



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

September 12, 2015

The update-to-date version of this slide is available from [my GitHub page](#).

# About me

Take a look at [my homepage](#) and [my technical blog](#).

# Multilabel classification

- ▶ It is an important research field in machine learning.
- ▶ Input variable  $\mathbf{x} \in \mathcal{X}$  lives in some input space  $\mathcal{X}$ .
- ▶ Output variable  $\mathbf{y} = (y_1, \dots, y_\ell) \in \mathcal{Y}$  is a vector of  $\ell$  binary variables  $y_j \in \{+1, -1\}$ .
- ▶  $\mathbf{y}$  is called *multilabel*,  $y_j$  is called *microlabel*.
- ▶ Output space  $\mathcal{Y}$  is composed by a tensor product of  $\ell$  sets

$$\mathcal{Y} = \mathcal{Y}_1 \times \dots \times \mathcal{Y}_\ell, \mathcal{Y}_i = \{+1, -1\}.$$

- ▶ For example, in document classification, a document  $\mathbf{x}$  could be tagged with “news” “movie” “science” but not “sports” “politics” “finance”.

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

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

# Concerns

- ▶ Dimension of the search space: exponential in the number of microlabels.

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

- ▶ The dependency of microlabels needs to be exploited.
  - ▶ If a document is tagged with “movie”, then it is more likely to be in the category of “art” than “science”.

# 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.



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

- 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

- ▶ The scheme is proposed in [Tsoumakas et al., 2010]
- ▶ The output variable  $\mathbf{y}$  is assumed to be a flat vector.
- ▶ Problem transformation
  - ▶ Model the problem as a collection of single-label classification problems and solve each problem independently.
  - ▶ E.g., ML-KNN [Zhang and Zhou, 2007], CC [Read et al., 2011], IBLR [Cheng and Hüllermeier, 2009].
- ▶ Algorithm adaptation
  - ▶ Adapt single-label classification models to multilabel classification problems.
  - ▶ E.g., CORRLOG [Bian et al., 2012], MTL [Argyriou et al., 2008], ADABOOST.MH [Schapire and Singer, 1999, Esuli et al., 2008].
- ▶ These approaches do not model the dependency structure of microlabels.

# Structured output prediction

- ▶ The scheme is proposed in [Su, 2015].
- ▶ Model the dependency by an *output graph* defined on microlabels.
- ▶ Hierarchical classification
  - ▶ The output graph is a rooted tree defining different levels of granularities.
  - ▶ E.g., 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],  $M^3N$  [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 by 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 tree predictors [Marchand et al., 2014].
- ▶ Codes for developed models are available from <http://hongyusu.github.io>.

# Outline

- ▶ Preliminaries
- ▶ Structured output prediction
  - ▶ Undirected graph structure
  - ▶ DAG structure
  - ▶ unknown structure
- ▶ Experimental evaluations
- ▶ Conclusion and future work

# Preliminaries

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

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

- ▶  $y_i$  is a *microlabel* and  $y_i \in \{1, \dots, r_i\}$ ,  $r_i \in \mathbb{Z}$ .
- ▶ 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  $\phi(\mathbf{x}, \mathbf{y})$ .
- ▶  $\mathbf{w}$  is the weight vector in the joint feature space.
- ▶ Define a linear score function  $F(\mathbf{w}, \mathbf{x}, \mathbf{y}) = \langle \mathbf{w}, \phi(\mathbf{x}, \mathbf{y}) \rangle$ .
- ▶  $\mathbf{w}$  makes sure example  $\mathbf{x}$  with correct multilabel  $\mathbf{y}$  achieves higher score than with any other incorrect multilabel  $\mathbf{y}' \in \mathcal{Y}$ .

# Inference problem

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

$$\mathbf{y}_w(\mathbf{x}) = \operatorname{argmax}_{\mathbf{y} \in \mathcal{Y}} \langle \mathbf{w}, \phi(\mathbf{x}, \mathbf{y}) \rangle. \quad (1)$$

- ▶ Search space is exponential in size,  $|\mathcal{Y}| = 2^\ell$ .
- ▶ (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  $\phi$  is a potential function on a Markov network (undirected graph)  $G = (E, V)$ .
- ▶ 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 and is given *a priori*.

