Методы машинного обучения
Метрические методы классификации и регрессии

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-24-25
орг.вопросы по курсу: ml.cmc@mail.ru

ВМК МГУ • 10 сентября 2024

Содержание

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):

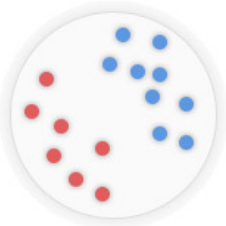
близким объектам соответствуют близкие ответы.

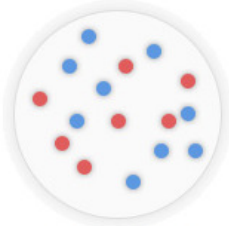
выполнена: 

не выполнена: 

Гипотеза компактности (для классификации):

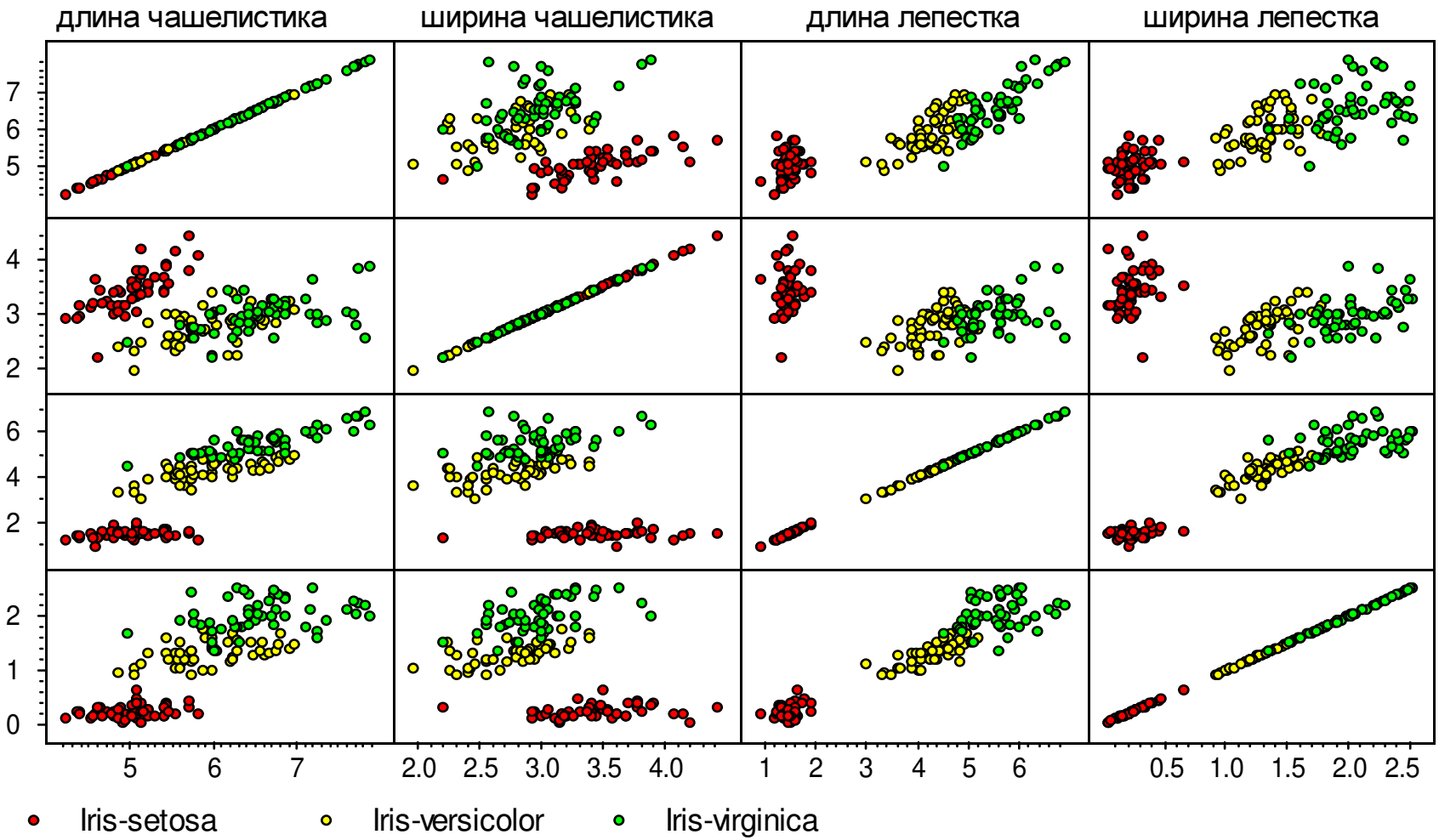
близкие объекты, как правило, лежат в одном классе.

выполнена: 

не выполнена: 

Пример: задача классификации цветков ириса [Фишер, 1936]

Классы — компактные сгустки точек



Формализация понятия «расстояние» (distance)

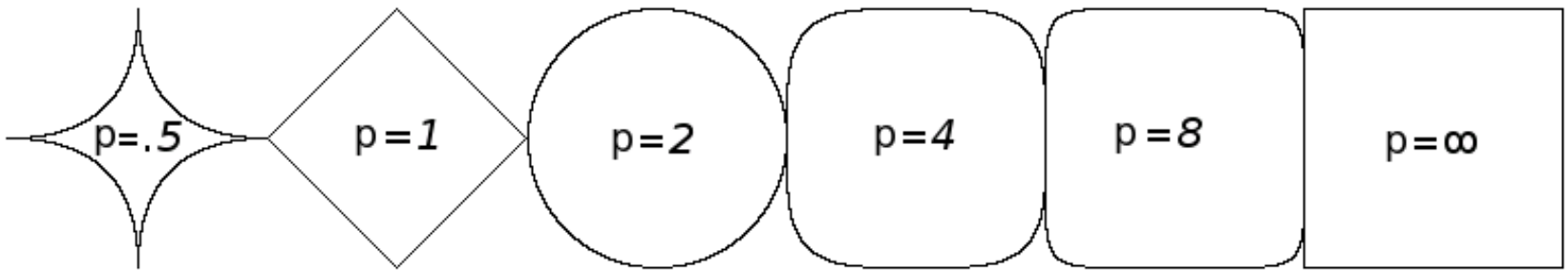
Евклидова метрика и *обобщённая метрика Минковского*:

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

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

$w_1, \ldots, w_n$ — обучаемые веса признаков, играющие две роли:
— нормировка, приведение к общему масштабу
— задание степени важности (информативности) признаков

Линии уровня (*эквидистантные поверхности*) при различных p:

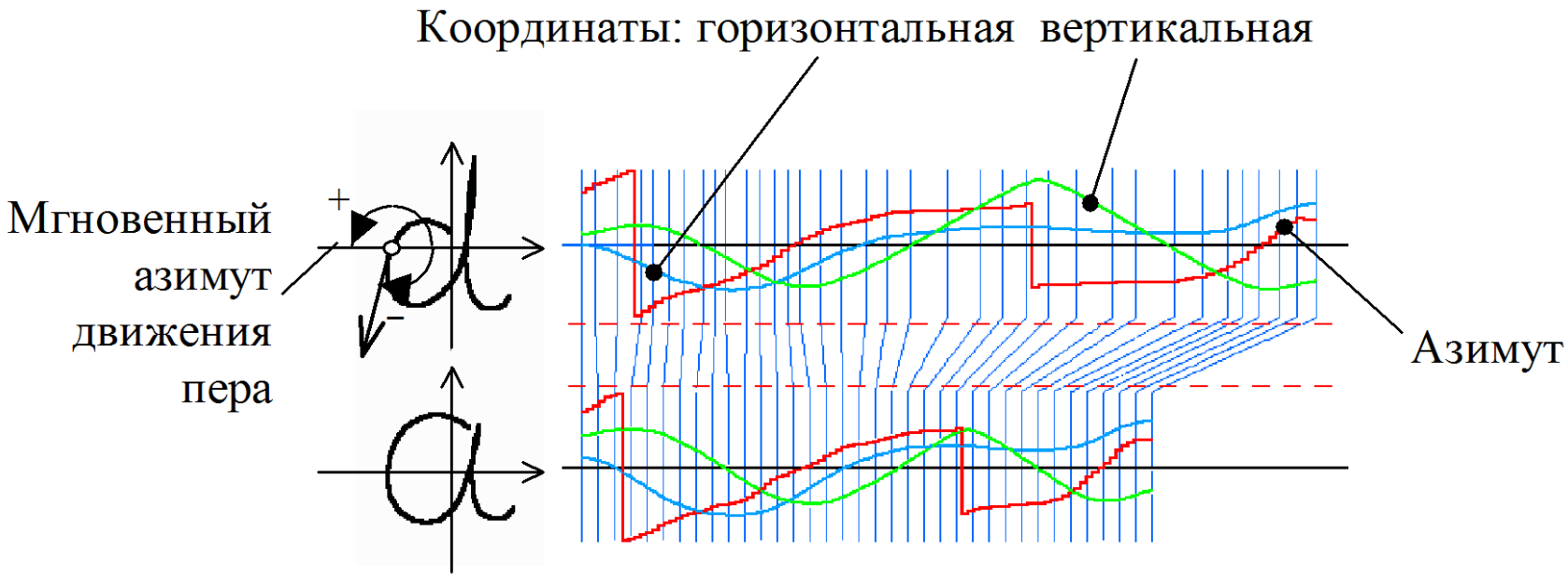


Расстояния между строками / сигналами

Для строк — редакторское расстояние Левенштейна:

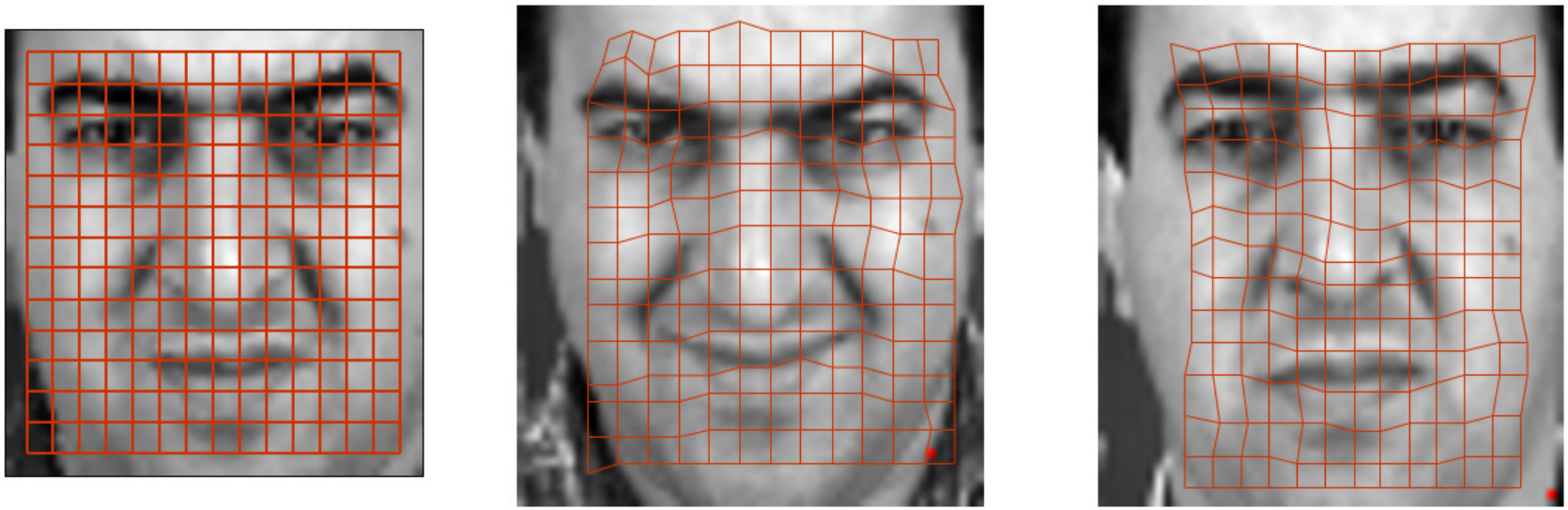
```
CTGGGCTAAAAGGTCCCTTAGCC..TTTAGAAAAA.GGGCCATTAGGAAATTGC
CTGGGACTAAA....CCTTAGCCTATTTACAAAAATGGGCCATTAGG...TTGC
```

Для сигналов — энергия сжатий и растяжений:



Расстояния между изображениями

Расстояние **между изображениями** на основе выравнивания:



Оценивается энергия растяжения прямоугольной сетки

Обобщённый метрический классификатор

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

$$\rho(x, x^{(1)}) \leqslant \rho(x, x^{(2)}) \leqslant \cdots \leqslant \rho(x, x^{(\ell)}),$$

$x^{(i)}$ — i-й сосед объекта x среди $x_1, \ldots, x_\ell$;
$y^{(i)}$ — ответ на i-м соседе объекта x.

Метрический алгоритм классификации относит объект x к тому классу, которому принадлежат его ближайшие соседи:

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

$w(i, x)$ — вес, *степень близости* к объекту x его i-го соседа, неотрицателен, не возрастает по i.
$\Gamma_y(x)$ — *оценка близости* объекта x к классу y.

Метод k ближайших соседей (k nearest neighbors, kNN)

$w(i,x) = [i \leqslant 1]$ — метод ближайшего соседа
$w(i,x) = [i \leqslant k]$ — метод k ближайших соседей

Преимущества:

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

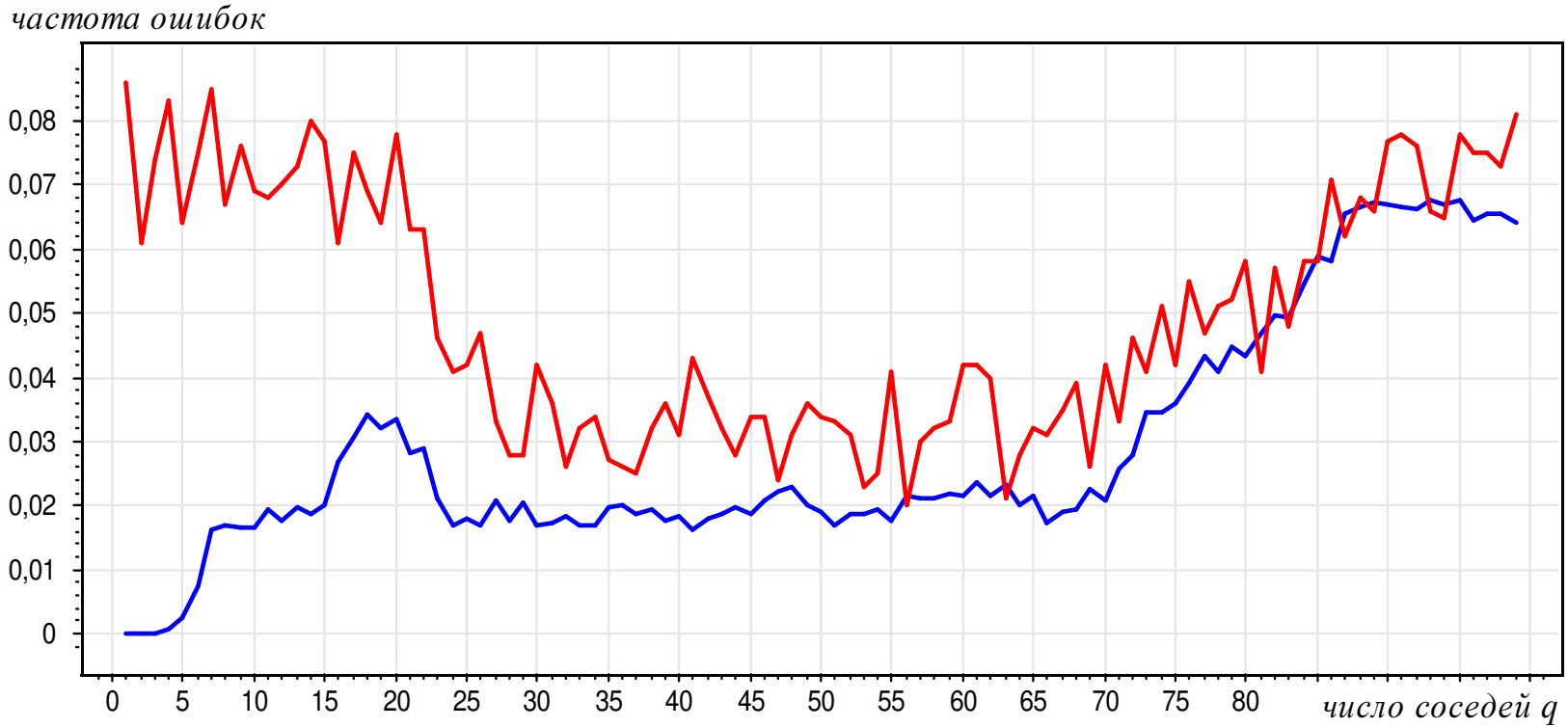
$$\mathrm{LOO}(k, X^\ell) = \sum_{i=1}^{\ell} \Big[a\big(x_i; X^\ell \setminus \{x_i\}, k\big) \neq y_i \Big] \to \min_k .$$

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.



— смещённое число ошибок, когда объект учитывается как сосед самого себя

— несмещённое число ошибок LOO

Метод k взвешенных ближайших соседей

$w(i,x) = [i \leqslant k]w_i$,
где w_i — вес, зависящий только от номера соседа;

Возможные эвристики:

$w_i = \frac{k+1-i}{k}$ — линейное 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 (bandwidth),
$K(r)$ — ядро (kernel), 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:

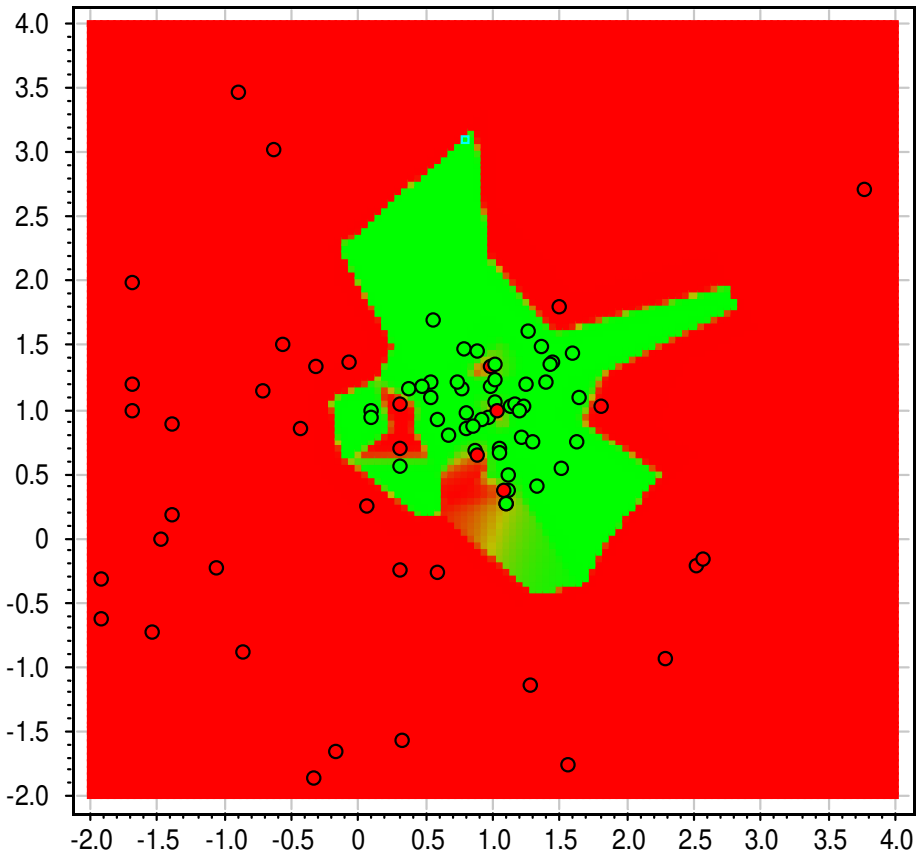
- выбор ширины окна h или числа соседей k
- выбор ядра K

