УНИВЕРСИТЕТ
ИННОПОЛИС

Регуляризация. Гребневая регрессия. Лассо

Воробьёва Мария

- maria.vorobyova.ser@gmail.com
- @SparrowMaria

Зачем нам регуляризация?

Переобучение может происходить по следующим причинам:

1. Данные могут быть зашумлены - модель вынуждена искать сложные закономерности на тренировочных данных
2. В данных может быть мультиколинеарность —> задача линейной регрессии будет иметь бесконечно много решений —> на практике это может приводить к тому, что веса признаков по модулю будут сильно возрастать
3. и другие причины, такие как неправильно подобранная модель

Зачем нам регуляризация?

Вектора v и x будут **мультиколинеарны**, если $\langle v, x \rangle = 0$.

Допустим, мы решаем задачу оптимизации MSE для линейной регрессии и нашли оптимальный вектор *w*, тогда классификаторы $w + \alpha v$ будут давать точно такие же ответы, что и *w,* так как $\langle w + \alpha v, x \rangle = \langle w, x \rangle + \alpha \underbrace{\langle v, x \rangle}_{=0} = \langle w, x \rangle$

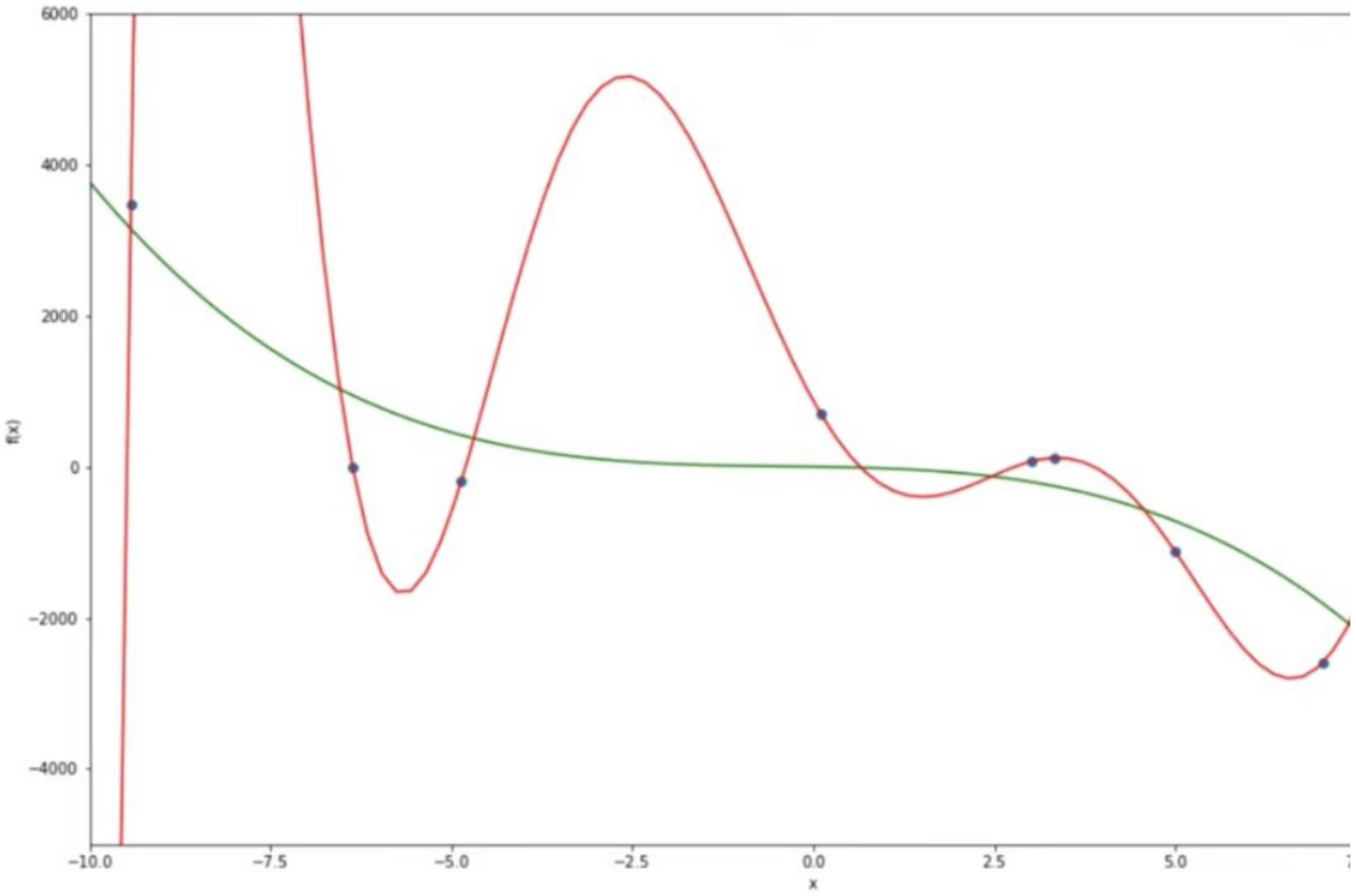
Это означает, что метод может найти решение с огромными коэффициентами при x, это плохо, так как классификатор будет чувствителен к крайне малым изменениям, что приведет к переобучению…

Зачем нам регуляризация?

