Задача обучения по прецедентам

X — множество *объектов*;
Y — множество *ответов*;
$y\colon X \to Y$ — неизвестная зависимость (target function).

Дано:
$\{x_1, \dots, x_\ell\} \subset X$ — *обучающая выборка* (training sample);
$y_i = y(x_i)$, $i = 1, \dots, \ell$ — известные ответы.

Найти:
$a\colon X \to Y$ — алгоритм, решающую функцию (decision function), приближающую y на всём множестве X.

Весь курс машинного обучения — это конкретизация:

- как задаются объекты и какими могут быть ответы;
- как строить функцию a;
- в каком смысле a должен приближать y.

Как задаются объекты. Признаковое описание

$f_j\colon X \to D_j,\ \ j = 1, \ldots, n$ — признаки объектов (features).

Типы признаков:

- $D_j = \{0, 1\}$ — *бинарный* признак f_j;
- $|D_j| < \infty$ — *номинальный* признак f_j;
- $|D_j| < \infty$, D_j упорядочено — *порядковый* признак f_j;
- $D_j = \mathbb{R}$ — *количественный* признак f_j.

Вектор $\bigl(f_1(x), \ldots, f_n(x)\bigr)$ — *признаковое описание* объекта x.

Матрица «*объекты–признаки*» (feature data)

$$F = \bigl\|f_j(x_i)\bigr\|_{\ell\times n} = \begin{pmatrix} f_1(x_1) & \ldots & f_n(x_1) \\ \ldots & \ldots & \ldots \\ f_1(x_\ell) & \ldots & f_n(x_\ell) \end{pmatrix}$$

Как задаются ответы. Типы задач

Задачи классификации (classification):

- $Y = \{-1, +1\}$ — *классификация* на 2 класса.
- $Y = \{1, \dots, M\}$ — на M непересекающихся классов.
- $Y = \{0, 1\}^M$ — на M классов, которые могут пересекаться.

Задачи восстановления регрессии (regression):

- $Y = \mathbb{R}$ или $Y = \mathbb{R}^m$.

Задачи ранжирования (ranking, learning to rank):

- Y — конечное упорядоченное множество.

Предсказательная модель

Модель (predictive model) — параметрическое семейство функций

$$A = \{a(x) = g(x, \theta) \mid \theta \in \Theta\},$$

где $g\colon X \times \Theta \to Y$ — фиксированная функция,
Θ — множество допустимых значений параметра θ.

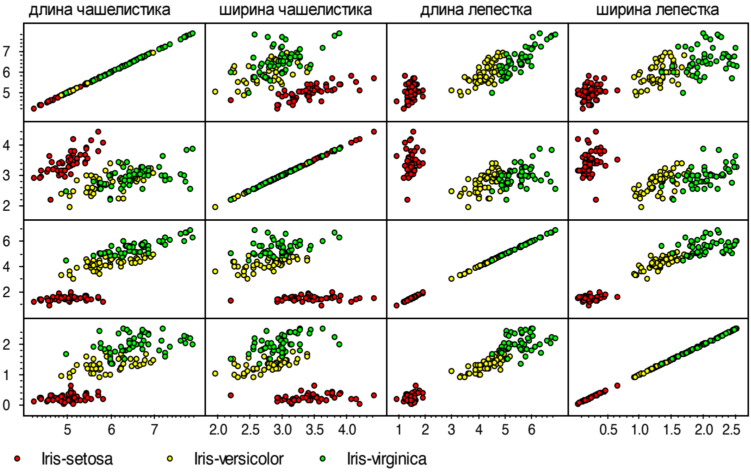
Пример.
Линейная модель с вектором параметров $\theta = (\theta_1, \dots, \theta_n)$, $\Theta = \mathbb{R}^n$:

$$g(x, \theta) = \sum_{j=1}^{n} \theta_j f_j(x) \quad \text{— для регрессии и ранжирования}, \; Y = \mathbb{R};$$

$$g(x, \theta) = \operatorname{sign} \sum_{j=1}^{n} \theta_j f_j(x) \quad \text{— для классификации}, \; Y = \{-1, +1\}.$$

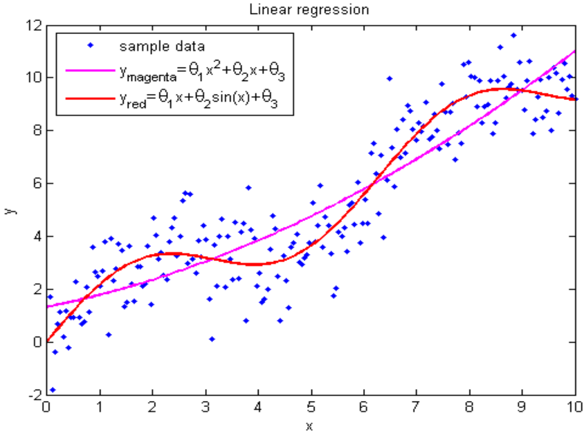
Пример: задача классификации цветков ириса [Фишер, 1936]

$n = 4$ признака, $|Y| = 3$ класса, длина выборки $\ell = 150$.