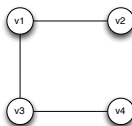


- ▶  $\varphi(\mathbf{x}) \in \mathbb{R}^d$  is the input feature map, e.g., bag-of-words of a document.
- ▶  $\psi(\mathbf{y}) \in \mathbb{R}^{|E|}$  is the output feature map which maps the multilabel  $\mathbf{y}$  into a collection of edges and labels

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

# An example of output feature map

- ▶ Markov network (undirected graph)  $G = (E, V)$



- ▶ Multilabel  $\mathbf{y}$

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

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

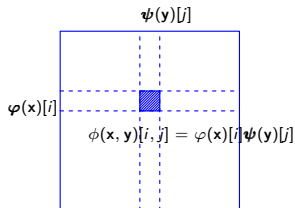
$$\psi(\mathbf{y}) = (\underbrace{0}_{--}, \underbrace{0}_{-+}, \underbrace{1}_{+-}, \underbrace{0}_{++}, \underbrace{0}_{--}, \underbrace{0}_{-+}, \underbrace{0}_{+-}, \underbrace{1}_{++}, \underbrace{0}_{--}, \underbrace{0}_{-+}, \underbrace{0}_{+-}, \underbrace{1}_{++})$$

$\underbrace{\hspace{10em}}_{(v_1, v_3)} \quad \underbrace{\hspace{10em}}_{(v_1, v_2)} \quad \underbrace{\hspace{10em}}_{(v_3, v_4)}$

# Joint feature map

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

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

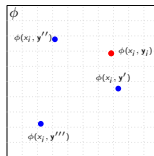


- The score function can be factorized by the output graph  $G$

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

# Optimization problem

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



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

$$\min_{\mathbf{w}, \xi_k} \quad \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{k=1}^m \xi_k \quad (2)$$

$$\text{s.t.} \quad \langle \mathbf{w}, \phi(\mathbf{x}_k, \mathbf{y}_k) \rangle - \langle \mathbf{w}, \phi(\mathbf{x}_k, \mathbf{y}) \rangle \geq \ell(\mathbf{y}_k, \mathbf{y}) - \xi_k, \\ \xi_k \geq 0, \forall \mathbf{y} \in \mathcal{Y}, k \in \{1, \dots, m\}.$$

- ▶  $\ell(\mathbf{y}, \mathbf{y}_k)$  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

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

- ▶ (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]

$$\mu(k, e, u_e) = \sum_{\mathbf{y}} \mathbf{1}_{\{\psi_e(\mathbf{y}) = u_e\}} \alpha(k, \mathbf{y}).$$

- ▶ The margin-dual optimization problem is

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

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

# Conditional gradient optimization

- ▶ (4) is optimized by conditional gradient decent.
- ▶ In each iteration it optimizes  $\mu_k$  that corresponds to a single example while keeps others ( $\mu_j, j \neq k$ ) fixed

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

- ▶ Current gradient of  $\mu_k$  is given by  $g_k = \ell_k - \sum_j K \mu_j$ .
- ▶ Compute the maximal feasible solution  $\mu_k^*$  as an update direction

$$\mu_k^* = \operatorname{argmax}_{\mu_k \in \mathcal{M}} \mu_k^\top g_k = \operatorname{argmax}_{\mu_k \in \mathcal{M}} \sum_e \mu(k, e)^\top g(k, e). \quad (5)$$

- ▶ (5) is an instantiation of MAP problem

Output graph	Inference problem	Inference algorithm
Tree	Polynomial	DP [Rousu et al., 2007]
Graph	$\mathcal{NP}$ -hard	LBP [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 ( $\tau$ )

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

- ▶  $\tau = 0$  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

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

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

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

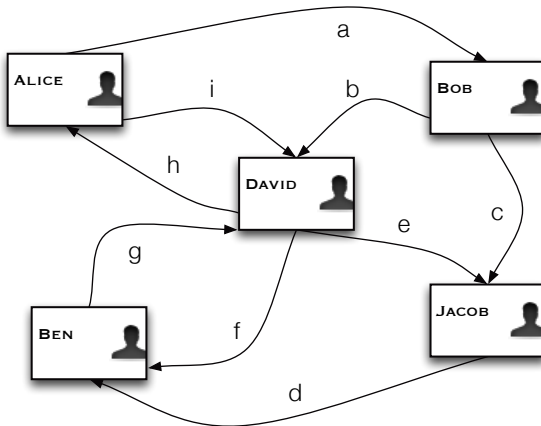
$$\begin{aligned} f(\mathbf{w}^t) - g(\mu^t) &= C \left( \ell - K^{\Delta \phi} \mu^t \right) - \mu^t \left( \ell - K^{\Delta \phi} \mu^t \right) \\ &= C^T \nabla g(\mu^t) - \mu^{t^T} \nabla g(\mu^t) \end{aligned}$$

