Логистическая регрессия и метод опорных векторов

МАКСИМОВСКАЯ АНАСТАСИЯ

Задача классификации

Задача классификации — задача, в которой имеется множество объектов, разделённых некоторым образом на классы. Задано конечное множество объектов, для которых известно, к каким классам они относятся.

Линейный классификатор — алгоритм классификации, основанный на построении линейной разделяющей поверхности.

Бинарная классификация — это задача классификации элементов заданного множества в две группы.

Целевая переменная в задаче классификации – класс, к которому принадлежит наблюдение.

Логистическая регрессия

ЭТО КЛАССИФИКАТОР :)

Линейная регрессия

$$a(x) = w_0 + w_1 \cdot x_1 + \ldots + w_d \cdot x_d = w_0 + \sum_{i=1}^{d} w_i \cdot x_i$$

Линейная регрессия

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

Параметры линейной регрессии – веса (коэффициенты w_j). Вес w_0 называется свободным коэффициентом или сдвигом (bias). Заметим, что после знака суммы написано скалярное произведение. Также добавим в выборку w_{d+1} признак, равный единице, тогда необходимость в свободном коэффициенте отпадет. Перепишем формулу в более компактном виде:

$$a(x) = \langle w, x \rangle$$

Логистическая регрессия

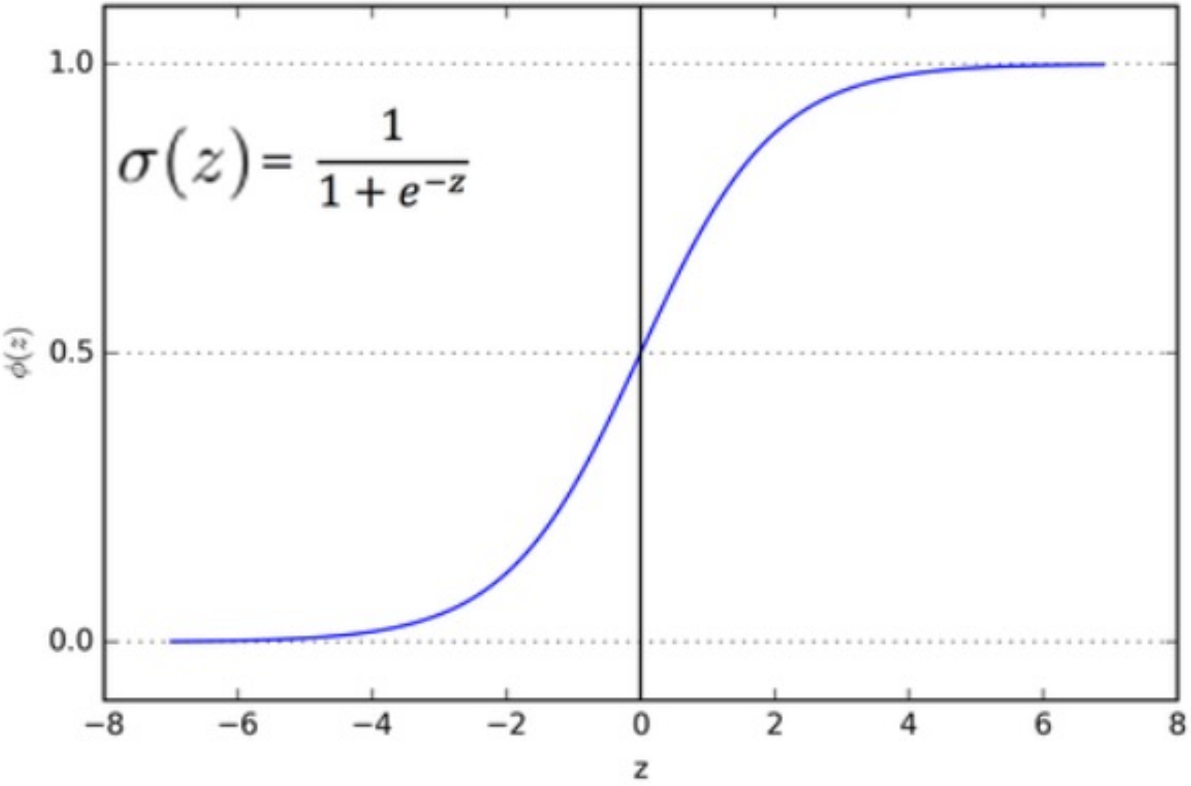
- Решаем задачу классификации и хотим предсказать вероятности классов
- Идея: давайте возьмем какую-нибудь функцию от $< w, x >$, чтобы результат попал в отрезок от 0 до 1

$$a(x, w) = g(< x, w >)$$

$$g(z) = \frac{1}{1 + e^{-z}}$$

- $g(z)$ – сигмоида

Логистическая регрессия



Логистическая регрессия

$$a(x, w) = P(y = +1|x; w)$$

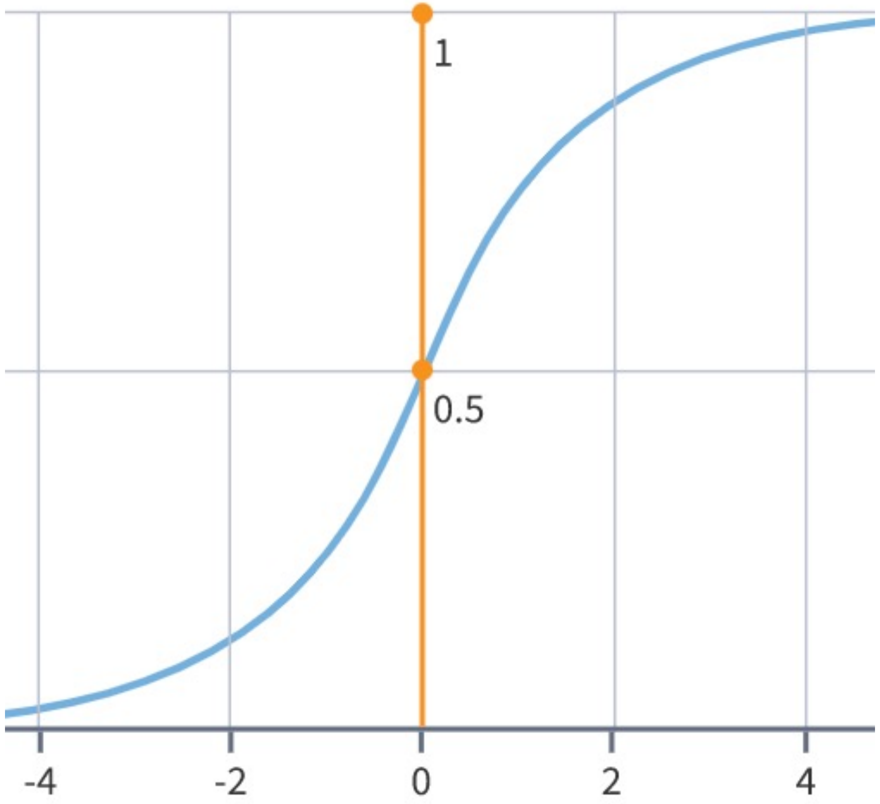
- Расшифровка: $a(x, w)$ – вероятность, что что объект x принадлежит к положительному классу
- Почему? Поможет записать и продифференцировать используя метода максимального правдоподобия

Логистическая регрессия

- Если $a(x, w) \geq 0.5$ предсказываем $y \ = \ +1$, иначе $y \ = -1$
- $a(x, w) = g(< x, w >) \geq 0.5$, если $< x, w > \geq 0$
- $< x, w > = 0$ – разделяющая гиперплоскость

