Передовые инженерные школы

МИНОБРНАУКИ РОССИИ

УНИВЕРСИТЕТ ИННОПОЛИС

онлайн университет

Программная инженерия. Разработка ПО (Python для продвинутых специалистов. Машинное обучение)

Модуль: Предобработка данных и машинное обучение

Лекция 5: Отбор признаков и уменьшение размерности

Дата: 26.05.2025

Содержание лекции

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА
УНИВЕРСИТЕТА ИННОПОЛИС

- Что такое признаки и зачем их отбирать?
- Проблемы высокой размерности
- Методы отбора признаков
  - Фильтрационные методы (Filter)
  - Обёрточные методы (Wrapper)
  - Встраиваемые методы (Embedded)

Что такое признаки (features)

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

Признаки (features) — это **отдельные измерения или характеристики** объекта, которые используются в модели машинного обучения для предсказания целевой переменной (target).

Например:

- В задаче предсказания стоимости квартиры:
  - Признаки: площадь, этаж, район, год постройки.
  - Целевая переменная: цена квартиры.
- В задаче диагностики болезни:
  - Признаки: возраст, давление, уровень глюкозы.
  - Целевая переменная: наличие заболевания (да/нет)

Массив признаков называют **признаковым пространством**, и именно в этом пространстве алгоритм "ищет закономерности".

Зачем отбирать признаки?

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

Большое количество признаков не всегда полезно. Наоборот, **избыточное или нерелевантное** множество признаков может:

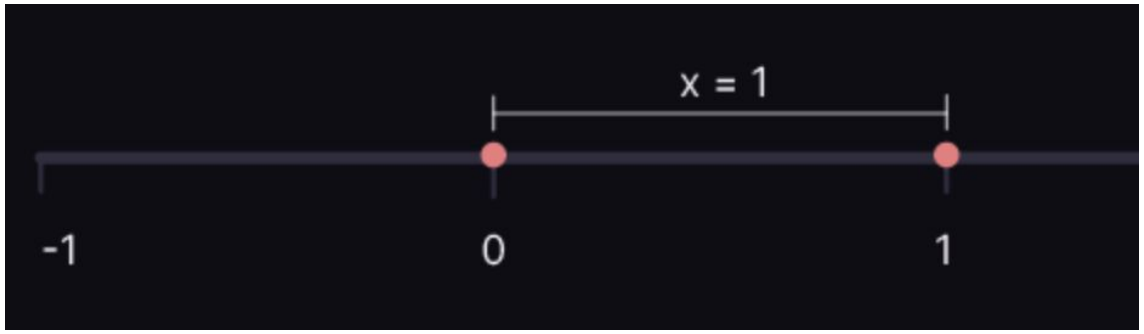
1. **Понижать качество модели**
   - Ненужные признаки добавляют шум, затрудняют обучение.
   - Возрастает риск **переобучения (overfitting)**.
2. **Увеличивать вычислительные затраты**
   - Большие датасеты с сотнями/тысячами признаков требуют больше памяти и времени.
3. **Ухудшать интерпретируемость**
   - Модель, основанная на 5 информативных признаках, проще понять и объяснить, чем модель с 200 признаками.

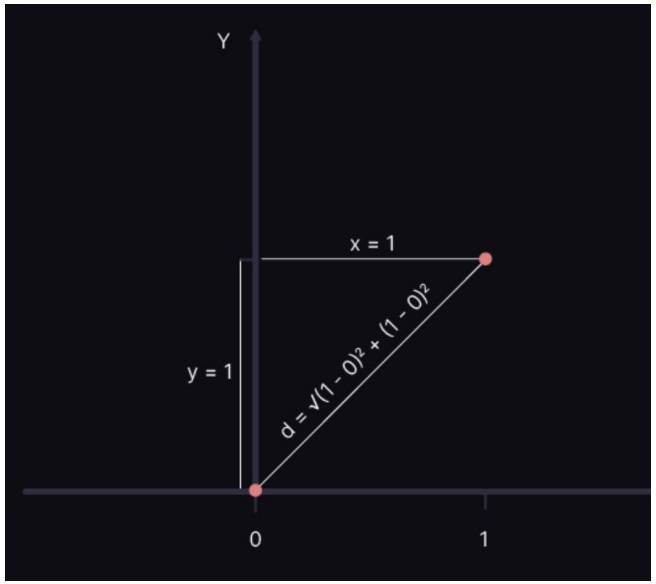
Проклятье размерности

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

Термин «**проклятие размерности**» в 1961 году ввел американский математик Ричард Беллман.