Пример: задача регрессии, модельные данные

$X = Y = \mathbb{R}$, $\ell = 200$, $n = 3$ признака: $\{x, x^2, 1\}$ или $\{x, \sin x, 1\}$



Вывод: признаковое описание можно задавать по-разному

Этап обучения (train):

Метод обучения (learning algorithm) $\mu\colon (X \times Y)^\ell \to A$
по выборке $X^\ell = (x_i, y_i)_{i=1}^\ell$ строит алгоритм $a = \mu(X^\ell)$:

$$\boxed{\begin{pmatrix} f_1(x_1) & \dots & f_n(x_1) \\ \dots & \dots & \dots \\ f_1(x_\ell) & \dots & f_1(x_\ell) \end{pmatrix} \xrightarrow{y} \begin{pmatrix} y_1 \\ \dots \\ y_\ell \end{pmatrix}} \xrightarrow{\mu} a$$

Этап применения (test):

алгоритм a для новых объектов $x'_1, \dots, x'_k$ выдаёт ответы $a(x'_i)$.

$$\begin{pmatrix} f_1(x'_1) & \dots & f_n(x'_1) \\ \dots & \dots & \dots \\ f_1(x'_k) & \dots & f_n(x'_k) \end{pmatrix} \xrightarrow{a} \begin{pmatrix} a(x'_1) \\ \dots \\ a(x'_k) \end{pmatrix}$$

Функционалы качества

$\mathscr{L}(a, x)$ — *функция потерь* (loss function) — величина ошибки алгоритма $a \in A$ на объекте $x \in X$.

Функции потерь для задач классификации:

- $\mathscr{L}(a, x) = \bigl[a(x) \neq y(x)\bigr]$ — индикатор ошибки;

Функции потерь для задач регрессии:

- $\mathscr{L}(a, x) = \bigl|a(x) - y(x)\bigr|$ — абсолютное значение ошибки;
- $\mathscr{L}(a, x) = \bigl(a(x) - y(x)\bigr)^2$ — квадратичная ошибка.

Эмпирический риск — функционал качества алгоритма a на X^ℓ:

$$Q(a, X^\ell) = \frac{1}{\ell} \sum_{i=1}^{\ell} \mathscr{L}(a, x_i).$$

Минимизация эмпирического риска (empirical risk minimization):

$$\mu(X^\ell) = \arg\min_{a\in A} Q(a, X^\ell).$$

Пример: *метод наименьших квадратов* ($Y = \mathbb{R}$, $\mathscr{L}$ квадратична):

$$\mu(X^\ell) = \arg\min_{\theta} \sum_{i=1}^{\ell} \bigl(g(x_i, \theta) - y_i\bigr)^2.$$

Понятие обобщающей способности (generalization performance):

- найдём ли мы «закон природы» или *переобучимся*, то есть подгоним функцию $g(x_i, \theta)$ под заданные точки?
- будет ли $a = \mu(X^\ell)$ приближать функцию y на всём X?
- будет ли $Q(a, X^k)$ мало́ на новых данных — *контрольной выборке* $X^k = (x'_i, y'_i)_{i=1}^k$, $y'_i = y(x_i)$?

Резюме

- **Основные понятия машинного обучения:** объект, ответ, признак, предсказательная модель, метод обучения, эмпирический риск, переобучение.
- **Прикладные задачи машинного обучения** встречаются во всех областях бизнеса, науки, производства — об этом в следующей лекции

Задача восстановления зависимости $y = y(x)$
по точкам *обучающей выборки* (x_i, y_i), $i = 1, \dots, \ell$.

