Метод опорных векторов для стандартной задачи классификации

Линейный классификатор

Рассмотрим задачу классификации на два класса. Пусть имеется обучающая выборка $(X, \boldsymbol{t}) = \{\boldsymbol{x}_n, \boldsymbol{t}_n\}_{n=1}^N$ объема N, где $\boldsymbol{x}_n \in \mathbb{R}^d$ – признаковое описание объекта, $t_n \in \{-1, +1\}$ – метка класса, принимающая одно из двух возможных значений. Задача состоит в том, чтобы на основе обучающей выборки спрогнозировать метку класса $\hat{t}$ для нового объекта, заданного своим признаковым описанием $\boldsymbol{x}$.

Метод опорных векторов (Support Vector Machines, SVM) относится к семейству линейных классификаторов. В этом семействе решение о принадлежности объекта $\boldsymbol{x}$ к классу $\hat{t}$ принимается по знаку линейного решающего правила:

$$f(\boldsymbol{x}) = \sum_{j=1}^{d} w_j x(j) + b = \boldsymbol{w}^T \boldsymbol{x} + b, \quad \hat{t}(\boldsymbol{x}) = \begin{cases} +1, & \text{если } f(\boldsymbol{x}) \geq 0, \\ -1, & \text{если } f(\boldsymbol{x}) < 0. \end{cases} \tag{1}$$

Здесь $w_j \in \mathbb{R}$ – некоторые веса, $b \in \mathbb{R}$ – параметр сдвига. С геометрической точки зрения линейный классификатор соответствует некоторой разделяющей гиперплоскости $f(\boldsymbol{x}) = 0$ в признаковом пространстве $\mathbb{R}^d$, при этом объект относится к первому классу, если он лежит с положительной стороны от гиперплоскости, и относится ко второму классу в противном случае (см. рис. 1a).

Решающее правило вида (1) применимо только для задачи двухклассовой классификации. Рассмотрим эквивалентную запись решающего правила (1), которую можно обобщить на случай более сложной структуры множества прогнозов $\mathcal{T}$:

$$\hat{t}(\boldsymbol{x}) = \arg \max_{t \in \{-1,+1\}} h(\boldsymbol{x}, t) = \arg \max_{t \in \{-1,+1\}} t f(\boldsymbol{x}). \tag{2}$$

Очевидно, что решающие правила (1) и (2) эквивалентны. Решающее правило типа (2), в частности, подходит для случая классификации на K классов. Введем для каждого класса свою линейную функцию $f_k(\boldsymbol{x})$, $k = 1, \ldots, K$. Тогда решающее правило можно записать как

$$\hat{t}(\boldsymbol{x}) = \arg \max_{t \in \{1,\ldots,K\}} h(\boldsymbol{x}, t) = \arg \max_{t \in \{1,\ldots,K\}} f_t(\boldsymbol{x}).$$

Оптимальная разделяющая гиперплоскость

Рассмотрим сначала случай, когда обучающая выборка $(X, \boldsymbol{t})$ является линейно разделимой (см. рис. 1b), т.е. существуют $\boldsymbol{w}, b$ такие, что

$$\boldsymbol{w}^T \boldsymbol{x}_n + b > 0, \text{ если } t_n = 1,$$
$$\boldsymbol{w}^T \boldsymbol{x}_n + b < 0, \text{ если } t_n = -1.$$

