Pairwise Multi-Layer Nets for Learning Distributed Representation of Multi-field Categorical Data

Ying Wen
University College London
London, UK
ying.wen@cs.ucl.ac.uk

Tianyao Chen
Shanghai Jiao Tong University
Shanghai, China
tianyao@try-skycn.net

Jun Wang
University College London
London, UK
jun.wang@cs.ucl.ac.uk

Weinan Zhang
Shanghai Jiao Tong University
Shanghai, China
wnzhang@sjtu.edu.cn

ABSTRACT

This paper presents a method of pairwise multi-layer networks for multi-field categorical data, which widely exists with various applications such as web search, recommender systems, social link prediction, and computational advertising. The success of non-linear models, e.g., factorization machines, boosted trees, has proved the potential of exploring the interactions among inter-field discrete categories. Inspired by Word2Vec, the distributed representation for natural language, we propose a PMLN (*Pairwise Multi-Layer Nets*) model to learn the distributed representation for multi-field categorical data. In PMLN, a low-dimensional continuous vector is automatically learned for each category in each field. The interactions among inter-field categories are explored by different neural gates and the most informative ones are selected by pooling layers. Such combined categories can be further explored by performing more gate interactions with another category and then selected by additional pooling operations. In our experiments, with the exploration of the interactions between pairwise categories over layers, the model outperforms state-of-the-art models in a supervised learning task, i.e., ad click prediction, while capturing the most significant interactions from the data in an unsupervised fashion.

CCS CONCEPTS

• **Information systems** → **Computational advertising**; *Information retrieval*; • **Computing methodologies** → *Knowledge representation and reasoning*.

ACM Reference Format:
Ying Wen, Tianyao Chen, Jun Wang, and Weinan Zhang. 2019. Pairwise Multi-Layer Nets for Learning Distributed Representation of Multi-field Categorical Data. In *1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data (DLP'19), August 4, 2019, Anchorage, AK, USA.* ACM, New York, NY, USA, 8 pages.

DLP-KDD'19, August 5, 2019, Anchorage, AK, USA

1 INTRODUCTION

There are different abstraction levels within data. For the low-abstraction continuous sensory data (e.g., images, videos, and audio) directly acquired from the physical world, the strong correlations (local patterns) are, quite often, known a priori within the data. As such, one can directly embed the prior knowledge into a learning model such as neural networks to automatically distill such patterns and consequently perform predictions [7, 13]. However, on the other hand, for high-abstraction data from our social and business activities, e.g., natural language and transactional log data, the data is commonly discrete and contains atomic symbols, whose meaning and correlation are unknown a priori. A typical solution is to employ embedding techniques [2, 17] to map the discrete tokens into a (low-dimensional) continuous space and further build neural networks to learn the latent patterns.

Multi-field categorical data is a type of high-abstraction data where the categories in each field are heterogeneous with those in other fields. Such a type of data is very widely used in data mining tasks based on transaction logs from many social or commercial applications, such as recommender systems [19], social link prediction [4], and computational advertising [32]. Table 1 gives an example of multi-field categorical data in user behavior targeting where we observe user browsing patterns; given those multi-field categorical features, a common task is to predict their actions such as clicks and conversions [14, 31, 33].

As there is no explicit dependency among these inter-field categories, two solutions are mainly used for building machine learning models that extract the local patterns of the data and make good predictions. The first solution is to create combining features across fields, such as City:Shanghai&Weekday:Friday [3]. Such feature engineering is expensive on human efforts and feature/parameter space. The second solution is to build functions [20] or employ neural networks over the embedded features [32]. These solutions are of low efficiency because of the brute-force feature engineering or aimless embedding interactions.

In this paper, we propose an unsupervised PMLN (Pairwise Multi-Layer Nets) model to learn the distributed representation of multi-field categorical data. The interactions among inter-field categories are explored by different neural gates and the informative ones are selected by K-max pooling layers. Note that the K-max pooling process acts like the classic Apriori algorithm in frequent item set

Table 1: A simple example of multi-field categorical data from iPinYou dataset .

TARGET	GENDER	WEEKDAY	CITY	BROWSER
1	MALE	TUESDAY	BEIJING	CHROME
1	FEMALE	TUESDAY	HONG KONG	IE
0	MALE	TUESDAY	BEIJING	CHROME
NUMBER	2	7	351	6

mining and association rule learning [1]. Repeating this pairwise interaction with K-max pooling, our PMLN model automatically extracts salient feature interactions and further explores higher-order interactions. Based on effective data representation by repeated interaction and pooling layers, fully connected layers are built to further learn discriminative patterns to make good predictions.

To train the pairwise interaction PMLN model effectively, we present a discriminant training method to estimating the category vectors. Furthermore, with the exploration of the pairwise and high-order category interactions, our PMLN model attains great performance improvement over state-of-the-art models in supervised learning tasks, such as user response rate prediction, while successfully captures the most significant interactions in unsupervised learning tasks.

2 RELATED WORK AND PRELIMINARIES

In this section, we outline the major data representation methods that are used for representing the *discrete categorical data*. These methods serve as the preliminaries of our PMLN model.

