Курс: Методы оптимизации в машинном обучении, осень 2015

Задание 2. Методы оптимизации для L_1-регуляризованной линейной регрессии

Начало выполнения задания: 23 ноября
Срок сдачи: **13 декабря, 23:59.**
Среда для выполнения задания – PYTHON.

Содержание

Модель L_1-регуляризованной линейной регрессии **1**

Формулировка задания **2**

Оформление задания **2**

Модель L_1-регуляризованной линейной регрессии

Рассматривается задача восстановления регрессии. Имеется обучающая выборка $(X, \boldsymbol{t}) = \{\boldsymbol{x}_n, t_n\}_{n=1}^N$, где $\boldsymbol{x}_n \in \mathbb{R}^D$ – вектор признаков для объекта n, а $t_n \in \mathbb{R}$ – его регрессионное значение. Задача заключается в прогнозировании регрессионного значения t_{new} для нового объекта, представленного своим вектором признаков $\boldsymbol{x}_{new}$. В линейной регрессии прогнозирование осуществляется с помощью линейной функции:

$$t(\boldsymbol{x}_{new}) = \sum_{d=1}^D w_d x_{new,d} = \boldsymbol{w}^T \boldsymbol{x}_{new},$$

где $\boldsymbol{w} \in \mathbb{R}^D$ – некоторые веса. Настройка весов осуществляется путем минимизации следующего регуляризованного критерия:

$$J = \frac{1}{2}\sum_{n=1}^N (t_n - \boldsymbol{w}^T\boldsymbol{x}_n)^2 + \lambda \sum_{d=1}^D |w_d| = \frac{1}{2}\|\boldsymbol{t} - X\boldsymbol{w}\|^2 + \lambda\|\boldsymbol{w}\|_1 \to \min_{\boldsymbol{w}}. \quad (1)$$

Здесь $\lambda \geq 0$ – задаваемый пользователем параметр регуляризации. Использование L_1-регуляризации позволяет, во-первых, снизить вероятность переобучения алгоритма, а во-вторых, получить т.н. разреженное решение. В разреженном решении часть компонент оптимального вектора весов $\boldsymbol{w}$ равно нулю (можно показать, что при $\lambda \geq \|X^T\boldsymbol{t}\|_\infty$ все веса будут равны нулю). Нулевые веса для некоторых признаков равносильны их исключению из модели (признание их неинформативными).

Рассмотрим получение двойственной задачи оптимизации для задачи (1). Для этого рассмотрим эквивалентную постановку данной задачи в виде задачи условной оптимизации:

$$\begin{aligned} &\frac{1}{2}\boldsymbol{z}^T\boldsymbol{z} + \lambda\|\boldsymbol{w}\|_1 \to \min_{\boldsymbol{z},\boldsymbol{w}}, \\ &X\boldsymbol{w} - \boldsymbol{t} = \boldsymbol{z}. \end{aligned} \quad (2)$$

Далее можно показать, что двойственной к данной задаче выпуклой оптимизации будет следующая:

$$\begin{aligned} &-\frac{1}{2}\boldsymbol{\mu}^T\boldsymbol{\mu} - \boldsymbol{\mu}^T\boldsymbol{t} \to \max_{\boldsymbol{\mu}}, \\ &\|X^T\boldsymbol{\mu}\|_\infty \leq \lambda. \end{aligned} \quad (3)$$

Здесь под $\|\boldsymbol{y}\|_\infty$ понимается $\max(|y_1|, \ldots, |y_D|)$.

Для произвольных весов $\boldsymbol{w} \in \mathbb{R}^D$ можно найти допустимую двойственную точку как

$$\mu = \frac{\lambda(X\boldsymbol{w} - \boldsymbol{t})}{\|X^T(X\boldsymbol{w} - \boldsymbol{t})\|_\infty}.$$

В результате находим оценку сверху на зазор между решениями прямой задачи (1) и двойственной (3):

$$\eta = \frac{1}{2}\|\boldsymbol{t} - X\boldsymbol{w}\|^2 + \lambda\|\boldsymbol{w}\|_1 + \frac{1}{2}\boldsymbol{\mu}^T\boldsymbol{\mu} + \boldsymbol{\mu}^T\boldsymbol{t}. \tag{4}$$

Задачу выпуклой негладкой оптимизации (1) можно также эквивалентно представить в виде задачи гладкой условной минимизации:

$$\begin{aligned} &\frac{1}{2}\|\boldsymbol{t} - X\boldsymbol{w}\|^2 + \lambda\sum_{d=1}^{D} u_d \to \min_{\boldsymbol{w},\boldsymbol{u}}, \\ &-u_d \le w_d \le u_d,\ \forall d = 1, \ldots, D. \end{aligned} \tag{5}$$