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

# Output graph is directed

## Predicting network response

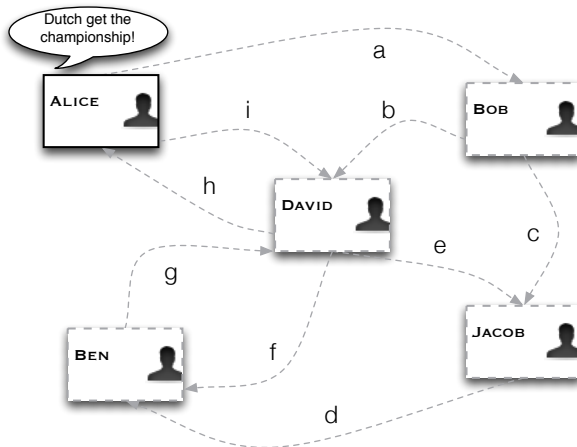
A twitter (follower-ship) network consists of five users.



# Output graph is directed

## Predicting network response

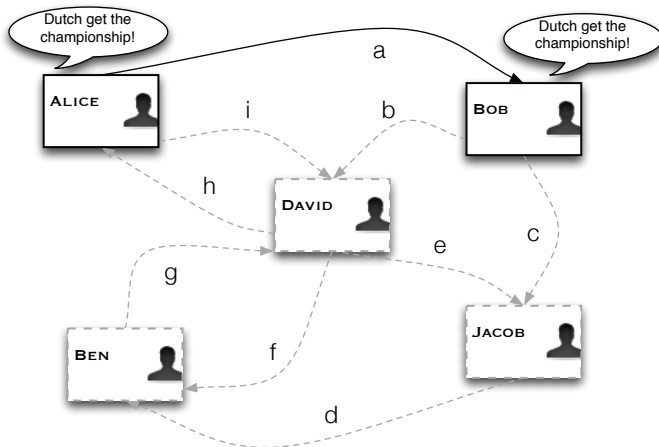
Alice tweets a message after World Cup final.



# Output graph is directed

## Predicting network response

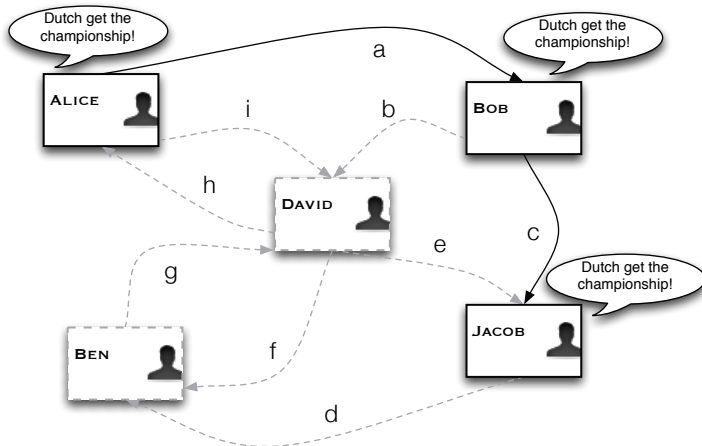
Bob sees the message and retweets the message from Alice.



# Output graph is directed

## Predicting network response

Jacob retweets the message from Bob.

