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 13, 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 $\mathbf{x} \in \boldsymbol{\mathcal{X}}$ lives in some input space $\boldsymbol{\mathcal{X}}$.
- Output variable $\mathbf{y} = (y_1, \cdots, y_l) \in \boldsymbol{\mathcal{Y}}$ is a vector of ℓ binary variables $y_j \in \{+1, -1\}$.
- $\mathbf{y}$ is called *multilabel*, y_j is called *microlabel*.
- Output space $\boldsymbol{\mathcal{Y}}$ is composed by a tensor product of ℓ sets

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

- 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 $\mathbf{x}$, $f : \boldsymbol{\mathcal{X}} \to \boldsymbol{\mathcal{Y}}$.

Concerns

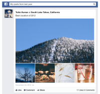
- Dimension of the search space: exponential in the number of microlabels.

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

- 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.


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

- The scheme is proposed in [Tsoumakas et al., 2010]
- The output variable **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].
- Models 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 has a more general form (e.g., a tree, a chain).
  - E.g., 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

- SOP models developed for observed output graph.
  - MMCRF on general output graph structures [Su et al., 2010].
  - SPIN on DAG for network influence prediction [Su et al., 2014].
- SOP models developed for 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
  - DAG
  - unknown output graph
- Experimental evaluations
- Conclusions and future work

Preliminaries

- Training examples come in pairs $(\mathbf{x}, \mathbf{y}) \in \boldsymbol{\mathcal{X}} \times \boldsymbol{\mathcal{Y}}$.
- $\boldsymbol{\mathcal{X}}$ 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 operates in the joint feature space.
- Define a linear score function $F(\mathbf{w}, \mathbf{x}, \mathbf{y}) = \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}, \mathbf{y}) \rangle$.
- $\mathbf{w}$ ensures that example $\mathbf{x}_i$ with correct multilabel $\mathbf{y}_i$ achieves higher score than with any other incorrect multilabel $\mathbf{y}' \in \boldsymbol{\mathcal{Y}}$.

Prediction

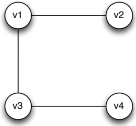
- The prediction $\mathbf{y}_\mathbf{w}(\mathbf{x})$ of an input $\mathbf{x}$ is the multilabel $\mathbf{y}$ that maximizes the score function

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

- Search space is exponential in size, $|\boldsymbol{\mathcal{Y}}| = 2^\ell$.
- (1) is called *inference* problem which is $\mathcal{NP}$-hard for most output feature maps.
- Often, we want a 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 (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 *apriori*.

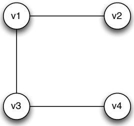


- $\boldsymbol{\varphi}(\mathbf{x}) \in \mathbb{R}^d$ is the input feature map, e.g., bag-of-words of a document.
- $\boldsymbol{\psi}(\mathbf{y}) \in \mathbb{R}^{4|E|}$ is the output feature map which maps the multilabel $\mathbf{y}$ into a collection of edges and labels

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

An example of $\psi(\mathbf{y})$

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



- Multilabel $\mathbf{y}$

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

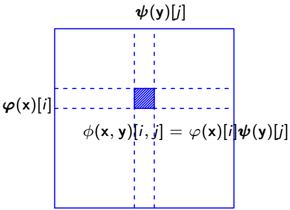
- Output feature map $\boldsymbol{\psi}(\mathbf{y})$

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

Joint feature map $\phi(\mathbf{x}, \mathbf{y})$

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

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



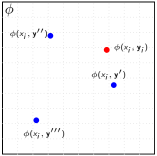
- The score function can be factorized by the output graph G

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

Optimization problem

- Max-margin learning for **w**

$$\gamma(\mathbf{w}, \mathbf{x}_i) = F(\mathbf{w}, \mathbf{x}_i, \mathbf{y}_i) - \max_{\mathbf{y} \in \boldsymbol{\mathcal{Y}}/\mathbf{y}_i} F(\mathbf{w}, \mathbf{x}_i, \mathbf{y})$$



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

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

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

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

- (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]

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

- 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 variables is $m \times 4|E|$.