Существует принцип под названием «**Бритва Оккама**», который гласит:

«Сталкиваясь с двумя одинаково хорошими гипотезами, всегда выбирайте более простую»

Зачем нам регуляризация?

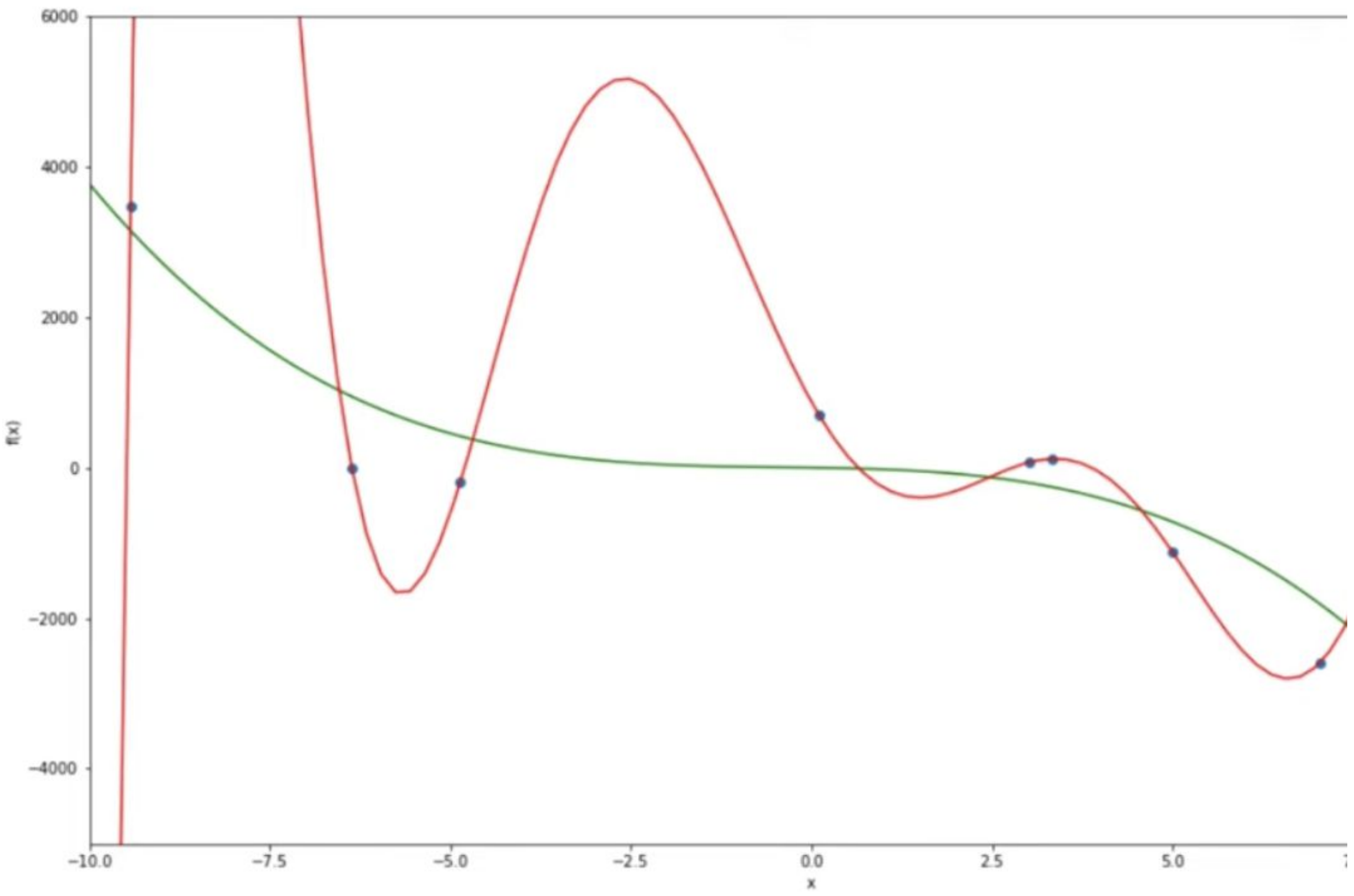


то есть при обучении мы получили коэффициенты:

[0, 2022.32, 4.42, 1.31, 3156.41, 2.4, 1]

Зачем нам регуляризация?

