ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240746

TabCGOK: Intra-Class Groups Retrieval and Inter-Class Ordinal Knowledge Augmented Network for Ordinal Tabular Data Prediction

Zhengdong Luo[a,b,c,*], **Abibulla Atawulla**[a,b,c], **Fengyi Yang**[a,b,c], **Yongqing Zhu**[d], **Yixiao Ren**[a,b,c], **Yunfei Han**[a,b,c] **and Xi Zhou**[a,b,c,**]

[a]Xinjiang Technical Institute of Physics and Chemistry, Chinese Academy of Sciences, Urumqi 830011, China.
[b]University of Chinese Academy of Sciences, Beijing 100049, China.
[c]Xinjiang Laboratory of Minority Speech and Language Information Processing, Urumqi 830011, China.
[d]Hithink RoyalFlush Information Network Co., Ltd. Hangzhou 310023, China.

Abstract. Ordinal tabular data, with advantages of structured knowledge representation in tabular data and the characteristic of inter-class ranks, has drawn increasing attention. However, existing retrieval-based tabular deep learning methods designed primarily for classical tabular data pay less attention to ordinal tabular data. Ordinal knowledge of ordinal tabular data provides a more explicit objective for tabular ordinal classification by considering both classification and regression properties. Furthermore, these approaches overlook the significance of intra-class group features which can balance the retrieved probability of various sample size groups and capture shared knowledge among multiple samples within same group. In this work, we propose the Intra-Class Groups Retrieval and Inter-Class Ordinal Knowledge Augmented Network (TabCGOK) model for ordinal tabular data prediction, equipped with Intra-Class Groups Retrieval (CG) module and Inter-Class Ordinal Knowledge Augmented (OK) module. The CG module provides intra-class group features candidate set for subsequent retrieval operation. It divides each class into several groups, then extracts the representation of each group as intra-class group features. And the intra-class group features candidate set consists of all intra-class group features from each class. The OK module is designed to capture inter-class ordinal knowledge. It estimates the ordinal distances by calculating inter-class feature distances, which could correspond to the inter-class non-isometric nature of ordinal knowledge, and then aggregates the previous ordinal distances to clarify the containment relationship of ordinal knowledge. OK module utilizes the attention mechanism for fusing the captured ordinal knowledge to retrieved intra-class group features. Finally, TabCGOK integrates fused intra-class group features with sample level features for ordinal tabular data prediction. Extensive experiments on several ordinal tabular datasets demonstrate the effectiveness of our method. The source code is available at https://github.com/luozhengdong/TabCGOK.

1 Introduction

Ordinal tabular data ubiquitous in daily life such as product quality evaluation and age prediction, whose labels represent ordinal ranks, is an important component of tabular data with samples (rows) sharing the same characteristics/attributes (columns) [36]. The classical tabular data can be categorized into two types: classification tabular data such as $Covertype$[1] dataset predicting forest types without ordinal relationship among labels, and regression tabular data such as $house\_16H$[2] dataset predicting house prices whose labels are continuous values. Ordinal tabular data, often treated as naive regression or classification data but actually falling in the intermediate between the two, is a type of tabular data.

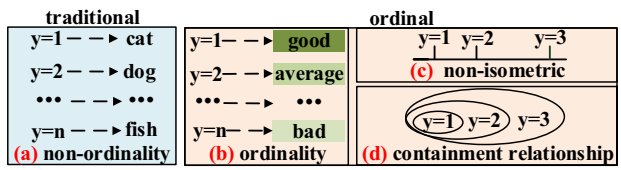

Figure 1: Properties of the inter-class ordinal knowledge. (a) Traditional classification data with disordered labels. (b) Ordinality of ordinal knowledge. (c) Ordinal knowledge distances are non-isometric. (d) Ordinal knowledge is in a certain containment relationship.

Compared to classical tabular data, ordinal tabular data offers additional inter-class ordinal knowledge [3] with three properties:

• **Ordinality**: ordinal knowledge is explicitly represented as ordinal scale labels. As show in Figure 1 (b), labels $y = 1, 2, ..., n$ denote $good$, $average$,..., and bad, which show different degrees of order. Whereas traditional classification labels (Figure 1 (a)) like "$fish$", "dog" and "cat" lack such correlation.

• **Non-isometric**: the distance between inter-class ordinal knowledge cannot be precisely quantified [35]. Popular example is customer satisfaction rating (an evaluation rating system from 1 to 5 levels), the distance between satisfaction level 2 and 1 is not equal to the distance between level 2 and 3, making it impossible to accurately quantify specific difference. As show in Figure 1 (c).

• **Containment relationship**: the containment relationship is reflected that higher level knowledge satisfies the condition of lower level knowledge. For example, if the label y denotes the level of knowledge, $y = 1$ for elementary school knowledge, $y = 2$ for junior high school knowledge, and $y = 3$ for senior high school

* Corresponding Author. Email: luozhengdong21@mails.ucas.edu.cn
** Corresponding Author. Email: zhouxi@ms.xjb.ac.cn

[1] https://openml.org/d/293
[2] https://openml.org/d/821

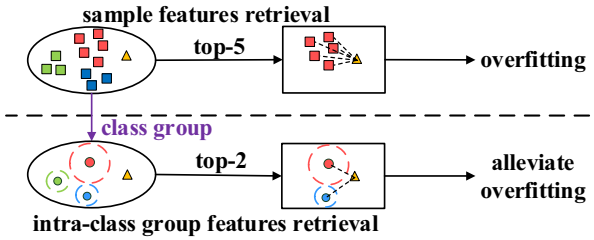

Figure 2: Intra-class group level retrieval vs. sample level retrieval. Samples from the same class can be divided into multiple groups. Red, blue, and green solid squares represent sample features. Red, blue and green solid circles represent group level features. Purple triangle represent query sample. Large solid circle indicates that the samples belong to the same category, and small dashed circles indicate that the samples belong to the same intra-class group. After similarity retrieval, top-k similar features are obtained.

knowledge. The containment relationship is shown by the second level ($y = 2$) meeting the requirements of the first level ($y = 1$), such as individuals who have mastered junior high school knowledge also have usually mastered elementary school knowledge. As shown in Figure 1 (d).

Despite many other fields of works [26, 34, 45, 46, 7, 9, 24] have demonstrated that ordinal knowledge can facilitate model prediction, few works specifically explore the use of ordinal knowledge in the tabular field. To this end, we should carefully design model to capture and utilize the inter-class ordinal knowledge for tabular data. Moreover, due to the distinct characteristics of tabular data compared to other data types like images and natural language, it's not feasible to directly apply ordinal prediction models from other fields.