Предположим, у нас есть две точки на прямой, 0 и 1. Эти две точки находятся на расстоянии друг от друга =1 Теперь мы вводим вторую ось Y – второе измерение. Положение точек определяется теперь списком из двух чисел – (0,0) и (1,1). Расстояние между точками теперь подсчитывается с помощью Евклидова расстояния и оно равно 1.44. В трехмерном пространстве будет 1.73



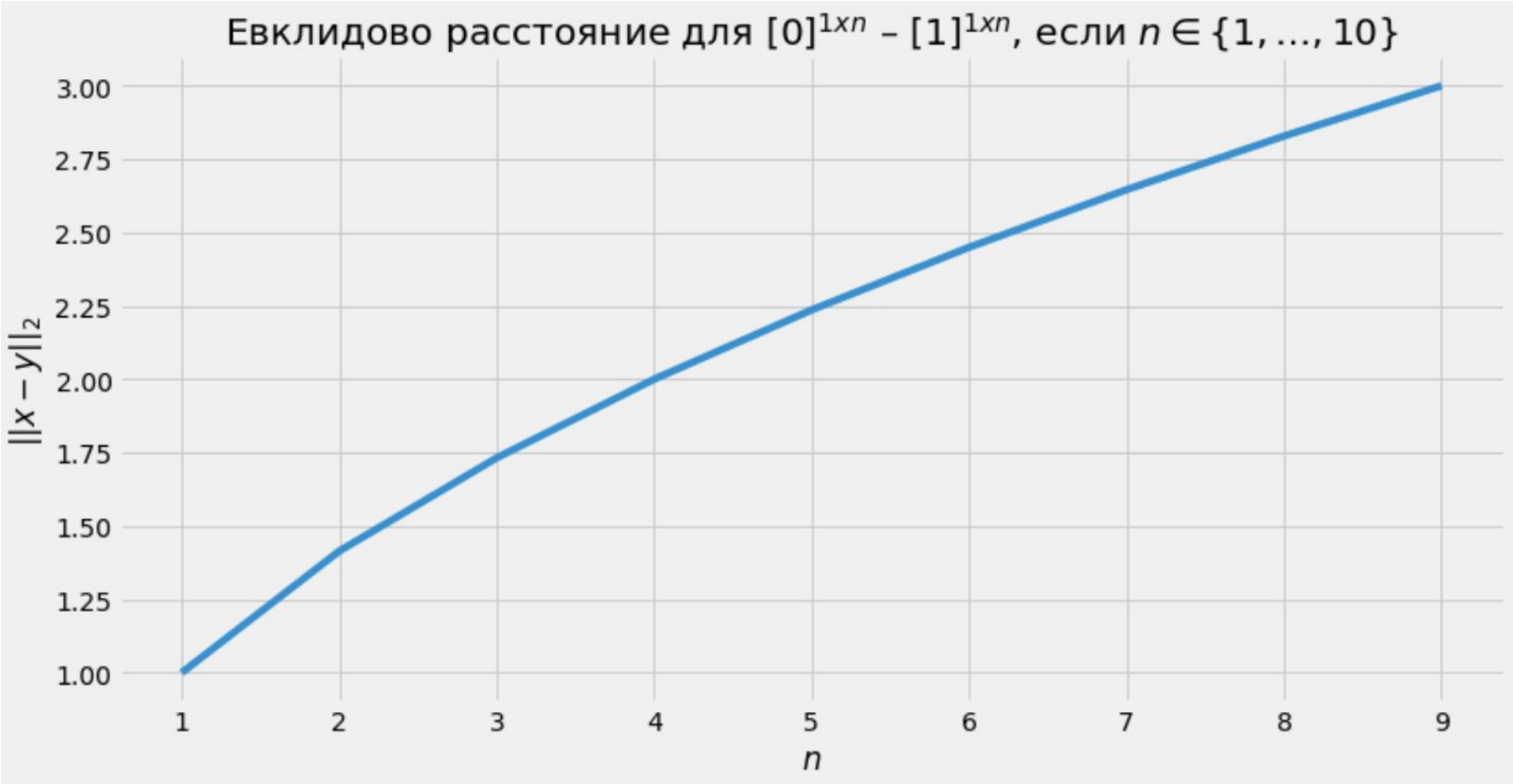


можно почитать - https://www.helenkapatsa.ru/prokliatiie-razmiernostiei

Проклятье размерности

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

Термин «**проклятие размерности**» в 1961 году ввел американский математик Ричард Беллман.



Проклятье размерности. Простой пример

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

Представьте, что вы находитесь в небольшой комнате и вы пытаетесь найти своего друга. В комнате всего один стол и стул, поэтому вам легко будет найти друга, потому что у вас мало места для поиска.

Совершенно другая ситуация, когда вы находитесь на огромном футбольном стадионе, и ваш друг тоже где-то там. Стадион в тысячу раз больше, чем комната, и вам гораздо сложнее найти друга, потому что у вас гораздо больше пространства для поиска.

Это и есть суть "проклятия размерности" — когда у вас много признаков (или измерений), поиск нужной информации становится очень трудным.

Цель отбора признаков

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

1. **Выделить наиболее информативные признаки**, которые максимально помогают предсказывать целевую переменную.
2. **Упростить модель**, сделав её быстрее, надёжнее и понятнее.
3. **Снизить размерность данных** перед применением других алгоритмов (кластеризации, визуализации и т.п.)

Пример:

Предположим, у нас есть 100 признаков, но только 10 из них действительно влияют на результат.

Если мы сможем отбросить "мусорные" признаки — наша модель будет:

- Обучаться быстрее
