Линейные модели регрессии

Викулин Всеволод

v.vikulin@corp.mail.ru

18 марта 2019

Часть 1

Повторение

Обучение с учителем

Обучение — приобретение нужной функциональности посредством опыта. (https://postnauka.ru/video/53575)

Множество объектов X, множество допустимых ответов Y. Существует $y : X \to Y$
Для $\{x_1, \dots, x_N\} = X_{train}$ известны $\{y_1, \dots, x_N\} = y_{train}$
Построить $a : X \to Y$, которая приближала бы y.

Пример

Строим рекомендательную систему музыки. Что является объектом?

Обучение с учителем

Объекты описываются набором признаков. Признак объекта x — это результат измерения некоторой его характеристики.

В курсе разберем 2 постановки задачи обучения с учителем:

- Классификация – $Y = \{1, \ldots, M\}$
- Регрессия – $Y = \mathbb{R}$

Пример

Строим рекомендательную систему музыки. Какой берем Y? Какие берем признаки?

Минимизация эмпирического риска

Функция потерь (loss function) $L(a, x, y)$ – неотрицательная функция, показывающая величину ошибки алгоритма a на объекте x с ответом y.

Функционал качества $Q(a, X_{train}, y_{train}) = \frac{1}{N}\sum_{i=1}^{N} L(a, x_i, y_i), x_i \in X_{train}, y_i \in y_{train}$

Принцип минимизации эмпирического риска:

$a^* = \underset{a}{\operatorname{argmin}} Q(a, X_{train}, y_{train})$

Пример

Строим рекомендательную систему музыки. Какую берем функцию потерь?

Обобщающая способность

Не обязательно, что $\underset{a}{\operatorname{argmin}} Q(a, X_{train}, y_{train})$ – полезный алгоритм. Можно просто запомнить обучающаю выборку.

Проблема **переобучения** – значения $Q(a, X_{train}, y_{train})$ значительно меньше, чем значение $Q(a, X_{test}, y_{test})$ на **контрольной** выборке.

Если $Q(a, X_{test}, y_{test})$ примерно равна $Q(a, X_{train}, y_{train})$, то говорят, что алгоритм обладает **обобщающей способностью**

Параметризация модели

Функций $a(x)$, которые идеально описывают обучающую выборку бесконечно много, и обобщающей способности у таких алгоритмов не будет. Нужно сузить функциональное пространство перебора. Будем искомую функцию **параметризовать** – модель $a(x, w)$ описывается вектором весов w.
Тогда задача превращается в поиск весов:

$$w^* = \underset{w}{\operatorname{argmin}} Q(w, X_{train}, y_{train})$$

Параметрическая модель, предсказание которой **линейно** зависит от признаков, называется **линейной моделью**.

Формула обучения

Не забываем нашу любимую формулу:

$$Learning = Representation + Evaluation + Optimization$$

Читаем статью `https://homes.cs.washington.edu/~pedrod/papers/cacm12.pdf`
(1500 цитирований)

Знакомьтесь, линейная регрессия!

Представление линейной регрессии

Пусть объект описывается D признаками $f_1, f_2, \dots f_D$. Тогда модель:
$a(x, w) = w_0 + \sum_{j=1}^{D} f_j w_j$ называется **линейной моделью**, где w – D-мерный вектор признаков $w_1, w_2, \dots w_D$.
Далее будем считать, что в векторе признаков есть тождественно равный единице признак f_0, тогда формула упростится до:
Representation: $a(x, w) = \sum_{j=0}^{D} f_j w_j = \boldsymbol{x} \cdot \boldsymbol{w}$
Веса модели интерпретируемы. w_i - значение, на которое изменится предсказание, если признак f_i увеличить на единицу.

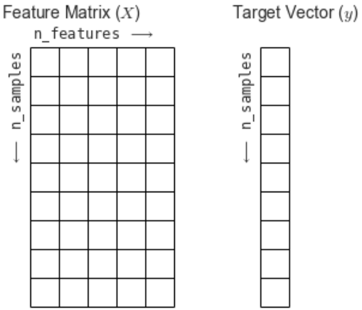
Функции потерь линейной регрессии

Evaluation.