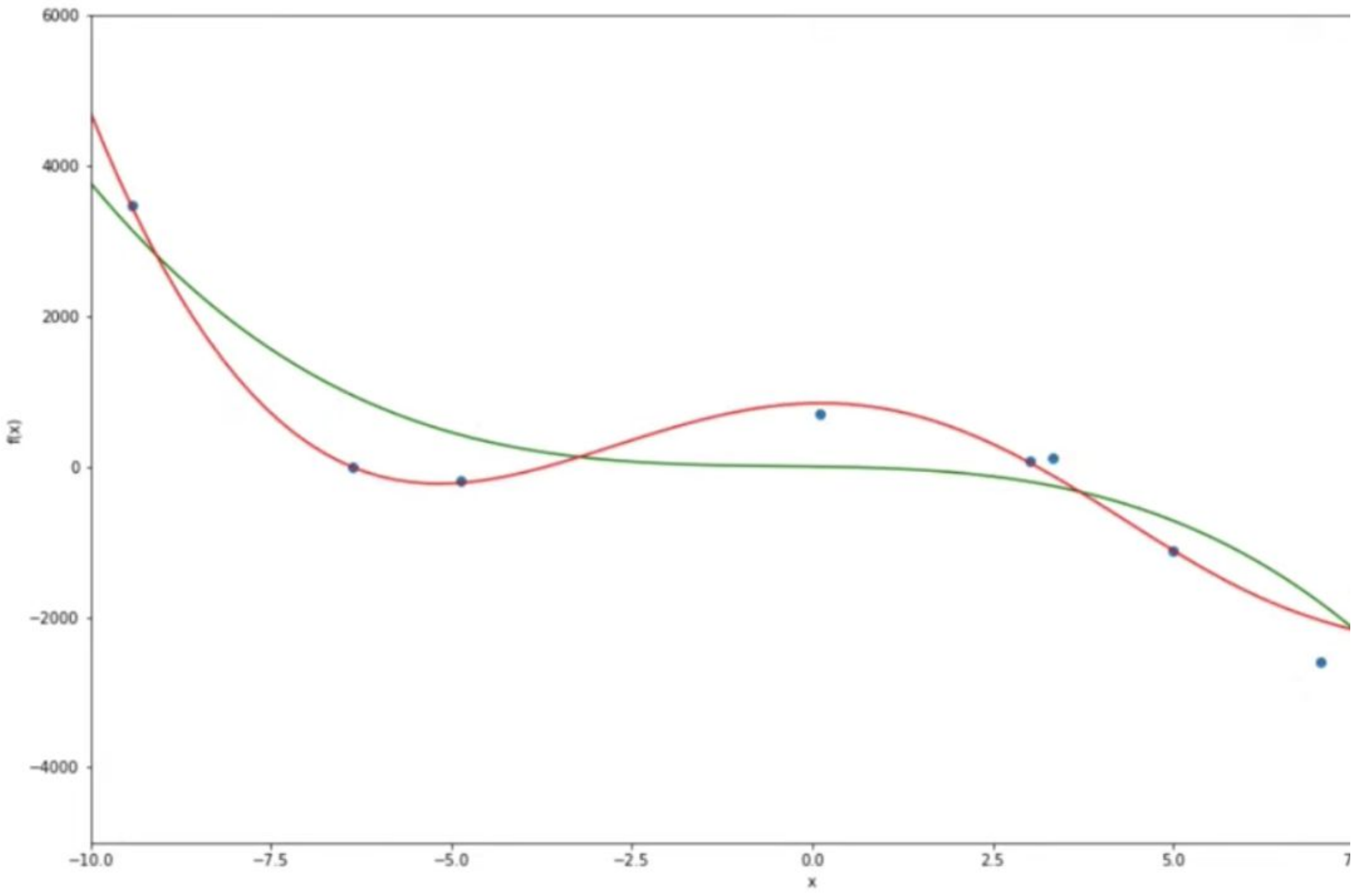
Будем штрафовать коэффициенты, которые слишком большие



то есть при обучении мы получили коэффициенты:

[0, **2022.32**, 4.42, 1.31, **3156.41**, 2.4, 1]

Зачем нам регуляризация?



то есть при обучении мы получили коэффициенты:

[0, 1.5, 4.42, 1.31, 0.5, 2.4, 1]

Линейная регрессия. Повторение

Линейная регрессия: $a(x) = w_0 + w_1 * x_1 + w_2 * x_2 + \ldots w_d * x_d$

В компактном виде $$a(x) = w_0 + \sum_{j=1}^{d} w_j x_j.$$

w_i - веса/коэффициенты

w_0 - свободным коэффициентом/сдвиг

можно еще в **более компактном виде** $a(x) = w_0 + \langle w, x \rangle,$

а если предположить, что существует еще один столбец со всеми 1, то можно записать

еще **компактнее** $a(x) = \langle w, x \rangle.$

http://wiki.cs.hse.ru/Машинное_обучение_1

Как штрафовать?