- Предсказывать точнее
- Меньше переобучаться
- Быть проще в интерпретации

Фильтрационные методы

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

- Фильтрационные методы оценивают **каждый признак по отдельности**, измеряя его статистическую связь с целевой переменной.
- Признаки сортируются по **оценке полезности**, и выбираются лучшие из них (например, k лучших или те, что выше порога).

Преимущества:

- Быстро работают, особенно на больших данных.
- Не зависят от конкретной модели.
- Часто полезны как **предварительный этап отбора**.

Недостатки:

- Не учитывают **взаимодействие признаков** между собой.
- Работают с каждым признаком **независимо**, что может игнорировать важные комбинации.

Фильтрационные методы.
Методы, не зависящие от таргета

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

Оценка дисперсии (Variance Threshold)

- признак с низкой дисперсией почти не меняется → бесполезен
- Например: если столбец почти везде = 0 (или любому другому значению) → он ничего не различает, следовательно бесполезен

Уникальность / частота значений

- Удаление признаков с одним значением (например, "город = Москва" во всех строках)
- Удаление признаков с **высокой кардинальностью** (много уникальных значений), для каждого объекта в обучающей выборке свое значение, например имя клиента

Корреляция между признаками

- **дублирующие или сильно скоррелированные признаки**
- Например: если `feature1` и `feature2` имеют корреляцию > 0.95 — можно оставить один из них
  a. Pearson (для числовых признаков)
  b. Cramér's V (для категориальных)

Много пропусков

- Удаление признаков с **высокой долей пропусков**

Фильтрационные методы.
Если целевая переменная непрерывная

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

Корреляция Пирсона (Pearson Correlation)

Что измеряет:

Измеряет **силу линейной зависимости** между признаком XXX и целевой переменной yyy.

