Машинное обучение (Machine Learning)

Визуализация данных. Математические модели и методы

Уткин Л.В.

ПОЛИТЕХ
Санкт-Петербургский политехнический университет Петра Великого

Содержание

Презентация является компиляцией и заимствованием материалов из замечательных курсов и презентаций по машинному обучению:

К.В. Воронцова, А.Г. Дьяконова, Н.Ю. Золотых, С.И. Николенко, Andrew Moore, Lior Rokach, Rong Jin, Luis F. Teixeira, Alexander Statnikov и других..

Метод визуализации

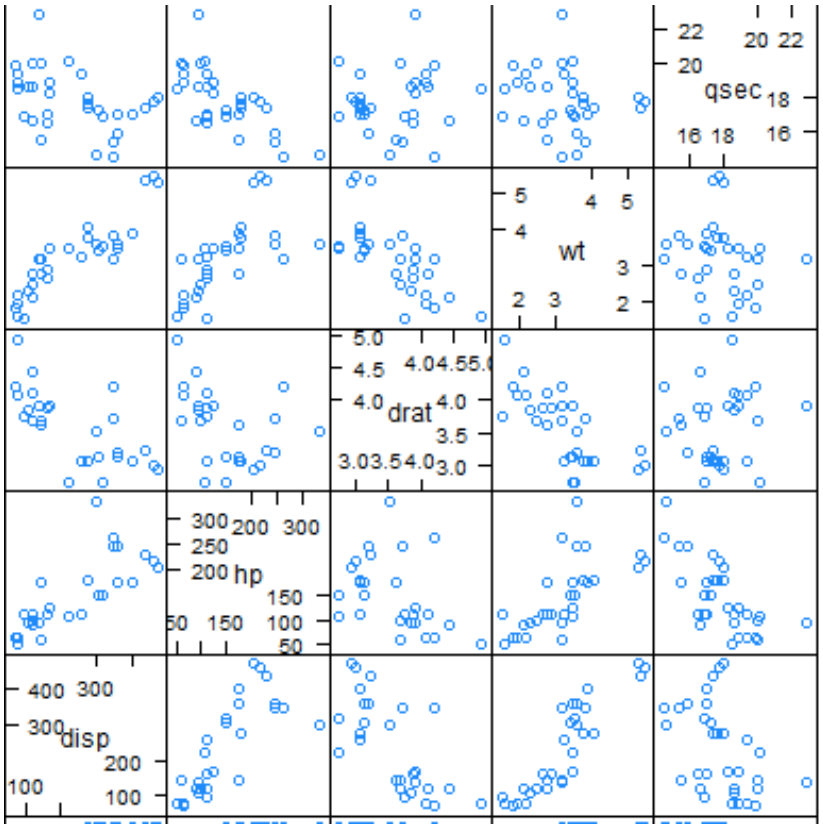
“Хорошая” визуализация (требования)

- Каждый объект с большой размерностью представляется объектом с малой размерностью.
- Сохранение “соседства ” двух объектов в разных пространствах.
- Удаленные точки соответствуют отличающимся объектам
- Масштабируемость

Более формально

- Точка данных - это точка x_i в исходном пространстве $\mathbb{R}^D$
- Образ - точка y_i в пространстве $\mathbb{R}^2$ или $\mathbb{R}^3$. Каждый образ соответствует одной исходной точке.
- Алгоритм визуализации выбирает положение образов в $\mathbb{R}^2$ или $\mathbb{R}^3$ в соответствии с определенными правилами (в основном для сохранения пространственной структуры данных)

Диаграммы рассеяния



Наивный байесовский классификатор

Теорема Байеса

Thomas Bayes

1702 - 1761

Теорема Байеса

$$P(y = c|x) = \frac{P(x|y = c)P(y = c)}{P(x)},$$

$P(y = c|x)$ - вероятность что объект x принадлежит классу c (апостериорная вероятность класса);
$P(x|y = c)$ - вероятность встретить объект x среди всех объектов класса c;
$P(y = c)$ - безусловная вероятность встретить объект класса c (априорная вероятность класса);
$P(x)$ - безусловная вероятность объекта x.

Теорема Байеса и классификация

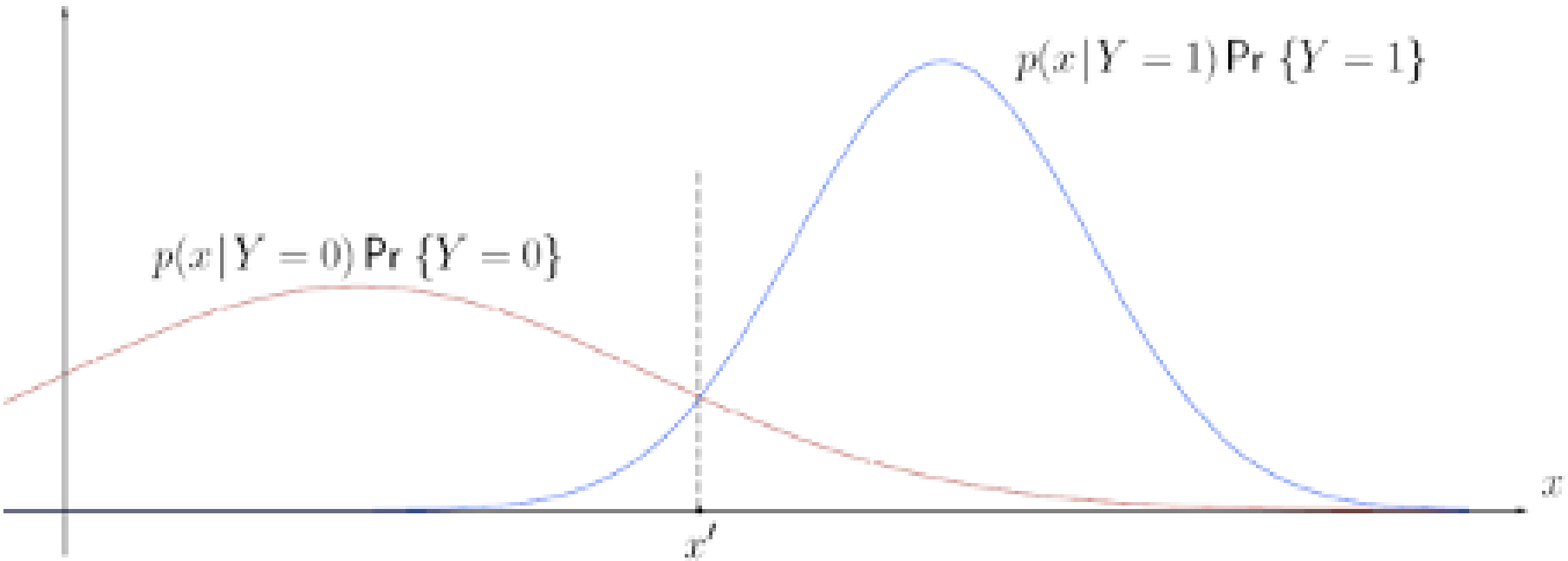
Цель классификации состоит в том чтобы понять к какому классу принадлежит объект x. Следовательно необходимо найти наиболее вероятный класс объекта x, т.е., необходимо из всех классов выбрать тот, который дает максимум вероятности $P(y = c|x)$:

$$c_{opt} = \arg\max_{c \in C} P(y = c|x) = \arg\max_{c \in C} \frac{P(x|y = c)P(y = c)}{P(x)}.$$

Для каждого класса c вычисляется $P(y = c|x)$ и выбирается класс, имеющий максимальную вероятность. Вероятность $P(x)$ не зависит от c и является константой::

