2022

Лекция 9. Метод опорных векторов

ПАПУЛИН С.Ю. (PAPULIN.STUDY@YANDEX.RU)

Дополнительные материалы доступны по ссылке: https://github.com/mlmethods

Содержание

1. Опорные векторы и зазор

1.1. Гиперплоскость

Пусть есть n точек в d размерном пространстве:

$$D = \{(x_i, y_i)\}_{i=1}^{n},$$

где

$$x_i = [x_{i1}\ x_{i2}\ \dots\ x_{id}]$$

и

$$y_i \in \{+1, -1\}$$

Гиперплоскость в d размерном пространстве представляется как множество всех точек $x \in \mathbb{R}^d$, которые удовлетворяют следующее условие:

$$h(x) = 0,$$

где $h(x)$ – функция гиперплоскости:

$$h(x) = w^T x + b = w_1 x_1 + w_2 x_2 + \cdots + w_d x_d + b,$$

где w – вектор весов размерности d; b – скалярное значение, смещение.

Для точек, лежащих на гиперплоскости, имеем

$$h(x) = w^T x + b = 0.$$

1.2. Разделяющая гиперплоскость

Гиперплоскость разбивает исходное d-размерное пространство на два полупространства.

Набор данных называется линейно-разделимым, если каждая полуплоскость содержит точки только одного класса.

Если входной набор данных линейно-разделим, то мы можем найти разделяющую гиперплоскость $h(\boldsymbol{x}) = 0$, такую что все точки $y_i = -1$ будут лежать $h(\boldsymbol{x}_i) < 0$ и все точки $y_i = +1$ будут находиться $h(\boldsymbol{x}_i) > 0$.

Функция гиперплоскости $h(\boldsymbol{x})$ служит как линейный классификатор или линейный дискриминант, который предсказывает класс y для любой заданной точки $\boldsymbol{x}$ в соответствии с правилом принятия решения:

$$y = \begin{cases} +1, \text{если } h(x) > 0 \\ -1, \text{если } h(x) < 0 \end{cases}$$

Пусть $\boldsymbol{a}_1$ и $\boldsymbol{a}_2$ две произвольные точки, которые лежат на гиперплоскости

$$h(a_1) = w^T a_1 + b = 0$$

$$h(a_2) = w^T a_2 + b = 0$$

Вычитаем одно выражение из другого, получаем

$$w^T(a_1 - a_2) = 0$$

Это означает, что вектор параметров $\boldsymbol{w}$ ортогонален гиперплоскости, потому что он ортогонален любому вектору $(\boldsymbol{a}_1 - \boldsymbol{a}_2)$ на гиперплоскости

$\boldsymbol{w}$ определяет направление нормальное к гиперплоскости, которое фиксирует ориентацию гиперплоскости

b фиксирует смещение гиперплоскости в d-размерном пространстве

Так как оба вектора $\boldsymbol{w}$ и $-\boldsymbol{w}$ нормальны к гиперплоскости, мы устраняем неопределенность за счет введения требования, что $h(\boldsymbol{x}_i) > 0$, когда $y_i = 1$, и $h(\boldsymbol{x}_i) < 0$ когда $y_i = -1$.

1.3. Расстояние от точки до гиперплоскости

Рассмотрим точку $\boldsymbol{x} \in R^d$, такие что x не лежит на гиперплоскости

Пусть $\boldsymbol{x}_p$ будет ортогональная проекция $\boldsymbol{x}$ на гиперплоскость и пусть $\boldsymbol{r} = \boldsymbol{x} - \boldsymbol{x}_p$. Тогда

$$\boldsymbol{x} = \boldsymbol{x_p} + \boldsymbol{r}$$

и

$$\boldsymbol{x} = \boldsymbol{x_p} + r\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}$$

где r – направленное расстояние до точки $\boldsymbol{x}$ из $\boldsymbol{x}_p$, т.е. r указывает на смещение $\boldsymbol{x}$ от $\boldsymbol{x}_p$ по направлению единичного вектора параметров $\frac{\mathbf{w}}{\|\mathbf{w}\|}$

Смещение r – положительно, если направления $\boldsymbol{r}$ и $\boldsymbol{w}$ совпадают, и отрицательно, если направлены противоположно

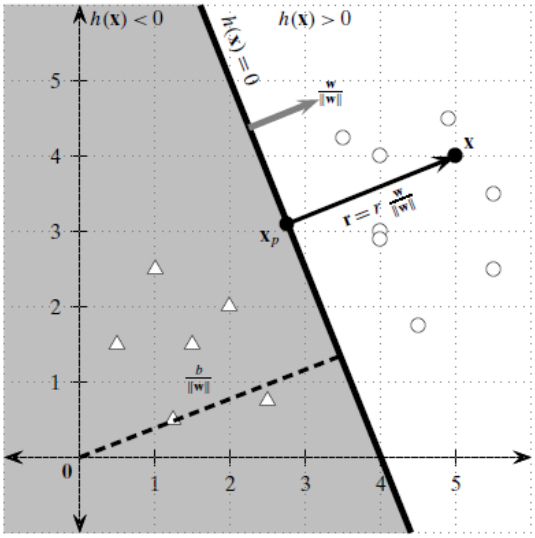


Рисунок – Гиперплоскость $h(x) = 0$ и проекция точки $\boldsymbol{x}$ на гиперплоскость [1]

Как получить расстояние между точкой $\boldsymbol{x}$ и гиперплоскостью?

Запишем уравнение гиперплоскости в следующем виде для заданной точки $\boldsymbol{x}$

$$h(\boldsymbol{x}) = h\left(\boldsymbol{x_p} + r\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}\right) = \boldsymbol{w}^T\left(\boldsymbol{x_p} + r\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}\right) + b = \underbrace{\boldsymbol{w}^T\boldsymbol{x_p} + b}_{h(\boldsymbol{x_p})} + r\frac{\boldsymbol{w}^T\boldsymbol{w}}{\|\boldsymbol{w}\|} = \underbrace{h(\boldsymbol{x_p})}_{0} + r\|\boldsymbol{w}\| = r\|\boldsymbol{w}\|$$

