



# Машинное обучение в науках о Земле

Михаил Криницкий

к.т.н., н.с.

Институт океанологии РАН им. П.П. Ширшова

Лаборатория взаимодействия океана и атмосферы и  
мониторинга климатических изменений (ЛВОАМКИ)



# Метод опорных векторов Support vector machines (SVM)

Михаил Криницкий

к.т.н., н.с.

Институт океанологии РАН им. П.П. Ширшова

Лаборатория взаимодействия океана и атмосферы и  
мониторинга климатических изменений (ЛВОАМКИ)

# ОБУЧЕНИЕ С УЧИТЕЛЕМ: задача классификации

Конечная задача: присвоить метку класса у новому объекту  $x_i$

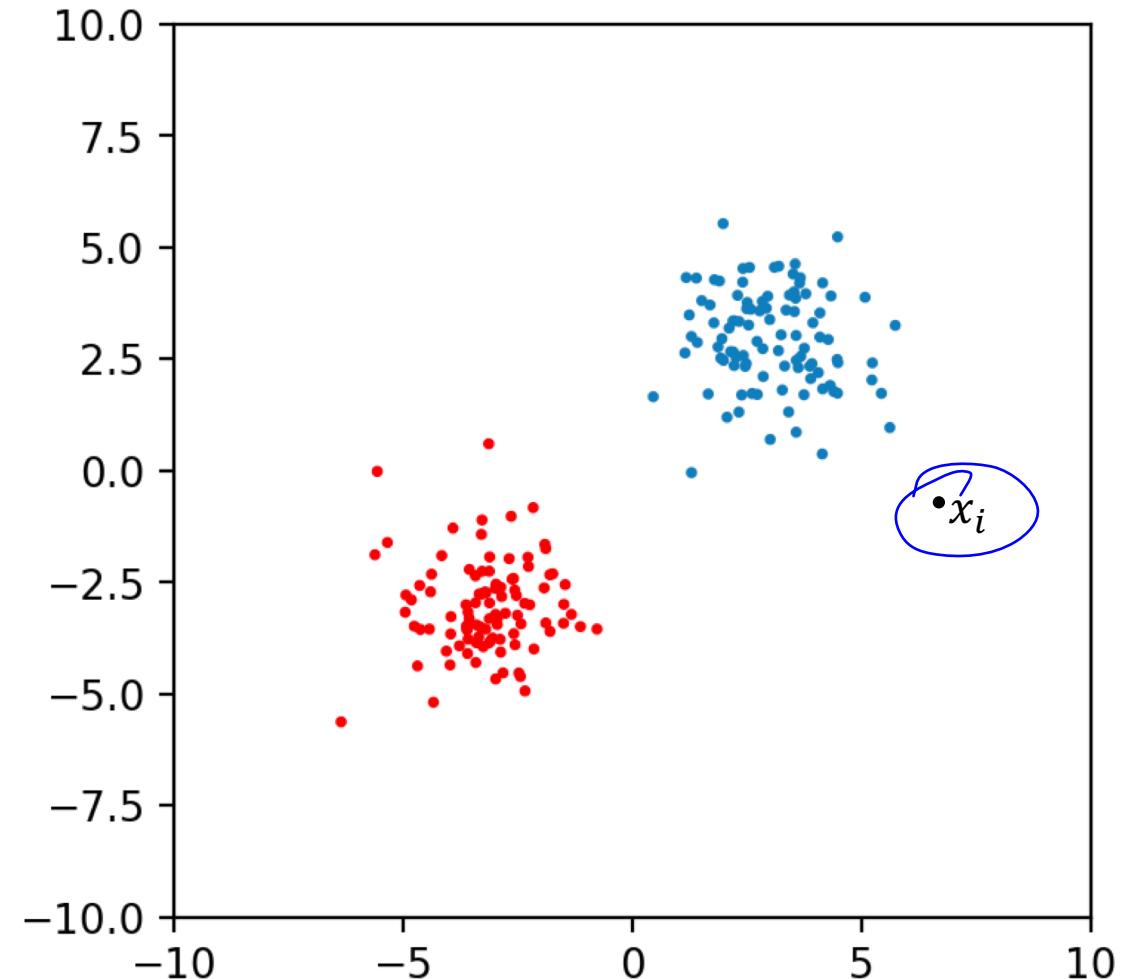
Статистический подход:

Промежуточная цель:  $p(y = 1|x_i)$

Методы моделирования:

- Байесовский классификатор (Bayes classifier)
- Наивный байесовский классификатор (Naïve bayes)
- Линейный дискриминантный анализ (LDA)
- Квадратичный дискриминантный анализ (QDA)
- Логистическая регрессия (Logistic Regression)
- Обобщенные линейные модели
- Обобщенные аддитивные модели
- Многослойный перцептрон (MLP, ANN, FCNN)

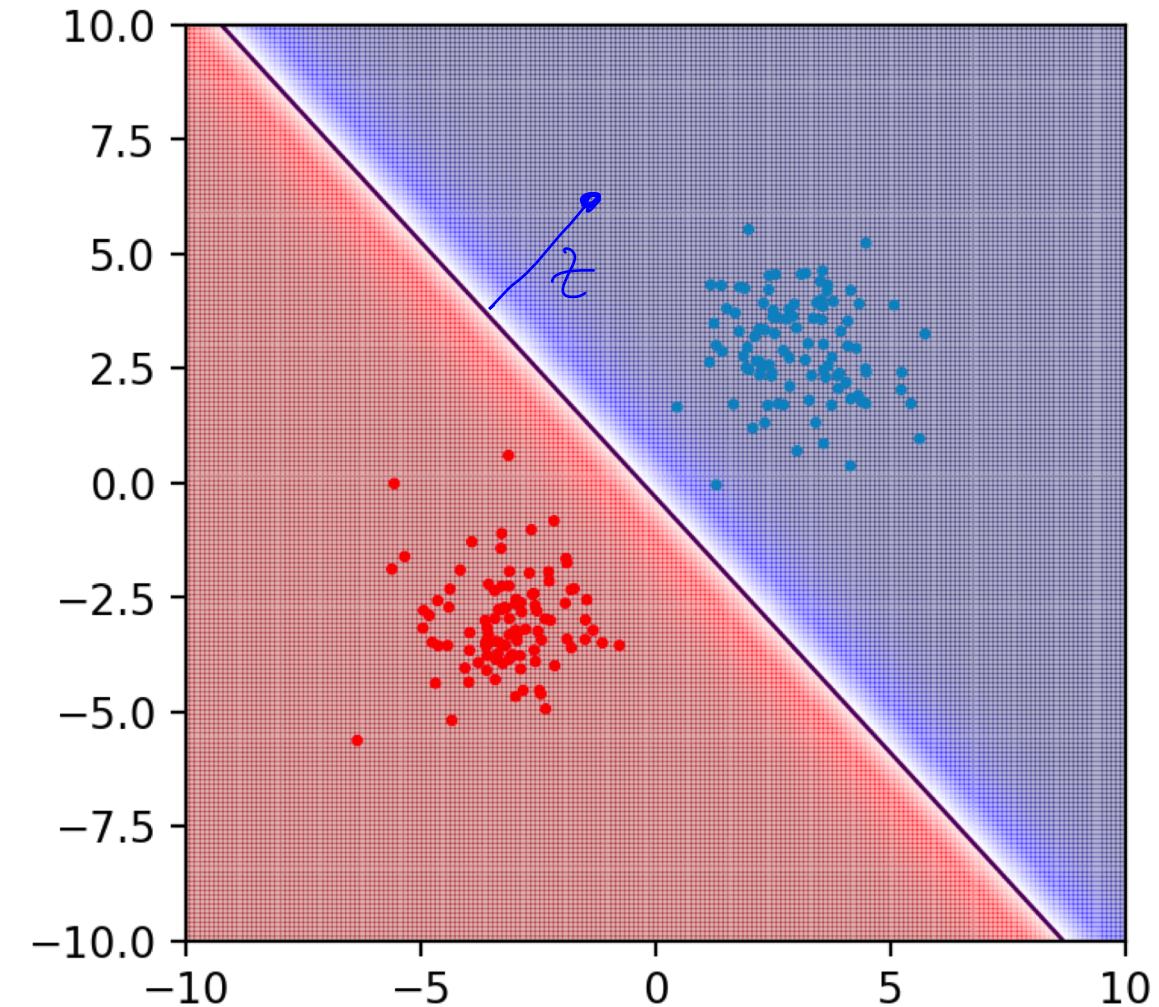
ML



# Разделяющая поверхность (на примере результатов логистической регрессии)

В модели логистической регрессии:

- разделяющая гиперповерхность линейна
- $z = \theta_0 + \theta_1 x$  – может интерпретироваться с точностью до масштаба как расстояние (со знаком) от объекта до разделяющей поверхности