Формулировка задания

1. Вывести двойственную задачу оптимизации (3); записать, как по решению двойственной задачи (3) можно получить решение прямой задачи (1);
2. Вывести необходимые формулы для решения гладкой задачи обучения (5) с помощью прямого метода барьерных функций; предложить свой способ эффективного решения соответствующей СЛАУ, необходимые для этого формулы вставить в отчёт;
3. Реализовать прямой метод барьерных функций для решения задачи (5); в качестве критерия останова использовать верхнюю оценку на зазор (4);
4. Вывести необходимые формулы для решения задачи (3) с помощью прямо-двойственного метода внутренней точки; предложить свой способ эффективного решения возмущенной ККТ-системы, необходимые для этого формулы вставить в отчёт;
5. Реализовать прямо-двойственный метод внутренней точки для задачи (3);
6. Реализовать проксимальный метод для решения задачи обучения (1);
7. Провести экспериментальное сравнение трёх реализованных методов (метод барьеров для задачи (5), прямо-двойственный метод внутренней точки для задачи (3) и проксимальный метод для задачи (1)) на предмет скорости работы при 1) различных соотношениях между количеством объектов и признаков в данных и 2) различных значениях параметра точности оптимизации (рассмотреть случай высокой точности, например, $\varepsilon = 10^{-10}$ и низкой точности, например, $\varepsilon = 10^{-2}$);
8. Написать отчёт в формате PDF с описанием всех проведённых исследований. Данный отчёт должен содержать, в частности, необходимые формулы для всех методов.

Оформление задания

Выполненное задание следует отправить письмом по адресу *bayesml@gmail.com* с заголовком письма

«[МОМО15] Задание 2, Фамилия Имя».

Убедительная просьба присылать выполненное задание только один раз с окончательным вариантом. Также убедительная просьба строго придерживаться заданных ниже прототипов реализуемых функций.

Все реализованные методы оптимизации должны располагаться в одном модуле под названием *l1linreg* с функциями *barrier*, *pd* и *prox*.

Таблица 1: Метод логарифмических барьеров

w = barrier(X, t, reg_coef, max_iter = 100, max_inner_iter = 20, tol_gap = 1e-5, tol_center = 1e-10, tau_params = np.array([1, 10]), bt_params = np.array([1e-4, 0.8]), display = 0)
ВХОД X – обучающая выборка, numpy.array, матрица размера NxD; t – регрессионные значения, numpy.array, вектор длины N; reg_coef – коэффициент регуляризации, число; ’max_iter’ – максимальное число внешних итераций; ’max_inner_iter’ – максимальное число внутренних итераций для центрирования; ’tol_gap’ – точность оптимизации по зазору (4); ’tol_center’ – точность центрирования; ’tau_params’ – стратегия увеличения параметра центрирования τ, вектор из двух чисел $[\tau_{start}, \nu]$; ’bt_params’ – параметры для стратегии backtracking при поиске длины шага, вектор из двух чисел $[c_1, \beta]$; ’display’ – режим отображения, 0 или 1, если 1, то отображаются текущие показатели метода на каждой итерации;
ВЫХОД w – найденные веса, numpy.array, вектор длины D.

Таблица 2: Прямо-двойственный метод

w, mu = pd(X, t, reg_coef, max_iter = 100, tol_feas = 1e-10, tol_gap = 1e-5, tau_param = 10, bt_params = np.array([1e-4, 0.8]), display = 0)
ВХОД X – обучающая выборка, numpy.array, матрица размера NxD; t – регрессионные значения, numpy.array, вектор длины N; reg_coef – коэффициент регуляризации; ’max_iter’ – максимальное число прямо-двойственных итераций; ’tol_feas’ – точность по двойственной невязке; ’tol_gap’ – точность по суррогатному зазору; ’tau_param’ – коэффициент увеличения параметра центрирования τ; ’bt_params’ – параметры для стратегии backtracking при поиске длины шага, вектор из двух чисел $[c_1, \beta]$; ’display’ – режим отображения, 0 или 1, если 1, то отображаются текущие показатели метода на каждой итерации;
ВЫХОД w – найденные веса, numpy.array, вектор длины D; mu – найденное решение двойственной задачи (3), numpy.array, вектор длины N.

Аналогично выглядят прототипы функции *prox* для проксимального метода.