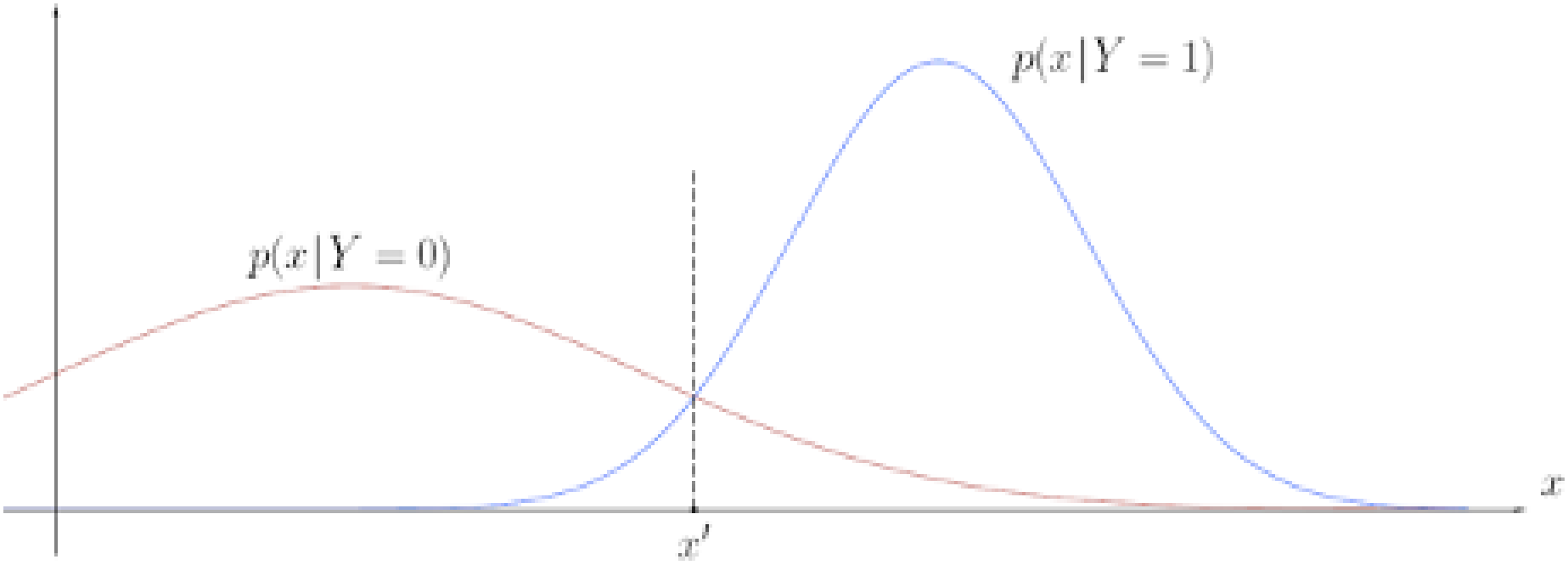
$$c_{opt} = \arg\max_{c \in C} P(x|y = c)P(y = c).$$

Принцип максимума апостериорной вероятности



При $x < x'$ считается $c_{opt} = 0$ иначе $c_{opt} = 1$

Принцип максимального правдоподобия



При $x < x'$ считается $c_{opt} = 0$ иначе $c_{opt} = 1$

Теорема Байеса и классификация (2 класса)

Выбор:

$$\begin{cases} \textit{класс } c_1, & \textit{если} P(y = c_1|x) > P(y = c_2|x) \\ \textit{класс} c_2, & \textit{иначе} \end{cases}$$

или

$$\begin{cases} \textit{класс} c_1, & \textit{если } \dfrac{P(x|y = c_1)}{P(x|y = c_2)} > \dfrac{P(y = c_2)}{P(y = c_1)} \\ \textit{класс } c_2, & \textit{иначе} \end{cases}$$

Байесовский классификатор минимизирует ошибку принятия решений

Наивность классификатора

Байесовский классификатор представляет объект как набор признаков (атрибутов), вероятности которых условно не зависят друг от друга:

$$P(x|y=c) = P(f_1|y=c)P(f_2|y=c)\cdots P(f_m|y=c) = \prod_{i=1}^{m} P(f_i|y=c).$$

Наивный байесовский классификатор:

$$c_{opt} = \arg\max_{c\in C} \left(P(y=c)\prod_{i=1}^{m} P(f_i|y=c)\right).$$

или

$$c_{opt} = \arg\max_{c\in C} \left(\log P(y=c) + \sum_{i=1}^{m} \log P(f_i|y=c)\right).$$

Математические модели и методы

Гипотезы компактности или непрерывности

Задачи классификации и регрессии:
X - объекты, Y - ответы; $X^n = (x_i, y_i)_{i=1}^n$ - обучающая выборка.

Гипотеза компактности (для классификации): Близкие объекты, как правило, лежат в одном классе.

Гипотеза непрерывности (для регрессии): Близким объектам соответствуют близкие ответы.

Метод k ближайших соседей

Метод k ближайших соседей (kNN — k nearest neighbours) метрический алгоритм для классификации объектов, основанный на оценивании сходства объектов. Классифицируемый объект относится к тому классу, которому принадлежат ближайшие к нему объекты обучающей выборки.

Алгоритм:

1. Вычислить расстояние до каждого из объектов обучающей выборки
2. Отобрать k объектов обучающей выборки, расстояние до которых минимально
3. Класс классифицируемого объекта — это класс, наиболее часто встречающийся среди k ближайших соседей

Мера близости

Что такое близкие объекты? Задана функция расстояния $\rho : X \times X \to [0, \infty)$.
Виды функций расстояния:

- Евклидово: $\rho(x_i, x_j) = \sqrt{\sum_{k=1}^{m} w_k \left(x_i^{(k)} - x_j^{(k)}\right)^2}$
- L_p-метрика: $\rho(x_i, x_j) = \left(\sum_{k=1}^{m} w_k \left|x_i^{(k)} - x_j^{(k)}\right|^p\right)^{1/p}$
- L_∞-метрика: $\rho(x_i, x_j) = \max_{k=1,\ldots,m} \left|x_i^{(k)} - x_j^{(k)}\right|$
- L_1-метрика: $\rho(x_i, x_j) = \sum_{k=1}^{m} \left|x_i^{(k)} - x_j^{(k)}\right|$

$x_i = (x_i^{(1)}, \ldots, x_i^{(m)})$ - вектор m признаков i-го объекта;
$x_j = (x_j^{(1)}, \ldots, x_j^{(m)})$ - вектор m признаков i-го объекта;

Мера близости

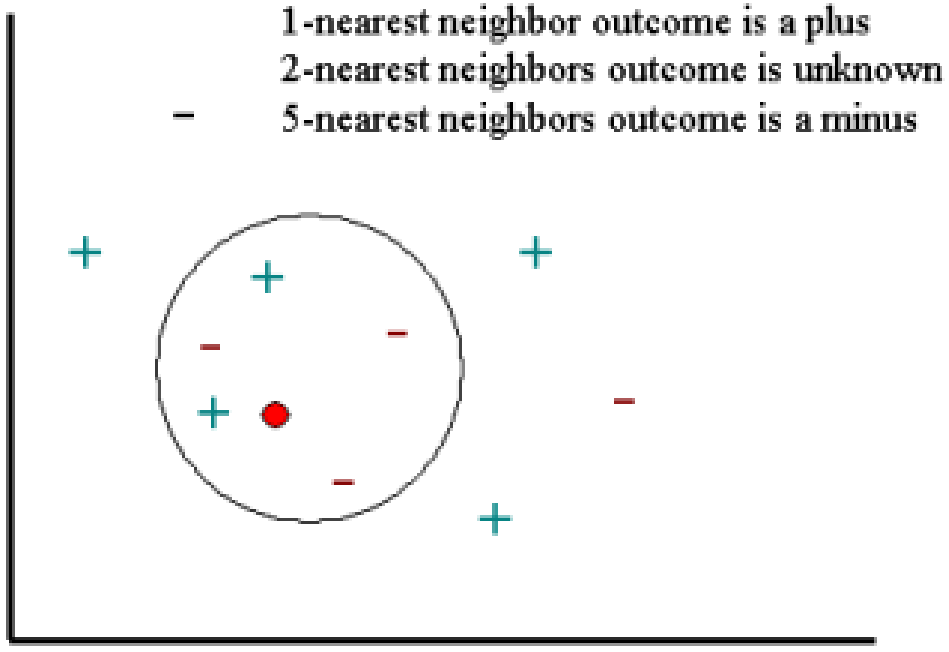
Что такое близкие объекты? Задана функция расстояния $\rho : X \times X \to [0, \infty)$.
Виды функций расстояния:

- Ланса-Уильямся: $\rho(x_i, x_j) = \frac{\sum_{k=1}^{m} \left| x_i^{(k)} - x_j^{(k)} \right|}{\sum_{k=1}^{m} \left(x_i^{(k)} + x_j^{(k)} \right)}$
- косинусная мера: $\rho(x_i, x_j) = \frac{\sum_{k=1}^{m} x_i^{(k)} x_j^{(k)}}{\sqrt{\sum_{k=1}^{m} \left(x_i^{(k)} \right)^2} \sqrt{\sum_{k=1}^{m} \left(x_j^{(k)} \right)^2}}$

