Схема обоняния фруктовой мухи, как новый вид локально-чувствительного хеширования.

Первая часть этой лекции посвящена хешированию и локально-чувствительному хешированию (LSH). Во второй части будет рассмотрена схема обоняния фруктовой мухи, как новый вид локально-чувствительного хеширования. А в третьей части приведено сравнение стандартного LSH алгоритма со схемой обоняния мухи на основе результатов задачи поиска ближайшего соседа.

1 Хеширование и локально-чувствительное хеширование (LSH).

Метод поиска с общим названием **хеширование** состоит в использовании некоторой частичной информации, полученной из входных данных, в качестве ключа поиска.
Определение 1. *Пусть у нас есть множество объектов X и множество чисел $Y(N) = \{0, 1, 2, ..., N-1\}$. Имеем функцию $f(x) = k$, где x – объект из X, k – число из $Y(N)$. Такая функция называется* **хеш-функцией**. *Она разбивает X на N непересекающихся подмножеств, это разбиение называется* **хешированием**.
Пример 1. *Пусть X – множество целых неотрицательных чисел, $f(x) = x \bmod 4$ (ищем остаток от деления). Тогда хеш-значения для чисел $\{8, 5, 10, 15\}$ будут равны $\{0, 1, 2, 3\}$.*

Хорошая хеш-функция должна удовлетворять следующим свойствам:

- быстрое вычисление;
- минимальное количество коллизий (для любых различных $x_1, x_2 \in X$ $f(x_1) = f(x_2)$).

Для разрешения коллизий разработаны некоторые интересные подходы [1]. Хеширование может применятся, например,

- для быстрого поиска в структуре данных;
- в криптографии.

Как правило, хеш-функции в структуре данных довольно просты и не подходят для применения в криптографии, а функции криптографического хеширования имеют довольно сложное тело.

Термин **локально-чувствительное хеширование** (LSH) был введен в 1998 году в работе [2], чтобы назвать рандомизированную структуру хеширования для эффективного поиска ближайшего соседа в многомерном пространстве. Он основан на определении семейства хеш-функций, отображающих одинаковые входные элементы в один и тот же хеш-код с большей вероятностью, чем разнородные элементы. Однако первое конкретное семейство LSH, **Minhash**, было изобретено Андреем Бродером [3] в 1997 году для обнаружения и кластеризации совпадающих веб-страниц. Это один из самых популярных методов LSH, который широко изучается в теории и широко используется на практике.

Идея локально-чувствительного хеширования противоположна идее традиционного хеширования. При традиционном хешировании два входящих значения, являющиеся похожими, но не одинаковыми, должны давать на выходе существенно различные результаты. При использовании LSH похожие входные значения должны давать похожие выходные значения, имеющие значительно меньшую размерность.

Очевидно, что если вектор большой размерности преобразуется в маленький хеш, часть данных теряется. Здесь на помощь приходит вероятность.
Определение 2. *Пусть M – метрическое пространство размерности d, S – корзина. Семейство хеш-функций $h : M \to S$ называется (R, cR, P_1, P_2)-локально-чувствительным, если для любых $p, q \in M$ выполняется следующее:*

1. *если* $\mathrm{dist}(p, q) \leq R$, *то* $\mathrm{Prob}[h(p) = h(q)] \geq P_1$,
2. *если* $\mathrm{dist}(p, q) \geq cR$, *то* $\mathrm{Prob}[h(p) = h(q)] \leq P_2$,

где $c > 1$, $P_1 > P_2 > 0$.

Существуют различные виды LSH-семейств для различных расстояний таких как, l_p-расстояние, расстояние Хэмминга, мера Жаккара.

Приведем несколько примеров LSH-семейств. Начнем с самого простого.
Пример 2. *Построим семейство LSH-функций для евклидового пространства* $\mathbb{R}^n$.