After exploring the inter-class ordinal knowledge, our exploration extends to the intra-class knowledge context. We discover two noteworthy phenomena: (1) individuals with similar characteristics assign identical satisfaction score; (2) individuals within the same category, but possessing diverse characteristics, exhibit uniform satisfaction score in customer satisfaction rating data. These observations suggest the existence of multiple groups of diverse sample features within a class. These group features are broader in granularity compared to sample features, encapsulating the collective knowledge shared among all samples within the group. However, existing retrieval-based tabular deep learning works [19, 29, 42, 22, 37, 14, 13] only focus on sample level similar features. A potential issue is that the retrieved sample level similar features are influenced by the sample distribution of n group features. The larger sample size group have more samples, and the retrieved probability of this group samples is also higher. This may result in the small sample size group contributing less or ignoring the expression of class information, which leads the model to incorrectly consider the characteristics of larger sample size group to be a representation of the full class.

Actually other groups' features within same class are also important. However, fewer sample features result in less retrieved features, potentially causing bias and overfitting. For example, if the banking industry has more samples compared to other industries among customers with satisfaction score of 5, the model might retrieve mostly bankers samples and ignoring the diversity of other industries, leading the model to treat bankers' characteristics as those of 5-score individuals. As shown in Figure 2, samples of each class comprise n groups with different characteristics but sharing the same label. The uniform representation of an intra-class group can convey a collective partial meaning (shared knowledge) among samples of the group, and possessing discriminative properties to other groups. Additionally, each group is represented as a single feature, which gives every group the same retrieved probability and increases the diversity of retrieved similar features. These characteristics of intra-class groups can mitigate bias and overfitting of the model to samples of a particular group. Although intra-class group knowledge is so important, existing approaches without considering them lead to limitations. To improve the limitations, it is essential to explore shared knowledge of intra-class group level features.

Inadequate utilization of inter-class ordinal knowledge and neglect of shared knowledge of intra-class group level features are two issues that have not been addressed by existing tabular data prediction methodologies. To tackle these challenges, we propose Intra-Class Groups Retrieval and Inter-Class Ordinal Knowledge Augmented Network (TabCGOK) for Ordinal Tabular Data Prediction, a novel retrieval-based tabular deep learning method, which alleviates these limitations by retrieving similar intra-class group features and utilizing inter-class ordinal knowledge. TabCGOK is equipped with two main modules: the intra-class group level features retrieval augmented module (CG) and the inter-class ordinal knowledge augmented module (OK). Different from retrieving sample level similar features [14], the CG module aims to retrieve similar features from intra-class group level candidate set. The intra-class group level candidate set is composed of all group features, and the group feature is a shared knowledge representation of all sample features within the group. The supplementation of intra-class group level features can balance the retrieved probability of each type of group knowledge and increase the diversity of retrieved features, thus mitigating bias and overfitting. The OK module exploits the non-isometric nature and containment relationship inherent in ordinal knowledge. It generates contribution weights for the retrieved intra-class group features. Rather than directly employing labels [14, 46], the OK module utilizes inter-class feature distances to estimate ordinal distances, and accumulate previous ordinal distances to determine ordinal weights. Then, an attention-like operation fuses retrieved intra-class group features to the ordinal weights. Finally, TabCGOK integrates fused intra-class group features with sample level features for model prediction. Additionally, to confront the limitation of lacking a benchmark for ordinal tabular dataset, we carefully select several ordinal tabular datasets from tabular data benchmarks [12, 15, 17, 28].

The contributions of our work can be summarized as follows:

(1) We propose a novel retrieval-based mechanism deep learning model for tabular data, which considers both sample level and intra-class group level candidate sets, incorporating similar intra-class group knowledge to the similar samples representation, thereby improving model performance for tabular data prediction.

(2) We introduce ordinal knowledge of tabular data, realizing the non-isometric of ordinal knowledge by estimating inter-class ordinal distances and the containment relation of ordinal knowledge by designing the cumulative ordinal weights of previous ordinal distances. The introduction of ordinal knowledge enriches prediction methods and improves model performance for ordinal tabular data prediction.

(3) Our method TabCGOK achieves comparable state-of-the-art performance on seven real-world ordinal tabular data tasks. Compare to retrieval-based tabular deep learning methods, TabCGOK achieves optimal accuracy (ACC) and root mean square error (RMSE) performance on six of the seven datasets.

2 Related Work

2.1 Retrieval-based Tabular Deep Learning.

Recently, tabular deep learning has emerged as a rapidly growing research direction. DeepGBM [21] and NODE [30] employed de-

cision tree algorithms within deep neural networks to integrate advantages of neural networks and tree-based models. However, these methods have the disadvantage of high variance and increased computational complexity. To capture complex relationships and dependencies, Tabtransformer [18] and Tabnet [2] adapted attention or Transformer architectures for tabular data. To further improve performance and generalization, TabPFN [17], CT-BERT [44], and XTab [47] used pre-training and transfer learning to tabular domain.

Among various tabular deep learning methodologies, retrieval-based methods that implicitly or explicitly retrieve data points as the reference for prediction have shown significant gains. In implicit retrieval, SAINT [37] learnt the association of the target row with other rows in tabular. NPTs [25] captured inter-data point relationships through self-attention. While in explicit retrieval, the classical examples of retrieval-based tabular models are the "shallow" neighbor-based and kernel methods [29, 19]. There are also "deep" retrieval-based models applicable to tabular data prediction [14, 29, 32], of which the TabR [14] implemented its retrieval component with just one single-head attention-like module and achieved exciting performance on multiple tabular datasets. Compared to these approaches that only focus on sample level retrieval, we additionally consider intra-class group level retrieval.

2.2 Ordinal Classification (Ordinal Regression)

Ordinal classification (also called ordinal regression) is a machine learning branch, whose objective is to predict the ordinal label y given an input x, where $x \in X \subseteq \mathbb{R}$ and $y \in Y = \{C_1, C_2, \ldots, C_Q\}$. And $C_1 \prec C_2 \prec \cdots \prec C_Q$, where $\prec$ is an order relation. It is distinct from sorting and ranking task in the test phase, where the goal of ordinal classification is to obtain correct labels, rather than correct relative partial order of the patterns (sorting) or the total order of patterns that matches the order of train set (ranking) [16]. Ordinal classification is an intermediate problem between classification and regression [35]. In distinction to traditional classification, ordinal classification considers both the differences between classes and the order of classes. In distinction to metric regression, ordinal classification cannot quantify the distances between classes.

