

Методы машинного обучения. Краткий итоговый обзор семестра

Воронцов Константин Вячеславович
www.MachineLearning.ru/wiki?title=User:Vokov
вопросы к лектору: k.vorontsov@iai.msu.ru

материалы курса:
github.com/MSU-ML-COURSE/ML-COURSE-25-26
орг.вопросы по курсу: mlcmc@mail.ru

1 Типы задач оптимизации в машинном обучении

- оптимизация, переобучение, регуляризация
- обучение с учителем
- обучение без учителя

2 Конструирование моделей: шесть научных школ

- символизм и эволюционизм
- аналогизм и байесионизм
- коннекционизм и композиционизм

3 Практика машинного обучения

- CRISP-DM: стандарт процесса анализа данных
- предобработка и векторизация данных
- постобработка и оценивание качества

Общая постановка большинства задач машинного обучения

Дано: X — пространство объектов

$X^\ell = \{x_1, \dots, x_\ell\} \subset X$ — обучающая выборка, прецеденты

$a(x, w)$, $a: X \times W \rightarrow Y$ — параметрическая модель, гипотеза

Найти $w \in W \subseteq \mathbb{R}^N$ — вектор параметров модели $a(x, w)$

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

$$Q(w) = \frac{1}{\ell} \sum_{i=1}^{\ell} \mathcal{L}(w, x_i) + \tau \mathcal{R}(w) \rightarrow \min_w$$

$\mathcal{L}(w, x)$ — функция потерь (loss function),

тем больше, чем хуже ответ модели $a(x, w)$ на объекте x

$\mathcal{R}(w)$ — регуляризатор, не прецедентные требования к модели

τ — коэффициент регуляризации для балансировки критериев

Градиентный метод минимизации эмпирического риска

$$Q(w) = \frac{1}{\ell} \sum_{i=1}^{\ell} \mathcal{L}(w, x_i) + \tau \mathcal{R}(w) \rightarrow \min_w$$

Метод градиентного спуска (GD):

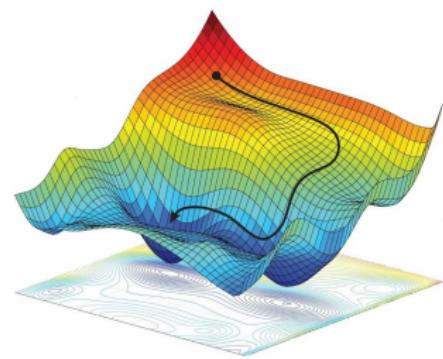
$w^{(0)} :=$ начальное приближение;

$$w^{(t+1)} := w^{(t)} - h \nabla Q(w^{(t)})$$

где $\nabla Q(w) = \left(\frac{\partial Q(w)}{\partial w_j}\right)_{j=1}^N$ — вектор градиента,

h — градиентный шаг, называемый также темпом обучения

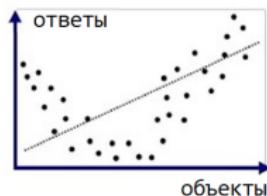
$$w^{(t+1)} := w^{(t)} - h \left(\frac{1}{\ell} \sum_{i=1}^{\ell} \nabla \mathcal{L}(w^{(t)}, x_i) + \tau \nabla \mathcal{R}(w^{(t)}) \right)$$



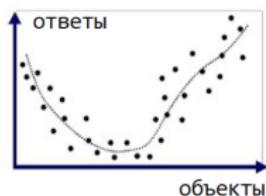
Метод стохастического градиента (Stochastic Gradient, SGD):

$$w^{(t+1)} := w^{(t)} - h \left(\nabla \mathcal{L}(w^{(t)}, x_i) + \tau \nabla \mathcal{R}(w^{(t)}) \right)$$

Фундаментальные проблемы: недообучение и переобучение

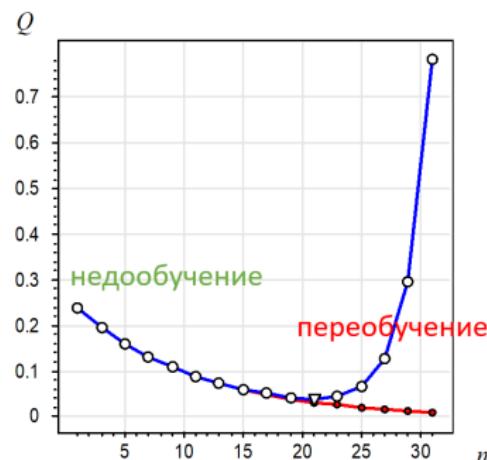


недообучение



переобучение

- **Недообучение (underfitting):**
данных много,
параметров недостаточно,
модель простая, негибкая
- **Переобучение (overfitting):**
параметров много, данных
недостаточно, модель
сложная, избыточно гибкая



Мультиколлинеарность и переобучение в линейных моделях

Мультиколлинеарность — линейная зависимость признаков:

- линейная модель регрессии/классификации $\langle w, x \rangle$
- мультиколлинеарность: $\exists u \in \mathbb{R}^n: \forall x \in X \quad \langle u, x \rangle = 0$
- неединственность решения: $\forall \gamma \in \mathbb{R} \quad \langle w, x \rangle = \langle w + \gamma u, x \rangle$

Мультиколлинеарность приводит к переобучению:

- неустойчивость: слишком большие веса $|w_j|$ разных знаков
- неинтерпретируемость веса w_j как важности признака f_j
- переобучение: $Q(w^*, X^\ell) \ll Q(w^*, X^k)$

Как уменьшить переобучение:

- регуляризация $\|w\| \rightarrow \min$ (сокращение весов, weight decay)
- отбор признаков: $f_1, \dots, f_n \rightarrow f_{j_1}, \dots, f_{j_m}, \quad m \ll n$
- преобразование признаков: $f_1, \dots, f_n \rightarrow g_1, \dots, g_m, \quad m \ll n$

Штраф за сложность модели — снижает переобучение

Регуляризатор — аддитивная добавка к основному критерию, где τ — коэффициент регуляризации, управляющий параметр:

$$\sum_{i=1}^{\ell} \mathcal{L}(w, x_i) + \tau \text{штраф}(w) \rightarrow \min_w$$

L_p -регуляризация линейных моделей $\langle w, x_i \rangle = \sum_{j=1}^n w_j f_j(x_i)$:

$$\text{штраф}(w) = \|w\|_p^p = \sum_{j=1}^n |w_j|^p$$

L_2 (Ridge, SVM) стабилизирует w при мультиколлинеарности
 L_1 (LASSO) и L_0 приводят к отбору информативных признаков