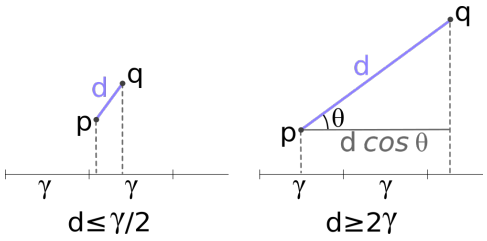
Напомним, расстояние между любыми векторами $p, q \in \mathbb{R}^n$ определяется, как

$$d(p,q) \equiv \|p-q\| = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}.$$

1. Берём случайный единичный вектор $u \in \mathbb{R}^n$, то есть $\|u\| = 1$. Это можно сделать, например, с помощью Гауссова распределения.
2. Проектируем $p, q \in \mathbb{R}^n$ на u:

$$p_u = (p,u) = \sum_{i=1}^{n} p_i u_i, \quad q_u = (q,u) = \sum_{i=1}^{n} q_i u_i.$$

3. Разбиваем u на отрезки длинной γ, это будут "корзины" . Индекс корзины, куда попала проекция произвольного вектора $x \in \mathbb{R}^n$ на u, определяет хеш-функцию $h(x) = (x,u) \mod \gamma$.



В случае одномерного пространства, если $d(p,q) \le \gamma/2$, то

$$\text{Prob}[h(p) = h(q)] \ge \frac{\gamma - d}{\gamma} \ge 1/2.$$

Пусть $d(p,q) \ge 2\gamma$. Тогда точки p и q попадут в одну корзину γ, если $d\cos\theta \le \gamma$. Откуда следует $\cos\theta \le \frac{1}{2}$ и $60° \le \theta \le 90°$. То есть вероятность, что точки p и q попадут в одну корзину не больше $1/3$. Получаем, что хеш-функции h будут $(\gamma/2, 2\gamma, 1/2, 1/3)$-чувствительны.
Пример 3. *Локально-чувствительное семейство для расстояния Хэмминга.*

Напомним, расстоянием Хэмминга $d(x,y)$ между двумя двоичными векторами x и y длины n называется число позиций, в которых они различны.

Сконструируем локально-чувствительное семейство функций для n-мерных двоичных векторов. Пусть $h_i(x)$ обозначает i-ю координату вектора x. Вероятность того, что x и y не совпадают, равна соотношению количества несовпадений координат векторов x и y на общее число координат, т.е.

$$\text{Prob}[h(x) \ne h(y)] = \frac{d(x,y)}{n}.$$

Тогда вероятность того, что x и y совпадают

$$\text{Prob}[h(x) = h(y)] = 1 - \frac{d(x,y)}{n}.$$

Если $d(x,y) \le R$, то $\text{Prob}[h(x) = h(y)] \ge 1 - \frac{R}{n}$. Если $d(x,y) \ge cR$, то $\text{Prob}[h(x) = h(y)] \le 1 - \frac{cR}{n}$. Мы получаем, что семейство $H = \{h_i : h_i(x) = x_i,\ i = 1, \ldots, n\}$ является $\left(R, cR, 1 - \frac{R}{n}, 1 - \frac{cR}{n}\right)$ локально-чувствительным.
Пример 4. *Локально-чувствительное семейство для меры Жаккара.*

Напомним определение меры Жаккара. Пусть S_1 и S_2 два подмножества из множества U. Мера Жаккара определяется, как отношение размера пересечения множеств S_1 и S_2 к размеру их объединения, то есть

$$J(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}.$$

А расстояние Жаккара:

$$D(S_1, S_2) = 1 - J(S_1, S_2).$$

Для меры Жаккара локально-чувсвительное семейство **minhash** впервые было построено в [3]. Пусть π – случайная перестановка (перемешивание) множества U. Для подмножества $S \subseteq U$ рассмотрим локально-чувствительное семейство

$$\{h(S) - \text{первый элемент } S \text{ после случайной перестановки } \pi \text{ множества } U\}. \quad (1)$$

Нетрудно проверить, что

$$\text{Prob}[h(S_1) = h(S_2)] = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|} = J(S_1, S_2).$$

