Multi-label Classification

Jingzhou Liu
Dec. 6 2016
Introduction

• Multi-class problem,
  • Training data \{ (x_i, y_i) \}_{i=1}^n, x_i \in \mathcal{X} \subseteq \mathbb{R}^D, y_i \in \mathcal{Y} = \{1,2,\ldots,L\}
  • Learn a mapping \( f: \mathcal{X} \rightarrow \mathcal{Y} \)
  • Each instance \( x_i \) is associated with a single relevant label \( y_i \)

• Multi-label problem
  • Training data \{ (x_i, y_i) \}_{i=1}^n, x_i \in \mathcal{X} \subseteq \mathbb{R}^D, y_i \in \{0,1\}^L
  • Learn a mapping \( g: \mathcal{X} \rightarrow \{0,1\}^L \)
  • Each instance \( x_i \) is associated with a set of relevant labels, denoted by label vector \( y_i \)
Introduction

• Applications

• Text categorization
• Image/video annotation
• Query/keyword suggestions
• Recommender system
• ...

Overview

• Introduction
• **Baseline Methods**
• Key Challenges
• Embedding-based Methods
• Tree-based Methods
• LETOR method
• Conclusion
1-vs-All

- Train a binary classifier for each label
- For label $j$, construct training data $D_j = \{(x_i, y_i^j) | i = 1..n\}$
- Learn a binary classifier $g_j : \mathcal{X} \rightarrow \mathbb{R}$

- For unseen instance $x$, predict its associated label set $Y$ by running all $L$ individual binary classifiers
  - $Y = \{y_j | g_j(x) > 0\}$

- Training: $O(L \cdot F_g(n, D))$  
- Testing: $O(L \cdot H_g(D))$
1-vs-1

- Train a binary classifier for each pair of labels
- For each pair of label \((j, k)\), construct training data
  \[
  D_{jk} = \{(x_i, y_i^j - y_i^k) | y_i^j \neq y_i^k, i = 1..n\}
  \]
- Learn a binary classifier \(g_{jk}: \mathcal{X} \to \mathbb{R}\)

- For unseen instance \(x\), take the overall votes on each label:
  \[
  \varphi(x, j) = \sum_{k=1}^{j-1} I(g_{kj}(x) \leq 0) + \sum_{k=j+1}^{L} I(g_{jk}(x) > 0)
  \]
- Training: \(O(L^2 \cdot F_g(n, D))\)  
  Testing: \(O(L^2 \cdot H_g(D))\)
ML-kNN

• For unseen instance $x$, let $\mathcal{N}(x)$ denote its $k$ nearest neighbors
• And calculate $C_j = \sum_{x_i \in \mathcal{N}(x)} I(y_i^j = 1)$

• Predicted label set $Y = \{j | \frac{P(H_j|C_j)}{P(\neg H_j|C_j)} > 1\}$
• where $H_j$ is the event that $x$ has label $j$

• Training: $O(n^2D + Ln k)$    Testing: $O(nD + L k)$
Decision Tree

• Starting from the root, identify the feature and the corresponding splitting value which maximizes the information gain.

\[ IG(T, l, v) = H(T) - \frac{|T^+|}{|T|} H(T^+) - \frac{|T^-|}{|T|} H(T^-) \]

Where \( T^+ = \{(x_i, y_i)| x_i^l > v \}, T^- = \{(x_i, y_i)| x_i^l \leq v \} \)

• For unseen instance \( x \), traverse the tree until reaching a leaf node, and predict labels that are in the majority of instances in the leaf.

• Training: \( O(nDLh) \)  

Testing: \( O(h + L_{leaf}n_{leaf}) \)
Overview

• Introduction
• Baseline Methods
• **Key Challenges**
  • Embedding-based Methods
  • Tree-based Methods
  • LETOR method
• Conclusion
Key Challenges

• Consider the output space: $2^L!$

• If treat labels separately: we lose correlations among labels

• 1-vs-All, ML-kNN, Decision Tree: no correlations

• 1-vs-1: pair-wise correlation

• It is crucial to exploit label correlations information.
## Key Challenges

- In practice, $n, L$ is large, and $D$ is usually large but sparse.

