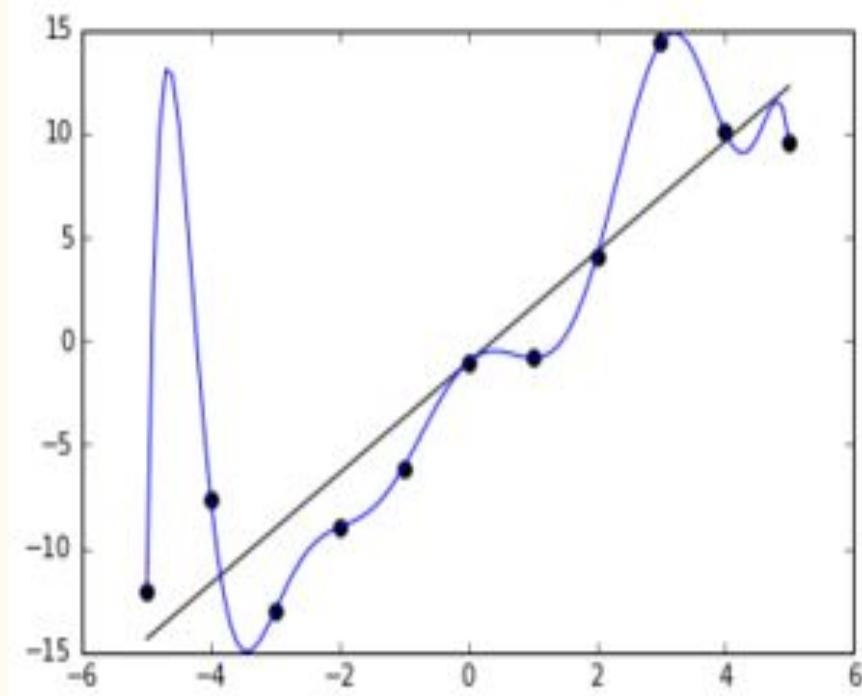
$R(w)$ – штраф за сложность модели

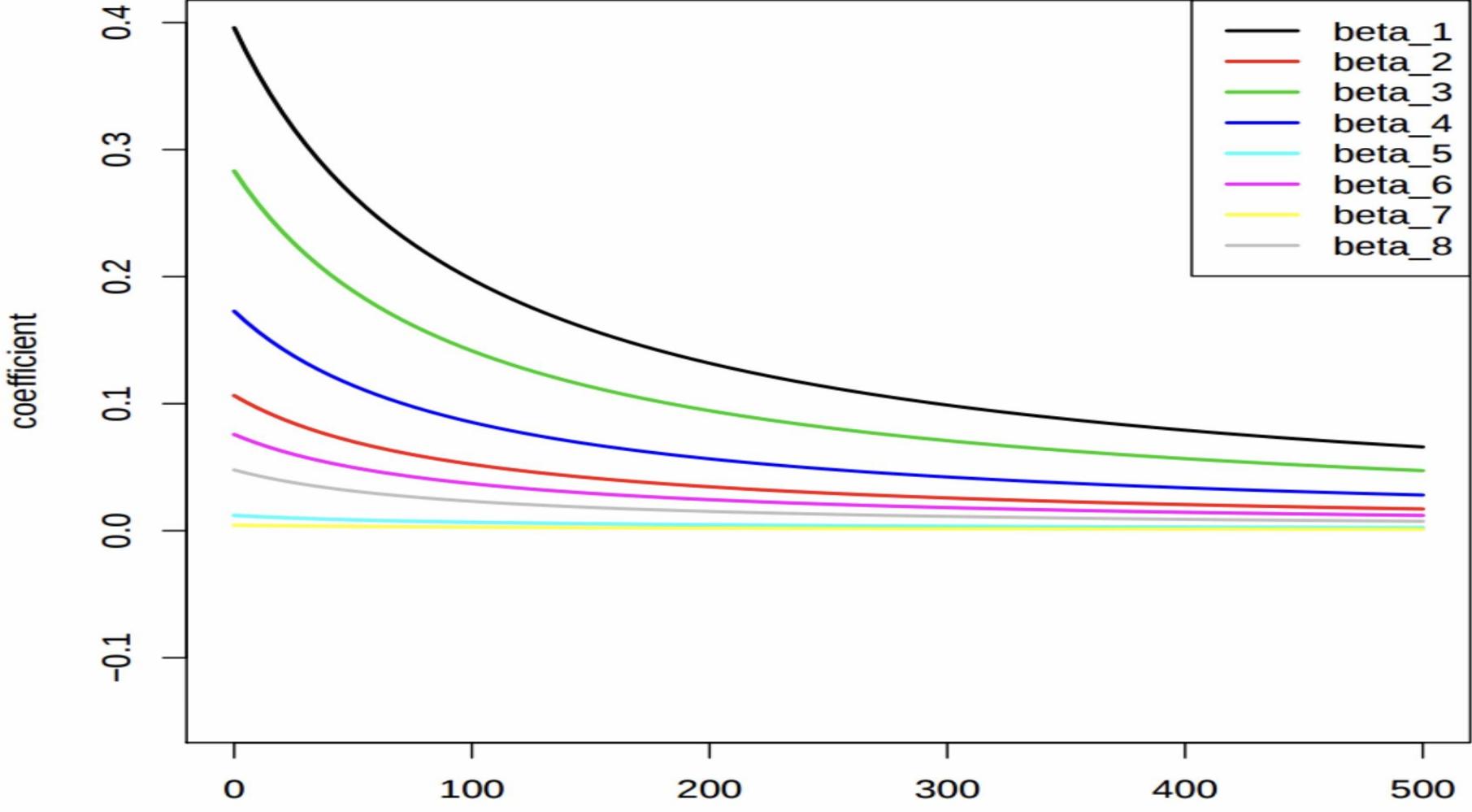
Переобучение

Большие веса модели, как правило, свидетельствуют о наличии переобучения

Для борьбы с ним, предлагается дополнительно штрафовать модель за большие веса

