Курс «Введение в машинное обучение»

Методология машинного обучения

Воронцов Константин Вячеславович
k.v.vorontsov@phystech.edu
http://www.MachineLearning.ru/wiki?title=User:Vokov

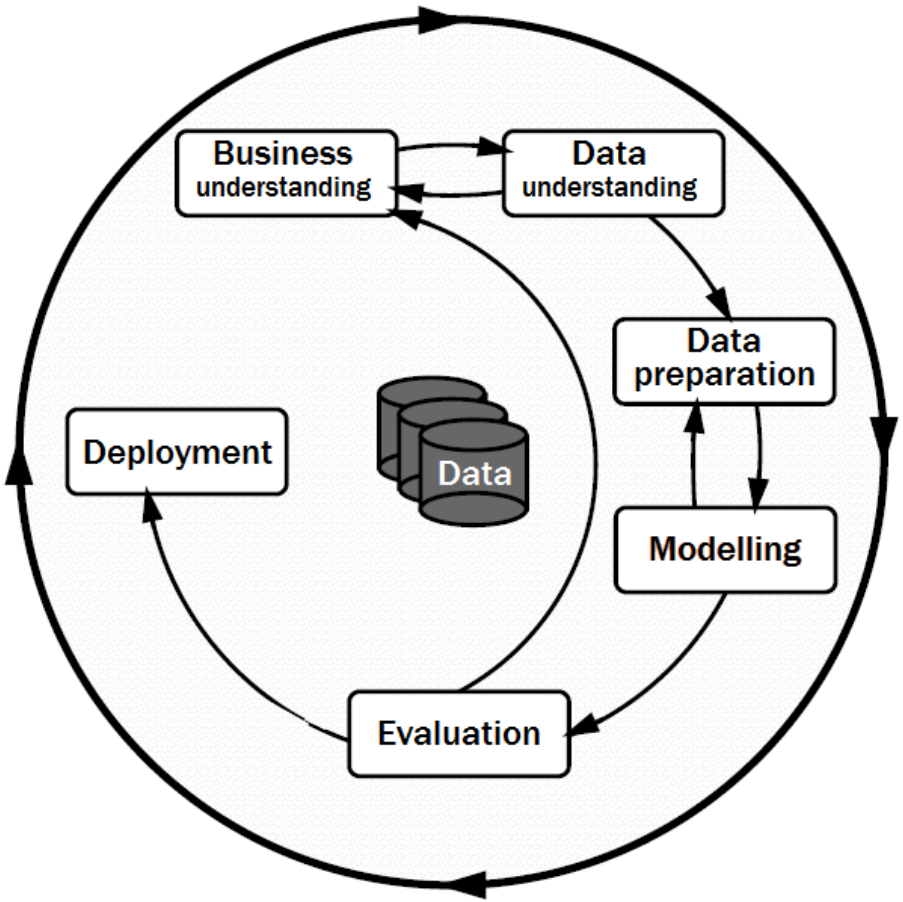
Этот курс доступен на странице вики-ресурса
http://www.MachineLearning.ru/wiki
«Введение в машинное обучение (курс лекций, К.В.Воронцов)»

МФТИ.ФПМИ.ИС.ИАД • 20 марта 2025

Содержание

Межотраслевой стандарт интеллектуального анализа данных

CRISP-DM: CRoss Industry Standard Process for Data Mining (1999)



Компании-инициаторы:

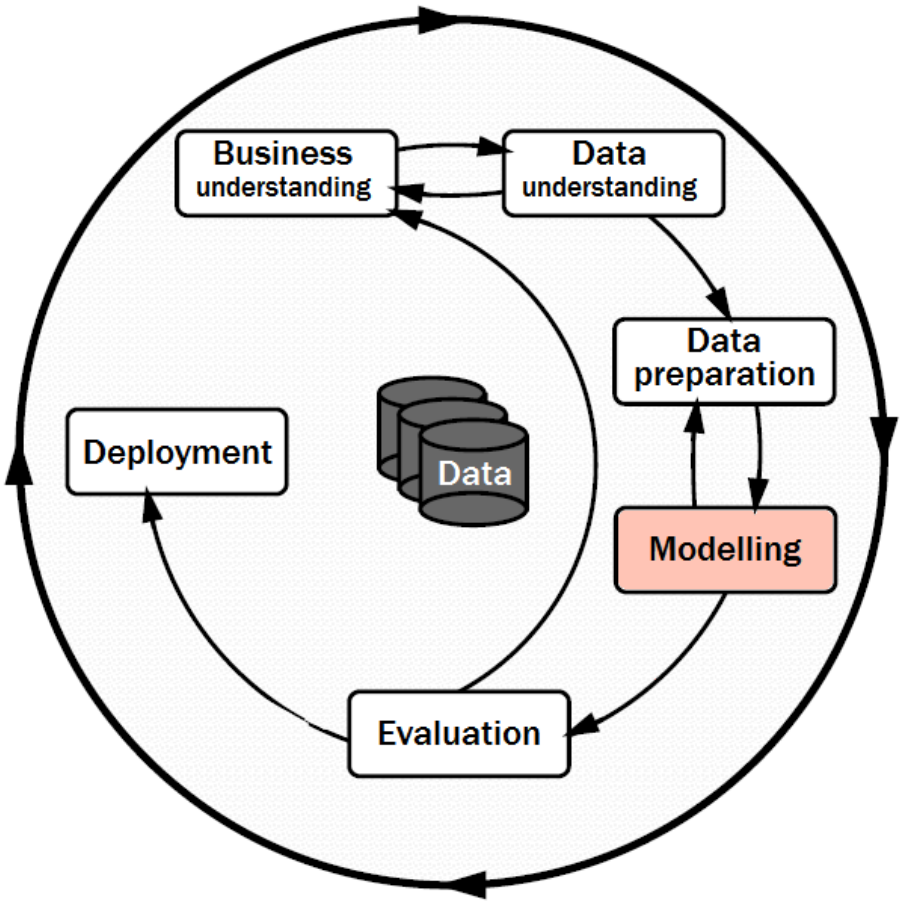
- SPSS
- Teradata
- Daimler AG
- NCR Corp.
- OHRA

Шаги процесса:

- понимание бизнеса
- понимание данных
- предобработка данных и инженерия признаков
- разработка моделей и настройка их параметров
- оценивание качества
- внедрение

Понимание эволюции ИИ как автоматизации шагов CRISP-DM

CRISP-DM: CRoss Industry Standard Process for Data Mining (1999)

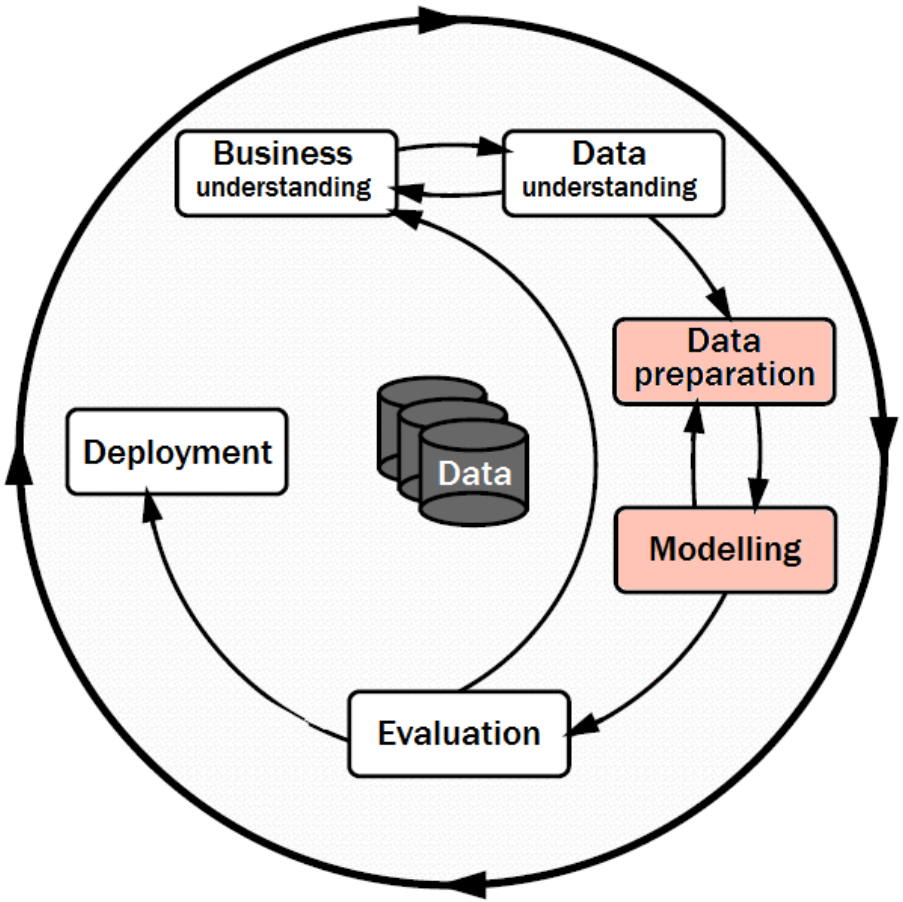


Эволюция ИИ:

- *Expert Systems:* жёсткие модели, основанные на правилах
- *Machine Learning:* параметрические модели, обучаемые по данным

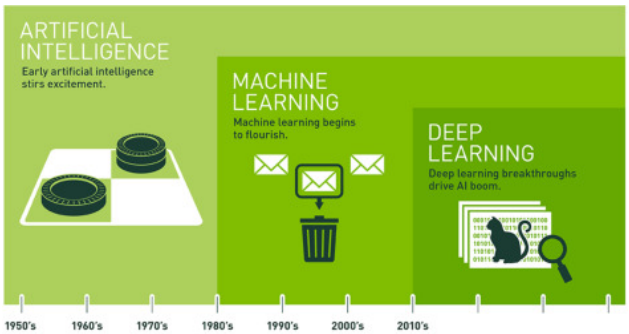
Понимание эволюции ИИ как автоматизации шагов CRISP-DM

CRISP-DM: CRoss Industry Standard Process for Data Mining (1999)



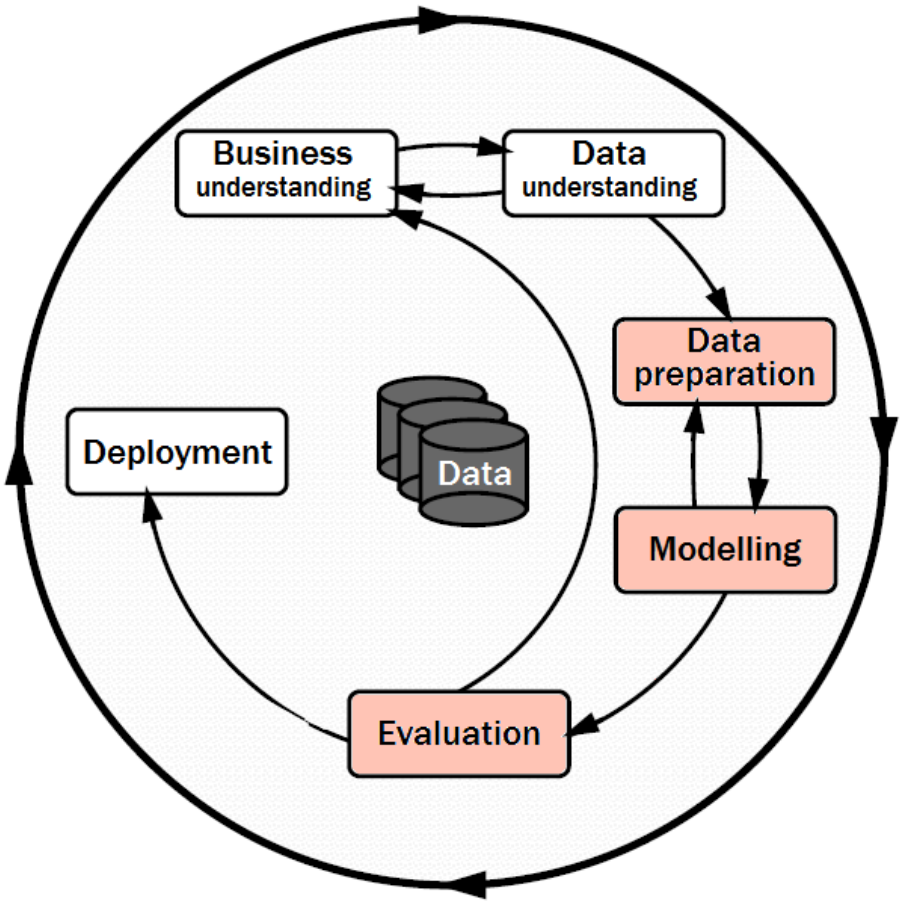
Эволюция ИИ:

- *Expert Systems:* жёсткие модели, основанные на правилах
- *Machine Learning:* параметрические модели, обучаемые по данным
- *Deep Learning:* модели с обучаемой векторизацией данных



Понимание эволюции ИИ как автоматизации шагов CRISP-DM

CRISP-DM: CRoss Industry Standard Process for Data Mining (1999)

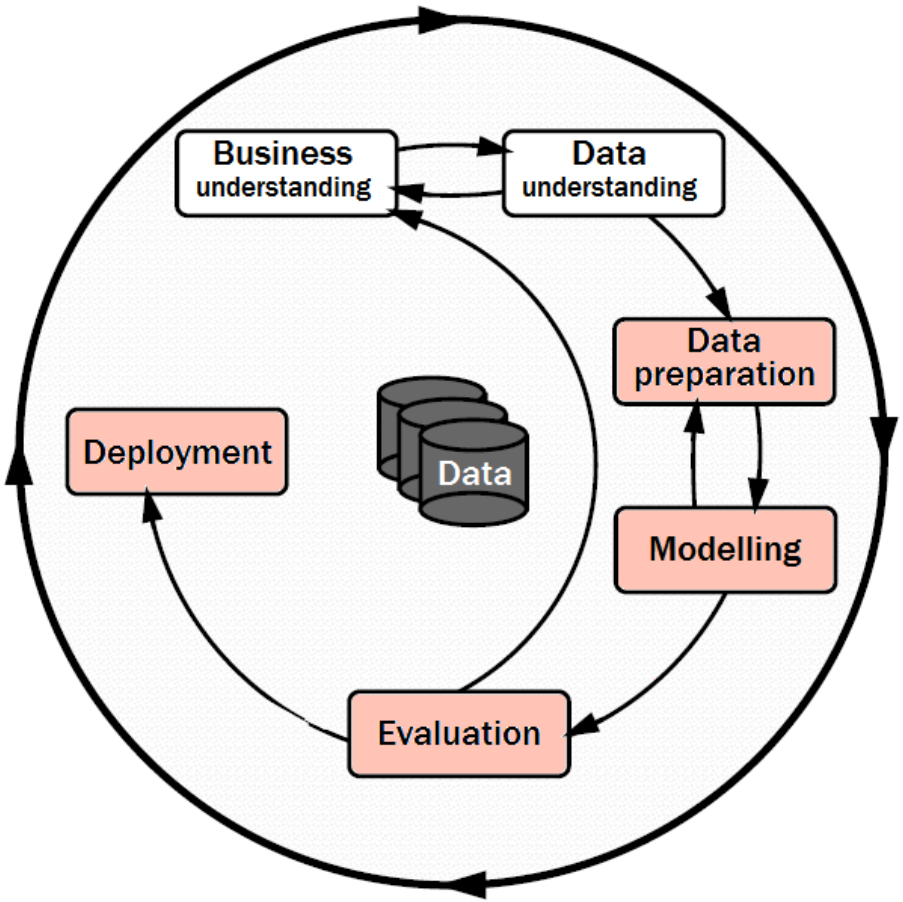


Эволюция ИИ:

- *Expert Systems:* жёсткие модели, основанные на правилах
- *Machine Learning:* параметрические модели, обучаемые по данным
- *Deep Learning:* модели с обучаемой векторизацией данных
- *AutoML:* автоматический выбор моделей и их строения

Понимание эволюции ИИ как автоматизации шагов CRISP-DM

CRISP-DM: CRoss Industry Standard Process for Data Mining (1999)



Эволюция ИИ:

- *Expert Systems:* жёсткие модели, основанные на правилах
- *Machine Learning:* параметрические модели, обучаемые по данным
- *Deep Learning:* модели с обучаемой векторизацией данных
- *AutoML:* автоматический выбор моделей и их строения
- *Lifelong Learning:* бесшовная интеграция в бизнес-процесс

Особенности данных и постановок прикладных задач

- разнородные (признаки измерены в разных шкалах)
- неполные (измерены не все, имеются пропуски)
- неточные (измерены с погрешностями)
- противоречивые (объекты одинаковые, ответы разные)
- избыточные (сверхбольшие, не помещаются в память)
- недостаточные (объектов меньше, чем признаков)
- сложно структурированные (нет признаковых описаний)

Риски, связанные с постановкой задачи:

- «грязные» данные
  (заказчик не обеспечивает качество данных)
- неясные критерии качества модели
  (заказчик не определился с целями или критериями)

Методы предварительной обработки данных

- Преобразование признаков (feature transformation)
  — усиление или ослабление шкалы измерения признака
  — нормализация, стандартизация
  — трансформация функции распределения признака
- Выделение признаков из сырых данных (feature extraction), конструирование признаков (feature engineering)
- Обучаемая векторизация данных (representation learning)
- Восполнение пропусков в данных (missing values imputation)
- Обнаружение выбросов (outlier/anomaly detection)
- Понижение размерности данных (dimensionality reduction)
- Отбор информативных признаков (feature selection)

Задачи оценивания и выбора моделей

Дано:

$X^\ell = (x_1, \ldots, x_\ell)$ — обучающая выборка

$A_t = \{a \colon X \times W_t \to Y\}$ — параметрические модели, $t \in T$

W_t — пространство параметров модели A_t

$\mu_t \colon (X \times Y)^\ell \to W_t$ — методы обучения, $t \in T$

Найти: метод μ_t с наилучшей *обобщающей способностью*.