Влияние ширины окна h на форму разделяющей поверхности

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

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

$h = 0.05$

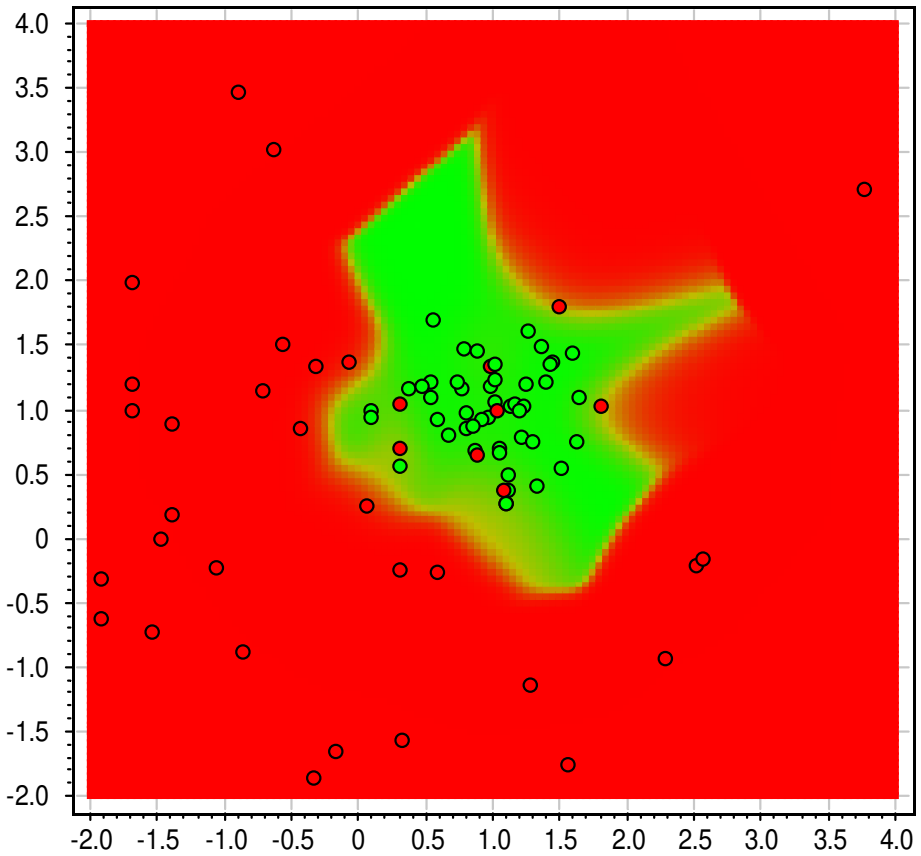


Влияние ширины окна h на форму разделяющей поверхности

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

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

$h = 0.2$

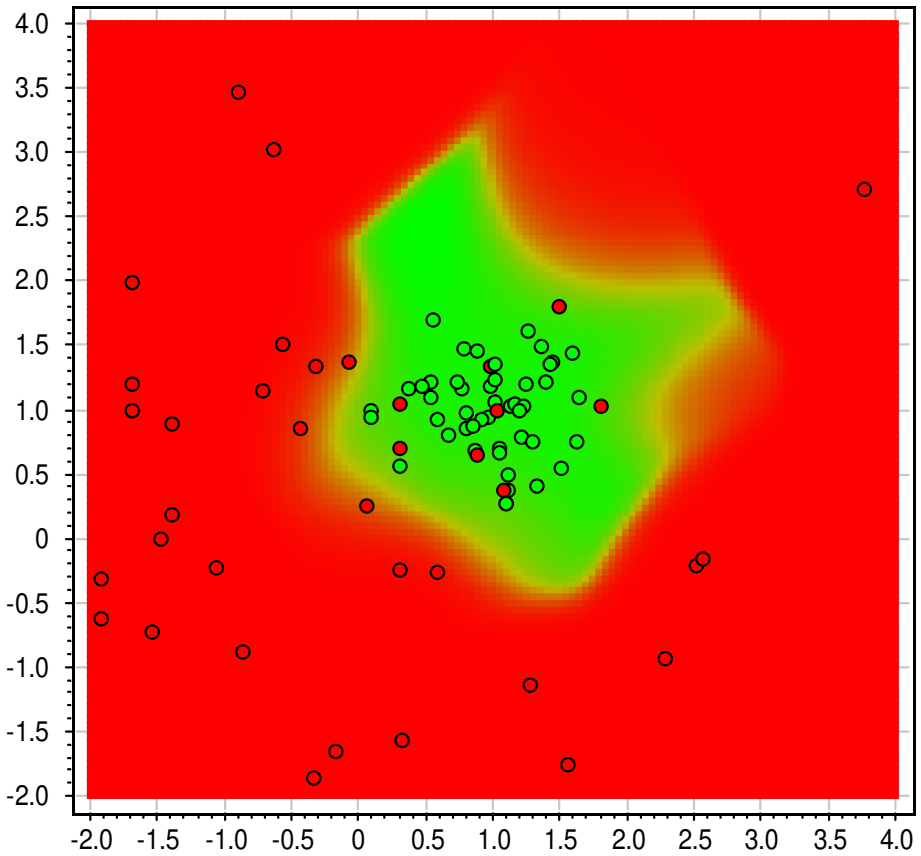


Влияние ширины окна h на форму разделяющей поверхности

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

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

$h = 0.3$

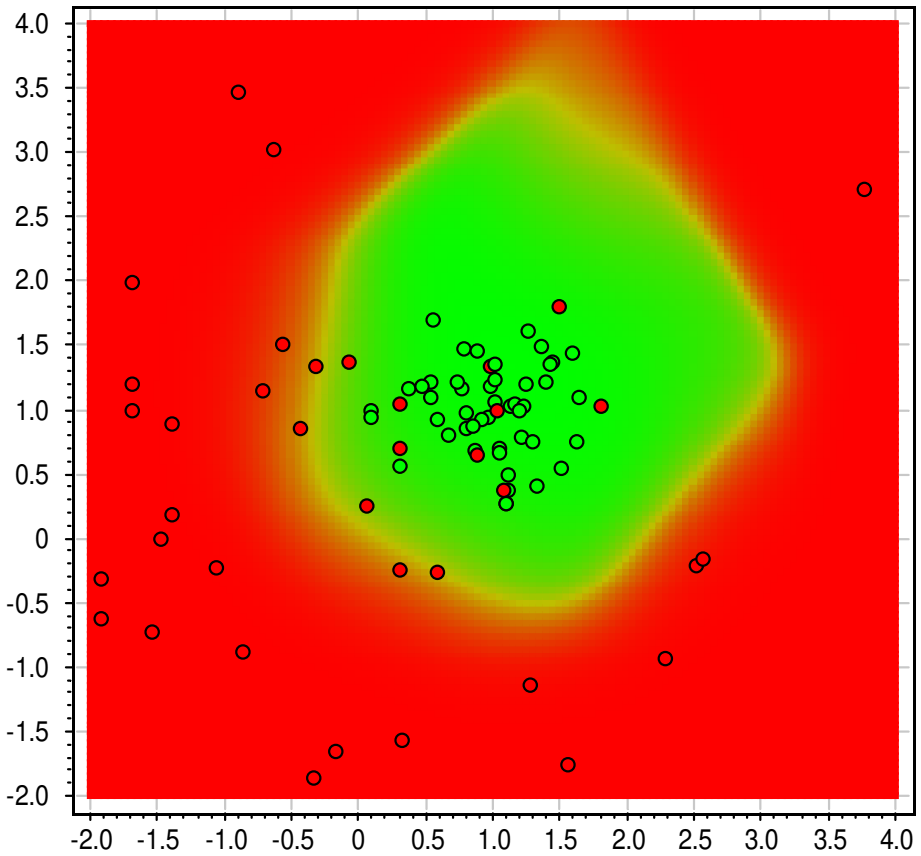


Влияние ширины окна h на форму разделяющей поверхности

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

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

$h = 0.5$

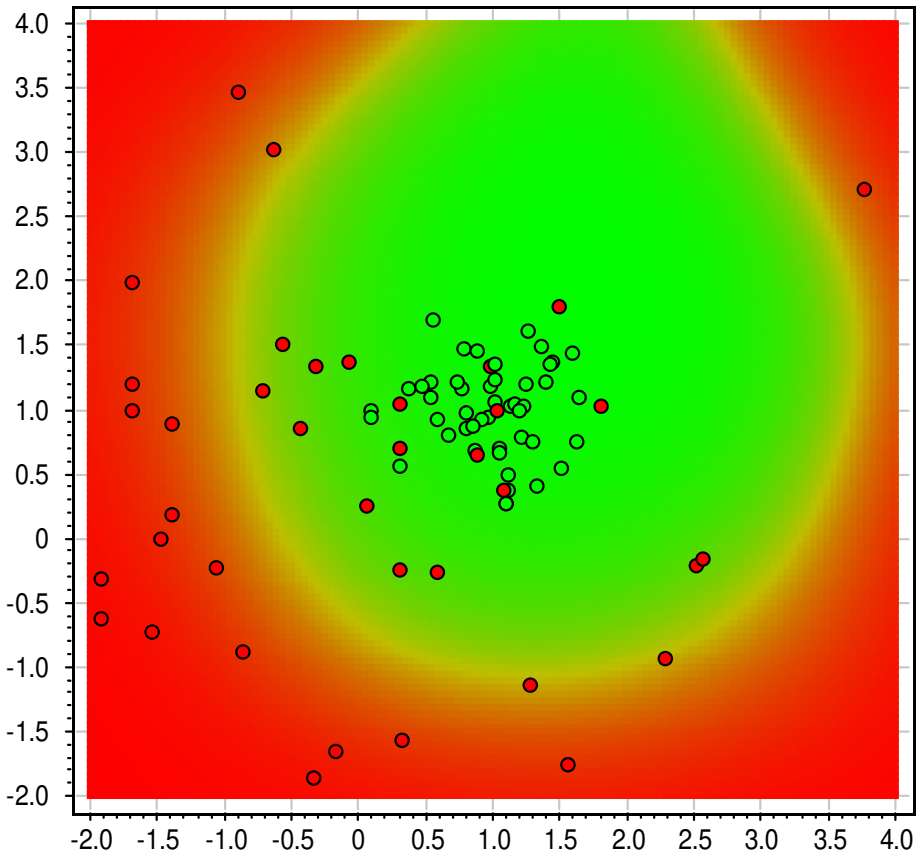


Влияние ширины окна h на форму разделяющей поверхности

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

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

$h = 1.0$

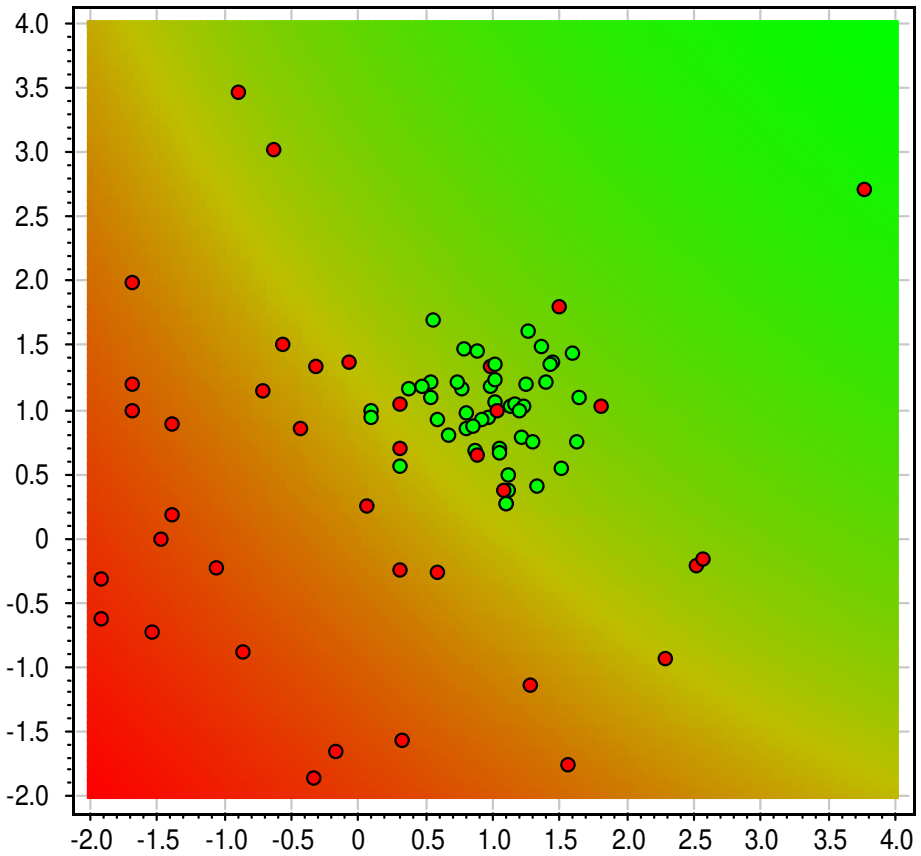


Влияние ширины окна h на форму разделяющей поверхности

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

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

$h = 5.0$



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 γ_i — веса объектов, $\gamma_i \geqslant 0$, $h_i > 0$.

Физическая аналогия из электростатики:

γ_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 $|Y|$.

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

