Multi-Label Classification

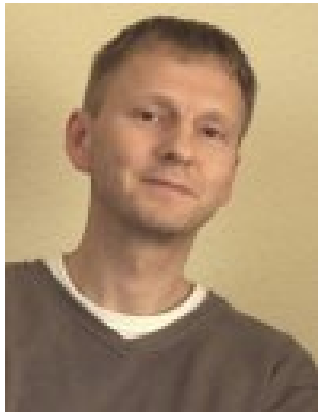
Krzysztof Dembczyński

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

PL-SIGML, Warszawa, November 12-13, 2013

Multi-label classification

Many thanks to Willem Waegeman and Eyke Hüllermeier for collaborating on this topic.



Multi-label classification

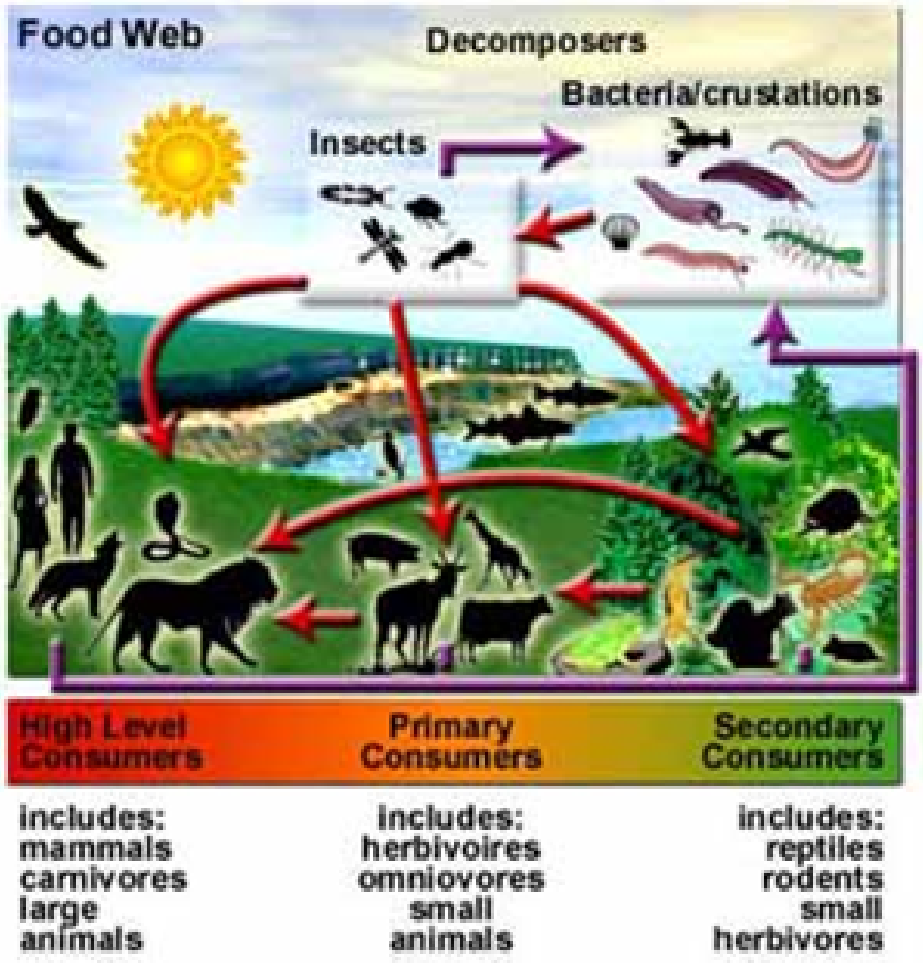
- A classification problem in which we consider **more than one** target variable.

Image annotation/retrieval

Target 1:	cloud	yes/no
Target 2:	sky	yes/no
Target 3:	tree	yes/no
...	...	...

Ecology

- Prediction of the presence or absence of species.



Multi-label classification

- Training data: $\{(\boldsymbol{x}_1, \boldsymbol{y}_1), (\boldsymbol{x}_2, \boldsymbol{y}_2), \ldots, (\boldsymbol{x}_n, \boldsymbol{y}_n)\}$, $\boldsymbol{y}_i \in \mathcal{Y} = \{0, 1\}^m$.
- **Predict** a vector $\boldsymbol{y} = (y_1, y_2, \ldots, y_m)$ for a given $\boldsymbol{x}$.

	X_1	X_2	Y_1	Y_2	$\ldots$	Y_m
$\boldsymbol{x}_1$	5.0	4.5	1	1		0
$\boldsymbol{x}_2$	2.0	2.5	0	1		0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$
$\boldsymbol{x}_n$	3.0	3.5	0	1		1
$\boldsymbol{x}$	4.0	2.5	?	?		?

Multi-label classification

- Training data: $\{(\boldsymbol{x}_1, \boldsymbol{y}_1), (\boldsymbol{x}_2, \boldsymbol{y}_2), \ldots, (\boldsymbol{x}_n, \boldsymbol{y}_n)\}$, $\boldsymbol{y}_i \in \mathcal{Y} = \{0, 1\}^m$.
- **Predict** a vector $\boldsymbol{y} = (y_1, y_2, \ldots, y_m)$ for a given $\boldsymbol{x}$.

	X_1	X_2	Y_1	Y_2	$\ldots$	Y_m
$\boldsymbol{x}_1$	5.0	4.5	1	1		0
$\boldsymbol{x}_2$	2.0	2.5	0	1		0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$
$\boldsymbol{x}_n$	3.0	3.5	0	1		1
$\boldsymbol{x}$	4.0	2.5	1	1		0

Multi-label classification

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

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

- **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

$$\ell(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x}))$$

- **Main question:**
  - Can we improve over independent models trained for each target?
- **Two views:**
  - The individual-target view
  - The joint-target view

The individual 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?

The joint target view

- 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 and joint target view

- The individual target view:
  - Goal: predict a value of y_i using $\boldsymbol{x}$ and any available information on other targets y_js.
  - 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 $\boldsymbol{y}$ using $\boldsymbol{x}$.
  - Multivariate distribution of $\boldsymbol{y}$.
  - The problem is defined through multivariate losses $\ell(\boldsymbol{y}, \hat{\boldsymbol{y}})$.
  - The problem is not easily decomposable over the targets.
  - Domain of $\boldsymbol{y}$ 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,multi-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

Target interdependences

- **Marginal** and **conditional dependence**:

$$P(\boldsymbol{Y}) \neq \prod_{i=1}^{m} P(Y_i) \qquad P(\boldsymbol{Y} \mid \boldsymbol{x}) \neq \prod_{i=1}^{m} P(Y_i \mid \boldsymbol{x})$$

marginal (in)dependence $\not\Leftrightarrow$ conditional (in)dependence

Target interdependences

- **Model similarities**:

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

Similarities in the structural parts $g_i(\boldsymbol{x})$ of the models.

Target interdependences

- **Structure** imposed (domain knowledge) on targets
  - Chains,
  - Hierarchies,
  - General graphs,
  - ...

Target interdependences

- **Interdependence** vs. **hypothesis** and **feature space**:
  - Regularization constraints 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

$$L = \sum_{i=1}^{n} \ell(\boldsymbol{y}_i, \boldsymbol{h}(\boldsymbol{x}_i)) \quad L \neq \sum_{i=1}^{n} \ell(\boldsymbol{y}_i, \boldsymbol{h}(\boldsymbol{x}_i))$$

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

$$\ell(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x})) = \sum_{i=1}^{m} \ell(y_i, h_i(\boldsymbol{x})) \quad \ell(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x})) \neq \sum_{i=1}^{m} \ell(y_i, h_i(\boldsymbol{x}))$$

The joint target view

- 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.

A starting example

- Training data: $\{(\boldsymbol{x}_1, \boldsymbol{y}_1), (\boldsymbol{x}_2, \boldsymbol{y}_2), \ldots, (\boldsymbol{x}_n, \boldsymbol{y}_n)\}$, $\boldsymbol{y}_i \in \mathcal{Y} = \{0,1\}^m$.
- **Predict** the vector $\boldsymbol{y} = (y_1, y_2, \ldots, y_m)$ for a given $\boldsymbol{x}$.

	X_1	X_2	Y_1	Y_2	$\ldots$	Y_m
$\boldsymbol{x}_1$	5.0	4.5	1	1		0
$\boldsymbol{x}_2$	2.0	2.5	0	1		0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$
$\boldsymbol{x}_n$	3.0	3.5	0	1		1
$\boldsymbol{x}$	4.0	2.5	?	?		?

A starting example

- Training data: $\{(\boldsymbol{x}_1, \boldsymbol{y}_1), (\boldsymbol{x}_2, \boldsymbol{y}_2), \ldots, (\boldsymbol{x}_n, \boldsymbol{y}_n)\}$, $\boldsymbol{y}_i \in \mathcal{Y} = \{0,1\}^m$.
- **Predict** the vector $\boldsymbol{y} = (y_1, y_2, \ldots, y_m)$ for a given $\boldsymbol{x}$.

