Гипотезы компактности и непрерывности

Гипотеза непрерывности (для регрессии):
близким объектам соответствуют близкие ответы.

Гипотеза компактности (для классификации):
близкие объекты, как правило, лежат в одном классе.

Формализация понятия «близости»:
задана функция расстояния $\rho\colon X \times X \to [0, \infty)$.

Пример. Евклидово расстояние и его обобщение:

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

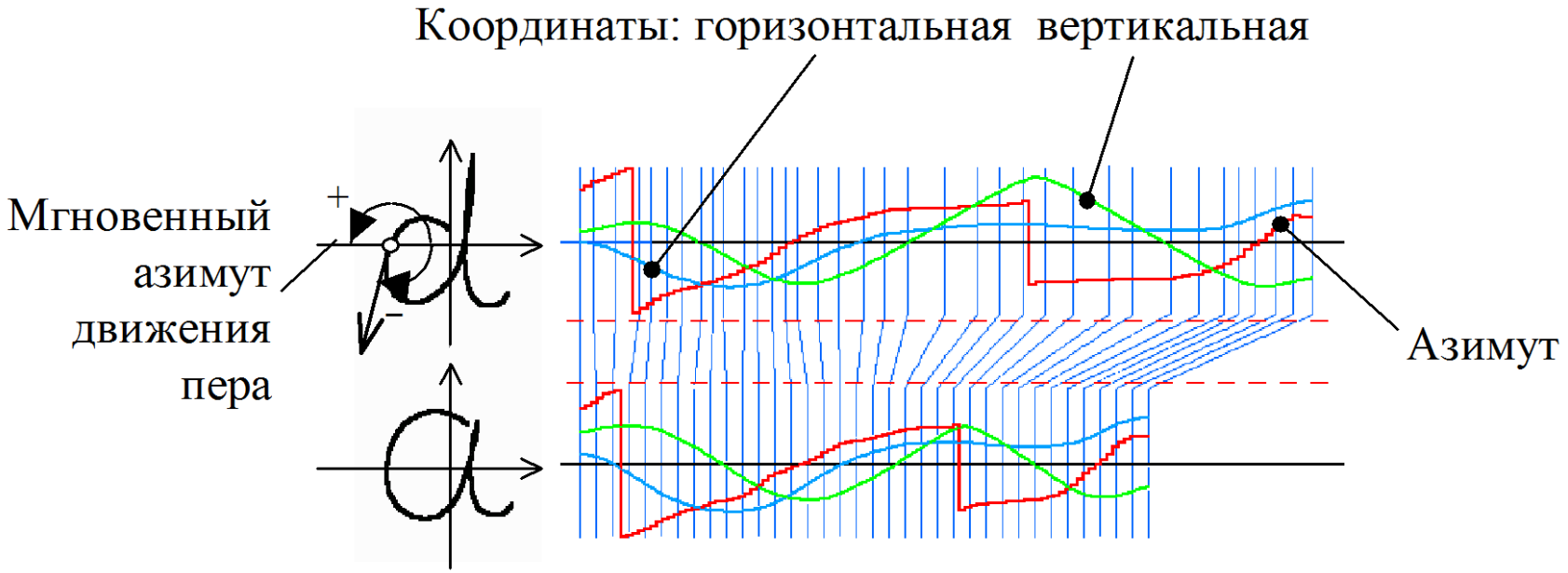
$x = (x^1, \ldots, x^n)$ — вектор признаков объекта x,
$x_i = (x_i^1, \ldots, x_i^n)$ — вектор признаков объекта x_i.

Ещё примеры расстояний:

— **между текстами** (редакторское расстояние Левенштейна):

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

— **между сигналами** (энергия сжатий и растяжений):



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:

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

$w(i, x)$ — вес, оценка сходства объекта x с его i-м соседом, неотрицательная, не возрастающая по i.

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

Метод k ближайших соседей (k nearest neighbors, kNN)

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

Преимущества:

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

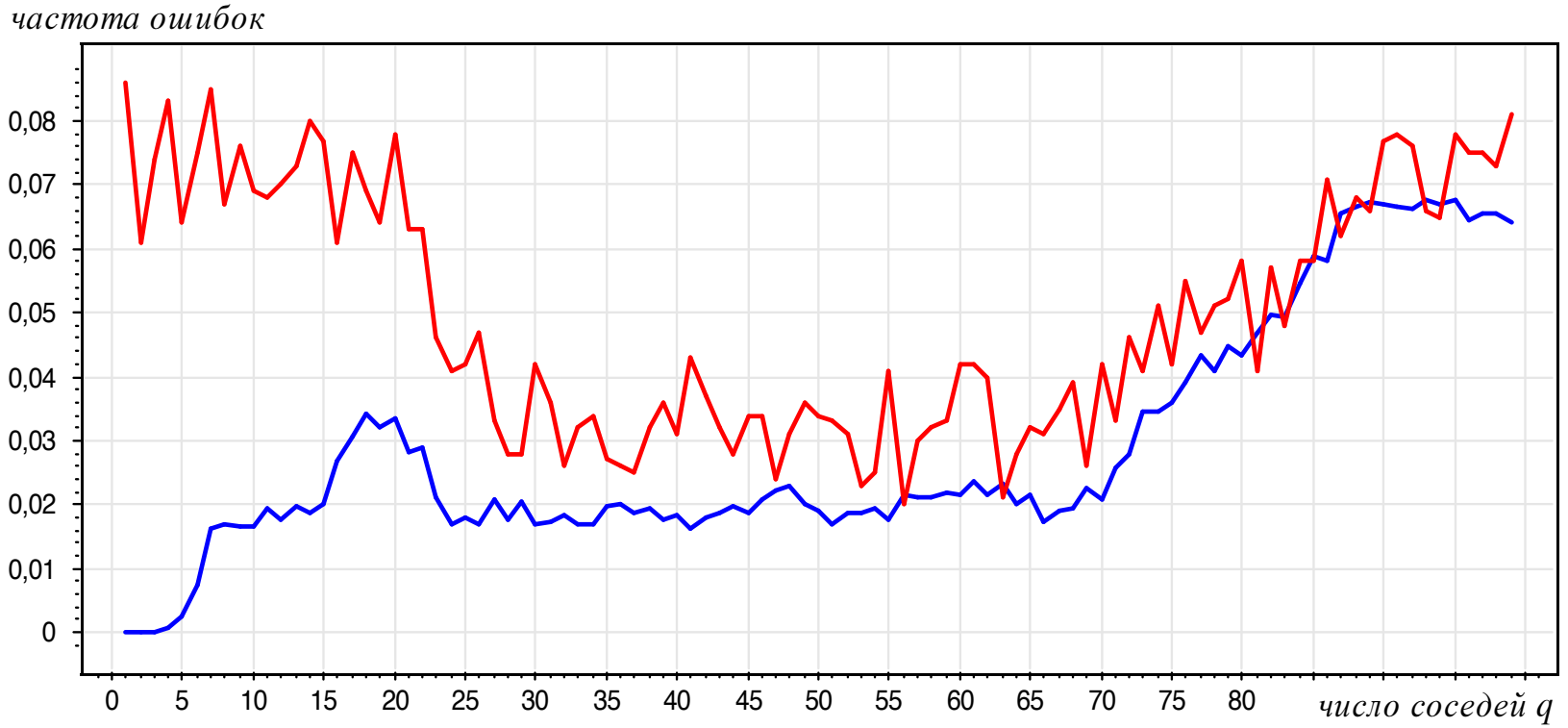
$$\mathrm{LOO}(k, X^\ell) = \sum_{i=1}^{\ell} \Big[a\big(x_i; X^\ell \setminus \{x_i\}, k\big) \neq y_i \Big] \to \min_k .$$

Проблемы:

- возможны ситуации, когда классификация не однозначна: $\Gamma_y(x) = \Gamma_s(x)$ для пары классов $y \neq s$
- учитываются не значения расстояний, а только их ранги

Пример зависимости LOO от числа соседей

Пример. Задача Iris.



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

— несмещённое число ошибок LOO

В реальных задачах минимум редко бывает при $k = 1$.