Частные случаи:

- выбор лучшей модели A_t (model selection);
- выбор метода обучения μ_t для заданной модели A (в частности, оптимизация *гиперпараметров*);
- отбор признаков (feature selection):
  $F = \{f_j \colon X \to D_j \colon j = 1, \ldots, n\}$ — множество признаков;
  метод обучения μ_J использует только признаки $J \subseteq F$.

Обобщающая (предсказательная) способность метода

$\mathscr{L}(w, x)$ — функция потерь модели $a(w, x)$ на объекте x

$Q(w, X^\ell) = \frac{1}{\ell} \sum_{i=1}^{\ell} \mathscr{L}(w, x_i)$ — *критерий качества* $a(w, x)$ на X^ℓ

Внутренний критерий оценивает качество на обучении X^ℓ:

$$Q_\mu(X^\ell) = Q\big(\mu(X^\ell), X^\ell\big).$$

Недостаток: эта оценка смещена, т.к. μ минимизирует её же.

Внешний критерий оценивает качество «вне обучения», например, по отложенной (hold-out) контрольной выборке X^k:

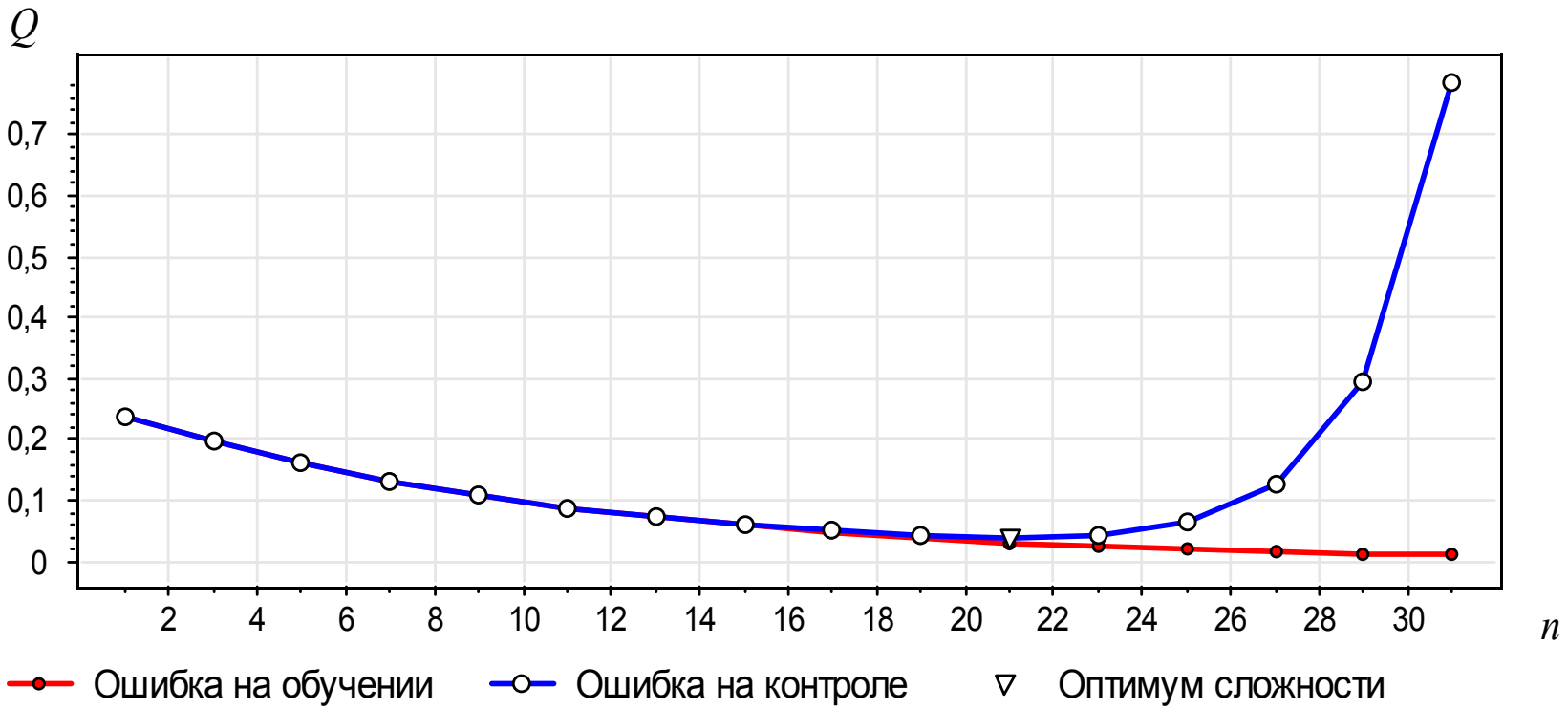
$$Q_\mu(X^\ell, X^k) = Q\big(\mu(X^\ell), X^k\big).$$

Недостаток: эта оценка зависит от разбиения $X^L = X^\ell \sqcup X^k$.

Основное отличие внешних критериев от внутренних

Внутренний критерий монотонно убывает с ростом сложности модели (числа признаков n или числа параметров $\dim w$).

Внешний критерий имеет характерный минимум, соответствующий оптимальной сложности модели.



Кросс-проверка (cross-validation, CV)

Усреднение оценок hold-out по заданному N — множеству разбиений $X^L = X_n^\ell \sqcup X_n^k$, $n = 1, \ldots, N$:

$$\mathrm{CV}(\mu, X^L) = \frac{1}{|N|} \sum_{n \in N} Q_\mu(X_n^\ell, X_n^k).$$

Частные случаи — разные способы задания множества N.

1. $|N| = 1$ — единственное разбиение: hold-out.

2. N — случайное множество разбиений: метод Монте-Карло.

3. N — множество всех $C_{\ell+k}^k$ разбиений:
полная кросс-проверка (complete cross-validation, CCV).

Недостаток: оценка CCV вычислительно слишком сложна.
Используются либо малые k, либо комбинаторные оценки CCV.

Скользящий контроль и поблочная кросс-проверка

4. Скользящий контроль (leave one out CV): $k = 1$,

$$\mathrm{LOO}(\mu, X^L) = \frac{1}{L} \sum_{i=1}^{L} Q_\mu(X^L \setminus \{x_i\}, \{x_i\}).$$

Недостатки LOO: ресурсоёмкость, высокая дисперсия.

5. Кросс-проверка по q блокам (q-fold CV): случайное разбиение $X^L = X_1^{\ell_1} \sqcup \cdots \sqcup X_q^{\ell_q}$ на q блоков (почти) равной длины,

$$\mathrm{CV}_q(\mu, X^L) = \frac{1}{q} \sum_{n=1}^{q} Q_\mu\left(X^L \setminus X_n^{\ell_n}, X_n^{\ell_n}\right).$$

Недостатки q-fold CV:

— оценка существенно зависит от разбиения на блоки;

— каждый объект лишь один раз участвует в контроле.

Многократная поблочная кросс-проверка

6. Контроль t раз по q блокам ($t \times q$-fold CV)
— стандарт «де факто» для тестирования методов обучения.

Выборка X^L разбивается t раз случайным образом на q блоков

$$X^L = X_{s1}^{\ell_1} \sqcup \cdots \sqcup X_{sq}^{\ell_q}, \quad s = 1, \ldots, t, \quad \ell_1 + \cdots + \ell_q = L;$$

$$\mathrm{CV}_{t \times q}(\mu, X^L) = \frac{1}{t} \sum_{s=1}^{t} \frac{1}{q} \sum_{n=1}^{q} Q_\mu\left(X^L \setminus X_{sn}^{\ell_n}, X_{sn}^{\ell_n}\right).$$

Преимущества $t \times q$-fold CV:

— увеличением t можно улучшать точность оценки (компромисс между точностью и временем вычислений);
— каждый объект участвует в контроле ровно t раз;
— оценивание доверительных интервалов (95% при $t = 40$).

Методология анализа ошибок (потерь)

$\mathscr{L}(w, x_i)$ — функция потерь (чем меньше, тем лучше).
Критерий средней потери модели $a(x, w)$ на выборке U:

$$Q(w, U) = \frac{1}{|U|} \sum_{x_i \in U} \mathscr{L}(w, x_i)$$

Анализ потерь на обучающей выборке:

- Ранжировать объекты по убыванию потерь $\mathscr{L}_i = \mathscr{L}(w, x_i)$
- Объекты со сверхбольшими потерями — выбросы?
- Если нет, то как улучшить модель на этих объектах?

