Multi-label Classification

Jesse Read

https://users.ics.aalto.fi/jesse/

Aalto University
Department of Information and Computer Science
Helsinki, Finland

Summer School on Data Sciences for Big Data
September 5, 2015
Multi-label Classification

Binary classification: Is this a picture of the sea?

∈ \{yes, no\}
Multi-label Classification

Multi-class classification: What is this a picture of?

$\in \{\text{sea, sunset, trees, people, mountain, urban}\}$
Multi-label Classification

Multi-label classification: Which labels are relevant to this picture?

\[ \subseteq \{ \text{sea, sunset, trees, people, mountain, urban} \} \]

i.e., multiple labels per instance instead of a single label!
Multi-label Classification

<table>
<thead>
<tr>
<th></th>
<th>$K = 2$</th>
<th>$K &gt; 2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L = 1$</td>
<td>binary</td>
<td>multi-class</td>
</tr>
<tr>
<td>$L &gt; 1$</td>
<td>multi-label</td>
<td>multi-output†</td>
</tr>
</tbody>
</table>

† also known as multi-target, multi-dimensional.

Figure: For $L$ target variables (labels), each of $K$ values.

- multi-output can be cast to multi-label, just as multi-class can be cast to binary.
- **tagging / keyword** assignment: set of labels ($L$) is not predefined
Increasing Interest

<table>
<thead>
<tr>
<th>year</th>
<th>in text</th>
<th>in title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1996-2000</td>
<td>23</td>
<td>1</td>
</tr>
<tr>
<td>2001-2005</td>
<td>188</td>
<td>18</td>
</tr>
<tr>
<td>2006-2010</td>
<td>1470</td>
<td>164</td>
</tr>
<tr>
<td>2011-2015</td>
<td>4550</td>
<td>485</td>
</tr>
</tbody>
</table>

Table: Academic articles containing the phrase ‘*multi-label classification*’ (Google Scholar)
## Single-label vs. Multi-label

**Table: Single-label** $Y \in \{0, 1\}$

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
<th>$X_4$</th>
<th>$X_5$</th>
<th>$Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.1</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0.9</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0.0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0.8</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0.0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0.0</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>?</td>
</tr>
</tbody>
</table>

**Table: Multi-label** $Y \subseteq \{\lambda_1, \ldots, \lambda_L\}$

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
<th>$X_4$</th>
<th>$X_5$</th>
<th>$Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.1</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>${\lambda_2, \lambda_3}$</td>
</tr>
<tr>
<td>0</td>
<td>0.9</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>${\lambda_1}$</td>
</tr>
<tr>
<td>0</td>
<td>0.0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>${\lambda_2}$</td>
</tr>
<tr>
<td>1</td>
<td>0.8</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>${\lambda_1, \lambda_4}$</td>
</tr>
<tr>
<td>1</td>
<td>0.0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>${\lambda_4}$</td>
</tr>
<tr>
<td>0</td>
<td>0.0</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>?</td>
</tr>
</tbody>
</table>
Single-label vs. Multi-label

Table: Single-label $Y \in \{0, 1\}$

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
<th>$X_4$</th>
<th>$X_5$</th>
<th>$Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.1</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0.9</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0.0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0.8</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0.0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0.0</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>?</td>
</tr>
</tbody>
</table>

Table: Multi-label $[Y_1, \ldots, Y_L] \in 2^L$

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
<th>$X_4$</th>
<th>$X_5$</th>
<th>$Y_1$</th>
<th>$Y_2$</th>
<th>$Y_3$</th>
<th>$Y_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.1</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0.9</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0.0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0.8</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0.0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0.0</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>?</td>
<td>?</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>
Text Categorization

For example, the news …

Novo Banco: Portugal bank sell-off hits snag

Portugal’s central bank has missed its deadline to sell Novo Banco, a bank created after the collapse of the country’s second-biggest lender.

- Reuters collection, newswire stories into 103 topic codes
Text Categorization

For example, the IMDb dataset: Textual movie plot summaries associated with genres (labels).
Text Categorization

For example, the IMDb dataset: Textual movie plot summaries associated with genres (labels).

<table>
<thead>
<tr>
<th>i</th>
<th>abandoned</th>
<th>accident</th>
<th>...</th>
<th>violent</th>
<th>wedding</th>
<th>horror</th>
<th>romance</th>
<th>...</th>
<th>comedy</th>
<th>action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>...</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>...</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>...</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>...</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>...</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>...</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>120919</td>
<td>1</td>
<td>1</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Labelling E-mails

For example, the *Enron* e-mails *multi-labelled* to 53 categories by the *UC Berkeley Enron Email Analysis Project*

- Company Business, Strategy, etc.
- Purely Personal
- Empty Message
- Forwarded email(s)
- …
- company image – current
- …
- Jokes, humor (related to business)
- …
- Emotional tone: worry / anxiety
- Emotional tone: sarcasm
- …
- Emotional tone: shame
- Company Business, Strategy, etc.
Labelling Images

Images are labelled to indicate
- multiple concepts
- multiple objects
- multiple people

E.g., Scene data with concept labels
\[ \subseteq \{\text{beach, sunset, foliage, field, mountain, urban}\} \]
Applications: Audio

Labelling music/tracks with genres / voices, concepts, etc.

e.g., Music dataset, audio tracks labelled with different moods, among: {

- amazed-surprised,
- happy-pleased,
- relaxing-calm,
- quiet-still,
- sad-lonely,
- angry-aggressive

}
Medical Diagnosis

- medical history, symptoms → diseases / ailments

  e.g., Medical dataset,
  - clinical free text reports by radiologists
  - label assignment out of 45 ICD-9-CM codes
Genes are associated with **biological functions**.

E.g. the Yeast dataset: 2, 417 genes, described by 103 attributes, labeled into 14 groups of the FunCAt functional catalogue.
Related Tasks

- **multi-output\(^1\) classification**: outputs are nominal
  
  \[ y_j \in \{1, \ldots, K\}, \mathbf{y} \in \mathbb{N}^L \]

- **multi-output regression**: outputs are real-valued
  
  \[ y_j \in \mathbb{R}, \mathbf{y} \in \mathbb{R}^L \]

- **label ranking**, i.e., preference learning

\[ \lambda_3 \succ \lambda_1 \succ \lambda_4 \succ \ldots \succ \lambda_2 \]

\(^1\)aka multi-target, multi-dimensional
Related Areas

- **multi-task learning**: multiple tasks, shared representation, data may come from different sources e.g., learn to recognise speech for different speakers, classify text from different corpora
- **sequential learning**: predict across time indices instead of across label indices
- **structured output prediction**: assume particular structure among outputs, e.g., pixels
Advanced Applications

Figure: Image Segmentation: Foreground $y_j = 1$
Figure: Localization: \( y_j = 1 \) if \( j \)-th tile occupied.
Advanced Applications

Figure: Demand prediction: $y_j = 1$ if high demand at $j$-th node.
Single-label Classification

\[ \hat{y} = h(x) \quad \text{classifier } h \]

\[ = \arg\max_{y \in \{0,1\}} p(y|x) \quad \text{MAP Estimate} \]
\[ \hat{y} = \operatorname{argmax}_{y \in \{0, 1\}} p(y)p(x|y) \quad \text{Generative, } p(y|x) \propto p(x|y)p(y) \]

\[ = \operatorname{argmax}_{y \in \{0, 1\}} p(y) \prod_{d=1}^{D} p(x_d|y) \quad \text{Naive Bayes} \]
Example: Logistic Regression

\[ \hat{y} = \arg\max_{y \in \{0,1\}} p(y|x) \quad \text{• MAP Estimate} \]

\[ p(y = 1|x) = f_w(x) = \frac{1}{1 + \exp(-w^\top x)} \quad \text{• Logistic Regression} \]

and find \( w \) to minimize \( E(w) \)
Focus on the Labels
Multi-label Classification

\[
\hat{y}_j = h_j(x) = \arg\max_{y_j \in \{0, 1\}} p(y_j | x) \quad \text{for index, } j = 1, \ldots, L
\]

and then,

\[
\hat{y} = h(x) = [\hat{y}_1, \ldots, \hat{y}_4]
\]

\[
= \left[ \arg\max_{y_1 \in \{0, 1\}} p(y_1 | x), \ldots, \arg\max_{y_4 \in \{0, 1\}} p(y_4 | x) \right]
\]

\[
= \left[ f_1(x), \ldots, f_4(x) \right] = f(W^\top x)
\]