$$Q_\tau = \|y - X\theta\|^2 + \tau \|\theta\|^2 \rightarrow \min_{\theta}$$





Lasso regression

100
010

Регуляризация

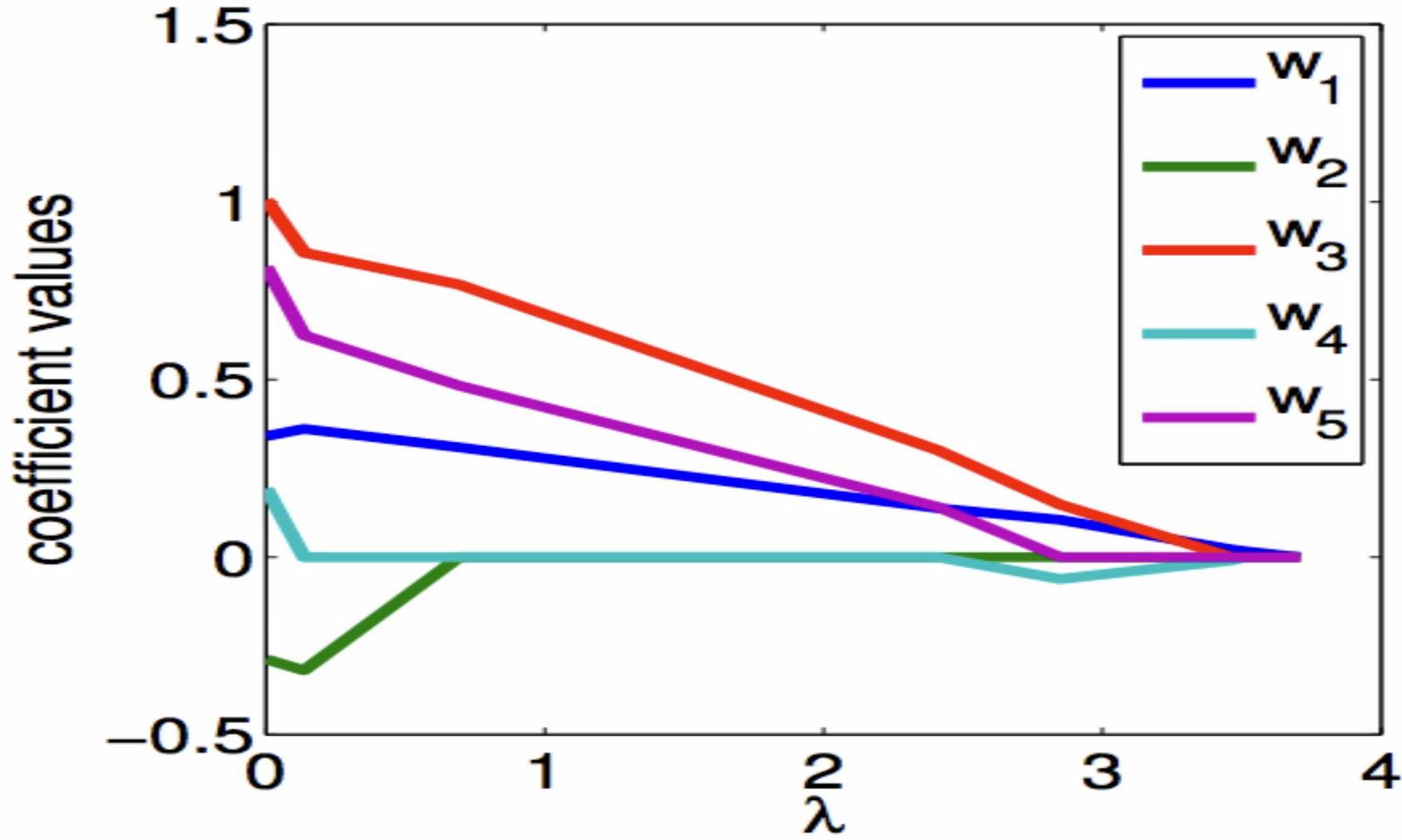
Для борьбы с переобучением в оптимизационную задачу добавляется регуляризация:

$$Q(w) = \sum_{i=1}^l \mathcal{L}_i(w) + R(w) \rightarrow \min_w$$

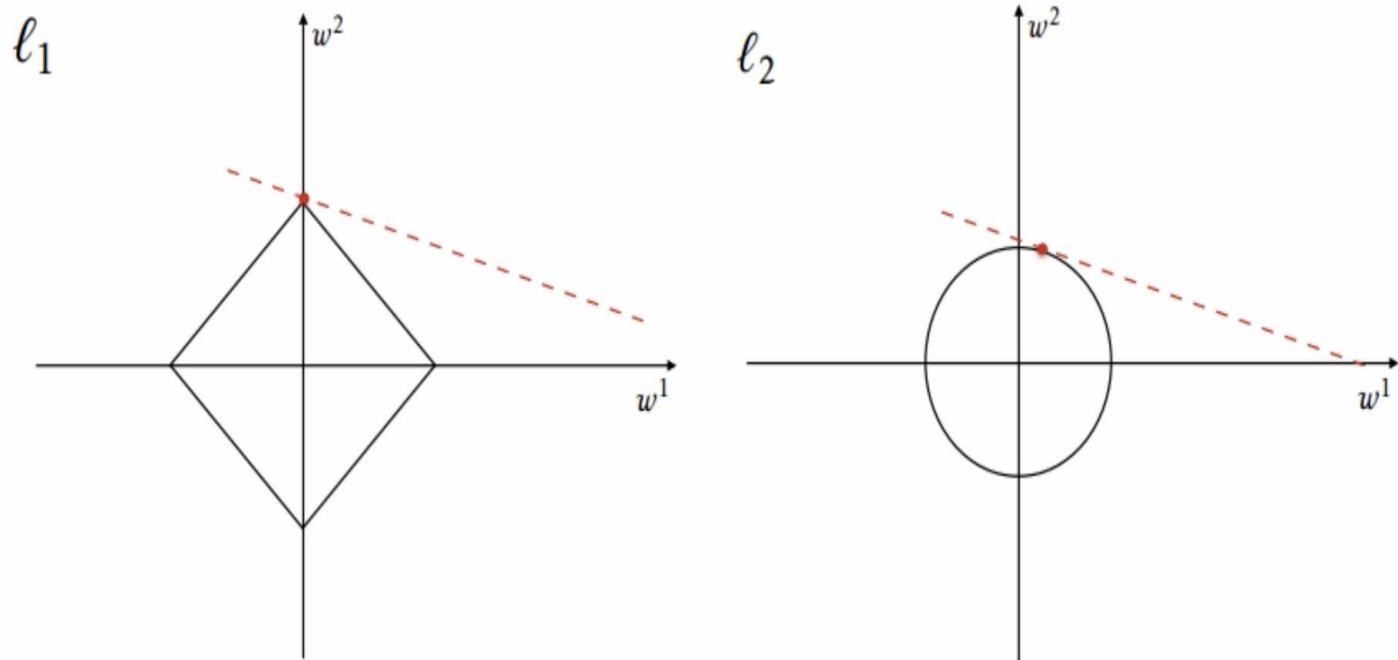
$R(w)$ – штраф за сложность модели

Lasso регрессия

$$\|y - X\theta\|^2 + \tau \sum_{j=1}^k |\theta_j| \rightarrow \min_{\theta}$$



Регуляризация



- L_1 : $R(w) = ||w||_1$ - обычно используется как метод отбора признаков
- L_2 : $R(w) = ||w||_2$ - улучшает качество модели

Linear classifiers

100
010

Решающее правило

Положим $Y = \{-1, +1\}$.

Линейным классификатором называется алгоритм классификации $a: X \rightarrow Y$ вида

$$a(x, w) = \text{sign} \left(\sum_{j=1}^n w_j f_j(x) - w_0 \right) = \text{sign} \langle x, w \rangle,$$

Обучение классификаторов

Метод минимизации эмпирического риска

Обучение (настройка) линейного классификатора методом [минимизации эмпирического риска](#) заключается в том, чтобы по заданной [обучающей выборке](#) пар «объект, ответ» $\mathbf{X}^m = \{(x_1, y_1), \dots, (x_m, y_m)\}$ построить алгоритм $a: \mathbf{X} \rightarrow Y$ указанного вида, минимизирующий [функционал эмпирического риска](#):

$$Q(w) = \sum_{i=1}^m [a(x_i, w) \neq y_i] \rightarrow \min_w.$$

Методы обучения линейных классификаторов различаются подходами к решению данной оптимизационной задачи.

Понятие отступа

