«Машинное обучение»

Линейная регрессия

Александр Дьяконов

План

Линейная регрессия
Линейная регрессия от одной переменной
Общий случай (многих переменных)
Решение задачи минимизации: прямой метод
Проблема вырожденности матрицы
Регуляризация (гребневая регрессия, LASSO, Elastic Net)
Градиентный метод обучения
Приложения

Линейная регрессия

Гипотеза о линейной зависимости целевой переменной, ищем решение в виде:

$$a(X_1,\ldots,X_n) = w_0 + w_1 X_1 + \ldots + w_n X_n$$

Практика:

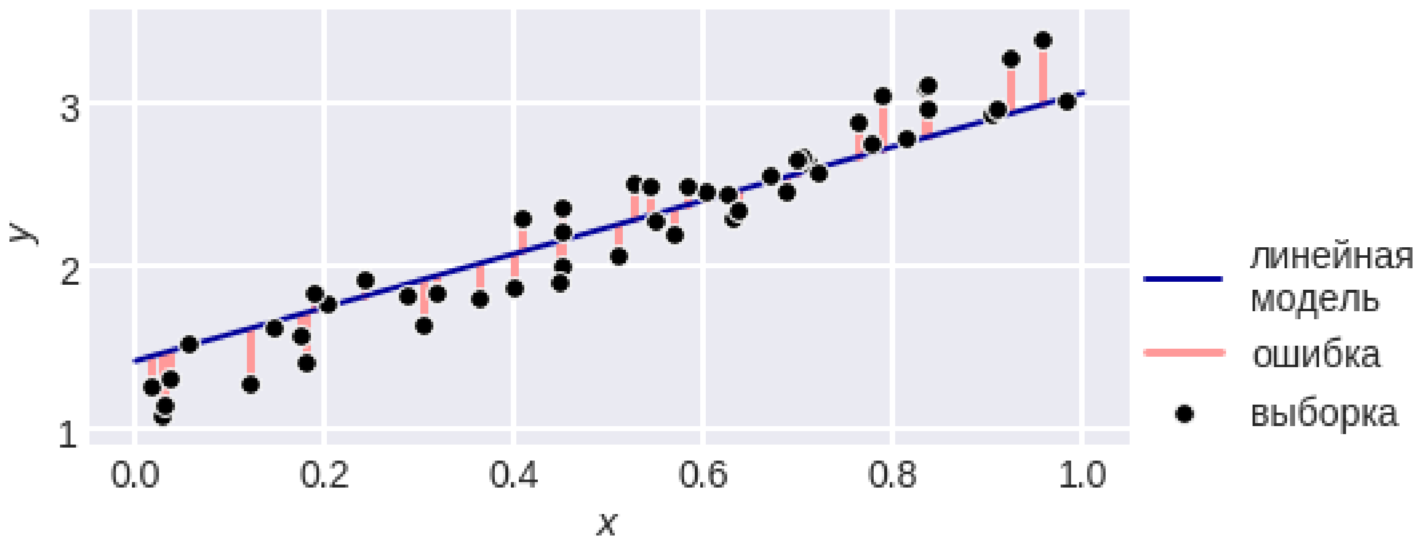
- часто неплохо работает и при монотонных зависимостях
- хорошо работает, когда есть много «однородных» зависимостей:

цель – число продаж
признак 1 – число заходов на страницу продукта
признак 2 – число добавлений в корзину
признак 3 – число появлений продукта в поисковой выдачи
...

Линейная регрессия от одной переменной

$$a(X_1) = w_0 + w_1 X_1$$

обучение: $\{(x_1, y_1), \ldots, (x_m, y_m)\},\ x_i \in \mathbb{R}$



хотели бы…

$$\begin{cases} w_0 + w_1 x_1 = y_1 \\ \cdots \\ w_0 + w_1 x_m = y_m \end{cases}$$

невязки / отклонения (residuals):

$$e_1 = y_1 - w_0 - w_1 x_1$$
$$\ldots$$
$$e_m = y_m - w_0 - w_1 x_m$$

Линейная регрессия от одной переменной

Задача минимизации суммы квадратов отклонений (residual sum of squares)

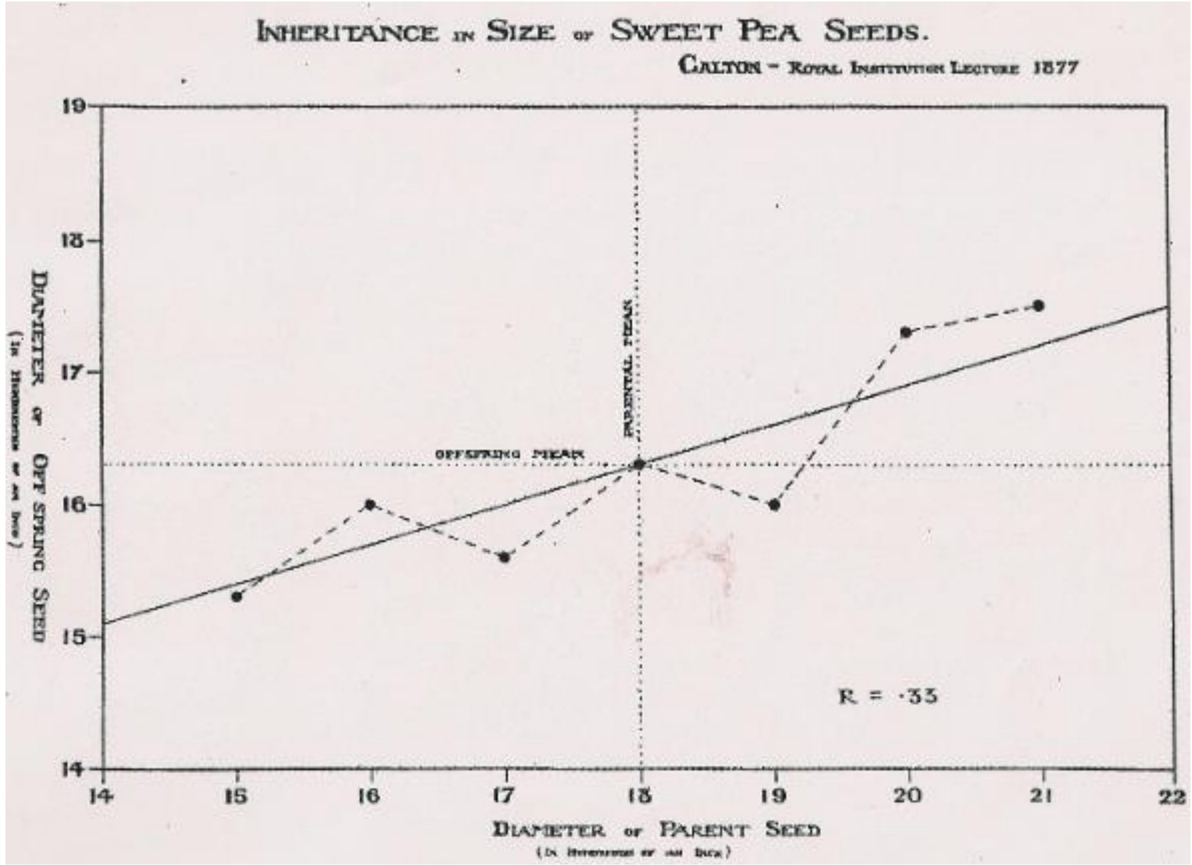
$$\mathrm{RSS} = e_1^2 + \ldots + e_m^2 \to \min$$

На это можно смотреть как на минимизацию эмпирического риска по параметрам $w = (w_0, w_1)$

$$L(w) = \sum_{i=1}^{m} (y_i - a_w(x_i))^2 = \sum_{i=1}^{m} (y_i - (w_0 + w_1 x_i))^2$$

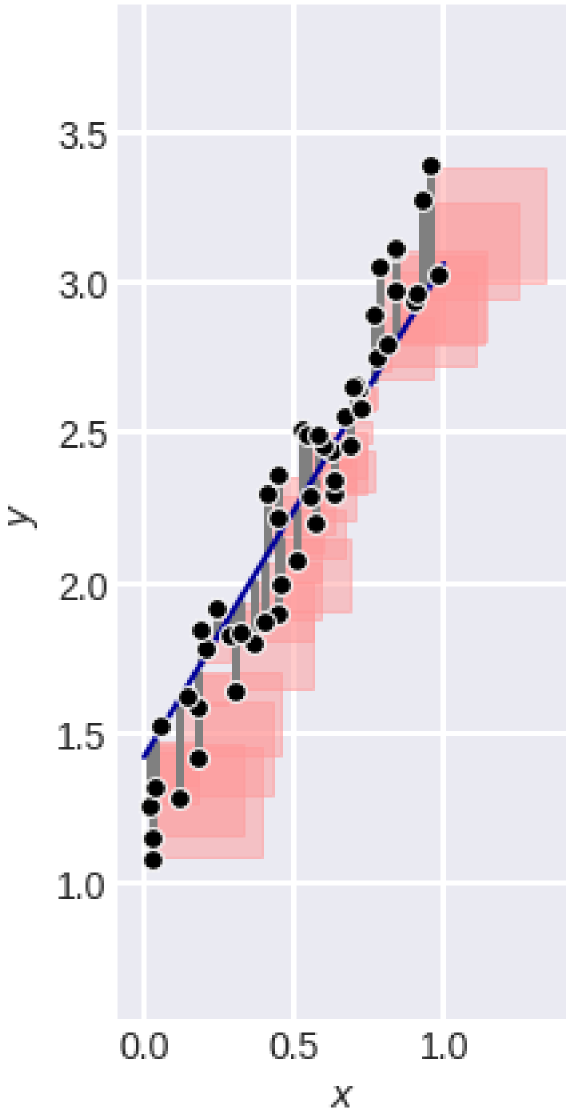
тут конкретная функция ошибки

Линейная регрессия от одной переменной

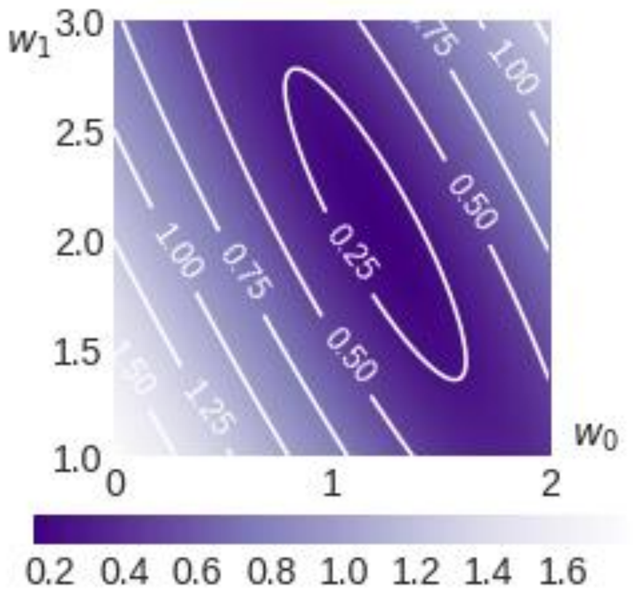


«Регрессия к посредственности при наследовании роста»
Francis Galton, 1877

Линейная регрессия от одной переменной: геометрический смысл ошибки



$$a(X_1) = w_0 + w_1 X_1$$



$$\sum_{i=1}^{m} (y_i - w_0 - w_1 x_i)^2$$

Отличается от суммы расстояний до поверхности!

Линейная регрессия от одной переменной

Нетрудно показать:

$$w_1 = \frac{\sum_{i=1}^{m}(x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{m}(x_i - \overline{x})^2} = \frac{\mathrm{cov}(\{x_i\},\{y_i\})}{\mathrm{var}(\{x_i\})},$$

$$w_0 = \overline{y} - w_1\overline{x},$$

где $\overline{x} = \frac{1}{m}\sum_{i=1}^{m} x_i, \ \overline{y} = \frac{1}{m}\sum_{i=1}^{m} y_i.$

Полученное уравнение прямой (проходит через «центр масс»):

$$(y - \overline{y}) = \frac{\mathrm{cov}(\{x_i\},\{y_i\})}{\mathrm{var}(\{x_i\})}(x - \overline{x})$$

Общий случай (многих переменных)

$$a(X_1,\ldots,X_n)=w_0+w_1X_1+\cdots+w_nX_n=x^{\mathrm{T}}w$$

веса (параметры) – $w=(w_0,w_1,\ldots,w_n)^{\mathrm{T}}$

объект – $x=(X_0,X_1,\ldots,X_n)^{\mathrm{T}}$

для удобства записи вводим фиктивный признак $X_0\equiv 1$

обучение: $\{(x_1,y_1),\ldots,(x_m,y_m)\}$, $x_i\in\mathbf{R}^{n+1}$,

опять хотим решить $Xw=y$**:**

$$\begin{cases} x_1^{\mathrm{T}}w=y_1 \\ \cdots \\ x_m^{\mathrm{T}}w=y_m \end{cases}$$

как решать?

