Product Quantization for Nearest Neighbor Search

Hervé Jégou, Matthijs Douze, Cordelia Schmid

Information Retrieval Course, A.Y. 2024/25

Student: Irene Testa

Outline

Methodology Overview

Product Quantization (PQ)

- **Goal**: Compress the data and enable efficient Approximate Nearest Neighbor (ANN) search.
- **Quantization process**: Split each vector into m subvectors, cluster each using K-Means with k^* centroids, and replace each subvector with its closest centroid index.
- **Distance Approximation**:
  - *Asymmetric Distance Computation* (ADC): Compute squared Euclidean distances from each query subvector to all subspace centroids and store them in lookup tables. Approximate the distance to a database vector by summing the stored distances corresponding to its centroid indices.
  - *Symmetric Distance Computation* (SDC): Precompute a look-up table of squared distances between centroid pairs. During search, quantize query subvectors and approximate the distance to a database vector by summing the corresponding precomputed distances.

Product Quantization with Inverted File Index (IVF)

- **Goal**: Limit distance computation to a subset of the database and reduce quantization error.
- **Indexing**: Database vectors are clustered into buckets using K-Means with k' centroids. Differences between each item and its bucket centroid (residuals) are encoded with PQ and stored in posting lists.
- **Non-exhaustive search**: Identify the w nearest buckets and compute distances only between the query residuals and items within those buckets.

Validation Experiments

Table 1: Summary of the datasets.

	Dimensions	Learning set size	Database set size	Queries set size
`siftsmall`	128	25 000	10 000	100
`sift`	128	100 000	1 000 000	10 000
`gist`	960	25 000	10 000	100
`glove`	300	25 000	10 000	100

Table 2: Comparison with the `faiss` library on the `siftsmall` dataset ($m = 8$, $k^* = 256$).

	Reconstruction Error	Nearest Recall@10	Average Kendall-τ
`faiss`	0.0899	0.8700	0.8734
`mine`	0.0893	0.8700	

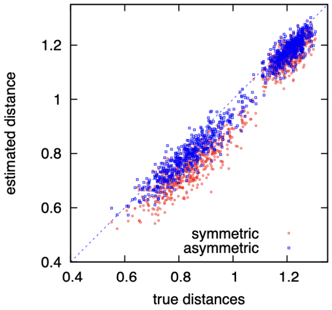

Figure 1: True vs. estimated distances between a typical SIFT query and database vectors ($m = 8$, $k^* = 256$). Results from the original paper.

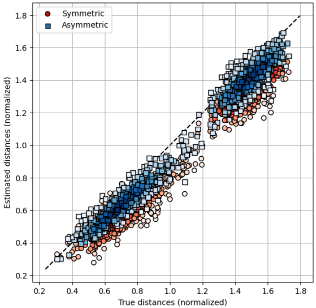

Figure 2: True vs. estimated distances between a typical SIFT query and database vectors ($m = 8$, $k^* = 256$). Replicated results.

Key Experiments

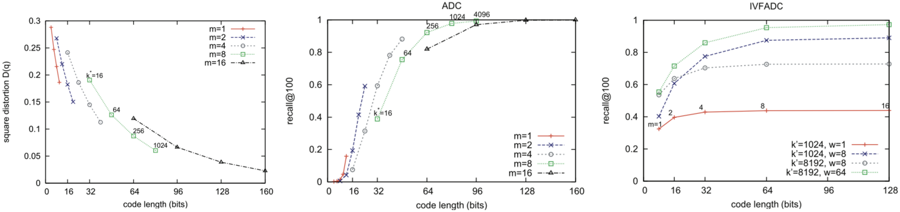

Figure 3: Experiments from the original paper on the `sift` dataset.

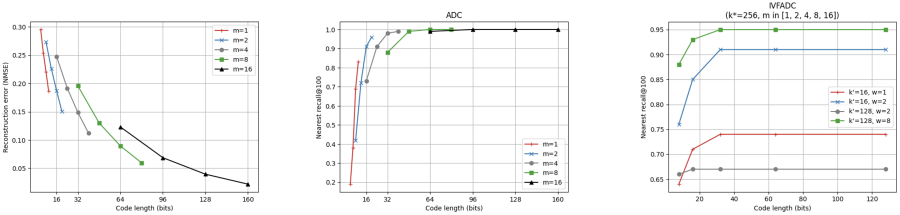

Figure 4: Reproduced experiments on the `siftsmall` dataset.

Data Pre-processing

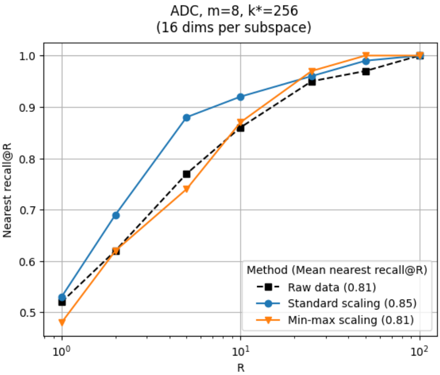

Figure 5: Effect of feature scaling on the `siftsmall` dataset.