В случае двух классов, $Y = \{-1, +1\}$, удобно определить для произвольного обучающего объекта $x_i \in \mathbf{X}^m$ величину [отступа](#) (margin):

$$M(x_i) = y_i \langle x_i, w \rangle.$$

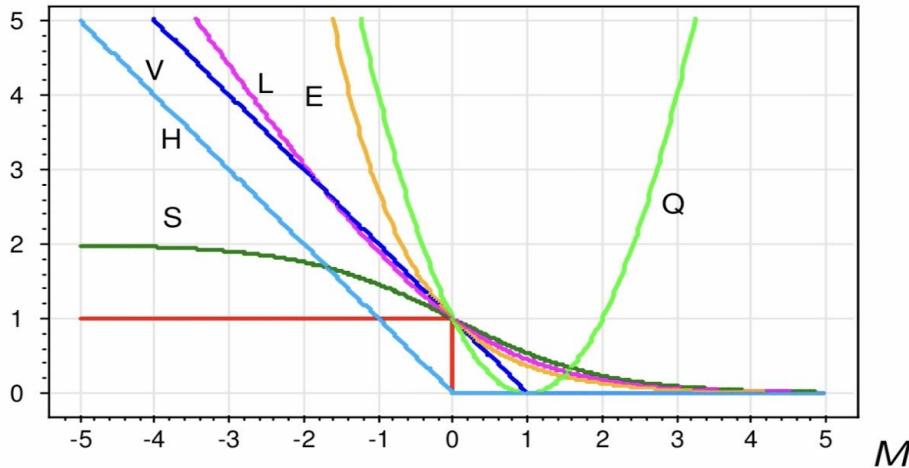
В случае произвольного числа классов отступ определяется выражением

$$M(x_i) = \langle x_i, w_{y_i} \rangle - \max_{y \in Y, y \neq y_i} \langle x_i, w_y \rangle.$$

Отступ можно понимать как «степень погруженности» объекта в свой класс. Чем меньше значение отступа $M(x_i)$, тем ближе объект подходит к границе классов, тем выше становится вероятность ошибки. Отступ $M(x_i)$ отрицателен тогда и только тогда, когда алгоритм $a(x)$ допускает ошибку на объекте x_i . Это наблюдение позволяет записать функционал [эмпирического риска](#) в следующем виде:

$$Q(w) = \sum_{i=1}^m [M(x_i) < 0].$$

Часто используемые непрерывные функции потерь $\mathcal{L}(M)$:



$$V(M) = (1 - M)_+$$

— кусочно-линейная (SVM);

$$H(M) = (-M)_+$$

— кусочно-линейная (Hebb's rule);

$$L(M) = \log_2(1 + e^{-M})$$

— логарифмическая (LR);

$$Q(M) = (1 - M)^2$$

— квадратичная (FLD);

$$S(M) = 2(1 + e^M)^{-1}$$

— сигмоидная (ANN);

$$E(M) = e^{-M}$$

— экспоненциальная (AdaBoost);

$$[M < 0]$$

— пороговая функция потерь.

