Машинное обучение

Лекция 11:
Метод опорных векторов

Докладчик: Тима Бовт

Что рассмотрим сегодня?

- Постановка задачи опорных векторов
- Метод опорных векторов
- Вывод задачи оптимизации двумя способами
- Нелинейное обобщение

Постановка задачи опорных векторов

Постановка задачи

Пусть задана выборка $\mathbb{D} = (X|y)_{i=1}^{n}$, где $X \subseteq \mathbb{X} = \mathbb{R}^{n \times m}$, $y \subseteq \mathbb{Y} = \{-1, 1\}$, то есть

$$\mathbb{D} = \begin{pmatrix} x_{11} & \dots & x_{1m} & y_1 \\ \vdots & \ddots & \vdots & \vdots \\ x_{n1} & \dots & x_{nm} & y_n \end{pmatrix}.$$

Добавляем фиктивный единичный столбец к матрице X. Введем параметр $w = (w_0, w_1, \dots, w_m) \in \mathbb{W} = \mathbb{R}^{m+1}$ и определим скалярное произведение

$$(x, w) = \sum_{i=0}^{m} x_i w_i = w_0 + w_1 x_1 + \dots + w_m x_m.$$

Построим линейную модель для бинарной классификации

$$f(x; w) = \operatorname{sgn}\big((x, w) - w_0\big), \ w \in \mathbb{R}^m, w_0 \in \mathbb{R}.$$

Постановка задачи

Зададим эмпирический риск как количество ошибок классификации (соответственно функция потерь – ошибка классификации)

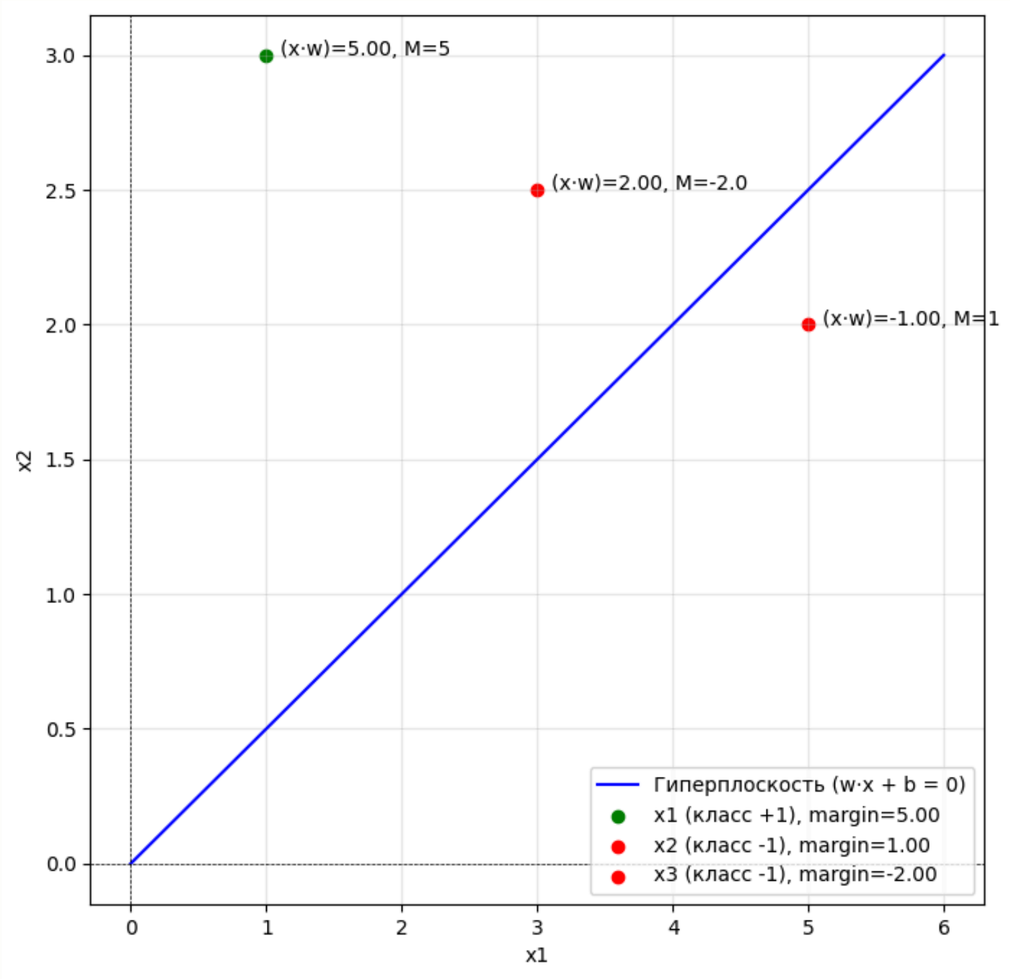
$$L(x; w) = \sum_{i=0}^{n} [f(x_i; w) \neq y_i].$$

