Scalable Multi-label Classification

Jesse Read
Supervised by Bernhard Pfahringer and Geoff Holmes
Machine Learning Group
University of Waikato
New Zealand

September 8, 2010
Supervised Classification: given *training data* (consisting of examples of input instances associated with an output e.g. labels) train a *classifier* that can predict output *automatically* for new instances.
Supervised Classification: given *training data* (consisting of examples of input instances associated with an output e.g. labels) train a *classifier* that can predict output *automatically* for new instances.

Multi-class (Single-label) Classification: predict a class label; e.g. $\in \{\text{Beach, Forest, Urban, Sunset}\}$
Supervised Classification: given training data (consisting of examples of input instances associated with an output e.g. labels) train a classifier that can predict output automatically for new instances.

Multi-class (Single-label) Classification: predict a class label; e.g. $\in \{\text{Beach, Forest, Urban, Sunset}\}$

Multi-label Classification: predict (potentially multiple) labels e.g. $\subseteq \{\text{Beach, Forest, Urban, Sunset}\}$
Supervised Classification: given *training data* (consisting of examples of input instances associated with an output e.g. labels) train a *classifier* that can predict output *automatically* for new instances.

- **Multi-class (Single-label) Classification:** predict a class label; e.g. $\in \{\text{Beach, Forest, Urban, Sunset}\}$
- **Multi-label Classification:** predict (potentially multiple) labels e.g. $\subseteq \{\text{Beach, Forest, Urban, Sunset}\}$

Multi-label classification is the supervised classification task where each data instance may be associated with *multiple* class labels.
Notation

- Input space: $\mathcal{X} = \mathbb{R}^d$
- Instance $\mathbf{x} = [x_1, \ldots, x_d]$
- Output space: $\mathcal{Y} = \{0, 1\}^L$
- Labels: $\mathbf{y} = [y_1, \ldots, y_L]$ where $y_j = 1$ if $j$th label relevant to $\mathbf{x}$ (else 0)
- Training examples: $\{(\mathbf{x}_i, \mathbf{y}_i) | i = 1, \ldots, N\} \subset (\mathcal{X} \times \mathcal{Y})$
- Classification: $\mathcal{X} \rightarrow \mathcal{Y}$
- Prediction: $\hat{\mathbf{y}} = \mathbf{h}(\mathbf{x})$
- Evaluation:
  - $\hat{y}_i = y_i$ ?
  - $\hat{y}_{ij} = y_{ij}$ ?
Multi-label classification is relevant to many domains:

- **Text**
  - text documents $\rightarrow$ subject categories
  - e-mails $\rightarrow$ labels
  - medical description of symptoms $\rightarrow$ diagnoses

- **Vision**
  - images/video $\rightarrow$ scene concepts
  - images/video $\rightarrow$ objects identified/recognised

- **Audio**
  - music $\rightarrow$ genres / moods
  - sound signals $\rightarrow$ events / concepts

- **Bioinformatics**
  - genes $\rightarrow$ biological functions

- **Robotics**
  - sensor inputs $\rightarrow$ states / error diagnosis
## Datasets and Statistics