Conditional gradient optimization

- (4) is optimized by conditional gradient decent.
- In each iteration it 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 the maximal feasible solution μ_k^* as an update direction

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

- (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 (τ)

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

- $\tau = 0$ 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).

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

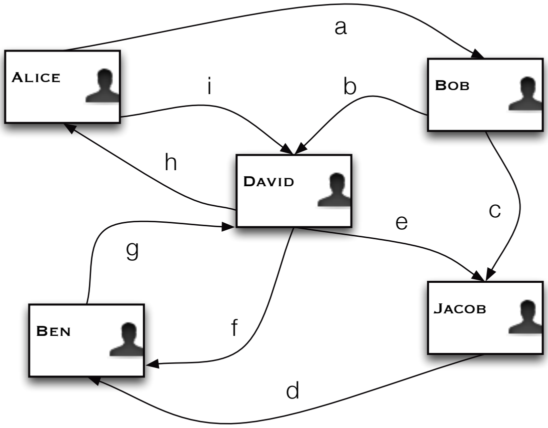
$$\begin{aligned} f(\mathbf{w}^t) - g(\mu^t) &= C\left(\ell - K^{\Delta\boldsymbol{\phi}}\mu^t\right) - \mu^t\left(\ell - K^{\Delta\boldsymbol{\phi}}\mu^t\right) \\ &= C^{\intercal}\nabla g(\mu^t) - \mu^{t\intercal}\nabla g(\mu^t) \end{aligned}$$

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 $C^{\intercal}\nabla g(\mu^t)$.

Output graph is directed

Predicting network response

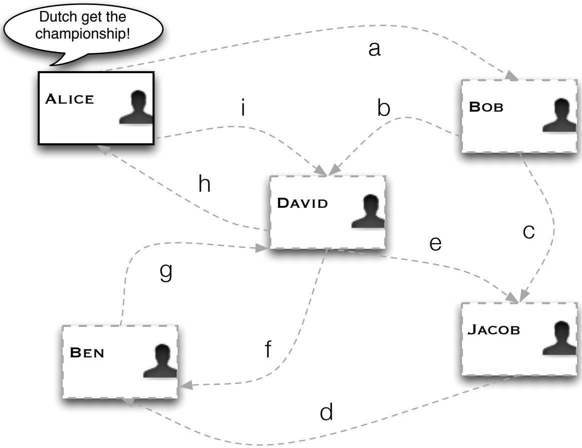
A twitter (follower-ship) network consists of five users.



Output graph is directed

Predicting network response

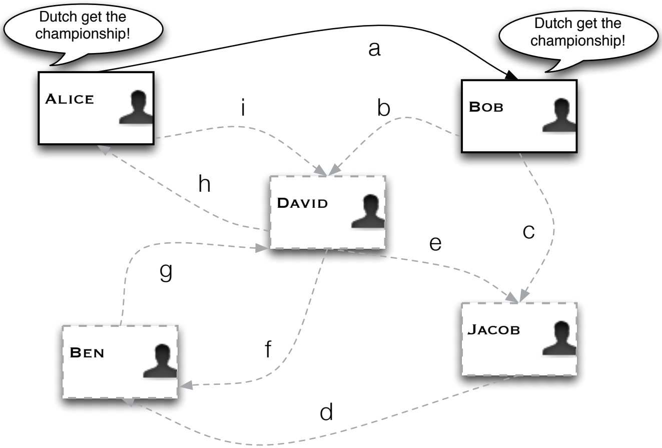
Alice tweets a message after World Cup final.