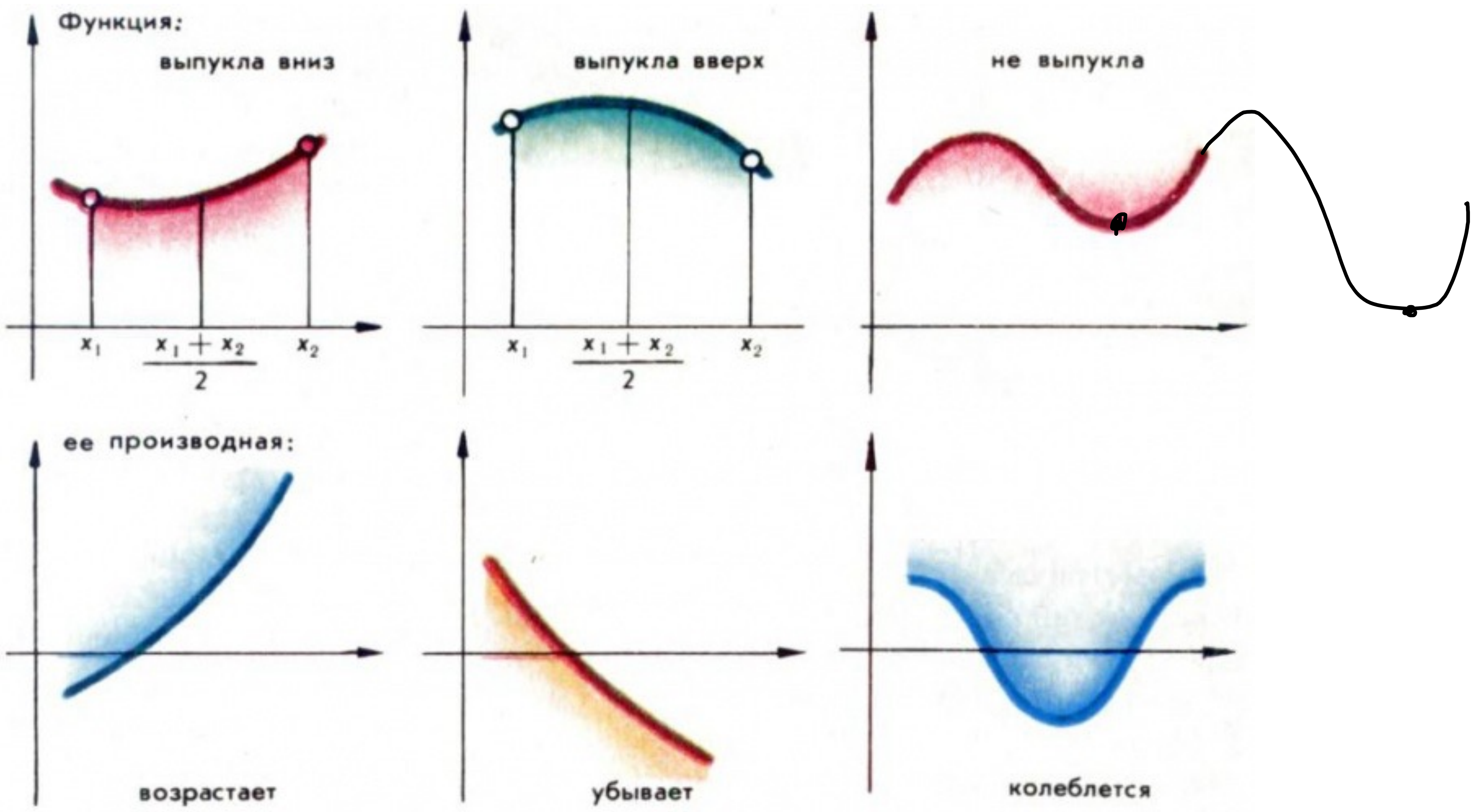
$y = +1 \quad < x, w > \geq 0$

$y = -1 \quad < x, w > < 0$



Функция потерь логистической регрессии

- Раньше брали квадратичную функцию потерь $L(a, y) = (a - y)^2$
- Но! Давайте подставим явную формулу $Q(a, X) = \frac{1}{l}\sum_{i=1}^{n}(\frac{1}{1+e^{-<x,w>}} - y)^2$ – это невыпуклая функция → можем не попасть в глобальный минимум при оптимизации


Функция:
выпукла вниз
выпукла вверх
не выпукла
x_1
$\frac{x_1 + x_2}{2}$
x_2
ее производная:
возрастает
убывает
колеблется

Функция потерь логистической регрессии

- Раньше брали квадратичную функцию потерь $L(a, y) = \ (a \ - y)^2$
- Но! Давайте подставим явную формулу $Q(a, X) = \ \frac{1}{l}\sum_{i=1}^{n}(\frac{1}{1+e^{<x,w>}} - y)^2$ – это невыпуклая функция → можем не попасть в глобальный минимум при оптимизации
- А еще будет низкий штраф за неверное предсказание: если у нас положительный объект, а модель предсказала $b(x) \ = 0$, то штраф будет всего лишь $(1 \ - 0)^2 = 1$

Функция потерь логистической регрессии

- Если наш алгоритм $b(x)$ и правда выдает вероятности, то они должны согласовываться с выборкой:

$$Q(a, X) = \prod_{i=1}^{\ell} b(x_i)^{[y_i=+1]} (1 - b(x_i))^{[y_i=-1]}.$$

- Оптимизировать удобнее логарифм данного функционала:

$$-\sum_{i=1}^{\ell} \left([y_i = +1] \log b(x_i) + [y_i = -1] \log(1 - b(x_i)) \right) \to \min$$

- Это log-loss – логарифмическая функция потерь

Функция потерь логистической регрессии

- Осталось показать, что мы корректно предсказываем вероятности. Запишем математическое ожидание функции потерь в точке x, продифференцируем по b и приравняем к 0:

$$\mathbb{E}\Big[L(y,b)\,|\,x\Big] = \mathbb{E}\Big[-[y=+1]\log b - [y=-1]\log(1-b)\,|\,x\Big] =$$
$$= -p(y=+1\,|\,x)\log b - (1-p(y=+1\,|\,x))\log(1-b)$$

Функция потерь логистической регрессии

$$\frac{\partial}{\partial b}\mathbb{E}\Big[L(y,b)\mid x\Big] = -\frac{p(y=+1\mid x)}{b} + \frac{1-p(y=+1\mid x)}{1-b} = 0$$

