Методы анализа данных

Александр Широков ПМ-1701

Преподаватель:

Ивахненко Дарья Александровна

Санкт-Петербург
2020 г., 7 семестр

Список литературы

[1]

Содержание

1 01.09.2020

1.1 Задача обучения по предедентам

Пусть X - множество объектов, а Y - множество ответов. $y : X \to Y$ - неизвестная зависимость.

Дано: $\{x_1, \ldots, x_l\} \subset X$ - обучающая выборка, а $y_i = y(x_i), i = 1, \ldots, l$ - известные ответы.

Требуется найти $a : X \to Y$ - алгоритм, решающую функцию, приближающую y на всем множестве X.

1.2 Типы задач

Задачи восстановления регрессии:

- $Y = \mathbb{R}$ - вся числовая ось:
  - определение температуры воздуха метеорологического поля
  - оценка влияния факторов потребления
- $Y \in [0; +\infty)$:
  - задачи медицинской диагностики: прогнозирование ожидаемого время действия препарата
  - задачи кредитного скоринга: определение величины кредитного лимита
  - определение расхода топлива по техническим характеристикам
- $Y \in [0, 1, \ldots, +\infty)$ - счетная целевая переменная

Задача классификации:

- $Y = \{-1, +1\}$ - классификация на два класса:
  - задачи кредитного скоринга: решение о выдаче кредита
  - предсказание оттока клиентов
- $Y = \{1, \ldots, K\}$ - классификация на K непересекающихся классов:
  - задачи медицинской диагностики: определение диагноза
  - распознавание символов
  - определение жанра
- $Y = \{0, 1\}^K$ - на K классов, которые могут пересекаться:
  - определение ключевых слов для оптимизации поиска
  - определение присутствующих на фото объектов

Типы признаков

- $D_j = \{0, 1\}$ - бинарный признак f_j:
  - пол
  - является ли..?
- $|D_j| < \infty$ - номинальный признак f_j:
  - город
  - цвет
- $|D_j| < \infty$, D_j - упорядочено - порядковый признак f_j:
  - уровень холестерина (ниже нормы, норма, выше нормы)
- $D_j = \mathbf{R}$ - количественный признак f_j:
  - длина и ширина объекта

1.3 08.09.2020

1.3.1 Задача бинарной классификации

Выведем функцию связи через биномиальное распределение через задачу классификации:

$y \in \{0, 1\}$

$$g^{-1}(p) = \overline{\theta}^T \overline{X}$$

$$f(y) = e^{\frac{y\alpha - c(\alpha)}{\varphi} + h(y,\varphi)}$$

Плотность биномиального распределения:

$$f(y) = p^y(1-p)^{1-y} = exp(y \log p + (1-y)\log(1-p)) = exp(y \log \frac{p}{1-p} + \log(1-p) \equiv c(\alpha)$$

$$\alpha = \log \frac{p}{1-p} = g(p) = \overline{\theta}^T \overline{X}$$

$$\frac{p}{1-p} = e^{\overline{\theta}^T \overline{X}}$$

$$p = \sigma\left(\overline{\theta}^T \overline{X}\right) = \frac{1}{1 + e^{-\overline{\theta}^T \overline{X}}}$$

Осталось получить функционал качества для задачи классификации. Функция называется логистической сигмоидой. Данная функция преобразует линейную комбинацию в интервал $[0, 1]$. Дальнейшее значение целевой переменной мы будем предсказывать в качестве:

$$\hat{y} = \sigma\left(\overline{\theta}^T \overline{X}\right)$$

Функционал качества найдём через метод максимального правдоподобия:

$$p(x, y, p) = \prod_{i=1}^{l} p_i^{y_i} (1 - p_i)^{1-y_i} \underset{\theta}{\rightarrow} \max$$

$$\sum_{i=1}^{l} y_i \log p_i + (1 - y_i) \log(1 - p_i) \underset{\theta}{\to} \max$$

