

Лекция 2:

Задачи машинного обучения, признаковое пространство, метрический подход

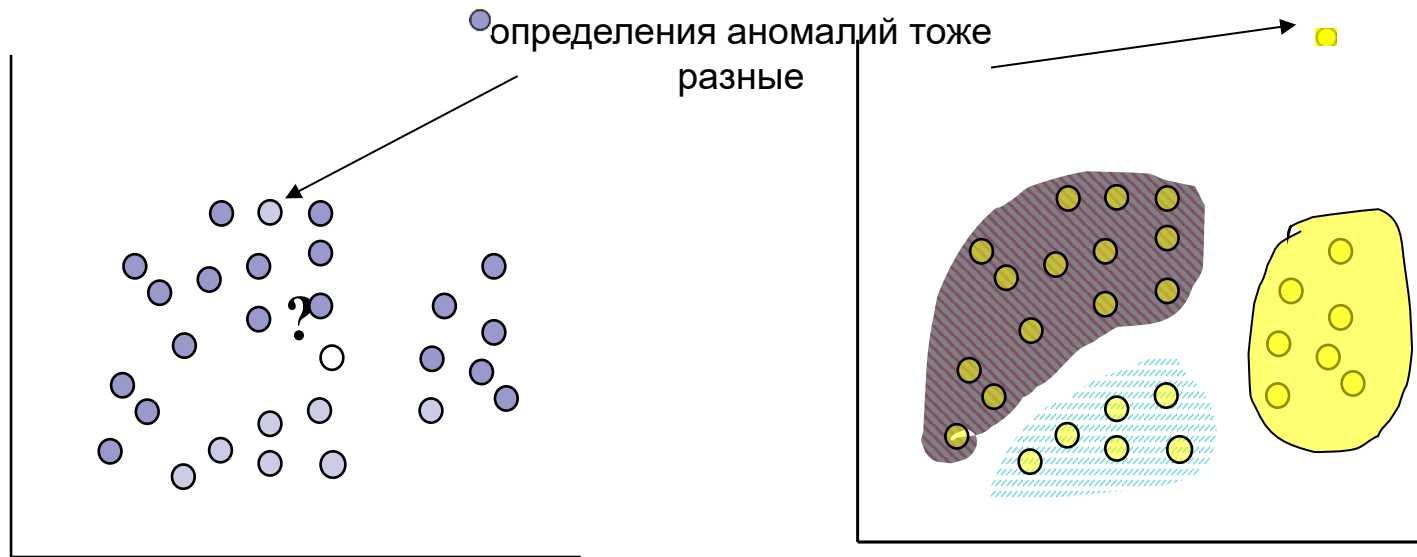
Базовые задачи машинного обучения = типы выявляемых закономерностей

- Классификация («Обучение с учителем»)
 - Отнесение объектов к заранее определенным категориям
- Ранжирование («Обучение с учителем»)
 - Оценка степени соответствия объектов одной или более заранее определенным категориям
- Прогнозирование («Обучение с учителем»)
 - На основании известных значений атрибутов анализируемого объекта определяются значения неизвестных атрибутов
- Ассоциации («Обучение без учителя»)
 - Выявление зависимостей между атрибутами в виде правил или аналитических зависимостей, выявление скрытых свойств объектов
- Кластеризация («Обучение без учителя»)
 - Выделение компактных подгрупп «похожих» объектов
- Выявление исключений («Обучение с учителем и без»)
 - Поиск объектов, которые своими характеристиками значительно отличаются от остальных

Примеры типов исходных данных

- Транзакционные
 - Объекты анализа – «события» различной структуры с числовыми и категориальными атрибутами и с временной меткой
- Табличные
 - Объекты анализа представлены в виде реляционных таблиц, возможно взаимосвязанных (заданно ER-схемой), имеют разнотипные атрибуты
- Временные ряды и числовые данные большого объема
 - Обработка результатов наблюдений, научных экспериментов, характеристик технологических процессов
- Электронные тексты на естественном языке
 - анализ содержимого документов
- Графовые данные
 - Анализ взаимосвязей (SNA)
- Специализированные данные
 - Мультимедия, геоданные, ДНК, программный код и многое другое

Обучение с учителем и без



- «Размеченный» набор данных – выделен один или более признаков, которые могут быть неизвестны и которые нужно предсказывать, тогда задача обучения «с учителем», иначе «без учителя» («неразмеченный» набор данных):
 - «Выходные» признаки - нужно предсказывать (они же отклики, или «зависимые переменные», или ...)
 - «Входные» признаки, которые считаются всегда известными (они же входы, или «независимые переменные», или регрессоры, ...)

Исходные данные и признаковое пространство

- Объект анализа (или прецедент, или кейс, или наблюдение, ...) x из некоторого возможно бесконечного множества объектов X задается набором признаков f_j (или атрибутов, или свойств, ...)

$$f_j: X \rightarrow D_j, x \in X$$

- Домен (область определения, множество значений, тип) признака:
 - Категориальный: D_j как правило конечно, нет расстояния, не задан порядок
 - Ординальный (порядковый): как категориальные, но задан порядок в виде транзитивного, антисимметричного отношения, но нет расстояний
$$R_j: D_j \times D_j, (x_a R_j x_b \wedge x_b R_j x_c) \Rightarrow x_a R_j x_c, (x_a R_j x_b \wedge x_b R_j x_a) \Rightarrow x_a = x_b$$
 - Числовой – есть расстояние, $D_j = \mathbb{R}$
 - Бинарный $D_j = \{0,1\}$
- Вектор признаков, описывающий объект $(f_1(x), f_2(x), \dots, f_p(x))$

Пространство признаков и выборки (наборы данных)

- Набор данных (выборка) - множество $Z = \{x_1, \dots, x_l\}$, включает l объектов (наблюдений) из X и может быть представлен:

- Матрицей признаков

$$\|F\|_{l \times p} = \begin{pmatrix} f_1(x_1) & f_2(x_1) & \dots & f_p(x_1) \\ \dots & \dots & \dots & \dots \\ f_1(x_l) & f_2(x_l) & \dots & f_p(x_l) \end{pmatrix}$$

- Симметричной матрицей сходства (или различия)

$$\|K\|_{l \times l} = \begin{pmatrix} d(x_1, x_1) & d(x_1, x_2) & \dots & d(x_1, x_l) \\ \dots & \dots & \dots & \dots \\ d(x_l, x_1) & d(x_l, x_2) & \dots & d(x_l, x_l) \end{pmatrix}$$

где $d: X \times X \rightarrow \mathbb{R}$ некая симметричная, мера сходства или различия, не обязательно расстояние (правило треугольника может не выполняться)

- Вопрос построения признакового пространства – не простой, а иногда и критический, причины:
 - Мусор на вход – мусор на выход
 - Сложные структурированные объекты (например, графы, тексты и т.д.)
 - Противоречия, пропуски, ошибки, артефакты
 - Взаимозависимые и незначимые признаки

Обучение без учителя

- Иногда называют задачей «самоорганизации»
- Все признаки равнозначны, нет отклика, поэтому:
 - Обучение без учителя более «субъективное» (нет единых интуитивно понятных мер оценки качества типа «точности») и менее «автоматизируемо»
 - Сложнее подбирать метапараметры и сравнивать полученные модели
- Важность этого направления велико:
 - «Истинный data mining» - ищет неизвестные заранее зависимости без «подсказок» эксперта. Много важных прикладных задач по сегментации, выявление скрытых характеристик, зависимостей между атрибутами и т.д.
 - Для больших данных получить качественно размеченный набор тяжело или невозможно, часто возникают задачи *semisupervised learning*, когда размечена лишь часть набора
- Большинство задач сводится к поиску скрытых (латентных) признаков или скрытых структур (групп, отношений, зависимостей ...) в данных:
$$z_j: D_1 \times \dots \times D_p \rightarrow D_{z_j}, z_1 = F_1(f_1, \dots, f_p), \dots, z_k = F_k(f_1, \dots, f_p)$$
 - Основные случаи: z_j или отношения (например, кластеры, ассоциативные правила) или новые признаки (например, главные компоненты и степень принадлежности кластеру), иногда как в SOM (сетях Кохонена) и то, и другое сразу.