Регуляризация накладывает штраф на большие веса модели

$$Q_\lambda(w) = Q(w) + \boldsymbol{\lambda} R(w).$$

Коэффициент λ - параметр регуляризации, он контролирует баланс между подгонкой под данные и штрафом за сложность

!!будем штрафовать коэффициенты только при неизвестных параметрах, свободный коэффициент не штрафуем

Регуляризация L2

Регуляризация Тихонова (или Ridge regularization, или гребневая регрессия) $R(w) = \|w\|_2 = \sum_{i=1}^{d} w_i^2$

$$Q(w, X) + \lambda||w||^2 \to \min_w.$$

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

Веса чаще всего не становятся нулевыми - они будут стремиться к 0, но не станут 0

Регуляризация L1

Регуляризация L1 или Lasso регуляризация

$$R(w) = \|w\|_1 = \sum_{i=1}^{d} |w_i|$$

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

$$||w||_1 = |w_1| + |w_2| + \ldots + |w_n|$$

Веса могут стать нулевыми , что полезно для задачи отбора переменных
или понижения размерности

Почему при L1 коэффициенты при регрессорах иногда зануляются?

Функция потерь (loss function) $\text{Loss} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 + \lambda\sum_{j=1}^{m}|w_j|$

Градиент 1-ого слагаемого $\frac{\partial \text{MSE}}{\partial w_j} = -\frac{2}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)x_{ij}$

Градиента 2-го слагаемого $\frac{\partial \lambda |w_j|}{\partial w_j} = \lambda \cdot \text{sign}(w_j)$

где $\text{sign}(w_j)$ — это функция знака, которая равна 1, если $w_j > 0$, и -1, если $w_j < 0$.

Обновление весов происходит следующим образом:

$$w_j = w_j - \alpha\left(-\frac{2}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)x_{ij} + \lambda \cdot \text{sign}(w_j)\right)$$

Предположим, что вес w_j очень мал (например, $w_j \approx 0.01$). Тогда градиент MSE может быть малым, но градиент L1 регуляризации ($\lambda \cdot \text{sign}(w_j)$) будет сравнительно большим и будет пытаться уменьшить w_j ещё больше.

Почему при L2 коэффициенты при регрессорах не зануляются?

Функция потерь (loss function) $\text{Loss} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 + \lambda\sum_{j=1}^{m} w_j^2$

Градиент 1-ого слагаемого $\frac{\partial \text{MSE}}{\partial w_j} = -\frac{2}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)x_{ij}$

Градиента 2-го слагаемого $\frac{\partial \lambda w_j^2}{\partial w_j} = 2\lambda w_j$

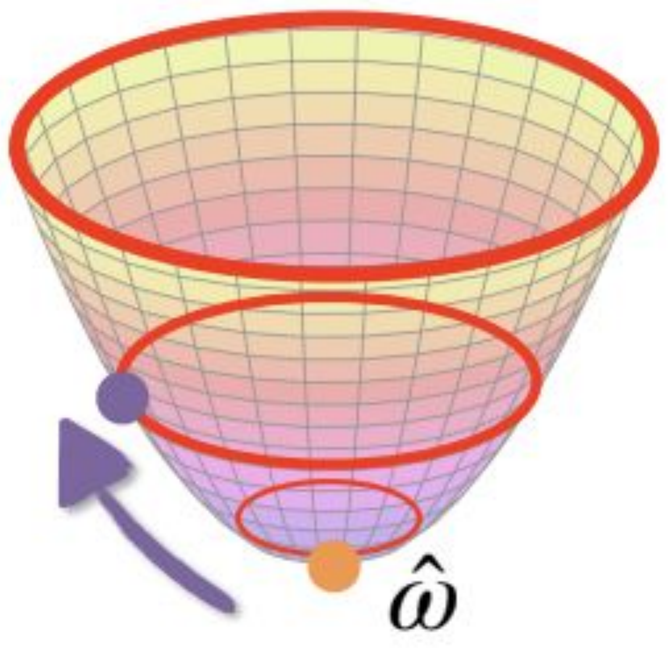
Обновление весов происходит следующим образом:

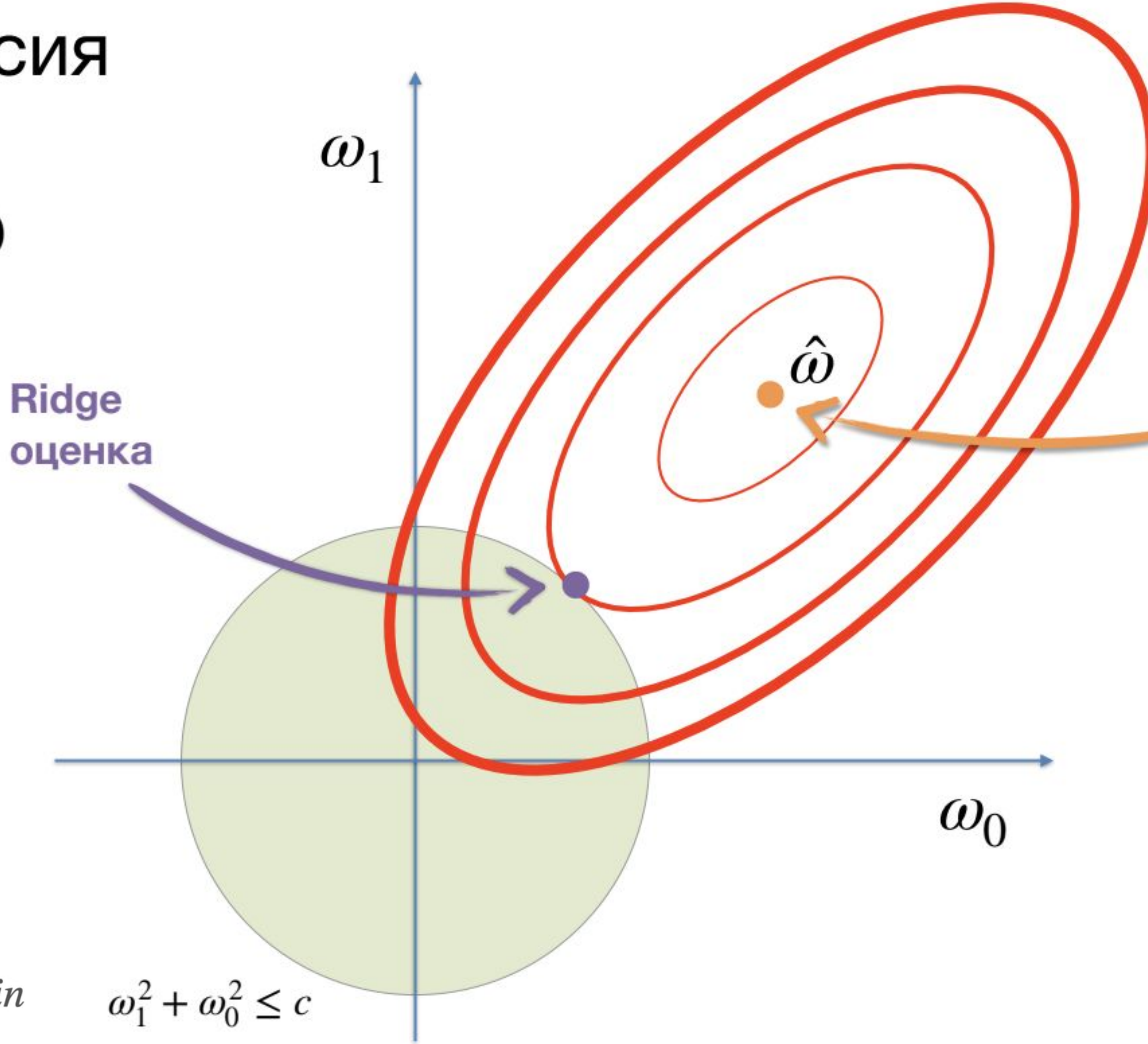
$$w_j = w_j - \alpha\left(-\frac{2}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)x_{ij} + 2\lambda w_j\right)$$

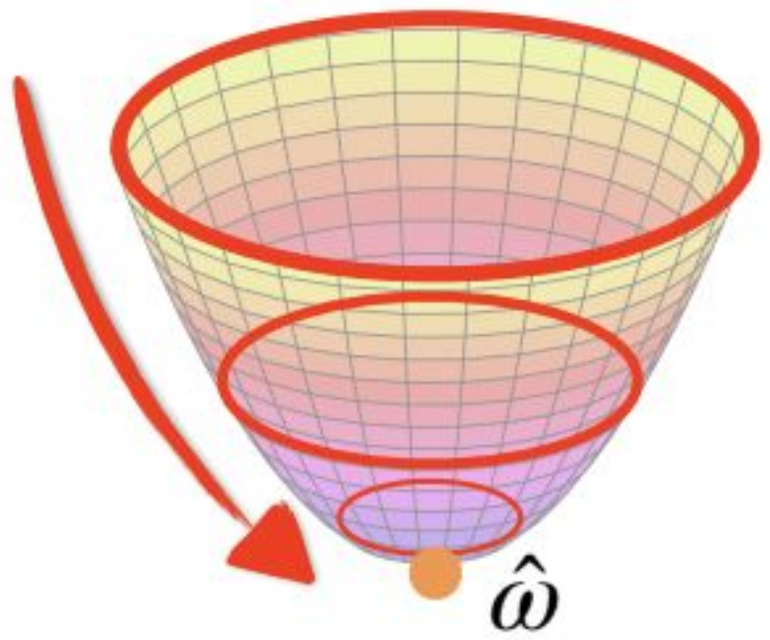
L2 регуляризация стремится уменьшить значения весов, но не "притягивает" их к нулю так же агрессивно, как это делает L1 регуляризация. Поскольку штраф в L2 регуляризации пропорционален квадрату веса, он становится меньше по мере уменьшения веса. Таким образом, когда вес становится малым, влияние регуляризационного члена также уменьшается, и вес стремится к малым значениям, но не становится нулевым.

