A Preference Model for Structured Supervised Learning Tasks

Fabio Aiolli
Dip. di Matematica Pura e Applicata, Università di Padova
Via G. Belzoni 7, 35131 Padova
aiolli@math.unipd.it

Abstract

The preference model introduced in this paper gives a natural framework and a principled solution for a broad class of supervised learning problems with structured predictions, such as predicting orders (label and instance ranking), and predicting rates (classification and ordinal regression). We show how all these problems can be cast as linear problems in an augmented space, and we propose an on-line method to efficiently solve them. Experiments on an ordinal regression task confirm the effectiveness of the approach.

1 Introduction

Many real-world learning problems are characterized by heterogeneous tasks which currently cannot be solved by general-purpose algorithms. These include *ordering* tasks (either label or instance ranking) where the required prediction concerns some order between labels/instances, and *rating* tasks where the required prediction consists of ratings (e.g. binary classification and ordinal regression). The typical approach followed to cope with these complex problems is to map them into a series of simpler, well-known settings and then to combine the resulting predictions. Often, these solutions lack a principled theory and/or require too much computational resources to be practical for data-mining applications. Although some efforts have been recently made to generalize label ranking tasks [5, 4, 2], a general framework and a theory encompassing all these supervised learning settings is missing. In this paper[1], we show that many types of supervision can be naturally seen as a set of preferences over the predictions of the learner and we show how they can be reduced to linear binary problems defined on an augmented space, thus suggesting very simple optimization procedures available for the binary case.

The main contribution of this paper is to define a flexible preference model which allows a practitioner to optimize the learning parameters on the basis of a proper evaluation function. In fact, while the goal of a problem in terms of its evaluation function is often clear, a crucial thing in the design of learning algorithms is how to define them in such a way to have some theoretical guarantee that a learning procedure leads to the effective minimization of that particular cost function. The model introduced here gives a natural and uniform way to encode the cost function of a supervised learning problem and plug it into a learning algorithm.

[1] An extended version of this paper can be found in [1].

2 A Model for Supervised Learning

For reasons that will be clear soon, we assume supervision to consist of (soft) constraints over the learner predictions, that is constraints whose violation entails a cost for the solution. Specifically, assuming a learner makes its predictions on the basis of a set of parameters Θ, supervision S causes the learner to suffer a cost $c(S|\Theta)$. Different settings are characterized by different types of prediction and supervision. Nevertheless, a broad set of them can be studied in the following framework.

We consider a space $\mathcal{X}$ of instances, a space $\mathcal{Y}$ of labels and an hypothesis space $\mathcal{H}$, based on which the learner makes its predictions, consisting of *relevance functions* $u : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$, depending on some set of parameters Θ. The goal of the learner is then to select a function $\hat{u} \in \mathcal{H}$ which is "consistent" with the supervision in a sense that will depend on the particular supervised setting.

In particular, we will consider hypotheses having form:

$$u(\mathbf{x}, y) = w \cdot \phi(\mathbf{x}, y) \tag{1}$$

where $\phi(\mathbf{x}, y) \in \mathbb{R}^d$ is a joint representation of instance-class pairs and $w \in \mathbb{R}^d$ is a weight vector. This form encompasses the more standard form $u(\mathbf{x}, y) = w_y \cdot \phi(\mathbf{x})$ which has a weight vector for each different label.

Common supervised learning tasks include ordering predictions and rating predictions (see [1] for details).

Ordering predictions are related to the ordering of classes (or instances) on a relevance basis in such a way

to be consistent with the supervision given as partial orders over the labels (or instances). These are referred to as *label ranking* (e.g. single-label multiclass classification and category ranking) and *instance ranking*, respectively.

Rating predictions aim at giving ranks from an ordinal scale to examples. The typical approach in this case is to associate the available ranks $\mathcal{Z} = \{0, \ldots, R-1\}$ to intervals of the real line by using an auxiliary set of thresholds $\tau = \{\tau_0 = -\infty, \tau_1, \ldots, \tau_{R-1}, \tau_R = +\infty\}$ and considering the rank z whenever $u(\mathbf{x}, y) \in (\tau_z, \tau_{z+1})$. Binary classification, ordinal regression, and their *multivariate* extensions belong to this family.