Output graph is directed

Predicting network response

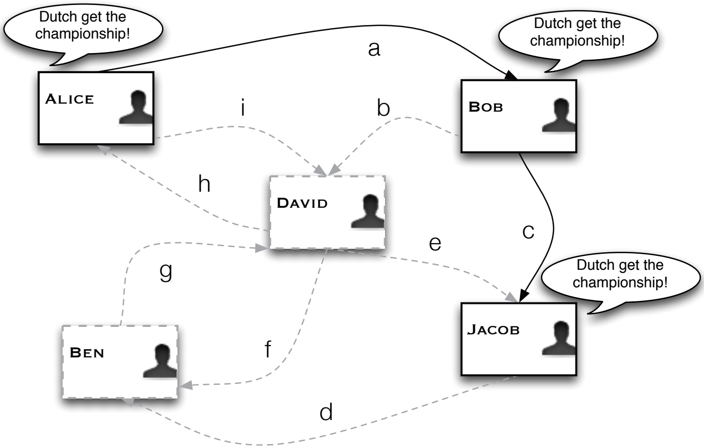
Bob sees the message and retweets the message from Alice.



Output graph is directed

Predicting network response

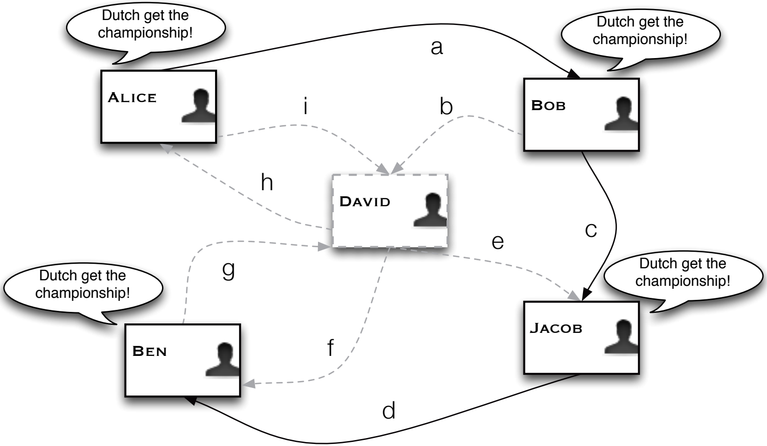
Jacob retweets the message from Bob.



Output graph is directed

Predicting network response

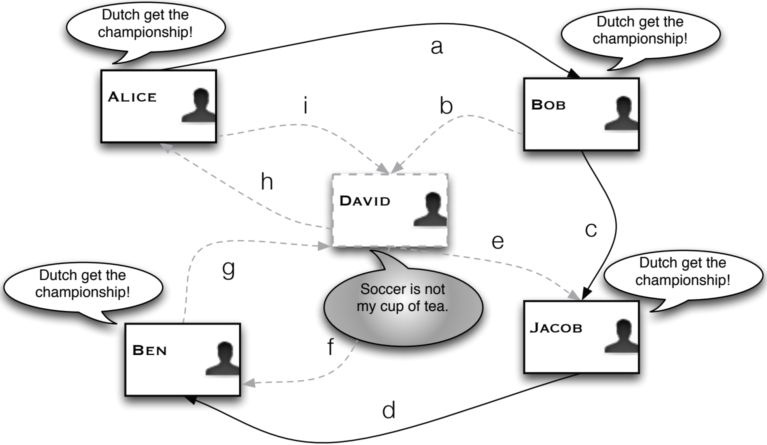
Ben retweets the message from Jacob.



Output graph is directed

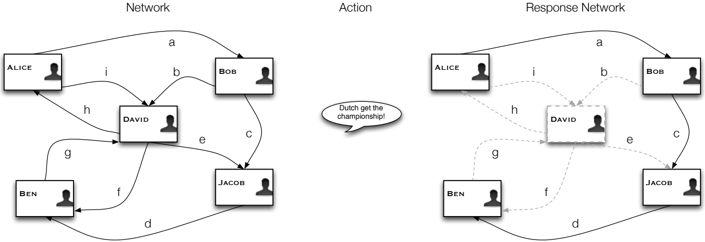
Predicting network response

David is not a fan.



