Семинары по линейным классификаторам

Евгений Соколов
sokolov.evg@gmail.com

1 декабря 2016 г.

1 Условия Куна-Таккера и SVM

§1.1 Условия Куна-Таккера и двойственность, продолжение

Задача 1.1. *Покажите, что задача минимизации регуляризованного функционала*

$$Q(w) + \tau \|w\|_p \to \min_w \tag{1.1}$$

с $p \geqslant 1$ *и* $\tau \geqslant 0$ *эквивалентна условной задаче*

$$\begin{cases} Q(w) \to \min\limits_w \\ \|w\|_p \leqslant C \end{cases} \tag{1.2}$$

если функционал $Q(w)$ *является выпуклым. Здесь имеется ввиду, что для любого* τ *найдется такое* C*, что эти задачи эквивалентны, и наоборот — для любого* C *найдется такое* τ*.*

Решение. Задача (1.2) является выпуклой и для нее выполнено условие Слейтера, поэтому вектор w_* является ее решением тогда и только тогда, когда он удовлетворяет условиям Куна-Таккера:

$$\begin{cases} \nabla_w \left(Q(w_*) + \lambda^* \|w_*\|_p\right) = 0 \\ \|w_*\|_p \leqslant C \\ \lambda^* \geqslant 0 \\ \lambda^* (\|w_*\|_p - C) = 0 \end{cases}$$

Пусть w_* — решение задачи (1.2), тогда из условий Куна-Таккера получаем, что градиент лагранжиана в данной точке равен нулю. Поскольку лагражиан является выпуклым, то из равенства нулю градиента в точке w_* следует, что w_* является глобальным минимумом лагранжиана. Следовательно, вектор w_* является решением задачи

$$Q(w) + \lambda^* \left(\|w\|_p - C\right) \to \min_w,$$

которая эквивалентна задаче (1.1) при $\tau = \lambda^*$. Значит, если w_* является решением задачи (1.2), то он является решением задачи (1.1).

Пусть теперь w_* — решение задачи (1.1). Положим $C = \|w_*\|_p$ и $\lambda^* = \tau$. Тогда пара (w_*, λ^*) удовлетворяет условиям Куна-Таккера и, следовательно, является решением задачи (1.2).

■

§1.2 Метод опорных векторов

1.2.1 Формулировка

Будем рассматривать линейные классификаторы вида

$$a(x) = \operatorname{sign}\langle w, x\rangle + b, \qquad w \in \mathbb{R}^d, b \in \mathbb{R}.$$

Линейной разделимая выборка. Будем считать, что существуют такие параметры w_* и b_*, что соответствующий им классификатор $a(x)$ не допускает ни одной ошибки на обучающей выборке. В этом случае говорят, что выборка *линейно разделима*.

Пусть задан некоторый классификатор $a(x) = \operatorname{sign}\langle w, x\rangle + b$. Заметим, что если одновременно умножить параметры w и b на одну и ту же положительную константу, то классификатор не изменится. Распорядимся этой свободой выбора и отнормируем параметры так, что

$$\min_{x\in X^\ell} |\langle w, x\rangle + b| = 1. \tag{1.3}$$

Расстояние от произвольной точки $x_0 \in \mathbb{R}^d$ до гиперплоскости, определяемой данным классификатором, равно

$$\rho(x_0, a) = \frac{|\langle w, x\rangle + b|}{\|w\|}.$$

Тогда расстояние от гиперплоскости до ближайшего объекта обучающей выборки равно

$$\min_{x\in X^\ell} \frac{|\langle w, x\rangle + b|}{\|w\|} = \frac{1}{\|w\|} \min_{x\in X^\ell} |\langle w, x\rangle + b| = \frac{1}{\|w\|}.$$

Данная величина также называется *отступом (margin)*.

Таким образом, если классификатор без ошибок разделяет обучающую выборку, то ширина его разделяющей полосы равна $\frac{2}{\|w\|}$. Известно, что максимизация ширины разделяющей полосы приводит к повышению обобщающей способности классификатора [1]. Вспомним также, что на повышение обобщающей способности направлена и регуляризация, которая штрафует большую норму весов — а чем больше норма весов, тем меньше ширина разделяющей полосы.