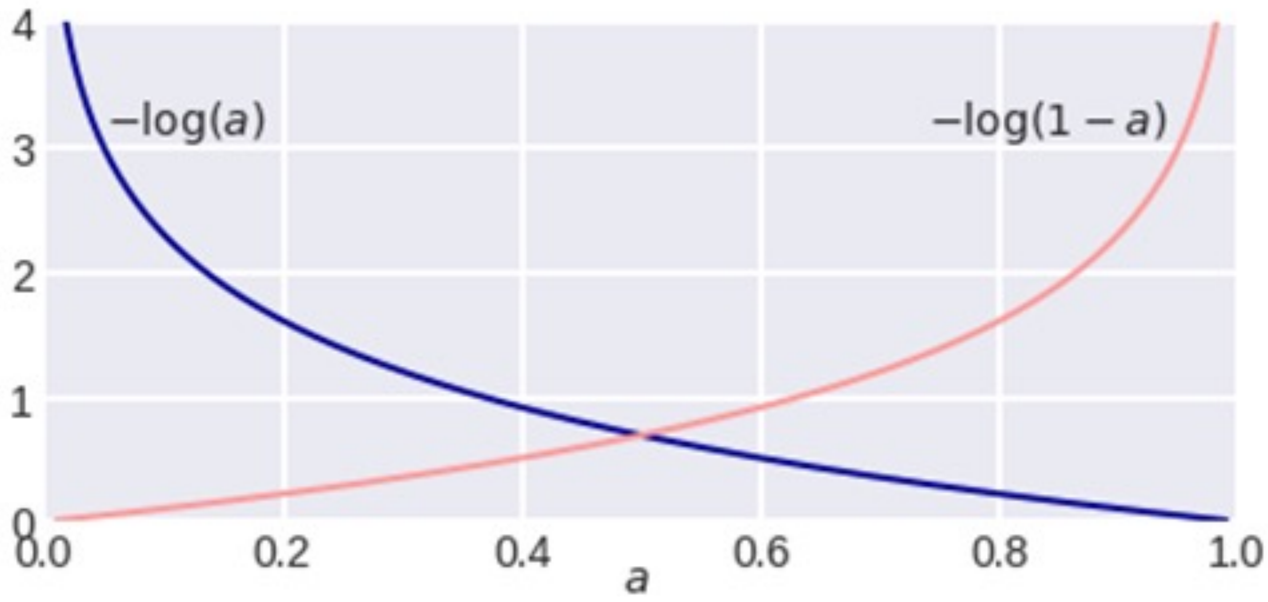
$$b_* = p(y=+1\mid x)$$

Функция потерь логистической регрессии

$$Q(w) = -\sum_{i=1}^{l}([y_i = +1] \cdot \log(a(x_i, w)) + [y_i = -1] \cdot \log(1 - a(x_i, w)))$$

Функция потерь логистической регрессии

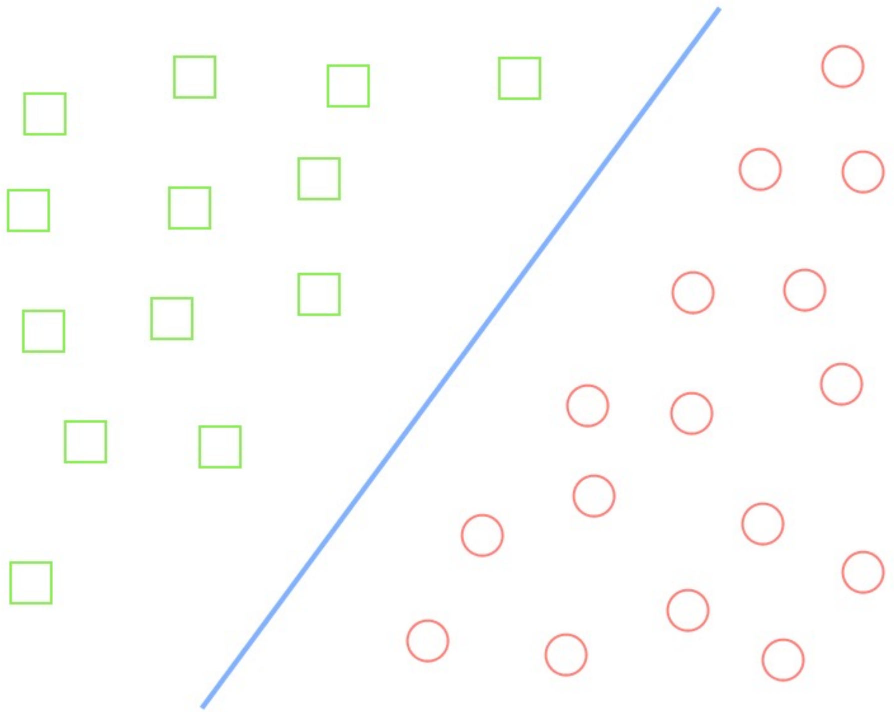
$$
-\begin{cases} \log a_i, & y_i = 1, \\ \log(1 - a_i), & y_i = 0. \end{cases}
$$



Метод опорных векторов

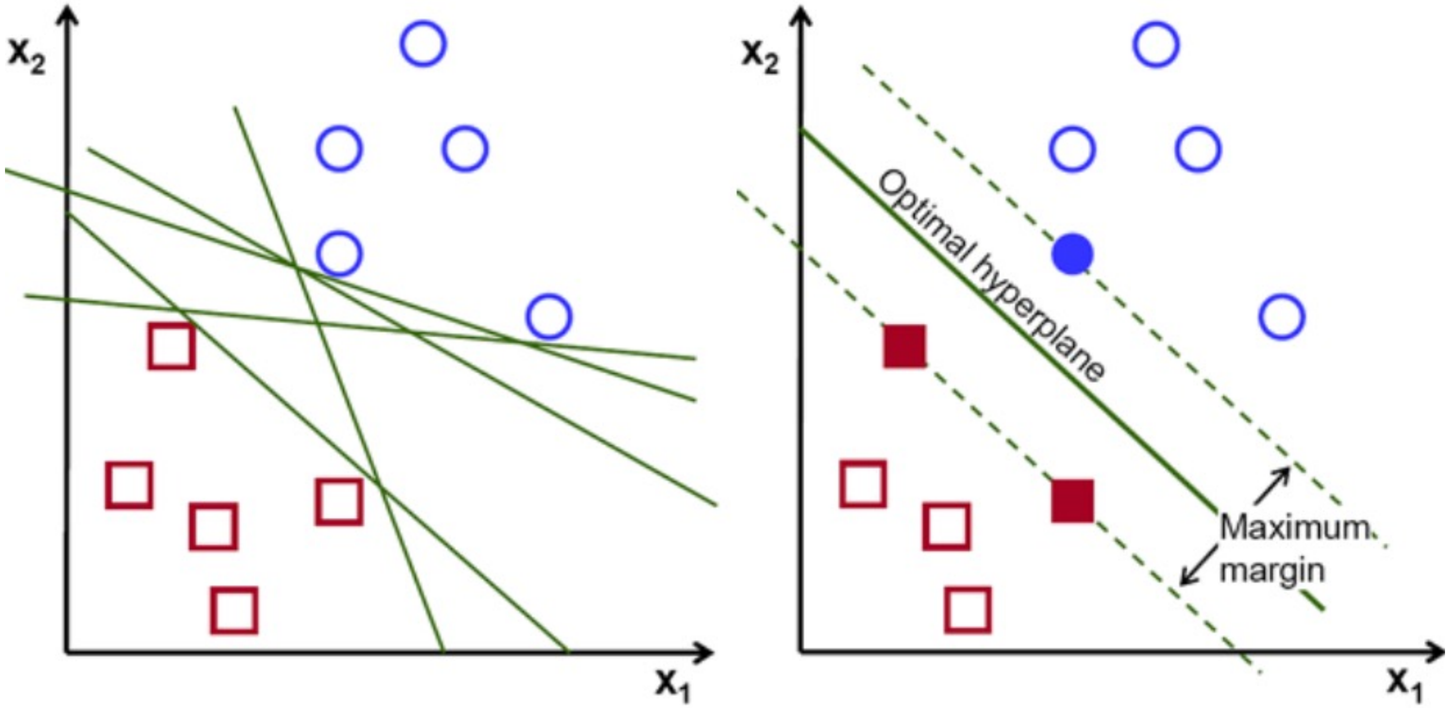
SVM: разделимый случай

Если существует такой вектор параметров w*, при котором алгоритм a(x) не ошибается на данной выборке, то выборка называется **линейно разделимой**



SVM: разделимый случай

- Но! На таких выборках можно провести не одну такую линию – нам хочется максимизировать ширину разделяющей полосы

