1 Метод опорных векторов

Основная идея метода опорных векторов состоит, в построении линейного классификатора, задающего разделяющую поверхность, имеющую равный и максимальный отступ от обоих классов. Для реализации этой идеи, с учётом линейной неразделимости классов, необходимо решить задачу:

$$\min_{w,\xi\geq 0}\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \tag{1}$$
$$y_i(w^Tx_i + w_0) \geq 1 - \xi_i,\ i = 1\ldots n$$

Или, если учитывать неотрицательность ξ_i:

$$\min_{w}\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\max(0, 1 - y_i(w^Tx_i + w_0)) \tag{2}$$

Где C – константа регуляризации, x_i – i-ый объект из обучающей выборки, y_i – метка класса i-ого объекта, n – количество ообъектов в обучающей выборке.

Воспользовавшись теоремой Куна-Таккера можно свести данную задачу к эквивалентной:

$$\max_{\lambda}\sum_{i=1}^{n}\lambda_i - \frac{1}{2}\sum_{i,j=1}^{n}y_iy_j\lambda_i\lambda_j(x_i, x_j) \tag{3}$$
$$0 \leq \lambda_i \leq C,\ i = 1, \ldots, n$$
$$\sum_{i=1}^{n}\lambda_iy_i = 0$$

Проблема заключается в том, что гиперплоскость часто не в может правильно разделить выборку, поэтому можно попробовать преобразовать признаковое пространство для получения хорошо разделимой выборки. Не трудно видеть, что для решения задачи 3 достаточно уметь считать скалярное произведение между преобразованными векторами. Скалярное произведение можно задать как ядерную функцию $K(x, y) = (\varphi(x), \varphi(y)$, где x, y вектора исходного пространства, φ некое преобразование на нём. В данных терминах задача 3 преобразовывается в:

$$\max_{\lambda}\sum_{i=1}^{n}\lambda_i - \frac{1}{2}\sum_{i,j=1}^{n}y_iy_j\lambda_i\lambda_jK(x_i, x_j) \tag{4}$$
$$0 \leq \lambda_i \leq C,\ i = 1, \ldots, n$$
$$\sum_{i=1}^{n}\lambda_iy_i = 0$$

2 Реализация метода опорных векторов

2.1 Метод субградиентного спуска

2.1.1 Метод градиентного спуска

Метод градиентного спуска решает задачу поиска минимума функции, путём итерационного построения решения. На шаге 1 берётся первое приближение решения – w_0. Пусть далее на шаге n построено решение w_n, тогда на шаге $n+1$ будет построено решение:

$$w_{n+1} = w_n - \eta_n \nabla_w Q(w) \tag{5}$$

2.1.2 Сведение метода градиентного спуска к методу субградиентного спуска

Идея метода градиентного спуска состоит в том, чтобы искать минимум функции "спускаясь"по направлению антиградиента. К сожалению функция из 2 не является дифферинцируемой всюду, а значит не имеет градиента в некоторых точках. Для этого вводится понятие субградиента:

Определение 1 *Вектор $g \in \mathbf{R}^n$ называется субградиентом функции $f : \mathbf{R}^n \longrightarrow \mathbf{R}$ в точке x, если $\forall z \in \mathbf{R}^n$ выполняется неравенство:*

$$f(z) \geq f(x) + g^T(z - x) \tag{6}$$

Замечание 1.1 *Если в точке x_0 $\exists$ градиент функции f, то субградиент функции f в точке x_0 совпадает с градиентом функции f в этой точке.*

Согласно 1.1, достаточно найти субградиент функции в точках, где эта функция не имеет градиент. Для функции из 2 – это все такие, и только такие точки, для которых $1 - y_i(w^T x_i + w_0) = 0,\ i = 1 \dots n$. Для удобства допишем каждому объекту X_i слева 1. Для такой выборки имеем задачу:

$$\min_w \frac{1}{2}\|w\| + C\sum_{i=1}^{n} \max(0, 1 - y_i w^T x_i) \tag{7}$$

При этом w_0 оказывается «забит» в w. Докажем следущую теорему:

Теорема 1 *Если у функций f_1 и f_2 существует субградиент в точке x_0, то у их суммы также существует субградиент в точке x_0 причём:*

$$\nabla^s f1 + \nabla^s f2 \subseteq \nabla^s (f1 + f2) \tag{8}$$

Доказательство.

$$\left.\begin{matrix} f_1(z) \geq f_1(x) + g_1^T(z - x) \\ f_2(z) \geq f_2(x) + g_2^T(z - x) \end{matrix}\right\} \Rightarrow (f_1 + f_2)(z) \geq (f_1 + f_2)(x) + (g_1 + g_2)^T(z - x)$$

■

Заметим, что в методе градиентного спуска градиент функции в точке определён однозначно, однако того же нельзя всегда сказать о субградиенте функции. Поэтому в методе субградиентного спуска, в случаях, где сохранена неоднозначность, мы будем брать только один вектор из субградиента. На основании этого, а также на основании теоремы 1, замечания 1.1 и общего вида функции из задачи 2 можно сделать вывод о том, что достаточно найти хотя бы один вектор, удовлетворяющий неравенству 6 для функции $f = 1 - y_i w_i^T x_i$ в точке w такой, что $1 - y_i w^T x_i = 0$. Найдём такой вектор:

$$\begin{gathered} 1 - y_i z^T x_i \geq 1 - y_i w^T x_i + g^T(z - w) \\ 1 - y_i z^T x_i \geq g^T z - g^T w \end{gathered} \tag{9}$$

Нетрудно видеть, что $g = -y_i x_i$ удовлетворяет неравенству 9. На основании вышеизложенного запишем формулу субградиента целевой функции $Q(w)$ для метода субградиентного спуска. Введём γ_i такую, что:

$$\gamma_i(w) = \begin{cases} 0, & \text{если } 1 - y_i w^T x_i < 0; \\ -y_i x_i, & \text{иначе.} \end{cases}$$

Тогда субградиент целевой функции $Q(w)$ равняется:

$$\nabla^s Q(w) = w + C \sum_{i=1}^{n} \gamma_i(w) \tag{10}$$

Осталось только заменить градиент в методе грдиентного спуска (5) на подсчитанный выше субградиент (10).

2.2 Метод стохастического субградиентного спуска

Идея метода стохастического субградиента в том, чтобы считать субградиент не по всей выборке, а только по какой-либо её части, которая генерируется на каждой итерации случайно.

2.3 Методы внутренней точки

2.3.1 Введение

Для реализации метода внутренней точки была использована функция $cvxopt.solvers.qp$, решающая следущую задачу:

$$\begin{gathered} \min_x \frac{1}{2} x^T P x + q^T x \\ Gx \preceq h \\ Ax = b \end{gathered} \tag{11}$$

Где x – вектор столбец.

2.3.2 Метод внутренней точки для прямой задачи

Необходимо свести задачу 1 к задаче 11.

$$x = \begin{pmatrix} w_0 & w_1 & \dots & w_l & \xi_1 & \xi_2 & \dots & \xi_n \end{pmatrix}^T$$

$$P = \begin{array}{c} \begin{array}{cccccccc} & & & & l+1 & & & \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0 & 0 & \dots & 0 \end{pmatrix} \end{array}$$

$$q = \begin{array}{c} \\ \\ \\ l+2 \\ \\ \\ \end{array} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ C \\ \vdots \\ C \end{pmatrix}$$

$$G = \begin{pmatrix} -y_1 & -1 & -x_1^1 & -x_1^2 & \dots & -x_1^l & -1 & -1 & \dots & -1 \\ -y_2 & -1 & -x_2^1 & -x_2^2 & \dots & -x_2^l & -1 & -1 & \dots & -1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ -y_n & -1 & -x_n^1 & -x_n^2 & \dots & -x_n^l & -1 & -1 & \dots & -1 \\ 0 & 0 & 0 & 0 & \dots & 0 & -1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & -1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & -1 \end{pmatrix}$$

$$h = \begin{array}{c} \\ \\ \\ \\ l+2 \\ \\ \\ \end{array} \begin{pmatrix} -1 \\ -1 \\ \vdots \\ -1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

$$A, B = None$$

Очевидно, что при данных значениях параметров, функция 2.3.1 решает задачу 1. При этом нам нужны только w и w_0. Поэтому данный подход не слишком эффективен.

