

ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ

КАФЕДРА АВТОМАТИЗАЦИЯ ПРОИЗВОДСТВЕННЫХ ПРОЦЕССОВ

ЛЕКЦИЯ № 05

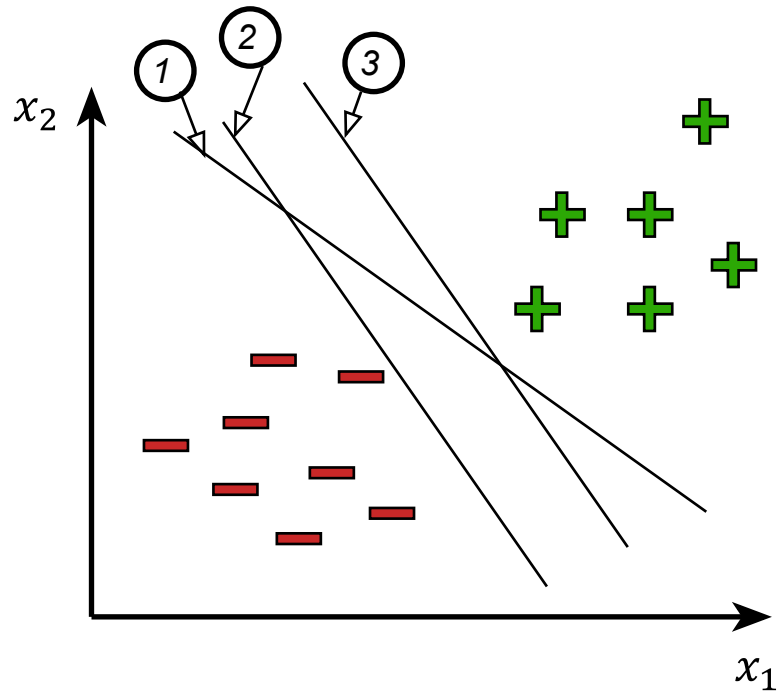
Метод опорных векторов. Ядерные методы

СОСТАВИТЕЛЬ: КАНД. ТЕХН. НАУК БЫКАДОР В.С.

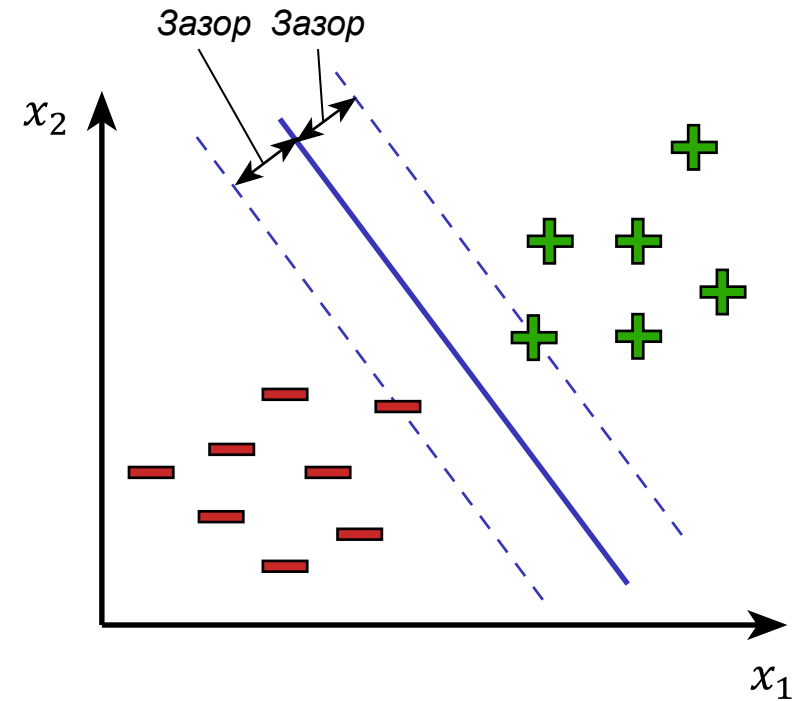
Общие положения метода опорных векторов

В линейных моделях, которые были рассмотрены ранее, между двумя классами объектов можно провести решающую границу по-разному (см. рисунок слева), но интуитивно есть желание провести решающую границу так, чтобы расстояние между решающей границей и ближайшими к ней точками было максимальным для каждого из классов (см. рисунок справа), т.к. такое расстояние будет наиболее робастным.

Расстояние между решающей границей и классами объектов называется **зазором**.



