INTERNATIONAL JOURNAL OF INTELLIGENT CONTROL AND SYSTEMS
VOL. 12, NO. 3, SEPTEMBER 2007, 230-236

Ordinal Regression as Multiclass Classification

Fen XIA, Liang ZHOU, Yanwu YANG, and Wensheng ZHANG

***Abstract*— Recently, an interesting framework was proposed to reduce the ordinal regression to the binary classification [1]. It made two independent assumptions: target functions are *rank-monotonic*; or rows of loss matrix are *convex*. Both of the two assumptions impose some restrictions on its application, because they may not be reliable in practice. This paper presents a novel reduction framework free of such restrictions, in this sense, which is more efficient and versatile in real world applications. Experiments on several datasets empirically validate its effectiveness. The contribution of our work is that it proves the fact that the ordinal regression is equivalent to the regular multiclass classification whose distribution is changed.**

***Index Terms*— Ordinal Regression, Ranking, Multiclass Classification, Cost-Sensitive.**

I. INTRODUCTION

ORDINAL regression is a supervised learning problem of predicting categories of ordinal scale. Training samples are labeled by a set of ranks, which exhibit an ordering among different categories. Ordinal regression lies somewhere between classification and regression. In contrast to classification, there is an ordinal relationship among categories. It also differs from regression in that the number of ranks is finite and the exact amount of differences among ranks is not defined.

Applications of the ordinal regression frequently occur in domains where human-generated data plays an important role. Examples of these domains include information retrieval, collaborative filtering, medicine, and psychology. For instance, in collaborative filtering, the Netflix Prize up to $\$1,000,000$ aims to substantially improve the accuracy of predictions about how much someone is going to love a movie based on their movie preferences[1]. When people assess objects of interest in a specific domain (e.g., in terms of their correctness, quality, or any other characteristics), they often resort to subjective evaluation and rating information that is typically imprecise. Besides, rating results or scores given by different people are usually not comparable. In practice, ordinal labels typically correspond to linguistic terms such as "bad", "good", and "very good".

Manuscript received July 20, 2007.

Fen Xia is with the Key Laboratory of Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences, Beijing, P.R. China, and also with the Graduate School, Chinese Academy of Sciences, Beijing, P.R. China (e-mail: fen.xia@ia.ac.cn).

Liang Zhou is with the Key Laboratory of Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences, Beijing, P.R. China, and also with the Graduate School, Chinese Academy of Sciences, Beijing, P.R. China (e-mail: xazl.ia.ac.cn@gmail.com).

Yanwu Yang is with with the Key Laboratory of Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences, Beijing, P.R China (e-mail: yanwu.yang@ia.ac.cn) .

Wensheng Zhang is with the Key Laboratory of Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences, Beijing, P.R China (e-mail: wensheng.zhang@ia.ac.cn) .

[1]It is available at http://www.netflixprize.com/.

Many approaches have been developed in machine learning literature to deal with the ordinal regression. One simple idea is to convert ordinal regressions to regular regression problems. For instance, Kramer *et.al* [2] investigated the use of a regression tree learner by mapping rating results to real values. However, an appropriate mapping is often difficult to construct, since the true, underlying metrics among ordinal scales are unknown for most of tasks. As a result, these regression algorithms are more sensitive to the representation of ranks rather than to ordinal relationships. Another idea is to convert ordinal regressions to a set of binary classification problems. In these approaches, results of these nested binary classifications are combined to produce rating predictions. For example, Frank and Hall [3] investigated the use of a classification tree learner to handle binary classification problems while Waegeman and Boullart [4] went into the case of Support Vector Machine (SVM). It is also possible to formulate ordinal regressions as preference judgment learning problems. Cohen *et.al* [5] considered these general ranking problems and presented a complexity gap between the classification and the ranking. Freund *et.al* [6] provided a formal framework and an efficient boosting algorithm for general ranking problems.

State-of-the-art approaches to ordinal regression assume that the ordinal response is a coarsely measured latent continuous variable, and model it with intervals on the real line. Based on this assumption, these algorithms usually seek a direction on which samples are well projected and a set of thresholds that divide this direction into consecutive intervals representing ordinal categories. They are called threshold model as a whole. Several methods have been proposed to seek the direction and thresholds, including a large margin algorithm based on a loss function defined on pair items of different ranks [7], a perceptron on-line algorithm [8], two large margin principles [9]. However, the formulation in [9] does not ensure that these thresholds are in the same order as the ranks. Chu and Keerthi [10] proposed two solutions to tackle this problem: the first is to add the ordering of the thresholds as a constraint to the original optimization problem; the second is to consider the training samples from all ranks to determine each threshold.

