Aalto University
School of Science
and Technology

Structured output prediction for multilabel classification

Hongyu Su

Helsinki Institute for Information Technology HIIT
Department of Computer Science, Aalto University

August 7, 2015

Multilabel classification

- *Multilabel classification* is an important research field in machine learning.
- Input variable $\mathbf{x} \in \boldsymbol{\mathcal{X}}$ is in d dimensional input space $\boldsymbol{\mathcal{X}} = \mathbb{R}^d$.
- Output variable $\mathbf{y} = (y_1, \cdots, y_l) \in \boldsymbol{\mathcal{Y}}$ is a binary vector consist of l binary variables $y_j \in \{+1, -1\}$.
- $\mathbf{y}$ is called a multilabel, y_j is called a microlabel.
- Output space is composed by a Cartesian product of l sets

$$\boldsymbol{\mathcal{Y}} = \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_l,\, \mathcal{Y}_i = \{+1, -1\}.$$

- For example, in document classification, a document $\mathbf{x}$ can be classified as "news", "movie", and "science"

$$\mathbf{y} = (\underbrace{+1}_{\text{news}}, \underbrace{+1}_{\text{movie}}, \underbrace{-1}_{\text{sports}}, \underbrace{-1}_{\text{politics}}, \underbrace{-1}_{\text{finance}}, \underbrace{+1}_{\text{science}}, \underbrace{-1}_{\text{art}}).$$

- The goal is to find a mapping function $f \in \mathcal{H}$ that predicts the best values of an output given an input $f : \boldsymbol{\mathcal{X}} \rightarrow \boldsymbol{\mathcal{Y}}$.

Central problems in multilabel classification

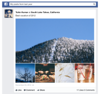
- The size of the output space (searching space) is exponential in the number of microlabels.

$$\boldsymbol{\mathcal{Y}} = \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_l,\ \mathcal{Y}_i = \{+1, -1\} \quad |\boldsymbol{\mathcal{Y}}| = 2^l.$$

- The dependency of microlabels needs to be exploited to improve the prediction performance.
  - If a document is about "movie", then it is more likely to be about "art" than "science".

Real world applications

- Social network, information can spread through multiple users.



$$\mathbf{y} = (\underbrace{+1}_{\text{Ted}}, \underbrace{-1}_{\text{Alice}}, \underbrace{+1}_{\text{David}}, \underbrace{-1}_{\text{Mark}}, \underbrace{+1}_{\text{Alex}}, \underbrace{-1}_{\text{Zoe}}, \underbrace{-1}_{\text{Frank}})$$

- Image annotation, an image can associate with multiple tags.

$$\mathbf{y} = (\underbrace{+1}_{\text{boat}}, \underbrace{+1}_{\text{sea}}, \underbrace{-1}_{\text{sun}}, \underbrace{-1}_{\text{beach}}, \underbrace{-1}_{\text{people}}, \underbrace{+1}_{\text{ice}}, \underbrace{+1}_{\text{land}})$$

- Document classification, an article can be assigned to multiple categories.

The New York Times.

TITANIC SINKS FOUR HOURS AFTER HITTING ICEBERG; 866 RESCUED BY CARPATHIA, PROBABLY 1250 PERISH; ISMAY SAFE, MRS. ASTOR MAYBE, NOTED NAMES MISSING

$$\mathbf{y} = (\underbrace{+1}_{\text{news}}, \underbrace{+1}_{\text{economics}}, \underbrace{-1}_{\text{sports}}, \underbrace{-1}_{\text{politics}}, \underbrace{-1}_{\text{movie}}, \underbrace{-1}_{\text{science}}, \underbrace{-1}_{\text{art}})$$

- Drug discovery, a drug can be effective for multiple symptoms.

