TOPRANKING : PREDICTING THE MOST RELEVANT ELEMENT OF A SET

Kristiaan Pelckmans , Johan A.K. Suykens

kristiaan.pelckmans@esat.kuleuven.be
SCD/sista - ESAT - KULeuven,
Kasteelpark Arenberg 10, Leuven (Heverlee), Belgium

ABSTRACT

This short paper concerns the task of identifying the element of a set which is probably the most useful, based on previous *incomplete* experiments on similar tasks. It is shown that this problem can be solved effectively using a quadratic program, while a probabilistic guarantee is given that such a prediction will solve the problem on the average. We comment on the relation and difference of this setting with amongst others the structured output learning model, transductive inference and the multi-task learning setting. Finally, a number of immediate applications are described.

Index Terms— Machine Learning, Ranking, PAC bound.

1. INTRODUCTION

Many problems in cognitive information processing can be reduced to the problem of predicting which element in a given set will be most relevant. For example, in statistical decision theory, the aim is to come up with an optimal action to cope with a given situation. An *intelligent* agent in this setting would provide the most relevant action in a yet unseen situation. We restrict attention to the case where only a finite number of such actions (elements) exist, as occurring in a context of discrete control where a plant can only choose amongst a limited number of control actions. The notion of generalization - or prediction - is tackled from a context of machine learning where one tries to come up with a good predictor based on past observations. This paper focusses on a setting where such past observations consist of partial information only. More specifically, one has access only to the most relevant element of a subset of all possible actions.

The following probabilistic model is adopted. Consider a set S consisting of m different elements, representing objects as e.g. documents, antennas or products. Let L tuples $T_L = \{(S_l, r_l^S, x_l)\}_{l=1}^L$ be observed, where S_l consists of n_l uniformly sampled elements of S, $r_l^S \in S_l$ is the index of the element in S_l which is found (observed) to be most relevant, and x_l denotes the remaining information available (taken from an appropriate domain X). We will use the index $r_l \in S_l$ and $r_l^S \in S_l$ interchangeably, to emphasize that r_l is the best one in S_l. To make life easier, we let $n_1 = \cdots = n_L = n$. We refer to such a pair as to a **task** denoted as $t_l = (S_l, r_l, x_l)$ in spirit of the work [1]. Now the observed tasks $T_L$are assumed to be sampled i.i.d. from a universe of possible tasks with a certain probability rule $Pr(\cdot)$, as are the task which will be encountered in the future.

The above question was already explored in various machine learning settings. We will comment on the relations and (subtle) differences.

1. **Structured Output Learning.** The framework resembles closely the setting studied in structured output learning. Specifically, the 'argmax' formulation, and the resulting optimization formulation will resemble closely the one studied in that context, see e.g. [2] and the recently edited book [3]. This setting improves on the structured output learning setting in that we assume each task gives only partial information about the most relevant element.

2. **Transductive Inference** Since S is a finite

This work was conducted while visiting CSML, UCL, London with professor J. Shawe-Taylor, supported by a FWO Visiting Grant. K. Pelckmans is supported by an FWO PDM

J.A.K. Suykens is a professor at the Katholieke Universiteit Leuven, Belgium. Research supported by Research Council KUL: GOA AMBioRICS, CoE EF/05/006 OPTEC, IOF-SCORES4CHEM, several PhD/postdoc & fellow grants; Flemish Government: FWO: PhD/postdoc grants, projects G.0452.04, G.0499.04, G.0211.05, G.0226.06, G.0321.06, G.0302.07, (ICCoS, ANMMM, MLDM); IWT: PhD Grants, McKnow-E, Eureka-Flite+ Belgian Federal Science Policy Office: IUAP P6/04, EU: ERNSI;

set, the setting hints towards the transductive learning setting, where one restricts attention to predicting the values of a finite number of predefined points. As such, there is no need for building a general predictive model which can be evaluated on any new point. If we would restrict attention to only one single task, this framework would be appropriate. Now, we only retain the use of the device of hypergeometric distributions which plays a crucial role in the transductive setting, see e.g. [4, 5, 6].