Тогда

$$r = \frac{h(\boldsymbol{x})}{\|\boldsymbol{w}\|}$$

Чтобы получить расстояние, которое должно быть неотрицательным, мы можем умножить r на метку класс y

$$\delta = y \cdot r = \frac{y \cdot h(\boldsymbol{x})}{\|\boldsymbol{w}\|}$$

Для точки начала координат $\boldsymbol{x} = \boldsymbol{0}$, направленное расстояние есть

$$r = \frac{h(\boldsymbol{0})}{\|\boldsymbol{w}\|} = \frac{\boldsymbol{w}^T\boldsymbol{0} + b}{\|\boldsymbol{w}\|} = \frac{b}{\|\boldsymbol{w}\|}$$

1.4. Зазор и опорные векторы гиперплоскости

Для каждой точки $\boldsymbol{x}_i$ мы можем найти её расстояние до гиперплоскости, как

$$\delta_i = \frac{y_i h(\boldsymbol{x_i})}{\|\boldsymbol{w}\|} = \frac{y_i(\boldsymbol{w}^T\boldsymbol{x_i} + b)}{\|\boldsymbol{w}\|}$$

Среди всех n точек мы определяем зазор линейного классификатора как минимальное расстояние от точки до разделяющей гиперплоскости, т.е.

$$\delta^* = \min_{\boldsymbol{x}_i}\left\{\frac{y_i(\boldsymbol{w}^T\boldsymbol{x_i} + b)}{\|\boldsymbol{w}\|}\right\}$$

Заметим, что $\delta^* \neq 0$, так как полагаем, что $h(\boldsymbol{x})$ разделяющая гиперплоскость.

Все точки (или векторы), которые соответствуют минимальному расстоянию, называются опорные векторы для гиперплоскости.

Другими словами, опорный вектор $\boldsymbol{x}^*$ есть точка, лежащая точно на зазоре классификатора, и таким образом удовлетворяет условие

$$\delta^* = \frac{y^*(\boldsymbol{w}^T\boldsymbol{x}^* + b)}{\|\boldsymbol{w}\|}$$

Числитель $y^*(\boldsymbol{w}^T\boldsymbol{x}^* + b)$ определяет абсолютное расстояние опорного вектора до гиперплоскости (функциональный зазор)

1.5. Уникальная гиперплоскость

Умножая обе части выражения гиперплоскости на некоторое скалярное значение s, получаем эквивалентную гиперплоскость:

$$s \cdot h(\boldsymbol{x}) = s \cdot \boldsymbol{w}^T \boldsymbol{x} + s \cdot b = (s \cdot \boldsymbol{w})^T \boldsymbol{x} + s \cdot b = 0$$

Чтобы получить уникальную (или каноническую) гиперплоскость, выберем такое значение s, при котором абсолютное расстояние опорного вектора до гиперплоскости будет равно 1:

$$s \cdot y^*(\boldsymbol{w}^T \boldsymbol{x}^* + b) = 1$$

Далее полагаем, что наша разделяющая гиперплоскость является канонической, т.е. полагаем, что выражение гиперплоскости масштабировано так, чтобы $y^* h(\boldsymbol{x}^*)$ = 1 для опорного вектора $\boldsymbol{x}^*$

Тогда зазор можно представить как

$$\delta^* = \frac{y^* h(\boldsymbol{x}^*)}{\|\boldsymbol{w}\|} = \frac{1}{\|\boldsymbol{w}\|}$$

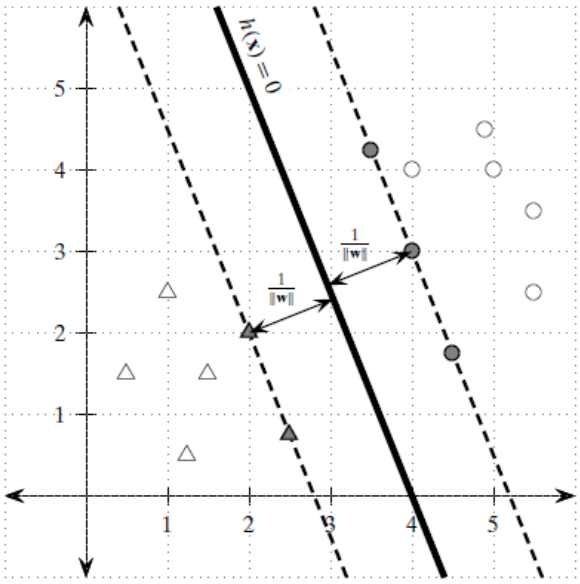


Рисунок – Зазор и опорные векторы при линейно разделимом случае [1]

В данном случае для каждого опорного вектора $\boldsymbol{x}_i^*$ с меткой y_i^* получаем

$$y_i^* h(\boldsymbol{x}_i^*) = 1$$

а для любой точки, которая не является опорным вектором

$$y_i^* h(\boldsymbol{x}_i^*) > 1$$

Для всех n точек из набор $\boldsymbol{D}$ получаем следующее неравенство

$$y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geq 1 \text{ для всех } \boldsymbol{x}_i \in D$$

2. Метод опорных векторов для линейно разделимой выборки

2.1. Максимизация зазора

Если предположить, что точки набора данных линейно разделимы, то существует гиперплоскость, которая идеально классифицирует каждую точку. В данном случае существует бесконечное количество разделяющих гиперплоскостей. Какую необходимо выбрать?

Фундаментальная идея, состоящая в основе метода опорных векторов, заключается в том, чтобы выбрать каноническую гиперплоскость, определяемую вектором параметров $\boldsymbol{w}$ и смещением b, которая дает максимальные зазор среди всех возможных гиперплоскостей.

Если δ_h^* представляет зазор для гиперплоскости $h(\mathbf{x}) = 0$, то целью является поиск оптимальной гиперплоскости h^*:

$$h^* = \underset{h}{\operatorname{argmax}}\{\delta_h^*\} = \underset{\boldsymbol{w},b}{\operatorname{argmax}}\left[\frac{1}{\|\boldsymbol{w}\|}\right]$$

при условии

$$y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 \text{ для всех } \boldsymbol{x}_i \in D$$

2.2. Прямая задача (primal problem)

Заметим, что вместо максимизации зазора $\frac{1}{\|\mathbf{w}\|}$, мы можем минимизировать $\|\mathbf{w}\|$. Тогда получим

$$\min_{\boldsymbol{w},b}\left[\frac{||\boldsymbol{w}||^2}{2}\right]$$

при условии

$$y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1, \forall \boldsymbol{x}_i \in D$$

Мы можем напрямую решить данную задачу по минимизации выпуклой функции с n линейными ограничениями, используя стандартные алгоритмы оптимизации.

