• Введение в машинное обучение •

Методология машинного обучения

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

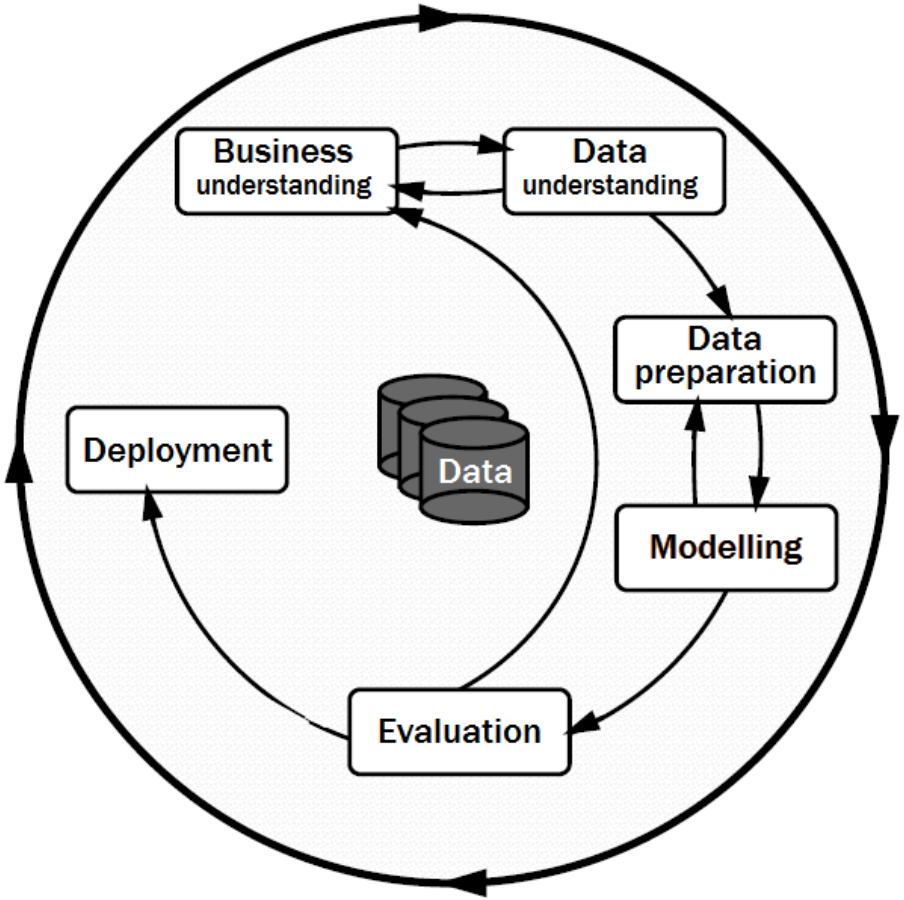
Этот курс доступен на странице вики-ресурса
http://www.MachineLearning.ru/wiki
«Введение в машинное обучение (курс лекций, К.В.Воронцов)»

МФТИ.ФПМИ.ИС.ИАД • 19 марта 2026

Содержание

Межотраслевой стандарт интеллектуального анализа данных

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



Компании-инициаторы:

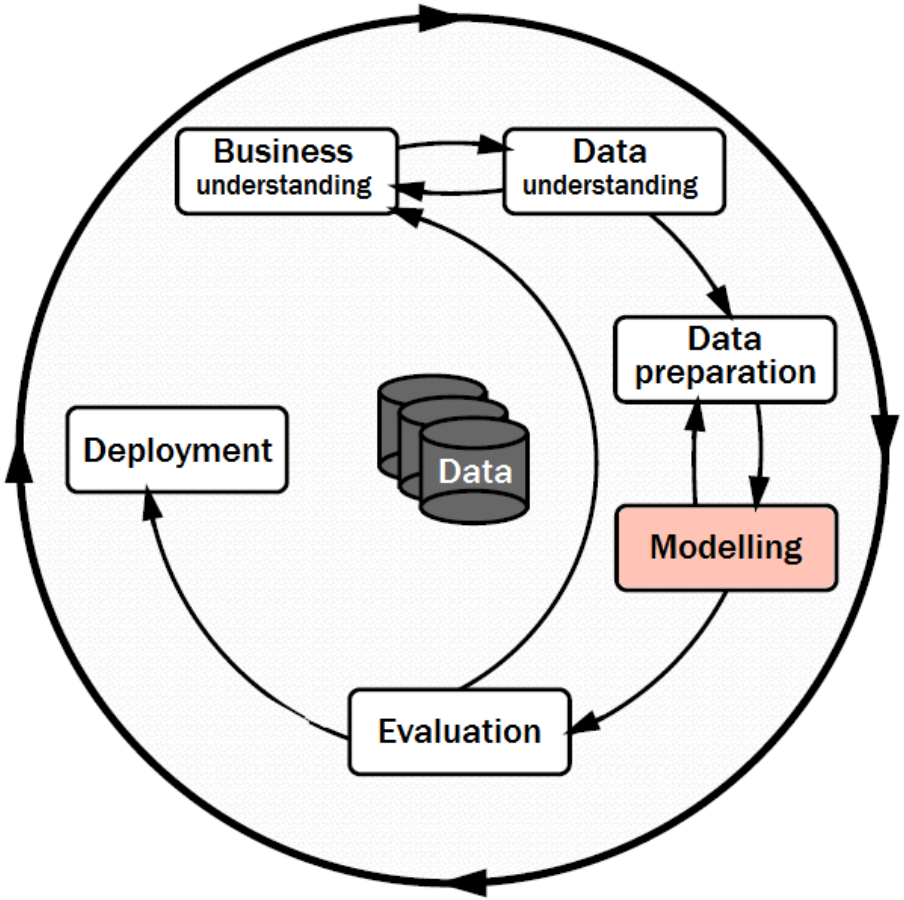
- SPSS
- Teradata
- Daimler AG
- NCR Corp.
- OHRA

Шаги процесса:

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

Эволюция ИИ как автоматизация шагов CRISP-DM

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

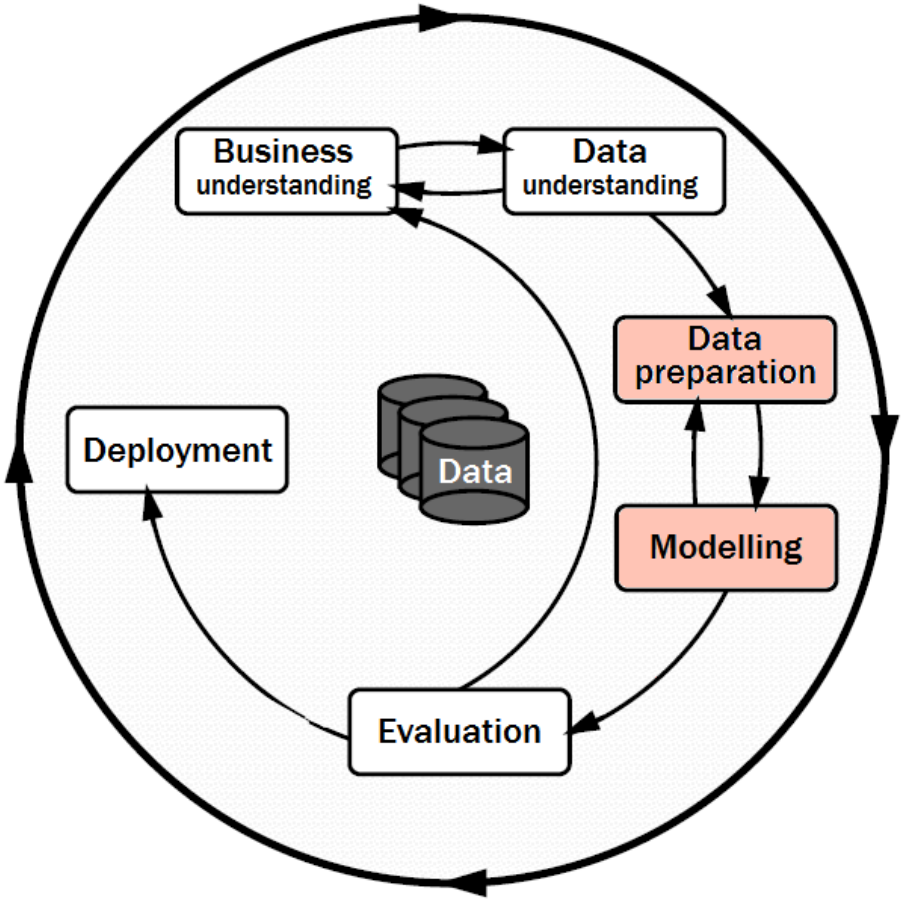


Эволюция ИИ:

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

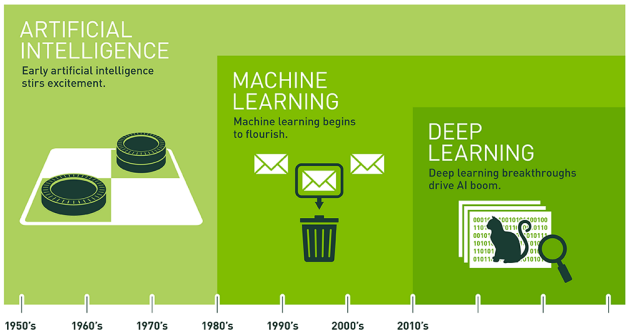
Эволюция ИИ как автоматизация шагов CRISP-DM

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



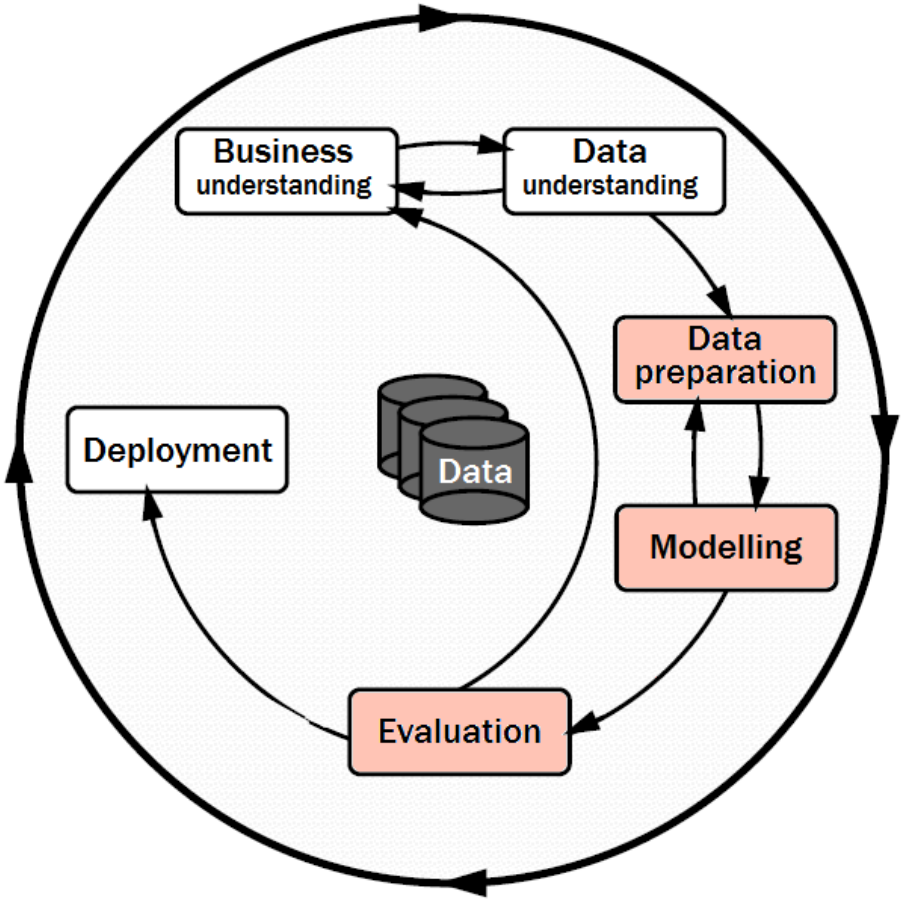
Эволюция ИИ:

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



Эволюция ИИ как автоматизация шагов CRISP-DM

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

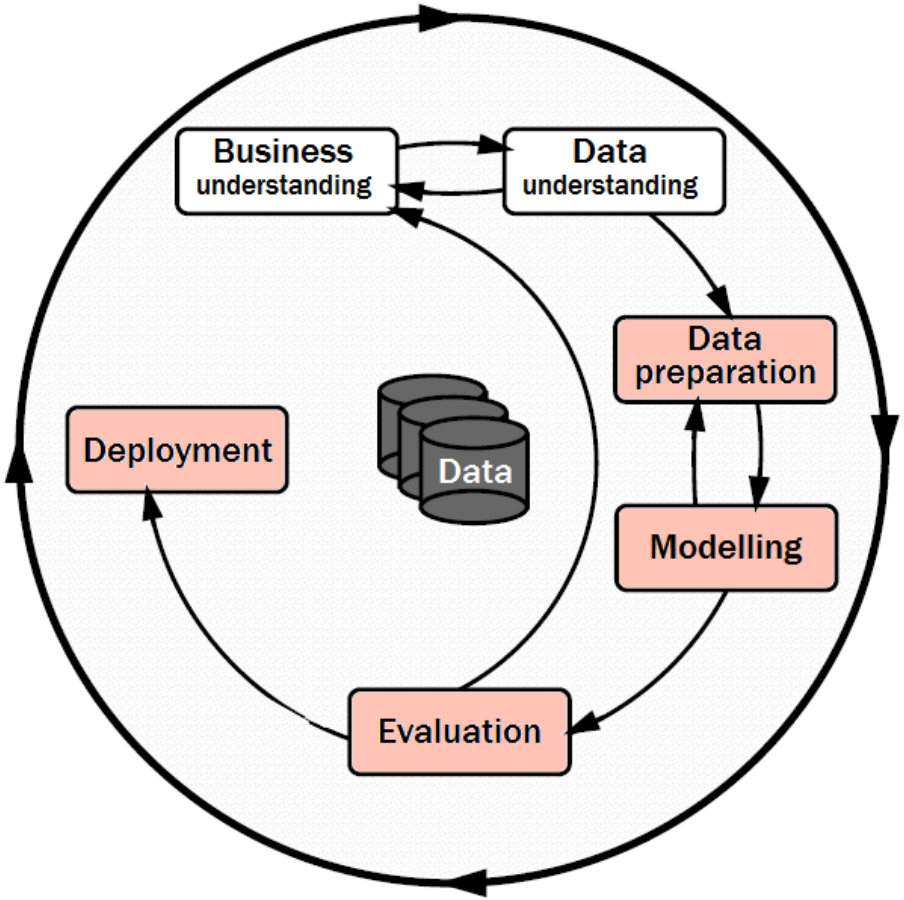


Эволюция ИИ:

- *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 структуры

Эволюция ИИ как автоматизация 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 структуры
- *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)
- понижение размерности данных (dimensionality reduction)
- отбор информативных признаков (feature selection)
- восполнение пропусков в данных (missing values imputation)
- выявление аномалий/выбросов (outlier/anomaly detection)

Граница между предобработкой и моделированием нечёткая, глубокое обучение (Deep Learning) стирает её окончательно

Методы обработки пропущенных значений

- игнорировать объекты или признаки с пропусками
  — ведёт к потере информации :(
- заполнить пропущенные значения признака f:
  — средним или медианным значением $\bar{f}$
- прогнозировать значения признака f по остальным:
  — регрессия для вещественного признака f
  — классификация для дискретного признака f
  — матричные разложения, автокодировщики
- использовать модели, способные игнорировать пропуски:
  — решающие деревья
  — голосование низкоразмерных базовых предикторов
- ввести бинарный признак $f'(x) = [f(x)$ не известно$]$

Y.Zhoua, S.Aryala, M.R.Bouadjeneka. A Comprehensive Review of Handling Missing Data: Exploring Special Missing Mechanisms. 2024

Методы выявления аномалий, выбросов, новизны

Аномальность объекта (anomaly/novelty/surprise score) — это значение функции потерь $\mathscr{L}\big(a(x_i, w), y_i\big)$ на данном объекте

Варианты оценивания аномальности:

- аномальность оценивается для объекта обучающей выборки (outlier) или для нового объекта (novelty)
- потеря зависит от y_i (supervised) или нет (unsupervised)
- при оценивании аномальности обучающего объекта он исключается из выборки $(a(x_i; X^\ell \setminus x_i))$ или нет $(a(x_i; X^\ell))$
- функция потерь та же, что в критерии обучения или другая

Варианты использования оценок аномальности:

- жёсткое удаление аномальных объектов из выборки
- мягкое перевзвешивание объектов в критерии обучения

M.Salehi et al. A unified survey on anomaly, novelty, open-set, and out-of-distribution detection: solutions and future challenges. 2021.