Though many approaches are proposed to deal with the ordinal regression, the learning foundation of the ordinal regression is still more desirable. For example, what is the relationship between the ordinal regression and other classical learning problems such as classification and regression? In this paper, we prove that the ordinal regression is equivalent to the multiclass classification whose distribution is changed. This proof is based on a reduction framework on extended samples, which are extracted from original samples and a given misclassification loss matrix. To guide the design of algorithms, we also prove that the empirical risk of classifiers

trained on multiclass classification samples is bounded by that on the weighted extended samples. Therefore, all algorithms and theories of the multiclass classification are immediately applied to the ordinal regression. Experimental results on some synthetic and benchmark datasets validate the effectiveness of our framework.

The remainder of this paper is organized as follows. Section 2 introduces related work about the learning foundation of the ordinal regression. The reduction framework along with its theoretical guarantee is presented in Section 3. Then we discuss multiclass classification algorithms, and propose one for extended samples in Section 4. In Section 5 we show experimental results, and conclude this paper in Section 6.

II. Related Work

During the past years, several pieces of work have been devoted to the learning foundation of the ordinal regression. They can be categorized into two groups: on inversion loss of pairs and on a given loss matrix. The former formulates the ordinal regression as a task of learning preference relations. In this formulation, a new loss function is proposed, which measures the $0/1$ loss of classifying preference relations of randomly drawn pairs. Based on the new loss function, the ordinal regression is converted to an augmented binary classification problem. The distribution of these pairs in the augmented space is derived from the product distribution of original samples. Therefore, the learning foundation of the ordinal regression is equivalent to that of the binary classification in an augmented space. Interested readers are referred to [6] and [7] for details. The latter formulates the ordinal regression as a special multiclass classification problem, where the loss of predicting an example of class y as class k is based on a pre-defined loss matrix instead of the commonly used $0/1$ loss. We prefer the latter as it truly reflects the nature of the ordinal regression . Moreover, the difference and relationship between the ordinal regression and the classification are clear in the latter.

Li and Lin [1] presented a reduction framework from the ordinal regression to the binary classification based on extended samples. In their framework, ordinal regression problems are converted to weighted binary classification problems, which assumes either rank functions f are *rank-monotonic* or rows of loss matrix are *convex*. The former requires that predicted values of ranks must be in the same order as true ranks, i.e., $f(\mathbf{x},1) \geq f(\mathbf{x},2) \geq \cdots \geq f(\mathbf{x},K-1)$ for each $\mathbf{x}$ with a K rank problem. The latter implies that the speed of loss increasing is not slower than that of the deviation of predicted ranks from true ranks. Actually, the former determines the function space of the approximator, thereby characterizing the range of datasets that can be applied to it. The latter restricts users' behaviors on the loss matrix. Unfortunately, in some datasets, these assumptions are not realistically reliable. For instance, Kosmelj and Vadnal [11] presented a case where the *rank-monotonic* function assumption cannot be met. In this paper, we present a novel reduction framework from the ordinal regression to the multiclass classification free of such assumptions. The advantage of eliminating assumptions is illustrated in our experiments.

III. The Reduction Framework

In this section, we give a formal definition of the ordinal regression. Then we introduce our reduction framework and its theoretical guarantee. Finally, the principle to guide the design of algorithms is given based on empirical risk bounds.

A. Definition of Ordinal Regression

The ordinal regression is similar to the multiclass classification, except for ordinal labels. The ordinal information can be interpreted as this way: the misclassification loss depends on the deviation degree of the prediction. For example, classifying an infant as an adult should loss more than as a child. If we encode the ordinal information in a loss matrix, the ordinal regression can be formulated in the same way as the multiclass classification. Consider basic assumptions made in supervised machine learning [12], the ordinal regression can be defined as follows.

Definition 1: Given an independently identically distributed (i.i.d) samples $S = \{(\mathbf{x_i}, y_i)\}_{i=1}^{l} \sim P_{\mathbf{X}Y}^{l}$ where $P_{\mathbf{X}Y}^{l} = \prod_{i=1}^{l} P_{\mathbf{X}Y}$, and a set $\mathcal{H}$ of mapping h from $\mathbf{X}$ to Y where $\mathbf{X}$ is the input vector space and $Y = \{1, 2, \ldots, K\}$ is set of ranks, an ordinal regression learning is to select one mapping h such that the risk functional $R(h)$ is minimized. The risk functional $R(h)$ is defined as

$$R(h) = E_{P_{\mathbf{X}Y}} l_{y,h(\mathbf{x})} = \int [\sum_{y=1}^{K} l_{y,h(\mathbf{x})} P(y|\mathbf{x})] p(\mathbf{x}) d\mathbf{x}, \quad (1)$$

where l is a $K \times K$ loss matrix with $l_{y,k}$ representing the cost of predicting an example $(\mathbf{x}, y)$ as rank k.

Naturally we assume $l_{y,y} = 0$ and $l_{y,k} > 0$ for $k \neq y$. Moreover, in order to reflect the ordinal information, each row of l must be *V-shaped*. That is, $l_{y,k-1}, > l_{y,k}$ if $k \leq y$ and $l_{y,k} < l_{y,k+1}$ if $k \geq y$.