Введем понятие отступа (margin) классификатора относительно объекта x_i – это проекция вектора, который указывает на объект x_i, на нормаль к плоскости

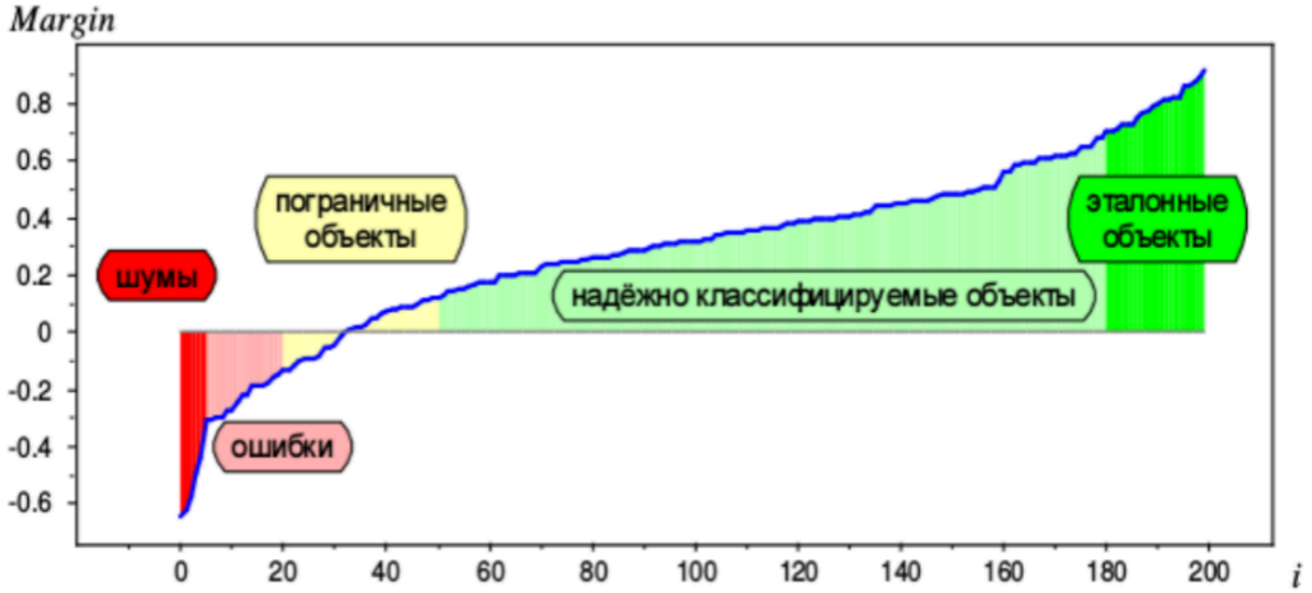
$$M_i(w, w_0) = \big((x, w) - w_0\big) \cdot y_i.$$

Отступ положителен $M_i(w, w_0) > 0$, когда класс угадан верно, и отрицателен $M_i(w, w_0) < 0$, когда класс угадан неверно.