Общий случай (многих переменных): в матричной форме

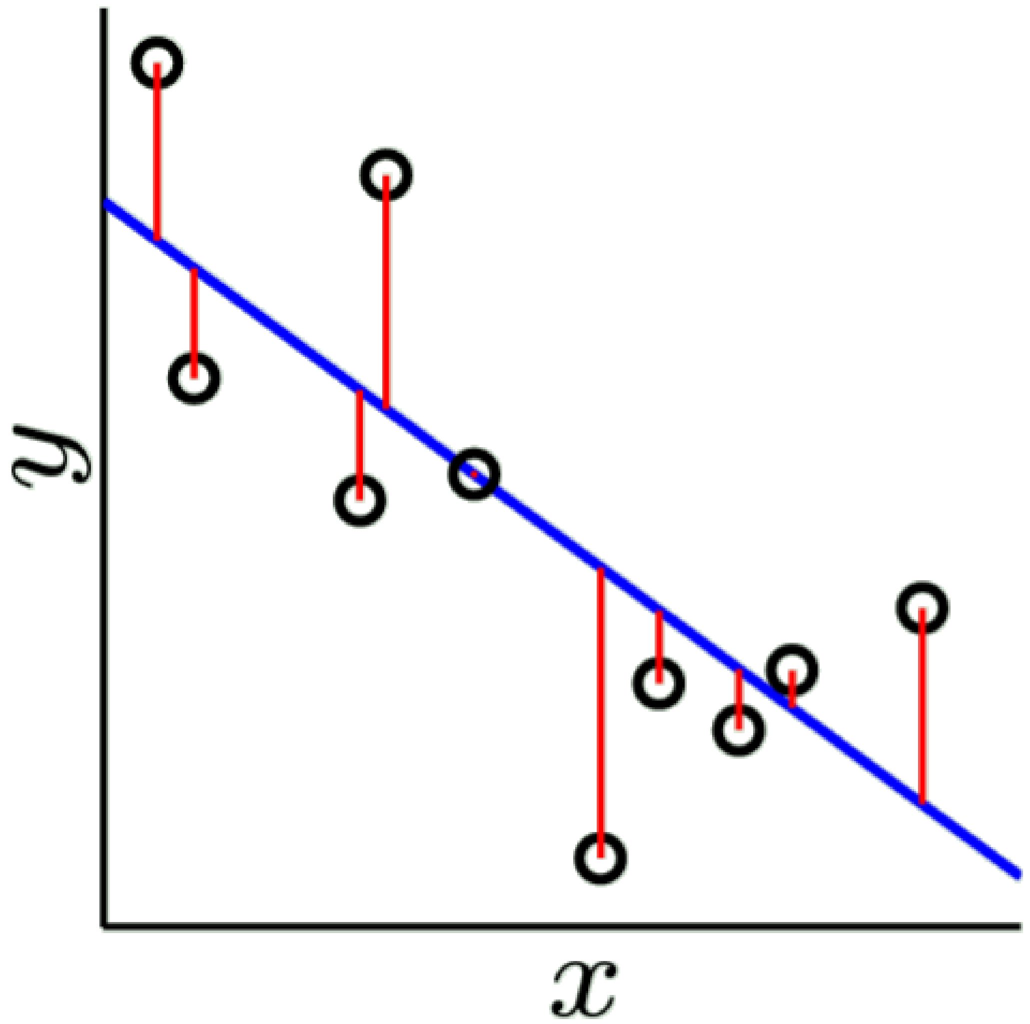
$$Xw = y$$

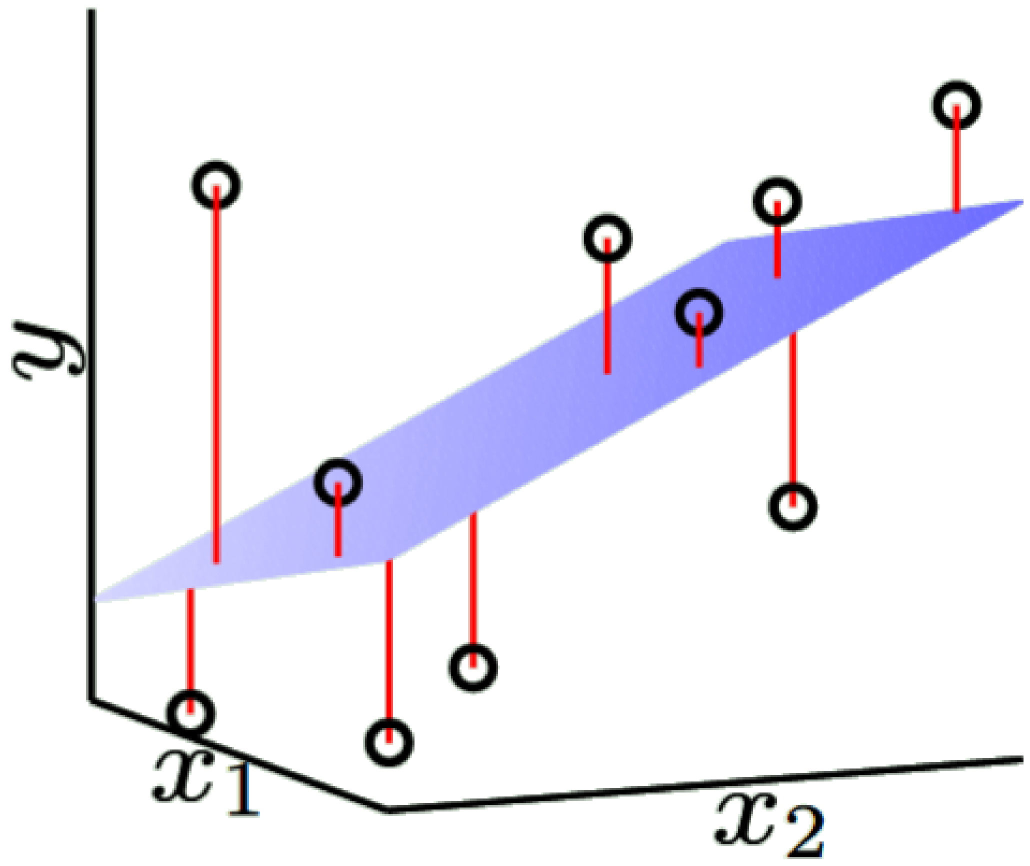
в матрице X по строкам записаны описания объектов,
в векторе y значения их целевого признака
(здесь есть коллизия в обозначении y)

будем решать так:

$$\| Xw - y \|_2^2 = \sum_{i=1}^{m} (x_i^{\mathrm{T}} w - y_i)^2 \to \min_w$$

Общий случай (многих переменных): геометрический смысл





Кстати, полученная задача оптимизации выпукла, единственный глобальный минимум

(кроме вырожденного случая)

Решение задачи минимизации: прямой метод

$$\| Xw - y \|_2^2 \to \min_w$$

$$\| Xw - y \|_2^2 = (Xw - y)^{\mathrm{T}}(Xw - y) = w^{\mathrm{T}}X^{\mathrm{T}}Xw - w^{\mathrm{T}}X^{\mathrm{T}}y - y^{\mathrm{T}}Xw + y^{\mathrm{T}}y$$

$$\nabla \| Xw - y \|_2^2 = 2X^{\mathrm{T}}Xw - 2X^{\mathrm{T}}y = 0$$

$$X^{\mathrm{T}}Xw = X^{\mathrm{T}}y$$

решение существует, если столбцы л/н

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

помним, что $\mathrm{rg}(X^{\mathrm{T}}X) = \mathrm{rg}(X)$

$(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}$ **– псевдообратная матрица Мура-Пенроуза**

обобщение обратной на неквадратные матрицы

Обобщённая линейная регрессия: вместо X – что угодно

выражаем целевое значение через л/к базисных функций
(они фиксированы)

$$a(X_1,\ldots,X_n)=w_0+w_1\varphi_1(X_1,\ldots,X_n)+\cdots+w_k\varphi_k(X_1,\ldots,X_n)$$

$$w=(w_0,w_1,\ldots,w_k)^{\mathrm{т}}$$
$$x=(X_0,X_1,\ldots,X_n)^{\mathrm{т}}$$

$$\varphi(x)=(\underbrace{\varphi_0(x)}_{\equiv 1},\varphi_1(x),\ldots,\varphi_k(x))^{\mathrm{т}}$$

$$a(x)=\sum_{i=1}^{k}w_i\varphi_i(x)=\varphi(x)^{\mathrm{т}}w$$

$$\varphi(X)=\begin{bmatrix}\varphi_0(x_1) & \cdots & \varphi_k(x_1)\\ \cdots & \cdots & \cdots\\ \varphi_0(x_m) & \cdots & \varphi_k(x_m)\end{bmatrix}$$

$$\|\varphi(X)w-y\|_2^2\to\min_w$$

Проблема вырожденности матрицы

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

Только ли вырожденность плоха?

Что делать?

Проблема вырожденности матрицы

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

Проблемы, когда матрица $X^{\mathrm{T}} X$ плохо обусловлена...

$$\mu(X^{\mathrm{T}} X) = \| X^{\mathrm{T}} X \| \cdot \| (X^{\mathrm{T}} X)^{-1} \| = \frac{\lambda_{\max}(X^{\mathrm{T}} X)}{\lambda_{\min}(X^{\mathrm{T}} X)}$$

Решения:

1. **Регуляризация – здесь и в «сложности»**
2. **Селекция (отбор) признаков – «селекция»**
3. **Уменьшение размерности (в том числе, PCA) – USL**
4. **Увеличение выборки**

Регуляризация: упрощённое объяснение смысла

$$a(X_1,\ldots,X_n) = w_0 + w_1X_1 + \ldots + w_nX_n$$

если есть два похожих объекта, то должны быть похожи метки, пусть отличаются в j-м признаке, тогда ответы модели отличаются на

$$\varepsilon_j w_j$$

Поэтому не должно быть <u>очень больших по модулю весов</u>
(у признаков, по которым могут отличаться похожие объекты)

Поэтому вместе с $\|Xw - y\|_2^2 \to \min$
хотим $\|w\|_2^2 \to \min$

Не на все коэффициенты нужна регуляризация! Почему?

Регуляризация: упрощённое объяснение смысла

Пусть есть какая-то зависимость и лишние признаки, например

$$y = X_1 \text{ и } X_2 = X_3$$

Можно получить ответ в таком виде:

$$a = X_1 + w'X_2 - w'X_3$$

Если теперь $X_2 \approx X_3$, **тогда** $\varepsilon = X_2 - X_3$

$$a = X_1 + w'\varepsilon$$

– может быть сколь угодно большим при больших (по модулю) w'

аналогично при линейных зависимостях!

Регуляризация

Иванова

$$\begin{cases} \| Xw - y \|_2^2 \to \min \\ \| w \|_2^2 \le \lambda \end{cases}$$

Тихонова

$$\| Xw - y \|_2^2 + \lambda \| w \|_2^2 \to \min$$

Удобнее: безусловная оптимизация

$\| w \|_2^2 = w_1^2 + w_2^2 + \ldots + w_n^2$ **– нет** w_0^2

эти две формы эквивалентны: решение одного можно получить как решение другого

На самом деле, регуляризация упрощает модель

Регуляризация и гребневая регрессия

$$\arg\min_{w} \| Xw - y \|_2^2 + \lambda \| w \|_2^2 = (X^{\mathrm{т}} X + \lambda I)^{-1} X^{\mathrm{т}} y$$

Доказать!

$$\lambda \geq 0$$

Такая регрессия называется гребневой регрессией (Ridge Regression)

Виден другой смысл регрессии: складываем две матрицы Грама, неотрицательно определённая + положительно определённая – боремся с вырожденностью матрицы

