Лекция 2

Метрические классификаторы

Екатерина Тузова

Разбор летучки

Мотивирующий пример

Мотивирующий пример

Pokemon with stats (https://www.kaggle.com/abcsds/pokemon)

Датасет

```
In [4]: pokemons.head()
```

Out[4]:

	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
0	Bulbasaur	Grass	Poison	318	45	49	49	65	65	45	1	False
1	Ivysaur	Grass	Poison	405	60	62	63	80	80	60	1	False
2	Venusaur	Grass	Poison	525	80	82	83	100	100	80	1	False
3	VenusaurMega Venusaur	Grass	Poison	625	80	100	123	122	120	80	1	False
4	Charmander	Fire	NaN	309	39	52	43	60	50	65	1	False

Какие признаки есть в датасете?

$f : X \to D_f$

- Бинарные ($D_f = \{0, 1\}$)
- Номинальные (D_f – конечное множество)
- Порядковые (D_f – конечное упорядоченное множество)
- Количественные ($D_f = \mathbb{R}$)

Признаки

– Бинарные (Legendary)

Признаки

- Бинарные (Legendary)
- Номинальные (Type 1, Type 2)

Признаки

- Бинарные (Legendary)
- Номинальные (Type 1, Type 2)
- Порядковые (Generation)

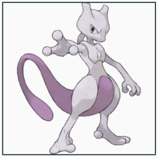
Признаки

- Бинарные (Legendary)
- Номинальные (Type 1, Type 2)
- Порядковые (Generation)
- Количественные (Attack, Defense, ...)

Легендарность

Легендарный покемон это чрезвычайно редкий и зачастую очень могущественных покемон, о нем слагаются мифы и легенды в мире покемонов.

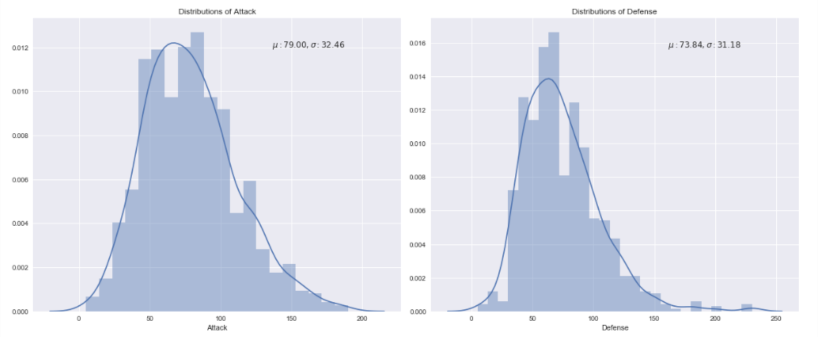
Легендарность



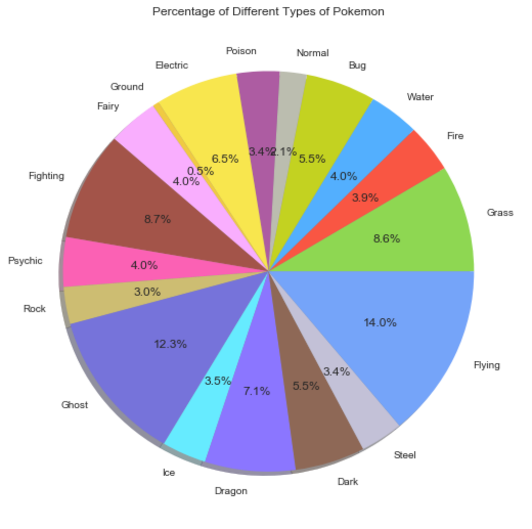


Распределения



Number of Pokemons = 800

Типы покемонов



Задача классификации

X - множество объектов
Y - множество классов
Обучающая выборка: $X^l = (x_i, y_i)_{i=1}^l$

Задача: Построить алгоритм $a \colon X \to Y$, способный классифицировать произвольный объект $x \in X$.

Задача классификации в нашем контексте

X - покемоны

Y - легендарность

Обучающая выборка: $X^l = (x_i, y_i)_{i=1}^l$

$l = 800$ покемонов в выборке

Задача: Построить алгоритм $a\colon X \to Y$, способный определить, является ли покемон легендарным.