This is the Binary Relevance method (BR).
Outline

1. Introduction
2. Applications
3. Background
4. Problem Transformation
5. Algorithm Adaptation
6. Label Dependence
7. Multi-label Evaluation
8. Summary & Resources
# BR Transformation

1. Transform dataset …

<table>
<thead>
<tr>
<th>X</th>
<th>Y₁</th>
<th>Y₂</th>
<th>Y₃</th>
<th>Y₄</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x^{(1)})</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>(x^{(2)})</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(x^{(3)})</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(x^{(4)})</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>(x^{(5)})</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

… into \(L\) separate binary problems (one for each label)

<table>
<thead>
<tr>
<th>X</th>
<th>Y₁</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(x^{(1)})</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(x^{(2)})</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(x^{(3)})</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(x^{(4)})</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(x^{(5)})</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>X</th>
<th>Y₂</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(x^{(1)})</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(x^{(2)})</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(x^{(3)})</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(x^{(4)})</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(x^{(5)})</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>X</th>
<th>Y₃</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(x^{(1)})</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(x^{(2)})</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(x^{(3)})</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(x^{(4)})</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(x^{(5)})</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>X</th>
<th>Y₄</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(x^{(1)})</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(x^{(2)})</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(x^{(3)})</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(x^{(4)})</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(x^{(5)})</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. and train with any off-the-shelf binary base classifier.
Why Not Binary Relevance?

BR ignores \textit{label dependence}, i.e.,

\[ p(y|x) \propto p(x) \prod_{j=1}^{L} p(y_j|x) \]

which may not always hold!

\textbf{Example (Film Genre Classification)}

\[ p(y_{\text{romance}}|x) \neq p(y_{\text{romance}}|x, y_{\text{horror}}) \]
Why Not Binary Relevance?

BR ignores label dependence, i.e.,

\[ p(y|x) \propto p(x) \prod_{j=1}^{L} p(y_j|x) \]

which may not always hold!

Table: Average predictive performance (5 fold CV, Exact Match)

<table>
<thead>
<tr>
<th></th>
<th>L</th>
<th>BR</th>
<th>MCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Music</td>
<td>6</td>
<td>0.30</td>
<td>0.37</td>
</tr>
<tr>
<td>Scene</td>
<td>6</td>
<td>0.54</td>
<td>0.68</td>
</tr>
<tr>
<td>Yeast</td>
<td>14</td>
<td>0.14</td>
<td>0.23</td>
</tr>
<tr>
<td>Genbase</td>
<td>27</td>
<td>0.94</td>
<td>0.96</td>
</tr>
<tr>
<td>Medical</td>
<td>45</td>
<td>0.58</td>
<td>0.62</td>
</tr>
<tr>
<td>Enron</td>
<td>53</td>
<td>0.07</td>
<td>0.09</td>
</tr>
<tr>
<td>Reuters</td>
<td>101</td>
<td>0.29</td>
<td>0.37</td>
</tr>
</tbody>
</table>
Classifier Chains

Modelling label dependence,

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

and,

\[ \hat{y} = \arg\max_{y \in \{0,1\}^L} p(y|x) \]
CC Transformation

Similar to BR: make $L$ binary problems, but include previous predictions as feature attributes,

<table>
<thead>
<tr>
<th>$X$</th>
<th>$Y_1$</th>
<th>$X$</th>
<th>$Y_1$</th>
<th>$Y_2$</th>
<th>$X$</th>
<th>$Y_1$</th>
<th>$Y_2$</th>
<th>$Y_3$</th>
<th>$X$</th>
<th>$Y_1$</th>
<th>$Y_3$</th>
<th>$Y_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x^{(1)}$</td>
<td>0</td>
<td>$x^{(1)}$</td>
<td>0</td>
<td>1</td>
<td>$x^{(1)}$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>$x^{(1)}$</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x^{(2)}$</td>
<td>1</td>
<td>$x^{(2)}$</td>
<td>1</td>
<td>0</td>
<td>$x^{(2)}$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>$x^{(2)}$</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x^{(3)}$</td>
<td>0</td>
<td>$x^{(3)}$</td>
<td>0</td>
<td>1</td>
<td>$x^{(3)}$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>$x^{(3)}$</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$x^{(4)}$</td>
<td>1</td>
<td>$x^{(4)}$</td>
<td>1</td>
<td>0</td>
<td>$x^{(4)}$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>$x^{(4)}$</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x^{(5)}$</td>
<td>0</td>
<td>$x^{(5)}$</td>
<td>0</td>
<td>0</td>
<td>$x^{(5)}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$x^{(5)}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

and, again, apply any classifier (not necessarily a probabilistic one)!
Greedy CC

$L$ classifiers for $L$ labels. For test instance $\tilde{x}$, classify [22],

1. $\hat{y}_1 = h_1(\tilde{x})$
2. $\hat{y}_2 = h_2(\tilde{x}, \hat{y}_1)$
3. $\hat{y}_3 = h_3(\tilde{x}, \hat{y}_1, \hat{y}_2)$
4. $\hat{y}_4 = h_4(\tilde{x}, \hat{y}_1, \hat{y}_2, \hat{y}_3)$

and return

$\hat{y} = [\hat{y}_1, \ldots, \hat{y}_L]$
\( \hat{y} = h(\tilde{x}) = [?, ?, ?] \)
\[ \hat{y} = h(\tilde{x}) = [1, ?, ?] \]

\[ \hat{y}_1 = h_1(\tilde{x}) = \arg\max_{y_1} p(y_1 | \tilde{x}) = 1 \]
\[ \hat{y} = h(\tilde{x}) = [1, 0, ?] \]

1. \( \hat{y}_1 = h_1(\tilde{x}) = \arg\max_{y_1} p(y_1 | \tilde{x}) = 1 \)
2. \( \hat{y}_2 = h_2(\tilde{x}, \hat{y}_1) = \ldots = 0 \)

Improves over BR; similar build time (if \( L < D \)); able to use any off-the-shelf classifier for \( h_j \); parallelizable

But, errors may be propagated down the chain
\[
\hat{y} = h(\tilde{x}) = [1, 0, 1]
\]

1. \(\hat{y}_1 = h_1(\tilde{x}) = \arg\max_{y_1} p(y_1 | \tilde{x}) = 1\)
2. \(\hat{y}_2 = h_2(\tilde{x}, \hat{y}_1) = \ldots = 0\)
3. \(\hat{y}_3 = h_3(\tilde{x}, \hat{y}_1, \hat{y}_2) = \ldots = 1\)
\hat{y} = h(\tilde{x}) = [1, 0, 1]

1. \hat{y}_1 = h_1(\tilde{x}) = \arg\max_{y_1} p(y_1 | \tilde{x}) = 1
2. \hat{y}_2 = h_2(\tilde{x}, \hat{y}_1) = \ldots = 0
3. \hat{y}_3 = h_3(\tilde{x}, \hat{y}_1, \hat{y}_2) = \ldots = 1

- Improves over BR; similar build time (if \( L < D \));
- able to use any off-the-shelf classifier for \( h_j \); parallelizable
- But, errors may be propagated down the chain
Bayes Optimal CC

Bayes-optimal Probabilistic CC [4] (PCC)

\[ \hat{y} = \arg\max_{y \in \{0,1\}^L} p(y|x) \]

\[ = \arg\max_{y \in \{0,1\}^L} \left\{ p(y_1|x) \prod_{j=2}^{L} p(y_j|x, y_1, \ldots, y_{j-1}) \right\} \quad \text{chain rule} \]

Test all possible paths (\( y = [y_1, \ldots, y_L] \in 2^L \) in total)
Bayes Optimal CC

Example

- Better accuracy than greedy CC but computationally limited to $L \lesssim 15$
Monte-Carlo search for CC

1. **For** $t = 1, \ldots, T$ iterations:
   - Sample $y_t \sim p(y|x)$ the chain [20]
     1. $y_1 \sim p(y_1|x)$ // $y_1 = 1$ with probability $p(y_1|x)$
     2. $y_2 \sim p(y_2|x, y_1, y_2)$
     3. ...
     4. $y_L \sim p(y_L|x, y_1, \ldots, y_{L-1})$

2. **Predict**

   $\hat{y} = \underset{y_t \in \{y_1, \ldots, y_T\}}{\text{argmax}} p(y_t|x)$

**Diagram**

- **x**
- **y1**, **y2**, **y3**, **y4**
Monte-Carlo search for CC

Example

Sample $T$ times . . .