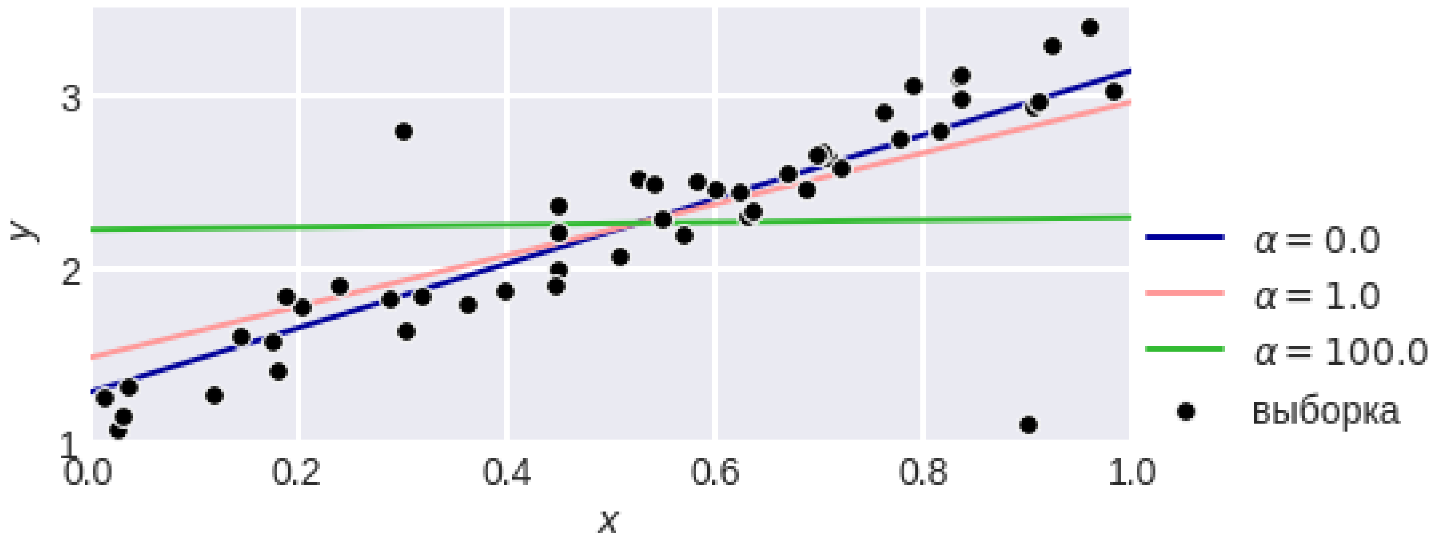
Коэффициент регуляризации (shrinkage penalty)

$\lambda = 0$ **– получаем классическое решение**

$\lambda \to +\infty$ **– меньше «затачиваемся на данные» и больше регуляризуем**

значение параметра регуляризации можно выбрать на скользящем контроле

Минутка кода: регуляризация и гребневая регрессия



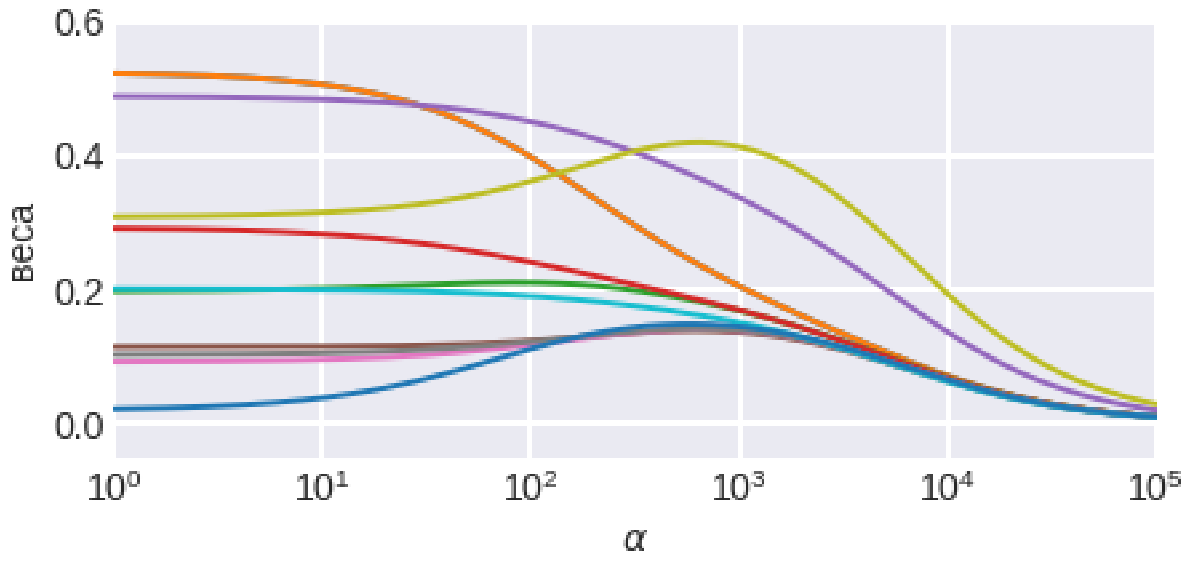

```
from sklearn.linear_model import Ridge

model = Ridge(alpha=0.0) # ридж-регрессия
# обучение
model.fit(x_train[:, np.newaxis], y_train)
# обратите внимание: np.newaxis
# контроль
a_train = model.predict(x_train[:, np.newaxis])
a_test = model.predict(x_test[:, np.newaxis])
```

Регуляризация и гребневая регрессия

$$\sum_{i=1}^{m}\left(y_i - a(x_i)\right)^2 + \lambda \sum_{j=1}^{n} w_j^2 \to \min$$

$$\lambda \geq 0$$

добавление shrinkage penalty (регуляризатора)



параметр регуляризации может подбираться с помощью скользящего контроля

Регуляризация и гребневая регрессия

Для ridge-регрессии нужна правильная нормировка признаков!
Нет инвариантности (в отличие от линейной) от умножения признаков на скаляры

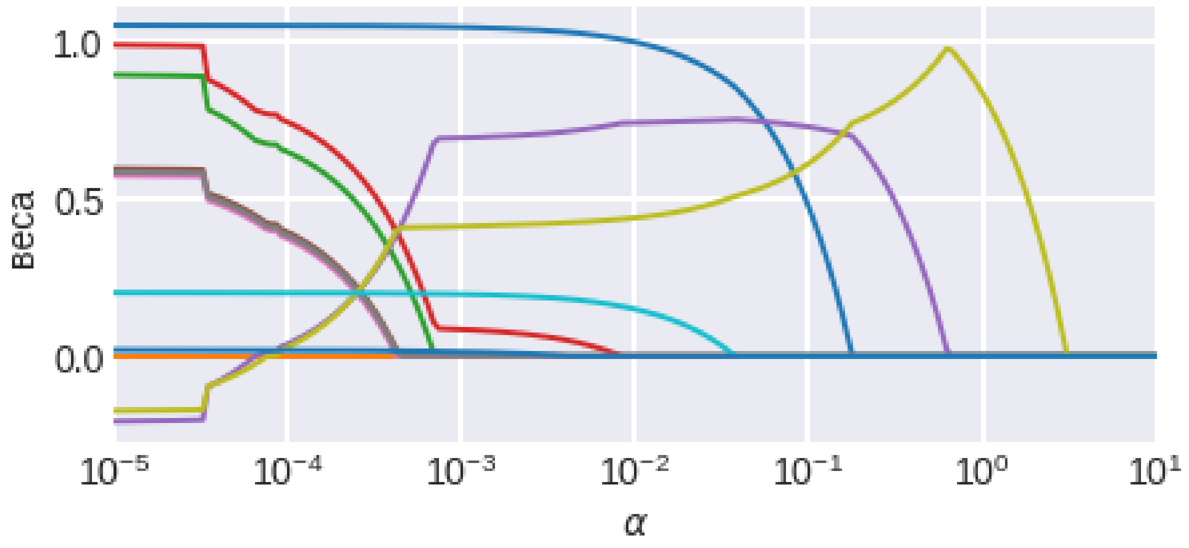
Перед регуляризацией – стандартизация!!!

LASSO (Least Absolute Shrinkage and Selection Operator)

Попробуем другой «штраф за сложность» (сейчас поймём название)

$$\sum_{i=1}^{m}(y_i - a(x_i))^2 + \lambda\sum_{j=1}^{n}|w_j| \to \min$$

$$\lambda \geq 0$$



Здесь значения коэффициентов существенно меньше (т.к. при $\Sigma|\cdot|$, а не $\Sigma(\cdot)^2$)

Здесь коэффициенты интенсивнее зануляются при увеличении $\lambda \geq 0$

Семейство регуляризированных линейных методов

Ridge

$$\| y - Xw \|_2^2 + \lambda \| w \|_2^2 \to \min_w$$

LASSO (Least Absolute Shrinkage and Selection Operator)

$$\| y - Xw \|_2^2 + \lambda \| w \|_1 \to \min_w$$

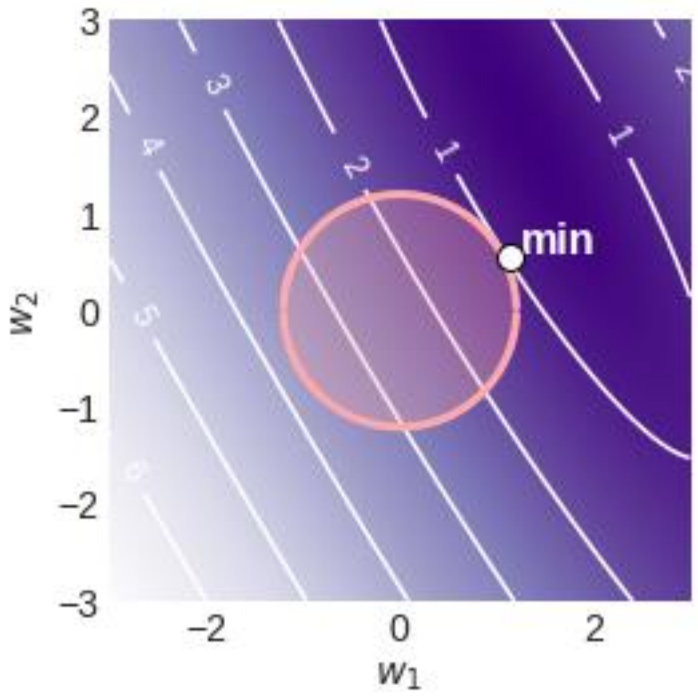
Elastic Net = LASSO + Ridge

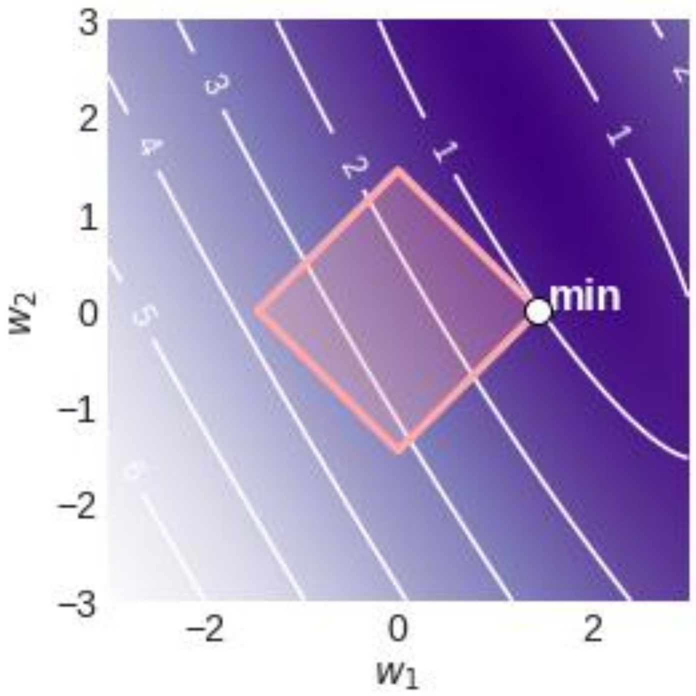
$$\| y - Xw \|_2^2 + \lambda_1 \| w \|_1 + \lambda_2 \| w \|_2^2 \to \min_w$$

Геометрический смысл Ridge и LASSO

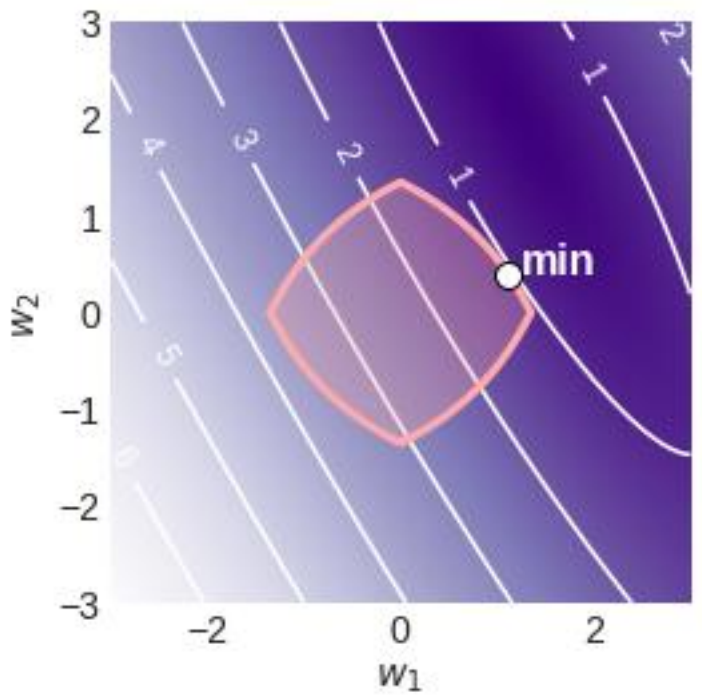
$$\sum_{i=1}^{m}\left(y_i - w_0 - \sum_{j=1}^{n} w_j x_{ij}\right)^2 \to \min_w, \quad \sum_{j=1}^{n} w_j^2 \le s$$

$$\sum_{i=1}^{m}\left(y_i - w_0 - \sum_{j=1}^{n} w_j x_{ij}\right)^2 \to \min_w, \quad \sum_{j=1}^{n} | w_j | \le s$$