3. **Selective Inference.** Given a finite set S, selective inference amounts to selecting an unlabeled point which belongs most certainly to the (true) positively (negatively) labeled set of points. This learning scheme was conjectured in [7] as being *easier* to learn than either inductive or transductive learning schemes. recently, in [8] one considers the task of finding the best instances based on a bipartite ranking. This setting is reminiscent of the adopted setting here, except for the assumption of observed binary labels.

4. **Missing Values.** In the analysis of missing values, one considers the case where in each sample some covariates are missing [9]. Specifically, the above setting can be viewed from this respective with the assumption of Missing Completely At Random (MCAR). Our setting deviates in that the considered universe has much more structure than in the typical cases, and in that no parametric assumptions are imposed on the involved probability laws.

5. **Multi-task Learning.** Our setup is directly related to the context of multitask learning where one tries to exploit the fact that many learning tasks in a certain context are related. This notion of relation is then used as a regularization mechanism to fill in the details of a learning task when one has not enough observations for this case at its disposal. A second objective is to find a model which generalizes well towards new sample tasks, see e.g. [1] and followup work. The difference in our case is that one observes only a partial piece of information in each task. and that we only try to come up with the most relevant element instead of learning the full labeling of all elements. The terminology in the multitask setup was used in our context.

A first application can be found in the context of *recommender systems*. Here, one has access the most relevant item bought by a customer, wile it is in general unlikely that the decision of the customer was preceeded by a study of the full catalogue. The task of the recommender system is to come up with a prediction of the globally most relevant product for the customer, indicating the applicability of the described learning model. Specifically, each user is modelled as an individual task, where at each instance, it is up to the learning system to predict the next purchase will be. The rationale is that it is most relevant for an advertiser or recommender system to predict the interest of the user in order to provide the most relevant information on products. Here, we are ignoring the fact that the set of items a customer did consider is probably not an independent sample of products. Extensions where the set S_l are non-uniform will be considered in future work.

Secondly, in *query-relevance learning* - or learning to rank answers to queries on a database - one is typically only interested in the top-ranked results. For example, in a search on the WWW, a user does typically only consider the first relevant 'hits' of the search query. In our framework, a task t_l would correspond to a query, and the bag of all results returned by the search-engine based on matching criteria. An application study towards this goal was described in [13]. Only in recent literature, it became apparent that one gets more efficient learning schemes when attention is restricted to the top-ranked results. This notion is often formalized in terms of the Discounted Cumulative Gain (DCG) measures and others as e.g. in [14]. The present work pushes this reasoning even further by only considering the topranked item.

The following notational convention is used throughout the paper. Scalars and vector are denoted using lower case letters, capitals denote matrices. The letters i, j, k, l are reserved for indices. The vector u_i of appropriate dimension is the unit coordinate vector consisting of zeros and having 1 at the i-th entry. This paper studies a practical algorithm based on a SVM (Section 2), derives PAC bounds using an elementary reduction argument (Section 3), and discusses the results of a preliminary experiment (Section 4).

2. MAXIMAL MARGIN MACHINE FOR TOPRANKING

A practical learning scheme is derived based on the Support Vector Machine (SVM). Let for each element i in the l-th task $\varphi_{t_l,i} \in \mathbb{R}^{d_\varphi}$ denote a feature vector capturing all relevant information which is known for this element - one has e.g. $\varphi_{t_l,i} = \phi(x_l)$. This includes for example external properties of the object represented by this element, or the set of closely related elements in S_l. Then given a global function f for task t_*, one would predict the most relevant element in S_* as

$$r(S_*) = \arg\max_i f\left(\varphi_{t_*,i}\right). \tag{1}$$

We consider the hypotheses f that are linear in the feature vector, or $\mathcal{H} = \{f(\varphi) = w^T\varphi\}$. At first, we start with the case where such a function can be assumed to capture the topranking exactly (the *realizable case*).

