Cost-based Ranking in Input Output Spaces

Ulf Brefeld

Max Planck Institute for Computer Science
Stuhlsatzenhausweg, 66123 Saarbrücken, Germany
brefeld@mpi-inf.mpg.de

Abstract. We study the problem of finding the most relevant candidates within a finite set of items under budget constraints. The choice of whether to bag an item not only depends on the actual sample but also on the associated costs and the remaining budget. We cast the problem of adapting a ranking function into the structural learning framework to capture the involved multiple-way dependencies. Key to our approach is the linearity of the rephrased task that can be solved optimally by the knapsack algorithm. Since inference is not tractable in general settings, we provide an ϵ-approximation that can be computed in polynomial time.

1 Introduction

Ranking approaches gain a lot of attention in recent years due to an omnipresence of search engines and recommender systems. In many applications, however, the goal is not to output a perfect ranking of items but to identify the maximal subset of items that fulfills certain additional constraints.

As an example consider software companies using the expiring time to the next deadline for final debugging. Since it is generally impossible to revise the whole program in only a few days, the companies face a combinatorial problem of deciding on which parts of their code they should have a second look at *and* on which parts they realistically can look at within the given time frame. Pure ranking approaches ignore the time constraint and only aim at finding the most buggy routines. However, this solution is inappropriate for the exemplary application since the time needed to debug the top-scoring routines might overrun the time left.

We propose a novel approach to the constraint-based identification of relevant items. To learn the ranking function, we cast the constraint-based task into the structured learning framework that allows for capturing the involved multiple-way dependencies. We devise a generalized linear model in joint input output space that can be optimized by structural support vector machines [3]. The derived model implements the knapsack criterion and is not tractable for large data sets. As a remedy, we propose a polynomial time approximation to the exact solution.

The remainder is organized as follows. The problem setting is introduced in Section 2. We then derive the constrained-based ranking method in Section 3 and Section 4 sketches the ϵ-approximation. Section 5 concludes.

2 Problem Setting

In this Section we abstract the task of finding the most relevant items under budget constraints. We decompose the task into a *decoding* and a parameter estimation step and present an appropriate loss function.

Given a training set $\{(\mathbf{x}^i, \mathbf{y}^i, \mathbf{c}^i, b^i)\}_{i=1}^n$, where $\mathbf{x}^i$ denotes a collection of items $\mathbf{x} = \{x_1^i, \ldots, x_{m_i}^i\}$ with associated costs $\mathbf{c}^i = (c_1^i, \ldots, c_{m_i}^i)^\mathsf{T} \in \mathbb{R}_+^{m_i}$, and b^i denotes the budget to spend. The corresponding output $\mathbf{y}^i$ is a subset of items contained in $\mathbf{x}^i$. We treat the output $\mathbf{y}^i = (y_1^i, \ldots, y_{m_i}^i)^\mathsf{T} \in \{0, 1\}^{m_i}$ as a binary vector, indicating whether items are contained in the solution, that is $y_j^i = 1$ if x_j^i is relevant and 0 otherwise. The true labeling is required to achieve the highest profit according to an (unknown) profit function p, that is,

$$\sum_{j=1}^{m_i} y_j^i \, p(x_j^i) \quad > \max_{\substack{\bar{\mathbf{y}} \neq \mathbf{y}^i \\ \sum \bar{y}_j c_j^i < b^i}} \sum_{j=1}^{m_i} \bar{y}_j \, p(x_j^i),$$

with $p(x) \geq 0$ for all x. Of course, having the true profit function renders the parameter estimation unnecessary. However, recall that we focus here on problems in which the true profit function is not accessible or unknown. We thus aim at finding a ranking function f that assigns higher decision scores to the true labeling than to all valid alternative labelings, that is,

$$\forall_{i=1}^n \forall_{\bar{\mathbf{y}} \neq \mathbf{y}^i \wedge \sum_j \bar{y}_j c_j^i < b^i} \quad f(\mathbf{x}^i, \mathbf{y}^i, \mathbf{c}^i, b^i) > f(\mathbf{x}^i, \bar{\mathbf{y}}, \mathbf{c}^i, b^i).$$

Similarly, at prediction time we are seeking the top-scoring hypothesis given by

$$\hat{\mathbf{y}} = \operatorname*{argmax}_{\bar{\mathbf{y}} : \sum_j \bar{y}_j c_j < b} f(\mathbf{x}, \bar{\mathbf{y}}, \mathbf{c}, b). \tag{1}$$

In the remainder we will refer to the computation of the argmax as *decoding* step which we will address in the next section.

We measure the quality of a hypothesis by a loss function Δ that details the differences between the true labeling and the prediction. For instance, an appropriate loss function in our discourse area is a Hamming-like loss that simply counts the number of errors in the prediction, that is, $\Delta_H(\mathbf{y}, \hat{\mathbf{y}}) = \sum_{j=1}^m 1_{[y_j \neq \hat{y}_j]}$, where $1_{[z]}$ equals 1 if z is true and 0 otherwise. In order to find a hypothesis that generalizes well on new and unseen data, we seek the minimizer of the regularized empirical risk

$$\hat{R}(f) = \sum_{i=1}^n \Delta_H(\mathbf{y}^i, \operatorname*{argmax}_{\bar{\mathbf{y}} : \sum_j \bar{y}_j c_j^i < b^i} f(\mathbf{x}^i, \bar{\mathbf{y}}, \mathbf{c}^i, b^i)) + \eta \|f\|^2.$$