Итак, требуется построить классификатор, идеально разделяющий обучающую выборку, и при этом имеющий максимальный отступ. Запишем соответствующую оптимизационную задачу:

$$\begin{cases} \dfrac{1}{2}\|w\|^2 \to \min\limits_{w,b} \\ y_i\left(\langle w, x_i\rangle + b\right) \geqslant 1, \quad i = 1, \ldots, \ell. \end{cases} \tag{1.4}$$

Здесь мы воспользовались тем, что линейный классификатор дает правильный ответ на объекте x_i тогда и только тогда, когда $y_i(\langle w, x_i\rangle + b) \geqslant 0$. Более того, из условия нормировки (1.3) следует, что $y_i(\langle w, x_i\rangle + b) \geqslant 1$.

В данной задаче функционал является строго выпуклым, а ограничения линейными, поэтому сама задача является выпуклой и имеет единственное решение. Более того, задача является квадратичной и может быть решена крайне эффективно.

Неразделимый случай. Рассмотрим теперь общий случай, когда выборку невозможно идеально разделить гиперплоскостью. Это означает, что какие бы w и b мы не взяли, хотя бы одно из ограничений в задаче (1.4) будет нарушено:

$$\exists x_i \in X^\ell : \; y_i\left(\langle w, x_i\rangle + b\right) < 1.$$

Сделаем эти ограничения «мягкими», введя штраф $\xi_i \geqslant 0$ за их нарушение:

$$y_i\left(\langle w, x_i\rangle + b\right) \geqslant 1 - \xi_i, \quad i = 1, \ldots, \ell.$$

Отметим, что если отступ объекта лежит между нулем и единицей ($0 \leqslant y_i\left(\langle w, x_i\rangle + b\right) < 1$), то объект верно классифицируется, но имеет ненулевой штраф $\xi > 0$. Таким образом, мы штрафуем объекты за попадание внутрь разделяющей полосы.

Величина $\frac{1}{\|w\|}$ в данном случае называется *мягким отступом (soft margin)*. С одной стороны, мы хотим максимизировать отступ, с другой — минимизировать штраф за неидеальное разделение выборки $\sum_{i=1}^{\ell} \xi_i$. Эти две задачи противоречат друг другу: как правило, излишняя подгонка под выборку приводит к маленькому отступу, и наоборот — максимизация отступа приводит к большой ошибке на обучении. В качестве компромисса будем минимизировать взвешенную сумму двух указанных величин. Приходим к оптимизационной задаче

$$\begin{cases} \dfrac{1}{2}\|w\|^2 + C\sum\limits_{i=1}^{\ell} \xi_i \to \min\limits_{w,b,\xi} \\ y_i\left(\langle w, x_i\rangle + b\right) \geqslant 1 - \xi_i, \quad i = 1, \ldots, \ell, \\ \xi_i \geqslant 0, \quad i = 1, \ldots, \ell. \end{cases} \tag{1.5}$$

Чем больше здесь параметр C, тем сильнее мы будем настраиваться на обучающую выборку.

Данная задача также является выпуклой и имеет единственное решение.

§1.3 Вывод двойственной задачи

Построим двойственную задачу к (1.5):

$$\begin{cases} \dfrac{1}{2}\|w\|^2 + C\sum\limits_{i=1}^{\ell} \xi_i \to \min\limits_{w,b,\xi} \\ y_i\left(\langle w, x_i\rangle + b\right) \geqslant 1 - \xi_i, \quad i = 1, \ldots, \ell, \\ \xi \geqslant 0, \quad i = 1, \ldots, \ell. \end{cases}$$

Запишем лагранжиан:

$$L(w,b,\xi,\lambda,\mu) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{\ell}\xi_i - \sum_{i=1}^{\ell}\lambda_i\left[y_i\left(\langle w,x_i\rangle + b\right) - 1 + \xi_i\right] - \sum_{i=1}^{\ell}\mu_i\xi_i.$$

