Общая постановка задачи обучения по прецедентам

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

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

Найти:
$a: X \to Y$ — 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;
- в каком смысле «a приближает y»;
- как строить функцию a.

$f_j\colon X \to D_j,\ \ j = 1, \ldots, n$ — 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 $\big(f_1(x), \ldots, f_n(x)\big)$ — *признаковое описание* объекта x.

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

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

Задачи классификации (classification):

- $Y = \{-1, +1\}$ — *классификация* на 2 класса.
- $Y = \{1, \ldots, M\}$ — 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.

Модель (predictive model) — параметрическое семейство функций

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

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

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

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

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

Метод обучения (learning algorithm) — это отображение вида

$$\mu\colon (X \times Y)^\ell \to A,$$

которое произвольной выборке $X^\ell = (x_i, y_i)_{i=1}^\ell$
ставит в соответствие некоторый алгоритм $a \in A$.

В задачах обучения по прецедентам всегда есть два этапа:

- Этап *обучения* (training):
  метод μ по выборке X^ℓ строит алгоритм $a = \mu(X^\ell)$.
- Этап *применения* (testing):
  алгоритм a для новых объектов x выдаёт ответы $a(x)$.

Этап *обучения* (train):

метод μ по выборке $X^\ell = (x_i, y_i)_{i=1}^\ell$ строит алгоритм $a = \mu(X^\ell)$:

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

Этап *применения* (test):

алгоритм a для новых объектов x'_i выдаёт ответы $a(x'_i)$.

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

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

Функции потерь для задач классификации:

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

Функции потерь для задач регрессии:

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

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

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

Метод минимизации эмпирического риска:

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

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

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

Проблема обобщающей способности:

- 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 $X^k = (x'_i, y'_i)_{i=1}^k$, $y'_i = y(x_i)$?

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

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

Обучение методом наименьших квадратов:

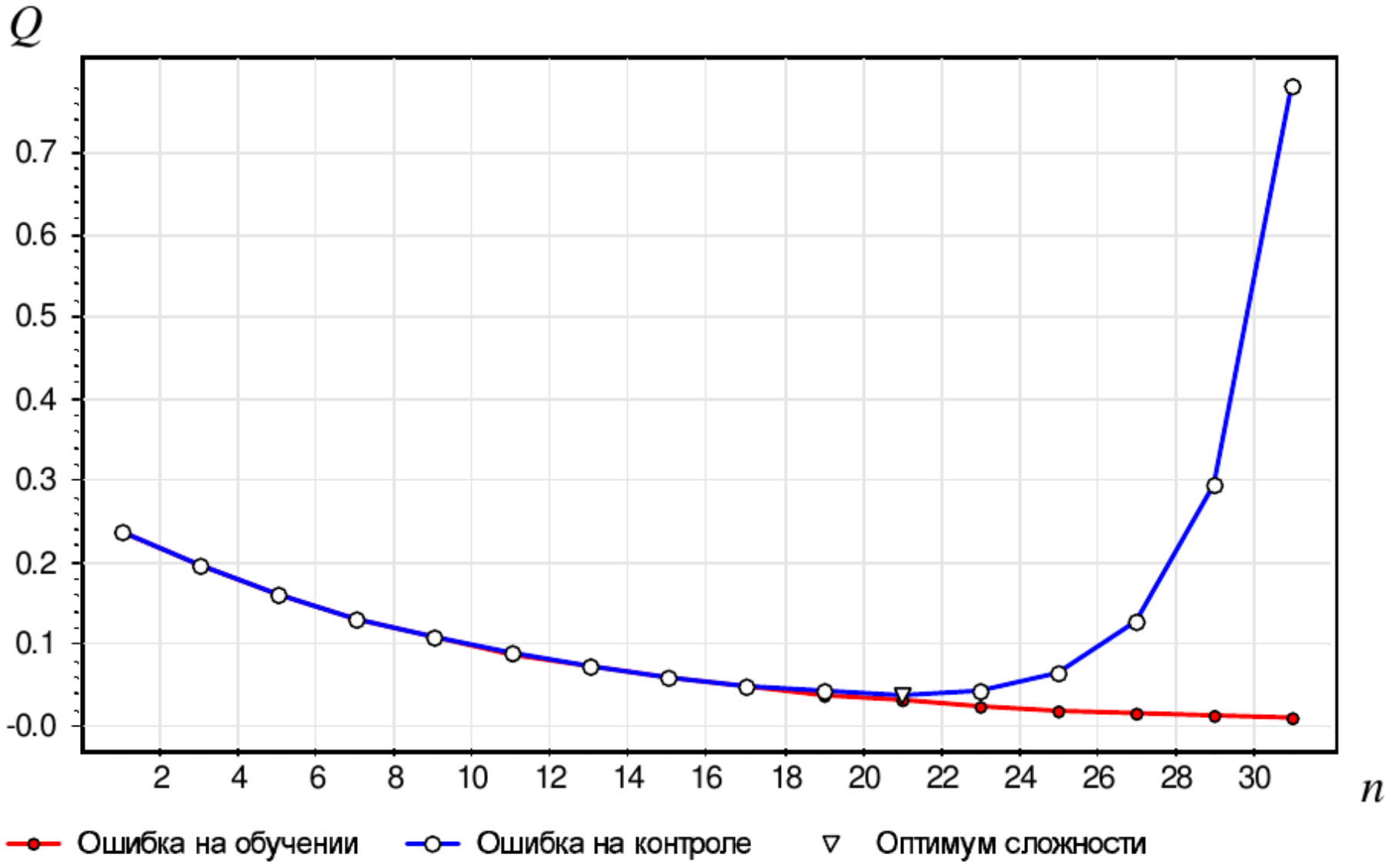
$$Q(\theta, X^\ell) = \sum_{i=1}^{\ell} (\theta_0 + \theta_1 x_i + \cdots + \theta_n x_i^n - y_i)^2 \to \min_{\theta_0, \ldots, \theta_n}.$$

