

# Прикладная статистика в машинном обучении

## Лекция 10. Часть 2

### Метод опорных векторов (SVM)

И. К. Козлов  
(Мехмат МГУ)

2022

## SVM и RVM

Сегодня мы изучим 2 схожие модели:

Support Vector Machine (SVM) и Relevance Vector Machine (RVM).

Подробнее о них — см. Главу 7



Bishop C.M.

*Pattern Recognition and Machine Learning.*

## Back to the 90s

В 2ой части лекции мы поговорим об одной  
олдскульной ML-модели из очень страшных времён - 90ых.

Сейчас

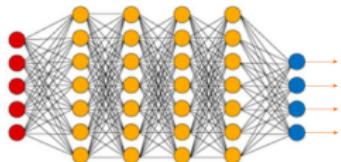
iPhone



BIG DATA



Neural Networks



90ые

Phone



Data



Support Vector Machine

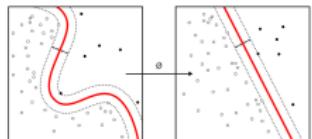
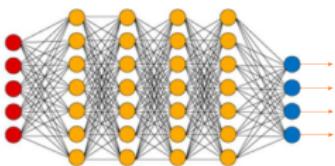


Рис.: До глубоких нейронок был SVM

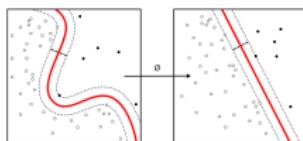
# SVM

Q: Как может выглядеть “промежуточное звено” между нейронками и KNN?

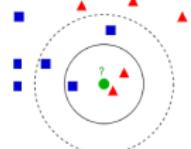
**Neural Networks**



**Support Vector Machine**



**KNN**



**TRAIN**

Обучаем веса  $w_i$  по  $x_j$

**TEST**

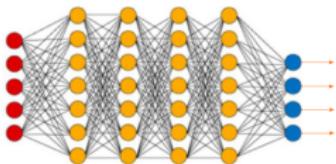
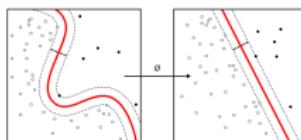
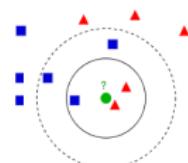
Вычисляем  $y = f(x; w)$   
Не нужен Train



Помним весь Train

Метка  $x$  определяется  
метками ближайших

$x_{(1)}, x_{(2)}, \dots, x_{(n)}$

**Neural Networks****Support Vector Machine****KNN**

TRAIN

Обучаем веса  $w_i$  по  $x_j$ Отбираем опорные векторы  $x_{i_1}, \dots, x_{i_M}$ 

TEST

Вычисляем  $y = f(x; w)$   
Не нужен Train

Метка определяется

$$y = \sum_{j=1}^M w_j K(x, x_j) + b$$

Ядро  $K(x, x_j)$  - насколько  $x$  похож на  $x_j$ 

Помним весь Train

Метка  $x$  определяется метками ближайших  $x_{(1)}, x_{(2)}, \dots, x_{(n)}$

# Линейный SVM

1 Линейный SVM

2 Kernel Trick

3 SVM vs RVM

## Разделяющая гиперплоскость

SVM — не вероятностная модель. Рассмотрим её в простейшем случае.

**Рассмотрим задачу классификации.** Даны точки

$$(x_1, y_1), \dots, (x_n, y_n).$$

Метки классов  $y_i = 1$  или  $-1$ .