Ridge регрессия

$$y = \omega_1 x + \omega_0$$





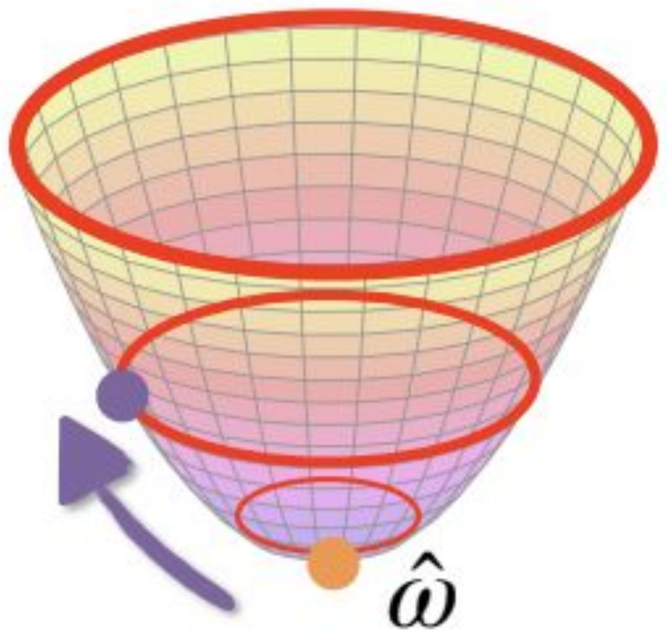


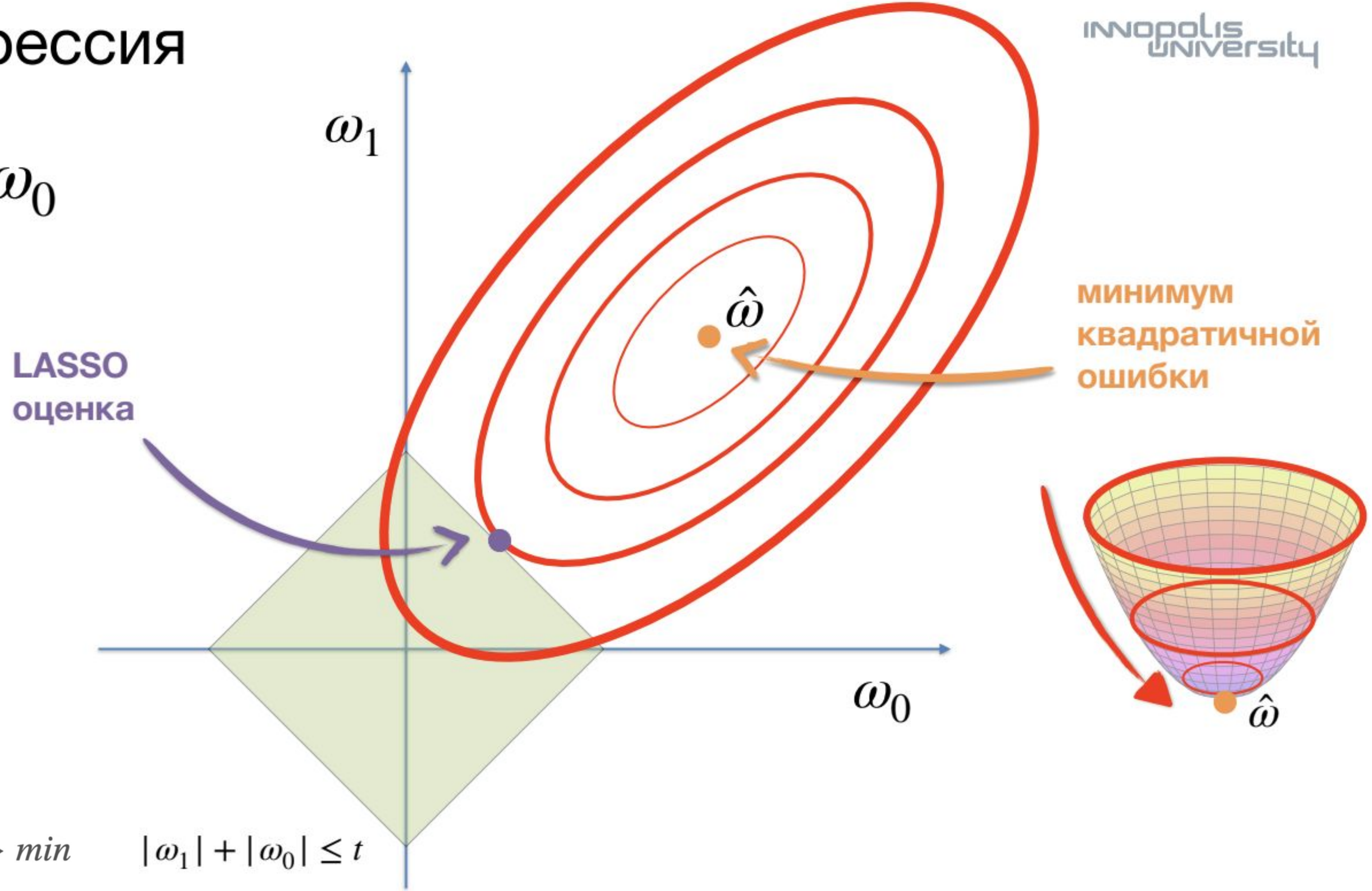
$Q(w, X) = (Xw - y)^2 \rightarrow min$

$\omega_1^2 + \omega_0^2 \leq c$

LASSO регрессия

$$y = \omega_1 x + \omega_0$$





$Q(w, X) = (Xw - y)^2 \to min$ $\quad |\omega_1| + |\omega_0| \le t$

Регуляризация

$$L(f(x,\omega),y) = \sum_{i=1}^{M}\left(y_i - \sum_{j=0}^{p}\omega_j x_{ij}\right)^2 + \lambda\sum_{j=0}^{p}|\omega_j|$$

