Decision-theoretic Approach to Multi-label Classification

Krzysztof Dembczyński

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

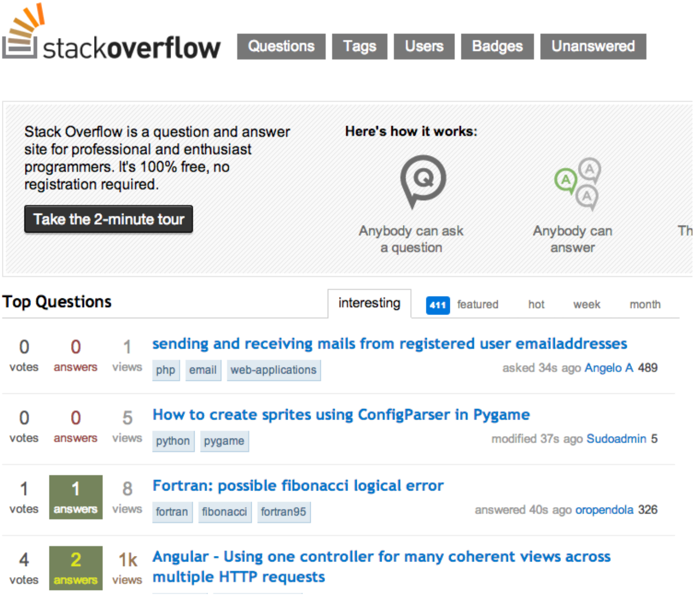
TFML, Bedlewo, Poland, 20.02.2015

Outline

Outline

Image annotation: cloud? sky? tree?

Ecology: Prediction of the presence or absence of species



Document tagging

Multi-label classification

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

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_p) \xrightarrow{\boldsymbol{h}(\boldsymbol{x})} \boldsymbol{y} = (y_1, y_2, \ldots, y_m) \in \mathcal{Y} = \{0, 1\}^m$$

Multi-label classification

- Training data: $\{(\boldsymbol{x}_1, \boldsymbol{y}_1), (\boldsymbol{x}_2, \boldsymbol{y}_2), \ldots, (\boldsymbol{x}_n, \boldsymbol{y}_n)\}$.
- **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: $\{(\boldsymbol{x}_1, \boldsymbol{y}_1), (\boldsymbol{x}_2, \boldsymbol{y}_2), \ldots, (\boldsymbol{x}_n, \boldsymbol{y}_n)\}$.
- **Predict** a vector $\boldsymbol{y} = (y_1, y_2, \ldots, y_m)$ for a given $\boldsymbol{x}$.

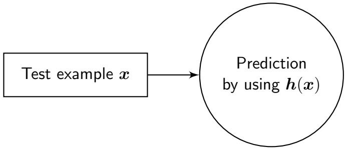
	x_1	x_2	y_1	y_2	$\ldots$	y_m
$\boldsymbol{x}_1$	5.0	4.5	1	1		0
$\boldsymbol{x}_2$	2.0	2.5	0	1		0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$
$\boldsymbol{x}_n$	3.0	3.5	0	1		1
$\boldsymbol{x}$	4.0	2.5	1	1		0