Метод потенциальных функций = линейный классификатор

Два класса: $Y = \{-1, +1\}$.

$$a(x; X^\ell) = \arg\max_{y \in Y} \Gamma_y(x) = \operatorname{sign}\big(\Gamma_{+1}(x) - \Gamma_{-1}(x)\big) =$$

$$= \operatorname{sign} \sum_{i=1}^{\ell} \gamma_i y_i \, K\left(\frac{\rho(x, x_i)}{h_i}\right).$$

Сравним с линейной моделью классификации:

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

- $f_j(x) = y_j K\big(\frac{1}{h_j}\rho(x, x_j)\big)$ — новые признаки объекта x, близость (сходство) объекта x и обучающего объекта x_j
- γ_j — веса линейного классификатора
- $n = \ell$ — число признаков равно числу объектов обучения

Оценка обобщающей способности (generalization performance)

Полный скользящий контроль (complete cross-validation, CCV):

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

— частота ошибок алгоритма на контрольной выборке X^k, усреднённая по всем C_L^ℓ разбиениям выборки $X^L = X^\ell \sqcup X^k$ на обучающую подвыборку X^ℓ и контрольную X^k.

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

Задача.
От чего зависит обобщающая способность CCV метода 1NN?
Возможно ли улучшить алгоритм 1NN, минимизируя CCV?

Mullin M., Sukthankar R. Complete cross-validation for nearest neighbor classifiers. 2000

Понятие профиля компактности

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

$$\Pi(m) = \frac{1}{L}\sum_{i=1}^{L}\left[y_i \neq y_i^{(m)}\right]; \quad m = 1, \ldots, L-1,$$

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

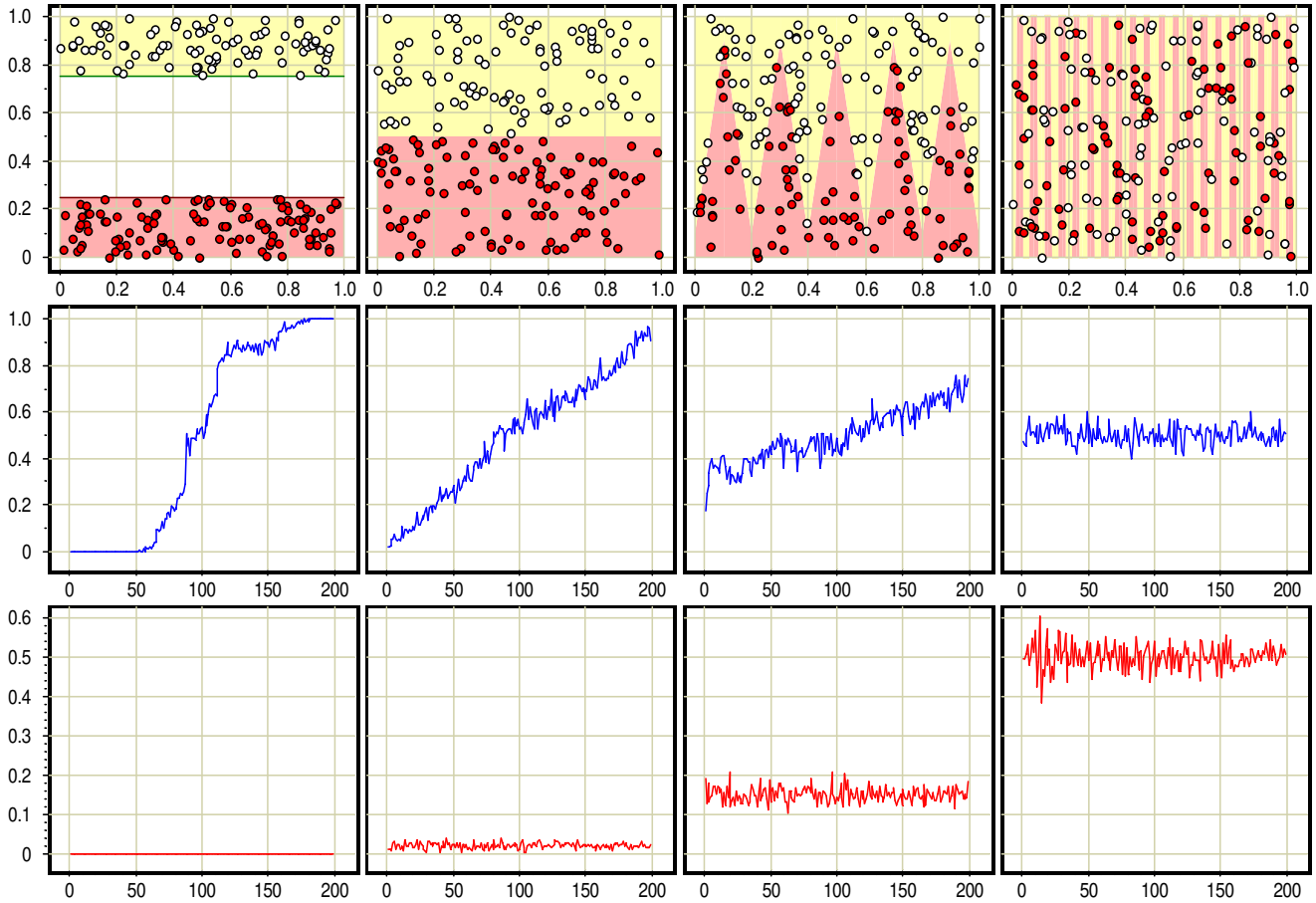
$y_i^{(m)}$ — ответ на m-м соседе объекта x_i.

