

# Метрические методы классификации и регрессии

Воронцов Константин Вячеславович

vokov@forecsys.ru

<http://www.MachineLearning.ru/wiki?title=User:Vokov>

Этот курс доступен на странице вики-ресурса

<http://www.MachineLearning.ru/wiki>

«Машинное обучение (курс лекций, К.В.Воронцов)»

Видеолекции: <http://shad.yandex.ru/lectures>

- 1 Определение расстояний между объектами**
  - Гипотезы компактности или непрерывности
  - Векторные меры близости
  - Беспознаковые способы вычисления расстояний
- 2 Метрические методы классификации**
  - Обобщённый метрический классификатор
  - От метода ближайшего соседа к потенциальным функциям
  - Задача отбора эталонных объектов
- 3 (Непара)метрические методы регрессии**
  - Формула Надарая–Ватсона
  - Выбор ядра  $K$  и ширины окна  $h$
  - Отсев выбросов

## Гипотезы непрерывности и компактности

**Задачи классификации и регрессии:**

$X$  — объекты,  $Y$  — ответы;

$X^\ell = (x_i, y_i)_{i=1}^\ell$  — обучающая выборка;

**Гипотеза непрерывности (для регрессии):**

*близким объектам соответствуют близкие ответы.*

выполнена:



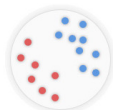
не выполнена:



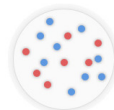
**Гипотеза компактности (для классификации):**

*близкие объекты, как правило, лежат в одном классе.*

выполнена:

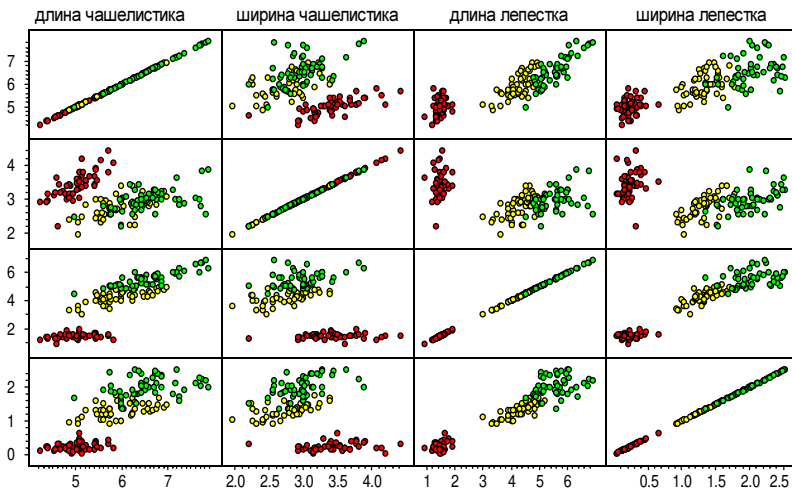


не выполнена:



## Пример: задача классификации цветков ириса [Фишер, 1936]

Привычная мера близости — евклидова метрика в  $\mathbb{R}^2$ .



## Формализация понятия «близости»

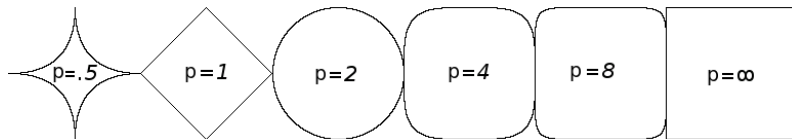
Евклидова метрика и обобщённая метрика Минковского:

$$\rho(x, x_i) = \left( \sum_{j=1}^n |x^j - x_i^j|^2 \right)^{1/2} \quad \rho(x, x_i) = \left( \sum_{j=1}^n w_j |x^j - x_i^j|^p \right)^{1/p}$$

$x = (x^1, \dots, x^n)$  — вектор признаков объекта  $x$ ,

$x_i = (x_i^1, \dots, x_i^n)$  — вектор признаков объекта  $x_i$ ,

$w_1, \dots, w_n$  — веса признаков, которые можно обучать.



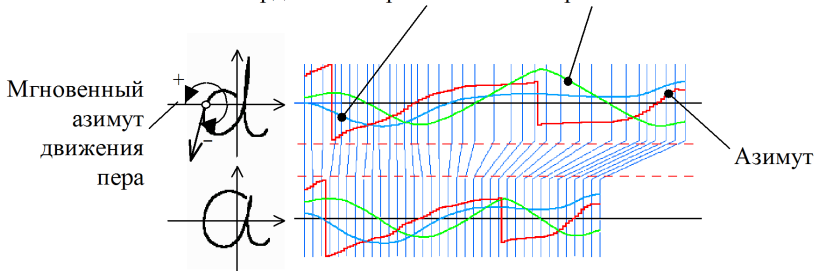
## Расстояния между строками / сигналами

Для строк — редакторское расстояние Левенштейна:

СТGGGCTAAAAGGTCCTTAGCC . . TTTAGAAAAA . GGGCCATTAGGAAATTGC  
СТGGGACTAAA . . . CCTTAGCCTATTTACAAAAATGGGCCATTAGG . . . TTGC