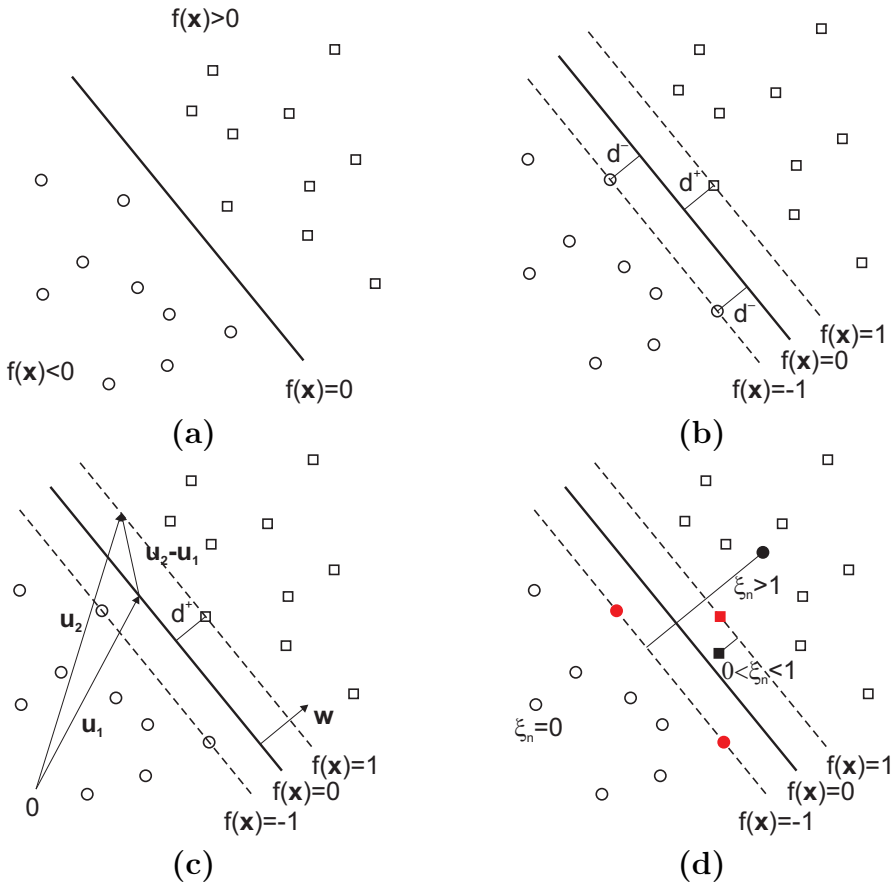


Рис. 1: Случай a — линейный классификатор соответствует построению разделяющей гиперплоскости. Случай b — зазор между гиперплоскостью и каждым из классов d^+, d^- определяется как расстояние до ближайшего объекта. Случай c — величина зазора d^+ определяется длиной проекции вектора $\boldsymbol{u}_2 - \boldsymbol{u}_1$ на вектор нормали $\boldsymbol{w}$. Случай d — ослабляющие коэффициенты ξ_n определяют наличие ошибки на объекте $\boldsymbol{x}_n$, только красные и черные объекты являются опорными и определяют расположение гиперплоскости в пространстве.

В этом случае провести гиперплоскость, корректно разделяющую данные, можно разными способами. Определим зазор между классом и гиперплоскостью как минимальное расстояние между гиперплоскостью и объектом класса. Обозначим через d^+ и d^- зазор между гиперплоскостью и первым и вторым классом соответственно (см. рис. 1b). Тогда определим *оптимальную гиперплоскость* как гиперплоскость, максимизирующую зазор:

$$\min(d^+, d^-) \to \max_{\boldsymbol{w}, b}. \tag{3}$$

Максимизация зазора между гиперплоскостью и данными позволяет надеяться на хорошую обобщающую способность в том случае, когда тестовая выборка является небольшой вариацией обучающей.

Выразим величину зазора d^+, d^- через параметры гиперплоскости $\boldsymbol{w}, b$. Рассмотрим линии уровня $f(\boldsymbol{x}) = a$, проходящие через ближайшие объекты классов к гиперплоскости (пунктирные линии на рис. 1b,c). Очевидно, что при фиксированном направлении гиперплоскости (фиксированном векторе нормали $\boldsymbol{w}$) оптимальная гиперплоскость проходит по середине между этими линиями уровня. Таким образом, можно считать, что $d^+(\boldsymbol{w}) = d^-(\boldsymbol{w})$. Заметим, что

гиперплоскость $\boldsymbol{w}^T\boldsymbol{x} + b = 0$ определена с точностью до масштаба шкалы измерения $\boldsymbol{w}$ и b. Действительно, если умножить $\boldsymbol{w}$ и b на одно и тоже число, то множество точек $\boldsymbol{x}$: $\boldsymbol{w}^T\boldsymbol{x} + b = 0$ не изменится. Однако, при таком умножении линии уровня $\boldsymbol{w}^T\boldsymbol{x} + b = a$ перемещаются. Потребуем, чтобы линии уровня, проходящие через ближайшие объекты классов к гиперплоскости, определялись как $\boldsymbol{w}^T\boldsymbol{x} + b = 1$ и $\boldsymbol{w}^T\boldsymbol{x} + b = -1$ (так всегда можно сделать, т.к. по рассуждениям выше оптимальная гиперплоскость всегда проходит по середине между этими двумя линиями уровня). Это требование однозначно фиксирует шкалу измерения для $\boldsymbol{w}$ и b.

Рассмотрим произвольный вектор $\boldsymbol{u}_1$, принадлежащий гиперплоскости, и произвольный вектор $\boldsymbol{u}_2$, принадлежащий линии уровня $\boldsymbol{w}^T\boldsymbol{x} + b = 1$. Очевидно, что величина зазора d^+ равна длине проекции вектора $\boldsymbol{u}_2 - \boldsymbol{u}_1$ на вектор нормали $\boldsymbol{w}$ (см. рис. 1c). Тогда:

$$\begin{array}{l}\boldsymbol{w}^T\boldsymbol{u}_1 + b = 0,\\ \boldsymbol{w}^T\boldsymbol{u}_2 + b = 1,\end{array} \Rightarrow \boldsymbol{w}^T(\boldsymbol{u}_2 - \boldsymbol{u}_1) = 1 \Rightarrow \frac{1}{\|\boldsymbol{w}\|}\boldsymbol{w}^T(\boldsymbol{u}_2 - \boldsymbol{u}_1) = \mathrm{pr}_{\boldsymbol{w}}(\boldsymbol{u}_2 - \boldsymbol{u}_1) = \frac{1}{\|\boldsymbol{w}\|}.$$

Таким образом, величина зазора $d^+ = d^- = 1/\|\boldsymbol{w}\|$. Теперь задачу максимизации зазора (3) с учетом корректного разделения данных можно записать как

$$\left\{\begin{array}{l}\frac{2}{\|\boldsymbol{w}\|} \to \max\limits_{\boldsymbol{w},b},\\ \boldsymbol{w}^T\boldsymbol{x}_n + b \geq 1, \text{ если } t_n = 1,\\ \boldsymbol{w}^T\boldsymbol{x}_n + b \leq -1, \text{ если } t_n = -1,\end{array}\right. \Leftrightarrow \left\{\begin{array}{l}\frac{1}{2}\|\boldsymbol{w}\|^2 \to \min\limits_{\boldsymbol{w},b},\\ t_n(\boldsymbol{w}^T\boldsymbol{x}_n + b) \geq 1.\end{array}\right. \tag{4}$$

Заметим, что в эквивалентном переходе выше был добавлен квадрат к $\|\boldsymbol{w}\|$. Добавление квадрата не меняет решение задачи, но делает саму задачу намного проще, т.к. теперь нужно минимизировать выпуклую функцию.