## Линейный SVM

### Линейный SVM

*Попробуем разделить классы гиперплоскостью.*

**Q:** Какой вид имеет уравнение гиперплоскости?

## Линейный SVM

### Линейный SVM

*Попробуем разделить классы гиперплоскостью.*

**Q:** Какой вид имеет уравнение гиперплоскости? **A:**

$$y = w^T x + b.$$

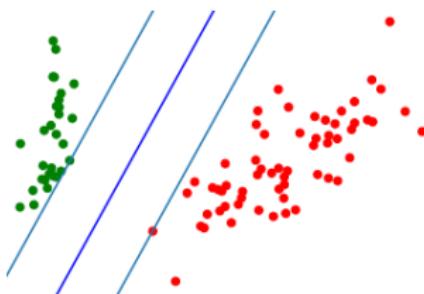
## Hard-Margin SVM

### Hard-Margin SVM

Предположим, что классы линейно разделимы. Отнормируем  $w$  и  $b$  так, чтобы

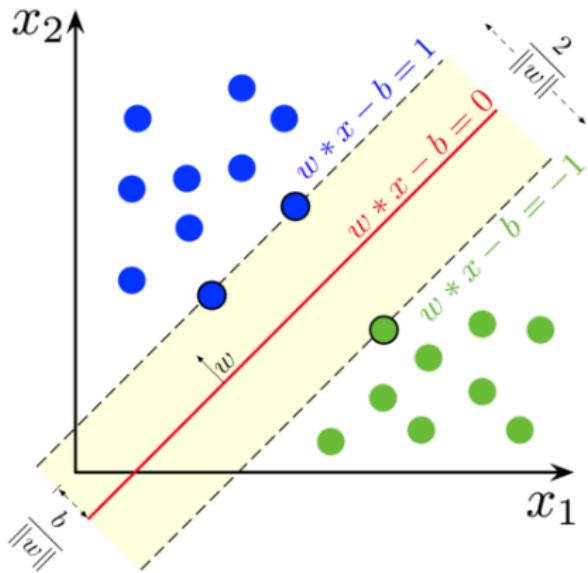
- Если  $y_i = 1$ , то  $w^T x_i - b \geq 1$ ;
- Если  $y_i = -1$ , то  $w^T x_i - b \leq -1$ ,

и в обоих случаях равенство достигается.



## Hard-Margin SVM

Найдём оптимальную гиперплоскость — с наименьшим зазором (margin = “ширина полосы”) между классами. Несложно посчитать, что зазор равен  $\frac{2}{\|w\|}$ .



## Классная задача оптимизации

В итоге мы получаем задачу оптимизации вида

$$\begin{aligned} \min \quad & \frac{1}{2} \|w\|^2 \\ \text{subject to} \quad & y_i(w^T x_i + b) \geq 1. \end{aligned}$$

Это задача **квадратичной оптимизации** с линейными ограничениями. Такие задачи отлично решаются.

Q: А как найти  $b$ ?

## Классная задача оптимизации

В итоге мы получаем задачу оптимизации вида

$$\begin{aligned} \min \quad & \frac{1}{2} \|w\|^2 \\ \text{subject to} \quad & y_i(w^T x_i + b) \geq 1. \end{aligned}$$

Это задача **квадратичной оптимизации** с линейными ограничениями. Такие задачи отлично решаются.

**Q:** А как найти  $b$ ?

**A:** Найти опорную точку, где  $y_i(w^T x_i + b) = 1$  и выразить  $b$ .

## Предсказание модели

Q: К какому классу относится новая точка  $x$ ?

## Предсказание модели

**Q:** К какому классу относится новая точка  $x$ ?

**A:** Определяется знаком

$$x \mapsto \text{sgn}(w^T x - b).$$

# Kernel Trick

1 Линейный SVM

2 Kernel Trick

3 SVM vs RVM

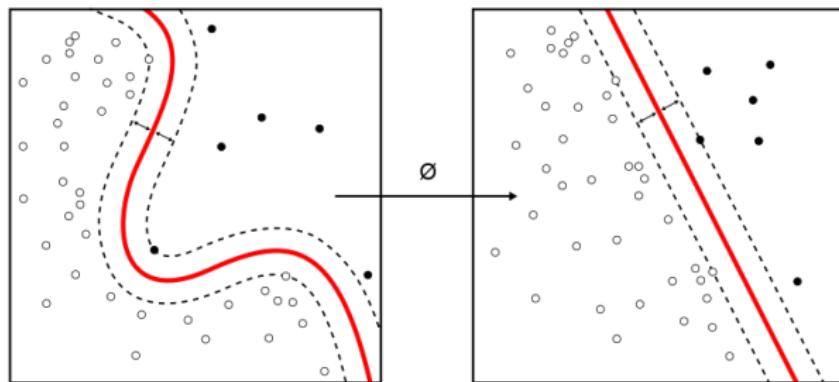
## Kernel Trick

Классы редко когда разделимы.

Идея. А пусть существует вложение

$$x \rightarrow \varphi(x),$$

в результате которого данные разделяются.