Обучение с учителем

- Дано: множество «размеченных» примеров :

- обучающая выборка или тренировочный набор:

$$Z = \{(x_i, y_i)\}_{i=1}^l \in X \times Y$$

- $y_i \in Y$: известный «отклик» и его множество значений
 - Неизвестная зависимость, которую необходимо «восстановить» $y: X \rightarrow Y$

- Постановка задачи:

- Найти алгоритм (или гипотезу, или модель, или решающую функцию)

$$a_Z: X \rightarrow Y$$

- «качественно» приближающую неизвестную $y: X \rightarrow Y$ на всем признаковом пространстве

- Два этапа:

- Обучение (построение модели), метод обучения μ выбирает «лучший» алгоритм a (модель, гипотезу) среди заданного семейства A :

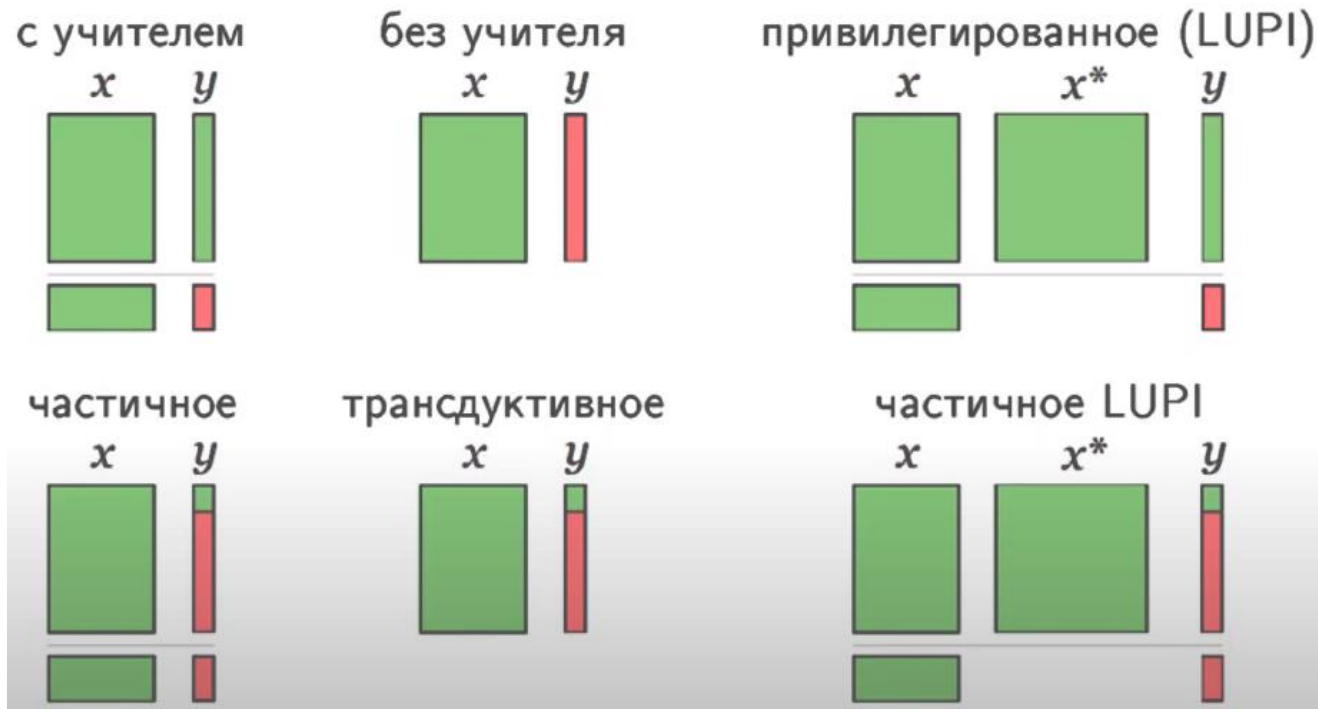
$$a_Z = \mu(Z) \in A$$

- Применение (скоринг модели), алгоритм a выдает (прогнозирует) значения отклика $\hat{y} = a_Z(\hat{x})$ для «новых» объектов (возможно для множества «новых» объектов) с неизвестным откликом

Типы задач обучения с учителем в зависимости от типа отклика

- Вообще тип задачи обучения с учителем определяются не только типом допустимых значений «отклика», но и **оценкой качества**, которая используется для выбора модели
- Типы задач:
 - Бинарная классификация $Y = \{0,1\}$ – решающая функция бинарная
 - Много-классовая классификация $Y = \{C_1, \dots, C_k\}$ – категориальный признак, значения взаимоисключающие, не задано отношение порядка, наблюдение не может принадлежать нескольким классам одновременно, решающая функция дискретная
 - Много-темная (multi-label) классификация $Y = \{0,1\}^k$ или Y – множество всех подмножеств $\{C_1, \dots, C_k\}$, решающая вектор функция выдает бинарный вектор релевантных тем
 - Регрессия $Y = \mathbb{R}$ или $Y = \mathbb{R}^k$, решающая функция – вещественная (вектор) функция
 - Порядковая регрессия $Y = \{1, \dots, K\}$ – дискретный признак, задано отношение порядка, решающая функция дискретная
 - Ранжирование $Y = \{C_1, \dots, C_k\}$ – категориальный признак, значения взаимоисключающие, решающая вектор функция выдает вектор степеней соответствия наблюдений каждому из классов

Типы задач обучения в зависимости от доступности разметки



- Трансдуктивное обучение – тестовая выборка известна заранее
- Привилегированное обучение – часть признаков известна только на этапе обучения

Другие виды обучения с учителем

- Обучение с подкреплением (reinforcement learning) – агент обучается итерационно в зависимости от отклика среды и своего состояния
- (До)обучение «на лету» (online learning) – корректировка модели по одному примеру или пакетом новых примеров
- Многозадачное (multi-task) – решается несколько ML задач сразу
- Активное обучение – процесс обучения влияет на формирование обучающей выборки
- Meta-learning – обучение как обучать модели машинного обучения
- Прогнозирование сложных откликов - отклик не атрибут, а сложный объект, например, граф или текст или последовательность действий как в MBR

Функция потерь

- $L: Y \times Y \rightarrow \mathbb{R}^+$ характеризует отличие истинного отклика от спрогнозированного $L(y(x), a(x))$

- Примеры:

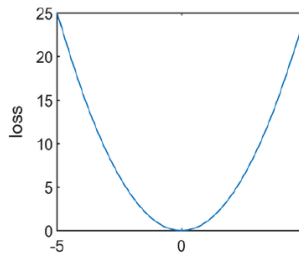
- Классификация и регрессия:

$$L(y, y') = [y \neq y'],$$

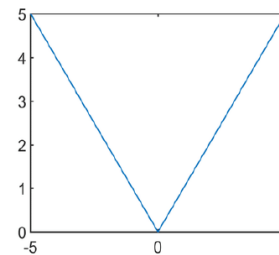
$$L(y, y') = |y - y'|,$$

$$L(y, y') = (y - y')^2,$$

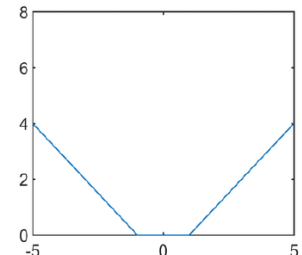
$$L(y, y') = [|y - y'| > \varepsilon]$$



(a) square loss



(b) absolute loss



(f) ϵ -insensitive loss ($\epsilon = 1$)

- Много-темная классификация:

$$L(y, y') = |y \nabla y'|, a \nabla b = (a \cap b) \setminus (a \cup b), a \subseteq Y, b \subseteq Y$$

- Ранжирование:

$$L(y, y') = \frac{|\{(r, s): y_r \leq y_s \wedge y'_r \leq y'_s\}|}{|y||y'|}$$

Функционал качества

- Теоретический риск:

$$E[L(y(x), a(x))] = \int_{X \times Y} L(y(x), a(x)) dP(x, y) \rightarrow \min$$

- Но $P(x, y)$ мы не знаем, а если бы знали, то и решать ничего не нужно, поэтому используем эмпирический риск:

$$Q(a, Z) = \frac{1}{l} \sum_Z L(y_i, a(x_i)) \rightarrow \min$$

- Получаем, что метод обучения μ выбирает «лучший» алгоритм a_Z^* (модель, гипотезу) среди заданного семейства A на наборе данных Z как:

$$a_Z^* = \mu(Z) = \arg \min_{a \in A} Q(a, Z)$$

- Можно ли использовать оценку качества на обучающей выборке как объективную? Ответ – НЕТ