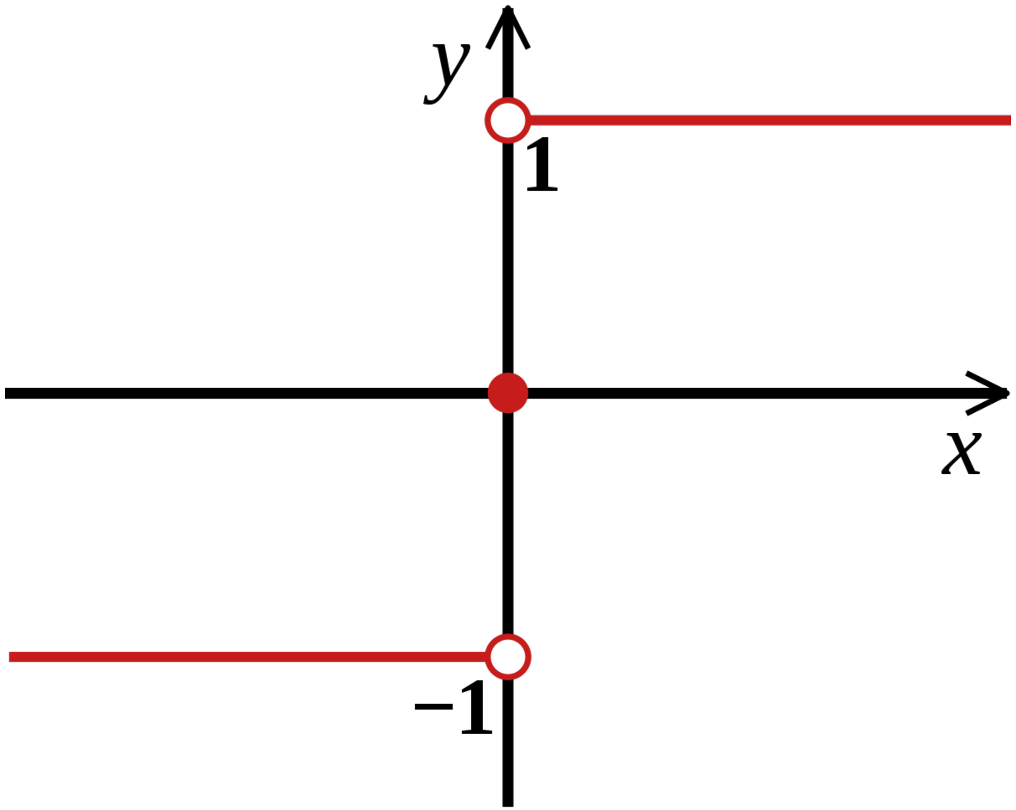


SVM: разделимый случай

- Зададим классификатор $a(x) = sign(< w, x > + w_0)$
- Раньше мы добавляли константный признак и не выносили w_0, сегодня нам удобнее его оставить

Напоминание

$$sgn(x) = \begin{cases} 1 & if \quad x > 0, \\ 0 & if \quad x = 0, \\ -1 & if \quad x < 0 \end{cases}$$



$w_0 + w_1 \cdot x_1 + w_2 \cdot x_2 + \dots$ −10 1

17 1.2

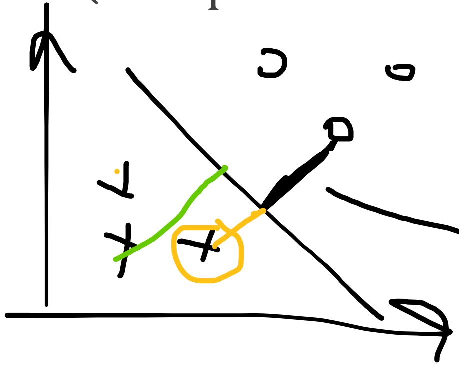
14 1.4

SVM: разделимый случай

- Зададим классификатор $a(x) = sign(\boxed{< w, x > \ + w_0})$
- Давайте нормируем значение того, что в скобочках, так, чтобы минимальное значение этого числа было равно 1:

$$\min_{x \in X} |< w, x > \ + w_0| = 1$$

- Можно показать, что расстояние от произвольной точки x_0 до разделяющей прямой (гиперплоскости, задаваемой нашим классификатором) будет равно:



$$\rho(x_0, a) = \frac{|\langle w, x \rangle + b|}{\|w\|}$$

SVM: разделимый случай

$$\rho(x_0, a) = \frac{|\langle w, x\rangle + b|}{\|w\|}$$

Сверху – модуль, снизу – норма вектора весов.

$$\|\mathbf{x}\|_1 = \sum_{i=1}^{m} |x_i| \qquad \|\mathbf{x}\|_2 = \left(\sum_{i=1}^{m} |x_i|^2\right)^{1/2} \qquad \|\mathbf{x}\|_\infty = \max_{1 \le i \le m} |x_i|$$

SVM: разделимый случай

- Имеем:

$$\min_{x \in X} |< w, x > \ + w_0| = 1$$

$$\rho(x_0, a) = \frac{|\langle w, x \rangle + b|}{||w||}$$

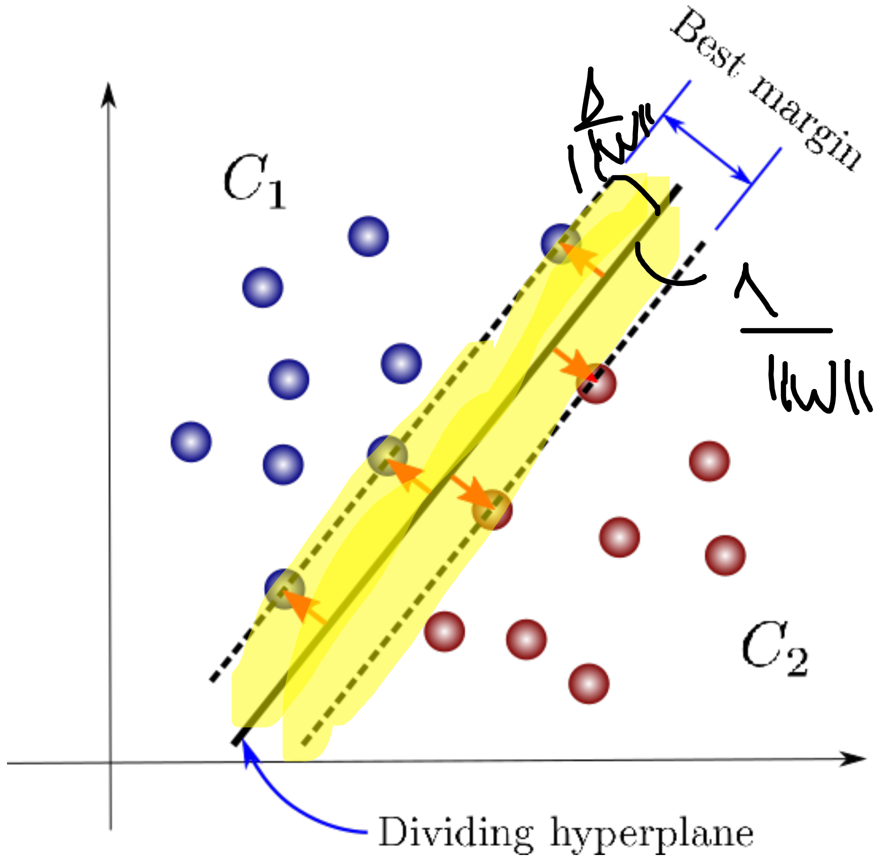
Значит, расстояние до ближайшего к разделяющей плоскости объекта будет $\frac{1}{||w||}$

SVM: разделимый случай

Значит, ширина разделяющей полосы – $\frac{2}{||w||}$

$$\frac{1}{||w||} + \frac{1}{||w||}$$

Цель метода опорных векторов – максимизировать ширину разделяющей полосы.