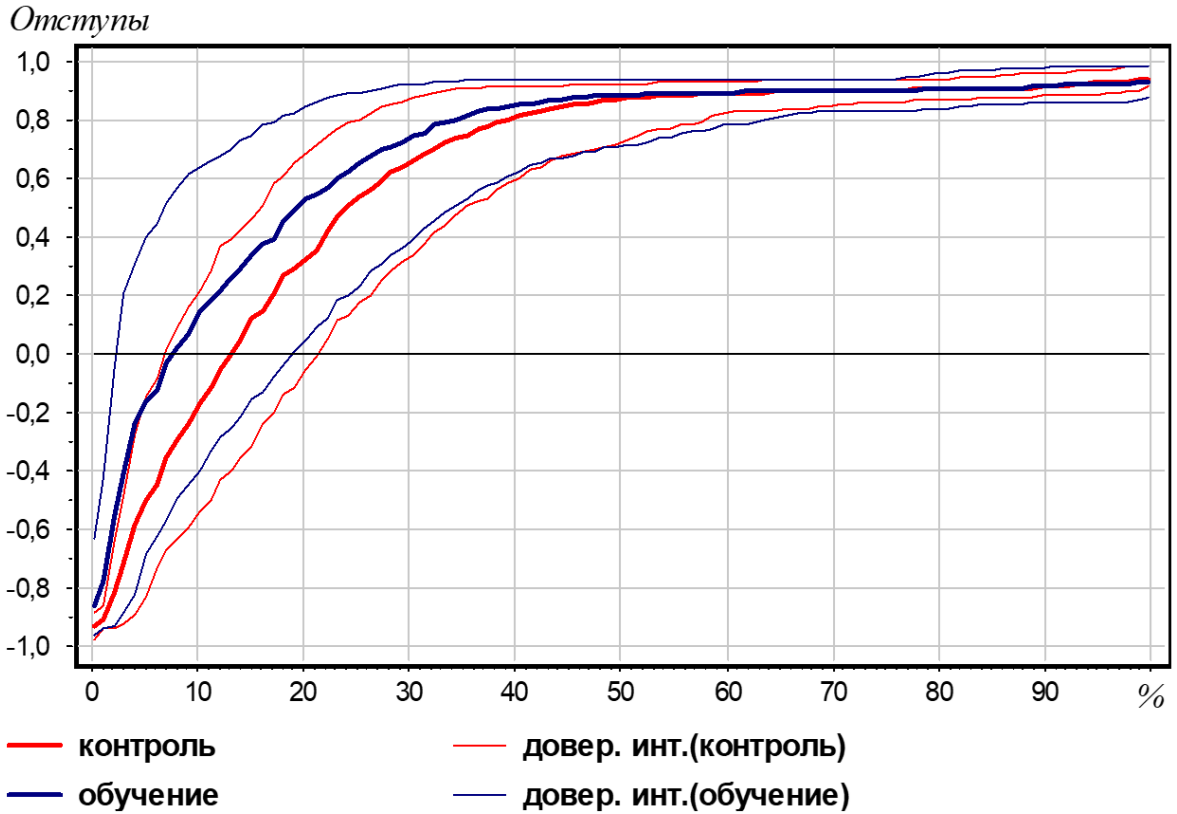
Сравнительный анализ потерь на обучении и тесте:

- Сильно ли отличаются распределения потерь?
- Если сильно, то как устранить переобучение?
- Объекты со сверхбольшими отличиями — выбросы?

Анализ распределения отступов в задаче классификации

Вместо потерь $\mathscr{L}_i = L(M_i)$ можно ранжировать отступы M_i

Видно: переобучение, зону неуверенной классификации

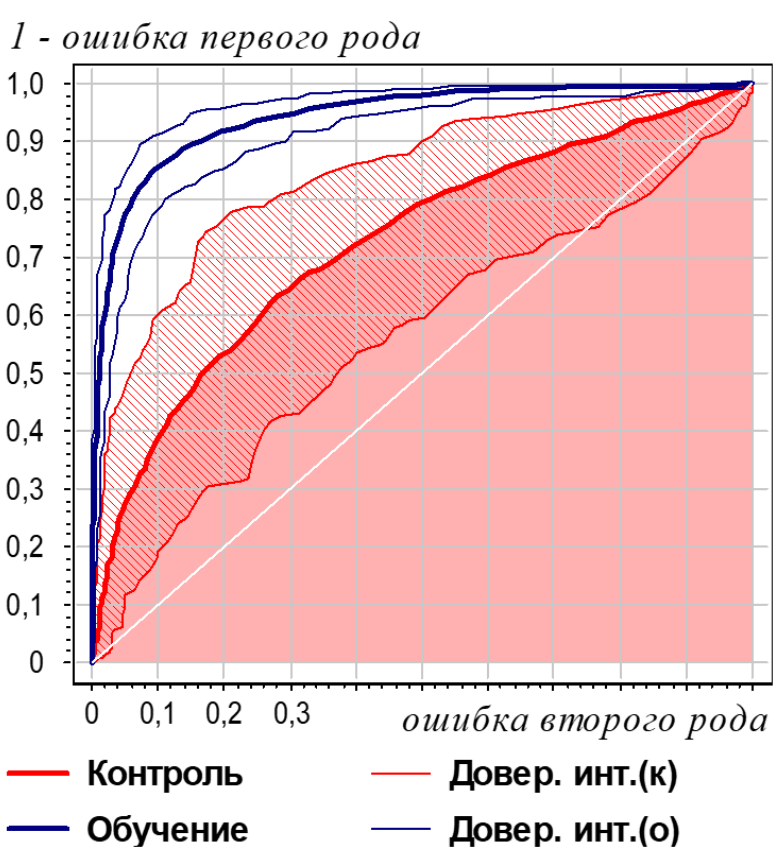


Задача UCI:australian, метод JRip

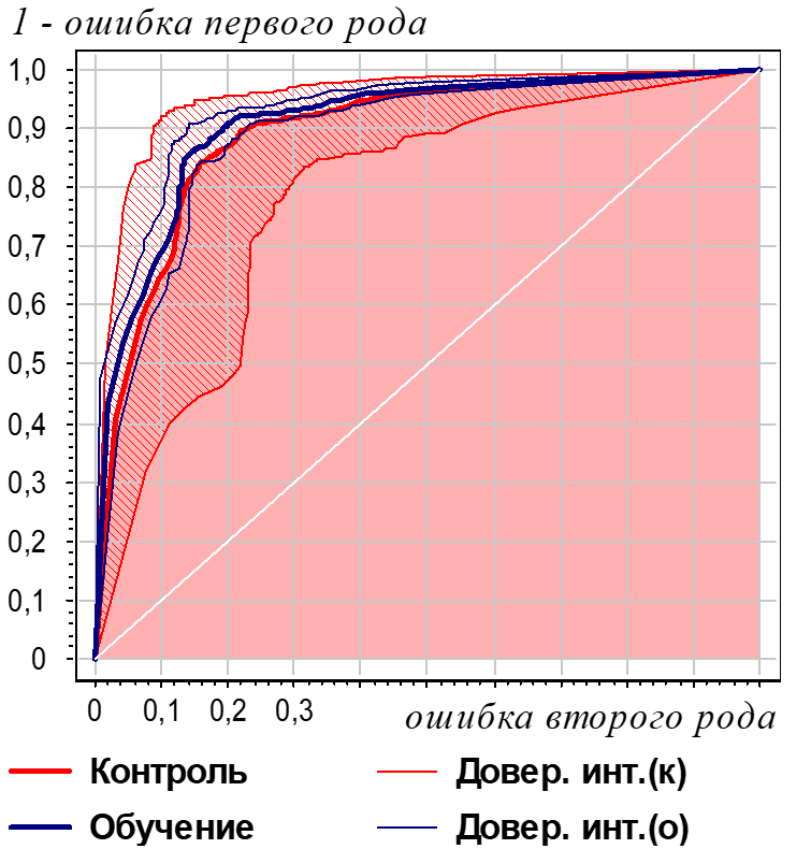
Анализ ROC-кривых

ROC-кривые можно строить отдельно для каждого класса

Видно: переобучение, устойчивость, различия классов



Задача UCI:liver, метод Bagging



Задача UCI:heart, метод Naïve Bayes

A/B тестирование (A/B testing, Split Testing)

Две модели, «базовая A» и «улучшенная B»,
построенные по историческим данным X^ℓ,
тестируются по метрике качества Q на новых данных X^k

В чём отличия A/B тестирования от обычного hold-out?

- X^k — это именно будущие данные (out-of-time), а не часть прошлых данных, исключённых из обучения (out-of-sample)
- больше реализма: за это время могут измениться свойства потока данных, реальные данные не обязаны быть i.i.d.
- однократный выбор модели почти не переобучается
- накопление данных X^k может потребовать много времени
- работа модели может влиять на формирование потока данных (например, в рекомендательных системах)

Мета-обучение (meta-learning, learning to learn)

Проблема: слишком много методов, слишком долго запускать

Дано: выборка «задача, метод» $\to$ критерии качества

Найти: модель многоклассовой классификации, предсказывающую, каким методом решать задачу

Критерий: точность предсказания оптимального метода

Признаки:

- размерные характеристики задачи
- характеристики пространства признаков: типы, выбросы, пропуски, корреляции
- результаты быстрых низкоразмерных методов

Joaquin Vanschoren. Meta-learning Architectures: Collecting, Organizing and Exploiting Meta-knowledge. 2009.