Гипотеза о компактности (непрерывности)

- Не формальная (эвристическая) гипотеза:

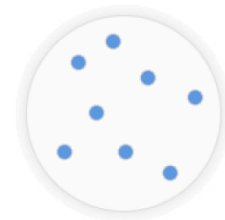
- «Близкие» наблюдения в признаковом пространстве должны обладать похожими свойствами, для прогнозирования - иметь «похожие» отклики

- Непрерывность (для регрессии)

выполнена:

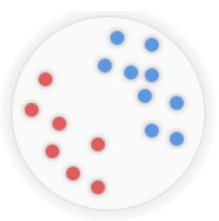


не выполнена:

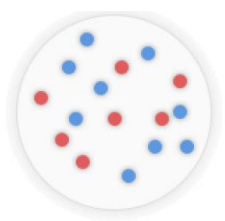


- Компактность (для классификации)

выполнена:



не выполнена:



Расстояния для числовых признаков

- Обычно строится на основе метрики:

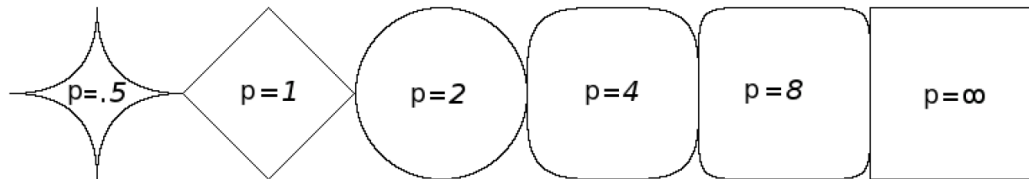
$$d(i,j) \geq 0, d(i,i) = 0, d(i,j) = d(j,i), d(i,j) \leq d(i,k) + d(k,j)$$

- Наиболее популярно расстояние Минковского:

$$d(i,j) = \sqrt[p]{|x_{i1} - x_{j1}|^p + |x_{i2} - x_{j2}|^p + \dots + |x_{in} - x_{jn}|^p}$$

где $i = (x_{i1}, x_{i2}, \dots, x_{in})$ и $j = (x_{j1}, x_{j2}, \dots, x_{jn})$ - два объекта с n числовыми атрибутами, p - положительное целое число

- «Изолинии» (области точек равноудаленных от заданной):



- $p = 2$ - Евклидово расстояние: $d(i,j) = \sqrt{(|x_{i1} - x_{j1}|^2 + \dots + |x_{in} - x_{jn}|^2)}$
- $p = 1$ - расстояние «Манхэттен»: $d(i,j) = |x_{i1} - x_{j1}| + \dots + |x_{in} - x_{jn}|$

Приведение к близким шкалам

- Цель – исключить влияние разброса значений признака и уменьшить влияние выбросов (l – размер выборки)
- Нормализация на абсолютное отклонение более робастно (устойчиво к выбросам), чем через стандартное отклонение:

$$x'_{if} = \frac{x_{if} - m_f}{s_f}, s_f = \frac{1}{l} (|x_{1f} - m_f| + |x_{2f} - m_f| + \dots + |x_{lf} - m_f|),$$

$$m_f = \frac{1}{l} (x_{1f} + x_{2f} + \dots + x_{lf}) \text{ // можно использовать медиану}$$

- Стандартизация:

$$x'_{if} = \frac{x_{if} - m_f}{std_f}, std_f = \sqrt{\frac{1}{l-1} [(x_{1f} - m_f)^2 + (x_{2f} - m_f)^2 + \dots + (x_{nf} - m_f)^2]}$$

Бинарные признаки

- Расстояние Хэмминга = сумма единиц после XOR: $(M_{10} + M_{01})$
- Таблица «сопряженных признаков»
 - В ячейках – число совпадающих и несовпадающих значений из p бинарных атрибутов для объектов j и i

		Объект j		
		1	0	sum
Объект i	1	M_{11}	M_{10}	$M_{11} + M_{10}$
	0	M_{01}	M_{00}	$M_{01} + M_{00}$
sum		$M_{11} + M_{01}$	$M_{10} + M_{00}$	$M_{00} + M_{01} + M_{10} + M_{11}$

- На основе коэффициента совпадения $d(i, j) = \frac{M_{10} + M_{01}}{M_{11} + M_{01} + M_{10} + M_{00}}$
 - для симметричных атрибутов (значения равнозначны)
- На основе коэффициента Джакара $d(i, j) = \frac{M_{10} + M_{01}}{M_{11} + M_{01} + M_{10}}$
 - для асимметричных атрибутов (единица важнее)

Пример

Имя	Пол	Жар	Кашель	Test-1	Test-2	Test-3	Test-4
Jack	M	Y	N	P	N	N	N
Mary	F	Y	N	P	N	P	N
Jim	M	Y	P	N	N	N	N

- пол - симметричный атрибут
- остальные ассиметричные
- пусть Y и P соответствует 1, а N соответствует 0

$$d(jack, mary) = \frac{0 + 1}{2 + 0 + 1} = 0.33$$

$$d(jack, jim) = \frac{1 + 1}{1 + 1 + 1} = 0.67$$

$$d(jim, mary) = \frac{1 + 2}{1 + 1 + 2} = 0.75$$

Категориальные признаки и шкалы

- Категориальные признаки:

- ☐ может быть много значений

- Простое совпадение

- ☐ m - число совпадений, p - число категориальных переменных (аналог нормированного расстояния Хэмминга):

$$d(i, j) = \frac{p - m}{p}$$

- То же самое - кодирование бинарными векторами

- ☐ Для каждого значения категориального атрибута создается отдельная бинарная переменная: один категориальный атрибут с k возможными значениями => бинарный вектор длины k

- Отображение на числовую шкалу:

- ☐ Могут быть и дискретными и непрерывными
- ☐ сводятся к числовым: заменить значение переменной на его ранг или отобразить на $[0, 1]$ с нормировкой
- ☐ затем использовать стандартные расстояния

Категориальные признаки

Пример переменной	Мощность	Подход
Physical characteristics	10	Бинарное кодирование
Region		
Partnership status		
Education level	100	Преобразования или отображение на числовую шкалу
Urbanicity codes		
State		
Ethnicity	1000	Связывание
Employment classification		
Postal Code		
Address	1000000	Текстовые модели
Social security number	100000000	
text	Бесконечность	

Смешанные типы признаков

- Нелинейные шкалы:

- ☐ логарифмические, экспоненциальные и другие
- ☐ «обратное» преобразование к линейной шкале
- ☐ «экспертное» преобразование к ранговой шкале

- «Взвешенное» нормированное расстояние:

$$d(i, j) = \frac{\sum_{f=1}^p w_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^p w_{ij}^{(f)}}$$

- ☐ f - бинарный или номинальный: $d_{ij}^{(f)} = 0$ если $x_{if} = x_{jf}$, иначе $d_{ij}^{(f)} = 1$
- ☐ f - числовой: использовать нормализованное расстояние
- ☐ f - шкала: рассчитать ранги r_{if} , нормализовать и считать числовым

- Другие меры сходства:

- ☐ Ядерные функции (будут позже)
- ☐ Для однородны признаков

На основе корреляции

$$s(\vec{x}_i, \vec{x}_j) = \frac{cov_{\vec{x}_i \vec{x}_j}}{\sigma_{\vec{x}_i \vec{x}_i} \sigma_{\vec{x}_j \vec{x}_j}}$$