Гипотеза компактности

Гипотеза компактности

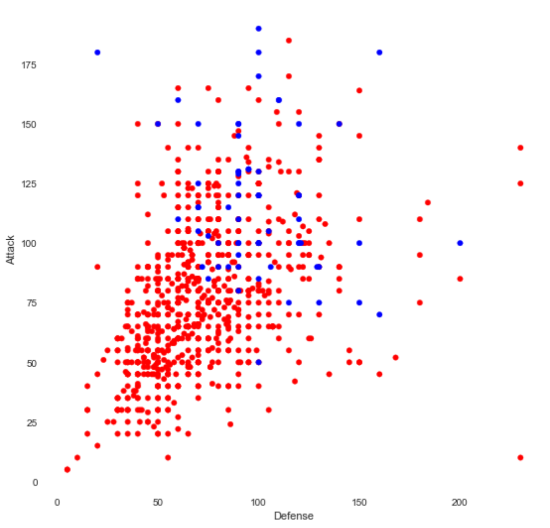
Схожие объекты, как правило, лежат в одном классе.

Гипотеза компактности

Схожие объекты, как правило, лежат в одном классе.

Как определить схожесть объектов?

Пример



Гипотеза компактности

Схожие объекты, как правило, лежат в одном классе.

Схожесть = Функция расстояния

$$\rho : X \times X \to [0, \infty)$$

Функции расстояния

Евклидово расстояние

$$\rho(u, v) = \sqrt{\sum_{j=1}^{n} |u^j - v^j|^2}, \qquad u, v \in X^l$$

Признаковые описания объектов:

$u = \left\{u^1, u^2, ..., u^n\right\}$

$v = \left\{v^1, v^2, ..., v^n\right\}$

Расстояние городских кварталов

$$\rho(u, v) = \sum_{j=1}^{n} |u^j - v^j|, \qquad u, v \in X^l$$

Признаковые описания объектов:
$u = \left\{u^1, u^2, ..., u^n\right\}$
$v = \left\{v^1, v^2, ..., v^n\right\}$

Расстояние Минковского

Обобщение евклидова расстояния и расстояния городских кварталов

$$\rho(u,v) = (\sum_{j=1}^{n} |u^j - v^j|^q)^{1/q}, \qquad u, v \in X^l$$

Признаковые описания объектов:

$u = \{u^1, u^2, ..., u^n\}$

$v = \{v^1, v^2, ..., v^n\}$

Расстояние Левенштейна

Минимальное количество операций вставки одного символа, удаления одного символа и замены одного символа на другой, необходимых для превращения одной строки в другую.

Расстояние Левенштейна

		E	L	E	P	H	A	N	T
	0	1	2	3	4	5	6	7	8
R	1	1	2	3	4	5	6	7	8
E	2	1	2	2	3	4	5	6	7
L	3	2	1	2	3	4	5	6	7
E	4	3	2	1	2	3	4	5	6
V	5	4	3	2	2	3	4	5	6
A	6	5	4	3	3	3	3	4	5
N	7	6	5	4	4	4	4	3	4
T	8	7	6	5	5	5	5	4	3

Метрический классификатор

Обобщенный метрический классификатор

$u \in X$ - произвольный объект, который собираемся классифицировать.

Обобщенный метрический классификатор

$u \in X$ - произвольный объект, который собираемся классифицировать.

Отсортируем объекты из X^l относительно u:
$\rho(u, x_1) \leq \rho(u, x_2) \leq \cdots \leq \rho(u, x_l)$

x_i – i-й сосед объекта u
y_i – класс i-го соседа u

Обобщенный метрический классификатор

$$\rho(u, x_1) \leq \rho(u, x_2) \leq \cdots \leq \rho(u, x_l)$$

x_i – i-й сосед объекта u
y_i – класс i-го соседа u

Обобщенный метрический классификатор

$\rho(u, x_1) \leq \rho(u, x_2) \leq \dots \leq \rho(u, x_l)$

x_i – i-й сосед объекта u
y_i – класс i-го соседа u

Идея 1: Посмотрим на ближайшие объекты и отнесем u к доминирующему классу.