$$\min_w \frac{1}{2}w^T w \quad \text{s.t.}$$
$$\begin{cases} w^T\varphi_{S_1,r_1} - w^T\varphi_{t_1,i} \geq 1 & \forall i \in S_l \backslash r_1 \\ \vdots \\ w^T\varphi_{S_L,r_L} - w^T\varphi_{t_L,i} \geq 1 & \forall \in S_L \backslash r_L. \end{cases} \tag{2}$$

Remark that there is no need for an intercept term here. Let N be defined as $N = Ln$, then there are exactly $N - L$ constraints in this problem. Let $\Phi \in \mathbb{R}^{N \times d_\varphi}$ denote the matrix containing all possible values of $\varphi_{t_l,i}$ for all $l = 1, \ldots, L$ and $i = 1, \ldots, n$, such that $\Phi_{(l-1)n+i} = \varphi_{t_l,i}$. Then we can write the learning problem (2) shortly as

$$\min_w \frac{1}{2}w^T w \quad \text{s.t.} \quad D(\Phi w) \geq 1_{N-L}, \tag{3}$$

with the matrix $D \in \{-1, 0, 1\}^{(N-L)\times N}$ defined as $D = [D^1; \ldots; D^L]$ and $D^l_i = u_{r_l} - u_{S_l(i+1)}$ for all $i = 1, \ldots, n-1$. This problem can be solved efficiently as a convex Quadratic Programming (QP) problem. The dual expression becomes

$$\min_{\alpha \geq 0_{N-L}} \frac{1}{2}\alpha^T(DKD^T)\alpha - \alpha^T 1_{N-L}, \tag{4}$$

where the kernel matrix $K \in \mathbb{R}^{N\times N}$ contains the kernel evaluations such that $K_{n(l-1)+i,n(h-1)+j} = \varphi^T_{t_l,i}\varphi_{t_h,j}$ for all $l, h = 1, \ldots, L$ and $i, j = 1, \ldots, n$. The prediction in t_* can be done by evaluating

$$\hat{r}(S_*) = \arg\max_i K_{t_*,i} D^T \hat{\alpha}, \tag{5}$$

where $\hat{\alpha}$ solves (4), $K_{t_*,i} \in \mathbb{R}^{1\times N}$ and $K_{t_*,i;n(l-1)+j} = \varphi^T_{t_*,i}\varphi_{t_l,j}$ for all $l = 1, \ldots, L$ and $j = 1, \ldots, n$. It is interesting to note that the design of the matrix decides on the comparisons which have to be (can be) made. If the elements r_l and j cannot be ranked disambiguously, one may omit the corresponding entry in D. Alternatively, if one believes extra ordering constraints have to be incorporated, this can be done via proper choice of D. The agnostic case deals with the case where one is not prepared to make the assumption that a function exist which will extract in all cases the most relevant element. Using slack variables as in soft margin SVMs, one can formalize the learning objective for a fixed value of $\gamma > 0$ as follows.

$$\min_{w,e} \frac{1}{2}w^T w + \gamma \sum_{l=1}^{L} e_l \quad \text{s.t.}$$
$$\begin{cases} w^T\left(\varphi_{S_1,r_1} - \varphi_{S_1,i}\right) \geq 1 - e_1 & \forall i \neq r_1 \\ \vdots \\ w^T\left(\varphi_{S_L,r_L} - \varphi_{S_L,i}\right) \geq 1 - e_L & \forall i \neq r_L \\ e_l \geq 0 & \forall l = 1, \ldots, L. \end{cases} \tag{6}$$

and the dual problems becomes

$$\min_\alpha \frac{1}{2}\alpha^T(DKD^T)\alpha - \alpha^T 1_{N-L}$$
$$\text{s.t.} \quad 0_{N-L} \leq \alpha \leq \gamma 1_{N-L}. \tag{7}$$

We sidestep the issue of how to choose the hyperparameters γ and the choice of the kernel (parameters) althoug of great concern in practice.