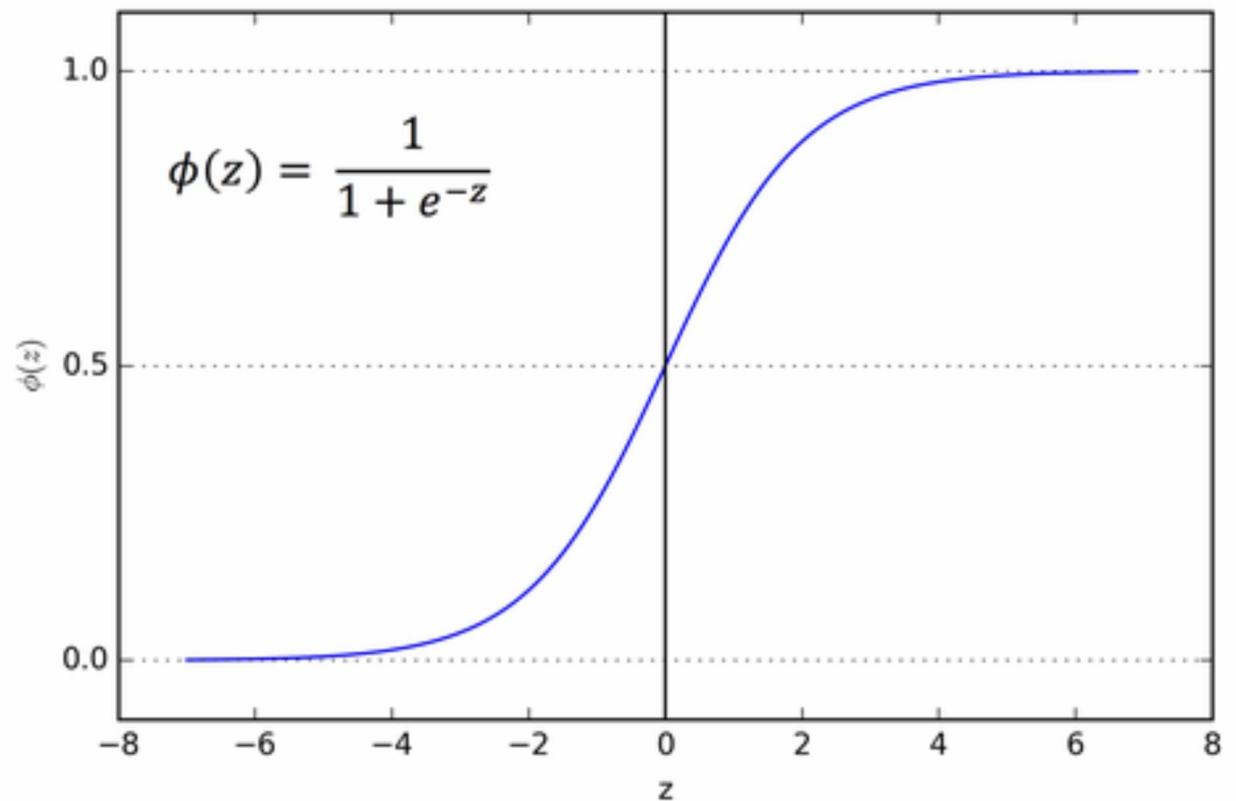
Logistic regression

100
010

Вопрос

Как по значению отступа M можно вычислить вероятность
принадлежности классу?

Сигмоида



Логистическая регрессия

Задача обучения линейного классификатора заключается в том, чтобы по выборке X^m настроить вектор весов w .

