Introduction: Multi-label Classification

Multi-label Classification is the supervised learning problem where an instance is associated with multiple classes, rather than with a single class, as in traditional binary or multi-class problems.

Task: learn, from training data \(D = ([x^{(1)}, y^{(1)}], \ldots, [x^{(n)}, y^{(n)}])\), a function:
\[
h : \mathcal{X} \rightarrow \mathcal{Y}
\]

Mapping input in \(\mathcal{X} = \mathbb{R}^D\) to some output in \(\mathcal{Y} = \{0, 1\}^L\); where \(x^{(i)} = [x_1, \ldots, x_D] \in \mathcal{X}\) is some data instance, and \(y^{(i)} = [y_{1}, \ldots, y_{L}] \in \mathcal{Y}\) is some label vector, where \(y_j = \text{1 if the } j\text{-th label is relevant to this } i\text{-th example (else } y_j = 0\text{)}\), e.g.,

\[
x = 1 0 1 0 0 0 \\
y = 1 0 1 0 0 0
\]

For a new test instance \(\tilde{x}\), we obtain \(\tilde{y} = h(\tilde{x})\).

Chain Classifiers

\[
p(y|x) = \prod_{j=1}^{L} p(y_j|x, y_1, \ldots, y_{j-1})
\]

Model label dependencies with:

\[
\hat{y}_j = \text{arg max}_{y_j} p(y_j|x, \hat{y}_1, \ldots, \hat{y}_{j-1}) \quad (1)
\]

- The Classifier Chain (CC) [3] is a greedy approximation:
  \[
  \hat{y}_j = h_j(x, \hat{y}_1, \ldots, \hat{y}_{j-1}) \equiv \text{arg max}_{y_j} p(y_j|x, \hat{y}_1, \ldots, \hat{y}_{j-1}) \quad (2)
  \]
  for each \(j = 1, \ldots, L\). May propagate errors along the chain.

- Probabilistic Classifier Chains (PCC) [1] tests all \(2^L\) possible \(y\) on (1):
  \[
  \hat{y} = \text{arg max}_{y} p(y|x) \quad (3)
  \]
  This is intractable (for \(L > 15\)), and also ignores chain order.

- Ensembles of CC (ECC) [3] averages results of 10 CCs each with random chain orders.

Example: Classifier Chains Prediction

\[
\text{Chain order? } [Y_1, Y_2, Y_3], [Y_2, Y_3, Y_1], \text{ etc.}
\]

Alternative Approach

- Conditional Dependency Networks (CDN) [2] fully connected network (among \(Y_1, \ldots, Y_L\)) instead of a chain.

Inference with Gibbs sampling (over \(T\) iterations):
\[
y_j \sim p(y_j|x, y_1, \ldots, y_{j-1}, y_{j+1}, \ldots, y_L) \quad (4)
\]

Monte Carlo Optimization for Classifier Chains (MCC)

We present:

\(\text{MCC}\) Monte Carlo search to find good \(\hat{y} = \text{max } p(y|x)\) (inference time)

\(\text{MCC}\) + find a good chain order \(\tilde{y}\) at training time where \(s\) parameterizes some order of the labels \(1, \ldots, L\) w.r.t. \(y\)

Training: Find good \(s\), build \(p(y|x, s)\)

Inference: Find a good \(\hat{y} = \text{max } p(y|x, s)\)

Results

Table: Other Applications / Datasets

<table>
<thead>
<tr>
<th>(\mathcal{X})</th>
<th>(\mathcal{Y})</th>
<th>(L)</th>
<th>(N)</th>
<th>(D)</th>
<th>(\sum \hat{y})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Music</td>
<td>audio data</td>
<td>6</td>
<td>593</td>
<td>72</td>
<td>1.87</td>
</tr>
<tr>
<td>Scene</td>
<td>image data</td>
<td>6</td>
<td>2407</td>
<td>294</td>
<td>1.07</td>
</tr>
<tr>
<td>Yeast</td>
<td>genes</td>
<td>14</td>
<td>2417</td>
<td>103</td>
<td>4.24</td>
</tr>
<tr>
<td>Genbase</td>
<td>biological</td>
<td>27</td>
<td>661</td>
<td>1185</td>
<td>1.25</td>
</tr>
<tr>
<td>Medical</td>
<td>medical text</td>
<td>45</td>
<td>978</td>
<td>1449</td>
<td>1.25</td>
</tr>
<tr>
<td>Enron</td>
<td>e-mails</td>
<td>53</td>
<td>1702</td>
<td>1001</td>
<td>3.38</td>
</tr>
<tr>
<td>Reuters</td>
<td>news articles</td>
<td>103</td>
<td>6000</td>
<td>500</td>
<td>1.48</td>
</tr>
</tbody>
</table>

Binary Relevance (3ax)

- The Binary Relevance method (3ax) builds one binary classifier for each label

\[
f_j = h_j(\tilde{x})
\]

A natural approach to multi-label classification, use any off-the-shelf binary classifier. However, does not model label dependencies;
\[
p(y|x) \neq \prod_{j=1}^{L} p(y_j|x)
\]

Key References


Source code available in Meka framework: http://meka.sourceforge.net