Вероятностная регуляризация (Maximum a Posteriori Probability):

$$-\sum_{i=1}^{\ell} \ln p(x_i|w) - \underbrace{\ln p(w, \gamma)}_{\text{регуляризатор}} \rightarrow \min_w$$

Негладкие регуляризаторы для отбора и группировки признаков

Общий вид регуляризаторов (μ — параметр селективности):

$$\sum_{i=1}^{\ell} \mathcal{L}(w, x_i) + \sum_{j=1}^n R_\mu(w_j) \rightarrow \min_w.$$

Регуляризаторы с эффектами отбора и группировки признаков:

LASSO (L_1): $R_\mu(w) = \mu|w|$

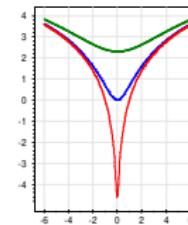
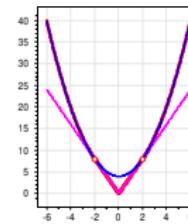
Elastic Net: $R_\mu(w) = \mu|w| + \tau w^2$

Support Feature Machine (SFM):

$$R_\mu(w) = \begin{cases} 2\mu|w|, & |w| \leq \mu; \\ \mu^2 + w^2, & |w| \geq \mu; \end{cases}$$

Relevance Feature Machine (RFM):

$$R_\mu(w) = \ln(\mu w^2 + 1)$$



Обучение модели регрессии

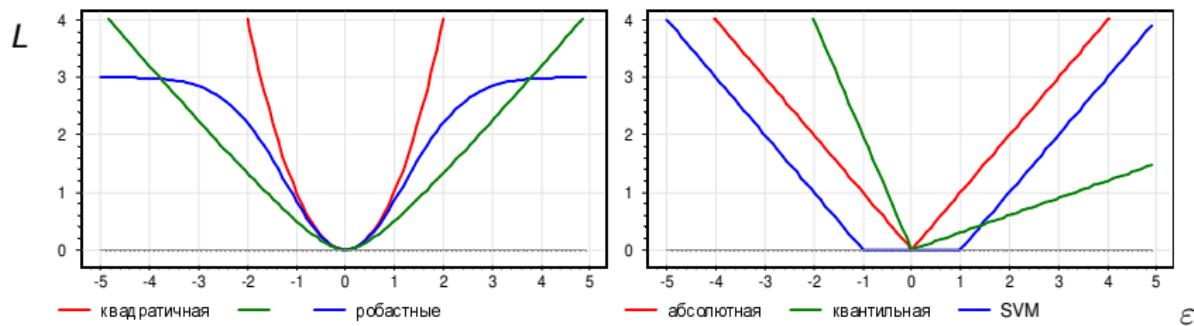
Дано: обучающая выборка $(x_i, y_i)_{i=1}^{\ell}$ с ответами $y_i \in \mathbb{R}$

Найти: вектор параметров w модели регрессии $a(x, w)$

Критерий: минимум эмпирического риска

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

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



Обучение бинарного классификатора

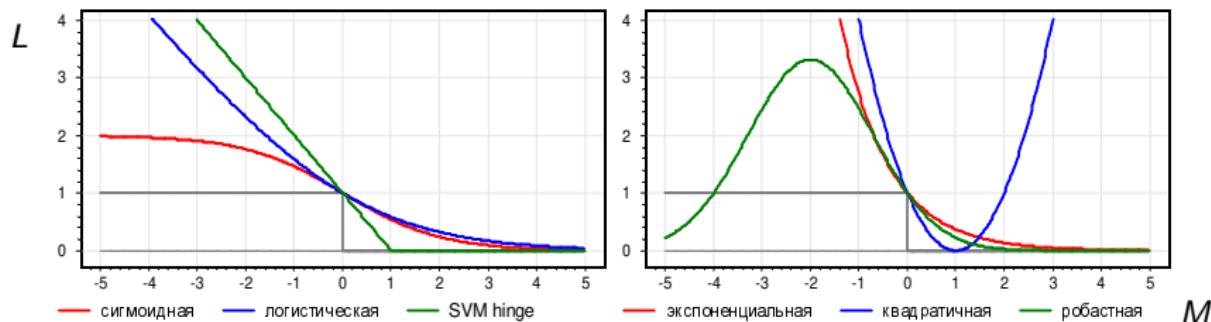
Дано: обучающая выборка $(x_i, y_i)_{i=1}^{\ell}$, $y_i \in \{-1, +1\}$

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

Критерий: \min аппроксимированного эмпирического риска

$$\sum_{i=1}^{\ell} [g(x_i, w)y_i < 0] \leq \sum_{i=1}^{\ell} L(g(x_i, w)y_i) \rightarrow \min_w$$

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



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»:

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

Критерий «каждый против всех»:

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

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

Обучение ранжированию, максимизация AUC-ROC

Дано: обучающая выборка (x_1, \dots, x_ℓ) ,

$i \prec j$ — отношение « x_j лучше, чем x_i » между объектами из X^ℓ

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

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

Критерий: число неверно ранжированных пар объектов

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

$$\sum_{i \prec j} L(a(x_j, w) - a(x_i, w)) \rightarrow \min_w$$

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

Задача восстановления плотности распределения

Дано: $X^\ell = \{x_1, \dots, x_\ell\} \stackrel{\text{i.i.d.}}{\sim} p(x)$ — обучающая выборка

Найти: вектор параметров θ в порождающей модели $p(x|\theta)$

Критерий: максимум правдоподобия (maximum likelihood, MLE)

$$\ln p(X^\ell | \theta) = \ln \prod_{i=1}^{\ell} p(x_i | \theta) = \sum_{i=1}^{\ell} \ln p(x_i | \theta) \rightarrow \max_{\theta}$$

или максимум апостериорной вероятности — совместного правдоподобия данных и модели с априорной плотностью $p(\theta|\gamma)$ (maximum a posteriori probability, MAP):

$$\ln p(X^\ell, \theta) = \ln p(X^\ell | \theta) p(\theta | \gamma) = \sum_{i=1}^{\ell} \ln p(x_i | \theta) + \underbrace{\ln p(\theta | \gamma)}_{\text{регуляризатор}} \rightarrow \max_{\theta}$$

где γ — вектор гиперпараметров априорного распределения

Метод главных компонент (Principal Component Analysis, PCA)

Дано: выборка объектов $\{x_1, \dots, x_\ell\}$;
 $f_1(x), \dots, f_n(x)$ — числовые признаки объектов

Найти преобразование признаков (Feature Transformation):
 $g_1(x), \dots, g_m(x)$ — новые числовые признаки, $m \leq n$, и
линейную реконструкцию старых признаков $f_j(x)$ по новым:

$$\hat{f}_j(x) = \sum_{t=1}^m g_t(x) u_{jt}, \quad j = 1, \dots, n, \quad \forall x \in X,$$