Рассмотрим теперь случай произвольных данных. В этом случае условия $t_n(\boldsymbol{w}^T\boldsymbol{x}_n + b) \geq 1$ не могут быть выполнены для всех объектов. Тогда добавим в эти условия т.н. ослабляющие коэффициенты $\xi_n \geq 0$:

$$t_n(\boldsymbol{w}^T\boldsymbol{x}_n + b) \geq 1 - \xi_n.$$

Очевидно, что возможны три ситуации (см. рис. 1d):

$\xi_n = 0,$	ошибки нет, объект $\boldsymbol{x}_n$ лежит за линиями уровня $\lvert f(\boldsymbol{x})\rvert = 1,$
$0 < \xi_n \leq 1,$	ошибки нет, объект $\boldsymbol{x}_n$ лежит внутри корридора $0 \leq t_n f(\boldsymbol{x}) < 1,$
$\xi_n > 1,$	ошибка есть, величина ошибки пропорциональна расстоянию от объекта $\boldsymbol{x}_n$ до гиперплоскости.

Модифицируем критерий оптимизации в задаче (4), включив в него минимизацию числа ошибок в выборке:

$$\begin{array}{l}\dfrac{1}{2}\|\boldsymbol{w}\|^2 + C\sum\limits_{n=1}^{N}\xi_n \to \min\limits_{\boldsymbol{w},b,\boldsymbol{\xi}},\\ t_n(\boldsymbol{w}^T\boldsymbol{x}_n + b) \geq 1 - \xi_n,\\ \xi_n \geq 0.\end{array} \tag{5}$$

Здесь $C \geq 0$ – коэффициент регуляризации, который определяет компромисс между количеством ошибок на обучающей выборке и простотой линейного решающего правила (близость весов w_j к нулю).

Таким образом, метод опорных векторов заключается в построении разделяющей гиперплоскости с помощью решения задачи оптимизации (5). При этом коэффициент регуляризации C задается пользователем до начала обучения. После обучения прогнозирование метки класса для нового объекта $\boldsymbol{x}$ происходит по схеме (1) или, эквивалентно, (2).

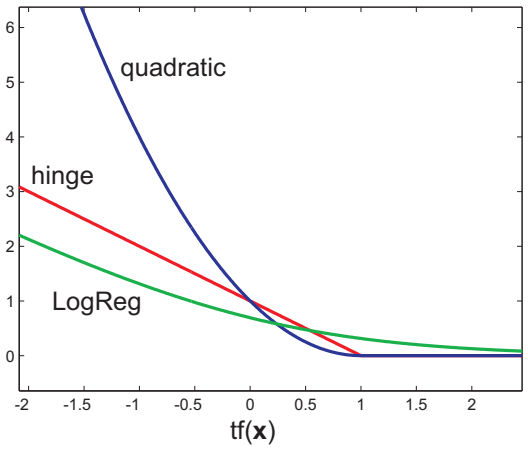


Рис. 2: Различные способы введения функции потерь – стандартная функция потерь в SVM l_{hinge} (красная кривая), квадратичная функция потерь (синяя кривая), логистическая функция потерь (зеленая кривая).

SVM как задача безусловной оптимизации

Задачу условной оптимизации (5) можно эквивалентно записать как задачу безусловной оптимизации. Действительно,

$$\begin{array}{l}\xi_n \geq 1 - t_n f(\boldsymbol{x}_n), \\ \xi_n \geq 0,\end{array} \quad \Rightarrow \quad \xi_n \geq \max(0, 1 - t_n f(\boldsymbol{x}_n)) \triangleq l_{hinge}(t_n, f(\boldsymbol{x}_n)).$$

В задаче (5) требуется минимизировать значения ξ_n. Очевидно, что при фиксированных $\boldsymbol{w}, b$ этот минимум достигается при $\xi_n = l_{hinge}(t_n, f(\boldsymbol{x}_n))$. Таким образом, задача (5) эквивалентна следующей задаче:

$$\sum_{n=1}^{N} l_{hinge}(t_n, f(\boldsymbol{x}_n)) + \frac{1}{2C}\|\boldsymbol{w}\|^2 \to \min_{\boldsymbol{w}, b}.$$

В этой задаче первое слагаемое соответствует ошибке классификатора на обучающей выборке, измеряемой с помощью функции потерь l_{hinge}. Второе слагаемое является регуляризатором, штрафующим излишнюю перенастройку классификатора на обучающую выборку.

