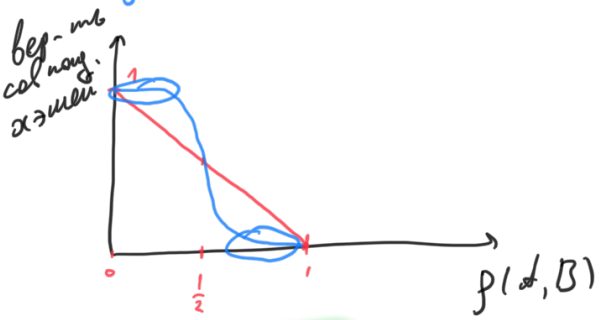
Locality-sensitive hashing (LSH)

① Расстояние Жаккара

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

$p(A, B) = d \Rightarrow \mathbb{P}[f_\pi(A) = f_\pi(B)] = 1 - d$



$x_1, \dots x_\ell$

$f(x_1), \dots f(x_\ell)$

u, $f(u) = 3$

t - порог, т.е.

$p(x,z) \le t \Rightarrow$ с большой вер-тью $f(x) = f(z)$

$p(x,z) > t \Rightarrow$ с большой вер-тью $f(x) \neq f(z)$

② Композиции хэш-функций

1) $f_1(x), \dots f_m(x)$ - хэш-функции из $\mathcal{F}$ (выбираем случайно)

$g(x) = (f_1(x), \dots f_m(x))$

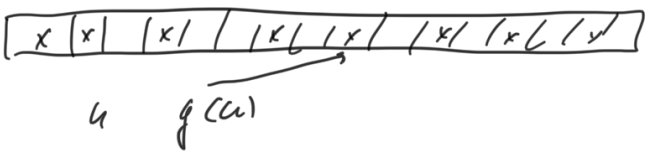
$(f_1(x), \dots f_m(x)) = g($
$(f_1(z), \dots f_m(z)) = g($

$\mathbb{P}_{f \in \mathcal{F}}[f(x) = f(z)] = p$

$\mathbb{P}[g(x) = g(z)] = p^m$

$f \in \{1, \dots k\}^k$

$g \in \{1, \dots k\}^{k^m}$



Может быть слишком много коллизий, можем не найти близких соседей

2) $g_1(x), \dots g_L(x)$ – L случ. векторных хэш-функций

$u \to g_1(u), \dots g_L(u)$

↑ новый объект

$C(u) = \{x \in X \mid g_1(x) = g_1(u)$ или $g_2(x) = g_2(u)$ или ... $g_L(x) = g_L(u)\}$

↑ кандидаты

ищем ближ. соседей u среди $C(u)$

$P[g_1(x) = g_1(z)] = p^m$

$P[g_1(x) = g_1(z)$ или ... или $g_L(x) = g_L(z)] = 1 - (1-p^m)^L$

p^m – вер-ть совпадения по g_1

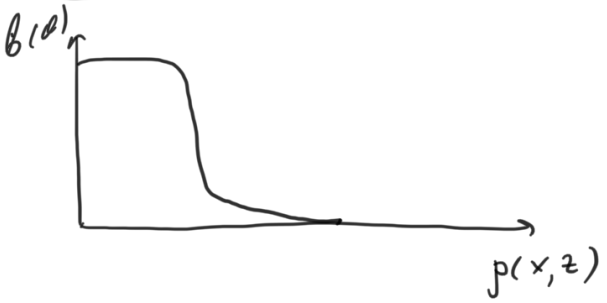
$(1-p^m)$ – вер-ть несовпадения по g_1

$(1-p^m)^L$ – вер-ть, что по всем $g_1, \dots g_L$ нет совпадения

$1-(1-p^m)^L$ – вер-ть совпадения хотя бы по одной $g_1, \dots g_L$

параметры

$(m, L) \to 1-(1-p^m)^L = b(d)$



Некоторая теория про LSH

Алгоритм решает задачу поиска c-ближайшего соседа, если он с вер-тью $1-\varepsilon$ для объекта u возвращает x_u : $\rho(u, x_u) \le c \cdot \rho(u, x_*)$

↑ истинный

Теорема: можно подобрать m и L так, что LSH будет решать задачу поиска c-ближ. соседа за $O(d \cdot l^r \log l)$, где r зависит от метрики и обычно $r \sim \frac{1}{c}$.