2.3. Двойственная задача (dual problem)

Функция Лагранжа

Функция Лагранжа применяется, когда необходимо найти максимум или минимум целевой функции $f(x)$ с ограничением $h(x) = 0$.

В этом случае для оптимального решения будет справедливо равенство

$$\nabla f(x^*) = \lambda^* \nabla h(x^*),$$

где λ^* – некоторая константа, называемая множителем Лагранжа.

Замечание: предыдущее выражение иногда записывают со знаком минус, т.е.

$$\nabla f(x^*) = -\lambda^* \nabla h(x^*)$$

Таким образом, функция Лагранжа с одним ограничением примет вид

$$L(x, \lambda) = f(x) - \lambda h(x)$$

Для поиска оптимальных значений решается система уравнений:

$$\nabla L(x, \lambda) = \begin{bmatrix} \frac{\partial L(x,\lambda)}{\partial x_1} \\ \frac{\partial L(x,\lambda)}{\partial x_2} \\ \vdots \\ \frac{\partial L(x,\lambda)}{\partial \lambda} \end{bmatrix} = \mathbf{0}$$

Если имеются p ограничений, то

$$L(x, \lambda) = f(x) - \sum_{i=1}^{p} \lambda_i h_i(x)$$

Прямая задача

Задача оптимизации в общей форме записывается как

$$\min_{\boldsymbol{x}} f(x)$$

при условии

$$f_i(x) \leq 0 \text{ для } i = 1, \dots, m$$

и

$$h_i(x) = 0 \text{ для } i = 1, \dots, p$$

Оптимальное решение представим как p^*

Данную задачу можно представить в виде функции Лагранжа

$$L(x, \lambda, \nu) = f(x) - \sum_{i=1}^{m} \lambda_i f_i(x) - \sum_{i=1}^{p} \nu_i h_i(x)$$

где λ_i и ν_i – множители Лагранжа

Заметим, что в такой записи x неограничен.

Двойственная задача

Двойственная функция Лагранжа:

$$g(\lambda, \nu) = \inf_{x \in D} L(x, \lambda, \nu) = \inf_{x \in D} \left[f(x) - \sum_{i=1}^{m} \lambda_i f_i(x) - \sum_{i=1}^{p} \nu_i h_i(x) \right],$$

где $g(\cdot,\cdot)$ – вогнутая функция

Двойственная задача примет вид:

$$\max_{\lambda, \nu} g(\lambda, \nu)$$

при условии

$$\lambda \geq 0$$

Оптимальное решение обозначим как d^*

Слабая двойственность:

$$p^* \geq d^*$$

Строгая двойственность:

$$p^* = d^*$$

Если $f(x)$ является выпуклой квадратичной функцией и функции f_i и h_i – аффинные функции, то

$$p^* = d^*$$

Условие Слейтера – достаточное условие для строгой двойственности при выпуклой оптимизации

Аффинная функция есть $\boldsymbol{y} = A\boldsymbol{x} + \boldsymbol{c}$

Двойственную задачу, как правило, решают через метод множителей Лагранжа. Основная идея заключается в том, что вводятся множители α_i для каждого ограничения, которые удовлетворяют условия Каруша-Куна-Таккера (ККТ) оптимального решения:

$$\alpha_i(y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) - 1) = 0$$

и

$$\alpha_i \geq 0$$

Новая целевая функция называется функцией Лагранжа (Lagrangian), которая имеет следующий вид

$$L = \frac{1}{2} ||\boldsymbol{w}||^2 - \sum_{i=1}^{n} \alpha_i(y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) - 1)$$

L должна быть минимизирована по $\boldsymbol{w}$ и b, и максимизирована по отношению к α_i

$$\max_{\alpha_i;\, \alpha_i \geq 0} \left[\min_{w,\, b} L(w, b, \alpha_i) \right]$$

Задача минимизации

Возьмем частные производные по параметрам и смещению:

- по параметрам $\boldsymbol{w}$:

$$\frac{\partial}{\partial \boldsymbol{w}} L = \boldsymbol{w} - \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i = 0$$

или

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

- по смещению b:

$$\frac{\partial}{\partial b} L = \sum_{i=1}^{n} \alpha_i y_i = 0$$

Таким образом, вектор параметров может быть представлен как линейная комбинация точек набора данных с множителем Лагранжа (с учетом знака метки класс) в качестве коэффициентов.

Сумма же множителей Лагранжа с учетом знака метки класса должна быть равна нулю

Задача максимизации

Подставляя ранее полученные выражения при минимизации по параметрам и смещению в общее выражение, получим двойственную целевую функцию:

$$L_{dual} = \frac{1}{2}\left|\left|\boldsymbol{w}\right|\right|^2 - \sum_{i=1}^{n} \alpha_i (y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) - 1) =$$

$$= \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} - \boldsymbol{w}^T \underbrace{\left(\sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i\right)}_{\boldsymbol{w}} - b \underbrace{\sum_{i=1}^{n} \alpha_i y_i}_{0} + \sum_{i=1}^{n} \alpha_i =$$

$$= -\frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \sum_{i=1}^{n} \alpha_i =$$

$$= \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j$$

Тогда задачу оптимизации можно записать как

$$\max_{\alpha} L_{dual} = \max_{\alpha} \left[\sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j\right]$$

при условии

$$\alpha_i \geq 0, \forall i \in D \text{ и } \sum_{i=1}^{n} \alpha_i y_i = 0$$

L_{dual} есть выпуклая квадратическая задача оптимизации (по α_i α_j), которая может быть решена с использованием стандартных техник оптимизации.

Вектор весов и смещение

После нахождения значений α_i мы можем рассчитать параметры $\boldsymbol{w}$ и смещение b. Заметим, что в соответствии с ККТ условиями мы имеем

$$\alpha_i(y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1) = 0$$

Рассмотрим два случая:

$$\alpha_i = 0$$

и

$$y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1 = 0$$

- Если $a_i > 0$, то $y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) = 1$ и тогда точка $\boldsymbol{x}_i$ должна быть опорным вектором;
- С другой стороны, если $y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) > 1$, то $a_i = 0$, т.е. если точка не является опорным вектором, тогда $\alpha_i = 0$.