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:

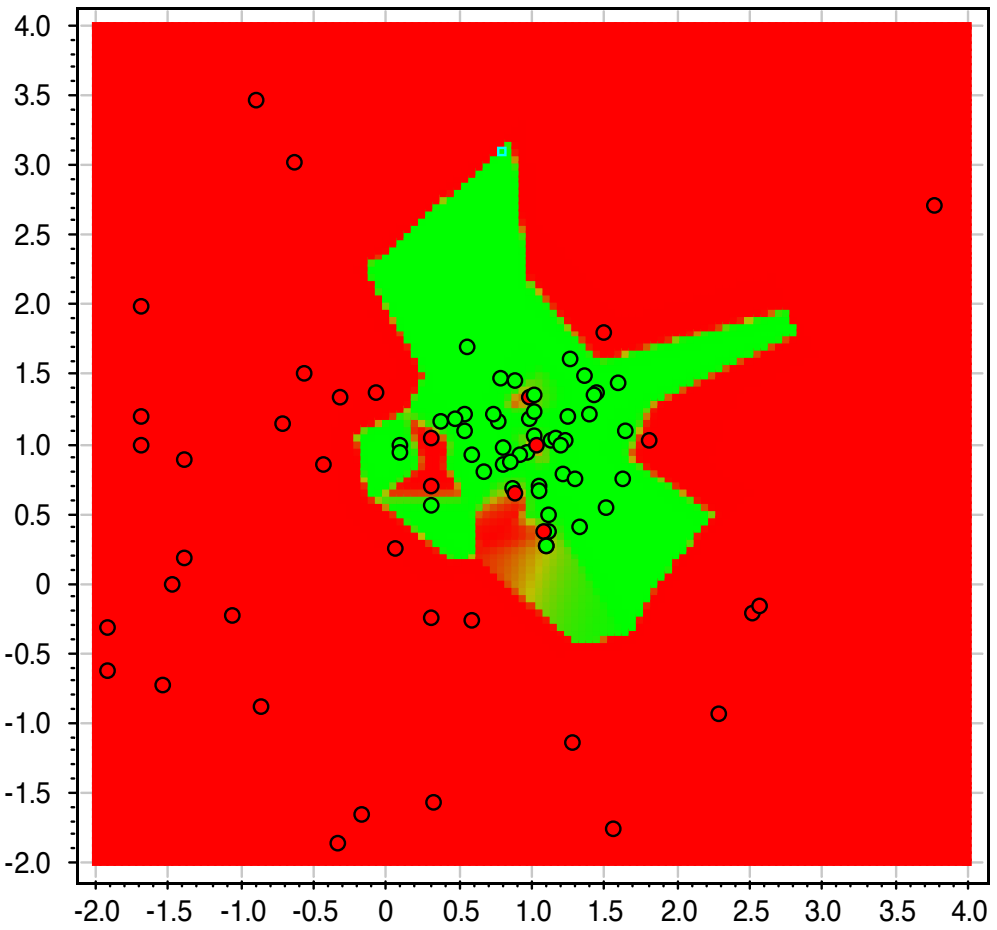
- выбор ширины окна h или числа соседей k
- выбор ядра K слабо влияет на качество классификации

