Чудесный мир регрессии, 2

Машинное обучение, 20!7

Спасибо К. В. Воронцову, МФТИ, Data Factory Яндекса, O.D.S. и кофеину.

Малютин Е. А.

Содержание

Планчик

- Трюки и финты
- Проблемы регрессии
- Приложения

w

В прошлы сериях:

Условности

Можно выписать модель регрессии явным образом для отдельного объекта:

$$y_i = \sum_{j=0}^{n} w_j X_{ij} + \epsilon$$

И, соответственно, мы приходим к некоторым условиям:

- матожидание: $\forall i : E[\epsilon_i] = 0$;
- гомоскедастичность: $\forall : Var(\epsilon_i) = \sigma^2 < \infty$
- некоррелированны: $\forall i \neq j : Cov(\epsilon_i, \epsilon_j) = 0$

В прошлых сериях:

Условности

Можно выписать модель регрессии явным образом для отдельного объекта:

$$y_i = \sum_{j=0}^{n} w_j X_{ij} + \epsilon$$

И, соответственно, мы приходим к некоторым условиям:

- матожидание: $\forall i : E[\epsilon_i] = 0$;
- гомоскедастичность: $\forall : Var(\epsilon_i) = \sigma^2 < \infty$
- некоррелированны: $\forall i \neq j : Cov(\epsilon_i, \epsilon_j) = 0$

В прошлых сериях:

Метод максимального правдоподобия

Правдоподобие: $L = \Pi p(y_i x_i, \alpha)$
Log-likelihood: $W(\alpha) = log(L) = \sum ln(P(y|X, \alpha))$
Положим ошибки нормально-распределёнными: $\epsilon_i \backsim \mathcal{N}(0, \sigma^2)$
И тогда для нашей модели: $p(y_i|x_i, w) = \mathcal{N}(\sum_j w_j X_{ij}, \sigma^2)$

$$log(p(X, y|w)) = log(\Pi\ \mathcal{N}(\sum_j w_j X_{ij}, \sigma^2)) = -\frac{n}{2} log 2\pi\sigma^2 - \frac{1}{2\sigma^2} \sum (y_i - \omega^T x_i)^2$$

И, максимизируя это вот всё:

$$w = arg \max_w p(X, y|w) = arg \max_\omega -\mathcal{L}(X, y, \omega)$$

Вывод: минимизация МНК эквивалентна максимизации МП.

В прошлых сериях:

Регуляризация

- При мультиколлинеарности в данных получаем $(X^TX)^{-1}$ – экстремально большие значения собственных чисел $(\frac{1}{\lambda_i})$
- Решение? Регуляризация по Тихонову:

$$\mathcal{L}(X, \overrightarrow{y}, \overrightarrow{\omega}) = \frac{1}{2n}||\overrightarrow{y} - X\overrightarrow{\omega}||^2 + ||\mathcal{G}\overrightarrow{\omega}||^2$$

- Часто используется в таком виде: $\mathcal{G} = \frac{\lambda}{2}E$
- Точное решение:

$$\overrightarrow{\omega} = (X^TX + \lambda E)^{-1}X^T\overrightarrow{y}$$

В прошлых сериях:

$\exists X, P(X); OR(X) \equiv \frac{P(X)}{1-P(X)}$; (отношение вероятностей) $P(X) \in [0,1]; OR(X) \in R$

Вычисляем лог. регрессию

- Вычислить значение $w_0 + w_1 x_1 + w_2 x_2 + ... = w^T \overrightarrow{x}$. (уравнение $\overrightarrow{w}^T \mathbf{x} = 0$ задает гиперплоскость, разделяющую примеры на 2 класса);
- Вычислить логарифм отношения шансов: $log(OR_+) = \overrightarrow{w}^T \overrightarrow{x}$.
- Вычисляем вероятность: $p_+ = \frac{OR_+}{1+OR_+} = \frac{exp^{\overrightarrow{w}^T \overrightarrow{x}}}{1+exp^{\overrightarrow{w}^T \overrightarrow{x}}} = \frac{1}{1-exp^{-\overrightarrow{w}^T \overrightarrow{x}}} = \sigma(\overrightarrow{w}^T \overrightarrow{x})$

В прошлых сериях:

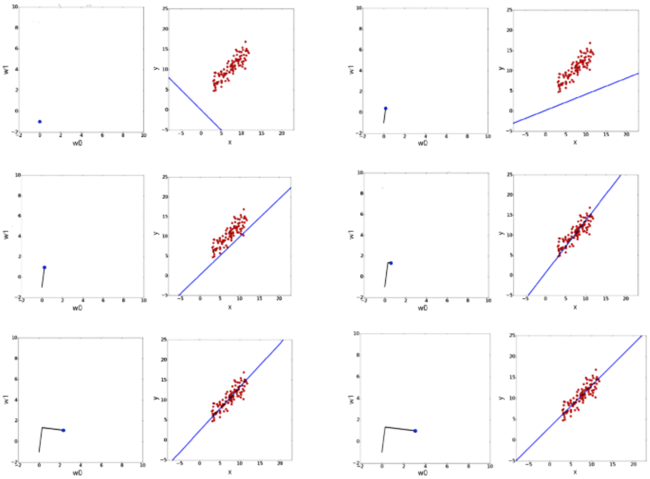


Рис.: Пример Gradient Descent на одномерной регрессии

Регрессия
В матрицах

В матричном мире:

- $g(x, a) = \sum\limits_{1}^{n} a_j f_j$; пусть y – вектор ответов, F – матрица объект-признак;

Регрессия
В матрицах

В матричном мире:

- $g(x,a) = \sum\limits_{1}^{n} a_j f_j$; пусть y – вектор ответов, F – матрица объект-признак;
- $Q(a) = ||Fa - y||^2$, – функционал ошибки;

В матричном мире:

- $g(x,a) = \sum\limits_{1}^{n} a_j f_j$; пусть y – вектор ответов, F – матрица объект-признак;
- $Q(a) = ||Fa - y||^2$, – функционал ошибки;
- Минимум в матричном виде: $\frac{\partial Q}{\partial a} Q(f) = 2F^T(Fa - y) = 0$

Регрессия

В матрицах

В матричном мире:

- $g(x,a) = \sum\limits_{1}^{n} a_j f_j$; пусть y – вектор ответов, F – матрица объект-признак;
- $Q(a) = ||Fa - y||^2$, – функционал ошибки;
- Минимум в матричном виде: $\frac{\partial Q}{\partial a} Q(f) = 2F^T(Fa - y) = 0$
- $a^* = (F^T F)^{-1} F^T y$ – аналитическое решение,
  $Q(a^*) = ||\mathcal{P}_F y - y||^2$ – функционал ошибки на решении

Регрессия

В матрицах

В матричном мире:

- $g(x,a) = \sum\limits_{1}^{n} a_j f_j$; пусть y – вектор ответов, F – матрица объект-признак;
- $Q(a) = ||Fa - y||^2$, – функционал ошибки;
- Минимум в матричном виде: $\frac{\partial Q}{\partial a} Q(f) = 2F^T(Fa - y) = 0$
- $a^* = (F^T F)^{-1} F^T y$ – аналитическое решение,
  $Q(a^*) = ||\mathcal{P}_F y - y||^2$ – функционал ошибки на решении
- $F^+ = (F^T F)^{-1} F^T$ – псевдообратная матрица
  $\mathcal{P}_F = FF^+$ – проекционная матрица

Singular value decomposition

- Произвольную $l \times n$-матрицу ранга n можно представить в виде сингулярного разложения, SVD

Singular value decomposition

- Произвольную $l \times n$-матрицу ранга n можно представить в виде сингулярного разложения, SVD
- $F = VDU^T$
  1. $n \times n$-матрица D диагональна, $D = diag(\sqrt{\lambda_1}, ..., \sqrt{\lambda_n})$ – общие ненулевые собственные значения F^TF и FF^T;

Singular value decomposition

- Произвольную $l \times n$-матрицу ранга n можно представить в виде сингулярного разложения, SVD
- $F = VDU^T$
  1. $n \times n$-матрица D диагональна, $D = diag(\sqrt{\lambda_1}, ..., \sqrt{\lambda_n})$ – общие ненулевые собственные значения F^TF и FF^T;
  2. $l \times n$-матрица $V = (v_1, ..., v_n)$ ортогональна, $V^TV = I^n$, столбцы v_j являются собственными векторами матрицы FF^T, соответствующими $\lambda_1, ..., \lambda_n$;