- $p([1, 0, 1]) = 0.6 \cdot 0.7 \cdot 0.6 = 0.252$
- $p([0, 1, 0]) = 0.4 \cdot 0.8 \cdot 0.9 = 0.288$

return $\text{argmax}_{y_t} p(y_t|x)$
Monte-Carlo search for CC

Example

Sample $T$ times …

- $p([1, 0, 1]) = 0.6 \cdot 0.7 \cdot 0.6 = 0.252$
- $p([0, 1, 0]) = 0.4 \cdot 0.8 \cdot 0.9 = 0.288$

return $\text{argmax}_{y_t} p(y_t|x)$

- Tractable, with similar accuracy to (Bayes Optimal) PCC
- Can use other search algorithms, e.g., beam search [13]
Does Label-order Matter?

Are these models equivalent?

$$\begin{align*}
\hat{p}(y_1|x) &= \hat{p}(y_2|\hat{y}_1, x) \\
\hat{p}(y_1|y_2, x) &= \hat{p}(y_2|x)
\end{align*}$$
Does Label-order Matter?

Are these models equivalent?

\[
p(x, y) = p(y_1|x)p(y_2|y_1, x) = p(y_2|x)p(y_1|y_2, x)
\]

but we are estimating \(p\) from finite and noisy data (and possibly doing a greedy search); thus

\[
\hat{p}(y_1|x)\hat{p}(y_2|\hat{y}_1, x) \neq \hat{p}(y_2|x)\hat{p}(y_1|\hat{y}_2, x)
\]
Searching the Chain Space

Can search the space of possible chain orderings [20] with, e.g., Monte Carlo walk.

For $u = 1, \ldots, U$:

1. propose $s_u = [s_1, \ldots, s_L] = \text{permute}([1, \ldots, L])$

2. build model on sequence $s_u$

3. evaluate; accept if better (if $J(s_u) > J(s_{u-1})$)

Use $h_{s_u}$ as the final model.

Example

<table>
<thead>
<tr>
<th>$u$</th>
<th>$s_u = [s_1, \ldots, s_L]$</th>
<th>$J(s_u)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>[4, 2, 0, 1, 3, 5]</td>
<td>0.623</td>
</tr>
<tr>
<td>1</td>
<td>[4, 2, 0, 3, 1, 5]</td>
<td>0.628</td>
</tr>
<tr>
<td>2</td>
<td>[4, 2, 0, 3, 5, 1]</td>
<td>0.638</td>
</tr>
<tr>
<td>3</td>
<td>[4, 0, 2, 3, 5, 1]</td>
<td>0.647</td>
</tr>
<tr>
<td>5</td>
<td>[4, 0, 5, 2, 3, 1]</td>
<td>0.653</td>
</tr>
<tr>
<td>18</td>
<td>[5, 1, 4, 3, 2, 0]</td>
<td>0.654</td>
</tr>
<tr>
<td>23</td>
<td>[5, 4, 0, 1, 2, 3]</td>
<td>0.664</td>
</tr>
<tr>
<td>128</td>
<td>[3, 5, 1, 0, 2, 4]</td>
<td>0.668</td>
</tr>
<tr>
<td>176</td>
<td>[5, 3, 1, 0, 4, 2]</td>
<td>0.669</td>
</tr>
<tr>
<td>225</td>
<td>[5, 3, 1, 4, 0, 2]</td>
<td>0.670</td>
</tr>
</tbody>
</table>

$J(s) := \text{EXACTMATCH}(Y, h_s(X))$ (higher is better)
Searching the Chain Space

- The space is of combinational proportions, ... but a little search can go a long way.
- Many other options:
  - add **temperature** to freeze $s_u$ from left to right over time
  - use a **population** of chain sequences: $s_u^{(1)}, \ldots, s_u^{(M)}$
  - use **beam search**
Chain Structure

We can formulate any structure,

\[ y_j = h_j(x, \text{pa}(y_j)) \]

where \( \text{pa}(y_j) \) = parents of node \( j \).

- If \( \text{pa}(y_j) := \{y_1, \ldots, y_{j-1}\} \) we recover CC
- ‘partial’ models are more efficient and interpretable
Structured Classifiers Chains

1. Measure some heuristic
   - marginal dependence [30]
   - conditional dependence [31]

2. Find a structure

3. Plug in base classifiers and run some CC inference

\[
p(y, \tilde{x}) = \prod_{j=1}^{L} p(y_j | \text{pa}(y_j), \tilde{x})
\]
Structured Classifiers Chains

1. Measure some heuristic
   - marginal dependence [30]
   - conditional dependence [31]
2. Find a structure
3. Plug in base classifiers and run some CC inference

Related to Bayesian networks, [1, 2]:

\[ p(y, \tilde{x}) = \prod_{j=1}^{L} p(y_j | \text{pa}(y_j), \tilde{x}) \]
Label Powerset (LP)

One multi-class problem (taking many values),

\[ \hat{y} = \arg\max_{y \in \{0,1\}^L} p(x) \prod_{j=1}^{L} p(y_j|x, y_1, \ldots, y_{j-1}) \quad \bullet \text{PCC} \]

\[ = \arg\max_{y \in \mathcal{Y}} p(y|x) \quad \bullet \text{LP, where } \mathcal{Y} \subset \{0,1\}^L \]

\[ \equiv \arg\max_{y \in \{0,\ldots,2^L-1\}} p(y|x) \quad \bullet \text{a multi-class problem!} \]
Label Powerset (LP)

One multi-class problem (taking many values),

\[
\hat{y} = \arg\max_{\mathbf{y} \in \{0,1\}^L} p(\mathbf{x}) \prod_{j=1}^{L} p(y_j|\mathbf{x}, y_1, \ldots, y_{j-1}) \quad \bullet \text{PCC}
\]

\[
= \arg\max_{\mathbf{y} \in \mathcal{Y}} p(\mathbf{y} | \mathbf{x}) \quad \bullet \text{LP, where } \mathcal{Y} \subset \{0, 1\}^L
\]

\[
\equiv \arg\max_{\mathbf{y} \in \{0, \ldots, 2^L - 1\}} p(\mathbf{y} | \mathbf{x}) \quad \bullet \text{a multi-class problem!}
\]

- Each value is a label vector,
- typically, the occurrences of the training set.
- In practice, $|\mathcal{Y}| \leq N$, and $|\mathcal{Y}| \ll 2^L$
Label Powerset Method (LP)

1. Transform dataset …

<table>
<thead>
<tr>
<th>( \mathbf{X} )</th>
<th>( Y_1 )</th>
<th>( Y_2 )</th>
<th>( Y_3 )</th>
<th>( Y_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathbf{x}^{(1)} )</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>( \mathbf{x}^{(2)} )</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( \mathbf{x}^{(3)} )</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>( \mathbf{x}^{(4)} )</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>( \mathbf{x}^{(5)} )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

… into a multi-
\textit{class} problem, taking \( 2^L \) possible values:

<table>
<thead>
<tr>
<th>( \mathbf{X} )</th>
<th>( Y \in 2^L )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathbf{x}^{(1)} )</td>
<td>0110</td>
</tr>
<tr>
<td>( \mathbf{x}^{(2)} )</td>
<td>1000</td>
</tr>
<tr>
<td>( \mathbf{x}^{(3)} )</td>
<td>0110</td>
</tr>
<tr>
<td>( \mathbf{x}^{(4)} )</td>
<td>1001</td>
</tr>
<tr>
<td>( \mathbf{x}^{(5)} )</td>
<td>0001</td>
</tr>
</tbody>
</table>

2. … and train any off-the-shelf multi-
\textit{class} classifier.
Issues with LP

- **complexity**: there is no greedy label-by-label option
- **imbalance**: few examples per class label
- **overfitting**: how to predict new value?