Различные варианты проведения решающей границы в линейных моделях



Решающая граница проведённая с максимальными удалением от двух классов

Общие положения метода опорных векторов

Формально, зазор будет иметь следующий вид:

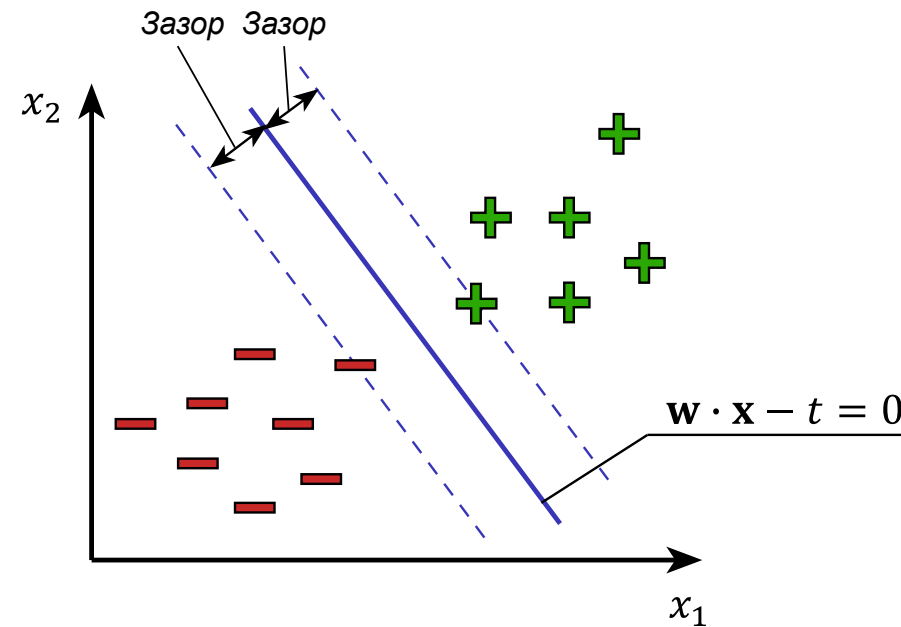
$$z(x) = y(x) \cdot \hat{s}(x)$$

где $y(x) \in \{-1, +1\}$ - метка объекта, которая в методе опорных векторов принимает значение -1 или +1 (в отличие от других методов, где допустимы как значения -1 и +1, так и значения 0 и +1);

Если принять, что оценка объекта \mathbf{x} , равна $\hat{s}(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} - t$

то положительная граница зазора будет равна: $\mathbf{w} \cdot \mathbf{x} - t > 0$

а отрицательная граница зазора будет равна: $-(\mathbf{w} \cdot \mathbf{x} - t) > 0$



Общие положения метода опорных векторов

Обучающие примеры ближе всего расположенные к решающей границе, называются **опорными векторами**.