Постановка задачи



Постановка задачи



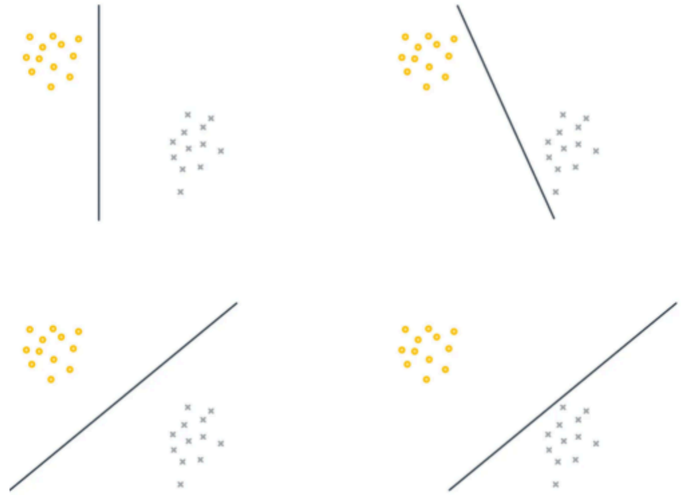
Тогда задачу оптимизации функции потерь можно сформулировать следующим образом:

$$L(x; w) = \sum_{i=0}^{n} [M_i(w, w_0) < 0] \to \min_{w, w_0}.$$

Эту функцию также можно называть **missclassification loss.**

Постановка задачи

Если мы попытаемся оптимизировать эту функцию градиентными методами (например, промажорируем ее какой-либо другой более гладкой функцией, так как в данном случае она кусочно-постоянная), то придем к тому, что, вообще говоря, решение этой задачи не единственно.



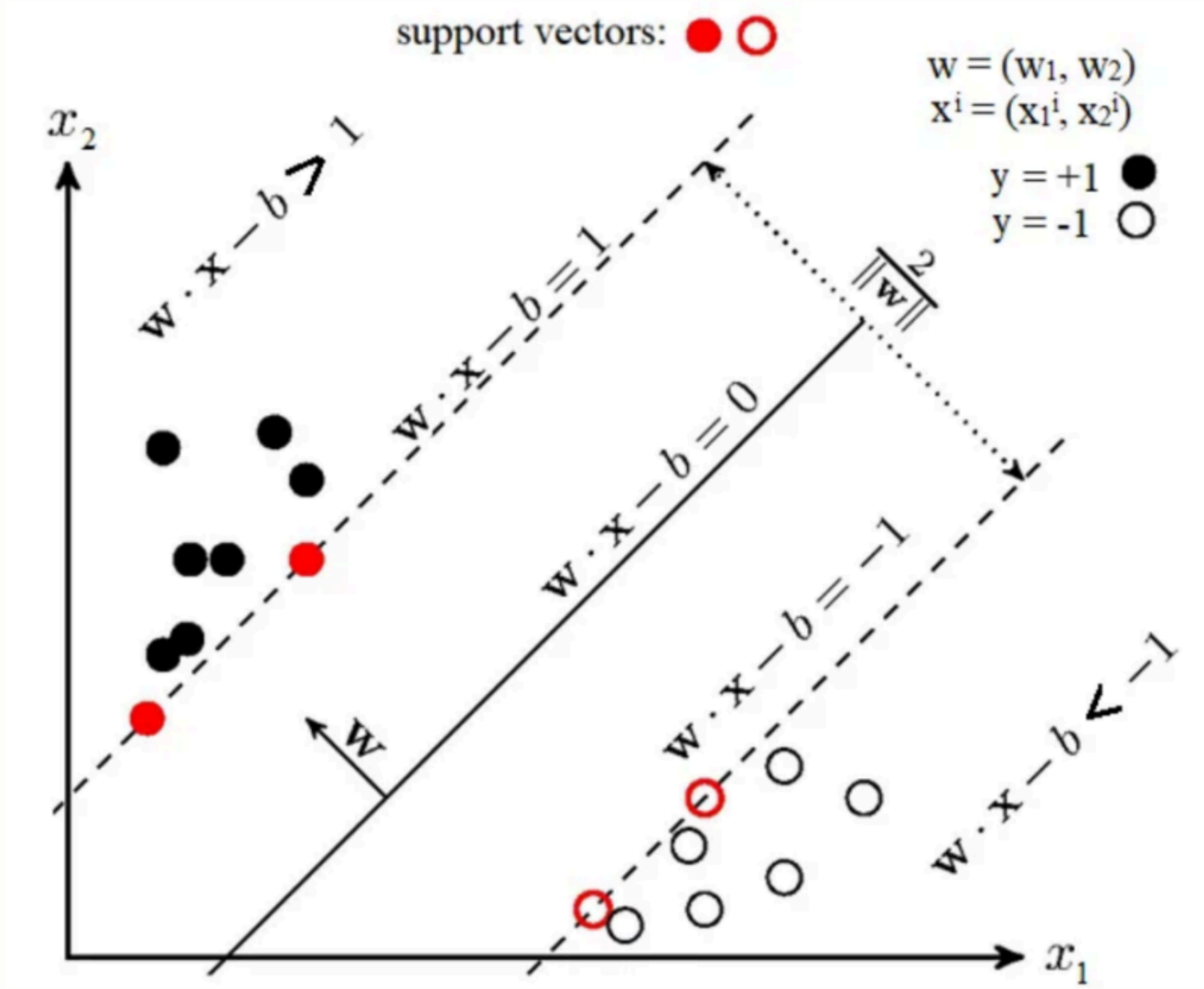
Важно учитывать тот факт, что не все из этих моделей будут устойчивыми, то есть, немного пошевелив эти прямые, мы можем случайно зацепить некоторые ненужные точки. Наилучшей же будет та, что находится в левом углу. Таким образом, для решения задачи нам нужно не только найти разделяющую прямую, но и постараться провести её на одинаковом удалении от обоих классов, то есть максимизировать минимальный отступ.