Геометрический смысл Elastic Net

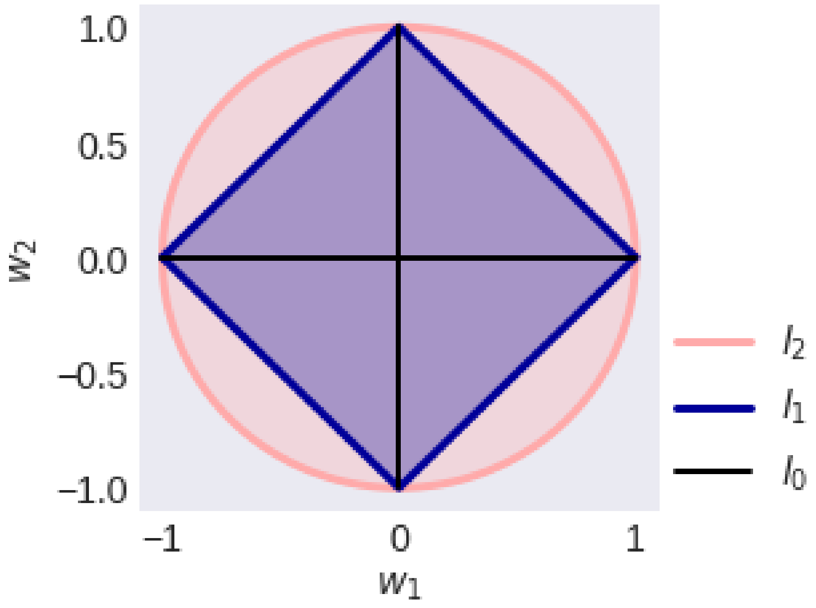


на практике часто модель и не может зависеть от небольшого числа переменных

Почему L1-норма ⇒ разреженность

1. Больше вероятность, что линии уровней функции ошибки касаются области ограничений в точках с нулевыми координатами

2. L1-норма больше похожа на L0, чем L2



$$\|w\|_0 = |\{t \mid w_t \neq 0\}|$$

При увеличении коэффициента регуляризации веса стремятся к нулю

Обеспечивается автоматическая селекция признаков!

Проблема вырожденности / плохой обусловленности матрицы

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

Решения:

1. Регуляризация
2. Селекция (отбор) признаков
3. Уменьшение размерности (в том числе, PCA)
4. Увеличение выборки

Какие признаки включить в модель

$$a(X_1, \ldots, X_n) = w_0 + w_1 X_1 + \ldots + w_n X_n$$

Маленький обзор стратегий:

1 стратегия – **перебор** – умный перебор подмножества признаков

2 стратегий – **оценка** – оценка качества признаков (фильтры)

3 стратегия – **автомат** – встроенные методы (ex: LASSO)

Проблема вырожденности матрицы

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

Решения:

1. Регуляризация
2. Селекция (отбор) признаков
3. Уменьшение размерности (в том числе, PCA)
4. Увеличение выборки

	x1	x2	x3	y
0	0.44	0.62	0.51	-0.25
1	0.03	0.53	0.07	-0.51
2	0.55	0.13	0.43	0.41
3	0.44	0.51	0.10	0.04
4	0.42	0.18	0.13	0.12
5	0.33	0.79	0.60	-0.45

	x1-x2	y
0	-0.18	-0.25
1	-0.50	-0.51
2	0.42	0.41
3	-0.07	0.04
4	0.24	0.12
5	-0.46	-0.45

обоснование необходимости аналогично селекции

Проблема вырожденности матрицы

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

Решения:

1. Регуляризация
2. Селекция (отбор) признаков
3. Уменьшение размерности (в том числе, PCA)
4. **Увеличение выборки**

на модельном примере:

$$m \le n \quad \Rightarrow \quad \mathrm{rg}(X^{\mathrm{T}} X)_{(n+1)\times(n+1)} < n+1$$

+ при увеличении выборки могут исчезнуть линейные зависимости между столбцами

Линейная регрессия: градиентный метод обучения

$$\frac{1}{2}\sum_{i=1}^{m}(a(x_i \mid w) - y_i)^2 \to \min$$

$$a(x \mid w) = w^{\mathrm{T}} x$$

недостатки прямого…

работа с большими матрицами (тем более обращение)

Оптимизация: градиент

$\nabla f(w_0)$ – **направление наискорейшего возрастания функции**

$$f(w) = f(w_0) + (w - w_0)^{\mathrm{т}} \nabla f(w_0) + o(\| w - w_0 \|)$$

$$f(w) - f(w_0) \approx (w - w_0)^{\mathrm{т}} \nabla f(w_0)$$

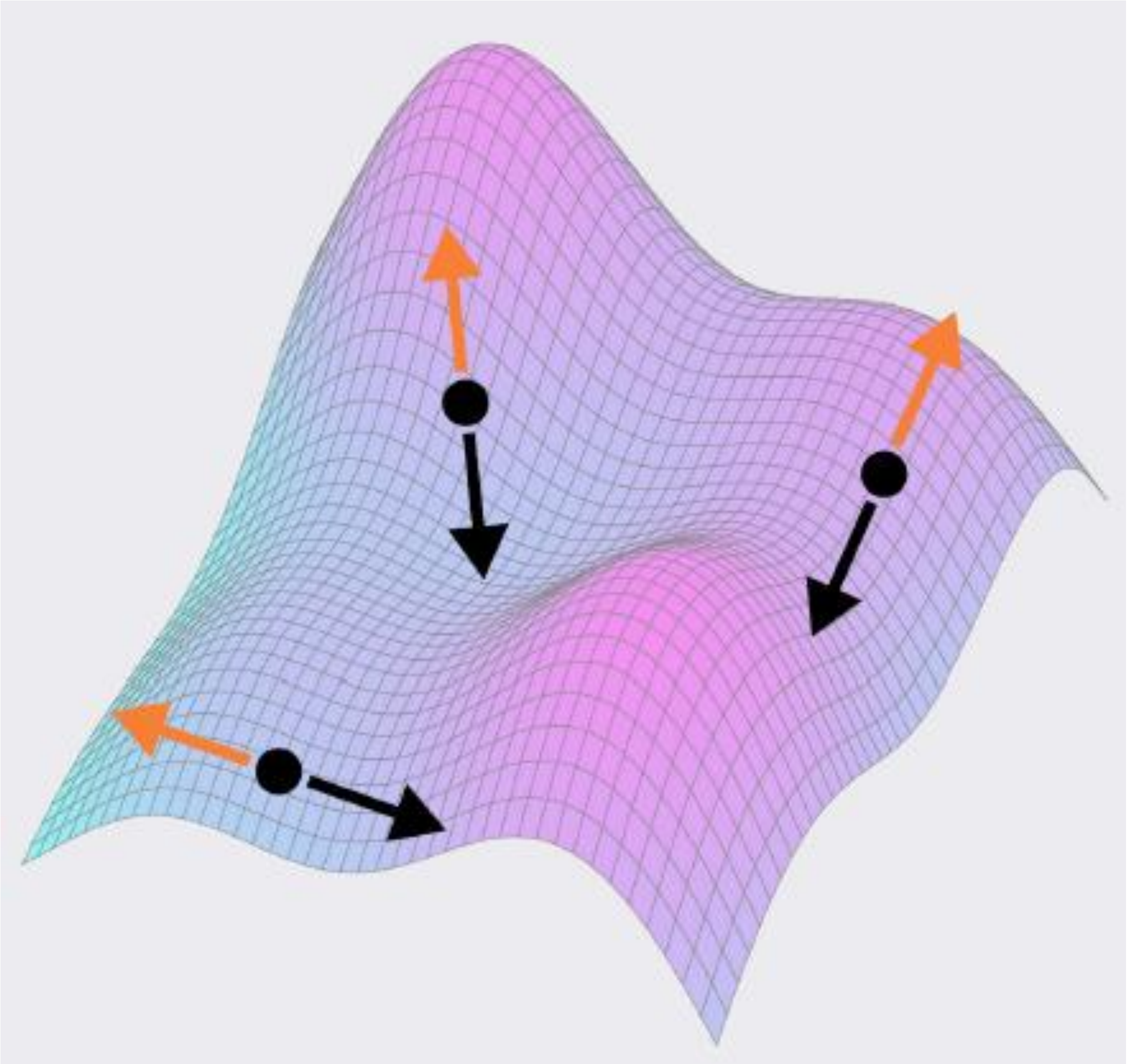
если выбирать из всех векторов $w - w_0$ **единичной нормы,**

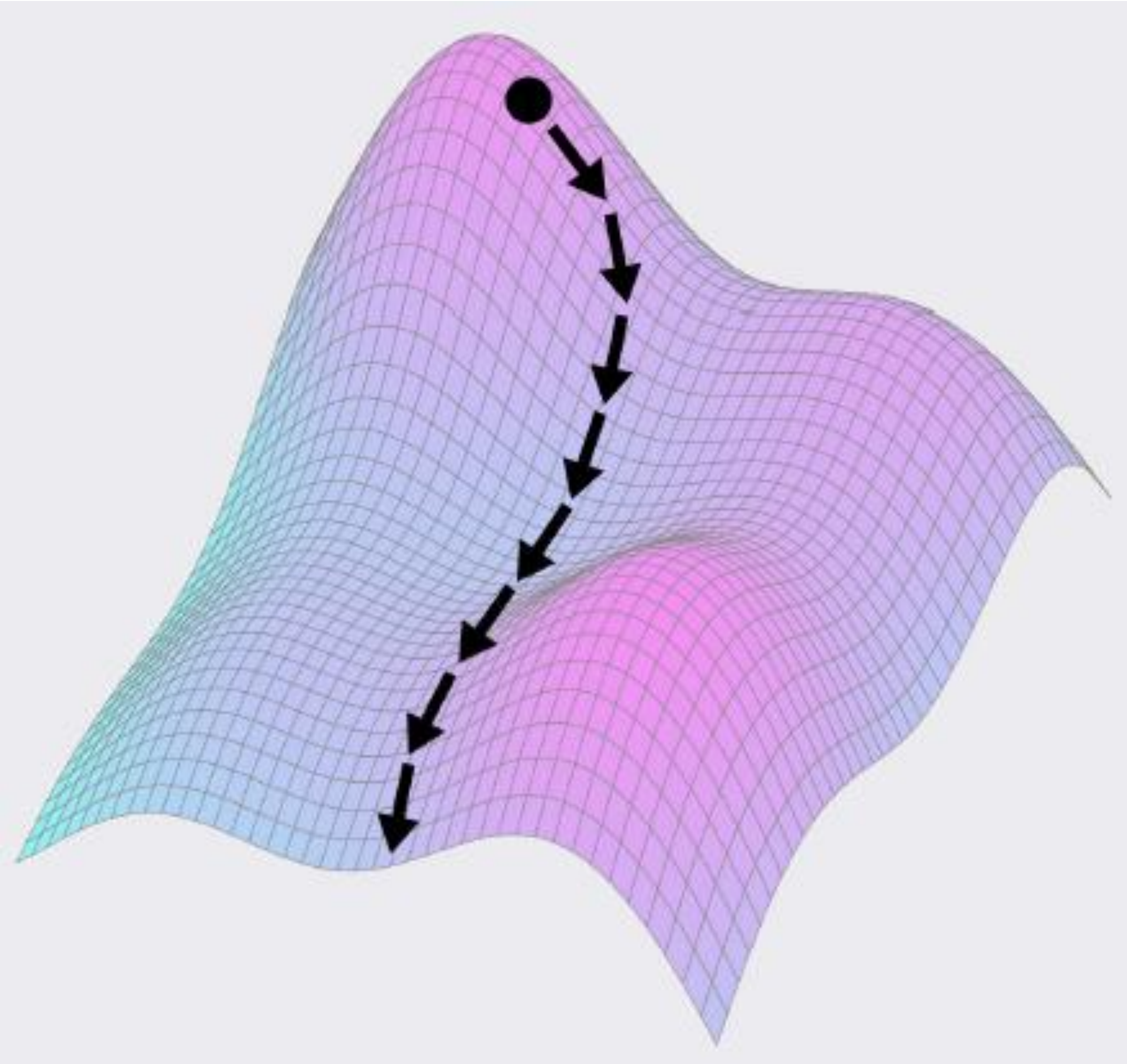
то по неравенству К-Б-Ш

$$| (w - w_0)^{\mathrm{т}} \nabla f(w_0) | \le 1 \| \nabla f(w_0) \| = \frac{\nabla f(w_0)^{\mathrm{т}}}{\| \nabla f(w_0) \|} \nabla f(w_0)$$