Например, если $U = \{1, 2, 3\}$, а $S_1 = \{1, 2\}$, $S_2 = \{2, 3\}$, то для всевозможных перестановок будем иметь такие хеш-функции:

$$\pi_1(U) = \{1, 2, 3\},\ h(S_1) = 1,\ h(S_2) = 2,$$
$$\pi_2(U) = \{1, 3, 2\},\ h(S_1) = 1,\ h(S_2) = 3,$$
$$\pi_3(U) = \{2, 1, 3\},\ h(S_1) = 2,\ h(S_2) = 2,$$
$$\pi_4(U) = \{2, 3, 1\},\ h(S_1) = 2,\ h(S_2) = 2,$$
$$\pi_5(U) = \{3, 1, 2\},\ h(S_1) = 1,\ h(S_2) = 3,$$
$$\pi_6(U) = \{3, 2, 1\},\ h(S_1) = 2,\ h(S_2) = 3.$$

Откуда следует $\text{Prob}[h(S_1) = h(S_2)] = \frac{2}{6} = \frac{1}{3} = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}$.

Пусть $r > 0$ и $c > 1$. Если $D(S_1, S_2) \leq r$, то

$$\text{Prob}[h(S_1) = h(S_2)] = 1 - D(S_1, S_2) \geq 1 - r.$$

Если $D(S_1, S_2) \geq cr$, то

$$\text{Prob}[h(S_1) = h(S_2)] = 1 - D(S_1, S_2) \leq 1 - cr.$$

Мы получили, что семейство хеш-функций (1) $(r, cr, 1 - r, 1 - cr)$ – локально-чувствительно.

Локально-чувствительное хеширование применяется для:

- поиска похожих объектов (документов, веб-страниц, картинок и т.д.);
- иерархической кластеризации объектов;
- задачи поиска ближайшего соседа.

Поиск похожих объектов является фундаментальной вычислительной задачей для больших информационно-поисковых систем. В работе [5] обнаружено, что схема обоняния фруктовой мухи решает эту задачу.

2 Система обоняния фруктовой мухи

Схема обоняния мухи присваивает похожие нейронные активные шаблоны похожим запахам так, что поведение, запомнившееся от одного аромата, может быть применено к похожему встретившемуся аромату. Опишем эту схему.

1. На первом шаге муха получает информацию об аромате. Обонятельные рецепторные нейроны (ORN), расположенные в периферических обонятельных органах мухи, соединяются с помощью аксонов (длинный отросток нервной клетки) с некоторыми переплетенными структурами нервных волокон, называемыми **клубочками** (glomeruli). Существует порядка 50-ти типов ORN с разной чувствительностью и избирательностью для разных ароматов. ORN, отвечающие одному и тому же типу сходятся к определенному клубочку, которых также порядка 50-ти. В этих клубочках ORN создают возбуждающие синапсы (особый тип контакта между нейронами или клетками) с проекционными нейронами (PN) (рисунок 1).

Также клубочки связаны между собой сетью локальных нейронов (LN), которые взаимодействуют с ORN и PN.

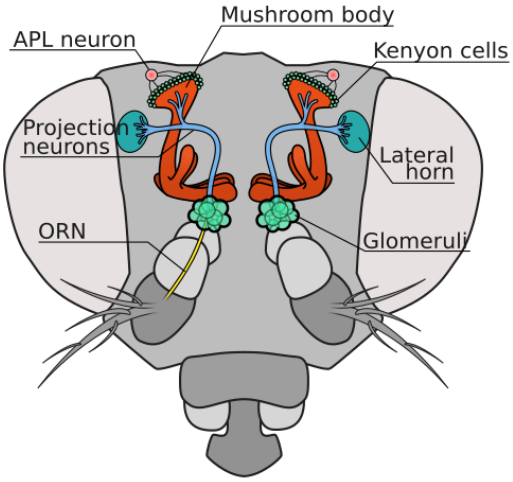


Рис. 1: Схема обоняния фруктовой мухи.

2. На втором шаге проекционные нейроны соединяются с двумя основными областями мозга: боковой рог (lateral horn) и грибовидное тело (mushroom body). Считается, что боковой рог отвечает за врожденную поведенческую реакцию на аромат, а грибовидное тело за обонятельную память. Между этими двумя областями также происходит взаимодействие. Например, изученный поведенческий сигнал от грибовидных тел может отменить врожденный сигнал поведения и наоборот. Ниже в математическом описании схемы обоняния участвует только грибовидное тело.