<table>
<thead>
<tr>
<th>Method</th>
<th>Train</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-vs-All</td>
<td>$O(L \cdot F_g(n, D))$</td>
<td>$O(L \cdot H_g(D))$</td>
</tr>
<tr>
<td>1-vs-1</td>
<td>$O(L^2 \cdot F_g(n, D))$</td>
<td>$O(L^2 \cdot H_g(D))$</td>
</tr>
<tr>
<td>ML-kNN</td>
<td>$O(n^2D + Lnk)$</td>
<td>$O(nD + Lk)$</td>
</tr>
<tr>
<td>Decision Tree</td>
<td>$O(nDLh)$</td>
<td>$O(h + L_{leaf}n_{leaf})$</td>
</tr>
</tbody>
</table>

- $n, L \sim$ million?
- We need testing time to be sublinear to $L$
Overview

• Introduction
• Baseline Methods
• Key Challenges
• Embedding-based Methods
• Tree-based Methods
• LETOR method
• Conclusion
Embedding-based Methods

• Reduce label and/or feature dimensions.

• Given a training example \((x_i, y_i)\), compress \(y_i\) to a \(\hat{L}\)-dimensional space
  \[ z_i = U y_i, U \in \mathbb{R}^{\hat{L} \times L} \text{ (e.g. SVD)} \]

• Learn \(V \in \mathbb{R}^{\hat{L} \times D}\) s.t. \(V x_i \approx z_i = U y_i\)

• For unseen instance \(x\), recover label vector \(y = U^*(V x)\)
• Where \(U^*(\cdot)\) is a decompression function
  (e.g. matrix multiplication, or involves some learning)
Low rank Empirical risk minimization for Multi-label Learning (LEML)  Yu et al. ICML’2014

• Directly optimize for $y = U^*(Vx)$

• Prediction is parameterized as
  $$f(x; Z) = Z^T x, \text{ where } Z \in \mathbb{R}^{D \times L}$$

• And loss function is decomposable
  $$\ell(y, f(x; Z)) = \sum_j \ell(y^j, f^j(x; Z))$$
To exploit label correlations, assume $Z$ is controlled by only a small number of latent factors

$$\hat{Z} = \arg \min_Z J(Z) = \sum_{i=1}^{n} \sum_{j=1}^{L} \ell(y_i^j, f^j(x_i; Z)) + \lambda \cdot r(Z)$$

s. t. $\text{rank}(Z) \leq k$
Low rank Empirical risk minimization for Multi-label Learning (LEML) Yu et al. ICML’2014

- Decompose $Z = WH^T$, where $W \in \mathbb{R}^{D \times k}, H \in \mathbb{R}^{L \times k}$
- $r(Z) = r_1(W) + r_2(H)$

$$
\arg \min_{W,H} J(W,H) = \sum_{i=1}^{n} \sum_{j=1}^{L} \ell(y_i^j, x_i^T W h_j) + \frac{\lambda}{2} (\|W\|_{F}^2 + \|H\|_{F}^2)
$$

$$
H^{(t)} \leftarrow \min_{H} J(W^{(t-1)}, H)
$$

$$
W^{(t)} \leftarrow \min_{W} J(W, H^{(t)})
$$
Low rank Empirical risk minimization for Multi-label Learning (LEML)  Yu et al. ICML’2014
Low rank Empirical risk minimization for Multi-label Learning (LEML)  
Yu et al. ICML’2014

<table>
<thead>
<tr>
<th>dataset</th>
<th>$k$</th>
<th>time (s)</th>
<th>LEML top-1</th>
<th>LEML top-3</th>
<th>LEML AUC</th>
<th>time (s)</th>
<th>WSABIE top-1</th>
<th>WSABIE top-3</th>
<th>WSABIE AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>eurlex</td>
<td>250</td>
<td>175</td>
<td>51.99</td>
<td>39.79</td>
<td>0.9425</td>
<td>373</td>
<td>33.13</td>
<td>25.01</td>
<td>0.8648</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>487</td>
<td>56.90</td>
<td>44.20</td>
<td>0.9456</td>
<td>777</td>
<td>31.58</td>
<td>24.00</td>
<td>0.8651</td>
</tr>
<tr>
<td>nus-wide</td>
<td>50</td>
<td>574</td>
<td>20.71</td>
<td>15.96</td>
<td>0.7741</td>
<td>4,705</td>
<td>14.58</td>
<td>11.37</td>
<td>0.7658</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>1,097</td>
<td>20.76</td>
<td>16.00</td>
<td>0.7718</td>
<td>6,880</td>
<td>12.46</td>
<td>10.21</td>
<td>0.7597</td>
</tr>
<tr>
<td>wiki</td>
<td>250</td>
<td>9,932</td>
<td>19.56</td>
<td>14.43</td>
<td>0.9086</td>
<td>79,086</td>
<td>18.91</td>
<td><strong>14.65</strong></td>
<td>0.9020</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>18,072</td>
<td>22.83</td>
<td>17.30</td>
<td>0.9374</td>
<td>139,290</td>
<td>19.20</td>
<td>15.66</td>
<td>0.9058</td>
</tr>
</tbody>
</table>
Low rank Empirical risk minimization for Multi-label Learning (LEML)  Yu et al. ICML’2014

• Exploit label correlations through a low rank assumption
• A general ERM framework
• Fairly scalable

• Prediction $f(x; Z) = HW^T x$
• Still $O(k \cdot (L + D))$ complexity
• Lose information during compression/decompression
Sparse Local Embedding for Extreme Classification (SLEEC)  Bhatia et al. NIPS’2015

• Project feature and label vectors to the same $\hat{L}$-dimensional space

$$x_i \rightarrow Vx_i, \ V \in \mathbb{R}^{\hat{L} \times D}$$

$$y_i \rightarrow zi, \ z_i \in \mathbb{R}^{\hat{L}}$$

s.t. \ $Vx_i \approx z_i$

$z_i$ preserves local properties of $y_i$

• At test phase, for unseen instance $x$, perform kNN search in the $\hat{L}$-dimensional space with $Vx$

• Can speed up testing by clustering training examples
Sparse Local Embedding for Extreme Classification (SLEEC)  Bhatia et al. NIPS’2015

• Instead of using a global projection that tries to preserve distances between all pairs of label vectors, only preserve those between nearest neighbors

• \( \Omega \) denotes the set of nearest neighbor pairs

\[
(i, j) \in \Omega \iff j \in \mathcal{N}(i)
\]

\[
\min_{Z \in \mathbb{R}^L \times n} \| P_{\Omega}(Y^TY) - P_{\Omega}(Z^T Z) \|_F^2 + \lambda \| Z \|_1
\]

\[
\min_{V \in \mathbb{R}^{L \times D}} \| P_{\Omega}(Y^TY) - P_{\Omega}(X^T V^TVX) \|_F^2 + \lambda \| V \|_F^2 + \mu \| VX \|_1
\]
Sparse Local Embedding for Extreme Classification (SLEEC)  Bhatia et al. NIPS’2015

- Divide the optimization into two phases

First

$$\min_{Z \in \mathbb{R}^{\hat{L} \times n}} \| P_\Omega (Y^T Y) - P_\Omega (Z^T Z) \|_F^2 = \min_{\substack{M \succeq 0 \\ \text{rank}(M) \leq \hat{L}}} \| P_\Omega (Y^T Y) - P_\Omega (M) \|_F^2$$

Then

$$\min_{V \in \mathbb{R}^{\hat{L} \times D}} \| Z - VX \|_F^2 + \lambda \| V \|_F^2 + \mu \| VX \|_1$$
Sparse Local Embedding for Extreme Classification (SLEEC)  Bhatia et al. NIPS’2015