# Линейность разделяющей поверхности для модели логистической регрессии

$$P = \frac{1}{1 + e^{-\theta \cdot x}}$$

$$P = 0.5$$

$$\frac{1}{1 + e^{-\theta \cdot x}} = c$$

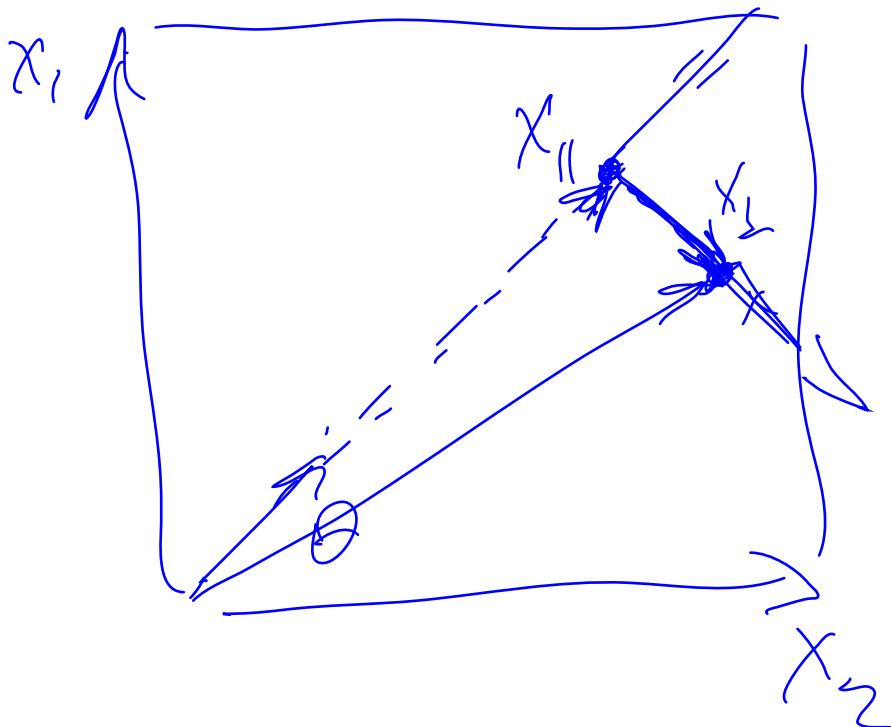
$$1 + e^{-\theta \cdot x} = c$$

$$e^{-\theta \cdot x} = c$$

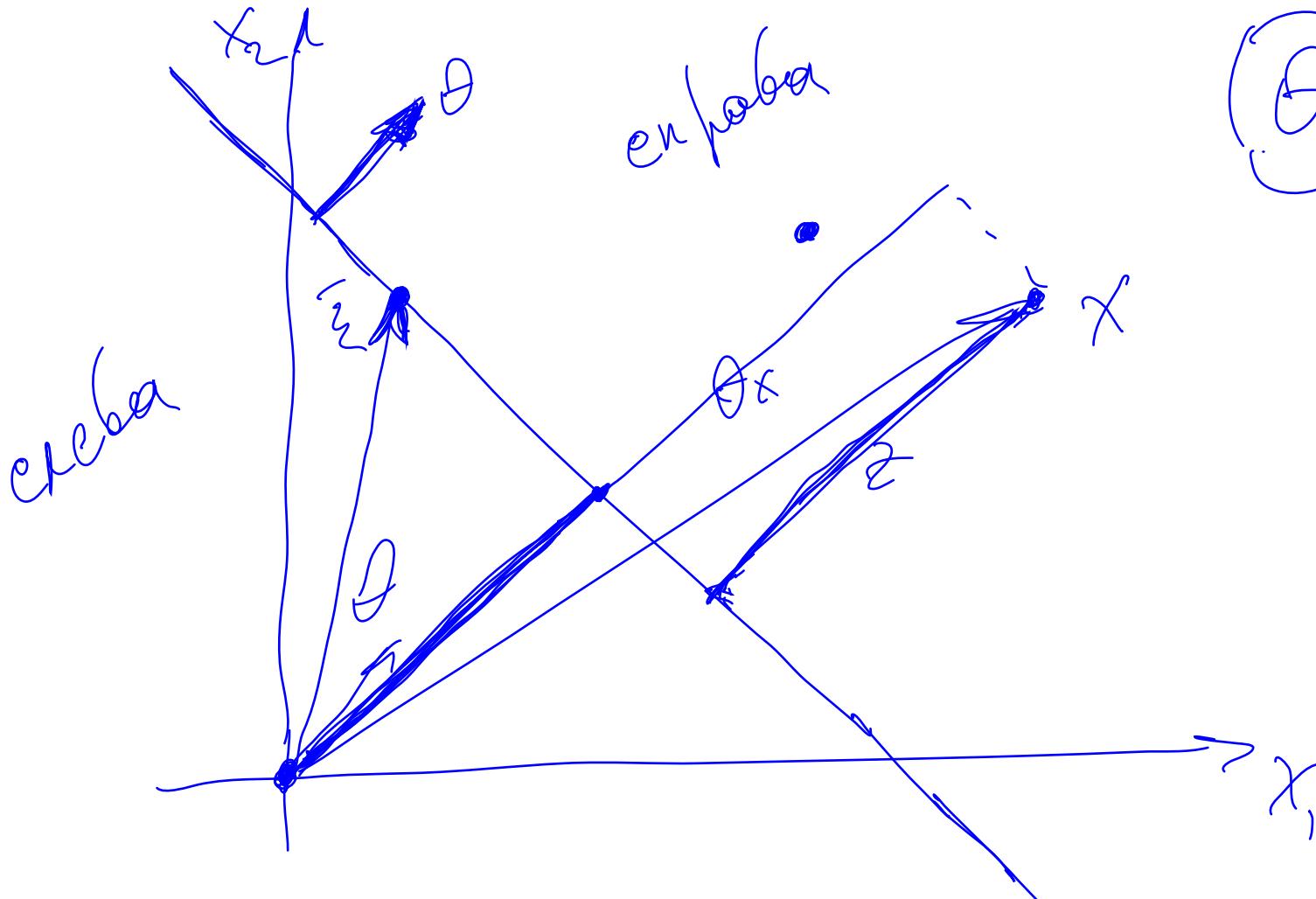
$$\theta \cdot x = -\ln c$$

$$\bar{x} = \bar{x}_{||} + \bar{x}_{\perp}$$

$$\begin{aligned}\bar{\theta} \cdot (\bar{x}_{||} + \bar{x}_{\perp}) &= \\ &= \bar{\theta} \cdot \bar{x}_{||} + \bar{\theta} \cdot \bar{x}_{\perp}\end{aligned}$$



Интерпретация  $z = \theta_0 + \theta \cdot x$



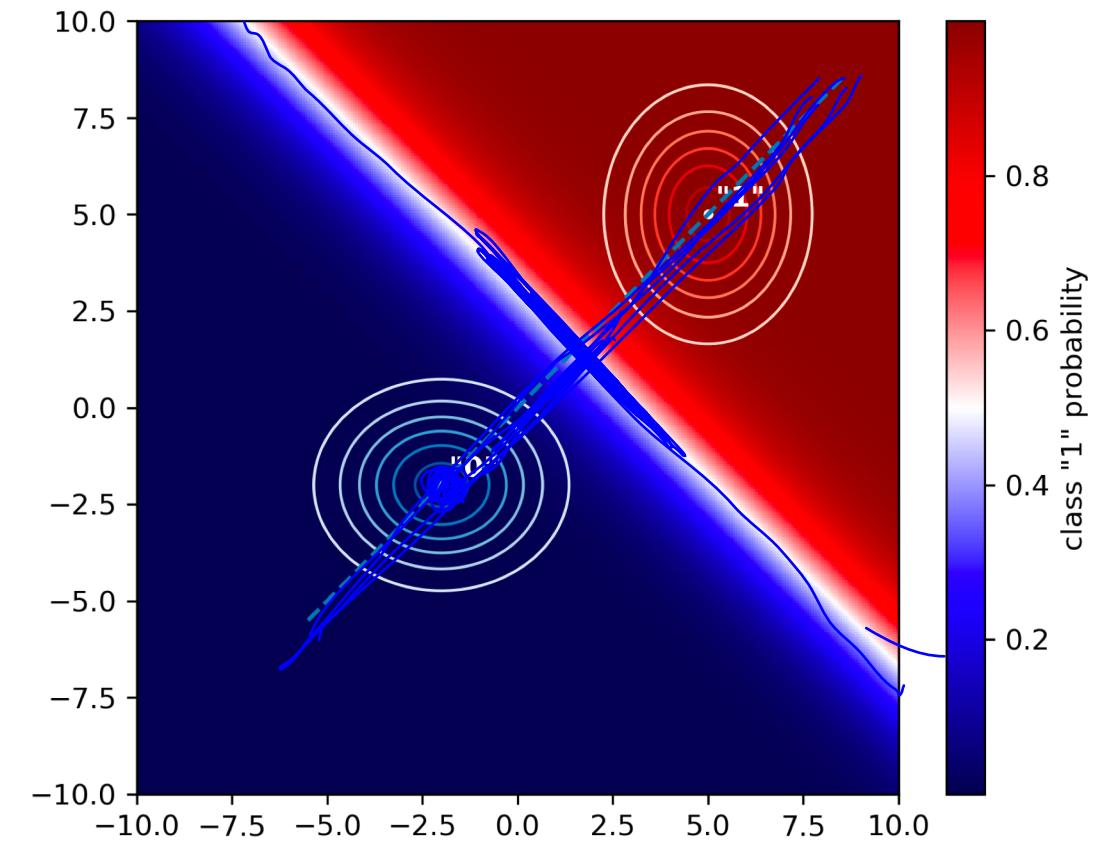
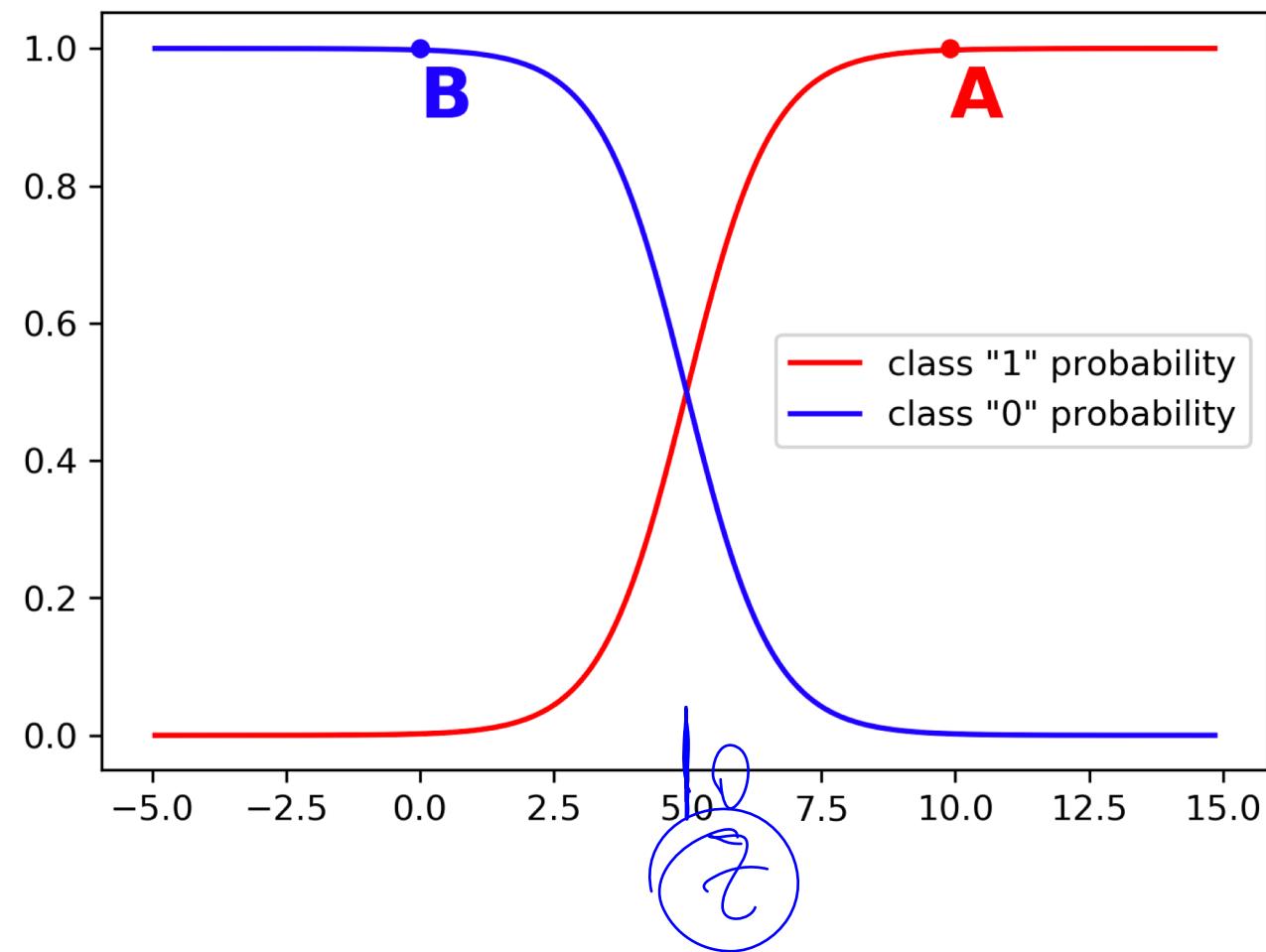
$$x^{(0)} \cancel{\rightarrow}$$