<table>
<thead>
<tr>
<th>Dataset</th>
<th>N</th>
<th>L</th>
<th>$(\sum y)/N$</th>
<th>uniq.y</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Music</td>
<td>593</td>
<td>6</td>
<td>1.87</td>
<td>0.046</td>
<td>media</td>
</tr>
<tr>
<td>Scene</td>
<td>2407</td>
<td>6</td>
<td>1.07</td>
<td>0.006</td>
<td>media</td>
</tr>
<tr>
<td>Yeast</td>
<td>2417</td>
<td>14</td>
<td>4.24</td>
<td>0.082</td>
<td>biology</td>
</tr>
<tr>
<td>Genbase</td>
<td>661</td>
<td>27</td>
<td>1.25</td>
<td>0.048</td>
<td>biology</td>
</tr>
<tr>
<td>Medical</td>
<td>978</td>
<td>45</td>
<td>1.25</td>
<td>0.096</td>
<td>medical text</td>
</tr>
<tr>
<td>Slashdot</td>
<td>3782</td>
<td>22</td>
<td>1.18</td>
<td>0.041</td>
<td>news</td>
</tr>
<tr>
<td>Lang.Log</td>
<td>1460</td>
<td>75</td>
<td>1.18</td>
<td>0.208</td>
<td>forum</td>
</tr>
<tr>
<td>Enron</td>
<td>1702</td>
<td>53</td>
<td>3.38</td>
<td>0.442</td>
<td>e-mail</td>
</tr>
<tr>
<td>Reuters(avg)</td>
<td>6000</td>
<td>103</td>
<td>1.46</td>
<td>0.147</td>
<td>news</td>
</tr>
<tr>
<td>OHSUMED</td>
<td>13929</td>
<td>23</td>
<td>1.66</td>
<td>0.082</td>
<td>medical text</td>
</tr>
<tr>
<td>tmc2007</td>
<td>28596</td>
<td>22</td>
<td>2.16</td>
<td>0.047</td>
<td>text</td>
</tr>
<tr>
<td>Media Mill</td>
<td>43907</td>
<td>101</td>
<td>4.38</td>
<td>0.149</td>
<td>media</td>
</tr>
<tr>
<td>Bibtex</td>
<td>7395</td>
<td>159</td>
<td>2.40</td>
<td>0.386</td>
<td>text</td>
</tr>
<tr>
<td>IMDB</td>
<td>95424</td>
<td>28</td>
<td>1.92</td>
<td>0.036</td>
<td>text</td>
</tr>
<tr>
<td>del.icio.us</td>
<td>16105</td>
<td>983</td>
<td>19.02</td>
<td>0.981</td>
<td>text</td>
</tr>
</tbody>
</table>
Issues and Challenges

Multi-label learning issues / challenges:
- correlations between labels
- dimensionality (output space $2^L$ instead of $L$)
- measures of evaluation / loss functions
- an emerging task; no ‘standardised’ datasets, measures, benchmark methods, etc.

(IMDB dataset: co-occurrences (subset), and conditional probabilities)
Aim

Existing methods:

- very computationally complex (often not applicable in practice);
- very specialised (for a specific domain, dimension, setting); or
- not very competitive (in terms of predictive performance).
Aim

Existing methods:
- very computationally complex (often not applicable in practice);
- very specialised (for a specific domain, dimension, setting); or
- not very competitive (in terms of predictive performance).

The aim of this research was to provide multi-label methods which are:
- scalable
- generally applicable; and
- competitive with state-of-the-art methods
Approach: Problem Transformation

Problem Transformation

- Transform a multi-label problem into single-label problems
- Flexible, general, can be more scalable
- Can use any off-the-shelf single-label classifier (kNN, Decision Trees, SVMs, Naive Bayes, etc.)
Problem Transformation

- Transform a multi-label problem into single-label problems
- Flexible, general, can be more scalable
- Can use any off-the-shelf single-label classifier (\(k\)NN, Decision Trees, SVMs, Naive Bayes, etc.)

For example:

- Label Combination method: each combination becomes a single class-label.
  - \(\mathcal{Y} = \text{distinct}({y_1, \ldots, y_N})\)
  - \(\hat{y} = h(x)\)

- Binary Relevance method: each label is a separate binary problem.
  - \(\mathcal{Y}_j = \{0, 1\}\)
  - \(\hat{y}_j = h_j(x)\)
Main Contribution 1: The Pruned Sets Method

The Label Combination method (each $y_i$ is a single class-label):
- Usually good performance, but
- worst-case complexity $\min(2^L, N)$ classes; and
- issues with label sparsity and overfitting.

Main Contribution 1: The Pruned Sets Method

The Label Combination method (each $y_i$ is a single class-label):

- Usually good performance, but
- worst-case complexity $\min(2^L, N)$ classes; and
- issues with label sparsity and overfitting.