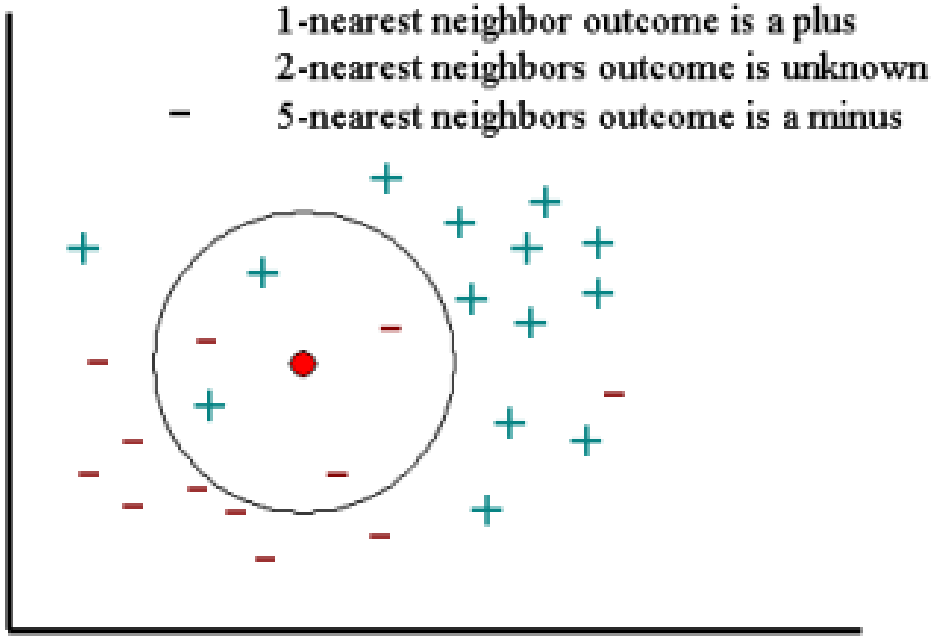
$x_i = (x_i^{(1)}, ..., x_i^{(m)})$ - вектор m признаков i-го объекта;
$x_j = (x_j^{(1)}, ..., x_j^{(m)})$ - вектор m признаков i-го объекта;

Метод k ближайших соседей (классификация)

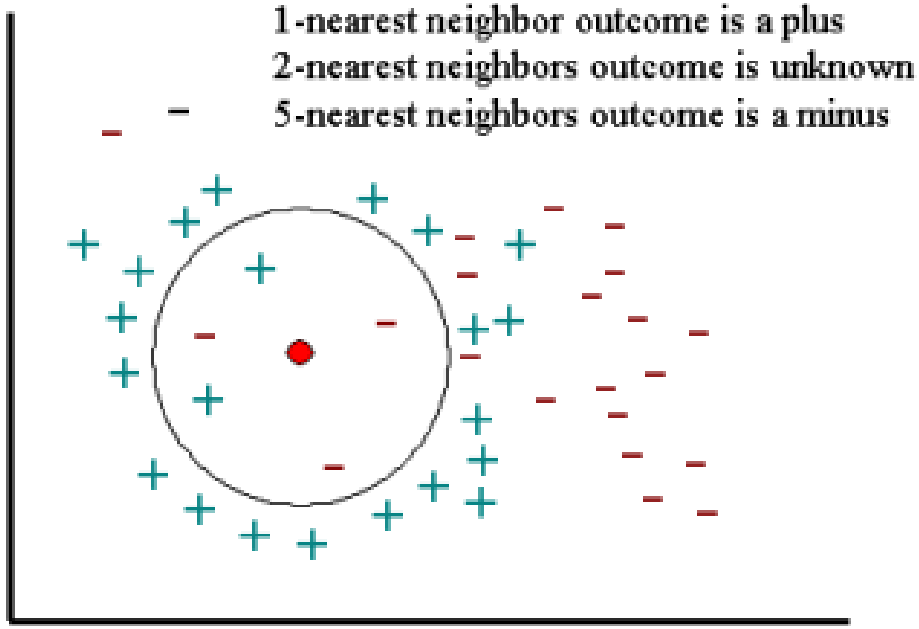
Метод k ближайших соседей (классификация)



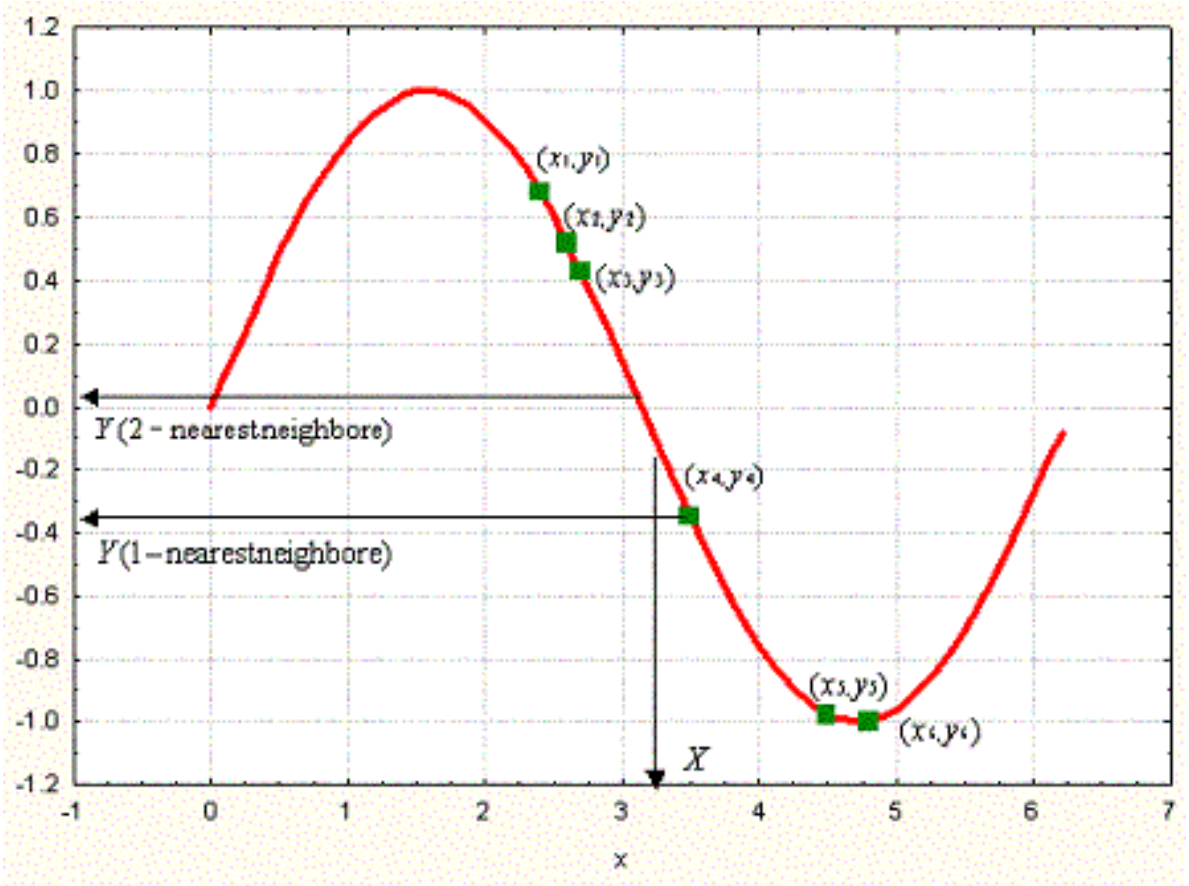
Метод k ближайших соседей: пример “правильной ” классификации



Метод k ближайших соседей: пример ошибочной классификаци



Метод k ближайших соседей (регрессия)



Метод k ближайших соседей

Достоинства:

- Простота реализации;
- Классификацию, проведенную алгоритмом, легко интерпретировать путем предъявления пользователю нескольких ближайших объектов.

Недостатки:

- Необходимость хранения обучающей выборки целиком.
- Поиск ближайшего соседа предполагает сравнение классифицируемого объекта со всеми объектами выборки.