Грибовидное тело – это большая структура, состоящая из нескольких слоев нейронов. Основной тип нейронов грибовидных тел – клетки Кеньона. Число клеток Кеньона у разных насекомых варьируется от 3000 у дрозофилы до 200 000 у тараканов. Каждая клетка Кеньона имеет около 6 отростков (claws), т.е. собирает информацию в среднем из 6 клубочков.

3. На последнем шаге APL нейрон (anterior paired lateral) (по одному на каждой стороне мозга) собирает выходящую информацию об аромате из аксонов клеток Кеньона и с помощью своих аксонов подавляет почти все клетки Кеньона за исключением часто активируемых 5% . Нарушение обратной связи между APL нейроном и клетками Кеньона уменьшает разреженность оставшихся клеток Кеньона после подавления, увеличивает корреляции между запахами и препятствует обучению мухи различать похожие запахи [6]. Частота активации этих оставшихся 5% соответствует определенному аромату (схема победитель получает все).

Перейдем к математическому описанию [5].

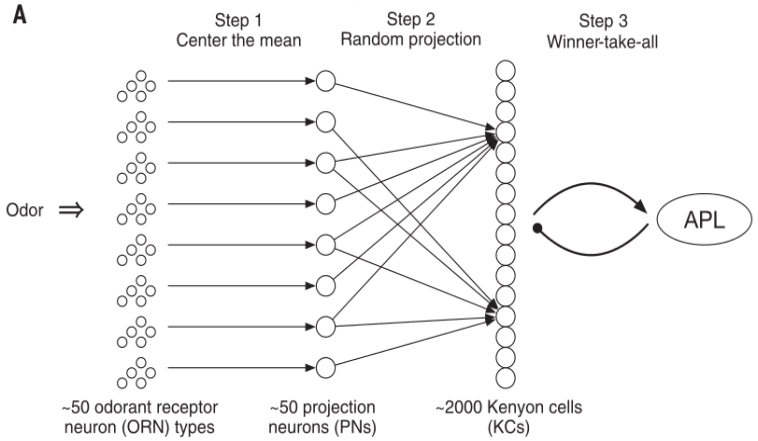


Рис. 2: Нейросеть мухи

1. Первый шаг заключается в передаче информации о запахе от рецепторных нейронов аромата (ORN) на проекционные нейроны (PN) через клубочки. Как уже было сказано, существует 50 типов ORN и каждый аромат расположен в 50-мерном пространстве, определяемом частотой активации ORN. Для каждого аромата распределение частот активации ORN в 50-мерном пространстве экспоненциально со средним значением, зависящим от концентрации аромата. Для PN зависимость от концентрации отсутствует. Поэтому распределение частот активации PN также экспоненциально, а среднее значение близко к одному и тому же значению для всех ароматов и всех концентраций ароматов. Поэтому первый шаг – это центрирование среднего значения, которое выполняется согласно модели "разделяющей нормализации" [7]. Эта модель состоит в том, что ответ нейрона на сложный стимул является суммой его ответов на каждую характеристику стимула в отдельности, деленной на величину, называемую энергией стимула, которая увеличивается с интенсивностью и сложностью стимула. То есть реакция нейрона на сложный стимул близка к среднему его ответов на каждую характеристику стимула [8].

2. Второй шаг – это 40-кратное увеличение числа нейронов, т.е. 50 проекционных нейронов (PN) $x_1, \ldots, x_d$ проектируются на 2000 клеток Кеньона (KC) $y_1, \ldots, y_m$, связанных разряженной бинарной случайной матрицей соединений M. Элементы этой матрицы определяются следующим образом:

$$M_{ij} = \begin{cases} 1, & \text{если } x_i \text{ связано с } y_j \\ 0, & \text{иначе.} \end{cases}$$

   В матричном виде будем иметь

$$y = Mx,$$

   где M – матрица размера $m \times d$. На практике часто делается шаг дискретизации $y = \lfloor \frac{Mx}{\omega} \rfloor$, где ω – константа. В работе [5] показано, что таким образом построенные разряженные бинарные случайные проекции сохраняют локальную метрическую структуру входящих ароматов в пространстве l_2, если число проекций m достаточно большое. Требуемое m зависит от того, насколько разряжен вектор x. Если x сильно разряжен, то $m = O(d)$ будет достаточно. Если x равномерно распределен, то $m = O(1)$ достаточно.