Значения коэффициента:

- от -1 до 1:
  - **+1** — сильная прямая связь
  - **-1** — сильная обратная связь
  - **0** — отсутствие линейной связи

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \cdot \sum(y_i - \bar{y})^2}}$$

Когда применять:

- Признак и target — **числовые**
- Интересует **линейная связь**

Ограничения:

- Не улавливает **нелинейные зависимости**
- Чувствителен к **выбросам**

Фильтрационные методы.
Если целевая переменная непрерывная

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

Mutual Information

- X и Y **независимы**, то знание одного **не даёт информации** о другом → **MI = 0**
- X и Y **сильно зависимы**, то знание одного **сильно уменьшает неопределённость** другого → **MI > 0**
- Когда есть две зависимые величины, можно говорить о том, сколько информации об одной переменной содержится в другой.
- Рассмотрим, для примера, (вес_человека, рост_человека). Информация о росте человека заставит нас пересмотреть распределение веса, например, сообщение "рост = 2 метра 10 см" сместит распределение веса в область больших значений
- Выявляет как линейные, так и **нелинейные** зависимости между признаком и target.

Фильтрационные методы.
Если целевая переменная категориальная

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА
УНИВЕРСИТЕТА ИННОПОЛИС

Категориальный target + числовой признак

- **ANOVA F-test**
- Сравнивает средние значений признака в разных классах

Что делает ANOVA F-тест в этой ситуации?

- Он проверяет гипотезу, что средние значения числового признака одинаковы во всех группах (классах) целевой переменной.
- Если средние по группам существенно отличаются, значит признак полезен для разделения классов.
- F-статистика измеряет отношение межгрупповой дисперсии к внутригрупповой дисперсии.

Фильтрационные методы.
Если целевая переменная категориальная

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА
УНИВЕРСИТЕТА ИННОПОЛИС

Категориальный target + категориальный признак

Chi-squared test (хи-квадрат)

Строится таблица сопряжённости, сравниваются ожидаемые и реальные частоты

Строится таблица сопряжённости (contingency table) — матрица, где строки — категории одного признака, столбцы — категории другого, а в ячейках — наблюдаемые частоты (число объектов, попавших в соответствующую пару категорий).

Вычисляются ожидаемые частоты при предположении независимости переменных.

$$\chi^2 = \sum \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

где O_{ij} — наблюдаемое значение, E_{ij} — ожидаемое.

Фильтрационные методы.
Weights of Evidence (WOE)

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

f(x|y=1) - относительные частоты “1”

f(x|y=0 - относительные частоты “0”

$$WoE = ln(\frac{f(x|y=0)}{f(x|y=1)})$$

- **Если значение WOE близко к 0**, значит, эта категория **встречается примерно одинаково часто** в классах **"1" (плохой)** и **"0" (хороший)**.

- **Если WOE < 0**, значит, эта категория **чаще встречается в классе "1" (плохой)**.

Возраст	количество всех объектов	количество "1"	f(x|y=1) - относительные частоты "1"	количество "0"	f(x|y=0) - относительные частоты "0"	event rate (bad rate)	WoE
<21	1000	300	6,6%	700	2,2%	30%	-1,110
(21;26]	4000	900	19,7%	3100	9,6%	23%	-0,720
(26;30]	6000	1080	23,6%	4920	15,2%	18%	-0,441
(30;35]	9000	900	19,7%	8100	25,0%	10%	0,240
(35;41]	10000	900	19,7%	9100	28,1%	9%	0,357
>41	7000	500	10,9%	6500	20,0%	7%	0,608

Фильтрационные методы.
Weights of Evidence (WOE)

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА
УНИВЕРСИТЕТА ИННОПОЛИС

$$WoE = ln(\frac{f(x|y=0)}{f(x|y=1)})$$

$$IV = \sum_x (f(x|1) - f(x|0))\, WoE(x)$$

IV	Variable Predictivenes
<0.01	Very weak
<0.1	Weak
<0.5	Medium
>0.5	High

Возраст	количество всех объектов	количество "1"	f(x|y=1) - относительные частоты "1"	количество "0"	f(x|y=0) - относительные частоты "0"	event rate (bad rate)	WoE
<21	1000	300	6,6%	700	2,2%	30%	-1,110
(21;26]	4000	900	19,7%	3100	9,6%	23%	-0,720
(26;30]	6000	1080	23,6%	4920	15,2%	18%	-0,441
(30;35]	9000	900	19,7%	8100	25,0%	10%	0,240
(35;41]	10000	900	19,7%	9100	28,1%	9%	0,357
>41	7000	500	10,9%	6500	20,0%	7%	0,608

Линейная регрессия. Повторение

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

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

Обёрточные методы (Wrapper methods)

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

Если все признаки были приведены к единой шкале, то есть нет такого, что в данных есть признаки очень маленькие и очень большие, то по коэффициентам можно оценивать важность признаков

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

Каждый признак x_i влияет на результат пропорционально своему **весу** w_i.