$$\theta_0 + \bar{\theta} \cdot \bar{x} = 0$$

$$\bar{\theta} \cdot \bar{x} = -\theta_0$$

$$\theta_0 + \bar{\theta} \cdot \bar{x} = z$$

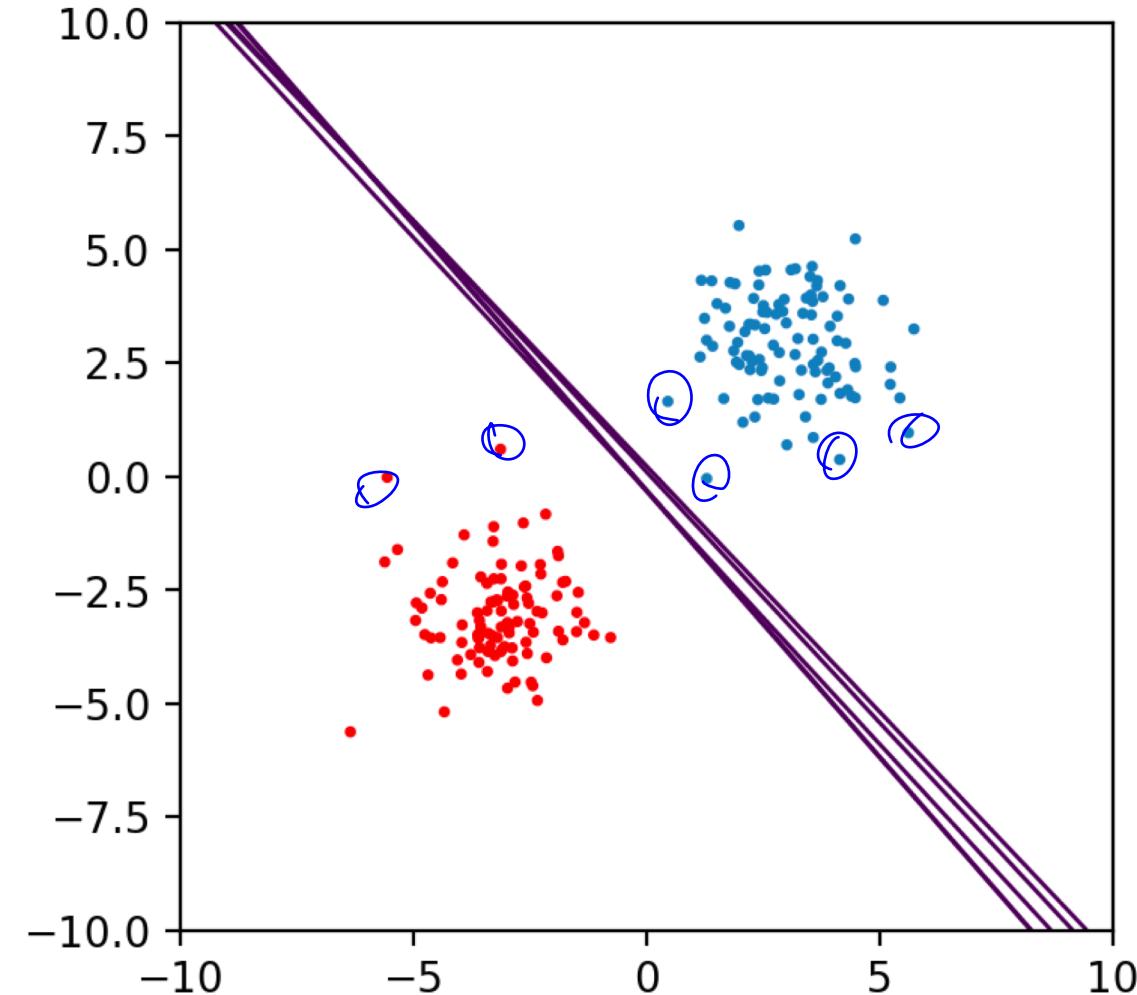
## Интерпретация $z = \theta_0 + \theta \cdot x$



# Метод опорных векторов (Support Vector Machines, SVM)

Идея: построить оптимальную разделяющую поверхность в пространстве признаков.

- пусть эта поверхность будет линейна
- пусть она будет как можно дальше от всех примеров обучающей выборки



# Метод опорных векторов (Support Vector Machines, SVM)

Идея: построить оптимальную разделяющую поверхность в пространстве признаков.

- пусть эта поверхность будет линейна
- пусть она будет как можно дальше от всех примеров обучающей выборки

Соглашение: классам присваиваются значения +1 и -1  
(было в логистической регрессии: 0 и 1)

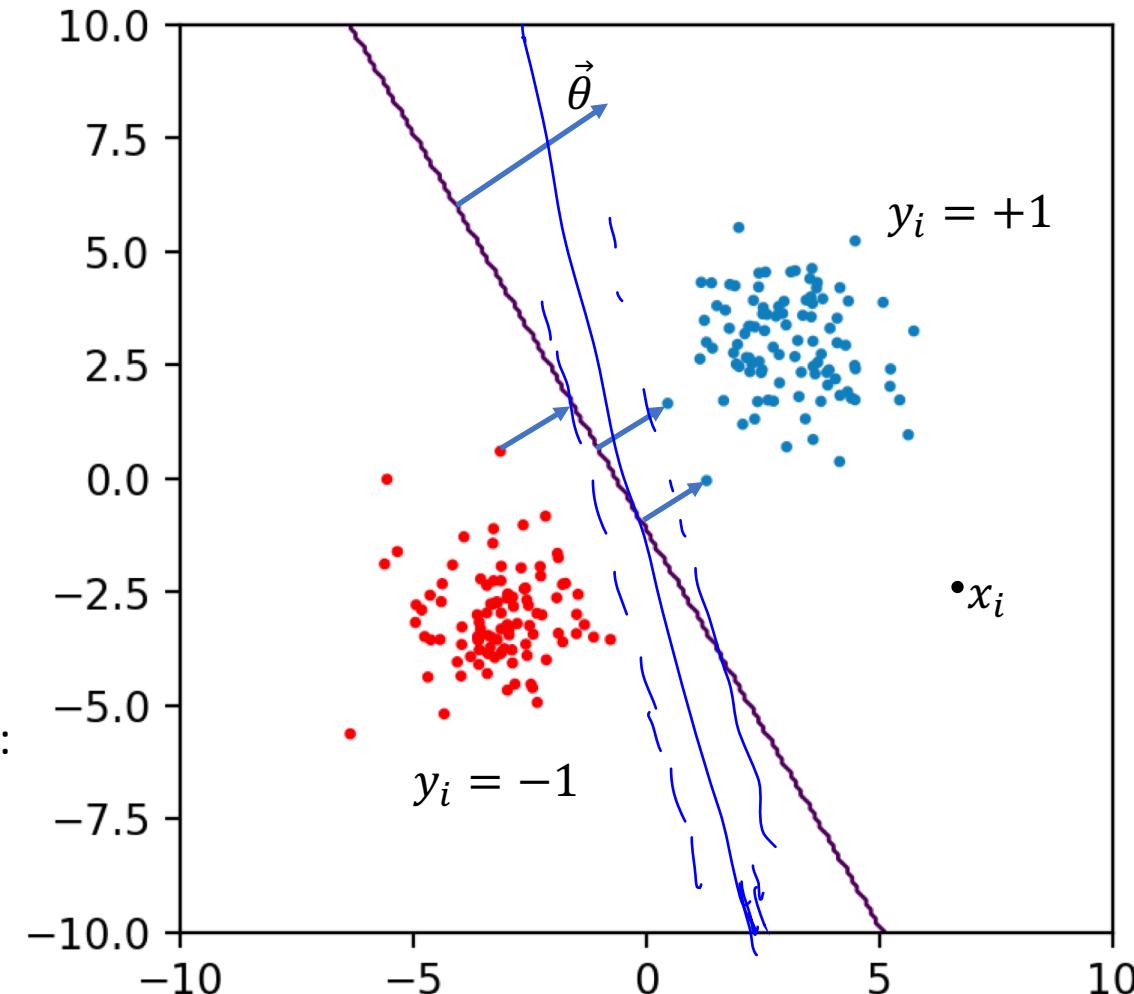
$z = \theta_0 + \theta \cdot x$  — евклидово расстояние (со знаком) от объекта  $x$  до разделяющей плоскости, если  $\|\vec{\theta}\| = 1$  или  $\sum_j \theta_j^2 = 1$

$\theta_0$  — расстояние от разделяющей поверхности до начала координат

Оптимизация:  $\theta^* = \max_{\theta} M$

Где для  $M$  и для всех элементов выборки  $(x_i, y_i)$  выполняется:

$$y_i * (\theta_0 + \theta \cdot x) \geq M$$



# Метод опорных векторов (Support Vector Machines, SVM)

Идея: построить оптимальную разделяющую поверхность в пространстве признаков.

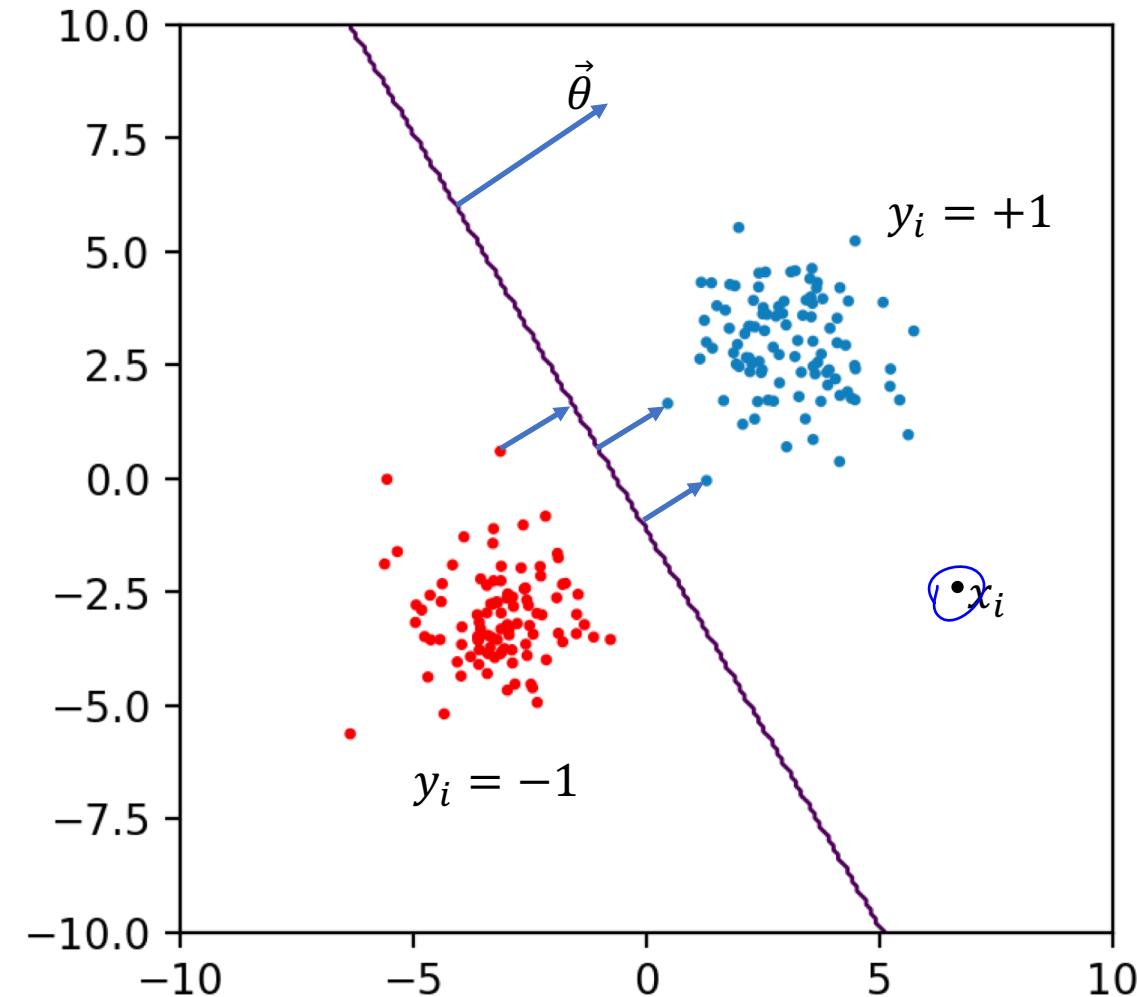
- пусть эта поверхность будет линейна
- пусть она будет как можно дальше от всех примеров обучающей выборки

