Kernelized Sorting for NLP

Jagadeesh Jagarlamudi*, Seth Juarez & Hal Daumé III*
{jags, seth, hal}@cs.utah.edu
School of Computing, Univ. of Utah

*Computer Science Dept., University of Maryland
Matching Problem

Hal

Obama

Bill Gates
Given sets of objects $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ from two domains, recover the underlying unknown alignment with out any cross-domain clues
Problem and Its relevancy

Given sets of objects $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ from two domains, recover the underlying unknown alignment without any cross-domain clues.

<table>
<thead>
<tr>
<th>Task</th>
<th>Objects</th>
<th>Domains</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document Alignment</td>
<td>Documents</td>
<td>Diff. languages</td>
</tr>
<tr>
<td>Sentence Alignment</td>
<td>Sentences</td>
<td>Diff. languages</td>
</tr>
<tr>
<td>Transliteration Mining</td>
<td>Names</td>
<td>Diff. languages</td>
</tr>
<tr>
<td>POS Tagging</td>
<td>Words, POS Tags</td>
<td>Vocabulary, POS Tag. set</td>
</tr>
</tbody>
</table>
Given sets of objects $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ from two domains, recover the underlying unknown alignment without any cross-domain clues.

<table>
<thead>
<tr>
<th>Task</th>
<th>Objects</th>
<th>Domains</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document Alignment</td>
<td>Documents</td>
<td>Diff. languages</td>
</tr>
<tr>
<td>Sentence Alignment</td>
<td>Sentences</td>
<td>Diff. languages</td>
</tr>
<tr>
<td>Transliteration Mining</td>
<td>Names</td>
<td>Diff. languages</td>
</tr>
<tr>
<td>POS Tagging</td>
<td>Words, POS Tags</td>
<td>Vocabulary, POS Tag. set</td>
</tr>
</tbody>
</table>

Rest of the talk

- Problem and its relevancy
- Kernelized Sorting
- Problems in Adapting to NLP
  - Sub-polynomial Kernels
  - Bootstrapping
- Experimental Setup
- Discussion & Conclusion
Intuition

Domain 1

Domain 2
Intuition

Domain 1

Domain 2
Intuition

Domain 1

Domain 2
Intuition

Align these points

Domain 1

Domain 2
Intuition

Domain 1

Align these points

Domain 2
Intuition

Align these points
Align another pair

Domain 1

Domain 2
Intuition

Domain 1

Domain 2

Align these points
Align another pair
... and continue
Intuition

- Uses the similarities between objects within same domain.
- Me and my counter part should have similar similarity vectors.
Kernelized Sorting (Quadrianto, N. et al. 2009)

- Uses the similarities between objects within same domain.
  - Kernel matrices (K, L)
- Me and my counter part should have similar similarity vectors.
Kernelized Sorting (Quadrianto, N. et al. 2009)

- Uses the similarities between objects within same domain.
  - Kernel matrices (K, L)
  - Me and my counterpart should have similar similarity vectors.

Formally:
Kernelized Sorting (Quadrianto, N. et al. 2009)

- Uses the similarities between objects within same domain.
  - Kernel matrices \((K, L)\)
  - Me and my counter part should have similar similarity vectors.

Formally:

\[
\begin{bmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

Permutation
Kernelized Sorting (Quadrianto, N. et al. 2009)

- Uses the similarities between objects within same domain.
  - Kernel matrices (K, L)
  - Me and my counter part should have similar similarity vectors.