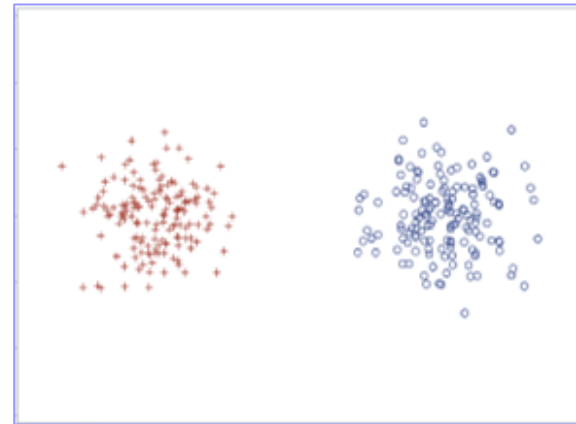
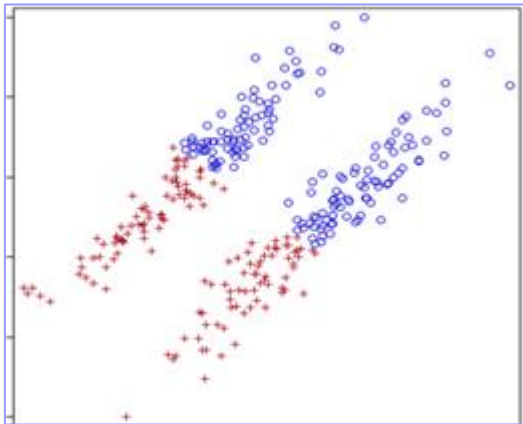
«Тригонометрические»:

$$s(\vec{x}_i, \vec{x}_j) = \frac{\langle \vec{x}_i, \vec{x}_j \rangle}{\sqrt{\langle \vec{x}_i, \vec{x}_i \rangle \langle \vec{x}_j, \vec{x}_j \rangle}}$$

Учет корреляций - расстояние Махалобиса

- Учет линейной корреляции в пространстве признаков, где Σ - ковариационная матрица по всей выборке

$$d(x^*, x) = (x^* - x)^T \Sigma^{-1} (x^* - x)$$



$$x \rightarrow z = \varphi(x) = \Sigma^{-1/2}(x - \mu)$$

Расстояние для сложных структур (пример)

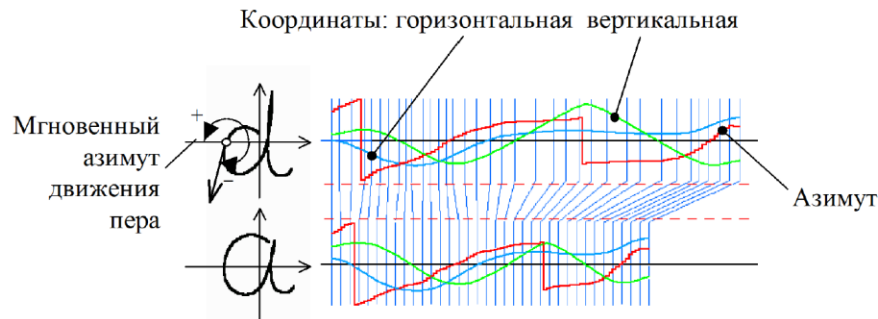
■ Расстояние Левенштейна для последовательностей:

- оценивает *минимальное* число операций (вставок и удалений) для сведения двух последовательностей к общей
- Эффективно считается алгоритмами динамического программирования

СТGGGCTAAAAGGTCCCTTAGCC...TTTAGAAAAA.GGGCCATTAGGAAATTGC
СТGGGACTAAA...CCTTAGCCATTTTACAAAAATGGGCCATTAGG...TTGC

■ DTW (и другие) – аналогично Левенштейну:

- Оценивают «затраты» на сведения двух структур к общей
- Эффективно считаются алгоритмами динамического программирования



Преобразованное пространство признаков с помощью ядер

- Преобразование пространства признаков с помощью отображения $\Phi: \mathbb{R}^p \mapsto H, x \mapsto \Phi(x)$, где скалярное произведение:

$$\langle \Phi(x_i), \Phi(x_j) \rangle$$

- НО: в явном виде не нужно вычислять новые признаки и образы, достаточно заменить скалярное произведение на ядерную функцию:

$$K(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$$

- Ядро – симметричная неотрицательно определенная функция $K: X \times X \rightarrow \mathbb{R}$:

□ симметрична: $\forall x_i, x_j: K(x_i, x_j) = K(x_j, x_i)$

□ неотрицательно определена: $\forall f: X \rightarrow \mathbb{R}$:

$$\int_X \int_X K(x_i, x_j) f(x_i) f(x_j) dx_i dx_j \geq 0$$

Примеры ядерных функций

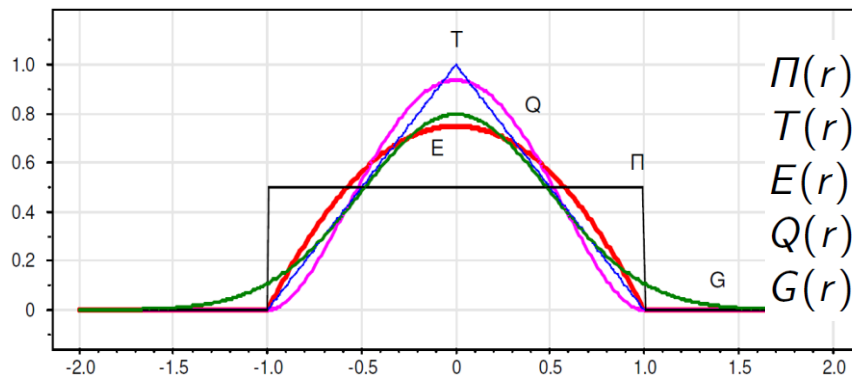
- Простые примеры:

- Линейное: $K(x_i, x_j) = \langle x_i, x_j \rangle$

- полиномиальное степени p : $K(x_i, x_j) = (1 + \langle x_i, x_j \rangle)^p$

- Радиально-базисные ядра:

- Одномерные ядра от расстояния (h – нормирующий параметр, ширина ядра): $K(x_j, x_i) = K\left(\frac{d(x_j, x_i)}{h}\right)$



$P(r) = [|r| \leq 1]$ — прямоугольное

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

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

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

$G(r) = \exp(-2r^2)$ — гауссовское

- K-spectrum ядро (для цепочек символов)

совпадение всех подстрок с весовым параметром $0 < \lambda < 1$

$$k_n(s, t) = \sum_{u \in \Sigma^n} (\varphi_u(s) \cdot \varphi_u(t)) = \sum_{u \in \Sigma^n} \sum_{i: u=s[i]} \sum_{j: u=t[j]} \lambda^{l(i)+l(j)}$$

Пропущенные значения

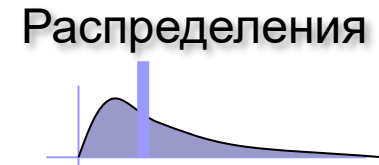
- Не все значения признаков известны или достоверны
 - важная задача, так как многие к ней сводятся (удаление шума, не консистентностей и т.д.)
- Причины появления пропущенных значений
 - Ошибки «оборудования» и/или ПО при получении данных от датчиков и из экспериментов
 - Удаление несогласованных значений атрибутов
 - Просто не введены в систему из-за халатности или ошибки
 - Часть данных может быть опциональна с точки зрения бизнес процессов организации, но важна для анализа
 - Не хранится правильная история изменений – невозможно правильно определить значение на момент анализа
- Пропущенные данные:
 - Ведут к неточным результатам анализа
 - Допускаются не всеми алгоритмами анализа

Методы обработки пропущенных значений

- Игнорировать объект или запись:
 - Можем потерять важные объекты (например, опорные вектора)
 - Можем «испортить» выборочное распределение
 - В некоторых задачах процент пропущенных значений велик ($>50\%$)
- Заполнение пропущенных значений «вручную»:
 - Нужен очень грамотный эксперт
 - Полностью «вручную» невозможно для больших объемов
 - Правила заполнения (импутации) трудно формулировать – проблема полноты, противоречивости, достоверности
- Использование глобальной спец. константы типа “unknown”
 - Не всеми алгоритмами анализа реализуемо
- Импутация «среднего» или «наиболее ожидаемого» значения
 - По всей выборке, по страте (срезу), по классу, по кластеру и т.д.
 - Наиболее популярный метод, но можем «испортить» выборочное распределение
 - Методы импутации на основе прогнозирования

Основные подходы к подстановке пропусков

- Импутация константным значением - все пропуски для переменной заменяются на:
 - Моду (для категориальных) или мат. ожидание, или пользовательскую константу или робастные оценки
- Импутация псевдослучайным значением:
 - В соответствии с распределением
- Импутация прогнозом (оценкой)
- Для неслучайных пропусков
 - индикаторные переменные
 - Одна на все наблюдение
 - Своя для каждой переменной



Оценки

$$x_i = f(x_1, \dots, x_p)$$

Что прогнозирует модель?