$$\mathbf{y} = (\underbrace{+1}_{\text{heart}}, \underbrace{+1}_{\text{stroke}}, \underbrace{+1}_{\text{blood}}, \underbrace{+1}_{\text{fever}}, \underbrace{-1}_{\text{digest}}, \underbrace{-1}_{\text{liver}}, \underbrace{+1}_{\text{swelling}})$$

Flat multilabel classification approaches

- The categorization is proposed in [Tsoumakas et al., 2010]
- Problem transformation
  - Model the multilabel classification as a collection of single-label classification problems and solve each problem independently.
  - For example, ML-KNN [Zhang and Zhou, 2007], CC [Read et al., 2009, Read et al., 2011], IBLR [Cheng and Hüllermeier, 2009].
- Algorithm adaptation
  - Modify the single-label classification algorithm for multilabel classification problems.
  - For example, ADABOOST.MH [Schapire and Singer, 1999, Esuli et al., 2008], CORRLOG [Bian et al., 2012], MTL [Argyriou et al., 2008].
- These approaches does not model the dependency structure explicitly.

Structured output prediction

- Model the dependency structure with an output graph defined on microlabels.
- The categorization is proposed in [Su, 2015].
- Hierarchical classification
  - The output graph is a rooted tree or a DAG defining different levels of granularities.
  - For example, SSVM [Tsochantaridis et al., 2004, Tsochantaridis et al., 2005].
- Graph labeling
  - The output graph takes a more general form (e.g., a tree, a chain).
  - For example, CRF [Lafferty et al., 2001, Taskar et al., 2002], $\mathrm{M}^3\mathrm{N}$ [Taskar et al., 2004], MMCRF [Rousu et al., 2007, Su et al., 2010], SPIN [Su et al., 2014].
- These approaches assume the output graph is known *apriori*.

Contributions

- Structured output prediction models when the output graph is known.
  - SPIN for network influence prediction [Su et al., 2014].
  - MMCRF to work with general output graph structures [Su et al., 2010].
- Structured output prediction models working with unknown output graph.
  - MVE to combine multiple structured output predictors with ensemble [Su and Rousu, 2011].
  - AMM and MAM to aggregate the inference results from multiple structured output predictors [Su and Rousu, 2013, Su and Rousu, 2015].
  - RTA to perform joint learning and inference over a collection of random spanning trees [Marchand et al., 2014].
- Codes for developed models are available from http://hongyusu.github.io.

Outline

- Preliminaries
- Structured output learning with known output graph
- Structured output learning with unknown output graph
- Future work
- Experimental results

Preliminaries

- Training examples come in pairs $(\mathbf{x}, \mathbf{y}) \in \boldsymbol{\mathcal{X}} \times \boldsymbol{\mathcal{Y}}$.
- $\mathbf{x} \in \boldsymbol{\mathcal{X}}$ is an arbitrary input space.
- $\boldsymbol{\mathcal{Y}}$ is an output space of a collection of ℓ-dimensional *multilabels*.

$$\mathbf{y} = (y_1, \cdots, y_\ell) \in \boldsymbol{\mathcal{Y}}.$$

- y_i is a *microlabel* and $y_i \in \{1, \cdots, r_i\}, r_i \in \mathbb{Z}$.
- For example, multilabel binary classification $y_i \in \{-1, +1\}$.
- We are given a set of m training examples $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^m$.
- Each example $(\mathbf{x}, \mathbf{y})$ is mapped into a joint feature space $\boldsymbol{\phi}(\mathbf{x}, \mathbf{y})$.
- $\mathbf{w}$ is the weight vector in the joint feature space.
- Define a linear score function $F(\mathbf{w}, \mathbf{x}, \mathbf{y}) = \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}, \mathbf{y}) \rangle$.
- $\mathbf{w}$ makes sure example $\mathbf{x}$ with correct multilabel $\mathbf{y}$ achieves higher score than with any other incorrect multilabel $\mathbf{y}' \in \boldsymbol{\mathcal{Y}}$.