<table>
<thead>
<tr>
<th>Data set</th>
<th>SLEEC</th>
<th>LEML</th>
<th>FastXML</th>
<th>LPSR-NB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wiki10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P@1</td>
<td>85.54</td>
<td>73.50</td>
<td>82.56</td>
<td>72.71</td>
</tr>
<tr>
<td>P@3</td>
<td>73.59</td>
<td>62.38</td>
<td>66.67</td>
<td>58.51</td>
</tr>
<tr>
<td>P@5</td>
<td>63.10</td>
<td>54.30</td>
<td>56.70</td>
<td>49.40</td>
</tr>
<tr>
<td>Delicious-Large</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P@1</td>
<td>47.03</td>
<td>40.30</td>
<td>42.81</td>
<td>18.59</td>
</tr>
<tr>
<td>P@3</td>
<td>41.67</td>
<td>37.76</td>
<td>38.76</td>
<td>15.43</td>
</tr>
<tr>
<td>P@5</td>
<td>38.88</td>
<td>36.66</td>
<td>36.34</td>
<td>14.07</td>
</tr>
<tr>
<td>WikiLSHTC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P@1</td>
<td>55.57</td>
<td>19.82</td>
<td>49.35</td>
<td>27.43</td>
</tr>
<tr>
<td>P@3</td>
<td>33.84</td>
<td>11.43</td>
<td>32.69</td>
<td>16.38</td>
</tr>
<tr>
<td>P@5</td>
<td>24.07</td>
<td>8.39</td>
<td>24.03</td>
<td>12.01</td>
</tr>
<tr>
<td>Amazon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P@1</td>
<td>35.05</td>
<td>8.13</td>
<td>33.36</td>
<td>28.65</td>
</tr>
<tr>
<td>P@3</td>
<td>31.25</td>
<td>6.83</td>
<td>29.30</td>
<td>24.88</td>
</tr>
<tr>
<td>P@5</td>
<td>28.56</td>
<td>6.03</td>
<td>26.12</td>
<td>22.37</td>
</tr>
<tr>
<td>Ads-1m</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P@1</td>
<td>21.84</td>
<td>-</td>
<td>23.11</td>
<td>17.08</td>
</tr>
<tr>
<td>P@3</td>
<td>14.30</td>
<td>-</td>
<td>13.86</td>
<td>11.38</td>
</tr>
<tr>
<td>P@5</td>
<td>11.01</td>
<td>-</td>
<td>10.12</td>
<td>8.83</td>
</tr>
</tbody>
</table>
Sparse Local Embedding for Extreme Classification (SLEEC)  Bhatia et al. NIPS’2015
Sparse Local Embedding for Extreme Classification (SLEEC)  Bhatia et al. NIPS’2015

• Preserve only local distances of label vectors, through nonlinear mapping

• Use kNN classifier and clustering on prediction, avoid a decompression step
Overview

• Introduction
• Baseline Methods
• Key Challenges
• Embedding-based Methods
• **Tree-based Methods**
• LETOR method
• Conclusion
Tree-based Methods

• Learn a hierarchy over the feature space, and assign a subset of labels to each leaf node.

• At test phase, an unseen instance traverses from root to its corresponding leaf, and prediction is restricted to labels in this leaf.

• Enjoy efficiency in both training and testing when the tree is balanced, and the number of instances in a leaf is relatively small.
Multi-Label Random Forests (MLRF) Agrawal et al. WWW’2013

• Learn an ensemble of decision trees

• Minimize Gini index over a set of randomly selected features

• At test time, label distributions are aggregated over the leaf nodes in each tree
Label Partitioning for Sublinear Ranking (LPSR)
Weston et al. ICML’2013

• Try to address slow prediction problem
• Suppose already trained a base scorer \( f(x, y) \) for an instance \( x \) and a single label \( y \), \( f(x) = [f(x, 1), \ldots, f(x, L)]^T \)

• Two components
• Input Partitioner
• Label Assignment
Label Partitioning for Sublinear Ranking (LPSR)
Weston et al. ICML’2013

• Input Partitioner
  • Over feature space
  • Examples for which the base scorer performs well should be prioritized
  • Hierarchical k-means

\[
\min_{l_i} \sum_{i=1}^{n} P(f(x_i), y_i) \|x_i - c_{l_i}\|^2
\]

where \(l_i\) is cluster assignment of \(x_i\), \(c_i\) is the centroid of the \(i\)th cluster
\(P(\cdot, \cdot)\) is a metric to maximize (e.g. precision@k)
Label Partitioning for Sublinear Ranking (LPSR)
Weston et al. ICML’2013

• Label Assignment

• Assign a set of labels $\alpha \in \{0,1\}^L$ to each leaf s. t. precision in each leaf is maximized through a relaxed integer programming

• Directly optimize the measurement of interest

• At test time, rank the labels assigned to the leaf according to base scorer

• Base scorer $O(nLF_f(n,D))$

• Test $O(L_{leaf}H_f(D))$
Fast eXtreme Multi-label Learning (FastXML)
Prabhu & Varma KDD’2014

