Extreme F-measure Maximization

Kalina Jasinska[1] Karlson Pfannschmidt[2]
Róbert Busa-Fekete[2] Krzysztof Dembczyński[1]

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

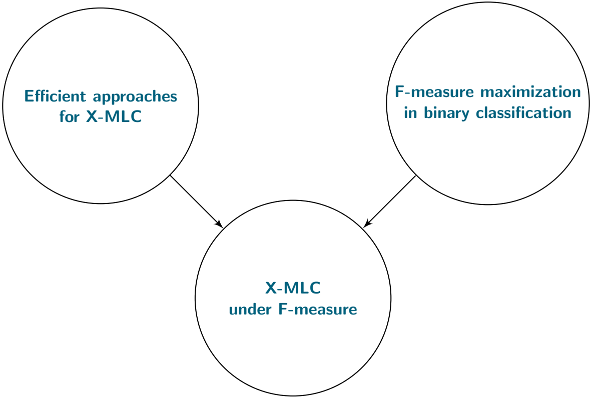
[2] Department of Computer Science, Paderborn University, Germany

XC15: Extreme Classification, The NIPS Workshop, 2015

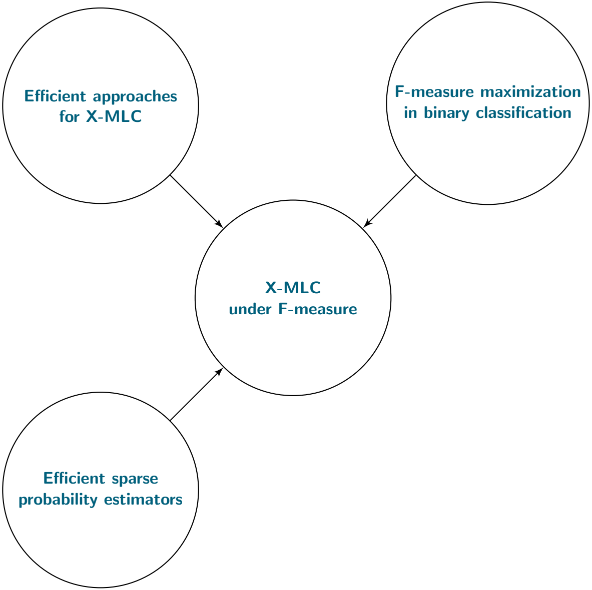
Efficient approaches for X-MLC

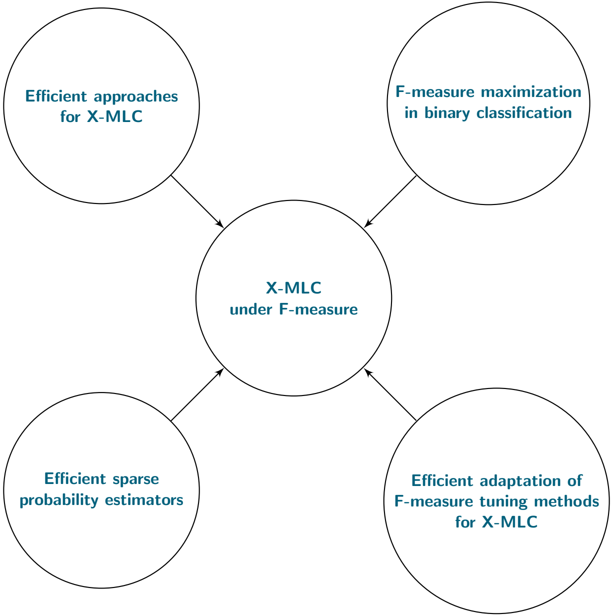
Efficient approaches for X-MLC

F-measure maximization in binary classification



Efficient approaches for X-MLC
F-measure maximization in binary classification
X-MLC under F-measure


Efficient approaches for X-MLC
F-measure maximization in binary classification
X-MLC under F-measure
Efficient sparse probability estimators


Efficient approaches for X-MLC
F-measure maximization in binary classification
X-MLC under F-measure
Efficient sparse probability estimators
Efficient adaptation of F-measure tuning methods for X-MLC

Outline

1. Extreme multi-label classification
2. The F-measure
3. Efficient sparse probability estimators
4. Experimental results
5. Summary

Outline

1 Extreme multi-label classification

2 The F-measure

3 Efficient sparse probability estimators

4 Experimental results

5 Summary

Multi-label classification

- For a feature vector $\boldsymbol{x}$ predict a binary vector $\boldsymbol{y}$ using a function $\boldsymbol{h}(\boldsymbol{x})$:

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

	x_1	x_2	$\ldots$	x_p	y_1	y_2	$\ldots$	y_m
$\boldsymbol{x}$	4.0	2.5		-1.5	?	?		?

Multi-label classification

- For a feature vector $\boldsymbol{x}$ predict a binary vector $\boldsymbol{y}$ using a function $\boldsymbol{h}(\boldsymbol{x})$:

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

	x_1	x_2	$\ldots$	x_p	y_1	y_2	$\ldots$	y_m
$\boldsymbol{x}$	4.0	2.5		-1.5	1	1		0

X-MLC – Extreme multi-label classification

- **Extreme** $\Rightarrow m \gg 10^4$

X-MLC – Extreme multi-label classification

- **Extreme** $\Rightarrow m \gg 10^4$
  - time and space complexity

X-MLC – Extreme multi-label classification

- **Extreme** $\Rightarrow m \gg 10^4$
  - time and space complexity
  - #examples vs. #features vs. #labels

X-MLC – Extreme multi-label classification

- **Extreme** $\Rightarrow m \gg 10^4$
  - time and space complexity
  - #examples vs. #features vs. #labels
  - training vs. validation vs. prediction

Outline

1 Extreme multi-label classification

2 The F-measure

3 Efficient sparse probability estimators

4 Experimental results

5 Summary

The F-measure

- Let $\boldsymbol{y} = (y_1, \ldots, y_m)$ be a binary label vector to be predicted.

The F-measure

- Let $\boldsymbol{y} = (y_1, \ldots, y_m)$ be a binary label vector to be predicted.
- Let $\hat{\boldsymbol{y}} = (\hat{y}_1, \ldots, \hat{y}_m)$ be a prediction of $\boldsymbol{y}$.

The F-measure

- Let $\boldsymbol{y} = (y_1, \ldots, y_m)$ be a binary label vector to be predicted.
- Let $\hat{\boldsymbol{y}} = (\hat{y}_1, \ldots, \hat{y}_m)$ be a prediction of $\boldsymbol{y}$.
- The **F-measure**:

$$F(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{2 \sum_{i=1}^{m} y_i \hat{y}_i}{\sum_{i=1}^{m} y_i + \sum_{i=1}^{m} \hat{y}_i} \in [0, 1] \, ,$$

where $0/0 = 1$ by definition.

The F-measure

- Let $\boldsymbol{y} = (y_1, \ldots, y_m)$ be a binary label vector to be predicted.
- Let $\hat{\boldsymbol{y}} = (\hat{y}_1, \ldots, \hat{y}_m)$ be a prediction of $\boldsymbol{y}$.
- The **F-measure**:

$$F(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{2 \sum_{i=1}^m y_i \hat{y}_i}{\sum_{i=1}^m y_i + \sum_{i=1}^m \hat{y}_i} \in [0, 1] \,,$$

  where $0/0 = 1$ by definition.
- It is a harmonic mean of precision $prec$ and recall $recl$:

$$prec(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{\sum_{i=1}^m y_i \hat{y}_i}{\sum_{i=1}^m \hat{y}_i} \,, \quad recl(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{\sum_{i=1}^m y_i \hat{y}_i}{\sum_{i=1}^m y_i} \,.$$

The F-measure

- The F-measure is better suited to **imbalanced** data than **accuracy**.

The F-measure

- The F-measure is better suited to **imbalanced** data than **accuracy**.
- **Example**:

The F-measure

- The F-measure is better suited to **imbalanced** data than **accuracy**.
- **Example**:
  - Let $P(y = 1) = 0.1$ and $P(y = 0) = 0.9$,

The F-measure

- The F-measure is better suited to **imbalanced** data than **accuracy**.
- **Example**:
  - Let $P(y = 1) = 0.1$ and $P(y = 0) = 0.9$,
  - Majority classifier $h(\boldsymbol{x})$ predicting always 0 will perform quite well in terms of accuracy, i.e., $P(y = h(\boldsymbol{x})) = 0.9$,

The F-measure

- The F-measure is better suited to **imbalanced** data than **accuracy**.
- **Example**:
  - Let $P(y = 1) = 0.1$ and $P(y = 0) = 0.9$,
  - Majority classifier $h(\boldsymbol{x})$ predicting always 0 will perform quite well in terms of accuracy, i.e., $P(y = h(\boldsymbol{x})) = 0.9$,
  - But the F-measure will be 0 in this case.

Optimal solution for the F-measure

- The F-measure in binary problems $\Rightarrow$ solved by **thresholding** conditional probabilities:

$$F(\tau) = \frac{2\int_{\mathcal{X}} \eta(\boldsymbol{x})\mathbb{I}\{\eta(\boldsymbol{x}) \geq \tau\}\,\mathrm{d}\mu(\boldsymbol{x})}{\int_{\mathcal{X}} \eta(\boldsymbol{x})\,\mathrm{d}\mu(\boldsymbol{x}) + \int_{\mathcal{X}} \mathbb{I}\{\eta(\boldsymbol{x}) \geq \tau\}\,\mathrm{d}\mu(\boldsymbol{x})}.$$

Optimal solution for the F-measure

- The F-measure in binary problems $\Rightarrow$ solved by **thresholding** conditional probabilities:

$$F(\tau) = \frac{2\int_{\mathcal{X}} \eta(\boldsymbol{x})\mathbb{I}\{\eta(\boldsymbol{x}) \geq \tau\}\,\mathrm{d}\mu(\boldsymbol{x})}{\int_{\mathcal{X}} \eta(\boldsymbol{x})\,\mathrm{d}\mu(\boldsymbol{x}) + \int_{\mathcal{X}} \mathbb{I}\{\eta(\boldsymbol{x}) \geq \tau\}\,\mathrm{d}\mu(\boldsymbol{x})}.$$

- The **optimal** threshold is

$$\tau^* = \arg\max_{\tau \in [0,1]} F(\tau)$$

Optimal solution for the F-measure

- The F-measure in binary problems $\Rightarrow$ solved by **thresholding** conditional probabilities:

$$F(\tau) = \frac{2\int_{\mathcal{X}} \eta(\boldsymbol{x})\mathbb{I}\{\eta(\boldsymbol{x}) \geq \tau\}\,\mathrm{d}\mu(\boldsymbol{x})}{\int_{\mathcal{X}} \eta(\boldsymbol{x})\,\mathrm{d}\mu(\boldsymbol{x}) + \int_{\mathcal{X}} \mathbb{I}\{\eta(\boldsymbol{x}) \geq \tau\}\,\mathrm{d}\mu(\boldsymbol{x})}.$$

- The **optimal** threshold is

$$\tau^* = \operatorname*{arg\,max}_{\tau \in [0,1]} F(\tau)$$

- The **optimal F-measure** is $F(\tau^*)$: no binary classifier can have a performance better than this.

Optimal solution for the F-measure

- Interestingly, the optimal solution satisfies the following condition:[1]

$$F^*(\tau) = 2\tau^* \,.$$

- Hence, it always holds that $\tau^* \leq 0.5$.
- This justifies the use of the F-measure in imbalance problems.

[1] Ming-Jie Zhao, Narayanan Edakunni, Adam Pocock, and Gavin Brown. Beyond Fano's inequality: Bounds on the Optimal F-Score, BER, and Cost-Sensitive Risk and Their Implications. *Journal of Machine Learning Research*, pages 1033–1090, 2013

Practical approaches

- **Tune** the threshold on **class probability estimates** (**CPEs**).
- At least three approaches:
  - **Fixed thresholds approach** (**FTA**),
  - **Sorting-based threshold optimization** (**STO**),
  - **Online F-measure optimization** (**OFO**).

Fixed thresholds approach

- Validate a **predefined set** of thresholds.

Fixed thresholds approach

- Validate a **predefined set** of thresholds.
- Performance depends on the number of thresholds used.