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 $a(x, w)$

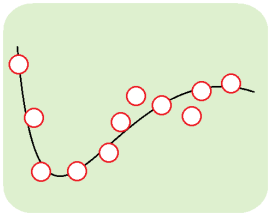
Критерий минимизации эмпирического риска $\mu\colon X^\ell \to W$ (empirical risk minimization, ERM), возможно, с регуляризацией:

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

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

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

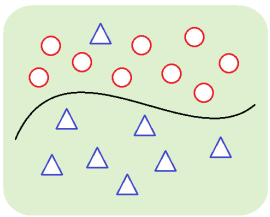
Напоминание. Задачи обучения с учителем



Регрессия: $y_i \in Y = \mathbb{R}$

$a(x, w) \in Y$

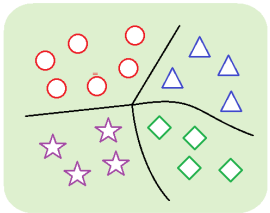
$\mathscr{L}(w, x_i) = L\big(a(x_i, w) - y_i\big)$, например: $L(\varepsilon) = \varepsilon^2$



Бинарная классификация: $y_i \in Y = \{-1, +1\}$

$a(x, w) = \operatorname{sign} g(x, w)$

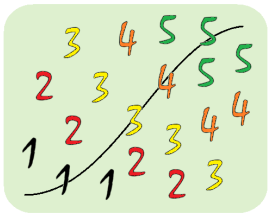
$\mathscr{L}(w, x_i) = L\big(g(x_i, w)y_i\big)$, например: $L(M) = (1-M)_+$



Многоклассовая классификация: $y_i \in Y$, $|Y| < \infty$

$a(x, w) = \arg\max\limits_{y \in Y} g_y(x, w_y), \quad w = (w_y)_{y \in Y}$

$\mathscr{L}(w, x_i) = \sum\limits_{y \neq y_i} L\big(g_{y_i}(x_i, w_{y_i}) - g_y(x_i, w_y)\big)$



Обучение ранжированию: $y_i \in \{1, \ldots, K\}$

$a(x, w) \in Y = \mathbb{R}$

$\mathscr{L}(w, x_i) = \sum\limits_{j:\, y_i < y_j} L\big(a(x_j, w) - a(x_i, w)\big)$

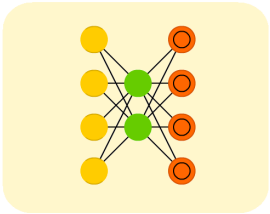
Обучение без учителя и с частичными данными от учителя

- матричные разложения (matrix factorization)
- автокодировщики (autoencoder)
- многомерное шкалирование (multidimensional scaling)
- восстановление плотности распределения (density estimation)
- восстановление смеси распределений (mixture estimation)
- кластеризация (clustering)
- поиск ассоциативных правил (association rule learning)

Частичное обучение (semi-supervised learning)

- трансдуктивное обучение (transductive learning)
- одноклассовая классификация (one-class classification)
- обучение только на положительных примерах (PU-learning)

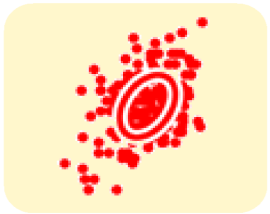
Задачи обучения без учителем



Автокодировщик:

$g(f(x_i,\alpha),\beta) = \hat{x}_i$ — реконструкция объекта

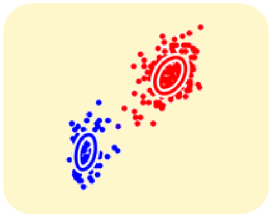
$\mathscr{L}(\alpha,\beta;x_i) = \|\hat{x}_i - x_i\|$



Восстановление плотности распределения:

$x_i \overset{\text{i.i.d.}}{\sim} p(x\,|\,\theta)$ — параметризация плотности

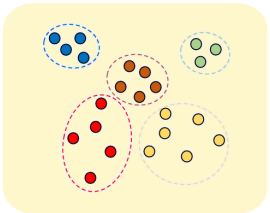
$\mathscr{L}(\theta,x_i) = -\ln p(x_i\,|\,\theta)$



Восстановление смеси плотностей:

$x_i \overset{\text{i.i.d.}}{\sim} p(x\,|\,\theta,w) = \sum_j w_j p(x\,|\,\theta_j),\quad w_j \geqslant 0,\ \sum_j w_j = 1$

$\mathscr{L}(\theta,w;x_i) = -\ln p(x_i\,|\,\theta,w)$



Кластеризация:

$a(x,w) = \arg\min\limits_{y\in Y} \|x_i - w_y\|,\quad |Y| < \infty$

$\mathscr{L}(w,x_i) = \|x_i - \mu_{a(x_i,w)}\|$

Задачи с совместным обучением нескольких моделей

Одновременное обучение:

- обучение автокодировщика (autoencoder)
- обучаемая векторизация данных (representation learning)
- многозадачное обучение (multi-task learning)
- состязательное обучение (adversarial learning, GAN)
- частичное обучение (semi-supervised learning)

Последовательное обучение:

- предобучение (pre-training)
- перенос обучения (transfer learning)
- самостоятельное обучение (self-supervised learning)
- дистилляция моделей или суррогатное моделирование
- обучение с привилегированной информацией (learning using privileged information, LUPI)

Задача частичного обучения (semi-supervised learning, SSL)

Дано:

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

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

Найти: $\{a_{k+1}, \ldots, a_\ell\} \in Y$ — метки неразмеченных объектов

Критерий SSL без модели классификации (transductive learning):

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

Найти–Критерий SSL с параметрической моделью $a(x_i, w)$:

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

Суть SSL — это общая параметризация w в двух критериях, не обязательно классификации, не обязательно кластеризации

Дистилляция моделей и 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:

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

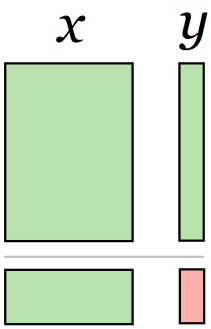
Примеры задач:

- 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

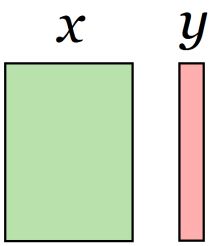
Обучение с использованием привилегированной информации

LUPI — Learning Using Privileged Information

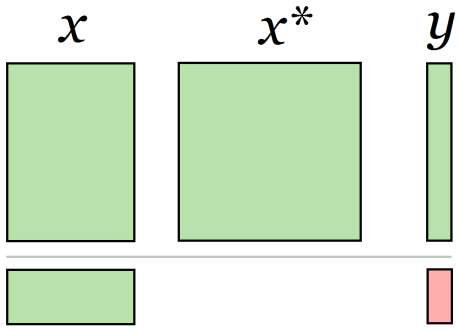
с учителем



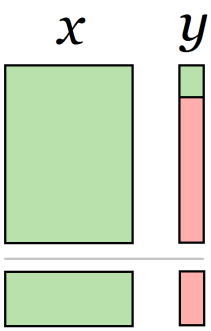
без учителя



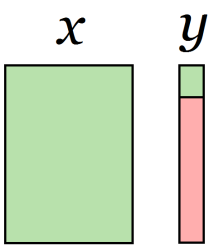
привилегированное (LUPI)



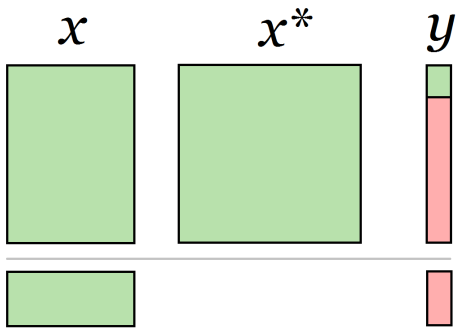
частичное



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 — данные баллистокардиографии
  x^* — данные ЭКГ (мониторирование по Холтеру)
  y — диагноз
- x — текстовый документ
  x^* — выделенные ключевые слова, фразы, фрагменты
  y — категория документа
- x — пара (запрос, документ)
  x^* — выделенные асессором ключевые слова или фразы
  y — оценка релевантности