Выбор k

- Малые значения k приведут к тому, что “шум” (выбросы) будет существенно влиять на результаты.
- Большие значения усложняют вычисления и искажают логику ближайших соседей, в соответствии с которой ближайшие точки могут принадлежать одному классу (гипотеза компактности).
- **Эвристика**: $k = \sqrt{n}$

Метод k ближайших соседей (пример)

Анализ брака древесины: по признакам средняя длина трещины и средний диаметр сучка

длина трещины	диаметр сучка	класс
7	7	брак
7	4	брак
3	4	не брак
1	4	не брак

Новый объект (длина трещины=3, диаметр сучка=7), $k = 3$

Метод k ближайших соседей (пример)

длина трещины	диаметр сучка	ρ
7	7	$(7-3)^2+(7-7)^2=16$
7	4	$(7-3)^2+(4-7)^2=25$
3	4	$(3-3)^2+(4-7)^2=9$
1	4	$(1-3)^2+(4-7)^2=13$

Метод k ближайших соседей (пример)

длина трещины	диаметр сучка	ρ	ранк	входит в 3 соседа?
7	7	16	3	да
7	4	25	4	нет
3	4	9	1	да
1	4	13	2	да

Метод k ближайших соседей (пример)

длина трещины	диаметр сучка	ρ	ранк	класс объекта
7	7	16	3	брак
7	4	25	4	-
3	4	9	1	не брак
1	4	13	2	не брак

*Объект (3,7) принадлежит классу "**не брак**"*

Вероятностная интерпретация метода ближайших соседей

- Метод ближайших соседей пытается аппроксимировать байесовское решающее правило на множестве обучающих данных;
- Для этого необходимо вычислить условную вероятность $P(x|y)$ äàíííûõ x данных x при условии их принадлежности классу y, априорную вероятность каждого класса $P(y)$ и маргинальную вероятность данных $P(x)$;
- Эти вероятности вычисляются для некоторой малой области вокруг нового примера, размер области будет зависеть от распределения вероятностей на тестовых примерах.

Вычисление вероятностей для kNN

- Пусть “шар” размерности m (m - число признаков) вокруг нового примера zсодержитk ближайших соседей для z
- Тогда

$$P(z) = \frac{k}{n},\ P(z|y=1) = \frac{k_1}{n_1},\ P(y=1) = \frac{n_1}{n}$$

- $P(z)$ - вероятность того, что случайная точка находится в “шаре”
- $P(z|y=1)$ - вероятность того, что случайная точка из класса 1 находится в“шаре”
- n_1, k_1 - число примеров из класса и 1 из класса 1 в k

Вычисление вероятностей для kNN

- $$P(z) = \frac{k}{n},\ P(z|y=1) = \frac{k_1}{n_1},\ P(y=1) = \frac{n_1}{n}$$
- Используем правило Байеса

$$P(y=1|z) = \frac{P(z|y=1)P(y=1)}{P(z)} =$$

$$= \frac{\frac{k_1}{n_1} \cdot \frac{n_1}{n}}{\frac{k}{n}} = \frac{k_1}{k}$$

Вычисление вероятностей для kNN

- Правило Байеса

$$P(y = 1|z) = \frac{k_1}{k}, \quad P(y = -1|z) = \frac{k_{-1}}{k}$$

Используя решающее правило Байеса, мы выбираем класс с наибольшей вероятностью, т.е. сравниваем $P(y = 1|z)$ *и* $P(y = -1|z)$. *А это тоже самое, что сравнение* k_1/k *и* k_{-1}/k.

Метод ближайшего соседа

Для произвольного $x^* \in X$ отсортируем объекты $x_1, \ldots, x_n$:

$$\rho(x^*, x_1) \leq \rho(x^*, x_2) \leq \ldots \leq \rho(x^*, x_n)$$

x_i - i-ый сосед объекта x^*; y_i - ответ на i-ом соседе объекта x^*.

Метрический алгоритм классификации:

$$a(x^*) = \arg\max_{y \in Y} \underbrace{\sum_{i=1}^{n} [y_i = y] \cdot w(i, x^*)}_{\Gamma_y(x^*)}$$

$w(i, x^*)$ - вес (степень важности) i-го соседа объекта x^*, неотрицателен, не возрастает по i.

$\Gamma_y(x^*)$ - оценка близости объекта x^* к классу y.

Как найти оптимальное значение k в различный ситуациях

Оптимизация числа соседей k: функционал скользящего контроля leave-one-out:

$$LOO(k, X) = \sum_{i=1}^{n} [a(x_i) \neq y_i] \rightarrow \min_k$$

Метод опорных векторов. Немного истории

- Первые идеи метода были предложены еще в 1950-е годы.
- Метод был создан на основе статистической теории обучения.
- Метод стал известен и популярен после замечательной статьи (Вапник и др.) в 1992 г.
- В настоящее время метод успешно используется во многих областях.
- Метод также был модифицирован для задач регрессии.

Что мы хотим?

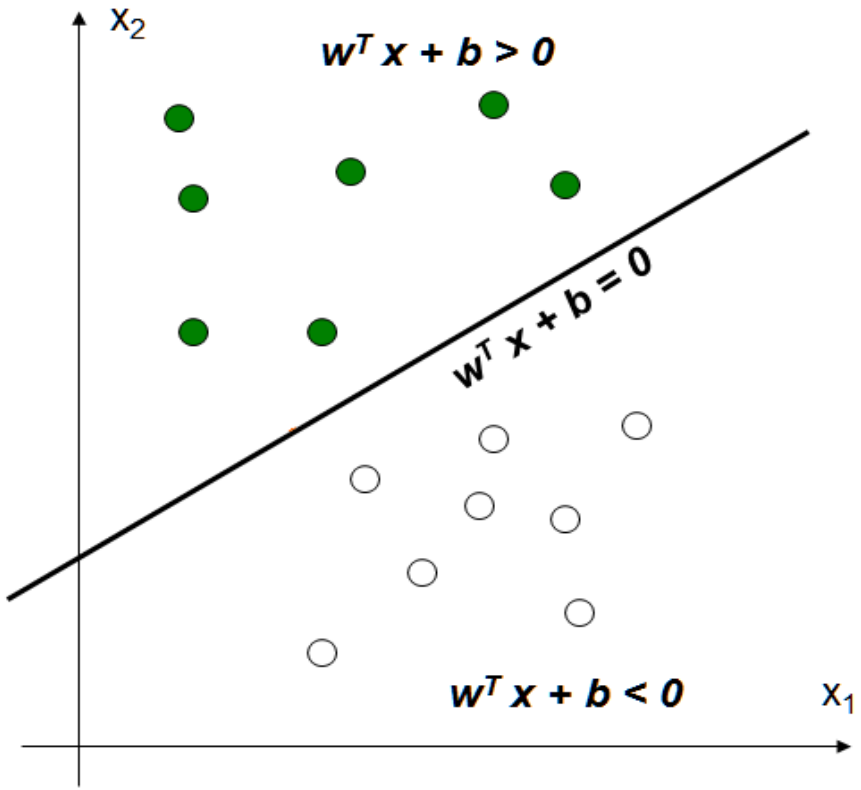
- Метод опорных векторов решает задачу классификации.
- Каждый элемент данных - точка m– мерном пространстве $\mathbb{R}_m$.
- Формально: есть точки x_i, $i = 1, ..., m$, у точек есть метки $y_i \in \{-1, +1\}$.

Можно ли разделить данные гиперплоскостью и какая она?

