Clustering methods: Part 5

Fast search methods

Pasi Fränti

5.5.2014

Speech and Image Processing Unit
School of Computing
University of Eastern Finland
Methods considered

Classical speed-up techniques
• Partial distortion search (PDS)
• Mean distance ordered partial search (MPS)

Speed-up of k-means
• Reduced-search based on centroid activity

External search data structures
• Nearest neighbor graph
• Kd-tree
Partial distortion search (PDS)


- Current best candidate gives upper limit.
- Distances calculated cumulatively.
- After each addition, check if the partial distortion exceeds the smallest distance found so far.
- If it exceeds, then terminate the search.

\[
e_{a,j} = \sum_{k=1}^{K} \left( c_{ak} - c_{jk} \right)^2
\]
Mean-distance ordered partial search (MPS)

- Calculate distance along projection axis.
- If distance is outside bounding circle defined by the best candidate, drop the vector.

\[
\left( \sum_{k=1}^{K} x_i^k - \sum_{k=1}^{K} c_b^k \right)^2 > K \cdot d(x_i, c_a)
\]
Bounds of the MPS method

Input vector

Bound

Best candidate

A

A'

B

B'

C

C'

C'
Pseudo code of MPS search

\textbf{SearchNearestNeighborUsingMPS}(c_a, c_j, d_{min}) \rightarrow nn_a, d_a;

\begin{align*}
  &d_{min} \leftarrow \infty; \\
  &\text{up} \leftarrow \text{TRUE}; \\
  &\text{down} \leftarrow \text{TRUE}; \\
  &j_1 \leftarrow a; \\
  &j_2 \leftarrow a; \\
  &\text{WHILE (up OR down) DO} \\
  &\quad \text{IF up THEN} \\
  &\quad \quad j_1 \leftarrow j_1 + 1; \\
  &\quad \quad \text{IF } j_1 > N \text{ THEN up \leftarrow FALSE} \\
  &\quad \quad \text{ELSE CheckCandidate}(s_a, s_{j1}, n_a, d_{min}, nn, \text{up}); \\
  &\quad \text{IF down THEN} \\
  &\quad \quad j_2 \leftarrow j_2 - 1; \\
  &\quad \quad \text{IF } j_2 < 1 \text{ THEN down \leftarrow FALSE} \\
  &\quad \quad \text{ELSE CheckCandidate}(s_a, s_{j2}, n_a, d_{min}, nn, \text{down}); \\
  &\quad \text{END-WHILE;}
\end{align*}

RETURN $nn, d_{min}$;
Activity classification


---

**Code vectors:**
- Active
- Static
- Previous

**Training vectors:**
- Moved farther
- Moved closer
- No change
Reduced search based on activity classification

Sentroidit:
- Aktiivinen, uusi sijainti
- Aktiivinen, vanha sijainti
- Staattinen

Alkiot:
- △ Sentroidi siirtynyt lähemmäksi alkiota
- ▲ Sentroidi siirtynyt kauemmaksi alkiosta
- ◇ Etäisyydessä sentroidiin ei muutosta

Lähimmän sentroidin haku määrytyy seuraavasti:

<table>
<thead>
<tr>
<th>▲</th>
<th>△</th>
<th>◇</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>O</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

T = täysi haku
O = osittainen haku

Etäisyyshakujen määrä:

<table>
<thead>
<tr>
<th>▲</th>
<th>△</th>
<th>◇</th>
</tr>
</thead>
<tbody>
<tr>
<td>100%</td>
<td>4%</td>
<td>4%</td>
</tr>
<tr>
<td>0%</td>
<td>0%</td>
<td>4%</td>
</tr>
</tbody>
</table>

Osuus alkioista:

<table>
<thead>
<tr>
<th>▲</th>
<th>△</th>
<th>◇</th>
</tr>
</thead>
<tbody>
<tr>
<td>3,6%</td>
<td>3,7%</td>
<td>0,1%</td>
</tr>
<tr>
<td>0%</td>
<td>0%</td>
<td>92,6%</td>
</tr>
</tbody>
</table>
Classification due to iterations

(Number of training vectors vs. Iteration)

- Moved closer
- Moved farther
- No change
Activity of vectors in Random Swap

![Graph showing the activity of vectors in Random Swap](image)

- **Y-axis (Amount of full searches):** 0 to 30%
- **X-axis (Iteration):** 0 to 10,000

The graph illustrates the activity of vectors over iterations, with the amount of full searches expressed as a percentage.
# Effect on distance calculations

## K-means

<table>
<thead>
<tr>
<th></th>
<th>Distance calculations / search</th>
<th>Dimensions / distance calculation</th>
<th>Dimensions / search</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full</td>
<td>255.97</td>
<td>16.00</td>
<td>4095.48</td>
</tr>
<tr>
<td>PDS</td>
<td>255.97</td>
<td>2.34</td>
<td>598.96</td>
</tr>
<tr>
<td>MPS+PDS</td>
<td>26.97</td>
<td>8.07</td>
<td>217.60</td>
</tr>
</tbody>
</table>

## K-means with activity classification

<table>
<thead>
<tr>
<th></th>
<th>Distance calculations / search</th>
<th>Dimensions / distance calculation</th>
<th>Dimensions / search</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full</td>
<td>61.44</td>
<td>16.00</td>
<td>983.07</td>
</tr>
<tr>
<td>PDS</td>
<td>61.44</td>
<td>2.08</td>
<td>127.98</td>
</tr>
<tr>
<td>MPS+PDS</td>
<td>5.35</td>
<td>6.72</td>
<td>35.97</td>
</tr>
</tbody>
</table>
Effect on processing time

For improving K-means algorithm

<table>
<thead>
<tr>
<th></th>
<th>Bridge</th>
<th>Miss America</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Without grouping</td>
<td>With grouping</td>
</tr>
<tr>
<td>Full</td>
<td>127.6</td>
<td>46.1</td>
</tr>
<tr>
<td>PDS</td>
<td>33.4</td>
<td>13.0</td>
</tr>
<tr>
<td>MPS+PDS</td>
<td>12.4</td>
<td>4.8</td>
</tr>
</tbody>
</table>

3.8 %  1.6 %
Comparison of speed-up methods

![Graph comparing running time with codebook size for Full, PDS, TIE, and MPS methods.](image_url)
Improvement of reduced search

![Graph showing the reduction of running time with varying codebook sizes for different methods: Full, PDS, TIE, and MPS. The y-axis represents the reduction of running time in percentage, ranging from 0% to 100%. The x-axis represents the codebook size, ranging from 16 to 1024. The graph illustrates the efficiency of each method as the codebook size increases.](image-url)
Neighborhood graph

Full search: \( O(N) \) distance calculations.

Graph structure: \( O(k) \) distance calculations.
Sample graph structure


5. C. Elkan. Using the Triangle Inequality to Accelerate k-Means. *Int. Conf. on Machine Learning*, (ICML'03), pp. 147-153.