Fixed thresholds approach

- Validate a **predefined set** of thresholds.
- Performance depends on the number of thresholds used.
- Implementations with different **trade-offs** between **computational** and **space costs**:

Fixed thresholds approach

- Validate a **predefined set** of thresholds.
- Performance depends on the number of thresholds used.
- Implementations with different **trade-offs** between **computational** and **space costs**:
  - Compute and optionally store CPEs for all examples in the validation set and check the F-measure by passing the set of CPEs once for each predefined threshold.

Fixed thresholds approach

- Validate a **predefined set** of thresholds.
- Performance depends on the number of thresholds used.
- Implementations with different **trade-offs** between **computational** and **space costs**:
  - Compute and optionally store CPEs for all examples in the validation set and check the F-measure by passing the set of CPEs once for each predefined threshold.
  - Compute the F-measure for all thresholds simultaneously by passing the validation set only once (auxiliary variables needed for each of predefined thresholds).

Sorting-based threshold optimization

- **No predefined** thresholds.

Sorting-based threshold optimization

- **No predefined** thresholds.
- Two steps:

Sorting-based threshold optimization

- **No predefined** thresholds.
- Two steps:
  - Compute CPEs for validation examples and sort them.

Sorting-based threshold optimization

- **No predefined** thresholds.
- Two steps:
  - Compute CPEs for validation examples and sort them.
  - Verify potential thresholds as values between consecutive CPEs.

Sorting-based threshold optimization

- **No predefined** thresholds.
- Two steps:
  - Compute CPEs for validation examples and sort them.
  - Verify potential thresholds as values between consecutive CPEs.
- Requires **one pass** over CPEs.

Theoretical results

- Estimation of the threshold on a validation set is statistically consistent with provable regret bounds.[2]

[2] N. Nagarajan, S. Koyejo, R. Ravikumar, and I. Dhillon. Consistent binary classification with generalized performance metrics. In *NIPS 27*, pages 2744–2752, 2014

H. Narasimhan, R. Vaish, and Agarwal S. On the statistical consistency of plug-in classifiers for non-decomposable performance measures. In *NIPS*, 2014

Shameem Puthiya Parambath, Nicolas Usunier, and Yves Grandvalet. Optimizing f-measures by cost-sensitive classification. In *NIPS 27*, pages 2123–2131, 2014

Wojciech Kotłowski and Krzysztof Dembczynski. Surrogate regret bounds for generalized classification performance metrics. In *ACML*, 2015

Online F-measure optimization

- **Online update** of the threshold by exploiting that $F^*(\tau) = 2\tau^*$.

[3] Róbert Busa-Fekete, Balázs Szörényi, Krzysztof Dembczynski, and Eyke Hüllermeier. Online f-measure optimization. In *NIPS 29*, 2015

Online F-measure optimization

- **Online update** of the threshold by exploiting that $F^*(\tau) = 2\tau^*$.
- **Converges** to the optimal threshold.[3]

[3] Róbert Busa-Fekete, Balázs Szörényi, Krzysztof Dembczynski, and Eyke Hüllermeier. Online f-measure optimization. In *NIPS 29*, 2015

Online F-measure optimization

- **Online update** of the threshold by exploiting that $F^*(\tau) = 2\tau^*$.
- **Converges** to the optimal threshold.[3]
- Requires to store only a **small** constant number of auxiliary variables.

[3] Róbert Busa-Fekete, Balázs Szörényi, Krzysztof Dembczynski, and Eyke Hüllermeier. Online f-measure optimization. In *NIPS 29*, 2015

Online F-measure optimization

- **Online update** of the threshold by exploiting that $F^*(\tau) = 2\tau^*$.
- **Converges** to the optimal threshold.[3]
- Requires to store only a **small** constant number of auxiliary variables.
- Can be either applied on a validation set or run **simultaneously** with training of the class probability model.

[3] Róbert Busa-Fekete, Balázs Szörényi, Krzysztof Dembczynski, and Eyke Hüllermeier. Online f-measure optimization. In *NIPS 29*, 2015

Online F-measure optimization

- **Online update** of the threshold by exploiting that $F^*(\tau) = 2\tau^*$.
- **Converges** to the optimal threshold.[3]
- Requires to store only a **small** constant number of auxiliary variables.
- Can be either applied on a validation set or run **simultaneously** with training of the class probability model.
- For large validation sets **one pass** over data should get an accurate estimate of the threshold.

[3] Róbert Busa-Fekete, Balázs Szörényi, Krzysztof Dembczynski, and Eyke Hüllermeier. Online f-measure optimization. In *NIPS 29*, 2015

Online F-measure Maximization

- In each round t:

Online F-measure Maximization

- In each round t:
  - Example $\boldsymbol{x}_t$ is observed,

$\boldsymbol{x}_1$

Online F-measure Maximization

- In each round t:
  - Example $\boldsymbol{x}_t$ is observed,
  - Model g applied to $\boldsymbol{x}_t$ to get $\hat{\eta}(\boldsymbol{x}_t) = \hat{P}(y_t = 1 \,|\, \boldsymbol{x}_t)$,

$\boldsymbol{x}_1$

$\hat{\eta}(\boldsymbol{x}_1)$

Online F-measure Maximization

- In each round t:
  - Example $\boldsymbol{x}_t$ is observed,
  - Model g applied to $\boldsymbol{x}_t$ to get $\hat{\eta}(\boldsymbol{x}_t) = \hat{P}(y_t = 1 \mid \boldsymbol{x}_t)$,
  - Prediction $\hat{y}_t$ is computed by $\hat{y}_t = [\![\hat{\eta}(\boldsymbol{x}_t) \geq \tau_{t-1}]\!]$

$\boldsymbol{x}_1$

$\hat{\eta}(\boldsymbol{x}_1)$

$\hat{y}_1$

Online F-measure Maximization

- In each round t:
  - Example $\boldsymbol{x}_t$ is observed,
  - Model g applied to $\boldsymbol{x}_t$ to get $\hat{\eta}(\boldsymbol{x}_t) = \hat{P}(y_t = 1 \mid \boldsymbol{x}_t)$,
  - Prediction $\hat{y}_t$ is computed by $\hat{y}_t = [\![\hat{\eta}(\boldsymbol{x}_t) \geq \tau_{t-1}]\!]$
  - Label y_t is revealed.

$\boldsymbol{x}_1$

$\hat{\eta}(\boldsymbol{x}_1)$

$\hat{y}_1$

y_1

- In each round t:
  - Example $\boldsymbol{x}_t$ is observed,
  - Model g applied to $\boldsymbol{x}_t$ to get $\hat{\eta}(\boldsymbol{x}_t) = \hat{P}(y_t = 1 \mid \boldsymbol{x}_t)$,
  - Prediction $\hat{y}_t$ is computed by $\hat{y}_t = [\![\hat{\eta}(\boldsymbol{x}_t) \geq \tau_{t-1}]\!]$
  - Label y_t is revealed.
  - Threshold τ_t is computed by

$$\tau_t = \frac{F_t}{2} = \frac{a_t}{b_t}\,,$$

with $a_t = a_{t-1} + y_t \hat{y}_t$ and $b_t = b_{t-1} + y_t + \hat{y}_t$ (a_0 and $b_0 \to$ prior).

$\boldsymbol{x}_1$

$\hat{\eta}(\boldsymbol{x}_1)$

$\hat{y}_1$

y_1

τ_1

Online F-measure Maximization

- In each round t:
  - Example $\boldsymbol{x}_t$ is observed,
  - Model g applied to $\boldsymbol{x}_t$ to get $\hat{\eta}(\boldsymbol{x}_t) = \hat{P}(y_t = 1 \,|\, \boldsymbol{x}_t)$,
  - Prediction $\hat{y}_t$ is computed by $\hat{y}_t = [\![\hat{\eta}(\boldsymbol{x}_t) \geq \tau_{t-1}]\!]$
  - Label y_t is revealed.
  - Threshold τ_t is computed by

$$\tau_t = \frac{F_t}{2} = \frac{a_t}{b_t}\,,$$

with $a_t = a_{t-1} + y_t\hat{y}_t$ and $b_t = b_{t-1} + y_t + \hat{y}_t$ (a_0 and $b_0 \to$ prior).

$\boldsymbol{x}_1$	$\boldsymbol{x}_2$
$\hat{\eta}(\boldsymbol{x}_1)$	
$\hat{y}_1$	
y_1	
τ_1	

Online F-measure Maximization

- In each round t:
  - Example $\boldsymbol{x}_t$ is observed,
  - Model g applied to $\boldsymbol{x}_t$ to get $\hat{\eta}(\boldsymbol{x}_t) = \hat{P}(y_t = 1 \mid \boldsymbol{x}_t)$,
  - Prediction $\hat{y}_t$ is computed by $\hat{y}_t = [\![\hat{\eta}(\boldsymbol{x}_t) \geq \tau_{t-1}]\!]$
  - Label y_t is revealed.
  - Threshold τ_t is computed by

$$\tau_t = \frac{F_t}{2} = \frac{a_t}{b_t}\,,$$

with $a_t = a_{t-1} + y_t \hat{y}_t$ and $b_t = b_{t-1} + y_t + \hat{y}_t$ (a_0 and $b_0 \to$ prior).

$\boldsymbol{x}_1$	$\boldsymbol{x}_2$
$\hat{\eta}(\boldsymbol{x}_1)$	$\hat{\eta}(\boldsymbol{x}_2)$
$\hat{y}_1$	
y_1	
τ_1	

Online F-measure Maximization

- In each round t:
  - Example $\boldsymbol{x}_t$ is observed,
  - Model g applied to $\boldsymbol{x}_t$ to get $\hat{\eta}(\boldsymbol{x}_t) = \hat{P}(y_t = 1 \,|\, \boldsymbol{x}_t)$,
  - Prediction $\hat{y}_t$ is computed by $\hat{y}_t = [\![\hat{\eta}(\boldsymbol{x}_t) \geq \tau_{t-1}]\!]$
  - Label y_t is revealed.
  - Threshold τ_t is computed by

$$\tau_t = \frac{F_t}{2} = \frac{a_t}{b_t} \,,$$

with $a_t = a_{t-1} + y_t \hat{y}_t$ and $b_t = b_{t-1} + y_t + \hat{y}_t$ (a_0 and $b_0 \to$ prior).

$\boldsymbol{x}_1$	$\boldsymbol{x}_2$
$\hat{\eta}(\boldsymbol{x}_1)$	$\hat{\eta}(\boldsymbol{x}_2)$
$\hat{y}_1$	$\hat{y}_2$
y_1	
τ_1	

Online F-measure Maximization

- In each round t:
  - Example $\boldsymbol{x}_t$ is observed,
  - Model g applied to $\boldsymbol{x}_t$ to get $\hat{\eta}(\boldsymbol{x}_t) = \hat{P}(y_t = 1 \mid \boldsymbol{x}_t)$,
  - Prediction $\hat{y}_t$ is computed by $\hat{y}_t = [\![\hat{\eta}(\boldsymbol{x}_t) \geq \tau_{t-1}]\!]$
  - Label y_t is revealed.
  - Threshold τ_t is computed by

$$\tau_t = \frac{F_t}{2} = \frac{a_t}{b_t} \,,$$

with $a_t = a_{t-1} + y_t \hat{y}_t$ and $b_t = b_{t-1} + y_t + \hat{y}_t$ (a_0 and $b_0 \rightarrow$ prior).

$\boldsymbol{x}_1$	$\boldsymbol{x}_2$
$\hat{\eta}(\boldsymbol{x}_1)$	$\hat{\eta}(\boldsymbol{x}_2)$
$\hat{y}_1$	$\hat{y}_2$
y_1	y_2
τ_1	

Online F-measure Maximization

- In each round t:
  - Example $\boldsymbol{x}_t$ is observed,
  - Model g applied to $\boldsymbol{x}_t$ to get $\hat{\eta}(\boldsymbol{x}_t) = \hat{P}(y_t = 1 \,|\, \boldsymbol{x}_t)$,
  - Prediction $\hat{y}_t$ is computed by $\hat{y}_t = [\![\hat{\eta}(\boldsymbol{x}_t) \geq \tau_{t-1}]\!]$
  - Label y_t is revealed.
  - Threshold τ_t is computed by

$$\tau_t = \frac{F_t}{2} = \frac{a_t}{b_t}\,,$$

with $a_t = a_{t-1} + y_t \hat{y}_t$ and $b_t = b_{t-1} + y_t + \hat{y}_t$ (a_0 and $b_0 \to$ prior).