Обёрточные методы (Wrapper methods)

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

Оценивают множество подмножеств признаков, используя модель как «чёрный ящик».

- RFE (Recursive Feature Elimination) — начинается с обучения полной модели на всех признаках, далее рекурсивное удаление признаков с меньшей важностью, то есть с маленькими по модулю значениями w_i

- Sequential Feature Selection (SFS) — пошаговое добавление или удаление признаков (вперёд/назад)
  - **Forward selection** (вперёд): начинаем с пустого множества и **добавляем признаки по одному**, которые больше всего улучшают общую метрику модели
  - **Backward selection** (назад): начинаем со всех признаков и **постепенно удаляем** наименее полезные.

Встроенные методы (Embedded methods)

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

Отбор признаков встроен в процесс обучения модели.

- Lasso (L1-регуляризация) — зануляет коэффициенты неинформативных признаков
- ElasticNet — комбинация L1 и L2
- Feature importance у деревьев — такие как feature_importances_ у RandomForest, XGBoost, CatBoost, LightGBM

Как штрафовать?

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

Регуляризация накладывает штраф на большие веса модели

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

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

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

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

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

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?

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

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?

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

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:

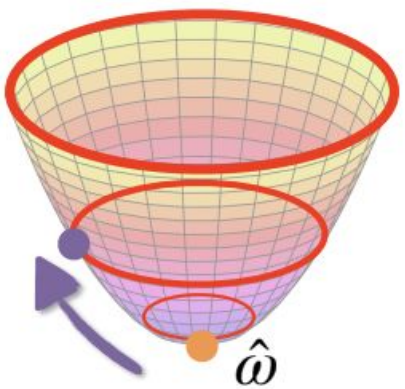
$$w_j = w_j - \alpha\left(-\frac{2}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)x_{ij} + 2\lambda w_j\right)$$