Дано: векторы $x_i = (x_i^1, \dots, x_i^n)$ — объекты обучающей выборки,
$y_i = y(x_i)$ — правильные ответы, $i = 1, \dots, \ell$:

$$\begin{pmatrix} x_1^1 & \dots & x_1^n \\ \dots & \dots & \dots \\ x_\ell^1 & \dots & x_\ell^n \end{pmatrix} \xrightarrow{y} \begin{pmatrix} y_1 \\ \dots \\ y_\ell \end{pmatrix}$$

Найти: функцию $a(x)$, способную давать правильные ответы
на *тестовых объектах* $\tilde{x}_i = (\tilde{x}_i^1, \dots, \tilde{x}_i^n)$, $i = 1, \dots, k$:

$$\begin{pmatrix} \tilde{x}_1^1 & \dots & \tilde{x}_1^n \\ \dots & \dots & \dots \\ \tilde{x}_k^1 & \dots & \tilde{x}_k^n \end{pmatrix} \xrightarrow{a?} \begin{pmatrix} a(\tilde{x}_1) \\ \dots \\ a(\tilde{x}_k) \end{pmatrix}$$

Задачи медицинской диагностики

Объект — пациент в определённый момент времени.

Классы: диагноз или способ лечения или исход заболевания.

Примеры признаков:

- **бинарные:** пол, головная боль, слабость, тошнота, и т. д.
- **порядковые:** тяжесть состояния, желтушность, и т. д.
- **количественные:** возраст, пульс, артериальное давление, содержание гемоглобина в крови, доза препарата, и т. д.

Особенности задачи:

- обычно много «пропусков» в данных;
- как правило, недостаточный объём данных;
- нужен интерпретируемый алгоритм классификации;
- нужна оценка вероятности (риска | успеха | исхода).

Задача кредитного скоринга

Объект — заявка на выдачу банком кредита.

Классы — bad или good.

Примеры признаков:

- **бинарные:** пол, наличие телефона, и т. д.
- **номинальные:** место проживания, профессия, работодатель, и т. д.
- **порядковые:** образование, должность, и т. д.
- **количественные:** возраст, зарплата, стаж работы, доход семьи, сумма кредита, и т. д.

Особенности задачи:

- нужно оценивать вероятность дефолта $P(\text{bad})$.

Задача предсказания оттока клиентов

Объект — абонент в определённый момент времени.

Классы — уйдёт или не уйдёт в следующем месяце.

Примеры признаков:

- **бинарные:** корпоративный клиент, включение услуг, и т. д.
- **номинальные:** тарифный план, регион проживания, и т. д.
- **количественные:** длительность разговоров (входящих, исходящих, СМС, и т. д.), частота оплаты, и т. д.

Особенности задачи:

- нужно оценивать вероятность ухода;
- сверхбольшие выборки;
- не ясно, какие признаки вычислять по «сырым» данным.

Объект — текстовый документ.

Классы — рубрики иерархического тематического каталога.

Примеры признаков:

- **номинальные:** автор, издание, год, и т. д.
- **количественные:** для каждого термина — частота в тексте, в заголовках, в аннотации, и т. д.

Особенности задачи:

- лишь небольшая часть документов имеют метки y_i;
- документ может относиться к нескольким рубрикам;
- в каждом ребре дерева свой классификатор на 2 класса.

Задача прогнозирования стоимости недвижимости

Объект — квартира в Москве.

Примеры признаков:

- **бинарные:** наличие балкона, лифта, мусоропровода, охраны, и т. д.
- **номинальные:** район города, тип дома (кирпичный/панельный/блочный/монолит), и т. д.
- **количественные:** число комнат, жилая площадь, расстояние до центра, до метро, возраст дома, и т. д.

Особенности задачи:

- выборка неоднородна, стоимость меняется со временем;
- разнотипные признаки;
- для линейной модели нужны преобразования признаков.

Задача прогнозирования объёмов продаж

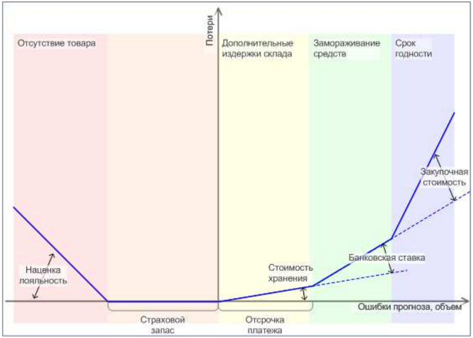
Объект — тройка $\langle$товар, магазин, день$\rangle$.

Примеры признаков:

- **бинарные:** выходной день, праздник, промоакция, и т. д.
- **количественные:** объёмы продаж в предшествующие дни.

Особенности задачи:

- функция потерь не квадратична и даже не симметрична;
- разреженные данные.



Конкурс kaggle.com: TFI Restaurant Revenue Prediction

Объект — место для открытия нового ресторана.

Предсказать — прибыль от ресторана через год.

Примеры признаков:

- демографическими свойствами района;
- цены на недвижимость поблизости;
- маркетинговые данные: наличие школ, офисов и т.д.