В логистической регрессии для этого решается задача [минимизации эмпирического риска](#) с функцией потерь специального вида:

$$Q(w) = \sum_{i=1}^m \ln(1 + \exp(-y_i \langle x_i, w \rangle)) \rightarrow \min_w \quad (1)$$

После того, как решение w найдено, становится возможным не только вычислять классификацию $a(x) = \text{sign}\langle x, w \rangle$ для произвольного объекта x , но и оценивать апостериорные вероятности его принадлежности классам:

$$\mathbb{P}\{y|x\} = \sigma(y \langle x, w \rangle), \quad y \in Y, \quad (2)$$

где $\sigma(z) = \frac{1}{1+e^{-z}}$ — [сигмоидная функция](#). Во многих приложениях апостериорные вероятности необходимы для оценивания рисков, связанных с возможными ошибками классификации.

SVM

100
010

SVM—Support Vector Machine

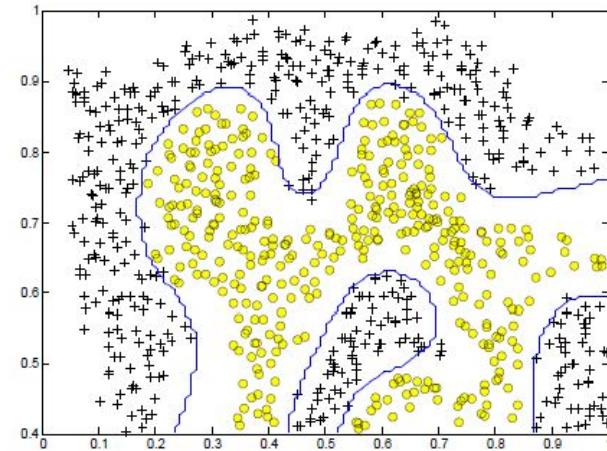
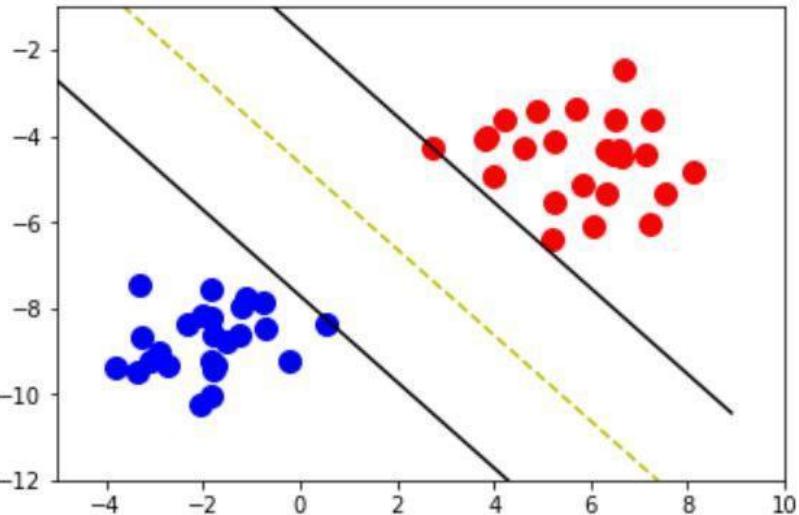


Figure 5: SVM (Gaussian Kernel) Decision Boundary (Example Dataset 2)

<http://www.machinelearning.ru/wiki/index.php?title=SVM>

Идея SVM

Линейный классификатор: $a(x, w) = \text{sign}(\langle w, x \rangle - w_0)$

Пусть выборка $X^\ell = (x_i, y_i)_{i=1}^\ell$ линейно разделима:

$$\exists w, w_0 : M_i(w, w_0) = y_i(\langle w, x_i \rangle - w_0) > 0, \quad i = 1, \dots, \ell$$

Нормировка: $\min_{i=1, \dots, \ell} M_i(w, w_0) = 1$

Разделяющая полоса (разделяющая гиперплоскость посередине):

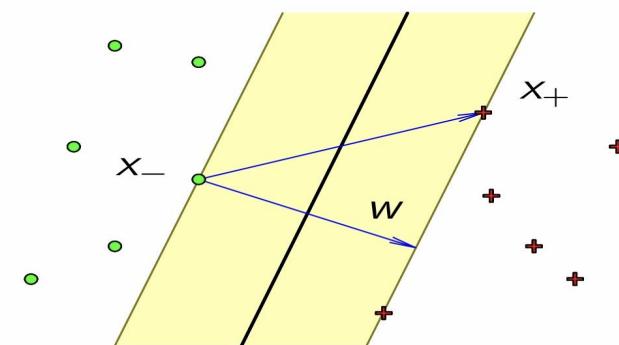
$$\{x : -1 \leq \langle w, x \rangle - w_0 \leq 1\}$$

$$\exists x_+ : \langle w, x_+ \rangle - w_0 = +1$$

$$\exists x_- : \langle w, x_- \rangle - w_0 = -1$$

Ширина полосы:

$$\frac{\langle x_+ - x_-, w \rangle}{\|w\|} = \frac{2}{\|w\|} \rightarrow \max$$



Что оптимизируем?

