Superset Learning Based on Generalized Loss Minimization

Eyke Hüllermeier[1] and Weiwei Cheng[2]

[1] Department of Computer Science
University of Paderborn, Germany
[2] Amazon Inc.,
Berlin, Germany

Abstract. In standard supervised learning, each training instance is associated with an outcome from a corresponding output space (e.g., a class label in classification or a real number in regression). In the superset learning problem, the outcome is only characterized in terms of a superset—a subset of candidates that covers the true outcome but may also contain additional ones. Thus, superset learning can be seen as a specific type of weakly supervised learning, in which training examples are ambiguous. In this paper, we introduce a generic approach to superset learning, which is motivated by the idea of performing model identification and "data disambiguation" simultaneously. This idea is realized by means of a generalized risk minimization approach, using an extended loss function that compares precise predictions with set-valued observations. As an illustration, we instantiate our meta learning technique for the problem of label ranking, in which the output space consists of all permutations of a fixed set of items. The label ranking method thus obtained is compared to existing approaches tackling the same problem.

1 Introduction

Superset learning is a specific type of learning from weak supervision, in which the outcome (response) associated with a training instance is only characterized in terms of a subset of possible candidates. Thus, superset learning is somehow in-between supervised and semi-supervised learning, with the latter being a special case (in which supersets are singletons for the labeled examples and cover the entire output space for the unlabeled ones). There are numerous applications in which only partial information about outcomes is available [13].

Correspondingly, the superset learning problem has received increasing attention in recent years, and has been studied under various names, such as *learning from ambiguously labeled examples* or *learning from partial labels* [6,11,15,5]. The contributions so far also differ with regard to their assumptions on the incomplete information being provided. In this paper, we only assume the actual outcome to be covered by the subset—hence the name *superset* learning.

We introduce an approach to superset learning based on direct loss minimization with a suitably generalized loss function. While previous work on superset learning has mainly been focused on (multi-class) classification, our approach is completely generic and does not make any specific assumptions about the output space. In fact, we argue that superset learning is specifically interesting for complex, structured output prediction, because information about such outputs is indeed often incomplete. This is why, in the second part of the paper, we apply our approach to the problem of label ranking, where outputs take the form of rankings. More specifically, by instantiating our approach to superset learning for the case of label ranking, we develop a new method for this problem, which turns out to perform quite strongly in first experimental studies.

The rest of the paper is organized as follows. In the next section, we introduce the basic problem setting and the main notation to be used throughout the paper. Our new approach to superset learning is then introduced in Section 3.[3] In Sections 4 and 5, we recall the label ranking problem and introduce our new method.[4] The paper concludes with a summary and an outlook on future work in Section 6.

2 Setting and Notation

Consider a standard setting of supervised learning with an input (instance) space $\mathcal{X}$ and an output space $\mathcal{Y}$. The goal is to learn a mapping from $\mathcal{X}$ to $\mathcal{Y}$ that captures, in one way or the other, the dependence of outputs (responses) on inputs (predictors). The learning problem essentially consists of choosing an optimal model (hypothesis) M^* from a given model space (hypothesis space) $\mathbf{M}$, based on a set of training data

$$\mathcal{D} = \big\{ (\boldsymbol{x}_n, y_n) \big\}_{n=1}^{N} \in (\mathcal{X} \times \mathcal{Y})^N \ . \tag{1}$$

More specifically, optimality typically refers to optimal prediction accuracy, i.e., a model is sought whose expected prediction loss or *risk*

$$\mathcal{R}(M) = \int L\big(y, M(\boldsymbol{x})\big) \, d\,\mathbf{P}(\boldsymbol{x}, y) \tag{2}$$

is minimal; here, $L : \mathcal{Y} \times \mathcal{Y} \longrightarrow \mathbb{R}$ is a loss function, and $\mathbf{P}$ is an (unknown) probability measure on $\mathcal{X} \times \mathcal{Y}$ modeling the underlying data generating process.

In this paper, we are interested in the case where output values $y_n \in \mathcal{Y}$ are not necessarily observed precisely; instead, only a superset $Y_n \subseteq \mathcal{Y}$ is observed. Therefore, the learning algorithm does not have direct access to the (precise) data (1), but only to the (imprecise, ambiguous) observations

$$\mathcal{O} = \big\{ (\boldsymbol{x}_n, Y_n) \big\}_{n=1}^{N} \in (\mathcal{X} \times 2^{\mathcal{Y}})^N \ . \tag{3}$$

[3] This approach is leaned on [8], where a similar problem is studied in the context of learning from “fuzzy data”.

[4] A first version of this method has been presented at M-PREF 2013, 7th Multidisciplinary Workshop on Advances in Preference Handling, Beijing, China.

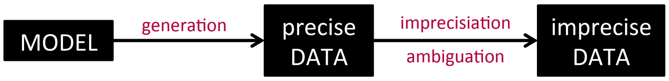

Fig. 1. Data generating process in the setting of superset learning.

More specifically, we assume a data generating process as sketched in Figure 1: Given an instance $\boldsymbol{x} \in \mathcal{X}$, an underlying process first generates a precise outcome $y \in \mathcal{Y}$, which is then turned into an imprecise observation in the form of a superset $Y \ni y$. We refer to this process of generating Y as "ambiguation" or "imprecisiation" of y.

In the following, we denote by $\mathbf{Y} = Y_1 \times Y_2 \times \cdots \times Y_N$ the (Cartesian) product of the supersets observed for $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$. Moreover, each $\boldsymbol{y} = (y_1, \ldots, y_N) \in \mathbf{Y}$ is called an *instantiation* of the imprecisely observed data. More generally, we call $\mathcal{D}$ in (1) an instantiation of $\mathcal{O}$ if the instances $\boldsymbol{x}_n$ coincide and $y_n \in Y_n$ for all $n \in [N] = \{1, \ldots, N\}$.