Парзеновское окно фиксированной ширины h

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

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

$h = 0.05$

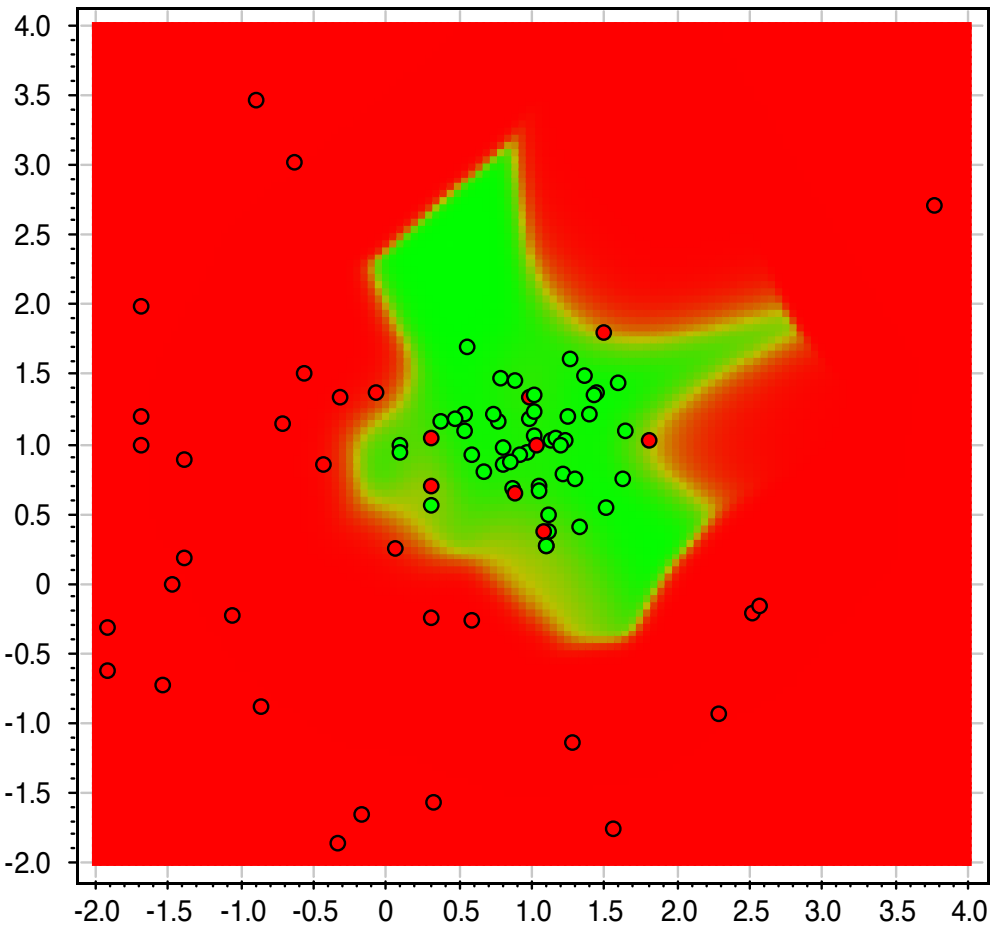


Парзеновское окно фиксированной ширины h

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

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

$h = 0.2$

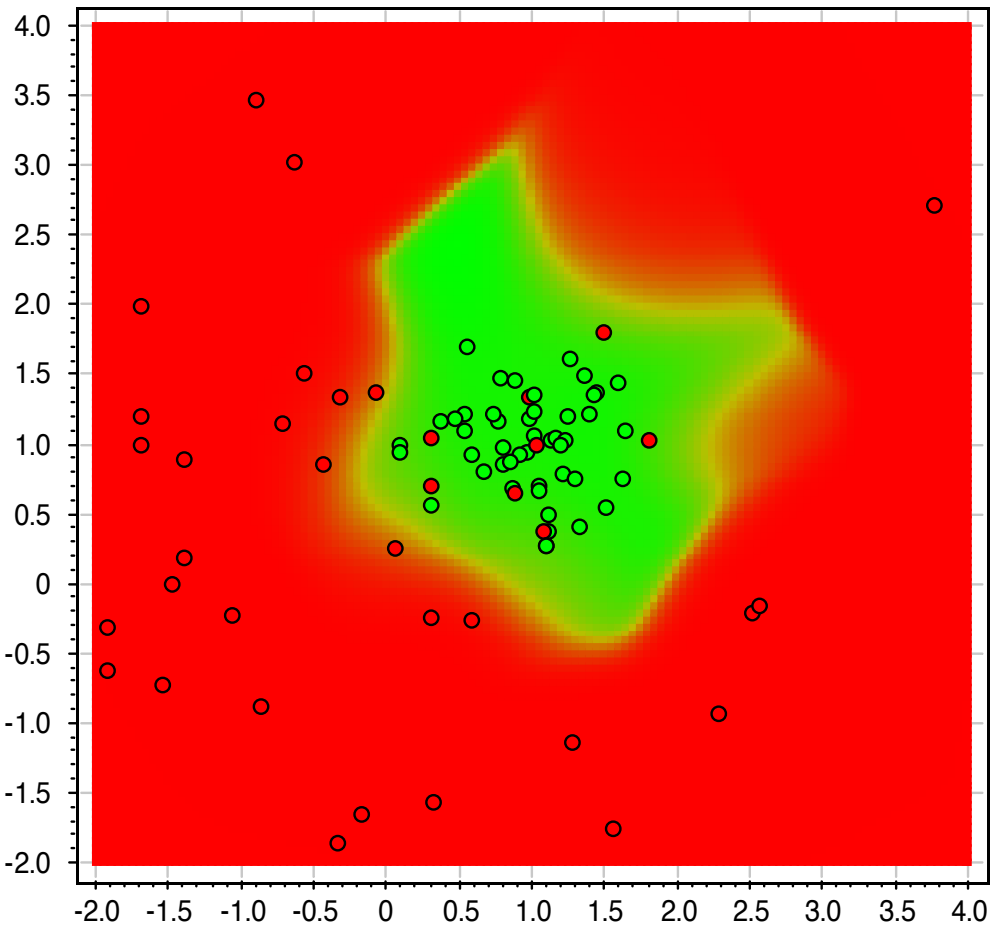


Парзеновское окно фиксированной ширины h

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

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

$h = 0.3$

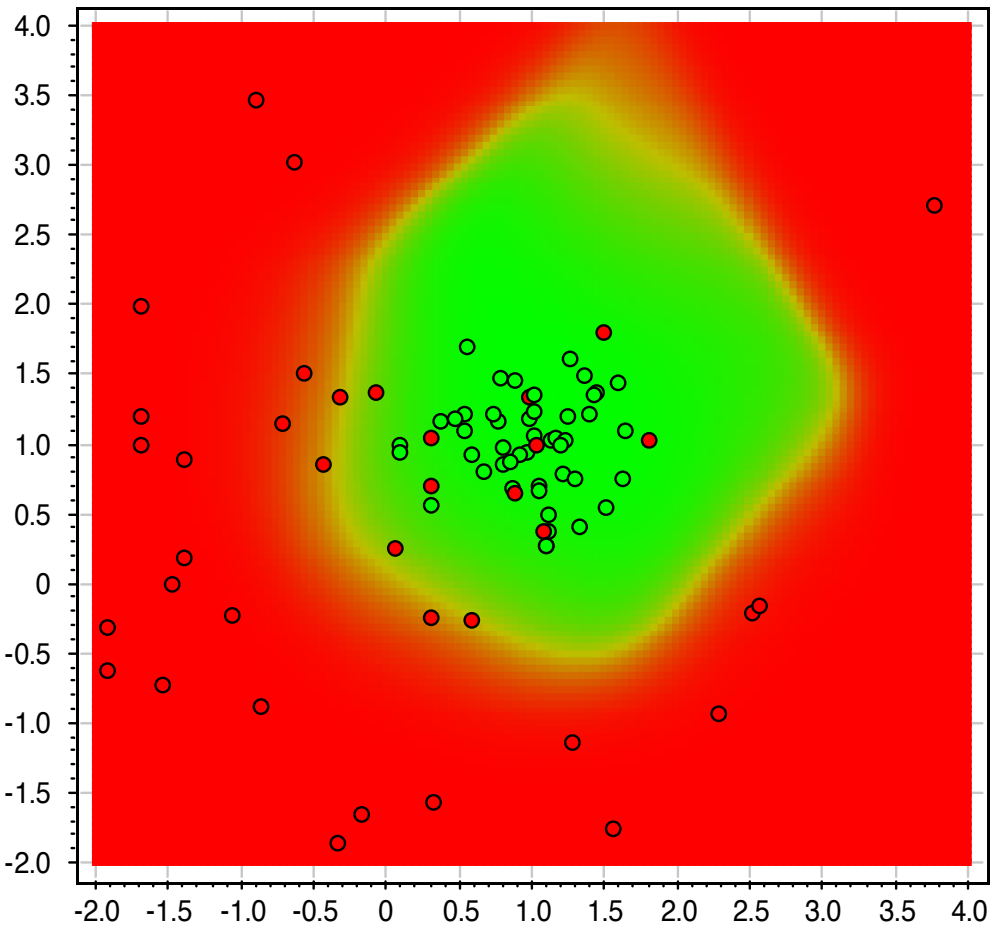


Парзеновское окно фиксированной ширины h

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

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

$h = 0.5$

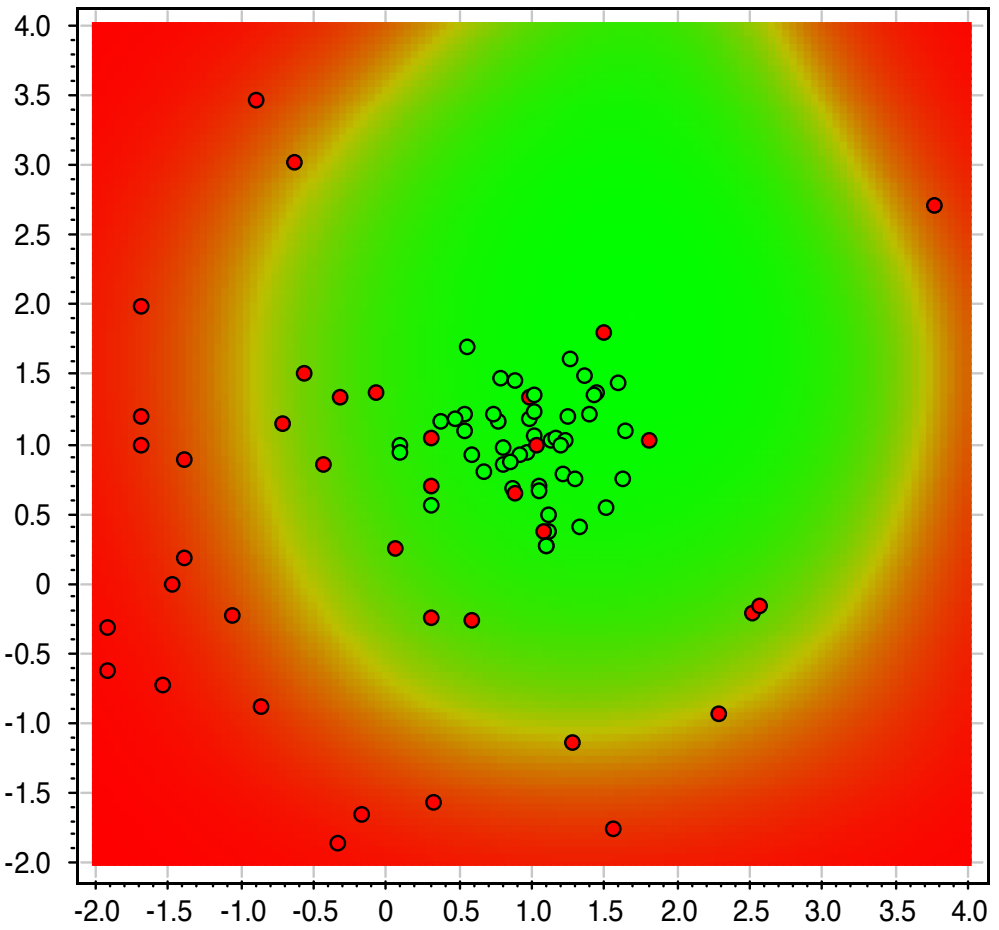


Парзеновское окно фиксированной ширины h

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

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

$h = 1.0$

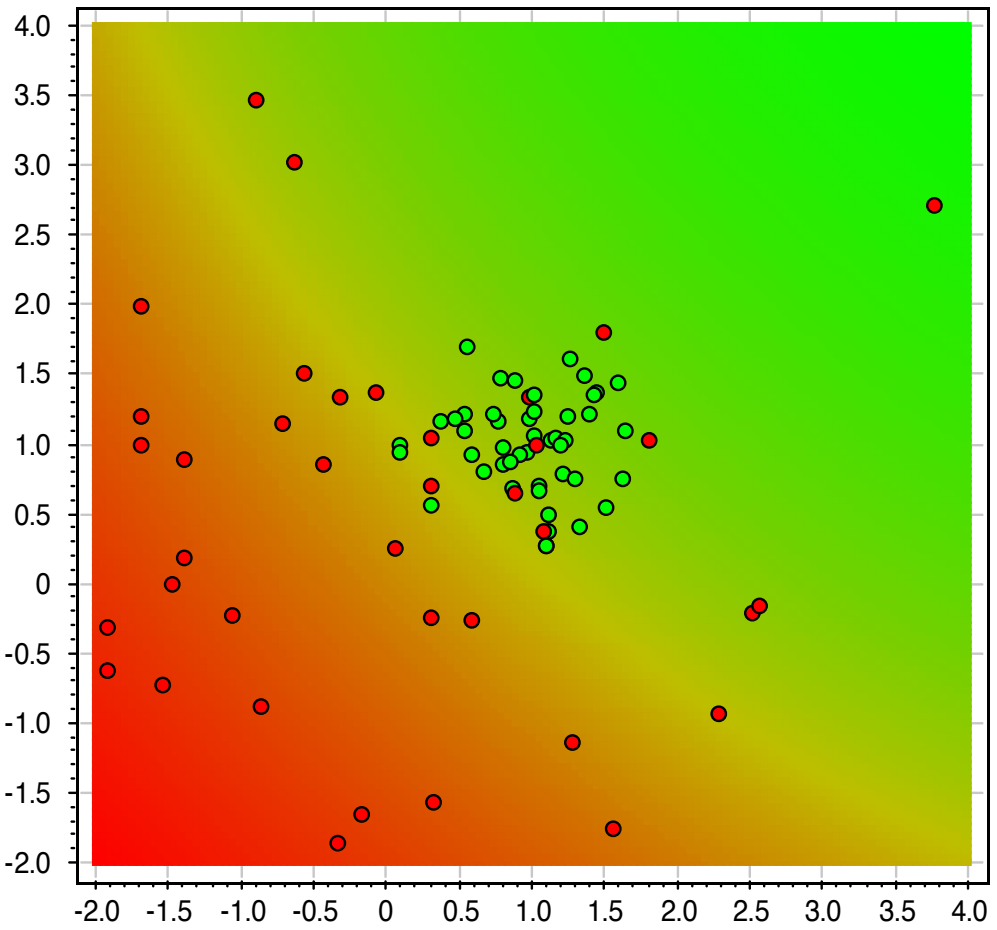


Парзеновское окно фиксированной ширины h

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

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

$h = 5.0$



Метод потенциальных функций

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

Более простая запись (без ранжирования объектов):

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

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

Физическая аналогия:

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 $Y = \{-1, +1\}$);

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

для задач классификации нет таких ограничений на K.