There are four main types of ordinal classification methods: (1) Naive methods [16, 39, 33, 1, 40], simplify ordinal classification to a traditional classification or regression problem. (2) Ordinal binary decomposition methods [3, 35, 41, 8, 23, 6], which decompose the ordinal target variable into several binary ones. (3) Threshold models [43, 11, 5], which map ordinal target variable to a one-dimensional space of continuous variables, determine the target by segmented space in which the predicted value is located. (4) Other approaches [26, 46, 38] explore utilizing ordinal label knowledge for feature extraction and fusion. Currently, ordinal classification is primarily investigated in the fields of Computer Vision [26, 34, 45, 46] and Natural Language Processing [7, 9, 24]. However, few works specifically focus on the problem of tabular ordinal prediction. To compensate for the scarcity of ordinal classification methods in tabular field, we design models that considers inter-class ordinal knowledge augmentation to achieve promising results.

3 Methods

Neglecting both the shared knowledge within intra-class groups and the inter-class ordinal knowledge poses a significant challenge to accurately predicting ordinal tabular data. To focus on these important information, we introduce Intra-Class Groups Retrieval and Inter-Class Ordinal Knowledge Augmented Network (TabCGOK). In TabCGOK, the CG module constructs candidate set of intra-class group level features and retrieves similar group features. Subsequently, the OK module estimates inter-class ordinal knowledge distances and converts them into inter-class ordinal weights. Finally, multiple retrieved similar group features and inter-class ordinal weights are fused to a single feature representation using an attention-like mechanism. This representation is then integrated with sample level feature representation for tabular prediction. In the inference stage, considering ordinal classification with regard to both classification and regression properties, the corresponding accuracy (ACC) and root mean square error (RMSE) are evaluated using cross entropy and mean squared error loss functions, respectively.

3.1 CG Module

As shown at the top of Figure 3, CG is a retrieval-augmented module that utilizes intra-class group level feature representations as the candidate set, unlike traditional retrieval-augmented module [19, 29, 42, 22, 37, 14, 13] which rely on sample level feature representations. Some datasets are preprocessed where the original samples (X) have been divided into numerical (X^{num}), binary (X^{bin}), and categorical (X^{cat}) subsamples, and to ensure alignment the columns of the group level candidate set and the columns of the original sample level candidate set, thus our framework begins with the preprocessed data. Since tabular data is inherently discrete, i.e., disrupting the positions of the columns does not affect the information content of the samples, we initially merge these subsample features ($X^{num}, X^{bin}, X^{cat}$) to reconstruct the original sample features X:

$$\begin{aligned} X_i &= concat(X_i^{\text{num}}, X_i^{\text{bin}}, X_i^{\text{cat}}) \\ X &= \{X_i\}, \quad i = 1, \ldots, c \end{aligned} \tag{1}$$

where c denotes the number of classes in dataset.

Then we utilize Grouping Algorithm (GA) (e.g. K-means) to divide samples X_i of each class into different groups G_{ij}:

$$\begin{aligned} G_{ij} &= GA(X_i), \quad X_i \in X \\ G_i &= \{G_{ij}\}, \quad j = 1, \ldots, \kappa \end{aligned} \tag{2}$$

where κ denotes the number of groups that the $i\text{-}th$ class is divided. For each group G_{ij}, which contains multiple sample instances, we construct a uniform representation R_{ij} by element-wise Mean-Pooling (MP). The group representations of each class R_i constitute the group level candidate set $G\_candidate$. The corresponding labels of these group representations are $Y\_candidate$:

$$\begin{aligned} &R_{ij} = MP(G_{ij}), \quad G_{ij} \in G_i \\ &R_i = \{R_{ij}\} \\ &G\_candidate = \{R_i\} \\ &Y\_candidate = \{Y : G\_candidate\} \\ &D\_candidate = \{G\_candidate, Y\_candidate\} \end{aligned} \tag{3}$$

where $D\_candidate$ is called the candidate set (support set).

Finally, standard retrieval operations are performed. x_query and $G\_candidate$ are encoded by linear encoder W_x and W_k, resulting in query feature f_k_q and group features candidate set $FG\_candidate$. Encoding functions for numerical, binary and categorical features are included in the encoder W_x. "Top-λ" similar group features $\widehat{F\_qg}$ and corresponding similar group labels $\widehat{Y\_qg}$

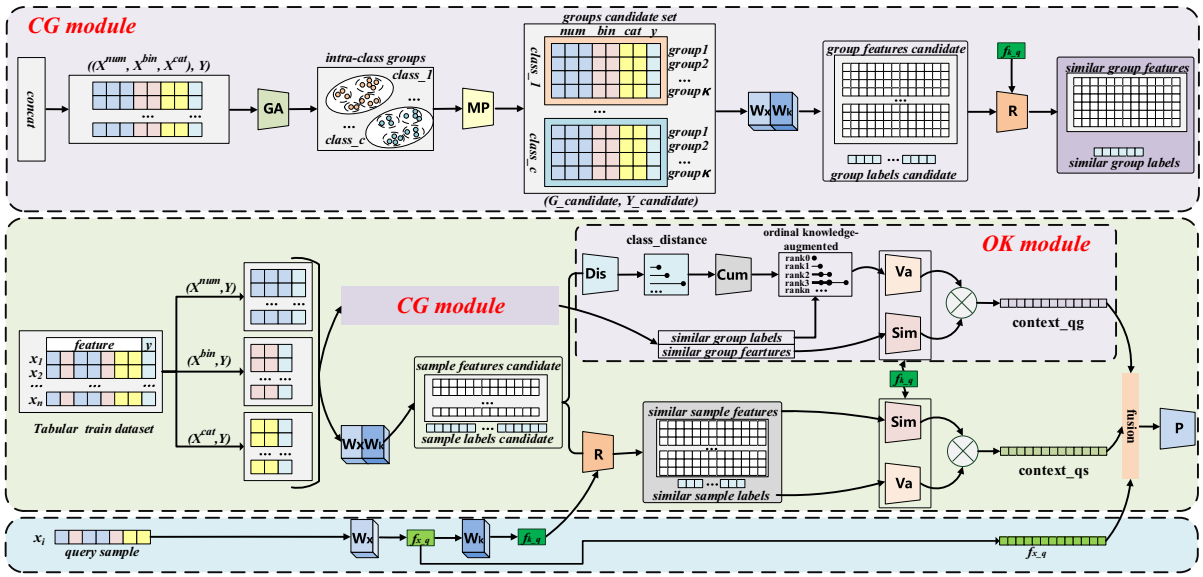

Figure 3: Framework of our approach. TabCGOK retrieves group-level similar features (CG) and fuses them with inter-class ordinal knowledge augmentation weights (OK) to obtain similar group-level contextual features, which are then fused with sample-level similar features and sample features to obtain the final feature representation. GA denotes group algorithm, MP denotes mean-pooling, W_x and W_k denote the encoder, R denotes the retriever, Dis denotes the distance algorithm, Cum denotes the cumulative algorithm, Va denotes the value algorithm, Sim denotes the similarity algorithm, P denotes the predictor, and x_i denotes query sample which is a validation or test sample instance.