$\boldsymbol{x}_1$	$\boldsymbol{x}_2$
$\hat{\eta}(\boldsymbol{x}_1)$	$\hat{\eta}(\boldsymbol{x}_2)$
$\hat{y}_1$	$\hat{y}_2$
y_1	y_2
τ_1	τ_2

Online F-measure Maximization

- In each round t:
  - Example $\boldsymbol{x}_t$ is observed,
  - Model g applied to $\boldsymbol{x}_t$ to get $\hat{\eta}(\boldsymbol{x}_t) = \hat{P}(y_t = 1 \,|\, \boldsymbol{x}_t)$,
  - Prediction $\hat{y}_t$ is computed by $\hat{y}_t = [\![\hat{\eta}(\boldsymbol{x}_t) \geq \tau_{t-1}]\!]$
  - Label y_t is revealed.
  - Threshold τ_t is computed by

$$\tau_t = \frac{F_t}{2} = \frac{a_t}{b_t}\,,$$

with $a_t = a_{t-1} + y_t \hat{y}_t$ and $b_t = b_{t-1} + y_t + \hat{y}_t$ (a_0 and $b_0 \to$ prior).

$\boldsymbol{x}_1$	$\boldsymbol{x}_2$	$\boldsymbol{x}_3$
$\hat{\eta}(\boldsymbol{x}_1)$	$\hat{\eta}(\boldsymbol{x}_2)$	
$\hat{y}_1$	$\hat{y}_2$	
y_1	y_2	
τ_1	τ_2	

Online F-measure Maximization

- In each round t:
  - Example $\boldsymbol{x}_t$ is observed,
  - Model g applied to $\boldsymbol{x}_t$ to get $\hat{\eta}(\boldsymbol{x}_t) = \hat{P}(y_t = 1 \mid \boldsymbol{x}_t)$,
  - Prediction $\hat{y}_t$ is computed by $\hat{y}_t = [\![\hat{\eta}(\boldsymbol{x}_t) \geq \tau_{t-1}]\!]$
  - Label y_t is revealed.
  - Threshold τ_t is computed by

$$\tau_t = \frac{F_t}{2} = \frac{a_t}{b_t} \,,$$

with $a_t = a_{t-1} + y_t \hat{y}_t$ and $b_t = b_{t-1} + y_t + \hat{y}_t$ (a_0 and $b_0 \to$ prior).

$\boldsymbol{x}_1$	$\boldsymbol{x}_2$	$\boldsymbol{x}_3$
$\hat{\eta}(\boldsymbol{x}_1)$	$\hat{\eta}(\boldsymbol{x}_2)$	$\hat{\eta}(\boldsymbol{x}_3)$
$\hat{y}_1$	$\hat{y}_2$	
y_1	y_2	
τ_1	τ_2	

Online F-measure Maximization

- In each round t:
  - Example $\boldsymbol{x}_t$ is observed,
  - Model g applied to $\boldsymbol{x}_t$ to get $\hat{\eta}(\boldsymbol{x}_t) = \hat{P}(y_t = 1 \mid \boldsymbol{x}_t)$,
  - Prediction $\hat{y}_t$ is computed by $\hat{y}_t = [\![\hat{\eta}(\boldsymbol{x}_t) \geq \tau_{t-1}]\!]$
  - Label y_t is revealed.
  - Threshold τ_t is computed by

$$\tau_t = \frac{F_t}{2} = \frac{a_t}{b_t} \,,$$

with $a_t = a_{t-1} + y_t \hat{y}_t$ and $b_t = b_{t-1} + y_t + \hat{y}_t$ (a_0 and $b_0 \to$ prior).

$\boldsymbol{x}_1$	$\boldsymbol{x}_2$	$\boldsymbol{x}_3$
$\hat{\eta}(\boldsymbol{x}_1)$	$\hat{\eta}(\boldsymbol{x}_2)$	$\hat{\eta}(\boldsymbol{x}_3)$
$\hat{y}_1$	$\hat{y}_2$	$\hat{y}_3$
y_1	y_2	
τ_1	τ_2	

Online F-measure Maximization

- In each round t:
  - Example $\boldsymbol{x}_t$ is observed,
  - Model g applied to $\boldsymbol{x}_t$ to get $\hat{\eta}(\boldsymbol{x}_t) = \hat{P}(y_t = 1 \mid \boldsymbol{x}_t)$,
  - Prediction $\hat{y}_t$ is computed by $\hat{y}_t = [\![\hat{\eta}(\boldsymbol{x}_t) \geq \tau_{t-1}]\!]$
  - Label y_t is revealed.
  - Threshold τ_t is computed by

$$\tau_t = \frac{F_t}{2} = \frac{a_t}{b_t} \,,$$

with $a_t = a_{t-1} + y_t \hat{y}_t$ and $b_t = b_{t-1} + y_t + \hat{y}_t$ (a_0 and $b_0 \rightarrow$ prior).

$\boldsymbol{x}_1$	$\boldsymbol{x}_2$	$\boldsymbol{x}_3$
$\hat{\eta}(\boldsymbol{x}_1)$	$\hat{\eta}(\boldsymbol{x}_2)$	$\hat{\eta}(\boldsymbol{x}_3)$
$\hat{y}_1$	$\hat{y}_2$	$\hat{y}_3$
y_1	y_2	y_3
τ_1	τ_2	

Online F-measure Maximization

- In each round t:
  - Example $\boldsymbol{x}_t$ is observed,
  - Model g applied to $\boldsymbol{x}_t$ to get $\hat{\eta}(\boldsymbol{x}_t) = \hat{P}(y_t = 1 \,|\, \boldsymbol{x}_t)$,
  - Prediction $\hat{y}_t$ is computed by $\hat{y}_t = [\![\hat{\eta}(\boldsymbol{x}_t) \geq \tau_{t-1}]\!]$
  - Label y_t is revealed.
  - Threshold τ_t is computed by

$$\tau_t = \frac{F_t}{2} = \frac{a_t}{b_t} \,,$$

with $a_t = a_{t-1} + y_t \hat{y}_t$ and $b_t = b_{t-1} + y_t + \hat{y}_t$ (a_0 and $b_0 \rightarrow$ prior).

$\boldsymbol{x}_1$	$\boldsymbol{x}_2$	$\boldsymbol{x}_3$
$\hat{\eta}(\boldsymbol{x}_1)$	$\hat{\eta}(\boldsymbol{x}_2)$	$\hat{\eta}(\boldsymbol{x}_3)$
$\hat{y}_1$	$\hat{y}_2$	$\hat{y}_3$
y_1	y_2	y_3
τ_1	τ_2	τ_3

Online F-measure Maximization

- In each round t:
  - Example $\boldsymbol{x}_t$ is observed,
  - Model g applied to $\boldsymbol{x}_t$ to get $\hat{\eta}(\boldsymbol{x}_t) = \hat{P}(y_t = 1 \,|\, \boldsymbol{x}_t)$,
  - Prediction $\hat{y}_t$ is computed by $\hat{y}_t = [\![\hat{\eta}(\boldsymbol{x}_t) \geq \tau_{t-1}]\!]$
  - Label y_t is revealed.
  - Threshold τ_t is computed by

$$\tau_t = \frac{F_t}{2} = \frac{a_t}{b_t}\,,$$

with $a_t = a_{t-1} + y_t\hat{y}_t$ and $b_t = b_{t-1} + y_t + \hat{y}_t$ (a_0 and $b_0 \to$ prior).

$\boldsymbol{x}_1$	$\boldsymbol{x}_2$	$\boldsymbol{x}_3$	$\boldsymbol{x}_4$
$\hat{\eta}(\boldsymbol{x}_1)$	$\hat{\eta}(\boldsymbol{x}_2)$	$\hat{\eta}(\boldsymbol{x}_3)$	
$\hat{y}_1$	$\hat{y}_2$	$\hat{y}_3$	
y_1	y_2	y_3	
τ_1	τ_2	τ_3	

Online F-measure Maximization

- In each round t:
  - Example $\boldsymbol{x}_t$ is observed,
  - Model g applied to $\boldsymbol{x}_t$ to get $\hat{\eta}(\boldsymbol{x}_t) = \hat{P}(y_t = 1 \,|\, \boldsymbol{x}_t)$,
  - Prediction $\hat{y}_t$ is computed by $\hat{y}_t = [\![\hat{\eta}(\boldsymbol{x}_t) \geq \tau_{t-1}]\!]$
  - Label y_t is revealed.
  - Threshold τ_t is computed by

$$\tau_t = \frac{F_t}{2} = \frac{a_t}{b_t} \,,$$

with $a_t = a_{t-1} + y_t \hat{y}_t$ and $b_t = b_{t-1} + y_t + \hat{y}_t$ (a_0 and $b_0 \rightarrow$ prior).

$\boldsymbol{x}_1$	$\boldsymbol{x}_2$	$\boldsymbol{x}_3$	$\boldsymbol{x}_4$
$\hat{\eta}(\boldsymbol{x}_1)$	$\hat{\eta}(\boldsymbol{x}_2)$	$\hat{\eta}(\boldsymbol{x}_3)$	$\hat{\eta}(\boldsymbol{x}_4)$
$\hat{y}_1$	$\hat{y}_2$	$\hat{y}_3$	
y_1	y_2	y_3	
τ_1	τ_2	τ_3	

Online F-measure Maximization

- In each round t:
  - Example $\boldsymbol{x}_t$ is observed,
  - Model g applied to $\boldsymbol{x}_t$ to get $\hat{\eta}(\boldsymbol{x}_t) = \hat{P}(y_t = 1 \,|\, \boldsymbol{x}_t)$,
  - Prediction $\hat{y}_t$ is computed by $\hat{y}_t = [\![\hat{\eta}(\boldsymbol{x}_t) \geq \tau_{t-1}]\!]$
  - Label y_t is revealed.
  - Threshold τ_t is computed by

$$\tau_t = \frac{F_t}{2} = \frac{a_t}{b_t} \,,$$

with $a_t = a_{t-1} + y_t \hat{y}_t$ and $b_t = b_{t-1} + y_t + \hat{y}_t$ (a_0 and $b_0 \rightarrow$ prior).

$\boldsymbol{x}_1$	$\boldsymbol{x}_2$	$\boldsymbol{x}_3$	$\boldsymbol{x}_4$
$\hat{\eta}(\boldsymbol{x}_1)$	$\hat{\eta}(\boldsymbol{x}_2)$	$\hat{\eta}(\boldsymbol{x}_3)$	$\hat{\eta}(\boldsymbol{x}_4)$
$\hat{y}_1$	$\hat{y}_2$	$\hat{y}_3$	$\hat{y}_4$
y_1	y_2	y_3	
τ_1	τ_2	τ_3	

Online F-measure Maximization

- In each round t:
  - Example $\boldsymbol{x}_t$ is observed,
  - Model g applied to $\boldsymbol{x}_t$ to get $\hat{\eta}(\boldsymbol{x}_t) = \hat{P}(y_t = 1 \,|\, \boldsymbol{x}_t)$,
  - Prediction $\hat{y}_t$ is computed by $\hat{y}_t = [\![\hat{\eta}(\boldsymbol{x}_t) \geq \tau_{t-1}]\!]$
  - Label y_t is revealed.
  - Threshold τ_t is computed by

$$\tau_t = \frac{F_t}{2} = \frac{a_t}{b_t}\,,$$

with $a_t = a_{t-1} + y_t\hat{y}_t$ and $b_t = b_{t-1} + y_t + \hat{y}_t$ (a_0 and $b_0 \to$ prior).