SVM: разделимый случай

Но мы любим минимизировать, а не максимизировать ☺

$$\begin{cases} \dfrac{1}{2}\|w\|^2 \to \min\limits_{w,b} \\ \underbrace{y_i\left(\langle w, x_i\rangle + b\right)}_{M} \underset{\geqslant 0}{\geqslant 1}, \quad i = 1, \ldots, \ell. \end{cases}$$

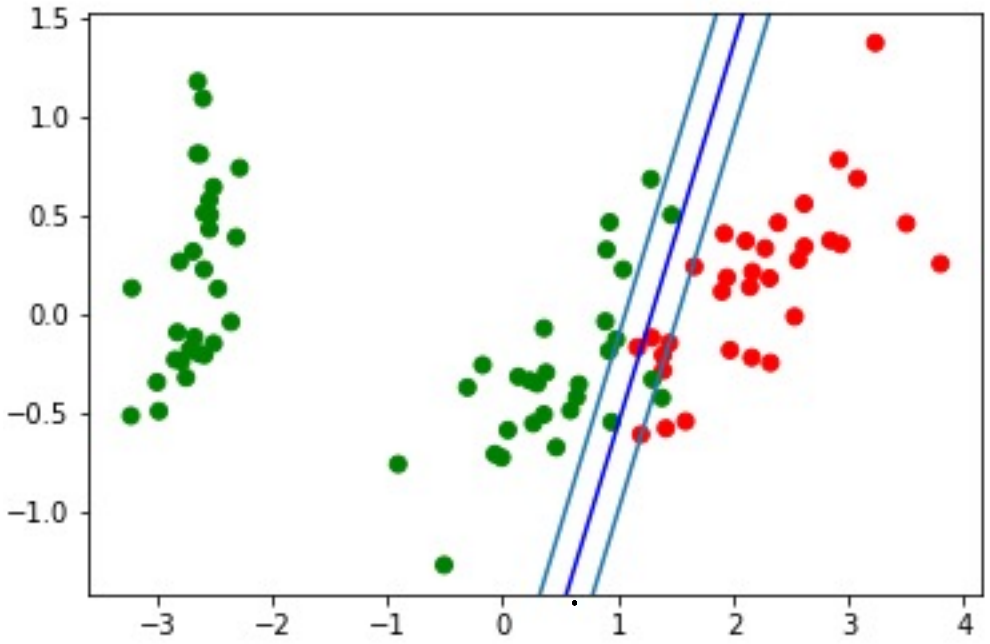
Второе условие означает, что выборка линейно разделима. Отступ ≥ 1

Функционал выпуклый, ограничение линейные – данная задача является выпуклой и имеет единственное решение.

SVM: неразделимый случай

Но в жизни все тяжело и выборка чаще всего линейно неразделима ☹

- Существует хотя бы один объект, такой, что $y(< w, x > \ + w_0) < 1$



SVM: неразделимый случай

- В чем беда?
- Мы все еще хотим разделяющую полосу пошире, чтобы классификатор был уверен в объектах, которые будут находиться вне разделяющей полосы
- Но теперь у нас будут объекты внутри разделяющей полосы. Мы хотим, чтобы их было поменьше
- Будем вводить штрафы $\xi_i \geq 0$, которые смягчат наше ограничение:

$$\overbrace{y_i\left(\langle w, x_i\rangle + b\right)}^{M} \geqslant 1 - \xi_i, \quad i = 1, \ldots, \ell$$

$0 \leq M < 1$

SVM: неразделимый случай

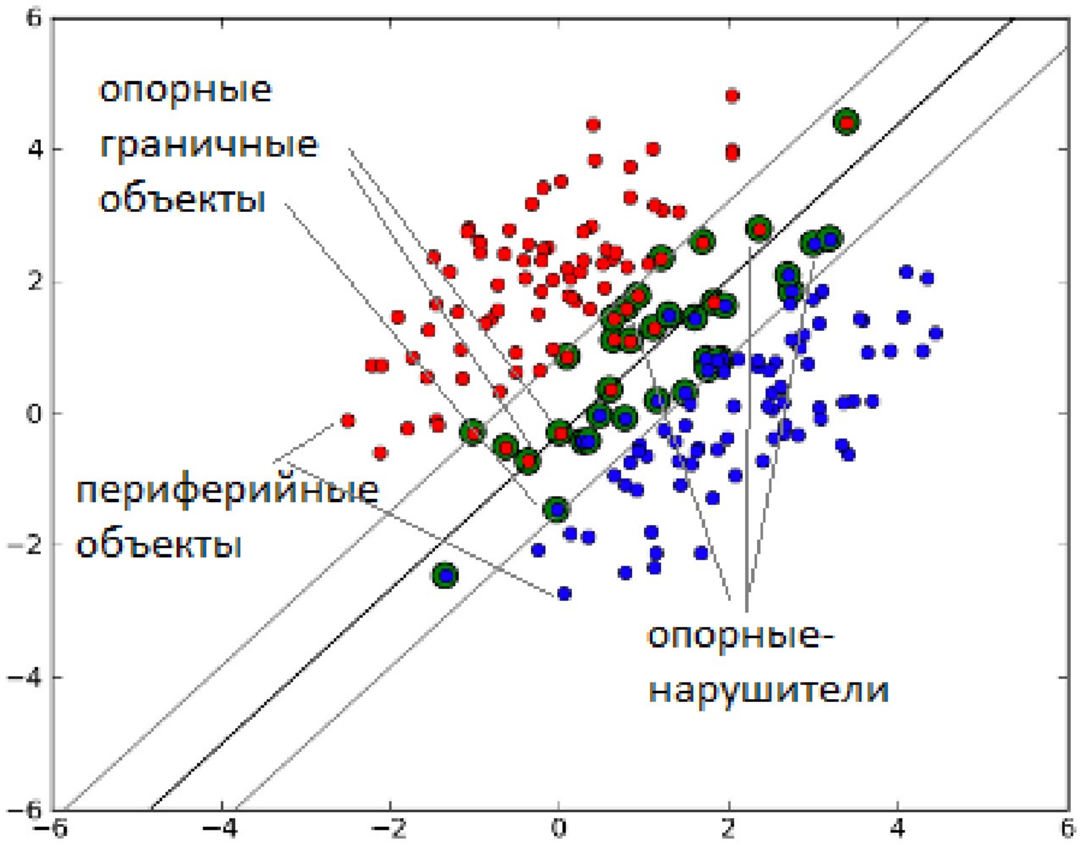
- Штрафы хотим минимизировать, а отступ – максимизировать

$$\begin{cases} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{\ell}\xi_i \to \min\limits_{w,b,\xi} \\ y_i\left(\langle w, x_i\rangle + b\right) \geqslant 1 - \xi_i, \quad i = 1, \dots, \ell, \\ \xi_i \geqslant 0, \quad i = 1, \dots, \ell. \end{cases}$$

- Параметр C отвечает за подгонку под обучающую выборку, позволяет находить компромисс между максимизацией разделяющей полосы и минимизацией суммарной ошибки

- Данная задача также является выпуклой и имеет единственное решение

Типы объектов



SVM: сведение к безусловной задаче

- Перепишем условия задачи:

$$\begin{cases} \xi_i \geqslant 1 - y_i(\langle w, x_i \rangle + b) \\ \xi_i \geqslant 0 \end{cases}$$