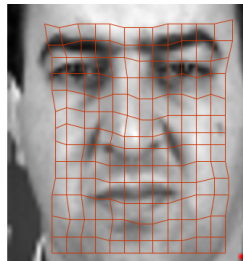
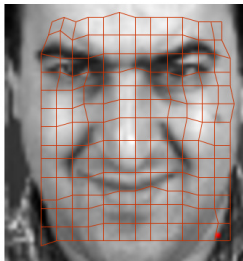
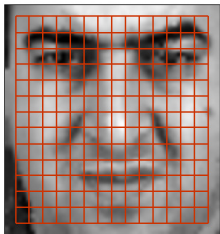
Для сигналов — энергия сжатий и растяжений:

Координаты: горизонтальная вертикальная



## Расстояния между изображениями

Расстояние между изображениями на основе выравнивания:



Оценивается энергия растяжения прямоугольной сетки

## Обобщённый метрический классификатор

Для произвольного  $x \in X$  отранжируем объекты  $x_1, \dots, x_\ell$ :

$$\rho(x, x^{(1)}) \leq \rho(x, x^{(2)}) \leq \dots \leq \rho(x, x^{(\ell)}),$$

$x^{(i)}$  —  $i$ -й сосед объекта  $x$  среди  $x_1, \dots, x_\ell$ ;

$y^{(i)}$  — ответ на  $i$ -м соседе объекта  $x$ .

Метрический алгоритм классификации:

$$a(x; X^\ell) = \arg \max_{y \in Y} \underbrace{\sum_{i=1}^{\ell} [y^{(i)} = y] w(i, x)}_{\Gamma_y(x)},$$

$w(i, x)$  — вес (степень важности)  $i$ -го соседа объекта  $x$ , неотрицателен, не возрастает по  $i$ .

$\Gamma_y(x)$  — оценка близости объекта  $x$  к классу  $y$ .



## Метод $k$ ближайших соседей ( $k$ nearest neighbors, $k$ NN)

$w(i, x) = [i \leq 1]$  — метод ближайшего соседа

$w(i, x) = [i \leq k]$  — метод  $k$  ближайших соседей

### Преимущества:

- простота реализации (lazy learning);
- параметр  $k$  можно оптимизировать по критерию скользящего контроля (leave-one-out):

$$\text{LOO}(k, X^\ell) = \sum_{i=1}^{\ell} [a(x_i; X^\ell \setminus \{x_i\}, k) \neq y_i] \rightarrow \min_k.$$

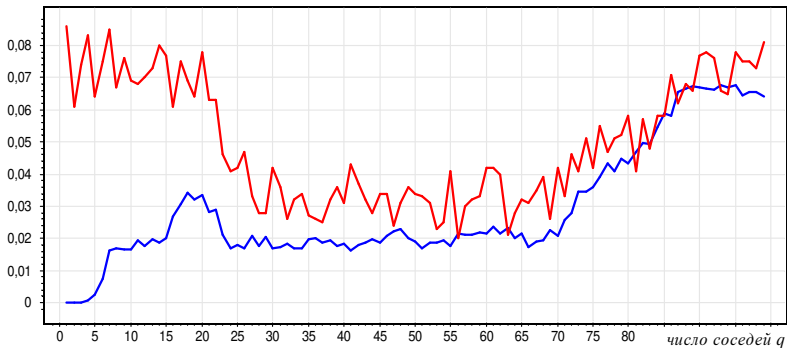
### Недостатки:

- неоднозначность классификации при  $\Gamma_y(x) = \Gamma_s(x)$ ,  $y \neq s$ .
- не учитываются значения расстояний

## Зависимость LOO от числа соседей

Пример. Задача UCI: Iris.

частота ошибок



- смещённое число ошибок, когда объект учитывается как сосед самого себя
- несмещённое число ошибок LOO

## Метод $k$ взвешенных ближайших соседей

$$w(i, x) = [i \leq k] w_i,$$

где  $w_i$  — вес, зависящий только от номера соседа;

**Возможные эвристики:**

$w_i = \frac{k+1-i}{k}$  — линейные убывающие веса;

$w_i = q^i$  — экспоненциально убывающие веса,  $0 < q < 1$ ;

**Проблемы:**

- как более обоснованно задать веса?
- возможно, было бы лучше, если бы вес  $w(i, x)$  зависел не от порядкового номера соседа  $i$ , а от расстояния до него  $\rho(x, x^{(i)})$ .

## Метод окна Парзена

$w(i, x) = K\left(\frac{\rho(x, x^{(i)})}{h}\right)$ , где  $h$  — ширина окна,  
 $K(r)$  — ядро, не возрастает и положительно на  $[0, 1]$ .