• Directly optimize the measurement of interest (nDCG)

• Jointly learn input partitioner (hyperplane) and label assignment (a ranking of labels)

• Learn an ensemble of trees instead of base scorers
  • Aggregate label distributions over all leaf nodes
Fast eXtreme Multi-label Learning (FastXML)
Prabhu&Varma KDD’2014

\[
\min_{w, \delta, r^+, r^-} \|w\|_1 + \sum_{i=1}^{n} C_{\delta}(\delta_i) \log(1 + e^{-\delta_i w^T x_i}) \\
- C_r \sum_{i=1}^{n} \frac{1}{2} (1 + \delta_i) \mathcal{L}_{nDCG}(r^+, y_i) \\
- C_r \sum_{i=1}^{n} \frac{1}{2} (1 - \delta_i) \mathcal{L}_{nDCG}(r^-, y_i)
\]

\(w \in \mathbb{R}^D\) is the hyperplane cutting the current feature space
\(\delta \in \{+1, -1\}^n\) indicating which half each instance is assigned to
\(r^+, r^-\) are rankings of labels assigned to each half
Fast eXtreme Multi-label Learning (FastXML)
Prabhu&Varma KDD’2014

\[
\min_{\mathbf{w}, \delta, r^+, r^-} \|\mathbf{w}\|_1 + \sum_{i=1}^{n} C_\delta(\delta_i) \log(1 + e^{-\delta_i \mathbf{w}^T \mathbf{x}_i}) \\
- C_r \sum_{i=1}^{n} \frac{1}{2} (1 + \delta_i) L_{nDCG}(r^+, \mathbf{y}_i) \\
- C_r \sum_{i=1}^{n} \frac{1}{2} (1 - \delta_i) L_{nDCG}(r^-, \mathbf{y}_i)
\]

If \(\mathbf{w}\) and \(\delta\) is fixed, optimize for \(r^+\) and \(r^-\): \(O(n\hat{L} + \hat{L} \log \hat{L})\)
If \(\mathbf{w}\) and \(r^\pm\) is fixed, optimize for \(\delta\): \(O(n\hat{L})\)
Fast eXtreme Multi-label Learning (FastXML)
Prabhu&Varma KDD’2014

\[
\min_{\mathbf{w}, \delta, r^+, r^-} \|\mathbf{w}\|_1 + \sum_{i=1}^{n} C_\delta(\delta_i) \log(1 + e^{-\delta_i \mathbf{w}^T \mathbf{x}_i})
\]

\[
- C_r \sum_{i=1}^{n} \frac{1}{2} (1 + \delta_i) \mathcal{L}_{nDCG}(r^+, y_i)
\]

\[
- C_r \sum_{i=1}^{n} \frac{1}{2} (1 - \delta_i) \mathcal{L}_{nDCG}(r^-, y_i)
\]

If \( r^\pm \) and \( \delta \) is fixed, optimize for \( \mathbf{w} \): \( \ell_1 \) regularized logistic regression

- The most costly step

- Fix \( \mathbf{w} \) and alternately update \( \delta \) and \( r^\pm \), perform update on \( \mathbf{w} \) only when updating \( \delta \) and \( r^\pm \) does not lead to decrease in objective
Fast eXtreme Multi-label Learning (FastXML)
Prabhu & Varma KDD’2014

- **BibTeX**
  \[ N = 4.8K, D = 1.8K, L = 159 \]

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>P1 (%)</th>
<th>P3 (%)</th>
<th>P5 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FastXML-T</td>
<td><strong>64.53 ± 0.72</strong></td>
<td><strong>40.17 ± 0.63</strong></td>
<td><strong>29.27 ± 0.53</strong></td>
</tr>
<tr>
<td>FastXML</td>
<td>63.26 ± 0.84</td>
<td>39.19 ± 0.66</td>
<td>28.72 ± 0.48</td>
</tr>
<tr>
<td>MLRF</td>
<td>62.81 ± 0.84</td>
<td>38.74 ± 0.69</td>
<td>28.45 ± 0.43</td>
</tr>
<tr>
<td>LPSR</td>
<td>62.95 ± 0.70</td>
<td>39.16 ± 0.64</td>
<td>28.75 ± 0.45</td>
</tr>
<tr>
<td>1-vs-All</td>
<td>63.39 ± 0.64</td>
<td>39.55 ± 0.65</td>
<td>29.13 ± 0.45</td>
</tr>
</tbody>
</table>