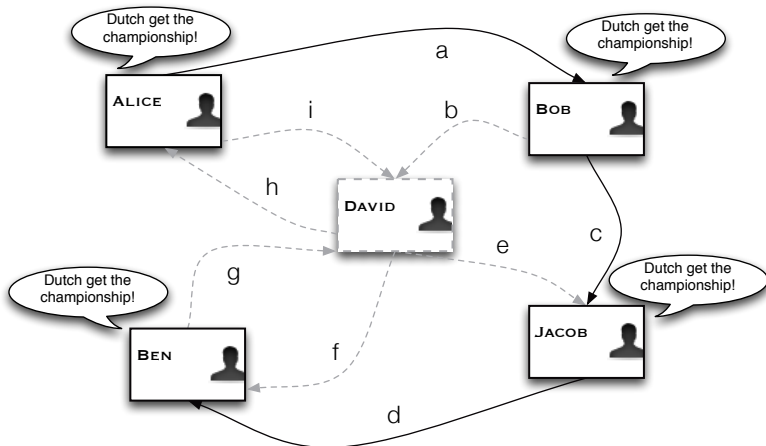




# Output graph is directed

## Predicting network response

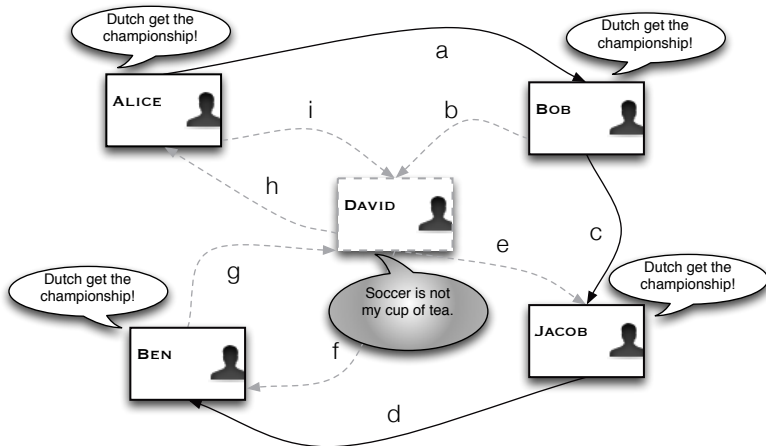
Ben retweets the message from Jacob.



# Output graph is directed

## Predicting network response

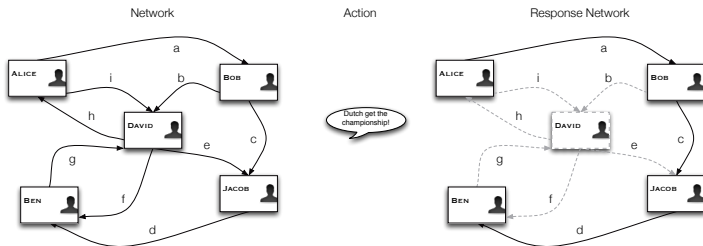
David is not a fan.



# Network response problem

► Definition:

- Given a complex network  $G = (E, V)$ , and an action  $x$  performed on the network.
- Task: predict the subnetwork that responds to the action.
  - Which nodes  $v \in V$  perform the action?  
 $V_x = \{\text{Alice}, \text{Bob}, \text{Jacob}, \text{Ben}\}$
  - Which directed edges  $e \in E_x$  relay the action from one node to its neighbors?  $E_x = \{a, c, d\}$

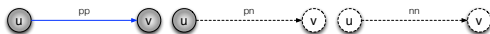


# Direct output graph

- ▶ Model is defined on directed network.
  - ▶ Any undirected network can be seen as special case by replacing undirected edges with two directed ones.



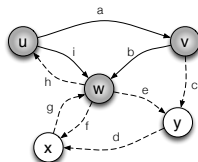
- ▶ Notation of edge labels:



- ▶ *Input feature*: Encode  $\mathbf{x}$  as  $\varphi(\mathbf{x})$  (e.g. bag-of-words of a tweet).
- ▶ *Output feature*: Encode  $G_y$  as  $\psi(y)$  (e.g. a set of edges and their labels)

$$\psi(y) = (\underbrace{1, 0, 0}_{a}, \underbrace{1, 0, 0}_{b}, \underbrace{0, 1, 0}_{c}, \dots)$$

$++ \quad +- \quad -- \quad ++ \quad +- \quad -- \quad ++ \quad +- \quad --$



# Structure output prediction model

- ▶ Compatibility score for  $(\mathbf{x}, \mathbf{y})$ :  $F(\mathbf{x}, \mathbf{y}, \mathbf{w}) = \langle \mathbf{w}, \phi(\mathbf{x}, \mathbf{y}) \rangle$ 
  - ▶  $\mathbf{w}$  is the feature weight to be learned.
  - ▶  $\phi(\mathbf{x}, \mathbf{y}) = \varphi(\mathbf{x}) \otimes \psi(\mathbf{y})$  is joint feature map.
  - ▶ Intuition: given an action  $\mathbf{x}$ , the score of correct response graph  $(\mathbf{x}, \mathbf{y})$  should be higher than any incorrect response graph  $(\mathbf{x}, \mathbf{y}')$

$$F(\mathbf{x}, \mathbf{y}, \mathbf{w}) > F(\mathbf{x}, \mathbf{y}', \mathbf{w}), \quad \forall \mathbf{y}' \in \mathcal{H}(G).$$

