BeyOND – Unleashing BOND

Thomas Bernecker, Franz Graf, Hans-Peter Kriegel, Christian Moennig and Arthur Zimek

Ludwig-Maximilians-Universität München (LMU)
Munich, Germany
http://www.dbs.ifi.lmu.de
{bernecker, graf, kriegel, zimek}@dbs.ifi.lmu.de
moennig@cip.ifi.lmu.de
Outline

1. Background
   - Motivation: k-nearest neighbor search in high-dimensional databases
   - BOND revisited

2. Introducing BeyOND
   - Filtering objects via distance approximations
   - Sub Cubes, MBRs

3. Experimental Evaluation

4. Conclusions
Motivation

• Similarity search in high-dimensional space is
  😊 important in cases of images, e-commerce, etc.
  😞 slow

• The suitability of index-based solutions depends on the data distribution

• Open question: relevant vs. irrelevant attributes

• Similarity search in subspaces:
  – Fix query attributes beforehand
  – Use multiple pivot points to derive upper and lower bounds
  – Process data vertically to reduce the high-dimensional space
• BOND\(^{[1]}\): k-nearest neighbor search on high-dimensional data
  – Resolves feature vectors (FVs) column-wise
  – Ranking of columns w.r.t. relevance
  – Pruning of columns using a branch-and-bound approach
  – Resolved part is known exactly
  – Unresolved part has to be approximated
  – Resolving stops when approximation is “good enough”
  – Support of subspace queries
  – Distance metrics:
    • Histogram intersection (uncorrelated dimensions)
    • Euclidean distance

\[1\] de Vries, Mamoulis, Nes, Kersten: Efficient k-NN Search On Vertically Decomposed Data (SIGMOD’02)
BOND Revisited (2)

• Restrictions of BOND:

1. The approach works only on Zipfian distributed data.

2. The feature values are normalized to [0,1] in each dimension.

3. The proposed bounds are loose. The validity of stricter bounds (Bond advanced) depends on a certain resolve order of the columns.
BOND Revisited (3)

- Notation:
  - query vector \( q \), database vector \( v \)
  - Splitting of \( v \): resolved part \( v^- \), unresolved part \( v^+ \) \( \Rightarrow \) \( v = v^- \cup v^+ \)

- Approximated distance:
  \[ S_{approx}(q,v) = S_1(q^-, v^-) + S_2(q^+, v^+) \]
  
  - Resolved part: \( S_1(q^-, v^-) = \sum_i (q_i^- - v_i^-)^2 \)
  
  - Unresolved part: \( S_2(q^+, v^+) = \sum_i \max\{q_i^+, 1 - q_i^+\}^2 \geq S_1(q^+, v^+) \)

- Distance bounds:
  \[ S_{upper}(q,v) = S_1(q^-, v^-) + S_2(q^+, v^+) \geq S_1(q,v) \]
  \[ S_{lower}(q,v) = S_1(q^-, v^-) + 0 \leq S_1(q,v) \]
Beyond BOND

• Benefits of BeyOND:
  1. Independence of the data distribution. 😊
  2. No restriction to a normalized data space. 😊
  3. No specific resolve order of the dimensions is needed. 😊

⇒ Price: Distance approximations are no more suitable! 😞

• Solution: Combining the idea of BOND with well-known techniques:
  – VA-file (data space partitioning)
  – MBR (Minimum Bounding Rectangle) approximation (data organizing)

⇒ Remaining restriction: minimum/maximum values for each dimension need to be known 😞
Sub Cubes (1)

- First extension: VA-file\textsuperscript{[2]} with one split
  \(\Rightarrow 2^d\) sub cubes
  \(\Rightarrow\) Addressing via Z-IDs
  \(\Rightarrow\) Improved bounds based on the close / far sub cube borders \(c_{v_i}^{\text{lower}}\) and \(c_{v_i}^{\text{upper}}\)

- Memory-efficient representation (8 bytes \(\rightarrow\) 1 bit)
  - Sub cube need not be kept in main memory

- Split positions stored in one separate array per dimension

- Dependence on split level:
  - FV: 8 bytes per dimension
  - \(s\) splits: \(s / 8\) bytes (\(s\) bits) per dimension

\textsuperscript{[2]} Weber, Schek, Blott. \textit{A Quantitative Analysis and Performance Study for Similarity Search Methods in High-Dimensional Spaces} (VLDB'98)
Sub Cubes (2)