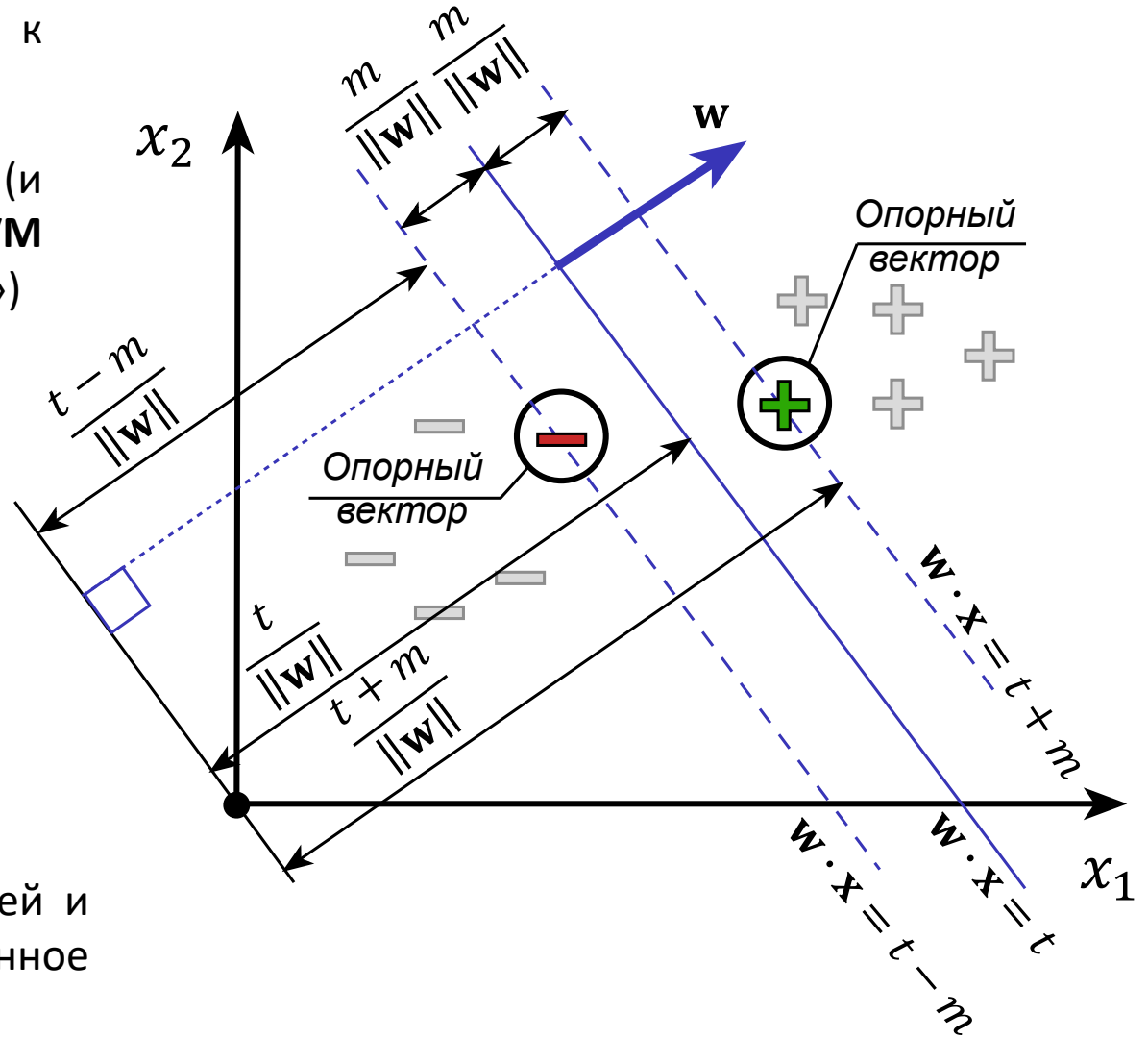
Метод опорных векторов в англоязычной литературе (и программных библиотеках) имеет сокращение **SVM** (**Support Vector Machines** – «машины опорных векторов»)

По схеме справа, зазор может быть найден любым из двух способов:

$$z = \frac{t}{\|\mathbf{w}\|} - \frac{t-m}{\|\mathbf{w}\|} = \frac{t-(t-m)}{\|\mathbf{w}\|} = \frac{t-t+m}{\|\mathbf{w}\|} = \frac{m}{\|\mathbf{w}\|}$$

$$z = \frac{t+m}{\|\mathbf{w}\|} - \frac{t}{\|\mathbf{w}\|} = \frac{t+m-t}{\|\mathbf{w}\|} = \frac{m}{\|\mathbf{w}\|}$$

t - формальное расстояние между решающей границей и ближайшими к ней обучающими объектами, измеренное вдоль вектора \mathbf{w} .



Общие положения метода опорных векторов

Так как мы можем менять масштаб t , $\|\mathbf{w}\|$ и m , то обычно принимают $m = 1$.

Тогда, чтобы зазор был максимальный, необходимо чтобы $\|\mathbf{w}\|$ было минимальными:

$$\left(\frac{m}{\|\mathbf{w}\|} \rightarrow \min \right) \rightarrow \max$$

В общем случае, мы приходим к задаче квадратичной оптимизации в которой $\|\mathbf{w}\|$ заменяется на $\frac{1}{2}\|\mathbf{w}\|^2$, при условии, что ни один из обучающих объектов не попадает внутрь зазора. Тогда математически можно записать:

$$\mathbf{w}^*, t^* = \operatorname{argmin}_{\mathbf{w}, t} \frac{1}{2} \|\mathbf{w}\|^2 \quad \text{при условии} \quad y_i(\mathbf{w} \cdot \mathbf{x} - t) \geq 1, 1 \leq i \leq n.$$

Общие положения метода опорных векторов

Для решения задачи оптимизации используются множители Лагранжа, тогда прибавление ограничений для всех обучающих объектов, помноженных на множители Лагранжа α_i , даёт функцию Лагранжа:

$$\Lambda(\mathbf{w}, t, \alpha_1, \dots, \alpha_n) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i (y_i (\mathbf{w} \cdot \mathbf{x}_i - t) - 1)$$