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

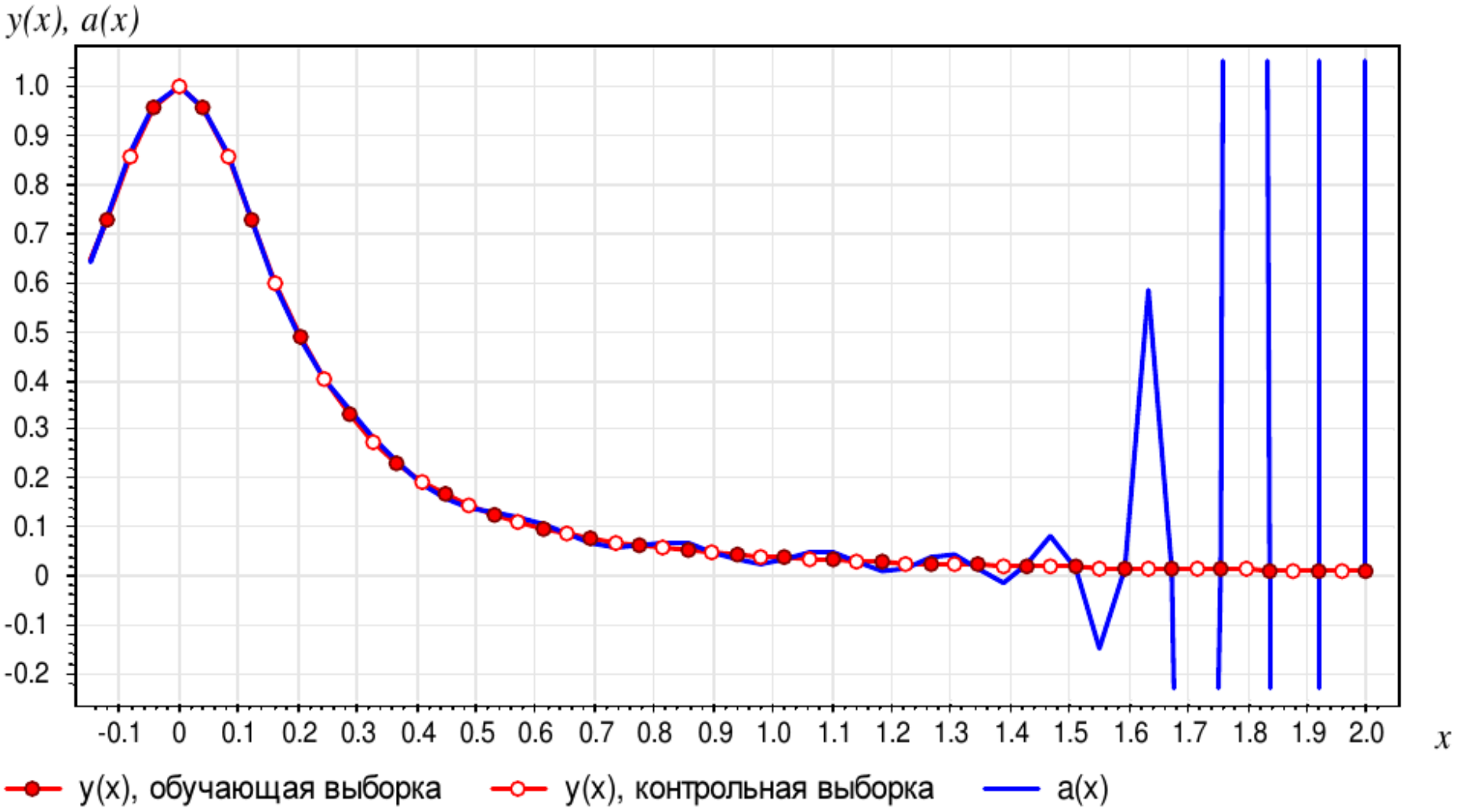
Контрольная выборка: $X^k = \left\{ x_i = 4\frac{i-0.5}{\ell-1} - 2 \;\middle|\; i = 1, \ldots, \ell - 1 \right\}$.

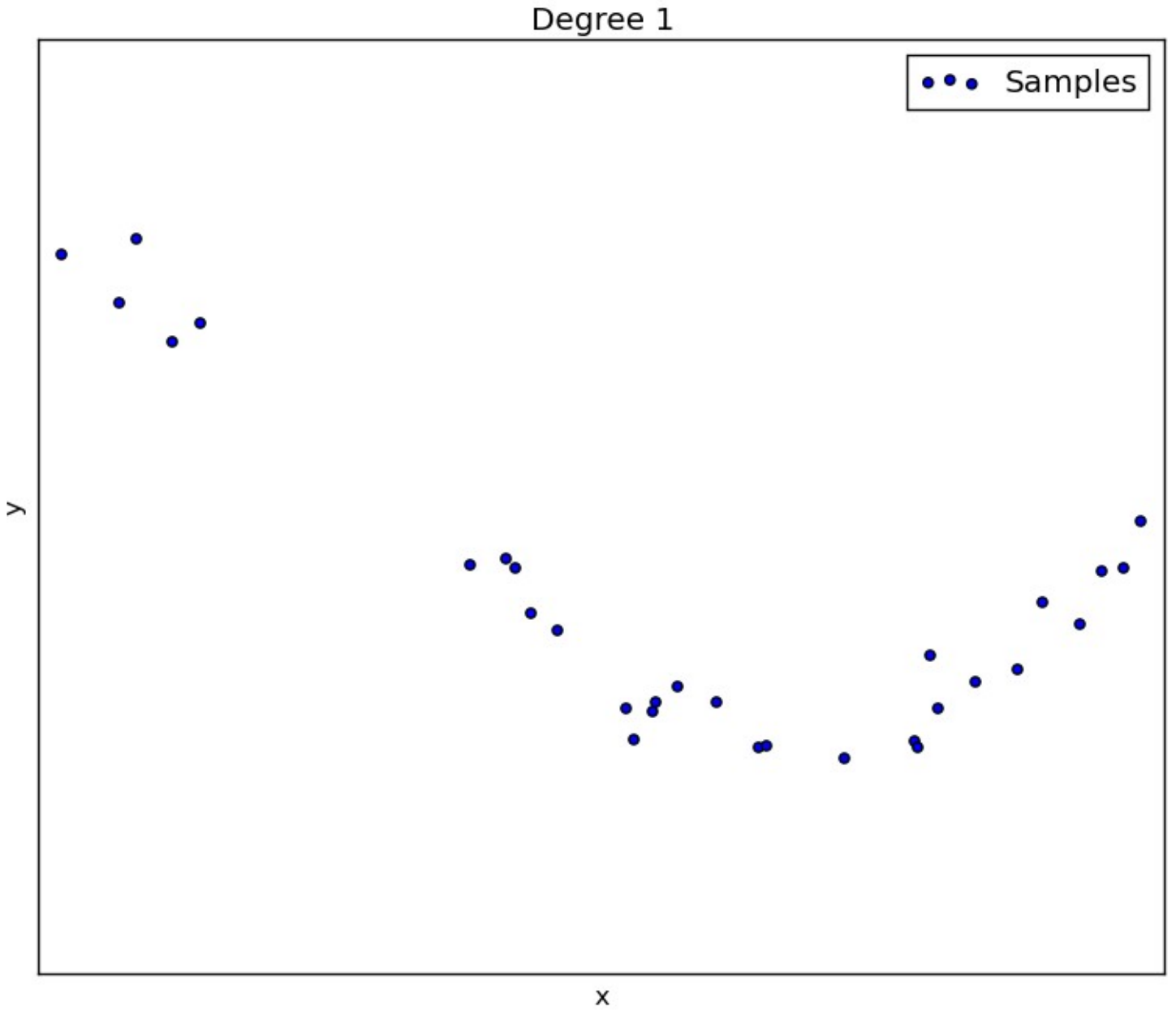
Что происходит с $Q\big(\theta, X^\ell\big)$ и $Q\big(\theta, X^k\big)$ при увеличении n?