If the loss matrix is with entries $l_{y,k} = 1$ for $y \neq k$ and $l_{y,y} = 0$, Definition 1 is actually a multiclass classification formulation. Therefore, the difference between the ordinal regression and the multiclass classification is equivalent to that between their loss matrixes.

B. Reducing Ordinal Regression to Multiclass Classification

Assume that $h(\mathbf{x})$ is a rank classifier and the most loss due to a misclassification on an example $(\mathbf{x}, y)$ is $l_{y,max}$. Then the loss of the rank classifier $h(\mathbf{x})$ on the example $(\mathbf{x}, y)$ is

$$\begin{aligned} l_{y,h(\mathbf{x})} &= \sum_{i=1}^{K} l_{y,i} - \sum_{i=1}^{K} l_{y,i} I(h(\mathbf{x}) \neq i) \\ &= \sum_{i=1}^{K} l_{y,i} - \sum_{i=1}^{K} l_{y,i} I(h(\mathbf{x}) \neq i) \\ &\quad + (K-1) l_{y,max} - (K-1) l_{y,max} \\ &= \sum_{i=1}^{K} (l_{y,max} - l_{y,i}) I(h(\mathbf{x}) \neq i) \\ &\quad - \sum_{i \in \{1,\ldots,K\} \setminus y} (l_{y,max} - l_{y,i}), \end{aligned} \quad (2)$$

where $I(x)$ is the step function, with value 1 if the inner condition is true, 0 otherwise.

Extended samples $(\mathbf{x}^{(k)}, y^{(k)})$ with weights $w_{y,k}$ are defined as

$$\mathbf{x}^{(k)} = \mathbf{x},\ y^{(k)} = k,\ w_{y,k} = (l_{y,max} - l_{y,k}). \tag{3}$$

Substituting (3) into (2) yields

$$l_{y,h(\mathbf{x})} = \sum_{k=1}^{K} w_{y,k} I(h(\mathbf{x}^{(k)}) \neq y^{(k)}) - f(y) \tag{4}$$

where $f(y) = \sum_{k\in\{1,\ldots,K\}\backslash y} w_{y,k}$.

Equation (4) shows that the loss of $h(\mathbf{x})$ on the example $(\mathbf{x}, y)$ equals to a weighted $0/1$ loss of $h(\mathbf{x})$ on extended samples minus a variable irrelevant to $h(\mathbf{x})$. Equation (4) also provides a way to transform the loss to weights of these samples. The weights can be used to modify $P_{\mathbf{X}Y}$ to produce a new distribution on $(\mathbf{X}, Y)$. In this way, the ordinal regression can be reduced to the multiclass classification. We formulate this as a theorem.

Theorem 2: Given an unknown probability distribution $P_{\mathbf{X}Y}$ on $(\mathbf{X}, Y)$, and a $K \times K$ loss matrix, there exists a distribution $\widehat{P}$ on the same space $(\mathbf{X}, Y)$, such that the solution to the ordinal regression on distribution P is equivalent to the solution to the multiclass classification on distribution $\widehat{P}$.

Proof: Substituting (4) into (1) yields

$$\begin{aligned} R(h) =& \int [\sum_{y=1}^{K} (\sum_{k=1}^{K} w_{y,k} I(h(\mathbf{x}^{(k)}) \neq y^{(k)}) \\ & - f(y)) P(y|\mathbf{x})] p(\mathbf{x}) d\mathbf{x} \\ =& \int [\sum_{y=1}^{K} (\sum_{k=1}^{K} w_{y,k} I(h(\mathbf{x}^{(k)}) \neq y^{(k)})) P(y|\mathbf{x})] p(\mathbf{x}) d\mathbf{x} \\ & - \int [\sum_{y=1}^{K} f(y) P(y|\mathbf{x})] p(\mathbf{x}) d\mathbf{x}. \end{aligned} \tag{5}$$

Using (3), reformulate (5) as

$$\begin{aligned} R(h) =& \int [\sum_{k=1}^{K} (\sum_{y=1}^{K} w_{y,k} P(y|\mathbf{x})) I(h(\mathbf{x}) \neq k)] p(\mathbf{x}) d\mathbf{x} \\ & - C_1 \\ =& C_2 \int [\sum_{k=1}^{K} I(h(\mathbf{x}) \neq k) \widehat{P}(k|\mathbf{x})] \widehat{p}(\mathbf{x}) d\mathbf{x} - C_1 \end{aligned} \tag{6}$$

where $C_1 = \int [\sum_{y=1}^{K} f(y) P(y|\mathbf{x})] p(\mathbf{x}) d\mathbf{x}$, $C_2 = \int g(\mathbf{x}) p(\mathbf{x}) d\mathbf{x}$, $g(\mathbf{x}) = \sum_{k=1}^{K} \sum_{y=1}^{K} w_{y,k} P(y|\mathbf{x})$ and $\widehat{P}(k|\mathbf{x}) = \sum_{y=1}^{K} w_{y,k} P(y|\mathbf{x}) / g(\mathbf{x})$, $\widehat{p}(x) = g(\mathbf{x}) p(\mathbf{x}) / C_2$ are probability distributions.

