Aalto University
School of Science
and Technology

Structured Output Learning with A Random Sample of Spanning Trees

Hongyu Su

Helsinki Institute for Information Technilogy HIIT
Department of Information and Computer Science
Aalto University

November 19, 2014

Multilabel Classification

- Multilabel classification is an important research field in machine learning.
  - For example, a document can be classified as "science", "genomics", and "drug discovery".
  - Each input variable $\mathbf{x} \in \boldsymbol{\mathcal{X}}$ is simultaneously associated with multiple output variables $\mathbf{y} \in \boldsymbol{\mathcal{Y}}, \boldsymbol{\mathcal{Y}} = \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_l$.
  - The goal is to find a mapping function that predicts the best values of an output given an input $f \in \mathcal{H} : \boldsymbol{\mathcal{X}} \rightarrow \boldsymbol{\mathcal{Y}}$.
- The central problems of multilabel classification:
  - The size of the output space $\boldsymbol{\mathcal{Y}}$ is exponential in the number of microlabels.
  - The dependency of microlabels needs to be exploited to improve the prediction performance.

Flat Multilabel Classification

- Multiple output variables are treated as a "flat" vector.
- For example, ML-KNN, ADABOOST.MH, MTL, ...
- It is difficult to take into consideration the correlation of labels.

Structured Output Learning

- There is an *output graph* connecting multiple labels.
  - A set of nodes corresponds to the multiple labels.
  - A set of edges represents the correlation between labels.
- Hierarchical classification:
  - The output graph is a rooted tree or a directed graph defining the different levels of granularities.
  - For example, SSVM, ...
- Graph labeling:
  - The output graph often takes a general form (e.g., a tree, a chain).
  - For example, $\mathrm{M^3N}$, CRF, MMCRF, ...
- The output graph is assumed to be known *apriori*.

The 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 focus on the problem in structured output learning when the output graph is not observed.
- Our approach:
  - Assume the dependency can be modeled by a complete set of pairwise correlations.
  - Build a structured output learning model with a complete graph as the output graph.
  - Solve the optimization problem.

Contributions

- A structured output learning model which performs max-margin learning on a random sample of spanning tree.
- The model is not constrained to the availability of the output graph.
- The $\mathcal{NP}$-hard inference problem can be solved by a polynomial time algorithm with a condition guaranteeing the exact solution.
- The theoretical analysis and the empirical results verify the performance of the proposed model.

Model

- The training examples are given in pair $S = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{m}$.
- A complete graph $G = (E, V)$ is used as the output graph.
- $\Gamma_G(\mathbf{y}_i)$ is the output feature map on a complete graph G.
- Each example is mapped to a joint feature space by a joint feature map

$$\phi_G(\mathbf{x}_i, \mathbf{y}_i) = \varphi(\mathbf{x}_i) \otimes \Gamma_G(\mathbf{y}_i).$$

- A compatibility score is defined as

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

- $\mathbf{w}$ ensures an input $\mathbf{x}_i$ with a correct multilabel $\mathbf{y}_i$ achieves a higher score than with any incorrect multilabel $\mathbf{y} \in \mathcal{Y}$.

Model (cont.)

- The predicted output $\mathbf{y}(\mathbf{x})$ for a given input $\mathbf{x}$ is computed by

$$\mathbf{y}(\mathbf{x}) = \underset{\mathbf{y} \in \mathcal{Y}}{\mathbf{argmax}}\, F(\mathbf{x}, \mathbf{y}; \mathbf{w}),$$

  which is called *inference problem*.
- The inference problem is $\mathcal{NP}$-hard for most joint feature maps on the complete graph.

Learning with A Complete Graph

- The *margin* of an example $\mathbf{x}_i$ is

$$\gamma_G(\mathbf{x}_i; \mathbf{w}) = \min_{\mathbf{y} \in \mathcal{Y}} [F(\mathbf{x}_i, \mathbf{y}_i; \mathbf{w}) - F(\mathbf{x}_i, \mathbf{y}; \mathbf{w})].$$

- $\mathbf{w}$ is solved by *maximum-margin principle* which aims to maximize $\gamma(\mathbf{x}_i; \mathbf{w})$ over all training example.
- The problems:
  - $\mathcal{NP}$-hardnessof the inference problem on a complete graph.
  - A large parameter space: $\Theta(k^2)$
- We aim to use a joint feature map that allows the inference problem be solved in polynomial time.

Random Spanning Tree Approximation

- We proved if a large margin structured output predictor exists, then combining a small sample of random trees will, with a high probability, generate a predictor with good generalization.
- $\mathcal{T} = \{T_1, \cdots, T_n\}$ is a set of spanning trees randomly sampled from the complete graph G.
- The compatibility score can be re-defined based on $\mathcal{T}$ as

$$F(\mathbf{x}_i, \mathbf{y}_i; \mathbf{w}) = \sum_{t=1}^{n} \langle \mathbf{w}_{T_t}, \phi_{T_t}(\mathbf{x}_i, \mathbf{y}_i) \rangle$$

- The inference problem of predicting the output $\mathbf{y}_{\mathcal{T}}(\mathbf{x})$ for a given input is

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

Optimization Problem

- The margin of an example $\mathbf{x}_i$ achieved by $\mathcal{T}$ is

$$\gamma_{\mathcal{T}}(\mathbf{x}_i; \mathbf{w}) = \min_{\mathbf{y} \in \mathcal{Y}} [\sum_{t=1}^{n} \langle \mathbf{w}_{T_t}, \phi_{T_t}(\mathbf{x}_i, \mathbf{y}_i) \rangle - \sum_{t=1}^{n} \langle \mathbf{w}_{T_t}, \phi_{T_t}(\mathbf{x}_i, \mathbf{y}) \rangle].$$