are retrieved from $FG\_candidate$ by retriever R based on f_k_q :

$$
\begin{aligned}
&f_k\_q = W_k(W_x(x\_query)) \\
&FG\_candidate = W_K(W_x(G\_candidate)) \\
&\widehat{f_k\_qgu} = R(f_k\_q, FG\_candidate) \\
&\widehat{F\_qg} = \{\widehat{f_k\_qgu}\}, \quad u = 1, \ldots, \lambda \\
&\widehat{y\_qgu} = D\_candidate[\widehat{f_k\_qgu}] \\
&\widehat{Y\_qg} = \{\widehat{y\_qgu}\}
\end{aligned}
\tag{4}
$$

where λ denotes the number of retrieved group features similar to f_k_q. $\widehat{f_k\_qgu}$ denotes one of similar group features $\widehat{F\_qg}$. It is worth mentioning that numerical, binary, and categorical features in $G\_candidate$ can be easily identified using positional index. They share the encoder W_x and W_k, which include components for encoding multiple types of features.

In summary, to implement the retrieval of intra-class group level features within the CG module, we divide samples of each class into distinct groups, obtaining representation for each group. These group representations serve as candidate set at group level, from which similar group features are retrieved. Importantly, each group utilizes a single feature for representation, ensuring that each group feature has the same retrieved probability and increasing the diversity of retrieved features. This process aims to validate the existence of multiple distinct groups in each class, which could provide shared knowledge, and mitigate bias and overfitting.

3.2 OK Module

Another module of our innovation is an attention mechanism based on ordinal knowledge-augmented, namely OK. CG finally outputs the set $\widehat{F\_qg}$ of "$top-\lambda$" retrieved group features similar to a sample instance x_query and the corresponding labels $\widehat{Y\_qg}$. Inspired by the vanilla attention mechanism, we redesign the attention mechanism with ordinal knowledge for retrieved intra-class group features.

Similarity calculation. Specifically, the similarity score S_qg between each group feature of $\widehat{F\_qg}$ and the query sample feature f_k_q is first computed:

$$
S\_qg = softmax(-\|f_k\_q - \widehat{F\_qg}\|^2) \tag{5}
$$

The value of attention. Different from using labels directly [14, 46], we consider that label values in the dataset are only indicative and may not accurately reflect the true ordinal rank distances. In ordinal tabular data, labels represent an rank relationship, and we need to estimate the distances that are more relevant to the meaning of ordinal ranks. To achieve this, we calculate the mean feature f_{class} for each class:

$$
f_{class}\_i = MP(W_k(W_x(X_i))), \quad i = 1, \ldots, c \tag{6}
$$

Then we calculate the distance from each class to first class:

$$
\begin{aligned}
&cd\_i = f_{class}\_i - f_{class}\_1 \\
&cd = \{cd\_i\}, \quad i = 1, \ldots, c
\end{aligned}
\tag{7}
$$

As shown in Figure 1 (d), the containment relationship exists among ordinal knowledge, wherein higher-ranked classes contain the relevant knowledge of the lower-ranked ones. Accumulated inter-class distances are utilized to signify this relationship, where higher-ranked distances contain lower-ranked distances. We interpret the class (rank) distances as indicative of the amount of knowledge. Although the distance from each class to the first class inherently includes the knowledge of the lower-ranked classes, we accumulate these distances c_rank for knowledge augmentation to further emphasize the containment relationships. Additionally, we construct a dictionary Y_rank to illustrate the connections between labels and ordinal relationships:

$$
\begin{aligned}
&c\_rank_i = \sum_{h=0}^{i} cd\_h, \quad i = 1, \ldots, c \\
&Y\_rank = \{key = y_i : value = c\_rank_i\}
\end{aligned}
\tag{8}
$$

Then, based on the retrieved $\widehat{Y\_qg}$, we can calculate the value of attention:

$$\begin{aligned} Rank\_qg &= Y\_rank[\widehat{Y\_qg}] \\ V\_qg &= W_y(Rank\_qg) + T(f_k\_q - \widehat{F\_qg}) \end{aligned} \tag{9}$$

where W_y is an embedding table for classification task and a linear layer for regression task, which can be seen as the contribution of context object. $T(\bullet)$ is $LinearWithoutBias(Droupout(ReLU(Linear(\bullet))))$, which can be seen as the "correction" term [14].

Contextual feature. Finally, we can calculate the contextual feature $context\_qg$ for query x_query based on S_qg and V_qg by attention-like mechanism:

$$context\_qg = S\_qg \odot V\_qg \tag{10}$$

where $\odot$ denotes pointwise multiplication. Finally, OK module generates a group level contextual feature $context\_qg$ corresponding to the query sample instance x_query.

3.3 Sample Level Features Retrieval & Feature Fusion

Here, we will introduce the backbone model, a sample level retrieval-based tabular deep learning model. It generates a sample level contextual feature representation.

Sample Level Tabular Retrieval. For sample level retrieval, the formalization is as follows:

$$\begin{aligned} f_k\_q &= W_k(W_x(x\_query)) \\ FS\_candidate &= W_k(W_x(X_{train})) \\ \widehat{f_k\_qsn} &= R(f_k\_q, FS\_candidate) \\ \widehat{F\_qs} &= \{\widehat{f_k\_qsn}\}, \quad n = 1, \ldots, \beta \\ \widehat{Y\_qs} &= \{Y : \widehat{F\_qs}\} \\ S\_qs &= softmax(-\|f_k\_q - \widehat{F\_qs}\|^2) \\ V\_qs &= W_y(\widehat{Y\_qs}) + T(f_k\_q - \widehat{F\_qs}) \\ context\_qs &= S\_qs \odot V\_qs \end{aligned} \tag{11}$$

where f_k_q denotes the encoded form of query x_query. X_train denotes all sample instances in the train set. The sample features candidate set $FS\_candidate$ is obtained from the train set by encoder W_k and W_x. $\widehat{F\_qs}$ denotes the set of retrieved features related to f_k_q from the sample features candidate set $FS\_candidate$, which includes β similar sample features $\widehat{f\_qsn}$. $\widehat{Y\_qs}$ denotes the labels corresponding to the retrieval results $\widehat{F\_qs}$. S_qs and V_qs denotes the similarity scores and attention values. Finally, the sample level contextual feature $context\_qs$ related to query x_query is obtained. Additionally, W_x, W_k, R, W_y, $T(\bullet)$, $\odot$ remain the same structure as mentioned above.

