Multi-Label Classification

Krzysztof Dembczyński

Intelligent Decision Support Systems Laboratory (IDSS)
Poznań University of Technology, Poland

PL-SIGML, Warszawa, November 12-13, 2013
Many thanks to Willem Waegeman and Eyke Hüllermeier for collaborating on this topic.
Multi-label classification

- A classification problem in which we consider more than one target variable.
Target 1: cloud yes/no
Target 2: sky yes/no
Target 3: tree yes/no
... ... ...
• Prediction of the presence or absence of species.
Multi-label classification

- Training data: \( \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \), \( y_i \in \mathcal{Y} = \{0, 1\}^m \).
- **Predict** a vector \( y = (y_1, y_2, \ldots, y_m) \) for a given \( x \).

<table>
<thead>
<tr>
<th></th>
<th>( X_1 )</th>
<th>( X_2 )</th>
<th>( Y_1 )</th>
<th>( Y_2 )</th>
<th>\ldots</th>
<th>( Y_m )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_1 )</td>
<td>5.0</td>
<td>4.5</td>
<td>1</td>
<td>1</td>
<td>\ldots</td>
<td>0</td>
</tr>
<tr>
<td>( x_2 )</td>
<td>2.0</td>
<td>2.5</td>
<td>0</td>
<td>1</td>
<td>\ldots</td>
<td>0</td>
</tr>
<tr>
<td>( \vdots )</td>
<td>( \vdots )</td>
<td>( \vdots )</td>
<td>( \vdots )</td>
<td>( \vdots )</td>
<td>( \vdots )</td>
<td>( \vdots )</td>
</tr>
<tr>
<td>( x_n )</td>
<td>3.0</td>
<td>3.5</td>
<td>0</td>
<td>1</td>
<td>\ldots</td>
<td>1</td>
</tr>
<tr>
<td>( x )</td>
<td>4.0</td>
<td>2.5</td>
<td>?</td>
<td>?</td>
<td>\ldots</td>
<td>?</td>
</tr>
</tbody>
</table>
Multi-label classification

- Training data: \( \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \), \( y_i \in \mathcal{Y} = \{0, 1\}^m \).
- **Predict** a vector \( y = (y_1, y_2, \ldots, y_m) \) for a given \( x \).

<table>
<thead>
<tr>
<th>( x )</th>
<th>( X_1 )</th>
<th>( X_2 )</th>
<th>( Y_1 )</th>
<th>( Y_2 )</th>
<th>\ldots</th>
<th>( Y_m )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_1 )</td>
<td>5.0</td>
<td>4.5</td>
<td>1</td>
<td>1</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>( x_2 )</td>
<td>2.0</td>
<td>2.5</td>
<td>0</td>
<td>1</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>( x_n )</td>
<td>3.0</td>
<td>3.5</td>
<td>0</td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>( x )</td>
<td>4.0</td>
<td>2.5</td>
<td>1</td>
<td>1</td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>
Multi-label classification

• **Multi-label classification:** For a feature vector $x$ predict accurately a vector of responses $y$ using a function $h(x)$:

$$x = (x_1, x_2, \ldots, x_p) \xrightarrow{h(x)} y = (y_1, y_2, \ldots, y_m)$$

• **Main challenges:**
  ▶ Appropriate modeling of target dependencies between targets $y_1, y_2, \ldots, y_m$
  ▶ A multitude of multivariate loss functions defined over the output vector $\ell(y, h(x))$

• **Main question:**
  ▶ Can we improve over independent models trained for each target?

• **Two views:**
  ▶ The individual-target view
  ▶ The joint-target view
• How can we improve the predictive accuracy of a single label by using information about other labels?
• What are the requirements for improvement?
• What are the specific multivariate loss functions we would like to minimize?
• How to perform minimization of such losses?
• What are the relations between the losses?
• The individual target view:
  ▶ Goal: predict a value of $y_i$ using $x$ and any available information on other targets $y_j$s.
  ▶ The problem is usually defined through univariate losses $\ell(y_i, \hat{y}_i)$.
  ▶ The problem is usually decomposable over the targets.
  ▶ Domain of $y_i$ is either continuous or nominal.
  ▶ Regularized (shrunken) models vs. independent models.

• The joint target view:
  ▶ Goal: predict a vector $y$ using $x$.
  ▶ Multivariate distribution of $y$.
  ▶ The problem is defined through multivariate losses $\ell(y, \hat{y})$.
  ▶ The problem is not easily decomposable over the targets.
  ▶ Domain of $y$ is usually finite, but contains a large number of elements.
  ▶ More expressive models vs. independent models.
Multi-target prediction

-the individual target view

-shrunken models
Reduce model complexity by model sharing.
Example: RR, FicyReg, Curds&Whey, mult-task learning methods, kernel dependency estimation, stacking, compressed sensing, etc.

-independent models
Fit one model for every target (independently).
Example: binary relevance in multi-label classification

-more expressive models
Introduce additional parameters or models for targets or target combinations.
Example: label powerset, structured SVMs, conditional random fields, probabilistic classifier chains (PCC), Max Margin Markov Networks, etc.

-the joint target view
• **Marginal and conditional dependence:**

\[
P(Y) \neq \prod_{i=1}^{m} P(Y_i) \quad P(Y \mid x) \neq \prod_{i=1}^{m} P(Y_i \mid x)
\]

marginal (in)dependence \(\not\equiv\) conditional (in)dependence
• **Model similarities**:

\[ f_i(x) = g_i(x) + \epsilon_i, \text{ for } i = 1, \ldots, m \]

Similarities in the structural parts \( g_i(x) \) of the models.
• **Structure** imposed (domain knowledge) on targets
  ▶ Chains,
  ▶ Hierarchies,
  ▶ General graphs,
  ▶ …
• **Interdependence vs. hypothesis and feature space:**
  - Regularization constrains the hypothesis space.
  - Modeling dependencies may increase the expressiveness of the model.
  - Using a more complex model on individual targets might also help.
  - Comparison between independent and multi-target models is difficult in general, as they differ in many respects (e.g., complexity)!
Multivariate loss functions

- **Decomposable and non-decomposable** losses over examples

\[ L = \sum_{i=1}^{n} \ell(y_i, h(x_i)) \quad L \neq \sum_{i=1}^{n} \ell(y_i, h(x_i)) \]

- **Decomposable and non-decomposable** losses over targets

\[ \ell(y, h(x)) = \sum_{i=1}^{m} \ell(y_i, h_i(x)) \quad \ell(y, h(x)) \neq \sum_{i=1}^{m} \ell(y_i, h_i(x)) \]
• Loss functions and probabilistic view
  ▶ Relations between losses.
  ▶ How to minimize complex loss functions.