- Хотим, чтобы штрафы были как можно меньше. Тогда можно записать такую формулу:

$$\xi_i = \max(0, 1 - y_i(\langle w, x_i \rangle + b)).$$

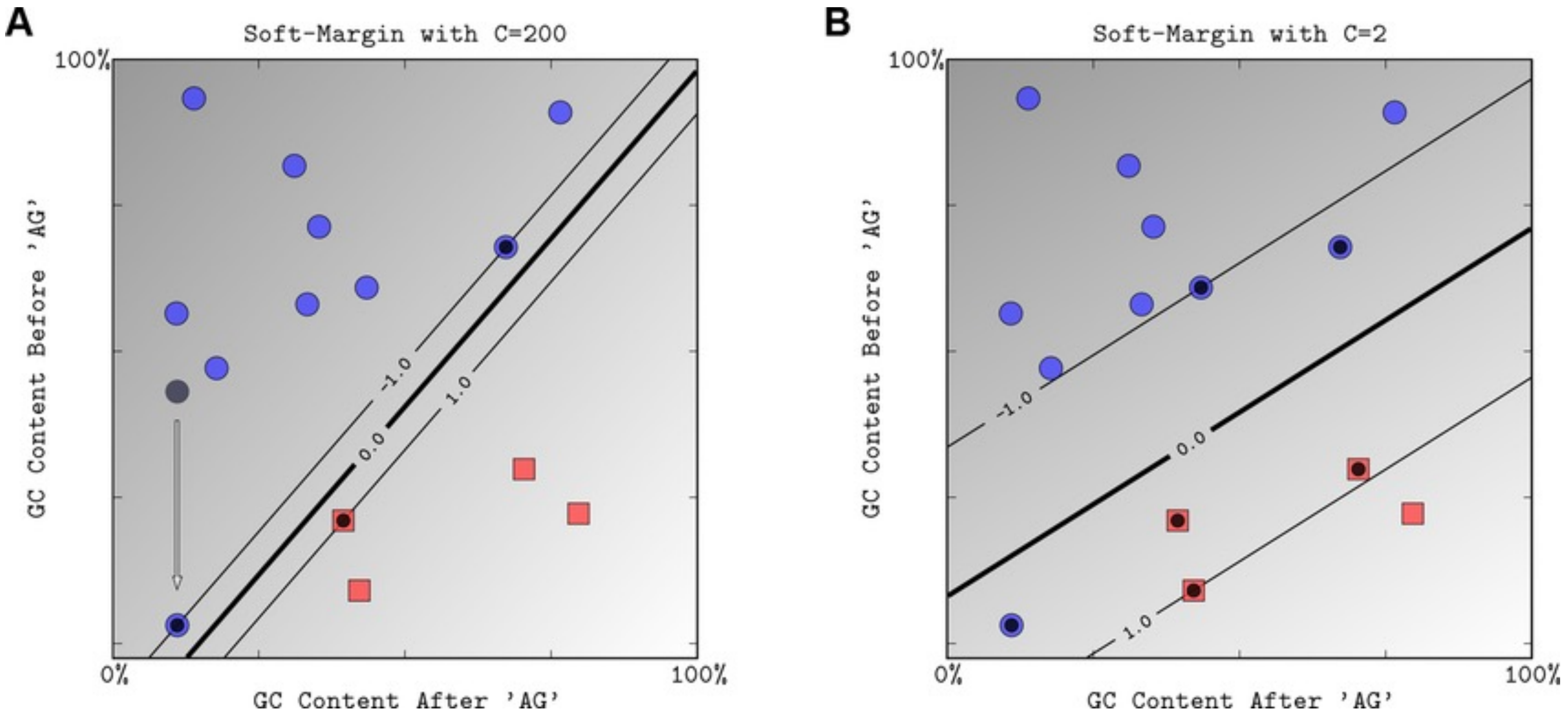
- Подставим ее функционал. Т.к. мы учли все ограничения, то можем решать безусловную задачу оптимизации:

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{\ell} \max(0, 1 - y_i(\langle w, x_i \rangle + b)) \to \min_{w,b}$$

SVM: задача оптимизации

- На задачу оптимизации SVM теперь можно смотреть как на оптимизацию функции потерь $L(M) = \max(0, 1 - M) = (1 - M)_+$ с регуляризацией:

$$Q(a, X) = \sum_{i=1}^{l} \left(1 - M_i(w, w_0)\right)_+ + \frac{1}{2C} ||w||^2 \to \min_{w, w_0}$$


A
Soft-Margin with C=200
100%
GC Content Before 'AG'
GC Content After 'AG'
0%
100%
-1.0
0.0
1.0
B
Soft-Margin with C=2
100%
GC Content Before 'AG'
GC Content After 'AG'
0%
100%
-1.0
0.0
1.0

Kernel trick

Иногда бывает так:

Fig.3

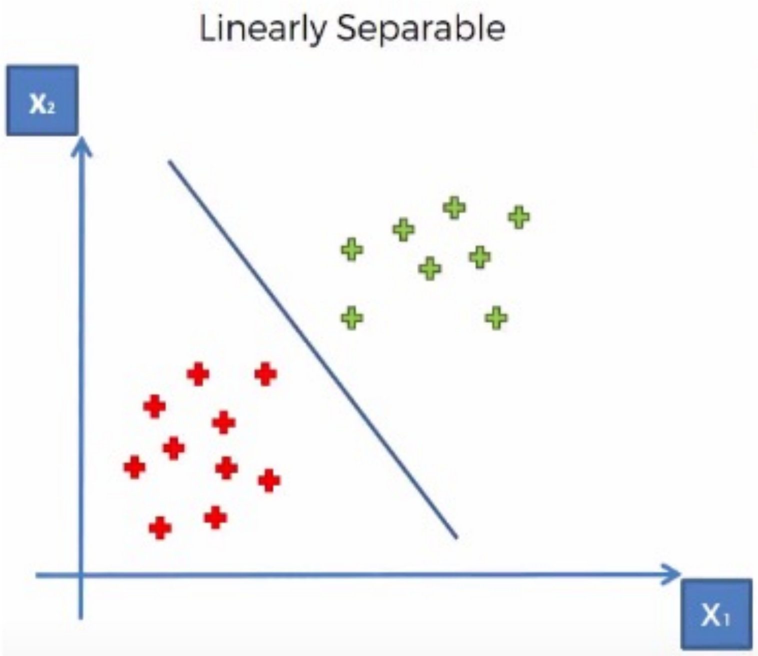
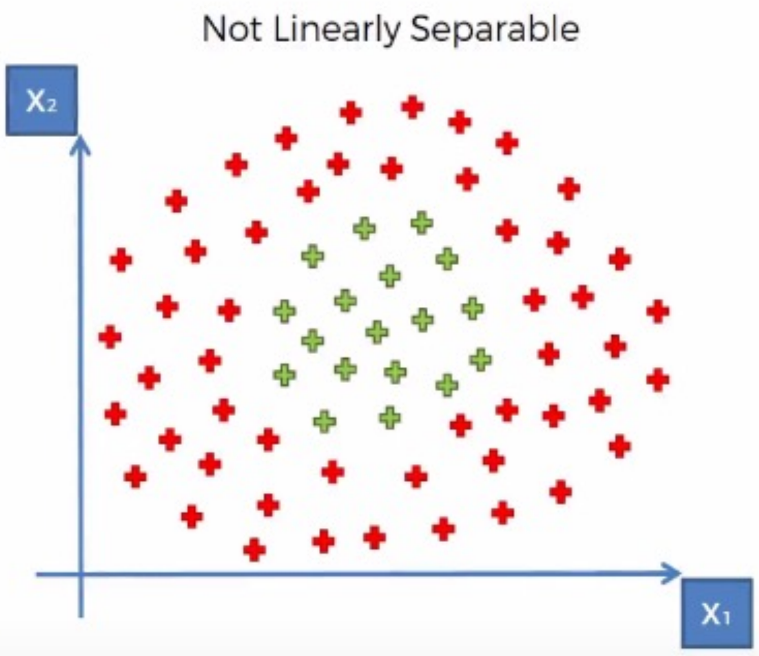
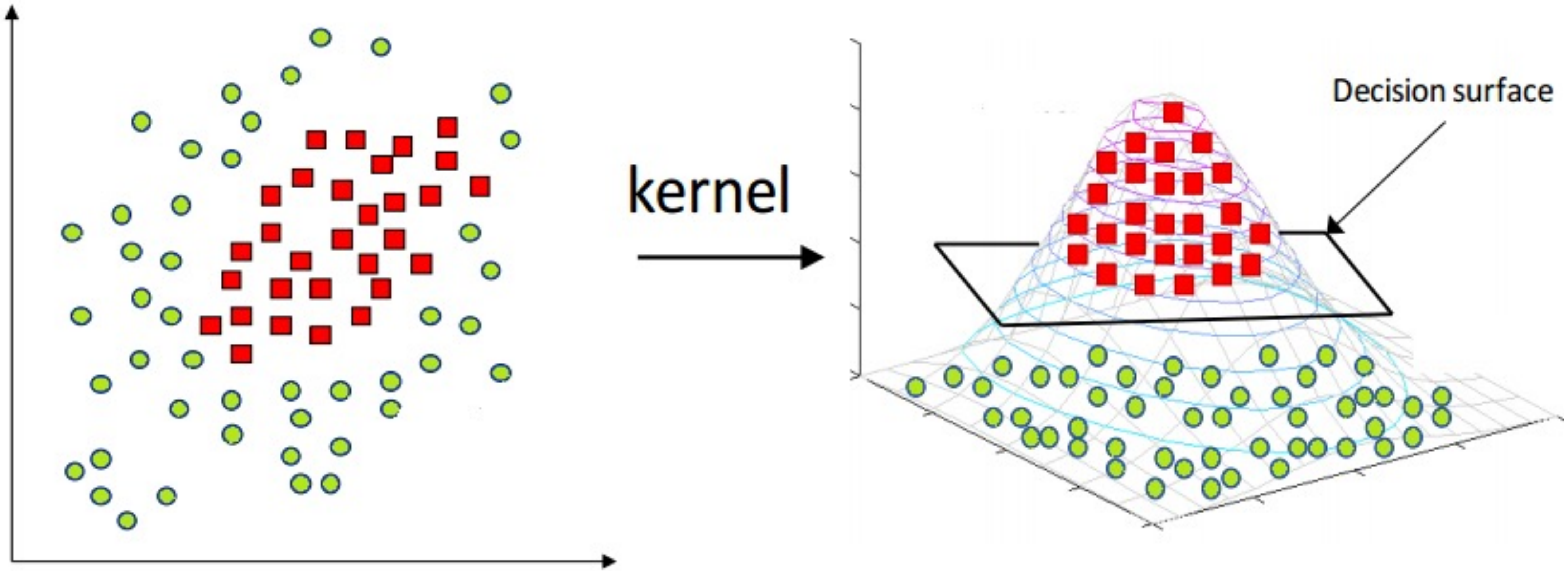


Fig.4

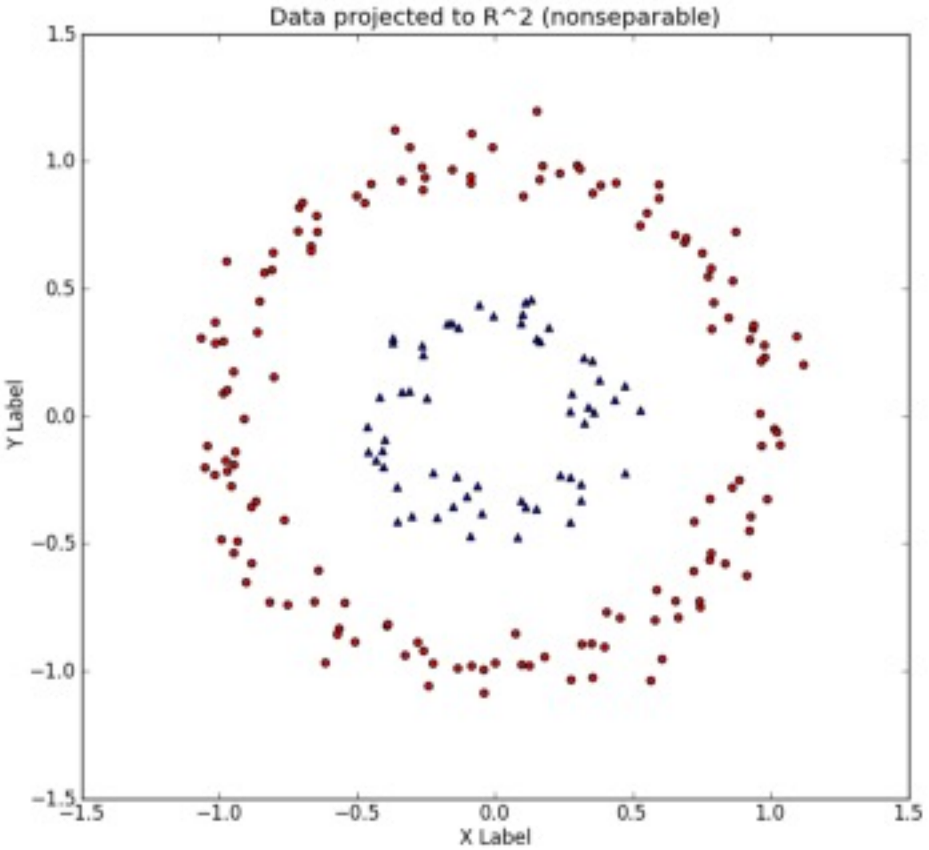


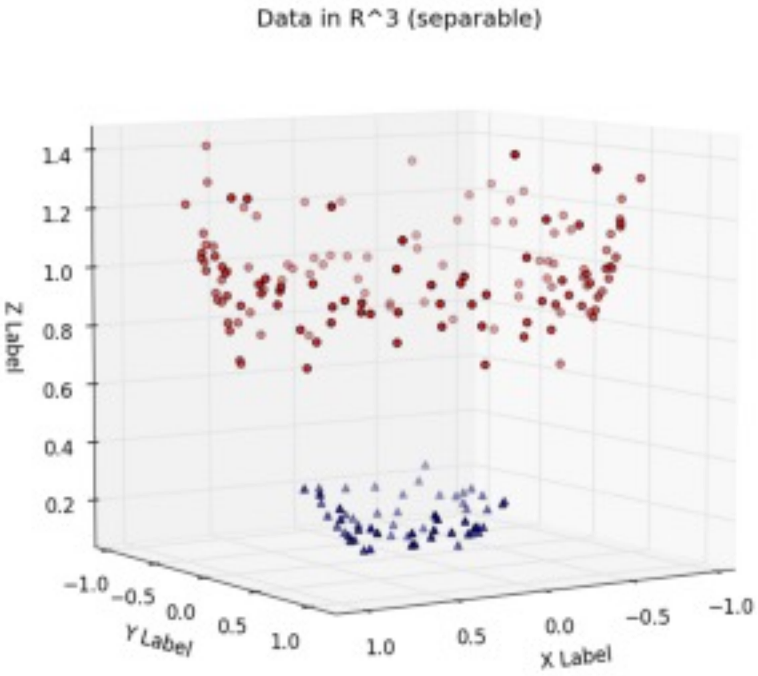
Kernel trick

- Если исходная выборка линейно неразделима, то может существовать такое преобразование координат, при котором выборка становится линейно разделимой
- Применение преобразования координат и метода главных компонент называется ядровым методом опорных векторов