Метод опорных векторов

Понятие опорного вектора

Мы можем интерпретировать SVM как построение гиперплоскости, которая находится как можно дальше от точек обоих классов. Это достигается за счет максимизации отступа между гиперплоскостью и объектами классов, которые расположены ближе всего к ней. Такие объекты и называют опорными векторами. Отсюда и название алгоритма.

Понятие опорного вектора



Вывод задачи оптимизации

Выведем это аналитически. Зададим условие нормировки весов

$$\min_{i=1,\dots,n} M_i(w, w_0) = 1,$$

то есть это значит, что расстояние от гиперплоскости отступа до самого близкого к ней объекта равно единице. Тогда

$$\forall x_{+1} \quad (w, x_{+1}) - w_0 \geqslant 1,$$

$$\forall x_{-1} \quad (w, x_{-1}) - w_0 \leqslant -1.$$

Домножим второе неравенство на -1 и сложим с первым:

$$(w, x_{+1} - x_{-1}) \geqslant 2.$$

Вектор w — это вектор нормали к разделяющей гиперплоскости. Найдем проекцию вектора, концами которого являются опорные векторы разных классов, на вектор w. Для этого разделим обе части неравенства на $||w||$:

$$\frac{(w, x_{+1} - x_{-1})}{||w||} \geqslant \frac{2}{||w||}.$$

Вывод задачи оптимизации

$$\frac{(w, x_{+1} - x_{-1})}{||w||} \geqslant \frac{2}{||w||}.$$

То есть это есть расстояние по нормали между ближайшими x_{+1} и x_{-1}. Теперь если мы утверждаем, что для всех точек выполняются неравенства приведенные выше, то решая задачу оптимизации

$$\frac{2}{||w||} \to \max,$$

мы будем максимизировать отступ между классами. Таким образом, если выборка линейно разделима, то нам удовлетворяет такая гиперплоскость, которая доставляет максимум функционала $\frac{2}{||w||}$.

Задача оптимизации

В итоге, преобразовав задачу максимизации к задаче минимизации и объединив с первым условием, мы получаем в случае линейной разделимости выборки задачу оптимизации

$$\begin{cases} \frac{1}{2}||w||^2 \to \min\limits_{w,w_0}, \\ M_i(w,w_0) \geqslant 1, \ \ i = 1,\ldots,n. \end{cases}$$

($||w||^2$ для того, чтобы не тратить ресурсы на извлечение корня например в случае с евклидовой нормой).