Антиградиент $(-\nabla f(w_0))$ – **направление наискорейшего убывания функции**

Оптимизация: градиент и антиградиент





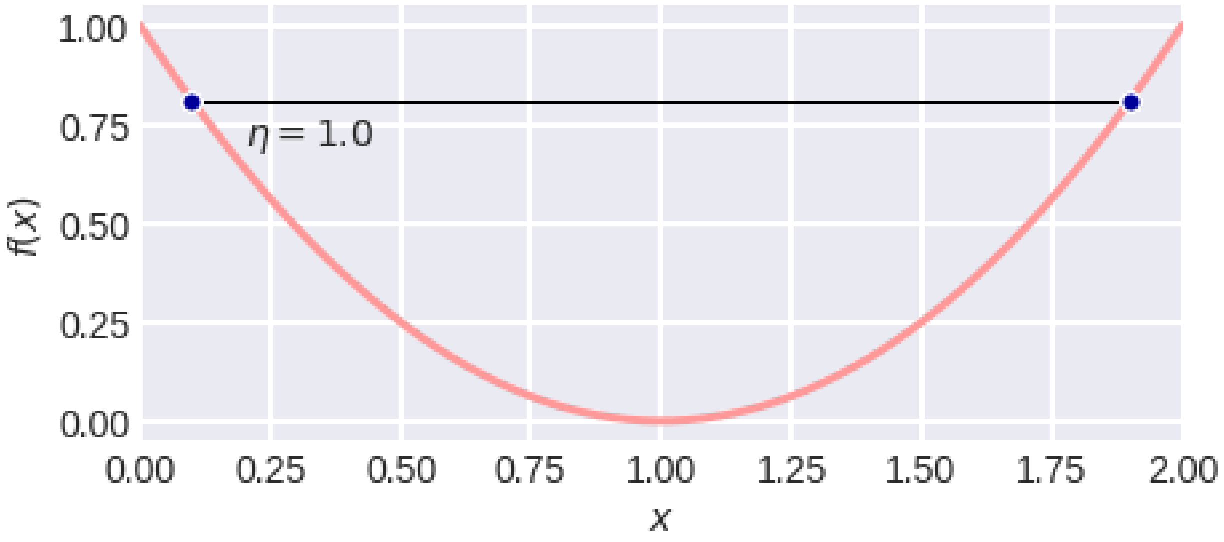
[Glassner]

Оптимизация: градиентный спуск (GD = Gradient Descent)

$$w^{(t+1)} = w^{(t)} - \eta \nabla L(w^{(t)})$$

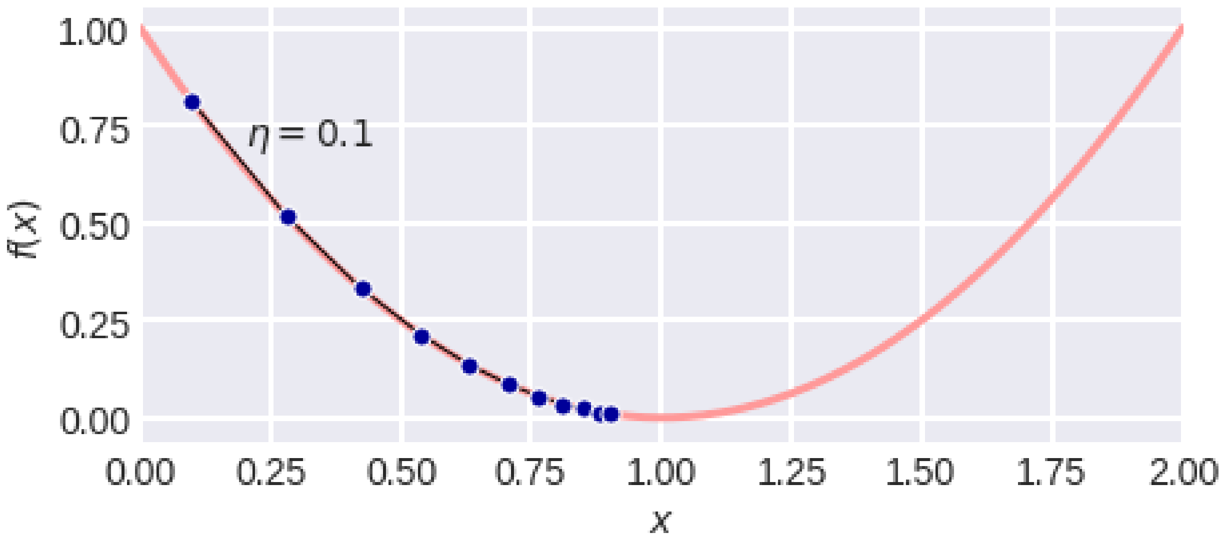
$\eta > 0$ **– шаг / темп обучения (step size / learning rate)**

Хотим $\lim\limits_{t\to\infty} w^{(t)} = \arg\min\limits_{w} L(w)$

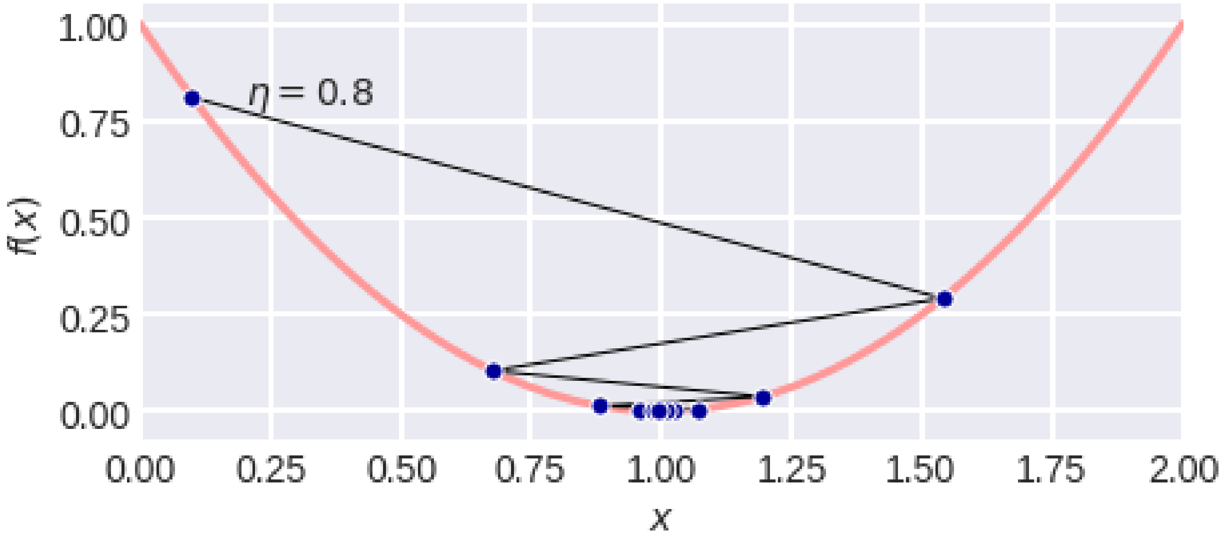


неудачно выбран темп

Оптимизация: градиентный спуск: проблема выбора темпа

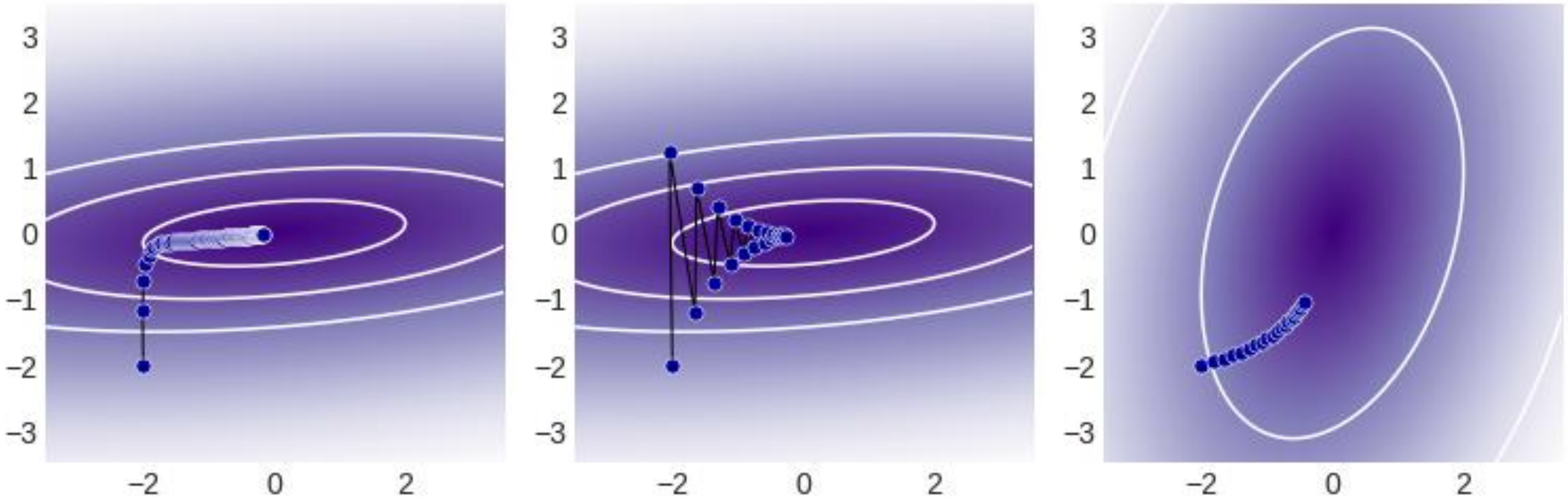


темп, возможно, маленький



темп, возможно, большой

Оптимизация: проблема масштаба признаков



вот для чего нормируют признаки

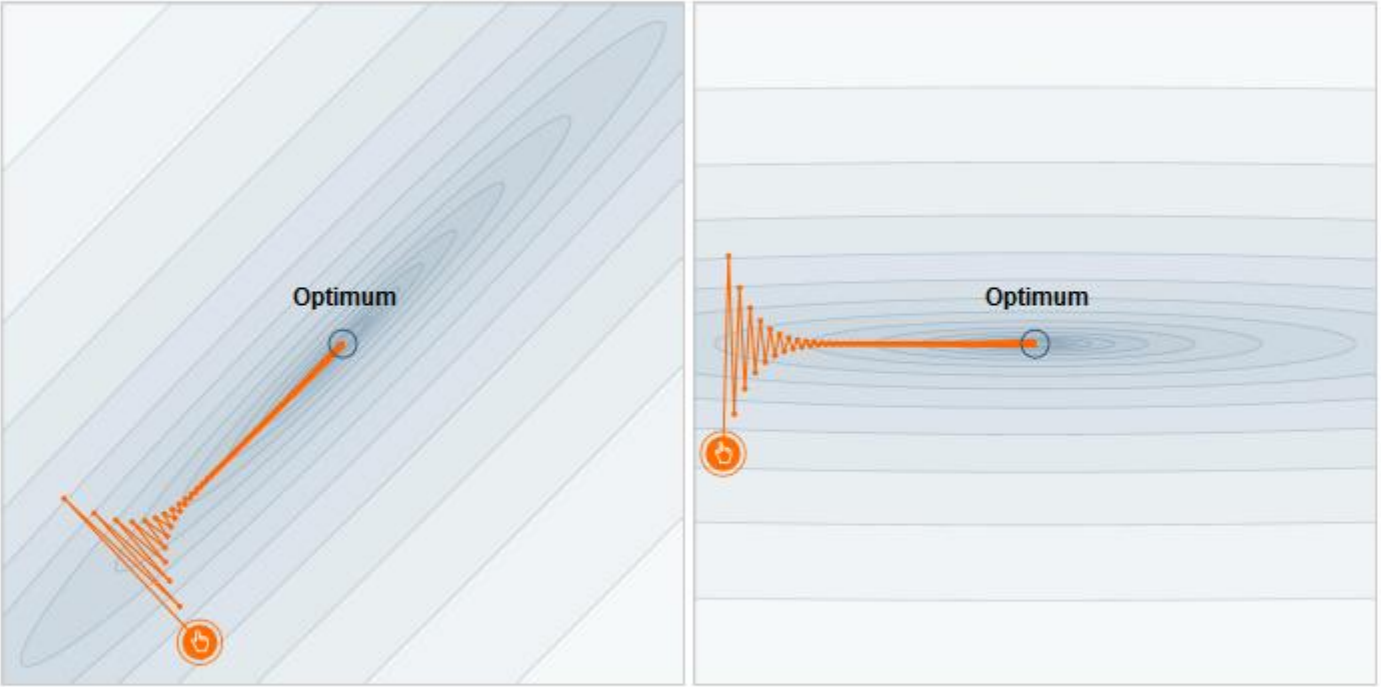
Оптимизация: свойства градиентного спуска

\+ если функция выпуклая градиентный спуск сойдётся в минимум (при правильном выборе шагов)

– если нет – в один из локальных минимумов

\+ простой метод

\+ при модификации можно использоваться в онлайн-режиме



https://distill.pub/2017/momentum/

Стохастический градиентный спуск (SGD = Stochastic gradient descent)

Если есть «большая» сумма (пока без регуляризации)

$$L(w)=\sum_{t=1}^{m}L_t(w)$$

Слишком долго вычислять полный градиент!