Как работает SVM в режиме исполнения:

$$z_i = \theta_0 + \theta \cdot x_i$$

Если  $z_i \geq 0$ , тогда  $y_i = +1$  иначе  $y_i = -1$

$$y_i = \text{sign}(\vec{\theta} \cdot \vec{x}_i)$$

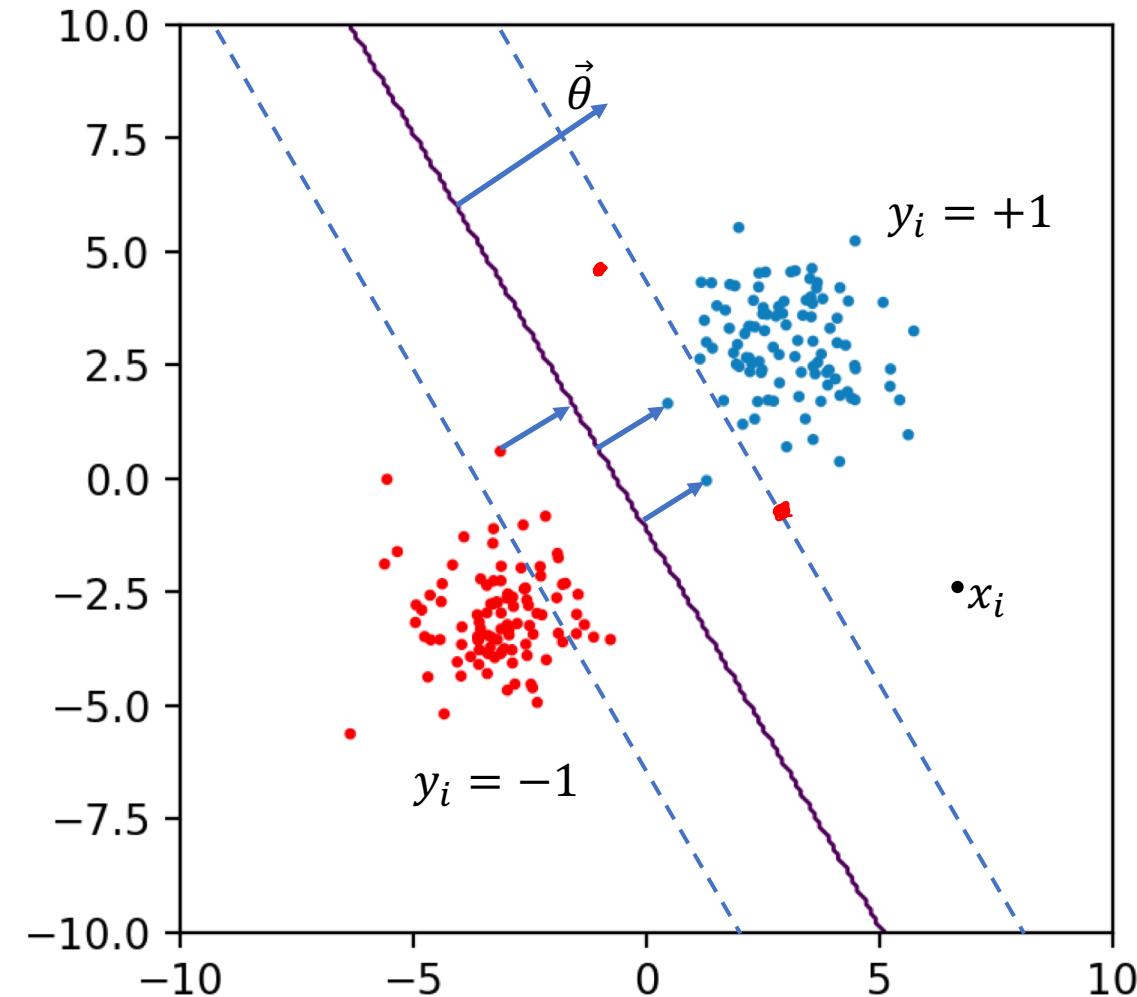


# Метод опорных векторов (Support Vector Machines, SVM)

Идея: построить оптимальную разделяющую поверхность в пространстве признаков.

- пусть эта поверхность будет линейна
- пусть она будет как можно дальше от всех примеров обучающей выборки

Что делать, если выборка шумная? Если выборка не разделима линейно?



# Метод опорных векторов (Support Vector Machines, SVM)

Идея: построить оптимальную разделяющую поверхность в пространстве признаков.

- пусть эта поверхность будет линейна
- пусть она будет как можно дальше от всех примеров обучающей выборки

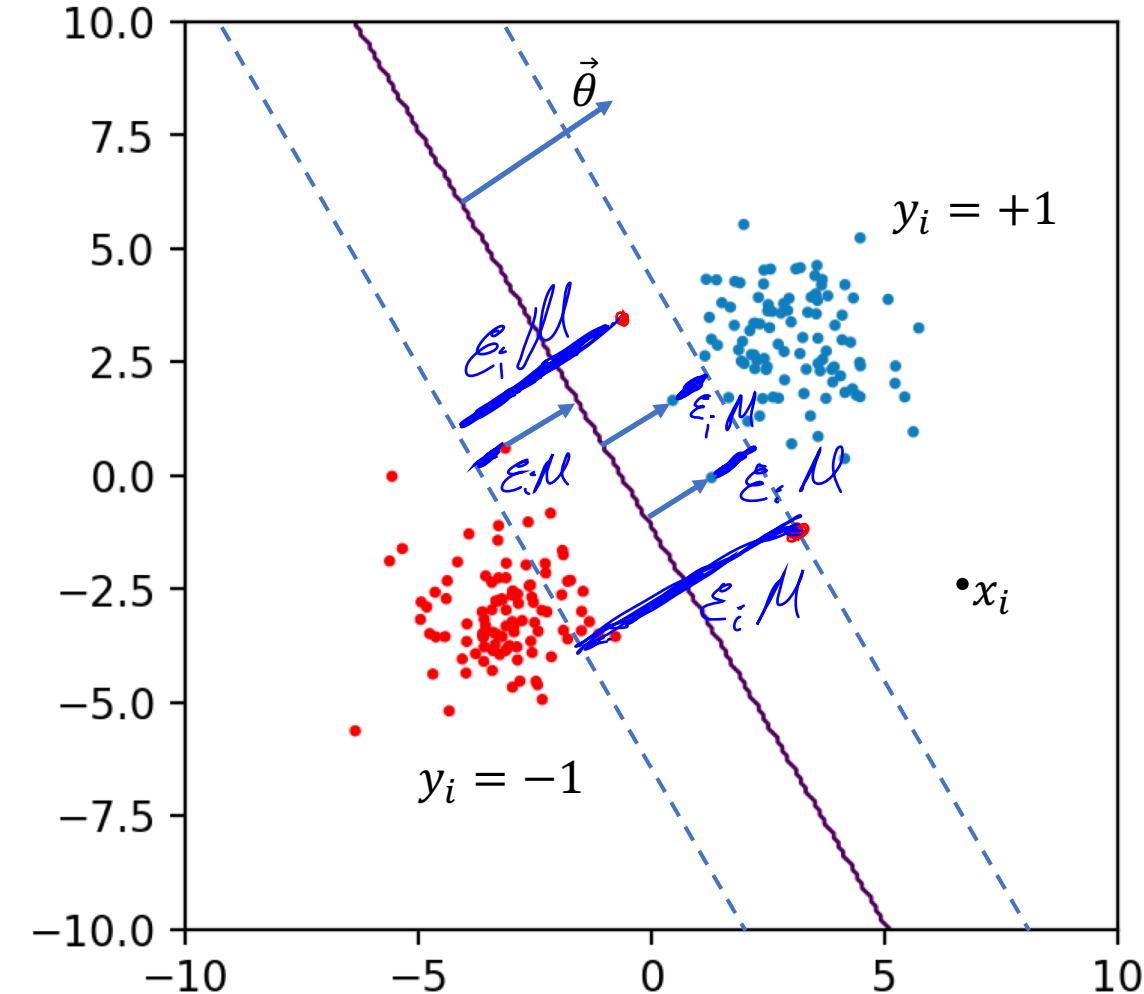
Что делать, если выборка шумная? Если выборка не разделима линейно?

Оптимизируется soft margin:

при условии, что:

$$M \rightarrow \max_{\theta, \theta_0, \epsilon_1, \dots, \epsilon_n}$$
$$\sum_j \theta_j^2 = 1;$$
$$y_i(\theta_0 + \theta \cdot x_i) \geq M(1 - \epsilon_i);$$
$$\epsilon_i > 0;$$
$$\sum_i \epsilon_i \leq C.$$

Задается некоторый «бюджет» для нарушений  $C$ , в который (суммарно  $C * M$ ) должны уложиться «нарушители» жестких границ при оптимизации.

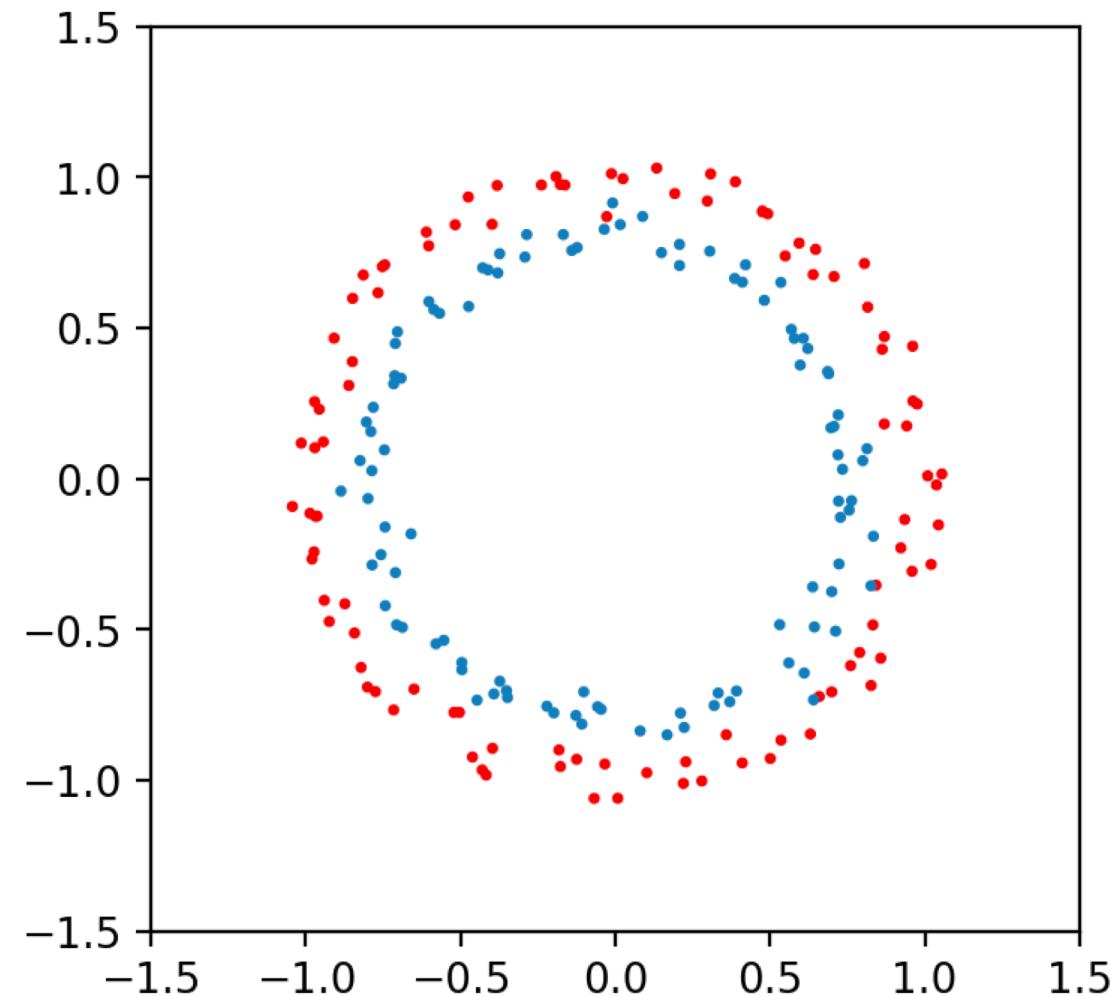


# Метод опорных векторов (Support Vector Machines, SVM)

Идея: построить оптимальную разделяющую поверхность в пространстве признаков.