	X_1	X_2	Y_1	Y_2	$\ldots$	Y_m
$\boldsymbol{x}_1$	5.0	4.5	1	1		0
$\boldsymbol{x}_2$	2.0	2.5	0	1		0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$
$\boldsymbol{x}_n$	3.0	3.5	0	1		1
$\boldsymbol{x}$	4.0	2.5	1	1		0

Two basic approaches

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

$$(\boldsymbol{x}, \boldsymbol{y}) \longrightarrow (\boldsymbol{x}, y = y_i), \quad i = 1, \ldots, m$$

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

$$(\boldsymbol{x}, \boldsymbol{y}) \longrightarrow (\boldsymbol{x}, y = \mathrm{metaclass}(\boldsymbol{y}))$$

	X_1	X_2	Y_1	Y_2	$\ldots$	Y_m
$\boldsymbol{x}_1$	5.0	4.5	1	1		0
$\boldsymbol{x}_2$	2.0	2.5	0	1		0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$
$\boldsymbol{x}_n$	3.0	3.5	0	1		1

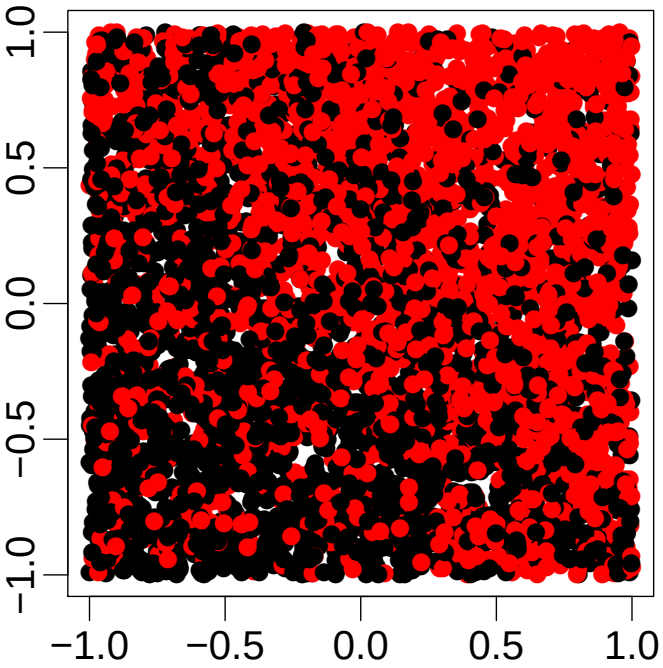
Synthetic data

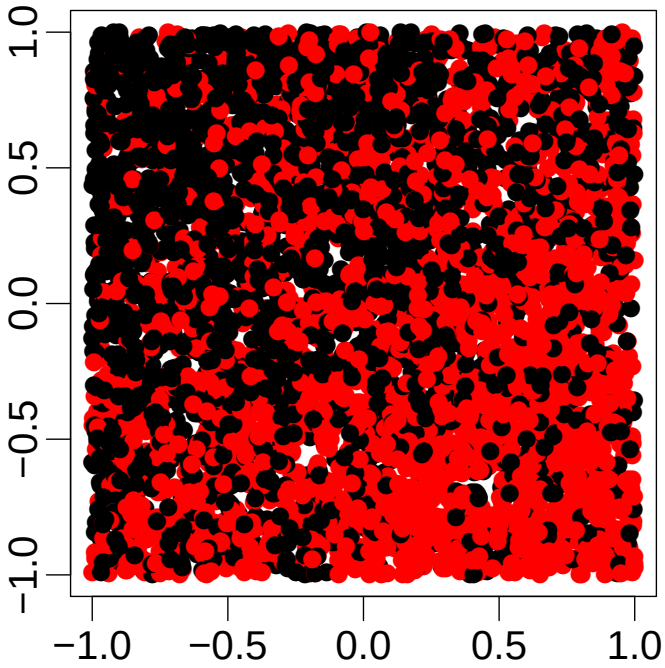
- Two independent models:

$$f_1(\boldsymbol{x}) = \frac{1}{2}x_1 + \frac{1}{2}x_2, \quad f_2(\boldsymbol{x}) = \frac{1}{2}x_1 - \frac{1}{2}x_2$$

- Logistic model to get labels:

$$P(y_i = 1) = \frac{1}{1 + \exp(-2f_i)}$$





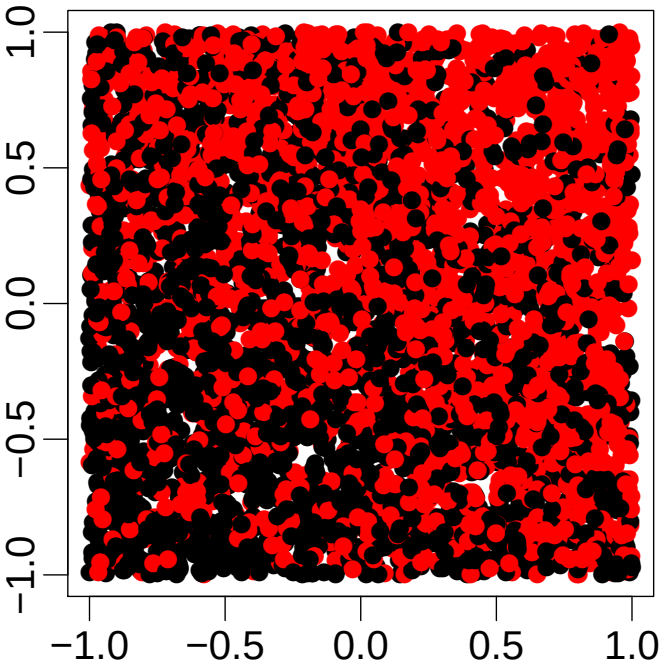
Synthetic data

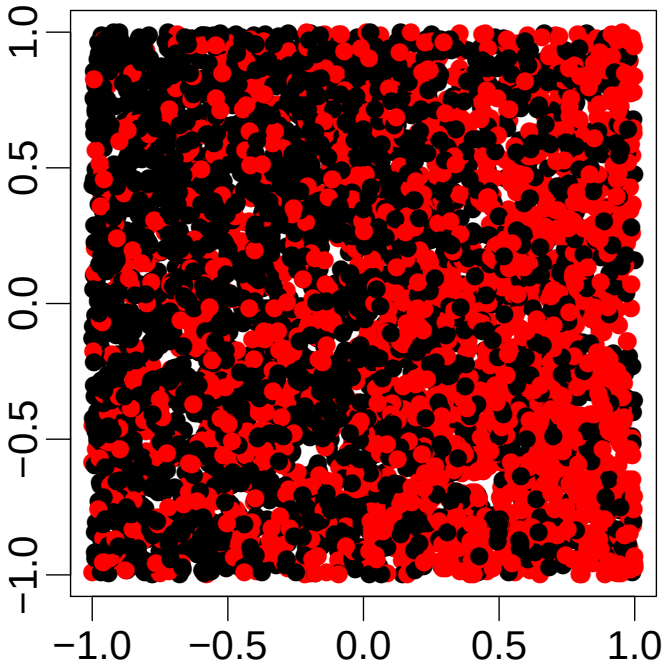
- Two dependent models:

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

- Logistic model to get labels:

$$P(y_i = 1) = \frac{1}{1 + \exp(-2f_i)}$$





Results for two performance measures

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

Conditional independence		
classifier	Hamming loss	subset 0/1 loss
BR LR	0.4232	0.6723
LP LR	0.4232	0.6725

Conditional dependence		
classifier	Hamming loss	subset 0/1 loss
BR LR	0.3470	0.5499
LP LR	0.3610	0.5146

Linear + XOR synthetic data

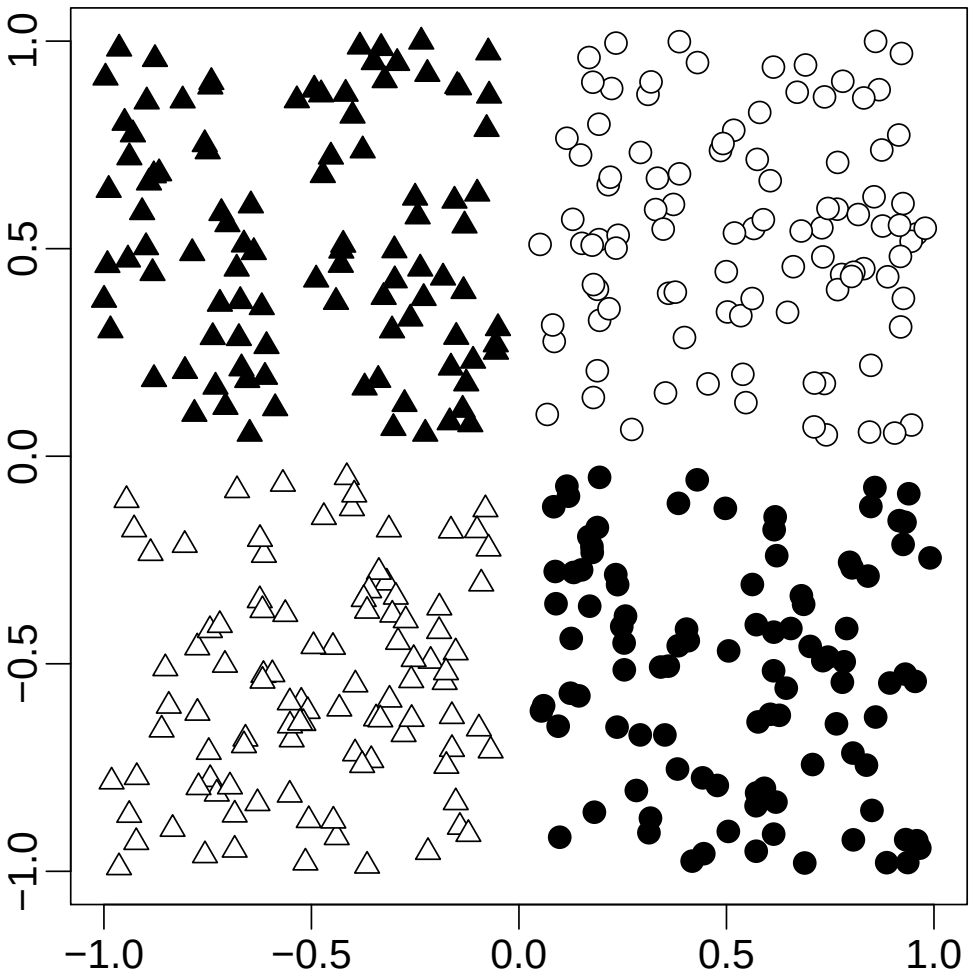


Figure : Problem with two targets: shapes (△ vs. ○) and colors (□ vs. ■).

