Team 9

Phase one submission

Team Members:

Ahmed Samy

Kareem Samy

Nancy Ayman

Yara Hisham

Table of Contents

1st Approach: IVFPQ (Inverted File with Product Quantization)

Technique for efficient approximate nearest neighbor (ANN) search in large, high-dimensional datasets.

Components of IVFPQ

Inverted File Index (IVF):

It clusters the dataset into groups, allowing the search to focus only on relevant clusters, reducing the number of comparisons.

Product Quantization (PQ):

Within each cluster, data points are compressed using PQ, which represents each point with a compact code by splitting vectors into sub-vectors and approximating each sub-vector with a centroid from a precomputed codebook.

How it Works

Offline Stage (Indexing):

- The dataset is clustered into inverted lists, and each point is assigned to a cluster based on its proximity to the cluster's centroid.
- Each data point is then compressed using product quantization, creating a PQ code that approximates the data point's position in the vector space.
- These PQ codes are stored in the inverted lists for each cluster.

Online Stage (Querying):

- When a query vector is submitted, the algorithm first finds the clusters closest to the query by comparing the query to the centroids of each cluster.
- After identifying the most relevant clusters, the algorithm retrieves the PQ codes for the data points in those clusters.
- It then computes approximate distances between the query and the PQ codes of the selected data points to find the nearest neighbors.

Benefits:

- Speed: Limits comparisons to relevant clusters and uses compact codes for fast, approximate distance calculations.
- Memory Efficiency: Reduces the data footprint, making it suitable for large datasets.

Trade-offs

- **Speed vs. Accuracy**: IVFPQ sacrifices some accuracy for speed by using approximate distance measures.

Further Optimization: OPQ (Optimized Product Quantization)

Technique that enhances the traditional **Product Quantization (PQ)** method, widely used in approximate nearest neighbor (ANN) search, by reducing the quantization error. OPQ achieves this through a learned rotation or transformation of the data that makes the quantization process more efficient and accurate.

How OPQ Improves PQ

- **Data Transformation**: OPQ applies a learned linear transformation (typically a rotation matrix) to the data before performing product quantization. This transformation reorients the data so that it better aligns with the predefined clusters in PQ.
- **Reduced Quantization Error**: By rotating the data into a new space, OPQ minimizes the quantization error, meaning that the distance between the original data points and their quantized representations is smaller. This leads to more accurate approximations.
- **Training the Transformation Matrix**: The rotation or transformation matrix is optimized through training, often by minimizing quantization error on a subset of the data. This learning process makes OPQ more flexible and adaptable to different data distributions.

Trade-Offs

- **Increased Preprocessing**: Learning and applying the rotation matrix introduces extra preprocessing time.
- **Memory Usage for Transformation Matrix**: Storing and applying the learned transformation matrix may slightly increase memory usage.

2nd Approach: IVFADC (Inverted File with Asymmetric Distance Computation)

Components of IVFADC

Inverted File Index (IVF):

- First, IVFADC divides the dataset into clusters using a method like k-means. Each cluster has a centroid, and data points are assigned to the closest cluster.
- This clustering allows the algorithm to limit search only to relevant clusters rather than searching through the entire dataset, greatly improving speed.

Asymmetric Distance Computation (ADC):