Inference problem

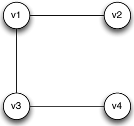
- The prediction $\mathbf{y_w}(\mathbf{x})$ of an input $\mathbf{x}$ is the multilabel $\mathbf{y}$ that maximizes the score function

$$\mathbf{y_w}(\mathbf{x}) = \underset{\mathbf{y} \in \mathcal{Y}}{\mathbf{argmax}} \langle \mathbf{w}, \phi(\mathbf{x}, \mathbf{y}) \rangle. \tag{1}$$

- Search space $|\mathcal{Y}| = 2^{\ell}$ is exponential in size.
- (1) is called *inference* problem which is $\mathcal{NP}$-hard for most output feature maps.
- We aim at using an output feature map in which the inference can be solved with a polynomial algorithm, e.g., dynamic programming.

Input-output feature maps

- We assume that the joint feature map ϕ is a potential function on a Markov network $G = (E, V)$.
- A vertex $v_i \in V$ corresponds to a microlabel y_i, an edge $(v_i, v_j) \in E$ corresponds to the pairwise correlation of the microlabel y_i and y_j.
- G models potential pairwise correlations.

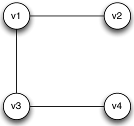


- $\varphi(\mathbf{x}) \in \mathbb{R}^d$ is the input feature map, e.g., bag-of-words of a document.
- $\psi(\mathbf{y}) \in \mathbb{R}^{4|E|}$ is the output feature map which maps the multilabel $\mathbf{y}$ into a collection of edges and labels

$$\varphi(\mathbf{y}) = (u_e)_{e \in E}, u_e \in \{-1, +1\}^2.$$

An example of output feature map

- Markov network $G = (E, V)$



- Multilabel $\mathbf{y}$

$$\mathbf{y} = (y_1, y_2, y_3, y_4) = (+1, -1, +1, +1)$$

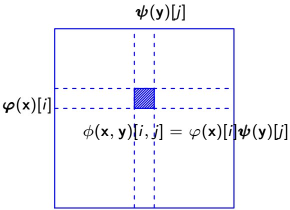
- Output feature map $\boldsymbol{\psi}(\mathbf{y})$

$$\boldsymbol{\psi}(\mathbf{y}) = (\underbrace{\underbrace{0}_{--}, \underbrace{0}_{-+}, \underbrace{1}_{+-}, \underbrace{0}_{++}}_{(v_1, v_3)}, \underbrace{\underbrace{0}_{--}, \underbrace{0}_{-+}, \underbrace{0}_{+-}, \underbrace{1}_{++}}_{(v_1, v_2)}, \underbrace{\underbrace{0}_{--}, \underbrace{0}_{-+}, \underbrace{0}_{+-}, \underbrace{1}_{++}}_{(v_3, v_4)})$$

Joint feature map

- The joint feature is the Kronecker product of $\varphi(\mathbf{x})$ and $\psi(\mathbf{y})$

$$\phi(\mathbf{x}, \mathbf{y}) = (\phi_e(\mathbf{x}, \mathbf{y}))_{e \in E} = (\varphi(\mathbf{x}) \otimes \psi_e(\mathbf{y}_e))_{e \in E}.$$

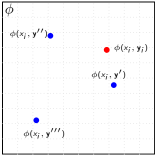


- The score function can be factorized by the output graph G

$$F(\mathbf{w}, \mathbf{x}, \mathbf{y}) = \langle \mathbf{w}, \phi(\mathbf{x}, \mathbf{y}) \rangle = \sum_{e \in E} \langle \mathbf{w}_e, \phi_e(\mathbf{x}, \mathbf{y}_e) \rangle.$$

Optimization problem

- To learn parameter $\mathbf{w}$, we aim to maximize the magin between correct pair $(\mathbf{x}_i, \mathbf{y}_i)$ and all the other incorrect pairs $(\mathbf{x}_i, \mathbf{y}), \mathbf{y} \in \boldsymbol{\mathcal{Y}}/\mathbf{y}_i$ in the joint feature space $\boldsymbol{\phi}$.