Особенности задачи:

- мало объектов, много признаков;
- разнотипные признаки;
- есть выбросы;
- разнородные объекты (возможно, имеет смысл строить разные модели для мелких и крупных городов).

Задача ранжирования поисковой выдачи

Объект — пара $\langle$запрос, документ$\rangle$.

Классы — релевантен или не релевантен,
разметка делается людьми — асессорами.

Примеры признаков:

- **количественные:**
  частота слов запроса в документе,
  число ссылок на документ,
  число кликов на документ: всего, по данному запросу,
  и т. д.

Особенности задачи:

- оптимизируется не число ошибок, а качество ранжирования;
- сверхбольшие выборки;
- проблема конструирования признаков по сырым данным.

Задача ранжирования в рекомендательных системах

Объект — пара $\langle$клиент, товар$\rangle$
(товары — книги, фильмы, музыка).

Предсказать: вероятность покупки или рейтинг товара.

Примеры признаков:

- **количественные:**
  частота покупок или средний рейтинг схожих товаров для данного клиента;
  частота покупок или средний рейтинг данного товара для схожих клиентов;
  оценки интересов клиента;
  оценки интересов товара;

Особенности задачи:

- сверхбольшие разреженные данные;
- интересы скрыты, их надо сначала выявить.

Конкурс kaggle.com: Avito Context Ad Clicks Prediction

Объект — тройка ⟨пользователь, объявление, баннер⟩.

Предсказать — кликнет ли пользователь по контекстной рекламе, которую показали в ответ на его запрос на avito.ru.

Сырые данные:

- все действия пользователя на сайте,
- профиль пользователя (бразуер, устройство и т. д.),
- история показов и кликов других пользователей по баннеру,
- ... всего 10 таблиц данных.

Особенности задачи:

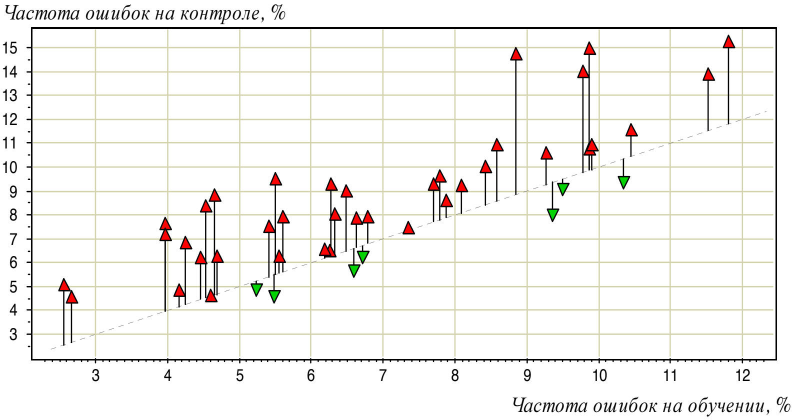
- признаки надо придумывать;
- данных много — сотни миллионов показов;
- основной критерий качества — доход рекламной площадки;
- но имеются и дополнительные критерии.

Резюме

- **Прикладные задачи машинного обучения** встречаются во всех областях бизнеса, науки, производства
- **Особенности данных в прикладных задачах:**
  — 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);
  — неструктурированные (нет признаковых описаний);
  — нетривиальные критерии качества.

Пример. Переобучение в задаче медицинской диагностики

Задача предсказания отдалённого результата хирургического лечения атеросклероза. Точки — различные алгоритмы.



Пример: переобучение полиномиальной регрессии

Зависимость $y(x) = \dfrac{1}{1+25x^2}$ на отрезке $x \in [-2, 2]$.

Признаковое описание $x \mapsto (1, x^1, x^2, \ldots, x^n)$.

Модель полиномиальной регрессии

$$a(x, \theta) = \theta_0 + \theta_1 x + \cdots + \theta_n x^n \quad \text{— полином степени } n.$$

Обучение методом наименьших квадратов:

$$Q(a, X^\ell) = \sum_{i=1}^{\ell} (\theta_0 + \theta_1 x_i + \cdots + \theta_n x_i^n - y_i)^2 \to \min_{\theta_0, \ldots, \theta_n}.$$