- пусть эта поверхность будет линейна
- пусть она будет как можно дальше от всех примеров обучающей выборки

Что делать, если выборка шумная? Если выборка ну никак не разделима линейно?



# Метод опорных векторов (Support Vector Machines, SVM)

Идея: построить оптимальную разделяющую поверхность в пространстве признаков.

- пусть эта поверхность будет линейна
- пусть она будет как можно дальше от всех примеров обучающей выборки

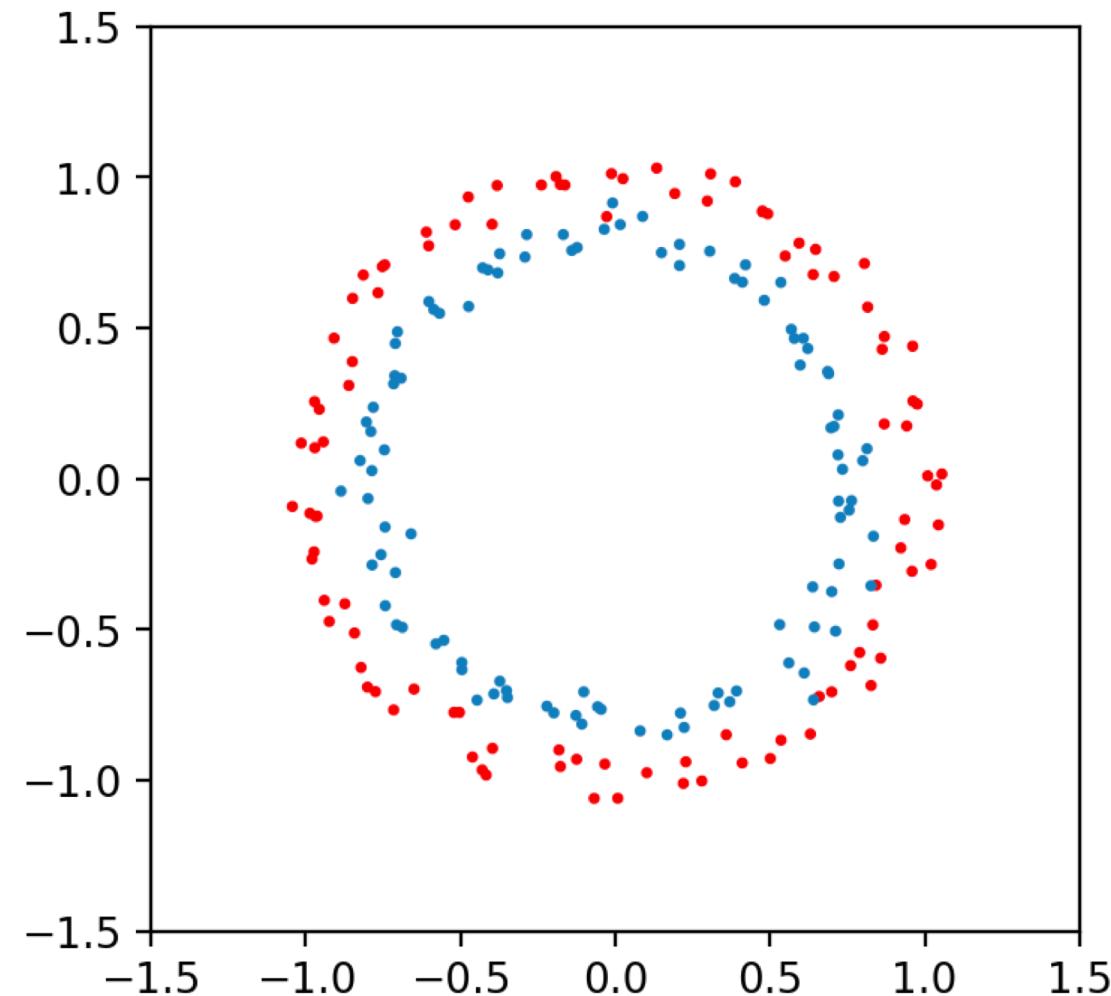
Что делать, если выборка шумная? Если выборка ну никак не разделима линейно?

Значит, надо расширять пространство признаков. Возможно, в более высокоразмерном пространстве эти примеры линейно разделимы.

Вариант №1: полиномиальные признаки

например:  $1, x_1, x_2, x_1^2, x_2^2, x_1x_2$

Проблема: масштаб полиномиальных признаков очень быстро становится слишком большим, что влечет вычислительные проблемы при оптимизации.



# Метод опорных векторов (Support Vector Machines, SVM)

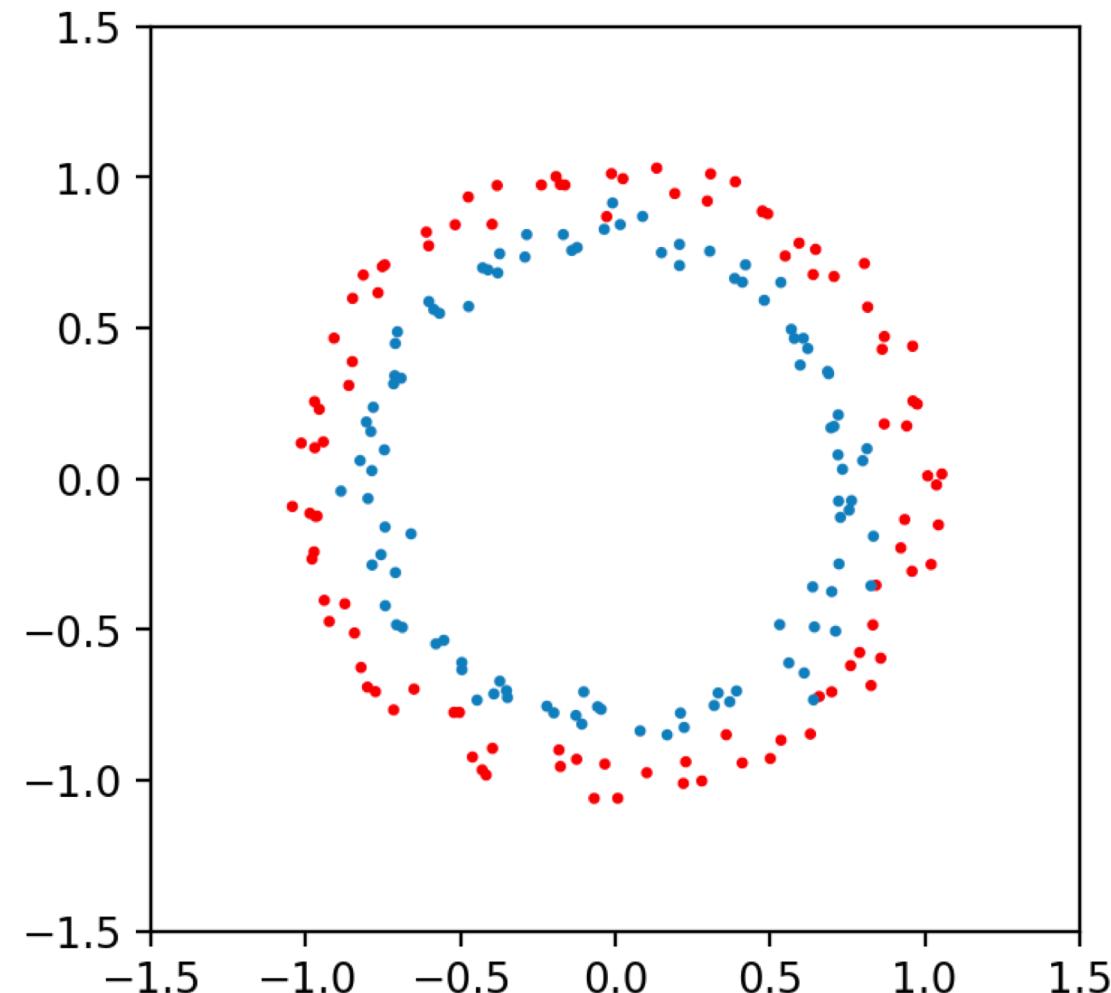
Идея: построить оптимальную разделяющую поверхность в пространстве признаков.

- пусть эта поверхность будет линейна
- пусть она будет как можно дальше от всех примеров обучающей выборки

Что делать, если выборка шумная? Если выборка ну никак не разделима линейно?

Значит, надо расширять пространство признаков. Возможно, в более высокоразмерном пространстве эти примеры линейно разделимы.

Вариант №2: "kernel trick"

