Aalto University
School of Science
and Technology

Max-Margin Learning with A Random Sample of Spanning Trees

Hongyu Su

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

February 12, 2015

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 associated with multiple output variables $\mathbf{y} \in \boldsymbol{\mathcal{Y}}, \boldsymbol{\mathcal{Y}} = \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_l, \mathcal{Y}_i = \{+1, -1\}$.
  - 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.

Structured output learning

- There is an *output graph* connecting multiple labels.
  - A set of nodes represents 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 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, M^3N, CRF, MMCRF, ...
- The output graph is assumed to be known *apriori*.

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 optimization problem and the inference problem ($\mathcal{NP}$-hard).

In this presentation

- A structured prediction model which performs max-margin learning on a random sample of spanning tree.
- Two ways to combine the set of random spanning trees.
  - conical combination
  - convex combination
- The corresponding optimization problem.

Model

- Training examples comes in pair $S = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^m \in \boldsymbol{\mathcal{X}} \times \boldsymbol{\mathcal{Y}}$.
- A complete graph $G = (E, V)$ is used as the output graph.
- $\Gamma_G(\mathbf{y}_i)$ is the output feature map of $\mathbf{y}_i$ on G

$$\Gamma_G(\mathbf{y}_i) = \{\Gamma_e(\mathbf{y}_{i,e})\}_{e\in G},$$
$$\Gamma_e(\mathbf{y}_{i,e}) = [\mathbf{1}_{\mathbf{y}_{i,e}==00}, \mathbf{1}_{\mathbf{y}_{i,e}==01}, \mathbf{1}_{\mathbf{y}_{i,e}==10}, \mathbf{1}_{\mathbf{y}_{i,e}==11}]$$

- A joint feature map of $(\mathbf{x}_i, \mathbf{y}_i)$

$$\phi_G(\mathbf{x}_i, \mathbf{y}_i) = \varphi(\mathbf{x}_i) \otimes \Gamma_G(\mathbf{y}_i) = \{\phi_e(x_i, \mathbf{y}_{i,e})\}_{e\in G}.$$

- A compatibility score is defined as

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

Model (cont.)

- $\mathbf{w}$ ensures an input $\mathbf{x}_i$ with a correct multilabel $\mathbf{y}_i$ achieves a higher score than with any incorrect multilabel $\mathbf{y} \in \mathcal{Y}$.
- The predicted output $\mathbf{y}(\mathbf{x})$ for a given input $\mathbf{x}$ is computed by

$$\mathbf{y}(\mathbf{x}) = \operatorname*{\mathbf{argmax}}_{\mathbf{y}\in\mathcal{Y}} F(\mathbf{x}, \mathbf{y}; \mathbf{w}_G) = \operatorname*{\mathbf{argmax}}_{\mathbf{y}\in\mathcal{Y}} \langle \mathbf{w}_G, \phi_G(\mathbf{x}, \mathbf{y}) \rangle,$$

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

How to learn w on a complete graph?

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

$$\gamma_G(\mathbf{x}_i; \mathbf{w}_G) = F(\mathbf{x}_i, \mathbf{y}_i; \mathbf{w}_G) - \max_{\mathbf{y} \in \mathcal{Y}/\mathbf{y}_i} F(\mathbf{x}_i, \mathbf{y}; \mathbf{w}_G).$$

- **w** is solved by *maximum-margin principle* which aims to maximize $\gamma(\mathbf{x}_i; \mathbf{w}_G)$ over all training example.
- The problems are:
  - The $\mathcal{NP}$-hardness of 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.

Superposition of random trees

- $S(G)$ is a complete set of spanning tree generate from G.
- $\mathbf{w}_T = \{\mathbf{w}_{G,e}\}_{e \in T}$ is the projection of $\mathbf{w}_G$ on T.
- $\phi_T(\mathbf{x}, \mathbf{y}) = \{\phi_e(\mathbf{x}, \mathbf{y})\}_{e \in T}$ is the projection of $\phi_G(\mathbf{x}, \mathbf{y})$ on T.
- Rewrite

$$
\begin{aligned}
F(\mathbf{x}, \mathbf{y}, \mathbf{w}_G) &= \sum_{e \in G} \langle \mathbf{w}_{G,e}, \phi_{G,e}(\mathbf{x}, \mathbf{y}_e) \rangle \\
&= \frac{1}{\ell^{\ell-2}} \sum_{T \in S(G)} \sqrt{\frac{\ell}{2}} \langle \mathbf{w}_T, \phi_T(\mathbf{x}, \mathbf{y}_e) \rangle \\
&= \frac{1}{n} \sum_{i=1}^{n} a_{T_i} \langle \hat{\mathbf{w}}_{T_i}, \hat{\phi}_{T_i}(\mathbf{x}, \mathbf{y}_e) \rangle,
\end{aligned}
$$

$$
\frac{1}{n} \sum_{i=1}^{n} a_{T_i}^2 = 1, \quad \frac{1}{n} \sum_{i=1}^{n} a_{T_i} \leq 1, \quad n = \ell^{\ell-2}.
$$

How many trees?