• Learning algorithms
  ▶ Reduction algorithms.
  ▶ Conditional random fields (CRFs).
  ▶ Structured support vector machines (SSVMs).
  ▶ Probabilistic classifier chains (PCCs).

• Problem settings
  ▶ Hamming and subset 0/1 loss minimization.
  ▶ Multilabel ranking.
  ▶ F-measure maximization.
• Training data: \( \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \), \( y_i \in \mathcal{Y} = \{0, 1\}^m \).

• **Predict** the vector \( y = (y_1, y_2, \ldots, y_m) \) for a given \( x \).

<table>
<thead>
<tr>
<th></th>
<th>( X_1 )</th>
<th>( X_2 )</th>
<th>( Y_1 )</th>
<th>( Y_2 )</th>
<th>( \ldots )</th>
<th>( Y_m )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_1 )</td>
<td>5.0</td>
<td>4.5</td>
<td>1</td>
<td>1</td>
<td>( \ldots )</td>
<td>0</td>
</tr>
<tr>
<td>( x_2 )</td>
<td>2.0</td>
<td>2.5</td>
<td>0</td>
<td>1</td>
<td>( \ldots )</td>
<td>0</td>
</tr>
<tr>
<td>( \vdots )</td>
<td>( \vdots )</td>
<td>( \vdots )</td>
<td>( \vdots )</td>
<td>( \vdots )</td>
<td>( \vdots )</td>
<td>( \vdots )</td>
</tr>
<tr>
<td>( x_n )</td>
<td>3.0</td>
<td>3.5</td>
<td>0</td>
<td>1</td>
<td>( \ldots )</td>
<td>1</td>
</tr>
<tr>
<td>( x )</td>
<td>4.0</td>
<td>2.5</td>
<td>?</td>
<td>?</td>
<td>( \ldots )</td>
<td>?</td>
</tr>
</tbody>
</table>
A starting example

- Training data: \( \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \), \( y_i \in \mathcal{Y} = \{0, 1\}^m \).
- **Predict** the vector \( y = (y_1, y_2, \ldots, y_m) \) for a given \( x \).

<table>
<thead>
<tr>
<th></th>
<th>( X_1 )</th>
<th>( X_2 )</th>
<th>( Y_1 )</th>
<th>( Y_2 )</th>
<th>\ldots</th>
<th>( Y_m )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_1 )</td>
<td>5.0</td>
<td>4.5</td>
<td>1</td>
<td>1</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>( x_2 )</td>
<td>2.0</td>
<td>2.5</td>
<td>0</td>
<td>1</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
</tr>
<tr>
<td>( x_n )</td>
<td>3.0</td>
<td>3.5</td>
<td>0</td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>( x )</td>
<td>4.0</td>
<td>2.5</td>
<td>1</td>
<td>1</td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>
Two basic approaches

- **Binary relevance**: Decomposes the problem to $m$ binary classification problems:

  $$(x, y) \rightarrow (x, y = y_i), \quad i = 1, \ldots, m$$

- **Label powerset**: Treats each label combination as a new meta-class in multi-class classification:

  $$(x, y) \rightarrow (x, y = \text{metaclass}(y))$$

<table>
<thead>
<tr>
<th></th>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$Y_1$</th>
<th>$Y_2$</th>
<th>...</th>
<th>$Y_m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>5.0</td>
<td>4.5</td>
<td>1</td>
<td>1</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>$x_2$</td>
<td>2.0</td>
<td>2.5</td>
<td>0</td>
<td>1</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>$x_n$</td>
<td>3.0</td>
<td>3.5</td>
<td>0</td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>
• Two independent models:

\[ f_1(\mathbf{x}) = \frac{1}{2}x_1 + \frac{1}{2}x_2, \quad f_2(\mathbf{x}) = \frac{1}{2}x_1 - \frac{1}{2}x_2 \]

• Logistic model to get labels:

\[ P(y_i = 1) = \frac{1}{1 + \exp(-2f_i)} \]
• Two dependent models:

\[ f_1(x) = \frac{1}{2}x_1 + \frac{1}{2}x_2 \quad f_2(y_1, x) = y_1 + \frac{1}{2}x_1 - \frac{1}{2}x_2 - \frac{2}{3} \]

• Logistic model to get labels:

\[ P(y_i = 1) = \frac{1}{1 + \exp(-2f_i)} \]
Results for two performance measures

- Hamming loss: \( \ell_H(y, h) = \frac{1}{m} \sum_{i=1}^{m} [y_i \neq h_i] \),
- Subset 0/1 loss: \( \ell_{0/1}(y, h) = [y \neq h] \).

<table>
<thead>
<tr>
<th>Classifiers</th>
<th>Conditional Independence</th>
<th>Conditional Dependence</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hamming loss</td>
<td>Subset 0/1 loss</td>
</tr>
<tr>
<td>BR LR</td>
<td>0.4232</td>
<td>0.6723</td>
</tr>
<tr>
<td>LP LR</td>
<td>0.4232</td>
<td>0.6725</td>
</tr>
<tr>
<td>BR LR</td>
<td>0.3470</td>
<td>0.5499</td>
</tr>
<tr>
<td>LP LR</td>
<td>0.3610</td>
<td>0.5146</td>
</tr>
</tbody>
</table>
Figure: Problem with two targets: shapes (△ vs. ○) and colors (□ vs. ■).
<table>
<thead>
<tr>
<th>CLASSIFIER</th>
<th>Hamming loss</th>
<th>Subset 0/1 loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>BR LR</td>
<td>0.2399(±.0097)</td>
<td>0.4751(±.0196)</td>
</tr>
<tr>
<td>LP LR</td>
<td>0.0143(±.0020)</td>
<td>0.0195(±.0011)</td>
</tr>
<tr>
<td>Bayes Optimal</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
### Linear + XOR synthetic data

<table>
<thead>
<tr>
<th>CLASSIFIER</th>
<th>Hamming Loss</th>
<th>Subset 0/1 Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>BR LR</td>
<td>0.2399(±.0097)</td>
<td>0.4751(±.0196)</td>
</tr>
<tr>
<td>LP LR</td>
<td>0.0143(±.0020)</td>
<td>0.0195(±.0011)</td>
</tr>
<tr>
<td><strong>BR MLRules</strong></td>
<td><strong>0.0011(±.0002)</strong></td>
<td><strong>0.0020(±.0003)</strong></td>
</tr>
<tr>
<td><strong>Bayes Optimal</strong></td>
<td><strong>0</strong></td>
<td><strong>0</strong></td>
</tr>
</tbody>
</table>
• BR LR uses two linear classifiers: cannot handle the label color ( □ vs. ■ ) – the XOR problem.
• LP LR uses four linear classifiers to solve 4-class problem ( △ , ▲ , ○ , ● ): extends the hypothesis space.
• BR MLRules uses two non-linear classifiers (based on decision rules): XOR problem is not a problem.
• There is no noise in the data.
• Easy to perform unfair comparison.
Multi-target prediction - probabilistic view