Prior to proceeding, let us emphasize that the Y_n are considered as constraints on *actual* outcomes y_n, not on any kind of *ideal* outcomes or predictions for the instance $\boldsymbol{x}_n$. In regression, for example, outcomes y_n could be random variables with expected value $\mu(\boldsymbol{x}_n)$ and standard deviation $\sigma(\boldsymbol{x}_n)$. What we assume, then, is $Y_n \ni y_n$ but not necessarily $Y_n \ni \mu(\boldsymbol{x}_n)$.

3 A Loss Minimization Approach

Given the data generating process as outlined above, the likelihood of a model $M \in \mathbf{M}$ can be defined by the probability of the data given the model, i.e.,

$$\ell(M) = \mathbf{P}\big(\mathcal{O}, \mathcal{D} \,|\, M\big) = \mathbf{P}(\mathcal{D} \,|\, M)\mathbf{P}(\mathcal{O} \,|\, \mathcal{D}, M) \ . \tag{4}$$

A reasonable assumption is that the imprecise observations Y_n only depend on the underlying true outcomes y_n but not on the model M or, in other words, that $\mathcal{O}$ is conditionally independent of M given $\mathcal{D}$. Under this assumption, $\mathbf{P}(\mathcal{O} \,|\, \mathcal{D}, M) = \mathbf{P}(\mathcal{O} \,|\, \mathcal{D})$ and (4) becomes

$$\ell(M) = \mathbf{P}\big(\mathcal{O}, \mathcal{D} \,|\, M\big) = \mathbf{P}(\mathcal{D} \,|\, M)\mathbf{P}(\mathcal{O} \,|\, \mathcal{D}) \ . \tag{5}$$

As can be seen, the likelihood of M under the superset data is a weighted average of standard likelihoods $\mathbf{P}(\mathcal{D} \,|\, M)$, with each precise data sample $\mathcal{D}$ being weighted by the probability $\mathbf{P}(\mathcal{O} \,|\, \mathcal{D})$ of observing $\mathcal{O}$ if the true underlying data were $\mathcal{D}$. In some cases, specific knowledge about these probabilities, i.e., about the process of imprecisiation, is available; for example, in a classification setting, a connection between true labels and observed partial labels is established in terms of a so-called mixing matrix in [17]. However, in lack of any specific knowledge of that kind, the most reasonable assumption we can make is

$$\mathbf{P}(Y \,|\, y) = \begin{cases} \text{const} & \text{if } Y \ni y \\ 0 & \text{if } Y \not\ni y \end{cases} \tag{6}$$

We call this the *superset assumption*, as it does not assume anything else than the observation Y being a superset of y; in fact, the uniform distribution (6) is the "weakest" distribution in accordance with this assumption, namely the one with the highest entropy among all distributions allocating the entire probability mass on supersets of y.

Now, it is easy to see that the likelihood (4) will vanish as soon as $y_n \notin Y_n$ for at least one of the observations, while $\mathbf{P}(\mathcal{O} \,|\, \mathcal{D})$ is a non-negative constant that does not depend on M if $y_n \in Y_n$ for all $n \in [N]$. Thus, maximizing the likelihood is equivalent to finding

$$M^* \in \operatorname*{argmax}_{M \in \mathbf{M}} \max_{\boldsymbol{y} \in \mathbf{Y}} \prod_{n=1}^{N} \mathbf{P}(y_n \,|\, M, \boldsymbol{x}_n) \tag{7}$$

or, equivalently,

$$M^* \in \operatorname*{argmin}_{M \in \mathbf{M}} \min_{\boldsymbol{y} \in \mathbf{Y}} \sum_{n=1}^{N} -\log \mathbf{P}(y_n \,|\, M, \boldsymbol{x}_n) \ . \tag{8}$$

3.1 Generalized Loss Minimization

Recall the principle of *empirical risk minimization* (ERM): A model M^* is sought that minimizes the *empirical risk*

$$\mathcal{R}_{emp}(M) \;=\; \frac{1}{N} \sum_{n=1}^{N} L\big(y_n, M(\boldsymbol{x}_n)\big) \ , \tag{9}$$

i.e., the average loss on the training data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$. The empirical risk (9) serves as a surrogate of the true risk (2). In order to avoid the problem of possibly *overfitting* the data, not (9) itself is typically minimized but a *regularized* version thereof. This is of minor importance here, however, and the approach outlined in the following can be generalized from standard ERM to regularized risk minimization in a straightforward way.

Now, coming back to our superset learning problem, it is interesting to note that the approach (8) can be seen as a special case of ERM, with the loss function $L(\cdot)$ given by the logistic loss: $L(y, \hat{y}) = L(y, M(\boldsymbol{x}))$ is the (negative) logarithm of the probability of y under the distribution specified by $M(\boldsymbol{x})$. For example, suppose that $\mathbf{M}$ is the class of linear regression models with normally distributed error term, i.e., $y = M_{\boldsymbol{w}}(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + \epsilon$. Then,

$$M^* \in \operatorname*{argmin}_{M_{\boldsymbol{w}} \in \mathbf{M}} \min_{\boldsymbol{y} \in \mathbf{Y}} \sum_{n=1}^{N} \big(y_n - \boldsymbol{w}^\top \boldsymbol{x}_n\big)^2 \ .$$

As can be seen, each candidate model M is evaluated optimistically according to

$$\overline{\mathcal{R}}_{emp}(M_{\boldsymbol{w}}) = \min_{\boldsymbol{y} \in \mathbf{Y}} \sum_{n=1}^{N} \big(y_n - \boldsymbol{w}^\top \boldsymbol{x}_n\big)^2 \ ,$$

i.e., the standard (squared) loss it makes on the instantiation $\boldsymbol{y}$ that is most favorable for M, and then the model M^* with the best optimistic evaluation is chosen.