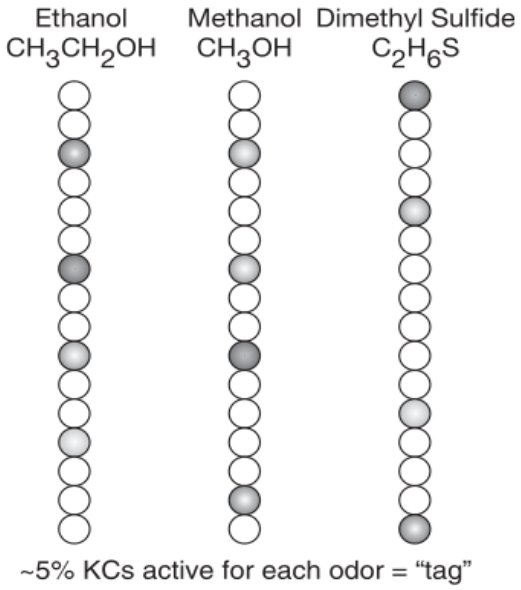


Рис. 3: Похожим ароматам (этанол и метанол) присвоены более похожие метки.

3. Третий шаг включает схему "победитель получает все" , в которой подавляющая обратная связь осуществляется одним APL нейроном. Как результат – все клетки Кеньона, за исключением 5 % часто активируемых, подавляются, то есть наиболее активные k клеток Кеньона сохраняют свои значения, а оставшиеся зануляются. В результате мы имеем разряженный вектор $z \in \mathbb{R}^m$, чьи элементы определяются следующим образом:

$$z_i = \begin{cases} y_i, & \text{если } y_i \text{ один из наибольших } k \text{ элементов } y; \\ 0, & \text{иначе} \end{cases}.$$

Частота активации этих оставшихся k клеток соответствует метке, присвоенной входящему аромату.

С точки зрение вычислительной перспективы, схема обоняния мухи – это хеш-функция, чьи входные данные – это аромат, а выходные – метки. Метки, полученные по данной схеме, различают не только ароматы, но также ассоциируют очень похожие ароматы с похожими метками.

Это говорит о том, что схема обоняния мухи – локально-чувствительная и может применятся, как LSH алгоритм для эффективного поиска ближайших соседей в большой базе данных.

3 Результаты.

Сравним результаты схемы обоняния мухи и традиционного LSH алгоритма, исходя из того, как точно каждый алгоритм может найти ближайшего соседа для входящих данных. Алгоритмы будем сравнивать на трех тестовых базах: SIFT($d = 128$), MNIST($d = 784$), GLOVE($d = 300$). Первые две базы содержат векторное представление картинок, в то время как третья база – векторное представление слов, используемых для поиска семантической похожести. Для каждой базы выбирается подмножество размером 10000 входных данных, в котором каждый вход представлен вектором признаков в d-мерном пространстве. Во всех базах каждый входящий вектор нормализован, чтобы иметь одно и тоже среднее значение. Чтобы сравнение двух алгоритмов было справедливым, устанавливаем одинаковую вычислительную сложность. Для вычисления эффективности выбирается 1000 случайных входных данных из 10000 и сравниваются настоящие "ближайшие соседи"с предсказанными с помощью средней точности.