- **Delicious**
  \[ N = 13K, D = 500K, L = 983 \]

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>P1 (%)</th>
<th>P3 (%)</th>
<th>P5 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FastXML</td>
<td><strong>69.65 ± 0.82</strong></td>
<td><strong>63.93 ± 0.50</strong></td>
<td><strong>59.36 ± 0.57</strong></td>
</tr>
<tr>
<td>MLRF</td>
<td>67.86 ± 0.70</td>
<td>62.02 ± 0.55</td>
<td>57.59 ± 0.43</td>
</tr>
<tr>
<td>LPSR</td>
<td>65.55 ± 0.99</td>
<td>59.39 ± 0.48</td>
<td>53.99 ± 0.31</td>
</tr>
<tr>
<td>1-vs-All</td>
<td>65.42 ± 1.05</td>
<td>59.34 ± 0.56</td>
<td>53.72 ± 0.50</td>
</tr>
</tbody>
</table>

- **MediaMill**
  \[ N = 30K, D = 120, L = 101 \]

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>P1 (%)</th>
<th>P3 (%)</th>
<th>P5 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FastXML</td>
<td><strong>87.35 ± 0.27</strong></td>
<td><strong>72.14 ± 0.20</strong></td>
<td><strong>58.15 ± 0.15</strong></td>
</tr>
<tr>
<td>MLRF</td>
<td>86.83 ± 0.18</td>
<td>71.18 ± 0.19</td>
<td>57.09 ± 0.16</td>
</tr>
<tr>
<td>LPSR</td>
<td>82.33 ± 2.15</td>
<td>66.37 ± 0.35</td>
<td>50.00 ± 0.20</td>
</tr>
<tr>
<td>1-vs-All</td>
<td>82.31 ± 2.19</td>
<td>66.17 ± 0.43</td>
<td>50.32 ± 0.56</td>
</tr>
</tbody>
</table>

- **RCV1-X**
  \[ N = 781K, D = 47K, L = 2.5K \]

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>P1 (%)</th>
<th>P3 (%)</th>
<th>P5 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FastXML</td>
<td><strong>91.23 ± 0.22</strong></td>
<td><strong>73.51 ± 0.25</strong></td>
<td><strong>53.31 ± 0.65</strong></td>
</tr>
<tr>
<td>MLRF</td>
<td>87.66 ± 0.46</td>
<td>69.89 ± 0.43</td>
<td>50.36 ± 0.74</td>
</tr>
<tr>
<td>LPSR</td>
<td>90.04 ± 0.19</td>
<td>72.27 ± 0.20</td>
<td>52.34 ± 0.61</td>
</tr>
<tr>
<td>1-vs-All</td>
<td>90.18 ± 0.18</td>
<td>72.55 ± 0.16</td>
<td>52.68 ± 0.57</td>
</tr>
</tbody>
</table>
Fast eXtreme Multi-label Learning (FastXML)
Prabhu&Varma KDD’2014

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>P1 (%)</th>
<th>P3 (%)</th>
<th>P5 (%)</th>
<th>Train Time (hr)</th>
<th>Test Time (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FastXML</td>
<td>49.78</td>
<td>33.06</td>
<td>24.40</td>
<td>9.14</td>
<td>5.10</td>
</tr>
<tr>
<td>LPSR-NB</td>
<td>27.91</td>
<td>16.04</td>
<td>11.57</td>
<td>1.59</td>
<td>3.52</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>P1 (%)</th>
<th>P3 (%)</th>
<th>P5 (%)</th>
<th>Train Time (hr)</th>
<th>Test Time (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FastXML</td>
<td>27.24</td>
<td>16.28</td>
<td>11.91</td>
<td>1.81</td>
<td>1.68</td>
</tr>
<tr>
<td>LPSR-NB</td>
<td>19.69</td>
<td>12.71</td>
<td>9.70</td>
<td>0.84</td>
<td>3.95</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>P1 (%)</th>
<th>P3 (%)</th>
<th>P5 (%)</th>
<th>Train Time (hr)</th>
<th>Test Time (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FastXML</td>
<td>23.45</td>
<td>14.21</td>
<td>10.41</td>
<td>8.09</td>
<td>6.26</td>
</tr>
<tr>
<td>LPSR-NB</td>
<td>17.08</td>
<td>11.38</td>
<td>8.83</td>
<td>3.78</td>
<td>19.32</td>
</tr>
</tbody>
</table>
Fast eXtreme Multi-label Learning (FastXML)
Prabhu & Varma KDD’2014