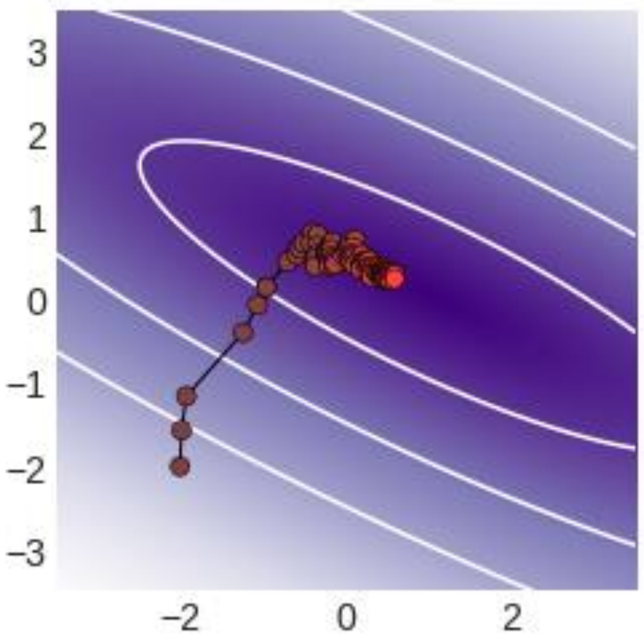
Не вычисляем полный градиент:

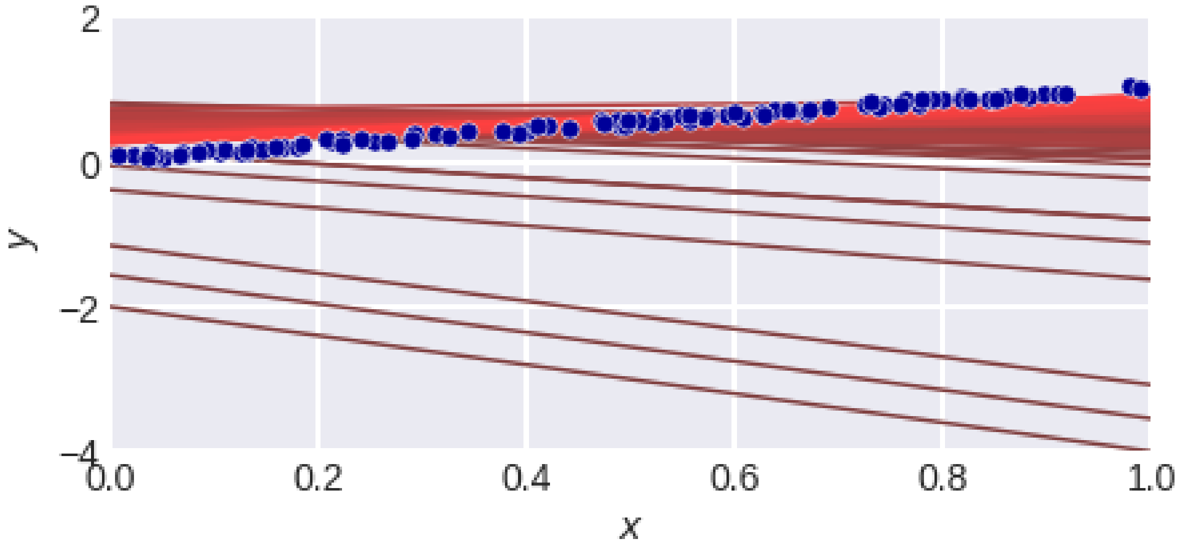
$$\nabla L(w)=\sum_{t=1}^{m}\nabla L_t(w)$$

А выцепляем случайное (!) слагаемое $i=i(t)\in\{1,2,\ldots,m\}$ **и делаем шаг с помощью такого частичного антиградиента:**

$$w^{(t+1)}=w^{(t)}-\eta\nabla L_i(w^{(t)})$$

Стохастический градиентный спуск (SGD)





- **Можно учиться в online-режиме**

(когда функция становится известна по частям – некоторые слагаемые),

но порядок здесь не совсем случайный

- **Метод быстрый**

(не надо вычислять градиенты всех слагаемых на каждом шаге)

темп сходимости определяется из графиков изменения ошибки

Линейная регрессия: градиентный метод обучения

В лекции «оптимизация»…

$$\frac{1}{2}\sum_{i=1}^{m}(a(x_i\,|\,w)-y_i)^2 \to \min$$

$$w^{(t+1)} = w^{(t)} - \eta\sum_{i=1}^{m}(a(x_i\,|\,w^{(t)})-y_i)\frac{\partial a(x_i\,|\,w^{(t)})}{\partial w}$$

$$a(x\,|\,w) = w^{\mathrm{T}}x$$

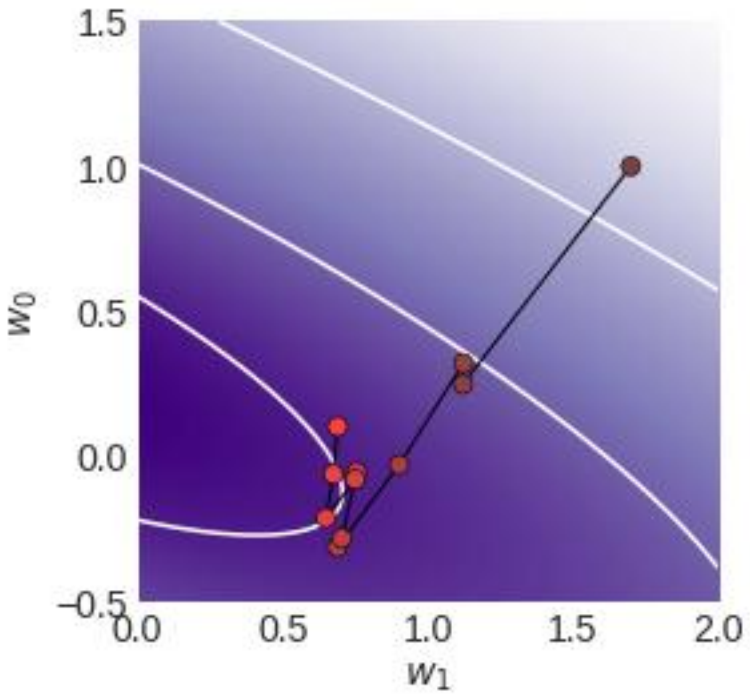
Gradient Descent

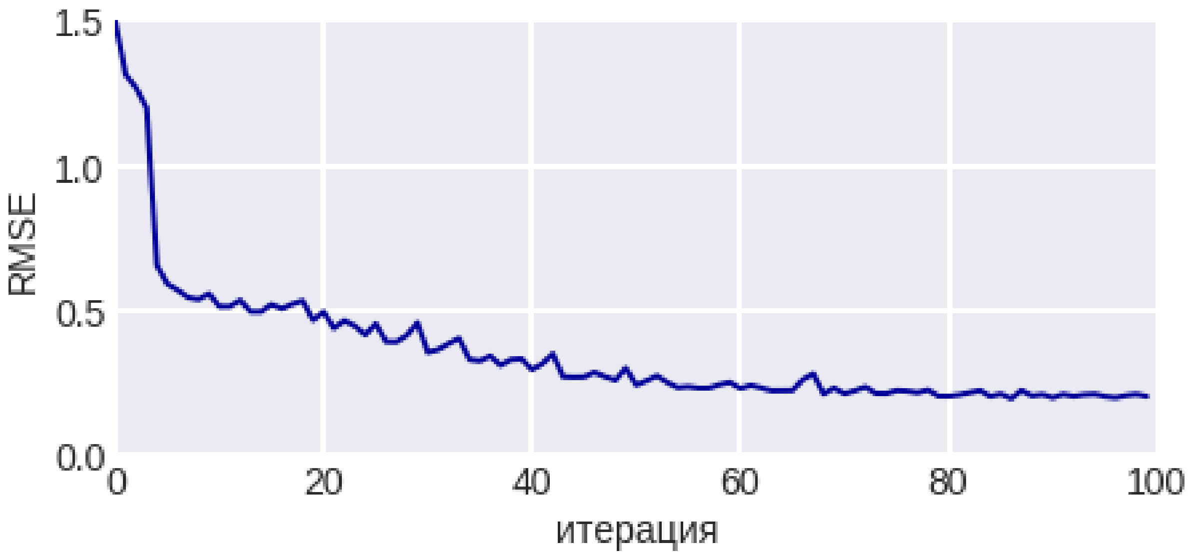
$$w^{(t+1)} = w^{(t)} - \eta\sum_{i=1}^{m}(a(x_i\,|\,w^{(t)})-y_i)x_i$$

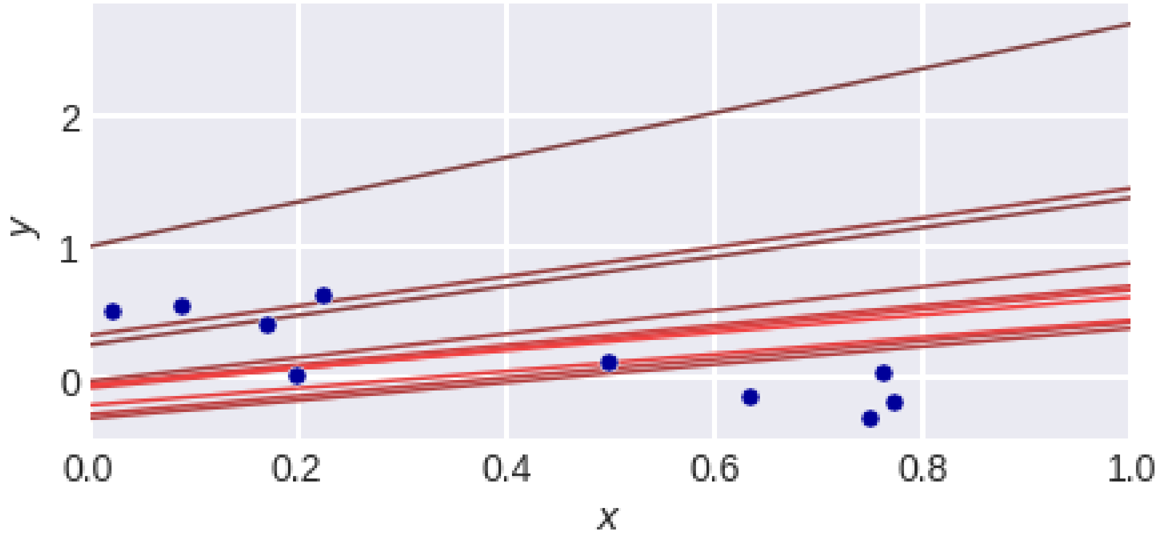
Stochastic Gradient Descent

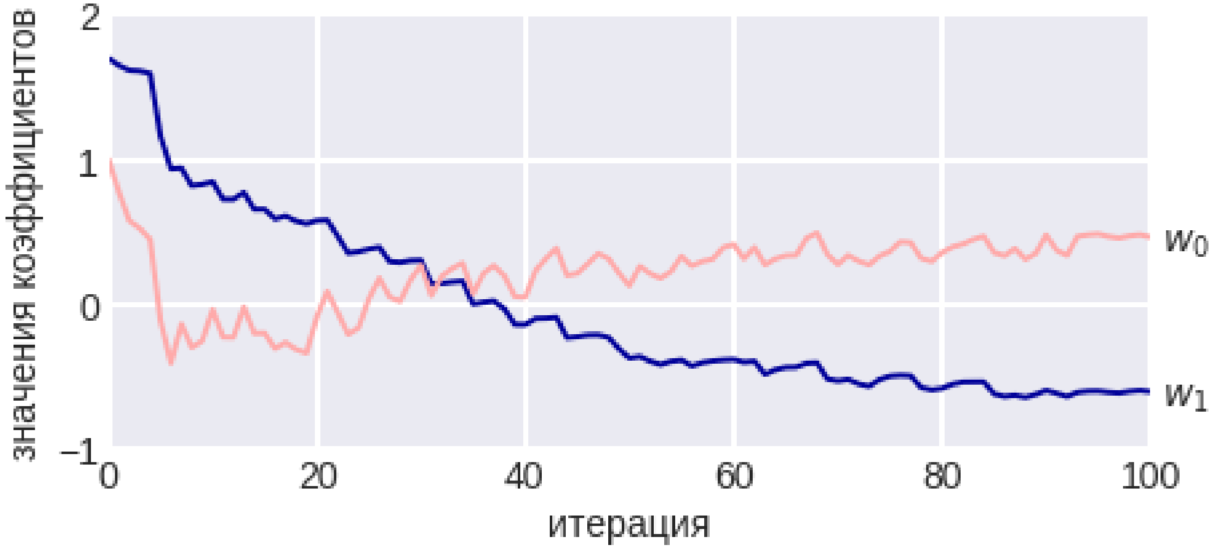
$$w^{(t+1)} = w^{(t)} - \eta_t(a(x_i\,|\,w^{(t)})-y_i)x_i$$