LASSO регрессия

$$L(f(x,\omega),y) = \sum_{i=1}^{M}\left(y_i - \sum_{j=0}^{p}\omega_j x_{ij}\right)^2 + \lambda\sum_{j=0}^{p}\omega_j^2$$

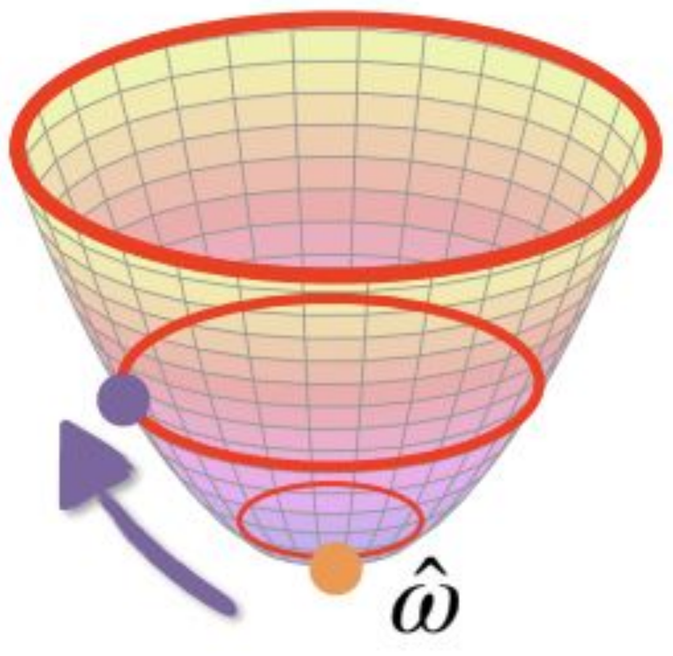
RIDGE регрессия

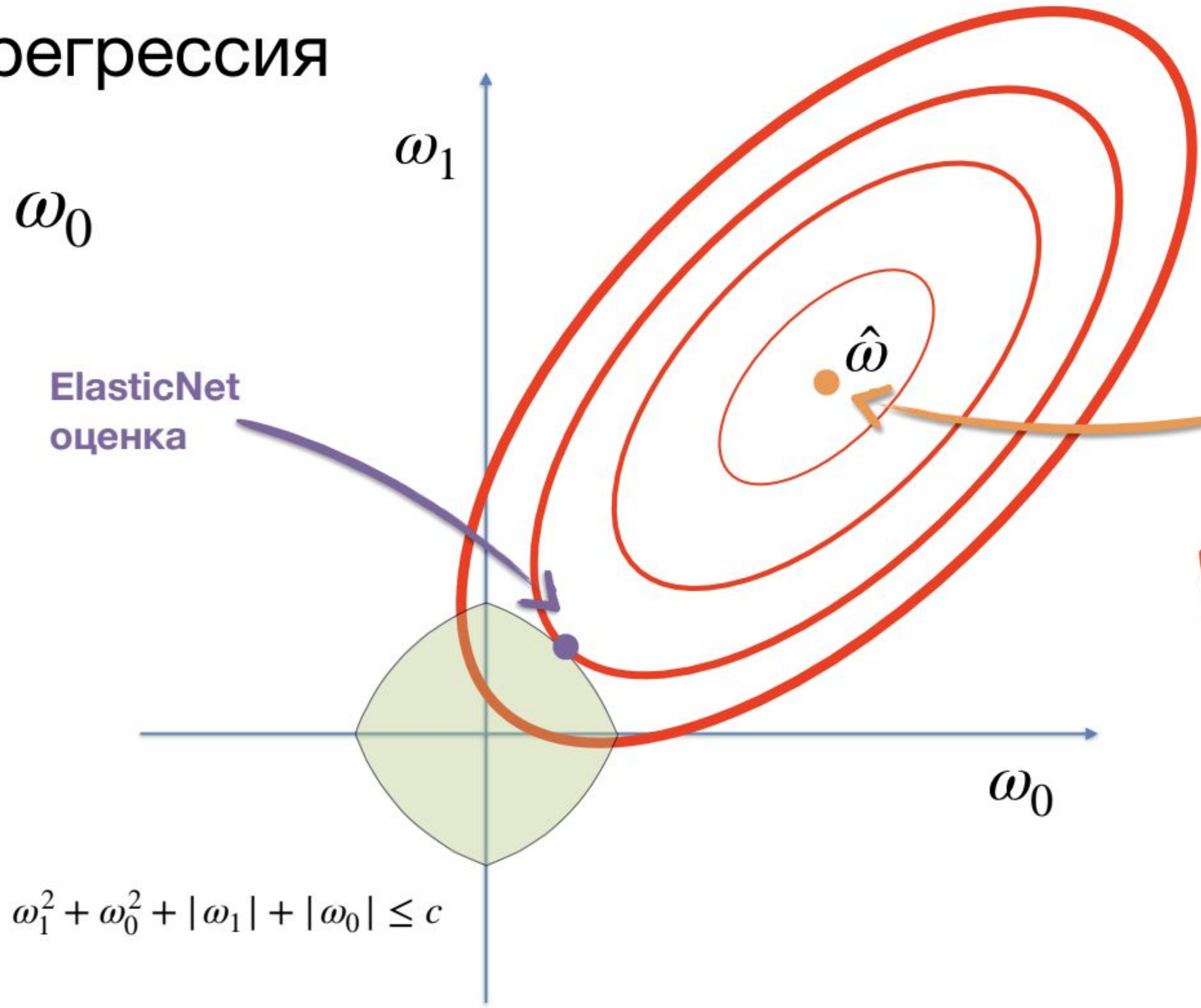
$$L(f(x,\omega),y) = \sum_{i=1}^{M}\left(y_i - \sum_{j=0}^{p}\omega_j x_{ij}\right)^2 + \lambda_1\sum_{j=0}^{p}\omega_j^2 + \lambda_2\sum_{j=0}^{p}|\omega_j|$$