Критерий: точность реконструкции f_j на обучающей выборке:

$$Q = \sum_{i=1}^{\ell} \sum_{j=1}^n (\hat{f}_j(x_i) - f_j(x_i))^2 = \|GU^\top - F\|^2 \rightarrow \min_{G, U}$$

где $G = (g_t(x_i))_{\ell \times m}$, $U = (u_{jt})_{n \times m}$

Спойлер. Автокодировщик — обучаемая векторизация

Дано: $X^\ell = \{x_1, \dots, x_\ell\}$ — обучающая выборка

Найти модель векторизации, сохраняющую информацию:

$f: X \rightarrow Z$ — кодировщик (encoder), кодовый вектор $z = f(x, \alpha)$

$g: Z \rightarrow X$ — декодировщик (decoder), реконструкция $\hat{x} = g(z, \beta)$

Критерий качества реконструкции объектов $g(f(x_i)) = \hat{x}_i \approx x_i$:

$$\sum_{i=1}^{\ell} \mathcal{L}(g(f(x_i, \alpha), \beta), x_i) \rightarrow \min_{\alpha, \beta}$$

Функция потерь может быть квадратичной: $\mathcal{L}(\hat{x}, x) = \|\hat{x} - x\|^2$

Пример. РСА это линейный автокодировщик: $x \in \mathbb{R}^n$, $z \in \mathbb{R}^m$

$$f(x, A) = {}_{m \times n}^A x, \quad g(z, B) = {}_{n \times m}^B z$$

При $m \ll n$ происходит сжатие данных об объектах

Спойлер. Автокодировщик — обучаемая векторизация для SSL

Данные: размеченные $(x_i, y_i)_{i=1}^k$, неразмеченные $(x_i)_{i=k+1}^\ell$

Найти:

$z_i = f(x_i, \alpha)$ — кодировщик

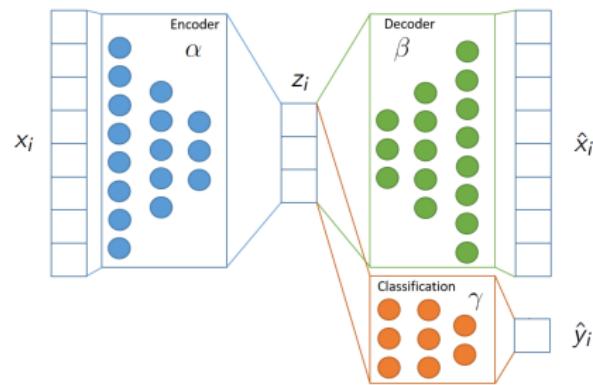
$\hat{x}_i = g(z_i, \beta)$ — декодировщик

$\hat{y}_i = \hat{y}(z_i, \gamma)$ — предиктор

Задаются функции потерь:

$\mathcal{L}(\hat{x}_i, x_i)$ — реконструкция

$\tilde{\mathcal{L}}(\hat{y}_i, y_i)$ — предсказание



Критерий: совместное обучение автокодировщика и предсказательной модели (классификации, регрессии или др.):

$$\sum_{i=1}^{\ell} \mathcal{L}(g(f(x_i, \alpha), \beta), x_i) + \lambda \sum_{i=1}^k \tilde{\mathcal{L}}(\hat{y}(f(x_i, \alpha), \gamma), y_i) \rightarrow \min_{\alpha, \beta, \gamma}$$

Основные научные школы машинного обучения

- ① **символизм** — поиск логических закономерностей
 - Decision Tree, Rule Induction
- ② **эволюционизм** — саморазвитие сложных моделей
 - Genetic Algorithms, Genetic Programming, Symbolic Regression
- ③ **аналогизм** — «у близких объектов близкие ответы»
 - kNN, RBF, SVM, Kernel Smoothing
- ④ **байесионизм** — оценивание распределений параметров
 - Naive Bayes, Bayesian Networks, Graphical Models
- ⑤ **коннекционизм** — обучение нейронных сетей
 - BackPropagation, Deep Belief Nets, Deep Learning
- ⊕ **композиционизм** — коопeração моделей
 - Weighted Voting, Boosting, Bagging, Stacking, Random Forest, Яндекс.CatBoost

Педро Домингос. Верховный алгоритм. 2016. 336 с.



Символизм. Научная школа М. М. Бонгарда

- 1958: Программа «Открой закон» восстанавливалась зависимость полным комбинаторным перебором формул
- 1959: Программа «Арифметика» для сокращения перебора использовала оценки информативности
- 1961: Программа «КоРа» перебирала информативные тройки признаков



Михаил Моисеевич
Бонгард
(1924–1971)

«КоРа-3»: первое применение распознавания незрительных образов для распознавания границы нефть-вода в скважине.

Введены принципы голосования, скользящего контроля, понятия информативности и предрассудка (переобучения).

Бонгард М. М., Вайнцвайг М. Н., Губерман Ш. А. Извекова М. Л., Смирнов М. С. Использование обучающейся программы для выявления нефтеносных пластов. 1966.

Понятие информативной логической закономерности

Модель классификации — взвешенное голосование правил:

$$a(x, w) = \arg \max_{y \in Y} \sum_{k=1}^{n_y} w_{yk} R_{yk}(x)$$

Правило — конъюнкция элементарных пороговых условий:

$$R(x) = \bigwedge_{j \in \omega} [a_j \leq f_j(x) \leq b_j]$$

Синдром — выполнены не менее d условий из множества ω ,

$$R(x) = \left[\sum_{j \in \omega} [a_j \leq f_j(x) \leq b_j] \geq d \right]$$

Правило R является закономерностью класса $y \in Y$, если

$$\begin{cases} p_y(R) = \#\{x_i : R(x_i) = 1 \text{ и } y_i = y\} \rightarrow \max \\ n_y(R) = \#\{x_i : R(x_i) = 1 \text{ и } y_i \neq y\} \rightarrow \min \end{cases}$$

Эволюционизм. Алгоритмы перебора структур модели

Дискретная оптимизация на основе дарвиновской эволюции:

- индивид — структурная формула модели
- популяция — множество различных формул
- приспособленность — внешний критерий, оценка модели
- естественный отбор лучших моделей в популяции
- скрещивание и мутация — операции порождения потомков

Частные задачи ML, решаемые эволюционными алгоритмами:

- отбор признаков (Feature Selection)
- поиск информативных логических закономерностей
- поиск архитектуры глубокой нейронной сети
- символическая регрессия (Symbolic Regression) — перебор структурных формул, генетическое программирование

Эволюционизм. Научная школа А. Г. Ивáхненко

Метод группового учёта аргументов (МГУА)
основан на самоорганизации моделей
— переборной оптимизации структуры модели

- отбор признаков или структуры модели
- качество моделей оценивается в процессе перебора по многим *внешним критериям*:
 - скользящий контроль
 - помехоустойчивость моделирования
 - баланс / согласованность прогнозов и др.
- первая 8-слойная глубокая нейросеть (1965)
- сотни применений, около 300 диссертаций



Алексей
Григорьевич
Ивáхненко
(1913–2007)

-
- Ивахненко А. Г., Лапа В. Г. Кибернетические предсказывающие устройства. 1965.
Ивахненко А. Г., Зайченко Ю. П., Димитров В. Д. Принятие решений на основе
самоорганизации. 1976.
Ивахненко А. Г. Индуктивный метод самоорганизации сложных систем. 1982.