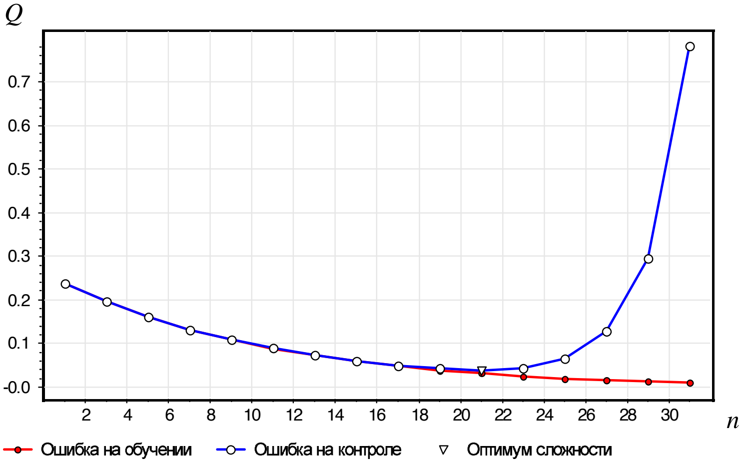
Обучающая выборка: $X^\ell = \left\{x_i = 4\frac{i-1}{\ell-1} - 2 \;\middle|\; i = 1, \ldots, \ell\right\}$.

Контрольная выборка: $X^k = \left\{x_i = 4\frac{i-0.5}{\ell-1} - 2 \;\middle|\; i = 1, \ldots, \ell-1\right\}$.

Что происходит с $Q(a, X^\ell)$ и $Q(a, X^k)$ при увеличении n?

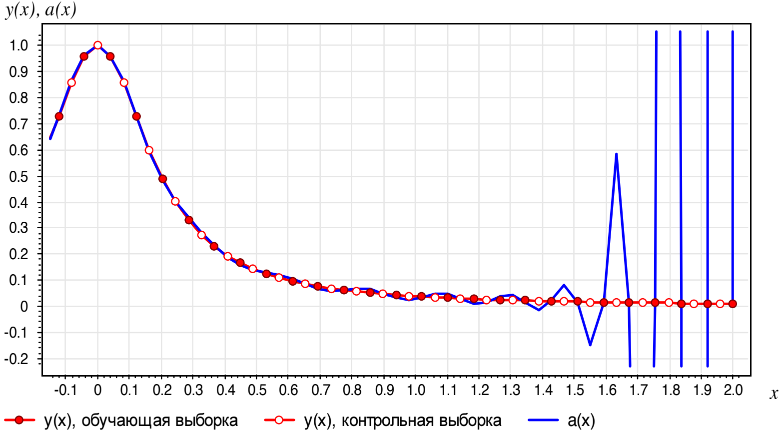
Пример переобучения: эксперимент при $\ell = 50$, $n = 1, \dots, 31$

Переобучение — это когда $Q(\mu(X^\ell), X^k) \gg Q(\mu(X^\ell), X^\ell)$:



Пример переобучения: эксперимент при $\ell = 50$, $n = 38$

$$y(x) = \frac{1}{1 + 25x^2}; \quad a(x) \text{ — полином степени } n = 38$$



Эмпирические оценки обобщающей способности

- Эмпирический риск на тестовых данных (hold-out):

$$\mathrm{HO}(\mu, X^\ell, X^k) = Q(\mu(X^\ell), X^k) \to \min$$

- Скользящий контроль (leave-one-out), $L = \ell + 1$:

$$\mathrm{LOO}(\mu, X^L) = \frac{1}{L} \sum_{i=1}^{L} \mathscr{L}(\mu(X^L \backslash \{x_i\}), x_i) \to \min$$

- Кросс-проверка (cross-validation) по N разбиениям, $X^L = X_n^\ell \sqcup X_n^k$, $L = \ell + k$:

$$\mathrm{CV}(\mu, X^L) = \frac{1}{N} \sum_{n=1}^{N} Q(\mu(X_n^\ell), X_n^k) \to \min$$

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 (308 задач, 09-02-2015)

Эксперименты на модельных (синтетических) данных

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

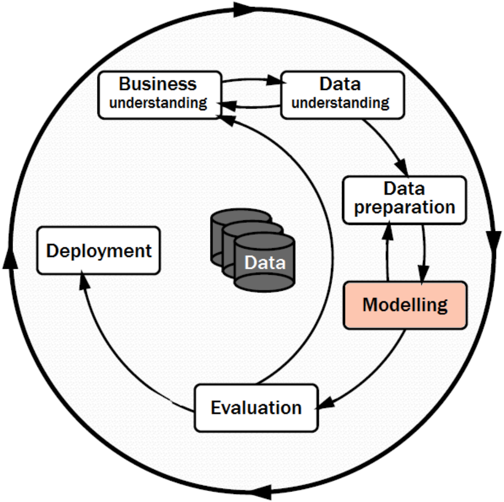
Эксперименты на модельных (synthetic) данных:

- 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 полумодельных (semi-synthetic) данных:

- цель — протестировать помехоустойчивость модели
- объекты x_i из реальной задачи (+ шум)
- ответы $y_i = a(x_i)$ для полученного решения $a(x)$ (+ шум)

CRISP-DM — межотраслевой стандарт решения задач интеллектуального анализа данных



Этапы решения задач машинного обучения:

- понимание задачи и данных;
- предобработка данных и изобретение признаков;
- построение модели;
- сведение обучения к оптимизации;
- решение проблем оптимизации и переобучения;
- оценивание качества решения;
- внедрение и эксплуатация.