Of course, the logistic loss could in principle be replaced by any other loss function $L(\cdot)$ of interest; this is in fact even a prerequisite for working with non-probabilistic models, i.e., if a model M merely produces predictions in $\mathcal{Y}$ but not complete probability distributions. A model M is then evaluated according to

$$\overline{\mathcal{R}}_{emp}(M) = \min_{\boldsymbol{y} \in \mathbf{Y}} \frac{1}{N} \sum_{n=1}^{N} L\big(y_n, M(\boldsymbol{x}_n)\big) \enspace .$$

Moreover, given a loss that is decomposable (over examples), the "optimism" can be moved into the loss:

$$\begin{aligned} \min_{\boldsymbol{y} \in \mathbf{Y}} \sum_{n=1}^{N} L\big(y_n, M(\boldsymbol{x}_n)\big) &= \sum_{n=1}^{N} \min_{y_n \in Y_n} L\big(y_n, M(\boldsymbol{x}_n)\big) \\ &= \sum_{n=1}^{N} L^*\big(y_n, M(\boldsymbol{x}_n)\big) \end{aligned}$$

with the generalized loss function

$$L^*(Y, \hat{y}) = \min \big\{ L(y, \hat{y}) \,|\, y \in Y \big\} \qquad (10)$$

that compares (precise) predictions with set-valued observations. We call this loss the *optimistic superset loss* (OSL). Note that this loss covers the *superset error* $[\![\hat{y} \not\in Y]\!]$, which is commonly used in superset label learning for classification [14], as a special case.

In summary, our approach to superset learning is based on the minimization of the empirical risk with respect to this generalized loss function. Thus, each candidate model $M \in \mathbf{M}$ is evaluated in terms of

$$\overline{\mathcal{R}}_{emp}(M) \;=\; \frac{1}{N} \sum_{n=1}^{N} L^*\big(Y_n, M(\boldsymbol{x}_n)\big) \enspace , \qquad (11)$$

and an optimal model M^* is one that minimizes (11) — or, as mentioned before, a regularized version thereof.

3.2 Data Disambiguation

In the context of learning from data, not only the data is providing information about the (unknown) model, but also the other way around. This view is made explicit in the Bayesian approach to data analysis, where the joint model/data probability $\mathbf{P}(M, \mathcal{D})$ can be written either way, as $\mathbf{P}(M)\mathbf{P}(\mathcal{D} \,|\, M)$ and $\mathbf{P}(\mathcal{D})\mathbf{P}(M \,|\, \mathcal{D})$. From a Bayesian perspective, the superset learning problem could be tackled quite naturally by not only starting with a prior on the model class $\mathbf{M}$ but also on the data, for example defining a uniform prior on each superset Y_n and zero probability outside. Inference would then come down to attuning these priors, e.g., by turning priors into posteriors on the model space and the data space in an alternating way. Eventually, this will yield a joint model/data (posterior) probability $\mathbf{P}(M, \mathcal{D})$ that will not only inform about a most plausible model M^* but also about a most plausible instantiation $\boldsymbol{y}^*$ of the imprecise data. In other words, it will help *disambiguating* the data.

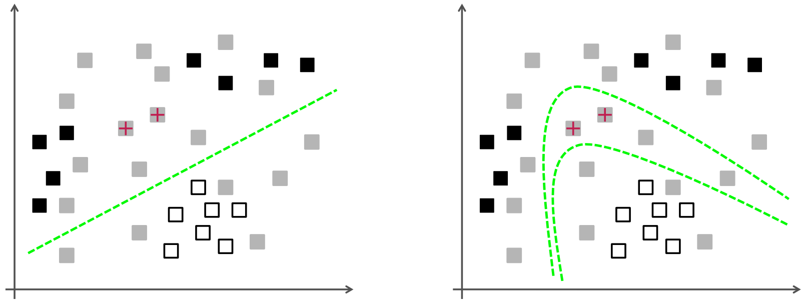

Fig. 2. Model identification and data disambiguation go hand in hand. Left: Assuming a linear model, the two example marked by a cross are most likely positive. Right: Fitting a nonlinear model, disambiguation of these examples is less obvious.

Our approach supports data disambiguation, too, albeit in a different way. As can be seen from the "double-max" operation in (7), model and data are selected in the most favorable combination. Thus, disambiguation essentially relies on the inductive bias implemented by the model class $\mathbf{M}$ [9]. In fact, against the background of the learning bias, some instantiations of the ambiguous data appear to be more plausible than others. This is illustrated in Figure 2 for a simple scenario of binary classification, in which some instances are known to be positive (marked in black, $y_n = +1$), some are known to be negative (white, $y_n = -1$), whereas some are unlabeled (grey, $Y_n = \{-1, +1\}$). Now, consider the two unlabeled instances marked with a cross, for example. Looking at each example in isolation, nothing can be said about the actual (precise) label. However, when looking at the data as a whole, in conjunction with the assumption of a linear decision boundary between the two classes, the positive class is clearly more plausible than the negative class (left picture). Yet, looking at the data with a slightly less biased view and also allowing for a nonlinear (e.g., quadratic) discriminant, these cases are more difficult to disambiguate: Both the positive and negative class appear to be plausible, since both can be obtained with plausible models $M \in \mathbf{M}$, i.e., models that are in agreement with the rest of the data. This example also shows that the stronger the bias, i.e., the more background knowledge is incorporated in the learning process, the easier disambiguation of the data becomes.