3 Learning Knapsack

Given an input triple $(\mathbf{x}, \mathbf{c}, b)$, the goal is to maximize the profit without exceeding the budget. This can be expressed as the following constrained optimization problem, also known as the *knapsack* problem [4],

$$\max_{\mathbf{y}} \sum_{i=j}^{m} y_j \, p(x_j) \quad \text{s.t.} \quad \sum_{j=1}^{n} y_j \, c_j \leq b. \tag{2}$$

Since the true profits are unknown, we employ a $\boldsymbol{\lambda}$-parameterized profit function by linearly combining features $\psi(x)$ drawn from objects,

$$\hat{p}(x) = \langle \boldsymbol{\lambda}, \psi(x) \rangle. \tag{3}$$

Depending on the problem at hand, features may capture the size, weight, or color of item x. Substituting Equation (3) into (2) allows to rewrite the objective as a generalized linear model in joint input output space. We have

$$\begin{aligned} \sum\nolimits_j y_j \, \hat{p}(x_j) &= \sum\nolimits_j y_j \langle \boldsymbol{\lambda}, \psi(x_j) \rangle \\ &= \langle \boldsymbol{\lambda}, \underbrace{\sum y_j \psi(x_j)}_{=:\Phi(\mathbf{x},\mathbf{y})} \rangle, \end{aligned}$$

where $\Phi(\mathbf{x}, \mathbf{y})$ is sometimes called the joint feature mapping of inputs and outputs. Thus, the decision value of the model f is determined by

$$f(\mathbf{x}, \mathbf{y}, \mathbf{c}, b) = \begin{cases} \langle \boldsymbol{\lambda}, \Phi(\mathbf{x}, \mathbf{y}) \rangle & : \quad \sum y_j \, c_j \leq b \\ \text{not defined} & : \quad \text{otherwise.} \end{cases}$$

The distinction of valid and illegal assignments can be augmented with the computation of the argmax at prediction time, leading to an elegant model that can be optimized by structural SVMs [3]. Let $\mathcal{Y}^i = \{\mathbf{y} : \sum y_j c_j^i \leq b^i\}$, we obtain,

$$\hat{\mathbf{y}} = \underset{\bar{\mathbf{y}} \in \mathcal{Y}^i}{\operatorname{argmax}} \, f(\mathbf{x}, \bar{\mathbf{y}}, \mathbf{c}, b) = \underset{\bar{\mathbf{y}} \in \mathcal{Y}^i}{\operatorname{argmax}} \langle \boldsymbol{\lambda}, \Phi(\mathbf{x}, \mathbf{y}) \rangle. \tag{4}$$

If the decision values were discrete, Equation (4) could be computed by the knapsack algorithm in time $\mathcal{O}(m^2 \max_j \langle \boldsymbol{\lambda}, \psi(x_j) \rangle)$ [4]. The next Section proposes an approximation by inducing equivalence classes on the decision values.

Notice, that the definition of $\Phi(\mathbf{x}, \mathbf{y})$ allows to rewrite the inner product in joint input output space in terms of a kernel function $k(x, x') = \langle \psi(x_j), \psi(x'_j) \rangle$, defined solely on pairs of input items; we have $\langle \Phi(\mathbf{x}, \mathbf{y}), \Phi(\mathbf{x}', \mathbf{y}') \rangle = \sum_j y_j \, y'_j \, k(x_j, x'_j)$.

4 ϵ-approximate Inference

Due to non-discrete profit estimates $\hat{p}(x) = \langle \boldsymbol{\lambda}, \psi(x) \rangle$, the knapsack algorithm needs to enumerate all possible candidate sets which is prohibitive for real world applications and renders the optimization problem intractable. As a remedy, we induce equivalence classes by some rounding operations on the estimates $\hat{p}$

as follows. Let $\epsilon > 0$, we define a constant $\kappa = \frac{\epsilon \cdot \max_j \langle \boldsymbol{\lambda}, \psi(x_j) \rangle}{m}$ that induces equivalence classes by normalizing and rounding the decision values; we have,

$$\hat{p}(x_j) \leftarrow \left\lfloor \frac{\langle \boldsymbol{\lambda}, \psi(x_j) \rangle}{\kappa} \right\rfloor .$$

Utilizing these rounded profits instead of the decision values leads to a fully polynomial time approximation scheme for the knapsack problem that can be solved in $\mathcal{O}(m^2 \lfloor \frac{m}{\epsilon} \rfloor)$ where the error is bounded by the factor ϵ [4].

Due to the approximate inference, convergence of the SVM to the global optimum for the parameter estimation step cannot be guaranteed. However, empirical studies on the benefits of approximate over exact inference techniques show that the former frequently leads to more accurate prediction models in many domains [1, 2].

5 Conclusion

We devised a novel large margin approach in joint input output space for relevance ranking under budget constraints. The proposed solution allows to capture multiple-way dependencies within the data by adapting a parameterized profit function to a labeled training sample. Since the exact solution is intractable for large data sets we propose an approximation based on equivalence classes. Initial empirical results are promising and will be presented at the workshop. The proposed algorithm can be easily generalized to maintain several budgets and/or bags.

Acknowledgments

This work has been funded by the German Science Foundation DFG under grant SCHE540/10-2.

References

1. U. Brefeld, T. Klein, and T. Scheffer. Support vector machines for collective inference. In *Proceedings of the International Workshop on Mining and Learning with Graphs*, 2007.
2. T. Finley and T. Joachims. Parameter learning for loopy markov random fields with structural support vector machines. In *Proceedings of the ICML Workshop on Constraint Optimization and Structured Output Spaces*, 2007.
3. I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005.
4. V. V. Vazirani. *Approximation Algorithms*. Springer, 2001.