- ▶  $\mathbf{w}$  is learned by solving structured output learning problem

$$\begin{aligned} \min_{\mathbf{w}, \xi} \quad & \frac{1}{2} \|\mathbf{w}\|_2^2 + C \sum_{i=1}^m \xi_i \\ \text{s.t.} \quad & F(\mathbf{x}_i, \mathbf{y}_i; \mathbf{w}) > \max_{\mathbf{y}'_i \in \mathcal{H}(G)} (F(\mathbf{x}_i, \mathbf{y}'_i, ) \\ & + \ell_G(\mathbf{y}_i, \mathbf{y}'_i)) - \xi_i, \xi_i \geq 0, \forall i \in \{1, \dots, m\}, \end{aligned}$$

# Inference problem

- ▶ To solve the optimization, we have to solve similar inference problem appeared both in training and in prediction.
- ▶ In prediction phase:
  - ▶ Given the feature weight  $\mathbf{w}$  and the complex network  $G$ .
  - ▶ To find out a DAG  $H^* = (V_H, E_H)$  that gives the maximal compatibility score for a given action  $\mathbf{x}$

$$H^*(\mathbf{x}) = \operatorname{argmax}_{H \in \mathcal{H}(G)} \sum_{e \in E^H} s_{y_e}(e, \mathbf{x}, \mathbf{w}). \quad (7)$$

## Lemma

*Finding the graph that maximizes Eq. (7) is an  $\mathcal{NP}$ -hard problem.*

## Proof.

Reduction from MAX-CUT problem. □

# Approximate inference via SDP relaxation

- ▶ We formulate the inference problem as *integer quadratic program* (IQP).
  - ▶ Introduce for each node  $u \in V$  a binary variable  $x_u \in \{-1, +1\}$ .
  - ▶ Introduce a special variable  $x_0 \in \{-1, +1\}$  to distinguish activated node.

$$\begin{aligned} \max \quad & \frac{1}{4} \sum_{(u,v) \in E} [s_{pn}(u,v)(1 + x_0 x_u - x_0 x_v - x_u x_v) \\ & + s_{nn}(u,v)(1 - x_0 x_u - x_0 x_v + x_u x_v) \\ & + s_{pp}(u,v)(1 + x_0 x_u + x_0 x_v + x_u x_v)] \\ \text{s.t.} \quad & x_0, x_u, x_v \in \{-1, +1\}, \text{ for all } u, v \in V, \end{aligned}$$

- ▶ IQP is relaxed into *quadratic program* (QP) and solved by *semidefinite programming relaxation* (SDP).
- ▶ Optimization guarantee  $E[Z] \geq (\alpha - \epsilon)Z_R$  with  $\alpha > 0.796$ ,  $Z$  is objective achieved by SDP,  $Z_R$  is objective of IQP.

# Short summary

- We have seen so far.

Output graph	Inference problem	Inference algorithm
Tree	Polynomial	DP [Rousu et al., 2007]
Graph	$\mathcal{NP}$ -hard	LBP [Su et al., 2010]
→ DAG	$\mathcal{NP}$ -hard	SDP [Su et al., 2014]

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

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

- ▶ 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 complete graph  $G$

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

# Inference in terms of all spanning trees

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

$$\mathbf{y}_w(\mathbf{x}) = \operatorname{argmax}_{\mathbf{y} \in \mathcal{Y}} F(\mathbf{w}, \mathbf{x}, \mathbf{y}) = \operatorname{argmax}_{\mathbf{y} \in \mathcal{Y}} \sum_{e \in E} \langle \mathbf{w}_e, \phi_e(\mathbf{x}, \mathbf{y}_e) \rangle.$$

$$\phi_G(\mathbf{x}, \mathbf{y}) = \{\phi_{G,e}(\mathbf{x}, \mathbf{y}_e)\}_{e \in G}, \mathbf{w}_G = \{\mathbf{w}_{G,e}\}_{e \in G}, \|\phi_G(\mathbf{x}, \mathbf{y})\| = \|\mathbf{w}_G\| = 1$$

- ▶ For a complete graph, there are  $\ell^{\ell-2}$  unique spanning trees.
- ▶  $\phi_T(\mathbf{x}, \mathbf{y}) = \{\phi_e(\mathbf{x}, \mathbf{y})\}_{e \in T}$  is the projection of  $\phi_G(\mathbf{x}, \mathbf{y})$  on  $T \in \mathcal{S}(G)$ .
- ▶  $\mathbf{w}_T = \{\mathbf{w}_{G,e}\}_{e \in T}$  is the projection of  $\mathbf{w}_G$  on  $T \in \mathcal{S}(G)$ .
- ▶ We can write  $F(\mathbf{w}, \mathbf{x}, \mathbf{y})$  as a conic combination of all spanning trees

$$F(\mathbf{w}, \mathbf{x}, \mathbf{y}) = \mathbf{E}_{T \in U(G)} a_T \langle \mathbf{w}_T, \phi_T(\mathbf{x}, \mathbf{y}) \rangle$$
$$\mathbf{E}_{T \in U(G)} a_T^2 = 1, \quad \mathbf{E}_{T \in U(G)} a_T < 1.$$