- The model is max-margin conditional random field MMCRF [Rousu et al., 2007, Su et al., 2010].
- The primal optimization problem is defined as

$$
\begin{aligned}
\min_{\mathbf{w}, \xi_k} \quad & \frac{1}{2} ||\mathbf{w}||^2 + C \sum_{k=1}^{m} \xi_k \qquad (2) \\
\textbf{s.t.} \quad & \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}_k, \mathbf{y}_k) \rangle - \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}_k, \mathbf{y}) \rangle \geq \ell(\mathbf{y}_k, \mathbf{y}) - \xi_k, \\
& \xi_k \geq 0\,, \forall\ \mathbf{y} \in \boldsymbol{\mathcal{Y}}, k \in \{1, \ldots, m\}.
\end{aligned}
$$

- $\ell(\mathbf{y}, \mathbf{y}_i)$ scales the margin according to the multilabel $\mathbf{y}$.

Marginal-dual optimization

- (2) is difficult as the number of the constraints is $m \times |\mathcal{Y}|$.
- The dual optimization problem is defined as

$$\begin{aligned} \max_{\alpha \geq 0} \quad & \alpha^\intercal \ell - \frac{1}{2}\alpha^\intercal K \alpha \\ \text{s.t.} \quad & \sum_{\mathbf{y}\in\mathcal{Y}} \alpha(i,\mathbf{y}) \leq C, \forall i \in \{1,\cdots,m\}. \end{aligned} \tag{3}$$

- (3) is also challenging due to the exponential number of dual variables.
- We use edge marginals to replace the dual variables [Taskar et al., 2004]

$$\mu(i,e,u_e) = \sum_{\mathbf{y}} \mathbf{1}_{\{\boldsymbol{\psi}_e(\mathbf{y})=u_e\}}\alpha(i,\mathbf{y}).$$

- The margin-dual optimization problem is

$$\max_{\mu\in\mathcal{M}} \quad \mu^\intercal \ell - \frac{1}{2}\mu^\intercal K \mu. \tag{4}$$

- The number of marginal-dual variable is $m \times 4|E|$.

Conditional gradient optimization

- (4) is optimized by conditional gradient decent which optimizes μ_k that corresponds to a single example while keeps others $(\mu_j, j \neq k)$ fixed

$$\max_{\mu_k \in \mathcal{M}} \quad \mu_k^\intercal \ell_k - \frac{1}{2} \sum_j \mu_k^\intercal K \mu_j, \, \forall k.$$

- Current gradient of μ_k is given by $g_i = \ell_i - \sum_j K \mu_j$.
- Compute a feasible solution μ_k^* as an update direction

$$\mu_k^* = \underset{\mu_k \in \mathcal{M}}{\mathbf{argmax}} \, \mu_k^\intercal g_k = \underset{\mu_k \in \mathcal{M}}{\mathbf{argmax}} \sum_e \mu(k,e)^\intercal g(k,e). \tag{5}$$

- (5) is an instantiation of MAP problem
  - G is tree, exact inference with polynomial time algorithm, e.g, dynamic programming in [Rousu et al., 2007]
  - G is general graph, approximate inference, e.g. loopy belief propagation in [Su et al., 2010]
- Perform the update via exact line search $\mu_k \leftarrow \mu_k + \tau(\mu_k^* - \mu_k)$.

Exact line search

- Line search gives the optimal feasible solution as a stationary point (τ)

$$\begin{aligned} \max_{\tau} \quad & g(\mu_k + \tau\Delta\mu_k) \\ \textbf{s.t.} \quad & 0 \le \tau \le 1. \end{aligned} \tag{6}$$

- $\tau = 0$ corresponds to no update.
- Feasible maximum update is achieved at $\tau = 1$.
- The cost of computing (6) is significantly smaller than the cost of computing (5).