In our approach, the disambiguated outcome $\boldsymbol{y}^*$ corresponds to those elements for which the minimizer M^* of (11) attains its (generalized) risk, i.e.,

$$y_n^* = \operatorname*{argmin}_{y_n \in Y_n} L\big(y_n, M^*(\boldsymbol{x}_n)\big) \enspace .$$

3.3 Examples

It is interesting to note that several methods proposed in the literature can be seen as special cases of our framework, i.e., these methods correspond to the minimization of the generalized loss (11) following to a suitable imprecisiation of the data. For example, the ϵ-insensitive loss $L(y, \hat{y}) = \max(|y-\hat{y}|-\epsilon, 0)$ used in support vector regression [16] corresponds to the OSL (10) with L the standard L_1 loss $L(y, \hat{y}) = |y - \hat{y}|$ and precise data y_n being replaced by interval-valued data $Y_n = [y_n - \epsilon, y_n + \epsilon]$ (cf. Figure 3).

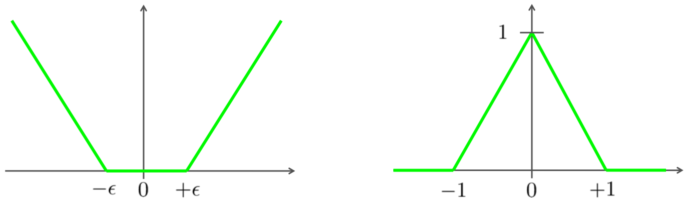

Fig. 3. The ϵ-insensitive loss (left) and the hat loss (right).

Perhaps more interestingly, we obtain semi-supervised learning with support vector machines as a special case by considering unlabeled data as instances labeled with the superset $\{-1, +1\}$ (like in our above example). The generalized loss (10), with L the standard hinge loss, then corresponds to the (non-convex) “hat loss” (cf. Figure 3). More generally, if the loss L is a margin loss of the form $L(y, s) = f(ys)$, comparing a class label $y \in \{-1, +1\}$ with a predicted score $s \in \mathbb{R}$ in terms of a non-increasing function $f : \mathbb{R} \longrightarrow \mathbb{R}$, it is easy to verify that (10) is given by $L^*(Y, S) = f(|ys|)$ for $Y = \{-1, +1\}$ (and, of course, $L^*(Y, S) = L(Y, s) = f(ys)$ for $Y = \{-1\}$ and $Y = \{+1\}$).

3.4 Superset Learning for Structured Output Prediction

Existing work on superset learning has been focused almost exclusively on (multi-class) classification. Obviously, our approach is not restricted to this problem; instead, the output space $\mathcal{Y}$ is completely generic. In fact, one may even argue that superset learning is more interesting for problems with complex, structured outcomes, since outcomes of that kind are often only partially specified in practice. A partial structure is then quite naturally associated with a subset of $\mathcal{Y}$, namely the set of all consistent completions—note that this view is somehow in contrast to the common view of a label set Y_n as a *corruption* of the true label, and of the additional labels as *distractors* [13]. In the following, we shall instantiate our approach for a problem of that kind, namely label ranking [19], where the output space consists of rankings (permutations).

4 Label Ranking

Let $\mathcal{C} = \{c_1, \ldots, c_K\}$ be a finite set of (choice) alternatives, referred to as *labels*. We consider total order relations $\succ$ on $\mathcal{C}$, where $c_i \succ c_j$ indicates that c_i precedes c_j in the order. Since a ranking can be seen as a special type of preference relation, we shall also say that $c_i \succ c_j$ indicates a preference for c_i over c_j. Formally, a total order $\succ$ can be identified with a permutation $\bar{\pi}$ of the set $[K]$, such that $\bar{\pi}(i)$ is the position of c_i in the order. Let the output space $\mathcal{Y}$ be given by the set of permutations of $[K]$ (the symmetric group of order K).

In the setting of label ranking, preferences are "contextualized" by instances $\boldsymbol{x} \in \mathcal{X}$. Thus, each instance $\boldsymbol{x}$ is associated with a ranking $\succ_{\boldsymbol{x}}$ of the label set $\mathcal{C}$ or, equivalently, a permutation $\bar{\pi}_{\boldsymbol{x}} \in \mathcal{Y}$. More specifically, since label rankings do not necessarily depend on instances in a deterministic way, each instance $\boldsymbol{x}$ is associated with a probability distribution $\mathbf{P}(\cdot \,|\, \boldsymbol{x})$ on $\mathcal{Y}$. Thus, for each $\bar{\pi} \in \mathcal{Y}$, $\mathbf{P}(\bar{\pi} \,|\, \boldsymbol{x})$ denotes the probability to observe $\bar{\pi}$ in the context specified by $\boldsymbol{x}$.

The goal in label ranking is to learn a "label ranker", that is, a model $\mathcal{M} : \mathcal{X} \longrightarrow \mathcal{Y}$ that predicts a ranking $\hat{\pi}$ for each instance $\boldsymbol{x}$ given as an input. As training data $\mathcal{D}$, a label ranker uses a set of instances $\boldsymbol{x}_n$ ($n \in [N]$), together with information about the associated rankings π_n. Ideally, complete rankings are given as training information, i.e., a single observation is a tuple of the form $(\boldsymbol{x}_n, \pi_n) \in \mathcal{X} \times \mathcal{Y}$; we call an observation of that kind a *complete* example. From a practical point of view, however, it is important to allow for incomplete information in the form of a ranking of some but not all of the labels in $\mathcal{C}$:

$$c_{\tau(1)} \succ_{\boldsymbol{x}} c_{\tau(2)} \succ_{\boldsymbol{x}} \ldots \succ_{\boldsymbol{x}} c_{\tau(J)} \quad , \tag{12}$$

where $J < K$ and $\{\tau(1), \ldots, \tau(J)\} \subset [K]$. In the following, we will write complete rankings $\bar{\pi}$ with an upper bar (as we already did above). If a ranking π is not complete, then $\pi(j)$ is the position of c_j in the incomplete ranking, provided this label is contained, and $\pi(j) = 0$ otherwise.