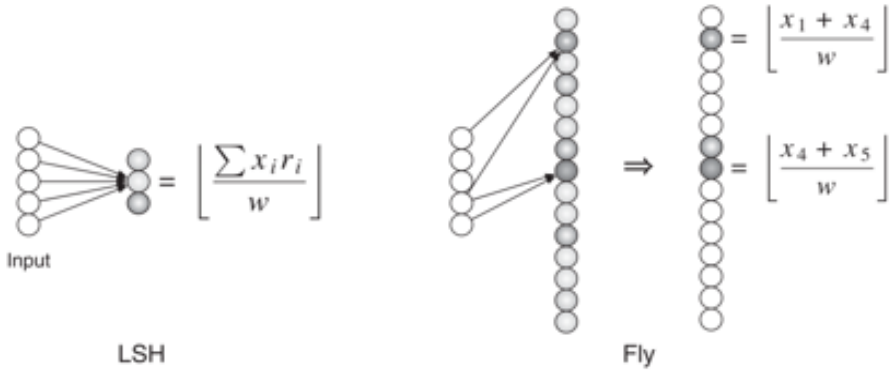


Рис. 4: Сравнение LSH-алгоритма и алгоритма обоняния мухи. Вычислительная сложность у обоих одинаковая. Входящая размерность $d = 5$. LSH вычисляет 3 случайных проекций, каждая из которых требует 10 операций. Схема обоняни мухи требует 15 случайных проекций, каждая из которых требует 2 операции сложения. $x = (x_1, \ldots, x_d)$ – вектор признаков входных данных, r – случайная гауссова переменная, w – размер корзины для дискретизации.

Остановимся на главных различиях в схеме обоняния фруктовой мухи и LSH алгоритма.

1. Схема обоняния мухи использует разреженные бинарные случайные проекции, в то время, как LSH функции используют плотные Гауссовы случайные проекции. Однако, это не вредит тому, как точно ближайшие соседи найдены.

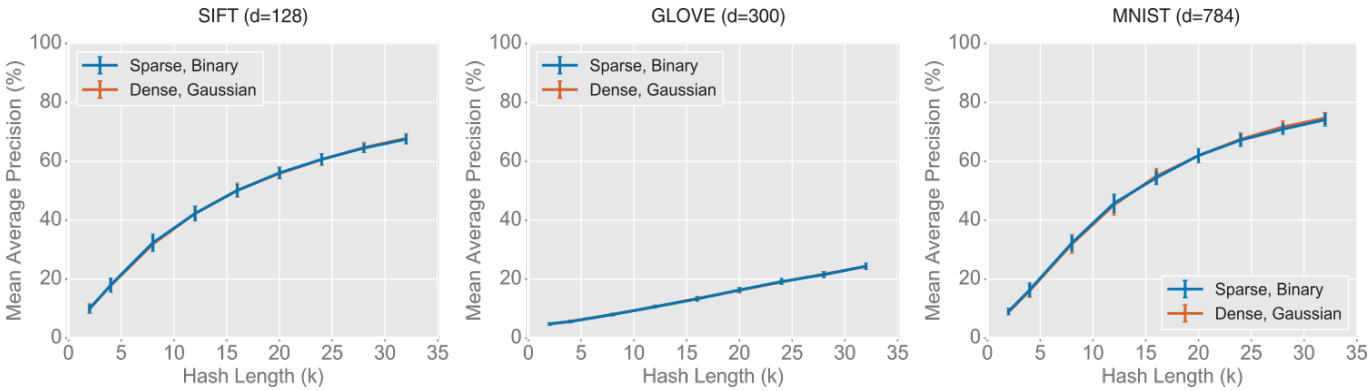


Рис. 5: Величина Mean Average Precision обозначает насколько точно найдены настоящие ближайшие соседи и зависит от длины хеша.

2. Схема обоняния фруктовой мухи не только увеличивает размерность входящих данных в отличии от LSH алгоритма, но и разреживает это представление с помощью схемы "победитель получает все". Эта схема выбирает k часто активируемых клеток Кеньона, как метки.