Метод потенциальных функций = линейный классификатор

Два класса: $Y = \{-1, +1\}$.

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

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

Сравним с линейной моделью классификации:

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

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

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

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

- Метод наименьших квадратов (МНК):

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

  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 $x \in X$:

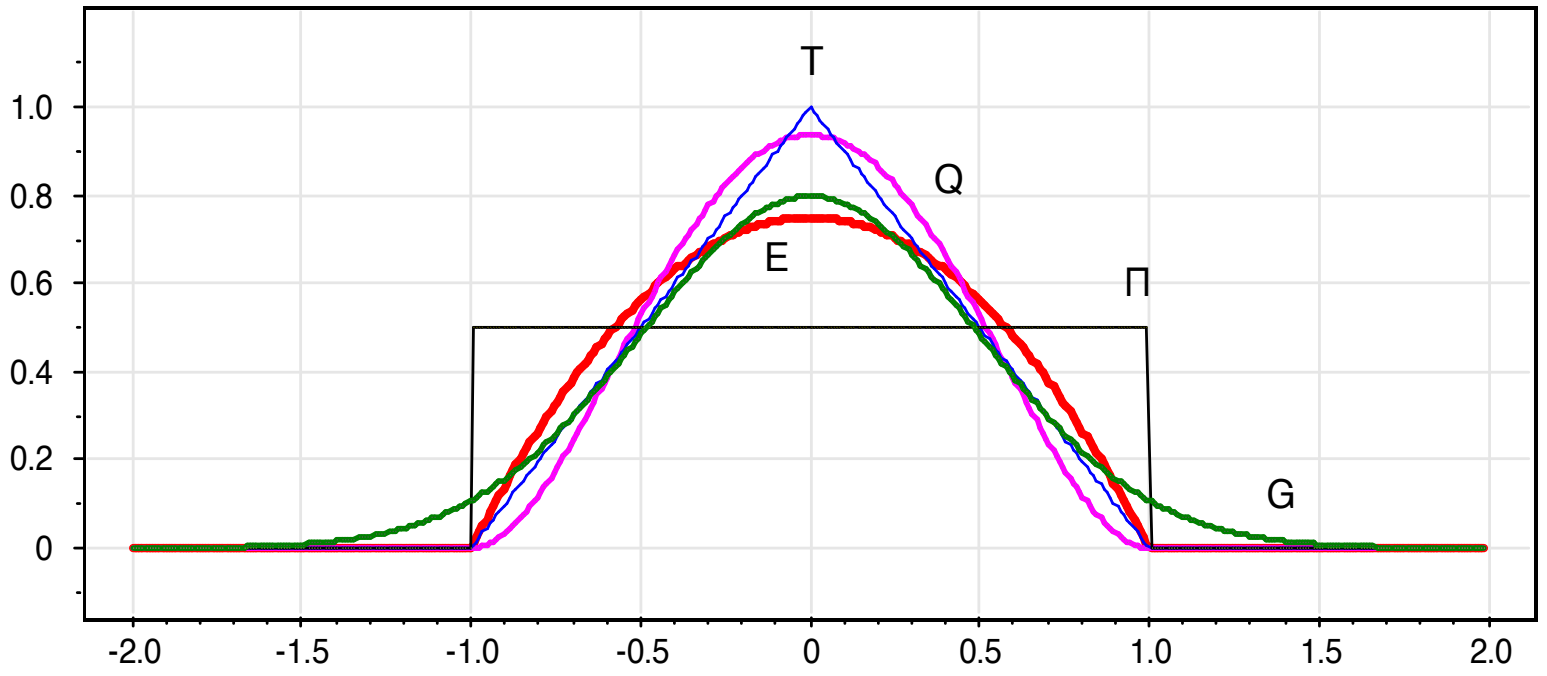
$$Q(\alpha; X^\ell) = \sum_{i=1}^{\ell} w_i(x)\big(\alpha - y_i\big)^2 \to \min_{\alpha\in\mathbb{R}};$$

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:

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

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

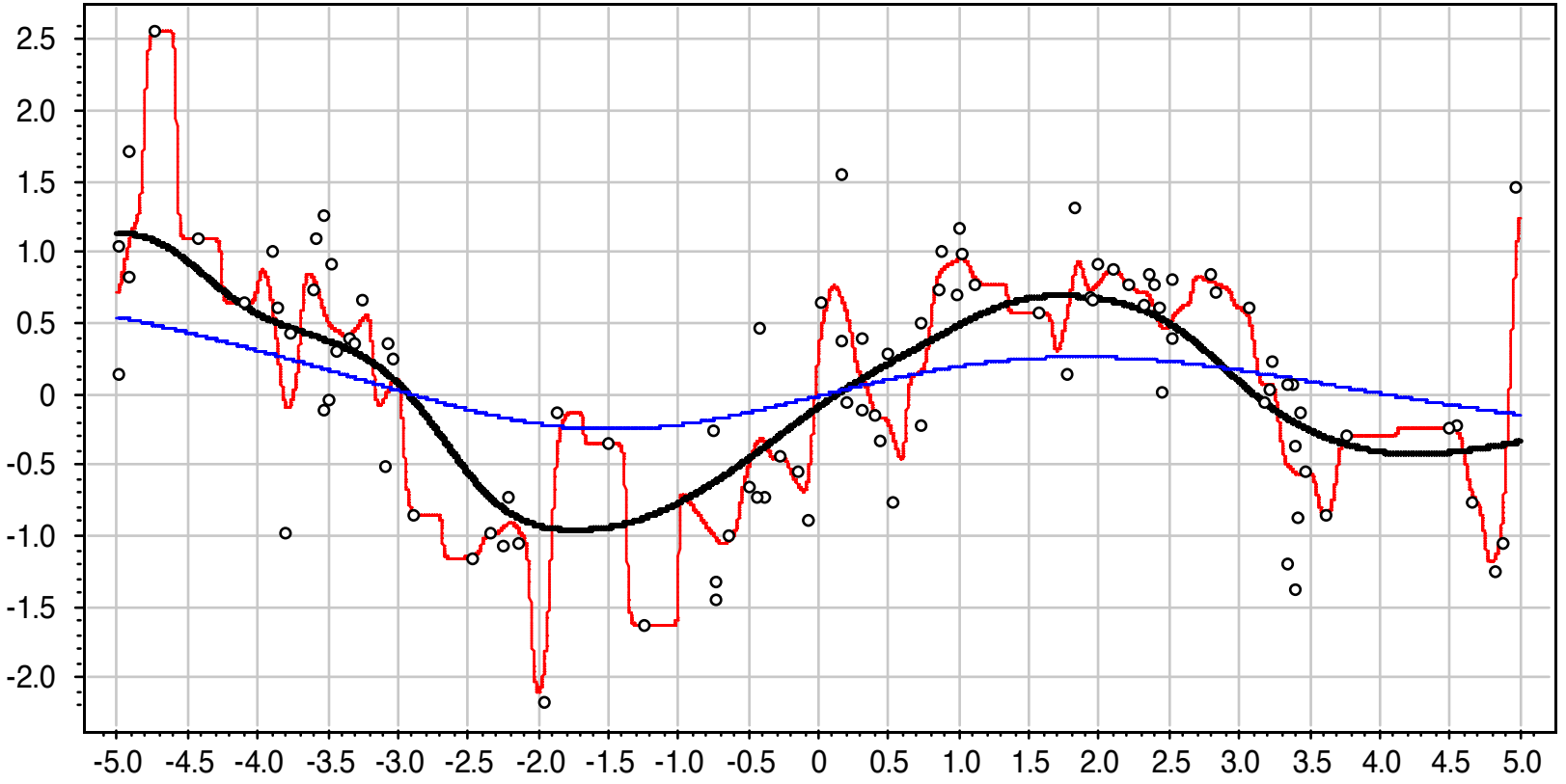
$E(r) = (1 - r^2)\big[|r| \leqslant 1\big]$ — квадратичное (Епанечникова)