- Если есть несколько наблюдений с одинаковым вектором признаков x_* , но разными откликами $y_{*1}, y_{*2}, \dots, y_{*k}$, то какой прогноз минимизирует эмпирический риск (ошибку)? Ответ:

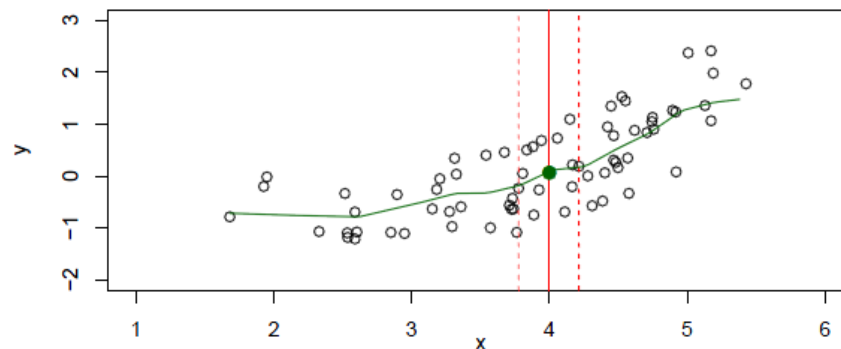
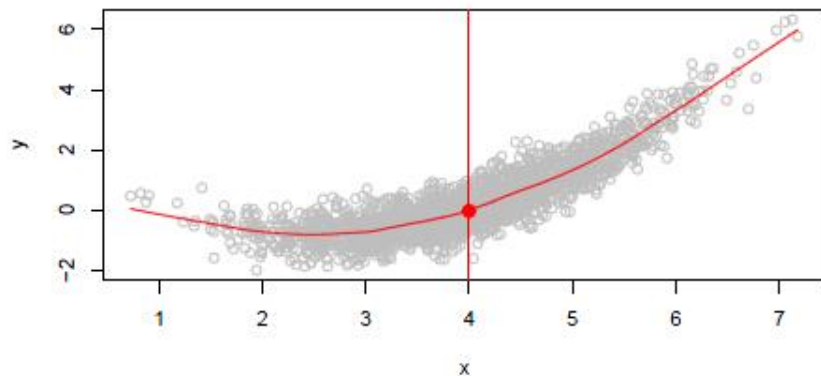
$$y_* = \arg \min_y Q(y, \{y_{*1}, y_{*2}, \dots, y_{*k}\}) = \arg \min_y \frac{1}{k} \sum_{*i} L(y_{*i}, y),$$

например, для квадратичной функции потерь: $y_* = E(y|x = x_*)$

- Если повторяющихся наблюдений нет, то можно «приблизить»:

$$y_* = E(y|x = x_*) \approx \text{mean}[y|x \in N(x_*)], \text{ где } N(x_*) - \text{«окрестность» } x_*$$

- Но должна выполняться гипотеза о компактности (или о непрерывности) и может быть проблема проклятия размерности



Обобщенный метрический классификатор

- Для выбранного x_* , ранжируем выборку $\{x^{(1)}, x^{(2)}, \dots, x^{(l-1)}\}$, так чтобы:

$$d(x_*, x^{(1)}) \leq d(x_*, x^{(2)}) \leq \dots \leq d(x_*, x^{(l-1)})$$

$x^{(i)}$ - i -й сосед x_* , а $y^{(i)}$ - отклик i -го соседа

- Метрический алгоритм классификации:

$$a(x, Z) = \arg \max_{y \in Y} \Gamma_y(x)$$

$\Gamma_y(x) = \sum_{i=1}^l [y = y^{(i)}] w(i, x)$ - оценка близости объекта x к классу y ,
 $w(i, x)$ – важность i -го соседа, не возрастает по i , неотрицательна

- Методы:

- «ближайшего соседа» $w(i, x) = [i \leq 1]$
- «к ближайших соседей» $w(i, x) = [i \leq k]$

Простые методы К ближайших соседей

■ Общая схема работы:

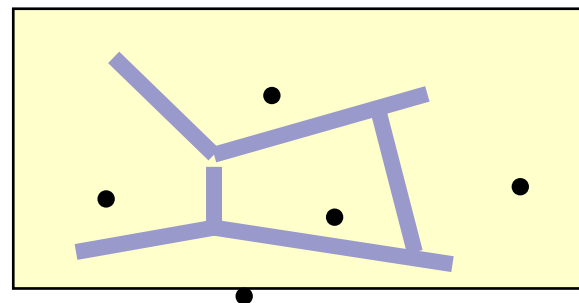
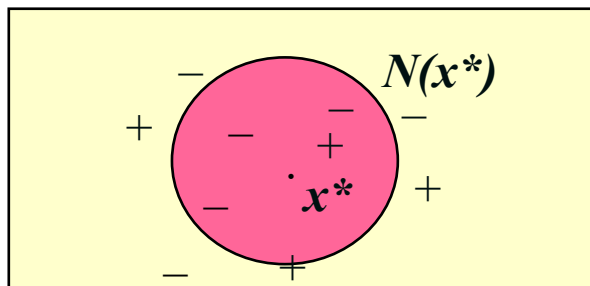
- Каждый пример – точка в пространстве, все примеры хранятся
- Вводится метрика расстояния с учетом нормирования координат
- Ищется К (от 1 до ...) ближайших соседей
- Прогноз вычисляется как агрегирующая функция от откликов найденных соседей

■ Метод KNN:

- Для задачи регрессии отклик считается как среднее по откликам всех соседей:
- Для классификации выбирается самый частый класс:

$$y^* = \frac{1}{K} \sum_{x_i \in N(x^*)} y_i$$

$$y^* = \arg \max_{c \in C, x_i \in N(x^*)} [|y_i = c|]$$



Метод «взвешенных» К ближайших соседей

- На базе KNN, но:
 - помимо распределения «отклика» учитываются и расстояния до соседей $w(i, x) = [i \leq k]w_i$, где w_i зависит от близости до x
- Примеры весов:
 - линейный вес $w_i = \frac{k+1-i}{k}$, экспоненциальный $w_i = q^i, 0 < q < 1$
 - ядра, например, $w_i = \exp(-d_i^2/h)$
- Усреднение отклика за счет:
 - «взвешенного» голосования для классификации и «взвешенного» среднего для регрессии (далее покажем откуда это следует)

$$y^* = \arg \max_{c \in C, x_i \in N(x^*)} \left[\frac{w_i |y_i = c|}{\sum_{x_j \in N(x^*)} w_j} \right]$$

$$y^* = \frac{\sum_{x_i \in N(x^*)} w_i y_i}{\sum_{x_i \in N(x^*)} w_i}$$

Метод К ближайших соседей с дискриминантным адаптивным расстоянием

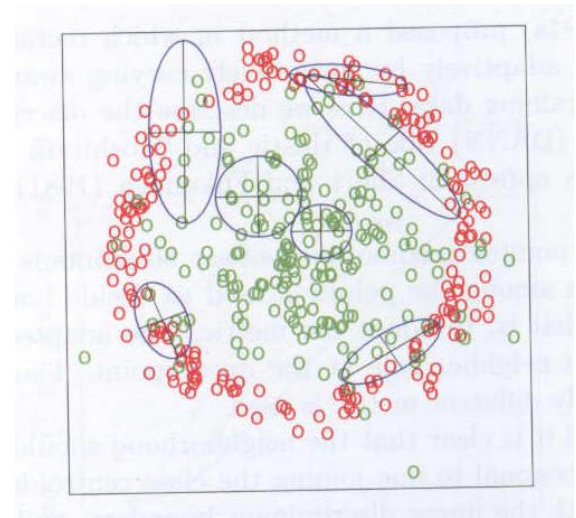
■ Метод DANN:

- На базе KNN, но используется итеративный (m-номер итерации) расчет локального расстояния Махаланобиса с учетом структуры распределения соседей в окрестности:

$$d^{(m)}(x^*, x_i) = (x^* - x_i)^T \Sigma_{(m)}^{-1} (x^* - x_i)$$

■ Параметры алгоритма:

- K_M – число соседей для оценки метрики (нужно побольше)
- K – число соседей для прогноза (лучше поменьше)



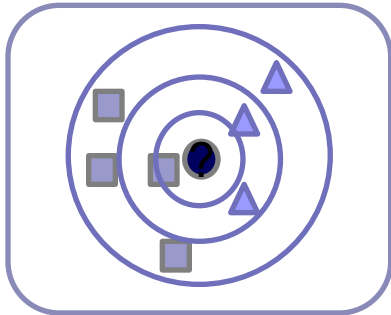
Выбор параметра K

Важность K:

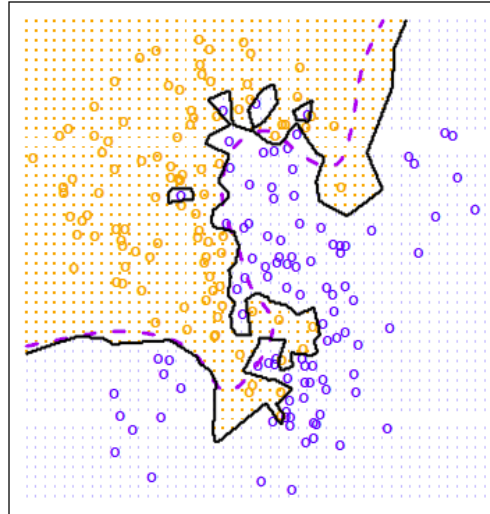
$k = 1$: Результат = квадрат

$k = 5$: Результат = треугольник

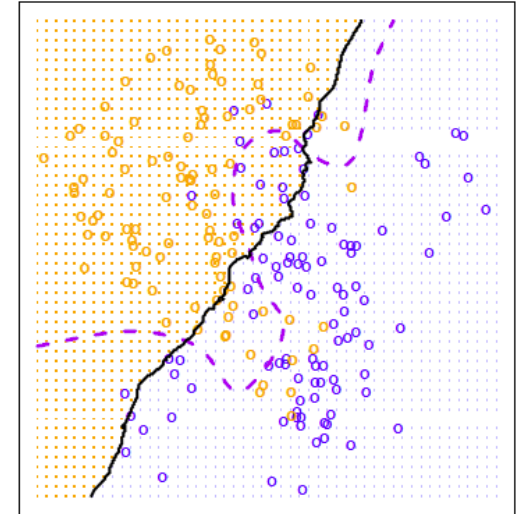
$k = 7$: Снова квадрат



KNN: K=1



KNN: K=100



■ Выбор k :

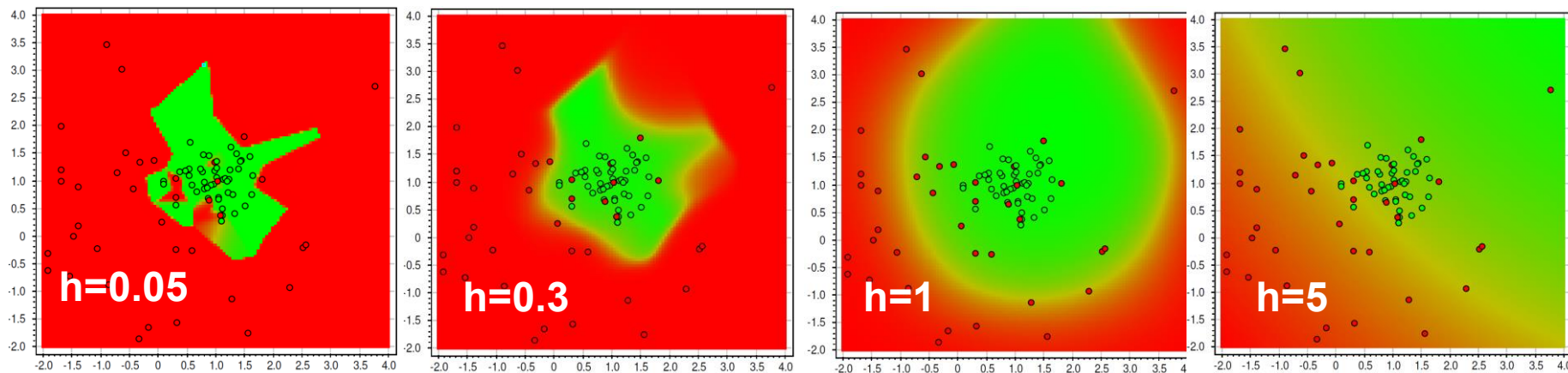
- Если k мал, то чувствительность к шуму, и негладкие границы классов (или линии уровня для регрессий)
- Если k велико, то окрестность может сильно «задеть» соседний класс, зато гладкие границы
- При классификации следует нечетный k , чтобы не было «ничьей»
- Выбирается кросс-валидацией или на валидационном наборе (далее)
- Стандартная эвристика $k = \sqrt{n}$

Метод окна Парзена (для классификации)

- Вес задается радиально-базисным ядром (h – «ширина ядра»):

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

- h – фиксировано $a(x, Z, h) = \arg \max_{y \in Y} \sum_{i=1}^l [y = y^{(i)}] K \left(\frac{d(x, x^{(i)})}{h} \right)$
- k – фиксировано $a(x, Z, k) = \arg \max_{y \in Y} \sum_{i=1}^l [y = y^{(i)}] K \left(\frac{d(x, x^{(i)})}{d(x, x^{(k)})} \right)$



Параметрические модели

- Параметрическое семейство функций

$$A = \{g(x, \theta) \mid \theta \in \Theta\}, g: X \times \Theta \rightarrow Y$$

Θ – множество допустимых параметров

- Линейная модель:

- Классификация $g(x, \theta) = \text{sign}[\sum_{i=1}^q \theta_i f_i(x)], Y = \{-1, 1\}$

- Регрессия $g(x, \theta) = \sum_{i=1}^q \theta_i f_i(x), Y = \mathbb{R}$

- θ -вектор параметров

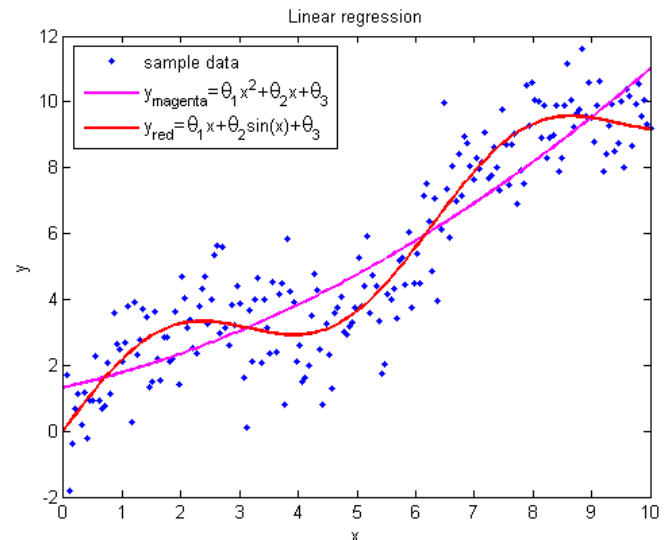
- f_i -не обязательно линейная

- если f_i -зависит от одного признака,

- то модель аддитивная

- Пример:

- Признаки $\{1, x, x^2\}$ vs $\{1, x, \sin(x)\}$



Метод наименьших квадратов для линейной регрессии

- Оценка ошибки - квадратичная функция потерь:

$$Q(\theta, Z) = \frac{1}{l} \sum_{i=1}^l (y_i - a(\bar{x}_i, \theta))^2 = \frac{1}{l} \sum_{i=1}^l \left(y_i - \theta_0 - \sum_{j=1}^p x_{ij} \theta_j \right)^2$$

- В матричной форме:

$$Q(\theta, Z) = (\bar{y} - X\theta)^T (\bar{y} - X\theta)$$

- Единственное оптимальное решение (если матрица данных не сингулярная)

$$\theta = (X^T X)^{-1} X^T \bar{y}$$

- Но не все так просто:

- Если сингулярная матрица данных из-за коррелированных факторов
- Большое число регрессоров – плохая точность и интерпретируемость

Непараметрическая (ядерная) регрессия Надарая-Ватсона

- Суть метода:

- МНК (квадратичная функция потерь)
- Локальный прогноз константой в точке
- Ядерные веса для определения локальной окрестности

- Постановка задачи:

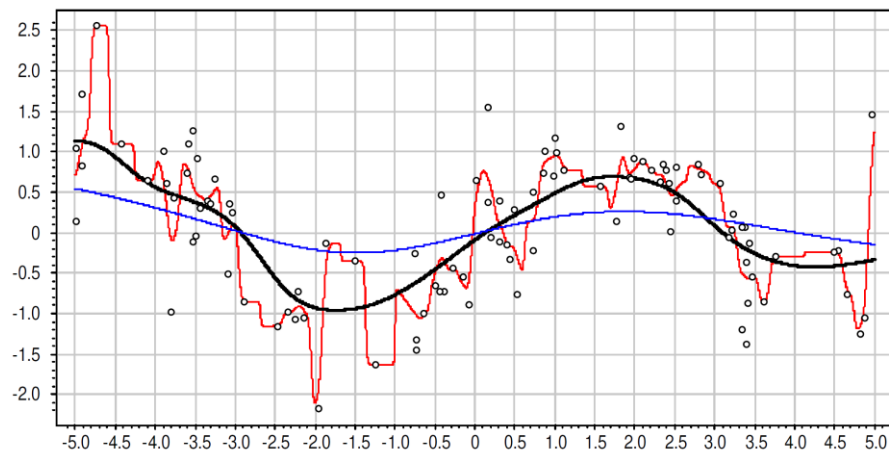
$$Q(Z, \theta) = \frac{1}{l} \sum_{i=1}^l K\left(\frac{d(x, x^{(i)})}{h}\right) (y_i - \theta)^2 \rightarrow \min_{\theta \in \mathbb{R}}$$

- Решающая функция:

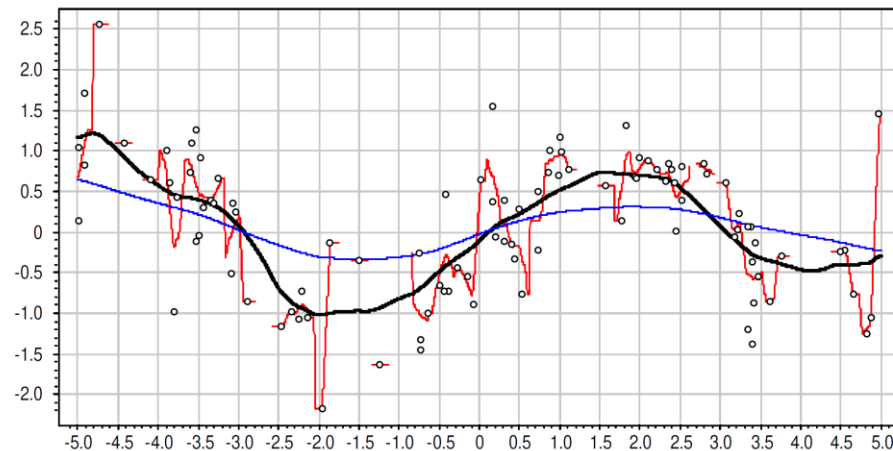
$$a(x, Z, h) = \theta = \frac{\sum_{i=1}^l y_i K\left(\frac{d(x, x^{(i)})}{h}\right)}{\sum_{i=1}^l K\left(\frac{d(x, x^{(i)})}{h}\right)}$$

Непараметрическая (ядерная) регрессия Надарая-Ватсона

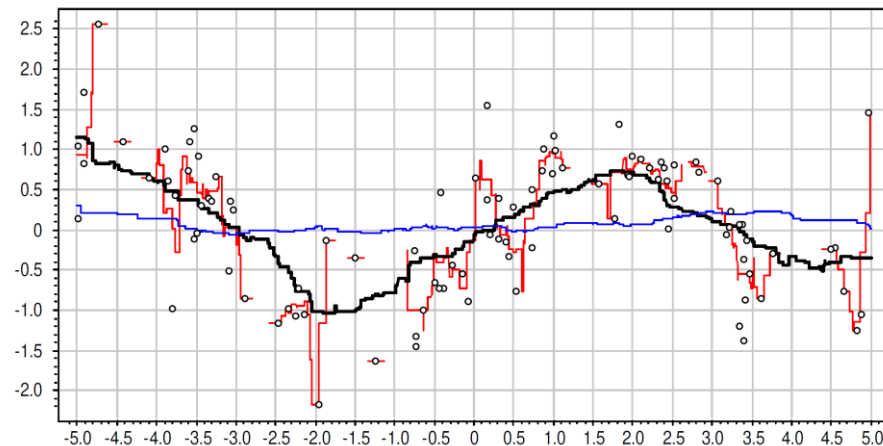
$h \in \{0.1, 1.0, 3.0\}$, гауссовское ядро $K(r) = \exp(-2r^2)$



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



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

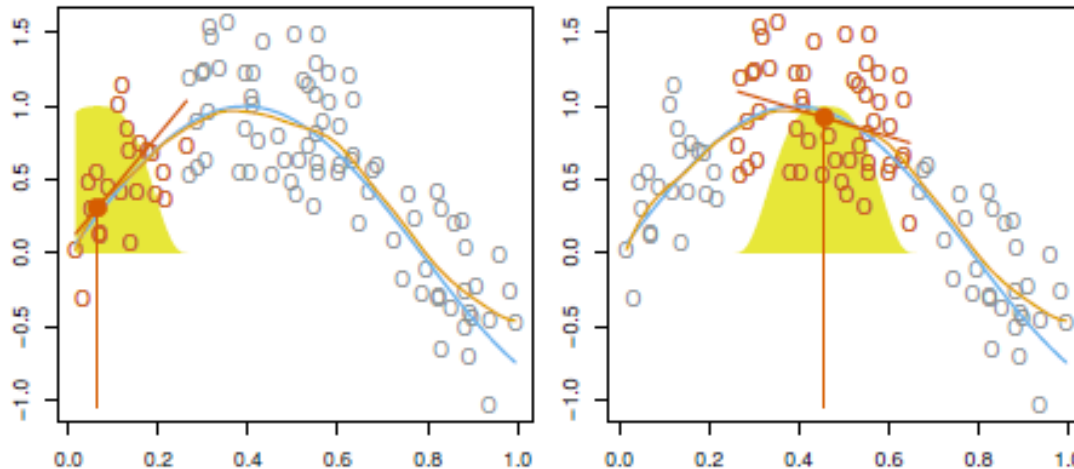


- Чем больше h тем «проще» функция
- Гладкость определяется ядром (непрерывная/кусочно линейная, кусочно постоянная аппроксимация)
- Разрыв, если нет точек в окрестности

Локальная взвешенная регрессия

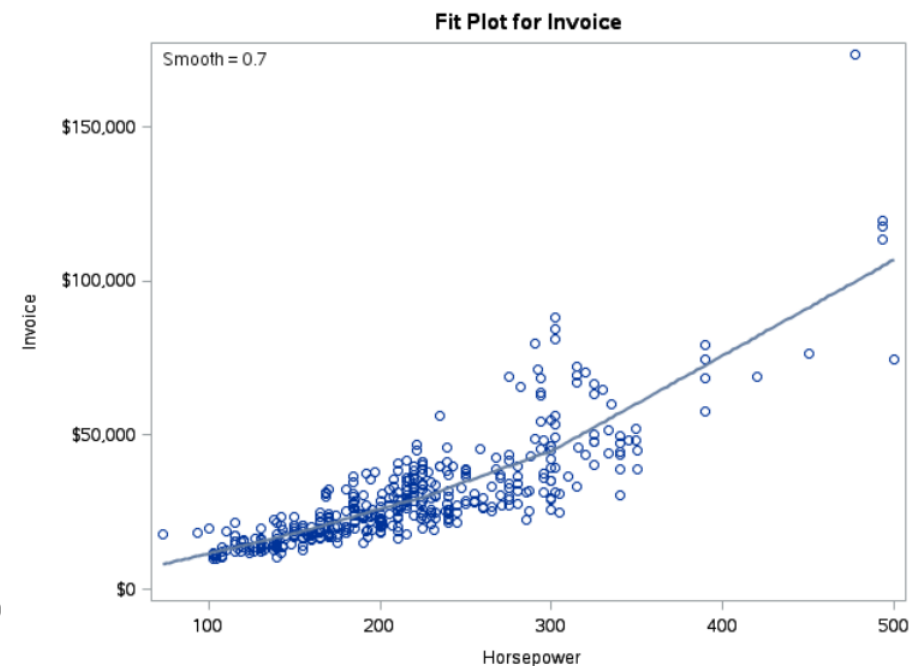
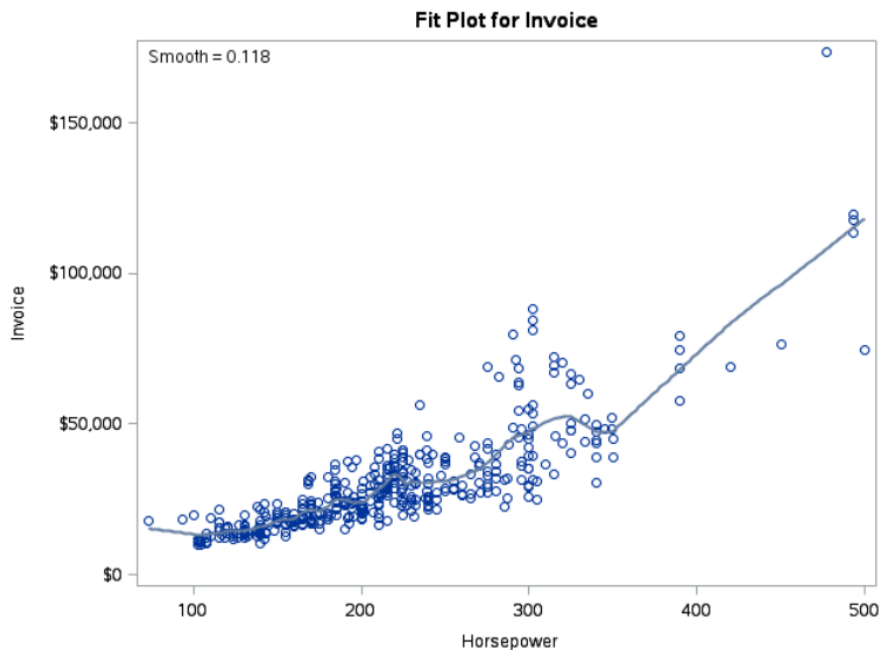
- Вместо константы (как в kernel regression) простая локальная параметрическая модель, например, линейная:

$$Q(Z, \theta) = \frac{1}{l} \sum_{i=1}^l K\left(\frac{d(x, x^{(i)})}{h}\right) (y_i - \mathbf{x}^T \boldsymbol{\theta})^2 \rightarrow \min_{\boldsymbol{\theta} \in \mathbb{R}^p}$$



Локальная взвешенная регрессия

- Нужно задавать параметр сглаживания (фактически – штраф за сложность), который определяет число точек окрестности, чтобы не усложнять модель:



Пример (Python)

Data Set Characteristics:

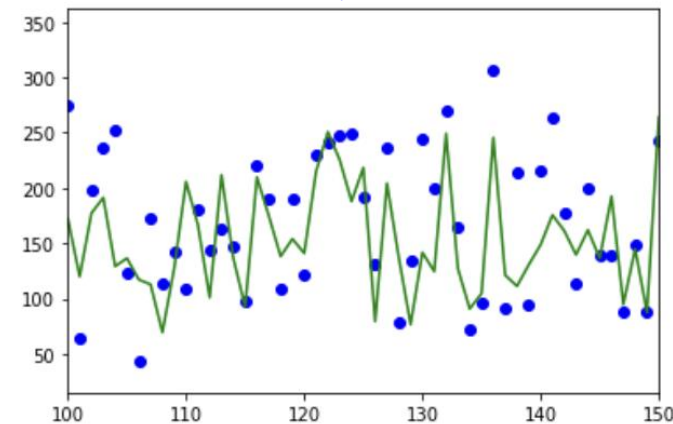
Number of Instances:	442
Number of Attributes:	First 10 columns are numeric predictive values
Target:	Column 11 is a quantitative measure of disease progression one year after baseline
Attribute Information:	<ul style="list-style-type: none">• age age in years• sex• bmi body mass index• bp average blood pressure• s1 tc, total serum cholesterol• s2 ldl, low-density lipoproteins• s3 hdl, high-density lipoproteins• s4 tch, total cholesterol / HDL• s5 ltg, possibly log of serum triglycerides level• s6 glu, blood sugar level

```
plt.scatter(range(len(y_test)), y_test, color="blue")
plt.plot(KNN.predict(X_test), color="green")
plt.xlim([100, 150])
```

```
from sklearn.neighbors import KNeighborsRegressor
from sklearn.datasets import load_diabetes
```

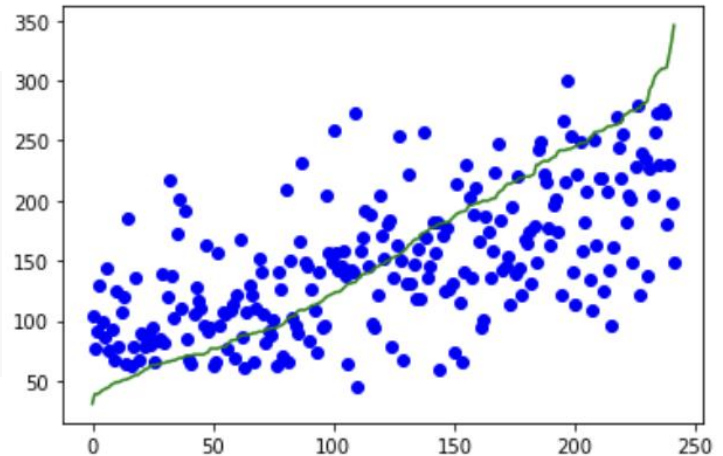
```
N = 200
data = load_diabetes()
X, X_test = data.data[:N], data.data[N:]
y, y_test = data.target[:N], data.target[N:]
```

```
# weights="uniform" is default
# weights="distance" is for KNN
# weights as user function: distances -> weight (implement DAN)
KNN = KNeighborsRegressor(n_neighbors=5, weights="distance")
KNN.fit(X, y)
pass
```

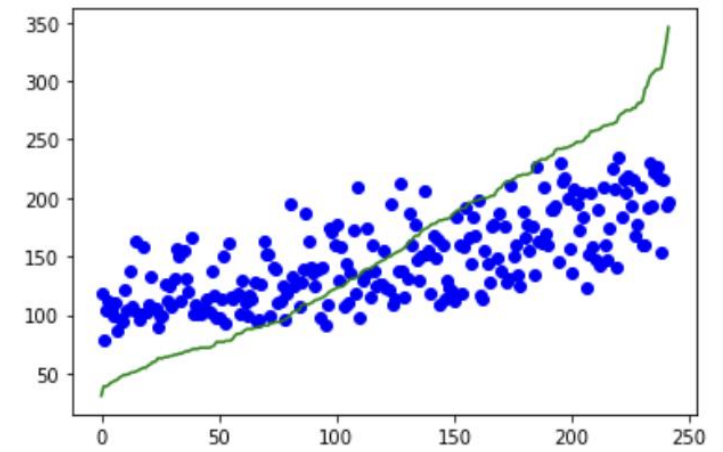


Пример (Python)

```
KNN = KNeighborsRegressor(n_neighbors=3, weights="distance")
KNN.fit(X, y)
y_pred=KNN.predict(X_test)
rs=pd.DataFrame([y_pred, y_test]).T
rs.sort_values(1,inplace=True)
plt.scatter(range(len(rs[0])), [rs[0]], color="blue")
plt.plot(range(len(rs[1])),rs[1], color="green")
```



```
KNN = KNeighborsRegressor(n_neighbors=30, weights="distance")
KNN.fit(X, y)
y_pred=KNN.predict(X_test)
rs=pd.DataFrame([y_pred, y_test]).T
rs.sort_values(1,inplace=True)
plt.scatter(range(len(rs[0])), [rs[0]], color="blue")
plt.plot(range(len(rs[1])),rs[1], color="green")
```



Свойства метрических методов

■ Основные свойства:

- ☐ «Ленивая модель» - не надо ничего обучать
- ☐ Обязательно нужна хорошая метрика и значимые признаки
- ☐ Есть критические **метапараметры**, определяющие сложность модели (гладкость границы или изолиний)

■ Достоинства:

- ☐ Простота реализации
- ☐ Один из самых точных методов (при корректной настройке)
- ☐ Легко адаптируется под сложные типы «откликов», включая ранжирование, многотемность и т.д.
- ☐ Можно интегрировать экспертные знания, задавая веса у примеров, или параметры у метрики

■ Недостатки:

- ☐ «черный ящик» - результат не интерпретируемый совсем
- ☐ Достаточно вычислительно трудоемкие
- ☐ **«Проклятие размерности»**