Аналогизм. Научная школа М. А. Айзера

- Гипотеза компактности: схожие объекты, как правило, находятся в одном классе
- Метод потенциальных функций: идея заимствуется из физики
- Линейная модель классификации: взвешенное голосование функций сходства $f_i(x) = K(x, x_i)$ между x и x_i :

$$a(x) = \arg \max_{y \in Y} \sum_{i: y_i=y} \alpha_{yi} K(x, x_i)$$



Марк Аронович
Айзerman
(1913–1992)

Айзерман М. А., Браверман Э. М., Розеноэр Л. И. Теоретические основы метода потенциальных функций в задаче об обучении автоматов разделению входных ситуаций на классы. 1964.

Айзерман М. А., Браверман Э. М., Розеноэр Л. И. Метод потенциальных функций в теории обучения машин. 1970.

Аркадьев А. Г., Браверман Э. М. Обучение машин распознаванию образов. 1964.

Аналогизм. Метрические (непараметрические) методы

Восстановление плотности. Метод Парзена–Розенблатта:

$$\hat{p}_h(x; X^\ell) = \frac{1}{\ell V(h)} \sum_{i=1}^{\ell} K\left(\frac{\rho(x, x_i)}{h}\right)$$

Классификация. Потенциальные функции, окно Парзена:

$$a_h(x; X^\ell, Y^\ell) = \arg \max_{y \in Y} \sum_{i=1}^{\ell} [y_i = y] K\left(\frac{\rho(x, x_i)}{h}\right)$$

Регрессия. Ядерное сглаживание Надара–Ватсона:

$$a_h(x; X^\ell, Y^\ell) = \frac{\sum_{i=1}^{\ell} y_i K\left(\frac{\rho(x, x_i)}{h}\right)}{\sum_{i=1}^{\ell} K\left(\frac{\rho(x, x_i)}{h}\right)}$$

Принципы обучаемости, ограничения сложности, разделимости

Семейство классификаторов A обучаемо:

$$P\left\{\sup_{a \in A} |P(a) - \nu(a, X^\ell)| > \varepsilon\right\} \leq \eta,$$

$P(a)$ — вероятность ошибки классификатора,
 $\nu(a, X^\ell)$ — эмпирический риск — частота ошибок классификатора a на выборке.



Владимир
Наумович Вапник



Алексей Яковлевич
Червоненкис
(1938–2014)

Основные результаты VC-теории:

- Обосновано ограничение сложности A
- Понятие ёмкости семейства, VCdim
- Метод структурной минимизации риска
- Метод опорных векторов, SVM

Вапник В. Н., Червоненкис А. Я.
Теория распознавания образов. М.: Наука, 1974.

Дискриминативные и генеративные модели классификации

Дано: простая выборка $X^\ell = (x_i, y_i)_{i=1}^\ell \sim p(x, y)$

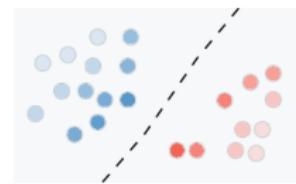
Дискриминативный подход (discriminative):

Найти модель $p(x, y) = P(y|x; w_y)p(x)$

(примеры: LR, GLM, SVM, RBF)

Критерий максимума правдоподобия:

$$\sum_{i=1}^{\ell} \ln p(x_i, y_i) = \sum_{i=1}^{\ell} \ln P(y_i|x_i; w_y) + \text{const} \rightarrow \max_{\{w_y\}}$$



Генеративный подход (generative):

Найти модель $p(x, y) = P(y)p(x|y; w_y)$

(примеры: NB, PW, FLD, RBF)

Критерий максимума правдоподобия:

$$\sum_{i=1}^{\ell} \ln p(x_i, y_i) = \sum_{i=1}^{\ell} \ln P(y_i) + \sum_{y \in Y} \sum_{x_i \in X_y} \ln p(x_i|y; w_y) \rightarrow \max_{\{P(y), w_y\}}$$



Байесовское обучение, MAP и регуляризация

Байесовский вывод апостериорного распределения $p(w|X)$ и точечная оценка максимума правдоподобия:

$$\text{Posterior}(w|X, \gamma) = \frac{p(X|w) \text{Prior}(w|\gamma)}{\int p(X|w) \text{Prior}(w|\gamma) dw}$$

$$\text{Posterior}(w|X, \gamma) \rightarrow \max_w \text{ или } \max_{w, \gamma}$$

Максимизация апостериорной вероятности (MAP) даёт точечную оценку w напрямую, без вывода Posterior:

$$\ln p(X|w) + \ln \text{Prior}(w|\gamma) \rightarrow \max_w \text{ или } \max_{\gamma} \max_w$$

Регуляризация, не обязательно вероятностная:

$$\ln p(X|w) + \gamma R(w) \rightarrow \max_w \text{ или } \max_{\gamma} \max_w$$

Коннекционизм. Линейная модель нейрона

Линейная модель нейрона (1943):

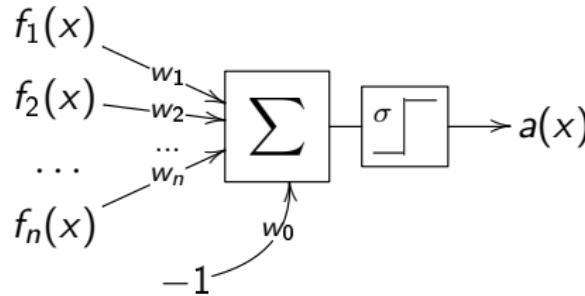
$$a(x, w) = \sigma \left(\sum_{j=1}^n w_j f_j(x) - w_0 \right)$$

