Добавление дополнительной функциональности в библиотеки поиска ближайших соседей

Келим Илья, 21.М07-мм

Научный руководитель: к.т.н., доцент Брыксин Т.А.

Консультант: Data scientist JetBrains Литвинов Д.В.

Контекст

Проблема поиска ближайших соседей

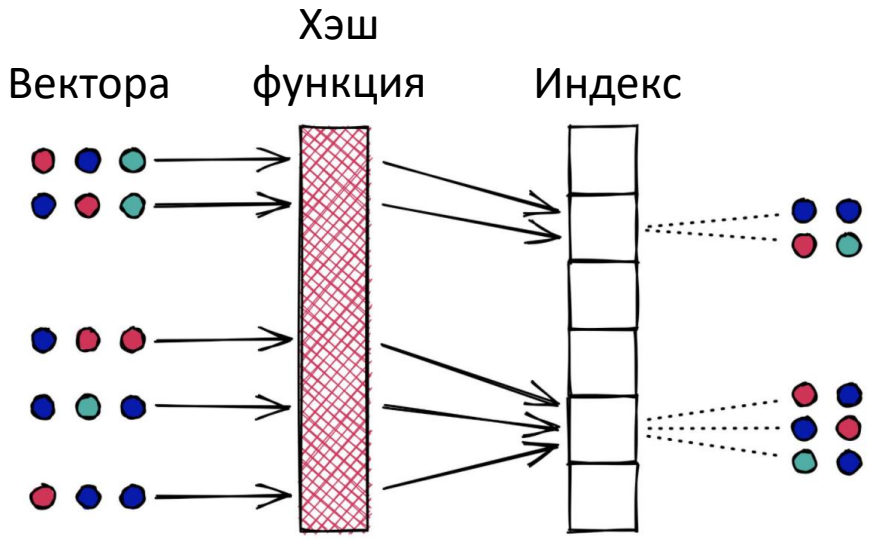
Задача нахождения в индексе векторов, наиболее близких к данному по заданной метрике.

Полный перебор

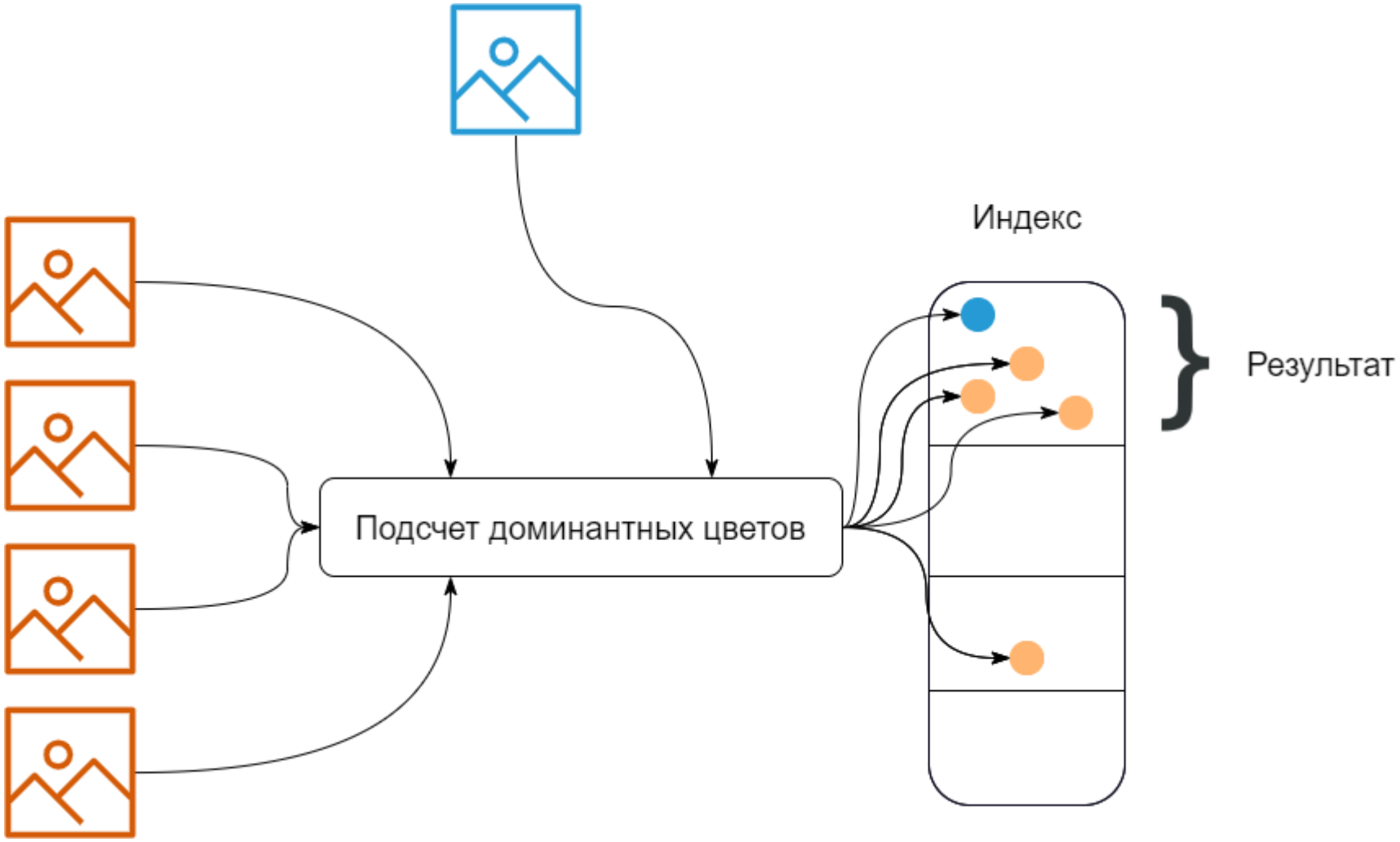
Считается расстояние между данным вектором и всеми сохраненными. Время поиска O(n).