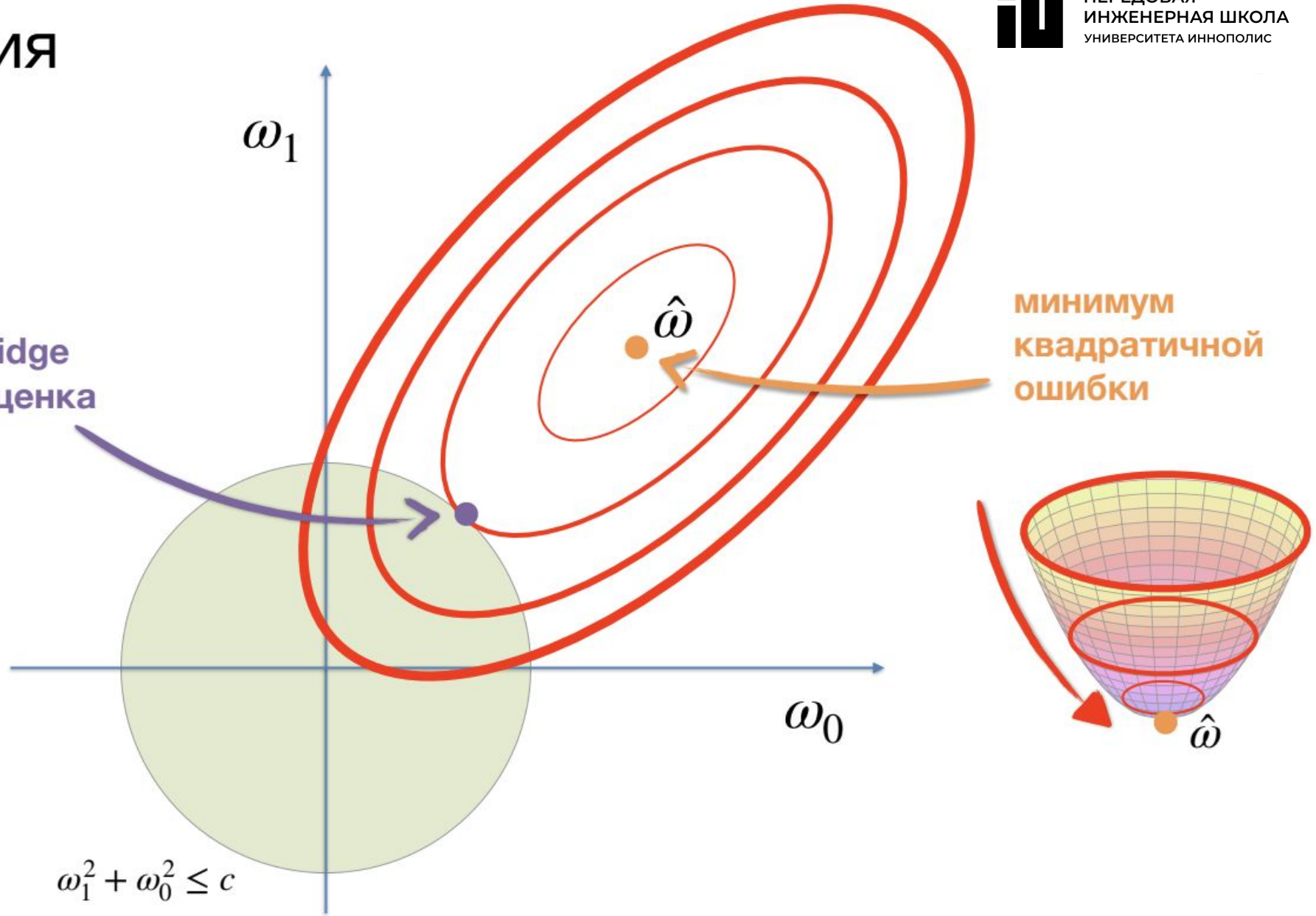
L2 регуляризация стремится уменьшить значения весов, но не "притягивает" их к нулю так же агрессивно, как это делает L1 регуляризация. Поскольку штраф в L2 регуляризации пропорционален квадрату веса, он становится меньше по мере уменьшения веса. Таким образом, когда вес становится малым, влияние регуляризационного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