Обобщенный метрический классификатор

$\rho(u, x_1) \leq \rho(u, x_2) \leq \dots \leq \rho(u, x_l)$

x_i – i-й сосед объекта u
y_i – класс i-го соседа u

Идея 1: Посмотрим на ближайшие объекты и отнесем u к доминирующему классу.

Идея 2: Более близкие объекты важнее для классификации.

Метрический алгоритм классификации

$$a(u, X^l) = \arg\max_{y \in Y} \sum_{y_i = y} w(i, u)$$

$w(i, u)$ - вес i-го соседа u, неотрицателен

Метод ближайшего соседа

$$\boxed{a(u, X^l) = \arg\max_{y \in Y} \sum_{y_i = y} w(i, u)}$$

Объект относится к тому классу, к которому относится ближайший в выборке.

$w(i, u) = [i = 1]$

Метод ближайшего соседа

$$\boxed{a(u, X^l) = \arg\max_{y \in Y} \sum_{y_i = y} w(i, u)}$$

$$w(i, u) = [i = 1]$$

\+ Простота

Метод ближайшего соседа

$$\boxed{a(u, X^l) = \arg\max_{y \in Y} \sum_{y_i = y} w(i, u)}$$

$$w(i, u) = [i = 1]$$

+ Простота
+ Интерпретируемость решения

Метод ближайшего соседа

$$\boxed{a(u, X^l) = \arg\max_{y \in Y} \sum_{y_i = y} w(i, u)}$$

$$w(i, u) = [i = 1]$$

+ Простота
+ Интерпретируемость решения

– Неустойчивость к шуму

Метод ближайшего соседа

$$\boxed{a(u, X^l) = \arg\max_{y \in Y} \sum_{y_i = y} w(i, u)}$$

$$w(i, u) = [i = 1]$$

+ Простота
+ Интерпретируемость решения

- Неустойчивость к шуму
- Отсутствие настраиваемых параметров

Метод ближайшего соседа

$$\boxed{a(u, X^l) = \arg\max_{y \in Y} \sum_{y_i = y} w(i, u)}$$

$$w(i, u) = [i = 1]$$

+ Простота
+ Интерпретируемость решения

– Неустойчивость к шуму
– Отсутствие настраиваемых параметров
– Низкое качество классификации

Метод ближайшего соседа

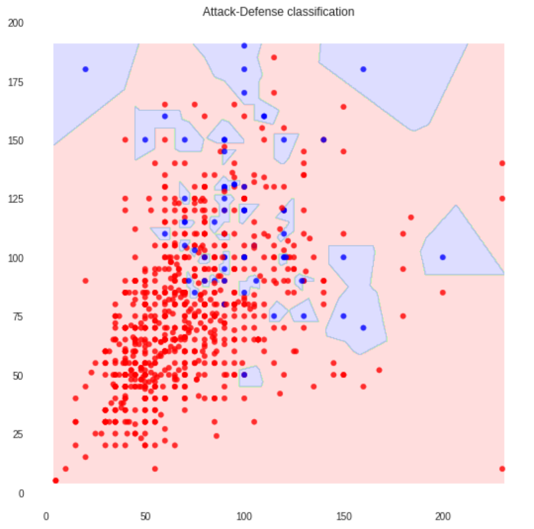
$$\boxed{a(u, X^l) = \arg\max_{y \in Y} \sum_{y_i = y} w(i, u)}$$

$$w(i, u) = [i = 1]$$

+ Простота
+ Интерпретируемость решения

– Неустойчивость к шуму
– Отсутствие настраиваемых параметров
– Низкое качество классификации
– Необходимость хранить всю выборку целиком

Пример


Attack-Defense classification
200
175
150
125
100
75
50
25
0
0
50
100
150
200

Метод k ближайших соседей

$$\boxed{a(u, X^l) = \arg\max_{y \in Y} \sum_{y_i = y} w(i, u)}$$

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

+ Менее чувствителен к шуму

Метод k ближайших соседей

$$\boxed{a(u, X^l) = \arg\max_{y \in Y} \sum_{y_i = y} w(i, u)}$$

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

+ Менее чувствителен к шуму
+ Появляется настраиваемый параметр k

Метод k ближайших соседей

$$\boxed{a(u, X^l) = \arg\max_{y \in Y} \sum_{y_i = y} w(i, u)}$$

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

+ Менее чувствителен к шуму
+ Появляется настраиваемый параметр k

– Неоднозначность при $\sum\limits_{y_i = y} w(i, u) = \sum\limits_{y_i = s} w(i, u) \qquad y \neq s$