Information in the form of an incomplete ranking π is naturally represented in terms of a subset $Y = E(\pi) \subseteq \mathcal{Y}$, namely the set of all of its linear extensions $E(\pi)$ (complete rankings preserving the order of those labels contained in π). Note that, if $\bar{\pi}$ is a completion of π, then $\bar{\pi}(k) \geq \pi(k)$ for all $k \in [K]$.

4.1 Prediction Accuracy

The prediction accuracy of a label ranker is typically assessed by comparing the true ranking $\bar{\pi}$ with the prediction $\hat{\pi}$ in terms of a distance measure on rankings.

Among the most commonly used measures is the Kendall distance, which is defined by the number of inversions, that is, index pairs $\{i,j\} \subset [K]$ such that the order of c_i and c_j in $\bar{\pi}$ is inverted in $\hat{\pi}$:

$$D(\bar{\pi}, \hat{\pi}) = \sum_{1 \leq i < j \leq K} [\![\operatorname{sign}(\bar{\pi}(i) - \bar{\pi}(j)) \neq \operatorname{sign}(\hat{\pi}(i) - \hat{\pi}(j))]\!] \tag{13}$$

The well-known Kendall rank correlation measure is an affine transformation of (13) to the range $[-1, +1]$. Besides, the sum of L_1 or L_2 losses on the ranks of the individual labels are often used as an alternative:

$$D_1(\bar{\pi}, \hat{\pi}) = \sum_{i=1}^{K} |\bar{\pi}(i) - \hat{\pi}(i)|, \quad D_2(\bar{\pi}, \hat{\pi}) = \sum_{i=1}^{K} (\bar{\pi}(i) - \hat{\pi}(i))^2 \tag{14}$$

These measures are closely connected with two other well-known rank correlation measures: Spearman's footrule is an affine transformation of D_1 to the interval $[-1, +1]$, and Spearman's rank correlation (Spearman's rho) is such a transformation of D_2.

4.2 Label Ranking Methods

Several methods for label ranking have been proposed that try to exploit, in one way or the other, the complex though highly regular structure of the output space $\mathbb{S}_K$. These include generalizations of standard machine learning methods such as nearest neighbor estimation and decision tree learning [4], as well as statistical inference based on parametrized models of rank data [3]. Moreover, several *reduction techniques* have been proposed, that is, meta-learning techniques that reduce the original label ranking problem into one or several classification problems that are easier to solve [7,10].

Since the (base) learner used to realize label ranking is actually of minor interest for our purpose, we shall stick to a simple nearest neighbor approach in this paper. The most obvious way of exploiting our framework for superset learning to realize such an approach consists of predicting, for a new query instance $\boldsymbol{x}_0$, the ranking

$$\hat{\pi} \in \operatorname*{argmin}_{\pi \in \mathcal{Y}} \sum_{n=1}^{nn} L^*(E(\pi_n), \pi) \ , \tag{15}$$

where $\pi_1, \ldots, \pi_{nn}$ are the (incomplete) rankings coming from the nn nearest neighbors of $\boldsymbol{x}_0$, and L^* is the OSL extension of a loss such as (13) or (14). However, depending on the loss chosen, the problem of finding a minimizer in (15) may become computationally expensive. Therefore, we subsequently introduce a new meta-learning technique for label ranking, which is based on the idea of reducing the original problem to standard classification problems.

5 Label Ranking based on Labelwise Decomposition

Unlike existing reduction techniques, which transform the original label ranking problem to a single large or a quadratic number of small binary classification problems [7,10], our approach is based on a *labelwise* decomposition into K ordinal classification problems. As will be explained in more detail in the following, the basic idea is to train one model per label, namely a model that maps instances to ranks.

5.1 Complete Training Information

If the training data $\mathcal{D}$ is precise, i.e., consists of complete examples $(\boldsymbol{x}_n, \bar{\pi}_n)$, then each such example informs about the rank $\bar{\pi}(k)$ of the label c_k in the ranking associated with $\boldsymbol{x}_n$. Thus, a quite natural idea is to learn a model

$$M_k : \mathcal{X} \longrightarrow [K]$$

that predicts the rank of c_k, given an instance $\boldsymbol{x} \in \mathcal{X}$ as an input. Indeed, such a model can be trained easily on the (label-specific) data

$$\mathcal{D}_k = \Big\{ (\boldsymbol{x}_n, r_n) \,|\, (\boldsymbol{x}_n, \bar{\pi}_n) \in \mathcal{D},\, r_n = \bar{\pi}_n(k) \Big\}. \tag{16}$$

The classification problems thus produced are multi-class problems with K classes, where each class corresponds to a possible rank. More specifically, since these ranks have a natural order, we are facing an *ordinal classification* problem. Thus, training of the models M_k ($k \in [K]$) can in principle be accomplished by any existing method for ordinal classification.

5.2 Incomplete Training Information

As mentioned before, the original training data is not necessarily precise; instead, for a training instance $\boldsymbol{x}_n$, only an incomplete ranking π_n of a subset of the labels in $\mathcal{C}$ might have been observed, while the complete ranking $\bar{\pi}_n$ is not given. In this case, the above method is not directly applicable: If at least one label is missing, i.e., $|\pi_n| < K$, then none of the true ranks $\bar{\pi}_n(k)$ is precisely known; consequently, the training data (16) cannot be constructed.