Задача обучения с привилегированной информацией

Раздельное обучение модели-ученика a и модели-учителя a^*:

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

Модель-ученик a обучается быть не хуже модели-учителя a^*:

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

Совместное обучение модели-ученика a и модели-учителя a^*:

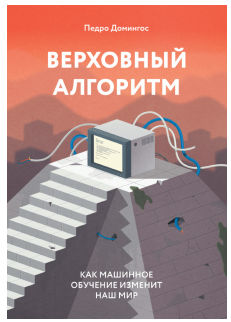
$$\sum_{i=1}^{\ell} \mathscr{L}\big(a(x_i, w), y_i\big) + \lambda \mathscr{L}\big(a^*(x_i, x_i^*, w^*), y_i\big) + \\ + \mu \mathscr{L}\big(a(x_i, w), a^*(x_i, x_i^*, w^*)\big) \to \min_{w, w^*}$$

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

Пять (шесть) школ машинного обучения по П.Домингосу

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



Педро Домингос. Верховный алгоритм. 2016. 336 с.

Анализ ошибок бинарной классификации

Задача бинарной классификации: $y_i, a(x_i) \in \{-1, +1\}$.

	модель классификации	учитель
TP, True Positive	$a(x_i) = +1$	$y_i = +1$
TN, True Negative	$a(x_i) = -1$	$y_i = -1$
FP, False Positive	$a(x_i) = +1$	$y_i = -1$
FN, False Negative	$a(x_i) = -1$	$y_i = +1$

FP: ложноположительно, ошибка I рода, «ложная тревога»
FN: ложноотрицательно, ошибка II рода, «пропуск цели»

Правильность классификации (чем больше, тем лучше):

$$\text{Accuracy} = \frac{1}{\ell}\sum_{i=1}^{\ell}\big[a(x_i) = y_i\big] = \frac{\text{TP} + \text{TN}}{\text{FP} + \text{FN} + \text{TP} + \text{TN}}$$

Недостаток: не учитывается дисбаланс численности классов, а также различие цены ошибки I и II рода.

ROC-кривая (Receiver Operating Characteristic)

Модель классификации: $a(x; w, w_0) = \text{sign}\big(g(x, w) - w_0\big)$

Кривая ROC: как меняется качество a при варьировании w_0 (чем больше w_0, тем больше x_i, на которых $a(x_i) = -1$)

- по оси X: доля *ошибочных положительных классификаций* (FPR — false positive rate):

$$\text{FPR}(a) = \frac{\text{FP}}{\text{FP} + \text{TN}} = \frac{\sum_{i=1}^{\ell}[y_i = -1][a(x_i; w, w_0) = +1]}{\sum_{i=1}^{\ell}[y_i = -1]}$$

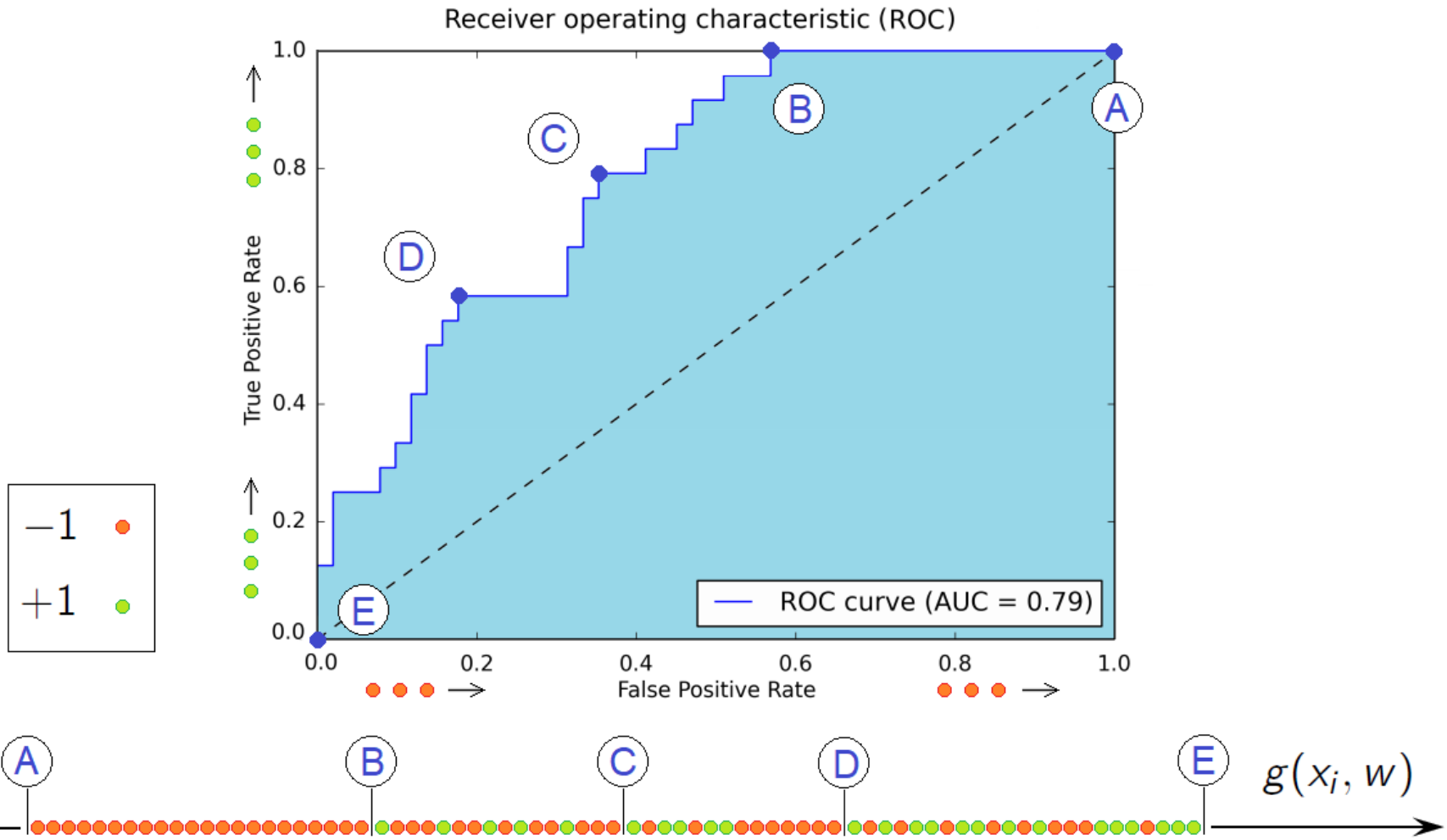
  $1 - \text{FPR}(a)$ называется *специфичностью* алгоритма a

- по оси Y: доля *правильных положительных классификаций* (TPR — true positive rate):

$$\text{TPR}(a) = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{\sum_{i=1}^{\ell}[y_i = +1][a(x_i; w, w_0) = +1]}{\sum_{i=1}^{\ell}[y_i = +1]}$$

  $\text{TPR}(a)$ называется также *чувствительностью* алгоритма a

ROC-кривая и площадь под кривой AUC (Area Under Curve)



ABCDE — положения порога w_0 на оси значений функции g

Задача максимизации AUROC — площади под ROC-кривой

Модель классификации: $a(x; w, w_0) = \text{sign}\big(g(x, w) - w_0\big)$

AUROC равна доле правильно упорядоченных пар (x_i, x_j), это критерий бинарного ранжирования с парной функцией потерь:

$$\text{AUROC}(w) = \frac{1}{\ell_-}\sum_{i=1}^{\ell}\big[y_i = -1\big]\text{TPR}_i =$$

$$= \frac{1}{\ell_-\ell_+}\sum_{i=1}^{\ell}\sum_{j=1}^{\ell}\big[y_i < y_j\big]\big[g(x_i, w) < g(x_j, w)\big] \to \max_w$$

Критерий максимума аппроксимированного AUROC:
вводится $L(M)$ — убывающая функция *парного отступа* $M_{ij}(w)$

$$1 - \text{AUROC}(w) \leqslant Q(w) = \sum_{i,j:\, y_i<y_j} L\underbrace{\big(g(x_j, w) - g(x_i, w)\big)}_{M_{ij}(w)} \to \min_w$$