Теорема (точное выражение CCV для метода 1NN)

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

Воронцов К.В. Комбинаторный подход к оценке качества обучаемых алгоритмов. 2004

Пример. Профили компактности для серии модельных задач



средний ряд — профиль компактности $\Pi(m)$

нижний ряд — зависимость $\mathrm{CCV}(k)$ от длины контроля k

Доказательство (точное выражение CCV для метода 1NN)

Основная идея — просуммировать по разбиениям аналитически:

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

$$= \sum_{X^\ell \sqcup X^k} \sum_{i=1}^{L} \sum_{m=1}^{k} \frac{\big[y_i^{(m)} \neq y_i\big]}{k C_L^\ell} \big[x_i^{(m)} \in X^\ell\big] \big[x_i, x_i^{(1)}, \ldots, x_i^{(m-1)} \in X^k\big] =$$

$$= \sum_{m=1}^{k} \sum_{i=1}^{L} \frac{\big[y_i^{(m)} \neq y_i\big]}{k C_L^\ell} \sum_{X^\ell \sqcup X^k} \big[x_i^{(m)} \in X^\ell\big] \big[x_i, x_i^{(1)}, \ldots, x_i^{(m-1)} \in X^k\big] =$$

$$= \sum_{m=1}^{k} \sum_{i=1}^{L} \frac{\big[y_i^{(m)} \neq y_i\big]}{k C_L^\ell} C_{L-1-m}^{\ell-1} = \sum_{m=1}^{k} \underbrace{\frac{1}{L} \sum_{i=1}^{L} \big[y_i^{(m)} \neq y_i\big]}_{\Pi(m)} \frac{C_{L-1-m}^{\ell-1}}{C_{L-1}^{\ell}}. \quad \blacksquare$$

Воронцов К.В. Комбинаторный подход к оценке качества обучаемых алгоритмов. 2004

Некоторые свойства профиля компактности и оценки CCV

- $R(m) = \frac{C_{L-1-m}^{\ell-1}}{C_{L-1}^{\ell}} \to 0$ быстрее геометрической прогрессии:

$$\frac{R(m+1)}{R(m)} = 1 - \frac{\ell-1}{L-1-m} < \frac{k}{L-1}.$$

- **Формализация гипотезы компактности:**
  $\text{CCV} = \sum_{m=1}^{k} \Pi(m) R(m)$ тем меньше,
  чем чаще близкие объекты лежат в одном классе.

- При малых k

$$k = 1: \quad \text{CCV} = \Pi(1) = \text{LOO};$$

$$k = 2: \quad \text{CCV} = \Pi(1)\tfrac{\ell}{\ell+1} + \Pi(2)\tfrac{1}{\ell+1};$$

$$k = 3: \quad \text{CCV} = \Pi(1)\tfrac{\ell}{\ell+2} + \Pi(2)\tfrac{2\ell}{(\ell+1)(\ell+2)} + \Pi(3)\tfrac{2}{(\ell+1)(\ell+2)}.$$

- CCV слабо зависит от длины контроля k

Задача отбора эталонов $\Omega \subseteq X^L$ (prototype learning)

$a(x; X^\ell \cap \Omega)$ — классификатор 1NN, использующий только объекты из Ω в качестве ближайших соседей.

Требуется найти оптимальное подмножество *эталонов* Ω.

Определение (*профиль компактности* относительно Ω)

$$\Pi(m, \Omega) = \frac{1}{L} \sum_{i=1}^{L} \left[y_i^{(m|\Omega)} \neq y_i \right]; \quad m = 1, \ldots, |\Omega|,$$

где $x_i^{(m|\Omega)}$ *— m-й сосед объекта* x_i *из множества* Ω

Теорема

$$\mathrm{CCV}(\Omega) = \frac{1}{L} \sum_{i=1}^{L} \underbrace{\sum_{m=1}^{k} \left[y_i^{(m|\Omega)} \neq y_i \right] \frac{C_{L-1-m}^{\ell-1}}{C_{L-1}^{\ell}}}_{T(x_i, \Omega) \text{ — вклад объекта } x_i \text{ в CCV}}$$

Жадный отбор эталонов Ω по критерию $\mathrm{CCV}(\Omega) \to \min$

Жадная стратегия удаления не-эталонов:

$\Omega := X^L$;

повторять

  найти $x \in \Omega$: $\mathrm{CCV}(\Omega \backslash \{x\}) \to \min$;

  $\Omega := \Omega \backslash \{x\}$; обновить $T(x_i, \Omega)$ для всех $x_i : x \in k\mathrm{NN}(x_i)$;

пока CCV уменьшается или почти не увеличивается;

Жадная стратегия добавления эталонов:

$\Omega := \{$по одному объекту от каждого класса$\}$;

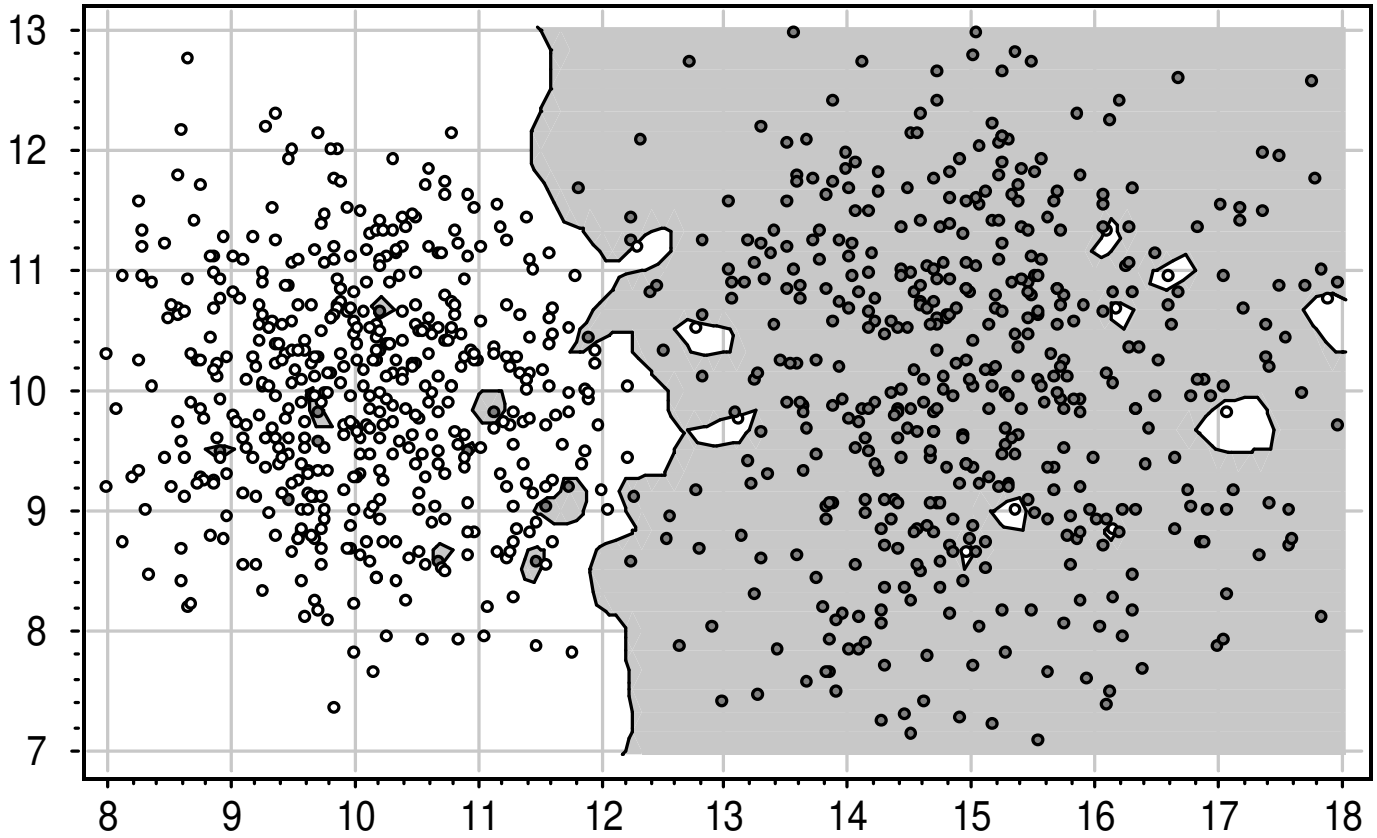
повторять

  найти $x \in X^L \backslash \Omega$: $\mathrm{CCV}(\Omega \cup \{x\}) \to \min$;

  $\Omega := \Omega \cup \{x\}$; обновить $T(x_i, \Omega)$ для всех $x_i : x \in k\mathrm{NN}(x_i)$;