Линейная регрессия: градиентный метод обучения





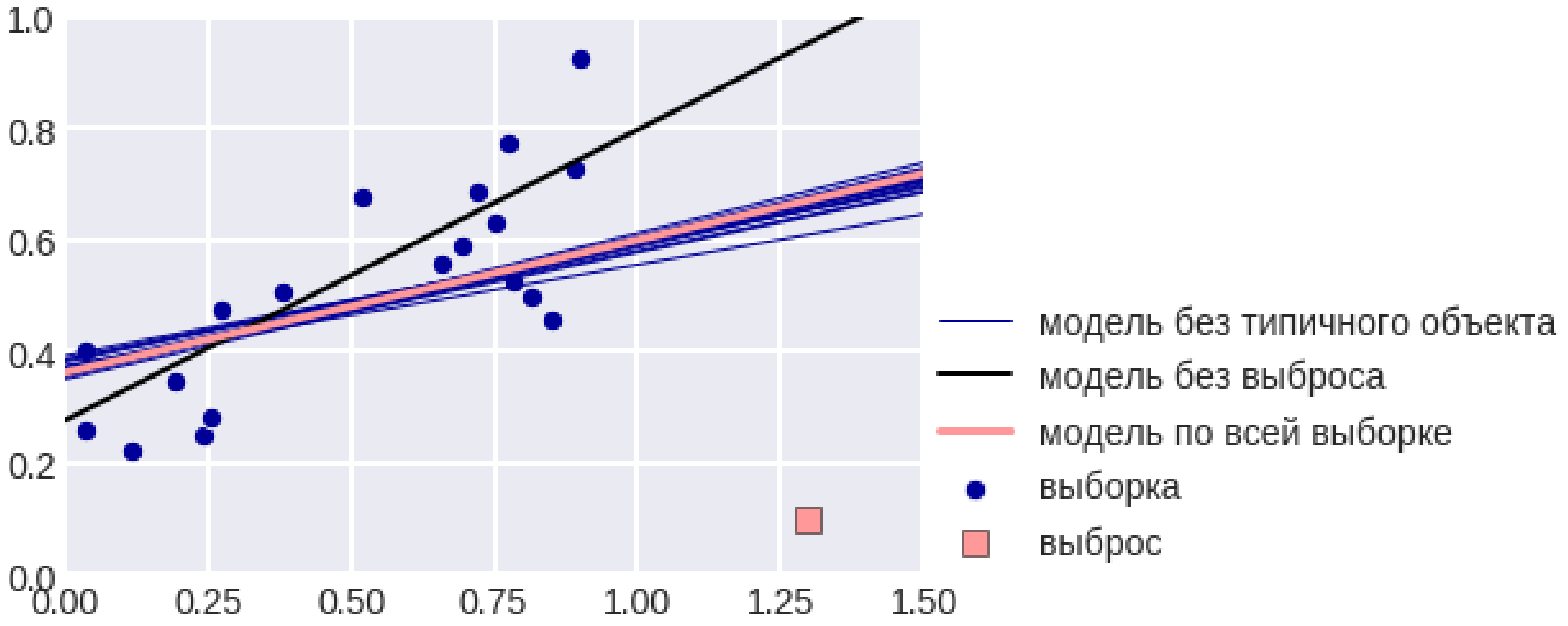




Реализация в scikit-learn

```
from sklearn.linear_model import Ridge
clf = Ridge(alpha=1.0,            # Коэффициент регуляризации, больше – сильнее
            fit_intercept=True,   # свободный член
            solver='auto',        # 'svd', 'cholesky', 'lsqr', 'sparse_cg',
                                  # 'sag', 'saga', 'lbfgs'}
            positive=False)       # условие неотрицательности коэффициентов
                                  # при lbfgs
            # normalize=False       - была раньше
clf.fit(X, y)

sklearn.linear_model.Lasso
sklearn.linear_model.ElasticNet
```

Линейная регрессия – неустойчивость к выбросам



- удаление выбросов
- устойчивая регрессия (ошибки с весами)

Приложения

Задачи с текстами

Бенчмарк для дебита нефти

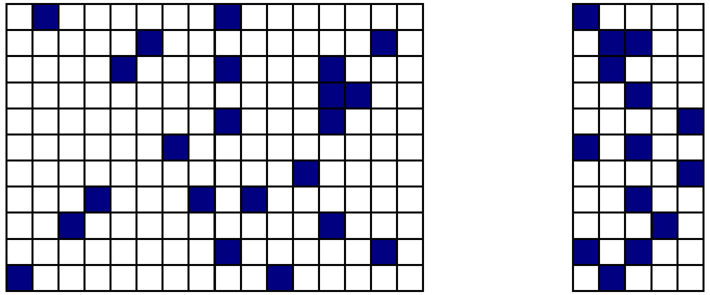
Прогнозирование спроса

Почти любые индустриальные задачи!

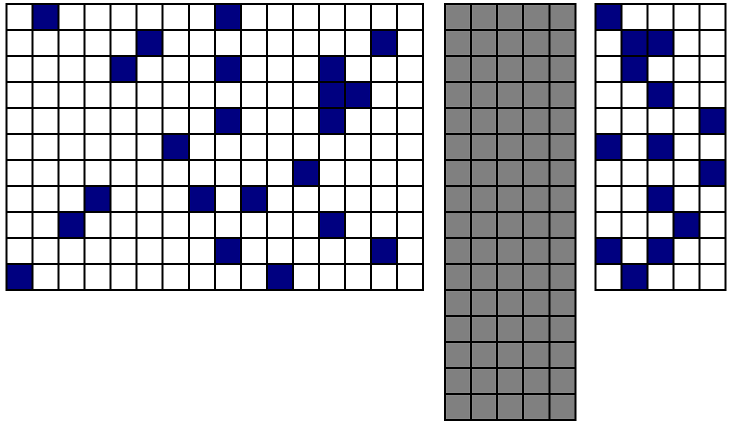
Задачи с текстами

Соревнование «Topical Classification of Biomedical Research Papers»

Данные



Логика решения

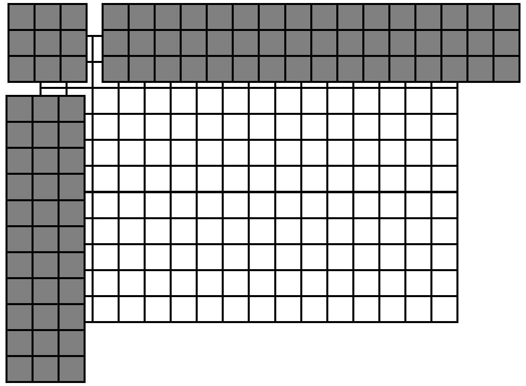


$$X_{q\times n}\cdot W_{n\times l}=Y_{q\times l}$$

$$q=10000,\ n=25000,\ l=83$$

нельзя решать напрямую

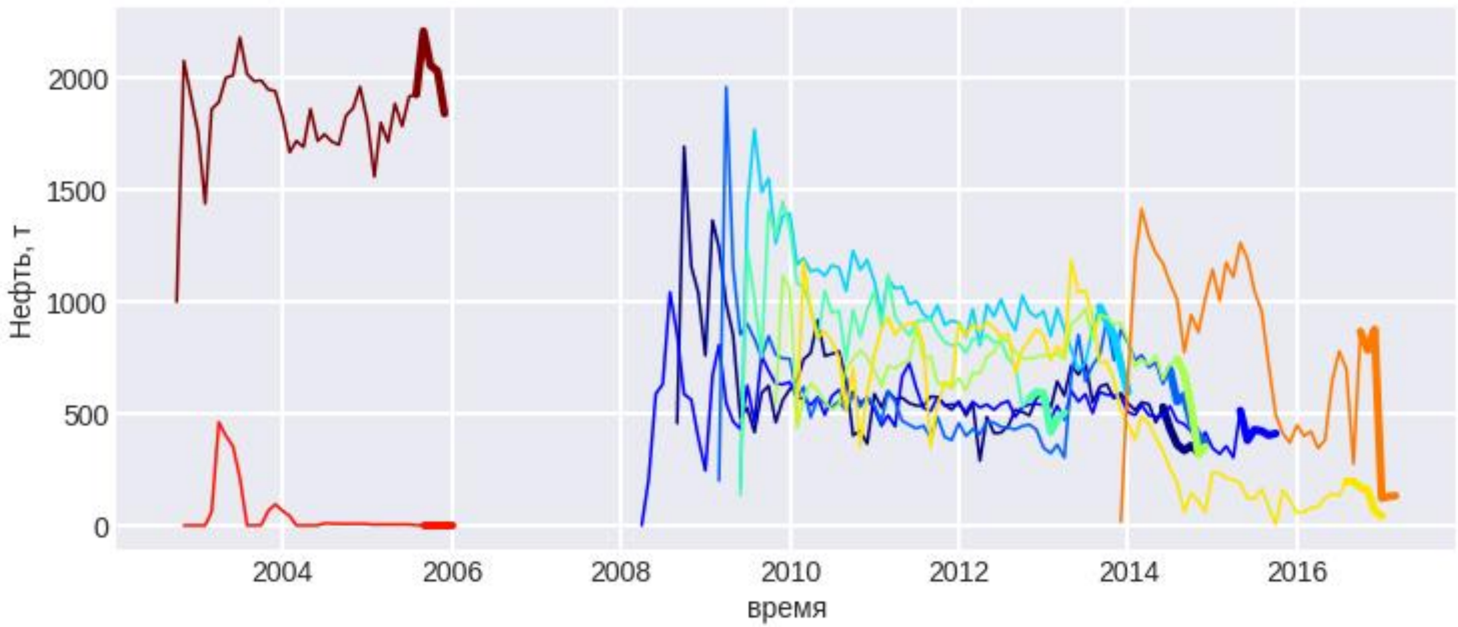
Упрощение: SVD



$$X_{q\times n}\approx U_{q\times k}L_{k\times k}V_{k\times n}$$

$$U_{q\times k}\cdot W_{k\times l}=Y_{q\times l}$$

Бенчмарк прогнозирования дебита нефти



Бенчмарк прогнозирования дебита нефти

$$y_t = \sum_{i=0}^{k} w_{ti} y_{-i}, \quad w_{t0} \geq w_{t1} \geq \ldots$$

соревнование на платформе boosters.pro

https://dyakonov.org/2018/12/23/

Прогнозирование спроса

Спрос товара конкретного id (покупок за следующую неделю)

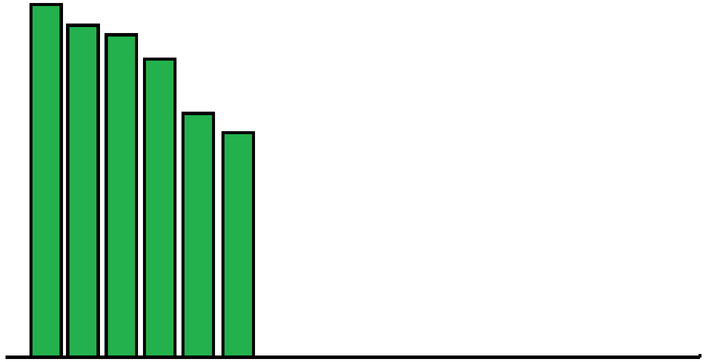
- **# покупок за k дней**
- **# просмотров за k дней**
- **# корзин за k дней**
- **# дней без покупок**
- **изменение цены за последние k дней**
- **есть ли маркетинговая акция**

…

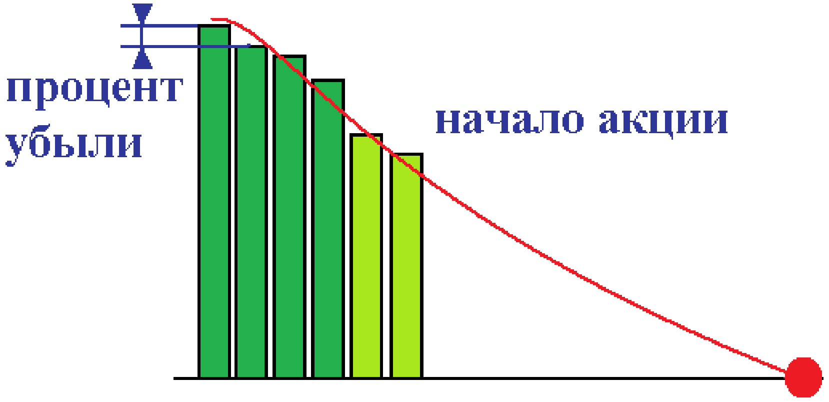
$$Y = \max\left[\sum_t w_t X_t, 0\right]$$