$f_i(x)$ — признаки объекта x

w_i — веса признаков

w_0 — порог активации

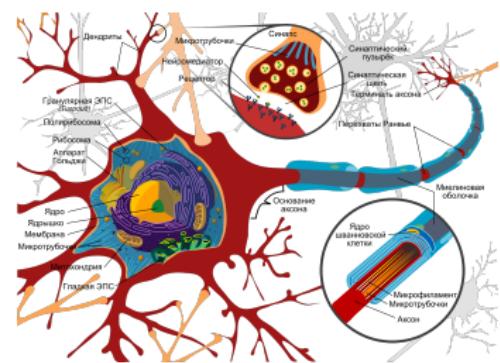
$\sigma(z)$ — функция активации



Уоррен
МакКаллок
(1898–1969)



Вальтер
Питтс
(1923–1969)



Персептрон Розенблатта (1957)

Mark-1 — первый нейрокомпьютер (1960)
для распознавания цифр и фигур
Обучение — метод коррекции ошибки
Архитектура — двухслойная сеть

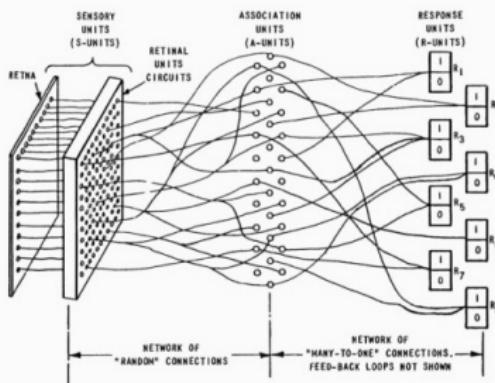


Figure 1 ORGANIZATION OF THE MARK I PERCEPTRON

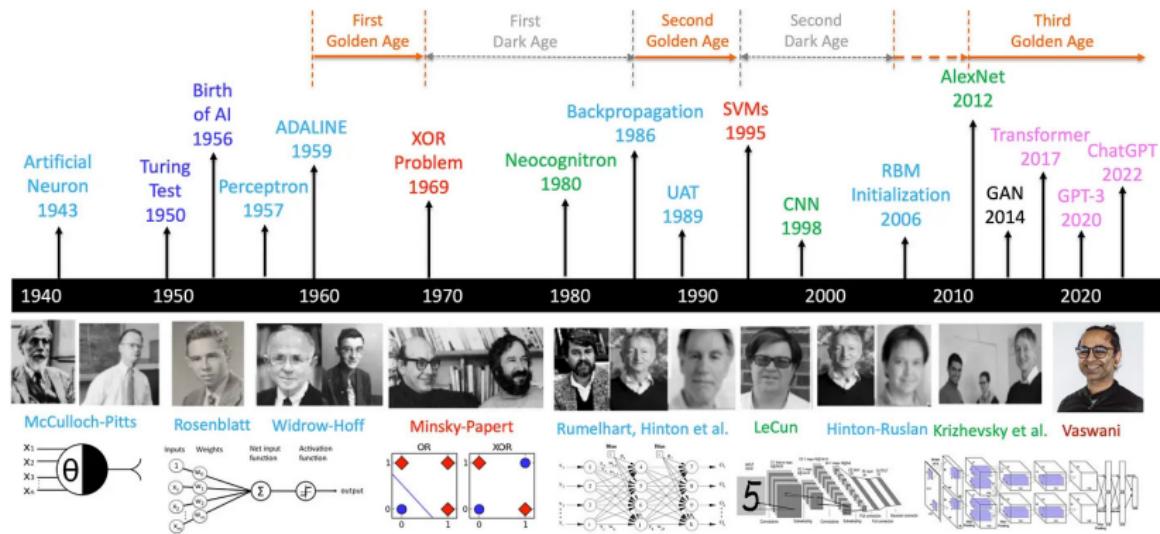


Фрэнк Розенблatt
(1928–1971)



Розенблатт Ф. Принципы нейродинамики. Перцептроны и теория механизмов мозга. 1965 (1962)

Основные вехи развития нейронных сетей (AI winters)



Минский М., Пайперт С. Персептроны. 1971 (1969)

Галушкин А. И. Синтез многослойных систем распознавания образов. 1974

Ивáхненко А. Г., Лапа В. Г. Кибернетические предсказывающие устройства. 1965

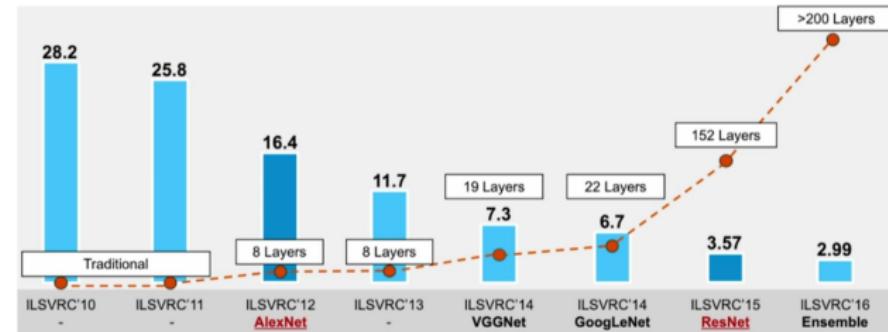
Rummelhart D. et al. Learning internal representations by error propagation. 1986

Krizhevsky A. et al. ImageNet classification with deep convolutional neural networks. 2012

Vaswani A. et al. Attention is all you need. 2017

Глубокие свёрточные сети для классификации изображений

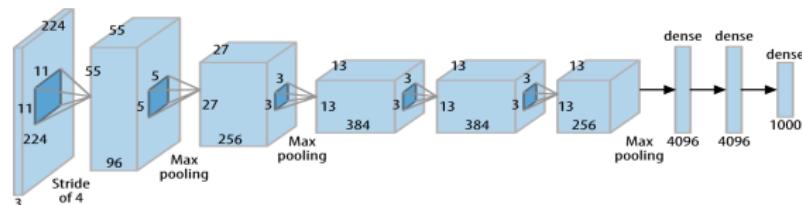
IMAGENET



Старт в 2009

Человеческий уровень ошибок 5% пройден в 2015

Свёрточные
нейронные сети
AlexNet (2012)
ResNet (2015)



Li Fei-Fei et al. ImageNet: A large-scale hierarchical image database. 2009

Krizhevsky A. et al. ImageNet classification with deep convolutional neural networks. 2012

Kaiming He et al. Deep residual learning for image recognition. 2015

Развитие ML/DL, этап 1: вектор → скаляр

Предсказательное моделирование векторных данных

Вход: векторные признаковые описания объектов

Выход: скалярные ответы (предсказания, прогнозы)



Приложения: медицинская диагностика,
геологическое прогнозирование, кредитный scoring,...