- Data are coming from distribution

\[ P(Y, X) . \]

- Since we predict the value of \( Y \) for a given object \( x \), we are interested in the conditional distribution:

\[ P(Y = y | X = x) = \frac{P(Y = y, X = x)}{P(X = x)} \]
• Data are coming from distribution

\[ P(Y, X) . \]

• Since we predict the value of \( Y \) for a given object \( x \), we are interested in the conditional distribution:

\[
P(Y = y | X = x) = \frac{P(Y = y, X = x)}{P(X = x)}
\]

• What is the most reasonable response \( y \)?
• Data are coming from distribution

\[ P(Y, X) \]

• Since we predict the value of \( Y \) for a given object \( x \), we are interested in the conditional distribution:

\[
P(Y = y | X = x) = \frac{P(Y = y, X = x)}{P(X = x)}
\]

• **What is the most reasonable response \( y \)?**
  
  ▶ \( P(Y = y | X = x) \) is the largest?
Multi-target prediction - probabilistic view

- Data are coming from distribution

\[ P(Y, X). \]

- Since we predict the value of \( Y \) for a given object \( x \), we are interested in the conditional distribution:

\[ P(Y = y | X = x) = \frac{P(Y = y, X = x)}{P(X = x)} \]

- What is the most reasonable response \( y \)?
  - \( P(Y = y | X = x) \) is the largest?
  - \( P(Y_i = y_i | X = x) \) are the largest?
Multi-target prediction - probabilistic view

- Data are coming from distribution

\[ P(Y, X) . \]

- Since we predict the value of \( Y \) for a given object \( x \), we are interested in the conditional distribution:

\[ P(Y = y|X = x) = \frac{P(Y = y, X = x)}{P(X = x)} \]

- What is the most reasonable response \( y \)?
  - \( P(Y = y|X = x) \) is the largest?
  - \( P(Y_i = y_i|X = x) \) are the largest?
  - \( \ldots \) ?
  - \( \ldots \) ?
  - \( \ldots \) ?
• Define your problem via **minimization** of a **loss** function $\ell(y, h(x))$.

• **Risk** (expected loss) of the prediction $h$ for a given $x$ is:

$$L_\ell(h, P \mid x) = \mathbb{E}_{Y \mid x} [\ell(Y, h(x))] = \sum_{y \in Y} P(Y = y \mid x) \ell(y, h(x))$$

• The risk minimization model $h^*(x)$, the so-called **Bayes classifier**, is defined for a given $x$ by

$$h^*(x) = \arg \min_{h(x)} L_\ell(h, P \mid x)$$

• Different formulations of loss functions possible:
  ▶ Set-based losses.
  ▶ Ranking-based losses.
Multi-target prediction - loss minimization view

- Subset 0/1 loss: \( \ell_{0/1}(y, h) = [y \neq h] \)

- Hamming loss: \( \ell_H(y, h) = \frac{1}{m} \sum_{i=1}^{m} [y_i \neq h_i] \)

- F-measure-based loss: \( \ell_F(y, h) = 1 - \frac{2 \sum_{i=1}^{m} y_i h_i}{\sum_{i=1}^{m} y_i + \sum_{i=1}^{m} h_i} \)

- Rank loss: \( \ell_{rnk}(y, h) = w(y) \sum_{y_i > y_j} \left( [h_i < h_j] + \frac{1}{2} [h_i = h_j] \right) \)

- ...
• Relations between losses.
• The form of the Bayes classifiers for different losses.
• How to optimize?
  ▶ Assumptions behind learning algorithms.
  ▶ Statistical consistency and regret bounds.
  ▶ Generalization bounds.
  ▶ Computational complexity.
The loss function $\ell(y, h)$ should fulfill some basic conditions:

- $\ell(y, h) = 0$ if and only if $y = h$.
- $\ell(y, h)$ is maximal when $y_i \neq h_i$ for every $i = 1, \ldots, m$.
- Should be monotonically non-decreasing with respect to the number of $y_i \neq h_i$.

- **In case of deterministic data (no-noise):** the optimal prediction should have the same form for all loss functions and the risk for this prediction should be 0.

- **In case of non-deterministic data (noise):** the optimal prediction and its risk can be different for different losses.
  ▶ The form of risk minimizers.
  ▶ Consistency of risk minimizers.
  ▶ Risk bound analysis.
  ▶ Regret bound analysis.
The risk minimizer for the Hamming loss is the marginal mode:

$$h_i^*(x) = \arg \max_{y_i \in \{0,1\}} P(Y_i = y_i \mid x), \quad i = 1, \ldots, m,$$

while for the subset 0/1 loss is the joint mode:

$$h^*(x) = \arg \max_{y \in \mathcal{Y}} P(y \mid x).$$

- Marginal mode vs. joint mode.

<table>
<thead>
<tr>
<th>$y$</th>
<th>$P(y)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0</td>
<td>0.30</td>
</tr>
<tr>
<td>0 1 1 1</td>
<td>0.17</td>
</tr>
<tr>
<td>1 0 1 1</td>
<td>0.18</td>
</tr>
<tr>
<td>1 1 0 1</td>
<td>0.17</td>
</tr>
<tr>
<td>1 1 1 0</td>
<td>0.18</td>
</tr>
</tbody>
</table>

Marginal mode: 1 1 1 1
Joint mode: 0 0 0 0
Consistency of risk minimizers and risk bounds

- The risk minimizers for $\ell_H$ and $\ell_{0/1}$ are **equivalent**, 

$$h^*_H(x) = h^*_{0/1}(x),$$

under specific conditions, for example, when:

- Targets $Y_1, \ldots, Y_m$ are conditionally independent, i.e,

$$P(Y|x) = \prod_{i=1}^{m} P(Y_i|x).$$

- The probability of the joint mode satisfies

$$P(h^*_{0/1}(x)|x) > 0.5.$$

- The following bounds hold for any $P(Y|x)$ and $h$:

$$\frac{1}{m} L_{0/1}(h, P|x) \leq L_H(h, P|x) \leq L_{0/1}(h, P|x)$$
• The previous results may suggest that one of the loss functions can be used as a proxy (surrogate) for the other:
  ▶ For some situations both risk minimizers coincide.
  ▶ One can provide mutual bounds for both loss functions.
The previous results may suggest that one of the loss functions can be used as a proxy (surrogate) for the other:

- For some situations both risk minimizers coincide.
- One can provide mutual bounds for both loss functions.