Выполнив математические преобразования и взяв частные производные по \mathbf{w} и t :

$$\frac{\partial \Lambda(\mathbf{w}, t, \alpha_1, \dots, \alpha_n)}{\partial \mathbf{w}} = 0$$

$$\frac{\partial \Lambda(\mathbf{w}, t, \alpha_1, \dots, \alpha_n)}{\partial t} = 0$$

$$\Rightarrow \mathbf{w} - \sum_{i=1}^n \alpha_i \cdot y_i \cdot \mathbf{x}_i = 0$$

\Downarrow

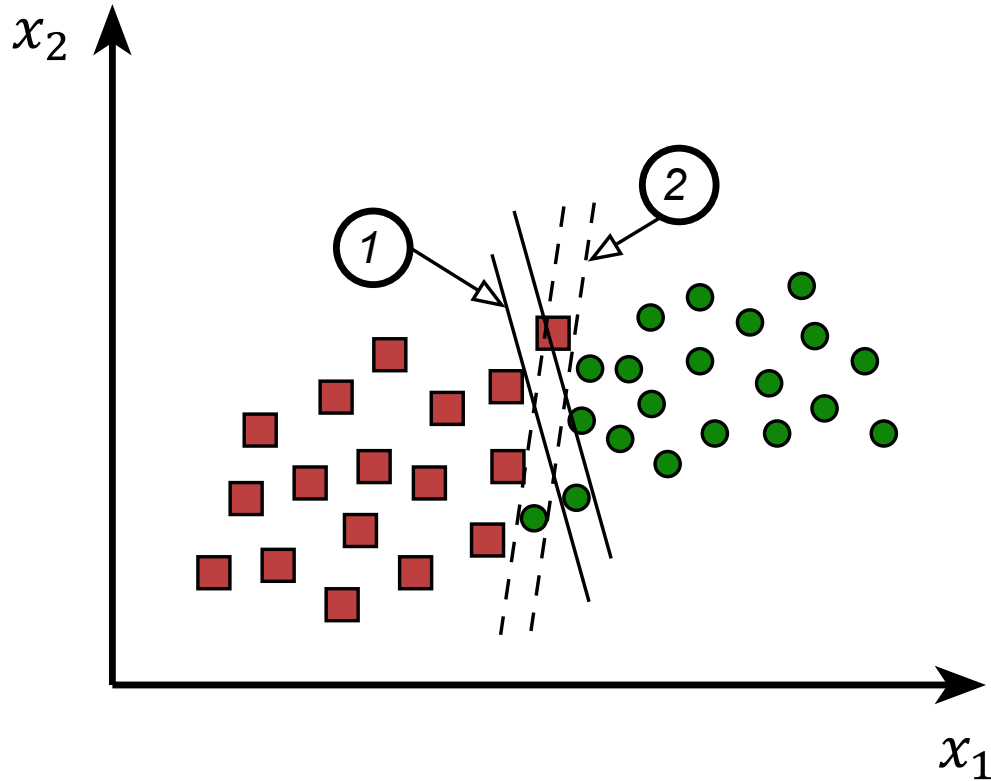
вес объекта

$$\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i$$

где $\alpha_i \geq 0$

$\alpha_i > 0$ только для опорных векторов, т.е. обучающих объектов, ближайших к решающей границе.

Метода опорных векторов с мягким зазором



Если данные не являются линейно разделимыми, то ограничения $\mathbf{w} \cdot \mathbf{x} - t \geq 1$, налагаемые обучающими объектами, невозможно удовлетворить повсеместно.

Для решения данной проблемы, предложено использовать так называемые **мягкие зазоры**.

Идея заключается в том, чтобы ввести некоторые **ослабляющие переменные** ξ_i для каждого обучающего объекта, благодаря которым некоторые обучающие объекты могут оказаться внутри зазора или с неправильной стороны от решающей границы. Такое явление принято называть **ошибками зазора**.

Тогда ограничения будут иметь вид: $\mathbf{w} \cdot \mathbf{x}_i - t \geq 1 - \xi_i$.