2.1 One-Hot Representation

It is common to use one-hot representation for discrete data in natural language processing or computational advertising tasks. For the first data sample as an example, the data is vectorized by one-hot encoding as

$$\underbrace{[0,1]}_{\text{GENDER:MALE}}, \underbrace{[0,1,\ldots,0,0]}_{\text{WEEKDAY:TUESDAY}}, \underbrace{[0,\ldots,1,\ldots,0]_{351}}_{\text{CITY:BEIJING}}, \underbrace{[1,\ldots,0]}_{\text{BROWSER:IE}}.$$

With each category as a dimension, one-hot representation preserves full information of the original data. Two main problems of one-hot representation are that (i) it may suffer from the curse of dimensionality, especially in deep learning-related applications; (ii) it cannot capture the similarity of each word/category pair, and we cannot even find any relationships among the synonyms or categories in the same field.

2.2 Distributed Representation

Distributed representation is first proposed by Hinton [11]. The basic idea of distributed representation is training the model to map each word into a d-dimensional vector (generally, d is the hyper-parameter of the model, and d is far smaller than whole vocabulary size N of words/categories), and the semantic similarity between the words/categories can be measured through the distance (such as cosine similarity, Euclidean distance) of their corresponding low dimension vectors. Word2Vec [17] is one of the most widely used methods to train the distributed word vector representation. Compared with text, with the local patterns among the neighbor words, multi-field categorical data has no explicit order relationships among inter-field categories. Also, the text vocabulary size (10^5) is often much smaller than the category size ($10^6 \sim 10^8$), making our problem more difficult. Another difference between our category and word is that category does not take the order into account or use any sliding window for context; in other words, we take all categories in the same training sample as the neighbor of a category.

Besides, typical deep network models learn the distributed representation of discrete categorical data implicitly, because the discrete data usually need an embedding layer to be feed into the deep networks. But, the common deep network models (e.g, restricted Boltzmann machines [22], multi-layer perceptrons [6], convolutional neural networks [15] , recurrent neural networks [34]) do not take the interactions among the categories in the different fields into account. Therefore, to explore the interaction between fields, product neural network (PNN) [18] uses deep neural network model with a product layer based on the embedded feature vectors to model the inter-field feature interactions. A weighted pairwise interaction model [5] is proposed for entity embedding of heterogeneous categorical events, and Factorization Machine (FM) [19] also uses pairwise interaction to learn implicit distributed representation. Meanwhile, both of them can only explore the interactions between pairwise categories in 2-order[1], the further higher order interactions still cannot be explored in a better way.

2.3 User Response Prediction

Learning and predicting user response is critical for personalizing tasks, including web search and online advertising, and the multi-field categorical data is widely available in the user response prediction task. In this task, most widely used linear model including logistic regression [21], nonlinear models including factorization machine [19], field-aware factorization machine [12], gradient tree models [10]. However, these models cannot explore the high order feature interactions or adaptively learn effective representations of the feature.

In recent years, deep neural networks have demonstrated their superior performance on this task. A factorization machine initialized feedforward neural network (FNN) is proposed by [32], which efficiently reduces the dimension from sparse features to dense continuous features. Convolutional neural networks are introduced for user click prediction (CCPM) in [15], but it only performs the convolution on the neighbor fields, which cannot explore the full interactions among the all fields. By adding a product layer between embedding layer and fully-connected layer, Product-based Neural Network (PNN) [18] achieves a better performance on CTR task, but this only explore low-order feature interactions. To take both advantages of linear model ('wide') and deep neural networks ('deep'), Cheng et al. [6] proposed a *Wide & Deep* model, which combines the wide net with deep net to form the final output. Based on *Wide & Deep* model, DeepFM [8] and DCN [28] enhence the wide part by cross network and factorization machine correspondingly; and in the meantime, self-attention models [23, 29, 30] and

[1] Although FM can be possibly extended to high orders, there is little literature trying such a setting.

NFM [9] focus on optimize the deep part by introducing the attention mechanism and bi-interaction layers via product operations over embeddings. More recently, DIN/DIEN [35, 36] also applied the attention mechanism, unlike AFM, they added the attention layer between the embedding layer and MLP, which models users' interests for different items. Compared with above deep net architectures, our PMLN contains all their advantages and is largely different as (i) the pairwise interaction layer can be implemented with various gates (interactive operations) to exploration different data interactions, and (ii) repetitive interaction and pooling operations distill non-trivial effective data pattern in an fully automatic manner, which is like a neural generalization of the Apriori data mining algorithm [1].

3 PAIRWISE MULTI-LAYER NETS

In this section, we introduce a PMLN (Pairwise Multi-Layer Network) model and its training method in detail. We design neural gates in the model to capture the interactions between each pair of categories, followed by the K-max pooling layers to select the most important interactions. We then repeat this processes to explore higher level interactions. Figure 1 illustrates the overview of the proposed architecture.

3.1 Interaction and Pooling Layers

Interaction Layer. To evaluate the interaction between each pair of categories, we use a gate to obtain the interaction result. Mathematically, a gate is a function $f : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ that takes any pair of category vectors $\boldsymbol{c}_i$ and $\boldsymbol{c}_j$ in the same sample $\boldsymbol{c}$ as input, and outputs interaction result vector $\boldsymbol{c}'_{i,j} = f(\boldsymbol{c}_i, \boldsymbol{c}_j)$. The interaction output vector $\boldsymbol{c}'_{i,j}$ acts as a certain combining feature of $\boldsymbol{c}_i$ and $\boldsymbol{c}_j$.

Note that $\boldsymbol{c}'_{i,j}$ keeps the same dimension as the category embedding vectors like $\boldsymbol{c}_i$ and $\boldsymbol{c}_j$ so that it can be further used to interact with other categories.