Feature fusion. In this stage, x_query is solely encoded by W_x to obtain f_x_q. For fusion, f_x_q is initially added $(+)$ to the sample level contextual feature $context\_qs$. Subsequently, it is concatenated $(\oplus)$ with the intra-class group level contextual feature $context\_qg$ to ultimately derive feature f_sq, which is fed into the predictor.

$$\begin{aligned} f_x\_q &= W_x(x\_query) \\ f\_sq &= f_x\_q + context\_qs \oplus context\_qg \end{aligned} \tag{12}$$

In summary, our approach emphasizes the ordinal knowledge within ordinal tabular data. We estimate ordinal distances based on inter-class feature distances, while also considering the containment relationship among these ordinal classes. Subsequently, we calculate inter-class ordinal weights, which are then fused to the retrieved group features by an attention-like mechanism. The fused similar group feature, similar sample feature and query sample feature are integrated into a uniform representation for tabular prediction.

3.4 Objective Optimization

Ordinal classification is an intermediate task between classification and regression. Hence, drawing from evaluation metrics in the other field [26], we assess both accuracy (ACC) of classification and root mean square error (RMSE) of regression performance. Different loss functions are employed for each performance:

$$\begin{aligned} \mathcal{L}(y, \hat{y}) &= -\sum_{i=1}^{c} y_i \log(\hat{y}_i), \quad \text{for ACC} \\ \mathcal{L}(y, \hat{y}) &= \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \quad \text{for RMSE} \end{aligned} \tag{13}$$

where c denotes the number of classes, n denotes the number of samples, y denotes the true label, $\hat{y}$ denotes the predicted label.

4 EXPERIMENTS

4.1 Datasets

Numerous benchmark datasets exist for tabular data [12, 15, 17, 28], but there is a lack of carefully distinguished datasets specifically tailored for ordinal tabular data. Some previous works[3] have published ordinal tabular dataset. However, the definition of these ordinal tabular datasets often relies solely on labels and does not incorporate deeper contextual meanings. Upon careful consideration of the dataset's intrinsic meaning and the relationships among the labels, we may come to realize that they do not strictly conform to the characteristics of ordinal tabular dataset. For example, the *ailerons*[4] dataset is derived from the state of an airplane to predict the aircraft's action. Although these actions are represented by ordinal labels ($y = 1, 2, ..., 9$), the actions are independent of each other, lacking ordinal class rank and containment relationship. Therefore, we carefully chose several ordinal tabular datasets from the tabular dataset benchmarks, including $Wine\_Quality$[5] [14], $Abalone$[6], $Eucalyptus$[7], $Microsoft(MSLR-WEB10K)$[8] [14], $Yahoo$[9], Car[10] and Cmc[11]. These datasets are briefly described in Table 1.

4.2 Experimental Setup

The experimental environment is python=3.9 and pytorch=1.12. Experiments for large tabular datasets $Microsoft$ and $Yahoo$ are performed on one 32G NVIDIA V100 GPU, the remaining datasets experiments are performed on four 12G Tesla K80 GPUs, and tree-based baselines are run on the CPU. The random state of the inference phase is set to values between 0 and 15 to mitigate the differ-

[3] https://github.com/gagolews/teaching-data.
[4] https://github.com/gagolews/teaching-data.
[5] https://openml.org/d/287
[6] https://archive.ics.uci.edu/dataset/1/abalone
[7] https://www.openml.org/d/188
[8] https://www.microsoft.com/en-us/research/project/mslr/
[9] https://webscope.sandbox.yahoo.com/catalog.php?datatype=c
[10] https://www.openml.org/d/40975
[11] https://www.openml.org/d/23

Table 1: Summary statistics of ordinal tabular datasets.

datasets	total sample volume	classes	feature dimension	train size	validation size	test size	Discription
Wine_Quality	6,497	7	11	4,547	585	1,365	wine quality prediction
Abalone	4,177	8	7	2,923	835	419	abalone age prediction
Eucalyptus	736	5	19	515	147	74	utility level prediction
Microsoft	1,200,192	5	136	723,412	235,259	241,521	query relevance prediction
Yahoo	709,877	5	699	473,134	71,083	165,660	query relevance prediction
Car	1,728	4	6	1,209	345	174	acceptability prediction
Cmc	1,473	3	9	1,031	294	148	contraceptive method choice prediction

Table 2: Comparison of RMSE on ordinal tabular datasets. The bold entries indicate the best performance among retrieval-based tabular deep learning models, while underlined items indicate that our approach outperforms tree-based tabular models. The experimental results marked with $*$ are sourced from backbone, and the remaining ones are tested by ourselves. The RMSE lower is better.

	Datasets (Models)	Wine_Quality ↓mean±std	Abalone ↓mean±std	Eucalyptus ↓mean±std	Microsoft ↓mean±std	Yahoo ↓mean±std	Car ↓mean±std	Cmc ↓mean±std
Tree-based	XGBoost	0.602±0.014*	1.418±0.002	0.717±0.002	0.741±0.000*	0.735±0.000	0.117±0.001	0.706±0.001
	CatBoost	0.606±0.014*	1.424±0.004	0.684±0.005	0.741±0.000*	0.740±0.000	0.165±0.005	0.707±0.001
	LightGBM	0.612±0.014*	1.400±0.002	0.741±0.007	0.741±0.000*	0.727±0.000	0.165±0.001	0.701±0.000
Retrieval DL	KNN	0.720	1.495	0.921	0.764*	0.802	0.211	0.740
	DNNR	0.687	1.419	0.728	0.765*	-	0.171	0.740
	DKL	0.678±0.003	1.359±0.016	0.751±0.002	-*	-	0.120±0.011	0.722±0.006
	ANP	0.647±0.001	1.333±0.011	0.755±0.039	-*	0.754±0.000	0.052±0.019	0.731±0.030
	SAINT	0.676±0.004	1.359±0.006	0.731±0.025	0.763±0.007*	-	0.090±0.008	0.717±0.007
	MLP-PLR	0.634±0.018*	1.346±0.005	0.728±0.004	**0.744±0.000***	0.753±0.001	0.076±0.003	0.736±0.001
	TabR (backbone)	0.620±0.007*	1.335±0.004	0.722±0.014	0.748±0.000*	0.751±0.001	0.057±0.011	0.716±0.004
	TabCGOK (ours)	**0.611±0.003**	**1.321±0.003**	**0.703±0.019**	0.747±0.000	**0.748±0.001**	**0.041±0.006**	**0.715±0.002**

ences arising by random seed bias. The label encoding is set to "standard" for RMSE and "null" for ACC. X^{num} encoder policy is "quantile", X^{cat} encoder policy is "ordinal". The parameters of feature encoding dimension (d_main), retrieval dropout ($context\_dropout$), learning rate (lr), and weight decay ($weight\_decay$) are tuned by allowing our model to automatically learn through n_trials times on the train and validation set. The hyperparameter $\beta = 96$, and others are shown in Table 3. We use AdamW optimizer [27] and ReLU activation function. The predictor is simply a combination of LayerNorm, Linear, ReLU, and Dropout layers [14].