SG: градиентные шаги по парам объектов (x_i, x_j): $y_i < y_j$

Точность и полнота бинарной классификации

В информационном поиске не важен TN:

$$\text{Точность, Precision} = \frac{TP}{TP+FP}$$

$$\text{Полнота, Recall} = \frac{TP}{TP+FN}$$

$$\text{F-мера} = \frac{2PR}{P+R}$$

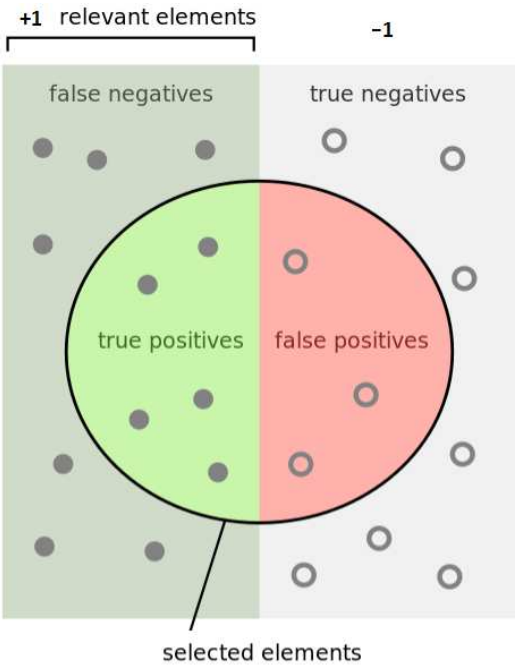
Precision — доля релевантных среди найденных
Recall — доля найденных среди релевантных

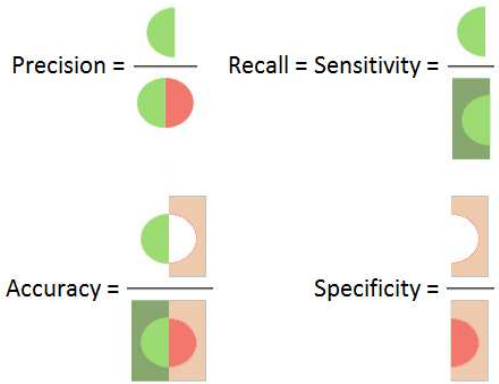
В медицинской диагностике важно всё:

$$\text{Чувствительность, Sensitivity} = \frac{TP}{TP+FN}$$

$$\text{Специфичность, Specificity} = \frac{TN}{TN+FP}$$

Sensitivity — доля верных положительных диагнозов
Specificity — доля верных отрицательных диагнозов





Точность и полнота многоклассовой классификации

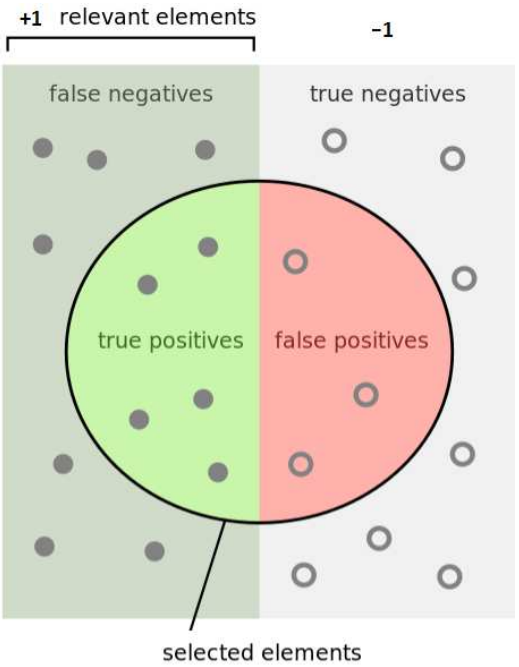
Для каждого класса $y \in Y$:
TP_y — верные положительные
FP_y — ложные положительные
FN_y — ложные отрицательные

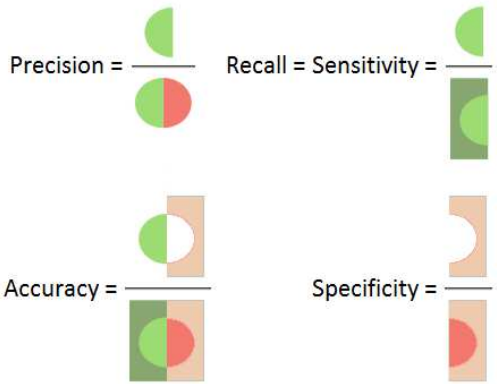
Точность и полнота с микроусреднением:

$$\text{Precision: } P = \frac{\sum_y \mathrm{TP}_y}{\sum_y (\mathrm{TP}_y + \mathrm{FP}_y)};$$

$$\text{Recall: } R = \frac{\sum_y \mathrm{TP}_y}{\sum_y (\mathrm{TP}_y + \mathrm{FN}_y)};$$

Микроусреднение не чувствительно
к ошибкам на малочисленных классах





Точность и полнота многоклассовой классификации

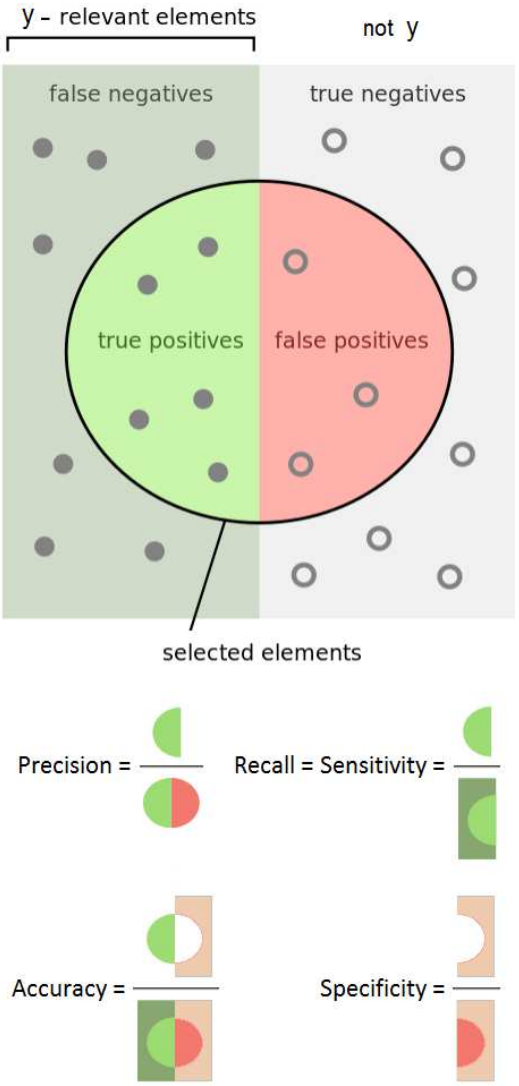
Для каждого класса $y \in Y$:
TP_y — верные положительные
FP_y — ложные положительные
FN_y — ложные отрицательные

Точность и полнота с макроусреднением:

$$\text{Precision: } P = \frac{1}{|Y|} \sum_y \frac{\mathrm{TP}_y}{\mathrm{TP}_y + \mathrm{FP}_y};$$

$$\text{Recall: } R = \frac{1}{|Y|} \sum_y \frac{\mathrm{TP}_y}{\mathrm{TP}_y + \mathrm{FN}_y};$$