Задача классификации (обучение с учителем)

Задача восстановления зависимости $y\colon X \to Y,\ \ |Y| < \infty$ по точкам *обучающей выборки* (x_i, y_i), $i = 1, \dots, \ell$.

Дано: векторы $x_i = (x_i^1, \dots, x_i^n)$ — объекты обучающей выборки, $y_i = y(x_i)$ — классификации, ответы учителя, $\ i = 1, \dots, \ell$:

$$\begin{pmatrix} x_1^1 & \dots & x_1^n \\ \dots & \dots & \dots \\ x_\ell^1 & \dots & x_\ell^n \end{pmatrix} \xrightarrow{y^*} \begin{pmatrix} y_1 \\ \dots \\ y_\ell \end{pmatrix}$$

Найти: функцию $a(x)$, способную классифицировать объекты произвольной *тестовой выборки* $\tilde{x}_i = (\tilde{x}_i^1, \dots, \tilde{x}_i^n)$, $\ i = 1, \dots, k$:

$$\begin{pmatrix} \tilde{x}_1^1 & \dots & \tilde{x}_1^n \\ \dots & \dots & \dots \\ \tilde{x}_k^1 & \dots & \tilde{x}_k^n \end{pmatrix} \xrightarrow{a?} \begin{pmatrix} a(\tilde{x}_1) \\ \dots \\ a(\tilde{x}_k) \end{pmatrix}$$

Определение бинарного решающего дерева

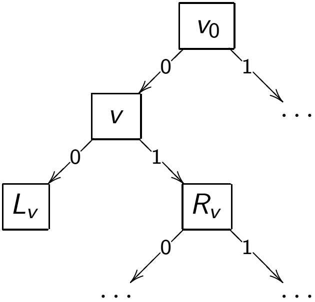
Бинарное решающее дерево — алгоритм классификации $a(x)$, задающийся бинарным деревом:

1) $\forall v \in V_{\text{внутр}} \;\to\;$ предикат $\beta_v : X \to \{0,1\}$, $\;\beta_v \in \mathscr{B}$,

2) $\forall v \in V_{\text{лист}} \;\to\;$ имя класса $c_v \in Y$,

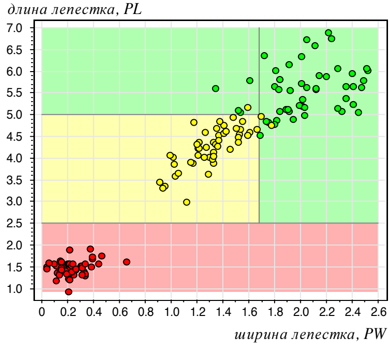
где $\mathscr{B}$ — множество бинарных признаков или предикатов (например, вида $\beta(x) = \bigl[x^j \geqslant \theta_j\bigr]$, $\;x^j \in \mathbb{R}$)

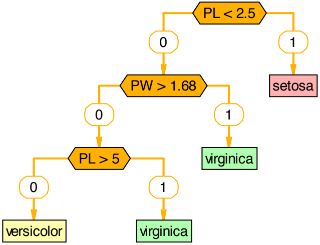
$v := v_0$;
пока $v \in V_{\text{внутр}}$
  если $\beta_v(x) = 1$ **то**
    переход вправо: $v := R_v$;
  иначе
    переход влево: $v := L_v$;
вернуть c_v.



Пример решающего дерева

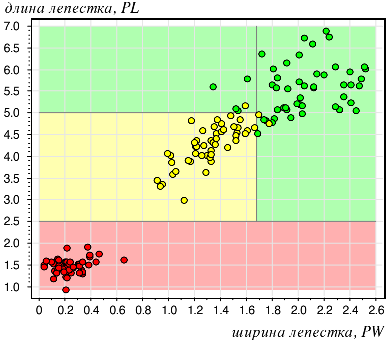
Задача Фишера о классификации цветков ириса на 3 класса, в выборке по 50 объектов каждого класса, 4 признака.