We provide several options of gate f as:

$$f^{\text{sum}}(\boldsymbol{c}_i, \boldsymbol{c}_j) = \boldsymbol{c}_i + \boldsymbol{c}_j, \tag{1}$$

$$f^{\text{mul}}(\boldsymbol{c}_i, \boldsymbol{c}_j) = \boldsymbol{c}_i \odot \boldsymbol{c}_j, \tag{2}$$

where $\odot$ is the element-wise multiplication operator. We can also can employ more complex gates, such as the highway gate [24], which is formulated as

$$\begin{aligned} f^{\text{highway}}(\boldsymbol{c}_i, \boldsymbol{c}_j) =& \boldsymbol{\tau} \odot g(\mathbf{W}_H(\boldsymbol{c}_i + \boldsymbol{c}_j) + \boldsymbol{b}_H) \\ &+ (1 - \boldsymbol{\tau}) \odot (\boldsymbol{c}_i + \boldsymbol{c}_j), \end{aligned} \tag{3}$$

where g is a nonlinear function and $\boldsymbol{\tau} = \sigma(\mathbf{W}_\tau(\boldsymbol{c}_i + \boldsymbol{c}_j) + \boldsymbol{b}_\tau)$ represents a "transform gate".

Then we apply the gate f on each pair of category vectors $\boldsymbol{c}_i$,$\boldsymbol{c}_j$, where n is the number of field:

$$\boldsymbol{c}' = [\boldsymbol{c}'_{1,2}, \boldsymbol{c}'_{1,3}, \cdots, \boldsymbol{c}'_{1,n}, \cdots, \boldsymbol{c}'_{n-2,n-1}, \boldsymbol{c}'_{n-1,n}]. \tag{4}$$

After the interaction, an activation function will be applied to implement the non-liner transformation.

K-Max Pooling Layer. We next describe a pooling operation that is a generalization of the max pooling based on the norm length of interaction outputs of each pair of category vectors. We keep the K maximum interaction output vectors $\boldsymbol{c}'_{i,j}$ according to their norm length, where K is the number of the original categories of the training sample. It would keep the max-pooling result $\boldsymbol{c}'_{\text{kmax}} = [\boldsymbol{c}'_1, \boldsymbol{c}'_2, \cdots, \boldsymbol{c}'_K]$ having the same size with the original embedding matrix $\boldsymbol{c}$ and $\boldsymbol{c}'_K$ is the embedding vector in $\boldsymbol{c}'$ in Eq. (4) that has top-K normal length.

Before producing an output for the interaction results, the interaction and K-max pooling operations will be repeated for several times in order to capture high-level interactions among the different field category vectors. After that, we output a prediction from the final interaction vector representation by a fully connected layer. Note that the above network structure can be used to build an auto-encoder to conduct unsupervised learning [26] since the intermediate layer can be regarded as an information-held low-dimensional representation of the input data. We leave this for future work, while staying with the label output network for both supervised (containing both negative and positive examples) and unsupervised (only containing positive examples where negative examples are generated randomly) learning tasks.

An interesting discussion is to compare our PMLN model with association rules mining, which aims to identify the most frequently appeared joint category instances (items), with or without a condition. *Apriori* [1] is a popular algorithm for association rules mining by exploiting dependencies between candidate frequent item sets of length K and frequent item sets of length $K - 1$. In our PMLN model, with neural networks, we provide an alternative way of generating such high-order interactions (i.e. item sets) among category instances. Via the pooling operation, our model can also find the most frequent category set automatically, which will be demonstrated and tested from our experiments in the following Sections 4 and 5. To our knowledge, this is the first work to compare Apriori data mining algorithm with the neural network pooling mechanism.

3.2 Discriminant PMLN for Training

To train the Pairwise Multi-Layer Network model, we design a training scheme called discriminant PMLN, which would train the model in a supervised way for unsupervised learning of the data.

In the discriminant PMLN, we feed the Sample Encoding Module showed in Figure 1 with a true or fake sample, the encoded sample vector will be followed by an MLP to predict the probability p of a true sample. As such, the generation of a fake sample would influence the learned category vector. In this paper, we generate a fake sample following this way: first, randomly choose a sample from the training set; second, randomly choose several categories in this sample and replace them with randomly chosen categories that belong to the same field. For example, we get a user behavior instance x = [Weekday:Wednesday, City:Beijing], and we randomly choose the category City:Beijing and replace it with City:Shanghai, then we build a fake sample x' = [Weekday:Wednesday, City:Shanghai]. The discriminant network is then trained to predict whether the new sample should be a true sample. The loss function of discriminant network is average cross entropy, which would maximize the likelihood of correct prediction:

$$L = \frac{1}{M} \sum_{i=1}^{M} -y_i \log(p_i) - (1 - y_i) \log(1 - p_i), \tag{5}$$