Задача оптимизации

Ослабим условие на margin, введя набор переменных $\varepsilon_i \geqslant 0$, характеризующих величину ошибки на каждом объекте x_i:

$$M_i(w, w_0) \geqslant 1 - \varepsilon_i, \;\; i = 1, \ldots, n.$$

Добавив в минимизируемый функционал суммарный штраф за ошибки, получим новую задачу оптимизации с ограничениями:

$$\begin{cases} \dfrac{1}{2}||w||^2 + C\sum\limits_{i=1}^{n} \varepsilon_i \to \min\limits_{w, w_0, \xi}, \\ M_i(w, w_0) \geqslant 1 - \varepsilon_i, \;\; i = 1, \ldots, n, \\ \varepsilon_i \geqslant 0 \;\; i = 1, \ldots, n. \end{cases}$$

Эту задачу оптимизации можно упростить, заменив на эквивалентную безусловную задачу оптимизации:

$$\frac{1}{2}||w||^2 + C\sum_{i=1}^{n} \big(1 - M_i(w, w_0)\big)_+ \to \min_{w, w_0}.$$

Получение задачи оптимизации другим способом

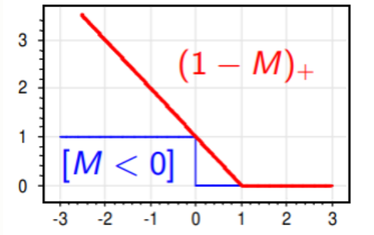
Альтернативный подход

Теперь вернемся к изначальной постановке задачи. Задача заключалась в минимизации функционала потерь:

$$L(x; w, w_0) = \sum_{i=0}^{n} [M_i(w, w_0) < 0] \to \min_{w, w_0}.$$

Подойдем к этой задаче немного с другой стороны. Из-за того, что эта функция является кусочно-постоянной (что делает невозможным оптимизацию этой функции градиентными методами), то функцию $\sum_{i=0}^{n} \mathbb{I}[M_i(w, w_0) < 0]$ мы можем аппроксимировать другой более гладкой функцией:

$$\sum_{i=0}^{n} [M_i(w, w_0) < 0] \leqslant \sum_{i=1}^{n} \big(1 - M_i(w, w_0)\big)_+ = \sum_{i=1}^{n} \max(0, 1 - M_i).$$



Альтернативный подход

То есть мы получим задачу оптимизации

$$\mathcal{L}(x; w, w_0) = \sum_{i=1}^{n} \big(1 - M_i(w, w_0)\big)_+ \to \min_{w, w_0}.$$

А теперь добавим к этой задаче член регуляризации и получим функционал потерь, который носит название Hinge Loss:

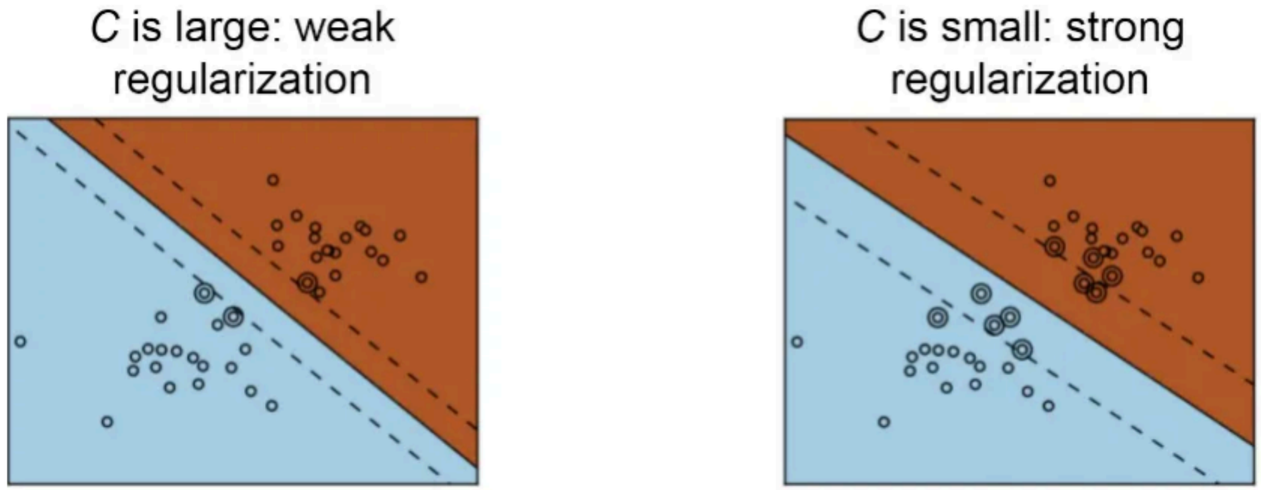
$$\mathcal{L}(x; w, w_0) = \sum_{i=1}^{n} \big(1 - M_i(w, w_0)\big)_+ + \frac{1}{2C}||w||^2 \to \min_{w, w_0},$$

что вернуло нас к тому результату, который мы вывели аналитически.

