Similarity search with polysemous codes

Matthijs Douze, Hervé Jégou, Florent Perronnin

Facebook AI Research

ECCV’2016
Problem setup

Build index for a collection:

\[ y_1, y_2, \ldots, y_n \in \mathbb{R}^d \]

Indexing

\[
\begin{bmatrix}
\vdots \\
\vdots \\
\vdots \\
\end{bmatrix}
\]

Media Description (CNN)

\[ x \in \mathbb{R}^d \]

Result:

\[ k = \text{argmin}_{i=1..n} \| x - y_i \|^2 \]

Query:

Approximate search
Criteria: compact, fast, accurate
Binary codes versus Product Quantization

Seen as concurrent methods in the literature

<table>
<thead>
<tr>
<th>Binarisation (ITQ)</th>
<th>PQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>comparison is context-free</td>
<td>need quantizer centroids</td>
</tr>
<tr>
<td>1190M comparisons / s</td>
<td>222M comparisons / s</td>
</tr>
<tr>
<td>precision=0.143</td>
<td>precision=0.442</td>
</tr>
<tr>
<td>Multi-index hashing</td>
<td>low memory overhead</td>
</tr>
<tr>
<td>high memory overhead</td>
<td>With an inverted file</td>
</tr>
</tbody>
</table>

[Iterative quantization: A procrustean approach to learning binary codes, Gong, Lazebnik, CVPR’11]

[Fast search in Hamming space with multi-index hashing, Norouzi, Pubjabi, Fleet, CVPR’12]

How to get the best of both worlds?

Polysemous codes, Douze, Jégou, Perronnin
Optimize the order of PQ centroids:
Index assignment

Given a k-means quantizer,
**learn** a permutation

So that  
binary comparison $\approx$ centroid distances
Optimize the order of PQ centroids: Index assignment

Before optimization

Optimization target

After optimization

Polysemous codes, Douze, Jégou, Perronnin
Binary codes for pre-filtering

Database polysemous codes

Query code

Query vector

PQ distance estimation

Binary distance filter

read

Nearest neighbor shortlist

Polysemous codes, Douze, Jégou, Perronnin

k ©2016.
Results on datasets of 1M-1G vectors

Memory usage is the same: focus on speed-accuracy tradeoff

Combines with non-exhaustive search: BIGANN (1G vectors)
2-2.5x faster, 0.5 ms per query for 1 thread