Compute duality gap

- We use duality gap to measure the progress of the optimization.
- Primal and marginal-dual objective functions

$$f(\mathbf{w}) = \frac{1}{2}||\mathbf{w}||^2 + C\sum_{k=1}^{m}(\ell_k - \langle \mathbf{w}, \Delta\boldsymbol{\phi}(\mathbf{x}_k, \mathbf{y}_k)\rangle)$$

$$g(\mu) = \sum_{k=1}^{m}\mu_k\ell_k - \frac{1}{2}\sum_{k=1}^{m}\sum_{j=1}^{m}\mu_k K^{\Delta\boldsymbol{\phi}}(\mathbf{x}_k, \mathbf{y}_k; \mathbf{x}_j, \mathbf{y}_j)\mu_j$$

- $\max_{\mu} g(\mu) \leq \min_{\mathbf{w}} f(\mathbf{w})$, gap is minimized at optimal.
- Duality gap at μ^t

$$\begin{aligned} f(\mathbf{w}^t) - g(\mu^t) &= C\left(\ell - K^{\Delta\boldsymbol{\phi}}\mu^t\right) - \mu^t\left(\ell - K^{\Delta\boldsymbol{\phi}}\mu^t\right) \\ &= C^{\intercal}\nabla g(\mu^t) - \mu^{t\intercal}\nabla g(\mu^t) \end{aligned}$$

1. Estimate the marginal-dual objective by linear approximation $\nabla g(\mu^t)$.
2. Marginal-dual objective value at μ^t is computed by $\mu^{t\intercal}\nabla g(\mu^t)$.
3. Primal objective value is estimate by $C^{\intercal}\nabla g(\mu^t)$.

So far

- We work with multilabel classification problems in general.
- In addition, we assume there is an output graph given *apriori*.
- We develop structured output prediction method which model the label correlations with an output graph.
- If the output graph is tree-like, the inference problem can be solved exactly with a polynomial time algorithm, e.g., dynamic programming.
- If the output graph is a general graph structure, the inference problem is $\mathcal{NP}$-hardand can be solved with approximation algorithm, e.g., loopy belief propagation.
- The next question is what if the output graph is not observed.

Research question

- The output graph is hidden in many applications.
  - For example, a surveillance photo can be tagged with "building", "road", "pedestrian", and "vehicle".
- We study the problem in structured output learning when the output graph is not observed.
- In particular:
  - Assume the dependency can be expressed by a complete set of pairwise correlations.
  - Build a structured output learning model with a complete graph as the output graph.
  - Solve the $\mathcal{NP}$-hardinference problem on the complete graph by a polynomial time algorithm.
- A structured prediction model which performs max-margin learning on a random collection of spanning trees sampled from the output graph.

Complete graph as output graph

- We assume that the joint feature map $\boldsymbol{\phi}$ is a potential function on a Markov network $G = (E, V)$.
- G is a complete graph with $|V| = \ell$ nodes and $|E| = \frac{\ell(\ell-1)}{2}$ undirected edges.
- G models all pairwise correlations.
- $\boldsymbol{\varphi}(\mathbf{x})$ is the input feature map, e.g., bag-of-words feature of an example $\mathbf{x}$.
- $\boldsymbol{\psi}(\mathbf{y})$ is the output feature map which is a collection of edges and labels

$$\boldsymbol{\varphi}(\mathbf{y}) = (u_e)_{e\in E}, u_e \in \{-1, +1\}^2.$$

- The joint feature is the Kronecker product of $\boldsymbol{\varphi}(\mathbf{x})$ and $\boldsymbol{\psi}(\mathbf{y})$

$$\boldsymbol{\phi}(\mathbf{x}, \mathbf{y}) = (\boldsymbol{\phi}_e(\mathbf{x}, \mathbf{y}))_{e\in E} = (\boldsymbol{\varphi}(\mathbf{x}) \otimes \boldsymbol{\psi}_e(\mathbf{y}_e))_{e\in E}.$$