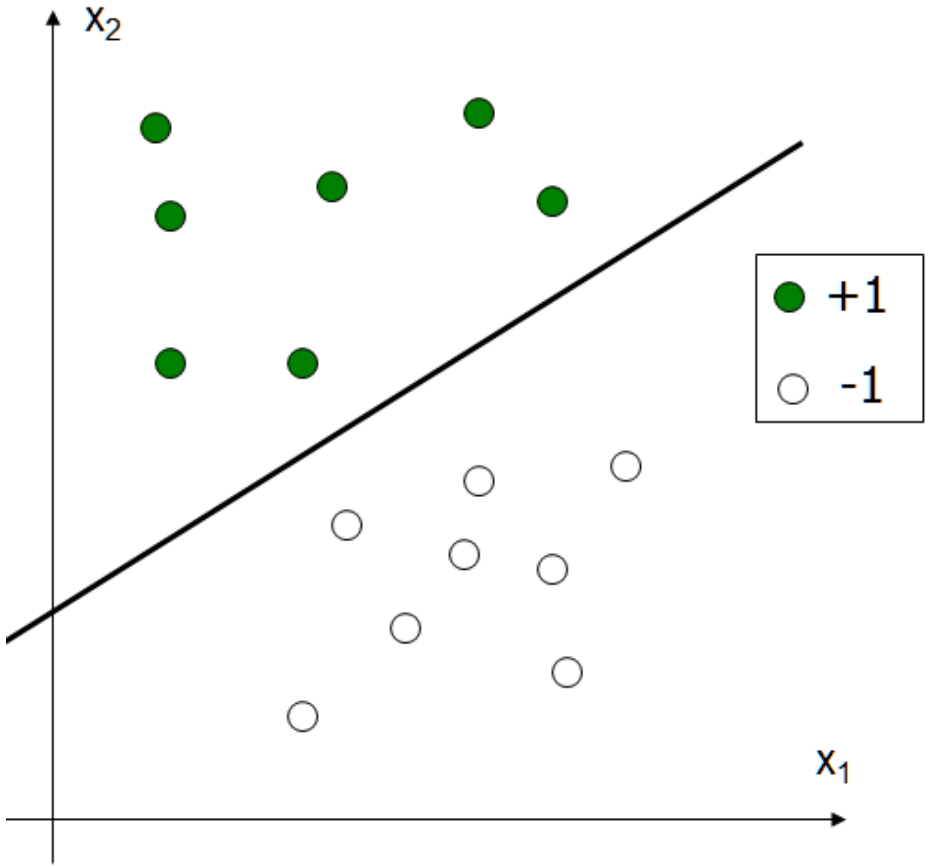
Классификация данных

$g(\mathrm{x}) = \mathrm{w}^T\mathrm{x} + b$ - линейная разделяющая функция (гиперплоскость))

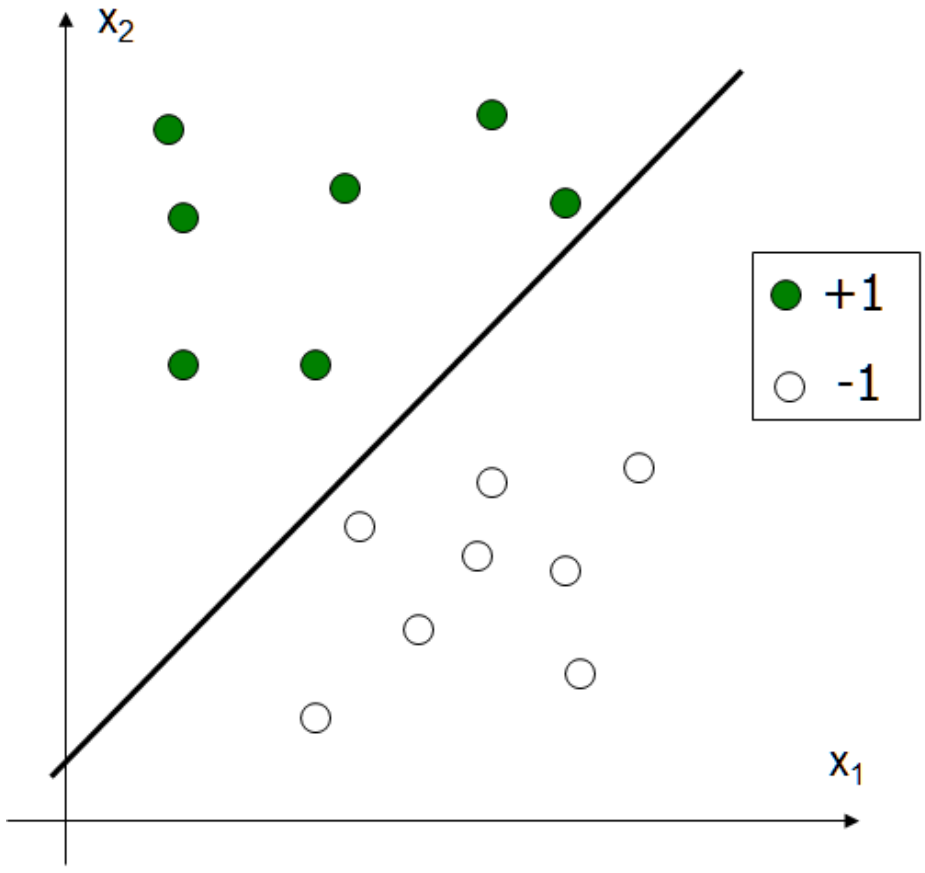
$g(x_1, ..., x_m) = \sum_{i=1}^{m} w_i x_i + b$



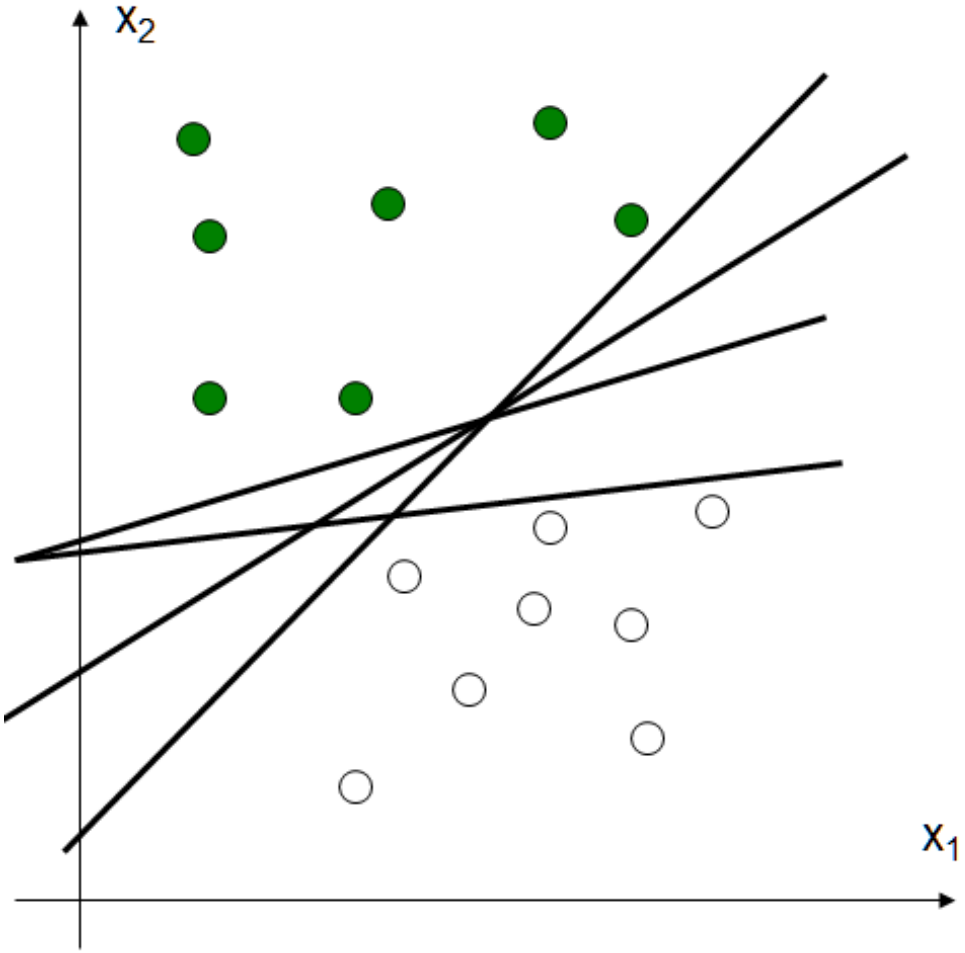
Классификация данных (один из вариантов)



Классификация данных (другой вариант)



Классификация данных (еще много вариантов)