Joaquin Vanschoren. Meta-Learning: A Survey. 2018.

Автоматический выбор моделей и гиперпараметров (AutoML)

Проблема:
подбор структуры модели (архитектуры нейросети)
и гиперпараметров требует слишком много ресурсов

Дано: выборка «задача, структура» $\to$ критерии качества

Найти: какой следующий эксперимент провести с моделью

Критерий:
минимизация затрат ресурсов на автоматический поиск
оптимальной модели, сопоставимой по качеству с моделями,
построенными профессиональными исследователями

Близкая классическая задача — *планирование экспериментов*

Xin He et al. AutoML: A Survey of the State-of-the-Art. 2019
https://github.com/sberbank-ai-lab/LightAutoML — AutoML от Сбербанка

Эксперименты на реальных данных

Эксперименты на конкретной прикладной задаче:

- цель — решить задачу как можно лучше
- важно понимание задачи и данных
- важно придумывать информативные признаки
- конкурсы по анализу данных: http://www.kaggle.com

Эксперименты на наборах прикладных задач:

- цель — протестировать метод в разнообразных условиях
- нет необходимости (и времени) разбираться в сути задач :(
- признаки, как правило, уже кем-то придуманы
- репозиторий UC Irvine Machine Learning Repository http://archive.ics.uci.edu/ml (668 задачи, 2024-09-01)

Эксперименты на синтетических данных

Используются для тестирования новых методов обучения.
Преимущество — мы знаем истинную $y(x)$ (ground truth)

Эксперименты на синтетических данных:

- цель — отладить метод, выявить границы применимости
- объекты x_i из придуманного распределения (часто 2D)
- ответы $y_i = y(x_i)$ для придуманной функции $y(x)$
- двумерные данные + визуализация выборки

Эксперименты на полу-синтетических данных:

- цель — протестировать помехоустойчивость модели
- объекты x_i из реальной задачи (признаки + шум)
- ответы $y_i = y(x_i)$ для придуманной функции $y(x)$ (+ шум)

Напоминание. Общая постановка задач машинного обучения

Дано: X — пространство *объектов*
$X^\ell = \{x_1, \ldots, x_\ell\} \subset X$ — *обучающая выборка* (training sample)
$a(x, w)$, $a\colon X \times W \to Y$ — параметрическая модель, гипотеза

Найти $w \in W$ — вектор параметров модели $a(x, w)$

Критерий минимизации эмпирического риска (empirical risk minimization, ERM):

$$\sum_{i=1}^{\ell} \mathscr{L}(w, x_i) + \tau \mathscr{R}(w) \to \min_w$$

$\mathscr{L}(w, x)$ — *функция потерь* (loss function), тем больше, чем хуже модель $a(x, w)$ обработала объект x

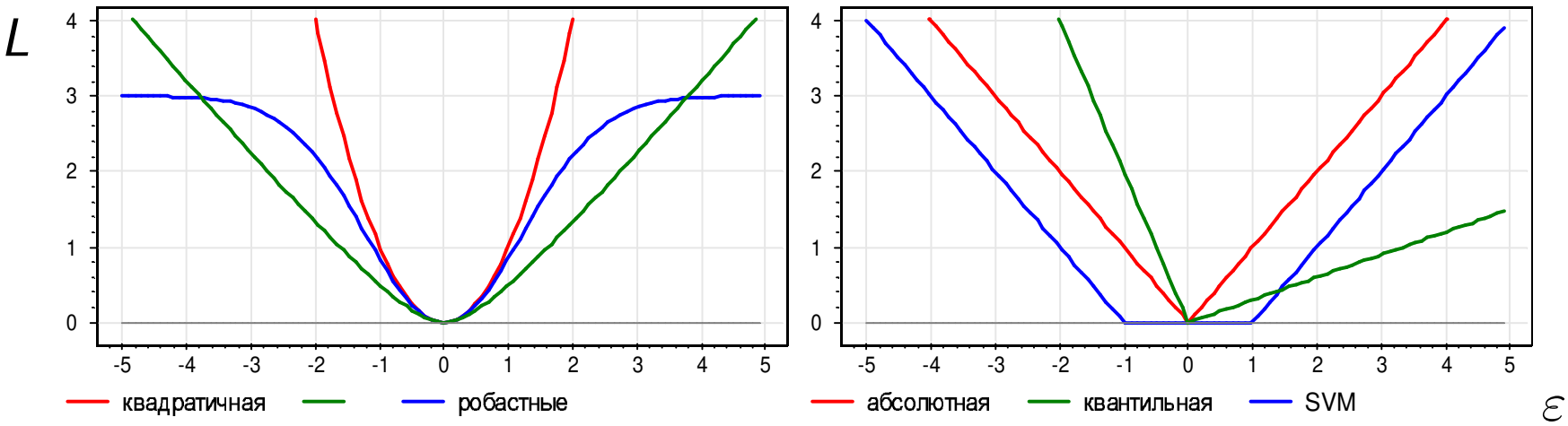
$\mathscr{R}(w)$ — *регуляризатор* для формализации дополнительных требований к модели, τ — коэффициент регуляризации

Напоминание. Обучение модели регрессии

Дано: обучающая выборка $(x_i, y_i)_{i=1}^{\ell}$ с ответами $y_i \in \mathbb{R}$
Найти: вектор параметров w модели регрессии $a(x, w)$
Критерий: минимум эмпирического риска

$$\sum_{i=1}^{\ell} L\big(a(x_i, w) - y_i\big) \;\to\; \min_w$$

Унимодальные функции потерь $L(\varepsilon)$ от невязки $\varepsilon = a(x, w) - y$:



Напоминание. Обучение бинарного классификатора

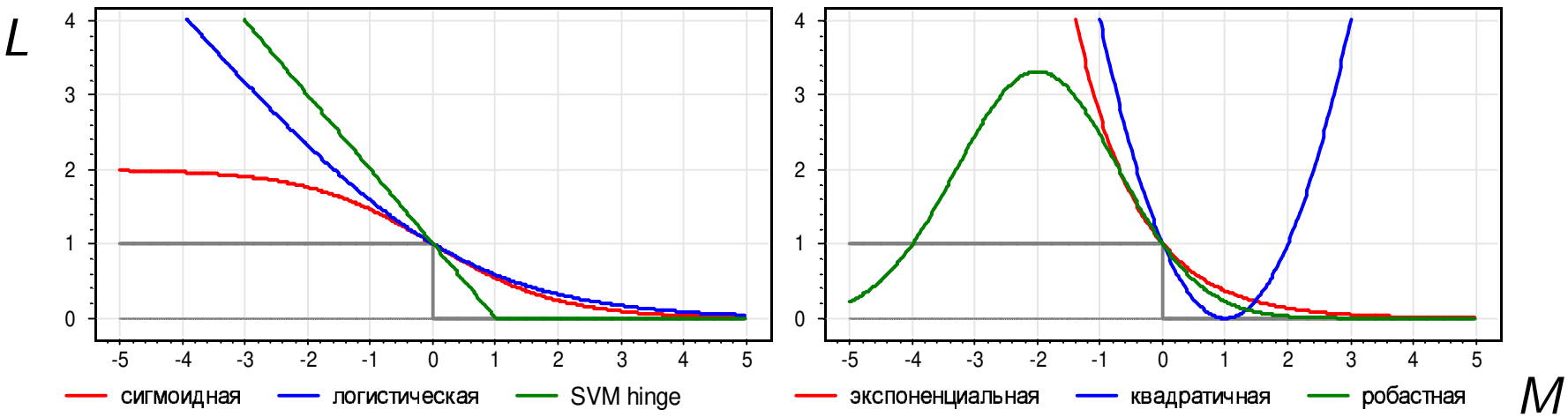
Дано: обучающая выборка $(x_i, y_i)_{i=1}^{\ell}$, $y_i \in \{-1, +1\}$

Найти: вектор w модели классификации $a(x, w) = \operatorname{sign} g(x, w)$

Критерий: min *аппроксимированного эмпирического риска*

$$\sum_{i=1}^{\ell} \big[g(x_i, w)y_i < 0\big] \leqslant \sum_{i=1}^{\ell} L\big(g(x_i, w)y_i\big) \;\to\; \min_w$$

Убывающие функции потерь $L(M)$ от *отступа* $M = g(x, w)y$:



Напоминание. Обучение многоклассового классификатора

Дано: обучающая выборка $(x_i, y_i)_{i=1}^{\ell}$, $y_i \in Y$, $|Y| < \infty$

Найти: вектор $w = (w_y : y \in Y)$ модели классификации

$$a(x, w) = \arg\max_{y \in Y} g_y(x, w_y)$$

Критерий «каждый против каждого»:

$$\sum_{i=1}^{\ell} \sum_{y \neq y_i} \Big[\underbrace{g_{y_i}(x_i, w_{y_i}) - g_y(x_i, w_y)}_{M_{iy}(w)} < 0\Big] \leqslant \sum_{i=1}^{\ell} \sum_{y \neq y_i} L\big(M_{iy}(w)\big) \to \min_w$$