Линейно разделимая выборка

$$\begin{cases} \frac{1}{2} \|w\|^2 \rightarrow \min_{w, w_0}; \\ M_i(w, w_0) \geq 1, \quad i = 1, \dots, \ell. \end{cases}$$

Переход к линейно неразделимой выборке (эвристика)

$$\begin{cases} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{\ell} \xi_i \rightarrow \min_{w, w_0, \xi}; \\ M_i(w, w_0) \geq 1 - \xi_i, \quad i = 1, \dots, \ell; \\ \xi_i \geq 0, \quad i = 1, \dots, \ell. \end{cases}$$

Эквивалентная задача безусловной минимизации:

$$C \sum_{i=1}^{\ell} (1 - M_i(w, w_0))_+ + \frac{1}{2} \|w\|^2 \rightarrow \min_{w, w_0}.$$

Kernel trick

Если честно решить задачу оптимизации эмпирического риска в случае SVM, то получим решение:

Линейный классификатор с признаками $f_i(x) = \langle x, x_i \rangle$:

$$a(x) = \text{sign} \left(\sum_{i=1}^{\ell} \lambda_i y_i \langle x, x_i \rangle - w_0 \right).$$

Kernel trick

Если честно решить задачу оптимизации эмпирического риска в случае SVM, то получим решение:

Линейный классификатор с признаками $f_i(x) = \langle x, x_i \rangle$:

$$a(x) = \text{sign}\left(\sum_{i=1}^{\ell} \lambda_i y_i \langle x, x_i \rangle - w_0\right).$$

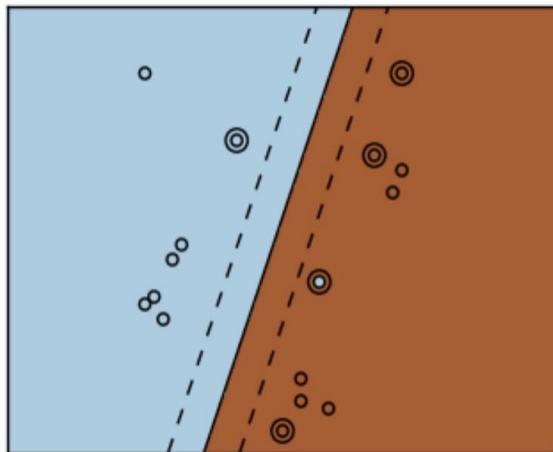
Идея: заменить $\langle x, x' \rangle$ нелинейной функцией $K(x, x')$.

Kernel trick

Примеры с различными ядрами $K(x, x')$

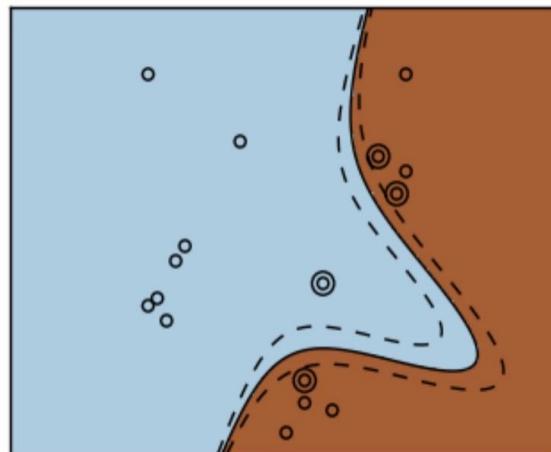
линейное

$$\langle x, x' \rangle$$



полиномиальное

$$(\langle x, x' \rangle + 1)^d, \quad d=3$$



гауссовское (RBF)

$$\exp(-\gamma \|x - x'\|^2)$$