$Q(r) = (1 - r^2)^2\big[|r| \leqslant 1\big]$ — 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 $K(r) = \exp\left(-2r^2\right)$

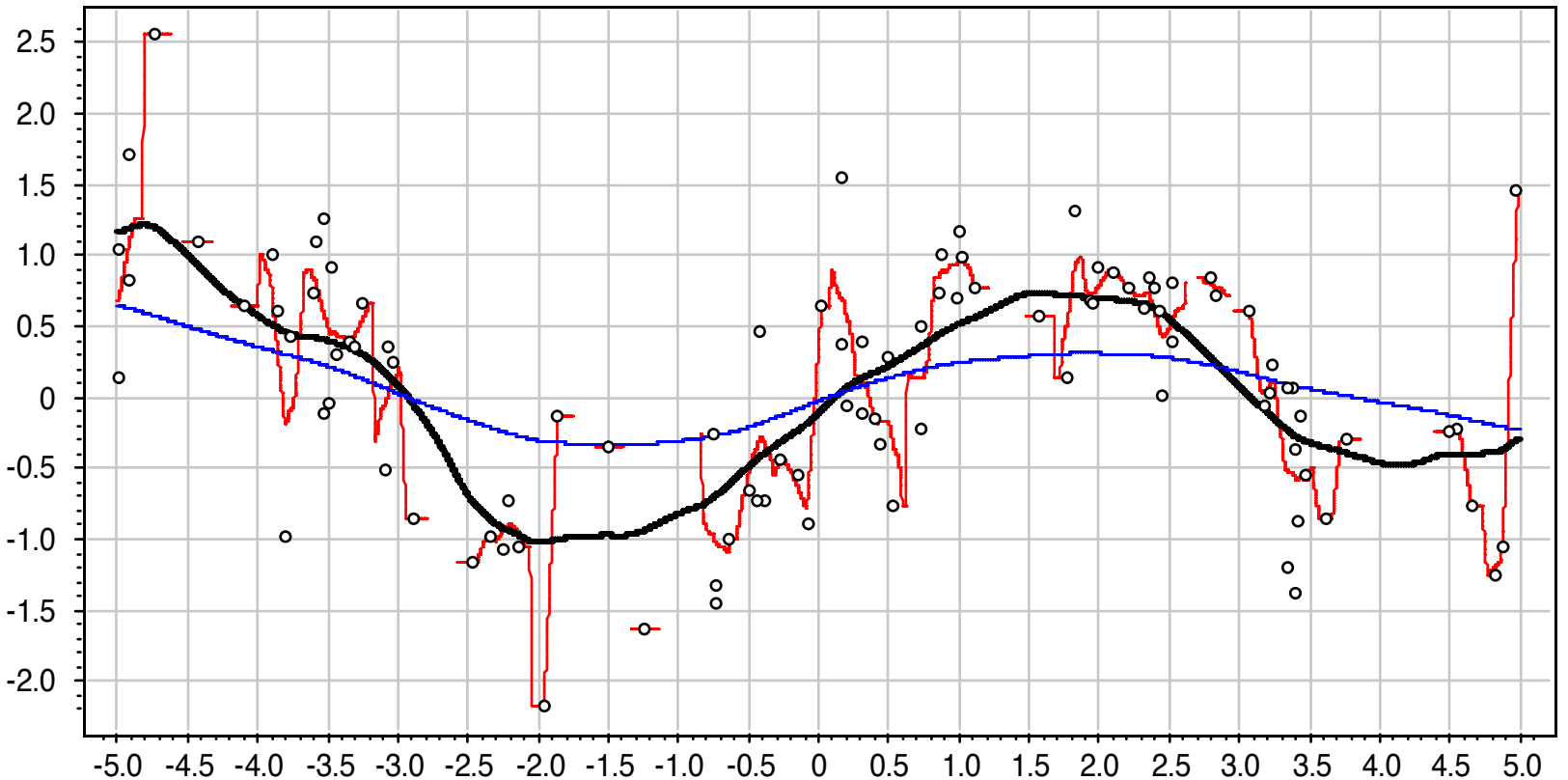


Гауссовское ядро $\Rightarrow$ гладкая аппроксимация

Ширина окна существенно влияет на точность аппроксимации

Выбор ядра K и ширины окна h

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

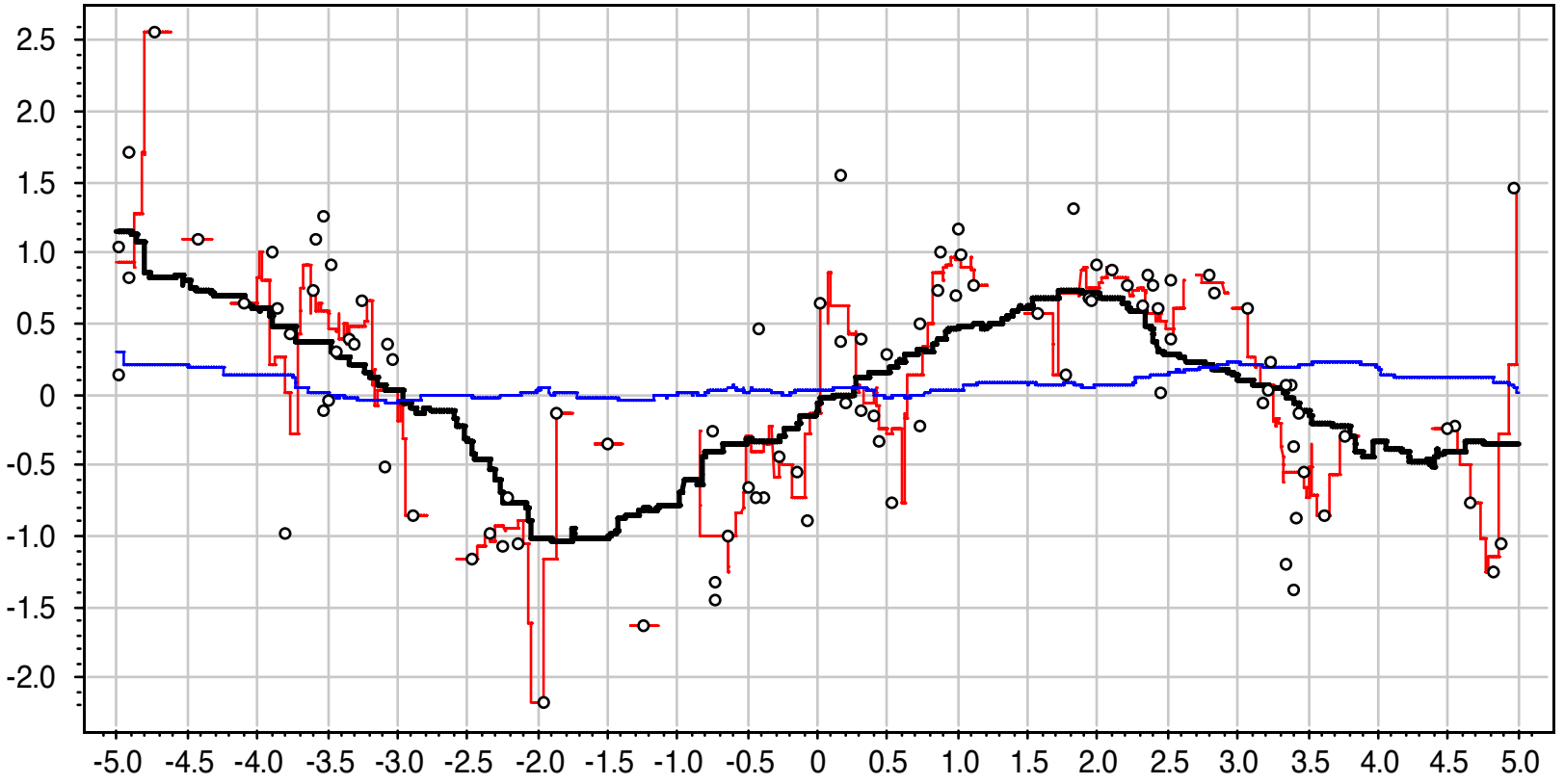


Треугольное ядро $\Rightarrow$ кусочно-линейная аппроксимация

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



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 $h(x)$ по k ближайшим соседям:

$$w_i(x) = K\left(\frac{\rho(x, x_i)}{h(x)}\right),$$

  где $h(x) = \rho(x, x^{(k+1)})$, $x^{(k)}$ — 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