Supervision for all the settings above can be described as conjunctive sets of preferences of two types: *qualitative* preferences

$$(u(\mathbf{x}_i, y_r), u(\mathbf{x}_j, y_s))$$

telling that the value of $u(\mathbf{x}_i, y_r)$ should be higher than the value of $u(\mathbf{x}_j, y_s)$, and *quantitative* preferences

$$(u(\mathbf{x}, y), \tau) \text{ or } (\tau, u(\mathbf{x}, y)), \tau \in \mathbb{R}$$

relating the value of $u(\mathbf{x}, y)$ to a given threshold τ.

Preference sets (or p-sets) for the supervision of the general settings described above are given in Table 1. Instantiations to more specific problems are immediate anyway.

Ordering	$\{(u(\mathbf{x}, y_r), u(\mathbf{x}, y_s))\}_{(\mathbf{x}, y_r) \succeq_S (\mathbf{x}, y_s)}$
Rating	$\{(u(\mathbf{x}, y), \tau_i)\}_{i<z} \cup \{(\tau_i, u(\mathbf{x}, y))\}_{i \geq z}$

Table 1. Ordering predictions have a preference for each order relation induced by the supervision S. In rating predictions, a preference are related to thresholds and $z \in \mathcal{Z}$ is the rank given by the supervision.

Exploiting equation (1) it is possible to conveniently reformulate qualitative and quantitative preferences as linear constraints. Specifically, in the qualitative case, we can express a preference $a \equiv (u(\mathbf{x}_i, y_r), u(\mathbf{x}_j, y_s))$ as

$$w \cdot \underbrace{(\phi(\mathbf{x}_i, y_r) - \phi(\mathbf{x}_j, y_s))}_{\psi(a)} > 0.$$

Viceversa, in the quantitative case, given $\delta \in \{-1, +1\}$, the preference $\delta(u(\mathbf{x}, y) - \tau_r) > 0$ can be expressed as

$$(w, \tau_1, \ldots, \tau_{R-1}) \cdot \underbrace{(\delta\phi(\mathbf{x}, y), 0, \ldots, 0, -\delta, 0, \ldots, 0)}_{\psi(a)} > 0.$$

In general, supervision constraints of all the problems discussed above can be reduced into sets of particular linear constraints of the form $\mathbf{w} \cdot \psi(a) > 0$ where $\mathbf{w} = (w, \tau_1, \ldots, \tau_R)$ is the vector of weights augmented with the set of available thresholds and $\psi(a)$ is an opportune representation of the preference under consideration.

The quantity $\rho_A(a|\mathbf{w}) = \mathbf{w} \cdot \psi(a)$ will be also referred to as the margin of the hypothesis w.r.t. the preference. This value is greater than zero when the preference is *consistent* with the hypothesis (denoted $a \sqsubset \mathbf{w}$) and less than zero otherwise. The margin of an hypothesis w.r.t. the whole supervision S can also be defined as the minimum of the margins of preferences in the associated p-set, here denoted $g[S]$, i.e. $\rho(S) = \min_{a \in g[S]} \rho_A(a)$. This definition is consistent with definitions of the margin commonly used in different problems. In particular, the margin is positive if and only if the prediction is consistent with the supervision.

Summarizing, all the settings above can be seen as (homogeneous) linear problems in a opportune augmented space. Specifically, any algorithm for linear optimization (e.g. perceptron or a linear programming package) can be used to solve them, provided the problem has a solution.

3 GPLM and Cost Functions

The mere consistency of supervision constraints is not necessarily the ultimate goal of a supervised learning setting. Rather, cost functions are often preferred measuring the disagreement between the current hypothesis and the supervision. These functions may either depend on the particular structure of the prediction or other factors.

In [2] a general model for label rankings has been proposed. Here, we extend the same idea to general supervised settings by mapping supervision into sets of preferences with costs. We will refer to this method as *Generalized Preference Learning Model* (GPLM).