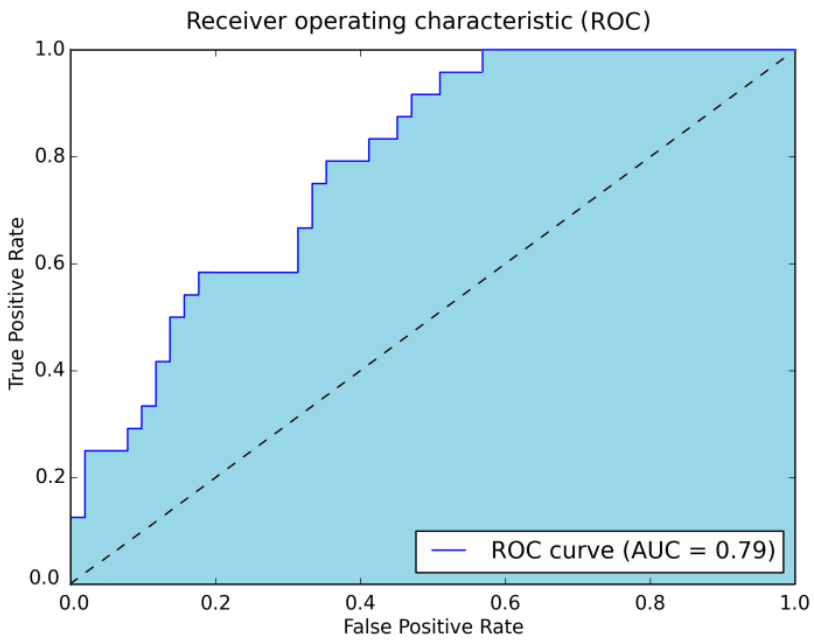
Макроусреднение чувствительно
к ошибкам на малочисленных классах

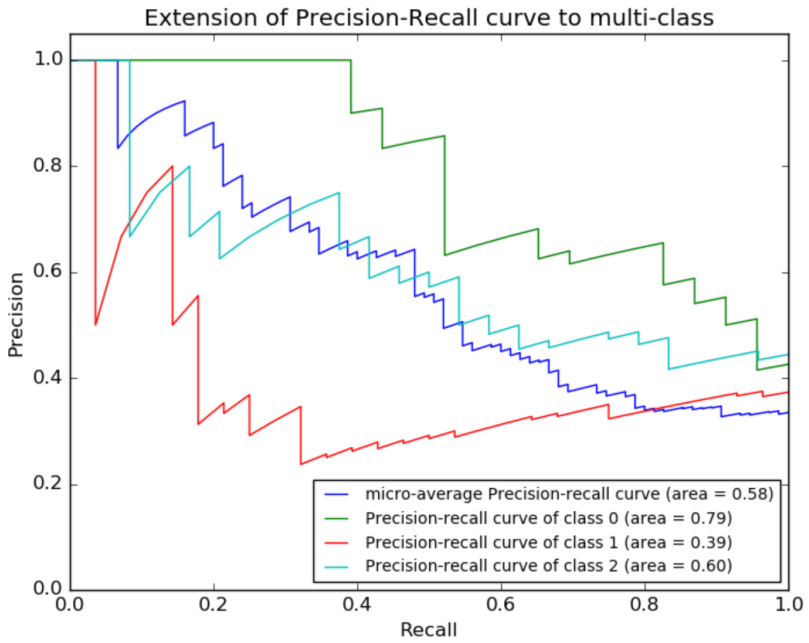


Кривые ROC и Precision-Recall

Модель классификации: $a(x) = \text{sign}\big(\langle x, w\rangle - w_0\big)$

Каждая точка кривой соответствует значению порога w_0





AUROC — площадь под ROC-кривой

AUPRC — площадь под кривой Precision-Recall

Примеры из Python scikit learn: `http://scikit-learn.org/dev`

Оценивание обобщающей способности и выбор 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

$\mu_t \colon X^\ell \to W_t$ — 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* $a(x, w)$ на X^ℓ

$\mu\colon X^\ell \to W$ — *метод обучения* модели $a(x, w)$, $w \in W$

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:

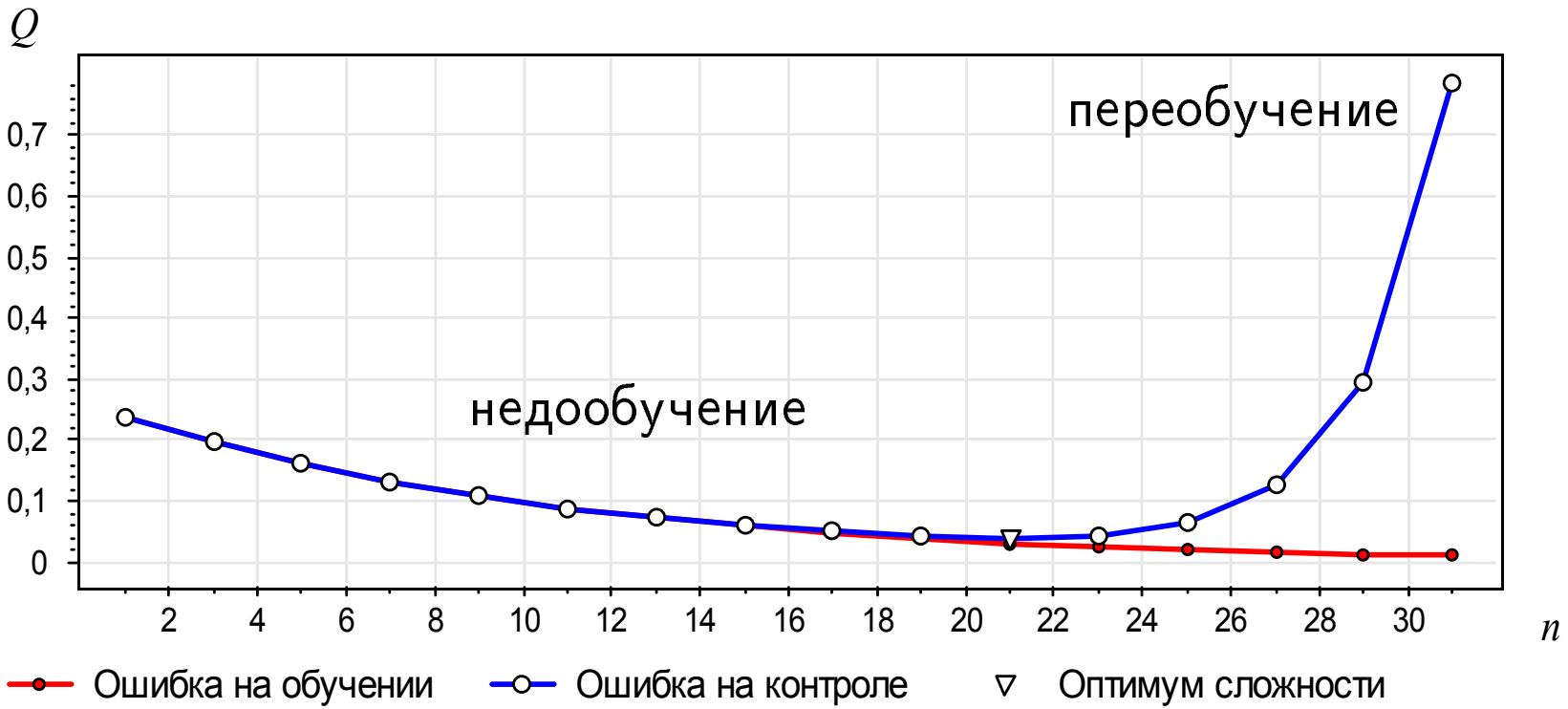
$$Q_\mu(X^\ell, X^k) = Q\big(\mu(X^\ell), X^k\big)$$

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 параметров $\dim w$ или числа признаков n).

Внешний критерий не подвержен переобучению, имеет минимум, соответствующий оптимальной сложности модели:



Кросс-проверка (cross-validation, CV)

Усреднение по множеству разбиений $X^L = X_n^\ell \sqcup X_n^k$, $n \in N$:

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

- $|N| = 1$ — единственное (случайное) разбиение: *hold-out*
- N — множество случайных разбиений: *метод Монте-Карло*
- $N = \left\{(X^L \setminus \{x_i\}) \sqcup \{x_i\}\right\}_{i=1..L}$, каждый объект становится контролем один раз, *скользящий контроль* (*leave one out*)
- $N = \left\{(X^L \setminus B_n) \sqcup B_n\right\}_{n=1..q}$, где $B_1 \sqcup \cdots \sqcup B_q = X^L$ — разбиение *на q блоков* равной± 1 длины (q-fold CV), каждый объект участвует в контроле один раз
- $N = \left\{(X^L \setminus B_n^s) \sqcup B_n^s\right\}_{n=1..q}^{s=1..t}$, где $B_1^s \sqcup \cdots \sqcup B_q^s = X^L$ — t разбиений *на q блоков* равной± 1 длины ($t \times q$-fold CV), каждый объект участвует в контроле ровно t раз
- N — все $C_{\ell+k}^k$ разбиений: *complete cross-validation*, CCV

Методология анализа ошибок (потерь)

$\mathscr{L}(w, x_i)$ — функция потерь (чем меньше, тем лучше).
Среднее потерь на выборке U и эмпирическое распределение:

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

$$F(\lambda; w, U) = \frac{1}{|U|} \sum_{x_i \in U} \big[\mathscr{L}(w, x_i) \leqslant \lambda\big]$$

Анализ потерь на обучающей выборке $U = X^\ell$:

- 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?