Так как мы знаем α_i для всех точек, мы можем вычислить параметры выполнив суммирование только по опорным векторам:

$$\boldsymbol{w} = \sum_{i,\alpha_i>0} \alpha_i y_i \boldsymbol{x}_i$$

Для вычисления смещения b сначала вычислим b_i для каждого опорного вектора:

$$b_i = \frac{1}{y_i} - \boldsymbol{w}^T\boldsymbol{x}_i = y_i - \boldsymbol{w}^T\boldsymbol{x}_i$$

(следует из $\alpha_i(y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1) = 0$ и $y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) = 1$)

Чтобы получить b, усредним значения по всем b_i, полученные на предыдущем шаге.

$$b = \operatorname*{avg}_{0<\alpha_i}\{b_i\}$$

Классификатор

Для любой точки $\boldsymbol{x}_*$ предсказание класса будет определяться как

$$\hat{y} = \operatorname{sign}\big(h(\boldsymbol{x}_*)\big) = \operatorname{sign}(\boldsymbol{w}^T\boldsymbol{x}_* + b),$$

где $\operatorname{sign}(\cdot)$ – функция возвращает $+1$, если аргумент положительный, и -1, если отрицательный.

3. Метод опорных векторов для линейно неразделимой выборки

3.1. Мягкий зазор

Рассмотрим случай, когда точки разных классов не могут быть идеально разделимы.

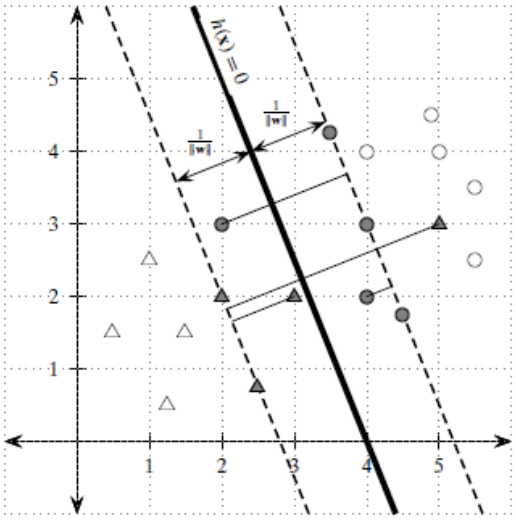


Рисунок – Мягкий зазор и опорные векторы при линейно неразделимом случае [1]

Метод опорных векторов может быть использован для неразделимых точек за счет введения переменных мягкого зазора ξ_i (slack variables). Таким образом, ранее введенное ограничение примет следующий вид:

$$y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geq 1 - \xi_i,$$

где $\xi_i \geq 0$ – переменная мягкого зазора для точки $\boldsymbol{x}_i$, которая показывает, как сильно точка может нарушить условие разделения, т.е. точка теперь может быть на расстоянии менее чем $\frac{1}{\|\mathbf{w}\|}$ от гиперплоскости.

Переменные ξ_i указывают на три типа точек:

- Если $\xi_i = 0$, то соответствующая точка $\boldsymbol{x}_i$ по крайней мере на расстоянии $\frac{1}{\|\mathbf{w}\|}$ от гиперплоскости
- Если $0 < \xi_i < 1$, то точка в пределах зазора и всё ещё правильно классифицирована, т.е. на правильной стороне гиперплоскости
- Если $\xi_i \geq 1$, то точка неправильно классифицирована и находится не на правильной стороне гиперплоскости

В случае с неразделимым пространством целью метода опорных векторов заключается в том, чтобы найти гиперплоскость с максимальным зазором, которая также минимизирует значения переменных ξ_i. Целевая функция примет следующий вид

$$\min_{\boldsymbol{w},b,\xi_i}\left[\frac{||\boldsymbol{w}||^2}{2}+C\sum_{i=1}^{n}(\xi_i)^k\right]$$

при условии

$$y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)\geq 1-\xi_i,\forall \boldsymbol{x}_i\in D,$$

$$\xi_i\geq 0\ \forall \boldsymbol{x}_i\in D,$$

где C и k – константы.

Величина $\sum_{i=1}^{n}(\xi_i)^k$ определяет потерю, т.е. оценку отклонения от случая, когда множество точек разделимы.

Скалярное значение C, которая выбирается эмпирически, есть константа регуляризации, которая контролирует компромисс между максимизацией зазора и минимизацией потерь.

- Если $C\to 0$, то компонент потери исчезает и целевая функция будет только максимизировать зазор
- Если $C\to\infty$, то зазор уже не имеет существенного значения, и целевая функция будет пытаться минимизировать потери.

Константа k управляет формой потерь

- Как правило, k устанавливается равным 1 (кусочно-линейная функция потерь – hinge loss) или 2 (квадратичная функция потерь)
- Когда $k=1$, целью является минимизации суммы ξ_i; когда $k=2$, то суммы квадратов ξ_i

3.2. Прямая задача

Задача оптимизации имеет следующий вид

$$\min_{\boldsymbol{w},b,\xi_i}\left[\frac{||\boldsymbol{w}||^2}{2}+C\sum_{i=1}^{n}(\xi_i)^k\right]$$

при условии

$$y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)\geq 1-\xi_i,\forall \boldsymbol{x}_i\in D,$$

$$\xi_i\geq 0\ \forall \boldsymbol{x}_i\in D,$$

Рассмотрим случай $k=1$. Вводимые условия можно записать в виде кусочно-линейной функции

$$l_{\text{hinge}}(\boldsymbol{x}_i,y,b,\boldsymbol{w})=\max\big(0,1-y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)\big)$$

или для всего набора данных

$$L_{\text{hinge}}=\sum_{i=1}^{n}\max\big(0,1-y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)\big)$$

Доказательство

Зафиксируем значения $\boldsymbol{w}$ **и** b и рассмотрим минимизацию по $\boldsymbol{\xi}$**.** Возьмем некоторое i. ξ_i должна быть неотрицательной. Лучшим вариантом является, когда ξ_i равна нулю, если $y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1$, иначе $1 - y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b)$. Таким образом, $\xi_i = \max\big(0, 1 - y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b)\big)$

Тогда задача оптимизации приобретает следующую форму

$$\min_{\boldsymbol{w},b}\left[\frac{\big|\big|\boldsymbol{w}\big|\big|^2}{2} + CL_{\text{hinge}}\right] = \min_{\boldsymbol{w},b}\left[\frac{\big|\big|\boldsymbol{w}\big|\big|^2}{2} + C\sum_{i=1}^{n}\max\big(0, 1 - y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b)\big)\right]$$

Логистическая регрессия и метод опорных векторов

Общая форма записи задачи безусловной оптимизации с регуляризацией L2

$$\min_{\boldsymbol{w},b}\left[L + \lambda\big|\big|\boldsymbol{w}\big|\big|^2\right]$$