**Example**

In the Enron dataset, 44% of labelsets are unique (a single training example or test instance). In del.icio.us dataset, 98% are unique.
RA$k$EL

<table>
<thead>
<tr>
<th>$X$</th>
<th>$Y \in 2^L$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x^{(1)}$</td>
<td>0110</td>
</tr>
<tr>
<td>$x^{(2)}$</td>
<td>1000</td>
</tr>
<tr>
<td>$x^{(3)}$</td>
<td>0110</td>
</tr>
<tr>
<td>$x^{(4)}$</td>
<td>1001</td>
</tr>
<tr>
<td>$x^{(5)}$</td>
<td>0001</td>
</tr>
</tbody>
</table>

Ensembles of RAAndom $k$-labEL subsets (RA$k$EL) [27]

- Do LP on $M$ subsets $\subset \{1, \ldots, L\}$ of size $k$

<table>
<thead>
<tr>
<th>$X$</th>
<th>$Y_{123} \in 2^k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x^{(1)}$</td>
<td>011</td>
</tr>
<tr>
<td>$x^{(2)}$</td>
<td>100</td>
</tr>
<tr>
<td>$x^{(3)}$</td>
<td>011</td>
</tr>
<tr>
<td>$x^{(4)}$</td>
<td>100</td>
</tr>
<tr>
<td>$x^{(5)}$</td>
<td>000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$X$</th>
<th>$Y_{124} \in 2^k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x^{(1)}$</td>
<td>010</td>
</tr>
<tr>
<td>$x^{(2)}$</td>
<td>100</td>
</tr>
<tr>
<td>$x^{(3)}$</td>
<td>010</td>
</tr>
<tr>
<td>$x^{(4)}$</td>
<td>101</td>
</tr>
<tr>
<td>$x^{(5)}$</td>
<td>001</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$X$</th>
<th>$Y_{234} \in 2^k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x^{(1)}$</td>
<td>110</td>
</tr>
<tr>
<td>$x^{(2)}$</td>
<td>000</td>
</tr>
<tr>
<td>$x^{(3)}$</td>
<td>110</td>
</tr>
<tr>
<td>$x^{(4)}$</td>
<td>001</td>
</tr>
<tr>
<td>$x^{(5)}$</td>
<td>001</td>
</tr>
</tbody>
</table>
## Pruned Sets

<table>
<thead>
<tr>
<th>$\mathbf{X}$</th>
<th>$Y \in 2^L$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathbf{x}^{(1)}$</td>
<td>0110</td>
</tr>
<tr>
<td>$\mathbf{x}^{(2)}$</td>
<td>1000</td>
</tr>
<tr>
<td>$\mathbf{x}^{(3)}$</td>
<td>0110</td>
</tr>
<tr>
<td>$\mathbf{x}^{(4)}$</td>
<td>1001</td>
</tr>
<tr>
<td>$\mathbf{x}^{(5)}$</td>
<td>0001</td>
</tr>
</tbody>
</table>

Ensembles of Pruned label Sets (EPS) [21]

- Do LP on $M$ pruned subsets (wrt class values)
- Can flip bits to reduce ratio of classes to examples

### Table Examples

<table>
<thead>
<tr>
<th>$\mathbf{X}$</th>
<th>$Y \in 2^L$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathbf{x}^{(1)}$</td>
<td>0110</td>
</tr>
<tr>
<td>$\mathbf{x}^{(2)}$</td>
<td>1000</td>
</tr>
<tr>
<td>$\mathbf{x}^{(3)}$</td>
<td>0110</td>
</tr>
<tr>
<td>$\mathbf{x}^{(4)}$</td>
<td>1001</td>
</tr>
<tr>
<td>$\mathbf{x}^{(5)}$</td>
<td>0001</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\mathbf{X}$</th>
<th>$Y \in 2^L$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathbf{x}^{(1)}$</td>
<td>0110</td>
</tr>
<tr>
<td>$\mathbf{x}^{(2)}$</td>
<td>1000</td>
</tr>
<tr>
<td>$\mathbf{x}^{(3)}$</td>
<td>0110</td>
</tr>
<tr>
<td>$\mathbf{x}^{(4)}$</td>
<td>0001</td>
</tr>
<tr>
<td>$\mathbf{x}^{(5)}$</td>
<td>1000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\mathbf{X}$</th>
<th>$Y \in 2^L$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathbf{x}^{(1)}$</td>
<td>0110</td>
</tr>
<tr>
<td>$\mathbf{x}^{(2)}$</td>
<td>1000</td>
</tr>
<tr>
<td>$\mathbf{x}^{(3)}$</td>
<td>0110</td>
</tr>
<tr>
<td>$\mathbf{x}^{(4)}$</td>
<td>1001</td>
</tr>
<tr>
<td>$\mathbf{x}^{(5)}$</td>
<td>0001</td>
</tr>
</tbody>
</table>
Ensemble-based Voting

Most problem-transformation methods are ensemble-based, e.g., ECC, EPS, RA$k$EL.

Ensemble Voting

<table>
<thead>
<tr>
<th>$h_1(\tilde{x})$</th>
<th>$\hat{y}_1$</th>
<th>$\hat{y}_2$</th>
<th>$\hat{y}_3$</th>
<th>$\hat{y}_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h_2(\tilde{x})$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>$h_3(\tilde{x})$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>$h_4(\tilde{x})$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Score | 0.75 | 0.25 | 0.75 | 0 |

$\hat{y}$ | 1 | 0 | 1 | 0 |

- more **predictive power** (ensemble effect)
- LP can predict **novel label combinations**
Scaling Up

LSHTC4: Large Scale Hierarchical Text Classification

A wikipedia-scale problem
- 325,056 labels
- 2.4M examples

- Even with only 1,000 features, have to learn over 300M parameters with BR (linear models)
- … plus 52,831M more with CC
- … plus ensembles (×10, ×50?)
- LP transformation generates around 1.47M classes
Scaling Up

Our approach [16, 23]:

1. Ignore the predefined hierarchy
2. Work with subsets of the labelset (RAkEL)
3. Prune them (pruned sets)
4. Chain these sets together (classifier chains)
5. Mix of base classifiers (centroid, decision trees, SVMs)
6. Ensemble with sample features and instances (random subspace)
7. Randomization: splits, pruning, reintroduction, chain links, base classifier parameters
8. Train models in parallel, weight according to score on hold-out sets (avoid overfitting!)
Pairwise Multi-label Classification

<table>
<thead>
<tr>
<th>( \mathbf{X} )</th>
<th>( Y_1 )</th>
<th>( Y_2 )</th>
<th>( Y_3 )</th>
<th>( Y_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathbf{x}^{(1)} )</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>( \mathbf{x}^{(2)} )</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( \mathbf{x}^{(3)} )</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( \mathbf{x}^{(4)} )</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>( \mathbf{x}^{(5)} )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

- Create a pairwise transformation, of up to \( \frac{L(L-1)}{2} \) binary classifiers (\textit{all-vs-all}), but smaller than in BR