However, the regret analysis of the worst case shows that minimization of the subset 0/1 loss may result in a large error for the Hamming loss and vice versa.
• The \textbf{regret} of a classifier with respect to $\ell$ is defined as:

$$
\text{Reg}_\ell(h, P) = L_\ell(h, P) - L_\ell(h^*_\ell, P),
$$

where $h^*_\ell$ is the Bayes classifier for a given loss $\ell$.

• Regret measures how worse is $h$ by comparison with the optimal classifier for a given loss.

• To simplify the analysis we will consider the conditional regret:

$$
\text{Reg}_\ell(h, P \mid x) = L_\ell(h, P \mid x) - L_\ell(h^*_\ell, P \mid x).
$$

• We will analyze the regret between:
  
  ▶ the Bayes classifier for Hamming loss $h^*_H$
  ▶ the Bayes classifier for subset 0/1 loss $h^*_{0/1}$

  with respect to both functions.

• It is a bit an unusual analysis.
The following **upper bound** holds:

\[
\text{Reg}_{0/1}(\mathbf{h}_H^*, P \mid x) = L_{0/1}(\mathbf{h}_H^*, P \mid x) - L_{0/1}(\mathbf{h}_{0/1}^*, P \mid x) < 0.5
\]

Moreover, this **bound is tight**.

**Example**:

<table>
<thead>
<tr>
<th>(y)</th>
<th>(P(y))</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0</td>
<td>0.02</td>
</tr>
<tr>
<td>0 0 1 1</td>
<td>0.49</td>
</tr>
<tr>
<td>1 1 0 0</td>
<td>0.49</td>
</tr>
</tbody>
</table>

Marginal mode: 0 0 0 0

Joint mode: 0 0 1 1 or 1 1 0 0
Regret analysis

- The following **upper bound** holds $m > 3$:

$$\text{Reg}_H(h^*_{0/1}, P \mid x) = L_H(h^*_{0/1}, P \mid x) - L_H(h^*_H, P \mid x) < \frac{m - 2}{m + 2}$$

- Moreover, this **bound is tight**.

- **Example:**

<table>
<thead>
<tr>
<th>$y$</th>
<th>$P(y)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0</td>
<td>0.170</td>
</tr>
<tr>
<td>0 1 1 1</td>
<td>0.166</td>
</tr>
<tr>
<td>1 0 1 1</td>
<td>0.166</td>
</tr>
<tr>
<td>1 1 0 1</td>
<td>0.166</td>
</tr>
<tr>
<td>1 1 1 0</td>
<td>0.166</td>
</tr>
<tr>
<td>1 1 1 1</td>
<td>0.166</td>
</tr>
</tbody>
</table>

Marginal mode: 1 1 1 1
Joint mode: 0 0 0 0
• **Summary:**
  ▶ The risk minimizers of Hamming and subset 0/1 loss are different: marginal mode vs. joint mode.
  ▶ Under specific conditions, these two risk minimizers are equivalent.
  ▶ The risks of these loss functions are mutually upper bounded.
  ▶ Minimization of the subset 0/1 loss may cause a high regret for the Hamming loss and vice versa.
• Both are commonly used.
• Hamming loss:
  ▶ Not too many labels.
  ▶ Well-balanced labels.
  ▶ **Application**: Gene function prediction.
• Subset 0/1 loss:
  ▶ Very restrictive.
  ▶ Small number of labels.
  ▶ Low noise problems.
  ▶ **Application**: Prediction of diseases of a patient.
What does the above analysis change in interpretation of the results of the starting examples?

BR trains for each label an independent classifier:
- Does BR assume label independence?
- Is it consistent for any loss function?
- What is its complexity?

LP treats each label combination as a new meta-class in multi-class classification:
- What are the assumptions behind LP?
- Is it consistent for any loss function?
- What is its complexity?
• Binary relevance (BR)
  ▶ BR is consistent for Hamming loss without any additional assumption on label (in)dependence.
  ▶ If this would not be true, then we could not optimally solve binary classification problems!!!
  ▶ For other losses, one should probably take additional assumptions:
    • For subset 0/1 loss: label independence, high probability of the joint mode (> 0.5), ...
  ▶ Learning and inference is linear in $m$ (however, faster algorithms exist).
• Label powerset (LP)
  ▶ LP is **consistent** for the subset 0/1 loss.
  ▶ In its basic formulation it is **not consistent** for Hamming loss.
  ▶ However, if used with probabilistic multi-class classifier, it estimates the joint conditional distribution for a given $x$: inference for any loss would be then possible.
  ▶ Similarly, by reducing to cost-sensitive multi-class classification LP can be used with **almost any loss function**.
  ▶ LP may gain from the implicit expansion of the feature or hypothesis space.
  ▶ Unfortunately, learning and inference is basically **exponential** in $m$ (however, this complexity is constrained by the number of training examples).
• The loss functions, like Hamming loss or subset 0/1 loss, often referred to as task losses, are usually neither convex nor differentiable.

• Therefore learning is a hard optimization problem.

• Two approaches try to make this task easier
  ▶ Reduction.
  ▶ Structured loss minimization.

• Two phases in solving multi-target prediction problems:
  ▶ Learning: Estimate parameters of the scoring function $f(x, y)$.
  ▶ Inference: Use the scoring function $f(x, y)$ to classify new instances by finding the best $y$ for a given $x$. 
Reduction

\[ \{(x, y)\}_{i=1}^{n} \]

\[ (x, y) \rightarrow (x', y') \]

\[ f(x', y') \]

Inference \[ x \rightarrow \text{Inference} \rightarrow \hat{y} \]

- **Reduce** the original problem into problems of simpler type, for which efficient algorithmic solutions are available.
- Reduction to one or a sequence of problems.
- Plug-in rule classifiers.
- BR and LP already discussed.
Structured loss minimization

\[ \{(x, y)\}_{i=1}^{n} \]

- Replace the task loss by a \textbf{surrogate loss} that is easier to cope with.
- Surrogate loss is typically a differentiable approximation of the task loss or a convex upper bound of it.

\[
\min \tilde{\ell}(y, x, f)
\]

\[ f(x, y) \]

\[ x \rightarrow \text{Inference} \rightarrow \hat{y} \]
Statistical consistency

- Analysis of algorithms in terms of their infinite sample performance.\(^2\)
- We say that a proxy loss \(\tilde{\ell}\) is consistent with the task loss \(\ell\) when the following holds:

\[
\text{Reg}_{\tilde{\ell}}(h, P) \to 0 \Rightarrow \text{Reg}_{\ell}(h, P) \to 0.
\]

- The definition concerns both structured loss minimization and reduction algorithms
  - Structured loss minimization: \(\tilde{\ell} = \) surrogate loss.
  - Reduction: \(\tilde{\ell} = \) loss used in the reduced problem.
- We already know: Hamming loss is not a consistent proxy for subset 0/1 loss and vice versa.

