Aalto University
School of Science
and Technology

Structured output prediction for multilabel classification

Hongyu Su

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

August 6, 2015

Multilabel classification

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

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

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

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

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

Central problems in multilabel classification

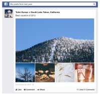
- The size of the output space (searching space) is exponential in the number of microlabels.

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

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

Real world applications

- Social network, information can spread through multiple users.



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

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

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

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

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

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

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

Flat multilabel classification approaches

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

Structured output prediction

- Model the dependency structure with an output graph defined on microlabels.
- The categorization is proposed in [Su, 2015].
- Hierarchical classification
  - The output graph is a rooted tree or a DAG defining different levels of granularities.
  - For example, SSVM [Tsochantaridis et al., 2004, Tsochantaridis et al., 2005].
- Graph labeling
  - The output graph takes a more general form (e.g., a tree, a chain).
  - For example, CRF [Lafferty et al., 2001, Taskar et al., 2002], $\mathrm{M^3N}$ [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 structure [Su et al., 2010].
- Structured output prediction models working with unknown output graph.
  - MVE to combine multiple structured output predictors with ensemble [Su and Rousu, 2011].
  - AMM and MAM to aggregate the inference results from multiple structured output predictors [Su and Rousu, 2013, Su and Rousu, 2015].
  - RTA to perform joint learning and inference over a collection of random spanning trees [Marchand et al., 2014].
- All developed models are available from hongyusu.github.io.

Outline

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, $\mathrm{M^3N}$, 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).

Today

- 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 in NIPS paper.
  - convex combination as future work.
- Derivations and the corresponding optimization problems.

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.
- $\varphi(\mathbf{x})$ is the input feature map, e.g., a feature vector of d dimension.
- $\Gamma_G(\mathbf{y})$ is the output feature map of $\mathbf{y}$ on G of $4 \times |E|$ dimension

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

- 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

$$\mathbf{y}(\mathbf{x}) = \underset{\mathbf{y} \in \boldsymbol{\mathcal{Y}}}{\mathbf{argmax}}\, F(\mathbf{x}, \mathbf{y}; \mathbf{w}_G) = \underset{\mathbf{y} \in \boldsymbol{\mathcal{Y}}}{\mathbf{argmax}} \sum_{e \in G} \langle \mathbf{w}_{G,e}, \phi_{G,e}(\mathbf{x}, \mathbf{y}_e) \rangle,$$

  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

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

- $\mathbf{w}$ is solved by *max-margin principle* which aims to maximize $\gamma(\mathbf{x}_i;\mathbf{w}_G)$ over all training example $\mathbf{x}_i$, $i \in \{1,\cdots,m\}$.
- The inference problem on a complete graph is $\mathcal{NP}$-hardness.
- The parameter space is quadratic in the number of microlabels k.
- 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, $|S(G)| = \ell^{\ell-2}$.
- Recall
  $\phi_G(\mathbf{x},\mathbf{y}) = \{\phi_{G,e}(\mathbf{x},\mathbf{y}_e)\}_{e\in G}, \mathbf{w}_G = \{\mathbf{w}_{G,e}\}_{e\in G}, ||\phi_G(\mathbf{x},\mathbf{y})|| = ||\mathbf{w}_G|| = 1.$
- $\phi_T(\mathbf{x},\mathbf{y}) = \{\phi_e(\mathbf{x},\mathbf{y})\}_{e\in T}$ 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 $T \in S(G)$.
- Rewrite

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

$$||\hat{\phi}_T(\mathbf{x},\mathbf{y})|| = ||\hat{\mathbf{w}}_T|| = 1,\ \frac{1}{n}\sum_{i=1}^{n} a_{T_i}^2 = 1,\ \frac{1}{n}\sum_{i=1}^{n} a_{T_i} \le 1,\ a_{T_i} \ge 0,\ n = \ell^{\ell-2}.$$

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

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

$$||\hat{\phi}_{\mathcal{T}}(\mathbf{x},\mathbf{y})|| = ||\hat{\mathbf{w}}_{\mathcal{T}}|| = 1,\ \frac{1}{n}\sum_{i=1}^{n} a_{T_i}^2 = 1,\ \frac{1}{n}\sum_{i=1}^{n} a_{T_i} \leq 1,\ a_{T_i} \geq 0,\ \cancel{n = \ell^{\ell-2}}.$$

Conical combination

- A sample $\mathcal{T} = \{T_1, \cdots, T_n\}$ of n spanning trees drawn from G.
- Normalized feature vectors $\hat{\phi}_{T_i}(\mathbf{x}, \mathbf{y}) = \frac{\phi_{T_i}(\mathbf{x},\mathbf{y})}{||\phi_{T_i}(\mathbf{x},\mathbf{y})||}$, $T_i \in \mathcal{T}$.
- Normalized feature weights $\hat{\mathbf{w}}_{T_i} = \frac{\mathbf{w}_{T_i}}{||\mathbf{w}_{T_i}||}$, $T_i \in \mathcal{T}$.
- Conical combination of spanning trees

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

$$\sum_{i=1}^{n} q_i^2 = 1,\ q_i \geq 0,\ \forall i \in \{1, \cdots, n\}.$$

Conical combination (cont.)