<table>
<thead>
<tr>
<th>( \mathbf{X} )</th>
<th>( Y_{1v2} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathbf{x}^{(1)} )</td>
<td>0</td>
</tr>
<tr>
<td>( \mathbf{x}^{(2)} )</td>
<td>1</td>
</tr>
<tr>
<td>( \mathbf{x}^{(3)} )</td>
<td>0</td>
</tr>
<tr>
<td>( \mathbf{x}^{(4)} )</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( \mathbf{X} )</th>
<th>( Y_{1v3} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathbf{x}^{(1)} )</td>
<td>0</td>
</tr>
<tr>
<td>( \mathbf{x}^{(2)} )</td>
<td>1</td>
</tr>
<tr>
<td>( \mathbf{x}^{(3)} )</td>
<td>1</td>
</tr>
<tr>
<td>( \mathbf{x}^{(4)} )</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( \mathbf{X} )</th>
<th>( Y_{1v4} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathbf{x}^{(1)} )</td>
<td>1</td>
</tr>
<tr>
<td>( \mathbf{x}^{(2)} )</td>
<td>1</td>
</tr>
<tr>
<td>( \mathbf{x}^{(3)} )</td>
<td>1</td>
</tr>
<tr>
<td>( \mathbf{x}^{(4)} )</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( \mathbf{X} )</th>
<th>( Y_{2v4} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathbf{x}^{(1)} )</td>
<td>1</td>
</tr>
<tr>
<td>( \mathbf{x}^{(2)} )</td>
<td>0</td>
</tr>
<tr>
<td>( \mathbf{x}^{(3)} )</td>
<td>0</td>
</tr>
</tbody>
</table>

- Ensemble voting, or calibrated label ranking [7]
- Can also model four classes (related to LP)
Hierarchy of MLC (HOMER)

1. Cluster labels (randomly, $k$-means) [28], or use pre-defined hierarchy
2. Apply problem transformation
Multi-label Regularization

\[ \hat{y} = b(h(x)), \text{ or } \hat{y} = b(h(x), x) \]

where
\[ \tilde{y} = h(x) \] is an initial classification; and
\[ b \] is some regularizer

Examples:

- **Meta BR**: A second (meta) BR \((b)\) takes as input the output from an initial BR \((h)\) [9]
- **Error Correcting Output Codes**: bit vector \(\tilde{y}\) has been distorted by noise; attempt to correct it [6]
- **Subset matching**: if \(\tilde{y}\) does not exist in training set, match it to the closest one that does
Problem Transformation Summary

Two ways of viewing a multi-label problem of $L$ labels:

1. $L$ binary problems (BR),
2. a multi-class problem with $2^L$ classes (LP)

or a combination of these.

General method:

1. Transform data into subproblems (binary or multi-class)
2. Apply some off-the-shelf base classifier
3. (Optional) Regularize
4. (Optional) Ensemble
Outline

1. Introduction
2. Applications
3. Background
4. Problem Transformation
5. Algorithm Adaptation
6. Label Dependence
7. Multi-label Evaluation
8. Summary & Resources
Algorithm Adaptation

1. Take your favourite (most suitable) classifier
2. Modify it for multi-label classification

- Advantage: a single model, usually very scalable
- Disadvantage: predictive performance depends on the problem domain
$k$ Nearest Neighbours ($k$NN)

Assign to $\tilde{x}$ the majority class of the $k$ ‘nearest neighbours’

$$\hat{y} = \arg\max_y \sum_{i \in N_k} y^{(i)}$$

where $N_k$ contains the training pairs with $x^{(i)}$ closest to $\tilde{x}$.
Multi-label $k$NN

Assigns the most common labels of the $k$ nearest neighbours

\[ p(y_j = 1|x) = \frac{1}{k} \sum_{i \in N_k} y_j^{(i)} \]

\[ \hat{y}_j = \arg\max_{y_j \in \{0,1\}} [p(y_j|x) > 0.5] \]

For example, [32]. Related to ensemble voting.
Decision Trees

- construct like C4.5 (multi-label entropy [3])
- multiple labels at the leaves
- predictive clustering trees [12] are highly competitive in an random forest/ensemble
Conditional Random Fields

\[
p(y|x) = \frac{1}{Z(x)} \prod_c \phi_c(x, y)
\]

\[
= \frac{1}{Z(x)} \exp\{\sum_c w_c f_c(x, y)\}
\]

where, e.g., \( \phi_3(x, y) = \phi_3(y_1, y_2) \propto p(y_2|y_1) \). Factors can be modelled with, e.g., with a problem transformation
where, e.g., $\phi_3(x, y) = \phi_3(y_1, y_2) \propto p(y_2|y_1)$. Factors can be modelled with, e.g., with a problem transformation
Conditional Random Fields

\[
\phi_3(x, y) = \phi_3(y_1, y_2) \propto p(y_2|y_1).
\]

Factors can be modelled with, e.g., with a problem transformation, but computational burden is shifted to inference, e.g.,

\[
\hat{y} = \arg\max_{y \in \{0, 1\}^L} f_1(x, y_1)f_2(x, y_2)f_3(y_2, y_1)
\]

- Gibbs simpling [10] (like an undirected PCC)
- Supported combinations [8] (i.e., \(\mathcal{Y}\) in LP)
Just include an output node for each label.

train with, e.g., gradient descent + error back-propagation
Other Algorithm Adaptations

- Max-margin methods / SVMs [29]
- Association rules [25]
- Boosting [24]
- Generative (Bayesian) [15]
Label Dependence in MLC

Common approach: Present methods to

1. measure label dependence
2. find a structure that best represents this
and then apply classifiers, compare results to BR.
Label Dependence in MLC

Common approach: Present methods to
1. measure **label dependence**
2. find a **structure** that best represents this
and then apply classifiers, compare results to BR.

**Example**

- Which labels (nodes) to link together? (CC, PGMs)
- Which subsets to form from the labelset? (RAkEL)
Label Dependence in MLC

Common approach: Present methods to

1. measure label dependence
2. find a structure that best represents this

and then apply classifiers, compare results to BR.

⚠️ Problem

Accuracy often indistinguishable to that of random ensembles, or slow! (although, may be more compact and/or interpretable)
Marginal label dependence

When the joint is **not** the product of the marginals, i.e.,

\[
p(y_2) \neq p(y_2 | y_1) \\
p(y_1)p(y_2) \neq p(y_1, y_2)
\]

- Estimate from co-occurrence frequencies in training data
Marginal label dependence

Example

Figure: Music dataset - Mutual Information
Marginal label dependence

Example

Figure: Scene dataset - Mutual Information
Exploiting marginal dependence

A Toy Dataset

<table>
<thead>
<tr>
<th></th>
<th>X₁</th>
<th>X₂</th>
<th>Y₁</th>
<th>Y₂</th>
<th>Y₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Measure marginal label dependence (i.e., do labels co-occur frequently, or does one exclude the other?).
Exploiting marginal dependence

A Toy Dataset

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$Y_1$</th>
<th>$Y_2$</th>
<th>$Y_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Measure marginal label dependence (i.e., do labels co-occur frequently, or does one exclude the other?).

- But all labels are interdependent! For example,

$$\hat{p}(y_2 = 1 | y_1 = 1) \neq \hat{p}(y_2 = 1)$$

$$\frac{1}{3} > \frac{1}{4}$$

- Could use a threshold, or statistical significance, …
- But how does this relate to classification, $p(y_j | x)$?
Conditional label dependence

... conditioned on input observation $x$.

$$p(y_2|y_1, x) \neq p(y_2|x)$$

- Have to build and measure models

Indication of conditional dependence if
  - the performance of LP/CC exceeds that of BR
  - errors among the binary models are correlated
Conditional label dependence

Conditional *independence*

...conditioned on input observation $\mathbf{x}$.

\[ p(y_2) \neq p(y_2|y_1) \]

, but $p(y_2 | \mathbf{x}) = p(y_2 | y_1, \mathbf{x})$

- Have to build and measure models

Indication of conditional dependence if

- the performance of LP/CC exceeds that of BR
- errors among the binary models are correlated
Exploiting conditional dependence

A Toy Dataset

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$Y_1$</th>
<th>$Y_2$</th>
<th>$Y_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Measure *conditional label dependence* (build models, measure the difference in error rate).
Exploiting conditional dependence

A Toy Dataset

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Measure *conditional label dependence* (build models, measure the difference in error rate).