ПЕРЕДОВАЯ
ИНЖЕНЕРНАЯ ШКОЛА
УНИВЕРСИТЕТА ИННОПОЛИС

Ridge регрессия

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

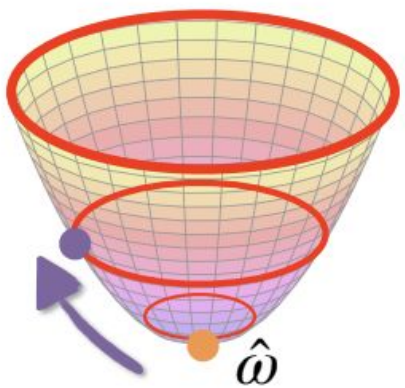


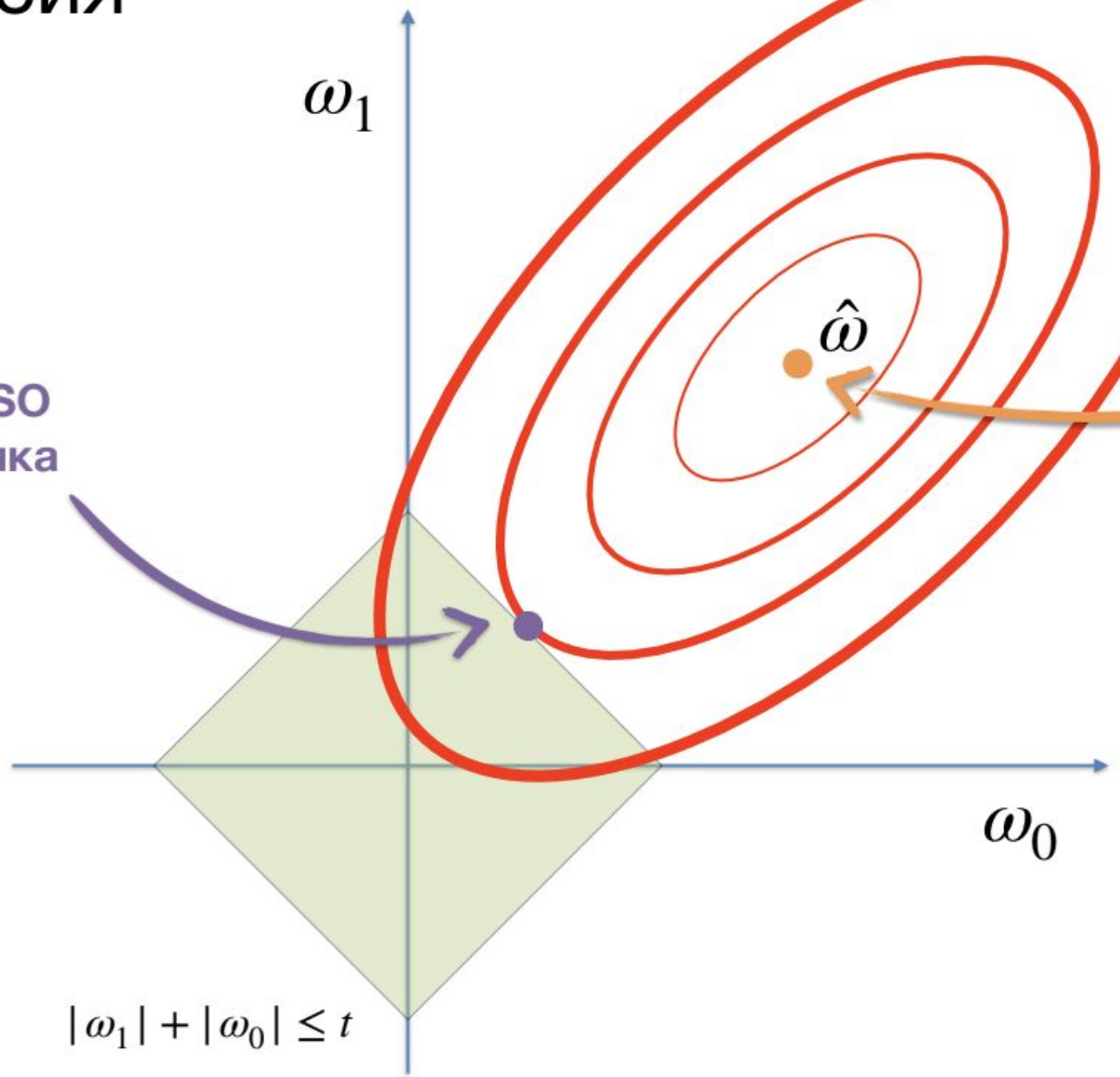


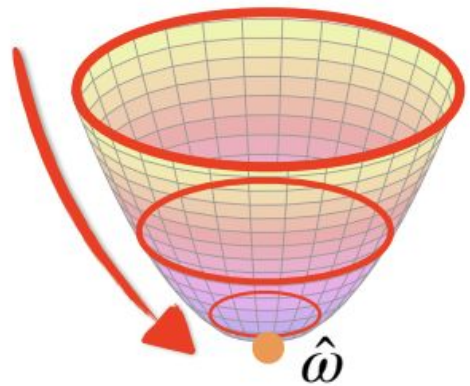
$$Q(w, X) = (Xw - y)^2 \rightarrow min$$