$\boldsymbol{x}_1$	$\boldsymbol{x}_2$	$\boldsymbol{x}_3$	$\boldsymbol{x}_4$
$\hat{\eta}(\boldsymbol{x}_1)$	$\hat{\eta}(\boldsymbol{x}_2)$	$\hat{\eta}(\boldsymbol{x}_3)$	$\hat{\eta}(\boldsymbol{x}_4)$
$\hat{y}_1$	$\hat{y}_2$	$\hat{y}_3$	$\hat{y}_4$
y_1	y_2	y_3	y_4
τ_1	τ_2	τ_3	

Online F-measure Maximization

- In each round t:
  - Example $\boldsymbol{x}_t$ is observed,
  - Model g applied to $\boldsymbol{x}_t$ to get $\hat{\eta}(\boldsymbol{x}_t) = \hat{P}(y_t = 1 \,|\, \boldsymbol{x}_t)$,
  - Prediction $\hat{y}_t$ is computed by $\hat{y}_t = [\![\hat{\eta}(\boldsymbol{x}_t) \geq \tau_{t-1}]\!]$
  - Label y_t is revealed.
  - Threshold τ_t is computed by

$$\tau_t = \frac{F_t}{2} = \frac{a_t}{b_t}\,,$$

with $a_t = a_{t-1} + y_t\hat{y}_t$ and $b_t = b_{t-1} + y_t + \hat{y}_t$ (a_0 and $b_0 \to$ prior).

$\boldsymbol{x}_1$	$\boldsymbol{x}_2$	$\boldsymbol{x}_3$	$\boldsymbol{x}_4$
$\hat{\eta}(\boldsymbol{x}_1)$	$\hat{\eta}(\boldsymbol{x}_2)$	$\hat{\eta}(\boldsymbol{x}_3)$	$\hat{\eta}(\boldsymbol{x}_4)$
$\hat{y}_1$	$\hat{y}_2$	$\hat{y}_3$	$\hat{y}_4$
y_1	y_2	y_3	y_4
τ_1	τ_2	τ_3	τ_4

Online F-measure Maximization

- In each round t:
  - Example $\boldsymbol{x}_t$ is observed,
  - Model g applied to $\boldsymbol{x}_t$ to get $\hat{\eta}(\boldsymbol{x}_t) = \hat{P}(y_t = 1 \,|\, \boldsymbol{x}_t)$,
  - Prediction $\hat{y}_t$ is computed by $\hat{y}_t = [\![\hat{\eta}(\boldsymbol{x}_t) \geq \tau_{t-1}]\!]$
  - Label y_t is revealed.
  - Threshold τ_t is computed by

$$\tau_t = \frac{F_t}{2} = \frac{a_t}{b_t}\,,$$

with $a_t = a_{t-1} + y_t \hat{y}_t$ and $b_t = b_{t-1} + y_t + \hat{y}_t$ (a_0 and $b_0 \to$ prior).

$\boldsymbol{x}_1$	$\boldsymbol{x}_2$	$\boldsymbol{x}_3$	$\boldsymbol{x}_4$	$\boldsymbol{x}_5$
$\hat{\eta}(\boldsymbol{x}_1)$	$\hat{\eta}(\boldsymbol{x}_2)$	$\hat{\eta}(\boldsymbol{x}_3)$	$\hat{\eta}(\boldsymbol{x}_4)$	
$\hat{y}_1$	$\hat{y}_2$	$\hat{y}_3$	$\hat{y}_4$	
y_1	y_2	y_3	y_4	
τ_1	τ_2	τ_3	τ_4	

Online F-measure Maximization

- In each round t:
  - Example $\boldsymbol{x}_t$ is observed,
  - Model g applied to $\boldsymbol{x}_t$ to get $\hat{\eta}(\boldsymbol{x}_t) = \hat{P}(y_t = 1 \,|\, \boldsymbol{x}_t)$,
  - Prediction $\hat{y}_t$ is computed by $\hat{y}_t = [\![\hat{\eta}(\boldsymbol{x}_t) \geq \tau_{t-1}]\!]$
  - Label y_t is revealed.
  - Threshold τ_t is computed by

$$\tau_t = \frac{F_t}{2} = \frac{a_t}{b_t} \,,$$

with $a_t = a_{t-1} + y_t \hat{y}_t$ and $b_t = b_{t-1} + y_t + \hat{y}_t$ (a_0 and $b_0 \to$ prior).

$\boldsymbol{x}_1$	$\boldsymbol{x}_2$	$\boldsymbol{x}_3$	$\boldsymbol{x}_4$	$\boldsymbol{x}_5$
$\hat{\eta}(\boldsymbol{x}_1)$	$\hat{\eta}(\boldsymbol{x}_2)$	$\hat{\eta}(\boldsymbol{x}_3)$	$\hat{\eta}(\boldsymbol{x}_4)$	$\hat{\eta}(\boldsymbol{x}_5)$
$\hat{y}_1$	$\hat{y}_2$	$\hat{y}_3$	$\hat{y}_4$	
y_1	y_2	y_3	y_4	
τ_1	τ_2	τ_3	τ_4	

- In each round t:
  - Example $\boldsymbol{x}_t$ is observed,
  - Model g applied to $\boldsymbol{x}_t$ to get $\hat{\eta}(\boldsymbol{x}_t) = \hat{P}(y_t = 1 \,|\, \boldsymbol{x}_t)$,
  - Prediction $\hat{y}_t$ is computed by $\hat{y}_t = [\![\hat{\eta}(\boldsymbol{x}_t) \geq \tau_{t-1}]\!]$
  - Label y_t is revealed.
  - Threshold τ_t is computed by

$$\tau_t = \frac{F_t}{2} = \frac{a_t}{b_t} \,,$$

with $a_t = a_{t-1} + y_t \hat{y}_t$ and $b_t = b_{t-1} + y_t + \hat{y}_t$ (a_0 and $b_0 \rightarrow$ prior).

$\boldsymbol{x}_1$	$\boldsymbol{x}_2$	$\boldsymbol{x}_3$	$\boldsymbol{x}_4$	$\boldsymbol{x}_5$
$\hat{\eta}(\boldsymbol{x}_1)$	$\hat{\eta}(\boldsymbol{x}_2)$	$\hat{\eta}(\boldsymbol{x}_3)$	$\hat{\eta}(\boldsymbol{x}_4)$	$\hat{\eta}(\boldsymbol{x}_5)$
$\hat{y}_1$	$\hat{y}_2$	$\hat{y}_3$	$\hat{y}_4$	$\hat{y}_5$
y_1	y_2	y_3	y_4	
τ_1	τ_2	τ_3	τ_4	

Online F-measure Maximization

- In each round t:
  - Example $\boldsymbol{x}_t$ is observed,
  - Model g applied to $\boldsymbol{x}_t$ to get $\hat{\eta}(\boldsymbol{x}_t) = \hat{P}(y_t = 1 \,|\, \boldsymbol{x}_t)$,
  - Prediction $\hat{y}_t$ is computed by $\hat{y}_t = [\![\hat{\eta}(\boldsymbol{x}_t) \geq \tau_{t-1}]\!]$
  - Label y_t is revealed.
  - Threshold τ_t is computed by

$$\tau_t = \frac{F_t}{2} = \frac{a_t}{b_t} \,,$$

with $a_t = a_{t-1} + y_t\hat{y}_t$ and $b_t = b_{t-1} + y_t + \hat{y}_t$ (a_0 and $b_0 \to$ prior).

$\boldsymbol{x}_1$	$\boldsymbol{x}_2$	$\boldsymbol{x}_3$	$\boldsymbol{x}_4$	$\boldsymbol{x}_5$
$\hat{\eta}(\boldsymbol{x}_1)$	$\hat{\eta}(\boldsymbol{x}_2)$	$\hat{\eta}(\boldsymbol{x}_3)$	$\hat{\eta}(\boldsymbol{x}_4)$	$\hat{\eta}(\boldsymbol{x}_5)$
$\hat{y}_1$	$\hat{y}_2$	$\hat{y}_3$	$\hat{y}_4$	$\hat{y}_5$
y_1	y_2	y_3	y_4	y_5
τ_1	τ_2	τ_3	τ_4	

Online F-measure Maximization

- In each round t:
  - Example $\boldsymbol{x}_t$ is observed,
  - Model g applied to $\boldsymbol{x}_t$ to get $\hat{\eta}(\boldsymbol{x}_t) = \hat{P}(y_t = 1 \,|\, \boldsymbol{x}_t)$,
  - Prediction $\hat{y}_t$ is computed by $\hat{y}_t = [\![\hat{\eta}(\boldsymbol{x}_t) \geq \tau_{t-1}]\!]$
  - Label y_t is revealed.
  - Threshold τ_t is computed by

$$\tau_t = \frac{F_t}{2} = \frac{a_t}{b_t}\,,$$

with $a_t = a_{t-1} + y_t\hat{y}_t$ and $b_t = b_{t-1} + y_t + \hat{y}_t$ (a_0 and $b_0 \to$ prior).

$\boldsymbol{x}_1$	$\boldsymbol{x}_2$	$\boldsymbol{x}_3$	$\boldsymbol{x}_4$	$\boldsymbol{x}_5$
$\hat{\eta}(\boldsymbol{x}_1)$	$\hat{\eta}(\boldsymbol{x}_2)$	$\hat{\eta}(\boldsymbol{x}_3)$	$\hat{\eta}(\boldsymbol{x}_4)$	$\hat{\eta}(\boldsymbol{x}_5)$
$\hat{y}_1$	$\hat{y}_2$	$\hat{y}_3$	$\hat{y}_4$	$\hat{y}_5$
y_1	y_2	y_3	y_4	y_5
τ_1	τ_2	τ_3	τ_4	τ_5

Online F-measure Maximization

- In each round t:
  - Example $\boldsymbol{x}_t$ is observed,
  - Model g applied to $\boldsymbol{x}_t$ to get $\hat{\eta}(\boldsymbol{x}_t) = \hat{P}(y_t = 1 \,|\, \boldsymbol{x}_t)$,
  - Prediction $\hat{y}_t$ is computed by $\hat{y}_t = [\![\hat{\eta}(\boldsymbol{x}_t) \geq \tau_{t-1}]\!]$
  - Label y_t is revealed.
  - Threshold τ_t is computed by

$$\tau_t = \frac{F_t}{2} = \frac{a_t}{b_t} \,,$$

with $a_t = a_{t-1} + y_t \hat{y}_t$ and $b_t = b_{t-1} + y_t + \hat{y}_t$ (a_0 and $b_0 \rightarrow$ prior).

$\boldsymbol{x}_1$	$\boldsymbol{x}_2$	$\boldsymbol{x}_3$	$\boldsymbol{x}_4$	$\boldsymbol{x}_5$	
$\hat{\eta}(\boldsymbol{x}_1)$	$\hat{\eta}(\boldsymbol{x}_2)$	$\hat{\eta}(\boldsymbol{x}_3)$	$\hat{\eta}(\boldsymbol{x}_4)$	$\hat{\eta}(\boldsymbol{x}_5)$	
$\hat{y}_1$	$\hat{y}_2$	$\hat{y}_3$	$\hat{y}_4$	$\hat{y}_5$	$\ldots$
y_1	y_2	y_3	y_4	y_5	
τ_1	τ_2	τ_3	τ_4	τ_5	

Beyond binary problems

- All the above approaches are working well.

Beyond binary problems

- All the above approaches are working well.
- Computational issues can almost be ignored in binary problems.

Beyond binary problems

- All the above approaches are working well.
- Computational issues can almost be ignored in binary problems.
- **Scaling** to X-MLC?

Macro-averaging of the F-measure

- m labels.

Macro-averaging of the F-measure

- m labels.
- Test set of size n, $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_1^n$.

Macro-averaging of the F-measure

- m labels.
- Test set of size n, $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_1^n$.
- The true label vector: $\boldsymbol{y}_i = (y_{i1}, \ldots, y_{im})$.

Macro-averaging of the F-measure

- m labels.
- Test set of size n, $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_1^n$.
- The true label vector: $\boldsymbol{y}_i = (y_{i1}, \ldots, y_{im})$.
- The predicted label vector: $\hat{\boldsymbol{y}}_i = (\hat{y}_{i1}, \ldots, \hat{y}_{im})$.