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

# A sample of $n$ spanning trees

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

$$F_{\mathcal{T}}(\mathbf{w}, \mathbf{x}, \mathbf{y}) = \frac{1}{n} \sum_{i=1}^n a_{T_i} \langle \mathbf{w}_{T_i}, \phi_{T_i}(\mathbf{x}, \mathbf{y}) \rangle$$
$$\frac{1}{n} \sum_{i=1}^n a_{T_i}^2 = 1, \quad \frac{1}{n} \sum_{i=1}^n a_{T_i} < 1.$$

- ▶ 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]

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

- ▶ The marginal-dual form is given by

$$\begin{aligned} \max_{\mu \in \mathcal{M}} \quad & \sum_{i=1}^n \left( \mu_{T_i} \ell_{T_i} - \frac{1}{2} \mu_{T_i} K_{T_i}^{\Delta \phi} \mu_{T_i} \right) \\ \text{s.t.} \quad & \sum_{u_e} \mu_{T_i, e}(u_e) \leq C. \end{aligned}$$

- ▶ Inside the summation, there is a structure output model with parameter  $\mu_{T_i}$  defined on a spanning tree  $T_i$ .
- ▶ The problem is how to jointly optimize structured output models defined on  $n$  spanning trees.

# 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

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

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

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

- ▶ 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 for 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

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

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

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

- ▶ We prove that with high probability the global best multilabel will exist in  $K$  best list.
- ▶ We have developed a condition to verify the global best multilabel from  $K$  best list in linear time  $\Theta(nK)$ .

# Short summary

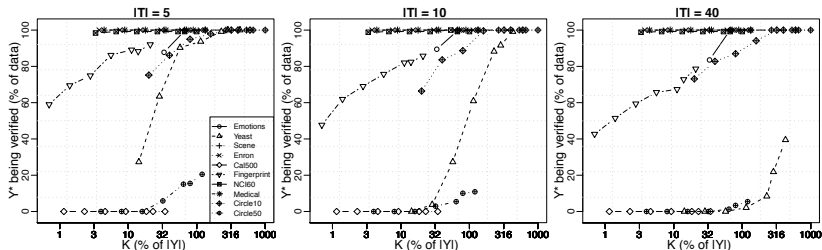
- We have seen so far.

Output graph	Inference problem	Inference algorithm
Tree	Polynomial	DP [Rousu et al., 2007]
Graph	$\mathcal{NP}$ -hard	LBP [Su et al., 2010]
DAG	$\mathcal{NP}$ -hard	SDP [Su et al., 2014]
→unknown	$\mathcal{NP}$ -hard	MVE AMM MAM [Su and Rousu, 2015] RTA [Marchand et al., 2014]



# RTA inference algorithm

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



# RTA on multilabel benchmark datasets

- ▶ Prediction performance on multilabel benchmark datasets.
- ▶ Measurement of success is microlabel accuracy and multilabel accuracy.
- ▶ The result is shown in the following table.

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	<b>18.8</b>	77.8	74.5	71.3	69.6	<b>66.3</b>
YEAST	20.0	20.7	21.7	20.1	<b>19.8</b>	85.9	88.7	93.0	86.0	<b>77.7</b>
SCENE	9.8	11.6	18.4	17.0	<b>8.8</b>	47.2	55.2	72.2	94.6	<b>30.2</b>
ENRON	6.4	6.5	6.2	<b>5.0</b>	5.3	99.6	99.6	92.7	87.9	<b>87.7</b>
CAL500	<b>13.7</b>	13.8	<b>13.7</b>	<b>13.7</b>	13.8	100.0	100.0	100.0	100.0	100.0
FINGERPRINT	<b>10.3</b>	17.3	10.5	10.5	10.7	99.0	100.0	99.6	99.6	<b>96.7</b>
NCI60	15.3	16.0	14.6	<b>14.3</b>	14.9	56.9	53.0	63.1	60.0	<b>52.9</b>
MEDICAL	2.6	2.6	<b>2.1</b>	<b>2.1</b>	<b>2.1</b>	91.8	91.8	63.8	63.1	<b>58.8</b>
CIRCLE10	4.7	6.3	2.6	2.5	<b>0.6</b>	28.9	33.2	20.3	17.7	<b>4.0</b>
CIRCLE50	5.7	6.2	<b>1.5</b>	2.1	3.8	69.8	72.3	<b>38.8</b>	46.2	52.8