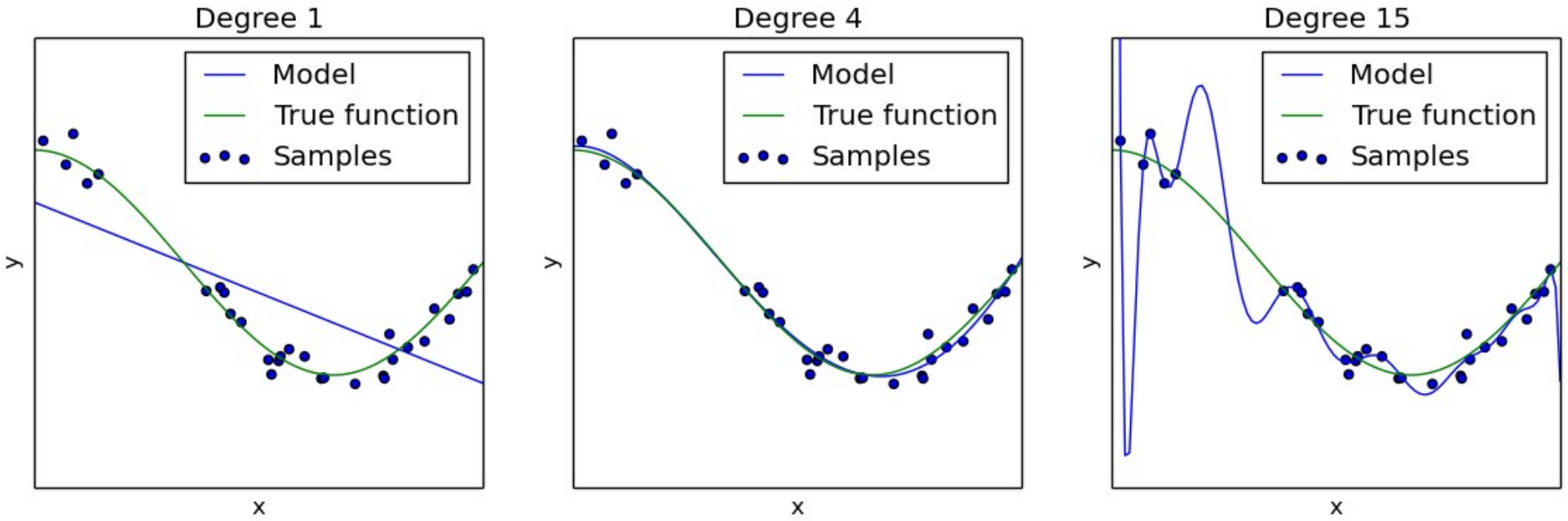
Переобучение — это когда $Q\big(\mu(X^\ell), X^k\big) \gg Q\big(\mu(X^\ell), X^\ell\big)$:

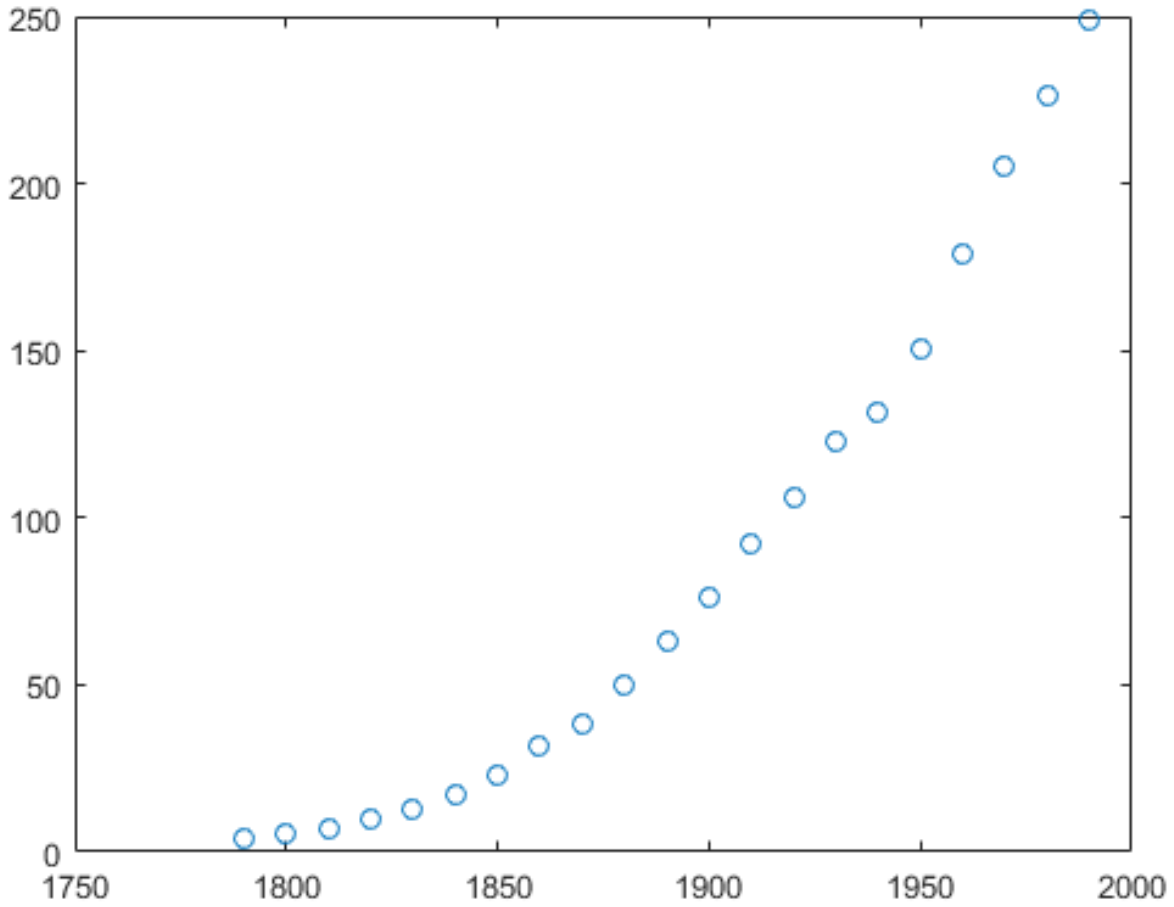


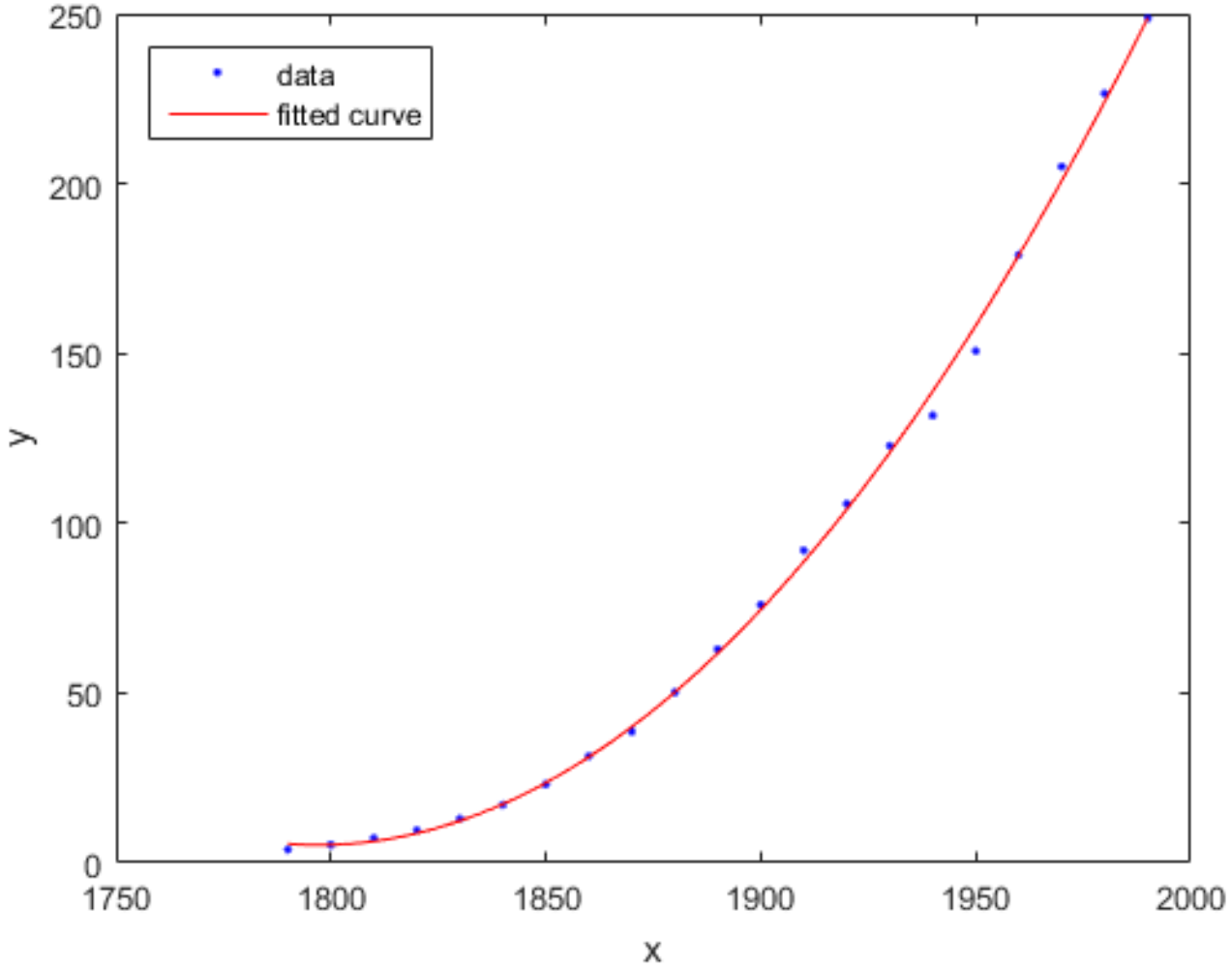
$$y(x) = \frac{1}{1 + 25x^2}; \quad a(x) \text{ — полином степени } n = 38$$