- The score function can be factorized by the complete graph G

$$F(\mathbf{w}, \mathbf{x}, \mathbf{y}) = \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}, \mathbf{y})\rangle = \sum_{e\in E} \langle \mathbf{w}_e, \boldsymbol{\phi}_e(\mathbf{x}, \mathbf{y}_e)\rangle.$$

Inference in terms of all spanning trees

- Solving the following inference problem on a complete graph is $\mathcal{NP}$-hard

$$\mathbf{y_w}(\mathbf{x}) = \underset{\mathbf{y}\in\mathcal{Y}}{\mathbf{argmax}}\, F(\mathbf{w},\mathbf{x},\mathbf{y}) = \underset{\mathbf{y}\in\mathcal{Y}}{\mathbf{argmax}} \sum_{e\in E} \langle \mathbf{w}_e, \boldsymbol{\phi}_e(\mathbf{x},\mathbf{y}_e)\rangle.$$

- For a complete graph, there are $\ell^{\ell-2}$ unique spanning trees.
- We can write $F(\mathbf{w},\mathbf{x},\mathbf{y})$ as a conic combination of all spanning trees

$$F(\mathbf{w},\mathbf{x},\mathbf{y}) = \underset{T\in U(G)}{\mathbf{E}} a_T \langle \mathbf{w}_T, \boldsymbol{\phi}_T(\mathbf{x},\mathbf{y})\rangle$$

$$\underset{T\in U(G)}{\mathbf{E}} a_T^2 = 1, \underset{T\in U(G)}{\mathbf{E}} a_T < 1.$$

- $U(G)$ is the uniform distribution over $\ell^{\ell-2}$ spanning trees.
- The number of spanning trees is exponentially dependent on the number of nodes ℓ.

A sample of n spanning trees

- Instead of using all spanning trees, we can just use n spanning trees

$$F_{\mathcal{T}}(\mathbf{w}, \mathbf{x}, \mathbf{y}) = \frac{1}{n} \sum_{i=1}^{n} a_{T_i} \langle \mathbf{w}_{T_i}, \boldsymbol{\phi}_{T_i}(\mathbf{x}, \mathbf{y}) \rangle$$

$$\frac{1}{n} \sum_{i=1}^{n} a_{T_i}^2 = 1, \frac{1}{n} \sum_{i=1}^{n} a_{T_i} < 1.$$

- When

$$n \geq \frac{\ell^2}{\epsilon^2} \left(\frac{1}{16} + \frac{1}{2} \ln \frac{8\sqrt{n}}{\delta} \right),$$

  we have $|F_{\mathcal{T}}(\mathbf{w}, \mathbf{x}, \mathbf{y}) - F(\mathbf{w}, \mathbf{x}, \mathbf{y})| \leq \epsilon$, with high probability.
- A sample of $n \in \Theta(\ell^2/\delta^2)$ random spanning tree is sufficient to estimate the score function.
- Margin achieved by $F(\mathbf{w}, \mathbf{x}, \mathbf{y})$ is also preserved by the sample of n random spanning trees $F_{\mathcal{T}}(\mathbf{w}, \mathbf{x}, \mathbf{y})$ [Marchand et al., 2014].

Random spanning tree approximation RTA

- The optimization problem of RTA is defined as [Marchand et al., 2014]

$$
\begin{aligned}
\min_{\mathbf{w}_{T_i},\xi_i} \quad & \frac{1}{2}\sum_{i=1}^{n}||\mathbf{w}_{T_i}||^2 + C\sum_{k=1}^{m}\xi_k \\
\textbf{s.t.} \quad & \frac{1}{\sqrt{n}}\sum_{i=1}^{n}\langle \mathbf{w}_{T_i}, \boldsymbol{\phi}_{T_t}(\mathbf{x}_k,\mathbf{y}_k)\rangle - \max_{\mathbf{y}\neq\mathbf{y}_k}\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\langle \mathbf{w}_{T_t}, \boldsymbol{\phi}_{T_i}(\mathbf{x}_k,\mathbf{y})\rangle \geq 1-\xi_k, \\
& \xi_k \geq 0\,, \forall\ k \in \{1,\dots,m\}.
\end{aligned}
$$