- To learn $\mathbf{w}_{T_t}, \forall T_t \in \mathcal{T}$ we solve the optimization problem

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

Inference Problem

- 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

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

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

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

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

Fast Inference Over a Collection of Trees

- Instead of compute the best multilabel $\mathbf{y}_{T_t}$ from each individual spanning tree $T_t \in \mathcal{T}$, we consider the K-best multilabel

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

- This gives a candidate list $\mathcal{Y}_{\mathcal{T},K} = \mathcal{Y}_{T_1,K} \cup \cdots \mathcal{Y}_{T_n,K}$ of $n \times K$ multilabels.
- We proved that with a high probability $\mathbf{y}_{\mathcal{T}}$ will appear in $\mathcal{Y}_{\mathcal{T},K}$.

Performance of the Inference Algorithm

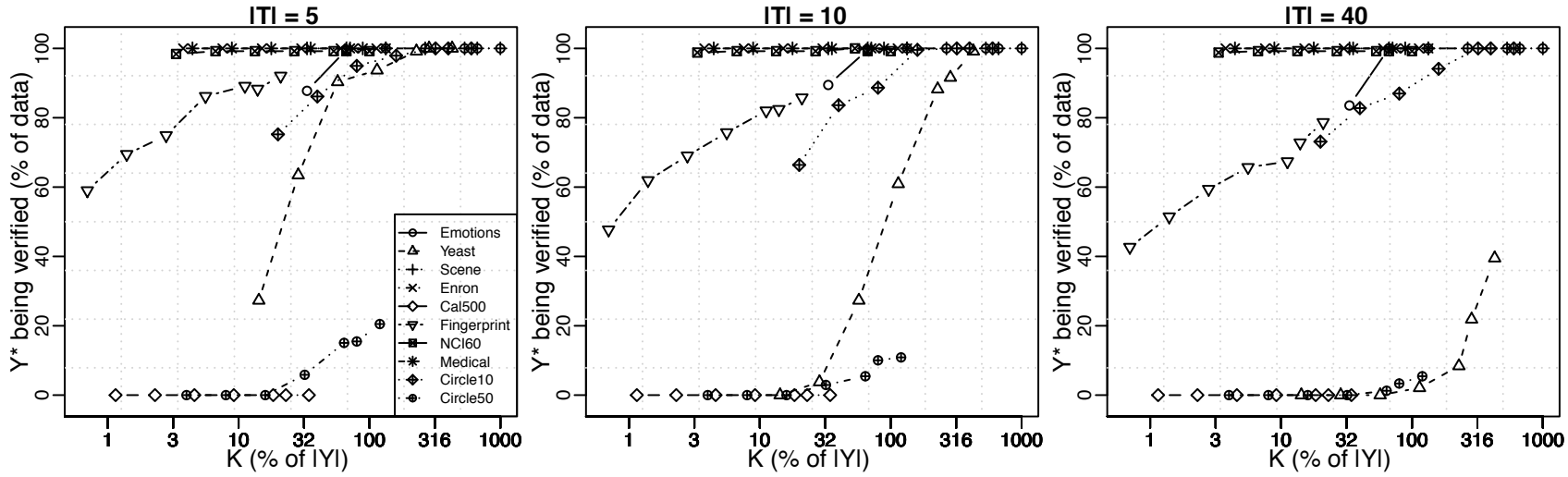

Figure : Percentage of examples with provably optimal **y** being in the K-best lists plotted as a function of K, scaled with respect to the number of microlabels in the dataset.

Prediction Performance

Dataset	Microlabel Loss (%)					0/1 Loss (%)				
	SVM	MTL	MMCRF	MAM	RTA	SVM	MTL	MMCRF	MAM	RTA
Emotions	22.4	20.2	20.1	*19.5*	**18.8**	77.8	74.5	71.3	*69.6*	**66.3**
Yeast	*20.0*	20.7	21.7	20.1	**19.8**	*85.9*	88.7	93.0	86.0	**77.7**
Scene	*9.8*	11.6	18.4	17.0	**8.8**	*47.2*	55.2	72.2	94.6	**30.2**
Enron	6.4	6.5	6.2	**5.0**	*5.3*	99.6	99.6	92.7	*87.9*	**87.7**
Cal500	**13.7**	*13.8*	**13.7**	**13.7**	*13.8*	100.0	100.0	100.0	100.0	100.0
Fingerprint	**10.3**	17.3	*10.5*	*10.5*	10.7	99.0	100.0	*99.6*	*99.6*	**96.7**
NCI60	15.3	16.0	*14.6*	**14.3**	14.9	56.9	*53.0*	63.1	60.0	**52.9**
Medical	*2.6*	*2.6*	**2.1**	**2.1**	**2.1**	91.8	91.8	63.8	*63.1*	**58.8**
Circle10	4.7	6.3	2.6	*2.5*	**0.6**	28.9	33.2	20.3	*17.7*	**4.0**
Circle50	5.7	6.2	**1.5**	*2.1*	3.8	69.8	72.3	**38.8**	*46.2*	*52.8*

Figure : Prediction performance of each algorithm in terms of microlabel loss and 0/1 loss. The best performing algorithm is highlighted with boldface, the second best is in italic

Conclusions

-