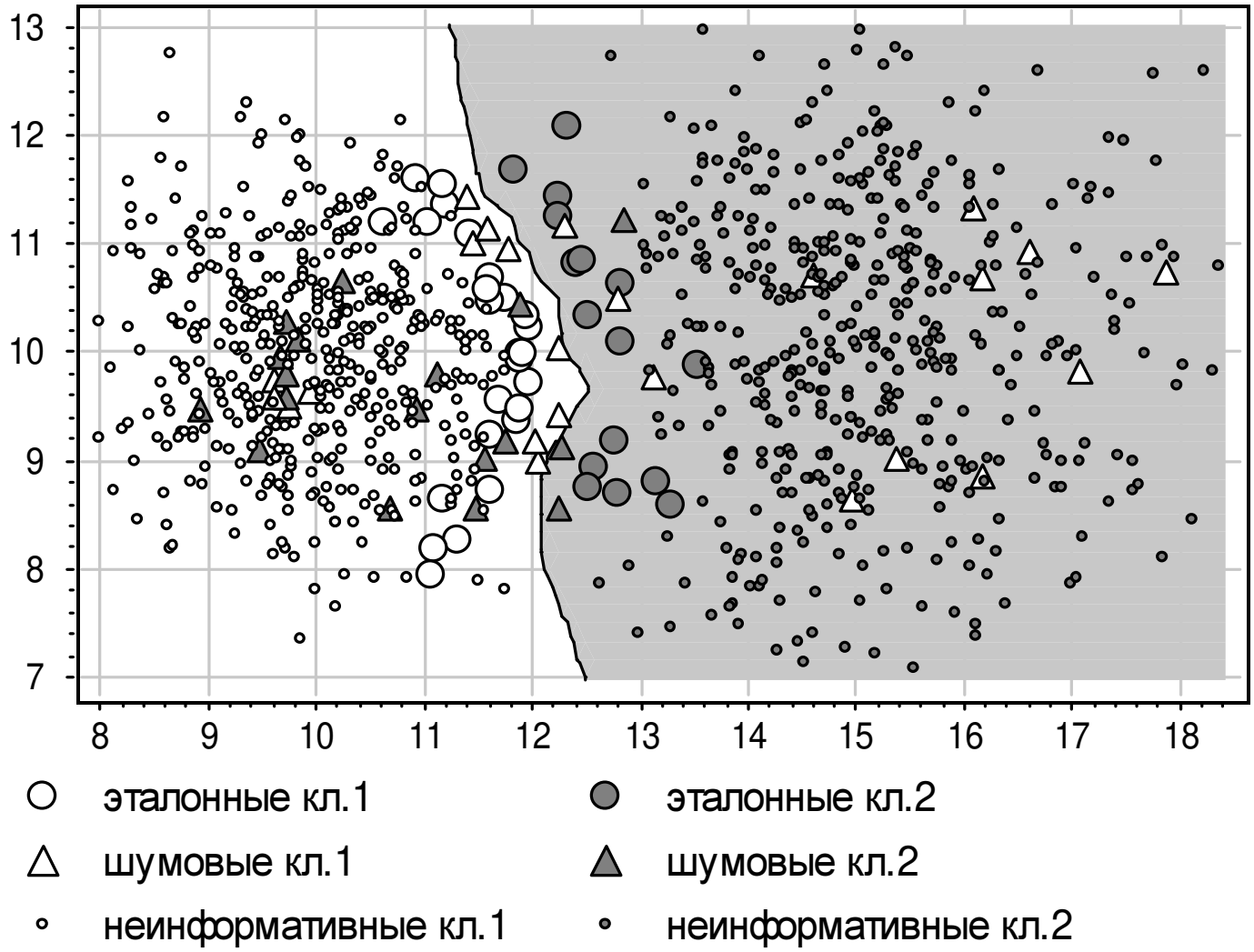
пока CCV уменьшается;

Пример. Эксперимент на синтетических данных

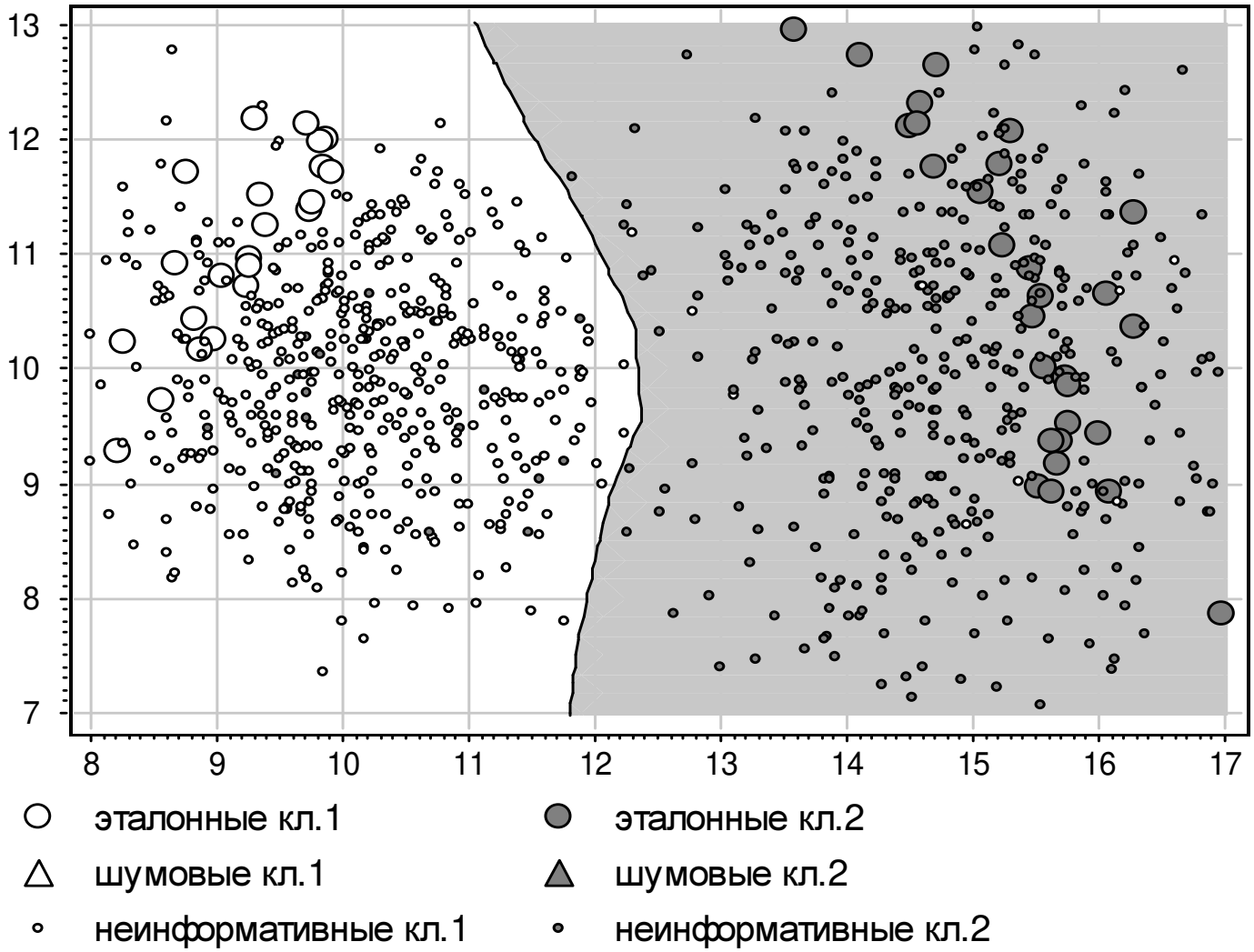


Синтетическая задача классификации:
2 класса по 500 объектов, добавлено 30 шумовых объектов

Последовательное жадное удаление не-эталонных объектов

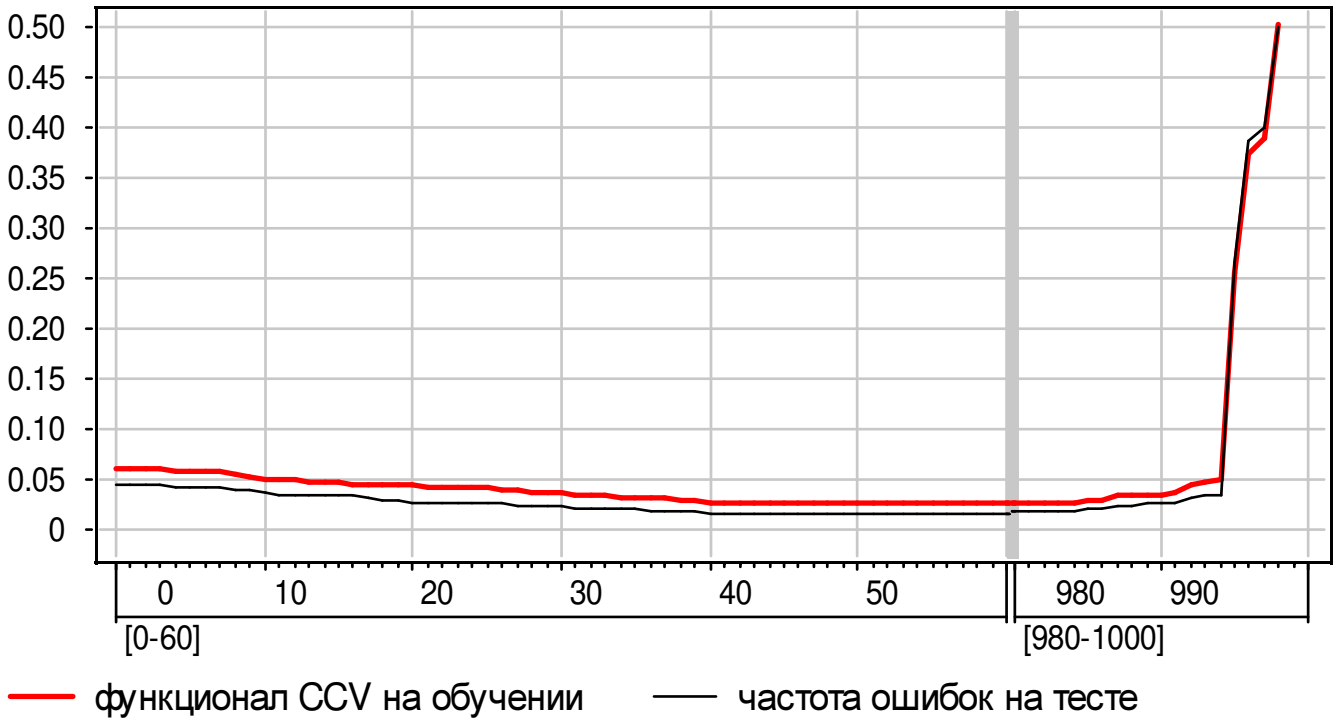


Последовательное жадное добавление эталонных объектов



Последовательное жадное удаление не-эталонных объектов

Зависимость CCV от числа удаленных неэталонных объектов.