Nevertheless, even in the case of incomplete rankings, non-trivial information can be derived about the rank $\bar{\pi}(k)$ for at least some of the labels c_k. In fact, if $|\pi| = J$ and $\pi(k) = r > 0$, then

$$\bar{\pi}(k) \in Y = \big\{ r, r+1, \ldots, r+K-J \big\} \ .$$

Of course, if $\pi(k) = 0$ (i.e., c_k is not present in the ranking), only the trivial information $\bar{\pi}(k) \in [K]$ can be derived. Yet, more precise information can be obtained under additional assumptions on the process of imprecisiation, which in this case is responsible for removing labels from the complete ranking. For example, if π is known to be the top of the ranking $\bar{\pi}$, then

$$\begin{cases} \bar{\pi}(k) = \pi(k) & \text{if } \pi(k) > 0 \\ \bar{\pi}(k) \in \{J+1, \ldots, K\} & \text{if } \pi(k) = 0 \end{cases} . \tag{17}$$

This scenario is practically relevant, since top-ranks are observed in many applications.

In general, the type of training data that can be derived for a label c_k in the case of incomplete rank information is of the form

$$\mathcal{O} = \big\{ \big(\boldsymbol{x}_n, Y_n \big) \big\}_{n=1}^{N} \subset \mathcal{X} \times 2^{[K]} \enspace , \tag{18}$$

that is, an instance $\boldsymbol{x}_n$ together with a set of possible ranks Y_n. Again, this is exactly the type of data assumed as an input by our approach to superset learning.

5.3 Generalized Nearest Neighbor Estimation

As already mentioned, we use a simple nearest neighbor approach for prediction: Given a new query instance $\boldsymbol{x}_0$, a prediction $\hat{\pi}$ is obtained by combining the (incomplete) rankings $\pi_1, \ldots, \pi_{nn}$ coming from the nn nearest neighbors of $\boldsymbol{x}_0$ in the training data $\mathcal{O}$. Denote by $Y_{k,n}$ ($k \in [K], n \in [nn]$) the (possibly imprecise) rank information for label c_k provided by π_n. Moreover, consider a distance $D(\cdot)$ on $\mathcal{Y}$ that is labelwise decomposable, i.e., which can be written in the form

$$D(\bar{\pi}, \hat{\pi}) = \sum_{k=1}^{K} L(\bar{\pi}(k), \hat{\pi}(k)).$$

Obviously, the L_1 and L_2 loss in (14) are both of this type. Then, the empirical risk of $\hat{\pi}$, i.e., the loss of this prediction in the neighborhood of $\boldsymbol{x}_0$, is given by

$$\begin{aligned} \sum_{n=1}^{nn} D(\bar{\pi}_n, \hat{\pi}) &= \sum_{n=1}^{nn} \sum_{k=1}^{K} L(\bar{\pi}_n(k), \hat{\pi}(k)) && (19) \\ &= \sum_{k=1}^{K} \sum_{n=1}^{nn} L(\bar{\pi}_n(k), \hat{\pi}(k)) && (20) \\ &= \sum_{k=1}^{K} L_k(\hat{\pi}(k)), && (21) \end{aligned}$$

where $L_k(r)$ is the cost of putting label c_k on position r. Taking into account that in general only incomplete rankings π_n are observed, the loss $L(\cdot)$ should be replaced by its generalization (10) and, therefore, L_k should be defined as

$$L_k(r) = \sum_{n=1}^{nn} L^*(Y_{k,n}, r) \enspace .$$

Thus, an optimal solution would consist of assigning c_k the position $\hat{\pi}(k) = r$ for which $L_k(r)$ is minimal. However, noting that each position $r \in [K]$ must be assigned at most once, this approach is obviously not guaranteed to produce a feasible solution. Instead, the minimization of (19) requires the solution of an *optimal assignment problem* [2]:

- labels $c_k \in \mathcal{C}$ must be uniquely assigned to ranks $r = \hat{\pi}(k) \in [K]$;
- assigning c_k to rank r causes a cost of $L_k(r)$;
- the goal is to minimize the sum of all assignment costs.

Assignment problems of that kind have been studied extensively in the literature, and efficient algorithms for their solution are available. The well-known Hungarian algorithm [12], for example, solves the above problem in time $O(K^3)$. Such algorithms can be used to produce a prediction $\hat{\pi}$ that minimizes the sum of assignment costs $L_1(\hat{\pi}(1)) + \ldots + L_K(\hat{\pi}(K))$, and therefore to realize our nearest neighbor approach to label ranking. In the next section, we experimentally analyze this approach with L given by D_1 in (14).

5.4 Experiments

In this section, we experimentally compare our new method, referred to as LWD (for Label-Wise Decomposition), with another nearest neighbor approach to label ranking. This approach is based on the (local) estimation of the parameters of a probabilistic model called the Plackett-Luce (PL) model [3]. It is known to achieve state-of-the-art performance, not only among the nearest neighbor approaches but among label ranking methods in general. Apart from that, the comparison with PL is specifically interesting for the following reason: The approach is based on finding the probabilistic model, identified by a parameter vector $\boldsymbol{v} = (v_1, \ldots, v_K)$, for which the likelihood of observing the (neighbor) rankings is maximized:

$$\boldsymbol{v}^* \in \operatorname*{argmax}_{\boldsymbol{v} \in \mathbb{R}_+^K} \prod_{n=1}^{nn} \mathrm{PL}(\pi_n \,|\, \boldsymbol{v})$$

Now, with PL being a probability measure on the set of permutations $\mathcal{Y}$, the probability of an incomplete ranking π_n is given by the corresponding marginal, namely

$$\mathbf{P}(\pi_n \,|\, \boldsymbol{v}) = \sum_{\overline{\pi} \in E(\pi_n)} \mathrm{PL}(\overline{\pi} \,|\, \boldsymbol{v}) \enspace .$$

Thus, as can be seen, ambiguous examples are dealt with by *summing* over the corresponding superset, as opposed to *maximizing* as suggested by our approach (7). Since summation is more in line with averaging over all candidates than selecting the most plausible one, this approach is obviously less in the spirit of superset learning through *data disambiguation*.