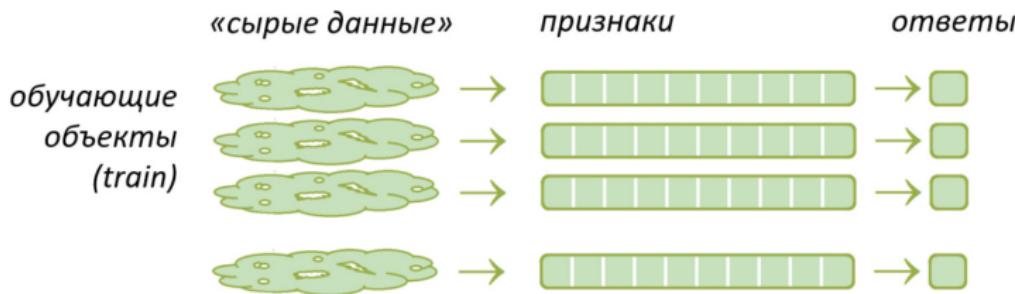
Модели: SVM, LR, MVR, RBF, MLP, ID3, CART, RF, GBM,...

Развитие ML/DL, этап 2: структура → вектор → скаляр

Обучаемая векторизация сложно структурированных данных

Вход: сложно структурированные «сырые» данные объектов

Выход: векторные представления объектов, затем ответы



Приложения: классификация изображений, текстов, сигналов, голосовых команд, биометрическая идентификация личности,...

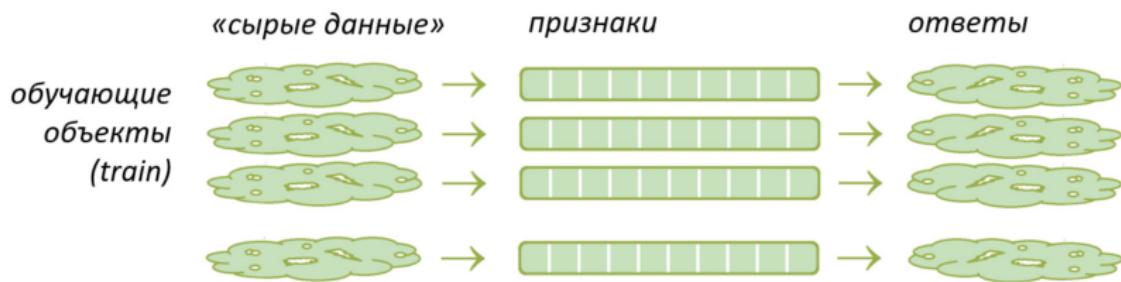
Модели: CNN, AlexNet, ResNet, word2vec, FastText, BERT,...

Развитие ML/DL, этап 3: структура → вектор → структура

Обучаемая генерация сложно структурированных данных

Вход: сложно структурированные объекты

Выход: сложно структурированные ответы



Приложения: аннотирование и синтез изображений, перенос стиля, распознавание речи, машинный перевод, суммаризация текстов, чат-боты с эмерджентными способностями,...

Модели: seq2seq, RNN, LSTM, GAN, VAE, GPT,...

Задача обучения ансамбля (композиции) моделей

Дано: $X^\ell = (x_i, y_i)_{i=1}^\ell \subset X \times Y$ — обучающая выборка
 $a_t(x, w) = C(b_t(x, w))$ — «слабые» обучаемые базовые модели
 $b_t: X \rightarrow R$ — алгоритмические операторы с параметрами w
 $C: R \rightarrow Y$ — решающее правило простого вида (без параметров)
 R — удобное пространство оценок

Найти: ансамбль $a(x) = C(F(b_1(x, w_1), \dots, b_T(x, w_T), x, \alpha))$
 $F: R^T \times X \rightarrow R$ — корректирующая функция с параметрами α

Критерий обучения «сильного» алгоритма как ансамбля из T по-отдельности «слабых» базовых алгоритмов:

$$Q(a, X^\ell) = \sum_{i=1}^{\ell} \mathcal{L}(a(x_i), y_i) \rightarrow \min_{w_1, \dots, w_T, \alpha}$$

Ю.И.Журавлëв. Об алгебраическом подходе к решению задач распознавания или классификации. Проблемы кибернетики, 1978.

M.Kearns, L.G.Valiant. Cryptographic limitations on learning Boolean formulae and finite automata. 1989.

Примеры корректирующих (агрегирующих) функций

- простое голосование (Simple Voting):

$$F(b_1, \dots, b_T) = \frac{1}{T} \sum_{t=1}^T b_t$$

- взвешенное голосование (Weighted Voting):

$$F(b_1, \dots, b_T, \alpha) = \sum_{t=1}^T \alpha_t b_t, \quad \sum_{t=1}^T \alpha_t = 1, \quad \alpha_t \geq 0$$

- взвешенный стэкинг (Feature-Weighted Linear Stacking):

$$F(b_1, \dots, b_T, x, v) = \sum_{t=1}^T \alpha_t(x) b_t, \quad \alpha_t(x) = \sum_{j=1}^n v_{tj} f_j(x)$$

- смесь моделей-экспертов (Mixture of Experts)

с функциями компетентности (gating function) $g_t: X \rightarrow \mathbb{R}$

$$F(b_1, \dots, b_T, x, \alpha) = \sum_{t=1}^T g_t(x, \alpha_t) b_t(x)$$

Обучение предсказательных моделей и их ансамблей

$\mathcal{L}(b, x_i)$ — функция потерь модели $b(x_i, w)$ при ответе y_i ;

Минимизация эмпирического риска для базовых алгоритмов:

$$\sum_{i=1}^{\ell} \mathcal{L}(b_t(x_i, w), y_i) \rightarrow \min_w$$

Минимизация эмпирического риска для добавления базового алгоритма b_T в ансамбль при фиксации предыдущих:

$$\sum_{i=1}^{\ell} \mathcal{L}\left(\sum_{t=1}^{T-1} \alpha_t b_t(x_i, w_t) + \alpha_T b_T(x_i, w_T), y_i\right) \rightarrow \min_{\alpha_T, w_T}$$

Ю.И.Журавлëв. Корректные алгебры над множествами некорректных (эвристических) алгоритмов (I, II, III). Кибернетика, Киев, 1977–1978.

M.Kearns, L.G.Valiant. Cryptographic limitations on learning Boolean formulae and finite automata. 1989.

Y.Freund, R.E.Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. 1995.

K.B.Рудаков, K.B.Воронцов. О методах оптимизации и монотонной коррекции в алгебраическом подходе к проблеме распознавания. Доклады РАН, 1999.