Функция потерь называется LogLoss:

$$LogLoss = L(x_i) = y_i \log p_i - (1 - y_i) \log p_i \underset{\theta}{\to} \min, p_i = \sigma(x_i)$$

Если мы правильно предсказываем отношение принадлежности класса к 1, то функция потерь будет равна нулю, если правильно предсказываем 0 правильно, то тоже 0, а если 1 - правильный ответ, а 0 - нет, то ошибка будет $+\infty$, и ошибку ограничивают значениями 100, чтобы ошибка не уходила далеко.

$$p = \sigma\left(\overline{\theta}^T \overline{X}\right) \in [0, 1]$$

$$L = -y \log p - (1 - y) \log(1 - p)$$

1.3.2 Задача многоклассовой классификации

Определение 1.3.1. Categorical Distribution: $y \in \{1, 2, \ldots, K\}$

$$f(y) = \prod_{i=1}^{K} p_i^{y_i}$$

Выведем функцию связи через Categorical Distribution:

$$f(y) = \prod_{i=1}^{K} p_i^{y_i} = \exp(\log \prod_{i=1}^{K} p_i^{y_i}) = \exp(\sum_{i=1}^{K} y_i \log p_i)$$

Рассмотрим определенный y_i:

$$f(y) = \exp(y_1 \log p_1 + y_2 \log p_2 + \ldots + y_K \log p_K)$$

$$\sum y_i = 1$$

$$\sum p_i = 1$$

Так как y_i имеет принадлежность определенному классу и может быть равен только одной единице в векторе:

$$\{0, 0, 1, 0\}$$

а сумма вероятностей по определению. Тогда:

$$f(y) = \exp(y_1 \log p_1 + \ldots + \left(1 - \sum_{i}^{K-1} y_i\right) \log p_k) =$$

$$= \exp\left(y_1 \log \frac{p_1}{p_k} + y_2 \log \frac{p_2}{p_k} + \ldots + \log p_k\right)$$

X для всех одинаковый, а y разные. Для определения y_i нам необхо-

димо определить α_i:

$$\alpha_i = \log \frac{p_i}{p_k} = \overline{\theta}^T \overline{X}$$

$$\frac{p_i}{p_k} = e^{\overline{\theta}^T \overline{X}}$$

$$\sum_i^K e^{\overline{\theta_i}^T \overline{X}} = \frac{\sum + i p_i}{p_k} = \frac{1}{p_k}$$

$$p_k = \frac{1}{\sum\limits_i^K e^{\overline{\theta_i}^T \overline{X}}}$$

$$\frac{p_i}{p_k} = e^{\overline{\theta_i}^T \overline{X}}$$

$$p_i = \frac{e^{\overline{\theta_i}^T \overline{X}}}{\sum\limits_{i=1}^K e^{\overline{\theta_i}^T \overline{X}}}$$

Данная функция называется функцией SoftMax.

Функционал качества мы можем легко получить:

$$L(x, y, p) = \prod_{i=1}^{l} \prod_{j=1}^{K} p_{ij}^{y_i j} \to \max$$

где i - i-ый объект, а j - принадлежность j-му классу.

Тогда функция потери для каждого класса:

$$-\sum_{j=1}^{K} y_j \log p_j \to \min$$

и данная функция называется кросс-энтропией.

Итого:

- $y \in \{0, 1\}$:

$$Q = -\left(\sum_{i=1}^{l} y_i \log \sigma_i + (1 - y_i) \log(1 - \sigma_i) \right) \xrightarrow[\theta]{\min}$$

- $y \in \{1, 2, \ldots, K\}$:

$$Q = -\sum_{i=1}^{l} y_{ij} \log G_{ij}$$

где y представляется вектором и матрица весов:

$$\Theta = \begin{bmatrix} \theta_{11} & \theta_{12} & \ldots & \theta_{1N} \\ \ldots & \ldots & \ldots & \ldots \\ \theta_{K1} & \ldots & \ldots & \theta_{KN} \end{bmatrix}$$