ближайший сосед

$O(d \cdot \sqrt{l} \cdot \log l)$

~~$O(d \cdot l)$~~

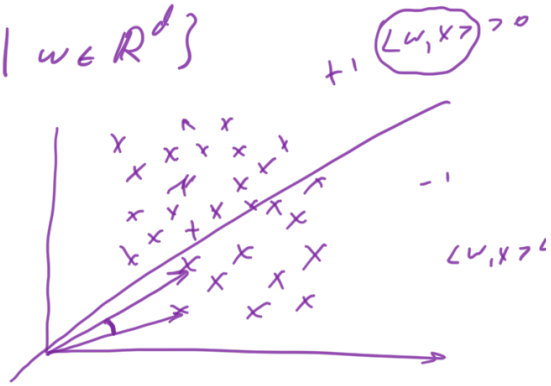
Пример для косинусного расстояния

$$\mathcal{F} = \{ f_w(x) = \underline{\operatorname{sign} \langle w, x \rangle} \mid w \in \mathbb{R}^d \}$$

$f_w(x) \in \{-1, 0, +1\}$

$=$

вер-ть совпадения хэшей обратно пропорц. углу

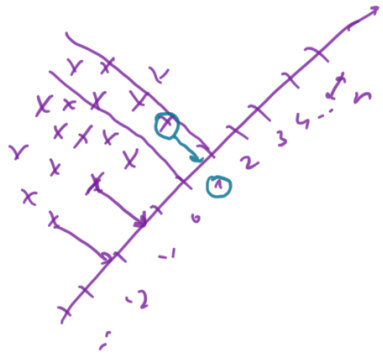


Пример для евклидова расстояния

$$\mathcal{F} = \left\{ f_{w,b}(x) = \left\lfloor \frac{\langle w, x \rangle + b}{r} \right\rfloor \;\middle|\; w \in \mathbb{R}^d, b \in [0, r) \right\}$$

$b \sim U[0, r]$

$w \sim \mathcal{N}(0, I)$



работает для расст. минковского
$< p \in (0, 2]$, если взять правильное
распределение на w

$p = 1 \Rightarrow w \sim Cauchy$

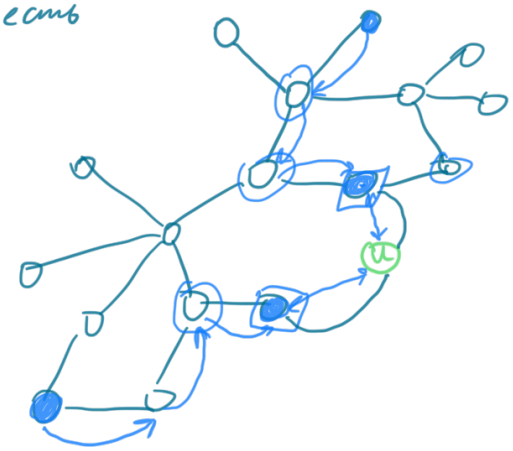
Поиск соседей с помощью NSW
(navigable small world)

$G = (X, E)$

① Поиск соседей

Считаем, что граф уже есть

- берем случ. вершину в графе
- в цикле: переходим в соседнюю вершину, самую близкую к (u) (если есть те, которые ближе чем мы)



v - текущая вершина
$v' : (v, v') \in E$
$p(u, v) > p(u, v')$

делаем мультистарт

$C(u)$ - результаты по всем запускам

ищем в $C(u)$ или в $C(u)$ + соседях этих вершин

② Построение графа

Добавление вершины u:

мультистарт → $C(u)$

$\{C(u) \cup$ соседей $C(u)\} = D(u)$

соединяем u с k ближ. соседями из D(u)

HNSW (Hierarchical NSW)