Лекция kNN деревья решения

Нахождение проклятие размерности

1. Точные методы

kNN - сложность $O(ld)$

Если $\vec{x} \in R^d$, $m < d$, то $\vec{x} \Rightarrow \vec{z} \in R^m$

$a(\vec{u}, X, \rho) \Rightarrow a(\vec{u}, Z, \rho)$ $O(lm)$

kd - деревья

$d = 10 \sim 20$ $O(\log l)$

$d > 20$ $O(ld)$

2. Приближенные методы

1. Запомнить не всю обучающую выборку $|\{X_i\}| = l$, а лишь эталоны

   STOLP

2. Искать k ближайших соседей приближенно

Опр 2.1

1. Если $\rho(x, y) \in d_1 \Rightarrow P_{f \in F}[f(x) = f(y)] \geq p_1$

2. Если $\rho(x, y) \notin d_2 \Rightarrow P_{f \in F}[f(x) = f(y)] \leq p_2$

где $d_1 \in d_2$ $p_1 \geq p_2$

Пример

$$p = 1 - \frac{S(A \cap B)}{S(A \cup B)}$$

$U = \{u_1, \dots, u_n\}$ – универсум

Π – перестановка

$f_\pi(A) = \min\{\pi(i) \mid u_i \in A\}$

U – это слова

π – степень важности слов, чем меньше $\pi(i)$, тем важнее i-е слово

Предпол. малый уровень влияния

Для нахожд. текстов, слово синхрофазатрон

Покажем, что некоторое все MinHash-функций $F = \{f_\pi \mid \pi \in Sym(U)\}$ является (d_1, d_2, p_1, p_2).

Сначала докажем:

Вероятность того что случайно выбранная функция $f_\pi \in F$ будет принимать одинаковые значения на двух заданных множествах A и B, равна коэф-ту Жаккарда $\frac{|A \cap B|}{|A \cup B|}$

1) $u \in A$, $u \in B$ — p

2) $u \in A$, $u \notin B$ или $u \in B$, $u \notin A$ — q

3) $u \notin A$, $u \notin B$ — 0

$\frac{p}{p+q}$ — вероятность

Пусть $\rho_j(A,B) \leq d_1 \Rightarrow 1-\rho_j(A,B) \geq 1-d_1$

$\Rightarrow p_1 \geq 1-d_2 \Rightarrow$

$(d_1, d_2, 1-d_1, 1-d_2)$

Композиция хэш функций

f — хэш функция

T — хэш таблица

размещения $\vec{x}_i$ в ячейке с номером $f(x)$

Пусть теперь требуется найти к ближайших n $\vec{u}$

Однако p_1 и p_2 мало отличаются поэтому мб k — ложных соседей либо много ложных

$g_1(x) = (f_{11}(x), \dots, f_{1m}(x)) - T_1$

...

$g_L(x) = (f_{L1}(x), \dots, f_{Lm}(x)) - T_L$

Чтобы найти k ближайших соседей для нового объекта u, выберем объекты из всех $g_1(x), \dots, g_L(x)$ таких $T_1 \dots T_L$ и вернем k наиболее близких из них

Два базовых параметра

число m – кол-во функ в одной таки

число L – кол-во комбинаций

$m \nearrow \Rightarrow P(f(\vec{x}) = f(\vec{y})) \searrow$ для $\rho(\vec{x}, \vec{y}) \geq d_2$

то есть $m \gg m_0$, то схожих объектов вообще не будет

$L \nearrow \Rightarrow P(f(\vec{x}) = f(\vec{y})) \nearrow$ для $\rho(\vec{x}, \vec{y}) < d_1$

то есть $L \gg L_0$, то будет слишком много кандидатов

$(d_1, d_2, 1 - (1 - p_1^m)^L, 1 - (1 - p_2^m)^L)$

Если расстояние между этими объектами велико т.е. $\rho(\vec{x}, \vec{y}) \geq d_2$ то вероятность совпадения m базовых функций p_2^m

Теоретические гарантии

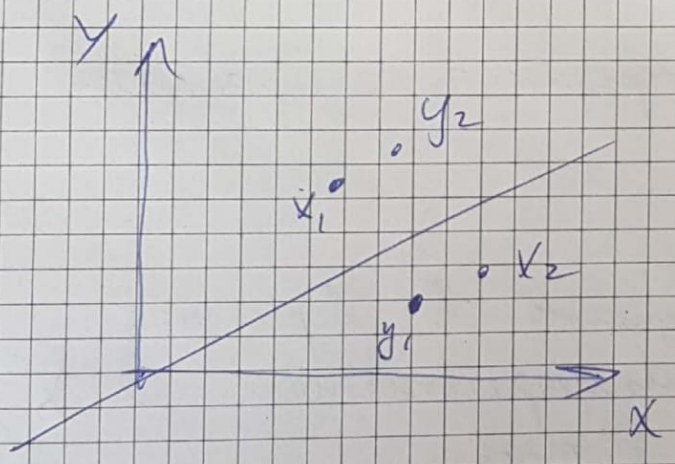
Будем говорить, что алгоритм решает задачу поиска c-ближайшего соседа если для $\vec{u}$ он с $p = 1 - \varepsilon$ возвращает объект выборки удалённый от $\vec{u}$ не более чем в c ру

$O(dl^r \log l)$, где r для многих функций расстояния имеет порядок $\frac{1}{c}$

Хэш функция для косинусного расстояния

$h = \{ f_w(\vec{x}) = \operatorname{sign} \langle \vec{w}, \vec{x} \rangle \mid w \in R^d \}$

Покажите, что Хэш-ф. косинуса является (d_1, d_2, p_1, p_2) – чувствительным



$\angle \vec{x}, \vec{y} = \theta$

$\frac{180 - \theta}{180}$

$(d_1, d_2, 1 - \frac{d_1}{\pi}, 1 - \frac{d_2}{\pi})$

Хэш евклидовой метрики

$$H = \{ f_{w,b}(x) = \left\lfloor \frac{\langle w, x \rangle + b}{r} \right\rfloor \mid w \in R^d, b \in [0, r] \}$$

Деревья

$\forall$ node : $\beta_v : X \to \{0,1\}$

$\forall$ term : $c_v \in Y$

Пример с кредитом

а) одномерные предикаты

г) линейные предикаты

$$\beta_v(x) = [\langle \vec{w}, \vec{x} \rangle < S]$$

з) $\beta_v(x) = [\rho(x, x_v) < S]$ x_v - эталон

многомерные предикаты с пропусками

Алгоритм:

- выбор предиката в вершине
- критерием информативности $Q(X, \beta_v)$
- критерием останова
- метод обр-ки пр-ых значений
- методы стрижки

R_m – индекс объектов обучающей выборки попавших в вершину m

$N_m = |R_m|$

p_{mk} – доля объектов класса k $k \in \{1 \dots K\}$ в вершине m.

$$p_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} [y_i = k]$$

k_m = обозначим класса с которым придусь отнести вершину

$k_m = \arg\max_k p_{mk}$