Singular value decomposition

- Произвольную $l \times n$-матрицу ранга n можно представить в виде сингулярного разложения, SVD
- $F = VDU^T$
  1. $n \times n$-матрица D диагональна, $D = diag(\sqrt{\lambda_1}, ..., \sqrt{\lambda_n})$ – общие ненулевые собственные значения $F^T F$ и FF^T;
  2. $l \times n$-матрица $V = (v_1, ..., v_n)$ ортогональна, $V^T V = I^n$, столбцы v_j являются собственными векторами матрицы FF^T, соответствующими $\lambda_1, ..., \lambda_n$;
  3. $n \times n$-матрица $U = (u1, ..., u_n)$ ортогональна, $U^T U = I^n$, столбцы u_j являются собственными векторами матрицы $F^T F$, соответствующими $\lambda_1, ..., \lambda_n$;

В контексте регрессии:

- Псевдообратную матрица:
$$F^+ = (UDV^T VDU^T)^{-1} UDV^T = UD^{-1}V^T = \sum_{j=1}^{n} \frac{1}{\sqrt{\lambda_j}} u_j v_j^T$$

Регрессия
SVD

В контексте регрессии:

- Псевдообратную матрица:
  $$F^+ = (UDV^T VDU^T)^{-1} UDV^T = UD^{-1}V^T = \sum_{j=1}^{n} \frac{1}{\sqrt{\lambda_j}} u_j v_j^T$$
- Вектор МНК-решения: $a^* = F^+ y = UD^{-1}V^T y = \sum_{j=1}^{n} \frac{1}{\sqrt{\lambda_j}} u_j (v_j^T y)$

Регрессия
SVD

В контексте регрессии:

- Псевдообратную матрица:
$$F^{+} = (UDV^TVDU^T)^{-1}UDV^T = UD^{-1}V^T = \sum_{j=1}^{n} \frac{1}{\sqrt{\lambda_j}} u_j v_j^T$$
- Вектор МНК-решения: $a^* = F^{+}y = UD^{-1}V^Ty = \sum_{j=1}^{n} \frac{1}{\sqrt{\lambda_j}} u_j (v_j^T y)$
- МНК-аппроксимация целевого вектора y:
$Fa = P_F y = y^T VD^{-1}UD^{-1}V^T y = y^T VD^{-2}V^T y = \sum \frac{1}{\lambda_j}(v_j^T y)^2$

Регрессия
SVD

В контексте регрессии:

- Псевдообратную матрица:
$$F^+ = (UDV^TVDU^T)^{-1}UDV^T = UD^{-1}V^T = \sum_{j=1}^{n} \frac{1}{\sqrt{\lambda_j}} u_j v_j^T$$
- Вектор МНК-решения: $a^* = F^+y = UD^{-1}V^Ty = \sum_{j=1}^{n} \frac{1}{\sqrt{\lambda_j}} u_j(v_j^Ty)$
- МНК-аппроксимация целевого вектора y:
$Fa = P_Fy = y^TVD^{-1}UD^{-1}V^Ty = y^TVD^{-2}V^Ty = \sum \frac{1}{\lambda_j}(v_j^Ty)^2$
- Норма вектора: $||a^*||^2 = yVD^{-1}U^TUD^{-1}V^Ty = y^TVD^{-2}V^Ty = \sum \frac{1}{\lambda_j}(v_jy)^2$

Число обусловленности:

- *Матрица ковариации:* $\Sigma = F^T F$, на практике частенько попадается Σ – *матрица неполного псевдоранга*;

Число обусловленности:

- *Матрица ковариации:* $\Sigma = F^T F$, на практике частенько попадается Σ – *матрица неполного псевдоранга*;
- Число обусловленности:

$$\mu(\Sigma) = ||\Sigma|| ||\Sigma^{-1}|| = \frac{\lambda_{max}}{\lambda_{min}}$$

Число обусловленности:

- *Матрица ковариации:* $\Sigma = F^T F$, на практике частенько попадается Σ – *матрица неполного псевдоранга*;
- Число обусловленности:

$$\mu(\Sigma) = ||\Sigma|| ||\Sigma^{-1}|| = \frac{\lambda_{max}}{\lambda_{min}}$$

- При умножении обратной матрицы на вектор, $z = \Sigma^{-1} u$, относительная погрешность усиливается в $\mu(\Sigma)$ раз:

$$\frac{||\delta z||}{||z||} \leq \mu(\Sigma) \frac{||\delta u||}{||\delta u||}$$

Скучно:

- $Q(a) = ||Fa - y||^2 + \tau ||a||^2$
- $\frac{\delta Q_\tau}{\delta a} = 0 \Rightarrow a_\tau^* = (F^T F + \tau I_n)^{-1} F^T y$
- $||a_*||^2 = \sum \frac{1}{\lambda_j + \tau} (v_j^T y)^2 < ||a||^2$

Как выбрать τ?