- But building models is *expensive*!
- Which *structure* to construct?
Exploiting conditional dependence

A Toy Dataset

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$Y_1$</th>
<th>$Y_2$</th>
<th>$Y_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Oracle: complete conditional independence!

Complete conditional independence,

$$p(Y_j|Y_k, X_1, X_2) = p(Y_j|X_1, X_2), \forall j, k : 0 < j < k \leq L$$

Then the binary relevance (BR) classifier should suffice?
The LOGICAL Problem

Figure: BR (left), CC (middle), LP (right)

Table: The LOGICAL problem, base classifier logistic regression.

<table>
<thead>
<tr>
<th>Metric</th>
<th>BR</th>
<th>CC</th>
<th>LP</th>
</tr>
</thead>
<tbody>
<tr>
<td>HAMMING SCORE</td>
<td>0.83</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>EXACT MATCH</td>
<td>0.50</td>
<td>1.00</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Why didn’t BR work?
XOR Solution

- Only one of these works (with greedy inference)!
- The ground truth (oracle) is

\[ p(y_{\text{XOR}}|y_{\text{AND}}, x) = p(y_{\text{XOR}}|x) \]

but, recall: we have an estimation of this,

\[ \hat{f}(y_{\text{XOR}}|y_{\text{AND}}, x) \neq \hat{f}(y_{\text{XOR}}|x) \]

(fininite data, finite training time, limited class of model \( \hat{f} \), i.e., linear): dependence depends on the model!
Solutions

1. Use a suitable *structure*
2. Use a suitable *base classifier*
3. Ensure that labels are conditionally *independent*. 
Solutions

1. Use a suitable *structure* How to find it?
2. Use a suitable *base classifier* Which one is suitable?
3. Ensure that labels are conditionally *independent*. How to do that?

Main limiting factor: *computational complexity*. 
The LOGICAL Problem

Figure: Binary Relevance (BR): linear decision boundary (solid line, estimated with logistic regression) not viable for $Y_{\text{XOR}}$ label
Solution via Structure

Figure: Solution via structure: linear model now applicable to $Y_{\text{XOR}}$
Solution via Structure

Figure: Solution via structure: two labels have augmented decision space

Can also use undirected connections

- directionality not an issue,
- but implies greater computational burden ($\approx$ LP)
- ...possibly shifted to inference ($\approx$ PCC, CDN)
Solution via Multi-class Decomposition

Figure: Label Powerset (LP): solvable with one-vs-one multi-class decomposition for any (e.g., linear) base classifier
Solution via Multi-class Decomposition

Figure: Label Powerset (LP): solvable with one-vs-one multi-class decomposition for any (e.g., linear) base classifier. Also possible with RAKEL subsets $Y_{OR,XOR}$ and $Y_{AND}$
Solution via Con. Independence

Figure: Solution via non-linear (e.g., random RBF) transformation on input to new space $z$ (creating independence).
Solution via Suitable Base-classifier

Figure: Solution via non-linear classifier (e.g., Decision Tree). Leaves hold examples, where $y = [y_{or}, y_{and}, y_{xor}]$
On Real World Problems . . .

Figure: Music dataset, kernel PCA
Latent Variables

\[
p(y|\mathbf{x}) = \frac{\sum_z p(x, y, z)}{p(x)} = \frac{1}{Z} \sum_z p(x, y, z)
\]

- Can view label dependencies as having marginalized out latent variables
Inner Layer Methods

1. Use an inner layer
   \[ z = f(x), \quad z \in \mathbb{R}^H \]

2. Apply a classifier
   \[ y = h(z) \]

- PCA, CCA [17]
- Kernel PCA [29]
- Mixture models [15]
- Clustering [28]
- Compressive Sensing [11]
- Deep Learning [18]
- Auto Encoders
Another look: Problem Transformation

Figure: Methods CC and RAKEL (among others) can be viewed as using an inner layer [18].
What about marginal dependence?

- Can be seen as a kind of constraint
- used for regularization
  (recall: e.g., ECOC, subset matching)
Label Dependence: Summary

- Marginal dependence for regularization
- Conditional dependence
  - ...depends on the model
  - ...may be introduced
- Should consider together:
  - base classifier
  - structure
  - inner layer
- An open problem
- Much existing research is relevant
Outline

1. Introduction
2. Applications
3. Background
4. Problem Transformation
5. Algorithm Adaptation
6. Label Dependence
7. Multi-label Evaluation
8. Summary & Resources
Multi-label Evaluation

In single-label classification, simply compare true label $y$ with predicted label $\hat{y}$ [or $p(y|\tilde{x})$]. What about in multi-label classification?

**Example**

If true label vector is $y = [1, 0, 0, 0]$, then $\hat{y} =$?

- compare bit-wise? too lenient?
- compare vector-wise? too strict?
# Hamming Loss

## Example

<table>
<thead>
<tr>
<th>$\hat{y}^{(i)}$</th>
<th>$\hat{y}^{(i)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tilde{x}^{(1)}$</td>
<td>[1 0 1 0]</td>
</tr>
<tr>
<td>$\tilde{x}^{(2)}$</td>
<td>[0 1 0 1]</td>
</tr>
<tr>
<td>$\tilde{x}^{(3)}$</td>
<td>[1 0 0 1]</td>
</tr>
<tr>
<td>$\tilde{x}^{(4)}$</td>
<td>[0 1 1 0]</td>
</tr>
<tr>
<td>$\tilde{x}^{(5)}$</td>
<td>[1 0 0 0]</td>
</tr>
</tbody>
</table>

Hamming Loss = \[
\frac{1}{NL} \sum_{i=1}^{N} \sum_{j=1}^{L} \mathbb{I}[\hat{y}_j^{(i)} \neq y_j^{(i)}]
\]

= 0.20
0/1 Loss

Example

<table>
<thead>
<tr>
<th>$\tilde{x}^{(1)}$</th>
<th>$[1 0 1 0]$</th>
<th>$[1 0 0 1]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tilde{x}^{(2)}$</td>
<td>$[0 1 0 1]$</td>
<td>$[0 1 0 1]$</td>
</tr>
<tr>
<td>$\tilde{x}^{(3)}$</td>
<td>$[1 0 0 1]$</td>
<td>$[1 0 0 1]$</td>
</tr>
<tr>
<td>$\tilde{x}^{(4)}$</td>
<td>$[0 1 1 0]$</td>
<td>$[0 1 0 0]$</td>
</tr>
<tr>
<td>$\tilde{x}^{(5)}$</td>
<td>$[1 0 0 0]$</td>
<td>$[1 0 0 1]$</td>
</tr>
</tbody>
</table>

$$0/1 \text{ LOSS} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(\hat{y}^{(i)} \neq y^{(i)})$$

$$= 0.60$$
Other Metrics

- **JACCARD INDEX** – often called multi-label **ACCURACY**
- **RANK LOSS** – average fraction of pairs not correctly ordered
- **ONE ERROR** – if top ranked label is not in set of true labels
- **COVERAGE** – average “depth” to cover all true labels
- **LOG LOSS** – i.e., cross entropy
- **PRECISION** – predicted positive labels that are relevant
- **RECALL** – relevant labels which were predicted
- **PRECISION vs. RECALL** curves
- **F-MEASURE**
  - *micro-averaged* (‘global’ view)
  - *macro-averaged* by label (ordinary averaging of a binary measure, changes in infrequent labels have a big impact)
  - *macro-averaged* by example (one example at a time, average across examples)

*For general evaluation, use multiple and contrasting evaluation measures!*
**HAMMING LOSS vs. 0/1 LOSS**

**Hamming loss**
- *evaluation by example*, suitable for evaluating
  \[
  \hat{y}_j = \arg\max_{y_j \in \{0,1\}} p(y_j|x)
  \]
  i.e., BR
- favours sparse labelling
- does not benefit directly from modelling label dependence