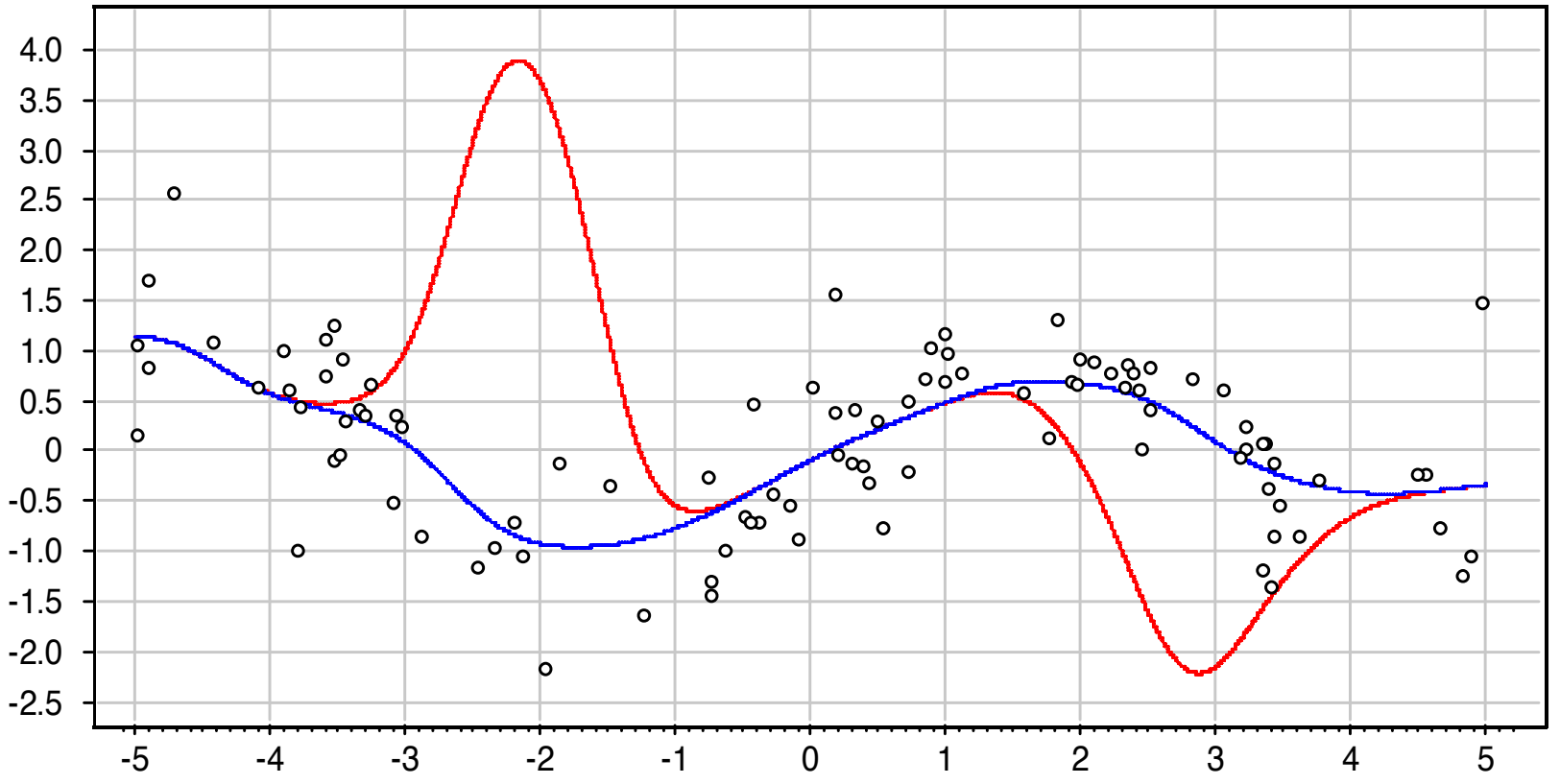
- Непараметрическая регрессия избегает использования параметрической модели зависимости $f(x, \alpha)$.
- Неявно моделью является функция расстояния $\rho(x, x_i)$ между объектами.
- Что можно обучать:
  — число ближайших соседей k или ширину окна h;
  — веса объектов (обнаруживать выбросы);
  — метрику (distance learning, similarity learning);
  — в частности, веса признаков в метрике.
- Непараметрическая регрессия часто используется как инструмент предварительной обработки данных для сглаживания шумов в данных.

Проблема выбросов (эксперимент на синтетических данных)

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

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

Синяя кривая — выбросов нет



Проблема выбросов и локально взвешенное сглаживание

Проблема выбросов: большие случайные ошибки в значениях y_i сильно искажают оценку Надарая–Ватсона

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

Идея:
чем больше величина невязки $\varepsilon_i = \left|a_h\left(x_i; X^\ell \setminus \{x_i\}\right) - y_i\right|$,
тем меньше должен быть вес i-го объекта $w_i(x)$.

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

Рекомендация:
использовать квартическое ядро $\tilde{K}(\varepsilon) = K_Q\left(\frac{\varepsilon}{6\,\mathrm{med}\{\varepsilon_i\}}\right)$,
где $\mathrm{med}\{\varepsilon_i\}$ — медиана множества значений ε_i.

Алгоритм LOWESS (LOcally WEighted Scatter plot Smoothing)

Вход: X^ℓ — обучающая выборка;
Выход: коэффициенты γ_i, $i = 1, \ldots, \ell$;

1: инициализация: $\gamma_i := 1$, $i = 1, \ldots, \ell$;
2: **повторять**
3: **для всех** объектов $i = 1, \ldots, \ell$
4: вычислить оценки скользящего контроля:

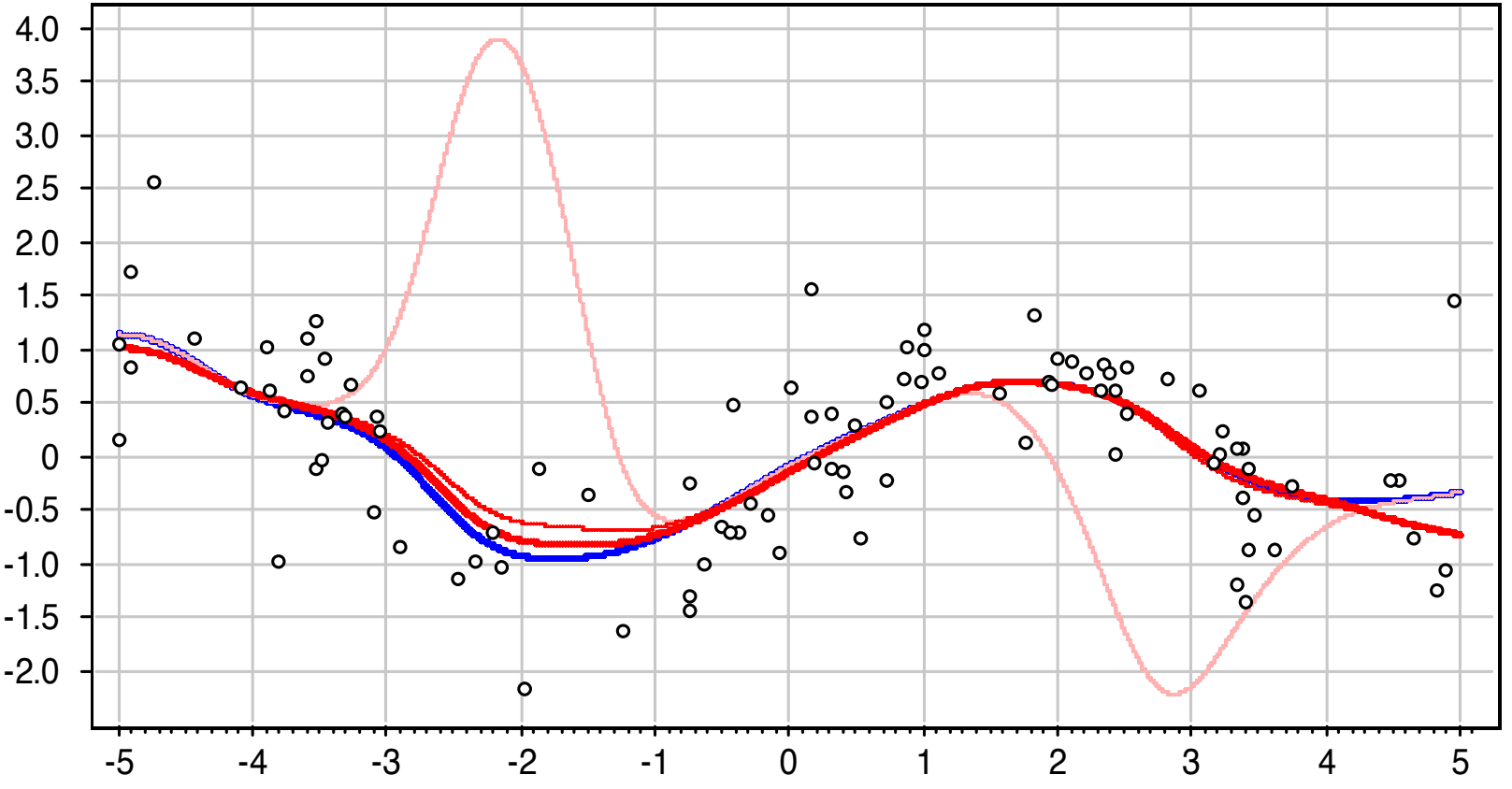
$$a_i := a_h\big(x_i; X^\ell \setminus \{x_i\}\big) = \frac{\sum\limits_{j=1,\, j\neq i}^{\ell} y_j \gamma_j K\left(\frac{\rho(x_i,x_j)}{h(x_i)}\right)}{\sum\limits_{j=1,\, j\neq i}^{\ell} \gamma_j K\left(\frac{\rho(x_i,x_j)}{h(x_i)}\right)};$$