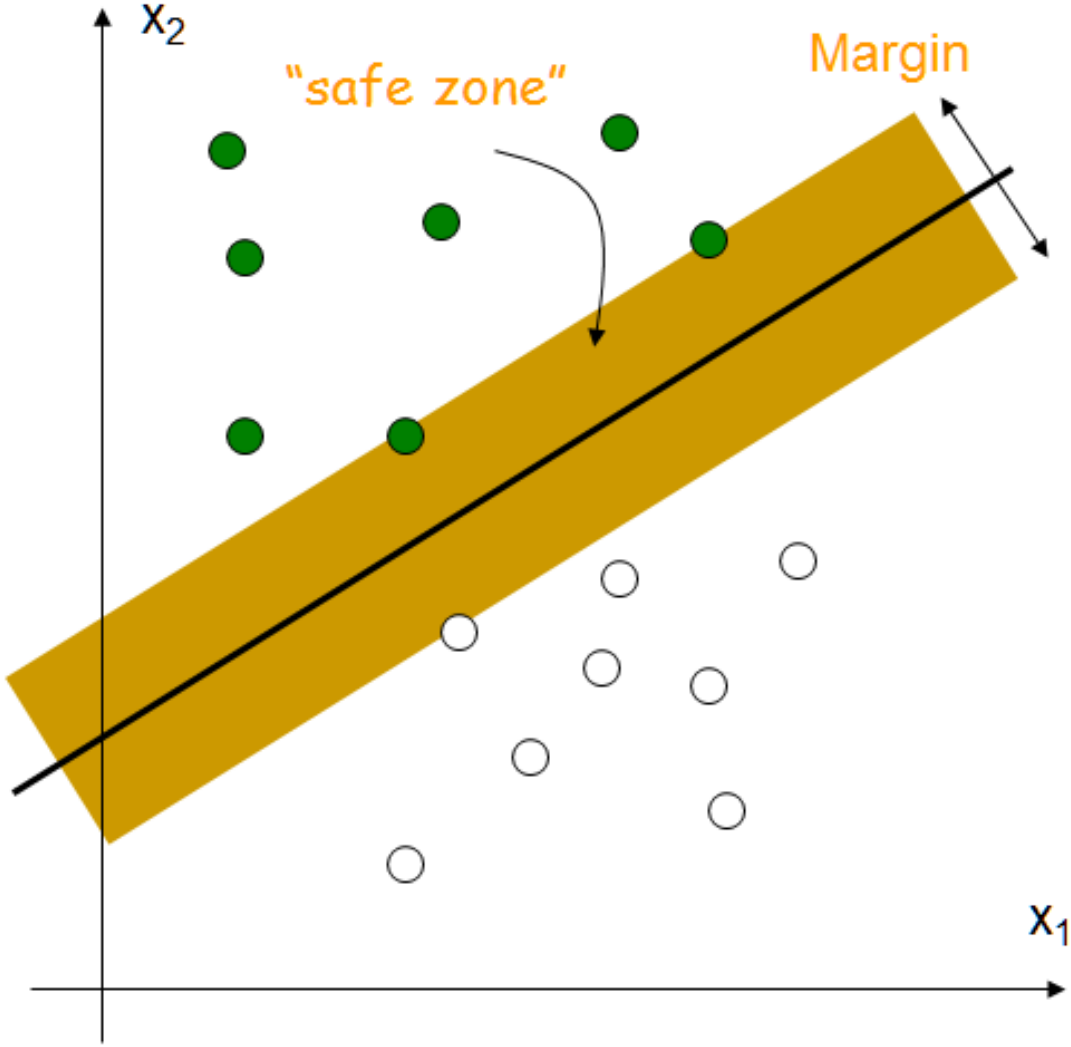
Какой вариант оптимальный?

Оптимальная разделяющая гиперплоскость — это гиперплоскость, максимизирующая ширину разделяющей полосы и лежащая в середине этой полосы.

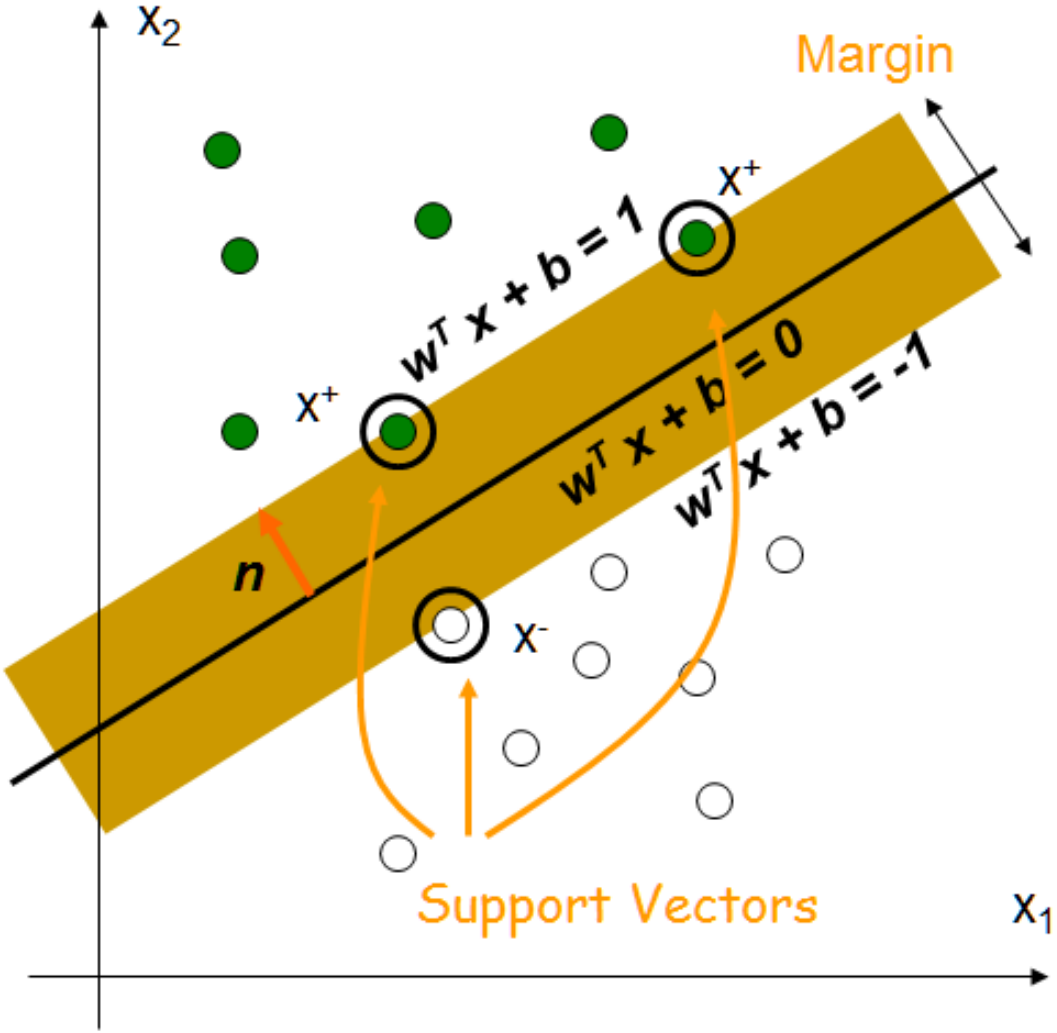
Иными словами, оптимальная разделяющая гиперплоскость максимизирует **зазор** (margin) между плоскостью и данными из обучающей выборки.

Если классы линейно разделимы и каждый содержит не менее одного элемента, то оптимальная разделяющая гиперплоскость единственна.

Разделяющая полоса



Какая полоса лучше?



Как определить ширину полосы?

Без доказательства:

$$M = \frac{2}{\|\mathrm{w}\|} = \frac{2}{w_1^2 + w_2^2 + \ldots + w_m^2}$$

Формулировка задачи оптимизации:

$$\frac{2}{\|\mathrm{w}\|} \to \max_{\mathrm{w}}$$

при условии:

$$\begin{aligned} \text{для } y_i = +1: &\quad \mathrm{w}^T\mathrm{x} + b \geq 1 \\ \text{для } y_i = -1: &\quad \mathrm{w}^T\mathrm{x} + b \leq -1 \end{aligned}$$

Задача оптимизации

$$\frac{\|\mathrm{w}\|}{2} = \frac{1}{2}(w_1^2 + w_2^2 + ... + w_m^2) \to \min_{\mathrm{w}}$$

при условии:

$$y_i \left(\mathrm{w}^T \mathrm{x} + b\right) \geq 1$$

Это задача квадратической оптимизации с линейными ограничениями!