3. PROBABILISTIC GUARANTEES

The PAC-Bayesian framework is adopted to provide a probabilistic guarantee that this mechanism indeed fulfills the objective on the average. This is somewhat surprising in that each 'sample task' t_l is never required to reveal his *globally* most relevant index, only the index of the most relevant entry in the *set of observations* S_l. The rationale is that the full ranking function emerges through the few orderings which can be extracted of the sample tasks. To make precise statements, the following notion of *actual risk* $R(f)$ of a specific function f is used

$$R(f) = Pr\left(r_* \neq \arg\max_{i\in S} f(\varphi_{t_*,i})\right), \tag{8}$$

where the probability concerns the choice of the task $t_* = (S, r_*, X_t)$. This quantity is equivalent to $Pr\left(\exists i \neq r_* : \ f(\varphi_{t_*,i}) > f(\varphi_{t_*,r_*})\right)$, or the probability that one can find an element besides r_* which is deemed more relevant by f. Furthermore, we will need the notion of risk restricted to sets $|S_*| = n$, formally

$$R^n(f) = Pr\left(r_*^{S_*} \neq \arg\max_{i \in S_*} f(\varphi_{t_*,i})\right), \quad (9)$$

where the probability concerns the choice of the task t_* on the one hand, and the uniform sampled subset S_* with $|S_*| = n$ on the other. Finally, the empirical counterpart to $R^n(f)$ becomes

$$R_L^n(f) = \frac{1}{L}\sum_{l=1}^{L} I\left(r_l^{S_l} \neq \arg\max_{i \in S_l} f(\varphi_{t_l,i})\right), \quad (10)$$

with the indicator $I(z)$ equal to one if statement z holds, and zero otherwise. For later convenience, let the term $I\left(r_l^{S_l} \neq \arg\max_{i \in S_l} f(\varphi_{t_l,i})\right)$ be denoted as the random variable $Z(f;T_l) \in \{0,1\}$. The generalization analysis will approach the question how much $R(f)$ deviates from $R_L^n(f)$ for functions in $f \in \mathcal{H}$. The following reduction provides the crucial means for the analysis.

Lemma 1 (Reduction of $R^n(f)$ to $R(f)$) *For a $\delta > 0$, one has with probaility exceeding $1 - \delta$ that for any $f \in \mathcal{H}$*

$$R(f) \leq n_S R^n(f) \quad (11)$$

where $n_S \in \mathbb{N}$ is defined as

$$n_S \geq \ln\left(\frac{(m-1)^2}{m^2 - n^2}\right) - \ln(\delta). \quad (12)$$

Proof: Assume at first that an index $j \in S$ exists such that $j \neq r$ and $f(\varphi_{t_*,j}) > f(\varphi_{t_*,r_*})$. The probability that the comparison between elements (j,r) does occur in a set S_1 sampled from S equals $\frac{n(n-1)}{m(m-1)}$, or the probability that an random subset contains 2 of the 2 relevant elements j and r. This follows from an application of the hypergeometric distribution.

$$\begin{aligned} &Pr\left((j,r) \notin \{S_l\}_l\right) \\ &\quad \leq \left(1 - Pr\left((j,r) \in S_1\right)\right)^{n_S} \\ &= \left(1 - \frac{n(n-1)}{m(m-1)}\right)^{n_S} \leq \left(\frac{m^2 - n^2}{(m-1)^2}\right)^{n_S}. \end{aligned} \quad (13)$$

Suppose one like to guarantee the inequality to a level $1 - \delta$, then

$$\left(\frac{m^2 - n^2}{(m-1)^2}\right)^{n_S} \leq \delta \Leftrightarrow n_S \geq \ln\left(\frac{(m-1)^2}{m^2 - n^2}\right) - \ln(\delta). \quad (14)$$

In conclusion, assume the learning scheme errs with probability $R^n(f)$ on a random subset S_*. Following Bonferroni (or the union bound) guarantees that the probability of erring at n_S such cases is at most $n_S R^n(f)$, proving the above statement.