Definition 3.1 Preference Sets w/ Costs A (conjunctive) preference set with costs, *or simply "*cp-set*", is a preference set where preferences have costs associated. Preferences of a cp-set will be denoted by $a_{\gamma(a)}$. When the cost is not indicated $\gamma(a) = 1$ will be considered.*

Given a cp-set g, it is natural to define the cost for an hypothesis w.r.t. this set as the maximum of the costs of its unfulfilled preferences, i.e. $c(g|\mathbf{w}) = \max_{a \in g, a \not\sqsubset \mathbf{w}} \gamma(a)$. Then, similarly to [2], we consider a *cost mapping*:

$$\mathcal{G} : g[S] \mapsto \{g_1(S), \ldots, g_{q_S}(S)\}$$

where each cp-set $g_i(S)$ is a subset of $g[S]$ with some costs assigned to the preferences. Once the cost mapping $\mathcal{G}$ is given, the total cost suffered by an hypothesis $\mathbf{w}$ for the supervision S is defined as the cumulative cost of cp-sets

$$c(g[S]|\mathbf{w}) = \sum_{j=1}^{q_S} c(g_j(S)|\mathbf{w}).$$

Let g_p be a preference set, natural mappings already proposed in [4] for preference graphs are easily adapted by considering classes of equivalence among preferences and by defining mappings in which a different cp-set is built for each partition. Specifically, let $a \equiv (a_s, a_e)$ and $a' \equiv (a'_s, a'_e)$ be a pair of preferences, we have:

(i) the *identity mapping*, denoted by $\mathcal{G}_I$, where g_p is mapped on a single cp-set g_c. This corresponds to define the trivial equivalence relation $(a_s, a_e) \equiv (a'_s, a'_e)$;

(ii) the *domination mapping*, denoted by $\mathcal{G}_D$, where g_p is split into a set of cp-sets on the basis of the equivalence relation $(a_s, a_e) \equiv (a'_s, a'_e) \Leftrightarrow a_s = a'_s$;

(ii) the *dominated mapping*, denoted by $\mathcal{G}_{dom}$, where g_p is split into a set of cp-sets on the basis of the equivalence relation $(a_s, a_e) \equiv (a'_s, a'_e) \Leftrightarrow a_e = a'_e$;

(iv) the *disagreement mapping*, denoted by $\mathcal{G}_d$, where g_p is split into a set of cp-sets on the basis of equivalence relations $(a_s, a_e) \equiv (a'_s, a'_e) \Leftrightarrow a_s = a'_s \wedge a_e = a'_e$.

Now, we can show how many common cost functions can naturally be defined with the tools offered by our model.

Basic mappings for standard label ranking tasks can be found in [2] and can be reproduced with the model proposed in this paper. In fact, it can be shown quite easily that, for label rankings, PLM preference graphs and GPLM cp-sets with unitary costs are equivalent.

However, the extension presented here introduces far more flexibility on the choice of the cost function for label rankings because of the use of cp-sets in place of preference graphs. A typical example is classification where misclassifications can have different costs. This can be the case in single-label classification when categories are not represented with the same frequencies in the training and the test set. Another interesting case is when there is some structure between the available classes and a different metric for misclassification costs is introduced. For example, in hierarchical classification, it makes sense to pay costs proportional to the path length to reach the true class from the predicted one. In all these cases, a cost matrix Δ is used to have a better control over the learning algorithm, where the element $\Delta(y_r, y_s)$ represents the cost of classifying a pattern as y_r when it is actually in y_s. In GPLM, this can be easily obtained by exploiting the cost mapping feature.

Similar considerations can be made for instance ranking tasks. A common loss function used in instance ranking is the so called AUC (area under ROC curve) measure. It can be shown that it can be implemented by $\mathcal{G}_d$. Interestingly, our model suggests new settings and loss definitions one might use for the tasks in the family of instance rankings.

To illustrate standard loss functions used for rating tasks and the implementation in our model, we consider (univariate) ordinal regression problems as the classification setting is just a particular case. Multi-variate extensions are omitted for space reasons and can be found in [1].

Specifically, recalling the natural definition of cost for ordinal regression problems, i.e. $c = |\hat{z}(\mathbf{x}) - z(\mathbf{x})|$, where $\hat{z}(\mathbf{x})$ is the rank given as output by the hypothesis and $z(\mathbf{x})$ the correct rank, we would like to define a cost mapping for GPLM consistent with the same cost function. At least two different cost mappings have this property. The easiest one is the mapping $\mathcal{G}_d$. In this case, the resulting cost will be the number of thresholds which are not correctly ordered w.r.t. $u(\mathbf{x})$. This is exactly the cost as given before. A second possibility is to define a mapping $\mathcal{G}_I$ followed by an assignment of costs where the r-th preference is set to $(u(\mathbf{x}), \tau_r)_{z-i+r}$ whenever $r \leq z$, and $(\tau_r, u(\mathbf{x}))_{r-z}$ otherwise.