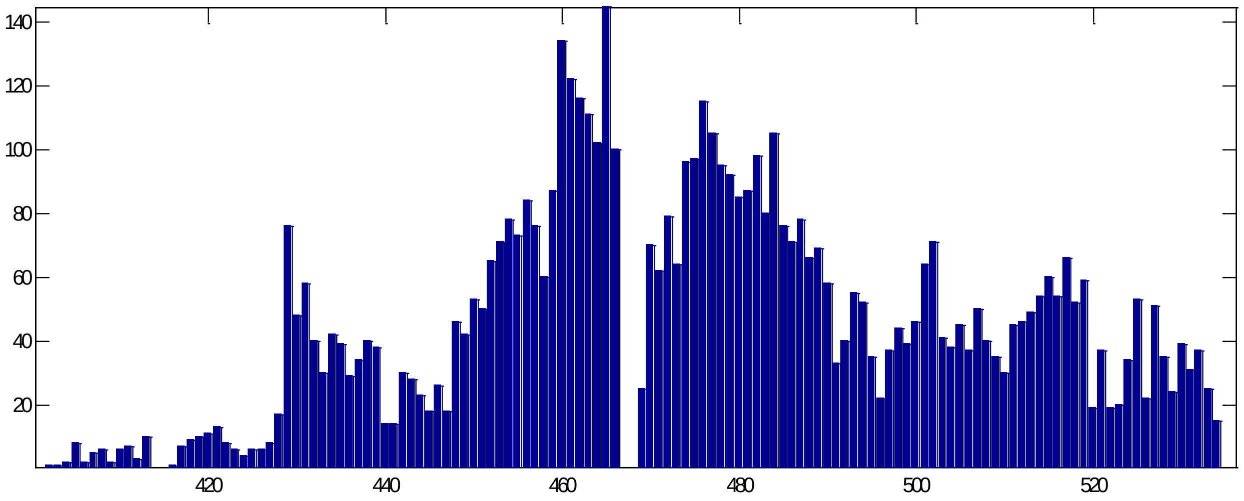
Прогнозирование раскупаемости

Остатки товара на складе:

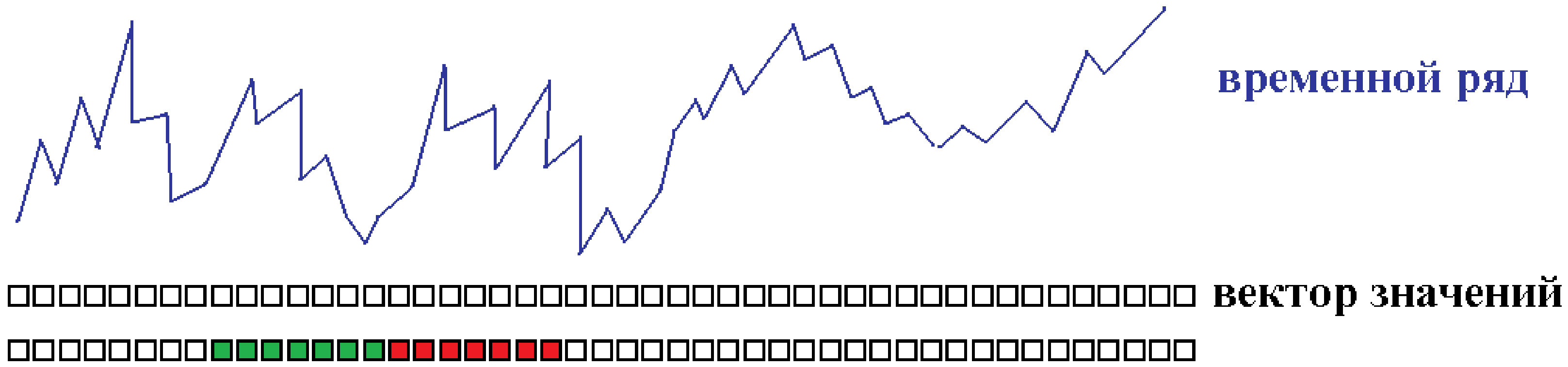


Прогноз точки раскупаемости

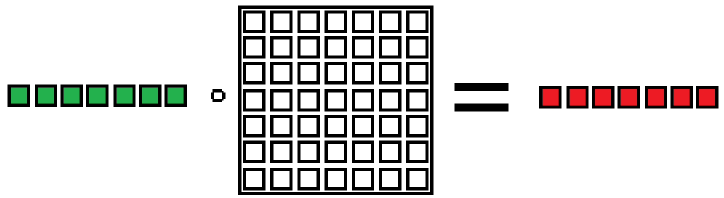




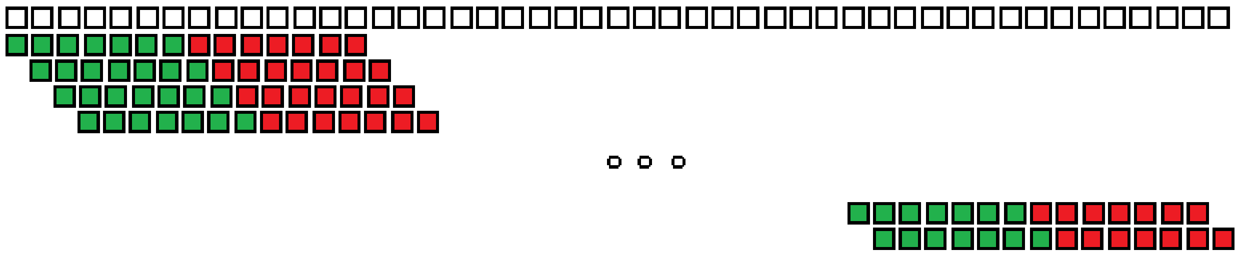
Линейный метод прогнозирования



Пусть существует линейный оператор

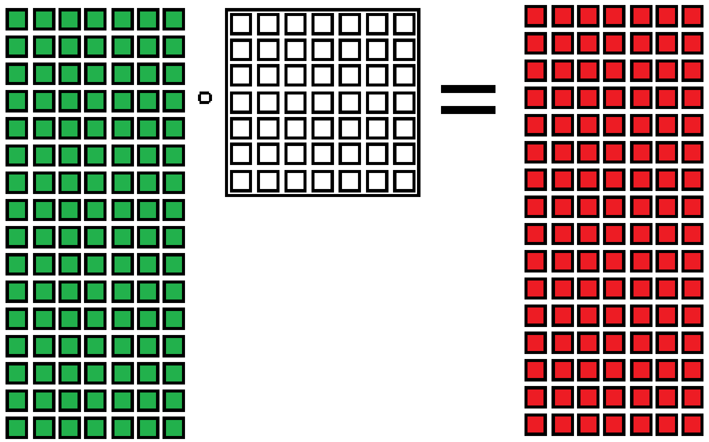


Обучающая выборка



т.е. данных много!

Линейный метод прогнозирования



Это матричное уравнение!

A*X = b

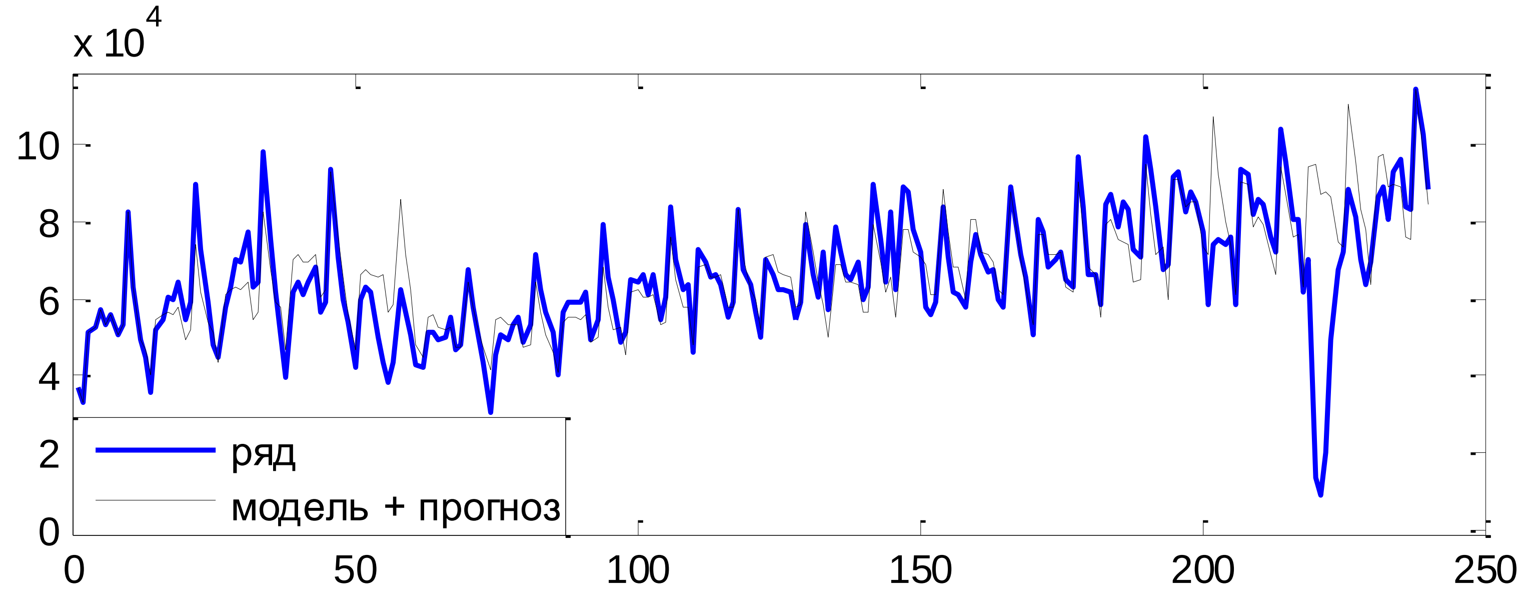
Можно также

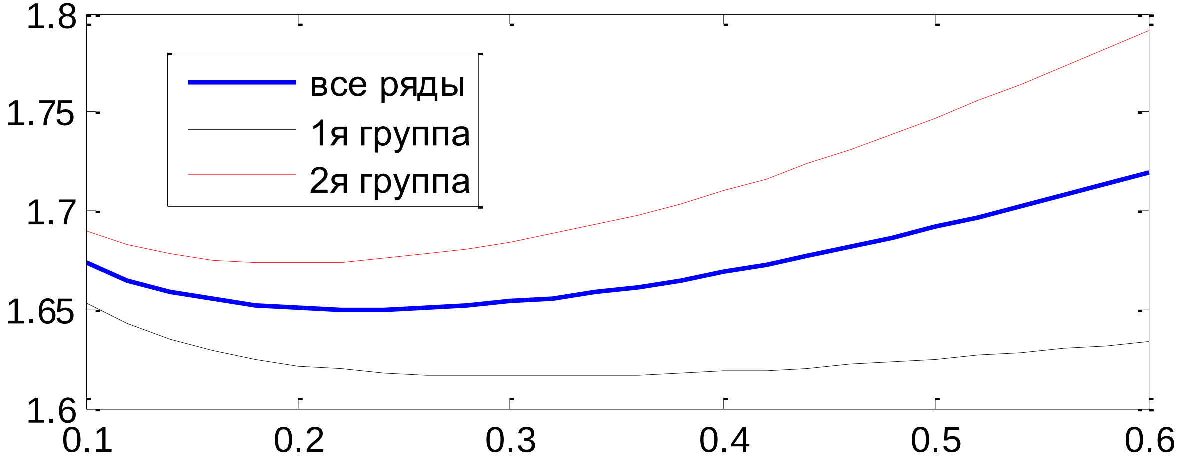
применять нелинейные операторы (как?)

Накладывать ограничения стабильности: $A(A(\tilde{x}_t)) \approx A(\tilde{x}_{t+1})$

Делать регуляризацию (и тут её надо правильно сделать – нормировки)

Линейный метод прогнозирования





точность от коэф. регуляризации

Плюсы и минусы линейных алгоритмов

+ простой, надёжный, быстрый, популярный метод
+ интерпретируемость (⇒ нахождение закономерностей)
+ интерполяция и экстраполяция
+ может быть добавлена нелинейность, с помощью генерации новых признаков
+ хороши для теоретических исследований (в Ridge есть явная формула)
+ коэффициенты асимптотически нормальны
(можно тестировать гипотезы о влиянии признаков)
+ глобальный минимум в оптимизируемом функционале

– линейная гипотеза вряд ли верна
– в теоретическом обосновании ещё предполагается нормальность ошибок
(зависит от функции ошибок)
– «страдает» из-за выбросов
– признаки в одной шкале и однородные (см. успешные примеры)
– проблема коррелированных признаков
⇒ необходимость регуляризации, селекции, PCA, data↑

Итог

Линейная регрессия ~ матричное уравнение

Но проблема вырожденности

Много методов решают эту проблему с разных сторон

Есть важные практические применения

Интересные ссылки

Курс Ramesh Sridharan «Statistics for Research Projects: IAP 2015»

http://www.mit.edu/~6.s085/

Про линейную регрессию

https://dyakonov.org/2019/10/31/линейная-регрессия/