LASSO регрессия

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







$$Q(w, X) = (Xw - y)^2 \rightarrow min \qquad |\omega_1| + |\omega_0| \leq t$$

Регуляризация

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

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

LASSO регрессия

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

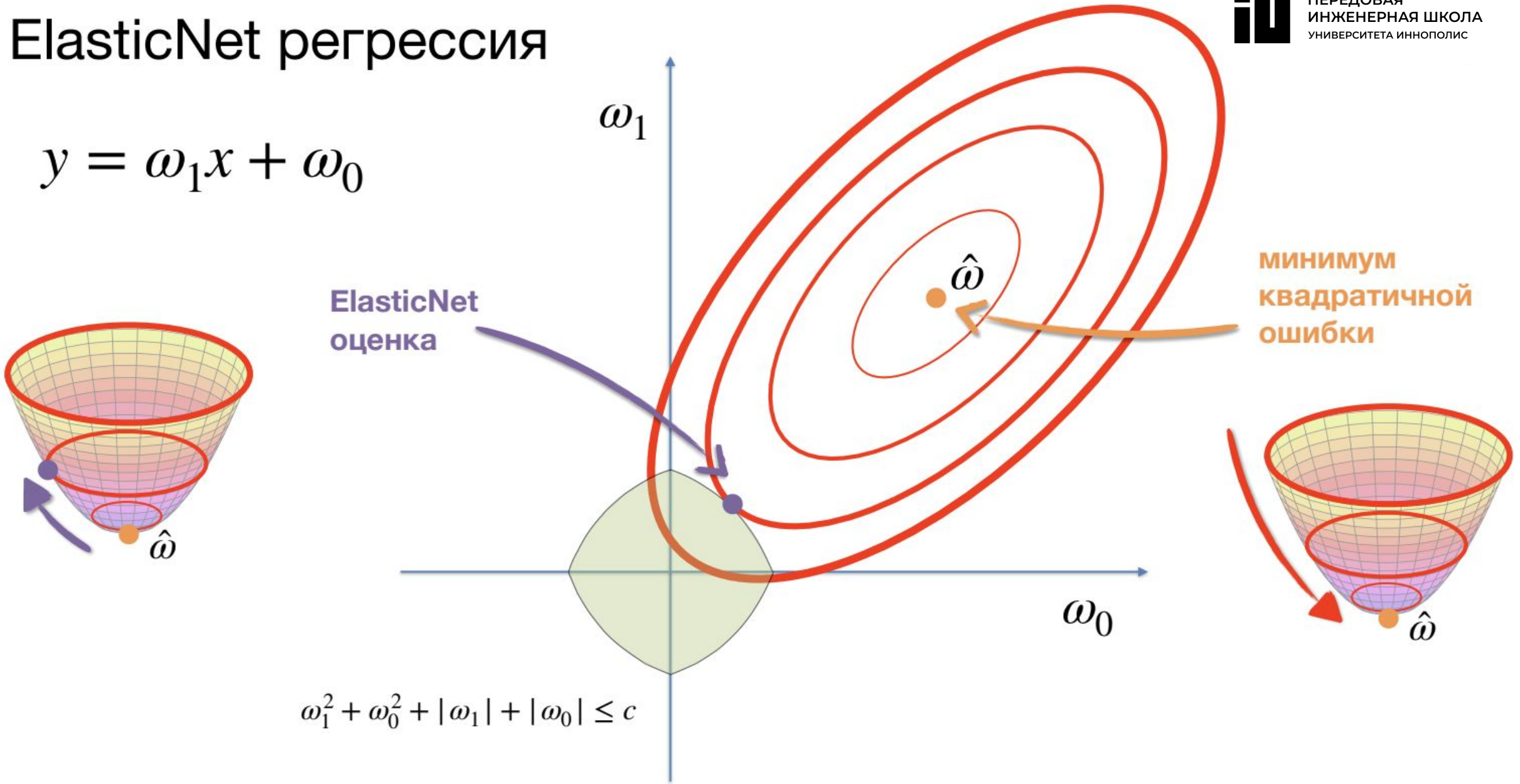
RIDGE регрессия

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

ElasticNet регрессия

ElasticNet регрессия

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



Регуляризация

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

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

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

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']

Передовые инженерные школы

МИНОБРНАУКИ РОССИИ

УНИВЕРСИТЕТ ИННОПОЛИС

онлайн университет

Спасибо за внимание