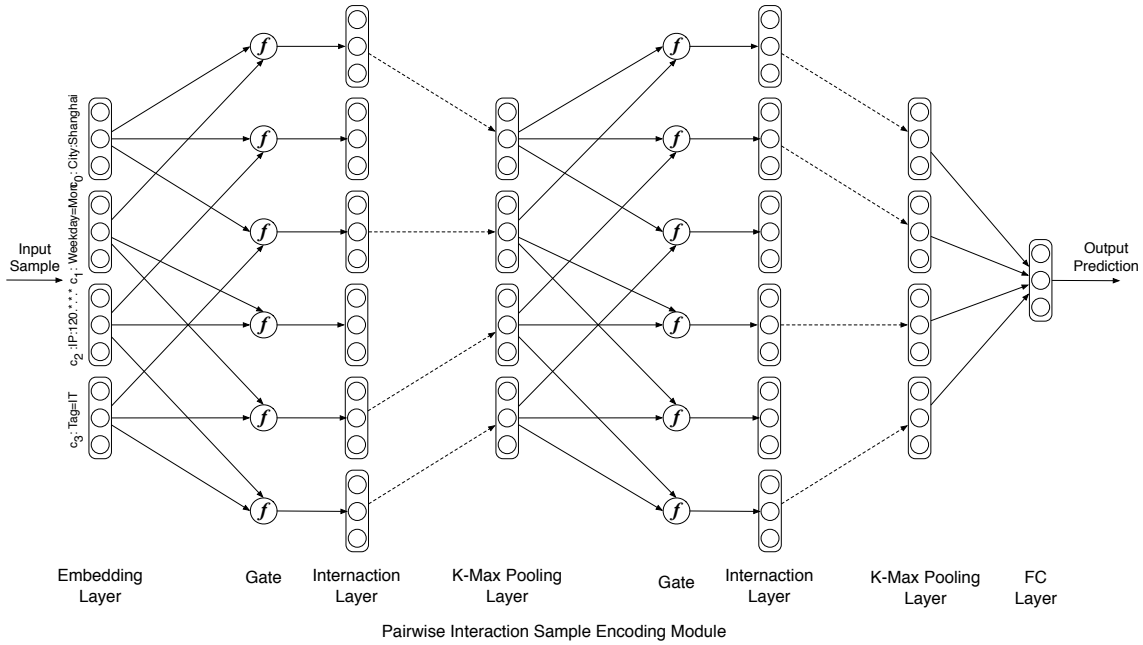

Figure 1: Sample encoding module. Each category pair will be fed into a gate to get the interaction between two categories. Next, using K-max pooling to capture important interactions. Repeat above two steps, which could capture higher level category interactions. Finally, we use a full connection layer to transform final interaction vectors into the prediction.

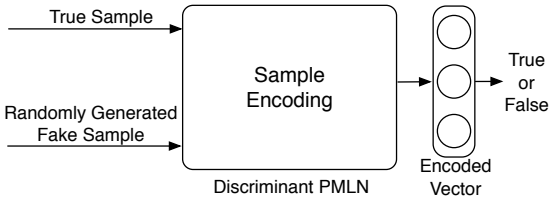

Figure 2: The discriminant PMLN model which learns the category embedding by training a discriminator to distinguish the true samples from the fake ones.

where M is the number of training samples. The i-th sample is labelled with $y_i \in \{1, 0\}$, which means true or fake sample, and p_i is the predicted probability that the given training sample is true.

4 SYNTHETIC DATA EXPERIMENTS

To explore and add our understanding of the PMLN model, we conduct a simulation test with synthetic data. In particular, we are interested in understanding how the learned vectors would be able to capture and leverage the most significant patterns embedded in the data.

4.1 Synthetic Dataset and Evaluation Metrics

To simulate the real-world multi-field categorical data, we use multivariate normal sampling to generate the true data distribution for the following experiments. Suppose the data has 4 fields $\{A, B, C, D\}$, each field contains 10 categories, and a sample can be represented as $\boldsymbol{x} = (a_i, b_i, c_i, d_i)$. We then randomly generate the means and covariance matrix for 4-dimensional truncated multivariate normal sampling with two-sided truncation. This sampling method can generate 4 float numbers between 0 and 10. We can convert the float numbers to integer which can represent the categories in 4 fields. In such a way, we can generate the data with specific joint distribution, which means certain categorical pair or 3-tuple like $p(a_4, b_4)$ or $p(a_3, c_5, d_6)$ may have a higher joint distribution probability. Recall that in our PMLN model, we have a K-max pooling layer, which will *select* the most popular category pairs in the dataset. Repeating the pairwise interaction layers and K-max pooling layers, we can also explore a high order categorical 3-tuple or 4-tuple etc. Therefore, our task here is to evaluate if our model would be able to capture these frequently occurred patterns from a given dataset; in other words, to test if our model would be able to keep the category pairs with the highest joint distribution probabilities in the K-max pooling results. This processes is in line with association rule mining [1], exploring the frequent categorical n-tuple from frequent categorical $(n-1)$-tuple.

We generate the positive data according to the above truncated multivariate normal sampling and choose uniform sampling to generate the fake (negative) data. We then apply discriminant PMLN to train the model. Because we know the true distribution of the generated real data, the most frequent category pairs/triples are known. We use precision and Spearman's rank correlation coefficient to evaluate the results of 1st/2nd K-max pooling layer (category pairs/triples pooling results), to see if the model can learn the true joint distribution in the real data. The details of the evaluation metrics are described in the following section.