Метод парзеновского окна *фиксированной ширины*:

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

Метод парзеновского окна *переменной ширины*:

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

**Оптимизация параметров** — по критерию LOO:

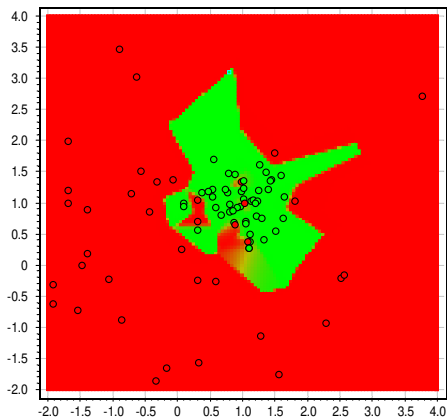
- выбор ширины окна  $h$  или числа соседей  $k$
- выбор ядра  $K$

## Парzenовское окно фиксированной ширины $h$

Пример: двумерная выборка, два класса  $Y = \{-1, +1\}$ .

$$a(x) = \arg \max_{y \in Y} \Gamma_y(x) = \text{sign}(\underbrace{\Gamma_{+1}(x) - \Gamma_{-1}(x)}})$$

$h = 0.05$

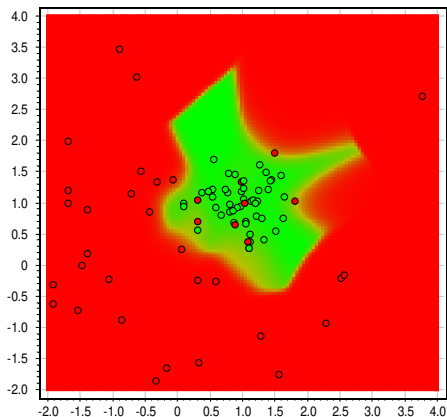


## Парzenовское окно фиксированной ширины $h$

Пример: двумерная выборка, два класса  $Y = \{-1, +1\}$ .

$$a(x) = \arg \max_{y \in Y} \Gamma_y(x) = \text{sign}(\underbrace{\Gamma_{+1}(x) - \Gamma_{-1}(x)}})$$

$h = 0.2$

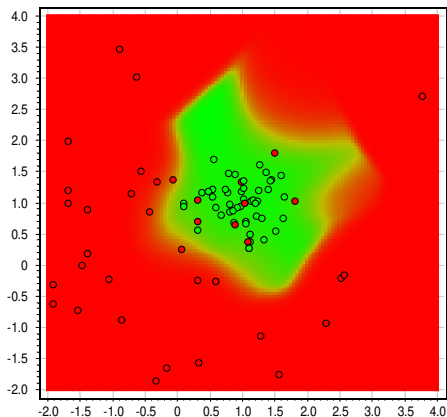


## Парzenовское окно фиксированной ширины $h$

Пример: двумерная выборка, два класса  $Y = \{-1, +1\}$ .

$$a(x) = \arg \max_{y \in Y} \Gamma_y(x) = \text{sign}(\underbrace{\Gamma_{+1}(x) - \Gamma_{-1}(x)}})$$

$h = 0.3$

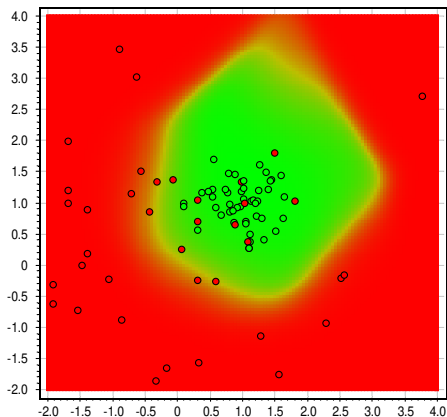


## Парzenовское окно фиксированной ширины $h$

Пример: двумерная выборка, два класса  $Y = \{-1, +1\}$ .

$$a(x) = \arg \max_{y \in Y} \Gamma_y(x) = \text{sign}(\underbrace{\Gamma_{+1}(x) - \Gamma_{-1}(x)}})$$

$h = 0.5$



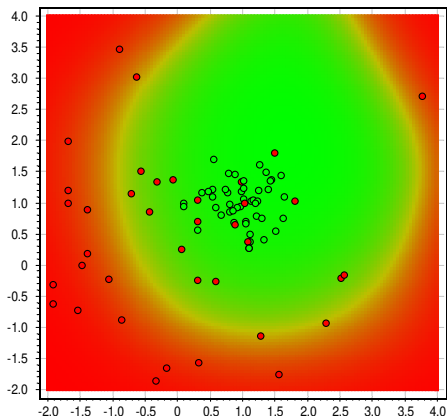


## Парzenовское окно фиксированной ширины $h$

Пример: двумерная выборка, два класса  $Y = \{-1, +1\}$ .

$$a(x) = \arg \max_{y \in Y} \Gamma_y(x) = \text{sign}(\underbrace{\Gamma_{+1}(x) - \Gamma_{-1}(x)}})$$

$h = 1.0$

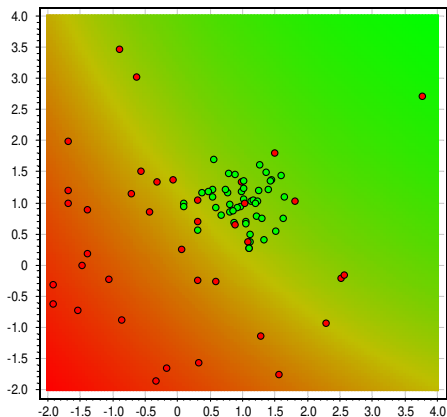


## Парzenовское окно фиксированной ширины $h$

Пример: двумерная выборка, два класса  $Y = \{-1, +1\}$ .