Equivalently, we can define a distribution $\widehat{P}(\mathbf{x}, k) = \widehat{P}(k|\mathbf{x}) \times \widehat{p}(\mathbf{x})$ that generates $(\mathbf{x}, k)$ from $P(\mathbf{x}, y)$ and the loss matrix l. The multiclass classification problem on distribution $\widehat{P}(\mathbf{x}, k)$ is to minimize

$$\begin{aligned} R(\widehat{h}) =& E_{\widehat{P}_{\mathbf{X}K}} l_{k,\widehat{h}(\mathbf{x})} \\ =& \int [\sum_{k=1}^{K} I(\widehat{h}(\mathbf{x}) \neq k) \widehat{P}(k|\mathbf{x})] \widehat{p}(\mathbf{x}) d\mathbf{x} \end{aligned} \tag{7}$$

Combining (6) and (7) results in

$$R(h) = C_2 R(\widehat{h}) - C_1 \tag{8}$$

It is obvious that $C_2 > 0$. Thus minimizing $R(\widehat{h})$ minimizes $R(h)$ as well. ■

Theorem 2 shows that, the methods solving an ordinal regression problem with an unknown probability distribution and a loss matrix can also be applied to a multiclass classification problem whose distribution is generated from the two features of the former.

It is easy to verify that, if the loss matrix is the multiclass classification loss matrix ($l_{y,k} = 1$ for $y \neq k$ and $l_{y,y} = 0$), then $C_2 = 1$, $C_1 = 0$, $\widehat{P}(k|\mathbf{x}) = P(k|\mathbf{x})$, $\widehat{p}(\mathbf{x}) = p(\mathbf{x})$ and $R(h) = R(\widehat{h})$. The multiclass classification problem is unchanged.

C. Empirical Risk Bounds

By (3), we transform a training sample set $S = \{(\mathbf{x_i}, y_i)\}$ to an extended training set $\widehat{S} = \{(\mathbf{x_i^{(k)}}, y_i^{(k)})\}$. If the number of original training samples is L, the extended training set contains at most $K \times N$ samples with the exact number depending on how many $w_{i,j}$s are positive. Each element is a possible outcome from distribution $\widehat{P}(\mathbf{x}, k)$ constructed in Theorem 2. However, not all elements are independent. Thus the whole extended set cannot directly be used as i.i.d outcomes from $\widehat{P}(\mathbf{x}, k)$. To get such outcomes, we choose an independent k_i from $\widehat{P}(k|\mathbf{x})$ for each $(\mathbf{x}_i, y_i)$ and produce a subset $T = \{(\mathbf{x}_i, k_i)\}_{i=1}^{l}$, which is a subset of $\widehat{S}$ do contain i.i.d outcomes from $\widehat{P}(\mathbf{x}, k)$.

The empirical risk on subset T is bounded by that on weighted extended training set $\widehat{S}$. This fact can guide the design of algorithms in our framework.

Theorem 3: Let $b_i^{(k)}$ be 0 if a classifier correctly predicts a sample $(\mathbf{x}_i^{(k)}, y_i^{(k)})$, and 1 otherwise. Let b_i be a Boolean random variable introduced by applying the classifier to a random sample $(\mathbf{x}_i, k_i)$ extracted from $(\mathbf{x}_i, y_i)$ according to $\widehat{P}(k_i|\mathbf{x}_i)$. For each sample $(\mathbf{x}_i, y_i)$ in the original training set $S = \{(\mathbf{x_i}, y_i)\}$, we assume that the posterior probability $P(Y = y_i|\mathbf{x}_i) = 1$. When each b_i is chosen independently, with probability at least $1 - \delta$ over this choice,

$$\frac{1}{N} \sum_{i=1}^{N} b_i \leq \frac{1}{N} \sum_{i=1}^{N} \frac{1}{c_{y_i}} \sum_{k=1}^{K} w_{y_i,k} b_i^{(k)} + O(\frac{1}{\sqrt{N}}, \sqrt{log\frac{1}{\delta}}) \tag{9}$$

where $c_{y_i} = \sum_{k=1}^{K} w_{y_i,k}$.