Supervised learning

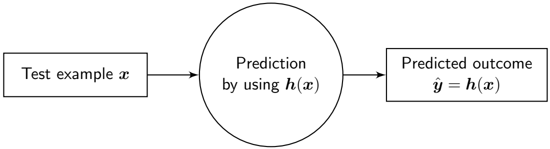
Supervised learning

Test example x

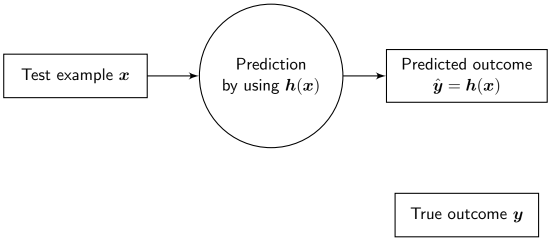
Supervised learning



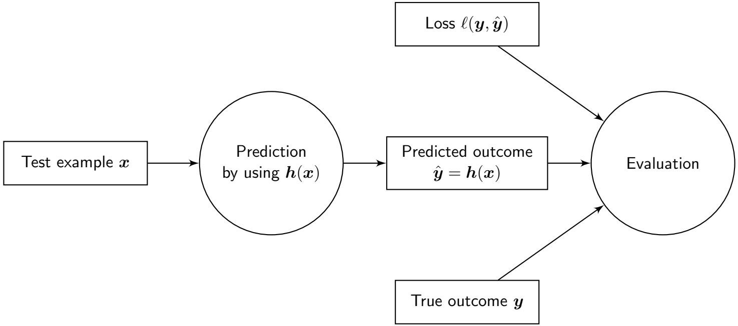
Supervised learning



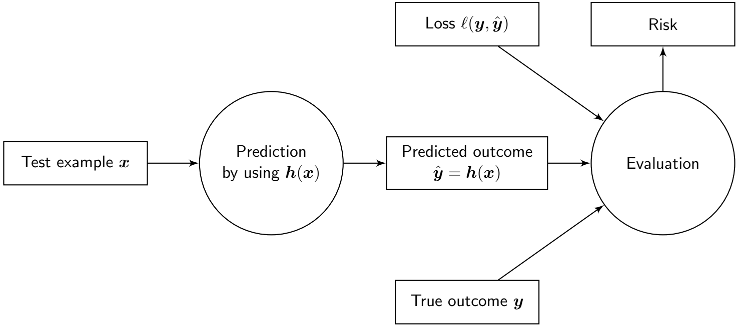
Supervised learning



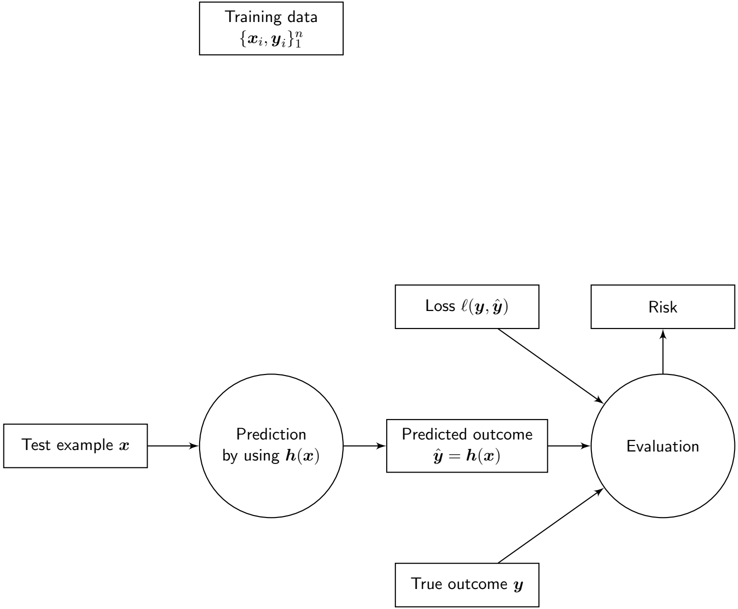
Supervised learning



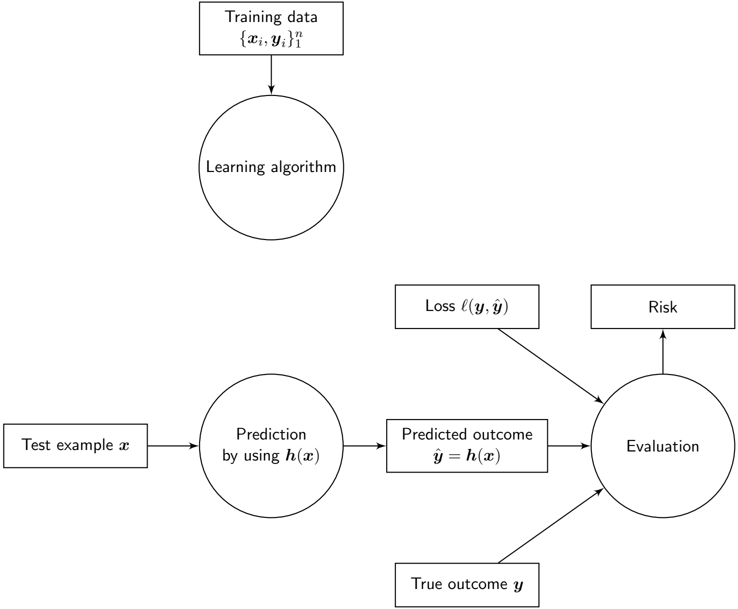
Supervised learning



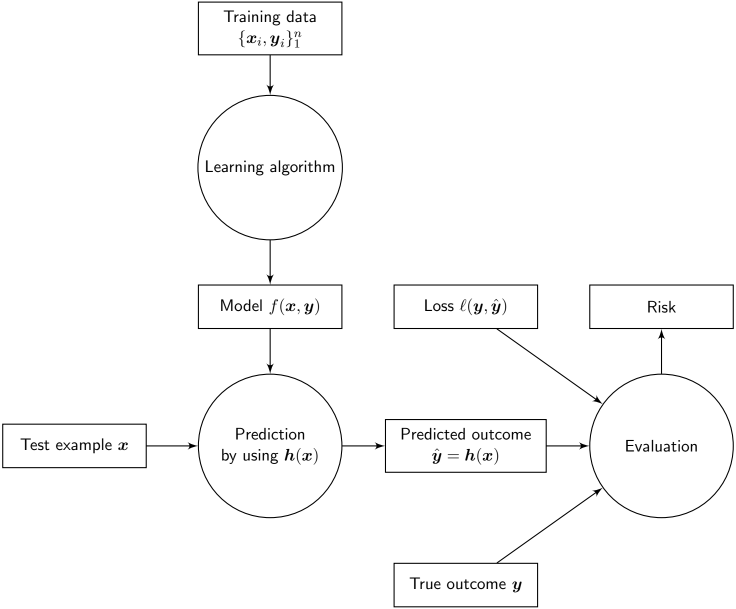
Supervised learning



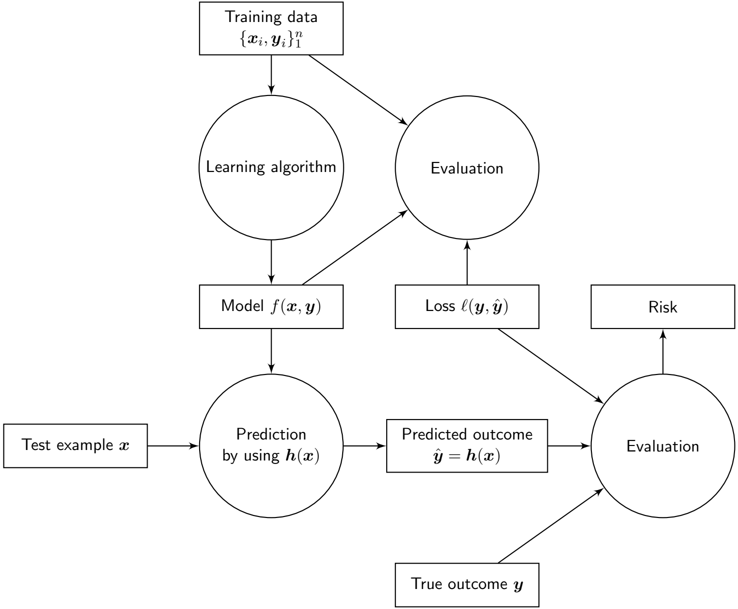
Supervised learning



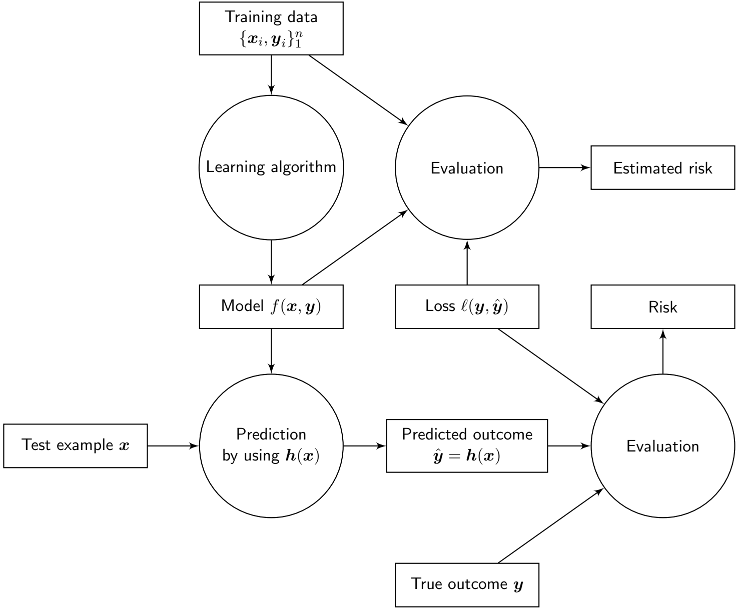
Supervised learning



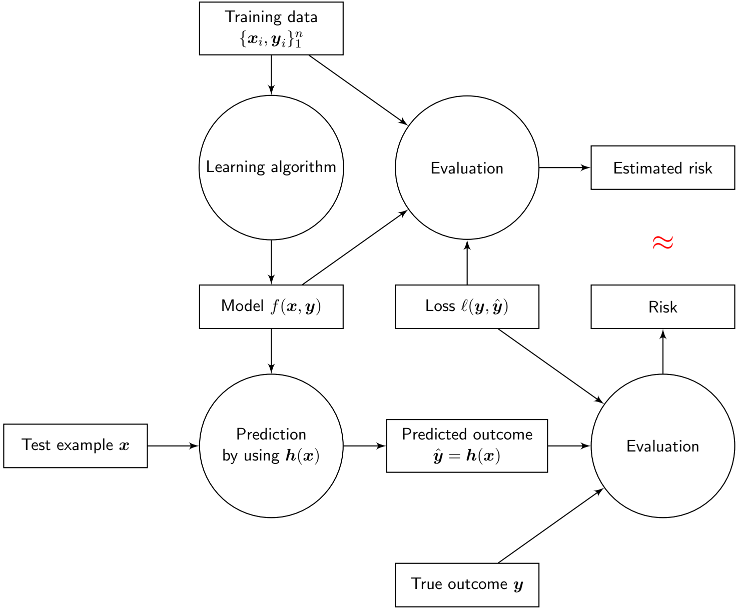
Supervised learning



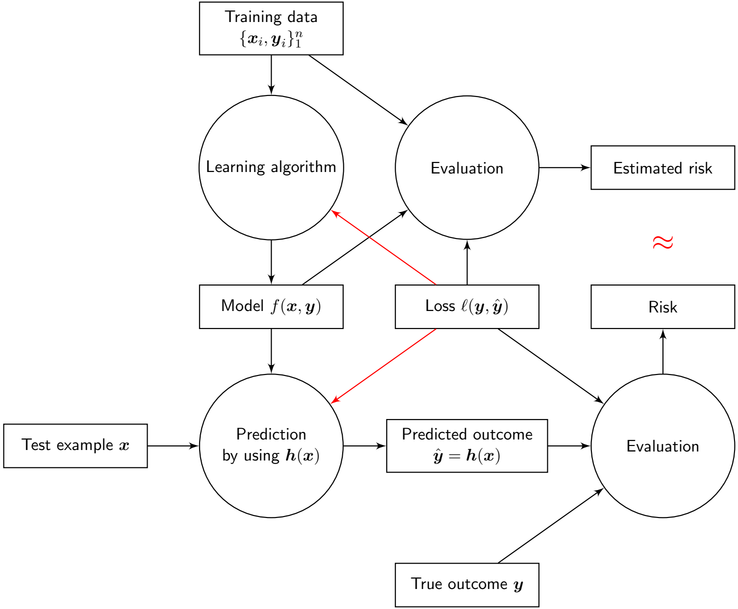
Supervised learning



Supervised learning



Supervised learning



Multi-label classification

- **Example** $\boldsymbol{x}$ is coming from an unknown input distribution $P(\boldsymbol{x})$.
- **True outcome** $\boldsymbol{y}$ is generated from $P(\boldsymbol{y} \,|\, \boldsymbol{x})$.
- **Predicted outcome** is given by $\hat{\boldsymbol{y}} = \boldsymbol{h}(\boldsymbol{x})$.
- The (**task**) **loss** of a single prediction is $\ell(\boldsymbol{y}, \hat{\boldsymbol{y}})$.

Multi-label classification

- The overall goal is to minimize the **risk**:

$$L_\ell(\boldsymbol{h}) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})}(\ell(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x})))$$

- The optimal prediction function, the so-called **Bayes classifier**, is:

$$\boldsymbol{h}^*_\ell = \arg\min_{\boldsymbol{h}} L_\ell(\boldsymbol{h})$$

- The **regret** of a classifier $\boldsymbol{h}$ with respect to ℓ is defined as:

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

Multi-label classification

- We use training examples $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}_1^n$ to find either:
  - a good approximation of $\boldsymbol{h}^*$, or
  - a good estimation of $P(\boldsymbol{y} \,|\, \boldsymbol{x})$ (or a function of it).
- In the second case, we need to apply an inference procedure to approximate $\boldsymbol{h}^*$.