□

Remark that this inequality is in general not tight as for a specific sample one can exploit transitivity properties (i.e. if $f_i \geq f_j$ and $f_j \geq f_k$, then $f_i \geq f_k$). Now, a similar argument can be used to give a guarantee on recovering the full ranking of all m items given $R^n(f)$. We however restrict attention to the case where $n = 2$ in order to guarantee that every pair of elements is ranked according to f with high probability.

Corollary 1 (Recovering the Full Ranking) *Fix $n = 2$ and $\delta > 0$. Let the actual ordering of the elements of S in task T_* be reflected as the set of couples $\pi_* = \{(i,j) : u(\varphi_{t_*,i}) \geq u(\varphi_{t_*,j})\}$ with u reflecting the actual ordering. Then with probability exceeding $1 - \delta$*

$$Pr\Big(\exists (i,j) \in \pi_* : f(\varphi_{t_*,i}) < f(\varphi_{t_*,j})\Big) \leq n'_S R^n(f), \quad (15)$$

where n'_S is defined as

$$n'_S \geq \ln\left(\frac{m(m-1)}{m(m-1) - 2}\right) - \ln(\delta). \quad (16)$$

Proof: Given a learning scheme which guarantees a risk $R^n(f)$, or which is expected as such to recover the best element of a subset $S_* \subset S$ with size $|S_*| = 2$. The question now reads how many such sets one would need to deduce the full ranking. Assume two different indices $(i,j) \in \pi_*$ exist such that $f(\varphi_{t_*,i}) > f(\varphi_{t_*,j})$. To deduce such a ranking from the toplearning scheme, element i should occur as the best element in a set S_1. As above, the hypergeometric distribution describes what the probability would be of sampling the set $S_* = \{i,j\}$, or $Pr(S_* = \{i,j\}) = \frac{2}{m(m-1)}$.

$$Pr\left((i,j) \notin \{S_l\}_l\right) \leq \left(\frac{m(m-1) - 2}{m(m-1)}\right)^{n_S}. \quad (17)$$

and inverting the statement again proves the result.

□

Remark that one runs into problems when $n > 2$, as one could never recover the ranking between the two lowest ranking entries in this way. Consequently, the above result is in a sense the best one could do. This is in direct contrast with the topranking case which improves if m grows. This bound again (even more so) ignores the transitivity properties, and it may be clear that incorporating this property should yields a more tight bound.

Given those results, we can proceed deriving the PAC guarantees as desired. First we show that the topranking function can be learned with fast rate if each task reveals its most relevant element, or equivalently $n_1 = \cdots = n_L = m$. This result follows completely the lines set out in the case of the zero/one loss, using each task as a full sample.

Proposition 1 (A PAC bound for $S_l = S$) *Given L tasks $\{T_l = (S_l, r_l, x_l)\}_{l=1}^L$ with $S_l = S$. Consider a class of hypothesis $\mathcal{H}$ with finite cardinality, i.e. $|\mathcal{H}| < \infty$, and say that one has always an $f \in \mathcal{H}$ with $R_L^n(f) = 0$. Then with probability exceeding $1 - \delta$, the inequality $R^n(f) \leq \epsilon$ is satisfied if the number of sample tasks L exceeds*

$$L \geq \frac{\ln(|\mathcal{H}|) - \ln(\delta)}{\epsilon} \tag{18}$$

The proof follows completely along the lines described in e.g. [7, 10, 11, 5]: the number (probability mass) of events where the hypothesis fails to reproduce the most relevant element cannot be too big. Indeed otherwise, such an event would turn up almost inevitably among the L samples with probability $1 - (1 - \epsilon)^L \geq 1 - \exp(-L\epsilon)$. The extension to the infinite case where $|\mathcal{H}| = \infty$ can be done using the device of VC dimensions as in one of the previous reference works. Lemma 1 results immediately results into the following generalization bound, much in the same vain as Corollary 1.