Table 3: The hyperparameters of TabCGOK. B denotes batch size.

Datasets	RMSE				ACC			
	κ	λ	B	n_trials	κ	λ	B	n_trials
Wine_Quality	4	17	256	100	3	10	256	100
Abalone	3	10	256	100	3	10	256	100
Eucalyptus	3	14	256	100	3	15	256	100
Microsoft	20	15	2048	20	5	10	4096	20
Yahoo	5	15	4096	50	5	10	2048	20
Car	6	11	256	100	3	10	96	100
Cmc	5	11	256	100	7	11	256	100

4.3 Evaluation Metrics

Ordinal classification is an intermediate task between classification and regression, with reference to the evaluation metrics of ordinal classification task in other fields [26, 24, 9], we adopt accuracy (ACC) and root mean square error (RMSE) as evaluation metrics.

4.4 Experimental Results

Our approach is compared with two types of methods: retrieval-based tabular deep learning models (KNN [19], DNNR [29], DKL [42], ANP [22], SAINT [37], MLP-PLR [13], TabR [14]) and tree-based tabular models (XGBoost [4], CatBoost [31], LightGBM [20]).

Despite tabular data classification and regression problems are still dominated by tree-based methods, varieties of deep learning-based methods have emerged aiming to narrow the gap between deep learning-based methods and tree-based methods [10], due to limitations of tree-based models [18] such as (1) tree-based models not allowing efficient end-to-end learning of sample encoders in presence of multi-modality along with tabular data; (2) state-of-the-art deep learning methods for processing missing and noisy data features not being applicable to tree-based models; (3) unsuitable for continual training from streaming data as compared to deep learning. Our method is compared to both retrieval-based tabular deep learning methods to demonstrate our strengths and tree-based models to show that we have further narrowed the gap between retrieval-based deep learning methods and tree-based models.

- For RMSE performance: our method surpasses the retrieval-based tabular deep learning baselines on six of seven ordinal tabular datasets, showing suboptimal performance on only one dataset, and outperforms the backbone (TabR) on all datasets. This further demonstrates the effectiveness of our approach, as shown in Table 2. When compared to the tree-based baselines, our method reaches the optimal on the $Abalone$ and Car datasets, and suboptimal on $Eucalyptus$. However, tree-based baselines are well-suited for CPU computation, which takes up more computational resources. Our experiments substantiate that we have successfully narrowed the gap between tabular deep learning methods and tree-based models. We remain committed to furthering this endeavor in the future.
- For ACC performance: as mentioned above, ordinal classification serves as an intermediate task, thus we also evaluate accuracy performance. Compared to retrieval-based tabular deep learning methods, our method achieves state-of-the-art (SOTA) performance for six of seven ordinal tabular datasets, and outperforms the backbone (TabR) on all datasets, as shown in Table 4. These ordinal tabular datasets are commonly treated as data for regression tasks [14, 12]. However, this perspective overlooks the impact of ordinal knowledge on model performance and achieves suboptimal ACC performance. Compared to tree-based tabular baselines, our method achieves optimal performance on five of the seven datasets. These results demonstrate the effectiveness of our method for tabular ordinal classification task, and highlight that our method further optimizes

Table 4: Comparison of ACC (%) on ordinal tabular datasets. The bold entries signify the best performance among retrieval-based tabular deep learning models, while underlined items indicate that our approach outperforms tree-based tabular models. The ACC higher is better.

	Datasets (Models)	Wine_Quality ↑mean±std	Abalone ↑mean±std	Eucalyptus ↑mean±std	Microsoft ↑mean±std	Yahoo ↑mean±std	Car ↑mean±std	Cmc ↑mean±std
Tree-based	XGBoost	65.52±0.6	32.94±0.5	66.22±2.3	57.13±0.0	55.48±0.0	98.85±0.0	56.76±0.7
	CatBoost	66.08±0.4	32.06±0.6	72.97±1.4	57.15±0.0	54.95±0.0	98.47±0.3	56.53±0.8
	LightGBM	64.74±0.3	33.33±0.4	68.47±0.8	57.10±0.0	55.65±0.0	99.43±0.0	56.08±1.8
Retrieval DL	KNN	57.36	30.79	63.51	55.85	41.15	93.67	50.67
	SAINT	63.44±0.1	34.13±0.9	72.07±2.1	56.95±0.1	-	99.04±0.3	56.31±1.4
	MLP-PLR	64.62±0.6	33.81±0.7	72.52±1.6	**57.20±0.0**	53.38±0.0	99.62±0.7	55.63±1.4
	TabR (backbone)	66.23±0.3	33.49±0.6	72.97±3.6	56.78±0.0	53.71±0.1	99.81±0.3	56.31±1.0
	TabCGOK (ours)	**66.91±0.7**	**34.37±0.6**	**73.42±0.8**	56.91±0.0	**53.85±0.0**	**99.99±0.01**	**57.21±1.0**

the performance gap with the tree-based tabular baselines. However, due to the unique characteristics of each dataset, optimal ACC is not achieved on $Microsoft$. This issue deserves further exploration in future work.

4.5 Ablation Study and Analysis

Table 5: Ablation experiments are conducted on $Wine\_Quality$ (WQ), $Eucalyptus$ (EU), $Abalone$ (AB) datasets for both CG and OK modules, with performance measured in terms of RMSE (where lower value is better). The backbone is TabR model. The bold entries represent the best performance.