## Условия Каруша — Куна — Таккера

Задача оптимизации принимает вид

$$\begin{aligned} \min \quad & \frac{1}{2} \|w\|^2 \\ \text{subject to} \quad & y_i(w^T \varphi(x_i) + b) \geq 1. \end{aligned}$$

**Проблема.** Отображение  $\varphi(x)$  мы не знаем.

## Двойственность

Следующие несколько слайдов могут показаться “шаманством”.

На самом деле стандартный приём в оптимизации: перейти от исходной задачи оптимизации к [двойственной задачи](#).

## Условия Каруша — Куна — Таккера

Оптимум задачи

$$\begin{array}{ll}\min & f(x) \\ \text{subject to} & g_i(x) \geq 0.\end{array}$$

находится при помощи [условий Каруша — Куна — Таккера](#).

## Условия Каруша — Куна — Таккера

Оптимум задачи

$$\begin{array}{ll} \min & f(x) \\ \text{subject to} & g_i(x) \geq 0. \end{array}$$

находится при помощи [условий Каруша — Куна — Таккера](#).

- Вводится **функция Лагранжа**

$$L = f(x) - \sum_i \alpha_i g_i(x).$$

## Условия Каруша — Куна — Таккера

Оптимум задачи

$$\begin{array}{ll} \min & f(x) \\ \text{subject to} & g_i(x) \geq 0. \end{array}$$

находится при помощи [условий Каруша — Куна — Таккера](#).

- Вводится **функция Лагранжа**

$$L = f(x) - \sum_i \alpha_i g_i(x).$$

- У него находятся экстремумы

$$\frac{\partial L}{\partial x} = 0, \quad \frac{\partial L}{\partial \alpha_i} = 0$$

- при условиях

$$g_i(x) \geq 0, \quad \alpha_i \geq 0, \quad \alpha_i g_i(x) = 0.$$

## Функция Лагранжа

Для SVM получаем функция Лагранжа

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_i \alpha_i [y_i(w^T \varphi(x_i) + b) - 1].$$

## Функция Лагранжа

Для SVM получаем функция Лагранжа

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_i \alpha_i [y_i(w^T \varphi(x_i) + b) - 1].$$

Условия на экстремум

$$\frac{\partial L}{\partial w} = w - \sum_i \alpha_i \varphi(x_i) \quad \Rightarrow \quad w = \sum_i \alpha_i y_i \varphi(x_i).$$

$$\frac{\partial L}{\partial b} = \sum \alpha_i y_i = 0.$$

## Функция Лагранжа

Для SVM получаем функция Лагранжа

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_i \alpha_i [y_i(w^T \varphi(x_i) + b) - 1].$$

Условия на экстремум

$$\frac{\partial L}{\partial w} = w - \sum_i \alpha_i \varphi(x_i) \Rightarrow w = \sum_i \alpha_i y_i \varphi(x_i).$$

$$\frac{\partial L}{\partial b} = \sum \alpha_i y_i = 0.$$

Подставляя  $\frac{\partial L}{\partial w}$  и  $\frac{\partial L}{\partial b}$  в  $L$ , получаем

$$L = \sum_i \alpha_i - \frac{1}{2} \sum_i \alpha_i \alpha_j y_i y_j \varphi(x_i)^T \varphi(x_j).$$

## Двойственная задача

Введём ядро

$$K(x, x') = \varphi(x)^T \varphi(x').$$

## Двойственная задача

Введём ядро

$$K(x, x') = \varphi(x)^T \varphi(x').$$

## Двойственная задача

$$\begin{aligned} \max \quad & \tilde{L}(\alpha) = \sum_i \alpha_i - \frac{1}{2} \sum_i \alpha_i \alpha_j y_i y_j K(x_i, x_j) \\ \text{subject to} \quad & \alpha_i \geq 0 \\ & \sum \alpha_i y_i = 0. \end{aligned}$$

## Двойственная задача

Введём ядро

$$K(x, x') = \varphi(x)^T \varphi(x').$$

## Двойственная задача

$$\begin{aligned} \max \quad & \tilde{L}(\alpha) = \sum_i \alpha_i - \frac{1}{2} \sum_i \alpha_i \alpha_j y_i y_j K(x_i, x_j) \\ \text{subject to} \quad & \alpha_i \geq 0 \\ & \sum \alpha_i y_i = 0. \end{aligned}$$

Опять же, это квадратичная задача оптимизации с развитыми методами решениями. Например, SMO — последовательная оптимизация по паре  $\alpha_i, \alpha_j$ :

[https://en.wikipedia.org/wiki/Sequential\\_minimal\\_optimization](https://en.wikipedia.org/wiki/Sequential_minimal_optimization).