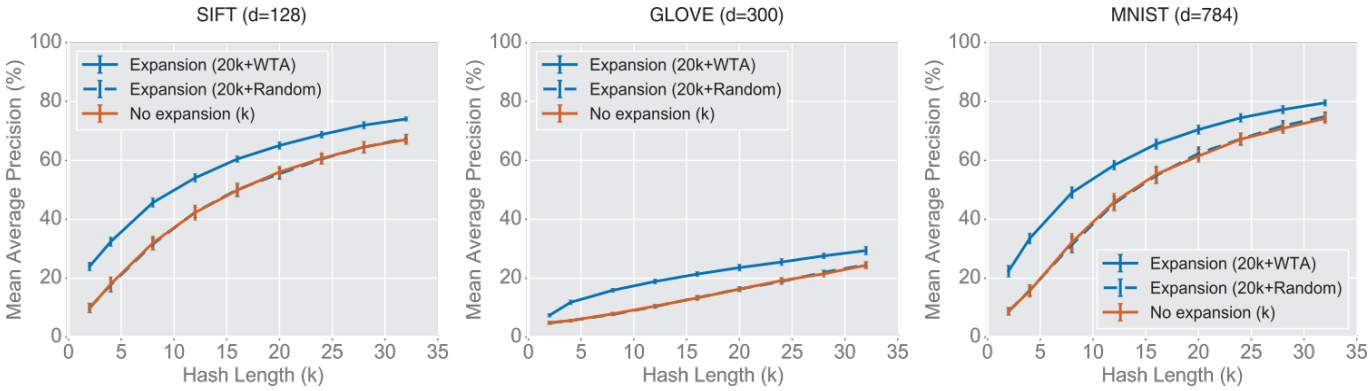
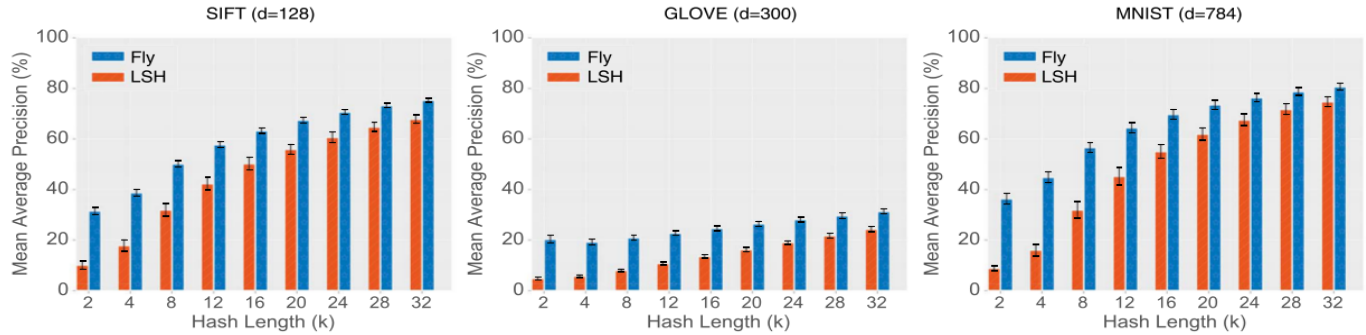


Рис. 6: Сравнение результатов для различных типов проекций: при увеличении размерности с k до $20k$ и использование схемы WTA или случайно выбранных k клеток Кеньона, и без расширения размерности.

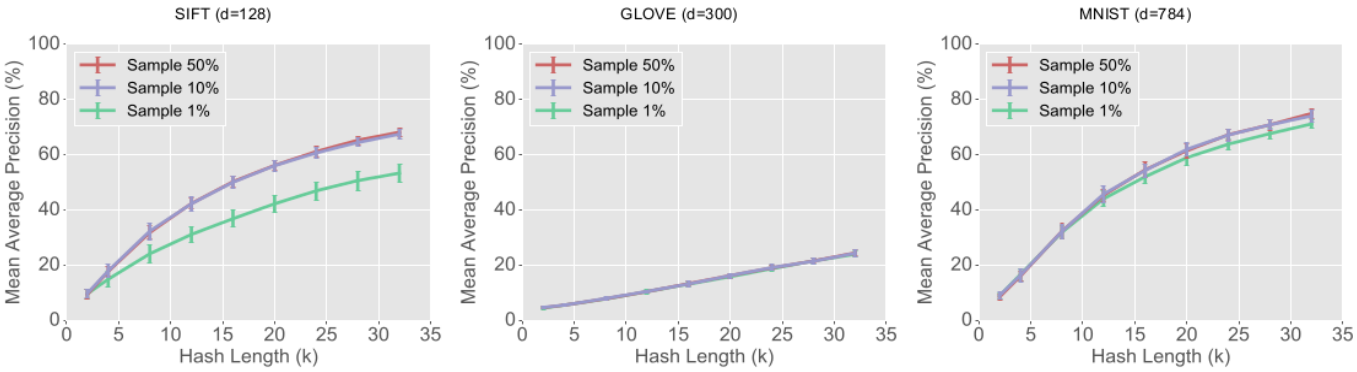
Дальнейшее увеличение размерности дает усиление результата схемы обоняния фруктовой мухи перед LSH для всех баз.