Inference Problem for a collection of trees

- The inference problem of RTA is defined as finding the multilabel $\mathbf{y}_{\mathcal{T}}(\mathbf{x})$ that maximizes the sum of scores over a collection of trees

$$\mathbf{y}_{\mathcal{T}}(\mathbf{x}) = \underset{\mathbf{y}\in\mathcal{Y}}{\mathbf{argmax}}\, F_{\mathcal{T}}(\mathbf{x},\mathbf{y};\mathbf{w}_{\mathcal{T}}) = \underset{\mathbf{y}\in\mathcal{Y}}{\mathbf{argmax}} \sum_{t=1}^{n} \langle \mathbf{w}_{T_t}, \phi_{T_t}(\mathbf{x},\mathbf{y})\rangle.$$

- The inference problem on each individual spanning tree can be solve efficiently in $\Theta(\ell)$ by *dynamic programming*

$$\mathbf{y}_{T_t}(\mathbf{x}) = \underset{\mathbf{y}\in\mathcal{Y}}{\mathbf{argmax}}\, F_{T_t}(\mathbf{x},\mathbf{y};\mathbf{w}_{T_t}) = \underset{\mathbf{y}\in\mathcal{Y}}{\mathbf{argmax}}\, \langle \mathbf{w}_{T_t}, \phi_{T_t}(\mathbf{x},\mathbf{y})\rangle.$$

- There is no guarantee that there exists a tree $T_t \in \mathcal{T}$ in which the maximizer of F_{T_t} is the maximizer of $F_{\mathcal{T}}$.

Fast inference over a collection of trees

- For each tree T_t, instead of computing the best multilabel $\mathbf{y}_{T_t}$, we compute K-best multilabels in $\Theta(K\ell)$ time

$$\mathcal{Y}_{T_t,K} = \{\mathbf{y}_{T_t,1}, \cdots, \mathbf{y}_{T_t,K}\}.$$

- Performing the same computation on all trees gives a candidate list of $n \times K$ multilabels in $\Theta(nK\ell)$ time

$$\mathcal{Y}_{\mathcal{T},K} = \mathcal{Y}_{T_1,K} \cup \cdots \mathcal{Y}_{T_n,K}.$$

- For now, we assume the best scoring multilabel of a collection of trees exists in the list $\mathcal{Y}_{\mathcal{T},K}$.
- We proved that with a high probability $\mathbf{y}_{\mathcal{T}}$ will appear in $\mathcal{Y}_{\mathcal{T},K}$.
- We can identify $\mathbf{y}_{\mathcal{T}}$ from $\mathcal{Y}_{\mathcal{T},K}$.

▶

Bibliography

Argyriou, A., Evgeniou, T., and Pontil, M. (2008).
Convex multi-task feature learning.
Machine Learning, 73(3):243–272.

Bian, W., Xie, B., and Tao, D. (2012).
Corrlog: Correlated logistic models for joint prediction of multiple labels.
Journal of Machine Learning Research - Proceedings Track, pages 109–117.

Cheng, W. and Hüllermeier, E. (2009).
Combining instance-based learning and logistic regression for multilabel classification.
Machine Learning, 76(2-3):211–225.

Esuli, A., Fagni, T., and Sebastiani, F. (2008).
Boosting multi-label hierarchical text categorization.
Information Retrieval, 11(4):287–313.

Bibliography (cont.)

Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001).
Conditional random fields: Probabilistic models for segmenting and labeling sequence data.
In *Proceedings of the 8th International Conference on Machine Learning (ICML 2001)*, pages 282–289. Morgan Kaufmann Publishers Inc.