Сравнение потерь на обучении $U = X^\ell$ и тесте $U = X^k$:

- Сильно ли отличаются распределения потерь?
- Если сильно, то это переобучение — как его устранить?

Автоматический выбор моделей и гиперпараметров (AutoML)

Проблема: подбор структуры модели, архитектуры нейросети, значений гиперпараметров требуют слишком много ресурсов

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

Найти: условия следующего эксперимента с моделью

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

Близкая классическая задача — *планирование экспериментов*

Ф.Хуттер, Л.Коттхофф, Х.Ваншорен. Введение в автоматизированное машинное обучение (AutoML). 2023.

П.Браздил, Я.В.Рейн, К.Соарес, Х.Ваншорен. Метаобучение. Применение в AutoML и науке о данных. 2023.

Xin He et al. AutoML: A Survey of the State-of-the-Art. 2019

https://github.com/sberbank-ai-lab/LightAutoML — AutoML от Сбербанка

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

Проблема: слишком много методов, слишком долго запускать

Дано: выборка «задача, метод» $\longrightarrow$ внешние критерии

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

Критерий: отличие предсказанного метода от оптимального по выбранному для задачи внешнему критерию качества

Признаки объекта «задача»:

- 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.

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

Две модели, «базовая A» и «улучшенная B»,
построенные по историческим данным X^ℓ,
тестируются по выбранным метрикам 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, но злоупотреблять выбором из многих моделей не стоит
- накопление данных X^k может потребовать много времени — отсечку времени между A и B можно двигать в прошлое
- работа модели может формировать смещённый поток данных (например, в рекомендательных системах)

Задача онлайнового обучения с учителем

Дано: $(x_i, y_i)_{i=1}^{\ell}$ — поток объектов с ответами

Найти: $a(x, w)$ — адаптивную модель $y(x)$ с параметром w и правило обновления параметров модели $w_{i-1} \mapsto w_i$

Критерий: убывание кривой обучения (LOO learning curve)

$$Q(t) = \frac{1}{t} \sum_{i=1}^{t} \mathscr{L}\big(a(x_i, w_{i-1}), y_i\big) \to \min,$$

где $\mathscr{L}(a, y)$ — потеря модели a в сравнении с y

инициализировать параметры модели w_0;
для всех $i = 1, \ldots, \ell$
- получить объект x_i и предсказать $a_i := a(x_i, w_{i-1})$;
- получить ответ y_i и оценить потерю $\mathscr{L}_i := \mathscr{L}(a_i, y_i)$;
- обновить модель $w_i := \text{Update}\,(w_{i-1}, x_i, y_i)$;

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 не забывать ранее выученный материал?
- Либо, наоборот, как забывать самые старые объекты?

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: $Y_t = (y_0, y_1, \ldots, y_t)$ — временной ряд, $y_i \in \mathbb{R}$

Найти: $\hat{y}_{t+d}(Y_t, w)$ — модель прогноза на момент $t+d$
где w — вектор параметров модели,
$d = 1, \ldots, D$, D — горизонт прогнозирования

Критерий: минимум среднеквадратичной ошибки прогнозов:

$$\sum_{d=1}^{D} \sum_{t=T_0}^{T} \left(\hat{y}_{t+d}(Y_t, w) - y_{t+d}\right)^2 \;\to\; \min_w$$

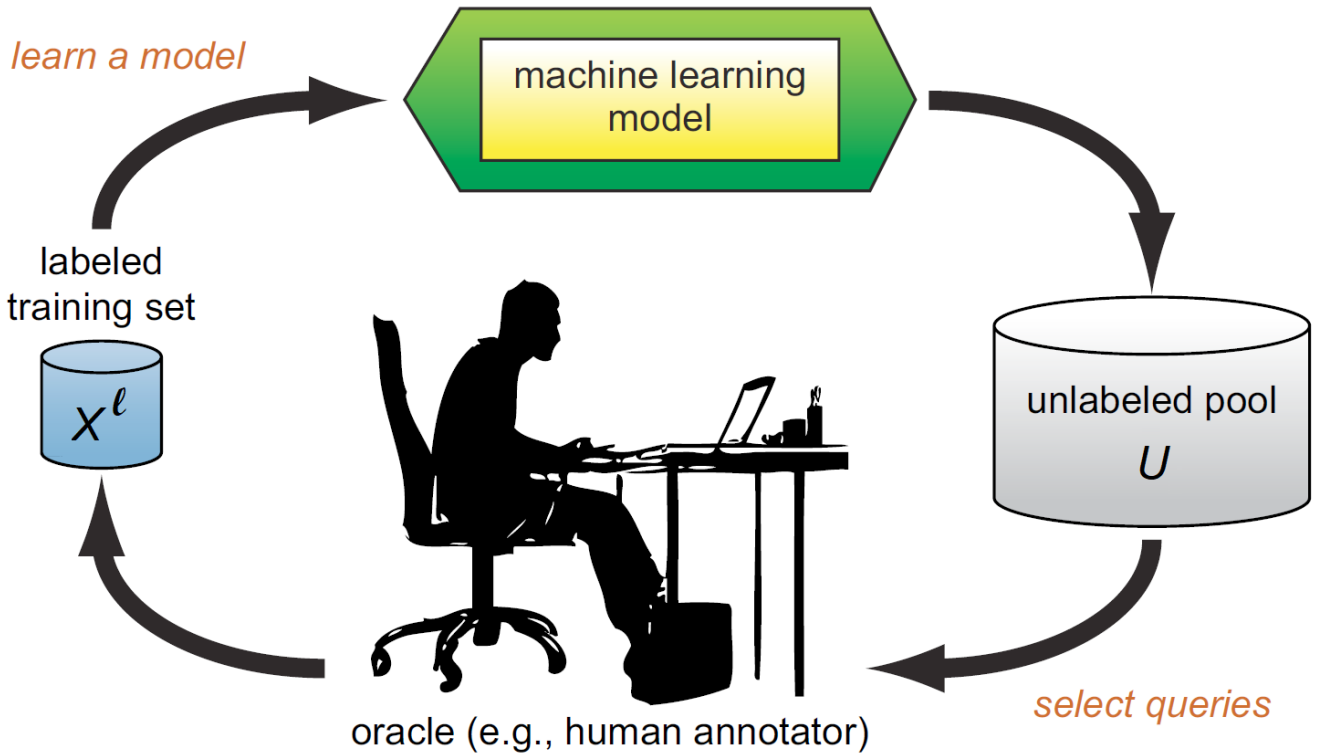
Пример: линейная модель *авторегрессии*. В роли признаков выступают непосредственно n предыдущих наблюдений ряда:

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

Обучающая выборка: $\ell = t - n + 1$ моментов истории ряда

Постановка задачи активного обучения

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



Burr Settles. Active Learning Literature Survey. 2010.