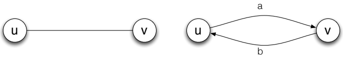
Network response problem

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

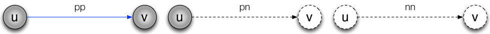


Direct output graph

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

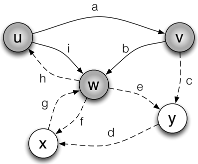


- Notation of edge labels:



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

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



Structure output prediction model

- Compatibility score for $(\mathbf{x}, \mathbf{y})$: $F(\mathbf{x}, \mathbf{y}, \mathbf{w}) = \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}, \mathbf{y}) \rangle$
  - $\mathbf{w}$ is the feature weight to be learned.
  - $\boldsymbol{\phi}(\mathbf{x}, \mathbf{y}) = \boldsymbol{\varphi}(\mathbf{x}) \otimes \boldsymbol{\psi}(\mathbf{y})$ 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

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

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 $\mathbf{x}$

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

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.

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

- 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]
$\rightarrow$ 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 $\mathcal{NP}$-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 (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.
- $\boldsymbol{\varphi}(\mathbf{x})$ 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

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

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

- 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 $T\in S(G)$.
- $\mathbf{w}_T = \{\mathbf{w}_{G,e}\}_{e\in T}$ is the projection of $\mathbf{w}_G$ on $T\in S(G)$.
- We can write $F(\mathbf{w},\mathbf{x},\mathbf{y})$ as a conic combination of all spanning trees

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

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

- $U(G)$ 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

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

  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]

$$\min_{\mathbf{w}_{T_i},\xi_i} \quad \frac{1}{2}\sum_{i=1}^{n}||\mathbf{w}_{T_i}||^2 + C\sum_{k=1}^{m}\xi_k$$

$$\text{s.t.} \quad \frac{1}{\sqrt{n}}\sum_{i=1}^{n}\langle \mathbf{w}_{T_i}, \boldsymbol{\phi}_{T_t}(\mathbf{x}_k,\mathbf{y}_k)\rangle - \max_{\mathbf{y}\neq\mathbf{y}_k}\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\langle \mathbf{w}_{T_t}, \boldsymbol{\phi}_{T_i}(\mathbf{x}_k,\mathbf{y})\rangle \geq 1-\xi_k,$$

$$\xi_k \geq 0\,, \forall\ k \in \{1,\dots,m\}.$$

- The marginal-dual form is given by

$$\max_{\mu\in\mathcal{M}} \quad \sum_{i=1}^{n}\left(\mu_{T_i}\ell_{T_i} - \frac{1}{2}\mu_{T_i}K_{T_i}^{\Delta\phi}\mu_{T_i}\right)$$

$$\text{s.t.} \quad \sum_{u_e}\mu_{T_i,e}(u_e) \leq C.$$

- Inside the summation, there is a structure output model with parameter μ_{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

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

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

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

- 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

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

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

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

- 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 $\Theta(nK)$.

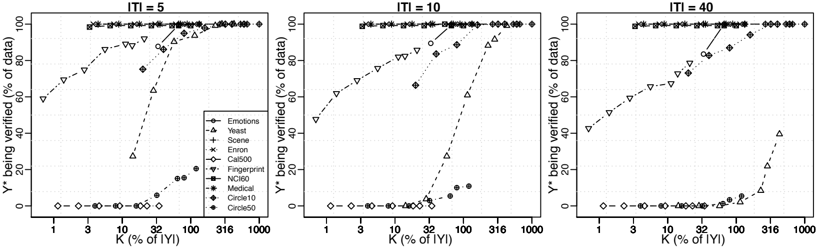
Short summary

- We have seem 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]
$\rightarrow$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*	**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*