Proof: For each sample $(\mathbf{x}_i, y_i)$ in the original training set, since the posterior probability $P(y_i|\mathbf{x}_i) = 1$, we have

$$\begin{aligned} \widehat{P}(k_i|\mathbf{x}_i) =& \sum_{y=1}^{K} w_{y,k} P(y|\mathbf{x}_i) / \sum_{k=1}^{K} \sum_{y=1}^{K} w_{y,k} P(y|\mathbf{x}_i) \\ =& w_{y_i,k} / c_{y_i} \end{aligned}$$

Thus the random variable b_i has mean $\sum_{k=1}^{K} w_{y_i,k} b_i^{(k)}/c_{y_i}$, which equals to the probability of $b_i = 1$. Now $b_1, \cdots, b_N$ are independent $\{0,1\}$-valued random variables with $Pr(b_i = 1) = \sum_{k=1}^{K} w_{y_i,k} b_i^{(k)}/c_{y_i}$. Following the Chernoff bounds [13], for $\epsilon \geq 0$,

$$Pr(\frac{1}{N}\sum_{i=1}^{N} b_i \geq (1+\epsilon)\mu) \leq exp(-\epsilon^2 \mu N/3), \tag{10}$$

where $\mu = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{c_{y_i}} \sum_{k=1}^{K} w_{y_i,k} b_i^{(k)}$.

Let $\delta = exp(-\epsilon^2 \mu N/3)$, then

$$\epsilon = \sqrt{\frac{3log(1/\delta)}{\mu N}}. \tag{11}$$

Thus, it yields (9) by combining (10)and (11). ■

Theorem 3 shows that a classifier aiming at a low weighted loss on extended sample set $\widehat{S} = \{(\mathbf{x_i^{(k)}}, y_i^{(k)})\}$ approximately aims at a low loss on subset $T = \{(\mathbf{x}_i, k_i)\}_{i=1}^{l}$.

IV. Algorithms Based on the Framework

As proved in the previous section, the empirical risk of extended training set $\widehat{S}$ bounds that of its subset T. Thus the rank classifier can be obtained by minimizing the empirical risk of extended set $\widehat{S}$. Usually a regularization term is added to expect a low generalization loss on unseen samples.

There are three basic approaches in the literature to solve multiclass classification problems. The first is the *one-versus-one* or *pairwise-decomposition* approach that assumes the existence of a separator between any two classes. The second one is the *one-versus-all-decomposition* approach that assumes that for each class there exists a single separator between that class and all the other classes. The last one is the *winner-take-all* approach that assumes that each class is associated with a score function and all score functions are jointly trained on the training data set simultaneously. For instance, when the score functions are linear, they are called "linear machine" [14].

There are many inconsistent samples with different weights in the extended set. Any algorithms on the extended set should take this fact into account. Thus, the *one-versus-one* approach and *one-versus-all* approach cannot be applied to the extended set due to its separation assumption. The *winner-take-all* approach takes all samples into consideration and allows the instance weights as a whole for optimization, which is very suitable for the extended data set.

This paper presents such a *winner-take-all* algorithm for the extended data set. It is based on an existing algorithm named LogitBoost, which minimizes the negative binomial log-likelihood. LogitBoost uses a weight trimming trick to reduce computational cost, often by factors of 10 to 50. When compared to other proposed multiclass boosting algorithms, LogitBoost exhibits performance comparable in most situations, and far superior in some [15]. Another attractive property of LogitBoost is that, during training, LogitBoost adds more and more weights on training samples that are estimated to be close to the boundary. While, other multiclass boosting, for instance Multiclass AdaBoost [16], gives more weights to currently misclassified training samples, especially those far from the boundary. Since the extended set contains many inconsistent samples, misclassified training samples exist everywhere. This fact suggests the algorithm should consider more about the decision bound than misclassified training samples. Therefore, LogitBoost is proposed on the extended set. A minor modification of LogitBoost is made to allow weights of instances. The modified LogitBoost is named as Weighted LogitBoost, whose detailed description is given in Algorithm 1.

Algorithm 1 Weighted LogitBoost (*J* classes)

1. Input the extended set $\widehat{S} = \{(\mathbf{x}, k)\}_{i=1}^{N}$ with sample weight set $V = \{v_i\}_{i=1}^{N}$.
2. Start with weights $w_{ij} = 1/N$, $i = 1, \ldots, N$, $j = 1, \ldots, J$, $F_j(x) = 0$ and $p_j(x) = 1/J \ \forall j$.
3. Repeat for $m = 1, 2, \ldots, M$:
   (a). Repeat for $j = 1, \ldots J$:
      i. Computer working responses and weight in the jth class
      $$z_{ij} = \frac{y_{ij}^* - p_j(x_i)}{p_j(x_i)(1-p_j(x_i))}$$
      $$w_{ij} = p_j(x_i)(1-p_j(x_i))$$
      ii. Fit function $f_{mj}(x)$ by a weighted least-squares regression of z_{ij} to x_i with weight $w_{ij} * v_i$.
   (b). Set $f_{mj}(x) \leftarrow \frac{J-1}{J}(f_{mj}(x) - \frac{1}{J}\sum_{k=1}^{j} f_{mk}(x))$, and $F_j(x) \leftarrow F_j(x) + f_{mj}(x)$
   (c). Update $p_j(x)$ via $p_j(x) = \frac{e^{F_j(x)}}{\sum_{k=1}^{J} e^{F_k(x)}}$.
4. Output the classifier $argmax_j F_j(x)$

V. Experiments and Discussions

We make some experiments on a synthetic dataset to validate the effectiveness of our reduction framework. Then the performance of algorithms based on our framework is compared against the best existing results reported in [1] on eight benchmark datasets for the ordinal regression. The absolute loss matrix is used throughout the experiments for fair comparisons. That is a matrix with entries defined by $l_{y,k} = |y - k|$.