Приближенный поиск с помощью LSH

Вектора разбиваются на группы с помощью хэш функции. В качестве результата выдается группа, в которую попал данный вектор. Время поиска O(1).



Пример применения



Суть работы

Мотивация:

Текущие реализации поиска ближайших соседей не позволяют автоматически сжимать индекс и динамически изменять его.

Невозможность динамически добавлять и удалять вектора из индекса приводит к необходимости его полного перестроения при каждом обновлении. Отсутствие возможности сжатия объема оперативной памяти, занимаемого индексом, требует использования существенных ресурсов при работе с большими объемами данных.

Идея:

Расширить возможности популярных библиотек для поиска ближайших соседей динамическим обновлением индекса и снижением объема оперативной памяти, требуемой для его использования.

Цели и задачи

Цель этой работы – Разработать надстройку над классом библиотек поиска ближайших соседей, позволяющую динамически обновлять индекс и сжимать его.

Задачи:

- Исследовать подходы к реализации приближенного поиска ближайших соседей
- Исследовать библиотеки на языке Python, предоставляющие функциональность поиска ближайших соседей
- Разработать надстройку над самыми популярными библиотеками, позволяющую динамически добавлять и удалять вектора из списка.
- Дополнить надстройку возможностью сжимать индекс.

Исследование библиотек

Название	Динамическое добавление	Динамическое удаление
Annoy	Нет	Нет
Faiss	Да (автоматически перестраивает индекс)	Да
NMSLIB	Нет	Нет
Flann	Нет	Нет
PANNS/MRPT+	Нет	Нет
Kgraph	Нет	Нет
RPForest	Нет	Нет
NGT	Нет	Нет
SPTAG	Да	Да
N2	Нет	Нет
ScaNN	Нет	Нет
scipy: cKDTree	Нет	Нет
datasketch	Нет	Нет
PyNNDescent	Нет	Нет
scikit-learn	Нет	Нет
NearPy	Да	Нет
PUFFINN	Нет	Нет

Исследование сервисов

В рамках исследования альтернатив производился поиск полноценных сервисов, предоставляющих функциональности поиска ближайших соседей.

Название	Динамическое добавление	Динамическое удаление
OpenSearch k-NN	Да	Да
DiskANN	Нет	Да
Vespa	Да	Да
qdrant	Да	Нет
EliasticNN	Да	Да

План работы

- Исследовать алгоритмы и структуры данных, позволяющие решить поставленные задачи
- Реализовать прототип, работающий со всеми популярными библиотеками
- Оценить скорость работы прототипа по сравнению с исходными библиотеками
- Подобрать алгоритм и структуру данных для прототипа, дающие оптимальный результат