• Old distance bounds:

\[ S_{\text{upper}}(q, v) = S_1(q^-, v^-) + \sum_i \max\{q_i^+, 1 - q_i^+\}^2 \]
\[ S_{\text{lower}}(q, v) = S_1(q^-, v^-) + 0 \]

• Approximations of unresolved dimensions:

\[ S'_2(q^+, v^+) = \sum_i \max\{q_i^+ - c_{v_i^+}^\text{lower}, |q_i^+ - c_{v_i^+}^\text{upper}|\}^2 \]
\[ S''_2(q^+, v^+) = \sum_i \min\{q_i^+ - c_{v_i^+}^\text{lower}, |q_i^+ - c_{v_i^+}^\text{upper}|\}^2 \]

if \( q_i^+ \in [c_{v_i^+}^\text{lower}, c_{v_i^+}^\text{upper}] \)

else

• New distance bounds:

\[ S'_{\text{upper}}(q, v) = S_1(q^-, v^-) + S'_2(q^+, v^+) \geq S_1(q, v) \]
\[ S'_{\text{lower}}(q, v) = S_1(q^-, v^-) + S''_2(q^+, v^+) \leq S_1(q, v) \]
MBR Caching (1)

- Most sub cubes are (very) sparse, i.e. occupied by at most one FV

- Dense sub cubes allow a tighter approximation via MBRs
  - Restrict the number of MBRs in order to avoid a memory overhead
  - Ranking function for MBRs:
    \[
    f(MBR) = \frac{V_{sub\ cube}}{V_{MBR}} \cdot card(MBR)
    \]
  - 8 byte coordinates: memory increase is limited to \( \frac{d \cdot 16}{card(MBR)} \) bytes per feature vector (+ pointer to Z-ID)
MBR Caching (2)

• Limit the number of MBRs to 1% of the database size
• Threshold as a trade-off between pruning power and additional memory consumption
• Requirements:
  – Either all MBRs can be kept in memory,
  – or the time for loading the MBRs is less than the time for resolving the respective FVs.
• Adaption of the equations for lower and upper bounds
Experimental Evaluation (1)

- Evaluated approaches:
  1. BondAdvanced (stricter bounds, but resolve order dependent)
  2. Bond (original bounds)*
  3. Sequential*
  4. Beyond-1 (1 split)
  5. BeyondMBR-1 (1 split + MBRs)
  6. Beyond-2
  7. BeyondMBR-2
  8. Beyond-3*
  9. BeyondMBR-3*
Experimental Evaluation (2)

- Data set descriptions:

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Dims</th>
<th>Size</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALOI</td>
<td>27</td>
<td>110,250</td>
<td>Color Histograms, Zipfian</td>
</tr>
<tr>
<td>CLUSTERED</td>
<td>20</td>
<td>500,000</td>
<td>Synthetic, 50 Clusters, Gaussian</td>
</tr>
<tr>
<td>PHOG(^3)</td>
<td>110</td>
<td>10,715</td>
<td>CT Histograms, PCA‘ed</td>
</tr>
<tr>
<td>SIFT(^4)</td>
<td>133</td>
<td>335,583</td>
<td>Image Features</td>
</tr>
</tbody>
</table>

\(^3\) Graf, Kriegel, Schubert, Poelsterl, Cavallaro. *2D Image Registration in CT Images Using Radial Image Descriptors* (MICCAI’11)

Experimental Evaluation (3)

- Experimental settings:
  - 50 k-nearest neighbor queries
  - $k = 10$
  - Averaged cumulative number of pruned FVs after resolving a column
  - AUC: data not resolved
  - AOC: data resolved for refinement
Experimental Evaluation (4)

| ALOI   |   27 | 110,250 | Color Histograms, Zipfian |

![Graph showing performance comparison of BondAdvanced, Bond, Beyond-1, BeyondMBR-1, and Beyond-2 algorithms.](image-url)
Experimental Evaluation (5)

CLUSTERED  20  500,000  Synthetic, 50 Clusters, Gaussian

![Graph showing performance comparison between BondAdvanced, Bond, Beyond-1, BeyondMBR-1, and Beyond-2 in a clustered environment with 20 dimensions and 500,000 synthetic data points. The graph measures the number of pruned vectors against the number of resolved dimensions.]
Experimental Evaluation (6)

PHOG | 110 | 10,715 | CT Histograms, PCA‘ed

- BondAdvanced
- Bond
- Beyond-1
- BeyondMBR-1
- Beyond-2

BeyOND – Unleashing BOND
### Experimental Evaluation (7)

