ML-NCA
Multi-Label Neighbourhood Component Analysis

Arjun Pakrashi, Payel Sadhukhan, Brian Mac Namee
Introduction
Introduction
Multi-label

- Multi-label classification:
  - One datapoint can be classified to belong to more than one class at the same time
    - $x$ has L1, L3 and L4 labels applied within five labels
    - $y = \{1, 0, 1, 1, 0\}$
  - Practical and naturally occurring datasets
  - Often performs better when label associations are considered
Introduction
Multi-label nearest neighbour

- Nearest neighbour method benefits
  - No or minimal training (lazy method)
  - Quick to update model with new data
  - Easier interpretation of prediction

- Multi-label nearest neighbour classification
  - MLkNN, BRkNN, DMLkNN, IBLR-ML, IBLR-ML+, Stacked-MLkN, etc.
  - Currently does not attain performance as good as the other types of algorithms [1]
  - Big room for improvement

Introduction
Supervised input space transformation

- Transform input space to in a way such that in the transformed space the classification task is relatively “easier”
- Nearest neighbour methods can benefit from such a transformation such that
  - A datapoint is likely to have more neighbours like itself
Introduction
Supervised input space transformation

- Neighbourhood Component Analysis (NCA) [2]
  - Finds a transformation matrix $A$ such that in the transformed space the test set nearest neighbour classification performance is maximised
  - Transformation is linear
  - Test set not known, leave-one-out performance is discontinuous
  - Optimises a smooth alternative based on stochastic neighbourhood assignment

Introduction
Supervised input space transformation

- Neighbourhood Component Analysis (NCA) [2]
  - Test set nearest neighbour performance is approximated using a stochastic neighbour selection rule
    - $x_i$ inherits class label of $x_j$ with the probability of $p_{ij}$
    - $C_i = \{ \text{index with the same class of } x_i \}$
  - Maximise $f(A)$ the probability that $x_i$ will be correctly classified

$$f(A) = \sum_i \sum_{j \in C_i} p_{ij} = \sum_i p_i$$

$$p_{ij} = \frac{\exp(-\|Ax_i - Ax_j\|^2)}{\sum_{k \neq i} \exp(\|Ax_i - Ax_k\|^2)}$$

Proposed Method
ML-NCA: Supervised linear input space transformation method for multi-label data
Proposed Method
ML-NCA

- Finds **one single transformation** with respect to all labels
  - Avoids training one transformation per label
  - Thus computationally beneficial
  - Label association considered implicitly in the transformed space
- After the transformation, multi-label based nearest neighbour algorithms can be used as usual
- **Objective**: to find **one** transformation $A$ with respect to all labels, such that in the transformed space the number of datapoints with correct label assignments are maximised under the stochastic neighbour selection rule
Method
ML-NCA

- \( p_{ij} \): Probability of point \( x_i \) selecting \( x_j \) as its neighbour under stochastic neighbour rule
- \( s_{ij} \): The similarity between the label assignments of \( x_i \) and \( x_j \)
- \( p_i \): \( x_i \) will be classified with the highest degree of label assignment
- Maximise \( f(A) \) to maximise the number of datapoints (for all \( i \)) with highest degree of label assignment
- NCA becomes a special case when \( s_{ij} \) is either 0 or 1

\[
p_{ij} = \frac{\exp(-||Ax_i - Ax_j||^2)}{\sum_{k \neq i} \exp(||Ax_i - Ax_k||^2)}, \quad p_{ii} = 0
\]

\[
s_{ij} = 1 - Jaccard(y_i, y_j)
\]

\[
p_i = \sum_{j} s_{ij} \times p_{ij}
\]

\[
f(A) = \sum_{i} \sum_{j} s_{ij} \times p_{ij} = \sum_{i} p_i
\]
Experiment

Do ML-NCA significantly improve kNN based multi-label methods?
Experiment Setup

- Eight multi-label datasets
- 5 times 2 folds cross-validation
- 12 values of k for both MLkNN and BRkNN explored
- ML-NCA trained using gradient descent method
- Comparisons to understand usefulness of ML-NCA in multi-label
  - BRkNN vs ML-NCA-BRkNN
  - MLkNN vs ML-NCA-MLkNN