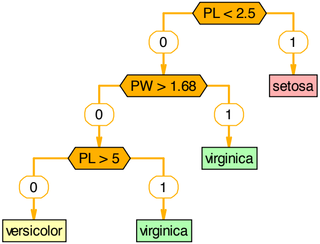




На графике: в осях двух самых информативных признаков (из 4) два класса разделились без ошибок, на третьем 3 ошибки.

Решающее дерево → покрывающий набор конъюнкций





setosa	$r_1(x) = \bigl[PL \leqslant 2.5\bigr]$
virginica	$r_2(x) = \bigl[PL > 2.5\bigr] \wedge \bigl[PW > 1.68\bigr]$
virginica	$r_3(x) = \bigl[PL > 5\bigr] \wedge \bigl[PW \leqslant 1.68\bigr]$
versicolor	$r_4(x) = \bigl[PL > 2.5\bigr] \wedge \bigl[PL \leqslant 5\bigr] \wedge \bigl[PW < 1.68\bigr]$

Жадный алгоритм построения дерева ID3

1: **ПРОЦЕДУРА** LearnID3 $(U \subseteq X^\ell)$;

2: **если** все объекты из U лежат в одном классе $c \in Y$ **то**

3:     **вернуть** новый лист v, $c_v := c$;

4: найти предикат с максимальной информативностью:
$\beta := \arg\max\limits_{\beta \in \mathscr{B}} I(\beta, U)$;

5: разбить выборку на две части $U = U_0 \sqcup U_1$ по предикату β:
$U_0 := \bigl\{x \in U \colon \beta(x) = 0\bigr\}$;
$U_1 := \bigl\{x \in U \colon \beta(x) = 1\bigr\}$;

6: **если** $U_0 = \varnothing$ или $U_1 = \varnothing$ **то**

7:     **вернуть** новый лист v, $c_v :=$ Мажоритарный класс(U);

8: создать новую внутреннюю вершину v: $\beta_v := \beta$;
построить левое поддерево: $L_v :=$ LearnID3 (U_0);
построить правое поддерево: $R_v :=$ LearnID3 (U_1);

9: **вернуть** v;

1. **Критерий Джини:**

$$I(\beta, X^\ell) = \#\bigl\{(x_i, x_j)\colon\ y_i = y_j \text{ и } \beta(x_i) = \beta(x_j)\bigr\}.$$

2. D-**критерий В.И.Донского:**

$$I(\beta, X^\ell) = \#\bigl\{(x_i, x_j)\colon\ y_i \neq y_j \text{ и } \beta(x_i) \neq \beta(x_j)\bigr\}.$$

3. **Энтропийный критерий:**

$$I(\beta, X^\ell) = \sum_{c \in Y} h\left(\frac{P_c}{\ell}\right) - \frac{p}{\ell} h\left(\frac{p_c}{p}\right) - \frac{\ell - p}{\ell} h\left(\frac{P_c - p_c}{\ell - p}\right),$$

где $h(z) \equiv -z \log_2 z$,
$P_c(X^\ell) = \#\{x_i\colon y_i = c\}$,
$p_c(X^\ell) = \#\{x_i\colon y_i = c \text{ и } \beta(x_i) = 1\}$,
$p(X^\ell) = \#\{x_i\colon \beta(x_i) = 1\}$.

На стадии обучения:

- $\beta_v(x)$ не определено $\Rightarrow$ x_i исключается из U для $I(\beta, U)$
- $q_v = \frac{|U_0|}{|U|}$ — оценка вероятности левой ветви, $\forall v \in V_{\text{внутр}}$
- $P(y|x,v) = \frac{1}{|U|}\#\bigl\{x_i \in U\colon y_i = y\bigr\}$ для всех $v \in V_{\text{лист}}$

На стадии классификации:

- $\beta_v(x)$ не определено $\Rightarrow$ *пропорциональное распределение*:

$$P(y|x,v) = q_v P(y|x,L_v) + \bigl(1 - q_v\bigr) P(y|x,R_v).$$

- $\beta_v(x)$ определено $\Rightarrow$ либо налево, либо направо:

$$P(y|x,v) = \bigl(1 - \beta_v(x)\bigr) P(y|x,L_v) + \beta_v(x) P(y|x,R_v).$$

- Окончательное решение — наиболее вероятный класс:

$$a(x) = \arg\max_{y\in Y} P(y|x,v_0).$$

Решающие деревья ID3: достоинства и недостатки

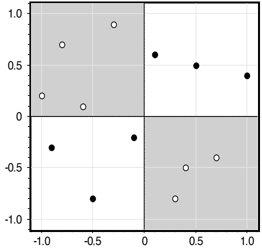
Достоинства:

- Интерпретируемость и простота классификации.
- Гибкость: можно варьировать множество $\mathscr{B}$.
- Допустимы разнотипные данные и данные с пропусками.
- Трудоёмкость линейна по длине выборки $O(|\mathscr{B}|h\ell)$.
- Не бывает отказов от классификации.