4 Learning in GPLM

We have already discussed the structure behind the supervision and how it can be modeled using cp-sets. Now, we see how to give a learning algorithm. We only consider the *on-line* learning setting, the motivation being that it is generally faster to train w.r.t. the *batch* counterpart and thus more suited for data-mining applications. Thus, we suppose supervision becomes available one by one and each time the learner updates the hypothesis to minimize future costs. Since the costs $c(S|\mathbf{w})$ are not continuous w.r.t. $\mathbf{w}$, we try to approximate them by introducing a continuous non-increasing function $l : \mathbb{R} \to \mathbb{R}^+$ approximating the indicator function and by defining the approximation

$$\tilde{c}(S|\mathbf{w}) = \sum_{g \in \mathcal{G}(g[S])} \max_{a \in g} \gamma(a) l(\rho_A(a|\mathbf{w})).$$

Examples of losses one can use are presented in Table 2 where $\beta > 0, \lambda > 0, \theta \in \mathbb{R}$ are external parameters.

β-margin	$[\beta - \rho]_+$	Exp	$e^{-\rho}$
Sigmoidal	$(1 + e^{\lambda(\rho-\theta)})^{-1}$	LogReg	$\log_2(1 + e^{-\rho})$

Table 2. Margin-based approximation losses.

In on-line learning, a suitable measure of performance after m rounds is the *cumulative cost* function

$$R_t^m[\mathbf{w}] = \sum_{i=1}^{m} c(g[S_i]|\mathbf{w}_i)$$

where $\mathbf{w}_i$ is the hypothesis obtained after seeing supervision $S_1, \ldots, S_{i-1}$.

Following a typical approach for on-line learning, we perform a stochastic gradient descent with respect to the instantaneous cost $\mathcal{Q}(\mathbf{w}_t) = \tilde{c}(S_t|\mathbf{w}_t)$. The update will be in the form $\mathbf{w}_t = \mathbf{w}_{t-1} - \eta \mathcal{Q}'_{\mathbf{w}}$, where

$$\mathcal{Q}'_{\mathbf{w}} = \textstyle\sum_{g \in \mathcal{G}(g[S_t])} \gamma(\hat{a}[g]) l'_\rho(\rho(\hat{a}[g])) \psi(\hat{a}[g]),$$

$\hat{a}[g] = \arg\max_{a \in g} \gamma(a) l(\rho(a))$ and $f'_x(v)$ stands for the gradient of f w.r.t. the parameters x evaluated in v.

It can be shown easily that these updates make the weight vector w taking the sparse form $w = \sum_{i,r} \alpha_i^r \phi(\mathbf{x}_i, y_r)$ thus obtaining an implicit representation of the relevance function as $u(\mathbf{x}, y) = \sum_{i,r} \alpha_i^r \phi(\mathbf{x}_i, y_r) \phi(\mathbf{x}, y)$.

5 Experiments and Results

To demonstrate the flexibility of the model proposed in this paper, we performed a set of experiments on a synthetic dataset. The explicit purpose was the one to try different cost mappings and loss functions in a relatively self-contained task in such a way to have a better control and to do fair comparisons between different configurations.

The experimental setting is the same used in [3]. Points $\mathbf{x} = (x_1, x_2)$ are uniformly distributed in the unit square $[0, 1]^2$. Ranks are assigned basing on the following rule:

$$r \in \{0, \ldots, 4\} : 10(x_1 - 0.5)(x_2 - 0.5) + \epsilon \in (b_r, b_{r+1})$$

where $b = \{b_0, \ldots, b_5\} = \{-\infty, -1, -0.1, 0.25, 1, +\infty\}$ and ϵ is a normally distributed noise $\epsilon \sim N(0, \sigma)$. We generated 100 sequences of 100,000 examples each. Moreover, a non-homogeneous second order polynomial kernel $K(\mathbf{x}_1, \mathbf{x}_2) = \Phi(\mathbf{x}_1) \cdot \Phi(\mathbf{x}_2) = (\mathbf{x}_1 \cdot \mathbf{x}_2 + 1)^2$ has been used. The performance on a sequence is obtained by feeding all the instances of the sequence and computing the *cumulative cost* at each iteration m as $c_m = \sum_{t=1}^{m} |\hat{r}_t - r_t|$. Finally, the obtained costs are averaged over the 100 sequences to obtain higher statistical significance.