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Instances</th>
<th>Inputs</th>
<th>Labels</th>
<th>Total Labelsets</th>
<th>Single Labelsets</th>
<th>Cardinality</th>
<th>Density</th>
<th>MeanIR</th>
</tr>
</thead>
<tbody>
<tr>
<td>yeast</td>
<td>2417</td>
<td>103</td>
<td>14</td>
<td>198</td>
<td>77</td>
<td>4.237</td>
<td>0.303</td>
<td>7.197</td>
</tr>
<tr>
<td>birds</td>
<td>322</td>
<td>260</td>
<td>20</td>
<td>89</td>
<td>55</td>
<td>1.503</td>
<td>0.075</td>
<td>13.004</td>
</tr>
<tr>
<td>emotions</td>
<td>593</td>
<td>72</td>
<td>6</td>
<td>27</td>
<td>4</td>
<td>1.869</td>
<td>0.311</td>
<td>1.478</td>
</tr>
<tr>
<td>cal500</td>
<td>502</td>
<td>68</td>
<td>174</td>
<td>502</td>
<td>502</td>
<td>26.044</td>
<td>0.150</td>
<td>20.578</td>
</tr>
<tr>
<td>foodtruck</td>
<td>407</td>
<td>21</td>
<td>12</td>
<td>116</td>
<td>74</td>
<td>2.290</td>
<td>0.191</td>
<td>7.094</td>
</tr>
<tr>
<td>medical</td>
<td>978</td>
<td>1449</td>
<td>45</td>
<td>94</td>
<td>33</td>
<td>1.245</td>
<td>0.028</td>
<td>89.501</td>
</tr>
<tr>
<td>PlantPseAAC</td>
<td>978</td>
<td>440</td>
<td>12</td>
<td>32</td>
<td>8</td>
<td>1.079</td>
<td>0.090</td>
<td>6.690</td>
</tr>
<tr>
<td>enron</td>
<td>1702</td>
<td>1001</td>
<td>53</td>
<td>753</td>
<td>573</td>
<td>3.378</td>
<td>0.064</td>
<td>73.953</td>
</tr>
</tbody>
</table>

Source code: https://github.com/phoxis/mlnca
Results

Do ML-NCA stage significantly improve kNN based multi-label methods? **ANALYSE**
### Results

#### F-Scores

Table 2: Label-based macro-averaged F-Scores results. Each values is the mean label-based macro-averaged F-Score (± standard deviation) for the best cross-validated k values.

<table>
<thead>
<tr>
<th></th>
<th>MLkNN</th>
<th>ML-NCA-MLkNN</th>
<th>BRkNN</th>
<th>ML-NCA-BRkNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>yeast</td>
<td>0.3674 ± 0.01</td>
<td><strong>0.3682 ± 0.01</strong></td>
<td><strong>0.3757 ± 0.01</strong></td>
<td>0.3622 ± 0.01</td>
</tr>
<tr>
<td>birds</td>
<td>0.2670 ± 0.03</td>
<td><strong>0.3050 ± 0.02</strong></td>
<td>0.2901 ± 0.03</td>
<td><strong>0.3174 ± 0.02</strong></td>
</tr>
<tr>
<td>emotions</td>
<td>0.6179 ± 0.02</td>
<td><strong>0.6394 ± 0.02</strong></td>
<td>0.6159 ± 0.02</td>
<td><strong>0.6438 ± 0.02</strong></td>
</tr>
<tr>
<td>CAL500</td>
<td>0.0604 ± 0.00</td>
<td><strong>0.0793 ± 0.00</strong></td>
<td>0.0831 ± 0.01</td>
<td><strong>0.0943 ± 0.00</strong></td>
</tr>
<tr>
<td>foodtruck</td>
<td>0.1114 ± 0.02</td>
<td><strong>0.1394 ± 0.01</strong></td>
<td>0.1373 ± 0.01</td>
<td><strong>0.1407 ± 0.01</strong></td>
</tr>
<tr>
<td>medical</td>
<td>0.2351 ± 0.01</td>
<td><strong>0.2865 ± 0.01</strong></td>
<td>0.1493 ± 0.02</td>
<td><strong>0.2583 ± 0.01</strong></td>
</tr>
<tr>
<td>PlantPseAAC</td>
<td>0.1077 ± 0.01</td>
<td><strong>0.2196 ± 0.02</strong></td>
<td>0.0701 ± 0.02</td>
<td><strong>0.1576 ± 0.02</strong></td>
</tr>
<tr>
<td>enron</td>
<td>0.0861 ± 0.01</td>
<td><strong>0.1106 ± 0.01</strong></td>
<td>0.0939 ± 0.01</td>
<td><strong>0.1309 ± 0.01</strong></td>
</tr>
</tbody>
</table>
Results
Per k-value Analysis