# SPIN for context-sensitive prediction

- ▶ We assume action  $\varphi(\mathbf{x})$  is known (e.g. bag-of-words of a tweet).
- ▶ Task is to predict the response network given the action.
- ▶ *Predicted Subgraph Coverage* (PSC) is the relative size of correctly predicted subgraph in terms of node labels.
- ▶ The result is shown in the following table.

Dataset	Node Accuracy			Node $F_1$ Score			Edge Acc		PSC		
	SVM	MMCRF	SPIN	SVM	MMCRF	SPIN	SVM	SPIN	SVM	MMCRF	SPIN
memeS	<b>73.4</b>	68.0	72.2	39.0	39.8	<b>47.1</b>	<b>62.7</b>	45.6	23.4	25.3	<b>33.6</b>
memeM	<b>82.1</b>	79.0	81.5	29.1	30.1	<b>38.0</b>	61.1	<b>68.8</b>	18.6	18.8	<b>28.3</b>
memeL	<b>89.9</b>	88.3	89.8	26.7	27.1	<b>35.0</b>	45.5	<b>80.0</b>	17.7	18.9	<b>27.6</b>
M700	91.9	<b>94.1</b>	92.1	13.8	7.3	<b>14.2</b>	26.3	<b>93.0</b>	29.4	23.9	<b>34.4</b>
M1k	94.1	<b>95.8</b>	94.2	<b>10.9</b>	3.5	9.3	26.6	<b>94.7</b>	33.7	16.6	<b>35.2</b>
M2k	96.8	<b>97.6</b>	96.7	<b>6.2</b>	1.4	3.4	25.3	<b>97.6</b>	<b>34.6</b>	9.6	14.7
L700	89.7	<b>92.4</b>	89.7	16.2	9.4	<b>17.3</b>	26.5	<b>90.4</b>	9.5	6.7	<b>12.5</b>
L1k	92.4	<b>94.4</b>	91.5	12.4	6.4	<b>13.9</b>	26.4	<b>92.3</b>	6.1	4.4	<b>8.4</b>
L2k	92.5	<b>94.5</b>	91.9	12.3	5.4	<b>12.7</b>	26.5	<b>93.2</b>	6.0	2.9	<b>7.2</b>
Geom.	85.5	86.4	<b>86.6</b>	19.8	12.6	<b>20.3</b>	32.6	<b>79.7</b>	18.9	14.2	<b>21.7</b>

# SPIN for context-free prediction

- ▶ We assume action is unknown during prediction phase.
- ▶ Task is to predict directed edges (network skeleton) from a cascade of actions.
- ▶ The measure of success is *Precision@K*, where we ask for top-*K* percent edge predictions and compute the precision.
- ▶ The result is shown in the following table.

Dataset	Model	T ( $10^3$ s)	Precision @ K					
			10%	20%	30%	40%	50%	60%
memeS	SPIN	5.50	<b>82.9</b>	<b>81.0</b>	<b>76.0</b>	<b>74.0</b>	<b>74.0</b>	<b>70.0</b>
	ICM-EM	<b>0.01</b>	60.3	63.5	65.1	62.0	62.0	61.5
	NETRATE	5.83	76.2	73.8	70.4	68.7	68.7	66.8
memeM	SPIN	5.52	<b>82.7</b>	<b>72.1</b>	<b>70.5</b>	<b>69.2</b>	<b>69.2</b>	<b>67.9</b>
	ICM-EM	<b>0.02</b>	56.3	55.3	56.8	57.4	57.4	56.3
	NETRATE	13.93	61.2	64.6	62.9	62.5	62.5	62.4
memeL	SPIN	4.75	<b>82.2</b>	<b>73.6</b>	<b>69.1</b>	<b>66.7</b>	<b>66.7</b>	<b>65.9</b>
	ICM-EM	<b>0.01</b>	52.1	55.7	54.2	56.5	56.5	56.7
	NETRATE	12.63	56.5	57.8	60.0	59.3	59.3	59.4