или

$$\min_{\boldsymbol{w},b}\left[CL + \frac{\big|\big|\boldsymbol{w}\big|\big|^2}{2}\right]$$

При использовании кросс-энтропии в качестве L получаем выражение для логистической регрессии с L2 регуляризацией:

$$\min_{\boldsymbol{w},b}\left[L_{CE} + \lambda\big|\big|\boldsymbol{w}\big|\big|^2\right] = \min_{\boldsymbol{w},b}\left[\sum_{i=1}^{n} y_i \log h_{\theta,i} + (1 - y_i)\log\big(1 - h_{\theta,i}\big) + \lambda\big|\big|\boldsymbol{w}\big|\big|^2\right]$$

или

$$\min_{\boldsymbol{w},b}\left[CL_{CE} + \frac{\big|\big|\boldsymbol{w}\big|\big|^2}{2}\right] = \min_{\boldsymbol{w},b}\left[C\sum_{i=1}^{n} y_i \log h_{\theta,i} + (1 - y_i)\log\big(1 - h_{\theta,i}\big) + \frac{\big|\big|\boldsymbol{w}\big|\big|^2}{2}\right]$$

Если использовать кусочно-линейную функцию потерь (hinge), то выражение примет вид

$$\min_{\boldsymbol{w},b}\left[L_{\text{hinge}} + \lambda\big|\big|\boldsymbol{w}\big|\big|^2\right] = \min_{\boldsymbol{w},b}\left[\sum_{i=1}^{n}\max\big(0, 1 - y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b)\big) + \lambda\big|\big|\boldsymbol{w}\big|\big|^2\right]$$

или

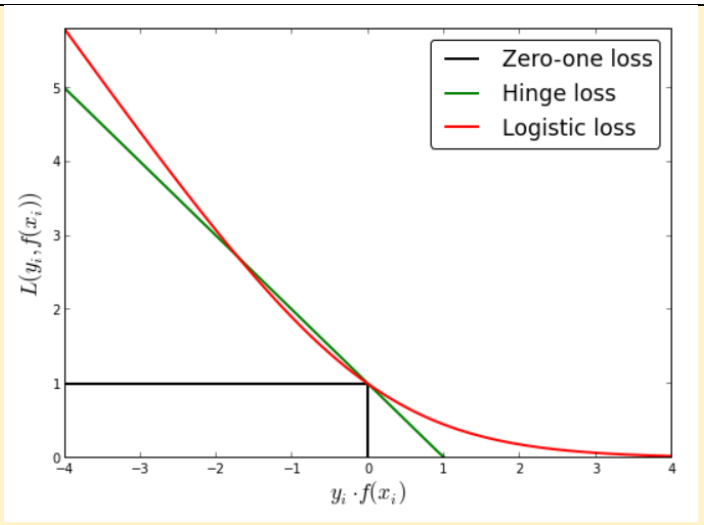
$$\min_{\boldsymbol{w},b}\left[CL_{\text{hinge}} + \frac{\big|\big|\boldsymbol{w}\big|\big|^2}{2}\right] = \min_{\boldsymbol{w},b}\left[C\sum_{i=1}^{n}\max\big(0, 1 - y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b)\big) + \frac{\big|\big|\boldsymbol{w}\big|\big|^2}{2}\right]$$

что соответствует методу опорных векторов.

Также возможен вариант

$$\min_{\boldsymbol{w},b}\left[\frac{1}{n}\sum_{i=1}^{n}\max\big(0, 1 - y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b)\big) + \lambda\big|\big|\boldsymbol{w}\big|\big|^2\right]$$

Сравнение функций потерь кросс-энтропии и кусочно-линейной



Такую задачу можно решить, например, посредством стохастического градиентного спуска и др.

3.3. Двойственная задача

3.3.1. Кусочно-линейная функция потерь (hinge loss)

Задача оптимизации

Если взять $k = 1$, то задача оптимизации примет вид

$$\min_{\boldsymbol{w},b,\xi_i} \left[\frac{||\boldsymbol{w}||^2}{2} + C \sum_{i=1}^{n} \xi_i \right]$$

при условии

$$y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geq 1 - \xi_i, \forall \boldsymbol{x}_i \in D,$$

$$\xi_i \geq 0 \; \forall \boldsymbol{x}_i \in D.$$

Данную задачу можно решить за счет введения множителей Лагранжа α_i и β_i, которые удовлетворяют следующие условия ККТ:

$$\alpha_i(y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) - 1 + \xi_i) = 0 \text{ при } \alpha_i \geq 0$$

и

$$\beta_i(\xi_i - 0) = 0 \text{ при } \beta_i \geq 0$$

Функция Лагранжа примет вид

$$L = \frac{1}{2}||\boldsymbol{w}||^2 + C \sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i(y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) - 1 + \xi_i) - \sum_{i=1}^{n} \beta_i \xi_i$$

Рассмотрим двойственную задачу. Для этого вычислим частные производные по $\boldsymbol{w}, b$ и ξ_i и приравняем их к нулю:

$$\frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i = 0 \text{ или } \boldsymbol{w} = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\frac{\partial L}{\partial \xi_i} = C - \alpha_i - \beta_i = 0 \text{ или } \beta_i = C - \alpha_i$$

Функция Лагранжа для двойственной задачи примет вид:

$$L_{dual} = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} - \boldsymbol{w}^T \underbrace{\left(\sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i\right)}_{\boldsymbol{w}} - b \underbrace{\sum_{i=1}^{n} \alpha_i y_i}_{0} + \sum_{i=1}^{n} a_i + \sum_{i=1}^{n} \underbrace{(C - \alpha_i - \beta_i)}_{0} \xi_i =$$

$$= \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j$$

Определим задачу оптимизации

$$\max_{\alpha} L_{dual} = \max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j$$

при условии

$$0 \le \alpha_i \le C, \forall i \in D \text{ и } \sum_{i=1}^{n} \alpha_i y_i = 0$$

Заметим, что целевая функция такая же, как и в линейно-разделимом случае. Однако ограничения по α_i отличаются, потому что появилось требования, что $\alpha_i + \beta_i = C$, $\alpha_i \ge 0$ и $\beta_i \ge 0$, что в свою очередь приводит к $0 \le \alpha_i \le C$.

Данная задача может быть решена, например, посредством стохастического градиентного подъема.