- Как выбрать τ?

Как выбрать τ?

- Как выбрать τ?
- Скользящий контроль

Как выбрать τ?

- Как выбрать τ?
- Скользящий контроль
- Практическая рекомендация: $\tau \in [0.1, 0.4]$

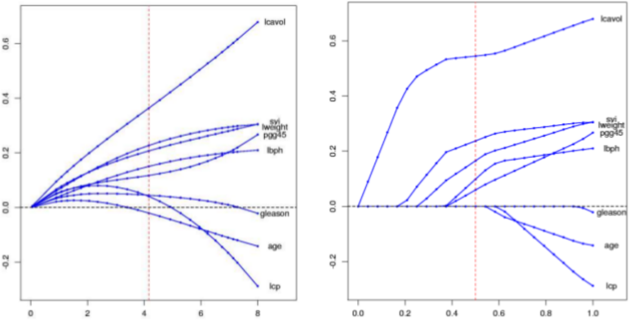
Как выбрать τ?

- Как выбрать τ?
- Скользящий контроль
- Практическая рекомендация: $\tau \in [0.1, 0.4]$
- Ограничить число обусловленности:
  $M_0 = \mu(F^T F + \tau I_n) = \frac{\lambda_{max}+\tau}{\lambda_{min}+\tau} \Rightarrow \tau^* = \frac{\lambda_{max}}{M_0}$

Скучно:

$$\begin{cases} Q(a) = ||F(a) - y||^2 \to min \\ \sum |a_j| \leq \aleph \end{cases}$$

Задача ЛП. Большие $\aleph$ обращают компонента вектора в 0.
Пусть $a_j = a_j^+ - a_j^-$, тогда ограничение на а принимает вид:
$\sum a_j^+ + a_j^- \leq \aleph; a_j^+ \geq 0, a_j^- \geq 0$



(а) зависимость $\{\alpha_j\}$ от $\sigma = 1/\tau$ (б) зависимость $\{\alpha_j\}$ от $\varkappa$

Рис.: Зависимость коэффициентов линейной модели от параметра $\sigma = 1/\tau$ для гребневой регрессии и от параметра κ для лассо Тибширани, по реальным данным задачи UCI.cancer

Регрессия

Метод главных компонент (PCA)

Постановка задачи:

- пусть дана: $F_{l\times n}$ – признаковое описание
- $G_{m\times n}$ – признаковое описание в новом пространстве R^m, $m < n$;
- F можно восстановить с помощью линейного преобразования $U = (u_{js})_{n\times m}$: $\widehat{f_j} = \sum\limits_s g_s u_{js}$ или $\widehat{f} = zU^T$
- причем $\Delta^2(G, U) = ||GU^T - F||^2 \to \min\limits_{G,U}$

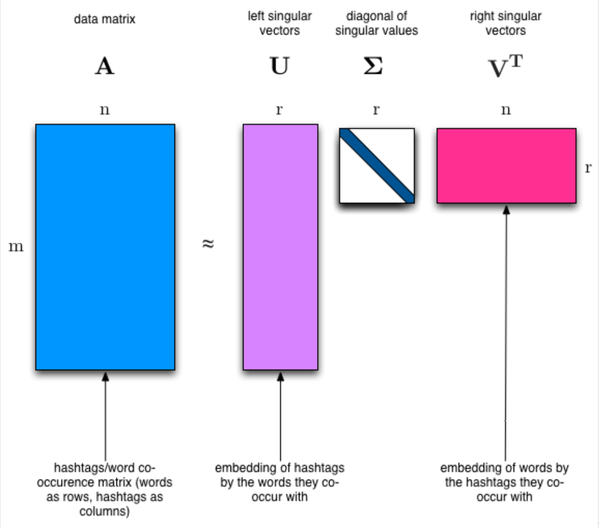
Теорема

Если $m < rkF$, то минимум $\Delta^2(G, U)$ достигается, когда столбцы матрицы U есть собственные векторы $F^T F$, соответствующие m максимальным собственным значениям. При этом $G = FU$, матрицы U и G ортогональны.