Алгоритмы вычисления оценок, АВО (Ю. И. Журавлёв)

Объединение основных на тот момент нестрогих (эвристических) принципов:

- символизм (вывод правил из данных)
- эволюционизм (отбор лучших правил)
- аналогизм (оценки сходства объектов)
- байесионизм (классы — смеси плотностей)
- коннекционизм (взвешенное голосование)



Юрий
Иванович
Журавлёв
(1935–2022)

$$a(x) = \arg \max_{y \in Y} \sum_{i: y_i=y} \sum_{\omega \in \Omega} w_{\omega i} B_{\omega i}(x, x_i)$$

где $B_{\omega i}$ — бинарные функции сходства по наборам признаков ω :

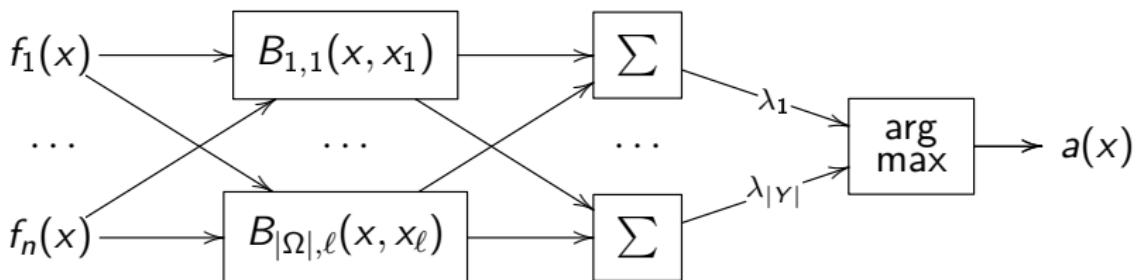
$$B_{\omega i}(x, x_i) = \bigwedge_{j \in \omega} [|f_j(x) - f_j(x_i)| < \varepsilon]$$

Журавлёв Ю. И., Никифоров В. В. Алгоритмы распознавания, основанные на вычислении оценок, 1971.

АВО объединяет многие эвристические принципы

≈ трёхслойная нейросеть RBF (Radial Basis Function):

$$a(x) = \arg \max_{y \in Y} \lambda_y \sum_{i,\omega} [y_i = y] w_{\omega i} B_{\omega i}(x, x_i)$$



- ≈ метод потенциальных функций $B_{\omega i}(x, x_i) = K\left(\frac{1}{h_{\omega i}} \rho_{\omega i}(x, x_i)\right)$
- ≈ линейный классификатор SVM с радиальным ядром
- ≈ байесовский классификатор с плотностями-смесями $p(x|y)$
- ≈ отбор эталонов: $w_{\omega i} = 0$ для не-эталонов x_i
- ≈ отбор признаков в сферических логических закономерностях

Задачи на малых данных

Особенности геологических данных в задачах поиска месторождений редкого типа (золото, уран, алмазы и т.д.)

- объектов мало ($7 + 11$), признаков много (более сотни)
- надёжной геофизической модели не существует
- в данных бывают пропуски — неизмеренные значения

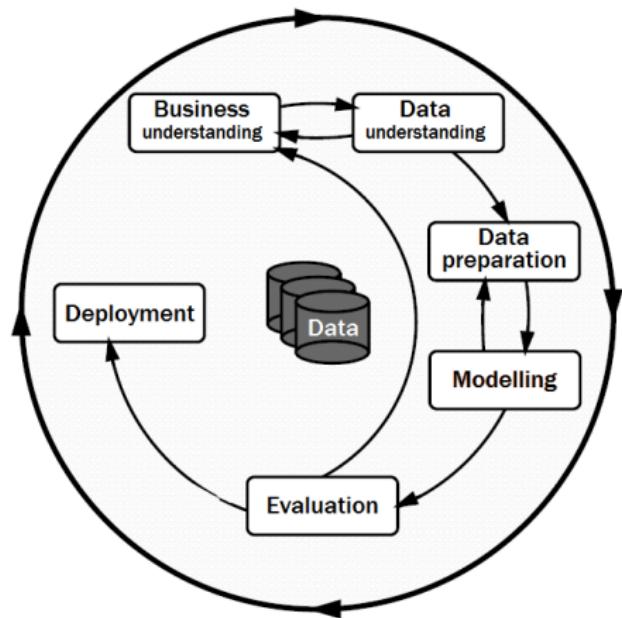
Группа признаков	Пространственно-временные										Вещественные				
	Признаки X_n	2	3	4	6	69	21	22	23	24	59
Месторождения M_i															
Витватерсrand (M ⁴ i)	1	0	0	1	0	1	1	1	1	0	1	0	0	0	1
Блайдс-Ривер (M ² i)	1	0	1	0	0	0	0	0	1	1	0	0	0	0	1
Жакобита (M ¹ i)	0	0	1	0	1	0	1	1	1	1	0	1	1	1	1
Мунана, Габон (M ⁴ i)	0	1	0	0	1	0	—	1	1	0	0	1	0	—	—
Тарква, Гана (M ⁸ i)	1	0	0	0	0	0	—	1	1	1	0	0	0	1	1
Австралия (M ⁴ i)	0	0	1	—	0	1	—	1	—	1	0	0	1	1	1
Эпо-Колия, Финляндия (M ⁷ i)	1	0	0	1	1	1	1	—	—	—	—	1	—	1	1

Кренделев Ф. П., Дмитриев А. Н., Журавлев Ю. И. Сравнение геологического строения зарубежных месторождений докембрийских конгломератов с помощью дискретной математики. Доклады АН СССР. 1967

Дмитриев А. Н., Журавлев Ю. И., Кренделев Ф. П. Об одном принципе классификации и прогноза геологических объектов и явлений. 1968.