Подбор параметров

Как выбрать k

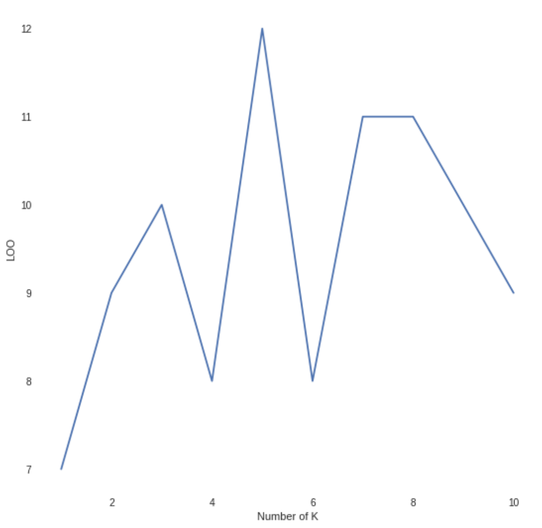
Функционал скользящего контроля (leave-one-out):

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

Вопрос

Правда ли нужно выбрасывать один объект?

Пример



Метод k взвешенных соседей

$$\boxed{a(u, X^l) = \arg\max_{y \in Y} \sum_{y_i = y} w(i, u)}$$

$w(i, u) = [i \leq k] * w_i$, где w_i это вес, зависящий только от номера соседа

Возможные эвристики:

- $w_i = \frac{k+1-i}{k}$ – линейное убывающие веса

Метод k взвешенных соседей

$$\boxed{a(u, X^l) = \arg\max_{y \in Y} \sum_{y_i = y} w(i, u)}$$

$w(i, u) = [i \leq k] * w_i$, где w_i это 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

Вопрос

Как более обоснованно задать веса?

Ядерная оценка плотности

Метод окна Парзена

$K(r)$ – ядро, невозрастающее, положительное на $[0, 1]$

Метод окна Парзена

$K(r)$ – ядро, невозрастающее, положительное на $[0,1]$

Фиксированной ширины:

$a(u, X^l, h, K) = \arg\max\limits_{y \in Y} \sum\limits_{y_i = y} K(\frac{\rho(u, x_i)}{h})$ $\qquad$ h – ширина окна

Ядерная оценка плотности

Метод окна Парзена

$K(r)$ – ядро, невозрастающее, положительное на $[0, 1]$

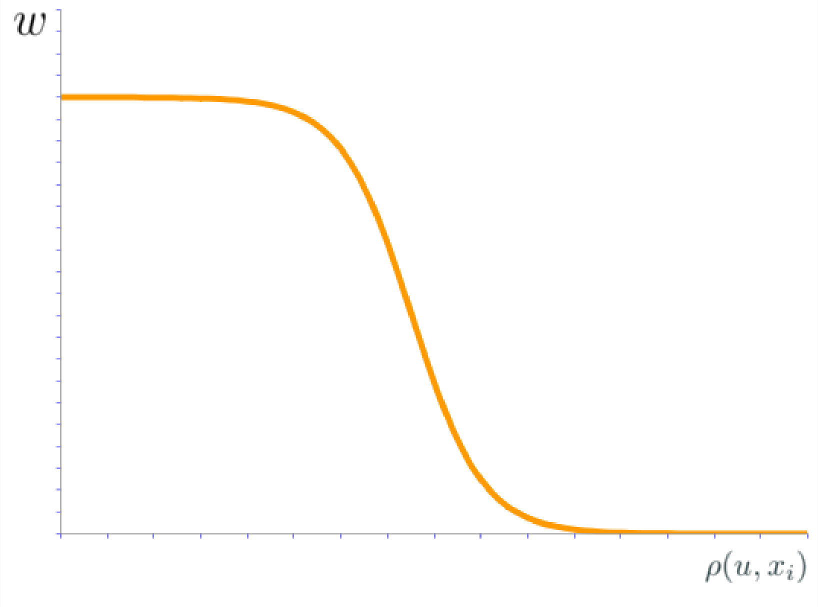
Фиксированной ширины:

$a(u, X^l, h, K) = \arg\max\limits_{y \in Y} \sum\limits_{y_i = y} K(\frac{\rho(u, x_i)}{h})$ $\qquad h$ – ширина окна

Переменной ширины:

$a(u, X^l, k, K) = \arg\max\limits_{y \in Y} \sum\limits_{y_i = y} K(\frac{\rho(u, x_i)}{\rho(u, x_k)})$

Более наглядно



Выбор метрики

Выбор метрики

Идея: Максимизировать сумму расстояний между объектами разных классов при этом сохраняя сумму расстояний между объектами одного класса небольшой.