$$a(x) = \arg \max_{y \in Y} \Gamma_y(x) = \text{sign}(\underbrace{\Gamma_{+1}(x) - \Gamma_{-1}(x)}})$$

$h = 5.0$



## Метод потенциальных функций

$$w(i, x) = \gamma^{(i)} K\left(\frac{\rho(x, x^{(i)})}{h^{(i)}}\right)$$

Более простая запись (здесь можно не ранжировать объекты):

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

где  $\gamma_i$  — веса объектов,  $\gamma_i \geq 0$ ,  $h_i > 0$ .

**Физическая аналогия** из электростатики:

$\gamma_i$  — величина «заряда» в точке  $x_i$ ;

$h_i$  — «радиус действия» потенциала с центром в точке  $x_i$ ;

$y_i$  — знак «заряда» (в случае двух классов  $Y = \{-1, +1\}$ );

$K(r) = \frac{1}{r}$  или  $\frac{1}{r+a}$

В задачах классификации нет ограничений ни на  $K$ , ни на  $|Y|$ .

## Метод потенциальных функций = линейный классификатор

Два класса:  $Y = \{-1, +1\}$ .

$$\begin{aligned} a(x; X^\ell) &= \arg \max_{y \in Y} \Gamma_y(x) = \text{sign}(\Gamma_{+1}(x) - \Gamma_{-1}(x)) = \\ &= \text{sign} \sum_{i=1}^{\ell} \gamma_i y_i K\left(\frac{\rho(x, x_i)}{h_i}\right). \end{aligned}$$

Сравним с линейной моделью классификации:

$$a(x) = \text{sign} \sum_{j=1}^n \gamma_j f_j(x).$$

- $f_j(x) = y_j K\left(\frac{1}{h_j} \rho(x, x_j)\right)$  — новые признаки объекта  $x$
- $\gamma_j$  — веса линейного классификатора
- $n = \ell$  — число признаков равно числу объектов обучения

## Полный скользящий контроль ССВ

Функционал *полного* скользящего контроля  
(complete cross-validation, ССВ):

$$\text{CCV}(X^L) = \frac{1}{C_L} \sum_{X^\ell \sqcup X^k} \frac{1}{k} \sum_{x_i \in X^k} [a(x_i, X^\ell) \neq y_i],$$

где  $X^\ell \sqcup X^k$  — все  $C_L^k$  разбиений выборки  $X^L$  на обучающую подвыборку  $X^\ell$  и контрольную  $X^k$ .

**Замечание 1.** При  $k = 1$  имеем:  $\text{CCV}(X^L) = \text{LOO}(X^L)$ .

**Замечание 2.** ССВ характеризует лишь среднюю частоту ошибок, но не учитывает её разброс.

## Понятие профиля компактности

### Определение

Профиль компактности выборки  $X^L$  — это функция доли объектов  $x_i$ , у которых  $m$ -й сосед  $x_i^{(m)}$  лежит в другом классе:

$$K(m, X^L) = \frac{1}{L} \sum_{i=1}^L [y_i \neq y_i^{(m)}]; \quad m = 1, \dots, L-1,$$

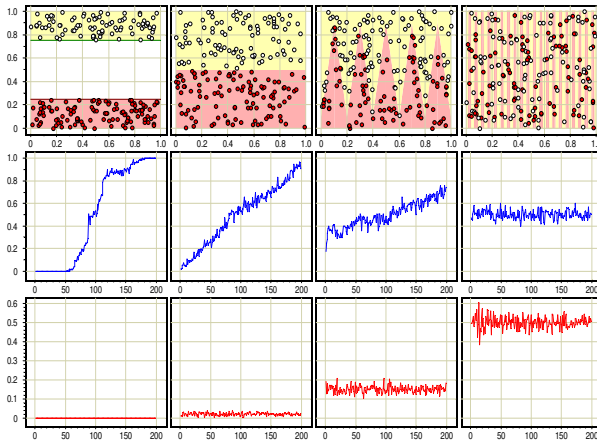
где  $x_i^{(m)}$  —  $m$ -й сосед объекта  $x_i$  среди  $X^L$ ;

$y_i^{(m)}$  — ответ на  $m$ -м соседе объекта  $x_i$ .

### Теорема (точное выражение CCV для метода 1NN)

$$\text{CCV}(X^L) = \sum_{m=1}^k K(m, X^L) \frac{C_{L-1-m}^{\ell-1}}{C_{L-1}^{\ell}}.$$

## Профили компактности для серии модельных задач



средний ряд: профили компактности,

нижний ряд: зависимость CCV от длины контроля  $k$ .