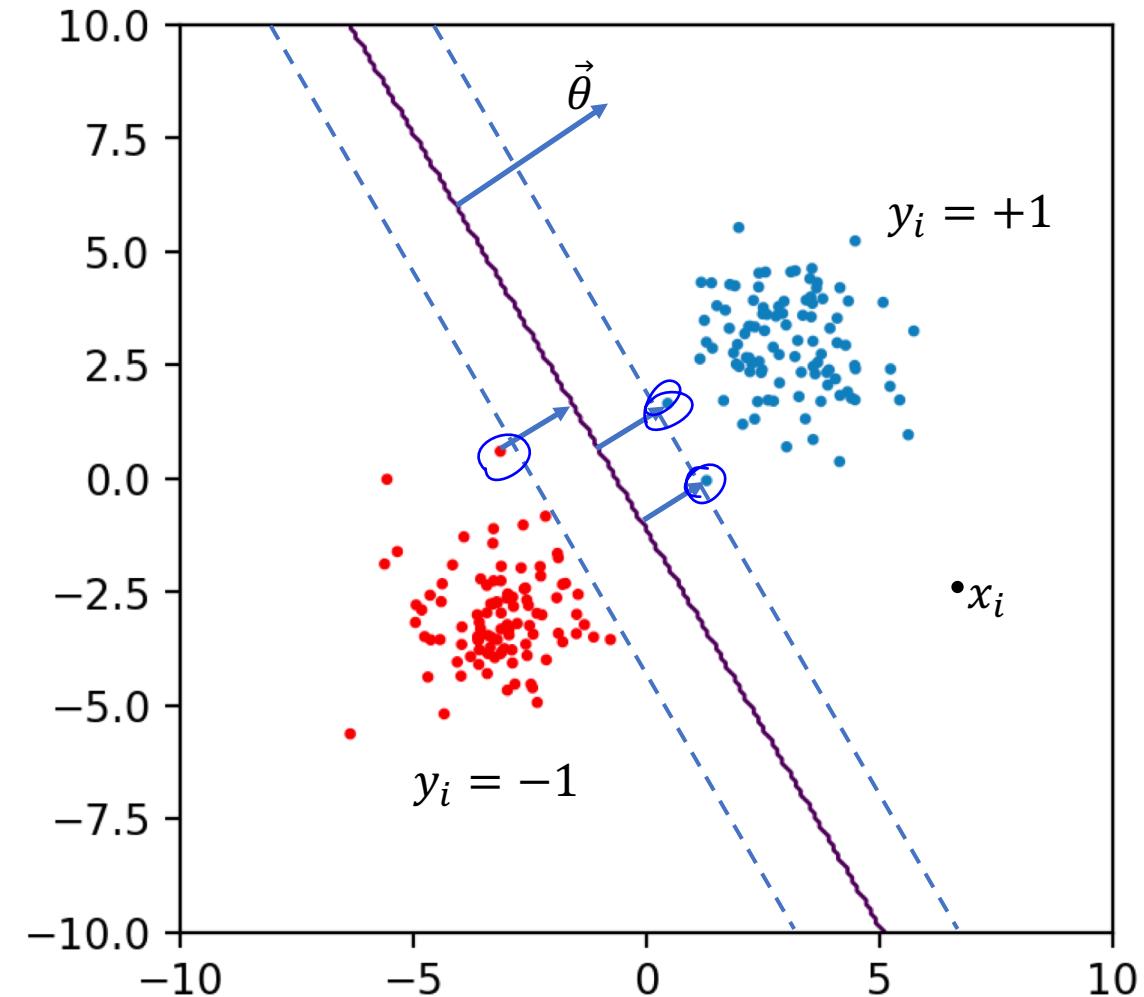


# Метод опорных векторов (Support Vector Machines, SVM)

Потребуем:

$$\frac{\theta \cdot x_+ + \theta_0}{\theta \cdot x_- + \theta_0} \geq +1$$
$$\Rightarrow y_i * (\theta \cdot x_i + \theta_0) \geq 1$$
$$y_i * (\theta \cdot x_i + \theta_0) - 1 \geq 0$$

$y_i * (\theta \cdot x_i + \theta_0) - 1 = 0$   
для элементов на границе



# Метод опорных векторов (Support Vector Machines, SVM)

Потребуем:

$$\begin{aligned} \theta \cdot x_+ + \theta_0 &\geq +1 \\ \theta \cdot x_- + \theta_0 &\leq -1 \end{aligned} \Rightarrow y_i * (\theta \cdot x_i + \theta_0) \geq 1$$

$$y_i * (\theta \cdot x_i + \theta_0) - 1 \geq 0$$

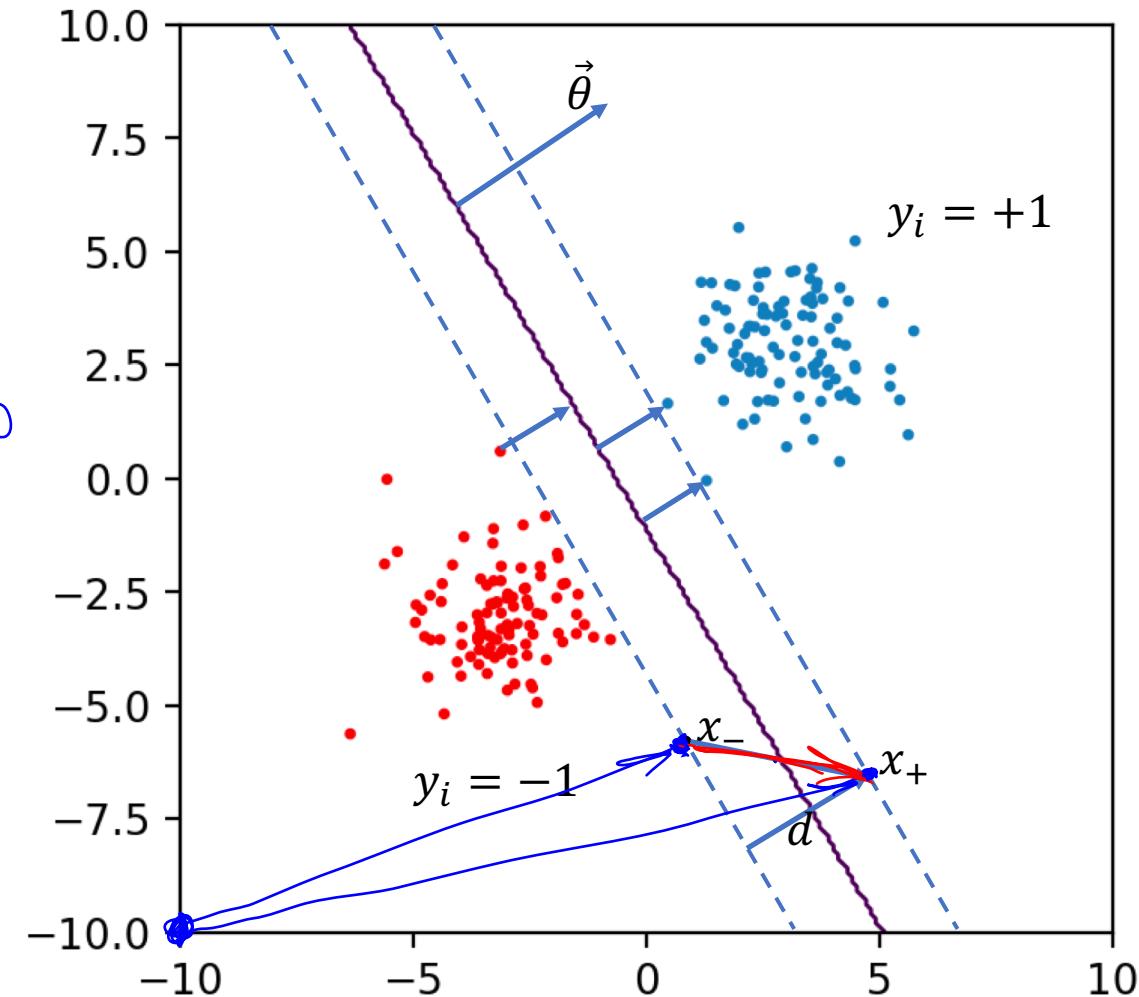
$y_i * (\theta \cdot x_i + \theta_0) - 1 = 0$   
для элементов на границе

ширина:  $d = (x_+ - x_-) \cdot \frac{\theta}{\|\theta\|}$

$$d = \frac{1}{\|\theta\|} (1 - \theta_0 + 1 + \theta_0) = \frac{2}{\|\theta\|}$$

$$d \rightarrow \max \Rightarrow \|\theta\| \rightarrow \min \Rightarrow \min_{\theta} \frac{1}{2} \theta^2$$

$$\begin{aligned} y_i &= 1 \\ \theta \cdot x_+ + \theta_0 - 1 &= 0 \\ \theta \cdot x_- + \theta_0 &= 1 \end{aligned}$$



# Метод опорных векторов (Support Vector Machines, SVM)

Потребуем:  $y_i * (\theta \cdot x_i + \theta_0) - 1 = 0$

для элементов на границе

Оптимизация:  $\min_{\theta} \frac{1}{2} \theta^2$  при условии, что  $y_i * (\theta \cdot x_i + \theta_0) - 1 = 0$

Выпишем лагранжиан:

$$L = \frac{1}{2} \theta^2 - \sum \alpha_i * (y_i * (\theta \cdot x_i + \theta_0) - 1) \rightarrow \min$$

$$\frac{\partial L}{\partial \theta} = \theta - \sum \alpha_i y_i x_i = 0 \Rightarrow \theta = \sum \alpha_i y_i x_i$$

$$\frac{\partial L}{\partial \theta_0} = -\sum \alpha_i y_i = 0 \Rightarrow \sum \alpha_i y_i = 0$$

$$L = \frac{1}{2} (\sum \alpha_i y_i x_i) * (\sum \alpha_j y_j x_j) - (\sum \alpha_i y_i x_i) * (\sum \alpha_j y_j x_j) - \theta_0 \sum \alpha_i y_i + \sum \alpha_i$$

$L = -\frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i x_j + \sum_i \alpha_i \rightarrow \min$

# Метод опорных векторов (Support Vector Machines, SVM)

Идея: построить оптимальную разделяющую поверхность в пространстве признаков.

- пусть эта поверхность будет линейна
- пусть она будет как можно дальше от всех примеров обучающей выборки

Что делать, если выборка шумная? Если выборка ну никак не разделима линейно?

Значит, надо расширять пространство признаков. Возможно, в более высокоразмерном пространстве эти примеры линейно разделимы.

Вариант №2: "kernel trick"

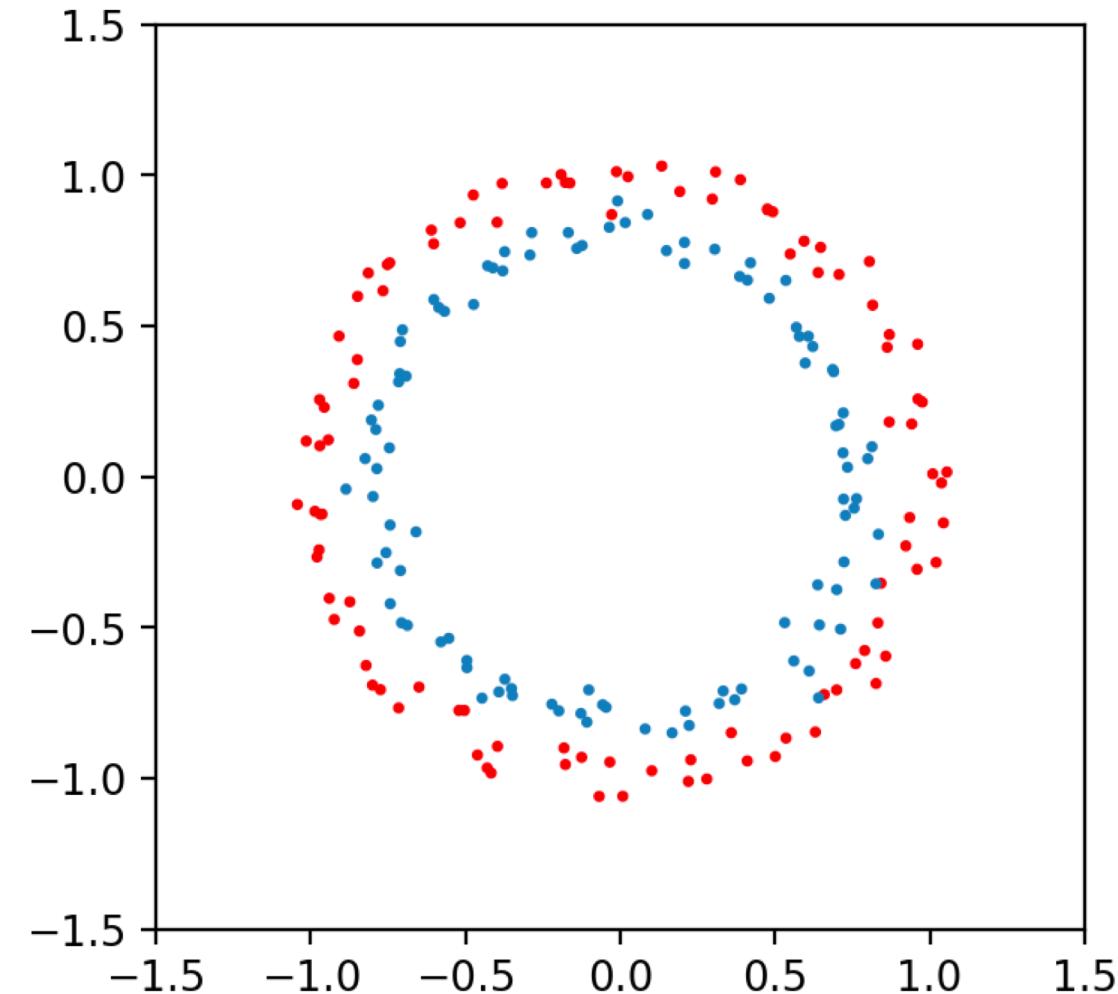
$$z = \theta_0 + \sum \alpha_i y_i x_i$$

причем, по элементам, находящимся на границе – т.н.  
«опорным векторам»

$$z = \theta_0 + \sum_{i \in S} \alpha_i y_i x_i \cdot x$$

значение  $z$  зависит от скалярных произведений  $x_i \cdot x$  с «опорными» элементами. Для таких элементов  $\alpha_i > 0$

$S$  – множество опорных векторов



# Метод опорных векторов (Support Vector Machines, SVM)

Идея: построить оптимальную разделяющую поверхность в пространстве признаков.