- квадратичная функция потерь (Q называется MSE) $L(a, x, y) = (a(x) - y)^2$
- абсолютная функция потерь (Q называется MAE) $L(a, x, y) = |a(x) - y|$
- логарифмическая функция потерь (Q называется $MSLE$) $L(a, x, y) = (\log(a(x) + 1) - \log(y + 1))^2$
- абсолютная-процентная функция потерь (Q называется $MAPE$) $L(a, x, y) = \frac{|a(x) - y|}{y}$
- все, что сами придумаете

Обычно используют *MSE*:

$$Q(\boldsymbol{w}) = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{x_i}\cdot\boldsymbol{w} - y_i)^2 = \frac{1}{N}||X\cdot\boldsymbol{w} - \boldsymbol{y}||^2$$



Источник: `jakevdp.github.io/PythonDataScienceHandbook`

Точное решение

$\nabla_{\boldsymbol{w}} Q(\boldsymbol{w})$ – градиент, вектор частных производных. Необходимое условие минимума – градиент равен нулю.

$$Q(\boldsymbol{w}) = \frac{1}{N}||X \cdot \boldsymbol{w} - \boldsymbol{y}||^2 = \frac{1}{N}(X \cdot \boldsymbol{w} - \boldsymbol{y})^T(X \cdot \boldsymbol{w} - \boldsymbol{y})$$

Два правила векторного дифференцирования:

$$1)\nabla_{\boldsymbol{w}} \boldsymbol{w}^T \boldsymbol{c} = \nabla_{\boldsymbol{w}} \boldsymbol{c}^T \boldsymbol{w} = \boldsymbol{c}; 2)\nabla_{\boldsymbol{w}}(\boldsymbol{w}^T C \boldsymbol{w}) = (C + C^T)\boldsymbol{w}$$

$$\nabla_{\boldsymbol{w}} Q(\boldsymbol{w}) = \nabla_{\boldsymbol{w}}(\boldsymbol{w}^T X^T X \boldsymbol{w} - \boldsymbol{w}^T X^T \boldsymbol{y} - \boldsymbol{y}^T X \boldsymbol{w} + \boldsymbol{y}^T \boldsymbol{y}) = (X^T X + X^T X)\boldsymbol{w} - 2X^T \boldsymbol{y} = 0$$

$$\boldsymbol{w} = (X^T X)^{-1} X^T \boldsymbol{y}$$

Домашнее задание

Покажите, что это действительно минимум.

Точное решение

Недостатки точного решения:

- Можно написать только для очень редких функций потерь
- Обращение матрицы – кубическая сложность
- Матрица $X^T X$ может быть плохо обусловленной, если признаки линейно зависимы

Итеративные методы оптимизации

Итеративные методы оптимизации:

- Нулевого порядка: золотое сечение, метод парабол, имитация отжига, генетические алгоритмы и т.д.
- Первого порядка: градиентный спуск, метод сопряженных градиентов, квазиньютоновские методы и т.д.
- Второго порядка: ньютоновские методы.

В курсе разбираем градиентный спуск.

Градиентный спуск

Антиградиент функции показывает направления наискорейшего убывания функции.

1. выбрать начальную длину шага α_0, начальное приближение $\boldsymbol{w_0}$
2. $\boldsymbol{w_{new}} = \boldsymbol{w_{old}} - \alpha \nabla_{\boldsymbol{w}} Q(\boldsymbol{w_{old}})$
3. $\alpha = f(k), k = k + 1$
4. Повторять (2), (3) до сходимости

$f(k) = \alpha_0, f(k) = \frac{\alpha_0}{k}, f(k) = \frac{\alpha_0}{k^p}, \ldots$

Стохастическая оптимизация

В задачах машинного обучения оптимизируемая функция специального вида:

$$Q(\boldsymbol{w}) = \frac{1}{N}\sum_{i=1}^{N} L(\boldsymbol{w}, \boldsymbol{x_i}, y_i)$$

$$\nabla_w Q(\boldsymbol{w}) = \frac{1}{N}\sum_{i=1}^{N} \nabla_{\boldsymbol{w}} L(\boldsymbol{w}, \boldsymbol{x_i}, y_i)$$

Пример

Для $L(\boldsymbol{w}, \boldsymbol{x_i}, y_i) = (\boldsymbol{x_i} \cdot \boldsymbol{w} - y_i)^2$ сможете посчитать $\nabla_{\boldsymbol{w}} L(\boldsymbol{w}, \boldsymbol{x_i}, y_i)$?