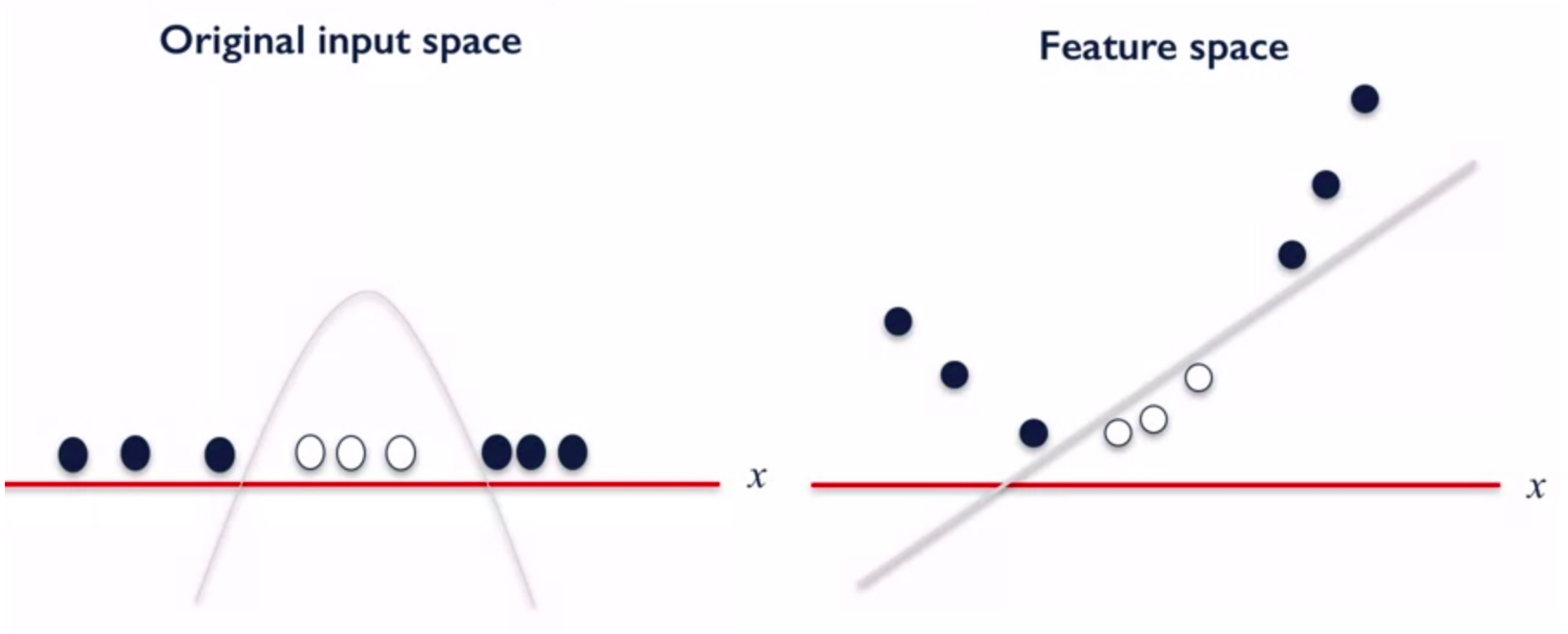

kernel
Decision surface

RBF – Радиальное ядро

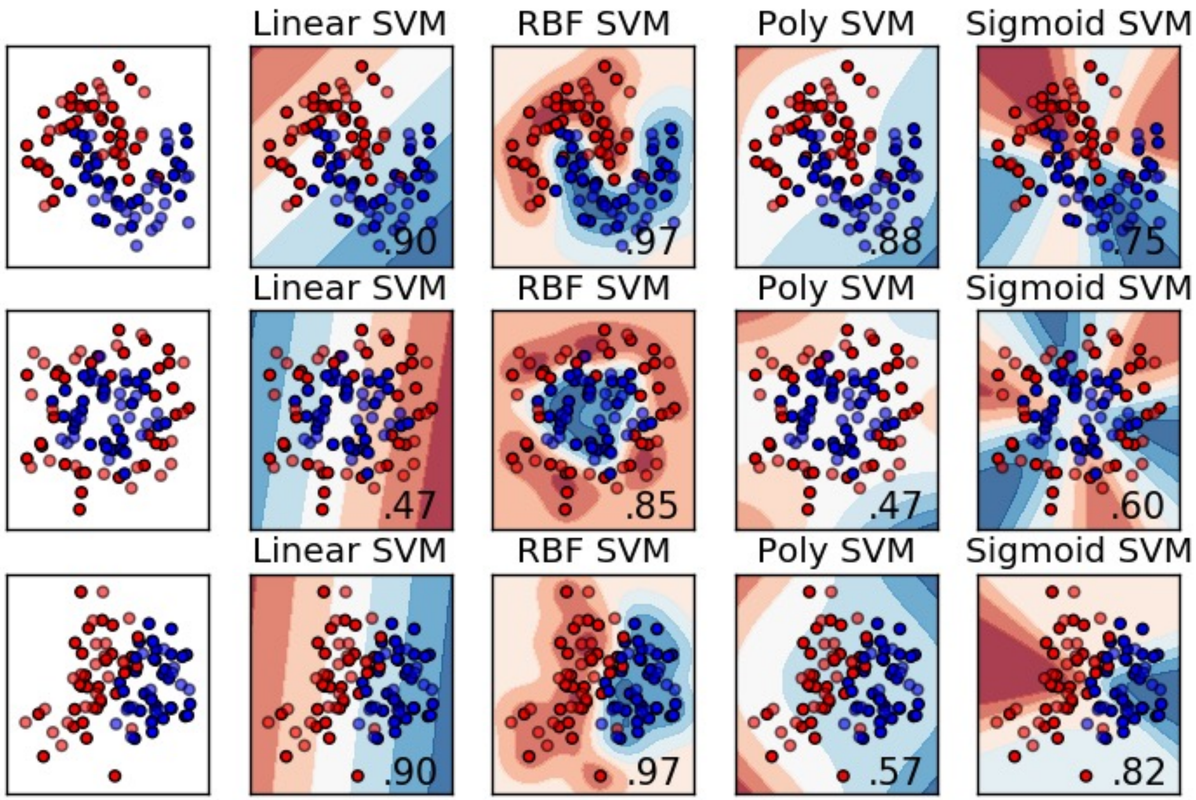




Полиномиальное ядро



Различные ядра



Калибровка вероятностей

Калибровка вероятностей – приведение ответов алгоритма к значениям, близким к вероятностям принадлежности объекта к конкретному классу

Это важно для:

1. Правильного понимания, насколько результатам алгоритма можно доверять
2. Упрощения интерпретации
3. Настройки на функции ошибки

Почитать подробнее: URL

Параметрическая калибровка Платта

- Идея метода заключается в обучении логистической регрессии на ответах классификатора

$$\pi(x; \alpha; \beta) = \sigma(\alpha \cdot a(x) + \beta) = \frac{1}{1+e^{-(\alpha \cdot a(x)+\beta)}}$$

- Находим α и β минимизируя логистическую функцию потерь (т. е. обучая линейную регрессию) на отложенной выборке (calibration set)

Реализация логистической или изотонической регрессии:
[URL]