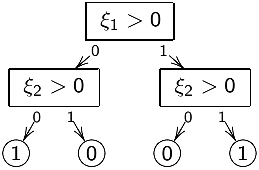
Недостатки:

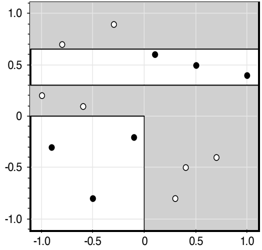
- Жадный ID3 переусложняет структуру дерева, и, как следствие, сильно переобучается.
- Фрагментация выборки: чем дальше v от корня, тем меньше статистическая надёжность выбора β_v, c_v.
- Высокая чувствительность к шуму, к составу выборки, к критерию информативности.

Жадный ID3 переусложняет структуру дерева

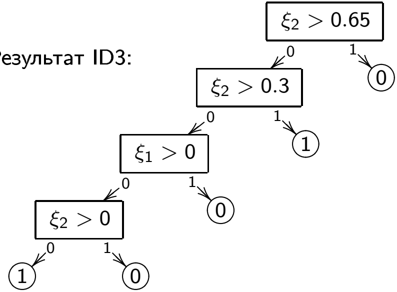


Оптимальное дерево для задачи XOR:





Результат ID3:



Усечение дерева (pruning). Алгоритм C4.5

X^k — независимая контрольная выборка, $k \approx 0.5\ell$.

для всех $v \in V_{\text{внутр}}$
    $S_v :=$ подмножество объектов X^k, дошедших до v;
    если $S_v = \varnothing$ **то**
        вернуть новый лист v, $c_v :=$ Мажоритарный класс(U);
    число ошибок при классификации S_v четырьмя способами:
        $r(v)$ — поддеревом, растущим из вершины v;
        $r_L(v)$ — поддеревом левой дочерней вершины L_v;
        $r_R(v)$ — поддеревом правой дочерней вершины R_v;
        $r_c(v)$ — к классу $c \in Y$.
    в зависимости от того, какое из них минимально:
        сохранить поддерево v;
        заменить поддерево v поддеревом L_v;
        заменить поддерево v поддеревом R_v;
        заменить поддерево v листом, $c_v := \arg\min\limits_{c \in Y} r_c(v)$.

CART: деревья регрессии и классификации

Обобщение на случай регрессии: $Y = \mathbb{R}$, $c_v \in \mathbb{R}$

Пусть U_v — множество объектов x_i, дошедших до вершины v

Значения в терминальных вершинах — МНК-решение:

$$c_v := \hat{y}(U_v) = \frac{1}{|U_v|} \sum_{x_i \in U_v} y_i$$

Критерий информативности — среднеквадратичная ошибка

$$I(\beta, U_v) = \sum_{x_i \in U_v} \bigl(\hat{y}_i(\beta) - y_i\bigr)^2,$$

где $\hat{y}_i(\beta) = \beta(x_i)\hat{y}(U_{v1}) + \bigl(1 - \beta(x_i)\bigr)\hat{y}(U_{v0})$
— прогноз после ветвления β и разбиения $U_v = U_{v0} \sqcup U_{v1}$

CART: критерий Minimal Cost-Complexity Pruning

Среднеквадратичная ошибка со штрафом за сложность дерева

$$C_\alpha = \sum_{x_i=1}^{\ell} \bigl(\hat{y}_i - y_i\bigr)^2 + \alpha |V_{\text{лист}}| \to \min$$

При увеличении α дерево последовательно упрощается.
Причём последовательность вложенных деревьев единственна.

Из этой последовательности выбирается дерево с минимальной ошибкой на тестовой выборке (Hold-Out).

Для случая классификации используется аналогичная стратегия усечения, с критерием Джини.

Резюме

- Преимущества решающих деревьев:
  - интерпретируемость,
  - допускаются разнотипные данные,
  - возможность обхода пропусков;
- Недостатки решающих деревьев:
  - переобучение,
  - фрагментация,
  - неустойчивость к шуму, составу выборки, критерию;
- Способы устранения этих недостатков:
  - редукция,
  - композиции (леса) деревьев.