При отборе эталонов по критерию CCV переобучения нет.

Воронцов К.В., Иванов М.Н. Отбор эталонов, основанный на минимизации функционала полного скользящего контроля. 2009

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;
  $y_i = y(x_i)$, $y : X \to Y$ — неизвестная зависимость;

- $a(x) = f(x, \theta)$ — параметрическая модель зависимости,
  $\theta \in \mathbb{R}^p$ — вектор параметров модели.

- Метод наименьших квадратов (МНК):

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

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

- **Недостаток:**
  надо иметь хорошую параметрическую модель $f(x, \theta)$

Непараметрическая регрессия, формула Надарая–Ватсона

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

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

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

Формула *ядерного сглаживания* Надарая–Ватсона:

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

Обоснование формулы Надарая–Ватсона (одномерный случай)

Теорема

Пусть выполнены следующие условия:

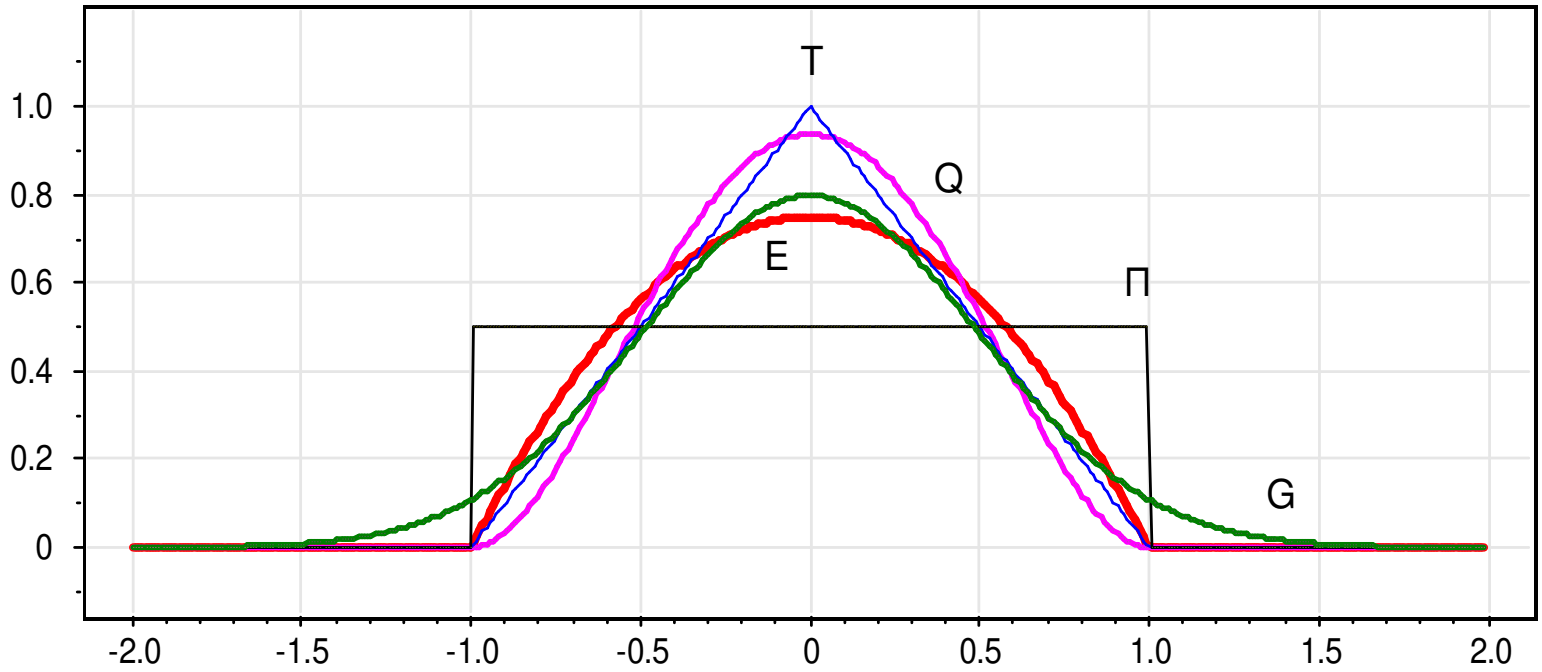
1) *выборка* $X^\ell = (x_i, y_i)_{i=1}^\ell$ *простая, из распределения* $p(x, y)$;
2) *ядро* $K(r)$ *ограничено:* $\int_0^\infty K(r)\,dr < \infty$, $\lim\limits_{r\to\infty} rK(r) = 0$;
3) *зависимость* $\mathsf{E}(y|x)$ *не имеет вертикальных асимптот:*
$\mathsf{E}(y^2|x) = \int_Y y^2 p(y|x)\,dy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

Тогда имеет место сходимость по вероятности:

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

в которой $\mathsf{E}(y|x)$, $p(x)$ *и* $\mathsf{D}(y|x)$ *непрерывны и* $p(x) > 0$.

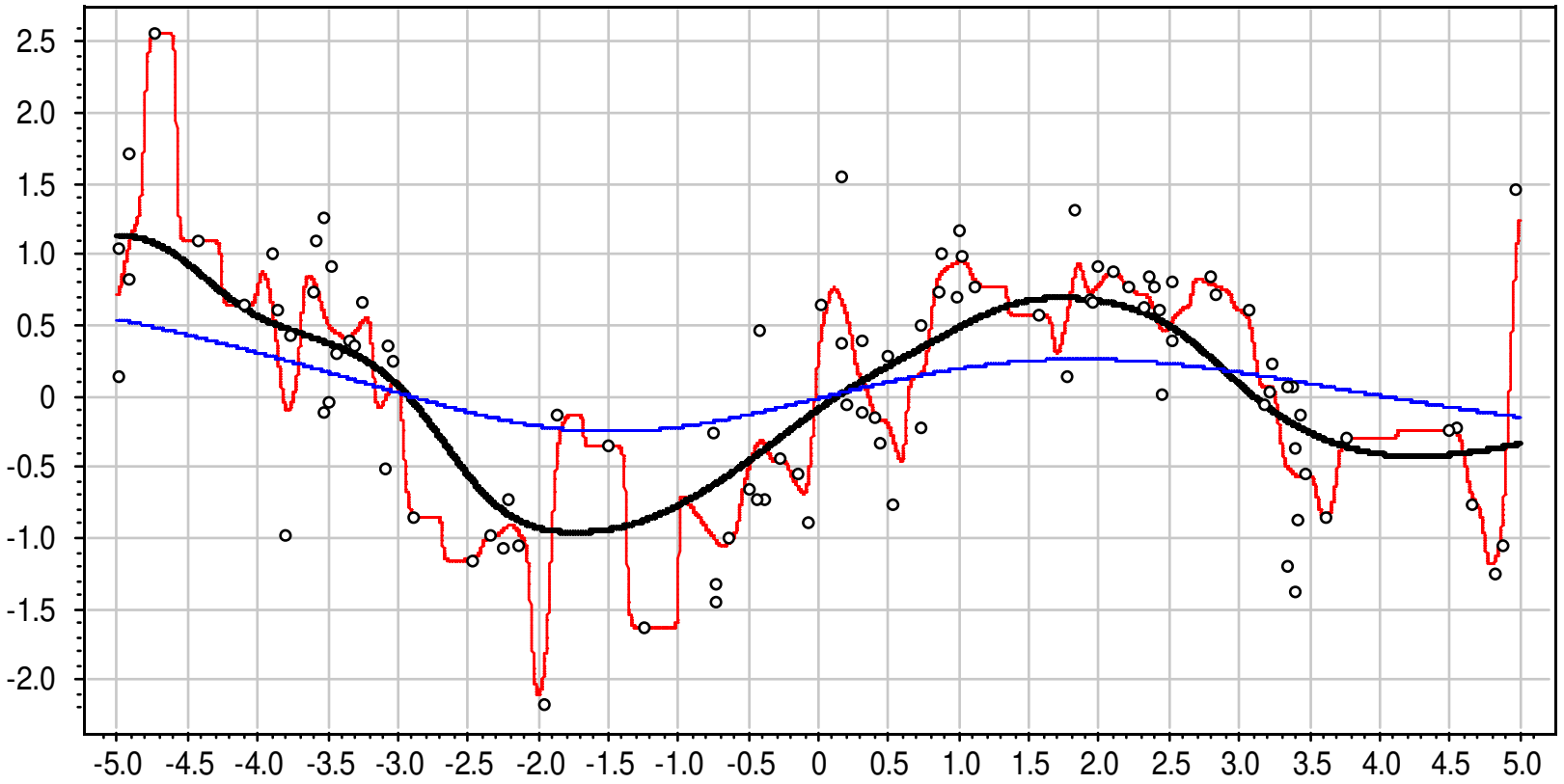
Часто используемые ядра $K(r)$



$П(r) = \big[|r| \leqslant 1\big]$ — прямоугольное
$T(r) = \big(1 - |r|\big)\big[|r| \leqslant 1\big]$ — треугольное
$E(r) = (1 - r^2)\big[|r| \leqslant 1\big]$ — квадратичное (Епанечникова)
$Q(r) = (1 - r^2)^2\big[|r| \leqslant 1\big]$ — квартическое
$G(r) = \exp\big(-2r^2\big)$ — гауссовское