Вектор параметров (весов)

Как и ранее $\alpha_i = 0$ для всех точек, которые не являются опорными векторами, и $\alpha_i > 0$ для опорных векторов. В последнем случае точки удовлетворяют равенство

$$y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) = 1 - \xi_i$$

Заметим, что опорные векторы включают все точки на зазоре ($\xi_i = 0$) так же, как и все точки с положительным ξ_i.

Вектор параметров (весов):

$$\boldsymbol{w} = \sum_{i, a_i > 0} \alpha_i y_i \boldsymbol{x}_i$$

Смещение

Рассчитаем для β_i:

$$\beta_i = C - \alpha_i$$

Заменим β_i в ККТ условиях на выражение выше

$$(C - \alpha_i)\xi_i = 0$$

Таким образом, для опорных векторов с $\alpha_i > 0$ существует два варианта:

- $\xi_i > 0$ откуда следует, что $C - \alpha_i = 0$ или $\alpha_i = C$
- $C - \alpha_i > 0$, т.е. $\alpha_i < C$. В этом случае $\xi_i = 0$ (опорные векторы на зазоре)

Использую опорные векторы на зазоре ($0 < \alpha_i < C$ и $\xi_i = 0$), можно определить смещение b_i. Так одно из условий выглядит как

$$a_i(y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b_i) - 1) = 0.$$

Если $\boldsymbol{x}_i$ является опорным вектором на зазоре, то

$$y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b_i) = 1.$$

Смещение в этом случае для i-ой точки есть

$$b_i = \frac{1}{y_i} - \boldsymbol{w}^T\boldsymbol{x}_i = y_i - \boldsymbol{w}^T\boldsymbol{x}_i$$

Общее смещение можно вычислить усреднением

$$b = \underset{0<\alpha_i<C}{\text{avg}} \{b_i\}$$

Предсказание

После определения оптимальной гиперплоскости предсказание для новой точки $\boldsymbol{x}_*$ есть

$$\hat{y} = \text{sign}\big(h(\boldsymbol{x}_*)\big) = \text{sign}(\boldsymbol{w}^T\boldsymbol{x}_* + b)$$

3.3.2. Квадратичная функция потерь (quadratic loss)

При использовании квадратичной формы записи ($k = 2$) задача оптимизации имеет вид

$$\min_{\boldsymbol{w},b,\xi_i} \left[\frac{\big||\boldsymbol{w}|\big|^2}{2} + C\sum_{i=1}^{n} {\xi_i}^2 \right]$$

при условии

$$y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 - \xi_i, \forall \boldsymbol{x}_i \in D.$$

Отметим, что отсутствует условие $\xi_i \geq 0\ \forall \boldsymbol{x}_i \in D$.

Функция Лагранжа примет вид

$$L = \frac{1}{2}||\boldsymbol{w}||^2 + C\sum_{i=1}^{n}\xi_i{}^2 - \sum_{i=1}^{n}\alpha_i(y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1 + \xi_i)$$

Дальнейшее решение аналогично ранее рассмотренному:

- Вычисляем частные производные по $\boldsymbol{w}, b$ и ξ_i и приравниваем их к нулю
- Формулируем двойственную задачу по отношению к α_i
- Максимизируем функцию Лагранжа для двойственной задачи и определяем α_i
- Вычисляем $\boldsymbol{w}$ и b

4. Нелинейный случай

Линейный метод опорных вектором может быть использован для набора данных с нелинейной границей принятия решения посредством трюка с ядром (kernel trick). Идея заключается в том, чтобы построить отображение точек $\boldsymbol{x}_i$ *исходного* d*-размерного пространства* в точки $\varphi(\boldsymbol{x}_i)$ многомерного *пространства признаков* посредством нелинейного преобразования φ. Обеспечивая дополнительную гибкость за счет большего количества признаков, более вероятно, что точки $\varphi(\boldsymbol{x}_i)$ будут линейно разделимы в пространстве признаков. Следуем отметить, что линейная поверхность принятия решения в действительности соответствует нелинейной поверхности принятия решения в исходном пространстве. Более того, трюк с ядром позволяет выполнять все операции посредством функции ядра, вычисляемой в исходном пространстве, нежели выполнять отображение точек в пространство признаков.

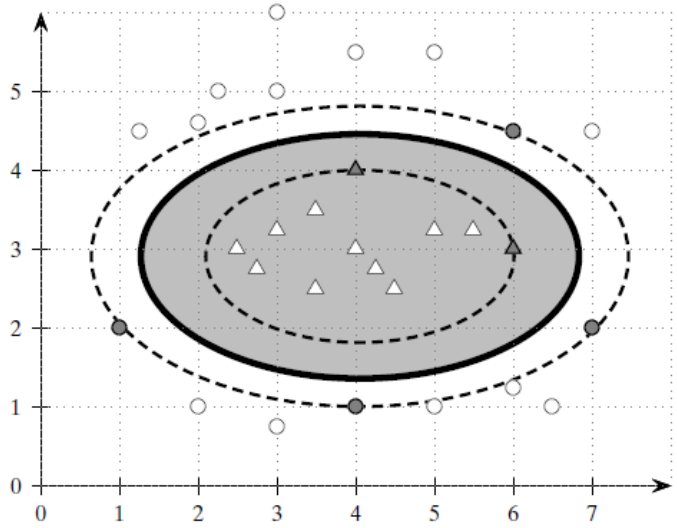


Рисунок – Пример с нелинейной границей принятия решения [1]

Функции ядра

Функция ядра должна удовлетворять следующее условие

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \varphi(\boldsymbol{x}_i)^T\varphi(\boldsymbol{x}_j)$$

где $\varphi(\cdot)$ – функция отображения точки исходного пространства в точку пространства признаков

Полиномиальное ядро:

- однородное

$$K_q(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(\boldsymbol{x}_i^T \boldsymbol{x}_j\right)^q$$

- неоднородное

$$K_q(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(1 + \boldsymbol{x}_i^T \boldsymbol{x}_j\right)^q$$

Ядро Гаусса:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left[-\frac{\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|}{2\sigma^2}\right]$$

Пример

Пример с полиномиальным неоднородным ядром второй степени
Исходные векторы:

$$\boldsymbol{x}_i = [x_{i1}\ x_{i2}]^T$$
$$\boldsymbol{x}_j = \left[x_{j1}\ x_{j2}\right]^T$$