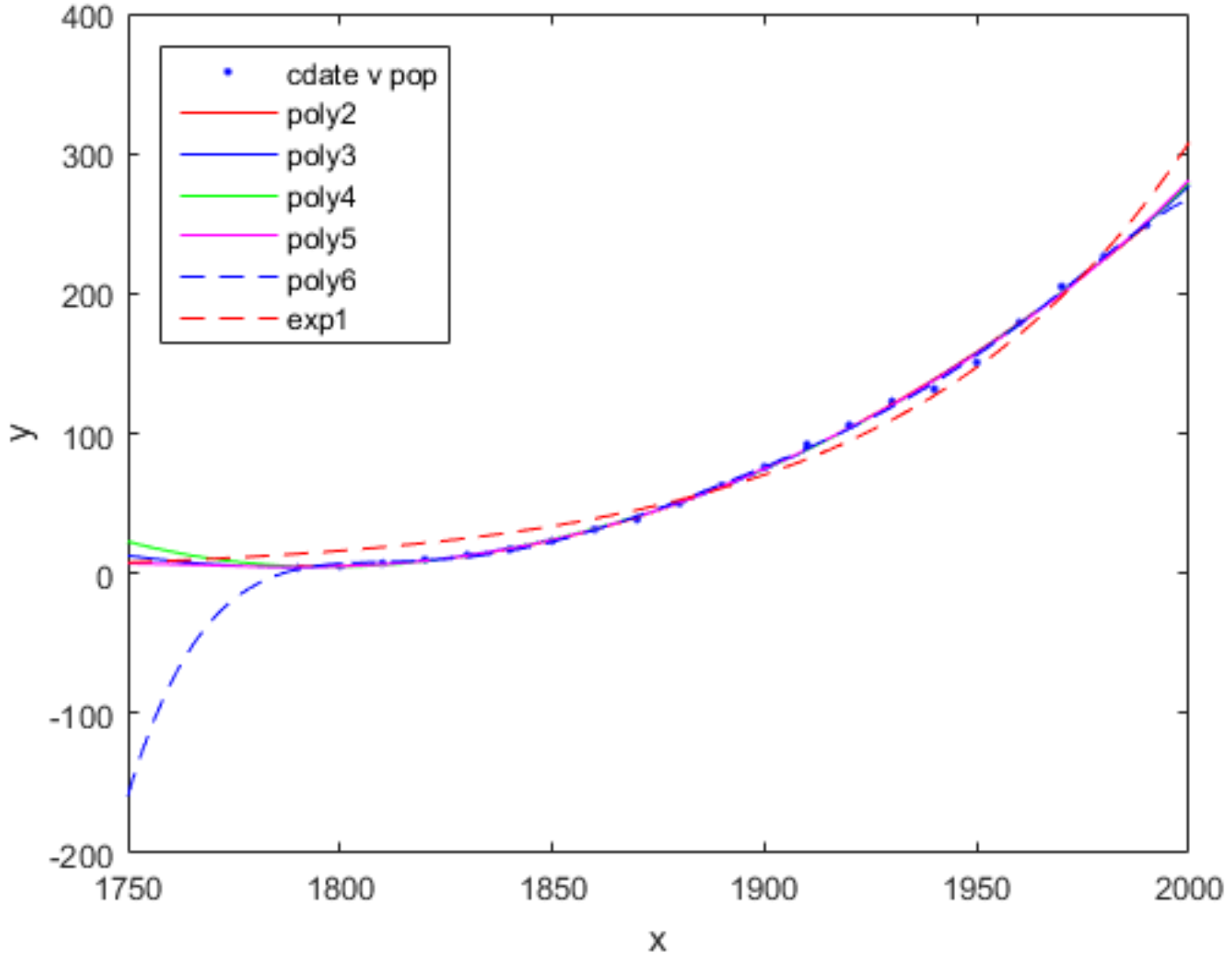


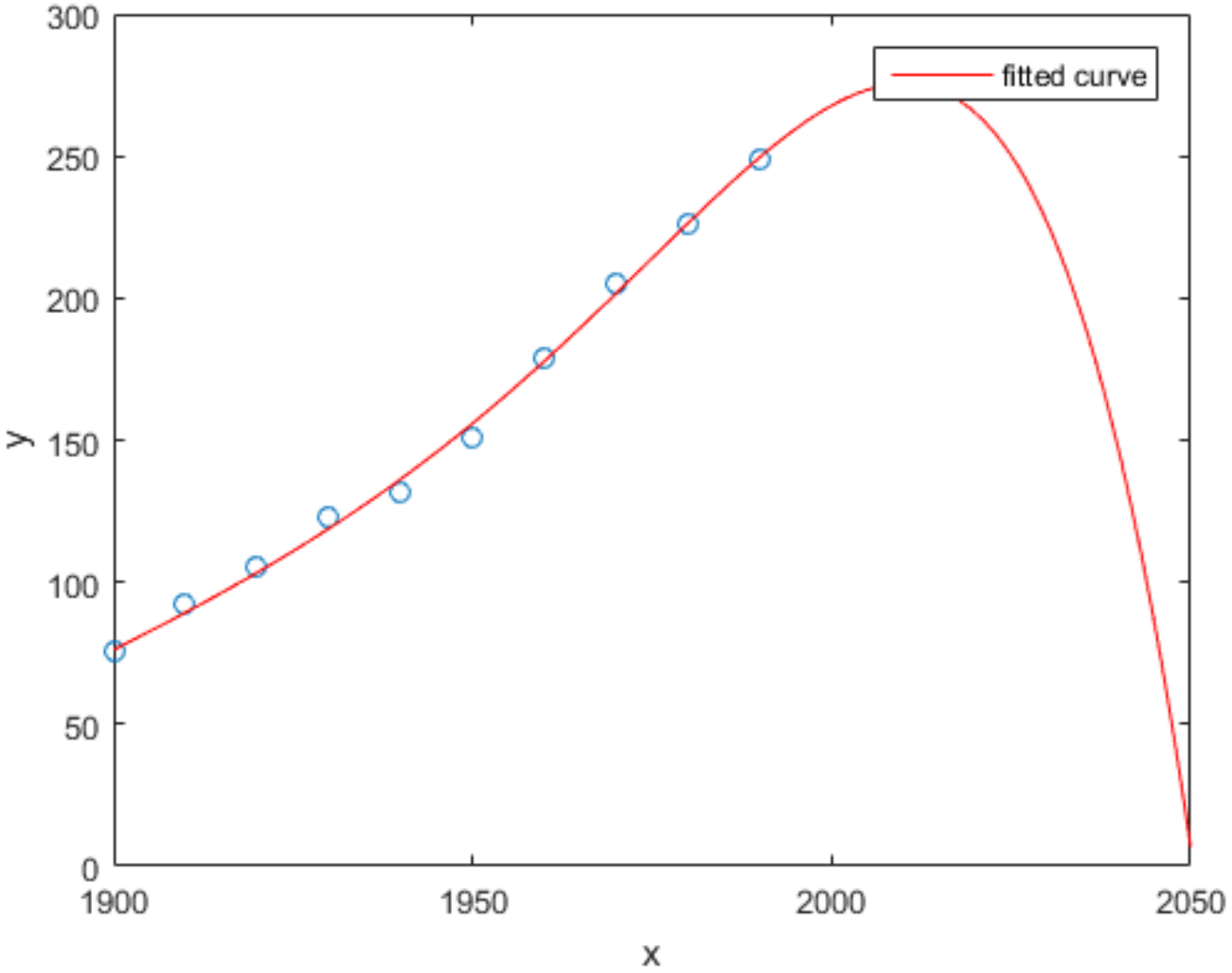

Degree 1
Samples
y
x


Degree 1
Model
True function
Samples
y
x
Degree 4
Model
True function
Samples
y
x
Degree 15
Model
True function
Samples
y
x


250
200
150
100
50
0
1750
1800
1850
1900
1950
2000


data
fitted curve
250
200
150
100
50
0
y
1750
1800
1850
1900
1950
2000
x


cdate v pop
poly2
poly3
poly4
poly5
poly6
exp1
400
300
200
100
0
-100
-200
y
1750
1800
1850
1900
1950
2000
x


300
250
200
150
100
50
0
y
1900
1950
2000
2050
x
fitted curve

1. **Из-за чего возникает переобучение?**
   — избыточная сложность пространства параметров Θ,
   лишние степени свободы в модели $g(x, \theta)$ «тратятся»
   на чрезмерно точную подгонку под обучающую выборку.
   — переобучение есть всегда, когда есть оптимизация
   параметров по конечной (заведомо неполной) выборке.

2. **Как обнаружить переобучение?**
   — эмпирически, путём разбиения выборки на train и test.

3. **Избавиться от него нельзя. Как его минимизировать?**
   — минимизировать одну из теоретических оценок;