Formally:

\[
\begin{pmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 \\
\end{pmatrix}
\]

Permutation

Find a \( \pi^* \) s.t. \( \text{tr}(K\pi^T L\pi) \) is maximized
Kernelized Sorting

Figure: Kernelized sorting to match observations in $Y$ & $X$
Problems in Adapting to NLP

1. Sensitivity to initialization
Problems in Adapting to NLP

1. Sensitivity to initialization
2. High Dimensionality data
Problems in Adapting to NLP

1. Sensitivity to initialization
2. High Dimensionality data
3. Diagonal Dominance
Problems in Adapting to NLP

1. Sensitivity to initialization
2. High Dimensionality data
3. Diagonal Dominance
   \[ \sum_{i \neq j} \tilde{K}_{ij}\hat{L}_{ji} + \sum_{i} \tilde{K}_{ii}\hat{L}_{ii} \]
Diagonal Dominance

- Sub-polynomial kernel

- Idea is to smooth the kernels
  - $k_{SP}(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle^p \quad x_i \in \mathbb{R}^d$
  - Ratio of diagonal elements to off-diagonal elements decrease

- Simply raising each element to power $p \in [0, 1]$
  - Normalize to unit length
  - $K_{SP} \leftarrow K_{SP} \times K_{SP}^T$
Bootstrapping

- Relax the binary constraint
- Also, provide some seed alignments
Bootstrapping

- Relax the binary constraint
- Also, provide some seed alignments

\[
\begin{align*}
y_1 & \quad x_1 \\
y_2 & \quad x_2 \\
y_3 & \quad x_3 \\
y_4 & \quad x_4
\end{align*}
\]
**Bootstrapping**

- Relax the binary constraint
- Also, provide some seed alignments

\[
\begin{bmatrix}
0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

Zero Wts
Bootstrapping

- Relax the binary constraint
- Also, provide some seed alignments

\[
\begin{pmatrix}
0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{pmatrix}
\]

Zero Wts

\[
\begin{pmatrix}
0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 \\
0 & 0 & 0.5 & 0.5 \\
0 & 0 & 0.5 & 0.5
\end{pmatrix}
\]

Uniform Wts
Bootstrapping

- Relax the binary constraint
- Also, provide some seed alignments

\[
\begin{bmatrix}
0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}
\]

Zero Wts

\[
\begin{bmatrix}
0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 \\
0 & 0 & 0.5 & 0.5 \\
0 & 0 & 0.5 & 0.5
\end{bmatrix}
\]

Uniform Wts

- We do this at each iteration
Results

- Image alignment, Doc. alignment, Transliteration Mining
Results

- Image alignment, Doc. alignment, Transliteration Mining
- Baseline System: KS, MCCA (Highighi, A. et al. 2008)
Results

- Image alignment, Doc. alignment, Transliteration Mining
- Baseline System: KS, MCCA (Highighi, A. et al. 2008)
- 10 Random seed alignments

<table>
<thead>
<tr>
<th></th>
<th>KS</th>
<th>MCCA</th>
<th>p-smooth</th>
<th>Seed-U</th>
<th>Seed-Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMG (320)</td>
<td>137</td>
<td>35</td>
<td>136</td>
<td>144</td>
<td>144</td>
</tr>
<tr>
<td>EP-P (250)</td>
<td>6</td>
<td>250</td>
<td>250</td>
<td>250</td>
<td>250</td>
</tr>
<tr>
<td>EP-C (226)</td>
<td>2</td>
<td>34</td>
<td>41</td>
<td>88</td>
<td>88</td>
</tr>
<tr>
<td>WP (115)</td>
<td>5</td>
<td>48</td>
<td>11</td>
<td>76</td>
<td>81</td>
</tr>
<tr>
<td>TL (300)</td>
<td>7</td>
<td>209</td>
<td>30</td>
<td>250</td>
<td>262</td>
</tr>
</tbody>
</table>
With Data

![Graph showing the number of recovered alignments against the number of pairs in the data. The graph includes two lines labeled 'KS' and 'p-Smooth.' The x-axis represents the number of pairs in the data, while the y-axis represents the number of recovered alignments.]
With Data and $p$
Assumption about alignment between objects