5: **для всех** объектов $i = 1, \ldots, \ell$
6: $\gamma_i := \tilde{K}\big(|a_i - y_i|\big)$;
7: **пока** коэффициенты γ_i не стабилизируются;

Пример работы LOWESS на синтетических данных

$\ell = 100$, $h = 1.0$, гауссовское ядро $K(r) = \exp\left(-2r^2\right)$
Две из 100 точек — выбросы с ординатами $y_i = 40$ и -40

В данном случае LOWESS сошёлся за 2–3 итерации:



Резюме

- В статистике методы, устойчивые к нарушениям модельных предположений о данных, называются *робастными*. Мы рассмотрели простой робастный метод, устойчивый к наличию небольшого числа выбросов.
- В этом методе происходит обучение весов объектов.

Обучение регрессии — это оптимизация

Обучающая выборка: $X^\ell = (x_i, y_i)_{i=1}^\ell, \quad x_i \in \mathbb{R}^n, \quad y_i \in \mathbb{R}$

1. Модель регрессии — *линейная*:

$$a(x, w) = \langle x, w \rangle = \sum_{j=1}^{n} f_j(x) w_j, \qquad w \in \mathbb{R}^n$$

2. Функция потерь — *квадратичная*:

$$\mathscr{L}(a, y) = (a - y)^2$$

3. Метод обучения — *метод наименьших квадратов*:

$$Q(w) = \sum_{i=1}^{\ell} \big(a(x_i, w) - y_i\big)^2 \to \min_w$$

4. Проверка по тестовой выборке $X^k = (\tilde{x}_i, \tilde{y}_i)_{i=1}^k$:

$$\bar{Q}(w) = \frac{1}{k} \sum_{i=1}^{k} \big(a(\tilde{x}_i, w) - \tilde{y}_i\big)^2$$

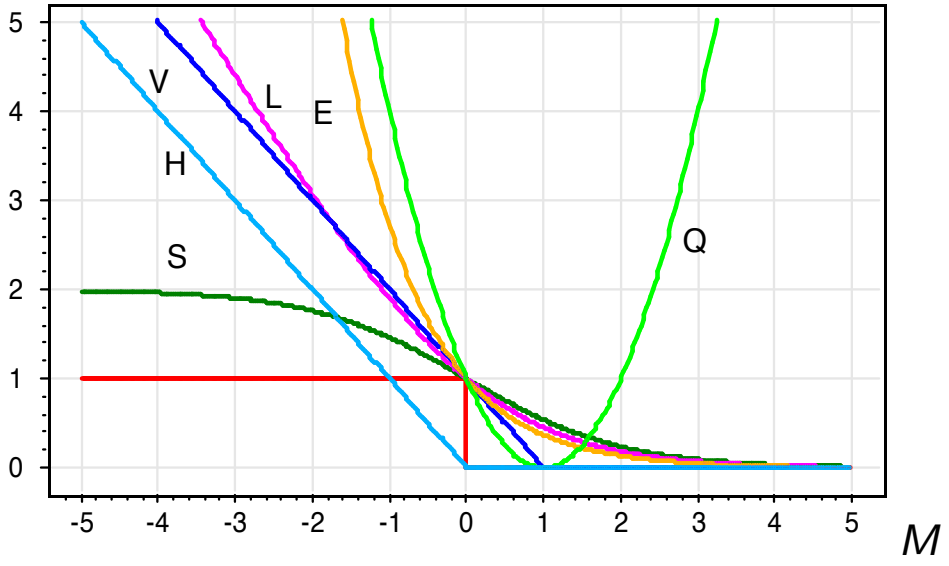
Обучение классификации — тоже оптимизация

Обучающая выборка: $X^\ell = (x_i, y_i)_{i=1}^\ell$, $x_i \in \mathbb{R}^n$, $y_i \in \{-1, +1\}$

1. Модель классификации — *линейная*:
$$a(x, w) = \text{sign}\langle x, w\rangle$$
2. Непрерывная аппроксимация бинарной функции потерь:
$$\mathscr{L}(a, y) = \big[\langle x_i, w\rangle y_i < 0\big] \leqslant \mathscr{L}\big(\langle x_i, w\rangle y_i\big),$$
где $M_i(w) = \langle x_i, w\rangle y_i$ — *отступ* (margin) объекта x_i
3. Метод обучения — *минимизация эмпирического риска*:
$$Q(w) = \sum_{i=1}^{\ell} \big[a(x_i, w) y_i < 0\big] \leqslant \sum_{i=1}^{\ell} \mathscr{L}\big(\langle x_i, w\rangle y_i\big) \to \min_w$$
4. Проверка по тестовой выборке $X^k = (\tilde{x}_i, \tilde{y}_i)_{i=1}^k$:
$$\bar{Q}(w) = \frac{1}{k} \sum_{i=1}^{k} \big[\langle \tilde{x}_i, w\rangle \tilde{y}_i < 0\big]$$

Непрерывные аппроксимации пороговой функции потерь

Часто используемые непрерывные функции потерь $\mathscr{L}(M)$:



$V(M) = (1 - M)_+$ — кусочно-линейная (SVM);

$H(M) = (-M)_+$ — кусочно-линейная (Hebb's rule);

$L(M) = \log_2(1 + e^{-M})$ — логарифмическая (LR);

$Q(M) = (1 - M)^2$ — квадратичная (FLD);

$S(M) = 2(1 + e^{M})^{-1}$ — сигмоидная (ANN);

$E(M) = e^{-M}$ — экспоненциальная (AdaBoost);

$[M < 0]$ — пороговая функция потерь.

Градиентный метод численной минимизации

Минимизация эмпирического риска:

$$Q(w) = \sum_{i=1}^{\ell} \mathscr{L}\big(\langle x_i, w\rangle y_i\big) = \sum_{i=1}^{\ell} \mathscr{L}_i(w) \to \min_w .$$

Численная минимизация методом *градиентного спуска*:

$$w^{(0)} := \text{начальное приближение;}$$

$$w^{(t+1)} := w^{(t)} - h \cdot \nabla Q(w^{(t)}), \qquad \nabla Q(w) = \left(\frac{\partial Q(w)}{\partial w_j}\right)_{j=0}^{n},$$

где h — *градиентный шаг*, называемый также *темпом обучения*.

$$w^{(t+1)} := w^{(t)} - h \sum_{i=1}^{\ell} \nabla \mathscr{L}_i\big(w^{(t)}\big).$$

Идея ускорения сходимости:
брать (x_i, y_i) по одному и сразу обновлять вектор весов.

Алгоритм SG (Stochastic Gradient)

Вход: выборка X^ℓ, темп обучения h, темп забывания λ;
Выход: вектор весов w;

инициализировать веса w_j, $j = 1, \ldots, n$;
инициализировать оценку функционала: $\bar{Q} := \frac{1}{\ell} \sum_{i=1}^{\ell} \mathscr{L}_i(w)$;
повторять
    выбрать объект x_i из X^ℓ случайным образом;
    вычислить потерю: $\varepsilon_i := \mathscr{L}_i(w)$;
    сделать градиентный шаг: $w := w - h\nabla\mathscr{L}_i(w)$;
    оценить функционал: $\bar{Q} := (1 - \lambda)\bar{Q} + \lambda\varepsilon_i$;
пока значение $\bar{Q}$ и/или веса w не сойдутся;

Robbins, H., Monro S. A stochastic approximation method // Annals of Mathematical Statistics, 1951, 22 (3), p. 400–407.