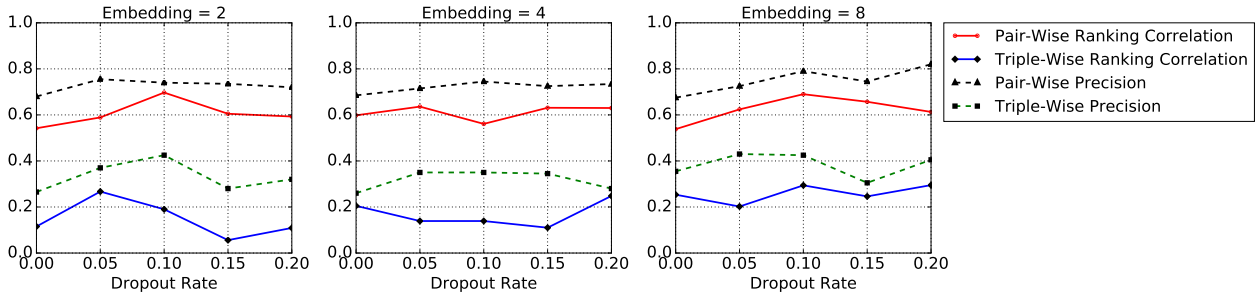

Figure 3: Precision and rank correlation on synthetic data, bigger embedding size and appropriate dropout rate leads to achieve better performance.

To evaluate how our network structure and K-max pooling help identify the significant n-tuples, we feed 1000 samples to the trained model and record the 1st and 2nd K-max pooling layers' results. Then we count the frequency of the category pairs/3-tuples in the real samples, and select top 20 ranked category pairs/3-tuples as target. Then we count the frequency of max-pooled category pairs/triples in the results and compare the top 20 frequent category pairs/3-tuples in the results to calculate precision and Spearman's rank correlation coefficient. Precision measures the fraction of category pairs/triples in the results that are also in the target. The Spearman's rank correlation coefficient measures the correlation between two ranked lists.

4.2 Result and Discussion

Figure 3 summarizes the results of the precision and the rank correlation on synthetic data. We can see that our model can easily find over 80% of the category pairs with high joint distribution probabilities under the best parameter settings. From the rank correlation, our model can make the ranking correlation over 0.6 of category pairs which means the category pairs with higher joint distribution probability would be more possible to appear in the K-max pooling result. As for the category triples case, the precision and rank correlation become lower than the category pairs', because finding 3-order combination is harder and relies on the accuracy from the 2-order. We also vary the dropout rate against those measures. It shows that dropout tends to help improving the accuracy of captured patterns. This can be explained by considering the fact that dropout brings randomness into the selection and allows exploration. But the best dropout rate seems rather arbitrary and highly dependent on the other parameter settings. [2]

5 REAL-WORLD DATA EXPERIMENTS

In this section, we continue our experiment using a real-world advertising dataset for click-through rate estimation. The iPinYou dataset [14] is a public real-world display ad dataset with each ad display information and corresponding user click feedback [33]. This dataset contains around 19.5M ad display instances with 14.8k positive user feedback (click). Each instance has 23 fields, and we choose 18 fields of them which have categories with occurrence larger than 10.[3]

5.1 Unsupervised Learning Experiment

We have tried different parameter settings and the performance is measured by the accuracy of our model to predict real samples. We also calculate the rank correlation coefficient and the precision to evaluate our model the same as we described in Section 4.1.

We continue our study on the model's ability of capturing the most significant patterns as we described in Section 3.2. Because the iPinYou dataset contains the unencrypted fields and categories, e.g. city, region and tag, so we choose the iPinYou dataset which has been introduced above as real (positive) data. As for the fake (negative) data, we randomly choose a sample in the iPinYou dataset and randomly replace some categories with other categories in the same field to generate the fake data, similar to what we have introduced in Section 3.2. We also set up two baseline models to compare the model accuracy performance: (i) DNN Concat model, which concatenates category embedding vectors to make prediction, and (ii) DNN Sum model, which sums up the category embedding vectors to make the prediction.

5.1.1 Result and Discussion. From Figure 5, we see that on the iPinYou dataset, our pairwise interaction models can achieve the accuracy of 85% which is about 1.7% improvement comparing with the simple DNN models' best case. Even the worst case in our model is better than the DNN models' best case. It means our model can find the extra information during the interactions and the K-max pooling processes. In addition, the model with interaction times as 3 usually yields better performance than that with interaction times as 2, which may be due to the fact that the more interaction times capture higher-order interactions and help make more accurate predictions. Besides, during the interactions, the pairs like (female, fashion), (male, IT) which would leads high Click-through Rate(CTR) are extracted which would be great helps to predict the CTR.

We next use the same evaluation metrics that described in Section 4.1 to test the ability of capturing data patterns. We find that in the real-world dataset, our model is still able to keep high precision and rank correlation and can achieve even better performance.

[2] iPinYou Dataset Link: https://contest.ipinyou.com/, and the code is available at GitHub Repo: https://github.com/ying-wen/pmln.

[3] The selected fields are WEEKDAY, HOUR, USER AGENT, IP, REGION, CITY, AD EXCHANGE, DOMAIN, URL, AD SLOT ID, AD SLOT WIDTH, AD SLOT HEIGHT, AD SLOT VISIBILITY, AD SLOT FORMAT, AD SLOT FLOOR PRICE, CREATIVE ID, KEY PAGE URL, AND USER TAGS.