The Pruned Sets Method [Read et al., 2008]$^1$: Prune and subsample infrequent label combinations.

- prune where $P(y_i) < p$, and subsample top $s$ best subsets $y_{i1}, \ldots, y_is$ (more frequent and more labels = better)
- e.g. $(x, [1^\text{beach}1^\text{urban}1^\text{forest}0^\text{sunset}]) \rightarrow (x, [1100]), (x, [1010])$
- up to two orders of magnitude faster (with SVMs)
- reduces label sparsity and overfitting

---

Main Contribution 1: The Pruned Sets Method

The Label Combination method (each $y_i$ is a single class-label):
- Usually good performance, but
- worst-case complexity $\min(2^L, N)$ classes; and
- issues with label sparsity and overfitting.

The **Pruned Sets Method** [Read et al., 2008]: Prune and subsample *infrequent* label combinations.

- prune where $P(y_i) < p$, and subsample top $s$ best subsets $y_{i1}, \ldots, y_{is}$ (more frequent and more labels = better)
- e.g. $(x, [1\text{beach}1\text{urban}1\text{forest}0\text{sunrise}]) \rightarrow (x, [1100]), (x, [1010])$
- up to two orders of magnitude faster (with SVMs)
- reduces label sparsity and overfitting

**Ensembles of Pruned Sets:**
- more robust; competes with state-of-the-art methods

---

Main Contribution 2: The Classifier Chains Method

The Pruned Sets method worked well, but had limitations:

- difficulty dealing with ‘extreme’ datasets; and
- worst-case time same as the Label Combination method.

---

Main Contribution 2: The Classifier Chains Method

The Pruned Sets method worked well, but had limitations:
- difficulty dealing with ‘extreme’ datasets; and
- worst-case time same as the Label Combination method.

The Binary Relevance method ($L$ separate problems; $\hat{y}_j = h_j(x)$):
- Relatively robust and good theoretical time complexity; but
- does not explicitly model label correlations (poor prediction).

---

Main Contribution 2: The Classifier Chains Method

The Pruned Sets method worked well, but had limitations:
- difficulty dealing with ‘extreme’ datasets; and
- worst-case time same as the Label Combination method.

The Binary Relevance method ($L$ separate problems; $\hat{y}_j = h_j(x)$):
- Relatively robust and good theoretical time complexity; but
- does not explicitly model label correlations (poor prediction).

The **Classifier Chains method** [Read et al., 2009]²: Pass information between binary classifiers.
- $\hat{y}_j = h_j(x, \hat{y}_1, \ldots, \hat{y}_{j-1})$; e.g. $\hat{\text{forest}} = h_3(x, 1^{\text{beach}}, 0^{\text{urban}})$
- improves prediction, and approximately as fast

---

Main Contribution 2: The Classifier Chains Method

The Pruned Sets method worked well, but had limitations:
- difficulty dealing with ‘extreme’ datasets; and
- worst-case time same as the Label Combination method.

The Binary Relevance method ($L$ separate problems; $\hat{y}_j = h_j(x)$):
- Relatively robust and good theoretical time complexity; but
- does not explicitly model label correlations (poor prediction).

The **Classifier Chains method** [Read et al., 2009]$^2$: Pass information between binary classifiers.
- $\hat{y}_j = h_j(x, \hat{y}_1, \ldots, \hat{y}_{j-1})$; e.g. $\hat{\text{forest}} = h_3(x, \text{1_{beach}}, \text{0_{urban}})$
- improves prediction, and approximately as fast

**Ensembles of Classifier Chains:**
- chain order not an issue (random)
- **highly competitive**

Contributions: an example

“Bayes Optimal Multilabel Classification via Probabilistic Classifier Chains” [Cheng et al., 2010]³

- “inspired by the classifier chain (CC) ... by [Read et al., 2009]”

- Probabilistic Classifier Chains (PCC): a Bayes optimal way of forming classifier chains. \( P_x(y) = \prod_{j=1}^{L} h_j(x, y_1, \ldots, y_{j-1}) \).