Linear + XOR synthetic data

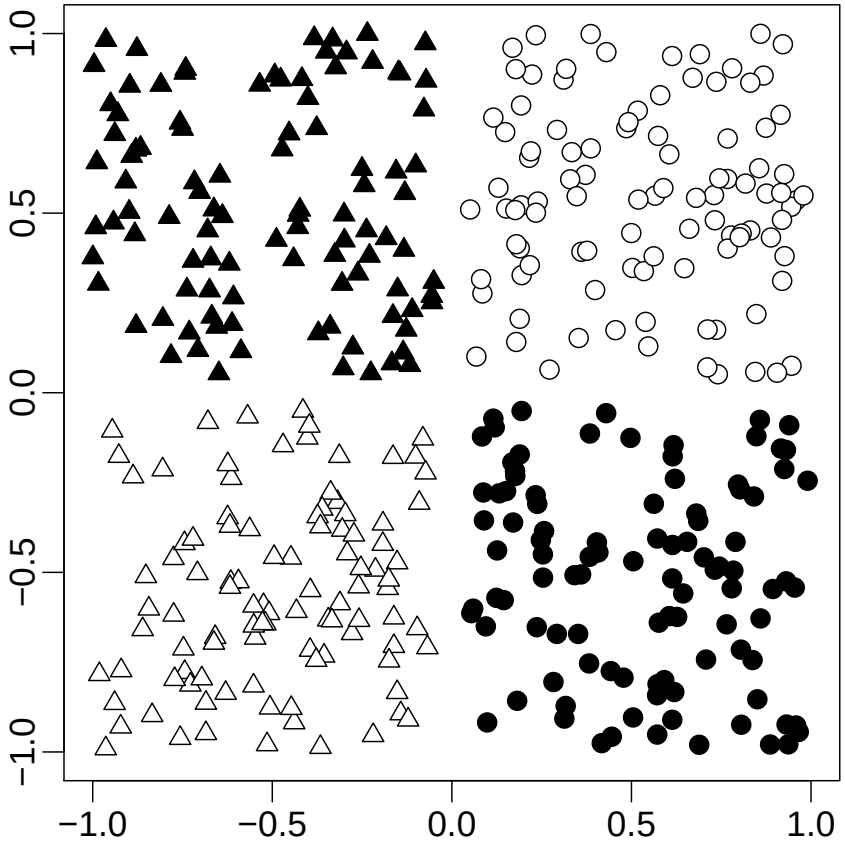
CLASSIFIER	HAMMING LOSS	SUBSET 0/1 LOSS
BR LR	0.2399(±.0097)	0.4751(±.0196)
LP LR	0.0143(±.0020)	0.0195(±.0011)
BAYES OPTIMAL	0	0

Linear + XOR synthetic data

CLASSIFIER	HAMMING LOSS	SUBSET 0/1 LOSS
BR LR	0.2399(±.0097)	0.4751(±.0196)
LP LR	0.0143(±.0020)	0.0195(±.0011)
BR MLRules	**0.0011(±.0002)**	**0.0020(±.0003)**
BAYES OPTIMAL	0	0

Linear + XOR synthetic data

- 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

$$P(\boldsymbol{Y}, \boldsymbol{X})\,.$$

- Since we predict the value of $\boldsymbol{Y}$ for a given object $\boldsymbol{x}$, we are interested in the conditional distribution:

$$P(\boldsymbol{Y} = \boldsymbol{y} | \boldsymbol{X} = \boldsymbol{x}) = \frac{P(\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{X} = \boldsymbol{x})}{P(\boldsymbol{X} = \boldsymbol{x})}$$

Multi-target prediction - probabilistic view

- Data are coming from distribution

$$P(\boldsymbol{Y}, \boldsymbol{X})\,.$$

- Since we predict the value of $\boldsymbol{Y}$ for a given object $\boldsymbol{x}$, we are interested in the conditional distribution:

$$P(\boldsymbol{Y} = \boldsymbol{y} | \boldsymbol{X} = \boldsymbol{x}) = \frac{P(\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{X} = \boldsymbol{x})}{P(\boldsymbol{X} = \boldsymbol{x})}$$

- **What is the most reasonable response y?**

Multi-target prediction - probabilistic view

- Data are coming from distribution

$$P(\boldsymbol{Y}, \boldsymbol{X})\,.$$

- Since we predict the value of $\boldsymbol{Y}$ for a given object $\boldsymbol{x}$, we are interested in the conditional distribution:

$$P(\boldsymbol{Y} = \boldsymbol{y} | \boldsymbol{X} = \boldsymbol{x}) = \frac{P(\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{X} = \boldsymbol{x})}{P(\boldsymbol{X} = \boldsymbol{x})}$$

- **What is the most reasonable response y?**
  - $P(\boldsymbol{Y} = \boldsymbol{y} | \boldsymbol{X} = \boldsymbol{x})$ is the largest?

Multi-target prediction - probabilistic view

- Data are coming from distribution

$$P(\boldsymbol{Y}, \boldsymbol{X})\,.$$

- Since we predict the value of $\boldsymbol{Y}$ for a given object $\boldsymbol{x}$, we are interested in the conditional distribution:

$$P(\boldsymbol{Y} = \boldsymbol{y} | \boldsymbol{X} = \boldsymbol{x}) = \frac{P(\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{X} = \boldsymbol{x})}{P(\boldsymbol{X} = \boldsymbol{x})}$$

- **What is the most reasonable response y?**
  - $P(\boldsymbol{Y} = \boldsymbol{y} | \boldsymbol{X} = \boldsymbol{x})$ is the largest?
  - $P(Y_i = y_i | \boldsymbol{X} = \boldsymbol{x})$ are the largest?

Multi-target prediction - probabilistic view

- Data are coming from distribution

$$P(\boldsymbol{Y}, \boldsymbol{X})\,.$$

- Since we predict the value of $\boldsymbol{Y}$ for a given object $\boldsymbol{x}$, we are interested in the conditional distribution:

$$P(\boldsymbol{Y} = \boldsymbol{y} | \boldsymbol{X} = \boldsymbol{x}) = \frac{P(\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{X} = \boldsymbol{x})}{P(\boldsymbol{X} = \boldsymbol{x})}$$

- **What is the most reasonable response y?**
  - $P(\boldsymbol{Y} = \boldsymbol{y} | \boldsymbol{X} = \boldsymbol{x})$ is the largest?
  - $P(Y_i = y_i | \boldsymbol{X} = \boldsymbol{x})$ are the largest?
  - ... ?
  - ... ?
  - ... ?

Multi-target prediction - loss minimization view

- Define your problem via **minimization** of a **loss** function $\ell(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x}))$.
- **Risk** (expected loss) of the prediction $\boldsymbol{h}$ for a given $\boldsymbol{x}$ is:

$$L_\ell(\boldsymbol{h}, P \,|\, \boldsymbol{x}) = \mathbb{E}_{\boldsymbol{Y}|\boldsymbol{x}}\left[\ell(\boldsymbol{Y}, \boldsymbol{h}(\boldsymbol{x}))\right] = \sum_{\boldsymbol{y} \in \mathcal{Y}} P(\boldsymbol{Y} = \boldsymbol{y} \,|\, \boldsymbol{x}) \ell(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x}))$$

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

$$\boldsymbol{h}^*(\boldsymbol{x}) = \underset{\boldsymbol{h}(\boldsymbol{x})}{\arg\min}\, L_\ell(\boldsymbol{h}, P \,|\, \boldsymbol{x})$$

- Different formulations of loss functions possible:
  - Set-based losses.
  - Ranking-based losses.

Multi-target prediction - loss minimization view

- Subset 0/1 loss: $\ell_{0/1}(\boldsymbol{y}, \boldsymbol{h}) = [\![\boldsymbol{y} \neq \boldsymbol{h}]\!]$
- Hamming loss: $\ell_H(\boldsymbol{y}, \boldsymbol{h}) = \frac{1}{m} \sum_{i=1}^{m} [\![y_i \neq h_i]\!]$
- F-measure-based loss: $\ell_F(\boldsymbol{y}, \boldsymbol{h}) = 1 - \frac{2 \sum_{i=1}^{m} y_i h_i}{\sum_{i=1}^{m} y_i + \sum_{i=1}^{m} h_i}$
- Rank loss: $\ell_{\mathsf{rnk}}(\boldsymbol{y}, \boldsymbol{h}) = w(\boldsymbol{y}) \sum_{y_i > y_j} \left([\![h_i < h_j]\!] + \frac{1}{2} [\![h_i = h_j]\!] \right)$
- ...

Loss minimization view - main issues

- 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.

Relations between losses

- The loss function $\ell(\boldsymbol{y}, \boldsymbol{h})$ should fulfill some basic conditions:
  - $\ell(\boldsymbol{y}, \boldsymbol{h}) = 0$ if and only if $\boldsymbol{y} = \boldsymbol{h}$.
  - $\ell(\boldsymbol{y}, \boldsymbol{h})$ 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.

Relations between losses

- Hamming loss vs. subset 0/1 loss:[1]
  - The form of risk minimizers.
  - Consistency of risk minimizers.
  - Risk bound analysis.
  - Regret bound analysis.

[1] K. Dembczyński, W. Waegeman, W. Cheng, and E. Hüllermeier. On loss minimization and label dependence in multi-label classification. *Machine Learning*, 88:5–45, 2012

Risk minimizers

- The risk minimizer for the Hamming loss is the **marginal mode**:

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

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

$$\mathbf{h}^*(\boldsymbol{x}) = \arg\max_{\boldsymbol{y} \in \mathcal{Y}} P(\boldsymbol{y} \,|\, \boldsymbol{x}).$$

- Marginal mode vs. joint mode.

$\boldsymbol{y}$	$P(\boldsymbol{y})$
0 0 0 0	0.30
0 1 1 1	0.17
1 0 1 1	0.18
1 1 0 1	0.17
1 1 1 0	0.18

Marginal mode: 1 1 1 1
Joint mode: 0 0 0 0

Consistency of risk minimizers and risk bounds

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

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

  under specific conditions, for example, when:
  - Targets $Y_1, \ldots, Y_m$ are conditionally independent, i.e,

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

  - The probability of the joint mode satisfies

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

- The following bounds hold for any $P(\boldsymbol{Y}\,|\,\boldsymbol{x})$ and $\boldsymbol{h}$:

$$\frac{1}{m} L_{0/1}(\boldsymbol{h}, P\,|\,\boldsymbol{x}) \leq L_H(\boldsymbol{h}, P\,|\,\boldsymbol{x}) \leq L_{0/1}(\boldsymbol{h}, P\,|\,\boldsymbol{x})$$

Regret analysis

- 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.