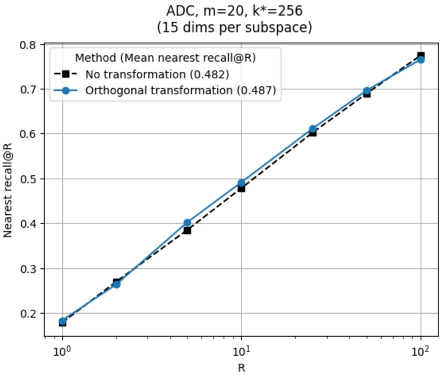

Figure 6: Effect of a random orthogonal transformation on the `glove` dataset.

Dimensionality Reduction

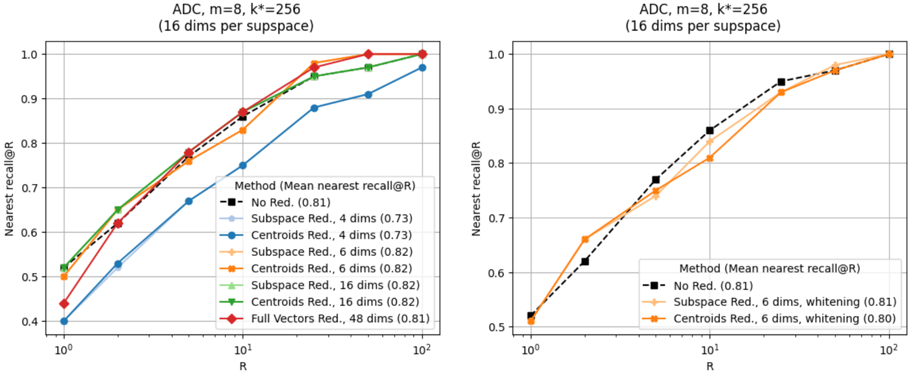

Figure 7: Effect of PCA on the `siftsmall` dataset. 'Subspace Red.': Subvectors are reduced, centroids computed and stored in the reduced space. 'Centroids Red.': Centroids are computed in the reduced space and then reprojected to the original space. 'Full Vectors Red.': Entire vectors are reduced before PQ training.

Figure 8: Effect of whitening on the `siftsmall` dataset.

Feature Partitioning

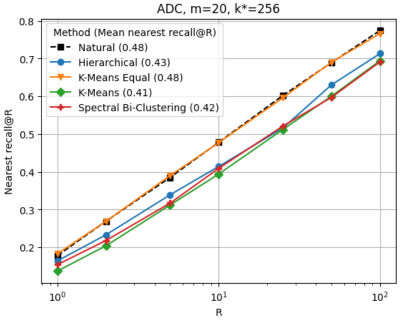

Figure 9: Effect of feature partitioning on the `glove` dataset.

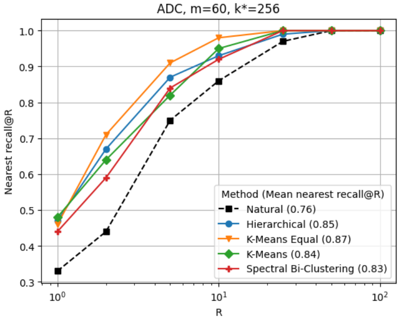

Figure 10: Effect of feature partitioning on the `gist` dataset.

Sample Weighting

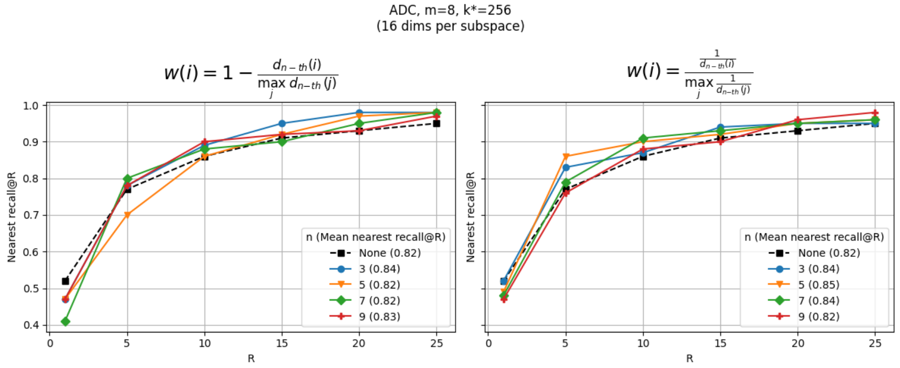

Figure 11: Effect of sample weighting in centroid computation on the `siftsmall` dataset.

Centroid Shrinkage

- Introduced in [1].
- Available in the `scikit-learn` implementation of `NearestCentroid`.
- Acts as **feature selection**: centroid features with little variation across centroids are set to zero.