SPIN for context-sensitive prediction

- We assume action $\varphi(\mathbf{x})$ is known (e.g. bag-of-word 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	**73.4**	68.0	*72.2*	39.0	*39.8*	**47.1**	**62.7**	45.6	23.4	*25.3*	**33.6**
memeM	**82.1**	79.0	*81.5*	29.1	*30.1*	**38.0**	61.1	**68.8**	18.6	*18.8*	**28.3**
memeL	**89.9**	88.3	*89.8*	26.7	*27.1*	**35.0**	45.5	**80.0**	17.7	*18.9*	**27.6**
M700	91.9	**94.1**	*92.1*	*13.8*	7.3	**14.2**	26.3	**93.0**	*29.4*	23.9	**34.4**
M1k	94.1	**95.8**	*94.2*	**10.9**	3.5	*9.3*	26.6	**94.7**	*33.7*	16.6	**35.2**
M2k	*96.8*	**97.6**	96.7	**6.2**	1.4	*3.4*	25.3	**97.6**	**34.6**	9.6	*14.7*
L700	89.7	**92.4**	*89.7*	*16.2*	9.4	**17.3**	26.5	**90.4**	*9.5*	6.7	**12.5**
L1k	*92.4*	**94.4**	91.5	*12.4*	6.4	**13.9**	26.4	**92.3**	*6.1*	4.4	**8.4**
L2k	*92.5*	**94.5**	91.9	*12.3*	5.4	**12.7**	26.5	**93.2**	*6.0*	2.9	**7.2**
Geom.	85.5	*86.4*	**86.6**	*19.8*	12.6	**20.3**	32.6	**79.7**	*18.9*	14.2	**21.7**

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 (10^3s)	Precision @ K					
			10%	20%	30%	40%	50%	60%
memeS	SPIN	*5.50*	**82.9**	**81.0**	**76.0**	**74.0**	**74.0**	**70.0**
	ICM-EM	**0.01**	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*	**82.7**	**72.1**	**70.5**	**69.2**	**69.2**	**67.9**
	ICM-EM	**0.02**	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*	**82.2**	**73.6**	**69.1**	**66.7**	**66.7**	**65.9**
	ICM-EM	**0.01**	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*

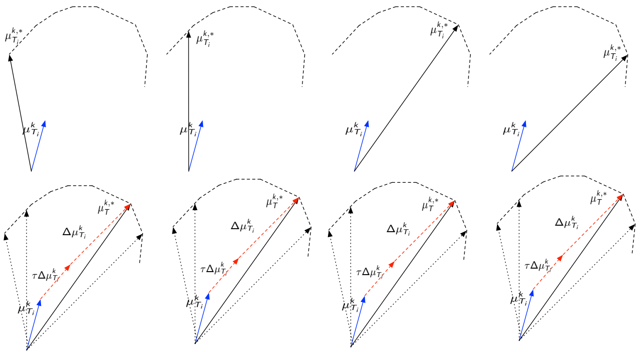
Conclusion

- Structured output prediction is family of methods designed for multilabel classification problems.
- The output graph is often assume to be known *apriori*.
  - 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

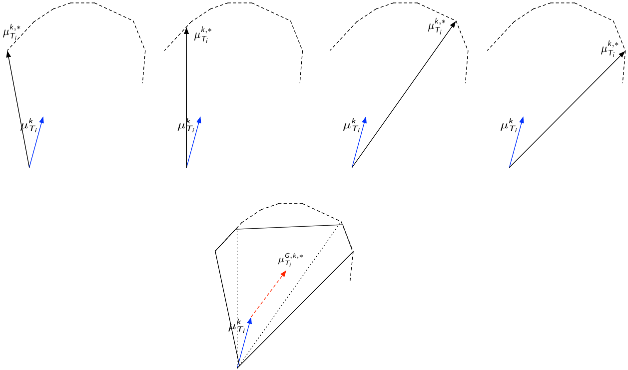
- From K-best inference algorithm developed for RTA to a Newton method that does a conic combination of multiple update directions.



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

$\mathbf{L}_1$ norm random spanning tree approximation $\mathrm{L_1RTA}$

- From conic combination of a collection of random spanning trees

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

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

- To convex combination of a collection of random spanning trees

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

$$\mathop{\mathbf{E}}_{T\in U(G)} a_T = 1, \ \mathop{\mathbf{E}}_{T\in U(G)} a_T < 1.$$

- Optimization problem

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

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).
Ml-knn: A lazy learning approach to multi-label learning.
Pattern Recognition, 40:2007.