Regret analysis

- 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.**

Regret analysis

- The **regret** of a classifier with respect to ℓ is defined as:

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

  where $\boldsymbol{h}^*_\ell$ is the Bayes classifier for a given loss ℓ.
- Regret measures how worse is $\boldsymbol{h}$ by comparison with the optimal classifier for a given loss.
- To simplify the analysis we will consider the conditional regret:

$$\mathrm{Reg}_\ell(\boldsymbol{h}, P \,|\, \boldsymbol{x}) = L_\ell(\boldsymbol{h}, P \,|\, \boldsymbol{x}) - L_\ell(\boldsymbol{h}^*_\ell, P \,|\, \boldsymbol{x})\,.$$

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

  with respect to both functions.
- It is a bit an unusual analysis.

Regret analysis

- The following **upper bound** holds:

$$\mathrm{Reg}_{0/1}(\boldsymbol{h}_H^*, P \,|\, \boldsymbol{x}) = L_{0/1}(\boldsymbol{h}_H^*, P \,|\, \boldsymbol{x}) - L_{0/1}(\boldsymbol{h}_{0/1}^*, P \,|\, \boldsymbol{x}) < 0.5$$

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

$\boldsymbol{y}$	$P(\boldsymbol{y})$
0 0 0 0	0.02
0 0 1 1	0.49
1 1 0 0	0.49

Marginal mode: 0 0 0 0

Joint mode: 0 0 1 1 or 1 1 0 0

Regret analysis

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

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

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

$\boldsymbol{y}$	$P(\boldsymbol{y})$
0 0 0 0	0.170
0 1 1 1	0.166
1 0 1 1	0.166
1 1 0 1	0.166
1 1 1 0	0.166
1 1 1 1	0.166

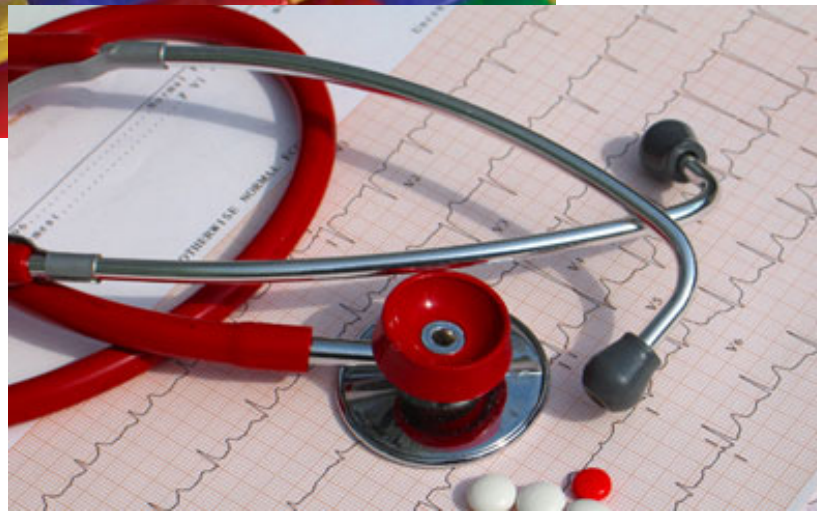
Marginal mode: 1 1 1 1
Joint mode: 0 0 0 0

Relations between losses

- 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.

Relations between losses

- 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.



BR vs. LP

- **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?

BR vs. LP

- 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).

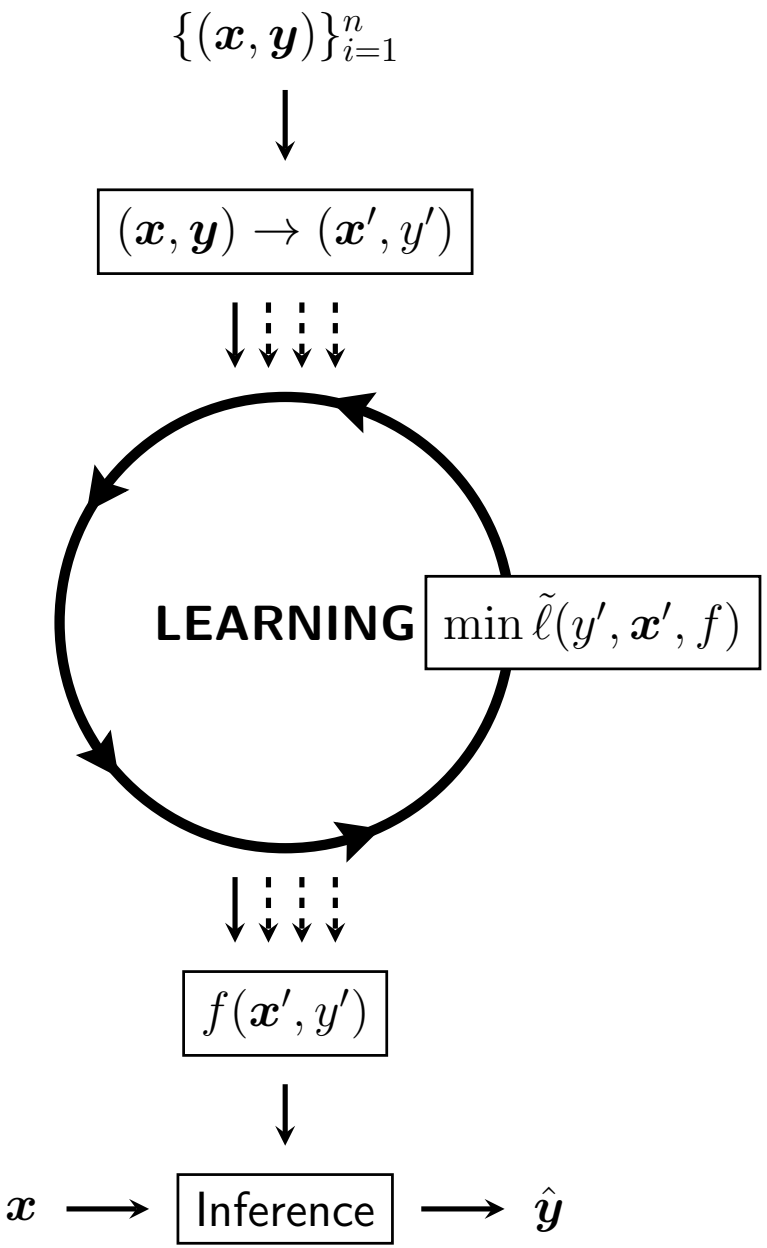
BR vs. LP

- 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 $\boldsymbol{x}$: 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).

Algorithmic approaches for multivariate losses

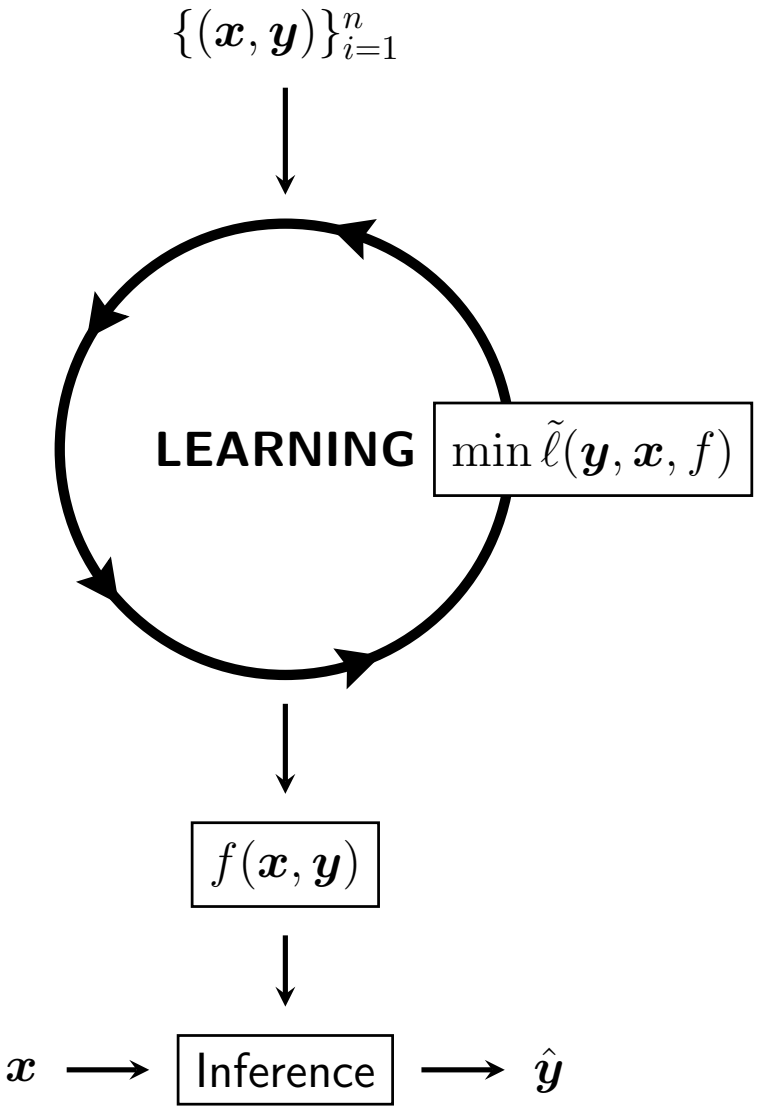
- The loss functions, like Hamming loss or subset 0/1 loss, often referred to as **task losses**, are usually neither convex nor differentiable.
- Therefore learning is a hard optimization problem.
- Two approaches try to make this task easier
  - Reduction.
  - Structured loss minimization.
- Two phases in solving multi-target prediction problems:
  - Learning: Estimate parameters of the scoring function $f(\boldsymbol{x}, \boldsymbol{y})$.
  - Inference: Use the scoring function $f(\boldsymbol{x}, \boldsymbol{y})$ to classify new instances by finding the best $\boldsymbol{y}$ for a given $\boldsymbol{x}$.