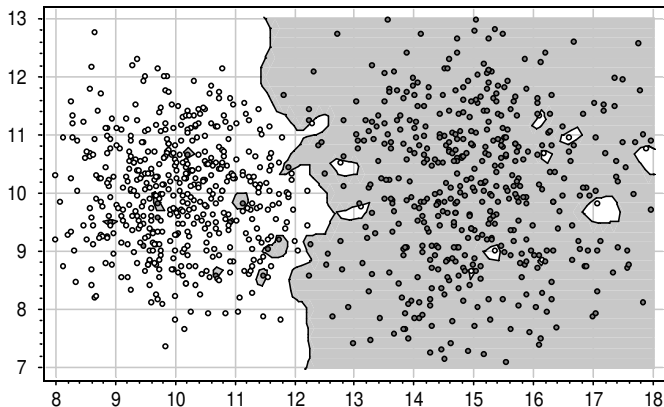
## Свойства профиля компактности и оценки CCV

$$\text{CCV}(X^L) = \sum_{m=1}^k K(m, X^L) \frac{C_{L-1-m}^{\ell-1}}{C_{L-1}^{\ell}}.$$

- $K(m, X^L)$  формализует гипотезу компактности, связывая свойства выборки с качеством классификации.
- CCV практически не зависит от длины контроля  $k$ .
- Для минимизации CCV важен только начальный участок профиля, т. к.  $\frac{C_{L-1-m}^{\ell-1}}{C_{L-1}^{\ell}} \rightarrow 0$  экспоненциально по  $m$ .
- Минимизация CCV позволяет делать отбор эталонов.

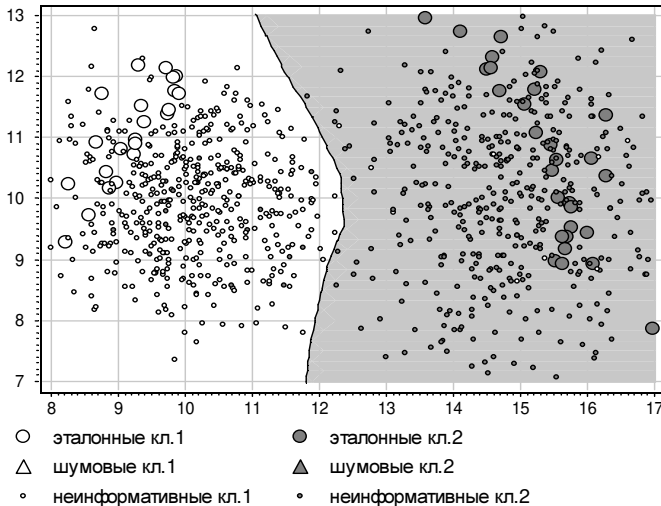


## Модельные данные

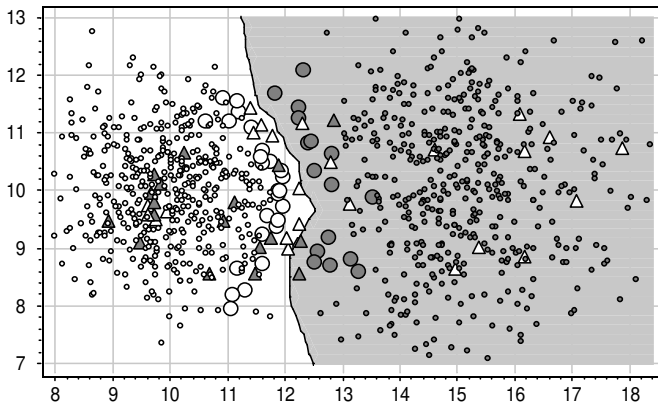


Модельная задача классификации: 1000 объектов.  
Алгоритм 1NN

## Последовательное добавление эталонных объектов



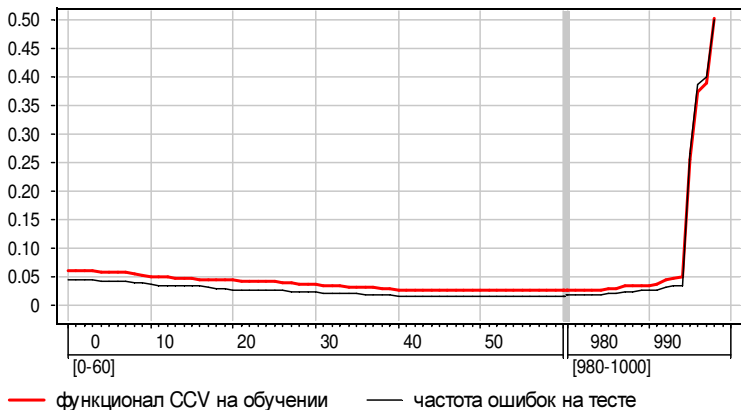
## Последовательный отсев не-эталонных объектов



- |                        |                        |
|------------------------|------------------------|
| ○ эталонные кл.1       | ● эталонные кл.2       |
| △ шумовые кл.1         | ▲ шумовые кл.2         |
| ◦ неинформативные кл.1 | ◦ неинформативные кл.2 |

## Последовательный отсев не-эталонных объектов

Зависимость CCV от числа удаленных неэталонных объектов.



