ВМК

Задание 3. Метод опорных векторов

Курс: Практикум на ЭВМ, осень 2016

Начало выполнения задания: 3 ноября.
Мягкий дедлайн: **14 декабря, 23:59.**
Жёсткий дедлайн: **28 декабря, 23:59.**

Формулировка задания

Данное задание направлено на ознакомление с методом опорных векторов и различными методами оптимизации для его обучения.

В задании необходимо:

1. Написать на языке Python собственные реализации различных процедур обучения метода опорных векторов. Прототипы функций должны строго соответствовать прототипам, описанным в спецификации и проходить все выданные тесты. Задание, не проходящее все выданные тесты, приравнивается к невыполненному. При написании необходимо пользоваться стандартными средствами языка Python, библиотеками numpy, scipy, cvxopt и matplotlib. Библиотекой scikit-learn пользоваться запрещается, если это не обговорено отдельно в пункте задания.

2. Вывести все необходимые формулы, привести выкладки в отчёте.

3. Провести описанные ниже эксперименты с модельными данными.

4. Написать отчёт о проделанной работе (формат PDF). Отчёт должен быть подготовлен в системе LaTeX.

Прямая задача SVM

Рассмотрим задачу бинарной классификации. Пусть дана обучающая выборка $X = (x_i, y_i)_{i=1}^{l}$, где $x_i \in \mathbb{R}^d$, $y_i \in \mathbb{Y} = \{1, -1\}$. Линейная модель классификации определяется следующим образом:

$$a(x) = \operatorname{sign}\left(\langle w, x\rangle + w_0\right), \quad \text{где } w \in \mathbb{R}^d \text{ — вектор весов, } w_0 \text{ — сдвиг.}$$

Уравнение $\langle w, x_i\rangle = -w_0$ описывает гиперплоскость, разделяющую классы в пространстве $\mathbb{R}^d$. Потребуем, чтобы разделяющая гиперплоскость правильно классифицировала все объекты обучающей выборки и максимально далеко отстояла от ближайших к ней точек обоих классов. Для линейно разделимой выборки задача оптимизации ставится так:

$$\begin{cases} \langle w, w\rangle \to \min\limits_{w, w_0} \\ y_i(\langle w, x_i\rangle + w_0) \geq 1, \quad i = 1, \dots, l \end{cases}$$

Задачу оптимизации можно обобщить на линейно неразделимый случай следующим образом:

$$\begin{cases} \langle w, w\rangle + C\sum\limits_{i=1}^{l}\xi_i \to \min\limits_{w, w_0, \xi} \\ y_i(\langle w, x_i\rangle + w_0) \geq 1 - \xi_i, \quad i = 1, \dots, l \\ \xi_i \geq 0, \quad i = 1, \dots, l \end{cases}$$

Записанную условную задачу оптимизацию будем называть *прямой задачей SVM*. Задача решается с помощью стадартных методов решения задач квадратичного программирования, например, с помощью метода внутренней точки. C — константа, которую необходимо подобрать по критерию скользящего контроля.

Двойственная задача SVM

Перейдём от прямой задачи SVM к двойственной:

$$\begin{cases} \sum\limits_{i=1}^{l}\lambda_i - \frac{1}{2}\sum\limits_{i=1}^{l}\sum\limits_{j=1}^{l}\lambda_i\lambda_j y_i y_j\langle x_i, x_j\rangle \to \max\limits_{\lambda} \\ 0 \le \lambda_i \le C, \quad i = 1, \dots, l \\ \sum\limits_{i=1}^{l}\lambda_i y_i = 0 \end{cases}$$

Решение исходной задачи w выражается через решение двойственной следующим образом:

$$w = \sum_{i=1}^{l}\lambda_i y_i x_i$$

$$w_0 = -\langle w, x_i\rangle + y_i$$

Нетрудно заметить, что на решение задачи оказывают влияние только те объекты x_i, для которых $\lambda_i \neq 0$. Такие объекты называются *опорными*, все остальные объекты называются *периферийными*. Записанную *двойственную задачу SVM* можно также решить с помощью стандартных методов решения задач квадратичного программирования.

Так как форма разделяющей поверхности — гиперплоскость, оба рассмотренных метода плохо работают с линейно неразделимыми объектами. Один из способов решения проблемы линейной неразделимости — переход в спрямляющее пространство H с помощью некоторого преобразования $\psi(x) : \mathbb{R}^d \to H$. Таким образом, скалярное произведение $\langle x_i, x_j\rangle$ в новом спрямляющем пространстве заменится на скалярное произведение $\langle \psi(x_i), \psi(x_j)\rangle = K(x_i, x_j)$. Более того, во многих случаях можно вообще не рассматривать функцию $\psi(x)$, а лишь задавать функцию ядра $K(x_i, x_j)$. Таким образом, двойственная задача в новом спрямляющем пространстве может быть сформулирована следующим образом:

$$\begin{cases} \sum\limits_{i=1}^{l}\lambda_i - \frac{1}{2}\sum\limits_{i=1}^{l}\sum\limits_{j=1}^{l}\lambda_i\lambda_j y_i y_j K(x_i, x_j) \to \max\limits_{\lambda} \\ 0 \le \lambda_i \le C, \quad i = 1, \dots, l \\ \sum\limits_{i=1}^{l}\lambda_i y_i = 0 \end{cases}$$