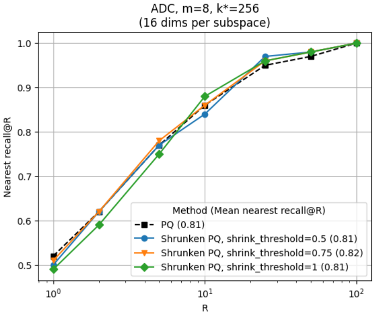

Figure 12: Effect of centroid shrinkage on the `siftsmall` dataset.

Fuzzy Product Quantization

- Proposed in [2], leveraging **Fuzzy C-Means** clustering:
  - 👍 Lower reconstruction error.
  - 👎 Slower convergence and an extra hyperparameter to tune.
- Records the **top two** centroids with highest membership probabilities and their **ratio** $r = \frac{p_2}{p_1}$:
  - 👎 Requires 4× PQ storage: 2 bytes for the indices (`np.int8`) and 2 bytes for the probability ratio (`np.float16`).
- Estimates distance via a **weighted average** of the distances between the query (ADC) and the two centroids, using $p_1 = \frac{1}{1+r}$ and $p_2 = \frac{r}{1+r}$ as weights.
  - 👍 Higher recall.
  - 👎 Two extra multiplications and one addition per vector.

Table 3: Comparison of PQ methods.

Method	Fuzzifier	Reconstruction Error	Avg. # iterations	Compression Factor	Recall@1
PQ	—	0.157	73	128	0.30
FuzzyPQ	1.1	0.139	196	32	0.39
FuzzyPQ	1.2	0.150	177	32	0.32
FuzzyPQ	1.3	0.207	94	32	0.23

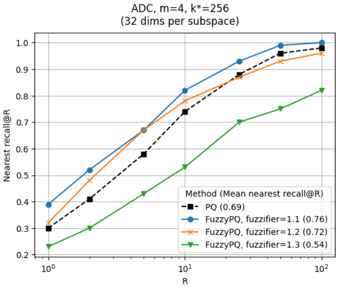

Figure 13: Search performance of FuzzyPQ on the `siftsmall` dataset with varying degree of fuzziness.

Coarse Clustering Algorithms

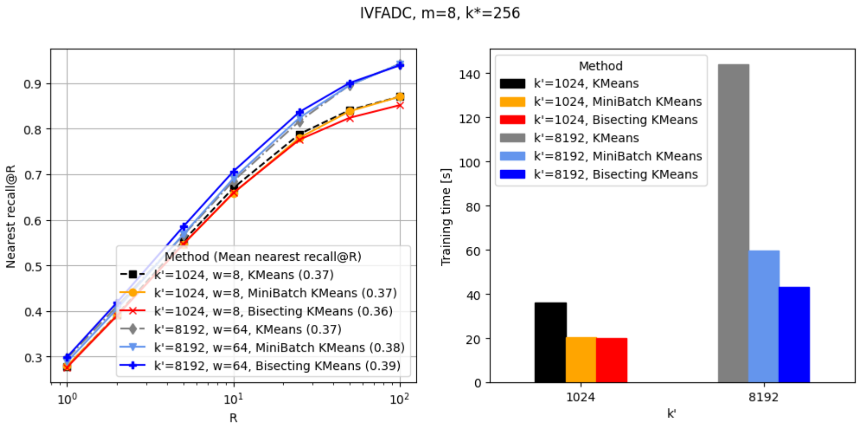

Figure 14: Comparison of K-Means variants as coarse clustering algorithms for building Inverted File Indexes: impact on search performance and time on the `sift` dataset.

Results Summary

- We **implemented** Product Quantization (PQ) and **validated** it against the `faiss` library.
- We successfully **replicated** the results from the original paper.
- We explored alternative approaches, with promising results from:
  - **Feature Scaling** and **Partitioning**: Often improve search performance with relatively low computational cost.
  - **Fuzzy PQ**: Reduces quantization error and boosts performance, at the cost of higher storage and training time.
  - **Bisecting K-Means** for coarse clustering: drastically reduces training time without affecting performance.

GitHub repository: https://github.com/iretes/IR-proj

References

[1] R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu, "Diagnosis of multiple cancer types by shrunken centroids of gene expression," *Proceedings of the National Academy of Sciences*, vol. 99, no. 10, pp. 6567–6572, 2002.

[2] X. Ding, Q. Hou, and X. Liu, "An improved product quantization method applying fuzzy clustering," in *2022 IEEE 6th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, IEEE, 2022, pp. 924–928.

[3] H. Jegou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 1, pp. 117–128, 2010.

[4] Y. Matsui, Y. Uchida, H. Jégou, and S. Satoh, "A survey of product quantization," *ITE Transactions on Media Technology and Applications*, vol. 6, no. 1, pp. 2–10, 2018.

Thank you!