Ablation module				Datasets		
exp.	backbone	CG	OK	WQ	EU	AB
(1)	✓			0.620	0.722	1.335
(2)	✓	✓		0.617	0.714	1.326
(3)	✓		✓	0.613	0.712	1.333
(4)	✓	✓	✓	**0.611**	**0.703**	**1.321**

CG and OK module ablation. We have developed intra-class groups retrieval (CG) and inter-class ordinal knowledge augmented (OK) components to enhance the model's ability for extracting crucial knowledge. To assess the effectiveness of these components, we conduct the ablation study on $Wine\_Quality$, $Abalone$ and $Eucalyptus$ datasets, as shown in Table 5. Our findings on these datasets reveal that : experiment (2) outperforms (1), suggesting that leveraging the shared knowledge from the CG module's intra-class groups improves prediction. Similarly, the performance of (3) surpasses (1), indicating that the ordinal knowledge from the OK module contributes to better prediction. The comparison between (4) and (2)(3) indicates that adding both CG module and OK module to backbone can achieve the maximum gain, meanwhile the comparison of (4) with (1) fully demonstrates the effectiveness of TabCGOK model. In summary, the performance gains vary under different experimental conditions affirming the effectiveness of two modules, and the best performance comes from the simultaneous equipping of the CG and OK modules in TabCGOK for ordinal tabular data prediction.

Analysis of confusion matrices. To verify the classification advantages of our method, we plot the confusion matrices (Figure 4) on test set based on the classification predicted labels y_pred and the true labels y_true, for both TabCGOK and backbone, respectively. Two main diagonals of the two subgraphs indicate the ratio of correct predictions. TabCGOK is worse than backbone only when $y\_true = y\_pred = 2$, but outperforms the backbone when considering the entire main diagonal. This suggests that our method is more effective. Next, we conduct classification error-free analysis focusing on the three categories of $y\_true = 2, 3, 4$, which have more samples. The total percentage of TabCGOK main diagonal and two neighboring lines are found to be 99%, 99.8%, and 98.7%, while 98.6%, 99.7%, and 95.1% for backbone. This suggests that TabCGOK closes the distance between the misclassified labels and the correct labels, demonstrating its feature representations are more accurate. In conclusion, our TabCGOK is superior than backbone because TabCGOK achieves higher prediction accuracy and narrows the distance between incorrectly predicted labels and true labels.

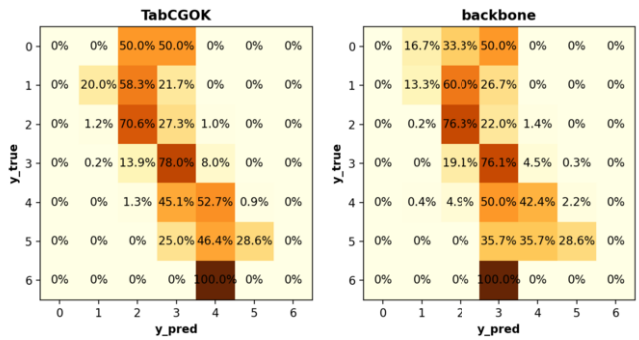

Figure 4: Confusion matrices of classification. Classification experiments of our methods TabCGOK (left) and backbone (right) on the $Wine\_Quality$ dataset. The value in the figure is ACC, where higher value is better.

5 Conclusion

In this work, to address the issues of existing tabular deep learning methods inadequate utilization of inter-class ordinal knowledge and neglect of shared knowledge of intra-class group level features, we propose TabCGOK model, a novel retrieval-based tabular deep learning framework for one type of tabular data, namely ordinal tabular data. Through the observed phenomenon of intra-class groups, we introduce the CG module of TabCGOK to capture the shared knowledge within each group and balance the retrieved probability of group level similar features, thus mitigating model bias and overfitting. Inspired by the properties of inter-class ordinal knowledge, we develop the OK module of TabCGOK to utilize the non-isometric nature and containment relationship of ordinal knowledge for obtaining ordinal weights. By employing an attention-like mechanism to fuse retrieved group level similar features to ordinal knowledge weights, we obtain a group level contextual feature representation. This representation is then integrated with a retrieved sample level contextual feature and query sample feature for ordinal tabular data prediction. The ACC and RMSE performance results, along with the ablation study and analysis, sufficiently demonstrate the effectiveness of our method. Our work provides a valuable exploration of ordinal tabular data prediction and lays foundational groundwork for future research on tabular ordinal classification (ordinal regression). Future work will focus on exploring automatic intra-class grouping strategies, and improving computational efficiency for large ordinal tabular datasets.

Acknowledgements

This research was supported partly by the Tianshan Talent Training Program (No. 2022TSYCLJ0035), partly by the Xinjiang Key Research and Development Task (No. 2023B01028), partly by the Xinjiang Tianchi Talents Program (Fengyi Yang), and partly by Xinjiang Major Scientific and Technological Project (No. 2023A01006).

References

[1] A. Agresti. *Analysis of ordinal categorical data*, volume 656. John Wiley & Sons, 2010.

[2] S. Ö. Arik and T. Pfister. Tabnet: Attentive interpretable tabular learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 6679–6687, 2021.

[3] W. Cao, V. Mirjalili, and S. Raschka. Rank consistent ordinal regression for neural networks with application to age estimation. *Pattern Recognition Letters*, 140:325–331, 2020.

[4] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[5] W. Chu and S. S. Keerthi. Support vector ordinal regression. *Neural computation*, 19(3):792–815, 2007.

[6] J. F. P. da Costa, H. Alonso, and J. S. Cardoso. The unimodal model for the classification of ordinal data. *Neural Networks*, 21(1):78–91, 2008.

[7] C. Fanconi, M. van Buchem, and T. Hernandez-Boussard. Natural language processing methods to identify oncology patients at high risk for acute care with clinical notes. *AMIA Summits on Translational Science Proceedings*, 2023:138, 2023.

[8] E. Frank and M. Hall. A simple approach to ordinal classification. In *Machine Learning: ECML 2001: 12th European Conference on Machine Learning Freiburg, Germany, September 5–7, 2001 Proceedings 12*, pages 145–156. Springer, 2001.

[9] B. J. Franks, B. Dinkelmann, S. Fellenz, and M. Kloft. Ordinal regression for difficulty estimation of stepmania levels. *arXiv preprint arXiv:2301.09485*, 2023.

[10] J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[11] T. S. Fuchs and J. Keshet. Thor: threshold-based ranking loss for ordinal regression. *arXiv preprint arXiv:2205.04864*, 2022.

[12] Y. Gorishniy, I. Rubachev, V. Khrulkov, and A. Babenko. Revisiting deep learning models for tabular data. *Advances in Neural Information Processing Systems*, 34:18932–18943, 2021.

[13] Y. Gorishniy, I. Rubachev, and A. Babenko. On embeddings for numerical features in tabular deep learning. *Advances in Neural Information Processing Systems*, 35:24991–25004, 2022.