Reduction



- **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



- Replace the task loss by a **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.

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 ℓ when the following holds:

$$\mathrm{Reg}_{\tilde{\ell}}(\boldsymbol{h}, P) \to 0 \Rightarrow \mathrm{Reg}_{\ell}(\boldsymbol{h}, P) \to 0\,.$$

- 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.

[2] A. Tewari and P.L. Bartlett. On the consistency of multiclass classification methods. *JMLR*, 8:1007–1025, 2007

D. McAllester and J. Keshet. Generalization bounds and consistency for latent structural probit and ramp loss. In *NIPS*, pages 2205–2212, 2011

W. Gao and Z.-H. Zhou. On the consistency of multi-label learning. *Artificial Intelligence*, 199-200:22–44, 2013

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 $\boldsymbol{Y}$ by:

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

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

$$Z(\boldsymbol{x}) = \sum_{\boldsymbol{y} \in \mathcal{Y}} \exp(f(\boldsymbol{x}, \boldsymbol{y}))$$

[3] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, pages 282–289, 2001

Conditional random fields

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

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

- Regularized log-likelihood optimization:

$$\min_f \frac{1}{n} \sum_{i=1}^{n} \ell_{\log}(\boldsymbol{y}, \boldsymbol{x}, f) + \lambda J(f)$$

- Inference for a new instance $\boldsymbol{x}$:

$$\boldsymbol{h}(\boldsymbol{x}) = \arg\max_{\boldsymbol{y} \in \mathcal{Y}} P(\boldsymbol{y} \,|\, \boldsymbol{x})$$

Conditional random fields

- Similar to LP, but with an internal structure of classes and scoring function $f(\boldsymbol{x}, \boldsymbol{y})$.
- Convex optimization problem, but depending on the structure of $f(\boldsymbol{x}, \boldsymbol{y})$ 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 $f(\boldsymbol{x}, \boldsymbol{y})$.

Conditional random fields

- Let $f(\boldsymbol{x}, \boldsymbol{y})$ be defined as:

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

- In this case, we have:

Conditional random fields

- Let $f(\boldsymbol{x}, \boldsymbol{y})$ be defined as:

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

- In this case, we have:

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

Conditional random fields

- Let $f(\boldsymbol{x}, \boldsymbol{y})$ be defined as:

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

- In this case, we have:

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

Conditional random fields

- Let $f(\boldsymbol{x}, \boldsymbol{y})$ be defined as:

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

- In this case, we have:

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

- **Optimal for Hamming loss!!!**
- The structure of $f(\boldsymbol{x}, \boldsymbol{y})$ is connected to the loss function.

Conditional random fields

- A different form of $f(\boldsymbol{x}, \boldsymbol{y})$:

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

- Models pairwise interactions, ...

Conditional random fields

- A different form of $f(\boldsymbol{x}, \boldsymbol{y})$:

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

- Models pairwise interactions, ... but in the **conditional sense**:

Conditional random fields

- A different form of $f(\boldsymbol{x}, \boldsymbol{y})$:

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

- Models pairwise interactions, ... but in the **conditional sense**:
  - Assume that $\boldsymbol{x}$ is not given:

$$P(\boldsymbol{y}) = \frac{\exp(\sum_i f_i(y_i) + \sum_{y_k, y_l} f_{k,l}(y_k, y_l))}{\sum_{\boldsymbol{y} \in \mathcal{Y}} \exp(\sum_i f_i(y_i) + \sum_{y_k, y_l} f_{k,l}(y_k, y_l))}$$

  - **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.

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 ...

[4] Y. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 6:1453–1484, 2005

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.[4]

[4] Y. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 6:1453–1484, 2005

Structured support vector machines

- SSVMs use, similarly to CRFs, a scoring function $f(\boldsymbol{x}, \boldsymbol{y})$.
- They minimize the **structured hinge loss**:

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

- Task loss $\ell(\boldsymbol{y}, \boldsymbol{y}')$ is used for margin rescaling.
- Regularized optimization problem:

$$\min_f \frac{1}{n} \sum_{i=1}^{n} \tilde{\ell}_h(\boldsymbol{y}, \boldsymbol{x}, f) + \lambda J(f)$$

- Predict according to:

$$\boldsymbol{h}(\boldsymbol{x}) = \arg\max_{\boldsymbol{y} \in \mathcal{Y}} f(\boldsymbol{x}, \boldsymbol{y}) \,.$$

Structured support vector machines

- Convex optimization problem with linear constraints.
- An exponential number of constraints $\longrightarrow$ Cutting-plane algorithms.
- The $\arg\max$ problem is hard for general structures.
- Different forms of $f(\boldsymbol{x}, \boldsymbol{y})$.

Structured support vector machines

- Let $f(\boldsymbol{x}, \boldsymbol{y})$ be defined as:

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

- Let us use it with the Hamming loss:

[5] B. Hariharan, L. Zelnik-Manor, S.V.N. Vishwanathan, and M. Varma. Large scale max-margin multi-label classification with priors. In *ICML*. Omnipress, 2010

Structured support vector machines

- Let $f(\boldsymbol{x}, \boldsymbol{y})$ be defined as:

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

- Let us use it with the Hamming loss:

$$\begin{aligned}
\tilde{\ell}_h(\boldsymbol{y}, \boldsymbol{x}, f) &= \max_{\boldsymbol{y}' \in \mathcal{Y}} \{\ell_H(\boldsymbol{y}, \boldsymbol{y}') + f(\boldsymbol{x}, \boldsymbol{y}')\} - f(\boldsymbol{x}, \boldsymbol{y}) \\
&= \max_{\boldsymbol{y}' \in \mathcal{Y}} \left\{ \sum_i [\![y_i \neq y_i']\!] + \sum_i f_i(\boldsymbol{x}, y_i') \right\} - \sum_i f_i(\boldsymbol{x}, y_i)
\end{aligned}$$

[5] B. Hariharan, L. Zelnik-Manor, S.V.N. Vishwanathan, and M. Varma. Large scale max-margin multi-label classification with priors. In *ICML*. Omnipress, 2010

Structured support vector machines

- Let $f(\boldsymbol{x}, \boldsymbol{y})$ be defined as:

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

- Let us use it with the Hamming loss:

$$\begin{aligned}
\tilde{\ell}_h(\boldsymbol{y}, \boldsymbol{x}, f) &= \max_{\boldsymbol{y}' \in \mathcal{Y}} \{\ell_H(\boldsymbol{y}, \boldsymbol{y}') + f(\boldsymbol{x}, \boldsymbol{y}')\} - f(\boldsymbol{x}, \boldsymbol{y}) \\
&= \max_{\boldsymbol{y}' \in \mathcal{Y}} \left\{ \sum_i [\![y_i \neq y_i']\!] + \sum_i f_i(\boldsymbol{x}, y_i') \right\} - \sum_i f_i(\boldsymbol{x}, y_i) \\
&= \sum_i \max_{y_i'} \left\{ [\![y_i \neq y_i']\!] + f_i(\boldsymbol{x}, y_i') - f_i(\boldsymbol{x}, y_i) \right\}
\end{aligned}$$

[5] B. Hariharan, L. Zelnik-Manor, S.V.N. Vishwanathan, and M. Varma. Large scale max-margin multi-label classification with priors. In *ICML*. Omnipress, 2010

Structured support vector machines

- Let $f(\boldsymbol{x}, \boldsymbol{y})$ be defined as:

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

- Let us use it with the Hamming loss:

$$\begin{aligned}
\tilde{\ell}_h(\boldsymbol{y}, \boldsymbol{x}, f) &= \max_{\boldsymbol{y}' \in \mathcal{Y}} \{\ell_H(\boldsymbol{y}, \boldsymbol{y}') + f(\boldsymbol{x}, \boldsymbol{y}')\} - f(\boldsymbol{x}, \boldsymbol{y}) \\
&= \max_{\boldsymbol{y}' \in \mathcal{Y}} \left\{ \sum_i [\![y_i \neq y_i']\!] + \sum_i f_i(\boldsymbol{x}, y_i') \right\} - \sum_i f_i(\boldsymbol{x}, y_i) \\
&= \sum_i \max_{y_i'} \left\{ [\![y_i \neq y_i']\!] + f_i(\boldsymbol{x}, y_i') - f_i(\boldsymbol{x}, y_i) \right\}
\end{aligned}$$

- **Structured hinge loss decomposes to hinge loss for each label.**[5]
- Consistent for the Hamming loss.

[5] B. Hariharan, L. Zelnik-Manor, S.V.N. Vishwanathan, and M. Varma. Large scale max-margin multi-label classification with priors. In *ICML*. Omnipress, 2010

Structured support vector machines

- The form $f(\boldsymbol{x}, \boldsymbol{y})$ that models pairwise interactions:

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

- **How important is the pairwise interaction part for different task losses?**
- For a general form of $f(\boldsymbol{x}, \boldsymbol{y})$, SSVMs are inconsistent for Hamming loss.[6]
- There are more results of this type.[7]

---

[6] W. Gao and Z.-H. Zhou. On the consistency of multi-label learning. *Artificial Intelligence*, 199-200:22–44, 2013

[7] A. Tewari and P.L. Bartlett. On the consistency of multiclass classification methods. *JMLR*, 8:1007–1025, 2007

D. McAllester. *Generalization Bounds and Consistency for Structured Labeling in Predicting Structured Data*. MIT Press, 2007

Table : SSVMs with pairwise term[8] vs. BR with LR[9].