Macro-averaging of the F-measure

- m labels.
- Test set of size n, $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_1^n$.
- The true label vector: $\boldsymbol{y}_i = (y_{i1}, \ldots, y_{im})$.
- The predicted label vector: $\hat{\boldsymbol{y}}_i = (\hat{y}_{i1}, \ldots, \hat{y}_{im})$.
- The **macro** F-measure:

$$F_M = \frac{1}{m} \sum_{j=1}^{m} F(\boldsymbol{y}_{\cdot j}, \hat{\boldsymbol{y}}_{\cdot j}) = \frac{1}{m} \sum_{j=1}^{m} \frac{2 \sum_{i=1}^{n} y_{ij} \hat{y}_{ij}}{\sum_{i=1}^{n} y_{ij} + \sum_{i=1}^{n} \hat{y}_{ij}} \,.$$

True labels

y_{11}	y_{12}	y_{13}	y_{14}
y_{21}	y_{22}	y_{23}	y_{24}
y_{31}	y_{32}	y_{33}	y_{34}
y_{41}	y_{42}	y_{43}	y_{44}
y_{51}	y_{52}	y_{53}	y_{54}
y_{61}	y_{62}	y_{63}	y_{64}

Predicted labels

$\hat{y}_{11}$	$\hat{y}_{12}$	$\hat{y}_{13}$	$\hat{y}_{14}$
$\hat{y}_{21}$	$\hat{y}_{22}$	$\hat{y}_{23}$	$\hat{y}_{24}$
$\hat{y}_{31}$	$\hat{y}_{32}$	$\hat{y}_{33}$	$\hat{y}_{34}$
$\hat{y}_{41}$	$\hat{y}_{42}$	$\hat{y}_{43}$	$\hat{y}_{44}$
$\hat{y}_{51}$	$\hat{y}_{52}$	$\hat{y}_{53}$	$\hat{y}_{54}$
$\hat{y}_{61}$	$\hat{y}_{62}$	$\hat{y}_{63}$	$\hat{y}_{64}$

Macro-averaging of the F-measure

- m labels.
- Test set of size n, $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_1^n$.
- The true label vector: $\boldsymbol{y}_i = (y_{i1}, \ldots, y_{im})$.
- The predicted label vector: $\hat{\boldsymbol{y}}_i = (\hat{y}_{i1}, \ldots, \hat{y}_{im})$.
- The **macro** F-measure:

$$F_M = \frac{1}{m} \sum_{j=1}^{m} F(\boldsymbol{y}_{\cdot j}, \hat{\boldsymbol{y}}_{\cdot j}) = \frac{1}{m} \sum_{j=1}^{m} \frac{2 \sum_{i=1}^{n} y_{ij} \hat{y}_{ij}}{\sum_{i=1}^{n} y_{ij} + \sum_{i=1}^{n} \hat{y}_{ij}} .$$

True labels

y_{11}	y_{12}	y_{13}	y_{14}
y_{21}	y_{22}	y_{23}	y_{24}
y_{31}	y_{32}	y_{33}	y_{34}
y_{41}	y_{42}	y_{43}	y_{44}
y_{51}	y_{52}	y_{53}	y_{54}
y_{61}	y_{62}	y_{63}	y_{64}

Predicted labels

$\hat{y}_{11}$	$\hat{y}_{12}$	$\hat{y}_{13}$	$\hat{y}_{14}$
$\hat{y}_{21}$	$\hat{y}_{22}$	$\hat{y}_{23}$	$\hat{y}_{24}$
$\hat{y}_{31}$	$\hat{y}_{32}$	$\hat{y}_{33}$	$\hat{y}_{34}$
$\hat{y}_{41}$	$\hat{y}_{42}$	$\hat{y}_{43}$	$\hat{y}_{44}$
$\hat{y}_{51}$	$\hat{y}_{52}$	$\hat{y}_{53}$	$\hat{y}_{54}$
$\hat{y}_{61}$	$\hat{y}_{62}$	$\hat{y}_{63}$	$\hat{y}_{64}$

Macro-averaging of the F-measure

- m labels.
- Test set of size n, $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_1^n$.
- The true label vector: $\boldsymbol{y}_i = (y_{i1}, \ldots, y_{im})$.
- The predicted label vector: $\hat{\boldsymbol{y}}_i = (\hat{y}_{i1}, \ldots, \hat{y}_{im})$.
- The **macro** F-measure:

$$F_M = \frac{1}{m}\sum_{j=1}^{m} F(\boldsymbol{y}_{\cdot j}, \hat{\boldsymbol{y}}_{\cdot j}) = \frac{1}{m}\sum_{j=1}^{m} \frac{2\sum_{i=1}^{n} y_{ij}\hat{y}_{ij}}{\sum_{i=1}^{n} y_{ij} + \sum_{i=1}^{n} \hat{y}_{ij}}\,.$$

True labels

y_{11}	y_{12}	y_{13}	y_{14}
y_{21}	y_{22}	y_{23}	y_{24}
y_{31}	y_{32}	y_{33}	y_{34}
y_{41}	y_{42}	y_{43}	y_{44}
y_{51}	y_{52}	y_{53}	y_{54}
y_{61}	y_{62}	y_{63}	y_{64}

Predicted labels

$\hat{y}_{11}$	$\hat{y}_{12}$	$\hat{y}_{13}$	$\hat{y}_{14}$
$\hat{y}_{21}$	$\hat{y}_{22}$	$\hat{y}_{23}$	$\hat{y}_{24}$
$\hat{y}_{31}$	$\hat{y}_{32}$	$\hat{y}_{33}$	$\hat{y}_{34}$
$\hat{y}_{41}$	$\hat{y}_{42}$	$\hat{y}_{43}$	$\hat{y}_{44}$
$\hat{y}_{51}$	$\hat{y}_{52}$	$\hat{y}_{53}$	$\hat{y}_{54}$
$\hat{y}_{61}$	$\hat{y}_{62}$	$\hat{y}_{63}$	$\hat{y}_{64}$

Macro-averaging of the F-measure

- m labels.
- Test set of size n, $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_1^n$.
- The true label vector: $\boldsymbol{y}_i = (y_{i1}, \ldots, y_{im})$.
- The predicted label vector: $\hat{\boldsymbol{y}}_i = (\hat{y}_{i1}, \ldots, \hat{y}_{im})$.
- The **macro** F-measure:

$$F_M = \frac{1}{m} \sum_{j=1}^{m} F(\boldsymbol{y}_{\cdot j}, \hat{\boldsymbol{y}}_{\cdot j}) = \frac{1}{m} \sum_{j=1}^{m} \frac{2 \sum_{i=1}^{n} y_{ij} \hat{y}_{ij}}{\sum_{i=1}^{n} y_{ij} + \sum_{i=1}^{n} \hat{y}_{ij}} .$$

True labels

y_{11}	y_{12}	y_{13}	y_{14}
y_{21}	y_{22}	y_{23}	y_{24}
y_{31}	y_{32}	y_{33}	y_{34}
y_{41}	y_{42}	y_{43}	y_{44}
y_{51}	y_{52}	y_{53}	y_{54}
y_{61}	y_{62}	y_{63}	y_{64}

Predicted labels

$\hat{y}_{11}$	$\hat{y}_{12}$	$\hat{y}_{13}$	$\hat{y}_{14}$
$\hat{y}_{21}$	$\hat{y}_{22}$	$\hat{y}_{23}$	$\hat{y}_{24}$
$\hat{y}_{31}$	$\hat{y}_{32}$	$\hat{y}_{33}$	$\hat{y}_{34}$
$\hat{y}_{41}$	$\hat{y}_{42}$	$\hat{y}_{43}$	$\hat{y}_{44}$
$\hat{y}_{51}$	$\hat{y}_{52}$	$\hat{y}_{53}$	$\hat{y}_{54}$
$\hat{y}_{61}$	$\hat{y}_{62}$	$\hat{y}_{63}$	$\hat{y}_{64}$

Macro-averaging of the F-measure

- m labels.
- Test set of size n, $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_1^n$.
- The true label vector: $\boldsymbol{y}_i = (y_{i1}, \ldots, y_{im})$.
- The predicted label vector: $\hat{\boldsymbol{y}}_i = (\hat{y}_{i1}, \ldots, \hat{y}_{im})$.
- The **macro** F-measure:

$$F_M = \frac{1}{m} \sum_{j=1}^{m} F(\boldsymbol{y}_{\cdot j}, \hat{\boldsymbol{y}}_{\cdot j}) = \frac{1}{m} \sum_{j=1}^{m} \frac{2 \sum_{i=1}^{n} y_{ij} \hat{y}_{ij}}{\sum_{i=1}^{n} y_{ij} + \sum_{i=1}^{n} \hat{y}_{ij}} .$$

True labels

y_{11}	y_{12}	y_{13}	y_{14}
y_{21}	y_{22}	y_{23}	y_{24}
y_{31}	y_{32}	y_{33}	y_{34}
y_{41}	y_{42}	y_{43}	y_{44}
y_{51}	y_{52}	y_{53}	y_{54}
y_{61}	y_{62}	y_{63}	y_{64}

Predicted labels

$\hat{y}_{11}$	$\hat{y}_{12}$	$\hat{y}_{13}$	$\hat{y}_{14}$
$\hat{y}_{21}$	$\hat{y}_{22}$	$\hat{y}_{23}$	$\hat{y}_{24}$
$\hat{y}_{31}$	$\hat{y}_{32}$	$\hat{y}_{33}$	$\hat{y}_{34}$
$\hat{y}_{41}$	$\hat{y}_{42}$	$\hat{y}_{43}$	$\hat{y}_{44}$
$\hat{y}_{51}$	$\hat{y}_{52}$	$\hat{y}_{53}$	$\hat{y}_{54}$
$\hat{y}_{61}$	$\hat{y}_{62}$	$\hat{y}_{63}$	$\hat{y}_{64}$

Macro-averaging of the F-measure

- Can be **solved** by **reduction** to m independent **binary** problems of F-measure maximization.[4]

[4] Oluwasanmi Koyejo, Nagarajan Natarajan, Pradeep Ravikumar, and Inderjit S. Dhillon. Consistent multilabel classification. In *NIPS 29*, dec 2015

Macro-averaging of the F-measure

- Can be **solved** by **reduction** to m independent **binary** problems of F-measure maximization.[4]
- Can we use the above threshold tuning methods?

[4] Oluwasanmi Koyejo, Nagarajan Natarajan, Pradeep Ravikumar, and Inderjit S. Dhillon. Consistent multilabel classification. In *NIPS 29*, dec 2015

Macro-averaging of the F-measure

- Can be **solved** by **reduction** to m independent **binary** problems of F-measure maximization.[4]
- Can we use the above threshold tuning methods?
- The **naive** adaptation of them can be **costly**!!!

[4] Oluwasanmi Koyejo, Nagarajan Natarajan, Pradeep Ravikumar, and Inderjit S. Dhillon. Consistent multilabel classification. In *NIPS 29*, dec 2015

Macro-averaging of the F-measure

- Can be **solved** by **reduction** to m independent **binary** problems of F-measure maximization.[4]
- Can we use the above threshold tuning methods?
- The **naive** adaptation of them can be **costly!!!**
  - We need CPEs for all labels and examples in the validation set.

[4] Oluwasanmi Koyejo, Nagarajan Natarajan, Pradeep Ravikumar, and Inderjit S. Dhillon. Consistent multilabel classification. In *NIPS 29*, dec 2015

Macro-averaging of the F-measure

- Can be **solved** by **reduction** to m independent **binary** problems of F-measure maximization.[4]
- Can we use the above threshold tuning methods?
- The **naive** adaptation of them can be **costly!!!**
  - We need CPEs for all labels and examples in the validation set.
  - For $m > 10^5$ and $n > 10^5$, we need at least 10^{10} predictions to be computed and potentially stored.