- пусть эта поверхность будет линейна
- пусть она будет как можно дальше от всех примеров обучающей выборки

“Kernel trick”

$$z = \theta_0 + \sum_S \alpha_i y_i * x_i \cdot x$$

значение  $z$  зависит от скалярных произведений  $x_i \cdot x$  с «опорными» элементами. Для таких элементов  $\alpha_i > 0$

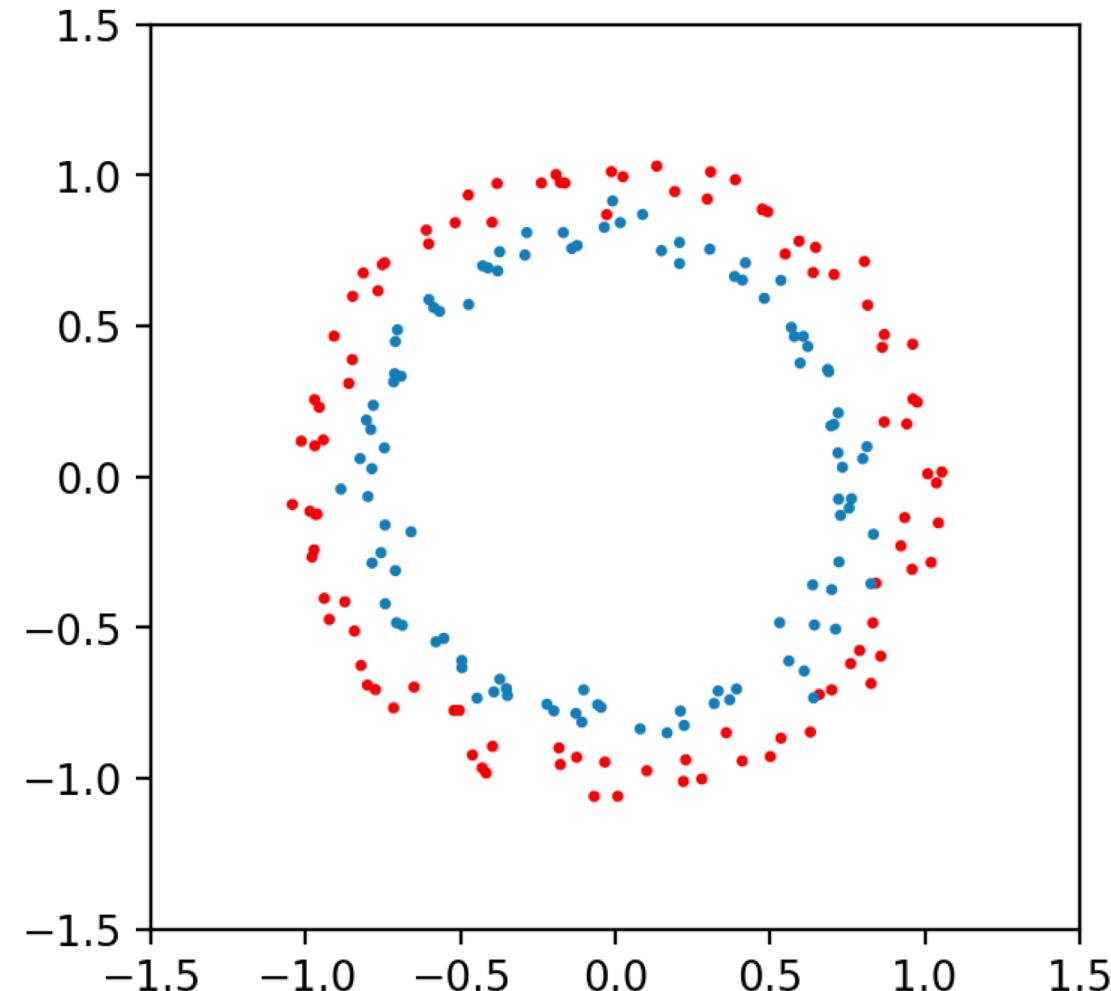
$x_i \cdot x$  – мера близости объекта  $x$  с опорными объектами  $x_i$

Можно заменить ее на любую другую, которой хочется задавать близость объектов.

Примеры:

$$K(x, x_i) = \left( 1 + \sum_{j=1}^p x^{(j)} * x_i^{(j)} \right)^d$$

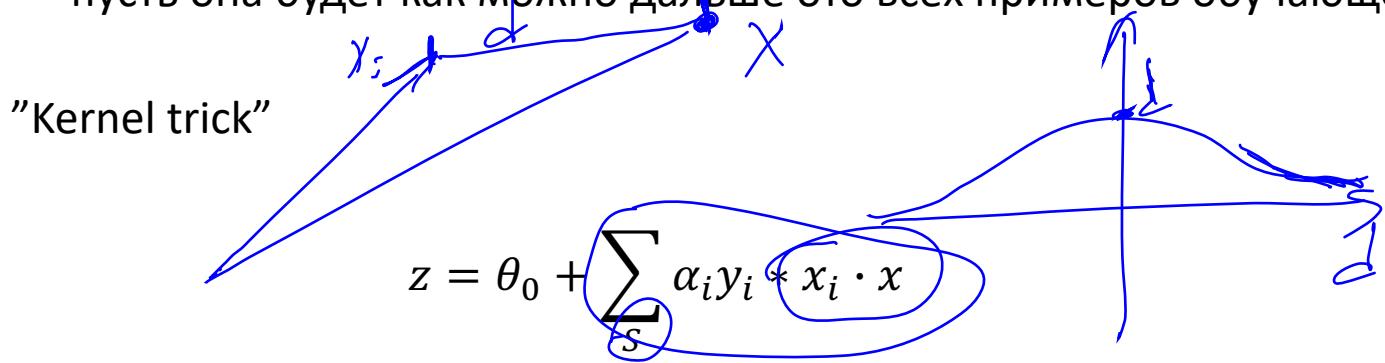
$d$ -полином ( $p$  – исходное количество признаков  $x$ ,  $d$  – степень полинома)



# Метод опорных векторов (Support Vector Machines, SVM)

Идея: построить оптимальную разделяющую поверхность в пространстве признаков.

- пусть эта поверхность будет линейна
- пусть она будет как можно дальше от всех примеров обучающей выборки



значение  $z$  зависит от скалярных произведений  $x_i \cdot x$  с «опорными» элементами. Для таких элементов  $\alpha_i > 0$

$x_i \cdot x$  – мера близости объекта  $x$  с опорными объектами  $x_i$

Можно заменить ее на любую другую, которой хочется задавать близость объектов.

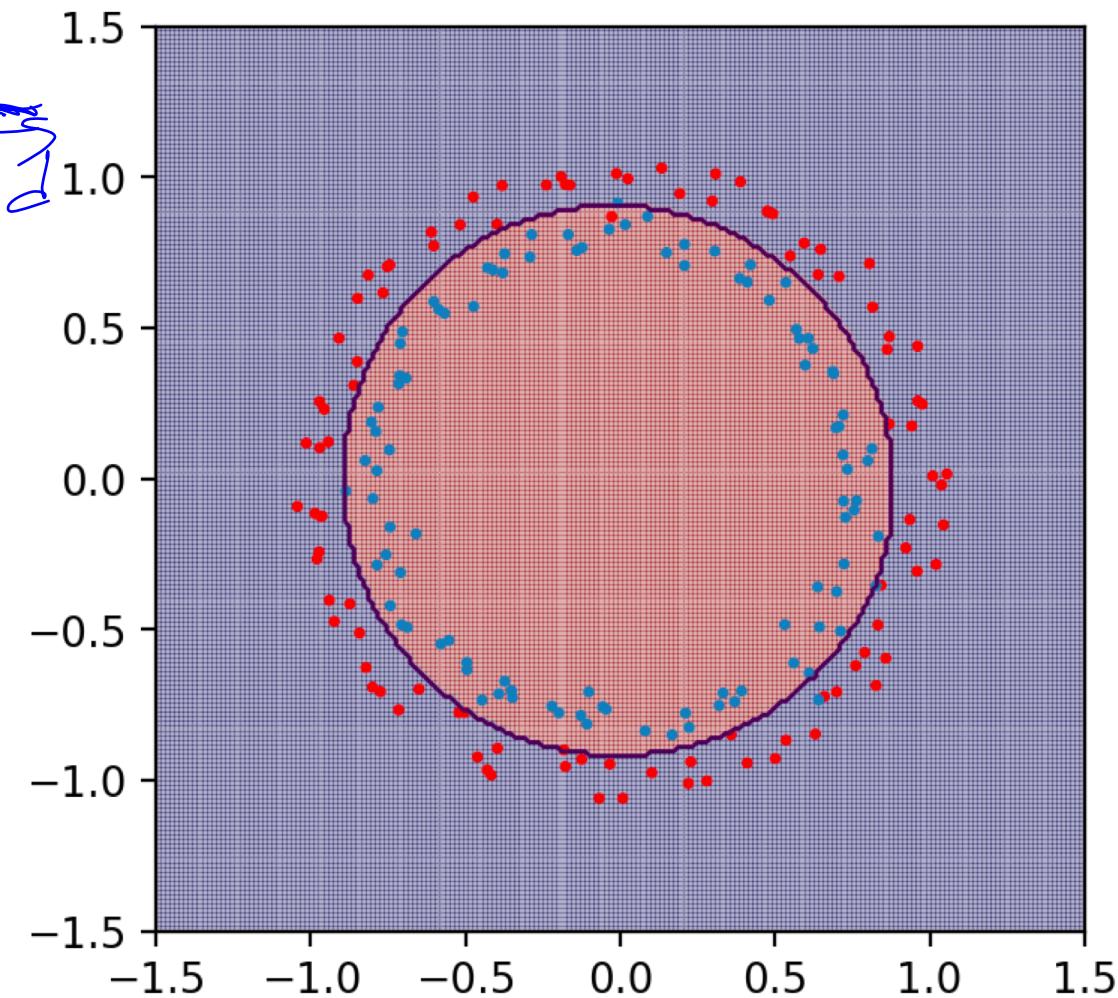
Примеры:

$$K(x, x_i) = \exp(-\gamma \|x - x_i\|^2)$$

т.н. радиально-базисное ядро

(radial kernel, radial-basis function kernel, RBF kernel)

$$z = \theta_0 + \sum_{S} d_i y_i K(x, x_i)$$



# Метод опорных векторов (Support Vector Machines, SVM)

Случай многоклассовой классификации:  $K$  классов

Подход один-против-всех  
One-versus-all (OVA)