Межотраслевой стандарт интеллектуального анализа данных

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



Компании-инициаторы:

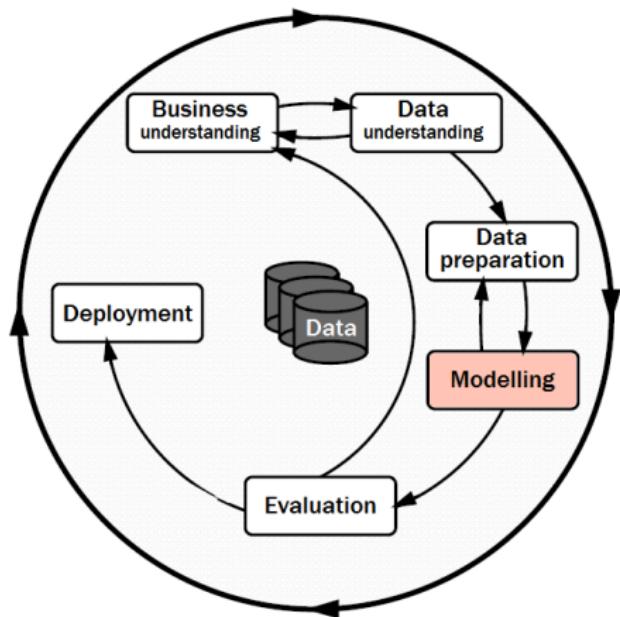
- SPSS
- Teradata
- Daimler AG
- NCR Corp.
- OHRA

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)

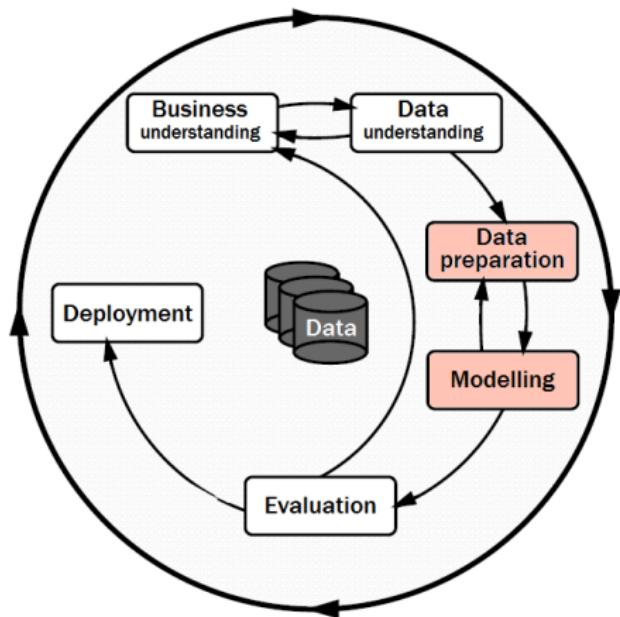


Эволюция ИИ:

- *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)



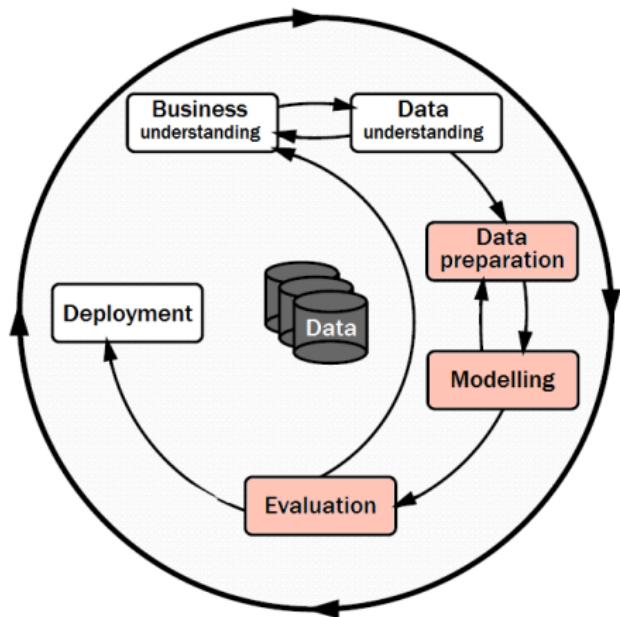
Эволюция ИИ:

- *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)

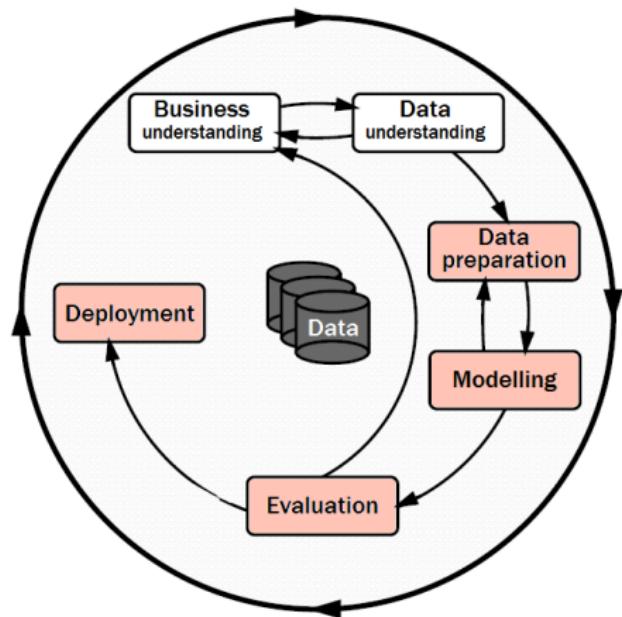


Эволюция ИИ:

- *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:

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

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

W_t — пространство параметров модели A_t

$\mu_t: (X \times Y)^\ell \rightarrow W_t$ — 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):
 $\mathcal{F} = \{f_j: X \rightarrow D_j: j = 1, \dots, n\}$ — множество признаков;
метод обучения $\mu_{\mathcal{G}}$ использует только признаки $\mathcal{G} \subseteq \mathcal{F}$.

Методология анализа ошибок (или потерь)

$\mathcal{L}(w, x_i)$ — функция потерь (чем меньше, тем лучше).

Среднее потерь на выборке U и эмпирическое распределение:

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

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

Анализ потерь на обучающей выборке:

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

Сравнительный анализ потерь на обучении и teste:

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

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)

Автоматический выбор моделей и гиперпараметров (AutoML)

Проблема:

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

Дано: выборка «задача, структура» → 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

Начинаем с постановки задачи: ДНК (дано, найти, критерий)

П.Домингос (2015) ошибался, что «верховный алгоритм» (AGI) появится как объединение подходов основных научных школ

Ю.И.Журавлёв в *алгоритмах вычисления оценок* (1971)
уже сделал это, но это не привело к появлению AGI

Теперь ясно, что слагаемые успеха — совсем другие:

- ① обучаемая векторизация данных → глубокое обучение
- ② большие данные + большие модели + нейропроцессоры
- ③ большие размерности + оптимизация без переобучения
- ④ большие языковые модели → эмерджентность

Воронцов К. В. Лекции по машинному обучению. <https://bit.ly/ML-Vorontsov>
Николенко С. Машинное обучение: основы, 2025

Мэрфи К.П. Вероятностное машинное обучение. В трёх томах, 2022–2024

Дайзенрот М.П. и др. Математика в машинном обучении, 2024

Кристофер Бишоп, Хью Бишоп. Глубокое обучение: принципы и концепции, 2025

Марков С. Охота на электроовец. Большая книга искусственного интеллекта. 2024

Домингос П. Верховный алгоритм. Как машинное обучение изменит наш мир. 2016