As data sets, we used several benchmarks for label ranking that have also been used in previous studies [10]; these are semi-synthetic data sets, namely label ranking versions of (real) UCI multi-class data. Moreover, we used two real label ranking data sets: The Sushi data[5] consists of 5000 instances (customers) described by 11 features, each one associated with a ranking of 10 types of sushis. The Students data [1] consists of 404 students (each characterized by 126 attributes) with associated rankings of five goals (want to get along with my parents, want to feel good about myself, want to have nice things, want to be different from others, want to be better than others). See Table 1 for a summary of the data.

Two missing label scenarios (imprecisiation procedures) were simulated, namely a "missing-at-random" setting and the top-rank setting (17). In the first case, a biased coin is flipped for every label in a ranking to decide whether to keep or delete that label; the probability for a deletion is specified by a parameter $p \in [0, 1]$. Thus, $p \times 100\%$ of the labels will be missing on average. Similarly, in the second case, only the J top-labels in a ranking are kept, where J has a binomial distribution with parameters K and $1 - p$.

Table 1. Properties of the data sets.

data set	# inst. (N)	# attr. (d)	# labels (K)
authorship	841	70	4
glass	214	9	6
iris	150	4	3
pendigits	10992	16	10
segment	2310	18	7
vehicle	846	18	4
vowel	528	10	11
wine	178	13	3
sushi	5000	11	10
students	404	126	5

The results in Tables 2 and 3 are presented as averages of 5 × 10-fold cross validation in terms of the Kendall correlation measure; other measures such as (14) led to similar results. The number of nearest neighbors was determined through internal cross-validation. As a distance measure on $\mathcal{X}$, the standard Euclidean distance was used.

These results clearly support the conclusion that, while LWD and PL are quite en par in the complete ranking case, the latter is much more sensitive toward missing label information than the former. In fact, the performance of LWD is comparably stable, and its drop in performance due to missing label information is less pronounced than in the case of PL; this observation is especially clear in the missing-at-random setting (see Figure 4), whereas the differences in

[5] http://kamishima.new/sushi/

Table 2. Performance in terms of Kendall's tau on synthetic data: missing-at-random (above) and top-rank setting (below).

	complete ranking		30% missing labels		60% missing labels	
	LWD	PL	LWD	PL	LWD	PL
authorship	.933±.016	.936±.015	.925±.018	.833±.030	.891±.021	.601±.054
glass	.840±.075	.841±.067	.819±.078	.669±.064	.721±.072	.395±.068
iris	.960±.036	.960±.036	.932±.051	.896±.069	.876±.068	.787±.111
pendigits	.940±.002	.939±.002	.924±.002	.770±.004	.709±.005	.434±.007
segment	.953±.006	.950±.005	.914±.009	.710±.013	.624±.020	.381±.020
vehicle	.853±.031	.859±.028	.836±.032	.753±.032	.767±.037	.520±.050
vowel	.876±.021	.851±.020	.821±.022	.612±.027	.536±.034	.327±.033
wine	.938±.050	.947±.047	.933±.054	.919±.059	.921±.062	.863±.094
authorship	.933±.016	.936±.015	.932±.017	.927±.017	.923±.015	.886±.022
glass	.840±.075	.841±.067	.838±.074	.809±.066	.815±.075	.675±.069
iris	.960±.036	.960±.036	.956±.036	.926±.051	.932±.048	.868±.070
pendigits	.940±.002	.939±.002	.933±.002	.918±.002	.837±.004	.794±.004
segment	.953±.006	.950±.005	.943±.005	.874±.008	.844±.010	.674±.015
vehicle	.853±.031	.859±.028	.851±.033	.838±.030	.818±.032	.765±.035
vowel	.876±.021	.851±.020	.867±.021	.785±.020	.800±.021	.588±.024
wine	.938±.050	.947±.047	.936±.049	.926±.061	.930±.059	.907±.066

performance are less visible in the top-rank setting. In any case, these results are very interesting in light of our previous remarks on the comparison between *averaging* (product-sum inference) and *maximizing* (product-maximum inference) and clearly provide first evidence in favor of the effectiveness of learning through disambiguation in the context of structured output prediction.

Table 3. Performance in terms of Kendall's tau on real-world data: missing-at-random (above) and top-rank setting (below).

sushi	0%	10%	20%	30%	40%	50%	60%	70%
LWD	.323±.012	.322±.011	.320±.011	.319±.010	.315±.011	.308±.011	.296±.011	.277±.010
PL	.321±.010	.320±.010	.318±.010	.311±.010	.298±.011	.278±.010	.246±.010	.203±.012
LWD	.325±.012	.324±.011	.324±.011	.323±.011	.323±.011	.323±.011	.321±.011	.316±.011
PL	.321±.010	.320±.010	.320±.011	.320±.011	.319±.010	.316±.010	.310±.010	.303±.011

students	0%	10%	20%	30%	40%	50%	60%	70%
LWD	.641±.051	.641±.051	.640±.050	.640±.051	.638±.052	.637±.051	.633±.054	.626±.055
PL	.386±.028	.384±.027	.382±.026	.377±.029	.365±.025	.350±.027	.327±.027	.274±.033
LWD	.641±.051	.641±.051	.641±.051	.641±.051	.640±.051	.640±.052	.638±.050	.628±.052
PL	.386±.028	.385±.028	.386±.028	.385±.027	.383±.029	.379±.026	.377±.026	.371±.028