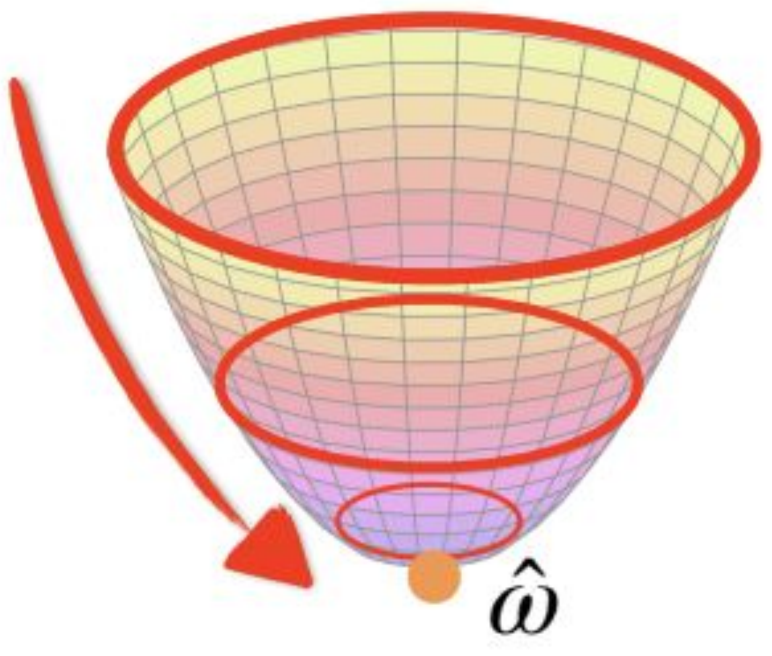
ElasticNet регрессия

ElasticNet регрессия

$$y = \omega_1 x + \omega_0$$







Регуляризация

sklearn.linear_model.Lasso

class `sklearn.linear_model.Lasso`(*alpha=1.0, *, fit_intercept=True, precompute=False, copy_X=True, max_iter=1000, tol=0.0001, warm_start=False, positive=False, random_state=None, selection='cyclic'*) [source]

sklearn.linear_model.Ridge

class `sklearn.linear_model.Ridge`(*alpha=1.0, *, fit_intercept=True, copy_X=True, max_iter=None, tol=0.0001, solver='auto', positive=False, random_state=None*) [source]

sklearn.linear_model.ElasticNet

class `sklearn.linear_model.ElasticNet`(*alpha=1.0, *, l1_ratio=0.5, fit_intercept=True, precompute=False, max_iter=1000, copy_X=True, tol=0.0001, warm_start=False, positive=False, random_state=None, selection='cyclic'*) [source]

Регуляризация

sklearn.linear_model.LogisticRegression

class `sklearn.linear_model.LogisticRegression`(*penalty='l2'*, ***, *dual=False*, *tol=0.0001*, *C=1.0*, *fit_intercept=True*, *intercept_scaling=1*, *class_weight=None*, *random_state=None*, *solver='lbfgs'*, *max_iter=100*, *multi_class='auto'*, *verbose=0*, *warm_start=False*, *n_jobs=None*, *l1_ratio=None*) [source]

Warning: The choice of the algorithm depends on the penalty chosen: Supported penalties by solver:

- 'newton-cg' - ['l2', 'none']
- 'lbfgs' - ['l2', 'none']
- 'liblinear' - ['l1', 'l2']
- 'sag' - ['l2', 'none']
- 'saga' - ['elasticnet', 'l1', 'l2', 'none']

УНИВЕРСИТЕТ
ИННОПОЛИС

ВОПРОСЫ И ОТВЕТЫ