Критерий «каждый против всех»:

$$\sum_{i=1}^{\ell} L\big(\min_{y \neq y_i} M_{iy}(w)\big) \to \min_w$$

где $M_{iy}(w)$ — отступ объекта x_i относительно класса y

Напоминание. Обучение ранжированию, максимизация AUC

Дано: обучающая выборка $(x_1, \ldots, x_\ell)$,
$i \prec j$ — отношение «x_j лучше, чем x_i» между объектами из X^ℓ

Найти: параметры w модели ранжирования $a(x, w)$,
восстанавливающей правильное отношение порядка:

$$i \prec j \quad \Rightarrow \quad a(x_i, w) < a(x_j, w)$$

Критерий: число неверно ранжированных пар объектов

$$Q(w) = \sum_{i \prec j} \big[\underbrace{a(x_j, w) - a(x_i, w)}_{M_{ij}(w)} < 0 \big]$$

$$\leqslant \sum_{i \prec j} L\big(a(x_j, w) - a(x_i, w)\big) \;\to\; \min_w$$

где $L(M)$ — убывающая функция парного отступа $M_{ij}(w)$

Обучение без учителя и с частичной разметкой

- восстановление плотности распределения (density estimation)
- восстановление смеси распределений (mixture estimation)
- кластеризация (clustering)
- обучение автокодировщика (autoencoder)
- одноклассовая классификация (one-class classification)
- обнаружение выбросов (outlier/anomaly/novelty detection)
- поиск ассоциативных правил (association rule learning)

Частичное обучение (semi-supervised learning)

- трансдуктивное обучение (transductive learning)
- обучение только на положительных примерах (PU-learning)

Задача кластеризации (clustering)

Дано: $X^\ell = \{x_1, \ldots, x_\ell\}$ — обучающая выборка, $x_i \in \mathbb{R}^n$

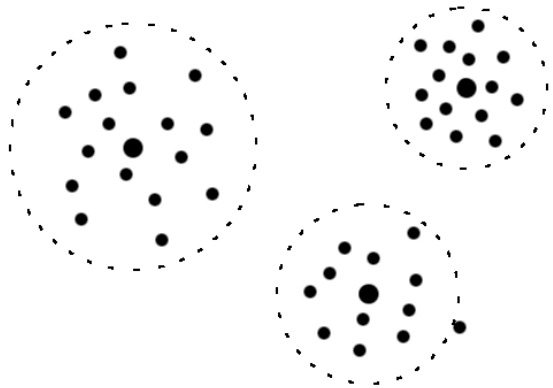
Найти:

— центры кластеров — параметры $\mu_j \in \mathbb{R}^n$, $j = 1, \ldots, K$

— какому кластеру принадлежит каждый объект $a_i \in \{1, \ldots, K\}$

Критерий: минимум суммы внутрикластерных расстояний

$$\sum_{i=1}^{\ell} \|x_i - \mu_{a_i}\|^2 \to \min_{\{a_i\}, \{\mu_j\}}$$



Метрика, как правило, евклидова (но может быть и другая):

$$\|x - \mu_j\|^2 = \sum_{d=1}^{n} \big(f_d(x) - \mu_{jd}\big)^2$$

Задача частичного обучения (semi-supervised learning, SSL)

Дано:

$X^k = \{x_1, \ldots, x_k\}$ — размеченные объекты (labeled data);
$\{y_1, \ldots, y_k\}$, $y_i \in Y$

$U = \{x_{k+1}, \ldots, x_\ell\}$ — неразмеченные объекты (unlabeled data).

Найти: классификации $\{a_{k+1}, \ldots, a_\ell\}$ неразмеченных, $a_i \in Y$

Критерий: без модели классификации (transductive learning):

$$\sum_{i=1}^{\ell} \|x_i - \mu_{a_i}\|^2 + \lambda \sum_{i=1}^{k} [a_i \neq y_i] \;\to\; \min_{\{a_i\},\{\mu_j\}}$$

Критерий с моделью классификации $a(x_i, w) = a_i \in Y$:

$$\sum_{i=1}^{\ell} \|x_i - \mu_{a(x_i,w)}\|^2 + \lambda \sum_{i=1}^{k} \mathscr{L}(w, x_i, y_i) + \tau R(w) \;\to\; \min_{w,\{\mu_j\}}$$

где $\mathscr{L}(w, x_i, y_i)$ — функция потерь для модели $a(x_i, w)$

Совместное обучение нескольких моделей

- частичное обучение (semi-supervised learning)
- обучение автокодировщика (autoencoder)
- обучаемая векторизация данных (representation learning)
- самостоятельное обучение (self-supervised learning)
- многомерное шкалирование (multidimensional scaling)
- предобучение (pre-training)
- перенос обучения (transfer learning)
- многозадачное обучение (multi-task learning)
- состязательное обучение (adversarial learning)
- дистилляция моделей или суррогатное моделирование
- обучение с привилегированной информацией

Дистилляция моделей или суррогатное моделирование

Обучение сложной модели $a(x, w)$ «долго, дорого»:

$$\sum_{i=1}^{\ell} \mathscr{L}\big(a(x_i, w), y_i\big) \;\to\; \min_{w}$$

Обучение простой модели $b(x, w')$, возможно, на других данных:

$$\sum_{i=1}^{k} \mathscr{L}\big(b(x'_i, w'), a(x'_i, w)\big) \;\to\; \min_{w'}$$

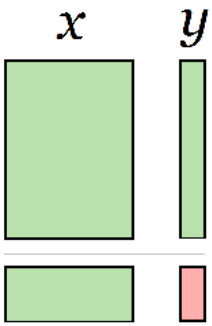
Примеры задач:

- замена сложной модели (климат, аэродинамика и др.), которая вычисляется на суперкомпьютере месяцами, «лёгкой» аппроксимирующей суррогатной моделью
- замена сложной нейросети, которая обучается неделями на больших данных, «лёгкой» аппроксимирующей нейросетью с минимизацией числа нейронов и связей

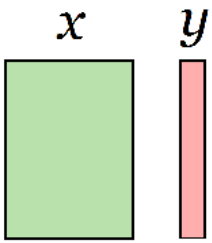
Обучение с использованием привилегированной информации

LUPI — Learning Using Privileged Information

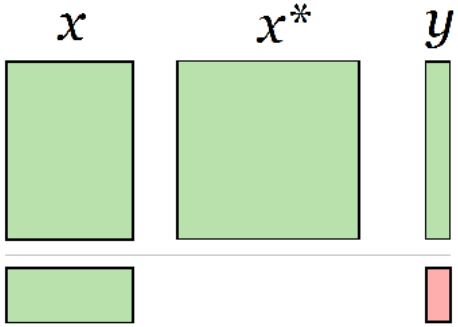
с учителем



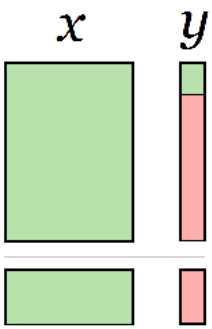
без учителя



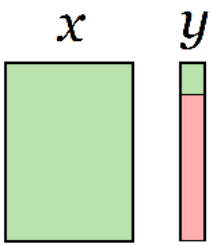
привилегированное (LUPI)



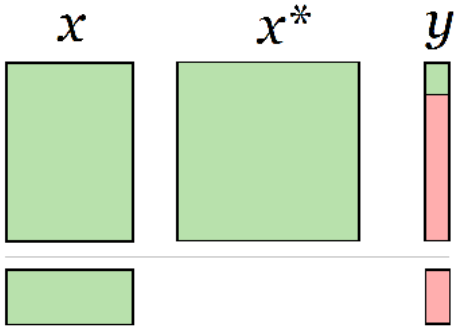
частичное



трансдуктивное



частичное LUPI



V.Vapnik, A.Vashist. A new learning paradigm: Learning Using Privileged Information // Neural Networks. 2009.

Примеры задач с привилегированной информацией x^*

- x — первичная (1D) структура белка
  x^* — третичная (3D) структура белка
  y — иерархическая классификация функции белка
- x — предыстория временного ряда
  x^* — информация о будущем поведении ряда
  y — прогноз следующей точки ряда
- x — текстовый документ
  x^* — выделенные ключевые слова или фразы
  y — категория документа
- x — пара (запрос, документ)
  x^* — выделенные асессором ключевые слова или фразы
  y — оценка релевантности

Обучение с привилегированной информацией

Раздельное обучение модели-ученика и модели-учителя:

$$\sum_{i=1}^{\ell} \mathscr{L}\big(a(x_i, w), y_i\big) \to \min_w \qquad \sum_{i=1}^{\ell} \mathscr{L}\big(a(x_i^*, w^*), y_i\big) \to \min_w$$

Модель-ученик обучается повторять ошибки модели-учителя:

$$\sum_{i=1}^{\ell} \mathscr{L}\big(a(x_i, w), y_i\big) + \mu \mathscr{L}\big(a(x_i, w), a(x_i^*, w^*)\big) \to \min_w$$

Совместное обучение модели-ученика и модели-учителя:

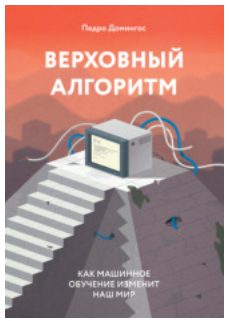
$$\sum_{i=1}^{\ell} \mathscr{L}\big(a(x_i, w), y_i\big) + \lambda \mathscr{L}\big(a(x_i^*, w^*), y_i\big) + \\ + \mu \mathscr{L}\big(a(x_i, w), a(x_i^*, w^*)\big) \to \min_{w, w*}$$

D.Lopez-Paz, L.Bottou, B.Scholkopf, V.Vapnik. Unifying distillation and privileged information. 2016.