2.3.3 Метод внутренней точки для двойственной задачи с линейным ядром

Решаем задачу 3. Сведём эту задачу к задаче 11.

$$x = \begin{pmatrix} \lambda_1 & \lambda_2 & \dots & \lambda_n \end{pmatrix}^T$$

$$P = \begin{pmatrix} y_1 y_1 (x_1, x_1) & y_1 y_2 (x_1, x_2) & \dots & y_1 y_n (x_1, x_n) \\ y_2 y_1 (x_2, x_1) & y_2 y_2 (x_2, x_2) & \dots & y_2 y_n (x_2, x_n) \\ \dots & \dots & \dots & \dots \\ y_n y_1 (x_n, x_1) & y_n y_2 (x_n, x_2) & \dots & y_n y_n (x_n, x_n) \end{pmatrix}$$

$$q = \begin{pmatrix} -1 & -1 & \dots & -1 \end{pmatrix}^T$$

$$G = \begin{pmatrix} -1 & 0 & 0 & \dots & 0 \\ 0 & -1 & 0 & \dots & 0 \\ 0 & 0 & -1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & -1 \\ 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix}$$

$$h = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ C \\ C \\ \vdots \\ C \end{pmatrix}$$

$$A = \begin{pmatrix} y_1 & y_2 & \dots & y_n \end{pmatrix}$$

$$b = \begin{pmatrix} 0 \end{pmatrix}$$

При данных параметрах функция 11 решает задачу 3.

2.3.4 Метод внутренней точки для двойственной задачи с ядром rbf

Будем решать задачу 4 с rbf ядром. Это значит, что функция $K(x, y)$ в задаче 4 примет значение $e^{-\frac{1}{2\gamma}\|x-y\|^2}$. Для того, чтобы свести задачу 4 к задаче 11 достаточно взять такие же параметры, что и для случая с линейным ядром, но в матрице P поменять скалярное произведение (x_i, x_j) на значение rbf-ядра в этих точках.

2.4 Методы реализованные в библиотеках liblin и libsvm.

При решении задачи 1 и задачи 4 с rbf ядром использовались методы из классов $LinearSVC$ и SVC соответственно, взятые из библиотеки $sklearn.svm$

2.5 Описание класса SVM

Для реализации описанных выше методов был создан класс SVM. Параметры инициализации:

C – константа регуляризаци(число), по умолчанию 1.

method – название метода решения задачи SVM (строка). Принимает значения: «$subgradient$», «$stoch\_subgradient$», «$primal$», «$dual$», «$liblinear$», «$libsvm$».

kernel – название ядра (строка). Только если $method =$«$dual$» или «$libsvm$». Принимает значения «$linear$» или «rbf»

gamma – параметр для rbf ядра (число). Только если $kernel =$«rbf».

Методы класса:

fit(tol=0.001, max_iter=10000, verbose=$False$, stop_criterion=«$objective$», batch_size=50, lamb=0.5, alpha=1, beta=1)

predict(X_test, return_classes=$True$) – возвращает вектор с метками классов для соответствующих объектов, если $return\_classes = True$, и значение целевой функции (если $method \neq$ «$libsvm$» или «$liblinear$»)

compute_support_vectors(X) – возвращает опорные вектора (только если решалась двойственная задача)

compute_w(X, y) – если решалась двойственная задача находит вектор w и отступ w_0 для прямой задачи.

Атрибуты класса:

w – вектор $l \times 1$ коэффициентов разделяющей гиперплоскости.

w_0 – число, смещение разделяющей гиперплоскости.

A – вектор $n \times 1$ двойственных коэффициентов.

В двумерном случае возможна визуализация выборки при помощи функции $visualize(X, y, alg\_svm, show\_vectors = False)$.