Выбор метрики

Идея: Максимизировать сумму расстояний между объектами разных классов при этом сохраняя сумму расстояний между объектами одного класса небольшой.

$$\max \sum_{x_i \in D, x_j \in F} \rho(x_i, x_j) \qquad D \neq F$$

Выбор метрики

Идея: Максимизировать сумму расстояний между объектами разных классов при этом сохраняя сумму расстояний между объектами одного класса небольшой.

$$\max \sum_{x_i \in D, x_j \in F} \rho(x_i, x_j) \qquad D \neq F$$

$$\sum_{x_i, x_j \in S} \rho^2(x_i, x_j) \leq 1$$

Проклятие размерности

Если используемая метрика $\rho(u, x_i)$ основана на суммировании различий по всем признакам, а число признаков очень велико, то все точки выборки могут оказаться практически одинаково далеки друг от друга.

Пример

Набор признаков объекта генерируется подбрасыванием честной монетки n раз. Соответственно каждый объект описывается вектором $[0, 1]^n$. При таких условиях все объекты будут равноудалены.

Предобработка

Предобработка данных

Что произойдет, если признаки представлены в разном масштабе?

Предобработка данных

Все признаки должны быть представлены в одном масштабе.
В противном случае признак с наибольшими числовыми значениями будет доминировать в метрике.

Жадное добавление признаков

1. $\rho_j(u, x_i) = |u^j - x_i^j|$ – расстояние по j-му признаку
   $LOO(j) \to \min$

Жадное добавление признаков

1. $\rho_j(u, x_i) = |u^j - x_i^j|$ – расстояние по j-му признаку
   $LOO(j) \to \min$
2. Добавляем признак и строим ρ'
   $\rho'(u, x_i) = \rho(u, x_i) + w_j \rho_j(u, x_i)$
   $LOO(j, w_j) \to \min$

Жадное добавление признаков

1. $\rho_j(u, x_i) = |u^j - x_i^j|$ – расстояние по j-му признаку
   $LOO(j) \to \min$
2. Добавляем признак и строим ρ'
   $\rho'(u, x_i) = \rho(u, x_i) + w_j \rho_j(u, x_i)$
   $LOO(j, w_j) \to \min$
3. Заменяем признак
   $\rho'(u, x_i) = \rho(u, x_i) - w_k \rho_k(u, x_i) + w_j \rho_j(u, x_i)$

Жадное добавление признаков

1. $\rho_j(u, x_i) = |u^j - x_i^j|$ – расстояние по j-му признаку
   $LOO(j) \to \min$
2. Добавляем признак и строим ρ'
   $\rho'(u, x_i) = \rho(u, x_i) + w_j \rho_j(u, x_i)$
   $LOO(j, w_j) \to \min$
3. Заменяем признак
   $\rho'(u, x_i) = \rho(u, x_i) - w_k \rho_k(u, x_i) + w_j \rho_j(u, x_i)$
4. Добавляем признаки, пока LOO не увеличивается

Сверхбольшие выборки

– Проблема хранения

Сверхбольшие выборки

- Проблема хранения
- Проблема быстрого поиска ближайших соседей

Отбор эталонов

Метрический алгоритм классификации

$$a(u, X^l) = \arg\max_{y \in Y} \underbrace{\sum_{y_i = y} w(i, u)}_{\Gamma_y(u)}$$

$w(i, u)$ - вес i-го соседа u, неотрицателен

$\Gamma_y(u)$ - оценка близости объекта u к классу y

Отступ

$\Gamma_y(u) = \sum\limits_{y_i = y} w(i, u)$ – оценка близости объекта u к классу y

Отступ показывает степень типичности объекта.