Dataset	SSVM Best	BR LR
Scene	0.101±.003	0.102±.003
Yeast	0.202±.005	0.199±.005
Synth1	0.069±.001	0.067±.002
Synth2	0.058±.001	0.084±.001

- 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

[9] K. Dembczyński, W. Waegeman, W. Cheng, and E. Hüllermeier. An analysis of chaining in multi-label classification. In *ECAI*, 2012

SSVMs vs. CRFs

- SSVMs and CRFs are quite similar to each other:

$$\begin{aligned}\tilde{\ell}_{\log}(\boldsymbol{y}, \boldsymbol{x}, f) &= \log\left(\sum_{\boldsymbol{y}\in\mathcal{Y}} \exp(f(\boldsymbol{x}, \boldsymbol{y}))\right) - f(\boldsymbol{x}, \boldsymbol{y}) \\ \tilde{\ell}_{h}(\boldsymbol{y}, \boldsymbol{x}, f) &= \max_{\boldsymbol{y}'\in\mathcal{Y}}\{\ell(\boldsymbol{y}, \boldsymbol{y}') + f(\boldsymbol{x}, \boldsymbol{y}')\} - f(\boldsymbol{x}, \boldsymbol{y})\end{aligned}$$

- The main differences are:
  - max vs. soft-max
  - margin vs. no-margin
- Many works on incorporating margin in CRFs.[10]

[10] P. Pletscher, C.S. Ong, and J.M. Buhmann. Entropy and margin maximization for structured output learning. In *ECML/PKDD*. Springer, 2010

Q. Shi, M. Reid, and T. Caetano. Hybrid model of conditional random field and support vector machine. In *Workshop at NIPS*, 2009

K. Gimpel and N. Smith. Softmax-margin crfs: Training log-linear models with cost functions. In *HLT*, page 733736, 2010

Probabilistic classifier chains

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

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

- They follow a reduction to a sequence of subproblems:

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

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

[11] J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. *Machine Learning Journal*, 85:333–359, 2011

K. Dembczyński, W. Cheng, and E. Hüllermeier. Bayes optimal multilabel classification via probabilistic classifier chains. In *ICML*, pages 279–286. Omnipress, 2010

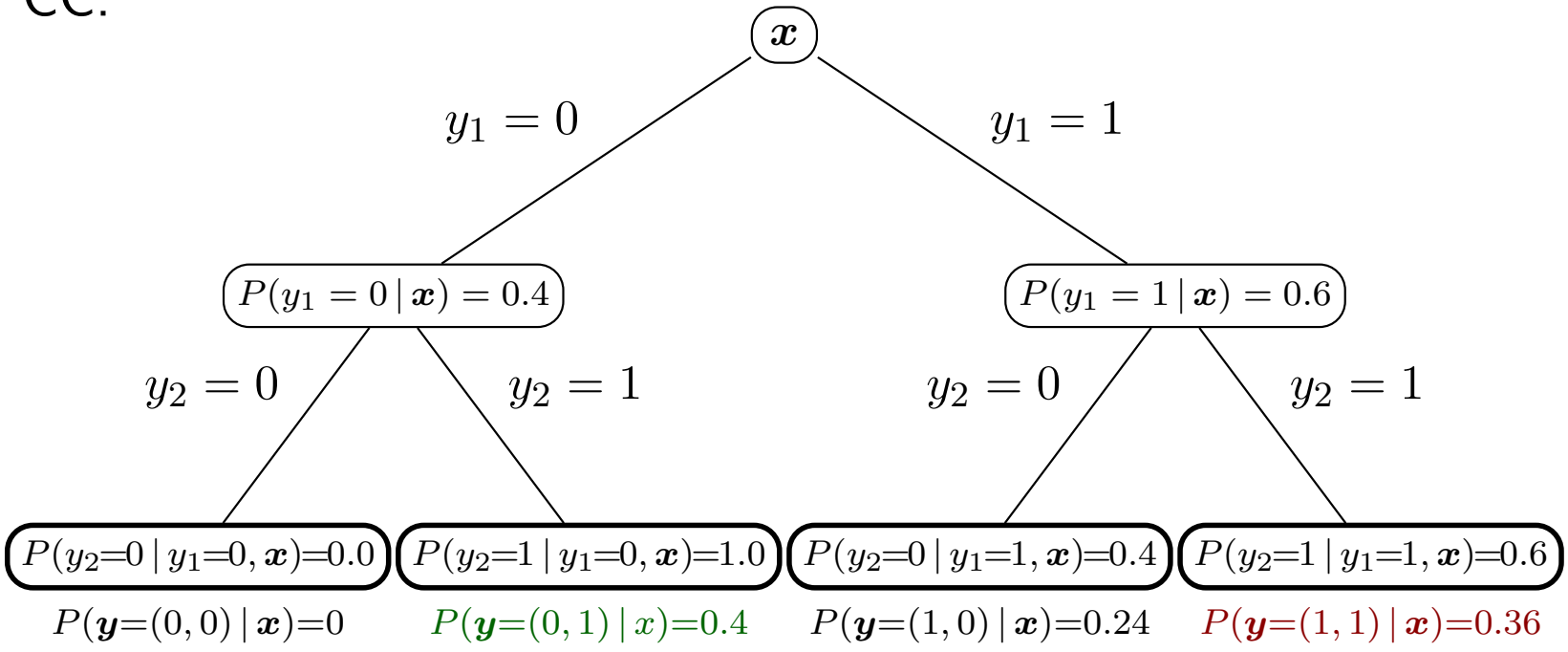
Probabilistic classifier chains

- Learning of PCCs relies on constructing probabilistic classifiers for estimating
$$P(Y_i = y_i | \boldsymbol{x}, y_1, \ldots, y_{i-1}),$$
independently for each $i = 1, \ldots, m$.
- One can use scoring functions $f_i(\boldsymbol{x}', y_i)$ and use logistic transformation.
- By using the linear models, the overall scoring function takes the form:

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

Probabilistic classifier chains

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



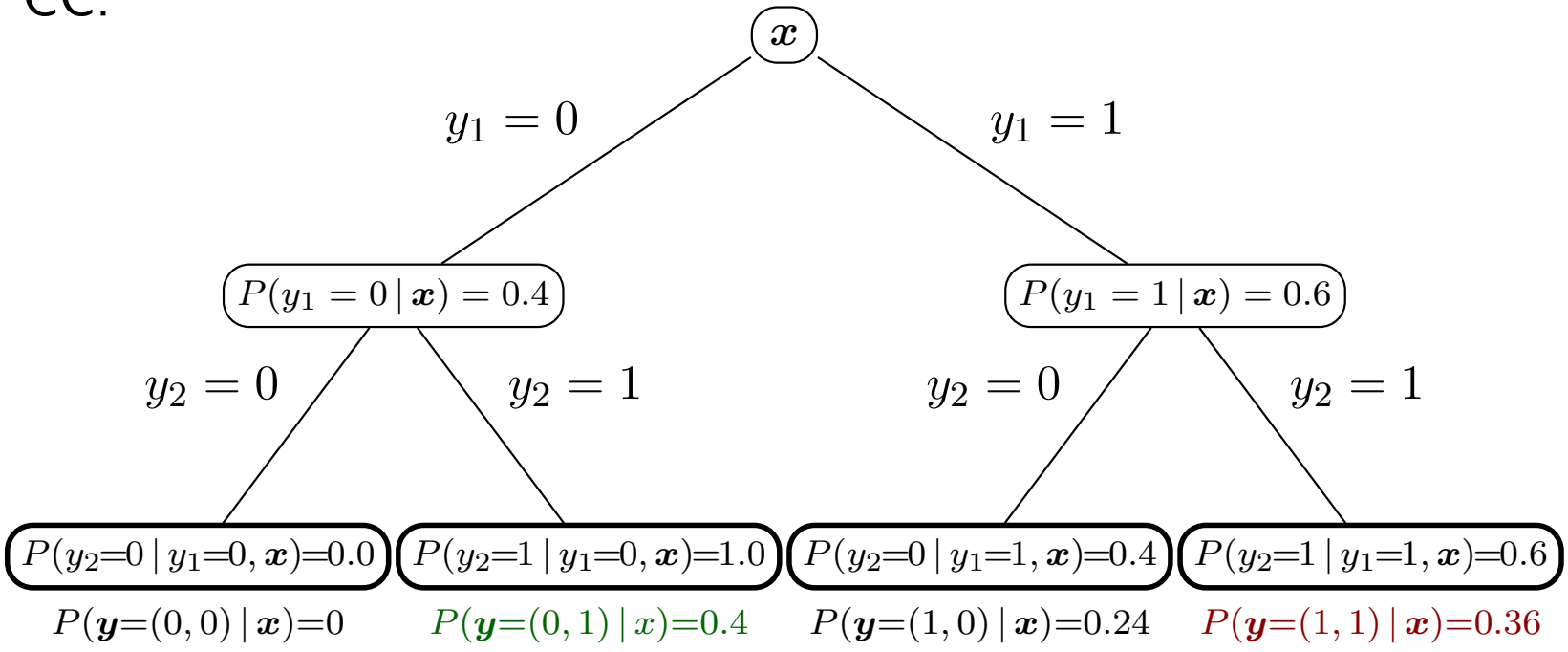
- For subset 0/1 loss one needs to find $\boldsymbol{h}(\boldsymbol{x}) = \arg\max_{\boldsymbol{y} \in \mathcal{Y}} P(\boldsymbol{y} \,|\, \boldsymbol{x})$.
- Greedy and approximate search techniques with guarantees exist.[12]

[12] K. Dembczyński, W. Waegeman, W. Cheng, and E. Hüllermeier. An analysis of chaining in multi-label classification. In *ECAI*, 2012

A. Kumar, S. Vembu, A.K. Menon, and C. Elkan. Beam search algorithms for multilabel learning. In *Machine Learning*, 2013