[4] Oluwasanmi Koyejo, Nagarajan Natarajan, Pradeep Ravikumar, and Inderjit S. Dhillon. Consistent multilabel classification. In *NIPS 29*, dec 2015

Macro-averaging of the F-measure

- Can be **solved** by **reduction** to m independent **binary** problems of F-measure maximization.[4]
- Can we use the above threshold tuning methods?
- The **naive** adaptation of them can be **costly!!!**
  - We need CPEs for all labels and examples in the validation set.
  - For $m > 10^5$ and $n > 10^5$, we need at least 10^{10} predictions to be computed and potentially stored.
- **Solution**:

[4] Oluwasanmi Koyejo, Nagarajan Natarajan, Pradeep Ravikumar, and Inderjit S. Dhillon. Consistent multilabel classification. In *NIPS 29*, dec 2015

Macro-averaging of the F-measure

- Can be **solved** by **reduction** to m independent **binary** problems of F-measure maximization.[4]
- Can we use the above threshold tuning methods?
- The **naive** adaptation of them can be **costly!!!**
  - We need CPEs for all labels and examples in the validation set.
  - For $m > 10^5$ and $n > 10^5$, we need at least 10^{10} predictions to be computed and potentially stored.
- **Solution**:
  - To compute the F-measure we need only true positive labels ($y_{ij} = 1$) and predicted positive labels ($\hat{y}_{ij} = 1$).

[4] Oluwasanmi Koyejo, Nagarajan Natarajan, Pradeep Ravikumar, and Inderjit S. Dhillon. Consistent multilabel classification. In *NIPS 29*, dec 2015

Macro-averaging of the F-measure

- Can be **solved** by **reduction** to m independent **binary** problems of F-measure maximization.[4]
- Can we use the above threshold tuning methods?
- The **naive** adaptation of them can be **costly!!!**
  - We need CPEs for all labels and examples in the validation set.
  - For $m > 10^5$ and $n > 10^5$, we need at least 10^{10} predictions to be computed and potentially stored.
- **Solution**:
  - To compute the F-measure we need only true positive labels ($y_{ij} = 1$) and predicted positive labels ($\hat{y}_{ij} = 1$).
  - Therefore to reduce the complexity we need to deliver **sparse probability estimates** (SPEs).

[4] Oluwasanmi Koyejo, Nagarajan Natarajan, Pradeep Ravikumar, and Inderjit S. Dhillon. Consistent multilabel classification. In *NIPS 29*, dec 2015

Outline

1. Extreme multi-label classification
2. The F-measure
3. Efficient sparse probability estimators
4. Experimental results
5. Summary

Efficient sparse probability estimators

- **Sparse propability estimates** (SPEs):

  CPEs of top labels or CPEs exceeding a given threshold

[5] Yashoteja Prabhu and Manik Varma. Fastxml: A fast, accurate and stable tree-classifier for extreme multi-label learning. In *KDD*, pages 263–272. ACM, 2014

[6] Kalina Jasinska and Krzysztof Dembczynski. Consistent label tree classifiers for extreme multi-label classification. In *The ICML Workshop on Extreme Classification*, 2015

Efficient sparse probability estimators

- **Sparse propability estimates** (SPEs):

  CPEs of top labels or CPEs exceeding a given threshold

- We need multi-label classifiers that efficiently deliver SPEs:

  Efficient sparse probability estimators

[5] Yashoteja Prabhu and Manik Varma. Fastxml: A fast, accurate and stable tree-classifier for extreme multi-label learning. In *KDD*, pages 263–272. ACM, 2014

[6] Kalina Jasinska and Krzysztof Dembczynski. Consistent label tree classifiers for extreme multi-label classification. In *The ICML Workshop on Extreme Classification*, 2015

Efficient sparse probability estimators

- **Sparse propability estimates** (SPEs):

  CPEs of top labels or CPEs exceeding a given threshold
- We need multi-label classifiers that efficiently deliver SPEs:

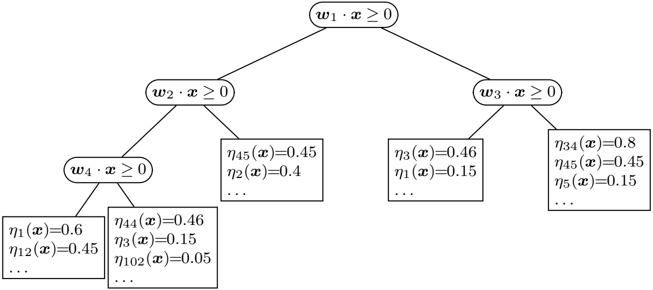
  Efficient sparse probability estimators
- Two examples: FastXML[5] and PLT[6]

[5] Yashoteja Prabhu and Manik Varma. Fastxml: A fast, accurate and stable tree-classifier for extreme multi-label learning. In *KDD*, pages 263–272. ACM, 2014

[6] Kalina Jasinska and Krzysztof Dembczynski. Consistent label tree classifiers for extreme multi-label classification. In *The ICML Workshop on Extreme Classification*, 2015

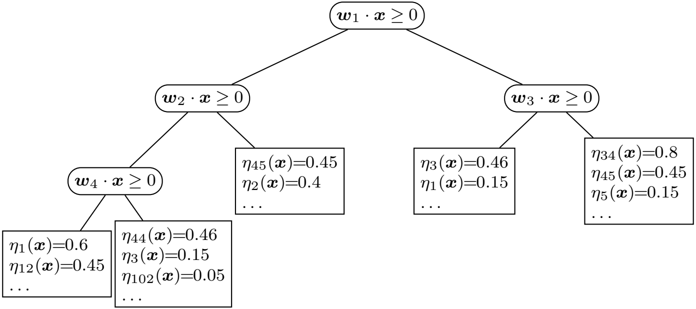
FastXML

- Based on standard **decision trees**.[7]
- Uses an **ensemble** of trees to improve predictive performance.
- **Sparse linear** classifiers trained to maximize **nDCG** in internal nodes.
- **Empirical distributions** in leaves.
- Very **efficient** training procedure.



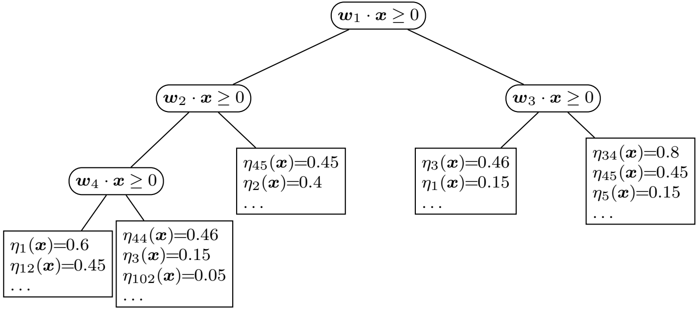
[7] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA, 1984

FastXML



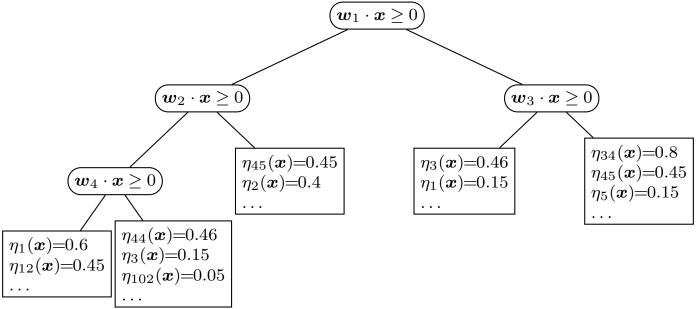
- Most importantly: **FastXML delivers SPEs**.

FastXML



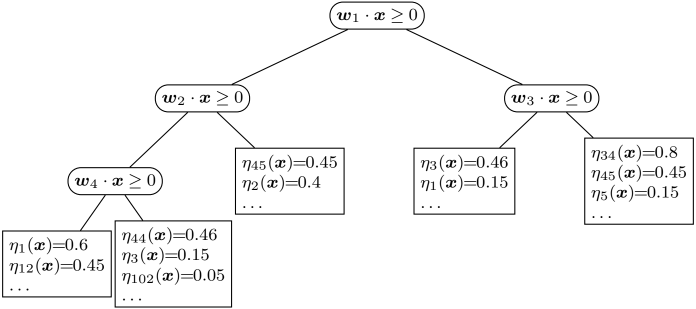
- Most importantly: **FastXML delivers SPEs**.
  - Leaf nodes cover only small feature space

FastXML



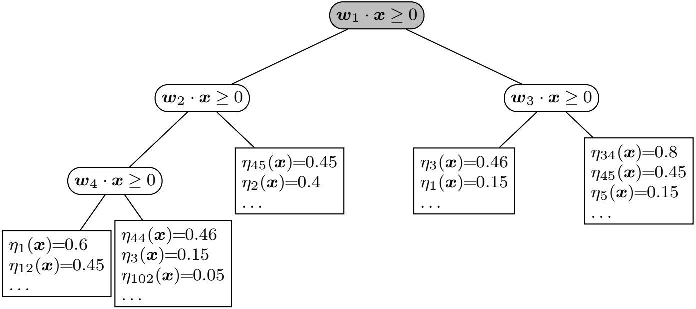
- Most importantly: **FastXML delivers SPEs**.
  - Leaf nodes cover only small feature space ⇒ small number of training examples in each leaf

FastXML



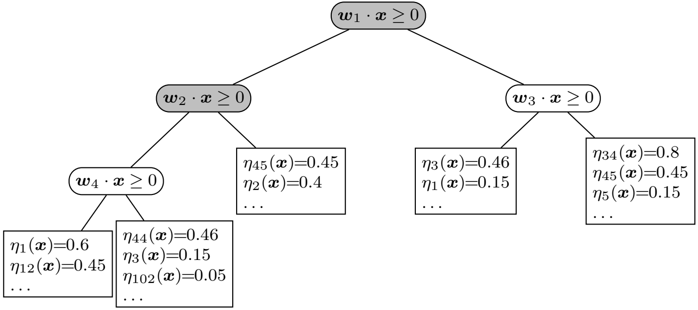
- Most importantly: **FastXML delivers SPEs**.
  - Leaf nodes cover only small feature space ⇒ small number of training examples in each leaf ⇒ small number of positive labels assigned to a leaf

FastXML



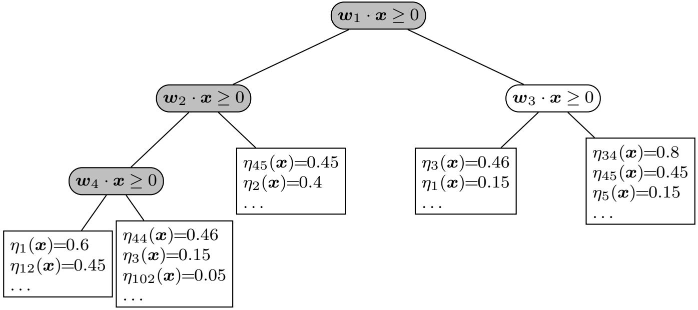
- Most importantly: **FastXML delivers SPEs**.
  - Leaf nodes cover only small feature space ⇒ small number of training examples in each leaf ⇒ small number of positive labels assigned to a leaf
  - Test example passes one path from the root to a leaf.

FastXML



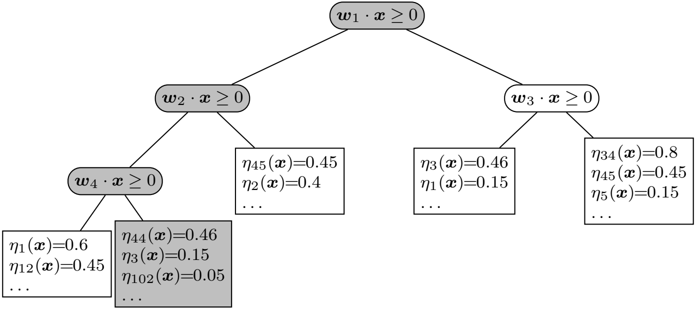
- Most importantly: **FastXML delivers SPEs**.
  - Leaf nodes cover only small feature space ⇒ small number of training examples in each leaf ⇒ small number of positive labels assigned to a leaf
  - Test example passes one path from the root to a leaf.