Отступом объекта $x_i \in X^l$ относительно классификатора a называется величина:

$$M(x_i) = \Gamma_{y_i}(x_i) - \max_{y \in Y \setminus y_i} \Gamma_y(x_i)$$

Типы объектов

1. Эталонные

Типы объектов

1. Эталонные
2. Надёжно классифицируемые (неинформативные)

Типы объектов

1. Эталонные
2. Надёжно классифицируемые (неинформативные)
3. Пограничные

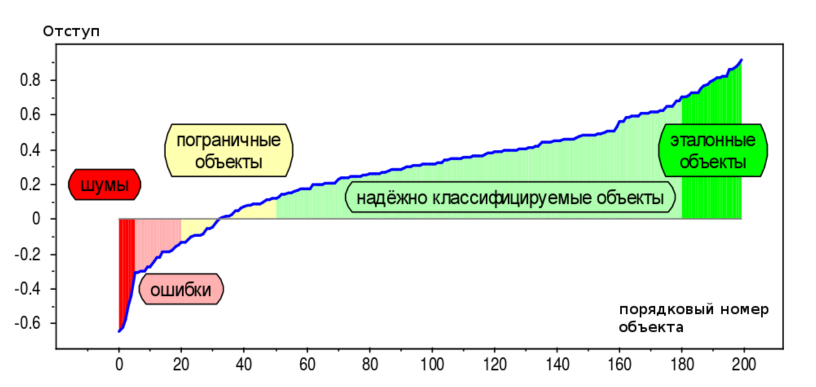
Типы объектов

1. Эталонные
2. Надёжно классифицируемые (неинформативные)
3. Пограничные
4. Ошибочные

Типы объектов

1. Эталонные
2. Надёжно классифицируемые (неинформативные)
3. Пограничные
4. Ошибочные
5. Шумовые

Типы объектов



картинка с machinelearning.ru

Отбор эталонных объектов

Задача:
Выбрать оптимальное подмножество эталонов $\Omega \subseteq X^l$

Классификатор будет иметь вид:

$$a(u, \Omega) = \arg\max_{y \in Y} \sum_{x_i \in \Omega} [y_i = y] w(i, u)$$

Алгоритм STOLP

1. Исключить выбросы и пограничные объекты
2. Найти по одному эталону в каждом классе
3. Добавлять эталоны, пока есть отрицательные отступы

Алгоритм STOLP

\+ Сокращается число хранимых объектов

Алгоритм STOLP

+ Сокращается число хранимых объектов
+ Сокращается время классификации

Алгоритм STOLP

+ Сокращается число хранимых объектов
+ Сокращается время классификации
+ Объекты разделяются по величине отступа

Алгоритм STOLP

+ Сокращается число хранимых объектов
+ Сокращается время классификации
+ Объекты разделяются по величине отступа

– Выбор параметра для определения выбросов

Алгоритм STOLP

+ Сокращается число хранимых объектов
+ Сокращается время классификации
+ Объекты разделяются по величине отступа

– Выбор параметра для определения выбросов
– Не высокая эффективность

Вопросы?

Быстрый поиск ближайших соседей

Быстрый поиск ближайших соседей

- граф ближайших соседей
- k-d дерево
- хеширование (LSH)

k-d дерево

Идея: разложим множество по поторому будем искать в бинарное дерево с простыми условиями и конкретными точками в узлах.

k-d дерево

Идея: разложим множество по поторому будем искать в бинарное дерево с простыми условиями и конкретными точками в узлах.

1. По циклу, или рандомно выбираем ось.
2. Ищем медиану (точку, разбивающую множество на как можно более равные части).
3. Повторяем 1-2 для каждого из получившихся подмножеств

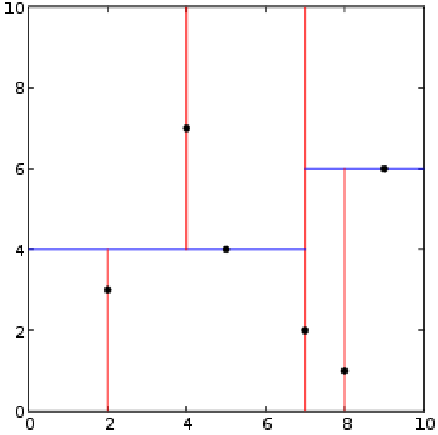
k-d дерево

Идея: разложим множество по поторому будем искать в бинарное дерево с простыми условиями и конкретными точками в узлах.