#### Pruning power (Sub cubes)

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Splits</th>
<th>25% pruned</th>
<th>50% pruned</th>
<th>90% pruned</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALOI</td>
<td>1</td>
<td>16 (59%)</td>
<td>19 (70%)</td>
<td>23 (85%)</td>
</tr>
<tr>
<td>CLUSTERED</td>
<td>1</td>
<td>7 (35%)</td>
<td>8 (40%)</td>
<td>10 (50%)</td>
</tr>
<tr>
<td>PHOG</td>
<td>1</td>
<td>45 (41%)</td>
<td>58 (53%)</td>
<td>80 (73%)</td>
</tr>
<tr>
<td>ALOI</td>
<td>2</td>
<td>7 (26%)</td>
<td>9 (33%)</td>
<td>21 (75%)</td>
</tr>
<tr>
<td>CLUSTERED</td>
<td>2</td>
<td>1 (5%)</td>
<td>1 (5%)</td>
<td>1 (5%)</td>
</tr>
<tr>
<td>PHOG</td>
<td>2</td>
<td>45 (41%)</td>
<td>55 (50%)</td>
<td>79 (72%)</td>
</tr>
</tbody>
</table>

#### Pruning power (Sub cubes + MBRs)

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Splits</th>
<th>25% pruned</th>
<th>50% pruned</th>
<th>90% pruned</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALOI</td>
<td>1</td>
<td>1 (4%)</td>
<td>1 (4%)</td>
<td>10 (37%)</td>
</tr>
<tr>
<td>CLUSTERED</td>
<td>1</td>
<td>1 (5%)</td>
<td>1 (5%)</td>
<td>1 (5%)</td>
</tr>
<tr>
<td>PHOG</td>
<td>1</td>
<td>37 (34%)</td>
<td>50 (45%)</td>
<td>77 (70%)</td>
</tr>
</tbody>
</table>

#### # Accessed columns

<table>
<thead>
<tr>
<th>Data Set</th>
<th>1 split</th>
<th>2 splits</th>
<th>1 split + MBR</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALOI</td>
<td>66.9%</td>
<td>38.3%</td>
<td>7.7%</td>
</tr>
<tr>
<td>CLUSTERED</td>
<td>34.1%</td>
<td>1.6%</td>
<td>1.4%</td>
</tr>
<tr>
<td>PHOG</td>
<td>52.6%</td>
<td>52.3%</td>
<td>45.4%</td>
</tr>
</tbody>
</table>
Experimental Evaluation (8)

<table>
<thead>
<tr>
<th>ALOI</th>
<th>27</th>
<th>110,250</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Color Histograms, Zipfian</td>
</tr>
</tbody>
</table>

Data resolve & pruning (all in RAM!)
Time for approximations
Amount of pruned data

BeyOND – Unleashing BOND
Experimental Evaluation (9)

PHOG 110 10,715 CT Histograms, PCA‘ed

BeyOND – Unleashing BOND
Experimental Evaluation (10)

**SIFT**

| 133 | 335,583 |

**Image Features**

**Time for approximations**

- Beyond-1: 0.355
- Beyond-2: 0.610
- Beyond-3: 0.935
- BeyondMBR-1: 0.629
- BeyondMBR-2: 0.853
- BeyondMBR-3: 0.526
- Bond: 0.000
- BondAdvanced: 0.000

**Data resolve & pruning (all in RAM!)**

- Beyond-1: 0.355
- Beyond-2: 0.610
- Beyond-3: 0.935
- BeyondMBR-1: 0.629
- BeyondMBR-2: 0.853
- BeyondMBR-3: 0.526
- Bond: 0.000
- BondAdvanced: 0.000

**Amount of pruned data**

- Beyond-1: 0.355
- Beyond-2: 0.610
- Beyond-3: 0.935
- BeyondMBR-1: 0.629
- BeyondMBR-2: 0.853
- BeyondMBR-3: 0.526
- Bond: 0.000
- BondAdvanced: 0.000
Conclusions

• Removed restrictions...
  1. Independence of the data distribution.
  2. No restriction to a normalized data space.
  3. No specific resolve order of the dimensions is needed.

• Combination of relevant techniques...
  – VA-file-based partitioning of the data space
  – MBR caching

• Still open issues...
  – Trade-off: split level vs. pruning power
  – Trade-off: MBR memory consumption vs. pruning power
  – Sophisticated techniques for the creation of the MBRs
  – Overcome the restriction that the vector lengths have to be known
Thank you for listening!

Any questions?

http://www.dbs.ifi.lmu.de/cms/Publications/BeyOND__Unleashing_BOND