Стохастическая оптимизация

1. выбрать начальную длину шага α_0, начальное приближение $\boldsymbol{w_0}$, размер батча n
2. выбрать случайно $\{j_1, i_2, ...j_n\}$
3. оценить градиент $\nabla_{\boldsymbol{w}} Q^*(w_{old}) = \frac{1}{n}\sum_{j=1}^{n} \nabla_{\boldsymbol{w}} L(\boldsymbol{w_{old}}, \boldsymbol{x_j}, y_j)$
4. $\boldsymbol{w_{new}} = \boldsymbol{w_{old}} - \alpha \nabla_{\boldsymbol{w}} Q^*(\boldsymbol{w_{old}})$
5. $\alpha = f(k), k = k + 1$
6. Повторять (2 - 5) до сходимости

В зависимости от n выделяют:

- $n = N-$ градиентный спуск (Gradient Descent, GD)
- $n = 1-$ стохастический градиентный спуск (Stochastic Gradient Descent, SGD)
- $n > 1$ и $n < N-$ мини-батч градиентный спуск (Mini-Batch Gradient Descent, MBGD)

Стохастическая оптимизация

Почему это вообще работает? Потому что такая оценка не является смещенной:

$$\mathbb{E}_{j_k} \nabla_{\boldsymbol{w}} L(\boldsymbol{w}, \boldsymbol{x_{j_k}}, y_{j_k}) = \frac{1}{N} \sum_{i=1}^{N} \nabla_{\boldsymbol{w}} L(\boldsymbol{w}, \boldsymbol{x_i}, y_i)$$

Градиентного спуск более точный и быстрее сходится (если мерить числом итераций до сходимости), но у стохастического спуска каждая итерация намного быстрее, то есть по времени он может сойтись быстрее!

Итоги

Линейная регрессия невероятно популярный алгоритм машинного обучения. Плюсы алгоритма:

- Быстро учится
- Быстро предсказывает
- Легко интерпретируется
- Легко хранить в памяти
- Легко применять с дифференцируемой функцией потерь

Весомый минус - не способен учитывать нелинейные зависимости в данных.

Пример

Строим рекомендательную систему музыки. Где могут появиться нелинейные зависимости?

Часть 3

Уменьшаем минусы, добавляем плюсы!

Правильные признаки для линейной регрессии

Категориальные признаки кодируем:

- One-hot кодирование - категориальный признак с k значениями превращаем в k бинарных признаков!
- Кодирование через целевую переменную (нельзя включать переменную самого объекта)
- Кодирование через вещественные признаки

Для вещественных признаков применяем нелинейные функции - возводим в степень, берем синус и т.д.

Учитываем взаимодействия:

- Пару вещественных перемножаем, делим и т.д.
- Для пары бинарных используем логические операции

Невозможно сделать правильное признаковое пространство без понимания самой задачи!

Нормализация данных

Масштаб признаков очень важен для скорости сходимости градиентного спуска.

Пример

$f(x, y, z) = (x-1)^2 + (y-1)^2 + (z-1)^2$. Стартуем из $(0, 0, 0)$ с шагом 0.5
А если так $f(x, y, z) = 1000(x-1)^2 + 200(y-1)^2 + (z-1)^2$?