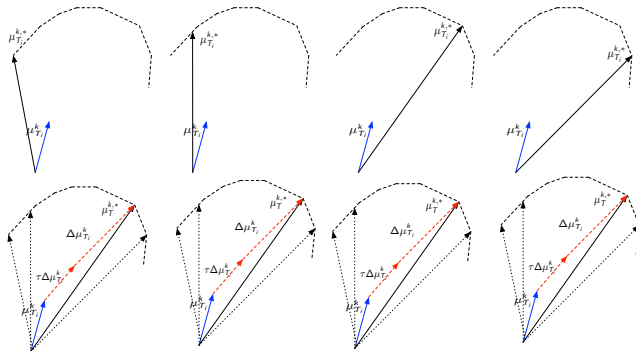
# Conclusion

- ▶ Structured output prediction is family of methods designed for multilabel classification problems.
- ▶ The output graph is often assume to be known *a priori*.
  - ▶ MMCRF assumes tree or general undirected graph as output graph.
  - ▶ SPIN assumes DAG as output graph.
- ▶ In addition, we focus on the problems where the output graph is unobserved.
  - ▶ MVE AMM MAM aggregates the inference results from based models.
  - ▶ RTA is a unified learning and inference framework.
    - ▶ Model all pairwise correlations with a complete graph.
    - ▶ Under margin assumption, the properties of a complete graph can be achieved by a collection of its spanning tree.
- ▶ All developed models are tested with real-world applications or benchmark datasets.
- ▶ Codes are available from <http://hongyusu.github.io>.

# Future work

## Inference algorithm

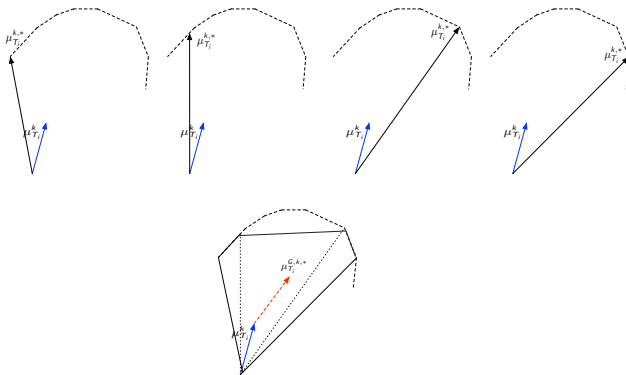
- From K-best inference algorithm developed for RTA to a Newton method that does a conic combination of multiple update directions.



# Future work

## Inference algorithm

- From K-best inference algorithm developed for RTA to a Newton method that does a conic combination of multiple update directions.



# Future work

## $L_1$ norm random spanning tree approximation $L_1RTA$

- From conic combination of a collection of random spanning trees

$$F(\mathbf{w}, \mathbf{x}, \mathbf{y}) = \mathbf{E}_{T \in U(G)} a_T \langle \mathbf{w}_T, \phi_T(\mathbf{x}, \mathbf{y}) \rangle$$
$$\mathbf{E}_{T \in U(G)} a_T^2 = 1, \quad \mathbf{E}_{T \in U(G)} a_T < 1.$$

- To convex combination of a collection of random spanning trees

$$F(\mathbf{w}, \mathbf{x}, \mathbf{y}) = \mathbf{E}_{T \in U(G)} a_T \langle \mathbf{w}_T, \phi_T(\mathbf{x}, \mathbf{y}) \rangle$$
$$\mathbf{E}_{T \in U(G)} a_T = 1, \quad \mathbf{E}_{T \in U(G)} a_T < 1.$$

- Optimization problem

$$\min_{\mathbf{w}_{T_i}, \xi_i} \quad \frac{1}{2} \left( \sum_{i=1}^n \|\mathbf{w}_{T_i}\| \right)^2 + C \sum_{k=1}^m \xi_k$$
$$\text{s.t.} \quad \frac{1}{n} \sum_{i=1}^n \langle \mathbf{w}_{T_i}, \phi_{T_t}(\mathbf{x}_k, \mathbf{y}_k) \rangle - \max_{\mathbf{y} \neq \mathbf{y}_k} \frac{1}{n} \sum_{i=1}^n \langle \mathbf{w}_{T_t}, \phi_{T_i}(\mathbf{x}_k, \mathbf{y}) \rangle \geq 1 - \xi_k,$$
$$\xi_k \geq 0, \forall k \in \{1, \dots, m\}.$$



# 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. (2011).

Classifier chains for multi-label classification.

*Machine Learning*, 85(3):333–359.

# Bibliography (cont.)



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).

MI-knn: A lazy learning approach to multi-label learning.

*Pattern Recognition*, 40:2007.