   — накладывать ограничения на θ (регуляризация);
   — минимизировать HoldOut, LOO или CV, но осторожно!

- 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), $L = \ell + 1$:

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

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

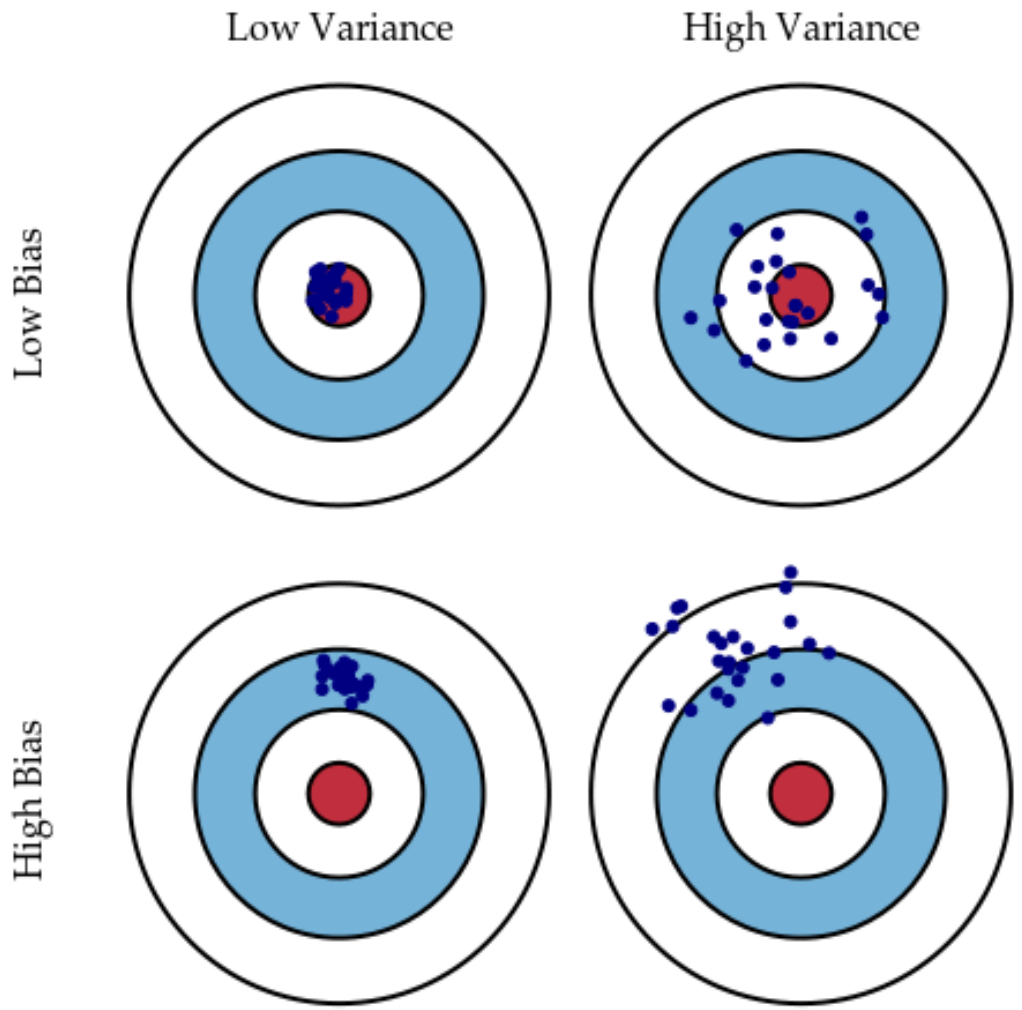
$$\mathrm{CV}(\mu, X^L) = \frac{1}{|N|}\sum_{n \in N} Q(\mu(X_n^\ell), X_n^k) \to \min$$

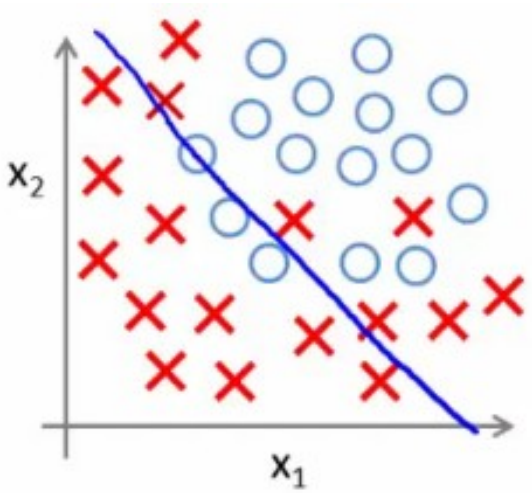
- Эмпирическая оценка вероятности переобучения:

$$Q_\varepsilon(\mu, X^L) = \frac{1}{|N|}\sum_{n \in N}\Big[Q\big(\mu(X_n^\ell), X_n^k\big) - Q\big(\mu(X_n^\ell), X_n^\ell\big) \geqslant \varepsilon\Big] \to \min$$

Bias–variance tradeoff

- The bias is error from erroneous assumptions in the learning algorithm. High bias can cause an algorithm to miss the relevant relations between features and target outputs (underfitting).
- The variance is error from sensitivity to small fluctuations in the training set. High variance can cause overfitting: modeling the random noise in the training data, rather than the intended outputs.