Регрессия
Свойства (PCA)

Связь с сингулярным разложением:

- Если $m = n$, то $\Delta^2(G, U) = 0$. В этом случае представление $F = GU^T$ является точным и совпадает с сингулярным разложением: $F = GU^T = VDU^T$
- Если $m < n$, то представление $F \approx GU^T$ является приближённым. Сингулярное разложение матрицы GU^T получается из сингулярного разложения матрицы F путём отбрасывания(обнуления) $n - m$ минимальных собственных значений.


data matrix
left singular vectors
diagonal of singular values
right singular vectors
A
U
Σ
V^T
n
r
r
n
m
≈
r
hashtags/word co-occurence matrix (words as rows, hashtags as columns)
embedding of hashtags by the words they co-occur with
embedding of words by the hashtags they co-occur with

В новом признаковом пространстве

- $||G\beta - y||^2 \to \min\limits_{\beta}$ – задача оптимизации в новом признаковом пространстве
- $\beta^* = D^{-1}V^Ty$
- $G\beta^* = VD\beta^* = VV^Ty.$
- Для вектора $a^* = U\beta^*$ МНК-решение выглядит так же, как и раньше, с той лишь разницей, что надо взять первые $m - n$ слагаемых, а оставшиеся $n - m$ просто отбросить

Pricipal Component Analysis

Снижение размерности

Эффективная размерность

- Сортируем числа: $\lambda_1 > ... > \lambda_n > 0$

Эффективная размерность

- Сортируем числа: $\lambda_1 > ... > \lambda_n > 0$
- Задаём $\epsilon \in [0, 1]$

Эффективная размерность

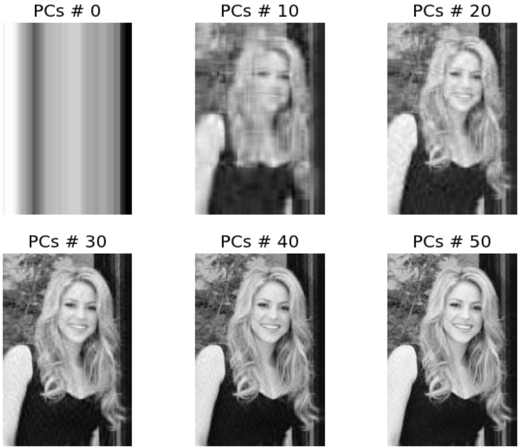
- Сортируем числа: $\lambda_1 > ... > \lambda_n > 0$
- Задаём $\epsilon \in [0, 1]$
- Считаем $E(m) = \frac{||GU - F||^2}{||F||^2} = \frac{\lambda_{m+1} + ... + \lambda_n}{\lambda_1 + ... + \lambda_m} \leq \epsilon$

Эффективная размерность

- Сортируем числа: $\lambda_1 > \ldots > \lambda_n > 0$
- Задаём $\epsilon \in [0, 1]$
- Считаем $E(m) = \frac{||GU - F||^2}{||F||^2} = \frac{\lambda_{m+1} + \ldots + \lambda_n}{\lambda_1 + \ldots + \lambda_m} \leq \epsilon$
- "Крутой обрыв"

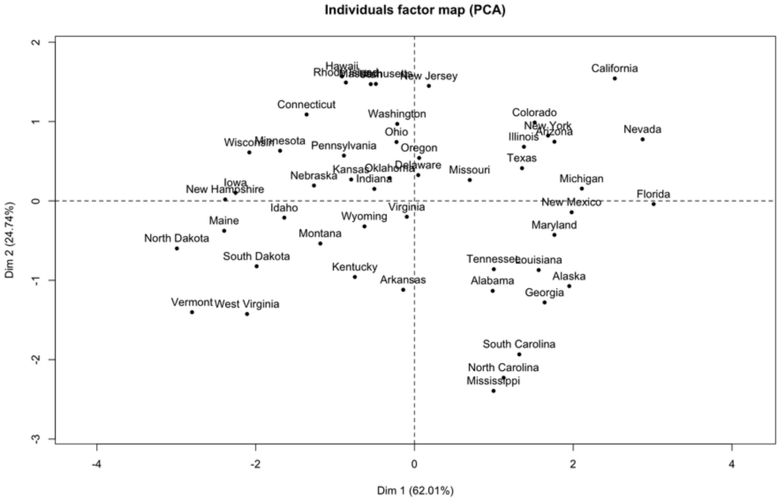
Pricipal Component Analysis

Снижение размерности



PCA

Визуализация многомерных данных



Регрессия
Категориальные признаки

Бинарное кодирование:

- Пусть j-й признак - категориальный: $f_j(x) = \{c_1, ..., c_n\}$

Категориальные признаки

Бинарное кодирование:

- Пусть j-й признак - категориальный: $f_j(x) = \{c_1, ..., c_n\}$
- вводятся n новых бинарных признаков: $b_1(x), ..., b_n(x)$

Бинарное кодирование:

- Пусть j-й признак - категориальный: $f_j(x) = \{c_1, ..., c_n\}$
- вводятся n новых бинарных признаков: $b_1(x), ..., b_n(x)$
- $b_i(x) = |f_j(x) = c_i|$

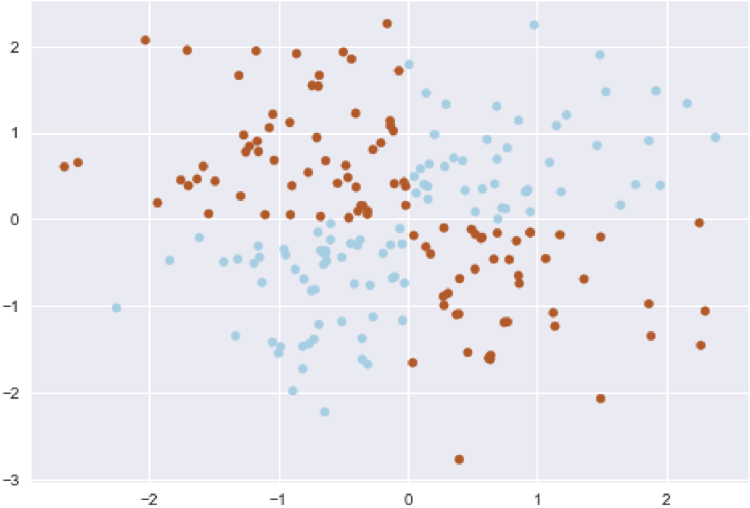
Бинарное кодирование:

- Пусть j-й признак - категориальный: $f_j(x) = \{c_1, ..., c_n\}$
- вводятся n новых бинарных признаков: $b_1(x), ..., b_n(x)$
- $b_i(x) = |f_j(x) = c_i|$
- **Вопрос:** как быть с $n+1$-м на рантайме?

Бинарное кодирование:

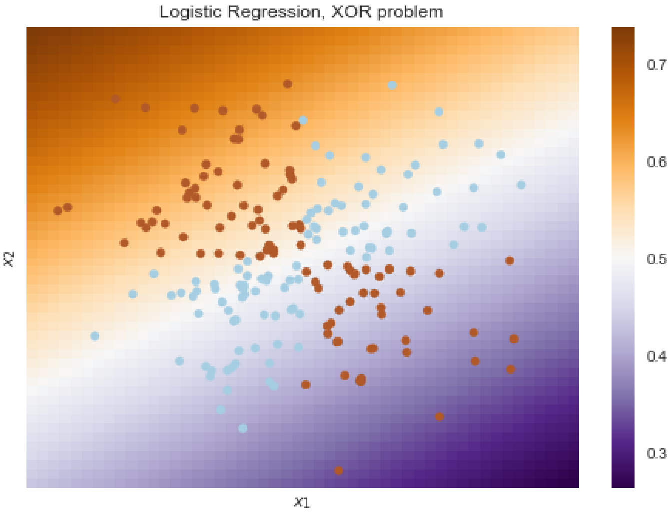
- Пусть j-й признак - категориальный: $f_j(x) = \{c_1, ..., c_n\}$
- вводятся n новых бинарных признаков: $b_1(x), ..., b_n(x)$
- $b_i(x) = |f_j(x) = c_i|$
- **Вопрос:** как быть с $n+1$-м на рантайме?
- $b_1 = b_2 = ... = b_n = 0$