Задача оптимизации

Функция Лагранжа:

$$\mathcal{L}(\mathrm{x}, \mathrm{w}, b, \alpha) = \frac{1}{2}\sum_{i=1}^{m} w_i^2 - \sum_{i=1}^{n} \alpha_i \left(y_i \left(\mathrm{w}^T \mathrm{x} + b\right) - 1\right)$$

$\alpha_i,\ i = 1, ..., n$ - множители Лагранжа.
Необходимые условия седловой точки функции Лагранжа

$$\frac{\partial \mathcal{L}}{\partial b} = -\sum_{i=1}^{n} \alpha_i y_i = 0, \quad \frac{\partial \mathcal{L}}{\partial w_k} = w_k - \sum_{i=1}^{n} \alpha_i y_i x_i^{(k)} = 0,\ k = 1, ..., m$$

Двойственная задача

Подставляя условия седловой точки в функцию Лагранжа, получаем двойственную задачу

$$\max\left(\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathrm{x}_i^T \mathrm{x}_j\right)$$

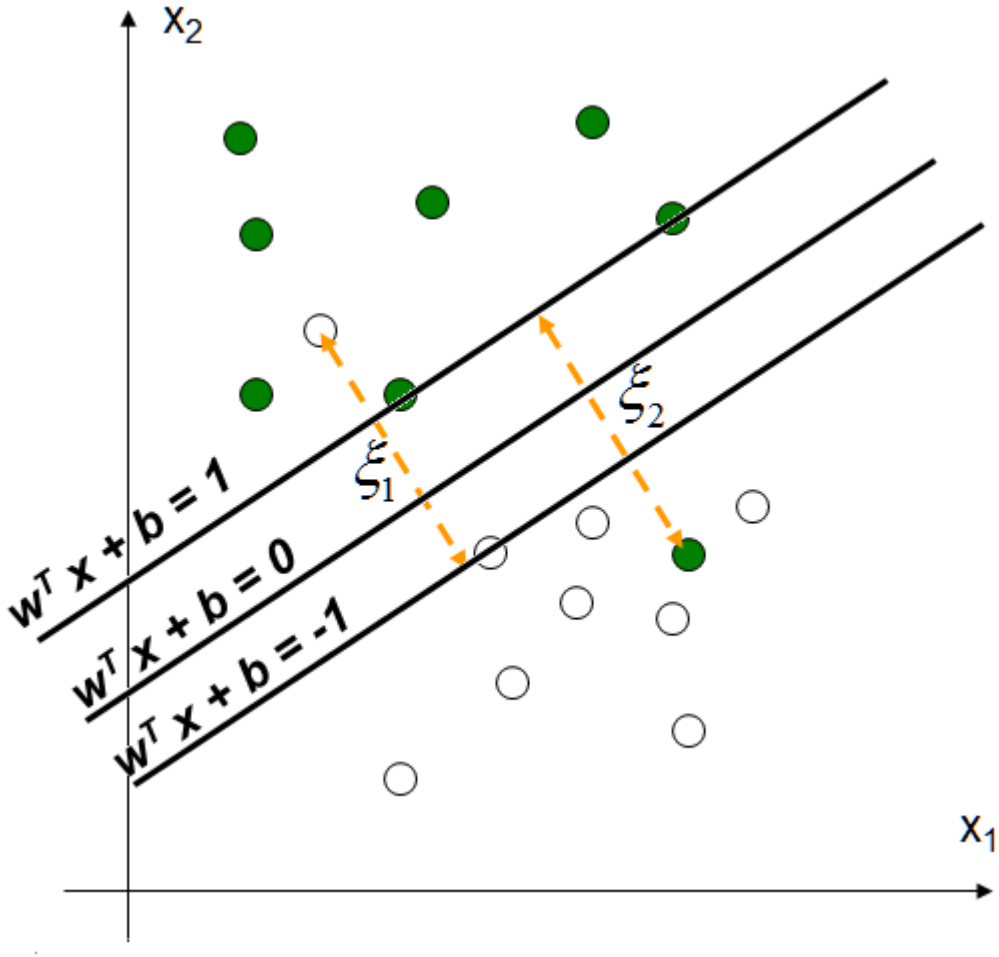
при ограничениях

$$\alpha_i \geq 0, i = 1, ..., n, \sum_{i=1}^{n}\alpha_i y_i = 0.$$

Разделяющая функция:

$$g(\mathrm{x}) = \mathrm{w}^T\mathrm{x} + b = \sum_{i=1}^{n}\alpha_i \mathrm{x}_i^T\mathrm{x} + b$$

Данные линейно не разделимы



Вспомогательные переменные ξ_i (неотрицательные ошибки) могут быть добавлены.

Ключевые особенности SVM

1. Максимизирует отступ между положительными и отрицательными объектами
2. Штрафует ошибки в случае неразделимой выборки
3. Только опорные векторы определяют решение
4. Отображение объектов при помощи ядер в новое нелинейное пространство

Преимущества и недостатки SVM

- Преимущества SVM:
  - Задача выпуклого квадратичного программирования имеет единственное решение.
  - Позволяет рассматривать различные виды нелинейности, изменяя ядра или их параметры.
- Недостатки SVM:
  - Неустойчивость к шуму.
  - Нет общих подходов к оптимизации ядра под задачу.
  - Приходится подбирать параметр C.
  - Нет отбора признаков.

Метод градиентного спуска

- Идея построения перцептрона - минимизация ошибки.
- Перцептронная функция $y^*(x_1, ..., x_d) = \sum_{i=0}^{d} x_i w_i$ должна быть приближена к функции, заданной примерами обучающей выборки: $y = g(x_1, ..., x_d)$.
- Мера ошибки - среднеквадратичное отклонение от целевых значений:

$$E(w_0, ..., w_d) = \frac{1}{2} \sum_{k=1}^{n} (y_k - y^*(x_1(k), ..., x_d(k)))^2$$

- Цель - минимизировать $E(w_0, ..., w_d)$ по $w_0, ..., w_d$.

Метод градиентного спуска

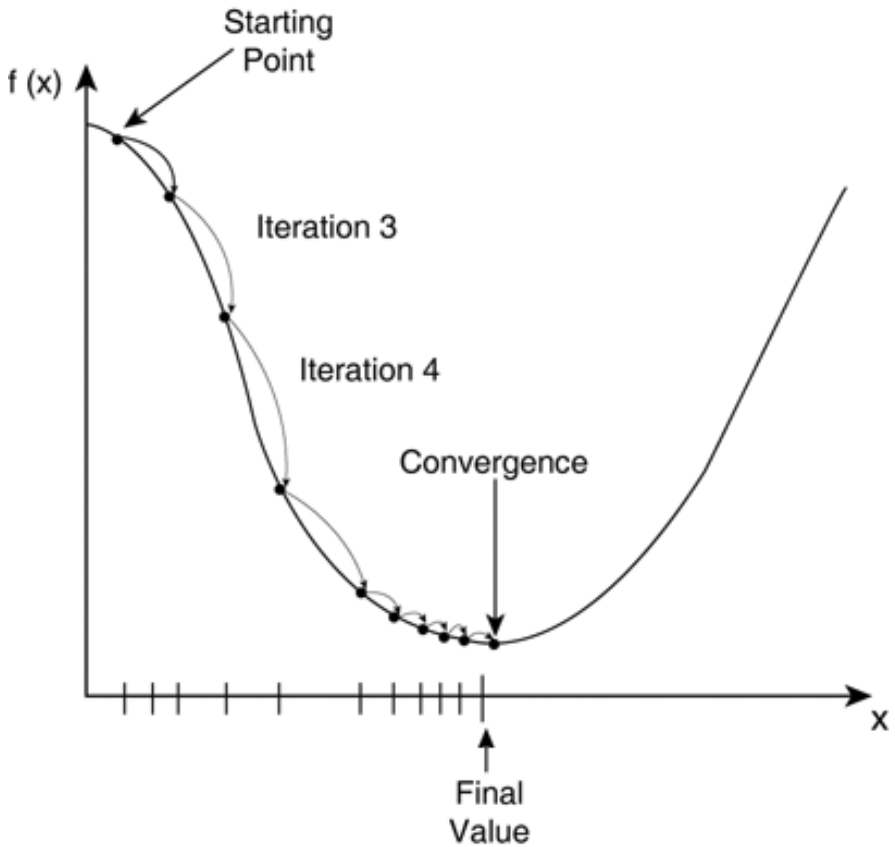
- $E(w_0, ..., w_d)$ - параболическую поверхность с единственным минимумом.
- Двигаемся в направлении, противоположном градиенту

$$-\nabla E(w_0, ..., w_d) = -\left[\frac{\partial E}{\partial w_0}, ..., \frac{\partial E}{\partial w_d}\right]$$