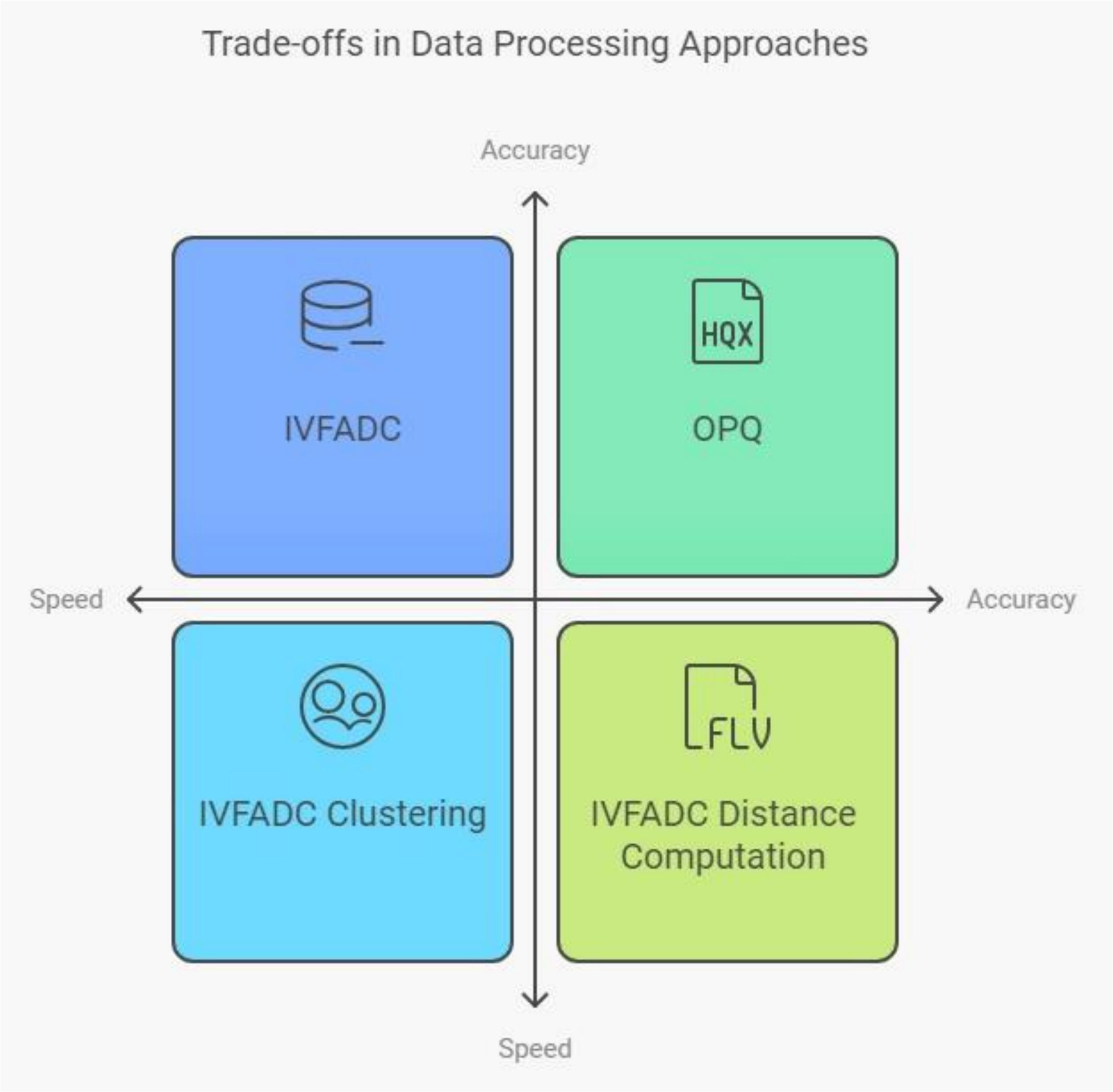
- After identifying the most relevant clusters for a query, IVFADC compares the query to the data points within those clusters.
- **Asymmetric** in this context means that, unlike Product Quantization (PQ) which quantizes both the database points and the query for comparisons, ADC quantizes only the database points and leaves the query uncompressed. This approach preserves more information from the query for more accurate distance computation.
- For each point within a cluster, IVFADC approximates the distance between the query and the database points using precomputed compressed representations of the database points.

How IVFADC Works During Search:

- When a query is provided, IVFADC identifies the top clusters based on which centroids are closest to the query.
- It then computes the distances from the query to the database points within these clusters using the compressed representations (using the ADC method).
- The system then ranks these points to find the closest ones to the query.

Trade-offs

- **Speed vs. Accuracy**: IVFADC sacrifices some speed for accuracy.


Trade-offs in Data Processing Approaches
Accuracy
IVFADC
HQX
OPQ
Speed
Accuracy
IVFADC Clustering
FLV
IVFADC Distance Computation
Speed