## Kernel Trick

- Находим  $\alpha_i$  из **двойственной задачи**.
- Тип новой точки  $\mathbf{x}$  определяется

$$\mathbf{x} \mapsto \operatorname{sgn}(\mathbf{w}^T \varphi(\mathbf{x}) - b) = \operatorname{sgn}\left(\left[\sum_{i=1}^n \alpha_i y_i \mathbf{K}(\mathbf{x}_i, \mathbf{x})\right] - b\right).$$

## Kernel Trick

В формулах больше нет вложения  $\varphi(x)$ . Только ядро  $K(x, x')$ .



## Kernel Trick

*Иногда ядро задать проще, чем вложение.*

Например, популярное RBF ядро

$$K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2)$$

## Kernel Trick

*Иногда ядро задать проще, чем вложение.*

Например, популярное RBF ядро

$$K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2)$$

задаёт вложение в бесконечномерное (!) пространство, поскольку

$$\exp\left(-\frac{1}{2}\|\mathbf{x} - \mathbf{x}'\|^2\right) = \sum_{j=0}^{\infty} \sum_{\sum n_i=j} \exp\left(-\frac{1}{2}\|\mathbf{x}\|^2\right) \frac{x_1^{n_1} \cdots x_k^{n_k}}{\sqrt{n_1! \cdots n_k!}} \exp\left(-\frac{1}{2}\|\mathbf{x}'\|^2\right) \frac{x'_1^{n_1} \cdots x'_k^{n_k}}{\sqrt{n_1! \cdots n_k!}}.$$

## Kernel Trick

Остается заметить:

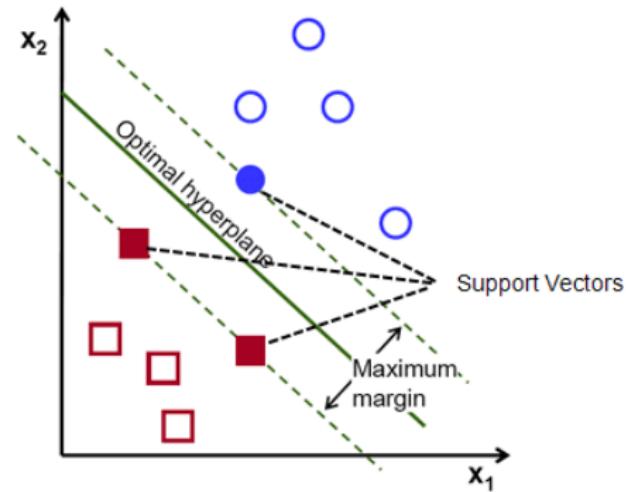
$$\alpha_i [y_i(\mathbf{w}^T \varphi(x_i) + b) - 1] \geq 0.$$

## Kernel Trick

Остается заметить:

$$\alpha_i [y_i(\mathbf{w}^T \varphi(x_i) + b) - 1] \geq 0.$$

- Если  $\alpha_i \neq 0$ , то  $x_i$  — **опорный вектор**, лежит на границе.
- Если  $\alpha_i = 0$ , то  $x_i$  — внутри полуплоскости, не влияет на классификацию.



# SVM vs RVM

1 Линейный SVM

2 Kernel Trick

3 SVM vs RVM

# SVM

## Преимущества SVM:

- У задачи квадратичной оптимизации единственное решение.  
(В отличие от нейронок!)
- Есть отбор опорных векторов.

## Недостатки SVM:

- Трудно подбирать ядро  $K(x, x')$ .
- Нет предсказания вероятностей.
- Не предназначен для мультиклассовой классификации.

# SVM

## Преимущества RVM:

- Обычно меньше опорных векторов, чем у SVM.
- Опорные векторы не на границе классов (более устойчивое решение).
- Все параметры подбираются автоматически.
- Можно добавить ядра как в SVM, заменив в Likelihood

$$\mu = w^T x \rightarrow \mu_i = \sum_j w_j K(x_i, x_j) + b$$

## Недостатки RVM:

- Обучается медленнее SVM.
- Не всегда эффективней SVM.