1.3.3 Принадлежность многим классам

$$f(y) = \prod_{i=1}^{K} p_i^{y_i}$$

$$f(y) = p^y (1-p)^{1-y}$$

Объединение двух распределений будет выглядеть следующим образом:

$$\prod_{i=1}^{K} p_i^{y_i}(1-p_i)^{1-y_i} \qquad p_i = G(\overline{\theta_i}^T \overline{X})$$

Матрица весов размерности $K \times M$ Необходимо:

- Вывести функционал качества

$$f(y) = \exp\left(\log \prod_{i=1}^{K} p_i^{y_i}(1-p_i)^{1-y_i}\right) = \exp\left(\sum_{i=1}^{K} y_i \log p_i + \sum_{i=1}^{K}(1-y_i)\log(1-p_i)\right)$$

$$Q = \prod_{i=1}^{l}\prod_{j=1}^{K} \sigma_{ij}^{y_{ij}}(1-\sigma_{ij})^{1-y_{ij}} \to \max$$

$$Q = -\sum_{i=1}^{l}\sum_{j=1}^{K}(y_{ij}\log\sigma_{ij} + (1-y_{ij})\log(1-\sigma_{ij})) \xrightarrow[\theta]{\min}$$

$$L = -\sum_{j=1}^{K}(y_j \log\sigma_j + (1-y_j)\log(1-\sigma_j)) \xrightarrow[\theta]{\min}$$

- Градиент функции потерь

$$\frac{\partial L}{\partial \theta_{KM}} = \frac{y_k}{\sigma_K}\cdot\sigma_k(1-\sigma_K)x_{KM} - (1-y_k)\frac{1}{1-\sigma_k}\sigma_k(1-\sigma_k)x_{KM}$$

$= y_k(1-\sigma_k)x_{KM} - (1-y_k)\sigma_k x_{KM} = x_{KM}(y_k - y_k\sigma_k - \sigma_k + \sigma_k y_k) = x_{KM}(y_k - \sigma_k)$

так как

$$\sigma_k = \frac{1}{1+e^{-\overline{\theta_k}^T\overline{x}}}$$

$$\sigma(z)' = \left(\frac{1}{1+e^{-z}}\right)' = -\frac{e^{-z}(-1)}{(1+e^{-z})^2} = \sigma(z)\cdot(1-\sigma(z))$$

1.4 Методы обработки текстов

b

1.5 Word2Vec

В модели Word2Vec есть фиксированный словарь, каждому слову сопоставляется вектор. Для каждого слова есть центральное слово и есть CONTEXT - слова вокруг центрального.

Мы используем сходство векторных представлений, чтобы предсказать вероятность слова на основании контекста. Данные вектора будем двигать в пространстве таким образом, чтобы вероятность слова в данном контексте была максимальной.

У нас есть корпус текста. Мы можем нарезать слово на контекст. Для каждого центрального слова из контекста имеем вероятности возникновения слова и мы предсказываем вероятности подобного слова.

У каждого слова есть своя вероятность встретиться со словом.

Нет такого понятия, как предсказание вероятности.

1.5.1 Cross-Entropy loss function

Пусть у нас есть некоторое количество вероятностей для всех слов, при условии, что у нас есть некоторые параметры, которые параметризуют функцию вероятности. Мы попробуем, чтобы получающаяся вероятность была похожа.

$$J(\theta) = -\frac{1}{T}\log(L(\theta)) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{\substack{-m\leq j\leq m\\ j\neq 0}}\log(w_{t+j}|w_t,\theta)$$

Наборы вероятностей мы предсказываем и мерим разницу между двумя вероятностями. В силу того, что мы работает с реальными корпусами. В реальном контесте встретился определенный набор слов.

$$J(\theta) = -\frac{1}{T}\log(L(\theta)) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{\substack{-m\leq j\leq m\\ j\neq 0}}\log P(w_{t+j}|w_t,\theta) \to \min$$

Мы хотим данную функцию максимизировать. Но как высчитывать вероятности? Будем использовать два вектора для каждого слова w: v_w, когда w - центральное слово и u_w, когда w - слово контеста.