• Exploit label correlations by directly optimize rank-sensitive loss function (nDCG)

• Linear separator to partition each node in the tree

• Jointly learn the partition and label assignment

• Sublinear to $L$ at prediction
Embedding-based vs Tree-based

• Embedding-based
  • Simple
  • Strong theoretical foundations
  • Easy to handle label correlations
  • Lose information during compression/decompression
  • Slow at prediction

• Tree-based
  • Efficient
  • Large model size
  • Not theoretically well supported
Overview

• Introduction
• Baseline Methods
• Key Challenges
• Embedding-based Methods
• Tree-based Methods
• LETOR Method
• Conclusion
Multilabel classification with meta-level features in a learning-to-rank framework  Yang & Gopal Mach Learn’2012

• The methods discussed above, have been focusing on low level features that do not characterize instance-label relationships

• Low level features may not be expressive enough for learning instance-label mapping

• Construct meta-level features based on both the original instance feature and labeled instances, and reformulate the problem as a standard learning-to-rank problem
Multilabel classification with meta-level features in a learning-to-rank framework  Yang & Gopal Mach Learn’2012

• Meta-level feature $\phi(x, y), x \in \mathbb{R}^D, y \in \{1, ..., L\}$
  • $\phi(x, y)$ should carry information about the relation of $x$ to $y$, with discriminative power
  • Obtain $\phi(x, y)$ directly from a given instance and a set of labeled instances

\[ \phi_{L_2}(x, y) = (d_{L_2}(x, \hat{x}_1), ..., d_{L_2}(x, \hat{x}_k)) \]

where $\hat{x}_1, ..., \hat{x}_k$ are the $k$ nearest neighbors to $x$ in $L_2$ distance

• Similarly, we can construct $\phi_{L_1}(x, y), \phi_{cos}(x, y)$
Multilabel classification with meta-level features in a learning-to-rank framework  
Yang & Gopal  
Mach Learn’2012

- Also include distance to centroids
  \[ \phi_{cen}(x, y) = (d_{L_2}(x, \bar{x}_y), d_{cos}(x, \bar{x}_y)) \]
  where \( \bar{x}_y \) is the centroid of all instances with label \( y \)

- Define \( \phi(x, y) = [\phi_{L_2}(x, y), \phi_{L_1}(x, y), \phi_{cos}(x, y), \phi_{cen}(x, y)] \)

- Plug in any learning-to-rank algorithm
  - SVM-Rank, SVM-MAP, ListNet, etc.
Multilabel classification with meta-level features in a learning-to-rank framework  Yang & Gopal Mach Learn’2012