- ML-NCA-BRkNN vs BRkNN
- Significant improvement achieved per dataset scores for almost all k-values, except *foodtruck* and *yeast* datasets
Results

Per k-value Analysis

- ML-NCA-MLkNN vs MLkNN
- Significant improvement achieved per dataset scores for almost all k-values, except foodtruck and yeast datasets
Results

Overall

- Wilcoxon’s Signed Rank Test shows that overall, the ML-NCA stage significantly improved the results
  - ML-NCA-MLkNN vs MLkNN
  - ML-NCA-BRkNN vs BRkNN

<table>
<thead>
<tr>
<th></th>
<th>MLkNN</th>
<th>ML-NCA-MLkNN</th>
<th>BRkNN</th>
<th>ML-NCA-BRkNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLkNN</td>
<td></td>
<td>0/8/0</td>
<td>3/5/0</td>
<td>1/7/0</td>
</tr>
<tr>
<td>ML-NCA-MLkNN</td>
<td>0.0059</td>
<td></td>
<td>6/2/0</td>
<td>3/5/0</td>
</tr>
<tr>
<td>BRkNN</td>
<td>0.3897</td>
<td>0.0344 **</td>
<td></td>
<td>1/7/0</td>
</tr>
<tr>
<td>ML-NCA-BRkNN</td>
<td>0.0086</td>
<td>0.5000</td>
<td>0.0178</td>
<td>**</td>
</tr>
</tbody>
</table>

Table 3: Significance test. Upper diagonal: win/lose/tie. Lower diagonal: Wilcoxon’s Signed Rank Test p-values. Significance levels: ***: $\alpha = 0.01$, **: $\alpha = 0.05$, *: $\alpha = 0.1$. 
Conclusion and Future Work

Do ML-NCA stage significantly improve kNN based multi-label methods? **YES**
Conclusion

- ML-NCA was able to significantly improve almost each dataset for all values of $k$ except yeast and foodtruck
- ML-NCA was able to attain best results for all results selected the best value of $k$
- Wilcoxon’s Signed Rank Test shows that overall, the ML-NCA stage significantly improved the results
  - ML-NCA-MLkNN vs MLkNN
  - ML-NCA-BRkNN vs BRkNN
- ML-NCA significantly improved kNN based multi-label methods
Future Work

- Compare with other transformation methods
- Test on other kNN based methods
- Compare based on other metrics
- More datasets

- Dimensionality reduction while retaining accuracy
- Label weighting
- Consider label association implicitly
- Non-linear transformation
Contact

Arjun Pakrashi$^{1,3,4}$, Payel Sadhukhan$^2$, Brian Mac Namee$^{1,3,4}$

1 School of Computer Science, University College Dublin, Ireland
2 TCG Crest, Kolkata, India
3 VistaMilk
4 Insight Centre for Data Analytics

Email: {arjun.pakrashi,brian.macnamee}@ucd.ie
payel0410@gmail.com
Twitter: @phoxis, @BrianMacNamee
GitHub: https://github.com/phoxis/mlnca