A. Synthetic Dataset

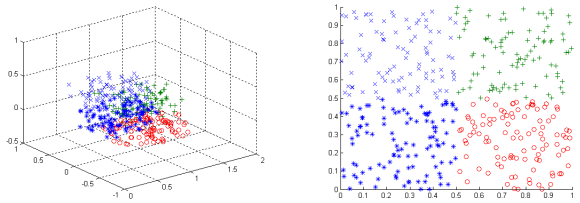
The synthetic dataset is generated in the following steps: Firstly, random points are generated according to the uniform distribution on the cubic area $[0,1] \times [0,1] \times [0,1]$. Then each point is assigned with one rank chosen from set {1,2,3,4} using the following rules:

If $x_1 < 0.5$ and $x_2 < 0.5$, the rank is 1;
If $x_1 < 0.5$ and $x_2 > 0.5$, the rank is 2.
If $x_1 > 0.5$ and $x_2 > 0.5$, the rank is 3.
If $x_1 > 0.5$ and $x_2 < 0.5$, the rank is 4.

Finally, each point is rotated by an angle $-\frac{\pi}{4}$ about x-axis and later by an angle $\arcsin\frac{1}{\sqrt{3}}$ about y-axis. This rotation can be written in term of a rotation matrix:

$$A = \begin{Bmatrix} \frac{\sqrt{6}}{3} & \frac{\sqrt{6}}{6} & \frac{\sqrt{6}}{6} \\ 0 & \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ -\frac{\sqrt{3}}{3} & \frac{\sqrt{6}}{6} & \frac{\sqrt{6}}{6} \end{Bmatrix}$$

We randomly generate 400 training samples to visualize the distribution of the dataset. Figure 1(a) shows the dataset in the rotated 3-dimensional space and Figure 1(b) shows the dataset in the $x - y$ plan before rotating.



(a) (b)

Fig. 1. The sample distribution in different feature space. (a) rotated 3-dimensional space; (b) $x - y$ plan before rotating.

In the reduction framework introduced in [1], the ordinal regression is converted to a weighted binary classification when rank functions are *rank-monotonic* or rows of loss matrix are *convex*. Since the loss matrix is user-related, we focus on the assumption of rank functions. As they proposed, one popular approach to obtain such rank functions is the threshold model. We thus take a Support Vector Ordinal Regression with IMplicit Constraints (SVOR-IMC) algorithm as a representative for algorithms of their framework [10]. To our best knowledge, SVOR-IMC is one of the best threshold model algorithms, which can achieve the best performance on some datasets. We use the Weighted LogitBoost as the representative algorithm of our framework. Both algorithms are implemented in the original feature space. In summary, the linear kernel is used in the SVOR-IMC and the linear regression is used as base learner in the Weighted LogitBoost.

We take 20 Monte-Carlo trials with 4000 training samples and a separate test set of 1000 samples to compare the performance of the two algorithms. The 5-fold cross validation on the training set is used to determine optimal values of model parameters: the regularization factor C for the SVOR-IMC and the number of iterations for the Weighted LogitBoost. Then the test error is obtained using the chosen parameters for each formulation. The search is done on a 7-grid linearly spaced $\{\log_{10} C | -3 \leq \log_{10} C \leq 3\}$ for C and up to 500 iterations for number of the iterations. The SVOR-IMC is implemented as [10] and the Weighted LogitBoost is implemented in *R Project* (*http://www.r-project.org*). Test results along with the standard deviation of two algorithms are recorded in Table I. The lowest values among the results are bold faced. MZE refers to the Mean Zero-one Error corresponding to the classification loss matrix, MAE the Mean Absolute Error corresponding to the absolute loss matrix.

Table I shows that the proposed Weighted LogitBoost algorithm significantly outperforms the SVOR-IMC in this synthetic dataset. The SVOR-IMC performs even worse than the half error. The reason for such large errors is the assumption of the rank function, which restricts the space of the hypotheses. As shown in Figure 1, any direction in the distribution of a dataset cannot be a *rank-monotonic* function. Thus in this dataset, it is problematic to reduce ordinal regression to binary classification in the original feature space.

TABLE I

TEST RESULTS OF THE TWO ALGORITHMS.