[14] Y. Gorishniy, I. Rubachev, N. Kartashev, D. Shlenskii, A. Kotelnikov, and A. Babenko. Tabr: Unlocking the power of retrieval-augmented tabular deep learning. *arXiv preprint arXiv:2307.14338*, 2023.

[15] L. Grinsztajn, E. Oyallon, and G. Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? *Advances in neural information processing systems*, 35:507–520, 2022.

[16] P. A. Gutiérrez, M. Perez-Ortiz, J. Sanchez-Monedero, F. Fernandez-Navarro, and C. Hervas-Martinez. Ordinal regression methods: survey and experimental study. *IEEE Transactions on Knowledge and Data Engineering*, 28(1):127–146, 2015.

[17] N. Hollmann, S. Müller, K. Eggensperger, and F. Hutter. Tabpfn: A transformer that solves small tabular classification problems in a second. *arXiv preprint arXiv:2207.01848*, 2022.

[18] X. Huang, A. Khetan, M. Cvitkovic, and Z. Karnin. Tabtransformer: Tabular data modeling using contextual embeddings. *arXiv preprint arXiv:2012.06678*, 2020.

[19] G. James, D. Witten, T. Hastie, R. Tibshirani, et al. *An introduction to statistical learning*, volume 112. Springer, 2013.

[20] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

[21] G. Ke, Z. Xu, J. Zhang, J. Bian, and T.-Y. Liu. Deepgbm: A deep learning framework distilled by gbdt for online prediction tasks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 384–394, 2019.

[22] H. Kim, A. Mnih, J. Schwarz, M. Garnelo, A. Eslami, D. Rosenbaum, O. Vinyals, and Y. W. Teh. Attentive neural processes. *arXiv preprint arXiv:1901.05761*, 2019.

[23] K.-j. Kim and H. Ahn. A corporate credit rating model using multi-class support vector machines with an ordinal pairwise partitioning approach. *Computers & Operations Research*, 39(8):1800–1811, 2012.

[24] S. Kolanu, S. Kommabathula, P. Peka, S. P. V. Gadiraju, and V. Pathibandla. Twitter sentiment analysis based on ordinal regression. In *AIP Conference Proceedings*, volume 2492. AIP Publishing, 2023.

[25] J. Kossen, N. Band, C. Lyle, A. N. Gomez, T. Rainforth, and Y. Gal. Self-attention between datapoints: Going beyond individual input-output pairs in deep learning. *Advances in Neural Information Processing Systems*, 34:28742–28756, 2021.

[26] W. Li, X. Huang, Z. Zhu, Y. Tang, X. Li, J. Zhou, and J. Lu. Ordinalclip: Learning rank prompts for language-guided ordinal regression. *NIPS*, 35:35313–35325, 2022.

[27] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[28] D. McElfresh, S. Khandagale, J. Valverde, V. Prasad C, G. Ramakrishnan, M. Goldblum, and C. White. When do neural nets outperform boosted trees on tabular data? *Advances in Neural Information Processing Systems*, 36, 2024.

[29] Y. Nader, L. Sixt, and T. Landgraf. Dnnr: Differential nearest neighbors regression. In *International Conference on Machine Learning*, pages 16296–16317. PMLR, 2022.

[30] S. Popov, S. Morozov, and A. Babenko. Neural oblivious decision ensembles for deep learning on tabular data. *arXiv preprint arXiv:1909.06312*, 2019.

[31] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31, 2018.

[32] H. Ramsauer, B. Schäfl, J. Lehner, P. Seidl, M. Widrich, T. Adler, L. Gruber, M. Holzleitner, M. Pavlović, G. K. Sandve, et al. Hopfield networks is all you need. *arXiv preprint arXiv:2008.02217*, 2020.

[33] J. Sánchez-Monedero, P. A. Gutiérrez, P. Tiňo, and C. Hervás-Martínez. Exploitation of pairwise class distances for ordinal classification. *Neural computation*, 25(9):2450–2485, 2013.

[34] J. Shah, M. M. R. Siddiquee, Y. Su, T. Wu, and B. Li. Ordinal classification with distance regularization for robust brain age prediction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 7882–7891, 2024.

[35] X. Shi, W. Cao, and S. Raschka. Deep neural networks for rank-consistent ordinal regression based on conditional probabilities. *Pattern Analysis and Applications*, 26(3):941–955, 2023.

[36] R. Shwartz-Ziv and A. Armon. Tabular data: Deep learning is not all you need. *Information Fusion*, 81:84–90, 2022.

[37] G. Somepalli, M. Goldblum, A. Schwarzschild, C. B. Bruss, and T. Goldstein. Saint: Improved neural networks for tabular data via row attention and contrastive pre-training. *arXiv preprint arXiv:2106.01342*, 2021.

[38] B.-Y. Sun, J. Li, D. D. Wu, X.-M. Zhang, and W.-B. Li. Kernel discriminant learning for ordinal regression. *IEEE Transactions on Knowledge and Data Engineering*, 22(6):906–910, 2009.

[39] V. Torra, J. Domingo-Ferrer, J. M. Mateo-Sanz, and M. Ng. Regression for ordinal variables without underlying continuous variables. *Information sciences*, 176(4):465–474, 2006.

[40] H.-H. Tu and H.-T. Lin. One-sided support vector regression for multiclass cost-sensitive classification. In *ICML*, volume 2, page 5, 2010.

[41] N. Twomey, R. Poyiadzi, C. Mann, and R. Santos-Rodríguez. Ordinal regression as structured classification. *arXiv preprint arXiv:1905.13658*, 2019.

[42] A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing. Deep kernel learning. In *Artificial intelligence and statistics*, pages 370–378. PMLR, 2016.

[43] H. Wu, H. Lu, and S. Ma. A practical svm-based algorithm for ordinal regression in image retrieval. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 612–621, 2003.

[44] C. Ye, G. Lu, H. Wang, L. Li, S. Wu, G. Chen, and J. Zhao. Ct-bert: learning better tabular representations through cross-table pre-training. *arXiv preprint arXiv:2307.04308*, 2023.

[45] K. Zha, P. Cao, J. Son, Y. Yang, and D. Katabi. Rank-n-contrast: Learning continuous representations for regression. *Advances in Neural Information Processing Systems*, 36, 2024.

[46] S. Zhang, L. Yang, M. B. Mi, X. Zheng, and A. Yao. Improving deep regression with ordinal entropy. *arXiv preprint arXiv:2301.08915*, 2023.

[47] B. Zhu, X. Shi, N. Erickson, M. Li, G. Karypis, and M. Shoaran. Xtab: Cross-table pretraining for tabular transformers. *arXiv preprint arXiv:2305.06090*, 2023.