Откуда взялась такая оценка функционала?

Проблема: после каждого шага w по одному объекту x_i, не хотелось бы оценивать Q по всей выборке $x_1, \ldots, x_\ell$.

Решение: использовать рекуррентную формулу.

Среднее арифметическое $\bar{Q}_m = \frac{1}{m}\sum_{i=1}^{m} \varepsilon_i$:

$$\bar{Q}_m = (1 - \tfrac{1}{m})\bar{Q}_{m-1} + \tfrac{1}{m}\varepsilon_m.$$

Экспоненциальное скользящее среднее

$$\bar{Q}_m := (1 - \lambda)\bar{Q}_{m-1} + \lambda\varepsilon_m;$$

$$\bar{Q}_m = \lambda\varepsilon_m + \lambda(1-\lambda)\varepsilon_{m-1} + \lambda(1-\lambda)^2\varepsilon_{m-2} + \lambda(1-\lambda)^3\varepsilon_{m-3} + \ldots$$

Чем больше λ, тем быстрее забывается предыстория ряда.
Параметр $\lambda \approx \frac{1}{m}$ называется *темпом забывания*.

Алгоритм SAG (Stochastic Average Gradient)

Вход: выборка X^ℓ, темп обучения h, темп забывания λ;
Выход: вектор весов w;

инициализировать веса w_j, $j = 1, \ldots, n$;
инициализировать градиенты: $G_i := \nabla \mathscr{L}_i(w)$, $i = 1, \ldots, \ell$;
инициализировать оценку функционала: $\bar{Q} := \frac{1}{\ell} \sum_{i=1}^{\ell} \mathscr{L}_i(w)$;
повторять
    выбрать объект x_i из X^ℓ случайным образом;
    вычислить потерю: $\varepsilon_i := \mathscr{L}_i(w)$;
    вычислить градиент: $G_i := \nabla \mathscr{L}_i(w)$;
    сделать градиентный шаг: $w := w - h \sum_{i=1}^{\ell} G_i$;
    оценить функционал: $\bar{Q} := (1 - \lambda)\bar{Q} + \lambda \varepsilon_i$;
пока значение $\bar{Q}$ и/или веса w не сойдутся;

Schmidt M., Le Roux N., Bach F. Minimizing finite sums with the stochastic average gradient // arXiv.org, 2013.

Резюме

- *Непрерывная аппроксимация пороговой функции потерь* $[M < 0] \leqslant \mathscr{L}(M)$ позволяет использовать градиентную оптимизацию и повышает качество классификации благодаря увеличению зазора между классами.
- SG легко обобщается для нелинейных моделей $g(x, w)$
- и для любых функций потерь, $\mathscr{L}_i(w) = \mathscr{L}\big(g(x_i, w), y_i\big)$.
- SG допускает онлайновое (потоковое) обучение.
- SG подходит для больших данных, т. к. даёт неплохие решения, даже не обработав всю выборку (x_i, y_i).

Метод стохастического градиента (Stochastic Gradient)

Достоинства:

1. легко реализуется;
2. применим к любым моделям и функциям потерь;
3. допускает онлайновое (потоковое) обучение;
4. на сверхбольших выборках позволяет получать неплохие решения, даже не обработав все (x_i, y_i);
5. всё чаще применяется для Big Data.

Недостатки:

1. возможно застревание в локальных экстремумах;
2. возможна расходимость или медленная сходимость;
3. возможно переобучение;
4. подбор комплекса эвристик является искусством.

Варианты инициализации весов

1. $w_j := 0$ для всех $j = 0, \ldots, n$;
2. небольшие случайные значения:
   $w_j := \text{random}\left(-\frac{1}{2n}, \frac{1}{2n}\right)$;
3. $w_j := \frac{\langle y, f_j \rangle}{\langle f_j, f_j \rangle}$, $f_j = \left(f_j(x_i)\right)_{i=1}^{\ell}$ — вектор значений признака;

   эта оценка w оптимальна при квадратичной функции потерь, если признаки некоррелированы, $\langle f_j, f_k \rangle = 0$, $j \neq k$.
4. $w_j := \ln \frac{\sum_i [y_i = +1] f_j(x_i)}{\sum_i [y_i = -1] f_j(x_i)} \frac{\sum_i [y_i = -1]}{\sum_i [y_i = +1]}$;

   эта оценка w оптимальна для задач классификации, $Y = \{-1, +1\}$, если признаки независимы.
5. оценки w_j по небольшой случайной подвыборке объектов;
6. мультистарт: многократные запуски из разных случайных начальных приближений и выбор лучшего решения.

Варианты порядка предъявления объектов

Возможны варианты:

1. *перетасовка объектов* (shuffling):
   попеременно брать объекты из разных классов;
2. чаще брать те объекты, на которых была допущена бóльшая ошибка
   (чем меньше M_i, тем больше вероятность взять объект)
   (чем меньше $|M_i|$, тем больше вероятность взять объект);
3. вообще не брать «хорошие» объекты, у которых $M_i > \mu_+$
   (при этом немного ускоряется сходимость);
4. вообще не брать объекты-«выбросы», у которых $M_i < \mu_-$
   (при этом может улучшиться качество классификации);

Параметры μ_+, μ_- придётся подбирать.

Варианты выбора градиентного шага

1. сходимость гарантируется (для выпуклых функций) при
$$h_t \to 0, \quad \sum_{t=1}^{\infty} h_t = \infty, \quad \sum_{t=1}^{\infty} h_t^2 < \infty,$$
в частности можно положить $h_t = 1/t$;
2. *метод скорейшего градиентного спуска*:
$$\mathscr{L}_i\big(w - h\nabla\mathscr{L}_i(w)\big) \to \min_h,$$
позволяет найти *адаптивный шаг* h^*;

   при квадратичной функции потерь $h^* = \|x_i\|^{-2}$;
3. периодически можно делать пробные случайные шаги для «выбивания» из локальных минимумов;
4. метод Левенберга-Марквардта (второго порядка)

Диагональный метод Левенберга-Марквардта

Метод Ньютона-Рафсона, $\mathscr{L}_i(w) \equiv \mathscr{L}(\langle w, x_i \rangle y_i)$:

$$w := w - h\big(\mathscr{L}_i''(w)\big)^{-1} \nabla \mathscr{L}_i(w),$$

где $\mathscr{L}_i''(w) = \left(\frac{\partial^2 \mathscr{L}_i(w)}{\partial w_j \partial w_{j'}}\right)$ — гессиан, $n \times n$-матрица

Эвристика: считаем, что гессиан диагонален. Тогда

$$w_j := w_j - h\left(\frac{\partial^2 \mathscr{L}_i(w)}{\partial w_j^2} + \mu\right)^{-1} \frac{\partial \mathscr{L}_i(w)}{\partial w_j},$$

h — темп обучения, можно полагать $h = 1$
μ — параметр, предотвращающий обнуление знаменателя.

Отношение h/μ есть темп обучения на ровных участках функционала $\mathscr{L}_i(w)$, где вторая производная обнуляется.

Проблема переобучения

Возможные причины переобучения:

1. линейная зависимость (мультиколлинеарность) признаков:
   пусть построен классификатор: $a(x, w) = \text{sign}\langle w, x\rangle$;
   мультиколлинеарность: $\exists u \in \mathbb{R}^{n+1}$: $\forall x \quad \langle u, x\rangle \equiv 0$;
   тогда $\forall \gamma \in \mathbb{R} \quad a(x, w) = \text{sign}\langle w + \gamma u, x\rangle$
2. слишком мало объектов; слишком много признаков;

Проявления переобучения:

1. слишком большие веса $|w_j|$ разных знаков;
2. неустойчивость классификаций $a(x, w)$ относительно погрешностей измерения признаков;
3. $Q(X^\ell) \ll Q(X^k)$;

Регуляризация (сокращение весов, weight decay)

Штраф за увеличение нормы вектора весов:

$$\widetilde{\mathscr{L}_i}(w) = \mathscr{L}_i(w) + \frac{\tau}{2}\|w\|^2 = \mathscr{L}_i(w) + \frac{\tau}{2}\sum_{j=1}^{n} w_j^2 \to \min_w .$$

Градиент:

$$\nabla\widetilde{\mathscr{L}_i}(w) = \nabla\mathscr{L}_i(w) + \tau w.$$

Модификация градиентного шага:

$$w := w(1 - h\tau) - h\nabla\mathscr{L}_i(w).$$

Подбор параметра τ — по скользящему контролю

Резюме в конце лекции

- В методе стохастического градиента необходимы различные эвристики для улучшения сходимости и получения лучшего решения.
- *Регуляризация* решает проблему мультиколлинеарности и снижает риск переобучения.