Main challenges

- Appropriate modeling of dependencies between labels

$$y_1, y_2, \ldots, y_m$$

- A multitude of multivariate loss functions defined over the output vector

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

Label interdependences

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

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

$$P(\boldsymbol{y}) \neq \prod_{i=1}^{m} P(y_i) \qquad P(\boldsymbol{y} \,|\, \boldsymbol{x}) \neq \prod_{i=1}^{m} P(y_i \,|\, \boldsymbol{x})$$

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

Multi-label loss functions

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

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

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

Multi-label loss functions

- **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**: $\ell_{\text{rnk}}(\boldsymbol{y}, \boldsymbol{f}) = w(\boldsymbol{y}) \sum_{y_i > y_j} \left([\![f_i < f_j]\!] + \frac{1}{2} [\![f_i = f_j]\!] \right)$
- ...

Relations between losses

- The set-based 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 $y_i \neq h_i$.

Relations between losses

- The set-based 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.

Relations between losses

- The set-based 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.

Learning and inference with multi-label losses

- The loss functions, like Hamming loss or subset 0/1 loss, often referred to as **task losses**, are usually neither convex nor differentiable.

Learning and inference with multi-label 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.

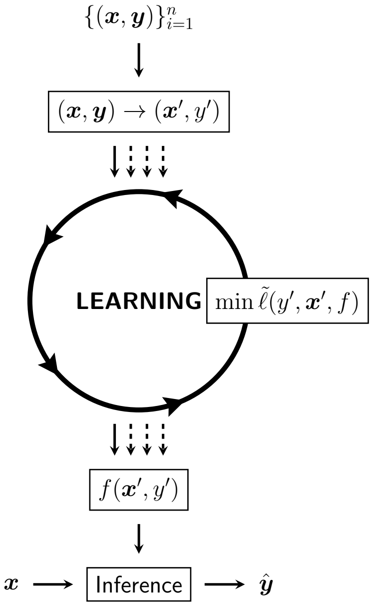
Learning and inference with multi-label losses

- 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.
  - Surrogate loss minimization.

Learning and inference with multi-label losses

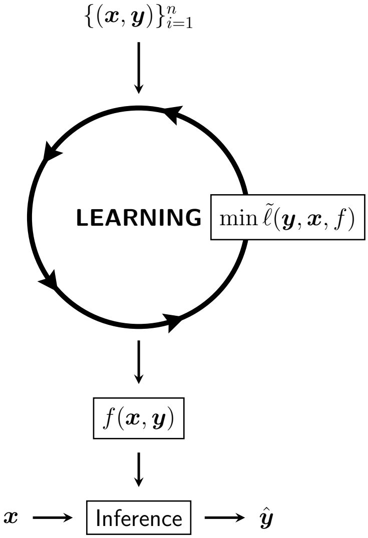
- The loss functions, like Hamming loss or subset $0/1$ loss, often referred to as **task losses**, are usually neither convex nor differentiable.
- Therefore learning is a hard optimization problem.
- Two approaches try to make this task easier
  - Reduction.
  - Surrogate loss minimization.
- Two phases in solving multi-label problems:
  - Learning: Estimate parameters of a 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 simple problems, for which efficient algorithmic solutions are available.
- Reduction to one or a sequence of problems.
- Plug-in rule classifiers.

Surrogate 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.[1]
- We say that a proxy loss $\tilde{\ell}$ is **consistent** (**calibrated**) with the task loss ℓ when the following holds:

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

- The definition concerns both surrogate loss minimization and reduction:
  - Surrogate loss minimization: $\tilde{\ell} =$ surrogate loss.
  - Reduction: $\tilde{\ell} =$ loss used in the reduced problem.

---

[1] 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

Outline

Basic reductions: Binary relevance

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

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

	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

Basic reductions: Binary relevance

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

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

	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

- Seems to be very simplistic.
- Ignores any dependencies.
- Is it good for any loss function?

Basic reductions: Label powerset

- **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

Basic reductions: Label powerset

- **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

- Any multi-class classification algorithm can be used, but the number of classes is huge.
- Takes other labels into account, but ignores internal structure of classes (label vectors).

What about task losses minimized by BR and LP?

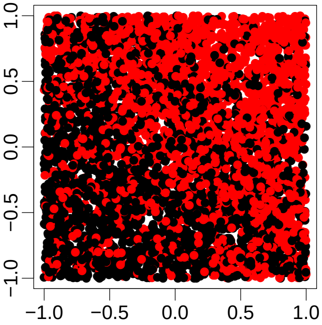
Synthetic data

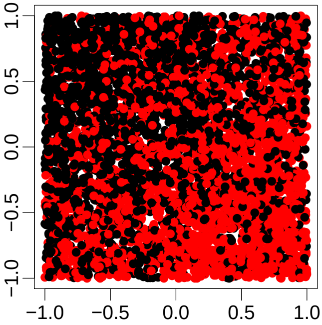
- Two independent models:

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

- Logistic model to get labels:

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





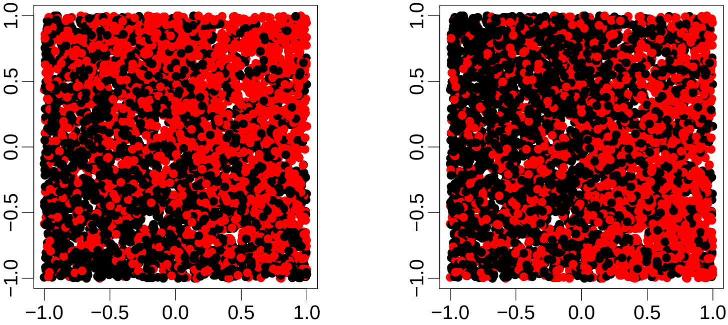
Synthetic data

- 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		
LP LR		

Conditional dependence		
classifier	Hamming loss	subset 0/1 loss
BR LR		
LP LR		

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		
LP LR		

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

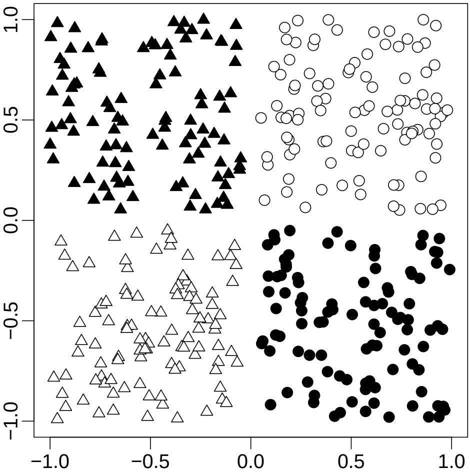


Figure: Problem with two targets: shapes (△ vs. ◦) and colors (□ vs. ■).

Linear + XOR synthetic data

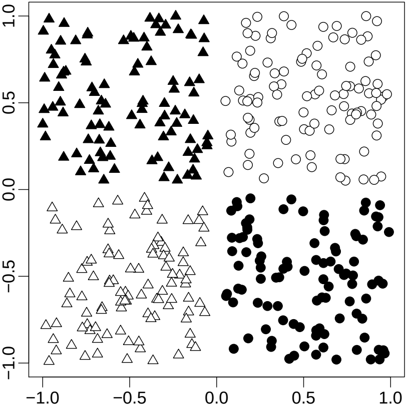
classifier	Hamming loss	subset 0/1 loss
BR LR	0.2399(±.0097)	0.4751(±.0196)
LP LR	0.0143(±.0020)	0.0195(±.0011)
Bayes Optimal	0	0

Linear + XOR synthetic data

CLASSIFIER	HAMMING LOSS	SUBSET 0/1 LOSS
BR LR	$0.2399(\pm.0097)$	$0.4751(\pm.0196)$
LP LR	$0.0143(\pm.0020)$	$0.0195(\pm.0011)$
BR MLRules	**$0.0011(\pm.0002)$**	**$0.0020(\pm.0003)$**
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-label loss functions

- The conditional risk in multi-label classification of $\boldsymbol{h}$ at $\boldsymbol{x}$:

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

- The risk-minimizing classifier for a given $\boldsymbol{x}$:

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

- Let us start with Hamming loss and subset 0/1 loss ...[2]

[2] 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

Hamming loss vs. subset 0/1 loss

- The risk minimizer for the Hamming loss is

Hamming loss vs. subset 0/1 loss

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

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

Hamming loss vs. subset 0/1 loss

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

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

while for the subset 0/1 loss is

Hamming loss vs. subset 0/1 loss

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

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

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

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

Hamming loss vs. subset 0/1 loss

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

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

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

Relations between losses

- The risk minimizers of Hamming and subset 0/1 loss are different: marginal mode vs. joint mode.

Relations between losses

- The risk minimizers of Hamming and subset $0/1$ loss are different: marginal mode vs. joint mode.
- Under specific conditions, like label independence or high probability of the joint mode (> 0.5), these two risk minimizers are equivalent.

Relations between losses

- The risk minimizers of Hamming and subset $0/1$ loss are different: marginal mode vs. joint mode.
- Under specific conditions, like label independence or high probability of the joint mode (> 0.5), these two risk minimizers are equivalent.
- The risks of these loss functions are mutually upper bounded.

Relations between losses

- The risk minimizers of Hamming and subset $0/1$ loss are different: marginal mode vs. joint mode.
- Under specific conditions, like label independence or high probability of the joint mode (> 0.5), 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.

BR vs. LP

BR vs. LP

- Binary relevance (BR)

BR vs. LP

- Binary relevance (BR)
  - BR is **consistent** for Hamming loss **without** any additional assumptions on **label (in)dependence**.

BR vs. LP

- Binary relevance (BR)
  - BR is **consistent** for Hamming loss **without** any additional assumptions on **label (in)dependence**.
  - If this would not be true, then **we could not optimally solve binary classification problems!!!**

BR vs. LP

- Binary relevance (BR)
  - BR is **consistent** for Hamming loss **without** any additional assumptions on **label (in)dependence**.
  - If this would not be true, then **we could not optimally solve binary classification problems!!!**
  - For other losses, one should take **additional assumptions**:
    - For subset $0/1$ loss: label independence, high probability of the joint mode (> 0.5), ...

BR vs. LP

- Binary relevance (BR)
  - BR is **consistent** for Hamming loss **without** any additional assumptions on **label (in)dependence**.
  - If this would not be true, then **we could not optimally solve binary classification problems!!!**
  - For other losses, one should 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)

BR vs. LP

- Label powerset (LP)
  - LP is **consistent** for the subset 0/1 loss.

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.

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 a probabilistic multi-class classifier, it estimates the joint conditional distribution for a given $\boldsymbol{x}$: inference for **any loss** would be then possible.

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 a 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**.

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 a 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**.

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 a 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).

Outline

Beyond LP

- **Classical multi-class classification algorithms**:
  - k-nearest neighbors,
  - Decision trees,
  - Logistic regression,
  - Multi-class SVMs,
  - ...
- **Reduction algorithms**:
  - 1 vs All,
  - 1 vs 1 and Weighted All-Pairs (WAP),
  - Directed acyclic graphs (DAG),
  - ECOC, PECOC, SECOC,
  - Filter Trees,
  - ...
- **Can we adapt these algorithms to multi-label classification and different task losses in a more direct way?**

Beyond LP

- Naive reduction to 1 vs. All:

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

Beyond LP

- Naive reduction to 1 vs. All:

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

- Reduction of **multi-class classification** to **binary classification**:

$$(\boldsymbol{x}, y = \mathrm{metaclass}(\boldsymbol{y})) \longrightarrow \{(\boldsymbol{x}, y, 1)\} \cup \{(\boldsymbol{x}, y', 0) : \forall y' \neq y\}$$

Beyond LP

- Naive reduction to 1 vs. All:

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

- Reduction of **multi-class classification** to **binary classification**:

$$(\boldsymbol{x}, y = \mathrm{metaclass}(\boldsymbol{y})) \longrightarrow \{(\boldsymbol{x}, y, 1)\} \cup \{(\boldsymbol{x}, y', 0) : \forall y' \neq y\}$$

- But we can reduce directly **multi-label classification** to **binary classification**:

$$(\boldsymbol{x}, \boldsymbol{y}) \longrightarrow \{(\boldsymbol{x}, \boldsymbol{y}, 1)\} \cup \{(\boldsymbol{x}, \boldsymbol{y}', 0) : \forall \boldsymbol{y}' \neq \boldsymbol{y}\}$$

Beyond LP

- Naive reduction to 1 vs. All:

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

- Reduction of **multi-class classification** to **binary classification**:

$$(\boldsymbol{x}, y = \mathrm{metaclass}(\boldsymbol{y})) \longrightarrow \{(\boldsymbol{x}, y, 1)\} \cup \{(\boldsymbol{x}, y', 0) : \forall y' \neq y\}$$

- But we can reduce directly **multi-label classification** to **binary classification**:

$$(\boldsymbol{x}, \boldsymbol{y}) \longrightarrow \{(\boldsymbol{x}, \boldsymbol{y}, 1)\} \cup \{(\boldsymbol{x}, \boldsymbol{y}', 0) : \forall \boldsymbol{y}' \neq \boldsymbol{y}\}$$

- **We can exploit now the internal structure of label vectors!!!**

Internal structure of classes

- The model can be given by a **scoring function** $f(\boldsymbol{x}, \boldsymbol{y})$.

Internal structure of classes

- The model can be given by a **scoring function** $f(\boldsymbol{x}, \boldsymbol{y})$.
- Different forms of $f(\boldsymbol{x}, \boldsymbol{y})$ are possible, for example:

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

Internal structure of classes

- The model can be given by a **scoring function** $f(\boldsymbol{x}, \boldsymbol{y})$.
- Different forms of $f(\boldsymbol{x}, \boldsymbol{y})$ are possible, for example:

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

  where the second term models pairwise interactions.

Internal structure of classes

- The model can be given by a **scoring function** $f(\boldsymbol{x}, \boldsymbol{y})$.
- Different forms of $f(\boldsymbol{x}, \boldsymbol{y})$ are possible, for example:

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

  where the second term models pairwise interactions.
- Prediction is given by:

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

Internal structure of classes

- Generalization of logistic regression and SVMs for $f(\boldsymbol{x}, \boldsymbol{y})$:
  - Conditional random fields (CRFs),[3]
  - Structured support vector machines (SSVMs).[4]

[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

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

CRFs and SSVMs

- CRFs use **logistic loss** as a surrogate:

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

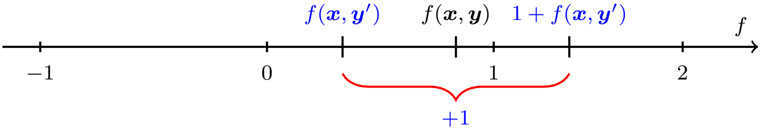
CRFs and SSVMs

- CRFs use **logistic loss** as a surrogate:

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

- SSVMs minimize the **structured hinge loss**:

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



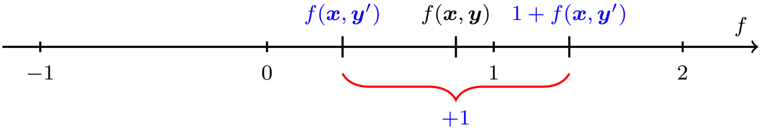
CRFs and SSVMs

- CRFs use **logistic loss** as a surrogate:

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

- SSVMs minimize the **structured hinge loss**:

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



- SSVMs and CRFs are quite similar to each other:
  - max vs. soft-max
  - margin vs. no-margin

CRFs and SSVMs

- Follow the general LP strategy, but can exploit the internal structure of classes within scoring function $f(\boldsymbol{x}, \boldsymbol{y})$.
- Convex optimization problem, but its hardness depends on the structure of $f(\boldsymbol{x}, \boldsymbol{y})$.
- Similarly, the inference (also known as decoding problem) is hard in the general case.
- For sequence and tree structures, the problem can be solved in polynomial time.

CRFs and SSVMs for different task losses

- In SSVMs, task loss $\ell(\boldsymbol{y}, \boldsymbol{y}')$ can be used for **margin rescaling**:

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

- SSVMs with Hamming loss and

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

  decompose to BR with SVMs.
- In general SSVMs are inconsistent.[5]

---

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

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

CRFs and SSVMs for different task losses

- CRFs are tailored for the subset 0/1 loss and cannot directly take other task losses into account.
- CRFs with the scoring function of the form

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

  minimize Hamming loss ($\rightarrow$ BR with logistic regression).
- Some works on incorporating margin into CRFs.[6]

---

[6] 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

SSVMs vs. BR

Table: SSVMs with pairwise term[7] vs. BR with LR[8].

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.

[7] Thomas Finley and Thorsten Joachims. Training structural SVMs when exact inference is intractable. In *ICML*. Omnipress, 2008

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

Probabilistic classifier chains

- Probabilistic classifier chains (PCCs)[9] are an efficient reduction method similar to conditional probability trees.[10]
- They estimate the joint conditional distribution $P(\boldsymbol{y} \,|\, \boldsymbol{x})$ as CRFs.
- The underlying idea is to repeatedly apply the **product rule of probability**:

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

[9] 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

[10] A. Beygelzimer, J. Langford, Y. Lifshits, G. B. Sorkin, and A. L. Strehl. Conditional probability tree estimation analysis and algorithms. In *UAI*, pages 51–58, 2009

Probabilistic classifier chains

- They follow a reduction to a sequence of subproblems:

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

- Each subproblem is solved independently by a probabilistic classifier estimating

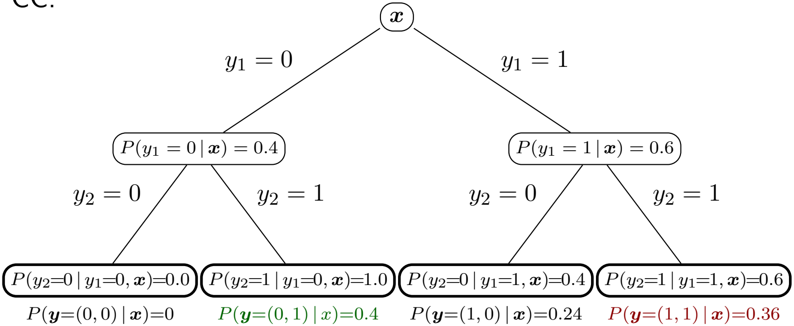
$$P(y_i = 1 \,|\, \boldsymbol{x}') \,.$$

- By using linear models in each task independently, 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:



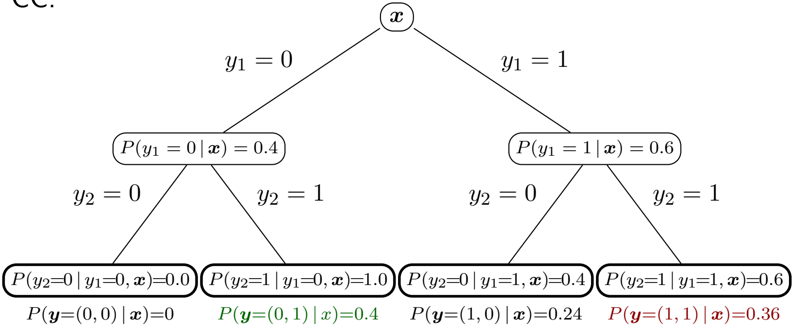
- For subset 0/1 loss one needs to find $\boldsymbol{h}(\boldsymbol{x}) = \arg\max_{\boldsymbol{y} \in \mathcal{Y}} P(\boldsymbol{y} \mid \boldsymbol{x})$.
- Greedy and approximate search techniques with guarantees exist.[11]

[11] 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 multi-label 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 $P(\boldsymbol{y} \mid \boldsymbol{x})$.[11]
- Sampling can be easily performed by using the probability tree.

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

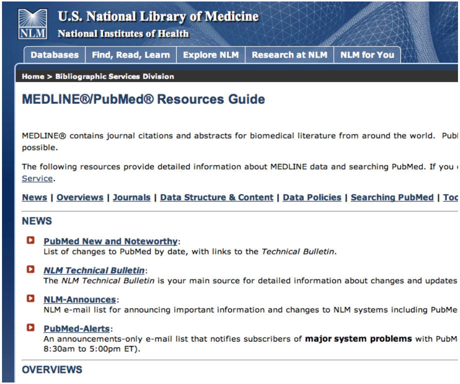
Table: PCC vs. SSVMs on Hamming loss and PCC vs. BR on subset 0/1 loss.

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

Outline

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:[12]

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.

---

[12] 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:[13]

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

- **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.

[13] Willem Waegeman, Krzysztof Dembczynski, Weiwei Cheng, and Eyke Hüllermeier. On the bayes-optimality of f-measure maximizers. *Journal of Machine Learning Research*, 15:3333–3388, 2014

Plug-in rule approach

- Approximation needed?

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

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

[16] 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!

---

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

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

[16] 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.[14]

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.[15]

Reduction to multinomial LR for each label.

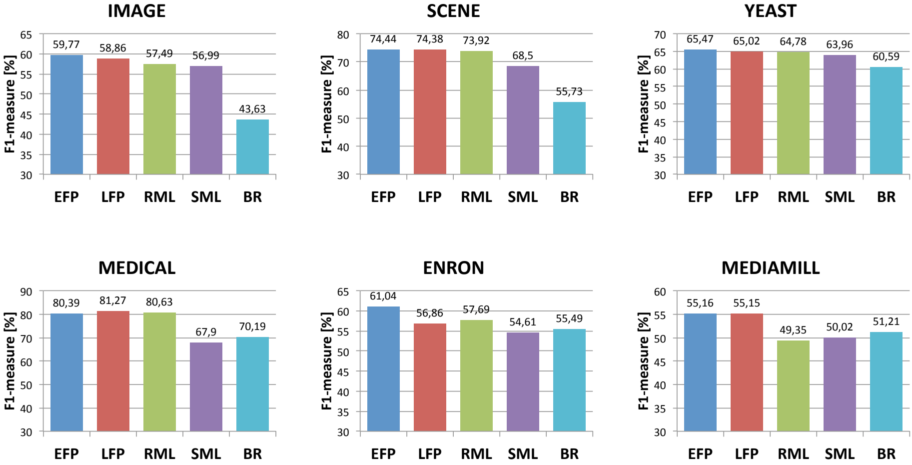
- EFP is consistent.[16]

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

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

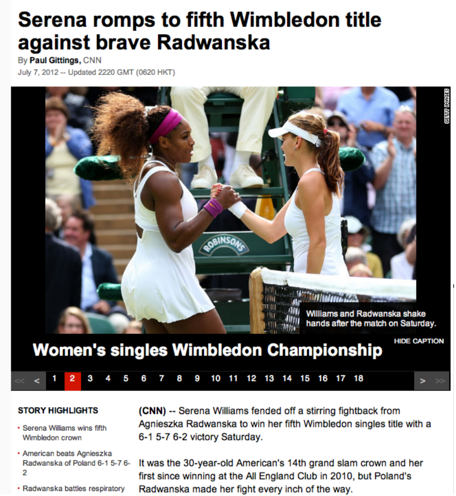
[16] K. Dembczynski, A. Jachnik, W. Kotlowski, W. Waegeman, and E. Hüllermeier. Optimizing the F-measure in multi-label classification: Plug-in rule approach versus structured loss minimization. In *ICML*, 2013

Maximization of the F-measure



Outline

Multi-label ranking



Serena romps to fifth Wimbledon title against brave Radwanska

By **Paul Gittings**, CNN

July 7, 2012 -- Updated 2220 GMT (0620 HKT)

STORY HIGHLIGHTS

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

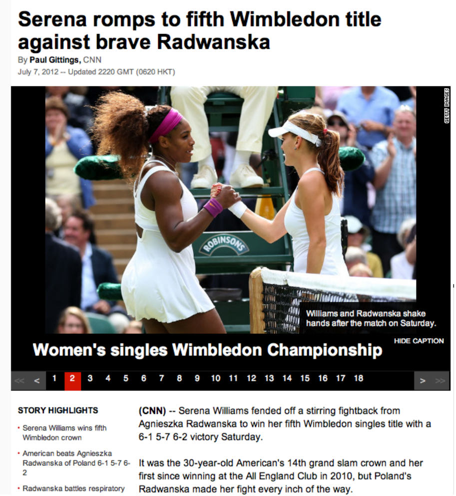
(CNN) -- Serena Williams fended off a stirring fightback from Agnieszka Radwanska to win her fifth Wimbledon singles title with a 6-1 5-7 6-2 victory Saturday.

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

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 ranking

tennis

$\succ$

sport

$\succ$

England

$\succ$

Poland

$\succ$

USA

$\succ$

$\vdots$

$\succ$

politics

Multi-label ranking

- **Ranking loss**:

$$\ell_{\mathrm{rnk}}(\boldsymbol{y}, \boldsymbol{f}) = w(\boldsymbol{y}) \sum_{(i,j)\,:\,y_i > y_j} \left([\![f_i(\boldsymbol{x}) < f_j(\boldsymbol{x})]\!] + \frac{1}{2} [\![f_i(\boldsymbol{x}) = f_j(\boldsymbol{x})]\!] \right) ,$$

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

Multi-label ranking

- **Ranking loss**:

$$\ell_{\mathrm{rnk}}(\boldsymbol{y}, \boldsymbol{f}) = w(\boldsymbol{y}) \sum_{(i,j)\,:\,y_i > y_j} \left([\![f_i(\boldsymbol{x}) < f_j(\boldsymbol{x})]\!] + \frac{1}{2} [\![f_i(\boldsymbol{x}) = f_j(\boldsymbol{x})]\!] \right) ,$$

  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

$$\tilde{\ell}_\phi(\boldsymbol{y}, \boldsymbol{f}) = \sum_{(i,j):\, y_i > y_j} w(\boldsymbol{y})\phi(f_i - f_j)\,,$$

  where ϕ is
  - an exponential function (BoosTexter)[17]: $\phi(f) = e^{-f}$,
  - logistic function (LLLR)[18]: $\phi(f) = \log(1 + e^{-f})$,
  - or hinge function (RankSVM)[19]: $\phi(f) = \max(0, 1 - f)$.

---

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

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

[19] 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.[20]
- The **consistent** classifier can be, however, obtained by using univariate loss functions[21] ...

[20] 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

[21] 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:

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

- 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:

$$\tilde{\ell}_{\exp}(\boldsymbol{y}, \boldsymbol{f}) = w(\boldsymbol{y}) \sum_{i=1}^{m} e^{-y' f_i} ,$$

$$\tilde{\ell}_{\log}(\boldsymbol{y}, \boldsymbol{f}) = w(\boldsymbol{y}) \sum_{i=1}^{m} \log \left(1 + e^{-y' f_i}\right) .$$

where $y' = 2y_i - 1$.

- The risk minimizer of these losses is:

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

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

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

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 $w = 1$.[22]
- Besides its **simplicity** and **efficiency**, this approach is **consistent** (regret bounds have also been derived).[23]

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

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

Weighted binary relevance

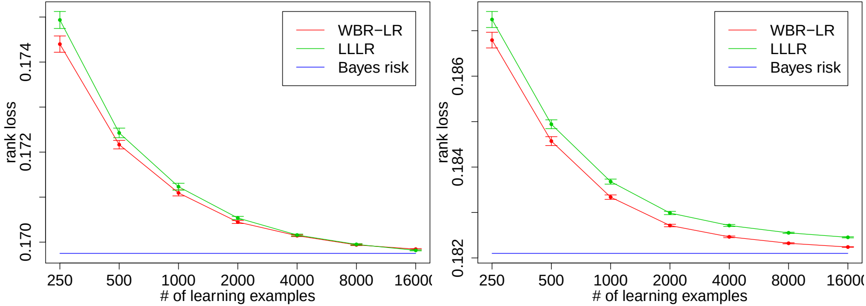


Figure: WBR LR vs. LLLR. Left: independent data. Right: dependent data.

- **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.

Outline

Open 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 issues: loss minimization and label dependence.
  - Two main approaches: surrogate loss minimization and reduction.
  - Consistency of algorithms.
  - High regret between solutions for different losses.
  - Proper modeling of label 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`