Постановка задачи активного обучения

Дано: $X^\ell = (x_i, y_i)_{i=1}^\ell$ — выборка размеченных объектов
$U = (u_i)_{i=1}^K$ — пул неразмеченных объектов

Найти: $\varphi(u)$ — модель перспективности разметки $u \in U$
$a(x, w)$ — модель с инкрементным обучением
$(u_i, y_i^*)_{i=1}^k$, $k \leqslant K$ — размеченная выборка

Критерий: достичь как можно лучшего качества модели a, разметив как можно меньше объектов из U

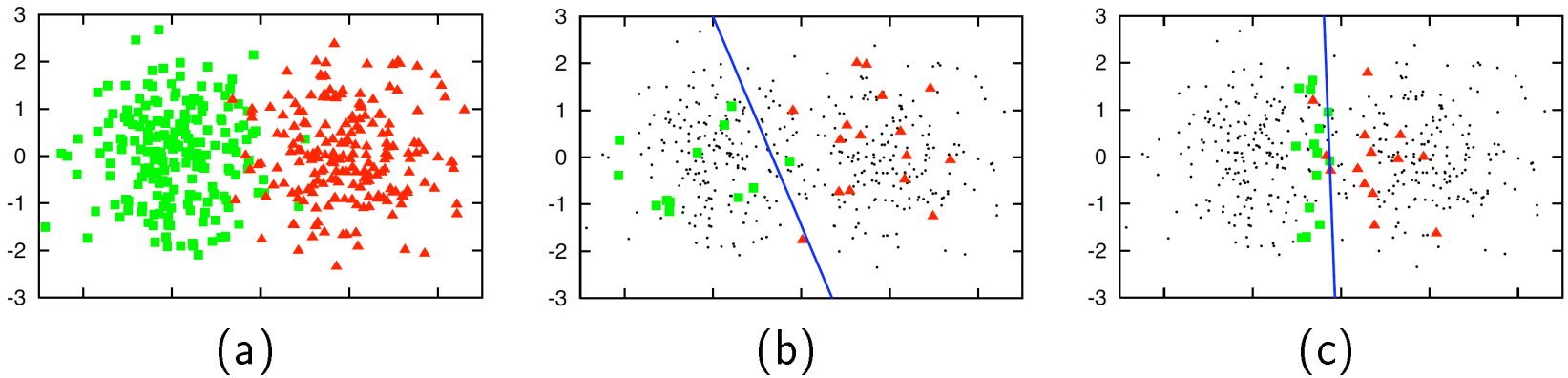
обучить модель a по начальной выборке $(x_i, y_i)_{i=1}^\ell$;
пока есть неразмеченные объекты и модель не обучилась
- $u_i = \arg\max_{u \in U} \varphi(u)$ — наиболее перспективный объект;
- $y_i^* = y(u_i)$ — запросить ответ учителя (оракула);
- дообучить модель $a(x, w)$ ещё на одном примере (u_i, y_i^*);

Burr Settles. Active Learning Literature Survey. 2010.

Почему активное обучение быстрее пассивного

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

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



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)

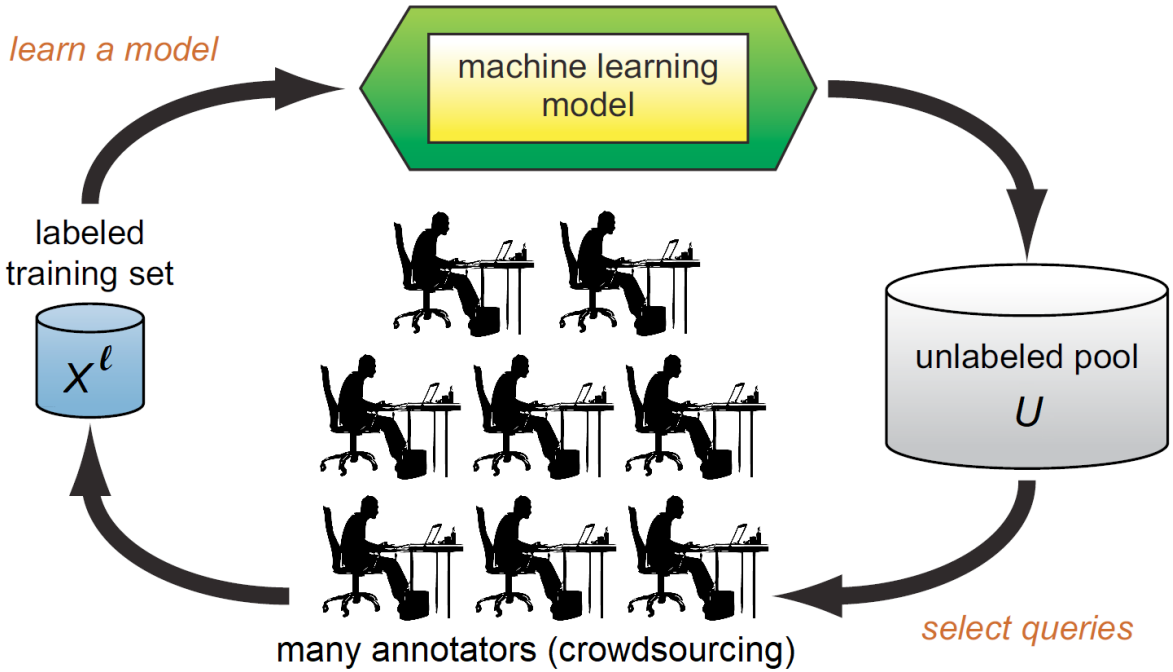
Применения в бизнесе:

- 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)

Дано: агент, действующий в среде $\langle A, p(r \mid a) \rangle$
A — конечное множество *действий* (action, arm)
$p(r \mid a)$ — неизвестное распределение *премии* за $a \in A$

Найти: $\pi_t(a)$ — *стратегию* (policy) агента в раундах t,
распределение на множестве действий A

Критерий: $\sum_{t=1}^{T} r_t \to \max$ — суммарная премия в конце игры

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

$$Q(a) = \lim_{T \to \infty} \frac{\sum_{t=1}^{T} r_t [a_t = a]}{\sum_{t=1}^{T} [a_t = a]} \to \max_{a \in A} \text{ — } \textit{ценность действия } a$$

Постановка задачи в случае, когда агент влияет на среду

Дано: агент, действующий в среде $\langle A, S, p(r \mid a, s), p(s \mid a, s) \rangle$
A — конечное множество *действий* (action, arm)
S — конечное множество состояний среды (state)
$p(r \mid a, s)$ — неизвестное распределение премий
$p(s' \mid a, s)$ — неизвестное распределение переходов

Найти: $\pi_t(a \mid s)$ — *стратегия* (policy) агента в раундах t

Критерий: $\sum_{t=1}^{T} r_t \to \max$ — суммарная премия в конце игры

Игра агента со средой: инициализация s_1, $\pi_1(a \mid s_1)$;
для всех раундов $t = 1, \ldots, T$
- 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 переходит в новое состояние $s_{t+1} \sim p(s \mid a_t, s_t)$;
- агент корректирует стратегию $\pi_{t+1}(a \mid s)$;

Отличия от обычных задач машинного обучения

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

Параметрические модели, которые можно обучать:

- функция ценности действия в состоянии $Q(s, a; w)$
  $Q^\pi(s, a) = \mathsf{E}_\pi\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k} \;\middle|\; s_t = s,\ a_t = a\right)$
- функция ценности состояния $V(s; w)$
  $V^\pi(s) = \mathsf{E}_\pi\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k} \;\middle|\; s_t = s\right)$
- стратегия $\pi_{t+1}(a \,|\, s; w)$
- предсказательная модель среды $(r_t, s_{t+1}) = \mu(s_t, a_t; w)$

Примеры прикладных задач

- Управление роботами, технологическими процессами
- Рекомендация товаров, новостей, рекламы
- Управление портфелем ценных бумаг, игра на бирже
- Управление ценами и ассортиментом в сетях продаж
- Маршрутизация в телекоммуникационных сетях
- Генерация движений персонажей в мультипликации
- Обучение LLM с обратной связью от пользователей (RLHF)
- Мультиагентные системы
- Стратегические игры: шахматы, го, Atari, Dota2, StarCraft2

Beam Rider | Breakout | Space Invaders | Seaquest | Pong | Enduro | Q*bert

H. Robbins. Some aspects of the sequential design of experiments. 1952.
V.Mnih et al. (DeepMind). Playing Atari with deep reinforcement learning. 2013

Резюме. Систематизация подходов машинного обучения

По исходным данным:

- данные табличные / сложно структурированные
- данные статичные / динамичные, потоковые, во времени
- данные малые / большие, в том числе распределённые

По постановкам задачи обучения:

- обучение с учителем / без учителя / частично с учителем
- обучение одной модели / совместно нескольких моделей
- обучение статичное / динамичное, с обратной связью

По подходам к построению моделей:

- эвристическая основа — 5(6) научных школ по Домингосу
- модель предиктивная / генеративная
- структура модели глубокая (deep) / неглубокая (shallow)

Николенко С. Машинное обучение: основы, 2025.
Николенко С., Кадурин А., Архангельская Е. Глубокое обучение: основы, 2024.
Марков С. Охота на электроовец. Большая книга искусственного интеллекта. В 2-х томах. 2024. (https://markoff.science)