Выбор ядра K и ширины окна h

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

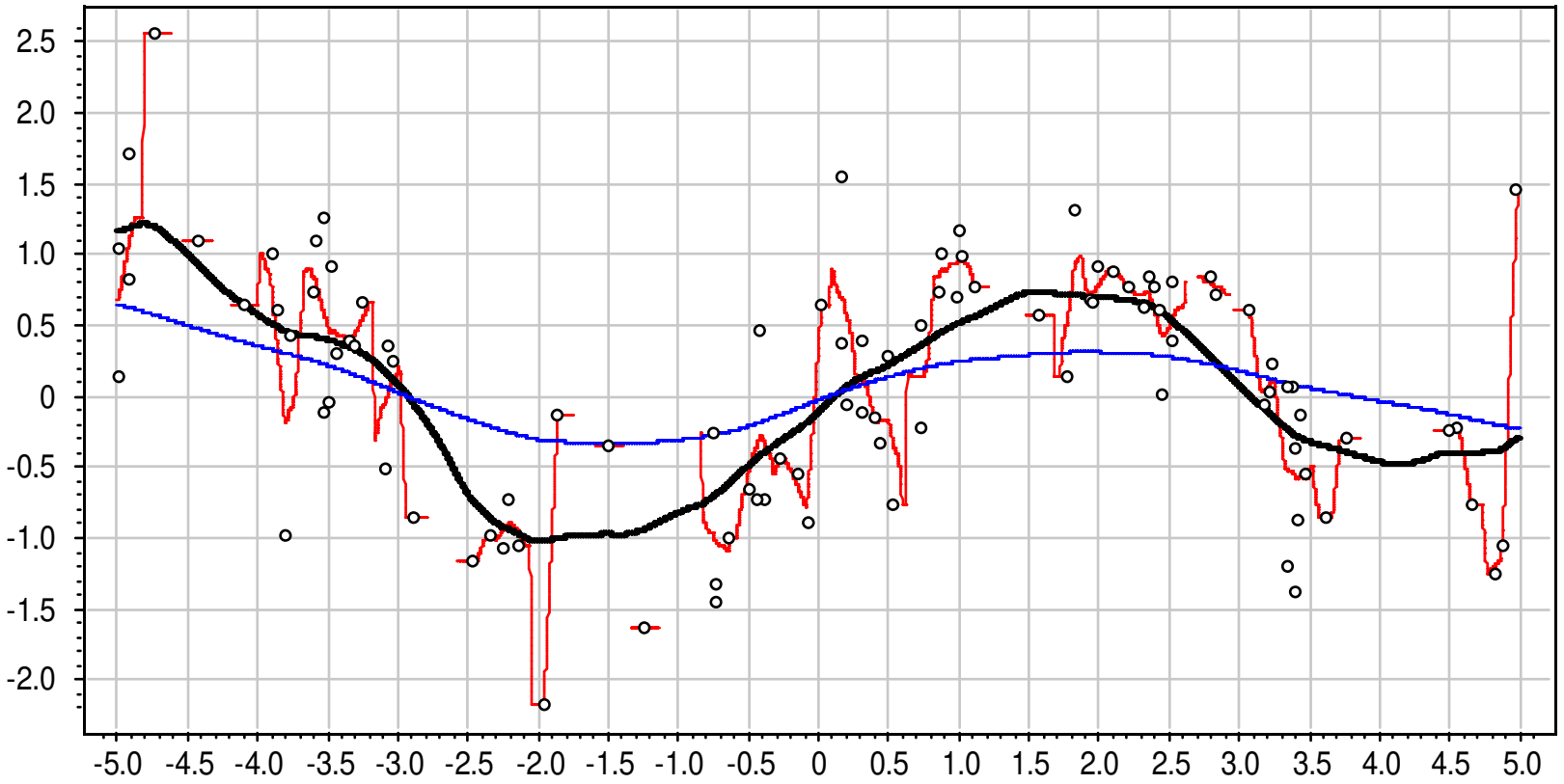


Гауссовское ядро $\Rightarrow$ гладкая аппроксимация

Ширина окна существенно влияет на точность аппроксимации

Выбор ядра K и ширины окна h

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

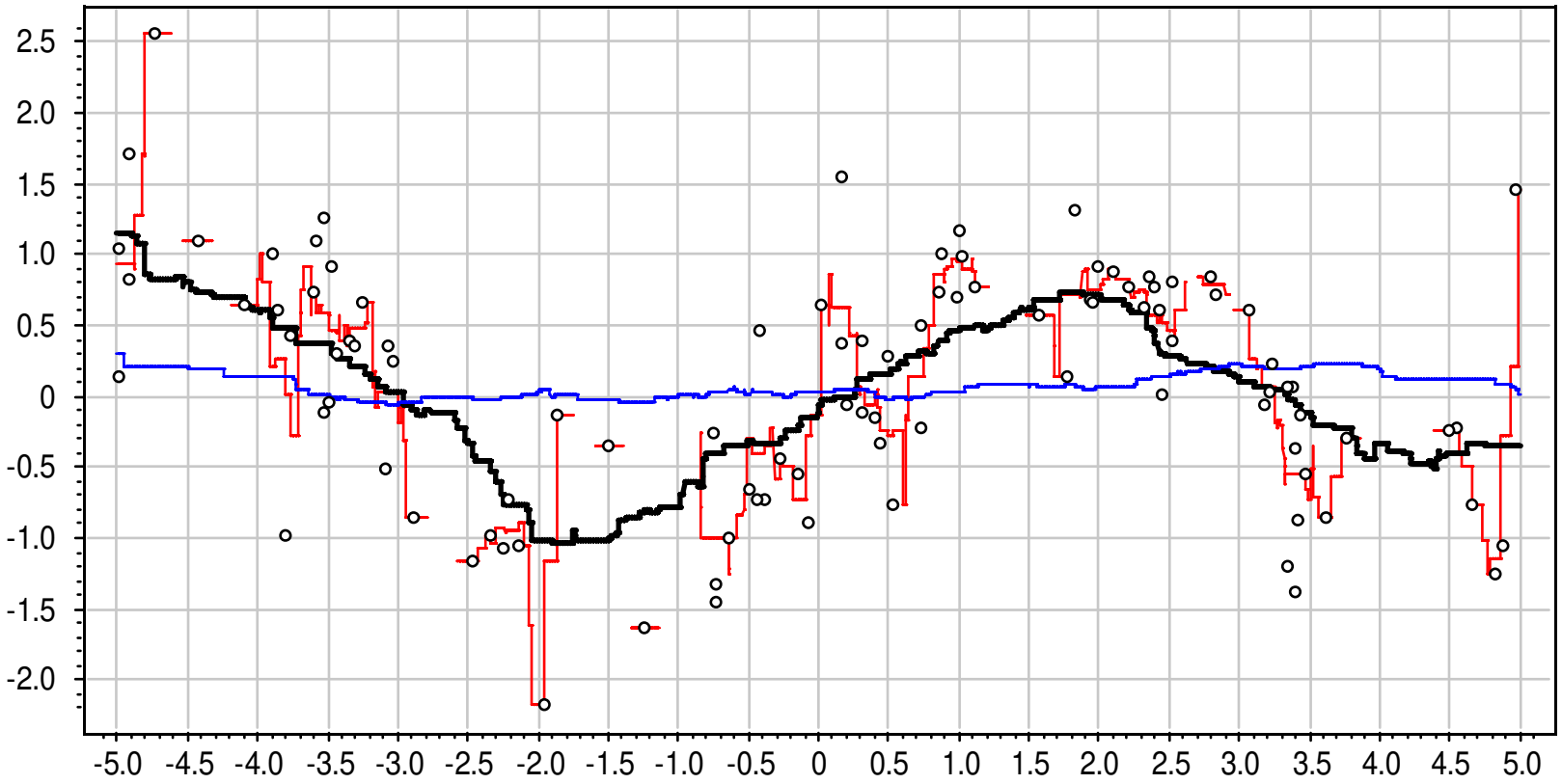


Треугольное ядро $\Rightarrow$ кусочно-линейная аппроксимация

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 $K(r) = \big[|r| \leqslant 1\big]$



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:

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

  где $x^{(k)}$ — k-й сосед объекта x.
- Оптимизация ширины окна по скользящему контролю:

$$\mathrm{LOO}(h, X^\ell) = \sum_{i=1}^{\ell} \Big(a_h\big(x_i; X^\ell \setminus \{x_i\}\big) - y_i\Big)^2 \to \min_h .$$

Резюме

Важные в ML принципы, на примере метрических методов

- *Эвристики* — изобретательные приёмы или рецепты, упрощающие построение модели и решение задачи, но не гарантирующие качество решения
- *Математические теории* в машинном обучении служат для
  — обоснования эвристических методов,
  — в том числе, выявление условий их применимости,
  — лучшего понимания моделей и методов,
  — на его основе, изобретения более совершенных эвристик
- Увы, легко доказываемые обоснования, свойства, оценки, как правило, не слишком полезны на практике
- Увы, наиболее практически полезные эвристики, как правило, не удаётся обосновать десятилетиями

Резюме

- Метрические методы — простейшие в машинном обучении, обучение сводится к запоминанию выборки (lazy learning)
- Усложняя метрические методы, можно обучать:
  — число ближайших соседей k или ширину окна h
  — веса (значимости, информативности) объектов
  — набор эталонов (prototype learning)
  — метрику (distance learning, similarity learning), в частности, веса признаков в метрике Минковского
- Метод потенциальных функций = линейный классификатор
  расстояние до опорного объекта = новый признак
- Качество обучения зависит от метрики и ширины окна, слабо зависит от вида ядра сглаживания
- Непараметрические методы обходятся без модели?
  Нет, модельные предположения закладываются в метрику