- To solve $\{\mathbf{w}_{T_i}\}_{T_i \in \mathcal{T}}$, we need to work on the optimization problem

$$\min_{\xi,\gamma,\mathbf{q},\mathcal{W}} \quad \frac{1}{2\gamma^2} + \frac{C}{\gamma}\sum_{k=1}^{m}\xi_k$$

$$\textbf{s.t.} \quad \frac{1}{\sqrt{n}}\sum_{i=1}^{n} q_i\langle \hat{\mathbf{w}}_{T_i}, \hat{\phi}_{T_i}(\mathbf{x}_k, \mathbf{y}_k)\rangle - \max_{\mathbf{y}\in\boldsymbol{\mathcal{Y}}} \frac{1}{\sqrt{n}}\sum_{i=1}^{n} q_i\langle \hat{\mathbf{w}}_{T_i}, \hat{\phi}_{T_i}(\mathbf{x}_k, \mathbf{y})\rangle$$

$$\geq \gamma - \xi_k, \xi_k \geq 0, \forall k \in \{1, \cdots, m\}, \sum_{i=1}^{n} q_i^2 = 1, q_i \geq 0, \forall i \in \{1, \cdots, n\}.$$

- This is equivalent to

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

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

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

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

Convex 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 $\hat{\phi}_{T_i}(\mathbf{x}, \mathbf{y}) = \frac{\phi_{T_i}(\mathbf{x},\mathbf{y})}{||\phi_{T_i}(\mathbf{x},\mathbf{y})||}$, $T_i \in \mathcal{T}$.
- Convex combination of spanning trees

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

$$\sum_{i=1}^{n} q_i = 1,\ q_i \geq 0,\ \forall i \in \{1, \cdots, n\}.$$

Convex combination (cont.)

- To solve $\{\mathbf{w}_{T_i}\}_{T_i \in \mathcal{T}}$, we need to work on the optimization problem

$$\min_{\xi,\gamma,\mathbf{q},\mathcal{W}} \quad \frac{1}{2\gamma^2} + \frac{C}{\gamma}\sum_{k=1}^{m}\xi_k$$

$$\text{s.t.} \quad \frac{1}{n}\sum_{i=1}^{n} q_i \langle \hat{\mathbf{w}}_{T_i}, \hat{\phi}_{T_i}(\mathbf{x}_k, \mathbf{y}_k)\rangle - \max_{\mathbf{y}\in\mathcal{Y}} \frac{1}{n}\sum_{i=1}^{n} q_i \langle \hat{\mathbf{w}}_{T_i}, \hat{\phi}_{T_i}(\mathbf{x}_k, \mathbf{y})\rangle$$

$$\geq \gamma - \xi_k, \xi_k \geq 0, \forall k \in \{1, \cdots, m\}, \sum_{i=1}^{n} q_i = 1, q_i \geq 0, \forall i \in \{1, \cdots, n\}.$$

- This is equivalent to

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

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

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

Convex combination (cont.)

- This can be expressed equivalently as

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

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

$$\xi_k \geq 0,\ \forall k\in\{1,\ldots,m\},\ \sum_{i=1}^{n}\lambda_i = 1,\ \lambda_i\geq 0,\ \forall i\in\{1,\cdots,n\}.$$

Conclusions

- We show that if there is a learner $\mathbf{w}_G$ defined on a complete graph achieves a margin on some training data, then with a random collection of spanning tree learners $\{\mathbf{w}_{T_i}\}_{i=1}^n$ we can achieve a similar margin with high probability. Besides, n is polynomial in k.
- We propose two methods to combine the random collection of trees, namely, convex combination and conical combination.

Bibliography

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

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

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

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

Bibliography (cont.)

Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001).

Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

In *Proceedings of the 8th International Conference on Machine Learning (ICML 2001)*, pages 282–289. Morgan Kaufmann Publishers Inc.

Marchand, M., Su, H., Morvant, E., Rousu, J., and Shawe-Taylor, J. (2014).

Multilabel structured output learning with random spanning trees of max-margin markov networks.

In *Advances in Neural Information Processing System NIPS2014*, page to appear.

Read, J., Pfahringer, B., Holmes, G., and Frank, E. (2009).

Classifier chains for multi-label classification.

In Buntine, W., Grobelnik, M., Mladenić, D., and Shawe-Taylor, J., editors, *Machine Learning and Knowledge Discovery in Databases*, volume 5782, pages 254–269. Springer Berlin Heidelberg.

Bibliography (cont.)

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

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

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

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

Bibliography (cont.)

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

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

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

Bibliography (cont.)

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

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

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

Bibliography (cont.)

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

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

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

Bibliography (cont.)

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

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