Задача оптимизации в таком случае будет называться задачей оптимизации с **мягким зазором** и иметь следующий общий вид:

$$\mathbf{w}^*, t^*, \xi_i^* = \underset{\mathbf{w}, t, \xi_i}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i \quad \text{при условии} \quad y_i(\mathbf{w} \cdot \mathbf{x} - t) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad 1 \leq i \leq n.$$

Свойства метода опорных векторов с мягким зазором

В библиотеках, используемых для машинного обучения, пользователю необходимо задавать значение параметра C . Данный параметр регулирует величину зазора за счёт уменьшения или увеличения ослабляющих переменных ξ_i .

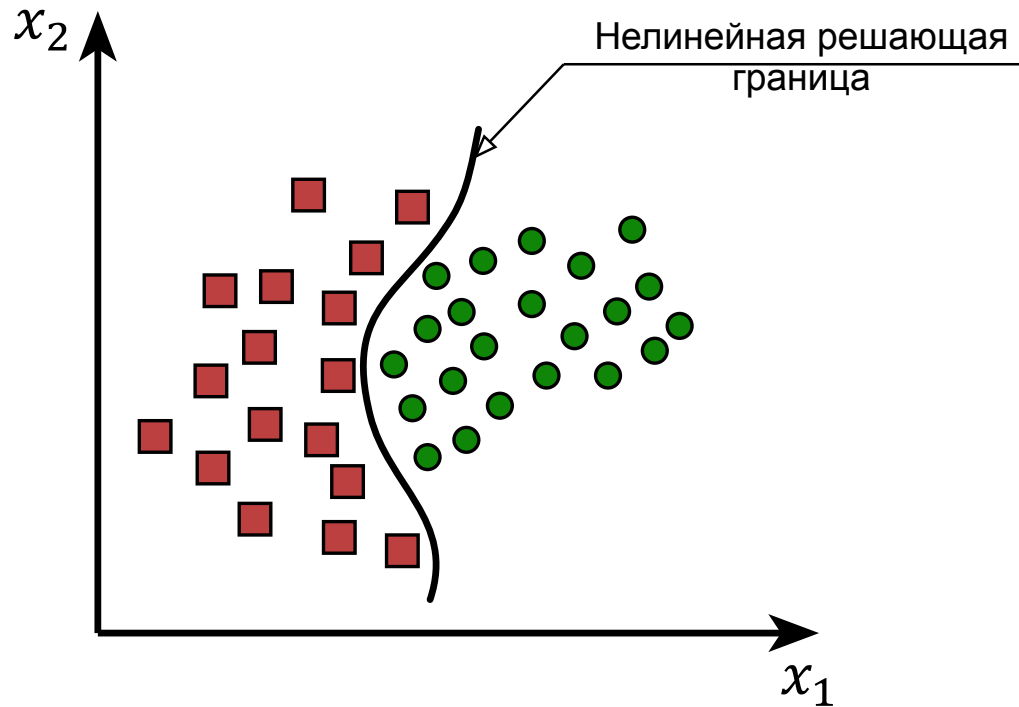
Большое значение $C \Rightarrow$ большой штраф за ошибки зазора, то есть зазор должен быть уменьшен, чтобы как можно меньше обучающих объектов попадали во внутрь зазора. Решение негибкое, но более точное.

Маленькое значение $C \Rightarrow$ штраф за ошибки зазора маленький, то есть зазор может быть увеличен, допускается неправильная классификация. Решение более гибкое, но менее точное.

Чем меньше значение параметра C , тем меньше опорных векторов требуется для построения зазора \rightarrow что модель машинного обучения будет проще, поэтому говорят, что параметр C управляет «сложностью» SVM, поэтому параметр C часто называют **параметром сложности**.

- $\alpha_i = 0$ – обучающие объекты находятся вне зазора или на границе зазора (не участвуют в создании границы);
- $0 < \alpha_i < C$ – обучающие объекты на границе зазора (участвуют в создании границы);
- $\alpha_i = C$ – обучающие объекты внутри или на границе зазора (участвуют в создании границы).

Ядерные методы опорных векторов



Ядерные методы могут применяться тогда, когда между классами объектов имеется нелинейная решающая граница. По своей сути ядерные методы являются более универсальными, чем методы линейных моделей. При выборе определенных ядер можно получить линейную модель.