Experiments have been performed using configurations produced according to three dimensions:

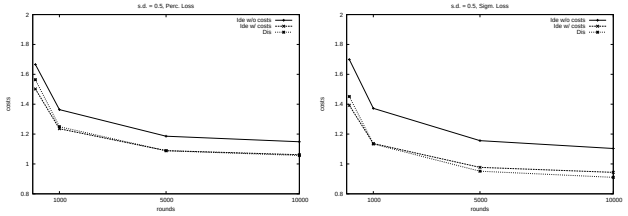
(i) Cost Mapping: Three cost mappings for ordinal regression have been used. Two of them are the ones presented in Section 3, i.e. the mapping $\mathcal{G}_I$ with costs (denoted $\mathcal{G}_I^c$) and the mapping $\mathcal{G}_d$. The last mapping is basically the mapping $\mathcal{G}_I$ where the cost assignment is not performed. Note that, this mapping represents the cost function which gives a unitary cost for incorrect predicted ranks.

(ii) Complexity of the task: Different values of the standard deviation $\sigma \in \{0, 0.125, 0.5, 1.0\}$ have been used. A greater σ leads to a more difficult task.

(iii) Preference Loss: Two losses from the ones in Table 2 were used, i.e. the Perceptron loss (β-margin with $\beta = 0$), and the Sigmoidal loss with parameter $\lambda = 1, \theta = -1$.

One may notice that the configuration $(\mathcal{G}_d, \cdot, \text{PLoss})$ is equivalent to the PRank algorithm proposed in [3].

In Fig. 1, the curves of cost obtained for the three mappings and $\sigma = 0.5$ are shown. Different plots refer to the two preference losses. In Table 3 a detail of results after 10000 presentations is shown. Results show that the baseline cost mapping $\mathcal{G}_I$ is consistently worse than the other two, while the performance of $\mathcal{G}_I^c$ and $\mathcal{G}_d$ ar quite similar. Interestingly, a far larger improvement is obtained for the sigmoidal loss and this can be due to the effective use of margins and/or a better approximation of the true cost.



(a) (b)

Figure 1. Curves of the cost (see text).

	—— Perc. Loss ——			—— Sigm. Loss ——		
σ	$\mathcal{G}_I$	$\mathcal{G}_I^c$	$\mathcal{G}_d$	$\mathcal{G}_I$	$\mathcal{G}_I^c$	$\mathcal{G}_d$
0.000	0.369	0.339	0.317	0.326	0.259	0.236
0.125	0.502	0.470	0.452	0.454	0.384	0.364
0.500	1.148	1.062	1.057	1.104	0.944	0.910
1.000	1.661	1.575	1.620	1.626	1.474	1.447

Table 3. Costs for different methods and task complexities.

6 Conclusion

We have proposed a general preference model for supervised learning and its application to on-line algorithms. The model allows to codify cost functions as preferences and naturally plug them into the same training shell. Furthermore, it gives a tool for comparing different methods and cost functions on a same learning problem. Experiments performed on an ordinal regression problem have confirmed the validity of the approach and highlighted the important role of the loss functions used for training.

References

[1] F. Aiolli. A unifying framework for supervised learning tasks. http://www.di.unipi.it/~ aiolli/slpref-long.pdf, 2005.

[2] F. Aiolli and A. Sperduti. Preference learning for multiclass problems. In *NIPS*, 2004.

[3] K. Crammer and Y. Singer. Pranking with ranking. In *NIPS*, 2001.

[4] O. Dekel, C. Manning, and Y. Singer. Log-linear models for label ranking. In *NIPS*, 2003.

[5] S. Har-Peled, D. Roth, and D. Zimak. Constraint classification for multiclass classification and ranking. In *NIPS*, 2002.