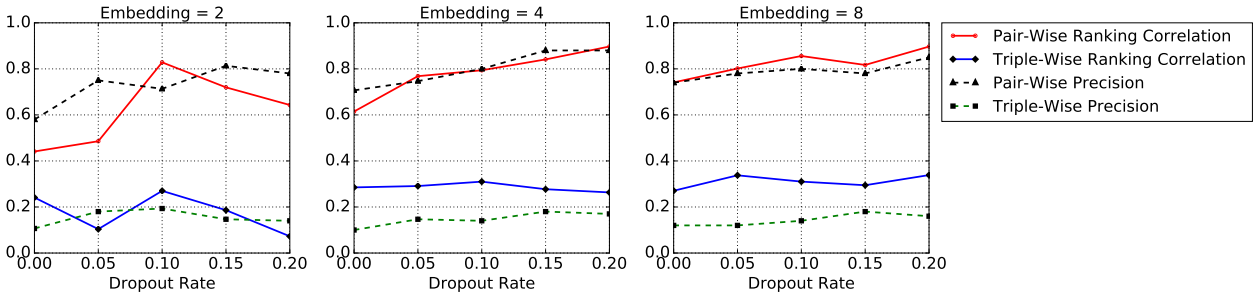

Figure 4: Precision and Rank Correlation on iPinYou Data; bigger embedding size and appropriate dropout rate leads to achieve better performance.

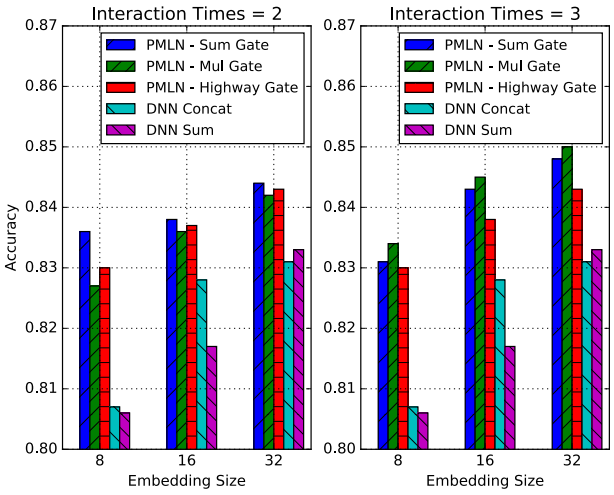

Figure 5: Accuracy of distinguishing true impression from fake impression on iPinYou dataset.

The precision and rank correlation on category pairs are over 0.8 which is a 30% improvement comparing to the performance on synthetic dataset. For the category triples case, we also have similar performance compared with the synthetic dataset.

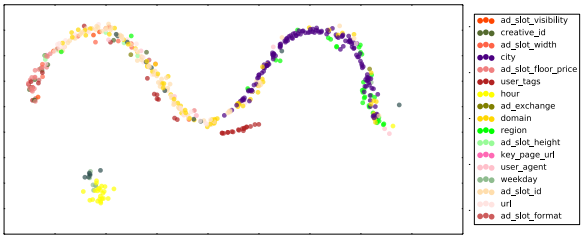

Figure 6: Visualization of learned PMLN embeddings in 2D via t-SNE which shows clustering by fields.

5.1.2 Field-Wise Clustering Property. We use the t-SNE [16] to visualize the learned category embeddings, which is shown in Figure 6. Every categories belong to same field will have same color in the figures. From the Figure 6, we can find that the category embeddings have clustering property which means categories belong to same field or similar field are intend to form a clustering. We also can find this pattern in Table 2, given a province category 'Henan', the nearest categories given by learned embeddings are three cities 'Hebi', 'Shangqiu' and 'Xuchang', which are belong to 'Henan'. Besides, the category embeddings learned by PMLN model is non-linear distribution cased by the sigmoid discriminator.

Table 2: Examples of the nearest categories of given categories

Category	region: Henan	user tags: Long-term/health
Top 3 Similar Categories	city: Hebi	user tags: Long-term/motherhood
	city: Shangqiu	tags: Long-term/outdoors
	city: Xuchang	user tags: Long-term/food

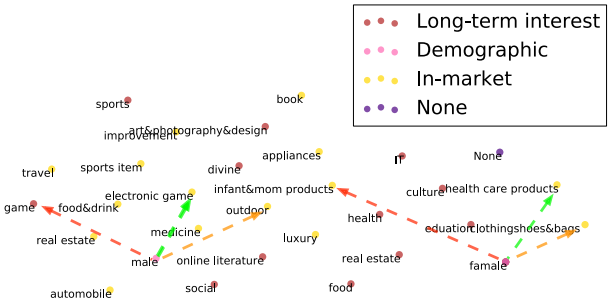

Figure 7: Tags embeddings visualization in 2D via t-SNE which shows analogy property.

5.1.3 Analogy Property. Similar to Word2Vec [17], we train high dimensional category vectors on a large amount of data. As Figure 7, we used t-SNE to visualize learned category embeddings of tags. The resulting embeddings show interesting semantic relationships between categories, such as a female/male user and their interests, e.g., the electronic game is to a male user as beauty is to a female user. Category embeddings with such analogy relationships could be used to improve many existing tasks, such as Click-Through Rate Prediction and Query Intent Prediction, and many other applications that have not yet be invented.