Таким образом, функционал потерь имеет структуру аппроксимация функции + член регуляризации.

Регуляризация и градиент

Визуализация того, как ведет себя гиперплоскость в зависимости от коэффициента регуляризации:



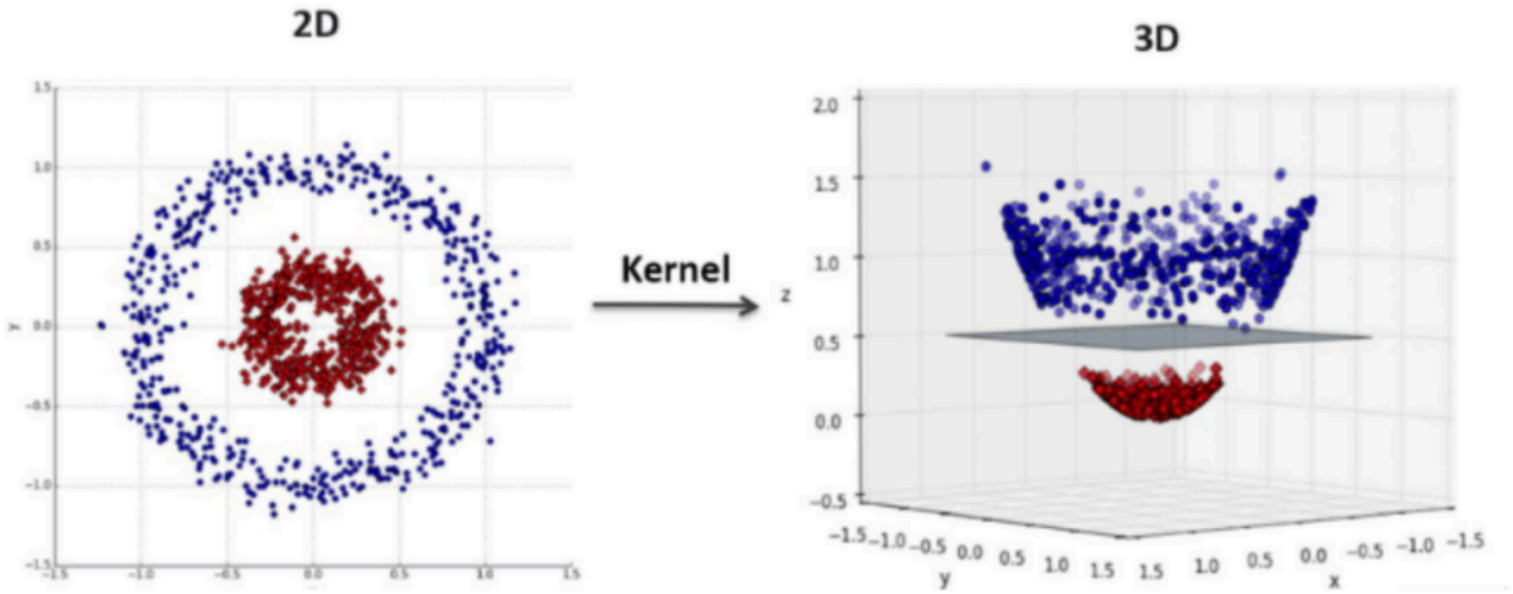
Получившийся функционал потерь мы и можем дальше оптимизировать градиентными методами. Запишем, чем равен градиент этого функционала:

$$\frac{\partial \mathcal{L}(x;w)}{\partial w} = \sum_{i=1}^{n} \begin{cases} 0, \ 1 - y_i(w_i, x) \leqslant 0, \\ -y_i x_i, \ 1 - y_i(w_i, x) > 0 \end{cases} + \frac{1}{C} w.$$