Algorithm	MZE	MAE
SVOR-IMC	0.505±0.013	0.508±0.135
Weighted-LogitBoost	**0.143±0.007**	**0.149±0.009**

B. Benchmark Datasets

For a comparison purpose, we use the datasets as in [1] and [10], i.e., eight benchmark datasets for regression problems. The same pre-processing as in [10] is done on each dataset. The target values are discretized into ten ordinal quantities using the equal-frequency binning. The input vectors are normalized to have zero mean and unit variance. Each dataset is randomly partitioned into training/test splits as specified in Table II.

The regression tree learner is used as a base learner in the Weighted LogitBoost algorithm. The number of iterations is determined by a 5-fold cross validation with maximum 500 iterations. The test error is obtained by using optimal model parameters for each dataset, and given in Table III together with those of [1]. These results are averaged over 20 trials. The lowest average values among results of four algorithms are marked with bold face. The absolute loss matrix are used for fair comparison. SVOR-IMC-P refers to SVOR-IMC with perceptron kernel and SVOR-IMC-G refers to SVOR-IMC with gaussian kernel .

TABLE II

THE PARTITION OF THE DATASETS.

Dataset	dimension	Training	test
Pyrimidines	27	50	24
MachineCpu	6	150	59
Boston	13	300	206
Abalone	8	1000	3177
Bank	32	3000	5192
Computer	21	4000	4192
California	8	5000	15640
Census	16	6000	16748

From Table III, we can observe that the Weighted LogitBoost outperforms the other three algorithms on some of the datasets, especially on "California", which demonstrates that the Weighted LogitBoost achieves decent out-of-sample performances. This implies that, on some datasets, our framework is more suitable than the one proposed by [1]. That is, the *rank-monotonic* assumption proposed by [1] cannot work well in several datasets. We also notice that results obtained by the Weighted LogitBoost have a larger variance than those of the other algorithms. This large variance might be caused by weights of samples.

TABLE III
TEST RESULTS OF WEIGHTED LOGITBOOST ALGORITHM, TOGETHER WITH THOSE OF [1].

Dataset	Weighted LogitBoost	SVM-perceptron	SVOR-IMC-P	SVOR-IMC-G
Pyrimidines	$\mathbf{1.271 \pm 0.205}$	1.304 ± 0.040	1.315 ± 0.039	1.294 ± 0.046
MachineCpu	$\mathbf{0.800 \pm 0.087}$	0.842 ± 0.022	0.814 ± 0.019	0.990 ± 0.026
Boston	0.816 ± 0.056	0.732 ± 0.013	$\mathbf{0.729 \pm 0.013}$	0.747 ± 0.011
Abalone	1.457 ± 0.014	1.383 ± 0.004	1.386 ± 0.005	$\mathbf{1.361 \pm 0.003}$
Bank	1.499 ± 0.016	1.404 ± 0.002	1.404 ± 0.002	$\mathbf{1.393 \pm 0.002}$
Computer	0.601 ± 0.007	$\mathbf{0.565 \pm 0.002}$	$\mathbf{0.565 \pm 0.002}$	0.596 ± 0.002
California	$\mathbf{0.882 \pm 0.009}$	0.940 ± 0.001	0.939 ± 0.001	1.008 ± 0.001
Census	$\mathbf{1.142 \pm 0.005}$	1.143 ± 0.002	1.143 ± 0.002	1.205 ± 0.002

VI. CONCLUSION

This paper presents a reduction framework from the ordinal regression to the multiclass classification based on extended samples. It proves the fact that the ordinal regression is equivalent to the regular multiclass problem whose distribution is changed. We also show that the empirical risk of the extended set bounds that of reduction multiclass classification samples. Based on the bound, a boosting-style algorithm named Weighted LogitBoost is proposed, and tailored to the extended set. Our reduction framework is compared with the one proposed by [1], which reduces the ordinal regression to the binary classification. Obviously, the multiclass classification is much more common and complex than the binary classification. Also our framework eliminates some restrictions introduced in their framework. Thus it is more versatile and efficient in real applications. Moreover, our framework can deal with an arbitrary loss matrix which enables it to solve other cost-sensitive multiclass problems. Experimental results on eight benchmark datasets empirically verify the effectiveness of our framework.

After many years' accumulation of research and application experiences, it produced a solid theoretical foundation for some learning algorithms such as SVM and boosting, and with proved effectiveness and satisfied performance. Different learning algorithms might be suitable in different model spaces accounting for different datasets. It is a promising and interesting research perspective to explore various learning algorithms in the proposed reduction framework.

ACKNOWLEDGMENT

This work was supported by the National Basic Research Program (2004CB318103), the National Science Foundation (60575001) of China, National Science Foundation of China (60033020), Overseas Outstanding Talent Research Program of Chinese Academy of Sciences(06S3011S01), and National Key Technology R&D Program (2006038097001).

REFERENCES

[1] L. Li and H-T. Lin, "Ordinal Regression by Extended Binary Classification," *Proceedings of the Conference on Neural Information Processing Systems 19*, Cambridge, MA: MIT Press, pp.865–872, 2007

[2] S. Kramer, G. Widmer, B. Pfahringer, and M. DeGroeve, "Prediction of ordinal classes using regression trees," *Fundamenta Informaticae, 47*, pp.1–13, 2001

[3] E. Frank and M. Hall, "A simple approach to ordinal classification," *In Proc 12th European Conference on Machine Learning*, pp.145–156. Springer, 2001.

[4] W. Waegeman and L. Boullart, "An ensemble of Weighted Support Vector Machines for Ordinal Regression," *Transactions on Engineering, Computing and Technology*, vol. 12, pp.71–75, Mar. 2006.

[5] W. W. Cohen, R. E. Schapire, and Y. Singer, "Learning to order things," *Journal of Artificial Intelligence Research*, vol. 10, pp. 243–270, 1999.

[6] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer, "An efficient boosting algorithm for combining preferences," *Journal of Machine Learning Research*, vol. 4, pp 933–969, 2003.

[7] R. Herbrich, T. Graepel, and K. Obermayer "Large margin rank boundaries for ordinal regression," *Advance in Large Margin Classifiers*, pp 115–132, 2000.

[8] K.Crammer and Y.Singer, " Pranking with ranking," *Proceedings of the Conference on Neural Information Processing Systems 14*, pp. 641–647, Cambridge, MA: MIT Press, 2001.

[9] A. Shashua and A Levin, "Ranking with Large Margin Principle: Two Approaches," *Advances in Neural Information Processing Systems 15*, pp. 937–944. MIT Press, 2003.

[10] W. Chu and S. Keerthi, "New Approaches to Support Vector Ordinal Regression," *Proceeding of the 22th International Conference on Machine Learning*, Bonn, Germany, pp. 321–328, 2005.

[11] K. Kosmelj and K. Vadnal, "Comparison of two generalized logistic regression models: a case study," *25th Int. Conf. Information Technology Interfaces*, pp. 199–204, 2003. Cavtat, Croatia.

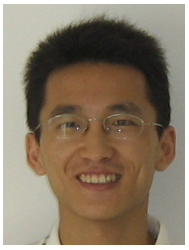
[12] V. Vapnik, *Statistical Learning Theory*, John Wiley and Sons Inc, 1998

[13] Chernoff. H, "A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations," *Annals of Mathematical Statistics*, vol. 23, pp. 493–509, 1952.

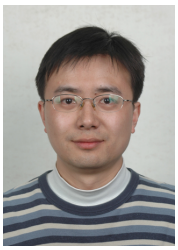
[14] R. O. Duda, P. E. Hart and D. G. Stork, *Pattern Classification and Scene Analysis*, John Wiley and Sons Inc, pp. 132–134, 1973.

[15] J. Friedman, T. Hastie and R. Tibshirani, "Additive logistic regression: A statistical view of boosting," *The Annals of Statistics*, vol. 28, pp. 337–407, 2000.

[16] J. Zhu., S. Rosset, H. Zou, and T. Hastie, "Multi-class adaboost," *Department of Statistics, University of Michigan*, 2005.



Fen Xia is a PhD candidate in the Key Lab of Complex System and Intelligent Science at Institute of Automation China Academy of Sciences. He received his Bachelor degree in Automation at the University of Science and Technology of China (USTC) in 2003. His research interests include statistical machine learning, ranking, regularization methods, efficient algorithms, image process.



Liang Zhou was born in HangZhou, China, on October 4, 1979. He received the B.S. degree from Xi'an Jiaotong University, Xi'an, China, in 2002, and he is currently working towards the Ph.D. degree in Machine Learning from The CAS Laboratory of Complex Systems and Intelligence Science(LCSIS), Institute of Automation, Chinese Academy of Sciences. He current research interests include pattern recognition, statistical machine learning, and data mining.

Yanwu Yang received the Ph.D degree in computer science from the doctoral school of the Ecole Nationale Superieure d'Arts et Metiers(ENSAM), France in 2006. He joined the lab of Complex Systems and Intelligence Sciences in 2007. His current interests include user model, Human-Computer Interaction and text mining.

Wensheng Zhang received the Ph.D. degree in Pattern Recognition and Intelligent Systems from the Institute of Automation, Chinese Academy of Sciences (CAS), in 2000. He joined the Institute of Software, CAS, in 2001. He is a Professor of Machine Learning and Data Mining and the Director of Research and Development Department, Institute of Automation, CAS. He has published over 32 papers in the area of Modeling Complex Systems, Statistical Machine Learning and Data Mining. His research interests include Intelligent Information Processing, Pattern Recognition, Artificial Intelligence and Computer Human Interaction.