Функцию потерь l и регуляризатор можно вводить различными способами (см. рис. 2). При этом можно получить разные модификации стандартного метода опорных векторов. Функция потерь l_{hinge} не является дифференцируемой в точке $tf(\boldsymbol{x}) = 1$. Это обстоятельство затрудняет решение соответствующей задачи оптимизации. Можно рассмотреть квадратичный аналог функции l_{hinge}: $l_2(t, f(\boldsymbol{x})) = \max(0, (1 - tf(\boldsymbol{x}))^2)$. Такая функция потерь является всюду дифференцируемой, что упрощает решение задачи оптимизации. Однако, при этом метод становится неустойчивым к выбросам в данных, т.к. большие ошибки штрафуются слишком сильно. Логистическая функция потерь $l_{logistic}(t, f(\boldsymbol{x})) = \log(1 + \exp(-tf(\boldsymbol{x})))$ является, с одной стороны, всюду дифференцируемой и, с другой стороны, растет линейно с увеличением ошибки. Метод поиска гиперплоскости с логистической функцией потерь и квадратичным регуляризатором получил название гребневой логистической регрессии.

Наряду с квадратичным регуляризатором $\|\boldsymbol{w}\|^2$ рассматривают также L_1-регуляризацию $\|\boldsymbol{w}\|_{L_1} = \sum_{j=1}^{d} |w_j|$ и L_0-регуляризацию $\|\boldsymbol{w}\|_{L_0} = \sum_{j=1}^{d} [w_j \neq 0]$. В отличие от квадратичного штрафа, L_1- и L_0-регуляризации позволяют получать т.н. разреженные решения для $\boldsymbol{w}$,

т.е. такие решения, в которых большинство весов w_j тождественно равны нулю. При этом L_0-регуляризация обеспечивает наиболее разреженное решение, но соответствующая задача оптимизации является негладкой и невыпуклой, что значительно усложняет метод оптимизации для такой задачи. L_1-регуляризация приводит к негладкой, но выпуклой задаче. Для такой задачи существуют эффективные методы оптимизации [4].

Двойственная задача оптимизации

Рассмотрим понятие двойственной задачи оптимизации в общем виде. Пусть имеется выпуклая задача условной оптимизации

$$\begin{aligned} & f(\boldsymbol{x}) \to \min_{\boldsymbol{x}}, \\ & f_i(\boldsymbol{x}) \le 0,\ i = 1, \dots, p, \\ & h_j(\boldsymbol{x}) = 0,\ j = 1, \dots, q. \end{aligned} \tag{6}$$

Здесь все функции f, f_i, h_j предполагаются выпуклыми. Хорошим свойством выпуклой задачи является то обстоятельство, что любой локальный оптимум задачи является ее глобальным оптимумом. Таким образом, в частности, не возникает проблем с локальными экстремумами в итерационных методах оптимизации. Введем функцию Лагранжа следующим образом:

$$L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\boldsymbol{x}) + \sum_{i=1}^{p} \lambda_i f_i(\boldsymbol{x}) + \sum_{j=1}^{q} \mu_j h_j(\boldsymbol{x}).$$

Тогда по выпуклому варианту теоремы Куна-Таккера: если $\boldsymbol{x}^*$ является решением задачи (6), то найдутся $(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ такие, что выполнены три условия:

1. *Принцип минимума:* $L(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = \min_{\boldsymbol{x}} L(\boldsymbol{x}, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$,
2. *Условия дополняющей нежесткости:* $\lambda_i^* f_i(\boldsymbol{x}^*) = 0$,
3. *Условия неотрицательности:* $\lambda_i^* \ge 0$.

Благодаря выпуклости, свойства 1)– 3) являются и достаточными условиями оптимального решения, если все функции h_j являются аффинными, а внутренность множества ограничений не пуста, т.е. найдется $\tilde{\boldsymbol{x}} : f_i(\tilde{\boldsymbol{x}}) < 0\ \forall i$ (т.н. условия Слейтера).

Введем на основе функции Лагранжа $L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu})$ *двойственную функцию*

$$g(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \min_{\boldsymbol{x}} L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}).$$

Тогда *двойственной задачей* к задаче условной оптимизации (6) называется следующая задача:

$$\begin{aligned} & g(\boldsymbol{\lambda}, \boldsymbol{\mu}) \to \max_{\boldsymbol{\lambda},\ \boldsymbol{\mu}}, \\ & \lambda_i \ge 0,\ i = 1, \dots, p. \end{aligned} \tag{7}$$

Обозначим через $(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ решение двойственной задачи (7). Легко показать, что оптимальное значение функционала двойственной задачи не превосходит аналогичное для прямой задачи, т.е. $g(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \le f(\boldsymbol{x}^*)$. Обозначим через $\mathcal{D}$ множество допустимых точек $\boldsymbol{x}$ прямой задачи:

$\mathcal{D} = \{\boldsymbol{x} : f_i(\boldsymbol{x}) \le 0 \ \forall i, \ h_j(\boldsymbol{x}) = 0 \ \forall j\}$. Очевидно, что при $\lambda_i \ge 0 \ \forall i$ и $\boldsymbol{x} \in \mathcal{D}$ выполнено $g(\boldsymbol{\lambda}, \boldsymbol{\mu}) \le f(\boldsymbol{x})$. Отсюда

$$g(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \triangleq \max_{\lambda \ge 0, \ \boldsymbol{\mu}} g(\boldsymbol{\lambda}, \boldsymbol{\mu}) \le \min_{\boldsymbol{x} \in \mathcal{D}} f(\boldsymbol{x}) = f(\boldsymbol{x}^*).$$

В том случае, если функции f, f_i, h_j является выпуклыми, то выполняется т.н. *условие сильной двойственности*, т.е. оптимальное значение функционала в прямой и двойственной задачах совпадают между собой

$$g(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = f(\boldsymbol{x}^*).$$

Заметим, что из совпадения оптимальных значений в прямой и двойственной задачах не следует, что можно выразить решение прямой задачи $\boldsymbol{x}^*$ через решение двойственной $(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ или наоборот. В общем случае знание оптимального значения $f(\boldsymbol{x}^*)$ или $g(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ никак не облегчает поиск самих решений.