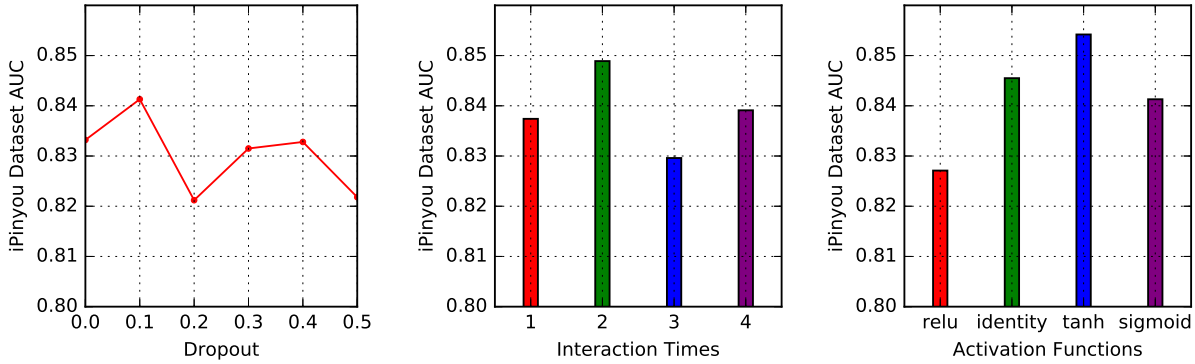

Figure 8: Performance Comparison over different Parameter Settings for PMLN models.

Table 3: AUC of CTR prediction on iPinYou dataset.

Model	LR	FM	FFM	CCPM	FNN	PNN	PMLN-FNN-1	PMLN-FNN-2
AUC	0.8323	0.8349	0.8449	0.8364	0.8453	0.8565	0.8599	**0.8640**

5.2 Click-through Rate Prediction Experiment

We now move to the evaluation on a supervised learning task. We consider click-through rate (CTR) prediction, which is important for many personalized Web services such as E-commerce, social recommendation and computational advertising [27, 31], and even 1% improvement in this task will bring great revenue. The most widely used CTR estimation model is the logistic regression based on one-hot data representation. Many deep learning models have been further investigated in CTR prediction. [32] proposed Factorization-Machine Supported Neural Networks (FNN) models for user response prediction. Convolutional Click Prediction Model (CCPM) [15] and Product Neural Network (PNN) [18] have been used in CTR prediction and gain some improvement on this task. To our knowledge, all of above previous work focuses on directly improving the prediction performance in supervised learning tasks and none of them investigates the learned representation of multi-field categorical data or how to learn the better representation.

In order to investigate our pairwise interaction model on the CTR task, we use the pairwise interaction sample encoding module to encode a training sample concatenated with the embedding vectors, which is followed by an MLP (multi-layer perceptron) to predict click-through probability. We choose following models as strong baselines:

- **Logistic Regression (LR)**: Logistic regression is the most widely used linear model [21].
- **Factorization Machine (FM)**: Simply apply the factorization machine on one-hot encoded sparse features of the training sample [19].
- **Field-aware Factorization Machine (FFM)**: A variant of Factorization Machine with pairwise interaction tensor factorization [12].
- **CCPM**: A deep neural network model by introducing convolutional layer in the model to make click prediction [15].
- **FNN**: A deep neural network model based on concatenated category vectors following with MLPs, being able to capture high-order latent patterns of multi-field categorical data [32].
- **PNN**: A deep neural network model with product layer, which is a concatenation of inner product and outer product [18].
- **PMLN-FNN-1**: This is our proposed architecture that only concatenates pairwise interaction output vectors among K-max pooling results to form the final vector representation and make prediction.
- **PMLN-FNN-2**: This is our proposed architecture that explore category vectors pairwise interaction result between K-max pooling results and category embeddings to form the final vector representation and make prediction.

5.2.1 Result and Discussion. We use Area Under ROC Curve (AUC) as the evaluation metrics to measure the performance of a prediction. We conduct the grid search for each model to make sure that each model has achieved its best performance, and because the iPinyou data is really sensitive to the downsampling rate, which would brings 10% improvement than without downsampling case, therefore, we choose the optimal downsampling rate to 0.1 to make sure all the models are compared in an equal setting. Specifically, empirically optimal hyper-parameters for the model are set as: the category embedding size is 16, the SGD batch size is 64, the Nadam [25] is set as SGD optimizer with default settings, the gate type is "Mul" and the norm type for K-max pooling is L2 norm. Then the model followed by three fully connected layer with width $[128, 32, 1]$. Besides, we evaluate the performance over different dropout rates, and find that set dropout rate as 0.1 would be the best. We also try to compare three different activation functions (sigmoid, tanh, relu) and set identity mapping as the baseline, the result shows that "tanh" yields the best performance, which has the advantages of non-linear transformation between $(-1, 1)$, and it may help gain more benefits on multi-field categorical data. Finally, we compare different interaction times and set it as two (3-tuple), suggesting that a high order of interactions helps improve the performance, but more than two would overfit the data and thus managed the performance.

Table 3 gives the results of our CTR experiment, compared with various baselines. We see that there is around 3% improvement over LR in terms of AUC. Our PMLN models also outperform the FM/FFM/CCPM/FNN/PNN model, with more than 2.2% over the FNN, and achieve the state of the art performance on CTR task. It

can be explained by their ability of taking higher order information into consideration, which helps make better decision.

In our pairwise interaction model, we also test different hyper-parameters and settings, and the result is given in Figure 8. First, we evaluate the performance over different dropout rates, and find that setting dropout as 0.1 would be the best, as shown in Figure 8. We also explore the impact of interaction. From the result, the model with 2 interaction times would have better generalization on the test set. Finally, we compare three different activation functions (sigmoid, tanh, relu) and set identity mapping as the baseline. The result shows that "tanh" yields the best performance, which has the advantages of non-linear transformation between $(-1, 1)$, and it may help gain more benefits on multi-field categorical data.