<table>
<thead>
<tr>
<th></th>
<th>#Aligned</th>
<th>#Noisy</th>
<th>MCCA</th>
<th>S-Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>EP-C</td>
<td>200</td>
<td>25</td>
<td>31 (34)</td>
<td>69 (88)</td>
</tr>
<tr>
<td>TL</td>
<td>300</td>
<td>50</td>
<td>185 (209)</td>
<td>236 (262)</td>
</tr>
</tbody>
</table>
Discussion

1. Assumption about alignment between objects

<table>
<thead>
<tr>
<th></th>
<th>#Aligned</th>
<th>#Noisy</th>
<th>MCCA</th>
<th>S-Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>EP-C</td>
<td>200</td>
<td>26</td>
<td>31 (34)</td>
<td>69 (88)</td>
</tr>
<tr>
<td>TL</td>
<td>300</td>
<td>50</td>
<td>185 (209)</td>
<td>236 (262)</td>
</tr>
</tbody>
</table>

2. Unequal number of objects on both sides
Discussion

1. Assumption about alignment between objects

<table>
<thead>
<tr>
<th></th>
<th>#Aligned</th>
<th>#Noisy</th>
<th>MCCA</th>
<th>S-Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>EP-C</td>
<td>200</td>
<td>26</td>
<td>31 (34)</td>
<td>69 (88)</td>
</tr>
<tr>
<td>TL</td>
<td>300</td>
<td>50</td>
<td>185 (209)</td>
<td>236 (262)</td>
</tr>
</tbody>
</table>

2. Unequal number of objects on both sides
   - Add dummy nodes with high weights
## Discussion

### Assumption about alignment between objects

<table>
<thead>
<tr>
<th></th>
<th>#Aligned</th>
<th>#Noisy</th>
<th>MCCA</th>
<th>S-Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>EP-C</td>
<td>200</td>
<td>26</td>
<td>31 (34)</td>
<td>69 (88)</td>
</tr>
<tr>
<td>TL</td>
<td>300</td>
<td>50</td>
<td>185 (209)</td>
<td>236 (262)</td>
</tr>
</tbody>
</table>

### Unequal number of objects on both sides
- Add dummy nodes with high weights

### Allow ambiguous alignments
Discussion

1. Assumption about alignment between objects

<table>
<thead>
<tr>
<th></th>
<th>#Aligned</th>
<th>#Noisy</th>
<th>MCCA</th>
<th>S-Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>EP-C</td>
<td>200</td>
<td>26</td>
<td>31 (34)</td>
<td>69 (88)</td>
</tr>
<tr>
<td>TL</td>
<td>300</td>
<td>50</td>
<td>185 (209)</td>
<td>236 (262)</td>
</tr>
</tbody>
</table>

2. Unequal number of objects on both sides
   - Add dummy nodes with high weights

3. Allow ambiguous alignments

4. Efficiency, by exploiting local neighbourhood
Discussion

1. Assumption about alignment between objects

<table>
<thead>
<tr>
<th></th>
<th>#Aligned</th>
<th>#Noisy</th>
<th>MCCA</th>
<th>S-Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>EP-C</td>
<td>200</td>
<td>26</td>
<td>31 (34)</td>
<td>69 (88)</td>
</tr>
<tr>
<td>TL</td>
<td>300</td>
<td>50</td>
<td>185 (209)</td>
<td>236 (262)</td>
</tr>
</tbody>
</table>

2. Unequal number of objects on both sides
   - Add dummy nodes with high weights

3. Allow ambiguous alignments

4. Efficiency, by exploiting local neighbourhood

5. Possibility to learn the value of $p$ automatically
Take away !!

- Identified problems with adapting KS to NLP
- Simple solutions
  - Sub-polynomial kernel smoothing
  - Seed alignments combined with relaxation
- More number of recovered alignments in parallel corpora
Take away !!

- Identified problems with adapting KS to NLP
- Simple solutions
  - Sub-polynomial kernel smoothing
  - Seed alignments combined with relaxation
- More number of recovered alignments in parallel corpora

Thank You !!