В заключение заметим, что двойственная задача всегда является выпуклой, даже в том случае, если прямая задача выпуклой не является. Как уже было отмечено выше, выпуклые задачи решать легче, чем невыпуклые (например, в выпуклых задачах все локальные оптимумы являются и глобальными).

Двойственная задача для SVM

Построим двойственную задачу для выпуклой задачи оптимизации (5). Для этого рассмотрим функцию Лагранжа:

$$L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{n=1}^{N}\xi_n - \sum_{n=1}^{N}\lambda_n(t_n(\boldsymbol{w}^T\boldsymbol{x}_n + b) - 1 + \xi_n) - \sum_{n=1}^{N}\mu_n\xi_n.$$

Найдем минимум функции Лагранжа по прямым переменным:

$$\nabla_{\boldsymbol{w}} L(\boldsymbol{w}, b, \boldsymbol{\xi}) = \boldsymbol{w} - \sum_{n=1}^{N}\lambda_n t_n \boldsymbol{x}_n = \boldsymbol{0} \quad \Rightarrow \quad \boldsymbol{w} = \sum_{n=1}^{N}\lambda_n t_n \boldsymbol{x}_n, \tag{8}$$

$$\frac{\partial}{\partial b} L(\boldsymbol{w}, b, \boldsymbol{\xi}) = -\sum_{n=1}^{N}\lambda_n t_n = 0 \quad \Rightarrow \quad \sum_{n=1}^{N}\lambda_n t_n = 0, \tag{9}$$

$$\frac{\partial}{\partial \xi_n} L(\boldsymbol{w}, b, \boldsymbol{\xi}) = -\lambda_n - \mu_n + C = 0 \quad \Rightarrow \quad \lambda_n + \mu_n = C. \tag{10}$$

Подставляя выражения (8)–(10) в функцию Лагранжа, найдем двойственную функцию $g(\boldsymbol{\lambda}, \boldsymbol{\mu})$:

$$g(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \frac{1}{2}\sum_{n,m=1}^{N}\lambda_n\lambda_m t_n t_m \boldsymbol{x}_n^T\boldsymbol{x}_m - \sum_{n,m=1}^{N}\lambda_n\lambda_m t_n t_m \boldsymbol{x}_n^T\boldsymbol{x}_m + \sum_{n=1}^{N}\lambda_n =$$

$$- \frac{1}{2}\sum_{n,m=1}^{N}\lambda_n\lambda_m t_n t_m \boldsymbol{x}_n^T\boldsymbol{x}_m + \sum_{n=1}^{N}\lambda_n = -\frac{1}{2}\boldsymbol{\lambda}^T \mathrm{diag}(\boldsymbol{t}) X^T X \mathrm{diag}(\boldsymbol{t})\boldsymbol{\lambda} + \boldsymbol{\lambda}^T\boldsymbol{1}.$$

Здесь $\mathbf{1}$ – вектор из единиц длины N. Таким образом, мы получаем следующую двойственную задачу для SVM:

$$\begin{aligned} &-\frac{1}{2}\boldsymbol{\lambda}^T \mathrm{diag}(\boldsymbol{t}) X^T X \mathrm{diag}(\boldsymbol{t})\boldsymbol{\lambda} + \boldsymbol{\lambda}^T \mathbf{1} \to \max_{\boldsymbol{\lambda}}, \\ &\boldsymbol{t}^T \boldsymbol{\lambda} = 0, \\ &0 \leq \lambda_n \leq C,\ n = 1, \ldots, N. \end{aligned} \tag{11}$$

Так как прямая задача оптимизации (5) является выпуклой, то оптимальные значения функционалов прямой и двойственной задачи совпадают. Более того, в данном случае решение прямой задачи $(\boldsymbol{w}^*, b^*)$ может быть выражено через решение двойственной задачи $\boldsymbol{\lambda}^*$. Действительно, решение для $\boldsymbol{w}^*$ получается из условия (8). Оптимальный сдвиг b^* можно получить из условий дополняющей нежесткости:

$$\lambda_n^*(t_n((\boldsymbol{w}^*)^T \boldsymbol{x}_n + b^*) + 1 - \xi_n^*) = 0,\ n = 1, \ldots, N,$$

Эти условия приводят к трем возможным ситуациям:

1) $\lambda_n^* = 0$,	объект x_n лежит внутри класса с правильной стороны от гиперплоскости;
2) $t_n((\boldsymbol{w}^*)^T \boldsymbol{x}_n + b^*) = 1$, $\xi_n^* = 0$,	объект x_n лежит на единичной линии уровня гиперплоскости (красные объекты на рис. 1d);
3) $t_n((\boldsymbol{w}^*)^T \boldsymbol{x}_n + b^*) = 1 - \xi_n^*$, $\xi_n^* > 0$,	на объекте x_n происходит ошибка, либо он лежит внутри корридора, образованного единичными линиями уровня гиперплоскости (черные объекты на рис. 1d).