Нелинейное обобщение

Нелинейное обобщение

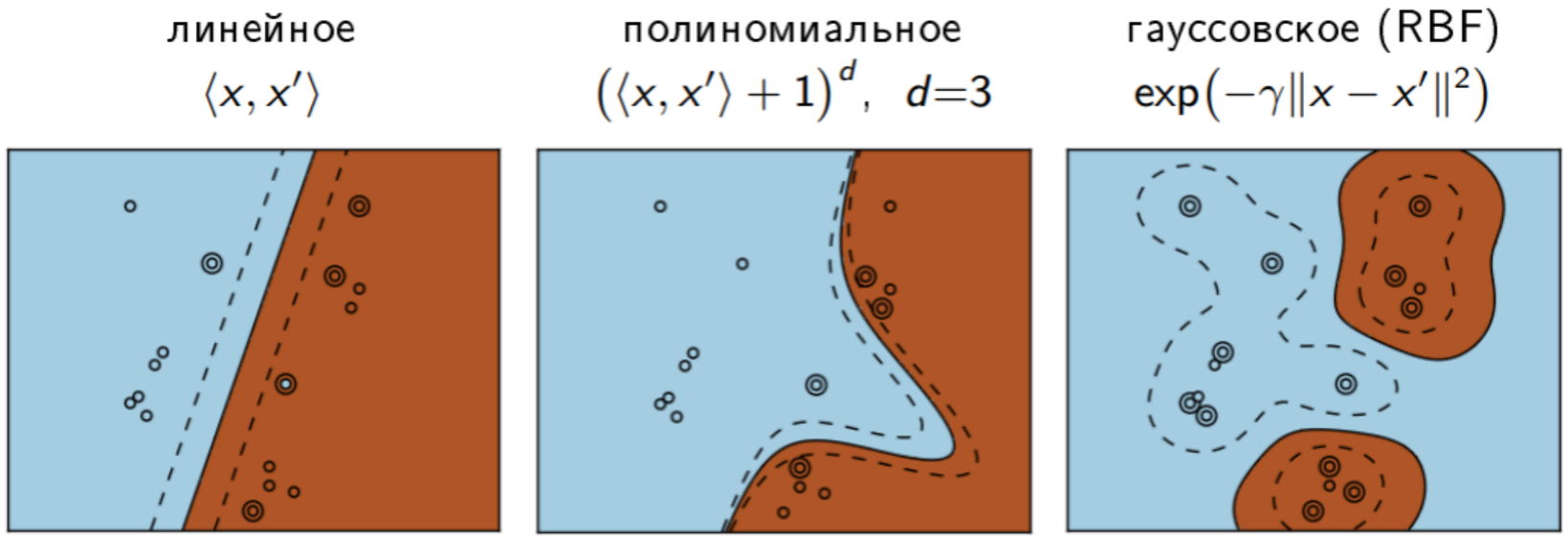
Идея состоит в переходе от исходного признакового пространства $\mathbb{X}$ в другое $\mathbb{H}$ где выборка может оказаться линейно разделимой, посредством преобразования $\psi : \mathbb{X} \to \mathbb{H}$.



При подсчете отступа мы считаем скалярное произведение. Причем мы можем изменять скалярное произведение в зависимости от функционального пространства, чтобы получить другие виды отступа.

Нелинейное обобщение

В гильбертовом пространстве скалярное произведение определяет норму, отсюда изменятся расстояния. Таким образом, мы неявно преобразовали признаковое пространство, сделав его нелинейным относительно исходных признаков.



Плюсы и минусы SVM

Рассмотрим плюсы и минусы классического SVM. Плюсы:

1. хорошо работает с пространством признаков большого размера;
2. хорошо работает с данными небольшого объема;
3. алгоритм максимизирует разделяющую полосу, которая, как подушка безопасности, позволяет уменьшить количество ошибок классификации;
4. так как алгоритм сводится к решению задачи квадратичного программирования в выпуклой области, то такая задача всегда имеет единственное решение (разделяющая гиперплоскость с определенными гиперпараметрами алгоритма всегда одна).

Минусы:

1. долгое время обучения (для больших наборов данных);
2. неустойчивость к шуму: выбросы в обучающих данных становятся опорными объектами-нарушителями и напрямую влияют на построение разделяющей гиперплоскости;
3. не описаны общие методы построения ядер и спрямляющих пространств, наиболее подходящих для конкретной задачи в случае линейной неразделимости классов. Подбирать полезные преобразования данных – искусство.