В данном практическом задании вам будет нужно использовать полиномиальное и rbf ядра:

$$K_{poly}(x_i, x_j) = (\langle x, x\rangle + 1)^d$$

$$K_{rbf}(x_i, x_j) = \exp(-\gamma ||x_i - x_j||^2)$$

Задача SVM без ограничений

Задачу SVM можно сформулировать без ограничений:

$$\langle w, w\rangle + C\sum_{i=1}^{l}\max\left(0, 1 - y_i(\langle w, x_i\rangle + w_0)\right) \to \min_{w, w_0}$$

Такую задачу оптимизации можно решить с помощью метода субградиентного спуска. Вектор $g \in \mathbb{R}^d$ является *субградиентом* выпуклой функции $f\colon \mathbb{R}^d \to \mathbb{R}$ в точке $x \in \mathbb{R}^d$, если $\forall z \in \mathbb{R}^d$ выполнено неравенство

$$f(z) \geqslant f(x) + \langle g, (z - x)\rangle.$$

Если функция f дифференцируема в точке x, ее субградиент в этой точке совпадает с градиентом. *Субдифференциалом* функции f в точке $\mathbf{x}$ называют множество субградиентов в этой точке, обозначают $\partial f(x)$. Субградиентный спуск — метод аналогичный методу градиентного спуска, в котором вместо градиента в точке используется какой-нибудь из субградиентов функции. Аналогично градиентному спуску, вместо вычисления субградиента по полной выборке на каждом шаге, можно вычислять субградиент по некоторой подвыборке. В отличие от градиентного метода, в субградиентном значение функции не обязано уменьшаться на каждой итерации, поэтому в качестве результата алгоритма должно выдаваться минимальное значение функционала

Есть и другие способы решения задачи без ограничений более эффективные по скорости чем субградиентный метод. Предположим, что первое слагаемое в задаче без ограничений содержит не только w, но и w_0. Это соответствует ситуации, когда в исходных данных есть константый признак и оптимизируется следующий функционал:

$$\langle w, w\rangle + C\sum_{i=1}^{l}\max\left(0, 1 - y_i(\langle w, x_i\rangle)\right) \to \min_w$$

Предложенный функционал предлагается оптимизировать с помощью алгортма PEGASOS:

Require: T — максимальное число итераций, λ, X, y;

Ensure: F_{best}

$w^{(1)} = 0, \quad w_0^{(1)} = 0, \quad F_{best} = F(w^{(1)})$

for $i = 1, \ldots, T$ **do**

  $I_k := Unif\{1, 2, \ldots l\}$ {индексы мини-батча размера B}

  $\alpha_k = (k\lambda)^{-1}$

  $w_{k+1} = (1 - \alpha_k\lambda)w_k + \frac{\alpha_k}{B}\sum_{i\in I_k}[y_i\langle w, x_i\rangle < 1]y_i x_i$

  $w_{k+1} = \min\left(1, \frac{1}{\sqrt{\lambda}\|w_{k+1}\|}\right)w_{k+1}$

  $F_{best} = \min(F_{best}, F(w_{k+1}))$

Требования реализации

Требуется реализовать следующие методы для решения задачи SVM (прототипы всех функций описаны в файлах, прилагающихся к заданию):

1. Метод внутренней точки для решения прямой задачи. Рекомендуется использовать библиотеку cvxopt, метод `cvxopt.solvers.qp`.

2. Метод внутренней точки для решения двойственной задачи, поддерживающий переход к rbf-ядру. Рекомендуется использовать библиотеку cvxopt, метод `cvxopt.solvers.qp`.

3. Метод субградиентного спуска для решения задачи без ограничений. Рекомендуется адаптировать код из 2 практического задания.

4. Метод стохастического субградиентного спуска для решения задачи без ограничений. Рекомендуется адаптировать код из 2 практического задания.

5. Метод PEGASOS для решения задачи без ограничений.

Дополнительно в бонусной части предлагается реализовать следующие методы:

1. Метод PEGASOS для решения задачи без ограничений, поддерживающий переход к ядрам.

2. Многоклассовый SVM.

Исследовательская часть.

Для проведения исследований необходимо генерировать модельные данные различной сложности. Обязательно необходимо рассматривать случаи:

- линейно разделимых данных

- хорошо разделимых данных, но линейно неразделимых

- плохо разделимых данных

- данные с несбалансированными классами, данные с выбросами

Требуется провести следующие исследования:

1. Исследовать зависимость времени работы реализованных различных реализаций (из обязательной части) линейного SVM от размерности признакового пространства и числа объектов в обучающей выборке. Исследовать скорость сходимости методов. Сравнить методы по полученным значениям целевой функции.