Lemma 2 (A PAC bound for $|S_l| = n < m$) *Given L tasks $\{T_l = (S_l, r_l, x_l)\}_{l=1}^L$ with $S_l = S$. Consider a class of hypothesis $\mathcal{H}$ with finite cardinality, i.e. $|\mathcal{H}| < \infty$, and say that one has always an $f \in \mathcal{H}$ with $R_L^n(f) = 0$. Then with probability exceeding $1 - \delta$, the difference $R(f) \leq \epsilon$ if the number of sample tasks L exceeds*

$$L \geq \frac{(\ln(|\mathcal{H}|) - \ln(\delta/2)) \ln(\delta/2)}{\epsilon \ln\left(1 - \frac{n}{m}\right)} \tag{19}$$

Proof: The proof consists of two main steps. At first, note that one has that for $f \in \mathcal{H}$ achieving zero empirical risk $R_L^n(f) = 0$ (such a function exists always due to the realizable assumption) that for any given $\epsilon > 0$, one has

$$Pr\left(\forall f \in \mathcal{H} : R^n(f) \geq \epsilon\right) \leq |\mathcal{H}| \exp(-L\epsilon) \tag{20}$$

using $Z_l(f) = I\left(r_i^S = \arg\min_{i \in S_l} f(\varphi_{t_l,i})\right) \in \{0, 1\}$ as the events of interest. In the second step, the relation between $R^n(f)$ and $R(f)$ is established. It will be argued that the guarantee on $R^n(f)$ has to hold uniformly over a (small) number of samples over $S_* \subset S$ with $|S_*| = n$. Indeed, if the performance is guaranteed for enough sets, the function f will prefer r over any other i at least once. A classical union bound argument for enforcing this, gives

$$Pr\left(\forall f \in \mathcal{H} : R^n(f) \geq \frac{\epsilon}{n_S}\right) \leq |\mathcal{H}| \exp\left(-L\frac{\epsilon}{n_S}\right), \tag{21}$$

or $\epsilon < \frac{\ln(|\mathcal{H}|) - \ln(\delta/2)}{n_S^{-1} L}$ as desired.

□

If $n = m$, the bound reduces to the statement of Proposition 2. If $n < m$, one needs slightly more samples L to learn effectively, governed by the fraction $(1 - n/m)$. This one does not need for n growing to m to have convergence, unlike one could expect. We now focus attention to infinite function sets using the practical device of Rademacher variables. Let the relevant Rademacher complexity expression be defined as

$$\mathcal{R}_L(\mathcal{H}) = E\left[\sup_{f \in \mathcal{H}} \frac{1}{L} \sum_{l=1}^{L} \sigma_l Z_l(f) \,\Big|\, T_1, \ldots, T_n\right] \tag{22}$$

This measure characterizes how flexible the hypothesis set is to either reconstruct or err in the topranking task as controlled by a random guidelines. It gives a datadependent measure of richness of the hypothesis set $\mathcal{H}$. Alternatively, one could think of this quantity as measuring how likely one is to *overfit* on the data. Many structural results and derivations of this quantity for different learning schemes were described in [12] and citations. This measure can then be used to characterize the generalization error, completely along the lines of the Rademacher results for the binary classification case.This result however slightly differentiates with the classical result described in [12] by considering the terms $R(f)$, $R^n(f)$ as well as $R_L^n(f)$.

Corollary 2 (Rademacher bound) *With probability exceeding $1 - \delta$ for fixed $\delta > 0$, one has for all $f \in \mathcal{H}$ that*

$$R(f) \leq n_S \left(R_L^n(f) + \mathcal{R}^L(\mathcal{H}) + \sqrt{\frac{\ln(1/\delta)}{L}}\right).$$

The proof again follows from Lemma 1 stating that $R(f) \leq n_S R^n(f)$. Application on the standard Rademacher bound gives the result.