Рассмотрим объекты, отвечающие случаю 2). Заметим, что такие объекты существуют всегда (за исключением вырожденного случая, когда в выборке присутствуют объекты только одного класса). Тогда оптимальную величину сдвига гиперплоскости b^* можно найти из соответствующих условий для этих объектов:

$$b^* = t_n(1 - (\boldsymbol{w}^*)^T \boldsymbol{x}_n).$$

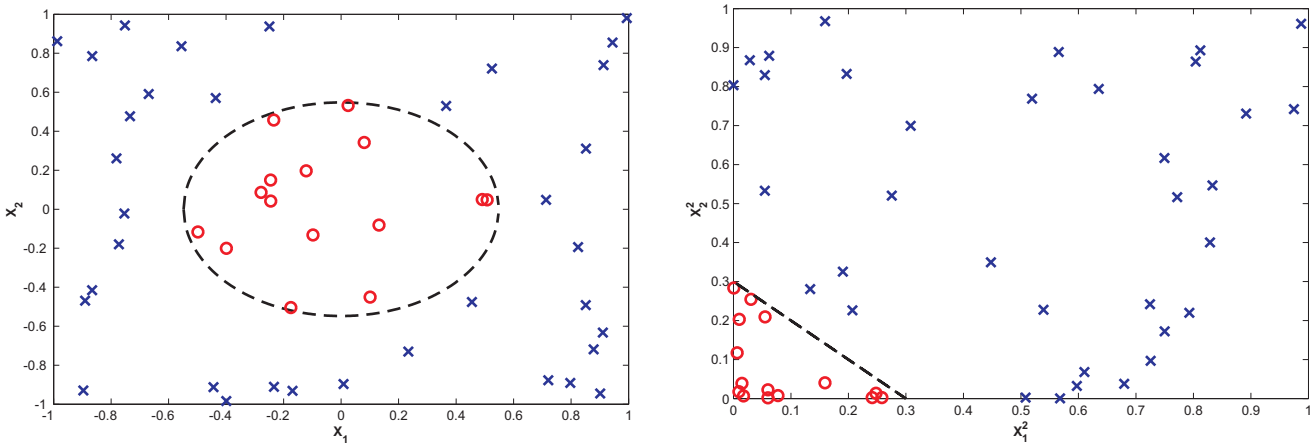
На практике для устойчивости обычно производят усреднение этих выражений для всех объектов, отвечающих условию 2).

Рассмотрим получившееся линейное решающее правило:

$$f(\boldsymbol{x}) = (\boldsymbol{w}^*)^T \boldsymbol{x} + b^* = \sum_{n=1}^{N} \lambda_n^* t_n \boldsymbol{x}_n^T \boldsymbol{x} + b^*. \tag{12}$$

Заметим, что только объекты обучения, отвечающие условиям 2) и 3) ($\lambda_n^* \neq 0$), влияют на данное решающее правило. Такие объекты называются *опорными*. Если из выборки удалить все неопорные объекты, то положение оптимальной разделяющей гиперплоскости не изменится. Следовательно, метод опорных векторов является разреженным по объектам обучения (но не по признакам!).

Таким образом, для обучения метода опорных векторов можно решать как прямую задачу оптимизации (5), так и двойственную задачу (11) (из решения которой легко получить решение прямой задачи). При этом двойственная задача содержит меньше переменных (N против $N +$



(a) (b)

Рис. 3: Иллюстрация ядрового перехода. Линейно неразделимые данные слева (a) становятся линейно разделимыми при переходе в пространство (x_1^2, x_2^2) (b).

$d+1$ в прямой задаче), а граничные условия имеют простую форму гиперпараллелепипеда $(0 \leq \lambda_n \leq C)$.

Популярным подходом к решению двойственной задачи (11) является метод декомпозиции. В этом методе на каждой итерации выбирается небольшое подмножество переменных λ, по которым осуществляется оптимизация. Если оптимизируемых переменных всего две, то соответствующая задача решается аналитически. Итерационный процесс останавливается, когда текущее решение удовлетворяет описанным выше достаточным условиям Куна-Таккера. Одной из наиболее популярных библиотек обучения SVM является LIBSVM[1]. В ней реализован метод декомпозиции, описанный в работе [2].

Ядровой переход

На практике проверхность, разделяющая два класса, может быть существенно нелинейной. Метод опорных векторов можно обобщить на случай построения нелинейных разделяющих поверхностей с помощью т.н. ядрового перехода. Рассмотрим модельный пример, показанный на рис. 3a. Здесь данные являются разделимыми с помощью окружности. После перехода в пространство квадратов исходных признаков (x_1^2, x_2^2) данные становятся линейно разделимыми (см. рис. 3b). В более общем случае поверхность второго порядка в двухмерном пространстве (x_1, x_2) задается уравнением:

$$ax_1^2 + bx_1x_2 + cx_2^2 + dx_1 + ex_2 + f = 0.$$

Данное уравнение соответствует гиперплоскости в пятимерном пространстве $(x_1^2, x_1x_2, x_2^2, x_1, x_2)$.