2. Провести эти исследования для случая SVM с полиномиальным и RBF ядрами для тех методов, где возможен ядровой переход.

3. Сравните точность различных подходов при решении задачи классификации. Проанализируйте, как зависит точность классификации от значения оптимизируемого функционала.

4. Реализовать процедуру поиска оптимального значения параметра C, ширины RBF ядра и степени полиномиального ядра с помощью кросс-валидации (можно воспользоваться библиотекой scikit-learn). Исследовать зависимость ошибки на валидационной выборке от значений этих параметров. Обязательно рассмотреть случаи хорошо и трудно разделимых выборок!

5. Сравнить (по скорости сходимости и точности решения) несколько стратегий выбора шага α_t в методе субградиентого спуска и стохастического субградиентного спуска: α, $\frac{\alpha}{t}$, $\frac{\alpha}{t^\beta}$, где α, β — некоторые константы, t — номер итерации. Сравнить рассмотренные стратегии с методом PEGASOS.

6. Исследовать, как размер подвыборки, по которой считается субградиент, в методе стохастического субградиентного спуска влияет на скорость сходимости метода и на точность решения. В этом и предыдущем пунктах за точное решение можно взять решение, полученное с помощью одного из методов внутренней точки.

7. Сравните результаты двух предыдущих экспериментов с аналогичными экспериментами из предыдущего практического задания, сделайте выводы. Дополнительно рассмотрите случай несбалансированных классов и классов с большим числом выбросов. Проанализируйте в каких случаях, какую модель (логистическую регрессию или SVM) следует использовать.

8. Для двумерного случая:

   - Визуализировать выборку.
   - Для линейного SVM для SVM с RBF ядром провести визуализацию разделяющей поверхности
   - Отобразить объекты, соответствующие опорным векторам.
   - Визуально показать, как зависит форма разделяющей поверхности для RBF ядра, от параметра γ

Бонусная часть

1. **(до 2 баллов)** Обработайте все некорректные вызовы в классе SVMSolver с помощью механизма исключений (например, вызовы функций для прямой задачи в случае, когда решается двойственная)

2. **(до 5 баллов)** Реализовать метод PEGASOS, поддерживающий переход к ядрам. Подробное описание метода можно найти в [1]. Сравнить предложенный метод по скорости и качеству с методами обязательной части, в которых возможен ядровой переход.

3. **(до 5 баллов)** Формально, в методе PEGASOS решается не традиционная задача SVM, из-за добавления в первое слагаемое вектора сдвига. Необходимо придумать, реализовать и протестировать другую стратегию нахождения вектора сдвига w_0 (в [1] рассмотрены несколько подходов), которая будет превосходить базовую по времени или качеству работы. Под качеством в этом эксперименте понимается какая-нибудь метрика классификации.

4. **(до 5 баллов)** Сгенерируйте несколько выборок с 2 признаками и 3 классами (достаточно 100 объектов каждого класса) на которых будете проводить эксперименты. Для этого можно воспользоваться функцией `make_blobs` из пакета `sklearn.datasets`.

   Рассмотрите четыре способа решения многоклассовых задач классификации линейными моделями: один против всех, каждый против каждого, мультиномиальная логистическая регрессия и многоклассовый SVM (можно прочитать в [2]). Сравните три способа из прошлого задания с многоклассовым SVM:

   - Укажите какие особенности, преимущества и недостатки с точки зрения построения разделяющих плоскостей, качества разделения классов и вычислительной эффективности характерны для многоклассового SVM.
   - Для каждой из стратегий подумайте над примерами ситуаций, когда стоит выбирать ее для решения задачи многоклассовой классификации. Рассмотрите выборки с несбалансированными классами и/или с большим числом выбросов.

5. **(до 3 баллов)** Примените многоклассовый SVM и мультиномиальную логистическую регрессию на датасете MNIST. Сравните качество, полученное с помощью линейных моделей, с качеством метрических методов. Проанализируйте результаты.

6. **(до 5 баллов)** Качественно проведите дополнительное (не пересекающееся с основным заданием) исследование по теме SVM: формулируется изучаемый вопрос, ставятся эксперименты, позволяющие на него ответить, делаются выводы. Перед исследованием необходимо обсудить тему с преподавателем.

Список литературы

[1] Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, Andrew Cotter — Pegasos: Primal Estimated sub-GrAdient SOlver for SVM

http://ttic.uchicago.edu/ nati/Publications/PegasosMPB.pdf

[2] Соколов Е.А. — Многоклассовая классификация и категориальные признаки

https://github.com/esokolov/ml-course-hse/blob/master/2017-fall/lecture-notes/lecture06-linclass.pdf