Рассмотрим некоторые дискуссионные вопросы.

3.1 Изменение количества проекционных нейронов.

Помимо упомянутых различий между LSH и схемой обоняния фруктовой мухи, также можно менять число проекционных нейронов (PN) в схеме мухи от 1% до 50%, которые соединяются с каждой клеткой Кеньона. Как видим, для базы GLOVE 1% проекционных нейронов достаточно.



3.2 Почему нельзя в качестве меток выбрать проекционные нейроны вместо клеток Кеньона?

Биологически проекционные нейроны активизируются в ответ на большинство ароматов. Если проекционный нейрон использовался бы для обучения, то каждый аромат модифицировал бы синаптическую силу проекционного нейрона, ассоциированную с большинством других ароматов и специфический аромат не возможно было бы выделить. Выбор клеток Кеньона в качестве меток дает непересекающие метки так, что синапсы, ассоциированные с одним ароматов могут быть модифицированные без модификации синапсов, ассоциированных с другим ароматом.

3.3 Количество APL нейронов в схеме победитель получает все.

Как уже выше было сказано, в схеме обоняния фруктовой мухи существует только один подавляющий APL нейрон, который подавляет редко активируемые нейроны. Однако в обонятельной схеме мыши существует аналоги подавляющих нейронов и их пять [9]. Вопрос, каким образом происходит обучение, остается неясным. В настоящее время ведутся активные исследования в создании биологически правдоподобных WTA сетей.

3.4 Хеширование, не зависящее от данных.

Заметим, что в схеме обоняния мухи хеширование не зависит от входящих данных, то есть хеш-функции не используют предварительные данные для получения меток. В последнее время было предложено много LSH-семейств, зависящих от данных, например, семантическое хеширование, спектральное хеширование, глубокое хеширование и другие. Вычислительное понимание, как изучать и модифицировать хеш-функцию на основе получаемых данных в течении времени, является важным вопросом.

Литература

[1] Д.Э. Кнут Искусство программирования. Том 3, Издание 2-е, 2001.

[2] P. Indyk and R. Motwani Approximate nearest neighbors: Towards removing the curse of dimensionality. In STOC, 604–613, 1998.

[3] A. Z. Broder On the resemblance and containment of documents. In Proceedings of the Compression and Complexity of Sequences, 21–29, Positano, Amalfitan Coast Salerno, Italy, IEEE Computer Society, 1997.

[4] C.F. Stevens What the fly's nose tells the fly's brain. Proc. Natl. Acad. July, 112 (30), 2015.

[5] S. Dasgupta, Ch. F. Stevens, S. Navlakha A neural algorithm for a fundamental computing problem. Science, Vol. 358, Issue 636, 2017.

[6] A.C. Lin, A. Bygrave, A. de Calignon, T. Lee, G. Miesenbock Sparse, decorrelated odor coding in the mushroom body enhances learned odor discrimination. Nat Neurosci 17(4), 2014.

[7] D.J. Heeger Normalization of cell responses in cat striate cortex. Vis Neurosci, 9,181–197, 1992.

[8] S. R. Olsen, V. Bhandawat and R. I. Wilson Divisive normalization in olfactory population codes. Neuron, 66(2), 287–299, 2010.

[9] J. M. Bekkers, N. Suzukie Neurons and circuits for odor processing in the piriform cortex. Trends Neurosci, v. 36, 429–438, 2013.