Probabilistic classifier chains

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



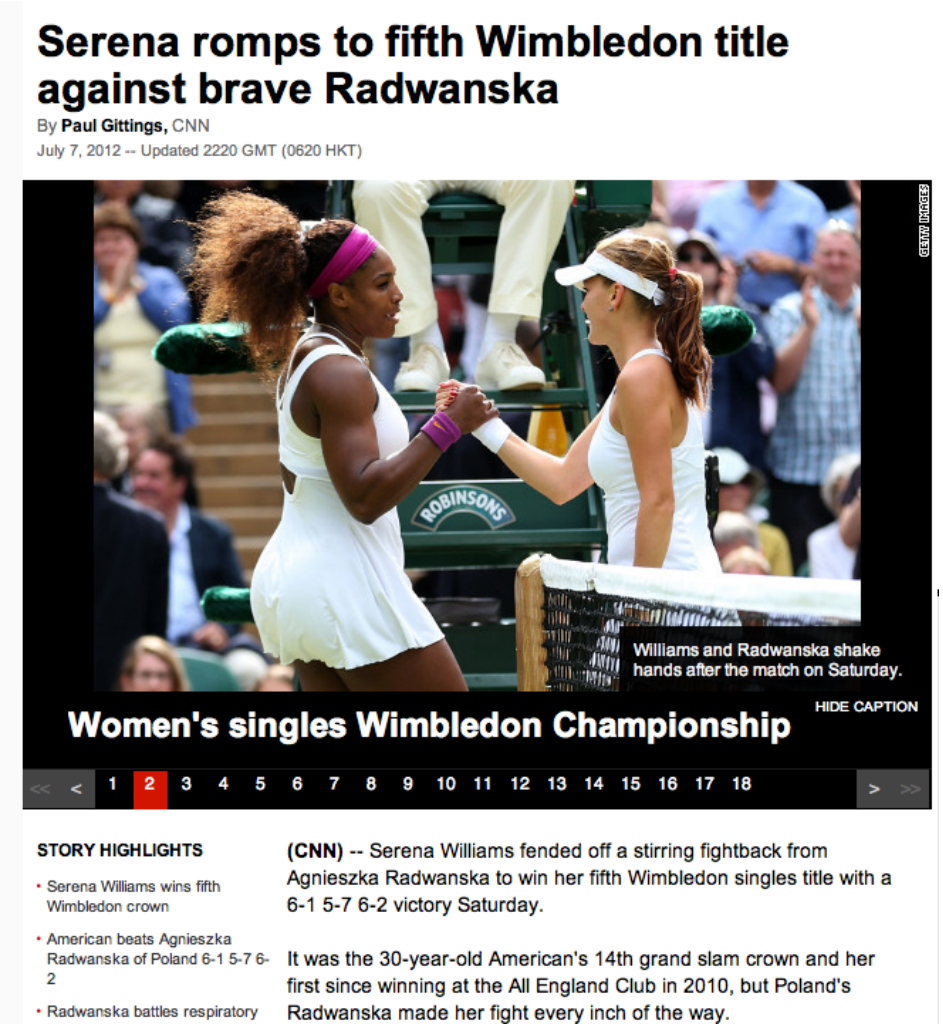
- Other losses: compute the prediction on a sample from $P(\boldsymbol{Y} \,|\, \boldsymbol{x})$.[12]
- Sampling can be easily performed by using the probability tree.

[12] K. Dembczyński, W. Waegeman, W. Cheng, and E. Hüllermeier. An analysis of chaining in multi-label classification. In *ECAI*, 2012

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

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

Multi-label ranking



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.

HIDE CAPTION

Women's singles Wimbledon Championship

<< < 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 > >>

STORY HIGHLIGHTS

- Serena Williams wins fifth Wimbledon crown
- American beats Agnieszka Radwanska of Poland 6-1 5-7 6-2
- Radwanska battles 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.

HIDE CAPTION

Women's singles Wimbledon Championship

<< < 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 > >>

STORY HIGHLIGHTS

- Serena Williams wins fifth Wimbledon crown
- American beats Agnieszka Radwanska of Poland 6-1 5-7 6-2
- Radwanska battles 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 ranking

tennis

Υ

sport

Υ

England

Υ

Poland

Υ

USA

Υ

$\vdots$

Υ

politics

Multi-label ranking

- **Ranking loss**:

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

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

	X_1	X_2	Y_1		Y_2		$\dots$		Y_m
$\boldsymbol{x}$	4.0	2.5	1		0				0
			h_2	$>$	h_1	$>$	$\dots$	$>$	h_m

- **Ranking loss**:

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

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

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

$$w(\boldsymbol{y}) = \frac{1}{n_+ n_-},$$

i.e., it is equal to the inverse of the total number of pairwise comparisons between labels.

Pairwise surrogate losses

- The most intuitive approach is to use pairwise **convex surrogate** losses of the form

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

where ϕ 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]: $\phi(f) = \max(0, 1 - f)$.

[13] R. E. Schapire and Y. Singer. BoosTexter: A Boosting-based System for Text Categorization. *Machine Learning*, 39(2/3):135–168, 2000

[14] O. Dekel, Ch. Manning, and Y. Singer. Log-linear models for label ranking. In *NIPS*. MIT Press, 2004

[15] A. Elisseeff and J. Weston. A kernel method for multi-labelled classification. In *NIPS*, pages 681–687, 2001

Multi-label ranking

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

[16] J. Duchi, L. Mackey, and M. Jordan. On the consistency of ranking algorithms. In *ICML*, pages 327–334, 2010

W. Gao and Z.-H. Zhou. On the consistency of multi-label learning. *Artificial Intelligence*, 199-200:22–44, 2013

[17] K. Dembczynski, W. Kotlowski, and E. Hüllermeier. Consistent multilabel ranking through univariate losses. In *ICML*, 2012

Reduction to weighted binary relevance

- The Bayes ranker can be obtained by sorting labels according to:

$$\Delta_i^1 = \sum_{\boldsymbol{y}\colon y_i=1} w(\boldsymbol{y}) P(\boldsymbol{y} \,|\, \boldsymbol{x})\,.$$

- For $w(\boldsymbol{y}) \equiv 1$, Δ_i^u reduces to **marginal probabilities** $P(Y_i = u \,|\, \boldsymbol{x})$.
- The solution can be obtained with BR or its weighted variant in a general case.

Reduction to weighted binary relevance

- Consider the sum of **univariate (weighted)** losses:

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

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

- The risk minimizer of these losses is:

$$h_i^*(\boldsymbol{x}) = \frac{1}{c} \log \frac{\Delta_i^1}{\Delta_i^0} = \frac{1}{c} \log \frac{\Delta_i^1}{W - \Delta_i^1} ,$$

which is a strictly increasing transformation of Δ_i^1, where

$$W = \mathbb{E}[w(\boldsymbol{Y}) \,|\, \boldsymbol{x}] = \sum_{\boldsymbol{y}} w(\boldsymbol{y}) P(\boldsymbol{y} \,|\, \boldsymbol{x}) .$$

Reduction to weighted binary relevance

- **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]

[18] R. E. Schapire and Y. Singer. BoosTexter: A Boosting-based System for Text Categorization. *Machine Learning*, 39(2/3):135–168, 2000

[19] K. Dembczynski, W. Kotlowski, and E. Hüllermeier. Consistent multilabel ranking through univariate losses. In *ICML*, 2012

Weighted binary relevance

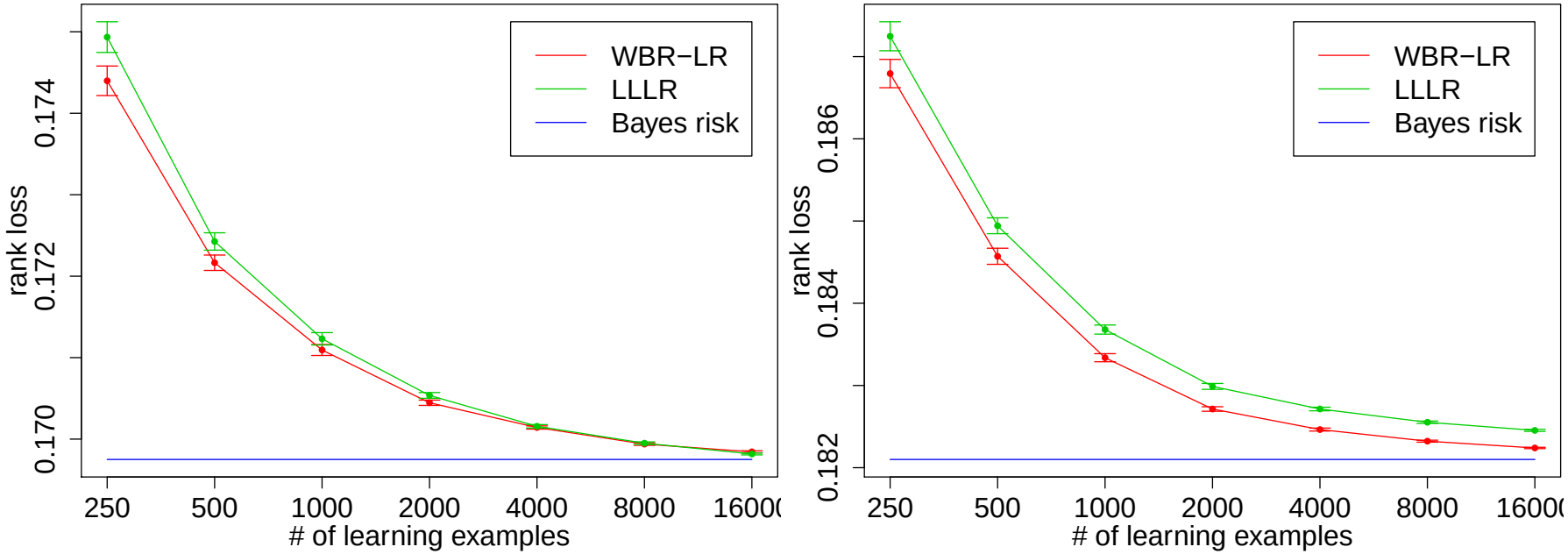


Figure : WBR LR vs. LLLR. Left: independent data. Right: dependent data.