Рассмотрим преобразование $\Phi : \mathbb{R}^d \to \mathcal{H}$ из исходного признакового пространства $\mathbb{R}^d$ в новое пространство $\mathcal{H}$. Будем искать оптимальную разделяющую гиперплоскость в новом пространстве $\mathcal{H}$ с помощью метода опорных векторов. Двойственная задача оптимизации (11) зависит от объектов обучения только в виде попарных скалярных произведений $\langle \Phi(\boldsymbol{x}_n), \Phi(\boldsymbol{x}_m) \rangle$.

[1] http://www.csie.ntu.edu.tw/∼cjlin/libsvm/

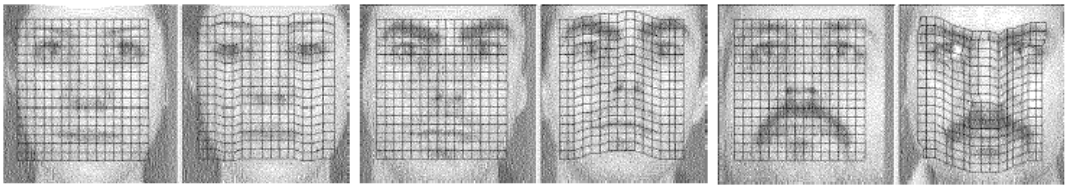


Рис. 4: Примеры эластичного преобразования растра при совмещении изображений.

Решающее правило (12) также зависит только от скалярных произведений между распознаваемым объектом и объектами обучения $\langle \Phi(\boldsymbol{x}_n), \Phi(\boldsymbol{x}) \rangle$. Предположим далее, что скалярное произведение в пространстве $\mathcal{H}$ известно как функция K в исходном пространстве $\mathbb{R}^d$:

$$\langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{y}) \rangle_{\mathcal{H}} = K(\boldsymbol{x}, \boldsymbol{y}).$$

Такую функцию будем в дальнейшем называть *ядровой функцией*. Тогда для обучения метода опорных векторов не требуется знать преобразование Φ, достаточно лишь задать ядровую функцию K.

Очевидно, что не для любой функции K найдется преобразование Φ и пространство $\mathcal{H}$, для которого K будет задавать скалярное произведение. Необходимыми и достаточными условиями для этого являются следующие:

1. *Симметричность*: $K(\boldsymbol{x}, \boldsymbol{y}) = K(\boldsymbol{y}, \boldsymbol{x})$,
2. *Условие Мерсера*: $\forall g : \int g(\boldsymbol{x})^2 d\boldsymbol{x} < \infty$ выполнено $\int K(\boldsymbol{x}, \boldsymbol{y}) g(\boldsymbol{x}) g(\boldsymbol{y}) d\boldsymbol{x} d\boldsymbol{y} \geq 0$.

Примеры ядровых функций, которые удовлетворяют условиям выше:

$$1)\ \textit{Линейная}: \quad K(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^T \boldsymbol{y} + \theta,\ \theta \geq 0,$$

$$2)\ \textit{Степенная}: \quad K(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^T \boldsymbol{y} + \theta)^d,\ \theta \geq 0,\ d \in \mathbb{N},$$

$$3)\ \textit{Радиальная}: \quad K(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\tfrac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{2\sigma^2}\right),\ \sigma > 0.$$

Использование ядровых функций также бывает оправдано в тех случаях, когда пространство объектов обладает сложной структурой (например, изображения), и задание скалярных произведений между парами объектов в таком пространстве оказывается легче, чем выбор пространства признаков. Такой подход получил название *беспризнакового распознавания образов* [5].

Рассмотрим для примера задачу идентификации личности по фотографии (см. рис. 4). Найдем эластичную деформацию между двумя изображениями (I_1, I_2) путем минимизации парно-сепарабельной энергии:

$$\sum_p (I_1(p) - I_2(p + x_p))^2 + \sum_{(i,j) \in \mathcal{E}} \|x_i - x_j\|^2 \to \min_X .$$

Здесь x_p – смещение пиксела p. Тогда скалярное произведение между парой изображений можно ввести следующим образом:

$$K(I_1, I_2) = \sum_p I_1(p) I_2(p + x_p).$$

Такая ядровая функция получила название *эластичной*. Другие примеры введения ядровых функций для различных задач беспризнакового распознавания образов можно найти в работе [5].

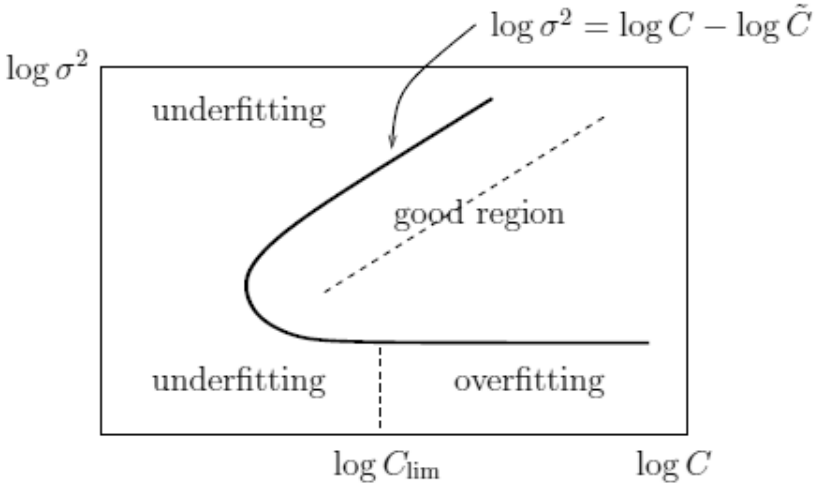


Рис. 5: Асимптотическое поведение SVM с радиальной ядровой функцией в пространстве параметров (C, σ).