1. По циклу, или рандомно выбираем ось.
2. Ищем медиану (точку, разбивающую множество на как можно более равные части).
3. Повторяем 1-2 для каждого из получившихся подмножеств

Сложность построения: $O(n \log n)$
Сложность поиска: в лучшем случае $O(\log n)$, в худшем – $O(n)$

2-d дерево



k-d дерево. Особенности

+ Один из наиболее простых методов

k-d дерево. Особенности

+ Один из наиболее простых методов
– Работает только при малом количестве параметров

k-d дерево. Особенности

- \+ Один из наиболее простых методов
- – Работает только при малом количестве параметров
- – Затратный алгоритм перестроения

Locality Sensitive Hash

Задача: Найти похожие документы в интернете

Locality Sensitive Hash

Проблема: Сколько сравнений нам понадобится для того, чтобы найти похожие среди N документов?

Locality Sensitive Hash

Проблема: Сколько сравнений нам понадобится для того, чтобы найти похожие среди N документов?

$C = \frac{N(N-1)}{2}$

$N = 10^6 \Rightarrow C = 5 * 10^{11}$

Locality Sensitive Hash

Идея: Давайте от каждого документа (строки из нулей и единиц) возьмем хэш h:

1. Если документы C_1 и C_2 похожи, то с большой вероятностью $h(C1) == h(C2)$

Locality Sensitive Hash

Идея: Давайте от каждого документа (строки из нулей и единиц) возьмем хэш h:

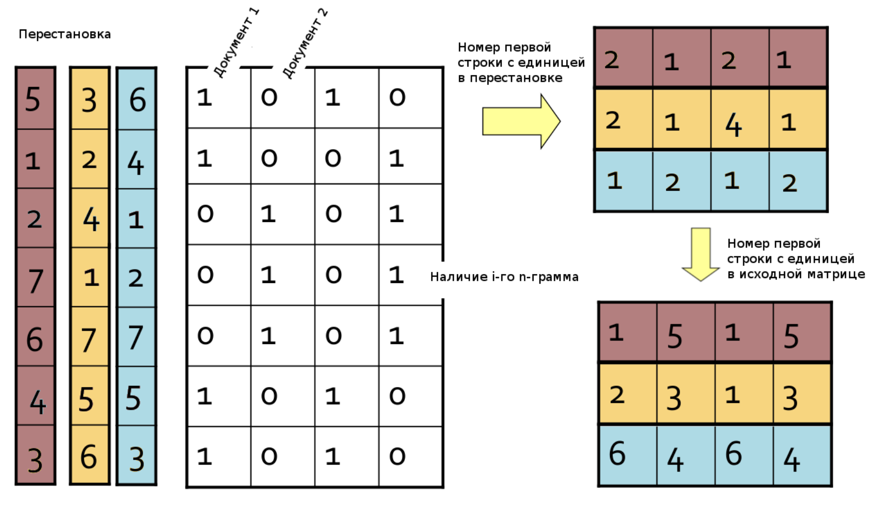
1. Если документы C_1 и C_2 похожи, то с большой вероятностью $h(C1) == h(C2)$
2. Иначе – с большой вероятностью $h(C1) \neq h(C2)$

Locality Sensitive Hash

Идея:

1. Разбить документ на n-граммы
2. Взять от каждого n-грамма хэш
3. Получим представление документа в виде строки из нулей и единиц. Длина такого вектора = количество всевозможных n-грамм.
4. Посчитаем документы похожими, если у них много совпадающих n-грамм

Locality Sensitive Hash



Что происходит сейчас в области knn

ICML'16: Fast k-Nearest Neighbour Search via Dynamic Continuous Indexing

NIPS'16: k^*-Nearest Neighbors: From Global to Local

NIPS'16: Finite-Sample Analysis of Fixed-k Nearest Neighbor Density Functional Estimators

На следующей лекции

- Кластеризация. K-means.
- Цели кластеризации.
- Типы кластерных структур.
- Функционал качества кластеризации
- K-средних
- Иерархическая кластеризация.