Для центрального слова c и контекстного слова o:

$$p(o|c) = \frac{e^{u_o^T v_c}}{\sum\limits_{w\in V} e^{u_w^T v_c}}$$

Данная функция называется Softmax - она делет контрастирование. У нас есть набор значений - косинусные состояния. Каждое значение будет возможно интерпретировать в качестве вероятности. Softmax выдает нормализованный вектор значений. Этот вектор значений мы можем интерпретировать, как вероятность распределений. Является вероятностью,

есть вход распределен нормально.

Softmax является дифференцируемой функцией, поэтому мы можем обучать нейронную сеть. Вычисление Softmax по большому словарю.

1.5.2 Skip-gram Model

Skipgram - мы пропускаем что-то, которая с помощью пропущенного работает. Есть центральное слово o, есть набор слов справа и мы будем предсказывать каждое из слов в контексте.

Тренировать модель будет с помощью оптимизации параметров - стохастический градиентный спуск.

У нас есть набор параметров - набор векторов слов для каждого слова. θ - будем оптимизировать все вектора:

$$\theta \in R^{2dV}$$

$$J(\theta) = -\frac{1}{T}\log L(\theta) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{\substack{-m\leq j\leq m\\ j\neq 0}}\log P(w_{t+j}|w_t,\theta) = -\frac{1}{T}\sum_{o\in context(c)}\sum_{c\in corpus}\log p(o|c;u,v)$$

$$\frac{\partial J(\theta)}{\partial\theta} = -\frac{1}{T}\sum_{o\in context(c)}\sum_{c\in corpus}\frac{\partial}{\partial\theta}\log p(o|c;u,v)$$

Подставим вместо p - Softmax:

$$\frac{\partial}{\partial v_c}\log p(o|c;u,v) = \frac{\partial}{\partial v_c}\log\frac{e^{u_o^T v_c}}{\sum\limits_{w\in V} e^{u_w^T v_c}} = \frac{\partial}{\partial v_c}\log\frac{e^{u_o^T v_c}}{\sum\limits_{w=1}^{V} e^{u_w^T v_c}} =$$

$$\frac{\partial}{\partial v_c}(u_o^T v_c) - \frac{\partial}{\partial v_c}\log\sum_{w=1}^{V} e^{u_w^T v_c} = u_0 - \frac{1}{\sum\limits_{w=1}^{V} e^{u_w^T v_c}}\cdot\frac{\partial}{\partial v_c}\sum_{w=1}^{V} e^{u_w^T v_c} =$$

$$= u_0 - \frac{1}{\sum\limits_{w=1}^{V} e^{u_w^T v_c}}\cdot\sum_{w=1}^{V}\frac{\partial}{\partial v_c}e^{u_w^T v_c} = u_0 - \frac{1}{\sum\limits_{w=1}^{V} e^{u_w^T v_c}}\sum_{w=1}^{V} e^{u_w^T v_c}u_w =$$

$$u_0 - \sum_{x=1}^{V}\frac{e^{u_x^T v_c}}{\sum\limits_{w=1}^{V} e^{u_w^T v_c}}u_x = u_0 - \sum_{x=1}^{V} p(x|c)u_x$$

Получаем, что частная производная по контекстному слову составляет:

$$\frac{\partial}{\partial v_c}J(\theta) = -\frac{1}{T}\sum_{o\in context(c)}\sum_{c\in corpus}(u_0 - \sum_{x=1}^{V} p(x|c)u_x)$$

$$\frac{\partial}{\partial u_0}J(\theta) = -\frac{1}{T}\sum_{o\in context(c)}\sum_{c\in corpus}(v_c - \sum_{x=1}^{V} p(o|x)v_x)$$