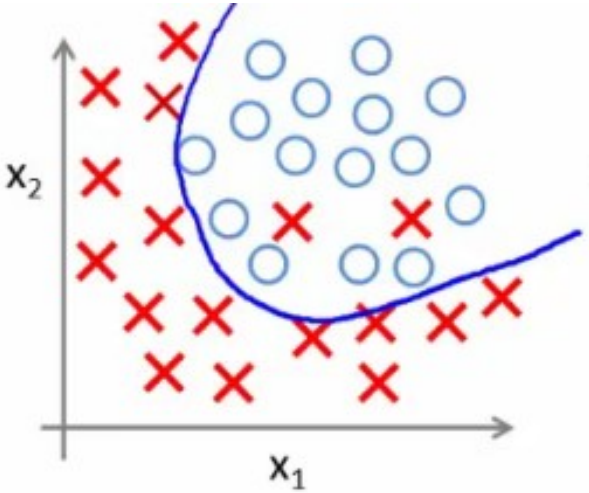

Low Variance
High Variance
Low Bias
High Bias



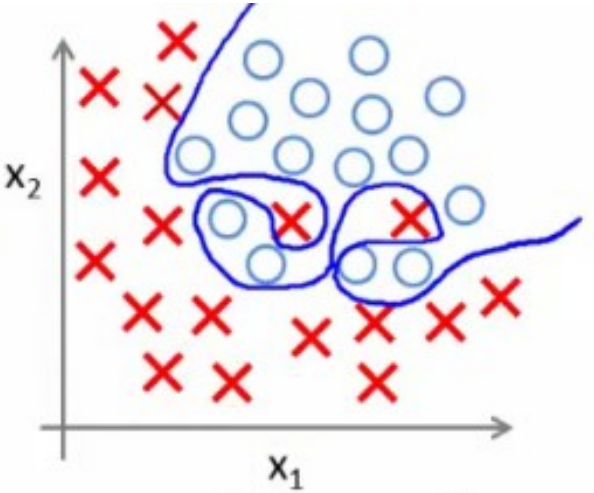
$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$

(g = sigmoid function)

UNDERFITTING
(high bias)

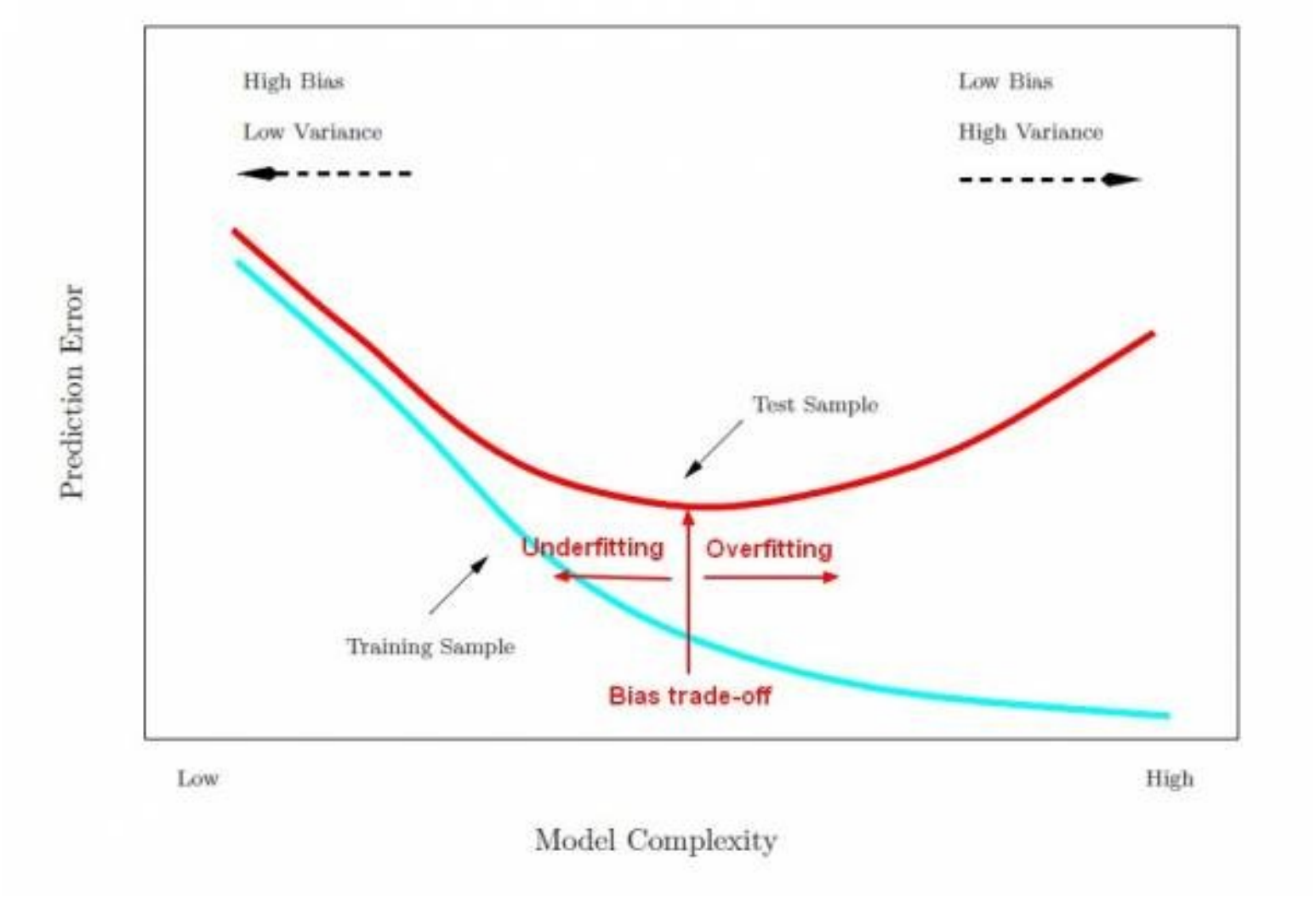


$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2)$



$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \ldots)$

OVERFITTING
(high variance)


High Bias
Low Variance
Low Bias
High Variance
Test Sample
Underfitting
Overfitting
Training Sample
Bias trade-off
Prediction Error
Low
High
Model Complexity

If we denote the variable we are trying to predict as Y and our covariates as X, we may assume that there is a relationship relating one to the other such as $Y = f(X) + \epsilon$ where the error term ϵ is normally distributed with a mean of zero like so $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon)$.

We may estimate a model $\hat{f}\,(X)$ of $f(X)$ using linear regressions or another modeling technique. In this case, the expected squared prediction error at a point x is:

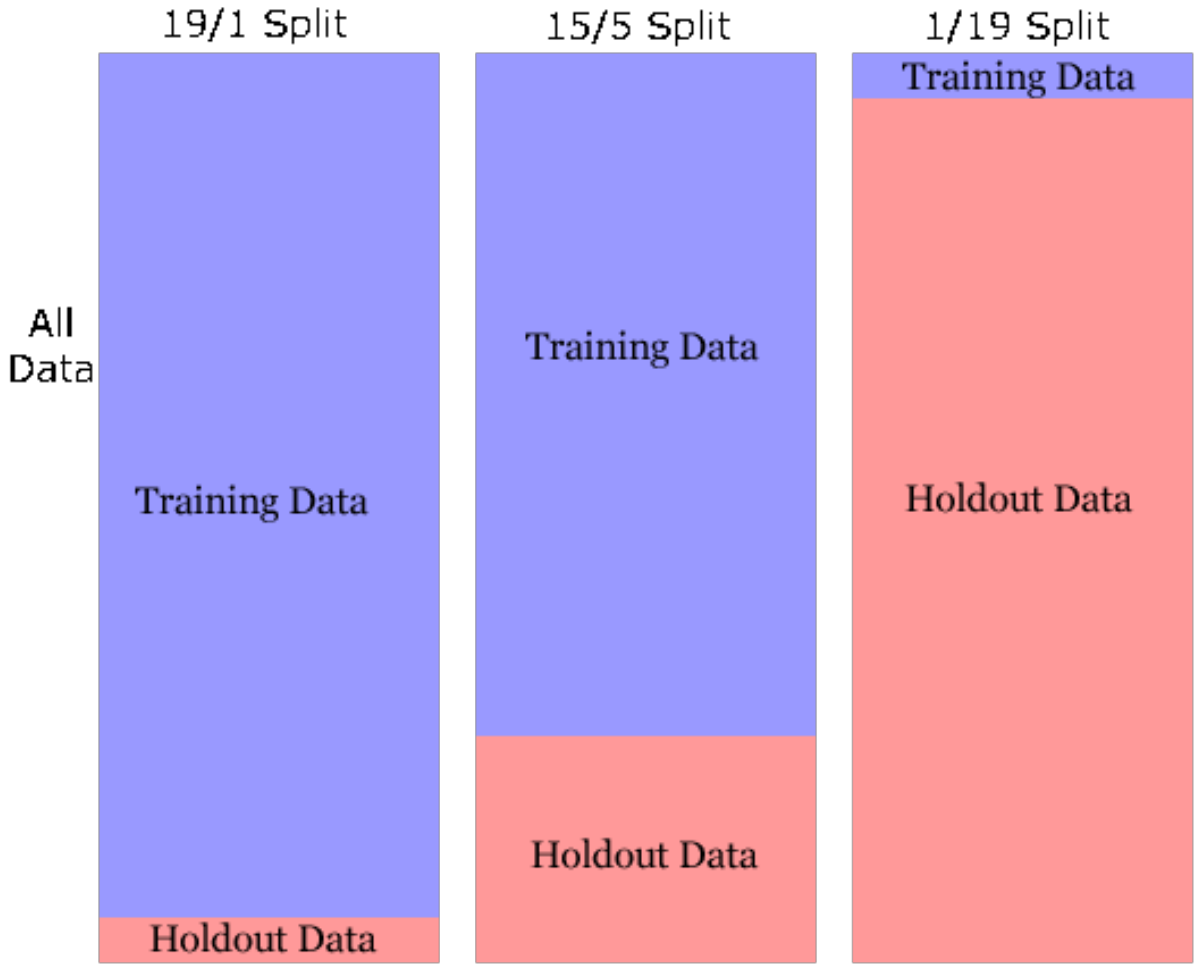
$$Err(x) = E\left[(Y - \hat{f}\,(x))^2\right]$$

This error may then be decomposed into bias and variance components:

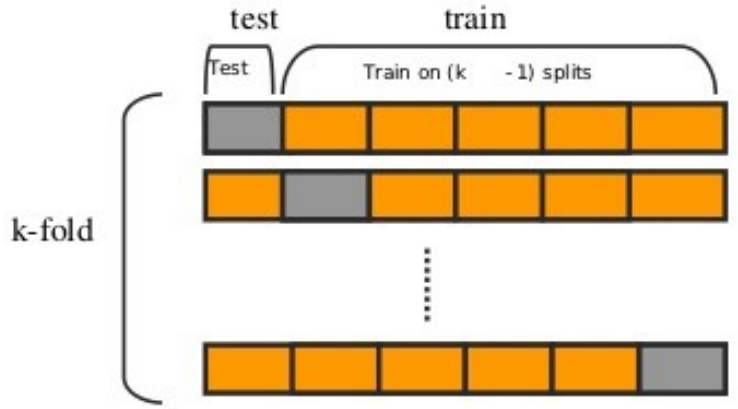
$$Err(x) = \Big(E[\hat{f}\,(x)] - f(x)\Big)^2 + E\Big[\hat{f}\,(x) - E[\hat{f}\,(x)]\Big]^2 + \sigma_e^2$$

$$Err(x) = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}$$

Hold-out validation



K-fold Cross Validation



- Randomly divide your data into K pieces/folds
- Treat 1st fold as the test dataset. Fit the model to the other folds (training data).
- Apply the model to the test data and repeat k times.
- Calculate statistics of model accuracy and fit from the test data only.

	True condition		
Total population	Condition positive	Condition negative	Prevalence $= \frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$
Predicted condition positive	**True positive**	**False positive** (Type I error)	Positive predictive value (PPV), Precision $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Test outcome positive}}$
Predicted condition negative	**False negative** (Type II error)	**True negative**	False omission rate (FOR) $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Test outcome negative}}$
Accuracy (ACC) $= \frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$	True positive rate (TPR), Sensitivity, Recall $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False positive rate (FPR), Fall-out $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	Positive likelihood ratio (LR+) $= \frac{\text{TPR}}{\text{FPR}}$
	False negative rate (FNR), Miss rate $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$	True negative rate (TNR), Specificity (SPC) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$	Negative likelihood ratio (LR−) $= \frac{\text{FNR}}{\text{TNR}}$

	Spam (Predicted)	Non-Spam (Predicted)	Accuracy
Spam (Actual)	27	6	81.81
Non-Spam (Actual)	10	57	85.07
Overall Accuracy			83.44

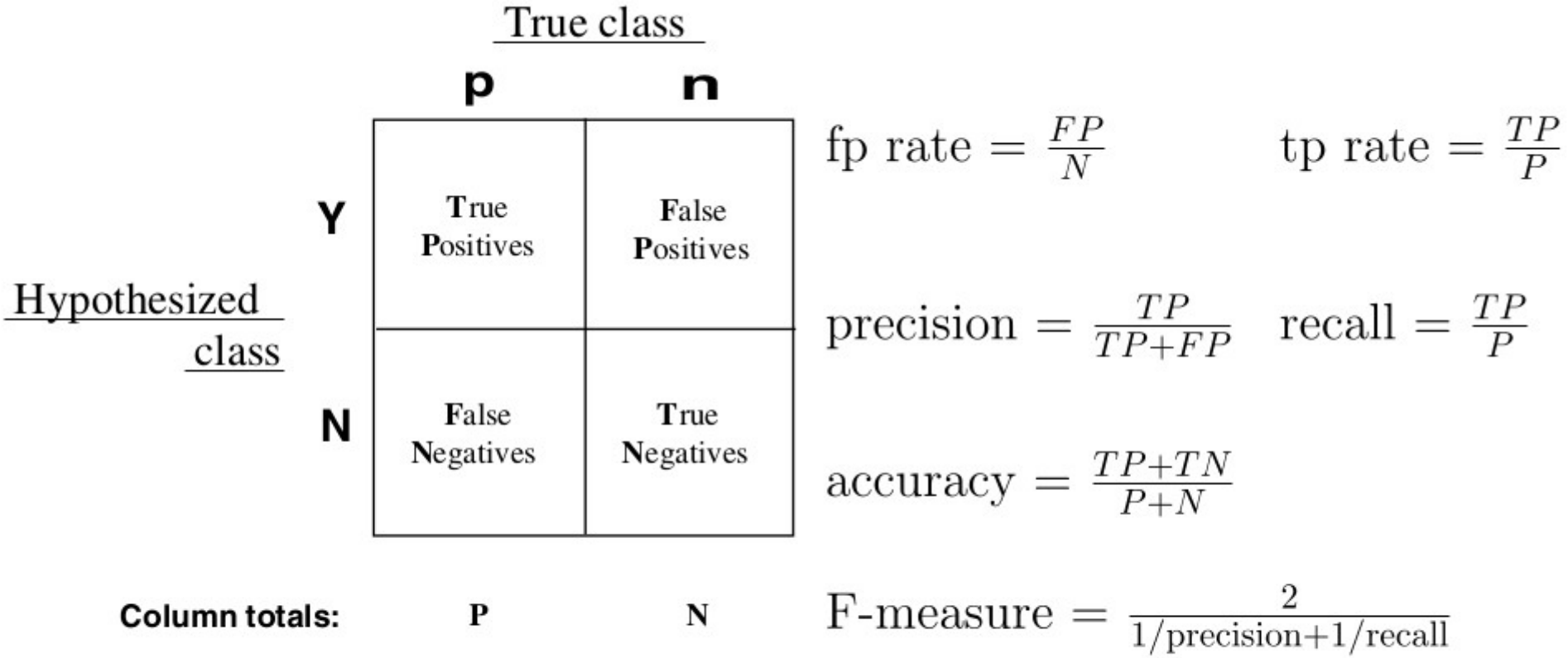
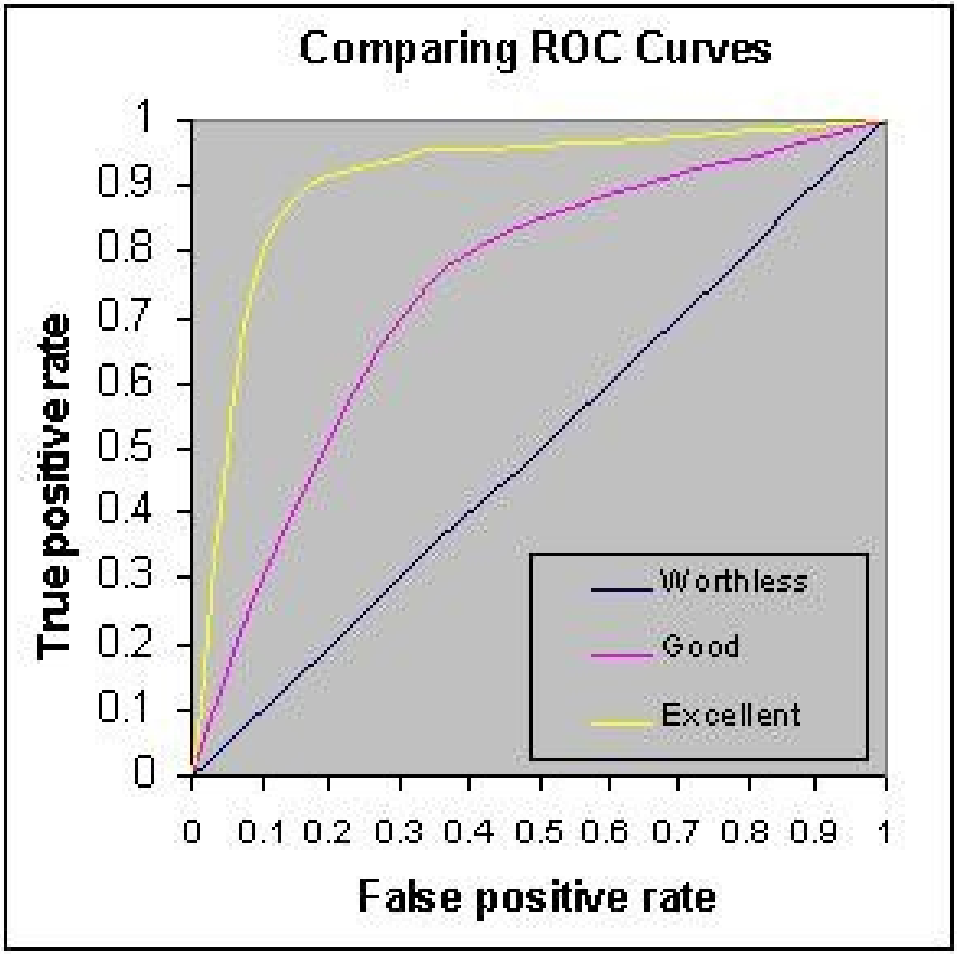


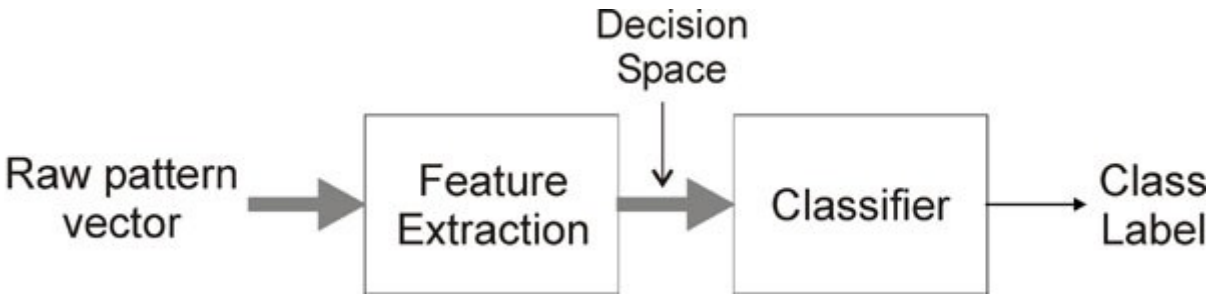
Fig. 1. Confusion matrix and common performance metrics calculated from it.


Comparing ROC Curves
True positive rate
False positive rate
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Worthless
Good
Excellent

- **Основные понятия машинного обучения:**
  объект, ответ, признак, алгоритм, модель алгоритмов, метод обучения, эмпирический риск, переобучение.

- **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;
  — внедрение и эксплуатация.

- **Прикладные задачи машинного обучения:**
  очень много, очень разных,
  во всех областях бизнеса, науки, производства.


Decision
Space
Raw pattern
vector
Feature
Extraction
Classifier
Class
Label