Выпишем условия Куна-Таккера:

$$\nabla_w L = w - \sum_{i=1}^{\ell}\lambda_i y_i x_i = 0 \quad \Longrightarrow \quad w = \sum_{i=1}^{\ell}\lambda_i y_i x_i \tag{1.6}$$

$$\nabla_b L = -\sum_{i=1}^{\ell}\lambda_i y_i = 0 \quad \Longrightarrow \quad \sum_{i=1}^{\ell}\lambda_i y_i = 0 \tag{1.7}$$

$$\nabla_{\xi_i} L = C - \lambda_i - \mu_i \quad \Longrightarrow \quad \lambda_i + \mu_i = C \tag{1.8}$$

$$\lambda_i\left[y_i\left(\langle w,x_i\rangle + b\right) - 1 + \xi_i\right] = 0 \quad \Longrightarrow \quad (\lambda_i = 0) \text{ или } \left(y_i\left(\langle w,x_i\rangle + b\right) = 1 - \xi_i\right) \tag{1.9}$$

$$\mu_i\xi_i = 0 \quad \Longrightarrow \quad (\mu_i = 0) \text{ или } (\xi_i = 0) \tag{1.10}$$

$$\xi_i \geqslant 0, \lambda_i \geqslant 0, \mu_i \geqslant 0. \tag{1.11}$$

Проанализируем полученные условия. Из (1.6) следует, что вектор весов, полученный в результате настройки SVM, можно записать как линейную комбинацию объектов, причем веса в этой линейной комбинации можно найти как решение двойственной задачи. В зависимости от значений ξ_i и λ_i объекты x_i разбиваются на три категории:

1. $\xi_i = 0$, $\lambda_i = 0$.
   Такие объекты не влияют решение w (входят в него с нулевым весом λ_i), правильно классифицируются ($\xi_i = 0$) и лежат вне разделяющей полосы. Объекты этой категории называются *периферийными*.

2. $\xi_i = 0$, $0 < \lambda_i < C$.
   Из условия (1.9) следует, что $y_i\left(\langle w,x_i\rangle + b\right) = 1$, то есть объект лежит строго на границе разделяющей полосы. Поскольку $\lambda_i > 0$, объект влияет на решение w. Объекты этой категории называются *опорными граничными*.

3. $\xi_i > 0$, $\lambda_i = C$.
   Такие объекты могут лежать внутри разделяющей полосы ($0 < \xi_i < 2$) или выходить за ее пределы ($\xi_i \geqslant 2$). При этом если $0 < \xi_i < 1$, то объект классифицируется правильно, в противном случае — неправильно. Объекты этой категории называются *опорными нарушителями*.

Отметим, что варианта $\xi_i > 0$, $\lambda_i < C$ быть не может, поскольку при $\xi_i > 0$ из условия дополняющей нежесткости (1.10) следует, что $\mu_i = 0$, и отсюда из уравнения (1.8) получаем, что $\lambda_i = C$.

Итак, итоговый классификатор зависит только от объектов, лежащих на границе разделяющей полосы, и от объектов-нарушителей (с $\xi_i > 0$).

Построим двойственную функцию. Для этого подставим выражение (1.6) в лагранжиан, и воспользуемся уравнениями (1.7) и (1.8) (данные три уравнения выполнены для точки минимума лагранжиана при любых фиксированных λ и μ):

$$L = \frac{1}{2}\left\|\sum_{i=1}^{\ell}\lambda_i y_i x_i\right\|^2 - \sum_{i,j=1}^{\ell}\lambda_i\lambda_j y_i y_j\langle x_i, x_j\rangle - b\underbrace{\sum_{i=1}^{\ell}\lambda_i y_i}_{0} + \sum_{i=1}^{\ell}\lambda_i + \sum_{i=1}^{\ell}\xi_i\underbrace{(C-\lambda_i-\mu_i)}_{0}$$

$$= \sum_{i=1}^{\ell}\lambda_i - \frac{1}{2}\sum_{i,j=1}^{\ell}\lambda_i\lambda_j y_i y_j\langle x_i, x_j\rangle.$$