<table>
<thead>
<tr>
<th></th>
<th>MLC-ListNet</th>
<th>Binary-SVM</th>
<th>RankSVM</th>
<th>IBLR</th>
<th>ML-kNN</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Micro-$F_1$</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scene</td>
<td><strong>0.74210</strong> (1)</td>
<td>0.66502** (4)</td>
<td>0.64072** (5)</td>
<td>0.71138** (2)</td>
<td>0.69864** (3)</td>
</tr>
<tr>
<td>Emotions</td>
<td><strong>0.72960</strong> (1)</td>
<td>0.64591** (3)</td>
<td>0.62162** (5)</td>
<td>0.69262** (2)</td>
<td>0.63900** (4)</td>
</tr>
<tr>
<td>Yeast</td>
<td><strong>0.67633</strong> (1)</td>
<td>0.63132** (5)</td>
<td>0.65620** (2)</td>
<td>0.63714** (3)</td>
<td>0.63681** (4)</td>
</tr>
<tr>
<td>Citeseer</td>
<td><strong>0.61711</strong> (1)</td>
<td>0.53448** (3)</td>
<td>0.56677** (2)</td>
<td>0.46148** (5)</td>
<td>0.52678** (4)</td>
</tr>
<tr>
<td>Reuters-21578</td>
<td>0.81658 (2)</td>
<td><strong>0.87084</strong> (1)</td>
<td>0.80416* (4)</td>
<td>0.72808** (5)</td>
<td>0.80484 (3)</td>
</tr>
<tr>
<td><strong>Macro-$F_1$</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scene</td>
<td><strong>0.75746</strong> (1)</td>
<td>0.66975* (5)</td>
<td>0.67308* (4)</td>
<td>0.71452 (2)</td>
<td>0.69162* (3)</td>
</tr>
<tr>
<td>Emotions</td>
<td><strong>0.72188</strong> (1)</td>
<td>0.66000* (3)</td>
<td>0.61644* (4)</td>
<td>0.68103* (2)</td>
<td>0.61392* (5)</td>
</tr>
<tr>
<td>Yeast</td>
<td><strong>0.46425</strong> (1)</td>
<td>0.32405** (5)</td>
<td>0.36645** (3)</td>
<td>0.37093** (2)</td>
<td>0.33614** (4)</td>
</tr>
<tr>
<td>Citeseer</td>
<td><strong>0.61326</strong> (1)</td>
<td>0.53523** (3)</td>
<td>0.56477** (2)</td>
<td>0.44948** (5)</td>
<td>0.50330** (4)</td>
</tr>
<tr>
<td>Reuters-21578</td>
<td><strong>0.56074</strong> (1)</td>
<td>0.52763** (2)</td>
<td>0.40487** (3)</td>
<td>0.31100** (5)</td>
<td>0.37045** (4)</td>
</tr>
<tr>
<td><strong>MAP</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scene</td>
<td><strong>0.87586</strong> (1)</td>
<td>0.85679** (4)</td>
<td>0.85962** (3)</td>
<td>0.85803** (3)</td>
<td>0.85118** (5)</td>
</tr>
<tr>
<td>Emotions</td>
<td><strong>0.82357</strong> (1)</td>
<td>0.76850** (5)</td>
<td>0.80223** (3)</td>
<td>0.81478 (2)</td>
<td>0.78969** (4)</td>
</tr>
<tr>
<td>Yeast</td>
<td><strong>0.76654</strong> (1)</td>
<td>0.74659** (5)</td>
<td>0.75678** (4)</td>
<td>0.75999** (2)</td>
<td>0.75846** (3)</td>
</tr>
<tr>
<td>Citeseer</td>
<td><strong>0.76948</strong> (1)</td>
<td>0.73615** (3)</td>
<td>0.75086** (2)</td>
<td>0.69643** (5)</td>
<td>0.73295** (4)</td>
</tr>
<tr>
<td>Reuters-21578</td>
<td>0.92375 (4)</td>
<td><strong>0.95432</strong> (1)</td>
<td>0.93332 (2)</td>
<td>0.85415 (5)**</td>
<td>0.92492 (3)</td>
</tr>
</tbody>
</table>
Multilabel classification with meta-level features in a learning-to-rank framework Yang & Gopal Mach Learn’2012

- Training: $nL$ pairs of meta-level features
- Testing: Need to construct features with all $L$ labels

- Restrict the label space to $\hat{L}$ candidates generated from other systems
- And re-rank these candidates
Overview

• Introduction
• Baseline Methods
• Key Challenges
• Embedding-based Methods
• Tree-based Methods
• LETOR method

• Conclusion
Conclusion

• Multi-label classification is fundamentally different from multi-class classification

• It is crucial for the success of multi-label classification methods to exploit label correlations

• Most baseline methods do not scale to large problems in practice

• Embedding-based and tree-based methods try to tackle these challenges from different angles
  • SLEEC and FastXML
Label Partitioning for Sublinear Ranking (LPSR)
Weston et al. ICML’2013

\[
\max_{\alpha} \sum_i \frac{1}{|y_i|} \sum_{y \in y_i} \alpha_y (1 - \Phi \left( \sum_{R_{i,j} < R_{i,y}} \alpha_j \right))
\]

\[s.t. \alpha \in [0,1]^L\]

• \(\Phi(r) = \frac{1}{1+e^{k-r}}\)

• \(R_{i,j}\) is the rank of label \(j\) for instance \(i\), according to base scorer

• back