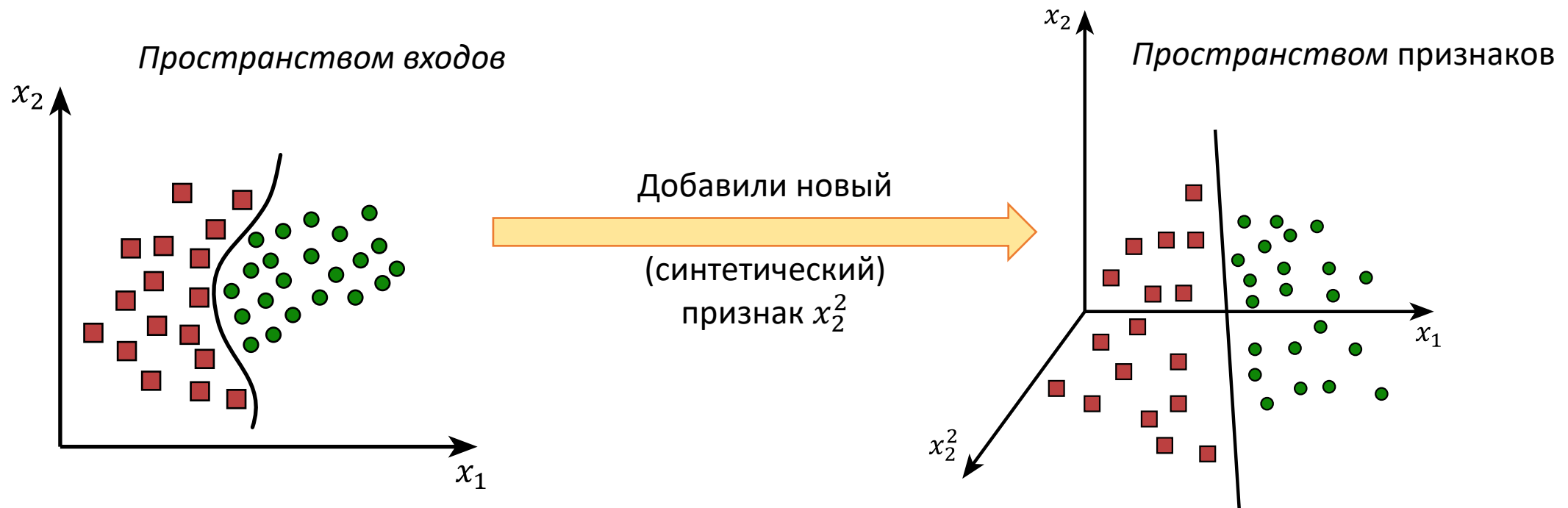
Основной принцип ядерных методов

Основной принцип ядерных методов: выполнить нелинейное преобразование данных в пространство признаков, в котором можно будет применить линейную классификацию.

Исходное пространство называется **пространством входов**.

Преобразованное пространство называется **пространством признаков**.

Для отображения пространства входов в пространства признаков как правило добавляют новый признак, который является комбинацией существующих входов.



Ядерные методы

Вывод: добавление нелинейных признаков может улучшить прогнозные свойства линейной модели.

Проблема: точно неизвестно какие признаки добавлять.

Перебор всех возможных вариантов => увеличивает стоимость вычислений и количество нелинейных комбинаций громадно.

Решение: использовать специальный математический приём, который позволит обучить классификатор в многомерном пространстве признаков при этом не прибегая к вычислению нового, вполне возможно, очень высокоразмерного пространства.



**ЯДЕРНЫЙ ТРЮК
(KERNEL TRICK)**

Ядерный трюк

Вернёмся к ранее полученному выражению вектора весов для метода опорных векторов:

$$\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i$$

Модифицируем это выражение. Заменяем i на j , $n = |D|$ - количество объектов в обучающем наборе, а \mathbf{x}_j на некоторую функцию $\kappa(\mathbf{x}_i, \mathbf{x}_j)$:

$$i \rightarrow j \Rightarrow \sum_{j=1}^n \alpha_j y_j \mathbf{x}_j$$

$$n \rightarrow |D| \Rightarrow \sum_{j=1}^{|D|} \alpha_j y_j \mathbf{x}_j$$

$$\mathbf{x}_j \rightarrow \kappa(\mathbf{x}_i, \mathbf{x}_j) \Rightarrow \sum_{j=1}^n \alpha_j y_j \kappa(\mathbf{x}_i, \mathbf{x}_j)$$



$$\mathbf{w} = \sum_{j=1}^{|D|} \alpha_j y_j \kappa(\mathbf{x}_i, \mathbf{x}_j)$$

Ядерный трюк

$$\mathbf{w}(\mathbf{x}_i) = \sum_{j=1}^{|D|} \alpha_j y_j \kappa(\mathbf{x}_i, \mathbf{x}_j) \quad - \text{вес объекта } \mathbf{x}_i.$$

где α_j - j -ый множитель Лагранжа, который следует ещё найти;

y_j - j -ая метка для обучающего объекта, $y_j \in \{-1 \quad +1\}$;

$\kappa(\mathbf{x}_i, \mathbf{x}_j)$ - ядро;

\mathbf{x}_i - объект подлежащий классификации;

\mathbf{x}_j - j -ый обучающий объект.