Мы должны потребовать выполнения условий (1.7) и (1.8) (если они не выполнены, то двойственная функция обращается в минус бесконечность), а также неотрицательность двойственных переменных $\lambda_i \geqslant 0$, $\mu_i \geqslant 0$. Ограничение на μ_i и условие (1.8), можно объединить, получив $\lambda_i \leqslant C$. Приходим к следующей двойственной задаче:

$$\begin{cases} \displaystyle\sum_{i=1}^{\ell}\lambda_i - \frac{1}{2}\sum_{i,j=1}^{\ell}\lambda_i\lambda_j y_i y_j\langle x_i, x_j\rangle \to \max\limits_{\lambda} \\ 0 \leqslant \lambda_i \leqslant C, \quad i = 1, \ldots, \ell, \\ \displaystyle\sum_{i=1}^{\ell}\lambda_i y_i = 0. \end{cases} \tag{1.12}$$

Она также является вогнутой, квадратичной и имеет единственный максимум.

Вернемся к тому, какое представление классификатора дает двойственная задача. Из уравнения (1.6) следует, что вектор весов w можно представить как линейную комбинацию объектов из обучающей выборки. Подставляя это представление w в классификатор, получаем

$$a(x) = \operatorname{sign}\left(\sum_{i=1}^{\ell}\lambda_i y_i\langle x_i, x\rangle + b\right). \tag{1.13}$$

Таким образом, классификатор измеряет сходство нового объекта с объектами из обучения, вычисляя скалярное произведение между ними.

В представлении (1.13) фигурирует переменная b, которая не находится непосредственно в двойственной задаче. Однако ее легко восстановить по любому граничному опорному объекту x_i, для которого выполнено $\xi_i = 0, 0 < \lambda_i < C$. Для него выполнено $y_i\left(\langle w, x_i\rangle + b\right) = 1$, откуда получаем

$$b = y_i - \langle w, x_i\rangle.$$

Как правило, для численной устойчивости берут медиану данной величины по всем граничным опорным объектам:

$$b = \operatorname{med}\{y_i - \langle w, x_i\rangle \mid \xi_i = 0, 0 < \lambda_i < C\}.$$

2 Логистическая регрессия

§2.1 Оценивание вероятностей

Метод обучения, которые получается при использовании логистической функции потерь, называется логистической регрессией. Основным его свойством является тот факт, что он корректно оценивает вероятность принадлежности объекта к каждому из классов.

Пусть в каждой точке пространства объектов $x \in \mathbb{X}$ задана вероятность $p(y = +1 \mid x)$ того, что объект x будет принадлежать классу $+1$. Это означает, что мы допускаем наличие в выборке нескольких объектов с одинаковым признаковым описанием, но с разными значениями целевой переменной; причём если устремить количество объекта x в выборке к бесконечности, то доля положительных объектов среди них будет стремиться к $p(y = +1 \mid x)$.

Примером может служить задача предсказания кликов по рекламным баннерам. При посещении одного и того же сайта один и тот же пользователь может как кликнуть, так и не кликнуть по одному и тому же баннеру, из-за чего в выборке могут появиться одинаковые объекты с разными ответами. При этом важно, чтобы классификатор предсказывал именно вероятности классов — если домножить вероятность первого класса на сумму, которую заплатит заказчик в случае клика, то мы получим матожидание прибыли при показе этого баннера. На основе таких матожиданий можно построить алгоритм, выбирающий баннеры для показа пользователю.