Преобразование исходного вектора в пространства полиномиальных признаков

$$\varphi(\boldsymbol{x}_i) = \left[1\ \sqrt{2}\,x_{i1}\ \sqrt{2}\,x_{i2}\ \sqrt{2}\,x_{i1}x_{i2}\ x_{i1}^2\ x_{i2}^2\right]$$
$$\varphi(\boldsymbol{x}_j) = \left[1\ \sqrt{2}\,x_{j1}\ \sqrt{2}\,x_{j2}\ \sqrt{2}\,x_{j1}x_{j2}\ x_{j1}^2\ x_{j2}^2\right]$$
$$K_q(\boldsymbol{x}_i, \boldsymbol{x}_j) = \varphi(\boldsymbol{x}_i)^T \varphi(\boldsymbol{x}_j) = 1 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2} + 2x_{i1}x_{j1}x_{i2}x_{j2} + x_{i1}^2x_{j1}^2 + x_{i2}^2x_{j2}^2$$

То же самое можно выполнить посредством функции ядра без преобразования в пространство признаков

$$K_q(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(1 + x_{i1}x_{j1} + x_{i2}x_{j2}\right)^2 = 1 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2} + 2x_{i1}x_{j1}x_{i2}x_{j2} + x_{i1}^2x_{j1}^2 + x_{i2}^2x_{j2}^2$$

Множество всех пар точек исходного пространства можно представить в виде матрицы ядра следующим образом

$$\mathbf{K} = \begin{bmatrix} K(\boldsymbol{x}_1, \boldsymbol{x}_1) & \cdots & K(\boldsymbol{x}_1, \boldsymbol{x}_n) \\ \vdots & \ddots & \vdots \\ K(\boldsymbol{x}_n, \boldsymbol{x}_1) & \cdots & K(\boldsymbol{x}_n, \boldsymbol{x}_n) \end{bmatrix}$$

где $K(\cdot,\cdot)$ – функция ядра.

Применение функции ядра в методе опорных векторов

Чтобы применить трюк с ядром для нелинейной классификации в методе опорных векторов, необходимо все операции свести к вычислению функции ядра

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \varphi(\boldsymbol{x}_i)^T \varphi(\boldsymbol{x}_j)$$

Новое пространство признаков будет иметь вид

$$D_\varphi = \{(\varphi(\boldsymbol{x}_i), y_i)\}_{i=1}^n$$

Сформулируем задачу оптимизации с учетом нового пространства

$$\min_{w,b,\xi_i}\left[\frac{||w||^2}{2} + C\sum_{i=1}^{n}(\xi_i)^k\right]$$

при условии

$$y_i(\boldsymbol{w}^T\varphi(\boldsymbol{x}_i) + b) \geq 1 - \xi_i, \text{и } \xi_i \geq 0, \forall \boldsymbol{x}_i \in D$$

Кусочно-линейная функция потерь

Двойственную задачу оптимизации при $k = 1$ для введенного пространства признаков можно представить как

$$\max_\alpha L_{dual} = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \varphi(\boldsymbol{x}_i)^T\varphi(\boldsymbol{x}_j) = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

при условии

$$0 \leq \alpha_i \leq C, \forall i \in D \text{ и } \sum_{i=1}^{n}\alpha_i y_i = 0$$

Вектор параметров

Для вычисления параметров используется выражение

$$\boldsymbol{w} = \sum_{i,a_i>0}\alpha_i y_i \varphi(\boldsymbol{x}_i)$$

Можно заметить, что в этом случае не используется функция ядра, а значит точки $\boldsymbol{x}_i$ необходимо в явном виде преобразовывать в пространство признаков $\varphi(\boldsymbol{x}_i)$. Однако как будет показано далее для предсказания нет необходимости отдельно вычислять параметры $\boldsymbol{w}$.

Смещение

Смещения для одной точки, лежащей на зазоре, вычисляется как

$$b_i = y_i - \boldsymbol{w}^T\varphi(\boldsymbol{x}_i) = y_i - \sum_{i,a_i>0}\alpha_i y_i \varphi(\boldsymbol{x}_i)^T\varphi(\boldsymbol{x}_j) = y_i - \sum_{i,a_i>0}\alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

Для определения общего смещения усредняем

$$b = \operatorname*{avg}_{0<\alpha_i<C}\{b_i\}$$

Классификация

Предсказание для новой точки $\boldsymbol{x}_*$ будет иметь вид

$$\hat{y} = \text{sign}\left(h\left(\varphi(\boldsymbol{x}_*)\right)\right) = \text{sign}(\boldsymbol{w}^T\varphi(\boldsymbol{x}_*) + b) = \text{sign}\left[\sum_{i,a_i>0} \alpha_i y_i \varphi(\boldsymbol{x}_i)^T \varphi(\boldsymbol{x}_*) + b\right]$$

$$= \text{sign}\left[\sum_{i,a_i>0} \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}_*) + b\right]$$

Таким образом, для обучения и классификации нам нет необходимости отображать точки исходного пространства в новое пространство признаков с большей размерностью, т.к. использование функции ядра позволяет производить вычисления над элементами исключительно исходного пространства.

5. Алгоритмы обучения

- Решение двойственной задачи. Стохастический градиентный подъем. Sequential minimal optimization (SMO)
- Решение прямой задачи. Оптимизация Ньютона

6. Метод опорных векторов для задачи регрессии

В качестве варианта применения методам опорных векторов к задаче регрессии рассмотрим линейную ε-SVM регрессию [2]. Наша задача найти такую функцию $h(x)$, которая отклонялась бы не более чем на величину ε от действительного значений y для каждого наблюдения из обучающего множества. При этом мы хотим получить как можно более плоскую функцию.