- If there is a predictor $\mathbf{w}_G$ on complete graph achieves a margin on some training data, with high probability we need n spanning tree predictors $\{\mathbf{w}_{T_i}\}_{i=1}^{n}$ to achieve a close margin. n is quadratic in terms of ℓ.
- Recall

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

$$\frac{1}{n}\sum_{i=1}^{n} a_{T_i}^2 = 1, \quad \frac{1}{n}\sum_{i=1}^{n} a_{T_i} \leq 1, \cancel{n = \ell^{\ell-2}}.$$

Conical combination

- A sample $\mathcal{T}$ of n spanning trees drawn from G.
- Normalized feature weights $\hat{\mathbf{w}}_{T_i} = \frac{\mathbf{w}_{T_i}}{||\mathbf{w}_{T_i}||}$, $T_i \in \mathcal{T}$.
- Normalized feature vectors $\hat{\phi}_{T_i}(\mathbf{x}, \mathbf{y}) = \frac{\phi_{T_i}(\mathbf{x},\mathbf{y})}{||\phi_{T_i}(\mathbf{x},\mathbf{y})||}$, $T_i \in \mathcal{T}$.
- Conical combination

$$F(\mathbf{x}, \mathbf{y}, \mathbf{w}_{\mathcal{T}}) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} q_i \langle \hat{\mathbf{w}}_{T_i}, \hat{\phi}_{T_i}(\mathbf{x}, \mathbf{y}) \rangle$$

$$\sum_{i=1}^{n} q_i^2 = 1, \sum_{i=1}^{n} q_i \geq 1.$$

Convex combination

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

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

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

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

Optimization Problem

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

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

- To learn $\{\mathbf{w}_{T_t}\}_{T_t\in\mathcal{T}}$ 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

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

- The inference problem on each individual spanning tree can be solve efficiently in $\Theta(l)$ 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 $\Theta(Kl)$ 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 $\Theta(nKl)$ 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 $\mathcal{Y}_{\mathcal{T},K}$.

Fast Inference Over a Collection of Trees (cont.)

- Assume

$$\mathbf{y}_K^* = \operatorname*{argmax}_{\mathbf{y} \in \mathcal{Y}_{\mathcal{T},K}} F_{\mathcal{T}}(\mathbf{x}, \mathbf{y}; \mathbf{w}_{\mathcal{T}}).$$

If

$$F_{\mathcal{T}}(\mathbf{x}, \mathbf{y}_K^*; \mathbf{w}_{\mathcal{T}}) \geq \frac{1}{n} \sum_{t=1}^{n} F_{T_t}(\mathbf{x}, \mathbf{y}_{T_t,K}, \mathbf{w}_{T_t}) = \theta_{\mathbf{x}}(K),$$

then

$$F_{\mathcal{T}}(\mathbf{x}, \mathbf{y}_K^*; \mathbf{w}_{\mathcal{T}}) = \max_{\mathbf{y} \in \boldsymbol{\mathcal{Y}}} F_{\mathcal{T}}(\mathbf{x}, \mathbf{y}; \mathbf{w}_{\mathcal{T}}).$$

Fast Inference Over a Collection of Trees (cont.)

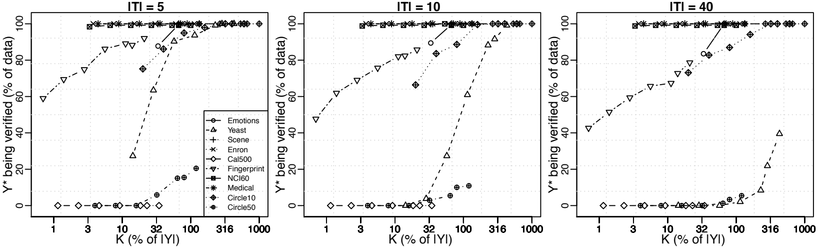
- For example, $\mathcal{T} = \{T_1, T_2\}, \boldsymbol{\mathcal{Y}} = \mathcal{Y}_1 \times \mathcal{Y}_2, \mathcal{Y}_i = \{+, -\}$

	T_1		T_2		$\theta_{\mathbf{x}}(K)$
	$\mathbf{y}_{T_1,K}$	$F_{T_1}(\mathbf{x}, \mathbf{y}_{T_1,K})$	$\mathbf{y}_{T_1,K}$	$F_{T_1}(\mathbf{x}, \mathbf{y}_{T_1,K})$	
$K = 1$	$+-$	5	$--$	4	9
$K = 2$	$++$	4	$-+$	3	7
$K = 3$	$-+$	3	$++$	3	6
$K = 4$	$--$	3	$+-$	2	5

- We proved that with a high probability $\mathbf{y}_{\mathcal{T}}$ will appear in $\mathcal{Y}_{\mathcal{T},K}$.

Performance of the Inference Algorithm

- 10 datasets, $|\mathcal{T}| = \{5, 10, 40\}$, $K = \{2, 4, 8, 16, 32, 40, 60\}$
- X-axis is the percentage of examples with exact inference.
- Y-axis is the value of K as the percentage of the number of microlabels.
- $K = 100\%|Y|$ corresponds to a complexity of $\Theta(nl^2)$.



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

- Theoretical study shows if a large margin structured output learner exists, then the combination of a random sample of spanning trees will achieve a similar margin with a high probability.
- The K-best inference algorithm is tractable which is proved theoretically and empirically.
- RTA is not constrained by the availability of the output graph, it can therefore be applied to a wider range of multilabel classification problem where the output graph is believed to play an important role during learning.