Ядерный трюк

$$\mathbf{w}(\mathbf{x}_i) = \sum_{j=1}^5 \alpha_j y_j \kappa(\mathbf{x}_i, \mathbf{x}_j) = \alpha_1 y_1 \kappa(\mathbf{x}_i, \mathbf{x}_1) + \alpha_2 y_2 \kappa(\mathbf{x}_i, \mathbf{x}_2) + \alpha_3 y_3 \kappa(\mathbf{x}_i, \mathbf{x}_3) + \alpha_4 y_4 \kappa(\mathbf{x}_i, \mathbf{x}_4) + \alpha_5 y_5 \kappa(\mathbf{x}_i, \mathbf{x}_5)$$

Можно сказать, что классифицируемый объект \mathbf{x}_i сопоставляется с каждым обучающим объектом \mathbf{x}_j , что приводит к вычислению веса \mathbf{w} классифицируемого объекта \mathbf{x}_i .

Так как для обучающих объектов, которые не являются опорными векторами, то есть не участвуют в формировании зазора множители Лагранжа $\alpha_i = 0$, то далеко не все слагаемые необходимо вычислять при нахождении веса классифицируемого объекта. Такая особенность машины опорных векторов как раз и привела к её частому использованию с ядерными методами.

$$\mathbf{w}(\mathbf{x}_i) = \sum_{j=1}^5 \alpha_j y_j \kappa(\mathbf{x}_i, \mathbf{x}_j) = \underline{\alpha_1 y_1 \kappa(\mathbf{x}_i, \mathbf{x}_1)} + \cancel{\alpha_2 y_2 \kappa(\mathbf{x}_i, \mathbf{x}_2)} + \cancel{\alpha_3 y_3 \kappa(\mathbf{x}_i, \mathbf{x}_3)} + \underline{\alpha_4 y_4 \kappa(\mathbf{x}_i, \mathbf{x}_4)} + \cancel{\alpha_5 y_5 \kappa(\mathbf{x}_i, \mathbf{x}_5)}$$

Ядро

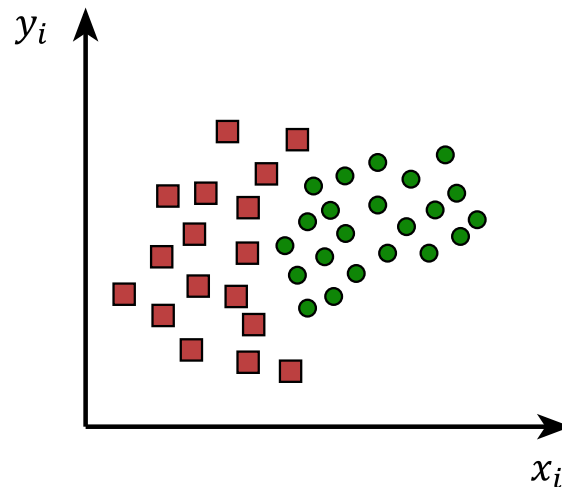
Ядром $\kappa(\mathbf{x}_i, \mathbf{x}_j)$ может быть любая функция, которая позволяет расширить пространство признаков, за счёт введения нелинейных признаков. Тем не менее существуют хорошо себя зарекомендовавшие ядра.

Но до начала давайте рассмотрим, как ядро может расширить пространство признаков, введя в пространство нелинейные признаки.

Будем рассматривать двумерный случай, тогда:

$$\mathbf{x}_i = (x_i \quad y_i) \quad \mathbf{x}_j = (x_j \quad y_j)$$

N.B.!: y_i и y_j не метки объектов, здесь это координаты в пространстве признаков.