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

Benchmark data

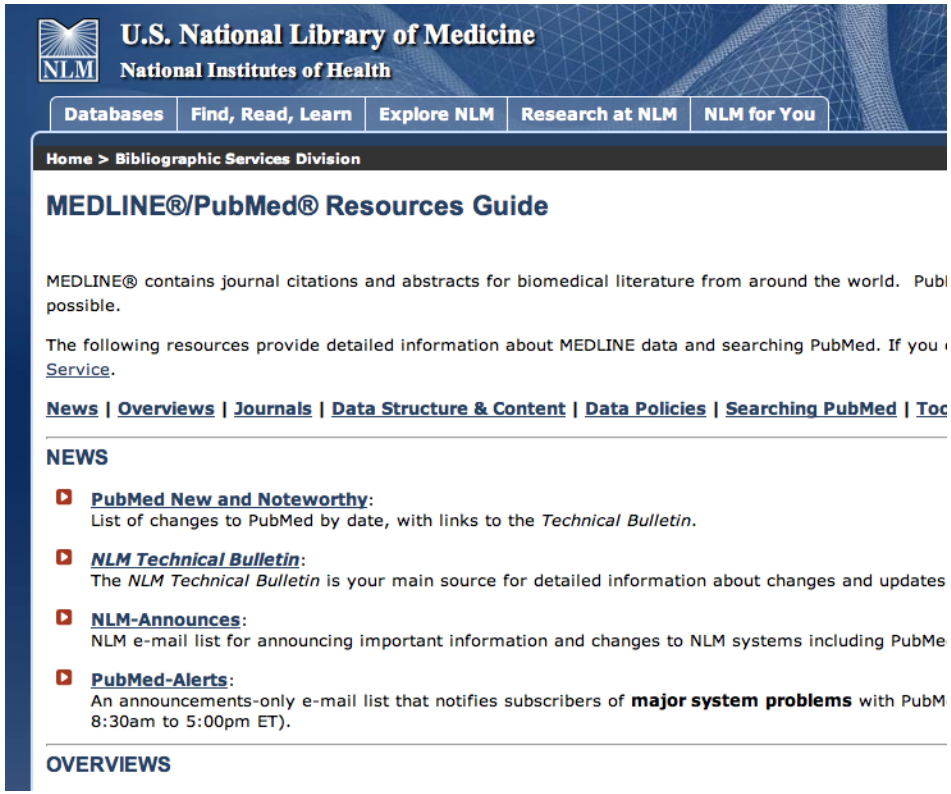
Table : WBR-AdaBoost vs. AdaBoost.MR (left) and WBR-LR vs LLLR (right).

DATASET	AB.MR	WBR-AB	LLLR	WBR-LR
IMAGE	0.2081	0.2041	0.2047	0.2065
EMOTIONS	0.1703	0.1699	0.1743	0.1657
SCENE	0.0720	0.0792	0.0861	0.0793
YEAST	0.2072	0.1820	0.1728	0.1736
MEDIAMILL	0.0665	0.0609	0.0614	0.0472

- 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_β-measure-based loss function (F_β-loss):

$$\begin{aligned}\ell_{F_\beta}(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x})) &= 1 - F_\beta(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x})) \\ &= 1 - \frac{(1+\beta^2)\sum_{i=1}^m y_i h_i(\boldsymbol{x})}{\beta^2 \sum_{i=1}^m y_i + \sum_{i=1}^m h_i(\boldsymbol{x})} \in [0,1]\,.\end{aligned}$$

- Provides a **better balance** between relevant and irrelevant labels.
- However, it **is not easy** to optimize.

SSVMs for F_β-based loss

- SSVMs can be used to minimize F_β-based loss
- Rescale the margin by $\ell_F(\boldsymbol{y}, \boldsymbol{y}')$.
- Two algorithms:[20]

RML

No label interactions:

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

Quadratic learning and linear prediction

SML

Submodular interactions:

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

More complex (graph-cut and approximate algorithms)

- Both are inconsistent.

[20] J. Petterson and T. S. Caetano. Reverse multi-label learning. In *NIPS*, pages 1912–1920, 2010
J. Petterson and T. S. Caetano. Submodular multi-label learning. In *NIPS*, pages 1512–1520, 2011

Plug-in rule approach

- Plug estimates of required parameters into the Bayes classifier.

$$\begin{aligned} \boldsymbol{h}^* &= \arg\min_{\boldsymbol{h}\in\mathcal{Y}} \mathbb{E}\left[\ell_{F_\beta}(\boldsymbol{Y},\boldsymbol{h})\right] \\ &= \arg\max_{\boldsymbol{h}\in\mathcal{Y}} \sum_{\boldsymbol{y}\in\mathcal{Y}} P(\boldsymbol{y}) \frac{(\beta+1)\sum_{i=1}^m y_i h_i}{\beta^2 \sum_{i=1}^m y_i + \sum_{i=1}^m h_i} \end{aligned}$$

- **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 ($P(\boldsymbol{y})$) is 2^m.

Plug-in rule approach

- Approximation needed?

[21] N. Ye, K. Chai, W. Lee, and H. Chieu. Optimizing F-measures: a tale of two approaches. In *ICML*, 2012

[22] K. Dembczyński, W. Waegeman, W. Cheng, and E. Hüllermeier. An exact algorithm for F-measure maximization. In *NIPS*, volume 25, 2011

[23] K. Dembczynski, A. Jachnik, W. Kotlowski, W. Waegeman, and E. Hüllermeier. Optimizing the F-measure in multi-label classification: Plug-in rule approach versus structured loss minimization. In *ICML*, 2013

Plug-in rule approach

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

[21] N. Ye, K. Chai, W. Lee, and H. Chieu. Optimizing F-measures: a tale of two approaches. In *ICML*, 2012

[22] K. Dembczyński, W. Waegeman, W. Cheng, and E. Hüllermeier. An exact algorithm for F-measure maximization. In *NIPS*, volume 25, 2011

[23] K. Dembczynski, A. Jachnik, W. Kotlowski, W. Waegeman, and E. Hüllermeier. Optimizing the F-measure in multi-label classification: Plug-in rule approach versus structured loss minimization. In *ICML*, 2013

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.

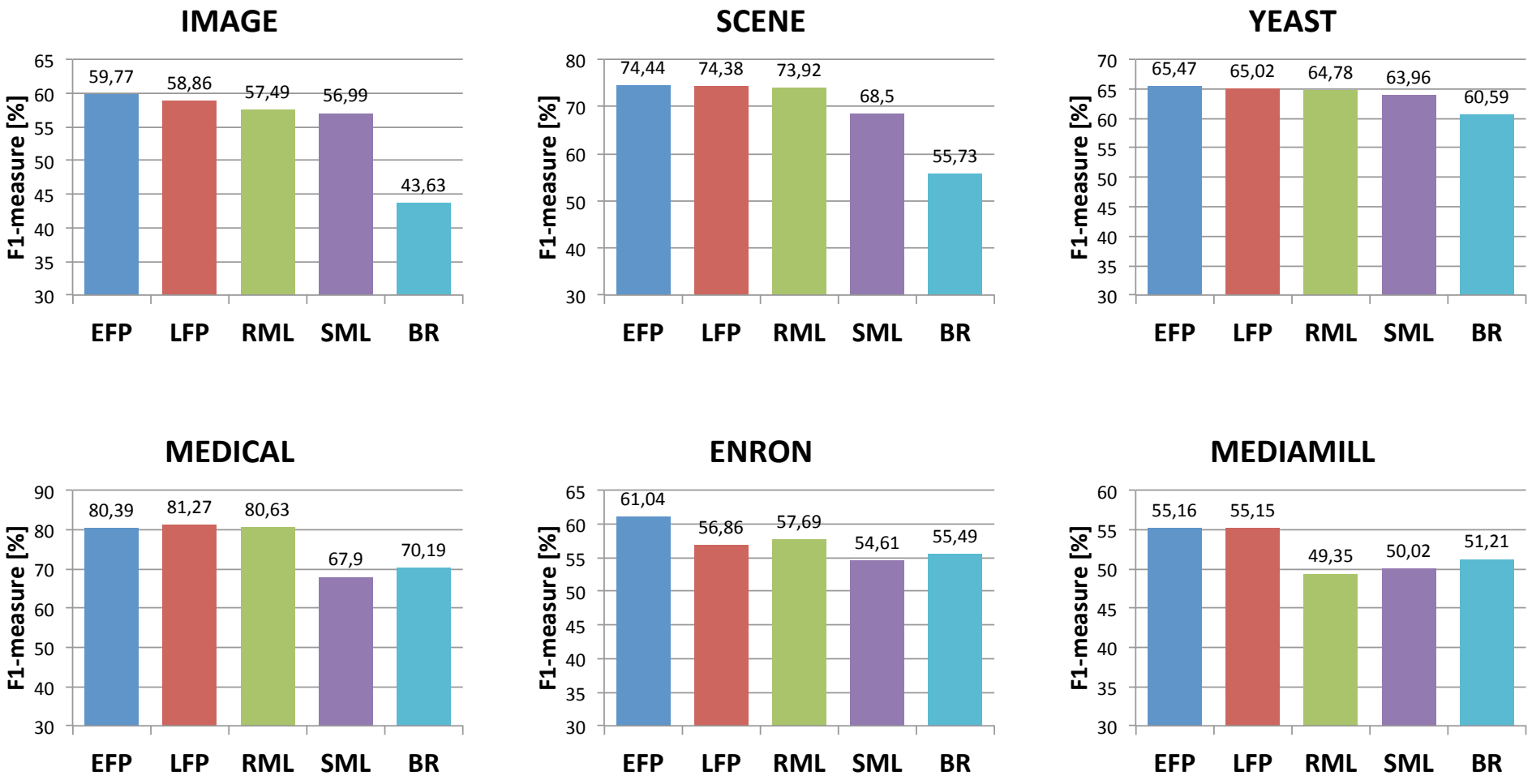
- EFP is consistent.[23]

[21] N. Ye, K. Chai, W. Lee, and H. Chieu. Optimizing F-measures: a tale of two approaches. In *ICML*, 2012

[22] K. Dembczyński, W. Waegeman, W. Cheng, and E. Hüllermeier. An exact algorithm for F-measure maximization. In *NIPS*, volume 25, 2011

[23] K. Dembczynski, A. Jachnik, W. Kotlowski, W. Waegeman, and E. Hüllermeier. Optimizing the F-measure in multi-label classification: Plug-in rule approach versus structured loss minimization. In *ICML*, 2013

Maximization of the F-measure



Challenges

- 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.

INNOVATIVE ECONOMY
NATIONAL COHESION STRATEGY

FNP
Foundation for Polish Science

EUROPEAN UNION
EUROPEAN REGIONAL
DEVELOPMENT FUND