Основные школы машинного обучения

1. *символизм* – поиск логических закономерностей
   - Decision Tree, Rule Induction
2. *коннекционизм* – обучаемые нейронные сети
   - BackPropagation, Deep Belief Nets, Deep Learning
3. *эволюционизм* – саморазвитие сложных моделей
   - Genetic Algorithms, Genetic Programming, Symbolic Regression
4. *байесионизм* – оценивание распределений параметров
   - Naive Bayes, Bayesian Networks, Graphical Models
5. *аналогизм* – «близким объектам близкие ответы»
   - kNN, RBF, SVM, Kernel Smoothing

⊕ *композиционизм* – кооперация моделей
   - Weighted Voting, Boosting, Bagging, Stacking, Random Forest, Яндекс.CatBoost



Домингос П. Верховный алгоритм. 2016. 336 с.

Задача онлайнового обучения

Задача обучения с учителем на потоке данных:

$(x_i, y_i)_{i=1}^{\ell}$ — последовательность прецедентов «объект, ответ»

$a(x, w)$ — параметрическая модель зависимости $y(x)$

$\mathscr{L}(a, y)$ — функция потерь

инициализировать параметры модели w_0;

для всех $i = 1, \ldots, \ell$

- получить очередной объект x_i;
- сделать предсказание $a_i := a(x_i, w_{i-1})$;
- получить ответ y_i и оценить потерю $\mathscr{L}_i := \mathscr{L}(a_i, y_i)$;
- обновить модель $w_i := \text{Update}(w_{i-1}, x_i, y_i)$;

$Q(t) = \frac{1}{t} \sum_{i=1}^{t} \mathscr{L}_i$ — кривая обучения (LOO learning curve)

Steven C. H. Hoi et al. Online learning: a comprehensive survey. 2018

Проблематика инкрементного и онлайнового обучения

- Как эффективно обновить модель по одному прецеденту?
- Как усложнять модель по мере роста объёма данных?
- Как обеспечить то же качество, что в оффлайне?
- Как избежать хранения всей выборки данных?
- Как при этом избежать «катастрофического забывания»?
- Как, добавляя новые объекты, ещё и удалять старые?

Что может добавляться в задачах машинного обучения:

- объекты — основной, но не единственный случай
- признаки
- размерность модели
- классы/кластеры
- подвыборки/подзадачи
- области пространства данных, разладки (concept drift)

Online Learning ≠ Incremental Learning. В чём отличия?

- Online обрабатывает объекты в потоке, по одному
  Incremental может накапливать пакеты обновлений
- Online может забывать старые данные (catastrophic forgetting)
  Incremental часто подразумевает эквивалентность результата оффлайновому обучению по полной выборке
- Online исследования озабочены теоретическими гарантиями
  Incremental сосредоточен на реализации быстрых алгоритмов
- Online обязательно является Incremental
  Incremental НЕ обязательно является Online

Continual (lifelong) learning — обучение одной модели разным задачам так, чтобы новые задачи не вытесняли старые

Anytime algorithm — алгоритм, который обучается по потоку, но в любой момент может быть использован для предсказаний

Задачи прогнозирования временных рядов

Дано: $y_0, y_1, \ldots, y_t, \ldots$ — временной ряд, $y_i \in \mathbb{R}$
Найти: $\hat{y}_{t+d}(w) = f_{t,d}(y_1, \ldots, y_t; w)$ — модель временного ряда,
где $d = 1, \ldots, D$, D — горизонт прогнозирования,
w — вектор параметров модели.

Критерий: минимум среднеквадратичной ошибки прогнозов:

$$\sum_{t=T_0}^{T} \left(\hat{y}_{t+d}(w) - y_{t+d}\right)^2 \;\to\; \min_w$$

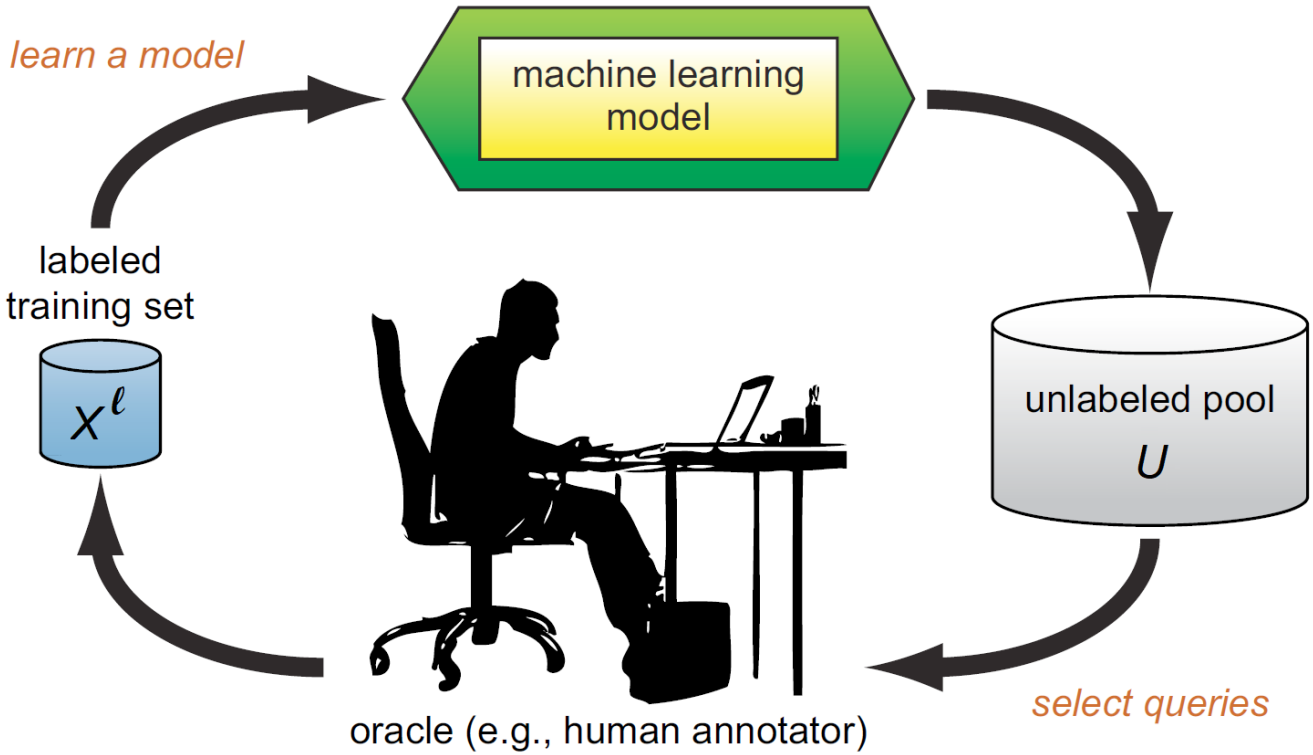
Пример: линейная модель авторегрессии.
В роли признаков выступают n предыдущих наблюдений ряда:

$$\hat{y}_{t+1}(w) = \sum_{j=1}^{n} w_j y_{t-j+1}, \qquad w \in \mathbb{R}^n$$

В роли объектов $\ell = t - n + 1$ моментов истории ряда.

Постановка задачи активного обучения

Задача: обучение модели $a: X \to Y$ по выборке (x_i, y_i), когда получение ответов $y_i = y(x_i)$ стоит дорого.



Burr Settles. Active Learning Literature Survey. 2010.

Постановка задачи активного обучения

Задача: обучение модели $a\colon X \to Y$ по выборке (x_i, y_i), когда получение ответов $y_i = y(x_i)$ стоит дорого.