Итак, рассмотрим точку x пространства объектов. Как мы договорились, в ней имеется распределение на ответах $p(y = +1 \mid x)$. Допустим, алгоритм $b(x)$ возвращает числа из отрезка $[0, 1]$. Наша задача — выбрать для него такую процедуру обучения, что в точке x ему будет оптимально выдавать число $p(y = +1 \mid x)$. Если в выборке объект x встречается n раз с ответами $\{y_1, \ldots, y_n\}$, то получаем следующее требование:

$$\arg\min_{b \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^{n} L(y_i, b) \approx p(y = +1 \mid x).$$

При стремлении n к бесконечности получим, что функционал стремится к матожиданию ошибки:

$$\arg\min_{b \in \mathbb{R}} \mathbb{E}\left[L(y, b) \mid x\right] = p(y = +1 \mid x).$$

На семинаре будет показано, что этим свойством обладает, например, квадратичная функция потерь $L(y, z) = (y - z)^2$, если в ней для положительных объектов использовать истинную метку $y = 1$, а для отрицательных брать $y = 0$.

Примером функции потерь, которая не позволяет оценивать вероятности, является модуль отклонения $L(y, x) = |y - z|$. Можно показать, что с точки зрения данной функции оптимальным ответом всегда будет либо ноль, либо единица.

Это требование можно воспринимать более просто. Пусть один и тот же объект встречается в выборке 1000 раз, из которых 100 раз он относится к классу $+1$, и 900 раз — к классу -1. Поскольку это один и тот же объект, классификатор должен выдавать один ответ для каждого из тысячи случаев. Можно оценить матожидание

функции потерь в данной точке по 1000 примеров при прогнозе b:

$$\mathbb{E}\Big[L(y,b)\,|\,x\Big] \approx \frac{100}{1000}L(1,b) + \frac{900}{1000}L(-1,b).$$

Наше требование, по сути, означает, что оптимальный ответ с точки зрения этой оценки должен быть равен $1/10$:

$$\arg\min_{b\in\mathbb{R}}\left(\frac{100}{1000}L(1,b) + \frac{900}{1000}L(-1,b)\right) = \frac{1}{10}.$$

Задача 2.1. *Покажите, что квадратичная функция потерь $L(y,z) = (y-z)^2$ позволяет предсказывать корректные вероятности.*

Решение. Заметим, что поскольку алгоритм возвращает числа от 0 до 1, то его ответ должен быть близок к единице, если объект относится к положительному классу, и к нулю — если объект относится к отрицательному классу.

Запишем матожидание функции потерь в точке x:

$$\mathbb{E}\Big[L(y,b)\,|\,x\Big] = p(y=1\,|\,x)(b-1)^2 + (1-p(y=1\,|\,x))(b-0)^2.$$

Продифференцируем по b:

$$\begin{aligned}\frac{\partial}{\partial b}\mathbb{E}\Big[L(y,b)\,|\,x\Big] &= 2p(y=1\,|\,x)(b-1) + 2(1-p(y=1\,|\,x))b = \\ &= 2b - 2p(y=1\,|\,x) = 0.\end{aligned}$$

Легко видеть, что оптимальный ответ алгоритма действительно равен вероятности:

$$b = p(y=1\,|\,x).$$

■

§2.2 Правдоподобие и логистические потери

Хотя квадратичная функция потерь и приводит к корректному оцениванию вероятностей, она не очень хорошо подходит для решения задачи классификации. Причиной этому в том числе являются и слишком низкие штрафы за ошибку — так, если объект положительный, а модель выдаёт для него вероятность первого класса $b(x) = 0$, то штраф за это равен всего лишь единице: $(1-0)^2 = 1$.

Попробуем сконструировать функцию потерь из других соображений. Если алгоритм $b(x) \in [0,1]$ действительно выдает вероятности, то они должны согласовываться с выборкой. С точки зрения алгоритма вероятность того, что в выборке встретится объект x_i с классом y_i, равна $b(x_i)^{[y_i=+1]}(1-b(x_i))^{[y_i=-1]}$. Исходя из этого, можно записать правдоподобие выборки (т.е. вероятность получить такую выборку с точки зрения алгоритма):

$$(a,X) = \prod_{i=1}^{\ell} b(x_i)^{[y_i=+1]}(1-b(x_i))^{[y_i=-1]}.$$

Данное правдоподобие можно использовать как функционал для обучения алгоритма — с той лишь оговоркой, что удобнее оптимизировать его логарифм:

$$-\sum_{i=1}^{\ell}\left([y_i=+1]\log b(x_i)+[y_i=-1]\log(1-b(x_i))\right)\to\min$$