FastXML



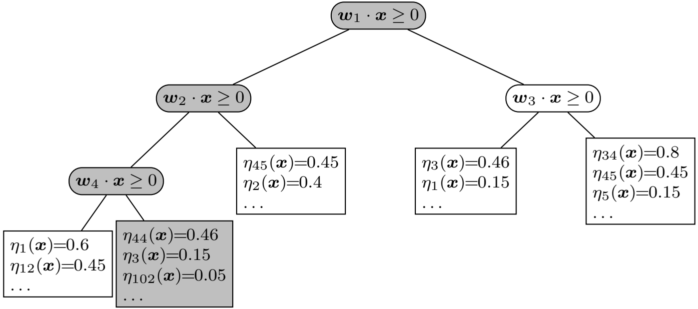
- Most importantly: **FastXML delivers SPEs**.
  - Leaf nodes cover only small feature space ⇒ small number of training examples in each leaf ⇒ small number of positive labels assigned to a leaf
  - Test example passes one path from the root to a leaf.

FastXML



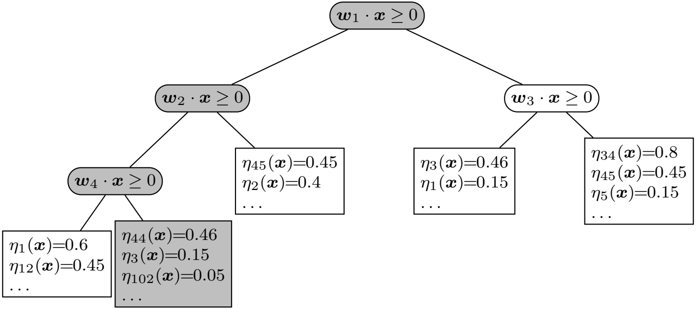
- Most importantly: **FastXML delivers SPEs**.
  - Leaf nodes cover only small feature space ⇒ small number of training examples in each leaf ⇒ small number of positive labels assigned to a leaf
  - Test example passes one path from the root to a leaf.

FastXML



- Most importantly: **FastXML delivers SPEs**.
  - Leaf nodes cover only small feature space ⇒ small number of training examples in each leaf ⇒ small number of positive labels assigned to a leaf
  - Test example passes one path from the root to a leaf.
  - Prediction based on the leaf node label distribution (zero probability for labels outside the leaf node).

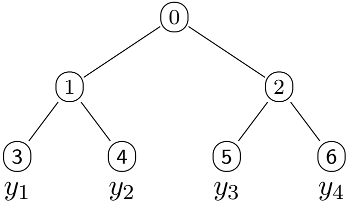
FastXML



- Most importantly: **FastXML delivers SPEs**.
  - Leaf nodes cover only small feature space ⇒ small number of training examples in each leaf ⇒ small number of positive labels assigned to a leaf
  - Test example passes one path from the root to a leaf.
  - Prediction based on the leaf node label distribution (zero probability for labels outside the leaf node).
  - The leaf node label distributions can be averaged over all trees in the ensemble.

Probabilistic label trees

- PLT are based on the **label tree approach**.[8]



- Each **leaf** node corresponds to one label.
- **Internal** node classifier decides whether to **go down the tree**.
- **Leaf** node classifier makes the **final prediction** about $\hat{y}_i$.
- A test example may follow many paths from the root to leaves.
- Each node j contains a class probability estimator $\eta(j)$ such that:

$$\eta_i(\boldsymbol{x}) = \prod_{j \in \mathrm{Path}(i)} \eta(j)\,.$$

[8] S. Bengio, J. Weston, and D. Grangier. Label embedding trees for large multi-class tasks. In *NIPS*, pages 163–171. Curran Associates, Inc., 2010

Probabilistic label trees

- Similar to **conditional probability trees**,[9] **probabilistic classifier chains**,[10] and **hierarchical softmax**,[11] but constructed to estimate **marginal** probabilities $\eta_i(\boldsymbol{x})$.
- Give probabilistic interpretation to **Homer**.[12]
- **Regret bounds**.[13]

---

[9] Alina Beygelzimer, John Langford, Yury Lifshits, Gregory B. Sorkin, and Alexander L. Strehl. Conditional probability tree estimation analysis and algorithms. In *UAI*, pages 51–58, 2009

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

[11] Frederic Morin and Yoshua Bengio. Hierarchical probabilistic neural network language model. In *AISTATS'05*, pages 246–252, 2005

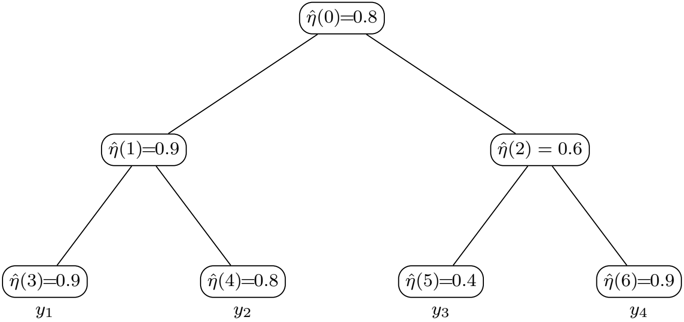
[12] G. Tsoumakas, I. Katakis, and I. Vlahavas. Effective and efficient multilabel classification in domains with large number of labels. In *Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data*, 2008

[13] Kalina Jasinska and Krzysztof Dembczynski. Consistent label tree classifiers for extreme multi-label classification. In *The ICML Workshop on Extreme Classification*, 2015

Probabilistic label trees

- Most importantly: **PLT delivers SPEs**.
  - Prediction relies on traversing the tree from the root to leaf nodes.
  - Pruning of subtrees if $p_j \leq t$ (e.g. $t = 0.5$):

Intermediate probability p_j: 1
Prediction $\hat{\boldsymbol{y}}$: $(0, 0, 0, 0)$

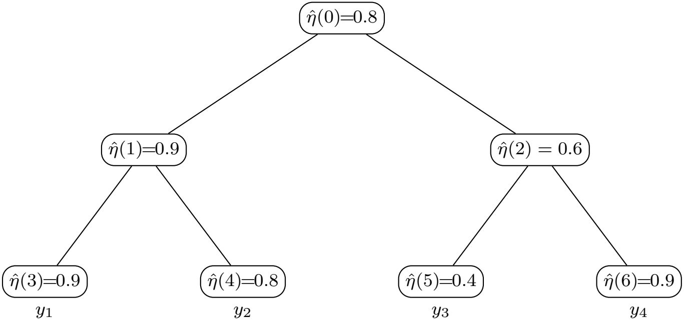


Queue $\mathcal{Q}$: []

Probabilistic label trees

- Most importantly: **PLT delivers SPEs**.
  - Prediction relies on traversing the tree from the root to leaf nodes.
  - Pruning of subtrees if $p_j \leq t$ (e.g. $t = 0.5$):

Intermediate probability p_j: 1
Prediction $\hat{\boldsymbol{y}}$: $(0, 0, 0, 0)$

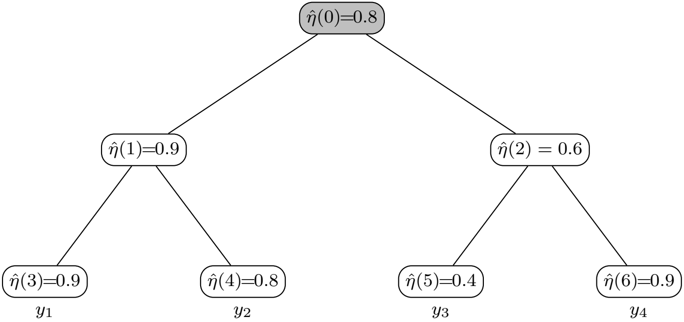


Queue $\mathcal{Q}$: $[(0, 1)]$

Probabilistic label trees

- Most importantly: **PLT delivers SPEs**.
  - Prediction relies on traversing the tree from the root to leaf nodes.
  - Pruning of subtrees if $p_j \leq t$ (e.g. $t = 0.5$):

Intermediate probability p_j: $\hat{\eta}(0) = 0.8$, $0.8 \geq 0.5$
Prediction $\hat{\boldsymbol{y}}$: $(0, 0, 0, 0)$

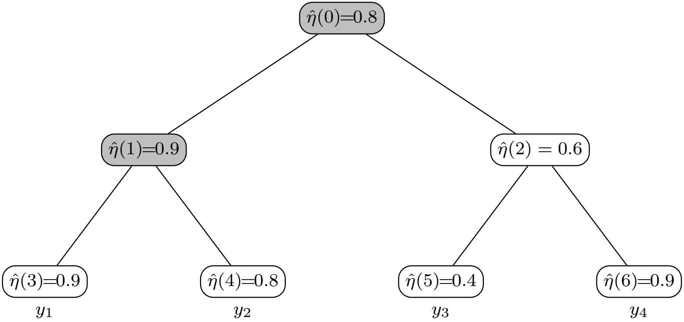


Queue $\mathcal{Q}$: $[(1, 0.8), (2, 0.8)]$

Probabilistic label trees

- Most important: **PLT delivers SPEs**.
  - Prediction relies on traversing the tree from the root to leaf nodes.
  - Pruning of subtrees if $p_j \leq t$ (e.g. $t = 0.5$):

Intermediate probability p_j: $\hat{\eta}(1) = 0.9$, $0.9 \cdot 0.8 = 0.72 \geq 0.5$
Prediction $\hat{\boldsymbol{y}}$: $(0, 0, 0, 0)$

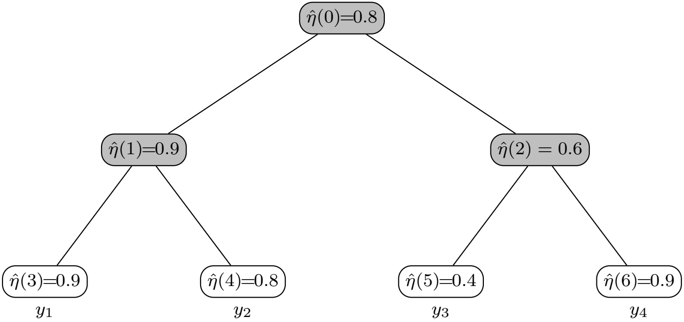


Queue $\mathcal{Q}$: $[(2, 0.8), (3, 0.72), (4, 0.72)]$

Probabilistic label trees

- Most importantly: **PLT delivers SPEs**.
  - Prediction relies on traversing the tree from the root to leaf nodes.
  - Pruning of subtrees if $p_j \leq t$ (e.g. $t = 0.5$):

Intermediate probability p_j: $\hat{\eta}(2) = 0.6$, $0.8 \cdot 0.6 = 0.48 < 0.5$
Prediction $\hat{\boldsymbol{y}}$: $(0, 0, 0, 0)$

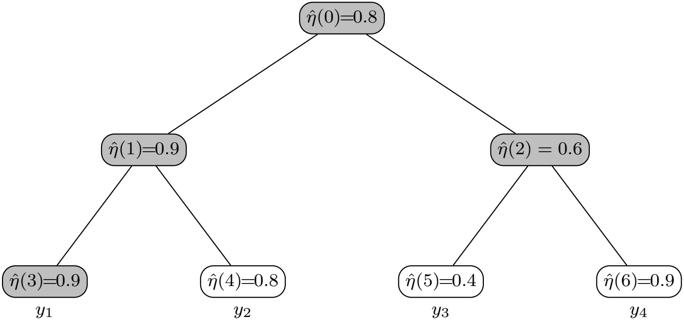


Queue $\mathcal{Q}$: $[(3, 0.72), (4, 0.72)]$

Probabilistic label trees

- Most importantly: **PLT delivers SPEs**.
  - Prediction relies on traversing the tree from the root to leaf nodes.
  - Pruning of subtrees if $p_j \leq t$ (e.g. $t = 0.5$):

Intermediate probability p_j: $\hat{\eta}(3) = 0.9$, $0.72 \cdot 0.9 \geq 0.5$
Prediction $\hat{\boldsymbol{y}}$: $(1, 0, 0, 0)$

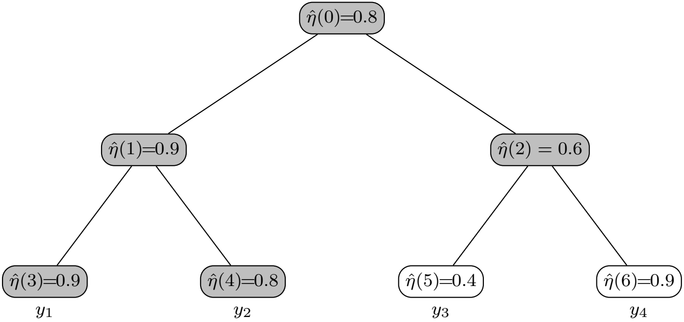


Queue $\mathcal{Q}$: $[(4, 0.72)]$

Probabilistic label trees

- Most importantly: **PLT delivers SPEs**.
  - Prediction relies on traversing the tree from the root to leaf nodes.
  - Pruning of subtrees if $p_j \leq t$ (e.g. $t = 0.5$):

Intermediate probability p_j: $\hat{\eta}(4) = 0.8$, $0.72 \cdot 0.8 \geq 0.5$
Prediction $\hat{\boldsymbol{y}}$: $(1, 1, 0, 0)$



Queue $\mathcal{Q}$: $[] \rightarrow STOP$

FastXML vs. PLT

	FastXML	PLT
tree structure	$\checkmark$	$\checkmark$
structure learning	$\checkmark$	$\times$
number of trees	≥ 1	1
number of leaves	$< m$	m
internal nodes models	linear	linear
leaves models	empirical distribution	linear
visited paths during prediction	1 per tree	several
sparse probability estimation	$\checkmark$	$\checkmark$

Outline

1 Extreme multi-label classification

2 The F-measure

3 Efficient sparse probability estimators

4 Experimental results

5 Summary

Experimental results

Table: Main statistics of datasets.