6 CONCLUSION

In this paper we have proposed a novel Pairwise Multi-Layer Network (PMLN) model working on the multi-field categorical data. Different from the other models, PMLN repetitively computes and selects inter-field category pairwise interactions to automatically explore high-level interactions, which is analogous to the Apriori algorithm in association rule mining. Moreover, we present an efficient discriminant training method to estimate the category vectors. We also apply our pairwise interaction model on CTR prediction, of which we have observed a significant performance gain over several strong baselines. For future work, we plan to design more sophisticated gates to explore different interaction patterns among inter-field categories; also leveraging PMLN in various data mining problems is of great interest to us.

ACKNOWLEDGMENTS

This research was partially funded by MediaGamma Ltd. during Ying Wen's internship.

REFERENCES

[1] Rakesh Agrawal, Ramakrishnan Srikant, et al. 1994. Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, Vol. 1215. 487–499.

[2] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *journal of machine learning research* 3, Feb (2003), 1137–1155.

[3] Olivier Chapelle, Eren Manavoglu, and Romer Rosales. 2015. Simple and scalable response prediction for display advertising. *ACM Transactions on Intelligent Systems and Technology (TIST)* 5, 4 (2015), 61.

[4] Kailong Chen, Tianqi Chen, Guoqing Zheng, Ou Jin, Enpeng Yao, and Yong Yu. 2012. Collaborative personalized tweet recommendation. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 661–670.

[5] Ting Chen, Lu-An Tang, Yizhou Sun, Zhengzhang Chen, and Kai Zhang. 2016. Entity embedding-based anomaly detection for heterogeneous categorical events. *IJCAI* (2016).

[6] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*. ACM, 7–10.

[7] Alan Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *ICASSP*. IEEE, 6645–6649.

[8] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017).

[9] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 355–364.

[10] Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, Stuart Bowers, et al. 2014. Practical lessons from predicting clicks on ads at facebook. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*. ACM, 1–9.

[11] Geoffrey E Hinton. 1986. Learning distributed representations of concepts. In *Proceedings of the eighth annual conference of the cognitive science society*, Vol. 1. Amherst, MA, 12.

[12] Yuchin Juan, Yong Zhuang, Wei-Sheng Chin, and Chih-Jen Lin. 2016. Field-aware factorization machines for CTR prediction. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 43–50.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*. 1097–1105.

[14] Hairen Liao, Lingxiao Peng, Zhenchuan Liu, and Xuehua Shen. 2014. iPinYou global rtb bidding algorithm competition dataset. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*. ACM, 1–6.

[15] Qiang Liu, Feng Yu, Shu Wu, and Liang Wang. 2015. A Convolutional Click Prediction Model. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 1743–1746.

[16] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9, Nov (2008), 2579–2605.

[17] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[18] Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang. 2016. Product-based neural networks for user response prediction. In *2016 IEEE International Conference on Data Mining*. IEEE, 995–1000.

[19] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International Conference on Data Mining*. IEEE, 995–1000.

[20] Steffen Rendle. 2012. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)* 3, 3 (2012), 57.

[21] Matthew Richardson, Ewa Dominowska, and Robert Ragno. 2007. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th international conference on World Wide Web*. ACM, 521–530.

[22] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. 2007. Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*. ACM, 791–798.

[23] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. 2018. AutoInt: Automatic Feature Interaction Learning via Self-Attentive Neural Networks. *arXiv preprint arXiv:1810.11921* (2018).

[24] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway networks. *arXiv preprint arXiv:1505.00387* (2015).

[25] Ilya Sutskever, James Martens, George E Dahl, and Geoffrey E Hinton. 2013. On the importance of initialization and momentum in deep learning. *ICML (3)* 28 (2013), 1139–1147.

[26] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*. ACM, 1096–1103.

[27] Jun Wang, Weinan Zhang, and Shuai Yuan. 2016. Display Advertising with Real-Time Bidding (RTB) and Behavioural Targeting. *arXiv preprint arXiv:1610.03013* (2016).

[28] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*. ACM, 12.

[29] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. 2017. Attentional factorization machines: Learning the weight of feature interactions via attention networks. *arXiv preprint arXiv:1708.04617* (2017).

[30] Chen Xu, Chengzhen Fu, Peng Jiang, and Wenwu Ou. 2018. Learning Representations of Categorical Feature Combinations via Self-Attention. (2018).

[31] Shuai Yuan, Jun Wang, and Xiaoxue Zhao. 2013. Real-time bidding for online advertising: measurement and analysis. In *Proceedings of the Seventh International Workshop on Data Mining for Online Advertising*. ACM, 3.

[32] Weinan Zhang, Tianming Du, and Jun Wang. 2016. Deep Learning over Multi-field Categorical Data. In *European Conference on Information Retrieval*. Springer, 45–57.

[33] Weinan Zhang, Shuai Yuan, Jun Wang, and Xuehua Shen. 2014. Real-time bidding benchmarking with iPinYou dataset. *arXiv preprint arXiv:1407.7073* (2014).

[34] Yuyu Zhang, Hanjun Dai, Chang Xu, Jun Feng, Taifeng Wang, Jiang Bian, Bin Wang, and Tie-Yan Liu. 2014. Sequential Click Prediction for Sponsored Search with Recurrent Neural Networks. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*.

[35] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2018. Deep Interest Evolution Network for Click-Through Rate Prediction. *arXiv preprint arXiv:1809.03672* (2018).

[36] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 1059–1068.