- Создать и оптимизировать  $K$  различных моделей SVM
- Для каждого тестового элемента  $x$  вычислить функцию

$$z_k = \theta_0 + \sum_S \alpha_i y_i \langle x_i \cdot x \rangle$$

$$z_k = \theta_0 + \sum \alpha_i y_i l^k(x, x_i)$$

- Выбрать класс, для которого  $z_k$  максимальный

# Метод опорных векторов (Support Vector Machines, SVM)

Случай многоклассовой классификации:  $K$  классов

Подход каждый-против-каждого  
One-versus-one (OVO)

- Создать и оптимизировать  $(K * K - K)/2$  различных моделей SVM для попарной классификации
- Для каждого тестового элемента  $x$  провести классификацию каждой из моделей
- На основании результатов - выбрать наиболее частый класс для этого элемента (голосование большинством)

# Метод опорных векторов (Support Vector Machines, SVM)

## Сравнение SVM и логистической регрессии

### Логистическая регрессия

- Моделируется условное распределение  $p(y|x)$
- Является частным случаем обобщенных линейных моделей
- Можно экспериментировать с расширением признакового описания, однако эти возможности ограничены
- $\mathcal{L}_{LR}(\mathcal{T}, \theta) = \sum_i (\log(1 + e^{\theta \cdot x_i}) - y_i * \theta \cdot x_i) + \lambda \sum_j \theta_j^2$

### SVM

- Напрямую моделируется разделяющая поверхность\*
- SVM (даже линейный) не является обобщенной линейной моделью\*
- Kernel trick предоставляет практически неограниченные возможности для расширения признакового пространства
- $\mathcal{L}_{SVM}(\mathcal{T}, \theta) = \sum_i (\max(0, 1 - y_i * \theta \cdot x_i)) + \lambda \sum_j \theta_j^2$

\* однако есть работы, показывающие эквивалентность SVM вероятностной модели со специальным априорным распределением : Franc, V., Zien, A., Schölkopf, B., 2011. **Support Vector Machines as Probabilistic Models**, in: Getoor, L., Scheffer, T. (Eds.), Proceedings of the 28th International Conference on Machine Learning (ICML-11), ICML '11. ACM, New York, NY, USA, pp. 665–672.

# ОБУЧЕНИЕ С УЧИТЕЛЕМ: задача классификации

## Логистическая регрессия («logistic regression»)

$$\log \frac{p(\theta, x)}{1-p(\theta, x)} = \theta \cdot x, \quad p(\theta, x) = \frac{1}{1+e^{-\theta \cdot x}}$$

Новое обозначение (для краткости):  $p_i \equiv p(\theta, x_i)$

$$L(\mathcal{T}) = \prod_{i=1}^N (p_i^{y_i} * (1 - p_i)^{(1-y_i)})$$

$$\begin{aligned} \ell(\mathcal{T}, \theta) &= \log(L(\mathcal{T}, \theta)) = \sum_i \log(p_i^{y_i}) + \sum_i \log((1 - p_i)^{(1-y_i)}) = \\ &= \sum_i (y_i * \log p_i + (1 - y_i) * \log(1 - p_i)) = \sum_i \log(1 - p_i) + \sum_i y_i * \log \frac{p_i}{1-p_i} = (*) \end{aligned}$$

Бинарная кросс-энтропия (отрицательная)  
Negative binary cross-entropy

$$\left\{ p_i = \frac{1}{1+e^{-\theta \cdot x_i}} : (1 - p_i) = \frac{1}{1+e^{\theta \cdot x_i}} \right\}$$

$$\left\{ \log \frac{p_i}{1-p_i} = \theta \cdot x_i \right\}$$

$$(*) = - \sum_i \log(1 + e^{\theta \cdot x_i}) + \sum_i y_i * \theta \cdot x_i$$

$$\frac{\partial \ell(\mathcal{T}, \theta)}{\partial \theta} = \sum_i (y_i - p(\theta, x_i)) x_i$$

# ОБУЧЕНИЕ С УЧИТЕЛЕМ: задача классификации

## Логистическая регрессия («logistic regression»)

Линейная регрессия:

набор предположений “LINE”

$$y_i \sim \mathcal{N}(\theta \cdot x_i, \sigma^2)$$

$$L(\mathcal{T}) = P(\{y_i, x_i\}) = \prod_{i=1}^N \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_i - \theta \cdot x_i)^2}{2\sigma^2}}$$

$$\ell(\mathcal{T}, \theta) = \log L(\mathcal{T}, \theta) = \sum_{i=1}^N \log \left( \frac{1}{\sqrt{2\pi\sigma^2}} \right) - \sum_{i=1}^N \frac{(y_i - \theta \cdot x_i)^2}{2\sigma^2}$$

$$\theta^* = \operatorname{argmax}_{\Theta} (\ell(\mathcal{T}, \theta)) = \operatorname{argmin}_{\Theta} \sum_{i=1}^N \frac{(y_i - \theta \cdot x_i)^2}{2\sigma^2}$$

Логистическая регрессия:

- (1) Предполагаем, что переменная  $Y$  распределена согласно распределению Бернулли с параметром  $p$ ;
- (2) Предполагаем, что отношение вероятностей классов «1» и «0» соотносятся как линейная функция  $f(\theta, x) = \theta \cdot x$

$$Y \sim \mathcal{B}(p(\theta, x)), \quad \log \frac{p(\theta, x)}{1-p(\theta, x)} = \theta \cdot x, \quad p(\theta, x) = \frac{1}{1+e^{-\theta \cdot x}}$$

$$L(\mathcal{T}) = \prod_{i=1}^N \left( p(x_i)^{y_i} * (1 - p(x_i))^{(1-y_i)} \right)$$

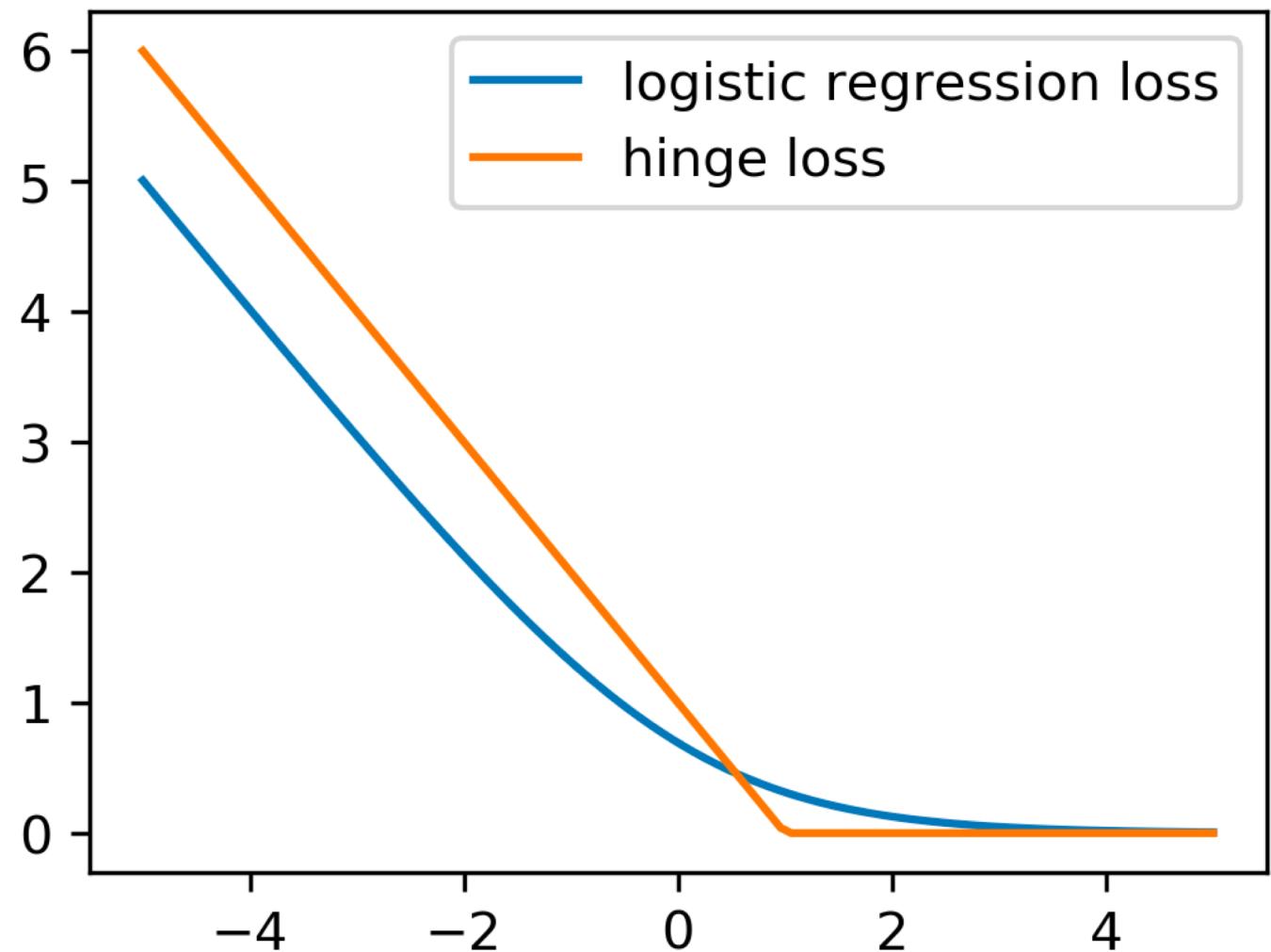
$$\ell(\mathcal{T}, \theta) = - \sum_i \log(1 + e^{\theta \cdot x_i}) + \sum_i y_i * \theta \cdot x_i$$

$$\theta^* = \operatorname{argmax}_{\Theta} (\ell(\mathcal{T}, \theta)) = \operatorname{argmin}_{\Theta} \left( \sum_i (\log(1 + e^{\theta \cdot x_i}) - y_i * \theta \cdot x_i) \right)$$

# Hinge loss

$$\mathcal{L}_{LR}(\mathcal{T}, \theta) = \sum_i (\log(1 + e^{\theta \cdot x_i}) - y_i * \theta \cdot x_i) + \lambda \sum_j \theta_j^2$$

$$\mathcal{L}_{SVM}(\mathcal{T}, \theta) = \sum_i (\max(0, 1 - y_i * \theta \cdot x_i)) + \lambda \sum_j \theta_j^2$$



# Метод опорных векторов (Support Vector Machines, SVM)

## Преимущества и недостатки SVM

### Преимущества

- Работает! И действительно эффективен в высокоразмерных пространствах (но не забываем про «проклятие размерности»)
- Хорошо работает, когда количество признаков превышает количество примеров обучающей выборки
- ОЧЕНЬ хорошо работает, когда классы линейно разделимы
- Разделяющая гиперплоскость (в случае линейного SVM) зависит только от пограничных элементов, то есть, влияние выбросов может быть меньше по сравнению с моделями, полностью учитывающими всю выборку
- Хорошо подходит для бинарной классификации (если это вообще преимущество)

### Недостатки

- Вычислительно дорогой, особенно при возрастании объема обучающей выборки
- Не очень хорошо работает на очень шумных данных и на данных с пересекающимися классами
- Необходимо настраивать гиперпараметры («бюджет»  $C$ , например)
- Ядро  $K$  – гиперпараметр, подбор которого – тоже вычислительно затратная процедура.