---

Algorithms

- Conditional random fields (CRFs)
- Structured support vector machines (SVMs)
- Probabilistic classifier chains (PCC)
Conditional random fields

- Conditional random fields (CRFs) extend logistic regression.\(^3\)
- CRFs model the conditional joint distribution of \(Y\) by:

\[
P(y \mid x) = \frac{1}{Z(x)} \exp(f(x, y))
\]

- \(f(x, y)\) is a scoring function that models the adjustment between \(y\) and \(x\).
- \(Z(x)\) is a normalization constant:

\[
Z(x) = \sum_{y \in Y} \exp(f(x, y))
\]

---

Conditional random fields

- The negative log-loss is used as a surrogate:

\[
\ell_{\log}(y, x, f) = - \log P(y|x) = \log \left( \sum_{y \in Y} \exp(f(x, y)) \right) - f(x, y)
\]

- Regularized log-likelihood optimization:

\[
\min_{f} \frac{1}{n} \sum_{i=1}^{n} \ell_{\log}(y, x, f) + \lambda J(f)
\]

- Inference for a new instance \(x\):

\[
h(x) = \arg \max_{y \in Y} P(y \mid x)
\]
Conditional random fields

- Similar to LP, but with an internal structure of classes and scoring function $f(x, y)$.
- Convex optimization problem, but depending on the structure of $f(x, y)$ its solution can be hard.
- Similarly, the inference (also known as decoding problem) is hard in the general case.
- Tailored for the subset 0/1 loss (estimation of the joint mode).
- Different forms of $f(x, y)$. 
• Let $f(x, y)$ be defined as:

$$f(x, y) = \sum_{i=1}^{m} f_i(x, y_i)$$

• In this case, we have:

$$P(y|x) = \frac{\exp(f(x, y))}{\sum_{y \in Y} \exp(f(x, y))} = \prod_{m} \frac{\exp(f_i(x, y_i))}{\sum_{y_i} \exp(f_i(x, y_i))} = \prod_{m} P(y_i|x)$$

• Optimal for Hamming loss!!!

• The structure of $f(x, y)$ is connected to the loss function.
Conditional random fields

• Let $f(x, y)$ be defined as:

$$f(x, y) = \sum_{i=1}^{m} f_i(x, y_i)$$

• In this case, we have:

$$P(y | x) = \frac{\exp(f(x, y))}{\sum_{y \in Y} \exp(f(x, y))} = \frac{\exp(\sum_{i=1}^{m} f_i(x, y_i))}{\sum_{y \in Y} \exp(\sum_{i=1}^{m} f_i(x, y_i))} = \frac{\prod_{i=1}^{m} \exp(f_i(x, y_i))}{\sum_{y \in Y} \prod_{i=1}^{m} \exp(f_i(x, y_i))} = \frac{\prod_{i=1}^{m} \exp(f_i(x, y_i))}{\prod_{i=1}^{m} \sum_{y_i} \exp(f_i(x, y_i))}$$

• Optimal for Hamming loss!!
• Let $f(x, y)$ be defined as:

$$f(x, y) = \sum_{i=1}^{m} f_i(x, y_i)$$

• In this case, we have:

$$P(y \mid x) = \frac{\exp(f(x, y))}{\sum_{y \in Y} \exp(f(x, y))} = \frac{\exp(\sum_{i=1}^{m} f_i(x, y_i))}{\sum_{y \in Y} \exp(\sum_{i=1}^{m} f_i(x, y_i))} = \frac{\prod_{i=1}^{m} \exp(f_i(x, y_i))}{\sum_{y \in Y} \prod_{i=1}^{m} \exp(f_i(x, y_i))} = \frac{\prod_{i=1}^{m} \exp(f_i(x, y_i))}{\prod_{i=1}^{m} \sum_{y_i} \exp(f_i(x, y_i))} = \prod_{i=1}^{m} P(y_i \mid x)$$
• Let $f(x, y)$ be defined as:

$$f(x, y) = \sum_{i=1}^{m} f_i(x, y_i)$$

• In this case, we have:

$$P(y | x) = \frac{\exp(f(x, y))}{\sum_{y \in \mathcal{Y}} \exp(f(x, y))} = \frac{\exp(\sum_{i=1}^{m} f_i(x, y_i))}{\sum_{y \in \mathcal{Y}} \exp(\sum_{i=1}^{m} f_i(x, y_i))}$$

$$= \frac{\prod_{i=1}^{m} \exp(f_i(x, y_i))}{\sum_{y \in \mathcal{Y}} \prod_{i=1}^{m} \exp(f_i(x, y_i))} = \frac{\prod_{i=1}^{m} \exp(f_i(x, y_i))}{\prod_{i=1}^{m} \sum_{y_i} \exp(f_i(x, y_i))}$$

$$= \prod_{i=1}^{m} P(y_i | x)$$

• **Optimal for Hamming loss!!!**

• The structure of $f(x, y)$ is connected to the loss function.
A different form of $f(x, y)$:

$$f(x, y) = \sum_{i=1}^{m} f_i(x, y_i) + \sum_{y_k, y_l} f_{k,l}(y_k, y_l)$$

Models pairwise interactions, ...
• A different form of $f(x, y)$:

$$f(x, y) = \sum_{i=1}^{m} f_i(x, y_i) + \sum_{y_k, y_l} f_{k,l}(y_k, y_l)$$

• Models pairwise interactions, ... but in the **conditional sense**: 
Conditional random fields

- A different form of $f(x, y)$:
  
  $$f(x, y) = \sum_{i=1}^{m} f_i(x, y_i) + \sum_{y_k, y_l} f_{k,l}(y_k, y_l)$$

- Models pairwise interactions, ... but in the **conditional sense**:
  
  - Assume that $x$ is not given:
    
    $$P(y) = \frac{\exp(\sum_i f_i(y_i) + \sum_{y_k, y_l} f_{k,l}(y_k, y_l))}{\sum_{y \in Y} \exp(\sum_i f_i(y_i) + \sum_{y_k, y_l} f_{k,l}(y_k, y_l))}$$
  
  - **Models a prior joint distribution over labels!!!**
  
  - The prior cannot be easily factorized to marginal probabilities.

- Should work better for subset 0/1 loss than for Hamming loss.
• CRFs do not directly take the task loss into account.
• We would like to have a method that could be used with any loss . . .