Проблемы с регрессией
XOR-проблема



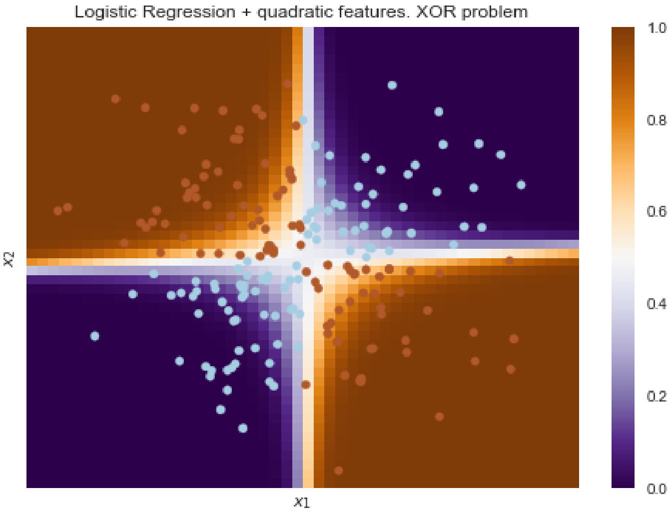
Проблемы с регрессией

XOR-проблема



Проблемы с регрессией

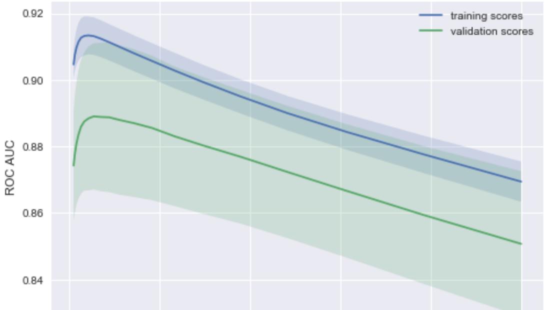
XOR-проблема



Как генерировать признаки?

- Квадратичные признаки: $(x_1, \dots, x_d) \to (x_1, \dots, x_d, x_2x_1, \dots, x_2x_d, x_1x_2, \dots, x_{d-1}x_d)$
- Полиномиальные признаки: $(x_1, \dots, x_d) \to (x_1, \dots, x_d, \dots, x_ix_j, \dots, x_ix_jx_k, \dots)$
- Логарифмирование: $x_i \to (x_i, log(|x_i| + 1))$

- Сделать модель сложнее или упростить?
- Добавить больше признаков?
- Или нам просто нужно больше данных для обучения?



- Для простых моделей тренировочная и валидационная ошибка находятся где-то рядом, и они велики. Это говорит о том, что модель **недообучилась**: то есть она не имеет достаточное кол-во параметров.
- Для сильно усложненных моделей тренировочная и валидационная ошибки значительно отличаются.

Сколько нужно данных?

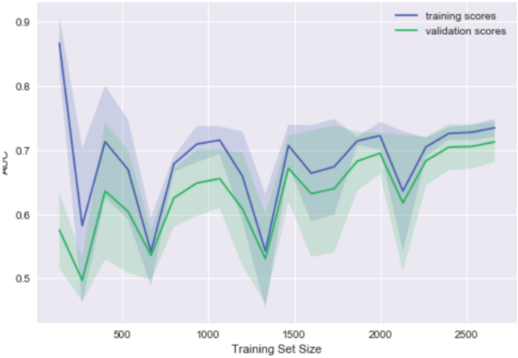


Рис.: Выставим τ большим

Сколько нужно данных?

Рис.: Выставим τ маленьким

Резюме:

Плюсы:

- Хорошо изучены
- Очень быстрые, могут работать на очень больших выборках
- Практически вне конкуренции, когда признаков очень много (от сотен тысяч и более), и они разреженные.
- Коэффициенты перед признаками могут интерпретироваться
- Логистическая регрессия выдает вероятности отнесения к разным классам.
- Модель может строить и нелинейную границу

Минусы:

- Плохо работают в задачах, в которых зависимость ответов от признаков сложная, нелинейная
- На практике предположения теоремы Маркова-Гаусса почти никогда не выполняются, поэтому чаще линейные методы работают хуже, чем,

Practice

- Идем сюда https://www.kaggle.com/c/bike-sharing-demand
- Собираем регрессию из scikit-learn с категориальными признаками
- Находим признак, который необходимо удалить из датасета (он почти повторяет целевой)
- Собираем без категориальных признаков регрессию
- Собираем с one-hot-encode признаками регрессию
- Сравниваем по MSE
- Находим способ найти и убрать лишние признаки
- Опять сравниваем