4. ILLUSTRATION

We conduct a Monte-Carlo experiment to illustrate the practical use of the method. The following setup was adopted. An artificial warehouse with $m = 100$ products is generated. Each customer was presented with n products, where in three different experiments $n = 2, 10, 50$. Furthermore, each customer is characterized with $d = 10$ different features - in this artificial case sampled from a distribution (which consists of a sum of Gaussians). The dataset is constructed that a function exists which puts the actual most relevant element on top - i.e. there is a $f \in \mathcal{H}$ such that $R(f) = 0$. A model was learned as described in (2), and the performance of the learned model $\hat{w}$ was assessed by trying to predict the most relevant product for 10000 new customers. Figure 1 indicates the performances as a function of the number of observed customers and of the size n of the given subsets.

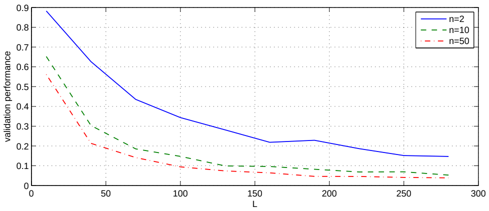

Fig. 1. Performances of the artificial recommender system for increasing number of observed tasks L, and for different sizes n of the considered subset m. Remark that the difference in performance in terms of m is multiplicative - giving evidence for the reduction argument.

5. DISCUSSION

This short work discussed the task of predicting the best element in a set, termed 'topranking'. Direct relationships with standard learning schemes as structured output learning, transductive and selective inference and multiple tasks learning were discussed, and a straightforward modification of the SVM for this setting was derived. The main contribution of this work is a simple reduction argument, indicating how one can cope with partial observations in order to learn a global scheme. Different application settings are described indicating the usefulness of the learning scheme. A imperative requirement is to conduct practical experiments to validate the proposed learning algorithm. indeed gives a formalization of what one would understand as 'cognitive intelligence' in a number of cases.

6. REFERENCES

[1] T. Evgeniou, C.A. Micchelli, and M. Pontil, "Learning Multiple Tasks with Kernel Methods," *Journal of Machine Learning Research*, vol. 6, pp. 615–637, 2005.

[2] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large Margin Methods for Structured and Interdependent Output Variables," *Journal of Machine Learning Research*, vol. 6, pp. 1453–1484, 2005.

[3] H. Bakir Editors Gökhan, T. Hofmann, B. Schölkopf, A. Smola, B Taskar, and S. V. N. Vishwanathan, *Predicting Structured Data (Neural Information Processing)*, MIT Press, 2007.

[4] A. Blum and S. Chawla, "Learning from labeled and unlabeled data using graph mincuts," in *Proceedings of ICML*, pp. 19–26. Morgan Kaufmann Publishers, 2001.

[5] A. Blum and J. Langford, "PAC-MDL bounds," in *Proceedings of COLT03*, 2003, pp. 344–357.

[6] K. Pelckmans, J. Shawe-Taylor, J.A.K. Suykens, and B. De Moor, "Margin based transductive graph cuts using linear programming," in *Proceedings of the AISTATS, pp. 360-367*, San Juan, Puerto Rico, 2007.

[7] V. Vapnik, *Estimation of Dependences Based on Empirical Data*, Springer, 2006.

[8] S. Clémenon and Vayatis, "Ranking the best instances," *Journal of Machine Learning Research*, vol. 8, pp. 2671–2699, 2007.

[9] D.B. Rubin, "Inference and missing data (with discussion)," *Biometrika*, vol. 63, pp. 581–592, 1976.

[10] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*, Springer-Verlag, 1996.

[11] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*, Cambridge University Press, 2004.

[12] P.L. Bartlett and S. Mendelson, "Rademacher and gaussian complexities: Risk bounds and structural results," *Journal of Machine Learning Research*, vol. 3, pp. 463–482, 2002.

[13] Y. Freund, R. Iyer, R.E. Schapire, and Y. Singer, "An Efficient Boosting Algorithm for Combining Preferences," *Journal of Machine Learning Research*, vol. 4, no. 6, pp. 933–969, 2004.

[14] D. Cossock and T. Zhang, "Subset ranking using regression," *Proceedings of COLT*, pp. 605–619, 2006.