---

Structured loss minimization

• CRFs do not directly take the task loss into account.
• We would like to have a method that could be used with any loss . . .
• Structured support vector machines (SSVMs) extends the idea of large-margin classifiers to structured output prediction problems.\textsuperscript{4}

Structured support vector machines

- SSVMs use, similarly to CRFs, a scoring function $f(x, y)$.
- They minimize the **structured hinge loss**:
  \[
  \tilde{\ell}_h(y, x, f) = \max_{y' \in Y} \{\ell(y, y') + f(x, y')\} - f(x, y).
  \]
- Task loss $\ell(y, y')$ is used for margin rescaling.
- Regularized optimization problem:
  \[
  \min_f \frac{1}{n} \sum_{i=1}^{n} \tilde{\ell}_h(y, x, f) + \lambda J(f)
  \]
- Predict according to:
  \[
  h(x) = \arg \max_{y \in Y} f(x, y).
  \]
Structured support vector machines

• Convex optimization problem with linear constraints.
• An exponential number of constraints $\rightarrow$ Cutting-plane algorithms.
• The $\text{arg max}$ problem is hard for general structures.
• Different forms of $f(x, y)$. 
Structured support vector machines

• Let $f(x, y)$ be defined as:

$$f(x, y) = \sum_{i=1}^{m} f_i(x, y_i)$$

• Let us use it with the Hamming loss:

$$\tilde{\ell}_h(y, x, f) = \max_{y' \in Y} \{\ell_H(y, y') + f(x, y')\} - f(x, y)$$

Structured hinge loss decomposes to hinge loss for each label.

Consistent for the Hamming loss.

---

Structured support vector machines

- Let $f(x, y)$ be defined as:
  $$f(x, y) = \sum_{i=1}^{m} f_i(x, y_i)$$

- Let us use it with the Hamming loss:
  $$\tilde{\ell}_h(y, x, f) = \max_{y' \in \mathcal{Y}} \{\ell_H(y, y') + f(x, y')\} - f(x, y)$$
  $$= \max_{y' \in \mathcal{Y}} \left\{\sum_i [y_i \neq y'_i] + \sum_i f_i(x, y'_i)\right\} - \sum_i f_i(x, y_i)$$

---

Structured support vector machines

- Let $f(x, y)$ be defined as:

$$f(x, y) = \sum_{i=1}^{m} f_i(x, y_i)$$

- Let us use it with the Hamming loss:

$$\tilde{\ell}_h(y, x, f) = \max_{y' \in \mathcal{Y}} \{\ell_H(y, y') + f(x, y')\} - f(x, y)$$

$$= \max_{y' \in \mathcal{Y}} \left\{ \sum_i [y_i \neq y'_i] + \sum_i f_i(x, y'_i) \right\} - \sum_i f_i(x, y_i)$$

$$= \sum_i \max_{y'_i} \{[y_i \neq y'_i] + f_i(x, y'_i) - f_i(x, y_i)\}$$

---

Structured support vector machines

- Let $f(x, y)$ be defined as:

$$f(x, y) = \sum_{i=1}^{m} f_i(x, y_i)$$

- Let us use it with the Hamming loss:

$$\tilde{\ell}_h(y, x, f) = \max_{y' \in \mathcal{Y}} \left\{ \ell_H(y, y') + f(x, y') \right\} - f(x, y)$$

$$= \max_{y' \in \mathcal{Y}} \left\{ \sum_i [y_i \neq y'_i] + \sum_i f_i(x, y'_i) \right\} - \sum_i f_i(x, y_i)$$

$$= \sum_i \max_{y'_i} \left\{ [y_i \neq y'_i] + f_i(x, y'_i) - f_i(x, y_i) \right\}$$

- **Structured hinge loss decomposes to hinge loss for each label.**

- Consistent for the Hamming loss.

---

Structured support vector machines

- The form $f(x, y)$ that models pairwise interactions:

$$f(x, y) = \sum_{i=1}^{m} f_i(x, y_i) + \sum_{y_k, y_l} f_{k,l}(y_k, y_l)$$

- How important is the pairwise interaction part for different task losses?

- For a general form of $f(x, y)$, SSVMs are inconsistent for Hamming loss.\(^6\)

- There are more results of this type.\(^7\)

---


Structured support vector machines

Table: SSVMs with pairwise term\(^8\) vs. BR with LR\(^9\).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>SSVM</th>
<th>BR</th>
<th>LR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scene</td>
<td>0.101±.003</td>
<td>0.102±.003</td>
<td></td>
</tr>
<tr>
<td>Yeast</td>
<td>0.202±.005</td>
<td>0.199±.005</td>
<td></td>
</tr>
<tr>
<td>Synth1</td>
<td>0.069±.001</td>
<td>0.067±.002</td>
<td></td>
</tr>
<tr>
<td>Synth2</td>
<td>0.058±.001</td>
<td>0.084±.001</td>
<td></td>
</tr>
</tbody>
</table>

- There is almost no difference between both algorithms.

---

\(^8\) Thomas Finley and Thorsten Joachims. Training structural SVMs when exact inference is intractable. In *ICML*. Omnipress, 2008

SSVMs vs. CRFs

• SSVMs and CRFs are quite similar to each other:

\[
\tilde{\ell}_{\log}(y, x, f) = \log \left( \sum_{y \in Y} \exp(f(x, y)) \right) - f(x, y)
\]

\[
\tilde{\ell}_h(y, x, f) = \max_{y' \in Y} \{ \ell(y, y') + f(x, y') \} - f(x, y)
\]

• The main differences are:
  ▶ max vs. soft-max
  ▶ margin vs. no-margin

• Many works on incorporating margin in CRFs.\(^\text{10}\)

---


• Probabilistic classifier chains (PCCs)\textsuperscript{11} similarly to CRFs estimate the joint conditional distribution $P(Y \mid x)$.
• Their idea is to repeatedly apply the \textbf{product rule of probability}:

$$P(Y = y \mid x) = \prod_{i=1}^{m} P(Y_i = y_i \mid x, y_1, \ldots, y_{i-1}).$$

• They follow a reduction to a sequence of subproblems:

$$(x, y) \rightarrow (x' = (x, y_1, \ldots, y_{i-1}), y = y_i), \quad i = 1, \ldots, m$$

• Their additional advantage is that one can easily sample from the estimated distribution.

• Learning of PCCs relies on constructing probabilistic classifiers for estimating

\[ P(Y_i = y_i \mid \mathbf{x}, y_1, \ldots, y_{i-1}) , \]

independently for each \( i = 1, \ldots, m \).

• One can use scoring functions \( f_i(\mathbf{x}', y_i) \) and use logistic transformation.