**0/1 loss**
- *evaluation by label*, suitable for evaluating
  \[
  y = \arg\max_{y \in \{0,1\}^L} p(y|x)
  \]
  i.e., PCC, LP
- does not favour sparse labelling
- benefits from models of label dependence
### Example: 0/1 LOSS vs. HAMMING LOSS

<table>
<thead>
<tr>
<th>$\mathbf{y}^{(i)}$</th>
<th>$\mathbf{\hat{y}}^{(i)}$</th>
<th>Ham. Loss 0.3</th>
<th>0/1 Loss 0.6</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tilde{x}^{(1)}$</td>
<td>[1 0 1 0]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\tilde{x}^{(2)}$</td>
<td>[1 0 0 1]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\tilde{x}^{(3)}$</td>
<td>[0 1 1 0]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\tilde{x}^{(4)}$</td>
<td>[1 0 0 0]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\tilde{x}^{(5)}$</td>
<td>[0 1 0 1]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
HAMMING LOSS vs. 0/1 LOSS

Example: 0/1 LOSS vs. HAMMING LOSS

<table>
<thead>
<tr>
<th>( \tilde{x}^{(i)} )</th>
<th>( y^{(i)} )</th>
<th>( \hat{y}^{(i)} )</th>
<th>Optimize HAMMING LOSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \tilde{x}^{(1)} )</td>
<td>[1 0 1 0]</td>
<td>[1 0 1 1]</td>
<td>…</td>
</tr>
<tr>
<td>( \tilde{x}^{(2)} )</td>
<td>[1 0 0 1]</td>
<td>[1 0 1 0]</td>
<td>HAM. LOSS 0.2</td>
</tr>
<tr>
<td>( \tilde{x}^{(3)} )</td>
<td>[0 1 1 0]</td>
<td>[0 1 1 0]</td>
<td>0/1 Loss 0.8</td>
</tr>
<tr>
<td>( \tilde{x}^{(4)} )</td>
<td>[1 0 0 0]</td>
<td>[1 0 1 0]</td>
<td>…0/1 LOSS goes up</td>
</tr>
<tr>
<td>( \tilde{x}^{(5)} )</td>
<td>[0 1 0 1]</td>
<td>[0 1 0 1]</td>
<td></td>
</tr>
</tbody>
</table>
### Hamming Loss vs. 0/1 Loss

**Example: 0/1 Loss vs. Hamming Loss**

<table>
<thead>
<tr>
<th>( \tilde{x}^{(i)} )</th>
<th>( y^{(i)} )</th>
<th>( \hat{y}^{(i)} )</th>
<th>Optimize 0/1 Loss…</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \tilde{x}^{(1)} )</td>
<td>[1 0 1 0]</td>
<td>[0 1 0 1]</td>
<td><strong>Ham. Loss 0.4</strong></td>
</tr>
<tr>
<td>( \tilde{x}^{(2)} )</td>
<td>[1 0 0 1]</td>
<td>[1 0 0 1]</td>
<td><strong>0/1 Loss 0.4</strong></td>
</tr>
<tr>
<td>( \tilde{x}^{(3)} )</td>
<td>[0 1 1 0]</td>
<td>[0 0 1 0]</td>
<td>… Hamming Loss goes up</td>
</tr>
<tr>
<td>( \tilde{x}^{(4)} )</td>
<td>[1 0 0 0]</td>
<td>[0 1 1 1]</td>
<td></td>
</tr>
<tr>
<td>( \tilde{x}^{(5)} )</td>
<td>[0 1 0 1]</td>
<td>[0 1 0 1]</td>
<td></td>
</tr>
</tbody>
</table>
HAMMING LOSS vs. 0/1 LOSS

**Example: 0/1 LOSS vs. HAMMING LOSS**

<table>
<thead>
<tr>
<th>$\tilde{x}^{(i)}$</th>
<th>$y^{(i)}$</th>
<th>$\hat{y}^{(i)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tilde{x}^{(1)}$</td>
<td>[1 0 1 0]</td>
<td>[0 1 0 1]</td>
</tr>
<tr>
<td>$\tilde{x}^{(2)}$</td>
<td>[1 0 0 1]</td>
<td>[1 0 0 1]</td>
</tr>
<tr>
<td>$\tilde{x}^{(3)}$</td>
<td>[0 1 1 0]</td>
<td>[0 0 1 0]</td>
</tr>
<tr>
<td>$\tilde{x}^{(4)}$</td>
<td>[1 0 0 0]</td>
<td>[0 1 1 1]</td>
</tr>
<tr>
<td>$\tilde{x}^{(5)}$</td>
<td>[0 1 0 1]</td>
<td>[0 1 0 1]</td>
</tr>
</tbody>
</table>

- Usually cannot minimize both at the same time ...
- … unless: labels are independent of each other! [5]
Threshold Selection

Methods often return a posterior probability, or ensemble votes \( p(\tilde{x}) \). Use a threshold of 0.5:

\[
\hat{y}_j = \begin{cases} 
1, & p_j(\tilde{x}) \geq 0.5 \\
0, & \text{otherwise}
\end{cases}
\]

**Example with threshold of 0.5**

<table>
<thead>
<tr>
<th>(\tilde{x}^{(i)})</th>
<th>(y^{(i)})</th>
<th>(p(\tilde{x}^{(i)}))</th>
<th>(\hat{y}^{(i)} := \mathbb{I}[p(\tilde{x}^{(i)}) \geq 0.5])</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\tilde{x}^{(1)})</td>
<td>[1 0 1 0]</td>
<td>[0.9 0.0 0.4 0.6]</td>
<td>[1 0 0 1]</td>
</tr>
<tr>
<td>(\tilde{x}^{(2)})</td>
<td>[0 1 0 1]</td>
<td>[0.1 0.8 0.0 0.8]</td>
<td>[0 1 0 1]</td>
</tr>
<tr>
<td>(\tilde{x}^{(3)})</td>
<td>[1 0 0 1]</td>
<td>[0.8 0.0 0.1 0.7]</td>
<td>[1 0 0 1]</td>
</tr>
<tr>
<td>(\tilde{x}^{(4)})</td>
<td>[0 1 1 0]</td>
<td>[0.1 0.7 0.4 0.2]</td>
<td>[0 1 0 0]</td>
</tr>
<tr>
<td>(\tilde{x}^{(5)})</td>
<td>[1 0 0 0]</td>
<td>[1.0 0.0 0.0 1.0]</td>
<td>[1 0 0 1]</td>
</tr>
</tbody>
</table>
Threshold Selection

Methods often return a posterior probability, or ensemble votes $p(\tilde{x})$. Use a threshold of 0.5?

$$\hat{y}_j = \begin{cases} 
1, & p_j(\tilde{x}) \geq 0.5 \\
0, & \text{otherwise}
\end{cases}$$

Example with threshold of 0.5

<table>
<thead>
<tr>
<th>$\tilde{x}^{(i)}$</th>
<th>$y^{(i)}$</th>
<th>$p(\tilde{x}^{(i)})$</th>
<th>$\hat{y}^{(i)} := \mathbb{I}[p(\tilde{x}^{(i)}) \geq 0.5]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tilde{x}^{(1)}$</td>
<td>[1 0 1 0]</td>
<td>[0.9 0.0 0.4 0.6]</td>
<td>[1 0 0 1]</td>
</tr>
<tr>
<td>$\tilde{x}^{(2)}$</td>
<td>[0 1 0 1]</td>
<td>[0.1 0.8 0.0 0.8]</td>
<td>[0 1 0 1]</td>
</tr>
<tr>
<td>$\tilde{x}^{(3)}$</td>
<td>[1 0 0 1]</td>
<td>[0.8 0.0 0.1 0.7]</td>
<td>[1 0 0 1]</td>
</tr>
<tr>
<td>$\tilde{x}^{(4)}$</td>
<td>[0 1 1 0]</td>
<td>[0.1 0.7 0.4 0.2]</td>
<td>[0 1 0 0]</td>
</tr>
<tr>
<td>$\tilde{x}^{(5)}$</td>
<td>[1 0 0 0]</td>
<td>[1.0 0.0 0.0 1.0]</td>
<td>[1 0 0 1]</td>
</tr>
</tbody>
</table>