- some improvement over CC, but ...
Contributions: an example

“Bayes Optimal Multilabel Classification via Probabilistic Classifier Chains” [Cheng et al., 2010]³

- “inspired by the classifier chain (CC) ... by [Read et al., 2009]”
- Probabilistic Classifier Chains (PCC): a Bayes optimal way of forming classifier chains. \( P_x(y) = \prod_{j=1}^{L} h_j(x, y_1, \ldots, y_{j-1}) \).
- some improvement over CC, but ...“PCC has to look at each of the \( 2^L \) paths ... which limits applicability to data sets with not more than ... about 15 labels” (they use 10 — ECC deals with 1000 in thesis).
- “the averaging method [of ECC] brings the predictions to the marginals”; ”overall good performance of ECC”.

---

³Weiwei Cheng and Krzysztof Dembczyński and Eyke Hüllermeier, Bayes optimal multilabel classification via probabilistic classifier chains. 27th International Conference on Machine Learning. 2010
Contributions: a summary

General contributions:

- most extensive empirical analysis in the multi-label literature
- large and varied dataset collection (± three new datasets)
- multiple evaluation measures (± introduced log loss)
- an open-source framework (MEKA\(^4\))

\(^4\)Multi-label wEKA: http://meka.sourceforge.net/
Contributions: a summary

General contributions:

- most extensive empirical analysis in the multi-label literature
- large and varied dataset collection (+ three new datasets)
- multiple evaluation measures (+ introduced log loss)
- an open-source framework (MEKA\textsuperscript{4})

Major contributions:

1. Pruned Sets
2. Classifier Chains

both of which are: scalable, generally applicable, and competitive (with state-of-the-art methods).

building blocks for other methods, as demonstrated by

1. Ensembles of Pruned Sets
2. Ensembles of Classifier Chains

and have already had impact in the literature: e.g. [Cheng et al., 2010, Zhang and Zhang, 2010].

\textsuperscript{4}Multi-label wEKA: http://meka.sourceforge.net/
Multi-label Data Streams

Data Streams

- data instances typically arrive continually and rapidly (data labelling often generated by machine)
- update model and predict in real time
- concept drift

Applications

- sensor data
- transactions (e.g. ATM, online)
- network traffic

---

Multi-label Data Streams

Data Streams
- data instances typically arrive continually and rapidly (data labelling often generated by machine)
- update model and predict in real time
- concept drift

Applications
- sensor data
- transactions (e.g. ATM, online)
- network traffic

Methods
- MOA\textsuperscript{5} framework: existing (single-label) incremental classifiers; concept-change-detection methods, now extended with multi-label classifiers, e.g. Multi-label Hoeffding Tree Classifier with Pruned Sets at the leaves [Read et al., 2010]

\textsuperscript{5}Massive Online Analysis: http://moa.cs.waikato.ac.nz/
Related Tasks

- tag/keyword-assignment
  - more labels, not predefined, more descriptive than categorical
    - e.g. $x \rightarrow$ truck, 4wd, snowing, mountain, cold, trees, fence

- label ranking
  - labels are associated with a rank / real value; $y \in \mathbb{R}^L$
    - e.g. given $Y = \{\text{beach, people, forest, mountain}\}; x \rightarrow [0.7, 0.0, 0.1, 0.2]$

- multi-task learning
  - learning a problem together with other related problems

- transfer learning
  - applying knowledge from one problem to a related problem

- structured outputs
  - labels are structured in some way: graph, hierarchy, coord.s, masks, mappings, bounding boxes, angles, etc.
    - e.g. $x \rightarrow (\text{bird}).\text{sits\_on.}(\text{truck}); \rightarrow \text{bird}@[x, y, z]; \rightarrow \text{fence}@[x_1, y_1][x_2, x_3]$
Thank you for your attention.

References:


