Data Mining and Knowledge Discovery (2021) 35:1134–1161
https://doi.org/10.1007/s10618-021-00746-8

Tackling ordinal regression problem for heterogeneous data: sparse and deep multi-task learning approaches

Lu Wang[1] · **Dongxiao Zhu**[1]

Received: 9 January 2019 / Accepted: 4 March 2021 / Published online: 23 March 2021

Abstract

Many real-world datasets are labeled with natural orders, i.e., ordinal labels. Ordinal regression is a method to predict ordinal labels that finds a wide range of applications in data-rich domains, such as natural, health and social sciences. Most existing ordinal regression approaches work well for independent and identically distributed (IID) instances via formulating a single ordinal regression task. However, for heterogeneous non-IID instances with well-defined local geometric structures, e.g., subpopulation groups, multi-task learning (MTL) provides a promising framework to encode task (subgroup) relatedness, bridge data from all tasks, and simultaneously learn multiple related tasks in efforts to improve generalization performance. Even though MTL methods have been extensively studied, there is barely existing work investigating MTL for heterogeneous data with ordinal labels. We tackle this important problem via sparse and deep multi-task approaches. Specifically, we develop a regularized multi-task ordinal regression (MTOR) model for smaller datasets and a deep neural networks based MTOR model for large-scale datasets. We evaluate the performance using three real-world healthcare datasets with applications to multi-stage disease progression diagnosis. Our experiments indicate that the proposed MTOR models markedly improve the prediction performance comparing with single-task ordinal regression models.

Keywords Ordinal regression · Multi-task learning · Heterogeneous data · Non-IID learning · Deep neural network · Multi-stage disease progression, Diagnosis

Responsible editor: Pierre Baldi.

✉ Dongxiao Zhu
dzhu@wayne.edu

Lu Wang
lu.wang3@wayne.edu

[1] Department of Computer Science, Wayne State University, Detroit, MI 48202, USA

1 Introduction

Ordinal regression is capable of exploiting ordinal labels to solve multi-ordered classification problems, which has been widely applied to diverse application domains (Domingo-Ferrer and Torra 2005; Henriques et al. 2015), e.g., medical diagnosis (Brookmeyer et al. 2007; Davis et al. 2010; Chan and Norat 2015; Cruickshank et al. 2015), social science (Kaplan 2004; O'Connell 2006; Grosskreutz and Rüping 2009; Lemmerich et al. 2016), education (Chen and John 2004; Hamidi et al. 2008), computer vision (Kim 2014; Liu et al. 2017; Niu et al. 2016; Liu et al. 2018) and marketing (Menon and Elkan 2010; Montañés et al. 2014; Lanfranchi et al. 2014). Specifically in medical diagnosis, many major diseases are multi-stage progressive, for example, Alzheimer's Disease (AD) progresses into three stages that are irreversible with orders, i.e., cognitively normal, mild cognitive impairment and AD (Brookmeyer et al. 2007). Conventional methods either convert ordinal regression problems into multiple binary classification problems (Frank and Hall 2001; Kato et al. 2008; Park and Fürnkranz 2012) (e.g., health and illness) or consider them as multi-class classification problems (Har-Peled et al. 2002; Gursoy et al. 2017). However, these methods fail to capture the key information of ordinal labels (e.g., the progression of multi-stage diseases). Therefore, ordinal regression is essential as it incorporates the ordinal labels in multi-class classification (Cruz et al. 2001; Tran et al. 2015; Hong and He 2010).

In the real-world scenario, there is an increasing need to build multiple related ordinal regression tasks for heterogeneous data sets. For instance, multi-stage disease diagnosis in multiple patient subgroups (e.g., various age groups, genders, races), student satisfaction questionnaire analysis in multiple student subgroups (e.g., various schools, majors), customer survey analysis in multiple communities (e.g., various incomes, living neighborhoods). However, most of the prior works merely concentrate on learning a single ordinal regression task, i.e., either build a global ordinal regression model for all sub-population groups, ignoring data heterogeneity among different subgroups (Chu and Keerthi 2005, 2007; Schmidt-Richberg et al. 2015; Gu et al. 2015); or build and learn an ordinal regression model for each subgroup independently, ignoring relatedness among these subgroups (Cruz et al. 2001; Tran et al. 2015; Hong and He 2010).

To overcome the aforementioned limitations, multi-task learning (MTL) is introduced to learn multiple related tasks simultaneously (Caruana 1998), which has been extensively researched in tackle classification and standard regression problems. By building multiple models for multiple tasks and learning them collectively, the training of each task is augmented via the auxiliary information from other related subgroups, leading to an improved generalization of prediction performance. MTL has achieved significant successes in analyzing heterogeneous data, such as prediction of patients' survival time for multiple cancer types (Wang et al. 2017), prioritzation of risk factors in obesity (Wang et al. 2019) and HIV therapy screening (Bickel et al. 2008). However, MTL for heterogeneous data with ordinal labels, such as multi-stage disease diagnosis of multiple patient subgroups, remains a largely unexplored and neglected domain. Multi-stage progressive diseases are rarely cured completely and the progression is often irreversible, e.g., AD, hypertension, obesity, dementia and multiple sclerosis (Brookmeyer et al. 2007; Chan and Norat 2015; Cruickshank et al. 2015). Hence new

ordinal regression approaches are urgently needed to analyze emerging heterogeneous and/or large-scale data sets.

To train multiple correlated ordinal regression models jointly, (Yu et al. 2006) connect these models using Gaussian process prior within the hierarchical Bayesian framework. However, multi-task models within the hierarchical Bayesian framework are not sparse or performed well in high dimensional data. In Gao and Zhao (2018), forecasting the spatial event scale is targeted using the incomplete labeled datasets, which means not every task has a complete set of labels in the training dataset. The objective function in Gao and Zhao (2018) is regularized logistic regression derived from logistic ordinal regression; therefore, their approach also suffers from the limitations of logistic regression, e.g., more sensitive to outliers comparing with our proposed methods based on maximum-margin classification (Rennie and Srebro 2005; Frome et al. 2007).

Here we propose a regularized multi-task ordinal regression (MTOR) model to analyze heterogeneous and smaller datasets. Moreover, we develop a deep neural networks (DNN) based model for heterogeneous and large-scale data sets. The proposed MTOR approach can be considered as the regularized MTL approach (Evgeniou and Pontil 2004), where the assumption of task relatedness is encoded via regularization terms that have been widely studied in the past decade (Argyriou et al. 2008; Liu et al. 2009). In this work, the task relatedness is encoded by shared representation layers. We note that Kato et al. (2008) formulates a single ordinal regression problem as a multi-task binary classification problem whereas in our work we solve multiple ordinal regression problems simultaneously within the MTL framework.

In this paper, we employ the alternating structure optimization to achieve an efficient learning scheme to solve the proposed models. In the experiments, we demonstrate the prediction performance of our models using three real-world datasets corresponding to three multi-stage progressive diseases, i.e., AD, obesity and hypertension with well-defined yet heterogeneous patient age subgroups. The main contributions of this paper can be summarized as follows:

- We propose a regularized MTOR model for smaller yet heterogeneous datasets to encode the task relatedness of multiple ordinal regression tasks using structural regularization term;
- We propose a DNN based MTOR model for large-scale datasets to encode the task relatedness via the shared hidden layers;
- We propose an alternating structure optimization framework to train our models, and within this framework the fast iterative shrinkage thresholding algorithm (FISTA) is employed to update the model weights;
- Our comprehensive experimental studies demonstrate the advantage of MTOR models over single-task ordinal regression models.

The rest of this paper is organized as follows: Sect. 2 summarizes relevant works on ordinal regression and MTL. In Sect. 3, we review the preliminary knowledge on the ordinal regression. Section 4 elaborates the details of MTOR models. In Section 5, we extend the MTOR model to deep learning using DNN to accommodate large-scale heterogeneous data sets. Section 6 demonstrates the effectiveness of the MTL ordinal

regression models using three real-world healthcare datasets for the multi-stage disease diagnosis. In Sect. 7, we conclude our work with discussion and future work.

2 Related works

In this section, we summarize the related works in the fields of ordinal regression and multi-task learning, and discuss the relationships and primary distinctions of the proposed methods compared to the existing methods in the literature.

2.1 Ordinal regression

Ordinal regression is an approach aiming at classifying the data with natural ordered labels and plays an important role in many data-rich science domains. According to the commonly used taxonomy of ordinal regression (Gutiérrez et al. 2016), the existing methods are categorized into: naive approaches, ordinal binary decomposition approaches and threshold models.

The naive approaches are the earliest approaches dealing with ordinal regression, which convert the ordinal labels into numeric and then implement standard regression or support vector regression (Witten et al. 2016; Kato et al. 2008). Since the distance between classes is unknown in this type of methods, the real values used for the labels may undermine regression performance. Moreover, these regression learners are sensitive to the label representation instead of their orders (Gutiérrez et al. 2016).

Ordinal binary decomposition approaches are proposed to decompose the ordinal labels into several binary ones that are then estimated by multiple models (Frank and Hall 2001; Li and Lin 2007). For example, (Frank and Hall 2001) transforms the data from U-class ordinal problems to $U - 1$ ordered binary classification problems and then they are trained in conjunction with a decision tree learner to encode the ordering of the original ranks, that is, train $U - 1$ binary classifiers using C4.5 algorithm as a decision tree learner.

Threshold models are proposed based on the idea of approximating the real value predictor followed with partitioning the real line of ordinal values into segments. During the last decade, the two most popular threshold models are support vector machines (SVM) models (Shashua and Levin 2003; Chu and Keerthi 2005, 2007; Gu et al. 2015) and generalized linear models for ordinal regression (Williams 2006; Baetschmann et al. 2015; Kockelman and Kweon 2002; Ye and Lord 2014); the former is to find the hyperplane that separates the segments by maximizing margin using the *hinge* loss and the latter is to predict the ordinal labels by maximizing the likelihood given the training data.

In Shashua and Levin (2003), support vector ordinal regression (SVOR) is achieved by finding multiple thresholds that partition the real line of ordinal values into several consecutive intervals for representing ordered segments; however, it does not consider the ordinal inequalities on the thresholds. In Chu and Keerthi (2005, 2007), the authors take into account ordinal inequalities on the thresholds and propose two approaches using two types of thresholds for SVOR by introducing explicit constraints. To deal

with incremental SVOR learning caused by the complicated formulations of SVOR, Gu et al. (2015) propose a modified SVOR formulation based on a sum-of-margins strategy to solve the computational scalability issue of SVOR.

Generalized linear models perform ordinal regression by fitting a coefficient vector and a set of thresholds, e.g., ordered logit (Williams 2006; Baetschmann et al. 2015) and ordered probit (Kockelman and Kweon 2002; Ye and Lord 2014). The margin functions are defined based on the cumulative probability of training instances' ordinal labels. Different link functions are then chosen for different models, i.e., logistic cumulative distribution function (CDF) for ordered logit and standard normal CDF for ordered probit. Finally, maximum likelihood principal is used for training.

With the development of deep learning, ordinal regression problems are transformed into binary classifications using convolutional neural network (CNN) to extract features (Niu et al. 2016; Liu et al. 2017). In Liu et al. (2018), CNN is also used to extract high-level features followed by a constrained optimization formulation minimizing the negative log-likelihood for the ordinal regression problems.

In this work, we propose novel ordinal regression models for heterogeneous data with subpopulation groups under the MTL framework. Particularly, we implement two different types of thresholds in the loss functions under different assumptions and use alternating structure optimization for training our models, which are different from existing threshold models using *hinge* loss or likelihood. Please refer to Sect. 4 for details.

2.2 Multi-task learning

To leverage the relatedness among the tasks and improve the generalization performance of machine learning models, MTL is introduced as an inductive transfer learning framework by simultaneously learning all the related tasks and transferring knowledge among the tasks. How task relatedness is assumed and encoded into the learning formulations is the central building block of MTL. In Evgeniou and Pontil (2004), the earliest MTL approach is to couple the learning process by using multi-task regularizations. Regularized MTL is able to leverage large-scale optimization algorithms such as proximal gradient techniques, so that the regularized MTL approach has a clear advantage over the other MTL approaches (Nesterov 2013; Liu et al. 2009; Ji and Ye 2009; Zhou et al. 2011). As a result, the regularized MTL can efficiently handle complicated constraints and/or non-smooth terms in the objective function.

Note that, we start this subsection by introducing some classical regularized MTL approaches. They demonstrate their models performance in different applications. For example on a benchmark dataset, i.e., School[1], which considers each school as one task to predict the same outcome exam scores in the multiple related tasks. Here we focus our literature review on the methods instead of applications.

MTL has been implemented with many deep learning approaches (Ruder 2017) in two ways, i.e., soft and hard parameter sharing of hidden layers. In the soft parameter sharing, all tasks do not share representation layers and the distance among their own representation layers are consytrained to encourage the parameters to be similar

[1] https://ttic.uchicago.edu/~argyriou/code/

(Ruder 2017), e.g., (Duong et al. 2015) and Yang and Hospedales (2016) use l_2-norm and the trace norm, respectively. Hard parameter sharing is the most commonly used approach in DNN based MTL (Ruder 2017) where all tasks share the representation layers to reduce the risk of overfitting (Baxter 1997) and keep some task-specific layers to preserve characteristics of each task (Lu et al. 2016). In this paper, we use the hard parameters sharing for DNN based MTOR. These existing methods are to solve either classification or standard regression problems. For the more challenging learning tasks of multiple ordinal regression. We describe our regularized MTOR model in Sect. 4 and deep learning based MTOR model in Sect. 5 to solve the multiple related ordinal regression problems simultaneously. Moreover, in the Sect. 6, the multi-stage disease diagnosis are demonstrated in experiments using the proposed MTOR models.

3 Preliminary: latent variable model in ordinal regression

Given N training instances denoted as $(X_i, Y_i)_{i\in\{1,...,N\}}$, the latent variable model is used to predict the ordinal label (Williams 2006):

$$\begin{aligned} Y^* &= XW + b, \\ \hat{Y}_i &= u \quad \text{if} \quad \vartheta_{\mu-1} < Y_i^* \leq \vartheta_\mu, \end{aligned} \tag{1}$$

where Y^* is the latent variable and $\hat{Y}_i$ is the ordered predicted label (i.e., $\hat{Y}_i = \mu \in \{1, ..., U\}$) for the i^{th} training instance. ϑ is a set of thresholds, where $\vartheta_0 = -\infty$ and $\vartheta_U = \infty$, so that we have $U - 1$ thresholds (i.e., $\vartheta_1 < \vartheta_2 < ... < \vartheta_{U-1}$) partitioning Y^* into U segments to obtain $\hat{Y}$, which can be expressed as:

$$\hat{Y} = \begin{cases} 1 \text{ if } \quad \vartheta_0 < Y^* \leq \vartheta_1, \\ \vdots \quad \vdots \qquad\quad \vdots \\ \mu \text{ if } \vartheta_{\mu-1} < Y^* \leq \vartheta_\mu, \\ \vdots \quad \vdots \qquad\quad \vdots \\ U \text{ if } \vartheta_{U-1} < Y^* \leq \vartheta_U. \end{cases} \tag{2}$$

As we see in Eq. (1) and Eq. (2), U ordered predicted labels, i.e., $\hat{Y}$, are corresponding to U ordered segments and each Y^* has the value within the range: $(\vartheta_{\mu-1}, \vartheta_\mu)$, the latter is immediate thresholds, for $\mu \in \{1, ..., U\}$.

4 Regularized multi-task ordinal regression (RMTOR) models

In this section, we formulate regularized multi-task ordinal regression (RMTOR) using two different types of thresholds: 1) Immediate thresholds: the thresholds between adjacent ordered segments including the first threshold ϑ_0 and last threshold ϑ_U. In the real-world problems, ϑ_0 and ϑ_U always remain in finite range. Hence, we can use the first and last thresholds to calculate the errors for training instances in

the corresponding segments. 2) All thresholds: the thresholds between adjacent and non-adjacent ordered segments followed the traditional definition of the first and last thresholds, i.e., $\vartheta_0 = -\infty$ and $\vartheta_U = \infty$. Thus, the first and last thresholds can not be used for calculating the errors of training instances.

4.1 Regularized multi-task learning framework

In the real-world scenario, multiple related tasks are more common comparing with many independent tasks. To employ MTL, many studies propose to solve a regularized optimization problem. Assume there are T tasks and G input variables/features in each corresponding dataset, then we have the weight matrix as $W \in R^{G\times T}$ and regularized MTL object function as:

$$\mathcal{J} = \min_{W} \mathcal{L}(W) + \Omega(W), \tag{3}$$

where $\Omega(W)$ is the regularization/penalty term, which encodes the task relatedness.

4.2 RMTOR using immediate thresholds (*RMTOR$_I$*)

4.2.1 *RMTOR$_I$* model

We define a margin function $M(D) := \log(1 + \exp(D))$ for the ordered pairwise samples as the logistic loss is a smooth loss that models the posterior probability and leads to better probability estimation at the cost of accuracy. The loss function of RMTOR with the immediate thresholds is formulated as:

$$\mathcal{L}_I = \sum_{t=1}^{T}\sum_{j=1}^{n_t}\left[M(\vartheta_{(Y_{tj}-1)} - X_{tj}W_t) + M(X_{tj}W_t - \vartheta_{Y_{tj}})\right], \tag{4}$$

where t is the index of task, n_t is the number of instances in the t^{th} task, j is the index of instance in the t^{th} task, Y_{tj} is the label of the j^{th} instance in the t^{th} task, $X_{tj} \in R^{1\times G}$, $W_t \in R^{G\times 1}$ and $\vartheta \in R^{T\times U}$. Note that, $\vartheta_{Y_{tj}}$ is a threshold in the t^{th} task, which is a scalar and its index is Y_{tj}. To visualize our immediate thresholds method, we show an illustration figure in Fig. 1.

Thus, we have the objective function $RMTOR_I$ as:

$$\begin{aligned} RMTOR_I = \min_{W,\vartheta} \sum_{t=1}^{T}\sum_{j=1}^{n_t}\big[&M(\vartheta_{(Y_{tj}-1)} - X_{tj}W_t) \\ &+ M(X_{tj}W_t - \vartheta_{Y_{tj}})\big] + \lambda||W||_{2,1}, \end{aligned} \tag{5}$$

where λ is the tuning parameter to control the sparsity and $\|W\|_{2,1} = \sum_{g=1}^{G} \sqrt{\sum_{t=1}^{T} |w_{gt}|^2}$. Note that, g is the index of feature and w_{gt} is the weight for the g^{th} feature in the t^{th} task.

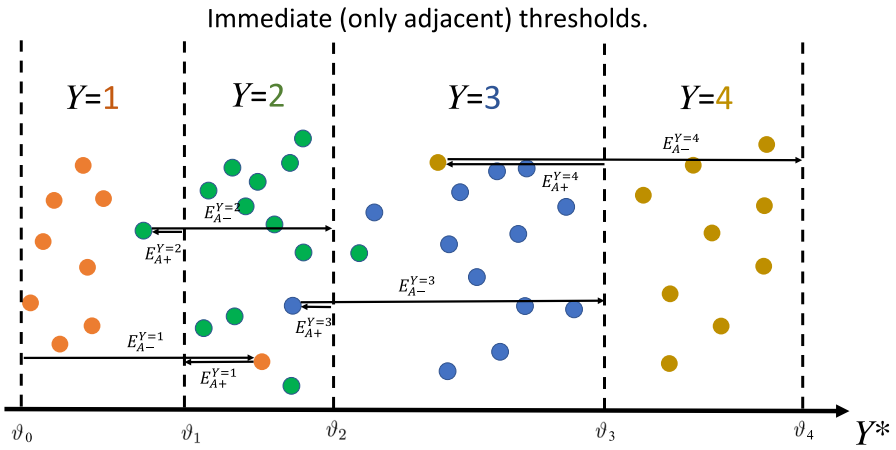

Fig. 1 Illustration of **immediate-thresholds loss** using four segments that only calculate the errors using the neighbor/adjacent thresholds of each segment when first and last thresholds remain in finite range. We denote $E_{A+/-}^{Y=\mu}$ as the error for a data point in the class μ, where A represents adjacent thresholds used and $+$ or $-$ indicates the error value is positive or negative. Note that, the solid arrow lines represent the errors calculated using neighbor/adjacent thresholds and the different direction of the arrow lines indicate the error direction. For example, $E_{A-}^{Y=1}$ denotes the error of a class 1 data point that equals $\vartheta_0 - X_{tj}^{Y=1} W_t$; this error is represented with a right direction arrow line in this figure and as ϑ_0 is smaller than $X_{tj}^{Y=1} W_t$, so its value is negative

4.2.2 Optimization

Alternating structure optimization (Ando and Zhang 2005) is a used to discover the shared predictive structure for all multiple tasks simultaneously, especially when the two sets of parameters W and ϑ in Eq. (5) can not be learned at the same time.

Optimization of **W** With fixed ϑ, the optimal W can be learned by solving:

$$\min_{W} \mathcal{L}_I(W) + \lambda ||W||_{2,1}, \tag{6}$$

where $\mathcal{L}_I(W)$ is a smooth convex and differentiable loss function, and the first order derivative can be expressed as:

$$\begin{aligned} \mathcal{L}_I'(W_t) &= \sum_{j=1}^{n_t} X_{tj}[G(X_{tj} W_t - \vartheta_{Y_{tj}}) \\ &\quad - G(\vartheta_{(Y_{tj}-1)} - X_{tj} W_t)], \\ \mathcal{L}_I'(W) &= \left[\frac{\mathcal{L}_I'(W_1)}{n_1}, \cdots, \frac{\mathcal{L}_I'(W_t)}{n_t}, \cdots, \frac{\mathcal{L}_I'(W_T)}{n_T} \right], \end{aligned} \tag{7}$$

where $G(D) := \frac{\partial M(D)}{\partial D} = \frac{1}{1+\exp(-D)}$.

To solve the optimization problem in Eq. (6), fast iterative shrinkage thresholding algorithm (FISTA) shown in Algorithm 1 is implemented with the general updating

steps:

$$W^{(l+1)} = \pi_P(S^{(l)} - \frac{1}{\gamma^{(l)}}\mathcal{L}'_I(S^{(l)})), \tag{8}$$

where l is the iteration index, $\frac{1}{\gamma^{(l)}}$ is the largest possible step-size that is chosen by line search (Beck and Teboulle 2009, Lemma 2.1, page 189) and $\mathcal{L}'_I(S^{(l)})$ is the gradient of $\mathcal{L}_I(\cdot)$ at search point $S^{(l)}$. $S^{(l)} = W^{(l)} + \alpha^{(l)}(W^{(l)} - W^{(l-1)})$ are the search points for each task, where $\alpha^{(l)}$ is the combination scalar. $\pi_P(\cdot)$ is $l_{2,1}-$regularized Euclidean project shown as:

$$\pi_P(H(S^{(l)})) = \min_W \frac{1}{2}||W - H(S^{(l)})||_F^2 + \lambda||W||_{2,1}, \tag{9}$$

where $||\cdot||_F$ is the Frobenius norm and $H(S^{(l)}) = S^{(l)} - \frac{1}{\gamma^{(l)}}\mathcal{L}'(S^{(l)})$ is the gradient step of $S^{(l)}$. An efficient solution (Theorem 1) of Eq. (9) has been proposed in Liu et al. (2009).

Theorem 1 *Given λ, the primal optimal point $\hat{W}$ of Eq. (9) can be calculated as:*

$$\hat{W}_g = \begin{cases} \left(1 - \frac{\lambda}{\|H(S^{(l)})_g\|_2}\right) H(S^{(l)})_g \text{ if } \lambda > 0, \| H(S^{(l)})_g \|_2 > \lambda \\ 0 \text{ if } \lambda > 0, \| H(S^{(l)})_g \|_2 \leq \lambda \\ H(S^{(l)})_g \text{ if } \lambda = 0, \end{cases} \tag{10}$$

where $H(S^{(l)})_g$ is the j^{th} row of $H(S^{(l)})$, and $\hat{W}_g$ is the g^{th} row of $\hat{W}$.

In lines 4-11 of Algorithm 1, the optimal $\gamma^{(l)}$ is chosen by the backtracking rule based on (Beck and Teboulle 2009 Lemma 2.1, page 189), $\gamma^{(l)}$ is greater than or equal to the Lipschitz constant of $\mathcal{L}_I(\cdot)$ at search point $S^{(l)}$, which means $\gamma^{(l)}$ is satisfied for $S^{(l)}$ and $\frac{1}{\gamma^{(l)}}$ is the possible largest step size.

In line 7 of Algorithm 1, $Q_\gamma(S^{(l)}, W^{(l+1)})$ is the tangent line of $\mathcal{L}_I(\cdot)$ at $S^{(l)}$, which can be calculated as:

$$\begin{aligned} Q_\gamma(S^{(l)}, W^{(l+1)}) = \mathcal{L}_I(S^{(l)}) + \frac{\gamma}{2} \| W^{(l+1)} - S^{(l)} \|^2 \\ + \langle W^{(l+1)} - S^{(l)}, \mathcal{L}'_I(S^{(l)}) \rangle. \end{aligned}$$

Algorithm 1: Fast iterative shrinkage thresholding algorithm (FISTA) for training RMTOR.

Input: A set of feature matrices $\{X_1, X_2, \cdots, X_T\}$, target value matrix Y for all T tasks, initial coefficient matrix $W^{(0)}$ and λ
Output: $\hat{W}$

Initialize: $W^{(1)} = W^{(0)}, d_{-1} = 0, d_0 = 1, \gamma^{(0)} = 1, l = 1$;
repeat
  Set $\alpha^{(l)} = \frac{d_{l-2}-1}{d_{l-1}}$, $S^{(l)} = W^{(l)} + \alpha^{(l)}(W^{(l)} - W^{(l-1)})$;
  for $j = 1, 2, \cdots$ **do**
    Set $\gamma = 2^j \gamma^{(l-1)}$;
    Calculate $W^{(l+1)} = \pi_P(S^{(l)} - \frac{1}{\gamma^{(l)}}\mathcal{L}'_I(S^{(l)}))$;
    Calculate $Q_\gamma(S^{(l)}, W^{(l+1)})$;
    if $\mathcal{L}_I(W^{(l+1)}) \leq Q_\gamma(S^{(l)}, W^{(l+1)})$ **then**
      $\gamma^{(l)} = \gamma$, **break** ;
    end
  end
  $d_l = \frac{1+\sqrt{1+4d_{l-1}^2}}{2}$;
  $l = l + 1$;
until Convergence of $W^{(l)}$;
$\hat{W} = W^{(l)}$;

Optimization of ϑ With fixed W, the optimal ϑ can be learned by solving $\min_\vartheta \mathcal{L}_I(\vartheta)$, where $\mathcal{L}_I(\vartheta)'$s first order derivative can be expressed as:

$$\begin{aligned}\mathcal{L}'_I(\vartheta_t) &= \sum_{j=1}^{n_t} \sum_{Y_{tj}-1=\mu}^{U} G(\vartheta_{t\mu} - X_{tj}W_t) \\ &\quad - \sum_{j=1}^{n_t} \sum_{Y_{tj}=\mu}^{U} G(X_{tj}W_t - \vartheta_{t\mu}), \\ \mathcal{L}'_I(\vartheta) &= \left[\frac{\mathcal{L}'_I(\vartheta_1)}{n_1}, \cdots, \frac{\mathcal{L}'_I(\vartheta_t)}{n_t}, \cdots, \frac{\mathcal{L}'_I(\vartheta_T)}{n_T}\right], \end{aligned} \tag{11}$$

where $\vartheta_{t\mu}$ is the μ^{th} threshold in task t, so that ϑ can be updated as:

$$\vartheta^{(l)} = \vartheta^{(l-1)} - \varepsilon^{(l)}\mathcal{L}'_I(\vartheta), \tag{12}$$

where ε is the step-size of gradient descent.

4.3 RMTOR using all thresholds (***RMTOR$_A$***)

Alternatively, we describe another possible way of formulating the loss function for ordinal regression, so-called all thresholds (Fig. 2), and use it as a strong baseline to compare with the loss function formulated using adjacent thresholds only.

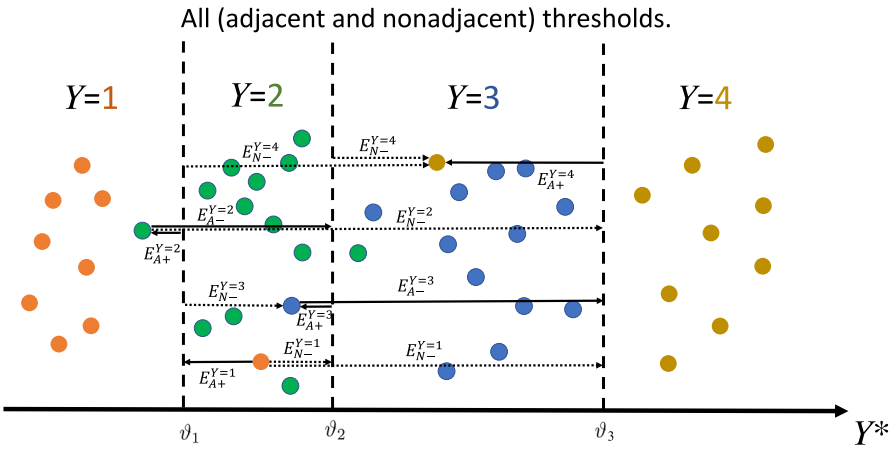

Fig. 2 Illustration of the **all-thresholds loss** using four segments that calculate the error using both neighbor/adjacent and non-neighbor/non-adjacent thresholds. We denote $E_{A+/-}^{Y=\mu}$ and $E_{N+/-}^{Y=\mu}$ as the error for a data point in the class μ, where A and N represent adjacent thresholds and non-adjacent used, respectively. In addition to Fig. 1, solid lines represent the errors calculated using adjacent thresholds, while dash lines represent the errors calculated using non-adjacent thresholds. Same as Fig. 1, $+$ or $-$ indicates the error value is positive or negative and the different direction of the arrow lines indicate the error direction. Due to the loss functions are different in immediate and all thresholds, the errors are also different in Fig. 1 and Fig. 2. For example, $E_{A+}^{Y=1}$ denotes the error of a class 1 data point using adjacent threshold that equals to $X_{tj}^{Y=1}W_t - \vartheta_1$; this error is represented with a left direction arrow line in Fig. 2 and as ϑ_1 is smaller than $X_{tj}^{Y=1}W_t$, so its value is positive. There are two $E_{N-}^{Y=1}$ in Fig. 2 denoting the errors of a class 1 data point using non-adjacent threshold that equal to $X_{tj}^{Y=1}W_t - \vartheta_2$ and $X_{tj}^{Y=1}W_t - \vartheta_3$, respectively; these two errors are represented with two right direction arrow dash lines in Fig. 2 and as ϑ_2 and ϑ_3 are smaller than $X_{tj}^{Y=1}W_t$, so their values are negative. Note that, in Eq. (13), the errors for data points in each class are calculated summing over from $\mu = 1$ to $U - 1$, so that $\vartheta = 0$ and $\vartheta = 4$ are not presented in Fig. 2

4.3.1 *RMTOR$_A$* model

RMTOR with the all thresholds, loss function is calculated as:

$$\mathcal{L}_A = \sum_{t=1}^{T}\sum_{j=1}^{n_t}\left[\sum_{\mu=1}^{Y_{tj}-1} M(\vartheta_{t\mu} - X_{tj}W_t) + \sum_{\mu=Y_{tj}}^{U-1} M(X_{tj}W_t - \vartheta_{t\mu})\right], \tag{13}$$

where $\sum_{\mu=1}^{Y_{tj}-1} M(X_{tj}W_t - \vartheta_{t\mu})$ is the sum of errors when $\mu < Y_{tj}$, which means the threshold's index μ is smaller than the j^{th} training instance label Y_{tj}, while $\sum_{\mu=Y_{tj}}^{U-1} M(\vartheta_{t\mu} - X_{tj}W_t)$ is the sum of errors when $\mu \geq Y_{tj}$. To visualize our all thresholds method, we show an illustration figure in Fig. 2.

Thus, its objective function $RMTOR_A$ is calculated as:

$$RMTOR_A = \min_{W,\vartheta} \sum_{t=1}^{T} \sum_{j=1}^{n_t} \left[\sum_{\mu=1}^{Y_{tj}-1} M(\vartheta_{t\mu} - X_{tj} W_t) + \sum_{\mu=Y_{tj}}^{U-1} M(X_{tj} W_t - \vartheta_{t\mu}) \right] + \lambda ||W||_{2,1}. \tag{14}$$

4.3.2 Optimization

We also implement an alternating structure optimization method to obtain the optimal parameters W and ϑ, which is similar as we perform for $RMTOR_I$ optimization.

Optimization of **W** With fixed ϑ, the optimal W can be learned by solving:

$$\min_{W} \mathcal{L}_A(W) + \lambda ||W||_{2,1}, \tag{15}$$

where $\mathcal{L}_A(W)$ is a smooth convex and differentiable loss function. First, we calculate its first order derivative w.r.t. W_t:

$$\mathcal{L}'_A(W_t) = \sum_{j=1}^{n_t} \left[\sum_{\mu=Y_{tj}}^{U-1} X_{tj} G(X_{tj} W_t - \vartheta_{t\mu}) - \sum_{\mu=1}^{Y_{tj}-1} X_{tj} G(\vartheta_{t\mu} - X_{tj} W_t) \right]. \tag{16}$$

We introduce an indicator variable z_μ:

$$z_\mu = \begin{cases} +1, & \mu \geq Y_{tj} \\ -1, & \mu < Y_{tj} \end{cases} \tag{17}$$

Then the updated formulation of Eq. (16) and the first order derivative w.r.t. W are calculated as:

$$\begin{aligned} \mathcal{L}'_A(W_t) &= \sum_{j=1}^{n_t} \sum_{\mu=1}^{U-1} X_{tj}^T \left[z_\mu \cdot G\left(z_\mu \cdot (X_{tj} W_t - \vartheta_{t\mu}) \right) \right], \\ \mathcal{L}'_A(W) &= \left[\frac{\mathcal{L}'_A(W_1)}{n_1}, \cdots, \frac{\mathcal{L}'_A(W_t)}{n_t}, \cdots, \frac{\mathcal{L}'_A(W_T)}{n_T} \right]. \end{aligned} \tag{18}$$

Similar as we did for $RMTOR_I$ optimization of W, we then use FISTA to optimize with the parameters in $RMTOR_A$ updating steps:

$$W^{(l+1)} = \pi_P(S^{(l)} - \frac{1}{\gamma^{(l)}} \mathcal{L}'_A(S^{(l)})), \tag{19}$$

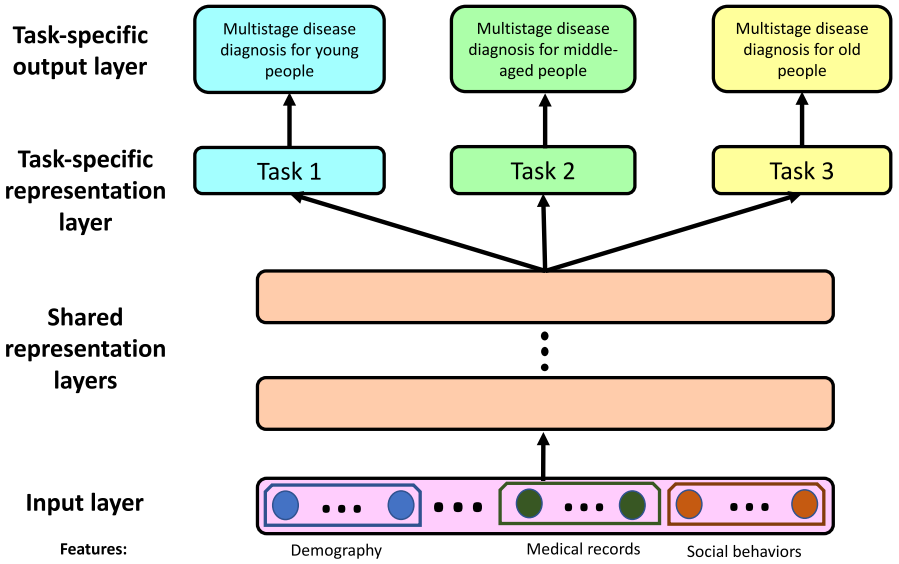

Fig. 3 Illustration of DNN based multi-task ordinal regression (DMTOR). All tasks share the input and representation layers, while all tasks keep several task-specific layers. Note that, circles represent the nodes at each layer and squares represent layers

which is solved in Algorithm 1.

Optimization of ϑ With fixed W, the optimal ϑ can be learned by solving $\min_{\vartheta} \mathcal{L}_A(\vartheta)$, where $\mathcal{L}_A(\vartheta)$'s first order derivative can be expressed as:

$$\begin{aligned} \mathcal{L}'_A(\vartheta_t) &= -\mathbf{1}^T \left[z_\mu \cdot G\left(z_\mu \cdot (X_{tj} W_t - \vartheta_{t\mu}) \right) \right], \\ \mathcal{L}'_A(\vartheta) &= \left[\frac{\mathcal{L}'_A(\vartheta_1)}{n_1}, \cdots, \frac{\mathcal{L}'_A(\vartheta_t)}{n_t}, \cdots, \frac{\mathcal{L}'_A(\vartheta_T)}{n_T} \right], \end{aligned} \tag{20}$$

and hence ϑ can be updated as:

$$\vartheta^{(l)} = \vartheta^{(l-1)} - \varepsilon^{(l)} \mathcal{L}'_A(\vartheta). \tag{21}$$

5 Deep multi-task ordinal regression (DMTOR) models

In this section, we introduce two deep multi-task ordinal regression (DMTOR) models implemented using deep neural networks (DNN). Fig. 3 illustrates the basic architecture of the DMTOR.

5.1 DMTOR architecture

We denote input layer, shared representation layers and task-specific representation layers as L_1, $L_{(R\cdot)}$ and $L_{(S\cdot)}$, respectively. Thus, we have the shared representation

layers as:

$$\begin{aligned} L_{R(1)} &= ReLU(W_1 \cdot L_1), \\ L_{R(2)} &= ReLU(W_2 \cdot L_{R(1)}), \\ &\cdots, \\ L_{R(r)} &= f(W_r, L_{R(r-1)}), \end{aligned} \tag{22}$$

where $\{W_1, \cdots, W_r\}$ are the coefficient parameters at different hidden layers, $ReLU(\cdot)$ stands for rectified linear unit that is the nonlinear activation function, r is the number of hidden layers and $f(\cdot)$ is a linear transformation.

Task-specific representation layers are expressed as:

$$\begin{aligned} L^t_{S(1)} &= ReLU(B^t_1 \cdot L_{R(r)}), \\ &\cdots, \\ L^t_{S(s)} &= ReLU(B^t_s \cdot L_{S(s-1)}), \end{aligned} \tag{23}$$

where B^t is the coefficient parameter corresponding to the t^{th} task and s is the number of task-specific representation layers.

5.2 Network training

Forward propagation calculation for the output is expressed as:

$$output^t = f(O^t, L^t_{S(s)}), \tag{24}$$

where O^t is the coefficient parameter corresponding to the t^{th} task.

Then the loss function of $DMTOR_I$ model can be calculated as:

$$\begin{aligned} \mathcal{L}_I = \sum_{t=1}^{T} \sum_{j=1}^{n_t} [& M(\vartheta_{(Y_{tj}-1)} - output^t) \\ & + M(output^t - \vartheta_{Y_{tj}})]. \end{aligned} \tag{25}$$

Similarly, the loss function of $DMTOR_A$ model can be calculated as:

$$\begin{aligned} \mathcal{L}_A = \sum_{t=1}^{T} \sum_{j=1}^{n_t} [& \sum_{\mu=1}^{Y_{tj}-1} M(\vartheta_{t\mu} - output^t) \\ & + \sum_{\mu=Y_{tj}}^{U-1} M(output^t - \vartheta_{t\mu})]. \end{aligned} \tag{26}$$

We use mini-batches to train our models' parameters for faster learning with partitioning the training dataset into small batches, and then calculate the model error and update the corresponding parameters.

Stochastic Gradient Descent (SGD) is used to iteratively minimize the loss and update all the model parameters (weights: W, B, O and thresholds: ϑ):

$$\begin{aligned} W^{(l)} &= W^{(l-1)} - \varepsilon^{(l)} \nabla_W \mathcal{L}, \\ &\cdots, \\ \vartheta^{(l)} &= \vartheta^{(l-1)} - \varepsilon^{(l)} \nabla_\vartheta \mathcal{L}. \end{aligned} \tag{27}$$

6 Experiments and results

To evaluate the performance of our proposed multi-task ordinal regression (MTOR) models, we extensively compare them with a set of selected single-task learning (STL) models. We first elaborate some details of the experimental setup and then describe three real-world medical datasets used in the experiments. Finally, we discuss the experimental results using accuracy and mean absolute error (MAE) as the evaluation metrics.

6.1 Experimental setup

We demonstrate the performance of proposed RMTOR and DMTOR models on small and large-scale medical datasets, respectively: 1). We use a small dataset (i.e., Alzheimer's Disease Neuroimaging Initiative) to experimentally compare $RMTOR_I$ and $RMTOR_A$ with their corresponding STL ordinal regression models denoted as $STOR_I$ and $STOR_A$. We also compare them with two SVM based ordinal regression (SVOR) models, i.e., support vector for ordinal regression with explicit constraints ($SVOREC$) (Chu and Keerthi 2007) and support vector machines using binary ordinal decomposition ($SVMBOD$) (Frank and Hall 2001). Both SVOR models are implemented in Matlab within $ORCA$ framework (Gutiérrez et al. 2016). 2). Our experiments on two large-scale healthcare datasets (i.e., Behavioral Risk Factor Surveillance System and Henry Ford Hospital hypertension) compare $DMTOR_I$ and $DMTOR_A$ with their corresponding STL ordinal regression models denoted as $DSTOR_I$ and $DSTOR_A$. In addition, we compare them with a neural network approach for ordinal regression, i.e., $NNRank$ (Cheng et al. 2008), which is downloaded from the $Multicom$ toolbox[2]. In our experiments, the models with DNN (i.e., $DMTOR_I$, $DMTOR_A$, $DSTOR_I$ and $DSTOR_A$) are implemented in Python using Pytorch and the other models without DNN ($RMTOR_I$, $RMTOR_A$, $STOR_I$ and $STOR_A$) are implemented in Matlab.

[2] http://sysbio.rnet.missouri.edu/multicom_toolbox/tools.html

Table 1 The accuracy of our proposed regularized MTOR model, i.e., $\boldsymbol{RMTOR_I}$ and compared with an alternative formulation $\boldsymbol{RMTOR_A}$, the corresponding single-task ordinal regression models (i.e., $\boldsymbol{STOR_I}$ and $\boldsymbol{STOR_A}$) and two SVM based STL ordinal regression models (i.e., $\boldsymbol{SVOREC}$ and $\boldsymbol{SVMBOD}$) using a small healthcare dataset, i.e., ADNI. Note that, standard deviations are shown at the second row in each cell that is under the accuracy. The first and second columns represent the age group (AG) of each task and number of instances in each task of testing dataset, respectively. The best performance results are in bold face

Task/	No. of	MTOR		Global setting				Individual setting			
AG	instances	$RMTOR_I$	$RMTOR_A$	$SVOREC$	$SVMBOD$	$STOR_I$	$STOR_A$	$SVOREC$	$SVMBOD$	$STOR_I$	$STOR_A$
50-59	72	**0.791**	0.783	0.572	0.522	0.493	0.489	0.554	0.633	0.473	0.459
		±0.055	±0.09	±0.08	±0.049	±0.04	±0.12	±0.065	±0.105	±0.05	±0.115
60-69	104	**0.739**	0.687	0.583	0.611	0.429	0.493	0.638	0.621	0.633	0.656
		±0.14	±0.02	±0.05	±0.072	±0.112	±0.018	±0.035	±0.046	±0.033	±0.081
70-79	142	**0.764**	0.659	0.533	0.661	0.572	0.478	0.602	0.645	0.674	0.629
		±0.218	±0.019	±0.255	±0.047	±0.023	±0.061	±0.038	±0.041	±0.029	±0.078
≥ 80	83	**0.747**	0.709	0.623	0.671	0.523	0.475	0.693	0.701	0.677	0.616
		±0.015	±0.09	±0.12	±0.04	±0.09	±0.031	±0.037	±0.044	±0.016	±0.03

Table 2 The MAE of our proposed regularized MTOR model, i.e., $\boldsymbol{RMTOR_I}$, compared with an alternative formulation $\boldsymbol{RMTOR_A}$, their corresponding STL ordinal regression models and two SVM based STL ordinal regression models using a small healthcare dataset, i.e., ADNI

Task/	No. of	MTOR		Global setting				Individual setting			
AG	instances	$RMTOR_I$	$RMTOR_A$	$SVOREC$	$SVMBOD$	$STOR_I$	$STOR_A$	$SVOREC$	$SVMBOD$	$STOR_I$	$STOR_A$
50-59	72	0.344	**0.307**	0.673	0.691	0.683	0.629	0.537	0.501	0.792	0.690
		±0.009	±0.022	±0.013	±0.087	±0.103	±0.028	±0.039	±0.043	±0.182	±0.207
60-69	104	**0.311**	0.362	1.014	0.892	1.033	1.098	0.911	0.837	0.894	1.063
		±0.005	±0.093	±0.088	±0.049	±0.052	±0.132	±0.095	±0.105	±0.077	±0.207
70-79	142	**0.401**	0.561	0.943	0.798	0.743	0.832	0.601	0.592	0.611	0.975
		±0.048	±0.05	±0.073	±0.082	±0.117	±0.131	±0.128	±0.092	±0.057	±0.155
≥ 80	83	**0.579**	0.619	0.912	0.593	0.840	0.983	0.812	0.727	0.930	1.091
		±0.051	±0.039	±0.133	±0.094	±0.078	±0.098	±0.109	±0.207	±0.118	±0.257

Table 3 The accuracy of the proposed DNN based MTOR model, i.e., $\boldsymbol{DMTOR_I}$, the alternative formulation $\boldsymbol{DMTOR_A}$, their corresponding STL ordinal regression models (i.e., $\boldsymbol{DSTOR_I}$ and $\boldsymbol{DSTOR_A}$) and a STL neural network approach for ordinal regression (i.e., ***NNRank***) using a large-scale medical dataset , i.e., BRFSS

Task/	No. of	MTOR		Global setting			Individual setting		
AG	instances	$DMTOR_I$	$DMTOR_A$	$NNRank$	$DSTOR_I$	$DSTOR_A$	$NNRank$	$DSTOR_I$	$DSTOR_A$
18-24	5,325	**0.532**	0.431	0.525	0.405	0.363	0.507	0.359	0.328
		±0.037	±0.071	±0.095	±0.039	±0.058	±0.009	±0.073	±0.098
25-34	5,693	**0.524**	0.452	0.521	0.432	0.379	0.513	0.325	0.389
		±0.052	±0.037	±0.112	±0.094	±0.075	±0.11	±0.046	±0.091
35-49	17,480	**0.577**	0.513	0.574	0.455	0.381	0.563	0.367	0.328
		±0.089	±0.076	±0.034	±0.078	±0.054	±0.093	±0.061	±0.052
50-79	55,388	**0.608**	0.529	0.580	0.421	0.276	0.585	0.293	0.284
		±0.101	±0.097	±0.063	±0.051	±0.077	±0.067	±0.035	±0.029
≥ 80	745	**0.451**	0.443	0.447	0.410	0.391	0.425	0.394	0.374
		±0.091	±0.085	±0.058	±0.039	±0.022	±0.081	±0.048	±0.053

6.1.1 MTL ordinal regression experimental setup

In the three real-world datasets, tasks are all defined based on various age groups in terms of the predefined age groups in MTOR models for the consistency. Also, all tasks share the same feature space, which follows the assumption of MTL that the multiple tasks are related.

For $RMTOR_I$ and $RMTOR_A$, we use 10-fold cross validation to select the best tuning parameter λ in the training dataset.

For $DMTOR_I$ and $DMTOR_A$, we use the same setting of DNN, i.e., three shared representations layers and three task-specific representation layers. For each dataset, we set the same hyper-parameters, e.g., number of batches and number of epochs; while these hyper-parameters are not the same in different datasets. We use random initialization for parameters. Please refer to Sect. 5.2 to see the details of the network training procedures.

6.1.2 STL ordinal regression experimental setup

In our experiments, STL ordinal regression methods are applied under two settings: 1) Individual setting, i.e., a prediction model is trained for each task; 2) Global setting, i.e., a prediction model is trained for all tasks. In the individual setting the heterogeneity among tasks are fully considered but not the task relatedness; on the contrary, in the global setting all the heterogeneities have been neglected.

For $DSTOR_I$ and $DSTOR_A$, the setting of DNN uses three hidden representation layers, where each layer's activation function is $ReLU(\cdot)$. During the training procedure, the loss functions use the same function $M(\cdot)$ with either immediate or all thresholds. Same as we did for DMTOR, we set the same hyper-parameters within each dataset and different ones among different datasets.

In the training of $NNRank$, we use the default setting, .e.g., number of epochs is 500, random seed is 999 and learning rate is 0.01. In testing, we also use the default setting, e.g., decision threshold is 0.5.

6.2 Data description

In this paper, Alzheimer's Disease Neuroimaging Initiative (ADNI) (Mueller et al. 2005) and Behavioral Risk Factor Surveillance System (BRFSS) are public medical benchmark datasets, while Henry Ford Hospital hypertension (FORD) is the private one. We divide these three datasets into training and testing using stratified sampling, more specifically, 80% of instances are used for training and the rest of instances are used for testing.

Age is a crucial factor when considering phenotypic changes in disease (Buja et al. 2014; Duricova et al. 2014; Westbrook and Viney 1983; Geifman et al. 2013). Thus, we define the tasks according to the disjoint age groups in ADNI, BRFSS and FORD datasets.

Table 4 The MAE of the proposed DNN based MTOR model, the alternative formulation $\boldsymbol{DMTOR_A}$, their corresponding STL models and ***NNRank*** using a large-scale BRFSS dataset

Task/	No. of	MTOR		Global setting			Individual setting		
AG	instances	$DMTOR_I$	$DMTOR_A$	$NNRank$	$DSTOR_I$	$DSTOR_A$	$NNRank$	$DSTOR_I$	$DSTOR_A$
18-24	5,325	**0.479**	0.582	0.793	0.783	1.020	0.802	0.745	1.055
		±0.071	±0.059	±0.037	±0.095	±0.107	±0.092	±0.093	±0.111
25-34	5,693	**0.521**	0.633	0.573	0.795	0.839	0.581	0.935	1.037
		±0.058	±0.079	±0.082	±0.094	±0.105	±0.057	±0.034	±0.125
35-49	17,480	**0.755**	0.924	0.915	1.090	0.927	0.790	0.954	1.077
		±0.102	±0.115	±0.059	±0.11	±0.098	±0.055	±0.072	±0.092
50-79	55,388	**0.536**	0.711	0.875	1.330	1.033	0.582	1.503	1.270
		±0.088	±0.042	±0.089	±0.107	±0.122	±0.197	±0.106	±0.14
≥ 80	745	**0.630**	0.681	0.833	0.961	0.902	0.710	1.027	1.009
		±0.108	±0.102	±0.133	±0.079	±0.082	±0.124	±0.21	±0.095

Table 5 The accuracy of the proposed DNN based MTOR models, their corresponding STL models and ***NNRank*** using a large-scale FORD dataset

Task/ AG	No. of instances	MTOR		Global setting			Individual setting		
		$DMTOR_I$	$DMTOR_A$	$NNRank$	$DSTOR_I$	$DSTOR_A$	$NNRank$	$DSTOR_I$	$DSTOR_A$
0-17	4,176	**0.732**	0.709	0.451	0.532	0.588	0.455	0.577	0.591
		±0.13	±0.105	±0.058	±0.092	±0.078	±0.074	±0.039	±0.102
18-24	5,284	**0.742**	0.697	0.551	0.530	0.592	0.479	0.635	0.671
		±0.085	±0.032	±0.049	±0.051	±0.069	±0.071	±0.083	±0.097
25-34	6,279	**0.722**	0.720	0.488	0.497	0.593	0.452	0.622	0.530
		±0.056	±0.072	±0.035	±0.042	±0.038	±0.092	±0.055	±0.094
35-49	9,516	**0.781**	0.737	0.667	0.649	0.563	0.619	0.620	0.565
		±0.081	±0.09	±0.033	±0.047	±0.04	±0.85	±0.029	±0.058
50-79	10,991	**0.755**	0.734	0.615	0.534	0.530	0.598	0.616	0.613
		±0.096	±0.075	±0.08	±0.09	±0.073	±0.069	±0.084	±0.106
≥ 80	1,070	**0.737**	0.733	0.690	0.570	0.539	0.658	0.609	0.579
		±0.089	±0.083	±0.036	±0.095	±0.047	±0.05	±0.104	±0.035

Table 6 The MAE of the proposed DNN based MTOR models, their corresponding STL models and ***NNRank*** using a large-scale FORD dataset

Task/ AG	No. of instances	MTOR		Global setting			Individual setting		
		$DMTOR_I$	$DMTOR_A$	$NNRank$	$DSTOR_I$	$DSTOR_A$	$NNRank$	$DSTOR_I$	$DSTOR_A$
0-17	4,176	**0.277**	0.303	0.654	0.745	0.894	0.531	0.845	0.919
		±0.007	±0.021	±0.008	±0.039	±0.089	±0.091	±0.013	±0.087
18-24	5,284	**0.298**	0.401	0.537	0.639	0.792	0.938	0.862	0.583
		±0.025	±0.028	±0.034	±0.023	±0.058	±0.086	±0.079	±0.093
25-34	6,279	**0.435**	0.539	0.680	1.032	0.794	0.902	0.883	0.895
		±0.061	±0.077	±0.062	±0.095	±0.054	±0.075	±0.098	±0.086
35-49	9,516	**0.301**	0.350	0.548	0.642	1.055	0.720	0.860	0.930
		±0.027	±0.019	±0.025	±0.092	±0.179	±0.032	±0.046	±0.071
50-79	10,991	0.379	**0.351**	0.537	0.665	0.995	0.850	0.990	1.034
		±0.039	±0.059	±0.024	±0.048	±0.064	±0.076	±0.096	±0.19
≥ 80	1,070	**0.383**	0.412	0.731	0.790	1.077	0.609	1.073	0.977
		±0.03	±0.052	±0.083	±0.078	±0.12	±0.065	±0.14	±0.098

6.2.1 Alzheimer's disease neuroimaging initiative (ADNI)

The mission of ADNI is to seek the development of biomarkers for the disease and advance in order to understand the pathophysiology of AD (Mueller et al. 2005). This data also aims to improve diagnostic methods for early detection of AD and augment clinical trial design. Additional goal of ADNI is to test the rate of progress for both mild cognitive impairment and AD. As a result, ADNI are trying to build a large repository of clinical and imaging data for AD research.

We pick one measurement from the participants of diagnostic file in this project and delete two participants whose age information are missing, which leaves us 1, 998 instances and 95 variables including 94 input variables that are corresponding to measurement of AD, e.g., FDG-PET is used to measure cerebral metabolic rates of glucose; plus one output variable that is phase used to represent three stages of AD (cognitively normal, mild cognitive impairment, and AD).

Since the age groups in ADNI dataset fall in mature adulthood and late adulthood, we divide mature adulthood into three subgroups. Hence, the tasks are defined in ADNI based on different stages of people shown as the first column in Tables 1 and 2, i.e., mature adulthood 1 (50 years to 59 years), mature adulthood 2 (60 years to 69 years), mature adulthood 3 (70 years to 79 years) and late adulthood (equal or older than 80 years).

6.2.2 Behavioral risk factor surveillance system (BRFSS)

The BRFSS dataset is a collaborative project between all the states in the U.S. and the Centers for Disease Control and Prevention (CDC), and aims to collect uniform, state-specific data on preventable health practices and risk behaviors that affect the health of the adult population (i.e., adults aged 18 years and older). In the experiment, we use the BRFSS dataset that is collected in 2016[3].

The BRFSS dataset is collected via the phone-based surveys with adults residing in private residence or college housing. The original BRFSS dataset contains 486, 303 instances and 275 variables, after deleting the entries with missing age information and the variables with all hidden values, the preprocessed dataset contains 459, 156 with 85 variables including 84 input variables and one output variable, i.e., categories of body mass index (underweight, normal weight, overweight and obese).

The tasks are defined in BRFSS based on different stages of people shown in the first column in Tables 3 and 4, i.e., early young (18 years to 24 years), young (25 years to 34 years), middle-aged (35 years to 49 years), mature adulthood (50 years to 70 years) and late adulthood (equal or older than 80 years).

6.2.3 Henry ford hospital hypertension (FORD)

FORD dataset is collected by our collaborator from Emergency Room (ER) of Henry Ford Hospital. All participants in this dataset are all from metro Detroit. All variables except for the outcomes are collected from the emergency department at Henry Ford

[3] https://www.cdc.gov/brfss/annual_data/annual_2016.html

Hospital. Some diagnostic variables are collected from any hospital admissions that occurred after the ER visits. The index date in FORD dataset for each patient started in 2014 and went through the middle of 2015. They then collect outcomes for each patient for one year after that index date. So, the time duration from the date that a patient seen in ER to his/her diagnostic variable collection date may be longer than one year. For example, a patient may have been seen in the ER on July 2, 2015 and they would have had diagnosis variable collected date up to July 2, 2016.

Originally, this FORD dataset contains 221, 966 instances and 63 variables including demographic, lab test and diagnosis related information. After deleting the entries with missing values, the preprocessed dataset contains 186, 572 instances and 23 variables including 22 input variables and one output, i.e., four stages of hypertension based on systolic and diastolic pressure: normal (systolic pressure: 90-119 and diastolic pressure: 60-79), pre-hypertension (120-139 and 80-89), stage 1 hypertension (140-159 and 90-99) and stage 2 hypertension ($\geq$ 160 and $\geq$ 160).

Since the number of instances in the age groups of infant, children and teenager are much less than other age groups, we combine these three age groups into one age group as minor. Hence, the tasks are defined in FORD based on different ages of people shown as the first column in Tables 5 and 6, i.e., minor (1 year to 17 years), early young (18 years to 24 years), young (25 years to 34 years), middle-aged (35 years to 49 years), mature adulthood (50 years to 70 years) and late adulthood (equal or older than 80 years).

6.3 Performance comparison

To evaluate the overall performance of each ordinal regression method, we use both accuracy and MAE as our evaluation metrics. Accuracy reports the proportion of accurate predictions, so that larger value of accuracy means better performance. With considering orders, MAE is capable of measuring the distance between true and predicted labels, so that smaller value of MAE means better performance.

To formally define accuracy, we use i and j to represent the index of true labels and the index of predicted labels. A pair of labels for each instance, i.e., $(Y_i, \hat{Y}_j)$, is positive if they are equal, i.e., $Y_i = \hat{Y}_j$, otherwise the pair is negative. We further denote N_T as the number of total pairs and N_P as the number of positive pairs. Thus, $accuracy = \frac{N_P}{N_T}$. MAE is calculated as $MAE = \frac{\sum_{i=1}^{n_s} |Y_i - \hat{Y}_i|}{n_s}$, where n_s is the number of instances in each testing dataset.

We show the performance results of prediction accuracy of different models along with their standard deviations using the aforementioned three medical datasets ADNI, BRFSS and FORD in Tables 1, 3 and 5, respectively. We also present the performance results of MAE of different models along with their standard deviations using the aforementioned three medical datasets ADNI, BRFSS and FORD in Tables 2, 4 and 6, respectively. Each task in our experiments is to predict the stage of disease for people in each age group. In the experiments of MTOR models, each task has its own prediction result. For each task, we build one STL ordinal regression model under the global and individual settings as comparison methods.

Overall, the experimental results show that the MTOR models perform better than other STL models in terms of both accuracy and MAE. MTOR models outperform STL ones across all the tasks in each dataset. MTOR models with immediate thresholds largely outperform the ones with all thresholds in both evaluation metrics, which confirms the assumption that first and last thresholds are always remaining in finite range in the real-world scenario.

Under the proposed MTOR framework, both deep and shallow models have descent performance for different types of datasets: RMTOR model with immediate thresholds performs better for small dataset whereas DMTOR model with immediate thresholds is more suitable for large-scale dataset. More specifically, the $DMTOR_I$ model outperforms the competing models in the most tasks of BRFSS and FORD datasets. In ADNI dataset, $RMTOR_I$ outperforms other models in terms of accuracy and MAE. Note that, the accuracy and MAE do not always perform consistently for all tasks. For example in the experiment using ADNI dataset, for the first task with ages ranging in (50-59), $RMTOR_I$ shows the best (largest) accuracy whereas $RMTOR_A$ exhibits the best (lowest) MAE.

For SVM based STL ordinal regression models, the distance between classes is unknown in this type of methods, the real values used for the labels may undermine regression performance. Moreover, these regression learners are sensitive to the label representation instead of their orders. While our MTOR models with predefining margin function that utilizes shared information between tasks can overcome the aforementioned shortcomings.

7 Conclusion

In this paper, we tackle multiple ordinal regression problem by proposing a regularized MTOR model for smaller data sets and a DNN based MTOR model for large-scale data sets. The former belongs to the regularized multi-task learning, where the ordinal regression is used to handle the ordinal labels and regularization terms are used to encode the assumption of task relatedness. The latter is based on DNN with shared representation layers to encode the task relatedness. Particularly, the DNN based MTOR outperforms other models for the large-scale datasets and the regularized MTOR are appropriate for small datasets. In the future, we plan to develop a weighted loss function for MTOR using both immediate and all thresholds in one unified function.

Acknowledgements This paper is based upon work supported by the National Science Foundation under grants CNS-1637312 and CCF-1451316.

References

Ando RK, Zhang T (2005) A framework for learning predictive structures from multiple tasks and unlabeled data. J Machine Learn Res 6:1817–1853

Argyriou A, Evgeniou T, Pontil M (2008) Convex multi-task feature learning. Machine Learn 73(3):243–272

Baetschmann G, Staub KE, Winkelmann R (2015) Consistent estimation of the fixed effects ordered logit model. J Royal Statistical Soc: Series A (Statistics Soc) 178(3):685–703

Baxter J (1997) A bayesian/information theoretic model of learning to learn via multiple task sampling. Machine learn 28(1):7–39
Beck A, Teboulle M (2009) A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J Imag sci 2(1):183–202
Cruickshank TM, Reyes AR, Ziman MR (2015) A systematic review and meta-analysis of strength training in individuals with multiple sclerosis or parkinson disease. Medicine 94:4
Brookmeyer R, Johnson E, Ziegler-Graham K, Arrighi HM (2007) Forecasting the global burden of alzheimer's disease. Alzheimer's & dementia: J Alzheimer's Assoc 3(3):186–191
Buja A, Damiani G, Gini R, Visca M, Federico B, Donato D, Francesconi P, Marini A, Donatini A, Brugaletta S et al (2014) Systematic age-related differences in chronic disease management in a population-based cohort study: a new paradigm of primary care is required. PLoS One 9(3):e91340
Grosskreutz H, Rüping S (2009) On subgroup discovery in numerical domains. Data min knowl discov 19(2):210–226
Chan DS, Norat T (2015) Obesity and breast cancer: not only a risk factor of the disease. Current treat opt oncol 16(5):22
Cheng J, Wang Z, Pollastri G (2008) A neural network approach to ordinal regression, in Neural Networks, IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on. IEEE 2008:1279–1284
Hamidi D. Yar, Wennberg K, Berglund H (2008) Creativity in entrepreneurship education. J small bus enterp dev 15(2):304–320
Chu W, Keerthi SS (2007) Support vector ordinal regression. Neural comput 19(3):792–815
Liu Y, Kong A. W.-K, Goh C. K (2017) "Deep ordinal regression based on data relationship for small datasets." in *IJCAI*, pp. 2372–2378
Cruickshank TM, Reyes AR, Ziman MR (2015) A systematic review and meta-analysis of strength training in individuals with multiple sclerosis or parkinson disease. Medicine 94:4
Cruz GD, Galvis DL, Kim M, Le-Geros RZ, Barrow S-YL, Tavares M, Bachiman R (2001) Self-perceived oral health among three subgroups of asian-americans in new york city: a preliminary study. Commun dent oral epidemiol 29(2):99–106
Davis DA, Chawla NV, Christakis NA, Barabási A-L (2010) Time to care: a collaborative engine for practical disease prediction. Data Min Knowl Discov 20(3):388–415
Domingo-Ferrer J, Torra V (2005) Ordinal, continuous and heterogeneous k-anonymity through microaggregation. Data Min Knowl Discov 11(2):195–212
Lanfranchi M, Giannetto C, Zirilli A, Alibrandi A (2014) Analysis of the demand of wine in sicily through ordinal logistic regression model. Calitatea 15(139):87
Duricova D, Burisch J, Jess T, Gower-Rousseau C, Lakatos PL (2014) ECCO-EpiCom, & Age-related differences in presentation and course of inflammatory bowel disease an update on the population-based literature. Journal of Crohn's and Colitis 8(11):1351–1361
Kato T, Kashima H, Sugiyama M, Asai K (2008) "Multi-task learning via conic programming," in *Advances in Neural Information Processing Systems*, pp. 737–744
Park S-H, Fürnkranz J (2012) Efficient prediction algorithms for binary decomposition techniques. Data Min Knowl Discov 24(1):40–77
Har-Peled S, Roth D, Zimak D, (2002) "Constraint classification: A new approach to multiclass classification and ranking," in *In Advances in Neural Information Processing Systems 15*. Citeseer,
Gursoy ME, Inan A, Nergiz ME, Saygin Y (2017) Differentially private nearest neighbor classification. Data Min Knowl Discov 31(5):1544–1575
Geifman N, Cohen R, Rubin E (2013) Redefining meaningful age groups in the context of disease. Age 35(6):2357–2366
Grosskreutz H, Rüping S (2009) On subgroup discovery in numerical domains. Data min knowl discov 19(2):210–226
Gu B, Sheng VS, Tay KY, Romano W, Li S (2015) Incremental support vector learning for ordinal regression. IEEE Trans Neural netw learn syst 26(7):1403–1416
Gursoy ME, Inan A, Nergiz ME, Saygin Y (2017) Differentially private nearest neighbor classification. Data Min Knowl Discov 31(5):1544–1575
Gutiérrez PA, Perez-Ortiz M, Sanchez-Monedero J, Fernandez-Navarro F, Hervas-Martinez C (2016) Ordinal regression methods: survey and experimental study. IEEE Trans Knowl Data Eng 28(1):127–146

Schmidt-Richberg A, Guerrero R, Ledig C, Molina-Abril H, Frangi A. F, Rueckert D, Initiative A. D. N *et al.*, (2015) "Multi-stage biomarker models for progression estimation in alzheimer's disease," in *International Conference on Information Processing in Medical Imaging*. Springer, pp. 387–398

Gu B, Sheng VS, Tay KY, Romano W, Li S (2015) Incremental support vector learning for ordinal regression. IEEE Trans Neural netw learn syst 26(7):1403–1416

Henriques R, Madeira SC, Antunes C (2015) Multi-period classification: learning sequent classes from temporal domains. Data Min Knowl Discov 29(3):792–819

Hong HG, He X (2010) Prediction of functional status for the elderly based on a new ordinal regression model. J Am Statistical Assoc 105(491):930–941

Wang L, Dong M, Towner E, Zhu D (2019) "Prioritization of multi-level risk factors for obesity," in *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, pp. 1065–1072

Kaplan D (2004) The Sage handbook of quantitative methodology for the social sciences. Sage

Yu S, Yu K, Tresp V, Kriegel H.-P (2006) "Collaborative ordinal regression," in *Proceedings of the 23rd international conference on Machine learning*. ACM, , pp. 1089–1096

Kim M (2014) Conditional ordinal random fields for structured ordinal-valued label prediction. Data min knowl discov 28(2):378–401

Kockelman KM, Kweon Y-J (2002) Driver injury severity: an application of ordered probit models. Accident Analysis & Prevention 34(3):313–321

Lanfranchi M, Giannetto C, Zirilli A, Alibrandi A (2014) Analysis of the demand of wine in sicily through ordinal logistic regression model. Calitatea 15(139):87

Lemmerich F, Atzmueller M, Puppe F (2016) Fast exhaustive subgroup discovery with numerical target concepts. Data Min Knowl Discov 30(3):711–762

Argyriou A, Evgeniou T, Pontil M (2008) Convex multi-task feature learning. Machine Learn 73(3):243–272

Liu J, Ji S, Ye J (2009) "Multi-task feature learning via efficient l 2, 1-norm minimization," in *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press, pp. 339–348

Gutiérrez PA, Perez-Ortiz M, Sanchez-Monedero J, Fernandez-Navarro F, Hervas-Martinez C (2016) Ordinal regression methods: survey and experimental study. IEEE Trans Knowl Data Eng 28(1):127–146

Witten IH, Frank E, Hall MA, Pal CJ (2016) Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, United States

Li L, Lin H.-T (2007) "Ordinal regression by extended binary classification," in *Advances in neural information processing systems*, pp. 865–872

Menon AK, Elkan C (2010) Predicting labels for dyadic data. Data Min Knowl Discov 21(2):327–343

Montañés E, Suárez-Vázquez A, Quevedo JR (2014) Ordinal classification/regression for analyzing the influence of superstars on spectators in cinema marketing. Expert Syst Appl 41(18):8101–8111

Mueller SG, Weiner MW, Thal LJ, Petersen RC, Jack C, Jagust W, Trojanowski JQ, Toga AW, Beckett L (2005) The alzheimer's disease neuroimaging initiative. Neuroimaging Clinics 15(4):869–877

Nesterov Y (2013) Introductory lectures on convex optimization: A basic course, vol 87. Springer Science & Business Media, Berlin

Ye F, Lord D (2014) Comparing three commonly used crash severity models on sample size requirements: multinomial logit, ordered probit and mixed logit models. Analyt methods accident res 1:72–85

Nesterov Y (2013) Introductory lectures on convex optimization: A basic course, vol 87. Springer Science & Business Media, Berlin

Park S-H, Fürnkranz J (2012) Efficient prediction algorithms for binary decomposition techniques. Data Min Knowl Discov 24(1):40–77

Zhou J, Chen J, Ye J (2011) "Clustered multi-task learning via alternating structure optimization," in *Advances in neural information processing systems*, pp. 702–710

Ruder S (2017) "An overview of multi-task learning in deep neural networks," *arXiv preprint* arXiv:1706.05098,

Duong L, Cohn T, Bird S, Cook P (2015) "Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, vol. 2, pp. 845–850

Yang Y, Hospedales T. M (2016) "Trace norm regularised deep multi-task learning," *arXiv preprint* arXiv:1606.04038,

Tran T, Phung D, Luo W, Venkatesh S (2015) Stabilized sparse ordinal regression for medical risk stratification. Knowl Info Syst 43(3):555–582

Lu Y, Kumar A, Zhai S, Cheng Y, Javidi T, Feris R (2016) "Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification," *arXiv preprint* arXiv:1611.05377,
Ando RK, Zhang T (2005) A framework for learning predictive structures from multiple tasks and unlabeled data. J Machine Learn Res 6:1817–1853
Beck A, Teboulle M (2009) A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J Imag sci 2(1):183–202
Williams R et al (2006) Generalized ordered logit/partial proportional odds models for ordinal dependent variables. Stata J 6(1):58
Witten IH, Frank E, Hall MA, Pal CJ (2016) Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, United States
Mueller SG, Weiner MW, Thal LJ, Petersen RC, Jack C, Jagust W, Trojanowski JQ, Toga AW, Beckett L (2005) The alzheimer's disease neuroimaging initiative. Neuroimaging Clinics 15(4):869–877
Yar Hamidi D, Wennberg K, Berglund H (2008) Creativity in entrepreneurship education. J small bus enterp dev 15(2):304–320
Ye F, Lord D (2014) Comparing three commonly used crash severity models on sample size requirements: multinomial logit, ordered probit and mixed logit models. Analyt methods accident res 1:72–85
Westbrook M. T, Viney L. L (1983) "Age and sex differences in patients' reactions to illness," *Journal of health and social behavior*, pp. 313–324,
Geifman N, Cohen R, Rubin E (2013) Redefining meaningful age groups in the context of disease. Age 35(6):2357–2366