• By using the linear models, the overall scoring function takes the form:

\[
f(\mathbf{x}, y) = \sum_{i=1}^{m} f_i(\mathbf{x}, y_i) + \sum_{y_k, y_l} f_{k,l}(y_k, y_l)
\]
• Inference relies on exploiting a probability tree being the result of PCC:

\[ P(y_1 = 0 | x) = 0.4 \]
\[ P(y_1 = 1 | x) = 0.6 \]
\[ P(y_2 = 0 | y_1 = 0, x) = 0.0 \]
\[ P(y_2 = 1 | y_1 = 0, x) = 1.0 \]
\[ P(y_2 = 0 | y_1 = 1, x) = 0.4 \]
\[ P(y_2 = 1 | y_1 = 1, x) = 0.6 \]

\[ P(y=(0, 0) | x) = 0 \]
\[ P(y=(0, 1) | x) = 0.4 \]
\[ P(y=(1, 0) | x) = 0.24 \]
\[ P(y=(1, 1) | x) = 0.36 \]

• For subset 0/1 loss one needs to find \( h(x) = \arg \max_{y \in Y} P(y | x) \).

• Greedy and approximate search techniques with guarantees exist.\(^\text{12}\)

---
• Inference relies on exploiting a probability tree being the result of PCC:

![Probability Tree Diagram]

- Other losses: compute the prediction on a sample from $P(Y | x)$.\(^{12}\)
- Sampling can be easily performed by using the probability tree.

Probabilistic classifier chains

Table: PCC vs. SSVMs on Hamming loss and PCC vs. BR on subset 0/1 loss.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>PCC Hamming loss</th>
<th>SSVM Best Hamming loss</th>
<th>PCC subset 0/1 loss</th>
<th>BR subset 0/1 loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scene</td>
<td>0.104±.004</td>
<td>0.101±.003</td>
<td>0.385±.014</td>
<td>0.509±.014</td>
</tr>
<tr>
<td>Yeast</td>
<td>0.203±.005</td>
<td>0.202±.005</td>
<td>0.761±.014</td>
<td>0.842±.012</td>
</tr>
<tr>
<td>Synth1</td>
<td>0.067±.001</td>
<td>0.069±.001</td>
<td>0.239±.006</td>
<td>0.240±.006</td>
</tr>
<tr>
<td>Synth2</td>
<td>0.000±.000</td>
<td>0.058±.001</td>
<td>0.000±.000</td>
<td>0.832±.004</td>
</tr>
<tr>
<td>Reuters</td>
<td>0.060±.002</td>
<td>0.045±.001</td>
<td>0.598±.009</td>
<td>0.689±.008</td>
</tr>
<tr>
<td>Mediamill</td>
<td>0.172±.001</td>
<td>0.182±.001</td>
<td>0.885±.003</td>
<td>0.902±.003</td>
</tr>
</tbody>
</table>
Multi-label ranking

Serena romps to fifth Wimbledon title against brave Radwanska
By Paul Graham, CNN
July 7, 2012 -- Updated 2220 GMT (0620 HKT)

Women's singles Wimbledon Championship

STORY HIGHLIGHTS
• Serena Williams wins fifth Wimbledon crown
• American beats Agnieszka Radwanska of Poland 6-1 5-7 6-2
• Radwanska battled respiratory

(CNN) -- Serena Williams fended off a stirring fightback from Agnieszka Radwanska to win her fifth Wimbledon singles title with a 6-1 5-7 6-2 victory Saturday.

It was the 30-year-old American's 14th grand slam crown and her first since winning at the All England Club in 2010, but Poland's Radwanska made her fight every inch of the way.

Multi-label classification

- politics: 0
- economy: 0
- business: 0
- sport: 1
- tennis: 1
- soccer: 0
- show-business: 0
- celebrities: 1
- England: 1
- USA: 1
- Poland: 1
- Lithuania: 0
Serena romps to fifth Wimbledon title against brave Radwanska

By Paul GitTINGS, CNN
July 7, 2012 – Updated 2220 GMT (0620 HKT)

Williams and Radwanska shake hands after the match on Saturday.

Women's singles Wimbledon Championship

STORY HIGHLIGHTS
• Serena Williams wins fifth Wimbledon crown
• American beats Agnieszka Radwanska of Poland 6-1 5-7 6-2
• Radwanska battles respiratory...
Multi-label ranking

- **Ranking loss**:  

\[
\ell_{\text{rnk}}(y, h) = w(y) \sum_{(i,j): y_i > y_j} \left( [h_i(x) < h_j(x)] + \frac{1}{2} [h_i(x) = h_j(x)] \right),
\]

where \( w(y) < w_{\text{max}} \) is a weight function.

| \( x \) | 4.0 | 2.5 | 1 | 0 | \( \ldots \) | 0 | \( h_2 > h_1 > \ldots > h_m \) | \( Y_1 \) | \( Y_2 \) | ... | \( Y_m \) |
Multi-label ranking

- **Ranking loss:**

\[
\ell_{\text{rnk}}(\mathbf{y}, \mathbf{h}) = w(\mathbf{y}) \sum_{(i,j): y_i > y_j} \left( \mathbb{I}[h_i(x) < h_j(x)] + \frac{1}{2} \mathbb{I}[h_i(x) = h_j(x)] \right),
\]

where \(w(\mathbf{y}) < w_{\text{max}}\) is a weight function.

The weight function \(w(\mathbf{y})\) is usually used to normalize the range of rank loss to \([0, 1]\):

\[
w(\mathbf{y}) = \frac{1}{n_+ n_-},
\]

i.e., it is equal to the inverse of the total number of pairwise comparisons between labels.
• The most intuitive approach is to use pairwise convex surrogate losses of the form

$$\tilde{\ell}_\phi(y, h) = \sum_{(i,j): y_i > y_j} w(y) \phi(h_i - h_j),$$

where $\phi$ is

- an exponential function (BoosTexter)$^{13}$: $\phi(f) = e^{-f}$,
- logistic function (LLLR)$^{14}$: $\phi(f) = \log(1 + e^{-f})$,
- or hinge function (RankSVM)$^{15}$: $\phi(f) = \max(0, 1 - f)$.

---


Multi-label ranking

- This approach is, however, **inconsistent** for the most commonly used convex surrogates.\textsuperscript{16}
- The **consistent** classifier can be, however, obtained by using univariate loss functions\textsuperscript{17} \ldots


\textsuperscript{17} K. Dembczynski, W. Kotlowski, and E. Hüllermeier. Consistent multilabel ranking through univariate losses. In *ICML*, 2012
• The Bayes ranker can be obtained by sorting labels according to:

\[
\Delta_i^1 = \sum_{y : y_i = 1} w(y) P(y \mid x).
\]

• For \( w(y) \equiv 1 \), \( \Delta_i^u \) reduces to **marginal probabilities** \( P(Y_i = u \mid x) \).

• The solution can be obtained with BR or its weighted variant in a general case.
• Consider the sum of \textit{univariate (weighted)} losses:

\[ \tilde{\ell}_{\text{exp}}(y, h) = w(y) \sum_{i=1}^{m} e^{-(2y_i-1)h_i}, \]

\[ \tilde{\ell}_{\text{log}}(y, h) = w(y) \sum_{i=1}^{m} \log \left( 1 + e^{-(2y_i-1)h_i} \right). \]

• The risk minimizer of these losses is:

\[ h_i^*(x) = \frac{1}{c} \log \frac{\Delta^1_i}{\Delta^0_i} = \frac{1}{c} \log \frac{\Delta^1_i}{W - \Delta^1_i}, \]

which is a strictly increasing transformation of \( \Delta^1_i \), where

\[ W = \mathbb{E}[w(Y) \mid x] = \sum_{y} w(y)P(y \mid x). \]
• **Vertical reduction**: Solving $m$ independent classification problems.

• Standard algorithms, like AdaBoost and logistic regression, can be adapted to this setting.

• AdaBoost.MH follows this approach for $w = 1$.\(^{18}\)

• Besides its **simplicity** and **efficiency**, this approach is **consistent** (regret bounds have also been derived).\(^{19}\)

---


\(^{19}\) K. Dembczynski, W. Kotlowski, and E. Hüllermeier. Consistent multilabel ranking through univariate losses. In *ICML*, 2012
Weighted binary relevance

Figure: WBR LR vs. LLLR. Left: independent data. Right: dependent data.

- **Label independence**: the methods perform more or less en par.
- **Label dependence**: WBR shows small but consistent improvements.
Benchmark data

Table: WBR-AdaBoost vs. AdaBoost.MR (left) and WBR-LR vs LLLR (right).

<table>
<thead>
<tr>
<th>DATASET</th>
<th>AB.MR</th>
<th>WBR-AB</th>
<th>LLLR</th>
<th>WBR-LR</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMAGE</td>
<td>0.2081</td>
<td>0.2041</td>
<td>0.2047</td>
<td>0.2065</td>
</tr>
<tr>
<td>EMOTIONS</td>
<td>0.1703</td>
<td>0.1699</td>
<td>0.1743</td>
<td>0.1657</td>
</tr>
<tr>
<td>SCENE</td>
<td>0.0720</td>
<td>0.0792</td>
<td>0.0861</td>
<td>0.0793</td>
</tr>
<tr>
<td>YEAST</td>
<td>0.2072</td>
<td>0.1820</td>
<td>0.1728</td>
<td>0.1736</td>
</tr>
<tr>
<td>MEDIAMILL</td>
<td>0.0665</td>
<td>0.0609</td>
<td>0.0614</td>
<td>0.0472</td>
</tr>
</tbody>
</table>

- WBR is at least competitive to state-of-the-art algorithms defined on pairwise surrogates.
Maximization of the F-measure

- Applications: Information retrieval, document tagging, and NLP.

- JRS 2012 Data Mining Competition: Indexing documents from MEDLINE or PubMed Central databases with concepts from the Medical Subject Headings ontology.
Maximization of the F-measure

- The \( F_\beta \)-measure-based loss function (\( F_\beta \)-loss):

\[
\ell_{F_\beta}(y, h(x)) = 1 - F_\beta(y, h(x))
\]

\[
= 1 - \frac{(1 + \beta^2) \sum_{i=1}^{m} y_i h_i(x)}{\beta^2 \sum_{i=1}^{m} y_i + \sum_{i=1}^{m} h_i(x)} \in [0, 1].
\]

- Provides a **better balance** between relevant and irrelevant labels.
- However, it is **not easy** to optimize.
SSVMs for $F_\beta$-based loss

- SSVMs can be used to minimize $F_\beta$-based loss
- Rescale the margin by $\ell_F(y, y')$.

**RML**

No label interactions:

$$f(y, x) = \sum_{i=1}^{m} f_i(y_i, x)$$

Quadratic learning and linear prediction

**SML**

Submodular interactions:

$$f(y, x) = \sum_{i=1}^{m} f_i(y_i, x) + \sum_{y_k, y_l} f_{k,l}(y_k, y_l)$$

More complex (graph-cut and approximate algorithms)

- Both are inconsistent.
Plug-in rule approach

- Plug estimates of required parameters into the Bayes classifier.
  
  \[ h^* = \arg \min_{h \in \mathcal{Y}} \mathbb{E} [\ell_{F \beta}(\mathbf{Y}, h)] \]
  
  \[ = \arg \max_{h \in \mathcal{Y}} \sum_{\mathbf{y} \in \mathcal{Y}} \mathcal{P}(\mathbf{y}) \frac{(\beta + 1) \sum_{i=1}^{m} y_i h_i}{\beta^2 \sum_{i=1}^{m} y_i + \sum_{i=1}^{m} h_i} \]
  
  - **No closed form** solution for this optimization problem.
  - The problem **cannot** be solved **naively** by brute-force search:
    - This would require to check all possible combinations of labels \(2^m\)
    - To sum over \(2^m\) number of elements for computing the expected value.
    - The number of parameters to be estimated \(\mathcal{P}(\mathbf{y})\) is \(2^m\).
• Approximation needed?

\[ P(y_i = 1) \]
\[ P(y_i = 1, s = \sum_i y_i) \]

Inference based on dynamic programming.

Inference based on matrix multiplication and top \( k \) selection.

Reduction to LR for each label.

Reduction to multinomial LR for each label.


Plug-in rule approach

- **Approximation needed?** Not really. The exact solution is **tractable**!

---


Plug-in rule approach

• **Approximation needed?** Not really. The exact solution is tractable!

**LFP:**
- Assumes label independence.
- Linear number of parameters: $P(y_i = 1)$.
- Inference based on dynamic programming.\(^{21}\)
- Reduction to LR for each label.

**EFP:**
- No assumptions.
- Quadratic number of parameters: $P(y_i = 1, s = \sum_i y_i)$.
- Inference based on matrix multiplication and top $k$ selection.\(^{22}\)
- Reduction to multinomial LR for each label.

• **EFP is consistent.**\(^{23}\)

---


Maximization of the F-measure
• We did not discuss:
  ◁ Individual target view.
  ◁ Hierarchical multi-label classification.
  ◁ Other task losses.
  ◁ ...

• Main challenges:
  ◁ Learning and inference algorithms for any task loss and output structure.
  ◁ Consistency of the algorithms.
  ◁ Large-scale datasets: number of instances, features, and labels.
Conclusions

- Take-away message:
  - Two main challenges: loss minimization and target dependence.
  - Two views: the individual target and the joint target view.
  - The joint target view: structured loss minimization and reduction.
  - Proper modeling of target dependence for different loss functions.
  - Be careful with empirical evaluations.
  - Independent models can perform quite well.

- For more check:
  
  http://www.cs.put.poznan.pl/kdembczynski

Many thanks to Eyke and Willem for collaboration on this tutorial and Arek for a help in preparing the slides.

This project is partially supported by the Foundation of Polish Science under the Homing Plus programme, co-financed by the European Regional Development Fund.