	Wiki1K	WikiLSHTC
#labels	933	325056
#features	196366	1617899
#examples	108738	2365435
avg. cardinality	1.71	3.26
max cardinality	14	198
cardinality $>$2	41%	72%
Hamming loss (%) of all-zero classifier	0.1833	1.003536E-05

Experimental results

Table: Results on Wiki1K.

τ	macro-F	HL
FastXML + FTA $\tau = 0.05$	0.303	3.038E-03
FastXML + FTA $\tau = 0.10$	**0.326**	1.680E-03
FastXML + FTA $\tau = 0.15$	0.315	1.285E-03
FastXML + FTA $\tau = 0.20$	0.298	1.128E-03
FastXML + FTA $\tau = 0.25$	0.277	1.058E-03
FastXML + FTA $\tau = 0.30$	0.254	1.031E-03
FastXML + FTA $\tau = 0.35$	0.233	**1.017E-03**
FastXML + FTA $\tau = 0.40$	0.215	1.018E-03
FastXML + FTA $\tau = 0.45$	0.196	1.029E-03
FastXML + FTA $\tau = 0.50$	0.179	1.051E-03
FastXML + STO	**0.379**	3.121E-03
FastXML + OFO (10 epoch, $a_0 = 0, b_0 = 350$)	0.353	7.353E-03

	P@1	P@2	P@3	P@4	P@5
FastXML	0.785	0.548	0.415	0.330	0.274

Experimental results

Table: Results on Wiki1K.

τ	macro-F	HL
PLT + FTA $\tau = 0.05$	0.301	3.895E-03
PLT + FTA $\tau = 0.10$	**0.313**	2.155E-03
PLT + FTA $\tau = 0.15$	0.299	1.600E-03
PLT + FTA $\tau = 0.20$	0.278	1.344E-03
PLT + FTA $\tau = 0.25$	0.252	1.219E-03
PLT + FTA $\tau = 0.30$	0.229	1.151E-03
PLT + FTA $\tau = 0.35$	0.206	1.122E-03
PLT + FTA $\tau = 0.40$	0.185	**1.114E-03**
PLT + FTA $\tau = 0.45$	0.165	1.120E-03
PLT + FTA $\tau = 0.50$	0.147	1.136E-03
PLT + STO	**0.331**	1.892E-03
PLT + OFO (1 epoch, $a_0 = 20, b_0 = 200$)	0.321	1.605E-03

	P@1	P@2	P@3	P@4	P@5
PLT	0.750	0.519	0.372	0.279	0.224

Experimental results

Table: Results on WikiLSHTC.

τ	macro-F	HL
FastXML + FTA $\tau = 0.05$	**0.076**	1.592E-05
FastXML + FTA $\tau = 0.10$	0.060	1.058E-05
FastXML + FTA $\tau = 0.15$	0.048	9.395E-06
FastXML + FTA $\tau = 0.20$	0.039	8.985E-06
FastXML + FTA $\tau = 0.25$	0.033	8.834E-06
FastXML + FTA $\tau = 0.30$	0.028	**8.789E-06**
FastXML + FTA $\tau = 0.35$	0.023	8.798E-06
FastXML + FTA $\tau = 0.40$	0.019	8.838E-06
FastXML + FTA $\tau = 0.45$	0.016	8.893E-06
FastXML + FTA $\tau = 0.50$	0.014	8.964E-06
FastXML + STO	**0.080**	8.121E-05
FastXML + OFO (1 epoch, $a_0 = 18, b_0 = 360$)	0.078	1.080E-05

	P@1	P@2	P@3	P@4	P@5
FastXML	0.492	0.390	0.322	0.272	0.235

Experimental results

Table: Results on WikiLSHTC.

τ	macro-F	HL
PLT + FTA $\tau = 0.05$		
PLT + FTA $\tau = 0.10$		
PLT + FTA $\tau = 0.15$		
PLT + FTA $\tau = 0.20$		
PLT + FTA $\tau = 0.25$		
PLT + FTA $\tau = 0.30$		
PLT + FTA $\tau = 0.35$		
PLT + FTA $\tau = 0.40$		
PLT + FTA $\tau = 0.45$		
PLT + FTA $\tau = 0.50$		
PLT + STO	**0.038**	4.115E-05
PLT + OFO (1 epoch, $a_0 =?, b_0 =?$)		

	P@1	P@2	P@3	P@4	P@5
PLT	0.387	0.295	0.220	0.165	0.132

Outline

1 Extreme multi-label classification

2 The F-measure

3 Efficient sparse probability estimators

4 Experimental results

5 Summary

Conclusions

- Presented approach can be **extended** to other **complex performance measures**.[14]

[14] N. Nagarajan, S. Koyejo, R. Ravikumar, and I. Dhillon. Consistent binary classification with generalized performance metrics. In *NIPS 27*, pages 2744–2752, 2014

H. Narasimhan, R. Vaish, and Agarwal S. On the statistical consistency of plug-in classifiers for non-decomposable performance measures. In *NIPS*, 2014

Wojciech Kotłowski and Krzysztof Dembczynski. Surrogate regret bounds for generalized classification performance metrics. In *ACML*, 2015

Conclusions

- Presented approach can be **extended** to other **complex performance measures**.[14]
- **Improving PLT** still in progress.

[14] N. Nagarajan, S. Koyejo, R. Ravikumar, and I. Dhillon. Consistent binary classification with generalized performance metrics. In *NIPS 27*, pages 2744–2752, 2014

H. Narasimhan, R. Vaish, and Agarwal S. On the statistical consistency of plug-in classifiers for non-decomposable performance measures. In *NIPS*, 2014

Wojciech Kotłowski and Krzysztof Dembczynski. Surrogate regret bounds for generalized classification performance metrics. In *ACML*, 2015

Conclusions

- Presented approach can be **extended** to other **complex performance measures**.[14]
- **Improving PLT** still in progress.
- **Ongoing work** on **online threshold** tuning.

[14] N. Nagarajan, S. Koyejo, R. Ravikumar, and I. Dhillon. Consistent binary classification with generalized performance metrics. In *NIPS 27*, pages 2744–2752, 2014

H. Narasimhan, R. Vaish, and Agarwal S. On the statistical consistency of plug-in classifiers for non-decomposable performance measures. In *NIPS*, 2014

Wojciech Kotłowski and Krzysztof Dembczynski. Surrogate regret bounds for generalized classification performance metrics. In *ACML*, 2015

Conclusions

- Presented approach can be **extended** to other **complex performance measures**.[14]
- **Improving PLT** still in progress.
- **Ongoing work** on **online threshold** tuning.
- Different **one dimensional optimization techniques**.

[14] N. Nagarajan, S. Koyejo, R. Ravikumar, and I. Dhillon. Consistent binary classification with generalized performance metrics. In *NIPS 27*, pages 2744–2752, 2014

H. Narasimhan, R. Vaish, and Agarwal S. On the statistical consistency of plug-in classifiers for non-decomposable performance measures. In *NIPS*, 2014

Wojciech Kotłowski and Krzysztof Dembczynski. Surrogate regret bounds for generalized classification performance metrics. In *ACML*, 2015

Conclusions

- Presented approach can be **extended** to other **complex performance measures**.[14]
- **Improving PLT** still in progress.
- **Ongoing work** on **online threshold** tuning.
- Different **one dimensional optimization techniques**.
- **Other** sparse probability estimators?

[14] N. Nagarajan, S. Koyejo, R. Ravikumar, and I. Dhillon. Consistent binary classification with generalized performance metrics. In *NIPS 27*, pages 2744–2752, 2014

H. Narasimhan, R. Vaish, and Agarwal S. On the statistical consistency of plug-in classifiers for non-decomposable performance measures. In *NIPS*, 2014

Wojciech Kotłowski and Krzysztof Dembczynski. Surrogate regret bounds for generalized classification performance metrics. In *ACML*, 2015

Conclusions

- **Take-away message**:

Conclusions

- **Take-away message**:
  - Extreme multi-label classification: #examples, #features, #labels

Conclusions

- **Take-away message**:
  - Extreme multi-label classification: #examples, #features, #labels
  - Complexity: training vs. validation vs. prediction, time vs. space

Conclusions

- **Take-away message**:
  - Extreme multi-label classification: #examples, #features, #labels
  - Complexity: training vs. validation vs. prediction, time vs. space
  - F-measure maximization by tuning threshold over probabilistic model.

Conclusions

- **Take-away message**:
  - Extreme multi-label classification: #examples, #features, #labels
  - Complexity: training vs. validation vs. prediction, time vs. space
  - F-measure maximization by tuning threshold over probabilistic model.
  - Naive generalization of tuning methods from binary to MLC scenario can be too expensive.

Conclusions

- **Take-away message**:
  - Extreme multi-label classification: #examples, #features, #labels
  - Complexity: training vs. validation vs. prediction, time vs. space
  - F-measure maximization by tuning threshold over probabilistic model.
  - Naive generalization of tuning methods from binary to MLC scenario can be too expensive.
  - Use sparse probability estimators.

Conclusions

- **Take-away message**:
  - Extreme multi-label classification: #examples, #features, #labels
  - Complexity: training vs. validation vs. prediction, time vs. space
  - F-measure maximization by tuning threshold over probabilistic model.
  - Naive generalization of tuning methods from binary to MLC scenario can be too expensive.
  - Use sparse probability estimators.
  - FastXML – decision tree-based approach.

Conclusions

- **Take-away message**:
  - Extreme multi-label classification: #examples, #features, #labels
  - Complexity: training vs. validation vs. prediction, time vs. space
  - F-measure maximization by tuning threshold over probabilistic model.
  - Naive generalization of tuning methods from binary to MLC scenario can be too expensive.
  - Use sparse probability estimators.
  - FastXML – decision tree-based approach.
  - PLT – label tree-based approach.

Conclusions

- **Take-away message**:
  - Extreme multi-label classification: #examples, #features, #labels
  - Complexity: training vs. validation vs. prediction, time vs. space
  - F-measure maximization by tuning threshold over probabilistic model.
  - Naive generalization of tuning methods from binary to MLC scenario can be too expensive.
  - Use sparse probability estimators.
  - FastXML – decision tree-based approach.
  - PLT – label tree-based approach.
  - Promising results, but many hopes for getting more ...

Conclusions

- **Take-away message**:
  - Extreme multi-label classification: #examples, #features, #labels
  - Complexity: training vs. validation vs. prediction, time vs. space
  - F-measure maximization by tuning threshold over probabilistic model.
  - Naive generalization of tuning methods from binary to MLC scenario can be too expensive.
  - Use sparse probability estimators.
  - FastXML – decision tree-based approach.
  - PLT – label tree-based approach.
  - Promising results, but many hopes for getting more ...
- For more check:

  `http://www.cs.put.poznan.pl/kdembczynski`