Два основных вида нормализации:

- Стандартизация. $f_j = \frac{f_j - mean(f_j)}{std(f_j)}$
- Min-max нормализация. $f_j = \frac{f_j - min(f_j)}{max(f_j) - min(f_j)}$

Регуляризация

Хотим еще сузить функциональное пространство $a(x)$, чтобы улучшить обобщающую способность. Наложим доп. штраф, если решение удаляется от нашего представления о правильном решении.

$$Q_r(\boldsymbol{w}) = Q(\boldsymbol{w}) + \alpha R(\boldsymbol{w}),$$

$R(\boldsymbol{w})$ - регуляризатор, α - параметр регуляризации.

Пример

Строим рекомендательную систему музыки. Какие регуляризаторы стоит сделать?

Регуляризация

Базовые регуляризаторы, штрафующие большие значения весов:

- L_1 регуляризация (Ridge регрессия). $R(\boldsymbol{w}) = \sum_{j=1}^{D} |w_j|$; С ней Q_r не будет гладким!
- L_2 регуляризация (Lasso регрессия). $R(\boldsymbol{w}) = \sum_{j=1}^{D} w_j^2$.

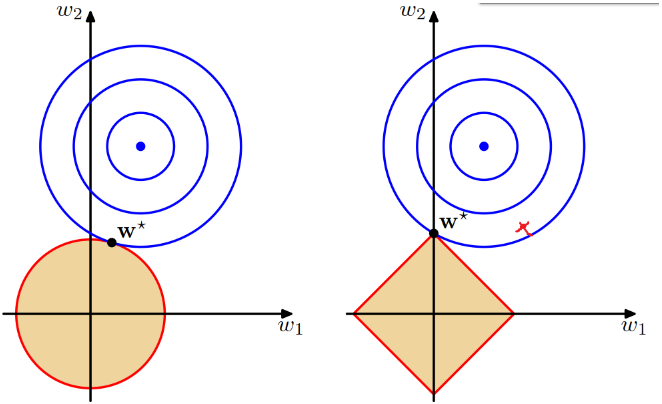
Домашнее задание

Показать, что с L_2 точное решение линейной регрессии станет $\boldsymbol{w} = (X^T X + \alpha I)^{-1} X^T \boldsymbol{y}$. Что это дает для нас?

Домашнее задание

Выписать правило обновления весов для SGD линейной регрессии с L_2.

Можно показать, что L_1 приводит к занулению весов, то есть автоматически отбирает важные признаки!



Источник: Bishop.

Big data и линейная регрессия

Стохастическая оптимизация позволяет не хранить данные в оперативной памяти! Взяли батч, обновили веса и сразу же его забыли. Позволяет легко обучаться на терабайтах данных. Рекомендуем ознакомиться с библиотекой **Vowpal Wabbit**.
Online learning – обучение, когда прецеденты поступают потоком.

Пример

Строим рекомендательную систему музыки. Как в ней устроим online learning?

Разреженные данные

Учим линейноую модель предсказывать спам/не спам почтовых писем. Делаем бинарные признаки f_i факт наличия слова $word_i$. Сколько таких будет признаков?
Чему равен градиент по j компоненте $\nabla_{\boldsymbol{w}}(\boldsymbol{x_i} \cdot \boldsymbol{w} - y_i)^2$ в точках $x_{i,j} = 0$?
Разреженный формат данных - для каждого объекта храним словарь вида {номер ненулевого признака : значение}

Пример

Строим рекомендательную систему музыки. Как сдалаем признаковое пространство?

Регрессия и причинно-следственные связи

Предсказываем сколько прослужит машина по ее цвету. Получили, что вес ненулевой: желтые машины дольше работают?

Предсказываем, результат физических упраженений по среднему баллу. Зависимостей нет, коэффицент около 0. А если добавить признак, поступил ли студент на военную кафедру МГУ?

Линейная регрессия позволяет находить причинно-следственные связи если:

- Линейная региессия должна содержать все признаки, являющиеся причинами признакам
- Динейная региессия не должна содержать признаки, которые являются следствиями одновременно самого признака и целевой переменной.

Рекомендую курс: `www.coursera.org/learn/stats-for-data-analysis`

Неправильные интерпретации:

`https://alexanderdyakonov.files.wordpress.com/2015/07/dyakonovfunnydm.pdf`

Заключение

Спасибо за внимание!