Вход: $X^\ell = (x_i, y_i)_{i=1}^\ell$ — выборка размеченных объектов;
$U = (u_i)_{i=1}^K$ — выборка (пул) неразмеченных объектов;
Выход: модель a и размеченная выборка $(u_i, y_i^*)_{i=1}^k$, $k \leqslant K$;

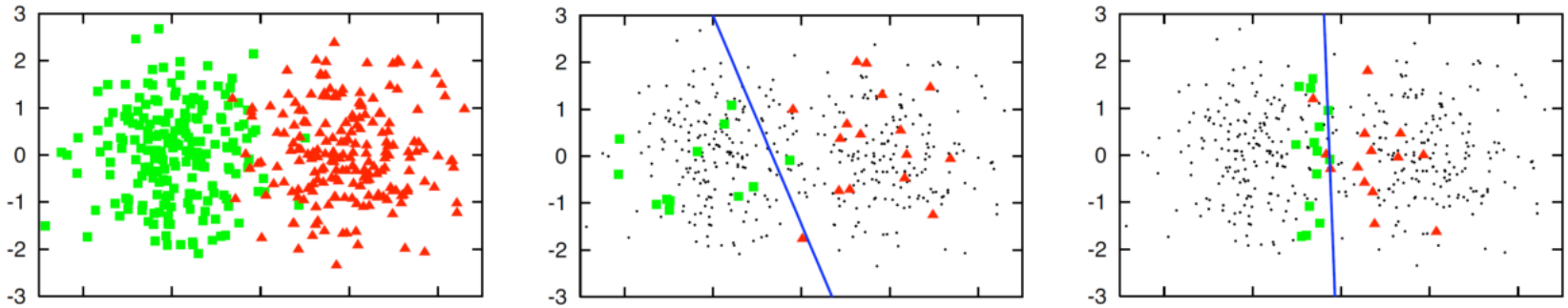
обучить модель a по начальной выборке $(x_i, y_i)_{i=1}^\ell$;
пока есть неразмеченные объекты и модель не обучилась
- $u_i = \arg\max\limits_{u \in U} \phi(u)$ максимум оценки перспективности;
- узнать для него $y_i^* = y(u_i)$;
- дообучить модель $a(x)$ ещё на одном примере (u_i, y_i^*);

Цель: достичь как можно лучшего качества модели a, использовав как можно меньше дополнительных примеров k.

Почему активное обучение быстрее пассивного

Пример. Синтетические данные: $\ell = 30$, $\ell + k = 400$;

(a) два гауссовских класса;
логистическая регрессия по 30 объектам:
(b) случайным;
(c) отобранным по максимуму неуверенности классификации.



(a) (b) (c)

Обучение по смещённой неслучайной выборке требует меньше данных для построения алгоритма сопоставимого качества.

Burr Settles. Active Learning Literature Survey. 2010.

Примеры приложений активного обучения

- сбор асессорских данных для информационного поиска, анализа текстов, сигналов, речи, изображений, видео
- в том числе на платформах краудсорсинга
- *планирование экспериментов* в естественных науках или на производстве (пример — комбинаторная химия)
- оптимизация трудно вычислимых функций (пример — оптимизация гиперпараметров, AutoML)

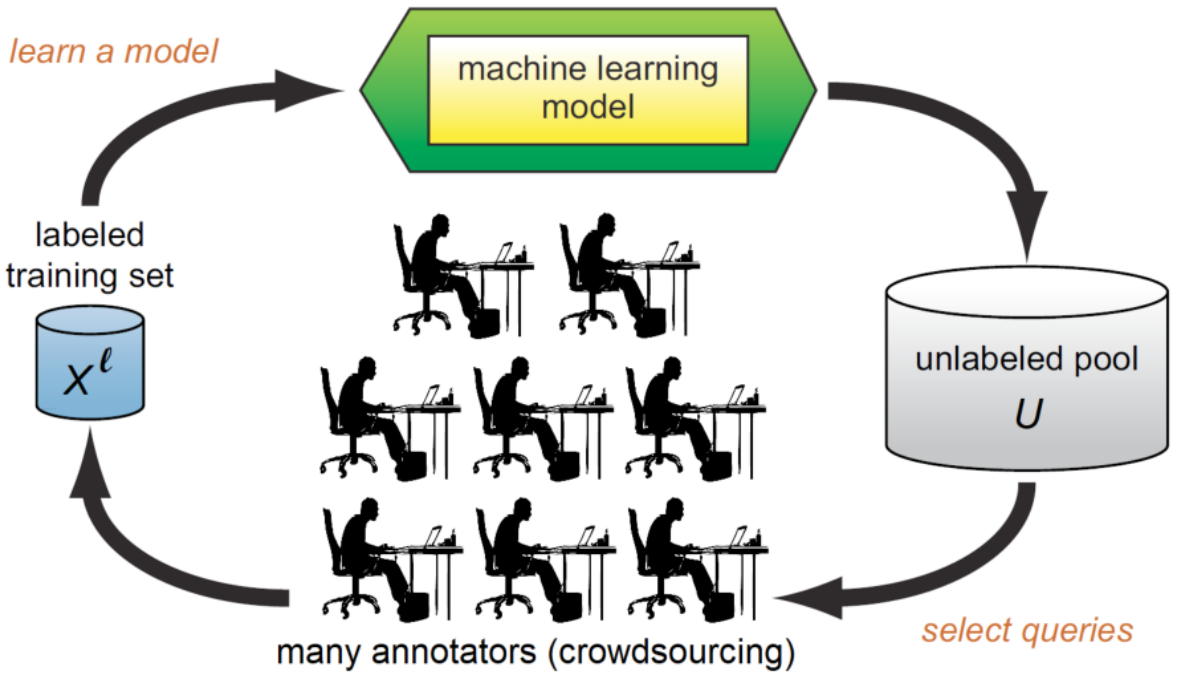
Применения в бизнесе:

- управление ценами и ассортиментом в торговых сетях
- выбор товара для проведения маркетинговой акции
- проактивное взаимодействие с клиентами
- выборочный контроль качества
- выявление аномалий в данных, случаев мошенничества

Краудсорсинг: активное обучение, когда аннотаторов много

y_{it} — ответы аннотаторов $t \in T$ на объекте u_i

Задача: сформировать согласованный ответ (консенсус) $\hat{y}_i$
и оценить надёжность каждого аннотатора $q_t = P[y_{it} = \hat{y}_i]$



Р.А.Гилязев, Д.Ю.Турдаков. Активное обучение и краудсорсинг: обзор методов оптимизации разметки данных. 2018.

Задача о многоруком бандите (multi-armed bandit)

A — конечное множество возможных *действий*
$p(r|a)$ — неизвестное распределение *премии* $r \in \mathbb{R}$ для $a \in A$
$\pi_t(a)$ — *стратегия* (policy) агента в раунде t, распределение на A

Игра агента со средой: инициализация стратегии $\pi_1(a)$;
для всех раундов $t = 1, \ldots, T, \ldots$
- агент выбирает действие $a_t \sim \pi_t(a)$;
- среда генерирует премию $r_t \sim p(r|a_t)$;
- агент корректирует стратегию $\pi_{t+1}(a)$;

$$Q_t(a) = \frac{\sum_{i=1}^{t} r_i[a_i = a]}{\sum_{i=1}^{t}[a_i = a]} \quad \text{— средняя премия в } t \text{ раундах}$$

$$Q^*(a) = \lim_{t\to\infty} Q_t(a) \to \max_{a\in A} \quad \text{— } \textit{ценность действия } a$$

Постановка задачи в случае, когда агент влияет на среду

A — конечное множество возможных *действий* (action)

S — конечное множество состояний среды (state)

Игра агента со средой: инициализация стратегии $\pi_1(a \mid s)$ и состояния среды s_1;

для всех раундов $t = 1, \ldots, T, \ldots$

- агент выбирает действие $a_t \sim \pi_t(a \mid s_t)$;
- среда генерирует премию $r_t \sim p(r \mid a_t, s_t)$
- и новое состояние $s_{t+1} \sim p(s \mid a_t, s_t)$;
- агент корректирует стратегию $\pi_{t+1}(a \mid s)$;

Функции ценности:

$$V^\pi(s) = \mathsf{E}_\pi\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k} \;\middle|\; s_t = s\right)$$ — состояния s

$$Q^\pi(s,a) = \mathsf{E}_\pi\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k} \;\middle|\; s_t = s,\ a_t = a\right)$$ — действия в состоянии

Примеры прикладных задач

- Управление роботами, технологическими процессами
- Генерация движений персонажей в мультипликации
- Рекомендация новостных статей пользователям
- Показ рекламы в Интернете
- Управление портфелем ценных бумаг, игра на бирже
- Управление ценами и ассортиментом в сетях продаж
- Маршрутизация в телекоммуникационных сетях
- Стратегические игры: шахматы, го, Dota2, StarCraft2, ...

Обобщения постановки задачи:

- Есть информация о состоянии среды или о контексте
- Есть параметрическая модель стратегии/ценности/среды

H. Robbins. Some aspects of the sequential design of experiments. 1952.

Отличия от обычных задач машинного обучения

- выборка (s_t, a_t, r_t) не является независимой
- распределение $p(s_t, a_t, r_t)$ может меняться во времени и зависеть от стратегии агента π
- премии могут быть
  — отложенными (оценивать действия с задержкой)
  — разреженными (почти всё время $r_t = 0$)
  — зашумлёнными (не ясно, за что именно премия)

Какие параметрические модели можно обучать:

- функцию ценности действия в состоянии $Q(s, a; w)$
- функцию ценности состояния $V(s; w)$
- стратегию $\pi_{t+1}(a|s; w)$
- модель среды $(r_t, s_{t+1}) = \mu(s_t, a_t; w)$