Настройка параметров

При обучении метода опорных векторов требуется выбрать коэффициент регуляризации C и, как правило, значения параметров ядровой функции, например, значение σ для семейства радиальных ядровых функций. Обычно эти параметры настраиваются с помощью скользящего контроля. Однако, скользящий контроль требует многократного запуска процедуры обучения SVM, что может требовать значительного времени. Последние исследования в области методов оптимизации для SVM показывают, что обучение SVM с линейной ядровой функцией можно проводить значительно быстрее, чем обучение с произвольной ядровой функцией [1]. Поэтому для обучения линейного SVM была разработана специальная библиотека LINLINEAR[2]. Например, обучение с помощью LIBLINEAR для задачи с 677399 объектами и 47236 признаками занимает всего около 2 секунд (подробнее см. статью [1]). В результате, с точки зрения времени обучения на практике оказывается выгодным подбирать нелинейное преобразование Φ и пространство $\mathcal{H}$ в явном виде, а затем запускать линейный SVM в этом пространстве. Обычно такой подбор ограничивается рассмотрением различных суперпозиций простых функций от исходных признаков, таких как произведение, сумма, возведение в квадрат и т.д.

Ручной подбор признакового пространства, все же, требует скрупулезного исследования. В том случае, если размер обучающей выборки не очень большой (несколько тысяч объектов и несколько сотен признаков), то для экономии времени обычно ограничиваются радиальной ядровой функцией, а параметры (C, σ) подбираются по сетке с помощью скользящего контроля. Тем не менее, подобный перебор по сетке может требовать значительного времени. Поэтому авторы работы [3] предложили универсальный способ ускорения вычислений в данном случае. Они обнаружили, что если количество объектов в двух классах не одинаково и больше двух, и в выборке отсутствуют идентичные объекты, то справедливы следующие асимптотические результаты:

1. Однозначное недообучение метода (отнесение всех объектов в класс большинства) обнаруживается в трех случаях: a) σ фиксировано и $C \to 0$, b) C фиксировано и мало,

[2]http://www.csie.ntu.edu.tw/∼cjlin/liblinear/

$\sigma \to 0$, c) $\sigma \to +\infty$ и C фиксировано;

2. Однозначное переобучение метода (малая окрестность объектов из класса меньшинства классифицируется верно, а остальные объекты относятся к классу большинства) обнаруживается, если C фиксировано и велико, а $\sigma \to 0$;

3. Если σ фиксировано и $C \to +\infty$, то метод обладает нулевой ошибкой на обучении; это соответствует переобучению, если выборка содержит хотя бы небольшой шум;

4. Если $\sigma \to +\infty$ и $C = \tilde{C}\sigma^2$, где $\tilde{C}$ фиксировано, то метод сходится к линейному SVM с параметром регуляризации $\tilde{C}$.

Все эти случаи схематично иллюстрируются на рис. 5. Результат 4, в частности, приводит к тому, что если осуществляется полный перебор по сетке в пространстве (C, σ), то нет необходимости запускать линейный SVM с подбором коэффициента C.

На основе полученных результатов авторы работы [3] предложили следующий простой способ ускорения вычислений при обучении SVM с радиальной ядровой функцией. Сначала обучается линейный SVM с подбором коэффициента C. Обозначим через $\tilde{C}$ наилучшее найденное значение C для линейного SVM. Затем, параметры (C, σ) подбираются в одномерном пространстве $C = \tilde{C}\sigma^2$. Таким образом, перебор по двумерной сетке заменяется на два перебора по одномерной сетке. При этом первый из этих переборов осуществляется с помощью сверхэффективного метода обучения LIBLINEAR.

Список литературы

[1] *C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S.S. Keerthi, S. Sundararajan.* A Dual Coordinate Descent Method for Large-scale Linear SVM // ICML, 2008.

[2] *R.-E. Fan, P.-H. Chen, C.-J. Lin.* Working set selection using second order information for training SVM // Journal of Machine Learning Research, V. 6, 2005, pp. 1889–1918.

[3] *S.S. Keerthi, C.-J. Lin.* Asymptotic Behaviors of Support Vector Machines for Gaussian Kernel // Neural Computation, V. 15, 2003, pp. 1667–1689.

[4] *J. Friedman, T. Hastie, R. Tibshirani.* Regularized Paths for Generalized Linear Models via Coordinate Descent // Journal of Statistical Software, V. 33, No. 1, 2010.

[5] *О.С. Середин.* Методы и алгоритмы беспризнакового распознавания образов // Дисс. к.ф.-м.н., Тульский государственный университет, 2001.