...but would eliminate two errors with a threshold of 0.4!
Threshold Selection

Threshold calibration strategies:

- **Ad-hoc**, e.g., $t = 0.5$
Threshold Selection

Threshold calibration strategies:

- **Ad-hoc**, e.g., $t = 0.5$
- **Internal validation**, e.g., $t \in \{0.1, 0.2, \ldots, 0.9\}$
Threshold Selection

Threshold calibration strategies:

- **Ad-hoc**, e.g., \( t = 0.5 \)
- **Internal validation**, e.g., \( t \in \{0.1, 0.2, \ldots, 0.9\} \)
- **PCut**: such that the training data and test data have similar average number of labels/example

\[
\hat{t} = \arg\min_t \left| \frac{1}{N} \sum_{i,j} \mathbb{1}(p_j^{(i)} > t) - \frac{1}{N} \sum_{i,j} y_j^{(i)} \right|
\]

- Can be done efficiently.
- Can also calibrate \( t_j \) for each label individually.
- Assumes training set similar to test set (i.e., not ideal for data streams)
- Can be viewed as another form of regularization

\[
\hat{y} = b(h(\tilde{x}))
\]
Various Real-World Concerns

- In **data streams**, label dependence (and therefore, appropriate structures/transformations/base classifiers)
  - may not be known in advance
  - must learn it incrementally
  - and adapt to change over time (**concept drift**)
  - **New labels** must be incorporated, old labels phased out
- Labels may be **missing** from training data,
  - but *we don’t know when they’re missing* (non-relevance ≠ missing)
  - Labelling is more intensive per example (affects both manual labelling and active learning)
Multi-label classification

- Can be approached via problem transformation or algorithm adaptation
- Label dependence and scalability are the main themes
- An active area of research and a gateway to many related areas
Resources

- Overview [26]
- Review/Survey of Algorithms [33]
- Extensive empirical comparison [14]
- Some slides: A, B, C
- http://users.ics.aalto.fi/jesse/
Software & Datasets

- Mulan (Java)
- Meka (Java)
- Scikit-Learn (Python) offers some multi-label support
- Clus (Java)
- LAMDA (Matlab)

Datasets

- http://mulan.sourceforge.net/datasets.html
- http://meka.sourceforge.net/#datasets
MEKA

- A WEKA-based framework for multi-label classification and evaluation
- Support for data-stream, semi-supervised classification

http://meka.sourceforge.net
A MEKA Classifier

```java
package weka.classifiers.multilabel;
import weka.core.*;

public class DumbClassifier extends MultilabelClassifier {

    /**
     * BuildClassifier
     */
    public void buildClassifier (Instances D) throws Exception {
        // the first L attributes are the labels
        int L = D.classIndex();
    }

    /**
     * DistributionForInstance – return the distribution p(y[j] | x)
     */
    public double[] distributionForInstance(Instance x) throws Exception {
        int L = x.classIndex();
        // predict 0 for each label
        return new double[L];
    }
}
```
Antonucci Alessandro, Giorgio Corani, Denis Mauá, and Sandra Gabaglio.
An ensemble of Bayesian networks for multilabel classification.

Hanen Borchani.
Multi-dimensional classification using Bayesian networks for stationary and evolving streaming data.

Amanda Clare and Ross D. King.
Knowledge discovery in multi-label phenotype data.

Krzysztof Dembczyński, Weiwei Cheng, and Eyke Hüllermeier.
Bayes optimal multilabel classification via probabilistic classifier chains.

Krzysztof Dembczyński, Willem Waegeman, Weiwei Cheng, and Eyke Hüllermeier.
On label dependence and loss minimization in multi-label classification.

Chun-Sung Ferng and Hsuan-Tien Lin.
Multi-label classification with error-correcting codes.

Johannes Fürnkranz, Eyke Hüllermeier, Eneldo Loza Mencía, and Klaus Brinker.
Multilabel classification via calibrated label ranking.

Nadia Ghamrawi and Andrew McCallum.
Collective multi-label classification.
Shantanu Godbole and Sunita Sarawagi.
Discriminative methods for multi-labeled classification.

Yuhong Guo and Suicheng Gu.
Multi-label classification using conditional dependency networks.
In IJCAI ’11: 24th International Conference on Artificial Intelligence, pages 1300–1305. IJCAI/AAAI, 2011.

Daniel Hsu, Sham M. Kakade, John Langford, and Tong Zhang.
Multi-label prediction via compressed sensing.

Dragi Kocev, Celine Vens, Jan Struyf, and Sašo Dzeroski.
Tree ensembles for predicting structured outputs.

Abhishek Kumar, Shankar Vembu, AdityaKrishna Menon, and Charles Elkan.
Beam search algorithms for multilabel learning.

Gjorgji Madjarov, Dragi Kocev, Dejan Gjorgjevikj, and Sašo Džeroski.
An extensive experimental comparison of methods for multi-label learning.
Pattern Recognition, 45(9):3084–3104, September 2012.

Andrew Kachites McCallum.
Multi-label text classification with a mixture model trained by em.
In AAAI 99 Workshop on Text Learning, 1999.

Antti Puurula, Jesse Read, and Albert Bifet.
Kaggle LSHTC4 winning solution.

Piyush Rai and Hal Daume.
Multi-label prediction via sparse infinite CCA.

Jesse Read and Jaakko Hollmén.
A deep interpretation of classifier chains.

Jesse Read and Jaakko Hollmén.
Multi-label classification using labels as hidden nodes.

Jesse Read, Luca Martino, and David Luengo.
Efficient monte carlo methods for multi-dimensional learning with classifier chains.

Jesse Read, Bernhard Pfahringer, and Geoff Holmes.
Multi-label classification using ensembles of pruned sets.

Jesse Read, Bernhard Pfahringer, Geoffrey Holmes, and Eibe Frank.
Classifier chains for multi-label classification.

Jesse Read, Antti Puurula, and Albert Bifet.
Multi-label classification with meta labels.

Robert E. Schapire and Yoram Singer.
Boostexter: A boosting-based system for text categorization.

F.A. Thabtah, P. Cowling, and Yonghong Peng.
MMAC: A new multi-class, multi-label associative classification approach.
Grigorios Tsoumakas and Ioannis Katakis.  
Multi label classification: An overview.  

Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas.  
Random k-labelsets for multi-label classification.  
*IEEE Transactions on Knowledge and Data Engineering*, 23(7):1079–1089, 2011.

Grigorios Tsoumakas, Ioannis Katakis, and Ioannis P. Vlahavas.  
Effective and efficient multilabel classification in domains with large number of labels.  
In *ECML/PKDD Workshop on Mining Multidimensional Data*, 2008.

Jason Weston, Olivier Chapelle, André Elisseeff, Bernhard Schölkopf, and Vladimir Vapnik.  
Kernel dependency estimation.  

Bayesian chain classifiers for multidimensional classification.  
In *24th International Joint Conference on Artificial Intelligence (IJCAI '11)*, pages 2192–2197, 2011.

Min-Ling Zhang and Kun Zhang.  
Multi-label learning by exploiting label dependency.  

Min-Ling Zhang and Zhi-Hua Zhou.  
ML-KNN: A lazy learning approach to multi-label learning.  

Min-Ling Zhang and Zhi-Hua Zhou.  
A review on multi-label learning algorithms.  
*IEEE Transactions on Knowledge and Data Engineering*, 99(PrePrints):1, 2013.
Multi-label Classification

Jesse Read
https://users.ics.aalto.fi/jesse/

Aalto University
Department of Information and Computer Science
Helsinki, Finland

Summer School on Data Sciences for Big Data
September 5, 2015