При отборе эталонов по критерию CCV переобучения нет.

## Резюме

- Метрические классификаторы — одни из самых простых. Качество классификации определяется качеством метрики.
- Что можно обучать:
  - число ближайших соседей  $k$  или ширину окна  $h$ ;
  - веса объектов;
  - набор эталонов (prototype selection);
  - метрику (distance learning, similarity learning);
  - в частности, веса признаков в метрике;
  - функцию ядра  $K(r)$ .

## Задачи регрессии и метод наименьших квадратов

- $X$  — объекты (часто  $\mathbb{R}^n$ );  $Y$  — ответы (часто  $\mathbb{R}$ , реже  $\mathbb{R}^m$ );  
 $X^\ell = (x_i, y_i)_{i=1}^\ell$  — обучающая выборка;  
 $y_i = y(x_i)$ ,  $y: X \rightarrow Y$  — неизвестная зависимость;
- $a(x) = f(x, \alpha)$  — параметрическая модель зависимости,  
 $\alpha \in \mathbb{R}^p$  — вектор параметров модели.

- Метод наименьших квадратов (МНК):

$$Q(\alpha, X^\ell) = \sum_{i=1}^{\ell} w_i (f(x_i, \alpha) - y_i)^2 \rightarrow \min_{\alpha},$$

где  $w_i$  — вес, степень важности  $i$ -го объекта.

- **Недостаток:**

надо иметь хорошую параметрическую модель  $f(x, \alpha)$

## Непараметрическая регрессия, формула Надарая–Ватсона

Приближение константой  $f(x, \alpha) = \alpha$  в окрестности  $x \in X$ :

$$Q(\alpha; X^\ell) = \sum_{i=1}^{\ell} w_i(x) (\alpha - y_i)^2 \rightarrow \min_{\alpha \in \mathbb{R}}$$

где  $w_i(x) = K\left(\frac{\rho(x, x_i)}{h}\right)$  — веса объектов  $x_i$  относительно  $x$ ;  
 $K(r)$  — ядро, невозрастающее, ограниченное, гладкое;  
 $h$  — ширина окна сглаживания.

### Формула ядерного сглаживания Надарая–Ватсона:

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

## Обоснование формулы Надарая–Ватсона

### Теорема

Пусть выполнены следующие условия:

- 1) выборка  $X^\ell = (x_i, y_i)_{i=1}^\ell$  простая, из распределения  $p(x, y)$ ;
- 2) ядро  $K(r)$  ограничено:  $\int_0^\infty K(r) dr < \infty$ ,  $\lim_{r \rightarrow \infty} rK(r) = 0$ ;
- 3) зависимость  $E(y|x)$  не имеет вертикальных асимптот:  
 $E(y^2|x) = \int_Y y^2 p(y|x) dy < \infty$  при любом  $x \in X$ ;
- 4) последовательность  $h_\ell$  убывает, но не слишком быстро:  
 $\lim_{\ell \rightarrow \infty} h_\ell = 0$ ,  $\lim_{\ell \rightarrow \infty} \ell h_\ell = \infty$ .

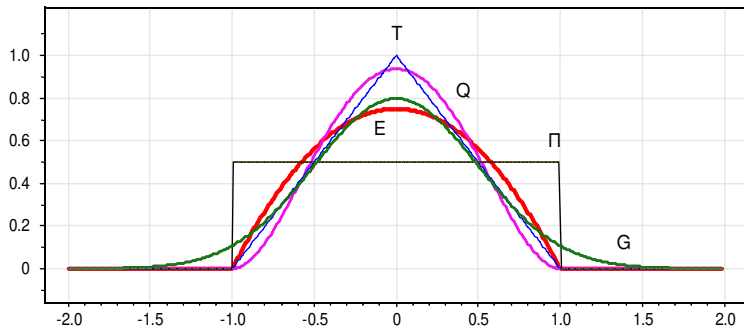
Тогда имеет место сходимость по вероятности:

$$a_{h_\ell}(x; X^\ell) \xrightarrow{P} E(y|x) \text{ в любой точке } x \in X,$$

в которой  $E(y|x)$ ,  $p(x)$  и  $D(y|x)$  непрерывны и  $p(x) > 0$ .



## Часто используемые ядра $K(r)$



$P(r) = [ |r| \leq 1 ]$  — прямоугольное

$T(r) = (1 - |r|) [ |r| \leq 1 ]$  — треугольное

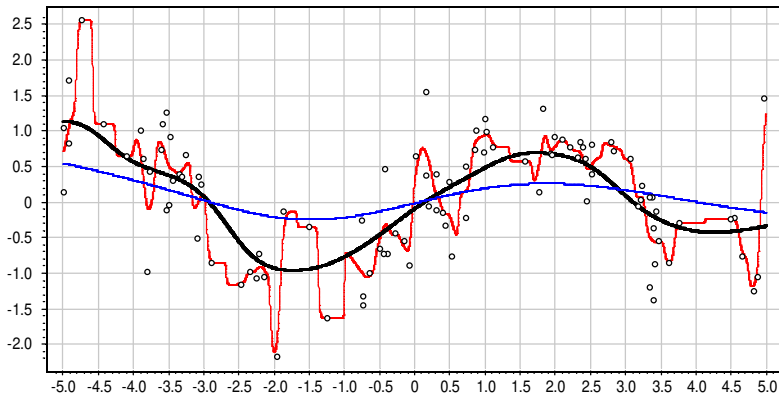
$E(r) = (1 - r^2) [ |r| \leq 1 ]$  — квадратичное (Епанечникова)

$Q(r) = (1 - r^2)^2 [ |r| \leq 1 ]$  — четвертое

$G(r) = \exp(-2r^2)$  — гауссовское

## Выбор ядра $K$ и ширины окна $h$

$h \in \{0.1, 1.0, 3.0\}$ , гауссовское ядро  $K(r) = \exp(-2r^2)$

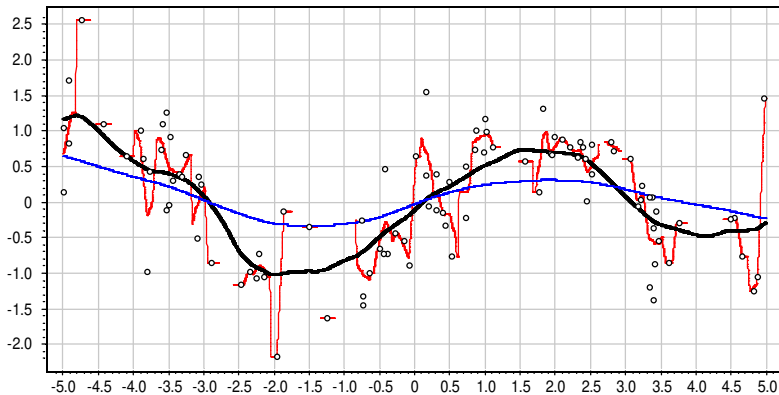


Гауссовское ядро  $\Rightarrow$  гладкая аппроксимация

Ширина окна существенно влияет на точность аппроксимации

## Выбор ядра $K$ и ширины окна $h$

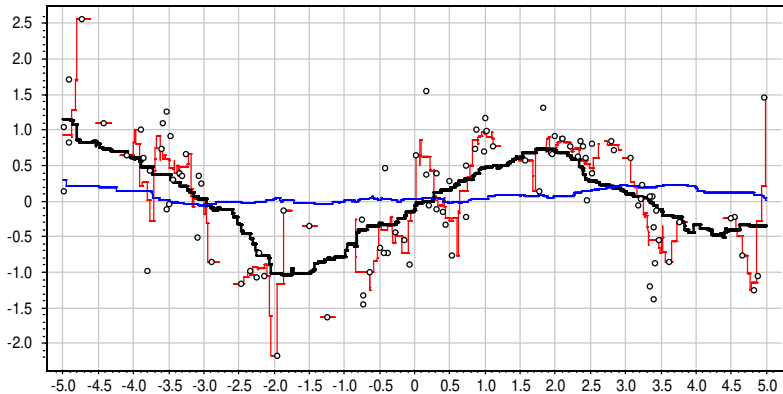
$h \in \{0.1, 1.0, 3.0\}$ , треугольное ядро  $K(r) = (1 - |r|) [|r| \leq 1]$



Треугольное ядро  $\Rightarrow$  кусочно-линейная аппроксимация  
Аппроксимация не определена, если в окне нет точек выборки

## Выбор ядра $K$ и ширины окна $h$

$h \in \{0.1, 1.0, 3.0\}$ , прямоугольное ядро  $K(r) = [|r| \leq 1]$



Прямоугольное ядро  $\Rightarrow$  кусочно-постоянная аппроксимация  
Выбор ядра слабо влияет на точность аппроксимации

## Выбор ядра $K$ и ширины окна $h$

- Ядро  $K(r)$ 
  - существенно влияет на гладкость функции  $a_h(x)$ ,
  - слабо влияет на качество аппроксимации.
- Ширина окна  $h$ 
  - существенно влияет на качество аппроксимации.
- Переменная ширина окна по  $k$  ближайшим соседям:

$$w_i(x) = K\left(\frac{\rho(x, x_i)}{h(x)}\right), \quad h(x) = \rho(x, x^{(k+1)})$$

где  $x^{(k)}$  —  $k$ -й сосед объекта  $x$ .