Marchand, M., Su, H., Morvant, E., Rousu, J., and Shawe-Taylor, J. (2014).
Multilabel structured output learning with random spanning trees of max-margin markov networks.
In *Advances in Neural Information Processing System NIPS2014*, page to appear.

Read, J., Pfahringer, B., Holmes, G., and Frank, E. (2009).
Classifier chains for multi-label classification.
In Buntine, W., Grobelnik, M., Mladenić, D., and Shawe-Taylor, J., editors, *Machine Learning and Knowledge Discovery in Databases*, volume 5782, pages 254–269. Springer Berlin Heidelberg.

Bibliography (cont.)

Read, J., Pfahringer, B., Holmes, G., and Frank, E. (2011).
Classifier chains for multi-label classification.
Machine Learning, 85(3):333–359.

Rousu, J., Saunders, C., Szedmak, S., and Shawe-Taylor, J. (2007).
Efficient algorithms for max-margin structured classification.
Predicting Structured Data, pages 105–129.

Schapire, R. and Singer, Y. (1999).
Improved boosting algorithms using confidence-rated predictions.
Machine Learning, 37(3):297–336.

Su, H. (2015).
Multilabel Classification through Structured Output Learning - Methods and Applications.
PhD thesis, Department of Information and Computer Science, Aalto University.

Bibliography (cont.)

Su, H., Gionis, A., and Rousu, J. (2014).
Structured prediction of network response.
In *Proceedings, 31th International Conference on Machine Learning ICML2014*, volume 32 of *Journal of Machine Learning Research WCP*, pages 442–450.

Su, H., Heinonen, M., and Rousu, J. (2010).
Structured output prediction of anti-cancer drug activity.
In *Proceedings, 5th IAPR International Conference on Pattern Recognition in Bioinformatics (PRIB2010)*, volume 6282 of *Lecture Note in Computer Science*, pages 38–49.

Su, H. and Rousu, J. (2011).
Multi-task drug bioactivity classification with graph labeling ensembles.
In *Proceedings, 6th IAPR International Conference on Pattern Recognition in Bioinformatics (PRIB2011)*, volume 7035 of *Lecture Note in Computer Science*, pages 157–167.

Bibliography (cont.)

Su, H. and Rousu, J. (2013).
Multilabel classification through random graph ensembles.
In *Proceedings, 5th Asian Conference on Machine Learning (ACML2013)*, volume 29 of *Journal of Machine Learning Research WCP*, pages 404–418.

Su, H. and Rousu, J. (2015).
Multilabel classification through random graph ensembles.
Machine Learning, 99(2):231–256.

Taskar, B., Abbeel, P., and Koller, D. (2002).
Discriminative probabilistic models for relational data.
In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI 2002)*, pages 485–492, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Bibliography (cont.)

Taskar, B., Guestrin, C., and Koller, D. (2004).
Max-margin markov networks.
In Thrun, S., Saul, L., and Schölkopf, B., editors, *Advances in Neural Information Processing Systems 16*, pages 25–32. MIT Press.

Tsochantaridis, I., Hofmann, T., Joachims, T., and Altun, Y. (2004).
Support vector machine learning for interdependent and structured output spaces.
In *Proceedings of the 21th International Conference on Machine Learning (ICML 2004)*, pages 823–830. ACM.

Tsochantaridis, I., Joachims, T., Hofmann, T., and Altun, Y. (2005).
Large margin methods for structured and interdependent output variables.
Journal of Machine Learning Research, 6:1453–1484.

Bibliography (cont.)

Tsoumakas, G., Katakis, I., and Vlahavas, I. (2010).
Mining multi-label data.
In Maimon, O. and Rokach, L., editors, *Data Mining and Knowledge Discovery Handbook*, pages 667–685. Springer US.

Zhang, M. and Zhou, Z. (2007).
Ml-knn: A lazy learning approach to multi-label learning.
Pattern Recognition, 40:2007.