Данная функция потерь называется логарифмической (log-loss). Покажем, что она также позволяет корректно предсказывать вероятности. Запишем матожидание функции потерь в точке x:

$$\mathbb{E}\Big[L(y,b)\,|\,x\Big]=\mathbb{E}\Big[-[y=+1]\log b-[y=-1]\log(1-b)\,|\,x\Big]=$$
$$=-p(y=+1\,|\,x)\log b-(1-p(y=+1\,|\,x))\log(1-b).$$

Продифференцируем по b:

$$\frac{\partial}{\partial b}\mathbb{E}\Big[L(y,b)\,|\,x\Big]=-\frac{p(y=+1\,|\,x)}{b}+\frac{1-p(y=+1\,|\,x)}{1-b}=0.$$

Легко видеть, что оптимальный ответ алгоритма равен вероятности положительного класса:

$$b_*=p(y=+1\,|\,x).$$

§2.3 Логистическая регрессия

Везде выше мы требовали, чтобы алгоритм $b(x)$ возвращал числа из отрезка $[0,1]$. Этого легко достичь, если положить $b(x)=\sigma(\langle w,x\rangle)$, где в качестве σ может выступать любая монотонно неубывающая функция с областью значений $[0,1]$. Мы будем использовать сигмоидную функцию: $\sigma(z)=\frac{1}{1+\exp(-z)}$. Таким образом, чем больше скалярное произведение $\langle w,x\rangle$, тем больше будет предсказанная вероятность. Как при этом можно интерпретировать данное скалярное произведение? Чтобы ответить на этот вопрос, преобразуем уравнение

$$p(y=1\,|\,x)=\frac{1}{1+\exp(-\langle w,x\rangle)}.$$

Выражая из него скалярное произведение, получим

$$\langle w,x\rangle=\log\frac{p(y=+1\,|\,x)}{p(y=-1\,|\,x)}.$$

Получим, что скалярное произведение будет равно логарифму отношения вероятностей классов (log-odds).

Как уже упоминалось выше, при использовании квадратичной функции потерь алгоритм будет пытаться предсказывать вероятности, но данная функция потерь является далеко не самой лучшей, поскольку слабо штрафует за грубые ошибки. Логарифмическая функция потерь подходит гораздо лучше, поскольку не позволяет алгоритму сильно ошибаться в вероятностях.

Подставим трансформированный ответ линейной модели в логарифмическую функцию потерь:

$$-\sum_{i=1}^{\ell}\left([y_i=+1]\log\frac{1}{1+\exp(-\langle w,x_i\rangle)}+[y_i=-1]\log\frac{\exp(-\langle w,x_i\rangle)}{1+\exp(-\langle w,x_i\rangle)}\right)=$$

$$=-\sum_{i=1}^{\ell}\left([y_i=+1]\log\frac{1}{1+\exp(-\langle w,x_i\rangle)}+[y_i=-1]\log\frac{1}{1+\exp(\langle w,x_i\rangle)}\right)=$$

$$=\sum_{i=1}^{\ell}\log\left(1+\exp(-y_i\langle w,x_i\rangle)\right).$$

Полученная функция в точности представляет собой логистические потери, упомянутые в начале. Линейная модель классификации, настроенная путём минимизации данного функционала, называется логистической регрессией. Как видно из приведенных рассуждений, она оптимизирует правдоподобие выборки и дает корректные оценки вероятности принадлежности к положительному классу.

Список литературы

[1] *Mohri, M., Rostamizadeh, A., Talwalkar, A.* Foundations of Machine Learning. // MIT Press, 2012.

[2] *Bishop, C.M.* Pattern Recognition and Machine Learning. // Springer, 2006.

[3] *Crammer, K., Singer, Y.* On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines. // Journal of Machine Learning Research, 2:265-292, 2001.