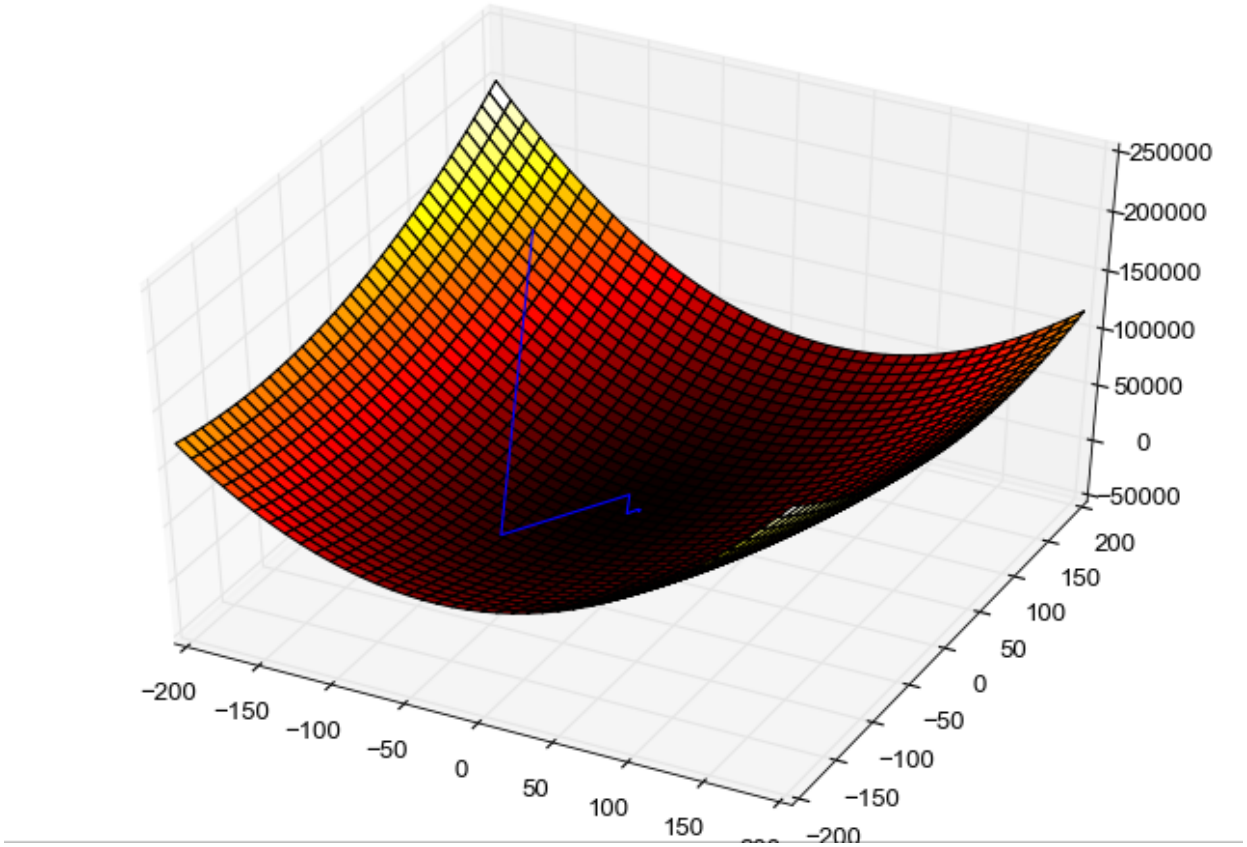
- Коррекция весов:

$$w_i(k+1) \leftarrow w_i(k) - \eta \frac{\partial E}{\partial w_i}$$

Движение к минимуму



Движение к минимуму



Метод градиентного спуска (далее)

- Вычислим $\partial E / \partial w_i$:
- Двигаемся в направлении, противоположном градиенту

$$\frac{\partial E}{\partial w_i} = \frac{1}{2} \sum_{k=1}^{n} \frac{\partial}{\partial w_i} \left(y_k - \sum_{i=0}^{d} x_i(k) w_i \right)^2$$
$$= \sum_{k=1}^{n} \left(y_k - \sum_{i=0}^{d} x_i(k) w_i \right) \left(-x_i(k) \right).$$

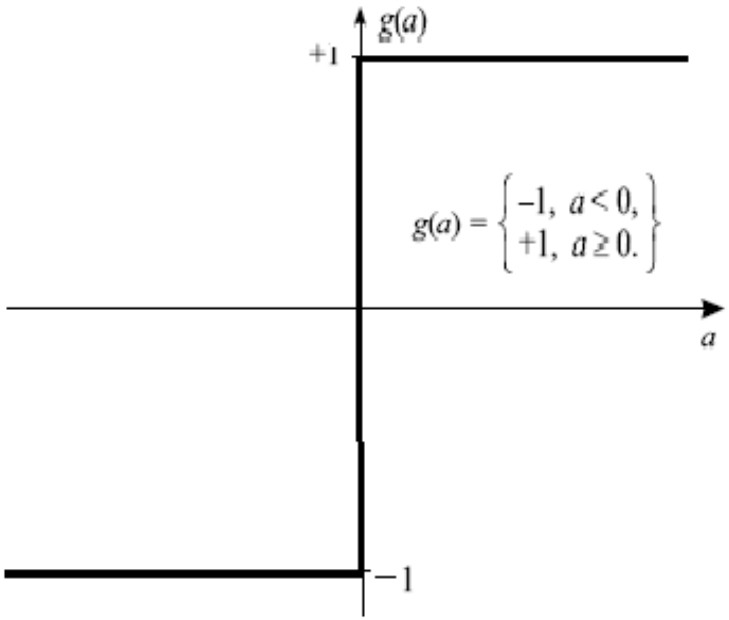
- Коррекция весов:

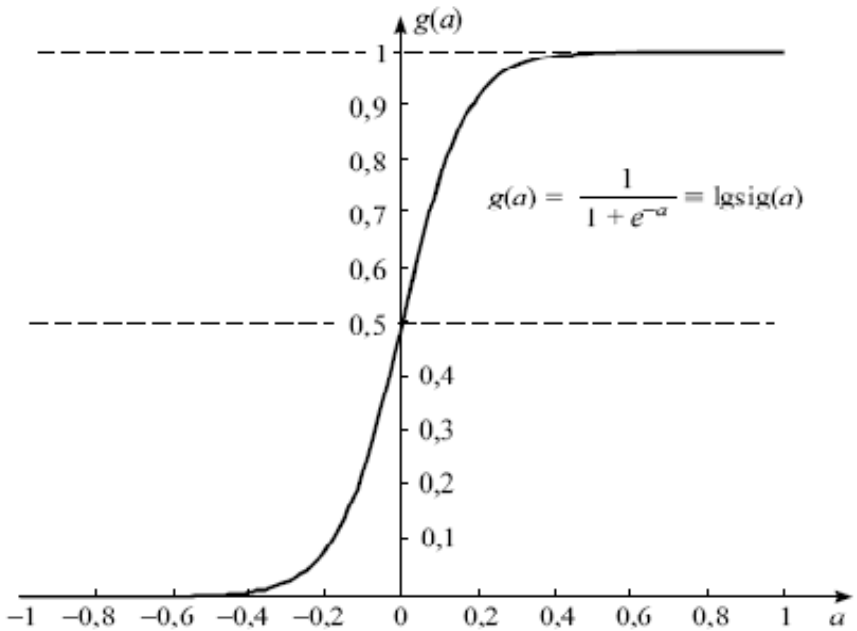
$$w_i(k+1) \leftarrow w_i(k) + \eta \sum_{k=1}^{n} \left(y_k - \sum_{i=0}^{d} x_i(k) w_i \right) x_i(k)$$

Пороговая функция или функция активации

Пороговая функция - **сигмоид**:

$$y(x) = \sigma(x) = \frac{1}{1 + e^{-x}}$$





Пороговая функция или функция активации

- Еще одна пороговая функция - **бисигмоид или гиперб. тангенс**:

$$y(x) = \sigma(x) = \frac{2}{1 + e^{-x}} - 1$$
$$= \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

- Изменяется от -1 до 1.