Ядро

$$\mathbf{x}_i = (x_i \ y_i) \quad \mathbf{x}_j = (x_j \ y_j)$$

Ядро

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j = x_i x_j + y_i y_j$$

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^2 \cdot \mathbf{x}_j^2 = x_i^2 x_j^2 + y_i^2 y_j^2$$

$$\begin{aligned} \kappa(\mathbf{x}_i, \mathbf{x}_j) &= (\mathbf{x}_i \cdot \mathbf{x}_j)^2 = (x_i x_j + y_i y_j)^2 = \\ &= x_i^2 x_j^2 + y_i^2 y_j^2 + 2x_i x_j y_i y_j \end{aligned}$$

Пространство признаков

$$\phi(\mathbf{x}) = (x, y)$$

$$\phi(\mathbf{x}) = (x^2, y^2)$$

$$\phi(\mathbf{x}) = (x^2, y^2, \sqrt{2}xy)$$

Проверим:

$$\begin{aligned} \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) &= (x_i^2, y_i^2, \sqrt{2}x_i y_i) \cdot (x_j^2, y_j^2, \sqrt{2}x_j y_j) = x_i^2 x_j^2 + y_i^2 y_j^2 + \sqrt{2}x_i y_i \sqrt{2}x_j y_j = \\ &= x_i^2 x_j^2 + y_i^2 y_j^2 + 2x_i y_i x_j y_j = (\mathbf{x}_i \cdot \mathbf{x}_j)^2 \end{aligned}$$

Полиномиальное ядро

Тогда можно получить одно из самых распространённых типов ядра – **полиномиальное ядро**.

Данный тип ядра позволяет преобразовать d -мерное пространство входов в пространство признаков большей размерности.

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j)^p$$

или в ещё более общем виде

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^p$$

Например, ядро $\kappa(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^p$ при $p = 2$, позволяет получить результирующее пространство признаков:

$$\phi(\mathbf{x}) = (x^2, y^2, \sqrt{2}xy, \sqrt{2}x, \sqrt{2}y, 1)$$

То есть входов (реальных признаков) два - $\mathbf{x} = (x \ y)$, но полиномиальное ядро во второй степени дало пространство признаков состоящее из шести признаков (один из них постоянный).

Гауссовское ядро

Второй тип часто используемых ядер – гауссовское (гауссово) ядро.

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right) = \exp\left(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2\right)$$

где $\|\mathbf{x}_i - \mathbf{x}_j\|$ - евклидово расстояние;

$\gamma = \frac{1}{2\sigma^2}$ - параметр, который регулирует ширину полосы пропускания гауссовского ядра.

Классификатор для ядерных методов

$$\hat{y}(\mathbf{x}_i) = \text{sign}(\mathbf{w}(\mathbf{x}_i))$$



-1

+1

0 – на разделяющей границе, очень мало вероятно.

$$\mathbf{w}(\mathbf{x}_i) = \sum_{j=1}^{|D|} \alpha_j y_j \kappa(\mathbf{x}_i, \mathbf{x}_j)$$

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^p$$

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$$

Алгоритм обучения ядерного перцептрона

Вход: помеченные входные данные $\mathbf{x} \in D$, ядерная функция $\kappa(\mathbf{x}_i, \mathbf{x}_j)$.

Выход: множители α_i , определяющих нелинейную решающую границу для классификатора $\hat{y} = \text{sign}(w(\mathbf{x}_i))$.

$\alpha_i \leftarrow 0$ для $1 \leq i \leq |D|$; /*инициализация вектора множителей нулями*/
 $converged \leftarrow \text{false}$; /*флаг сходимости алгоритма и завершения главного цикла*/

while $converged = \text{false}$ **do**

 /*коррекция вектора множителей α_i */

$converged \leftarrow \text{true}$;

 /*требуется добавить защиту от зацикливания*/

for $i \leftarrow 1$ **in** $|D|$ **do**

if $y_i \cdot \left(\sum_{j=1}^{|D|} \alpha_j y_j \cdot \kappa(\mathbf{x}_i, \mathbf{x}_j) \right) \leq 0$

then

$\alpha_i \leftarrow \alpha_i + 1$;

$converged \leftarrow \text{false}$; /*меняли α_i , значит алгоритм еще не сошёлся*/

end

end

end

return α_i ;

Использованные информационные источники

1. Флах П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных / пер. с англ. А. А. Слинкина. – М.: ДМК Пресс, 2015. – 400 с.: ил.
2. Мэрфи К. П. Вероятностное машинное обучение: введение / пер. с англ. А. А. Слинкина. – М.: ДМК Пресс, 2022. – 990 с.: ил.
3. Мюллер А., Гвидо С. Введение в машинное обучение с помощью Python. Руководство для специалистов по работе с данными.: Пер. с англ. - СПб.: ООО "Альфа-книга", 2017. - 480 с.: ил.