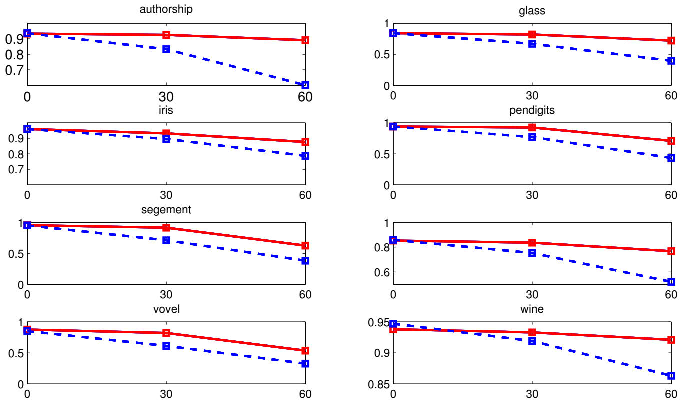

Fig. 4. Performance of LWD (solid lines) and PL (dashed line) in the missing-at-random setting.

6 Summary and Outlook

Our approach to superset learning is based on the idea of simultaneously finding the most plausible combination of model and data. As we explained, this idea could in principle also be realized by means of a probabilistic approach, and indeed, the principle of likelihood maximization was on the origin of our considerations. However, a full-fledged probabilistic approach is quite demanding and requires working with probability distributions both in the model and the data space. While perhaps being less principled, our approach relaxes these requirements: The plausibility of a model is captured in terms of how well it fits the data (according to a given loss); moreover, by merely distinguishing between possible and impossible instantiations of the imprecise data, plausibility in the data space is treated as a purely bivalent notion.

There are various directions for future work, notably the following:

- Depending on the underlying loss function $L(\cdot)$, the computation of the corresponding OSL (10) and solution of the optimization problem (11) may become complex, especially since (10) could be non-convex. Therefore, efficient algorithmic solutions need to be found for specific instantiations of our framework.
- Theoretical properties of our approach to superset learning need to be investigated. A specifically important question concerns conditions under which successful learning, for example in the sense of (stochastic) convergence toward an optimal model, is actually possible. An analysis of this kind obviously requires assumptions about the process of imprecisiation. Imagine, for example, a classification problem in which class A is *deterministically* added to the observed superset whenever the true class is B and vice versa. Learning to distinguish A from B is obviously impossible in that case. See [14]

for a first analysis of learnability in the context of superset label learning problem (superset learning for binary classification).

– The idea of tackling structured output prediction by superset learning appears to be interesting, and our results for label ranking are indeed promising. This idea should therefore be realized for other types of structured output prediction, too, for example multi-label classification [18].

References

1. M. Boekaerts, K. Smit, and F.M.T.A. Busing. Salient goals direct and energise students' actions in the classroom. *Applied Psychology: An International Review*, 4(S1):520–539, 2012.
2. R.E. Burkard, M. Dell'Amico, and S. Martello. *Assignment Problems.* SIAM, 2009.
3. W. Cheng, K. Dembczynski, and E. Hüllermeier. Label ranking based on the Plackett-Luce model. In *Proc. ICML–2010, Int. Conf. on Machine Learning*, Haifa, Israel, 2010.
4. W. Cheng, J. Hühn, and E. Hüllermeier. Decision tree and instance-based learning for label ranking. In *Proc. ICML–2009, 26th International Conference on Machine Learning*, Montreal, Canada, 2009.
5. T. Cour, B. Sapp, and B. Taskar. Learning from partial labels. *Journal of Machine Learning Research*, 12:1501–1536, 2011.
6. Y. Grandvalet. Logistic regression for partial labels. In *IPMU–02, Int. Conf. Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 1935–1941, Annecy, France, 2002.
7. Sariel Har-Peled, Dan Roth, and Dav Zimak. Constraint classification for multiclass classification and ranking. In *Proc. NIPS-02*, pages 785–792, 2003.
8. E. Hüllermeier. Learning from imprecise and fuzzy observations: Data disambiguation through generalized loss minimization. *International Journal of Approximate Reasoning*, 55(7):1519–1534, 2014.
9. E. Hüllermeier and J. Beringer. Learning from ambiguously labeled examples. *Intelligent Data Analysis*, 10(5):419–440, 2006.
10. E. Hüllermeier, J. Fürnkranz, W. Cheng, and K. Brinker. Label ranking by learning pairwise preferences. *Artificial Intelligence*, 172:1897–1917, 2008.
11. R. Jin and Z. Ghahramani. Learning with multiple labels. In *16th Annual Conference on Neural Information Processing Systems*, Vancouver, Canada, 2002.
12. H.W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1–2):83–97, 1955.
13. L.P. Liu and T.G. Dietterich. A conditional multinomial mixture model for superset label learning. In *Proc. NIPS*, 2012.
14. L.P. Liu and T.G. Dietterich. Learnability of the superset label learning problem. In *Proc. ICML 2014, Int. Conference on Machine Learning*, Beijing, China, 2014.
15. N. Nguyen and R. Caruana. Classification with partial labels. In *Proc. KDD 2008, 14th Int. Conf. on Knowledge Discovery and Data Mining*, Las Vegas, USA, 2008.
16. B. Schölkopf and AJ. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, 2001.
17. J. Sid-Sueiro. Proper losses for learning from partial labels. In *Proc. NIPS*, 2012.
18. Y.Y. Sun, Y. Zhang, and Z.H. Zhou. Multi-label learning with weak label. In *Proc. 24th AAAI Conference on Artificial Intelligence*, Atlanta, Georgia, USA, 2010.
19. Y. Zhou, Y. Lui, J. Yang, X. He, and L. Liu. A taxonomy of label ranking algorithms. *Journal of Computers*, 9(3), 2014.