- Оптимизация ширины окна по скользящему контролю:

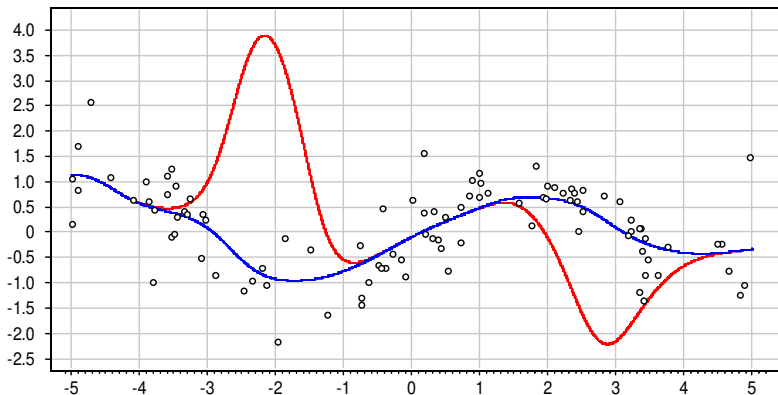
$$\text{LOO}(h, X^\ell) = \sum_{i=1}^{\ell} \left( a_h(x_i; X^\ell \setminus \{x_i\}) - y_i \right)^2 \rightarrow \min_h.$$

## Проблема выбросов (эксперимент на синтетических данных)

$\ell = 100$ ,  $h = 1.0$ , гауссовское ядро  $K(r) = \exp(-2r^2)$

Две из 100 точек — выбросы с ординатами  $y_i = 40$  и  $-40$

Синяя кривая — выбросов нет



## Проблема выбросов и локально взвешенное сглаживание

**Проблема выбросов:** точки с большими случайными ошибками  $y_i$  сильно искажают функцию  $a_h(x)$

**Основная идея:**

чем больше величина ошибки  $\varepsilon_i = |a_h(x_i; X^\ell \setminus \{x_i\}) - y_i|$ ,  
тем больше прецедент  $(x_i, y_i)$  похож на выброс,  
тем меньше должен быть его вес  $w_i(x)$ .

**Эвристика:**

домножить веса  $w_i(x)$  на коэффициенты  $\gamma_i = \tilde{K}(\varepsilon_i)$ ,  
где  $\tilde{K}$  — ещё одно ядро, вообще говоря, отличное от  $K(r)$ .

**Рекомендация:**

квартичное ядро  $\tilde{K}(\varepsilon) = K_Q\left(\frac{\varepsilon}{6 \operatorname{med}\{\varepsilon_i\}}\right)$ ,  
где  $\operatorname{med}\{\varepsilon_i\}$  — медиана вариационного ряда ошибок.

## Алгоритм LOWESS (LOcally WEighted Scatter plot Smoothing)

**Вход:**  $X^\ell$  — обучающая выборка;

**Выход:** коэффициенты  $\gamma_i$ ,  $i = 1, \dots, \ell$ ;

1: инициализация:  $\gamma_i := 1$ ,  $i = 1, \dots, \ell$ ;

2: **повторять**

3: **для всех** объектов  $i = 1, \dots, \ell$

4: вычислить оценки скользящего контроля:

$$a_i := a_h(x_i; X^\ell \setminus \{x_i\}) = \frac{\sum_{j=1, j \neq i}^{\ell} y_j \gamma_j K\left(\frac{\rho(x_i, x_j)}{h(x_i)}\right)}{\sum_{j=1, j \neq i}^{\ell} \gamma_j K\left(\frac{\rho(x_i, x_j)}{h(x_i)}\right)}$$

5: **для всех** объектов  $i = 1, \dots, \ell$

6:  $\gamma_i := \tilde{K}(|a_i - y_i|)$ ;

7: **пока** коэффициенты  $\gamma_i$  не стабилизируются;

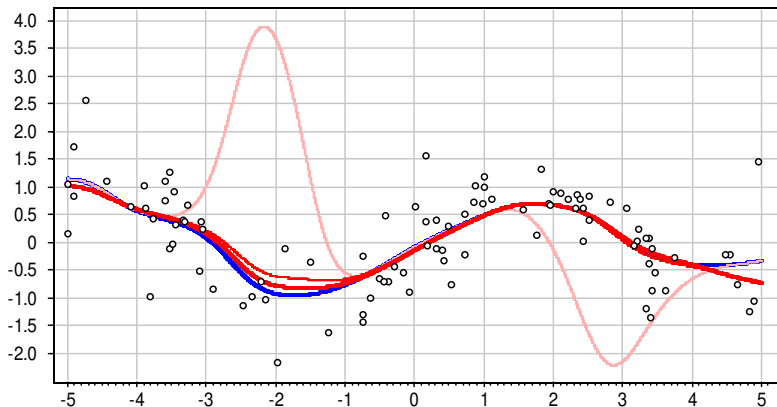


## Пример работы LOWESS на синтетических данных

$\ell = 100$ ,  $h = 1.0$ , гауссовское ядро  $K(r) = \exp(-2r^2)$

Две из 100 точек — выбросы с ординатами  $y_i = 40$  и  $-40$

В данном случае LOWESS сходится за несколько итераций:



## Резюме

- Непараметрическая регрессия обходится без явного задания параметрической модели зависимости  $f(x, \alpha)$ .
- Однако неявно модельные предположения закладываются в функцию расстояния  $\rho(x, x_i)$  между объектами.
- Что можно обучать:
  - число ближайших соседей  $k$  или ширину окна  $h$ ;
  - веса объектов (обнаруживать выбросы);
  - метрику (distance learning, similarity learning);
  - в частности, веса признаков в метрике.
- Непараметрическая регрессия часто используется как инструмент предварительной обработки данных для сглаживания шумов в данных.