Таким образом нам надо найти

$$h(x) = \boldsymbol{w}^T\boldsymbol{x} + b$$

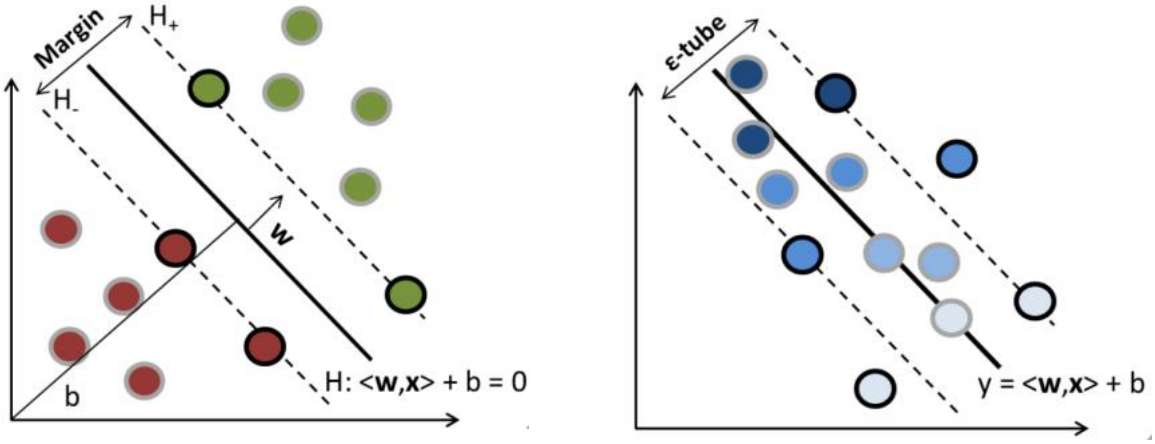
такую, что

$$\frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} \to \min$$

при условии

$$|y_i - (\boldsymbol{w}^T\boldsymbol{x}_i + b)| \leq \varepsilon$$

Чтобы ослабить условие и позволить наблюдениям выходить за установленный уровень ε (ε-канал), введем переменные ξ_i и ξ_i^* по аналогии с мягким зазором в задаче классификации. Данные переменные показывают на сколько действительное значение отклоняется от предсказанного за вычетом ε.



а) б)

Рисунок – Метод опорных векторов для классификация (а) и регрессия (б)[3]

Прямая задача

$$\min_{\boldsymbol{w},b,\xi_i}\left[\frac{||\boldsymbol{w}||^2}{2}+C\sum_{i=1}^{n}(\xi_i+\xi_i^*)\right]$$

при условии

$$y_i-(\boldsymbol{w}^T\boldsymbol{x}_i+b)\le\varepsilon+\xi_i,$$

$$(\boldsymbol{w}^T\boldsymbol{x}_i+b)-y_i\le\varepsilon+\xi_i^*,$$

$$\xi_i,\xi_i^*\ge 0\ \forall\boldsymbol{x}_i\in D,$$

Таким образом, мы наказываем те предсказания, которые выходят за пределы установленного значения ε. Штраф выражается в добавлении в целевую функцию значения ξ_i или ξ_i^* в зависимости положения наблюдения, выше или ниже функции предсказания.

Замечание

Прямую задачу можно представить с использованием кусочно-линейной функции

$$\min_{\boldsymbol{w},b}\left[C\sum_{i=1}^{n}\max(0,|y_i-(\boldsymbol{w}^T\boldsymbol{x}_i+b)|-\varepsilon)+\frac{||\boldsymbol{w}||^2}{2}\right]$$

Двойственная задача

$$\max_{\alpha}L_{dual}=\max_{\alpha}\left[\sum_{i=1}^{n}y_i(\alpha_i-\alpha_i^*)-\varepsilon\sum_{i=1}^{n}(\alpha_i+\alpha_i^*)-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)\boldsymbol{x}_i^T\boldsymbol{x}_j\right]$$

при условии

$$\sum_{i=1}^{n}(\alpha_i-\alpha_i^*)=0,$$

$$0 \le \alpha_i, \alpha_i^* \le C$$

Замечание

Двойственную форму записи можно представить в виде минимизации с сохранением условий следующим образом

$$\min_{\alpha} \left[\sum_{i=1}^{n} y_i(\alpha_i^* - \alpha_i) + \varepsilon \sum_{i=1}^{n} (\alpha_i + \alpha_i^*) + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \boldsymbol{x}_i^T \boldsymbol{x}_j \right]$$

или в матричном виде

$$\min_{\alpha} \left[\boldsymbol{y}^T(\boldsymbol{\alpha}^* - \boldsymbol{\alpha}) + \varepsilon \boldsymbol{e}^T(\boldsymbol{\alpha} + \boldsymbol{\alpha}^*) + \frac{1}{2}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)^T Q(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) \right]$$

где $\boldsymbol{e}$ – единичный вектор размера n; $Q_{ij} = \boldsymbol{x}_i^T \boldsymbol{x}_j$ или в более общем виде $Q_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$

ККТ

$$\alpha_i(\varepsilon + \xi_i - y_i + \boldsymbol{w}^T \boldsymbol{x}_i + b) = 0$$

$$\alpha_i^*(\varepsilon + \xi_i^* + y_i - \boldsymbol{w}^T \boldsymbol{x}_i - b) = 0$$

$$\xi_i(C - \alpha_i) = 0$$

$$\xi_i^*(C - \alpha_i^*) = 0$$

Для всех наблюдений, находящихся в пределах ε-канала, множители Лагранжа будут равны нулю, то есть $\alpha_i = 0$ и $\alpha_i^* = 0$. Если один из множителей α_i или α_i^* не равен нулю, то такое наблюдение называется опорным вектором.

Предсказание

$$\hat{y}_* = \boldsymbol{w}^T \boldsymbol{x}_* + b = \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) \boldsymbol{x}_i^T \boldsymbol{x}_* + b$$

Решение задачи оптимизации посредством алгоритмов квадратичного программирования (quadratic programming – QP) может быть ресурсозатратным. Поэтому используют такие методы как:

- Методы разложения (Decomposition methods)
- Последовательная минимальная оптимизация (Sequential minimal optimization – SMO)

7. Преимущества и недостатки SVM

Преимущества

- Подходит для многоразмерных данных

- Можно использовать, если количество наблюдений меньше количества признаков
- Подходит для нелинейной границы принятия решения
- Устойчив к шумам/выбросам
- Имеет хорошую обобщающую способность

Недостатки

- Плохо подходит для больших наборов данных
- Не дает вероятностные значения
- Выбор гиперпараметров (коэффициента регуляризации и ядра)
- При использовании ядра требуется хранить матрицу значений

Список литературы

1. Zaki, Mohammed & Meira Jr, Wagner. Data Mining and Analysis: Fundamental Concepts and Algorithms. 2014.
2. MathWorks Documentation. Understanding Support Vector Machine Regression. URL: https://www.mathworks.com/help/stats/understanding-support-vector-machine-regression.html
3. Rodríguez-Pérez, R., Bajorath, J. Evolution of Support Vector Machine and Regression Modeling in Chemoinformatics and Drug Discovery. J Comput Aided Mol Des 36, 355–362 (2022). https://doi.org/10.1007/s10822-022-00442-9