Deep Embedding Logistic Regression

Zhicheng Cui
Department of Computer Science and Engineering
Washington University in St. Louis
St Louis, USA
z.cui@wustl.edu

Muhan Zhang
Department of Computer Science and Engineering
Washington University in St. Louis
St Louis, USA
muhan@wustl.edu

Yixin Chen
Department of Computer Science and Engineering
Washington University in St. Louis
St Louis, USA
chen@cse.wustl.edu

***Abstract*—Logistic regression (LR) is used in many areas due to its simplicity and interpretability. While at the same time, those two properties limit its classification accuracy. Deep neural networks (DNNs), instead, achieve state-of-the-art performance in many domains. However, the nonlinearity and complexity of DNNs make it less interpretable. To balance interpretability and classification performance, we propose a novel nonlinear model, Deep Embedding Logistic Regression (DELR), which augments LR with a nonlinear dimension-wise feature embedding. In DELR, each feature embedding is learned through a deep and narrow neural network and LR is attached to decide feature importance. A compact and yet powerful model, DELR offers great interpretability: it can tell the importance of each input feature, yield meaningful embedding of categorical features, and extract actionable changes, making it attractive for tasks such as market analysis and clinical prediction.**

***Index Terms*—interpretability, accountability, actionability, classification**

I. Introduction

Classification has been studied for decades, during which numerous algorithms including logistic regression (LR), k-nearest neighbors (KNN), support vector machine (SVM), decision trees (DT), random forest (RF) and deep neural networks (DNNs) have been proposed. Many of them can achieve high accuracies. For example, DNNs can outperform humans in image recognition tasks [1]. However, classification remains a challenging problem as prediction accuracy is not the only concern in many scenarios.

In particular, many tasks calls for *interpretability* of the model, which entails: 1) accountability (revealing the significance of each feature), 2) actionability (identifying changes to input features that can turn the model output to a desired label), and 3) *quantification* of categorical features. In areas such as market analysis and clinical prediction, these three properties are critically needed in addition to high accuracy. For example, in clinical prediction of ICU transfers, in addition to accurate prediction, doctors need to know the contributing factors that triggered the alert (accountability), which factors can be quickly altered to prevent the ICU transfer (actionability), and how categorical features such as disease type and medication records affect the prediction (quantification). Unfortunately, most existing algorithms cannot balance these competing needs well.

LR is widely used, especially on high-dimensional cases, due to its simplicity and efficiency. Moreover, LR offers accountability and actionability. Weights in LR can measure feature importance and tell how we can alter certain features with a minimum cost to achieve a desired output. However, being a linear classifier, LR has limited separation ability and generally low accuracy. In addition, categorical features are not naturally supported by LR.

Kernel Logistic Regression (KLR) has nonlinear separation ability with the use of kernel functions, which implicitly maps features into a high-dimensional space [2]. Without explicit mapping, interpreting the model output is nearly impossible. This is also the case with most kernel methods such as kernel-based SVM. Density-based Logistic Regression (DLR) avoids this limitation through dimensional-wise transformation by applying kernel estimators [3]. However, DLR cannot handle large-scale dataset. For a dataset with N instances and D dimensions, the training time complicity for DLR is between $O(DN^2)$ and $O(DN^3)$, which is expensive for datasets with a large number of instances. Also, DLR requires $O(DN)$ time to evaluate an instance, which is again expensive for real-time testing.

DNNs keep breaking records in applications such as image classification [4], speech recognition [5] and objection detection [6]. However, the prediction results of DNNs are known to be very hard to interpret due to the extreme nonlinearity and complexity of the model [7]. The lack of interpretability greatly restricts their prevalence in fields such as clinical predictions [8] and business intelligence [9] where explanation, insights, and actionability are much needed.

In this paper, we proposed an end-to-end logistic regression model, deep embedding logistic regression (DELR), which incorporates LR with deep learning based feature embedding. By taking the advantage of DNNs' superior expressing power, each feature is first transformed into nonlinear representation before being fed into a LR layer. DELR has high efficiency by leveraging GPU computing. Nonlinear feature transformation equips DELR with nonlinear separation ability. Using deep embedding, we can also naturally handle and quantify categorical features, which is not supported by LR. Last but not least, accountability and actionability are offered by the LR layer.

In summary, our contributions are as follows.

1) We propose DELR, a classifier that offers scalability, nonlinearity, support for mixed data types and excellent inter-

pretability.

2) We analyze and demonstrate the model accountability and actionability of DELR through case studies.

3) We empirically validate the accuracy performance of DELR as compared with existing interpretable methods including LR, DT, DLR and gradient boosting decision stumps (GBDS). Commonly used non-interpretable models such as SVM-rbf, random forest (RF) are also tested for reference.

4) We visualize the quantification of categorical feature embedding and further verify the interpretability of DELR.

5) We apply DELR on a real-world clinical dataset and show how interpretability can help doctors make decisions.

II. Preliminaries

In this section, we introduce the notations and briefly review the limitations of LR and its extensions.

Suppose we are given a dataset $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^N$ with N instances, where $\mathbf{x}_i$ and y_i are D dimensional feature vector and label of instance i, respectively. Each feature vector is the concatenation of two types of feature vectors, numerical feature vector $\mathbf{x}_i^R \in \mathcal{R}^{D_1}$ and categorical feature vector $\mathbf{x}_i^{\mathcal{C}} = ([x_i^C]_1, [x_i^C]_2, ..., [x_i^C]_{D_2})$. Each element in $\mathbf{x}_i^R$ is a real number and each element in $\mathbf{x}_i^{\mathcal{C}}$ is an ordinal number. DELR can handle both binary and multi-class classification. For ease of presentation, we consider binary classification where $y_i \in \mathcal{C} = \{0, 1\}$. $\mathcal{D}_k$ contains all the data samples with label k. Multi-class classification can be easily supported by replacing LR with softmax regression classifier.

A common way for many classifiers to handle categorical features is one-hot encoding, which converts a categorical feature to a numerical vector. However, one-hot encoding is known to be prohibitively expensive when the cardinality is high.

LR models the conditional probability of y given an instance $\mathbf{x}$ using a sigmoid function:

$$p(y = 1|\mathbf{x}) = \sigma(\mathbf{w}^{\mathbf{T}}\mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{w}^{\mathbf{T}}\mathbf{x})} \qquad (1)$$

where $\mathbf{w}$ is weight parameters to be learned, making the decision boundary a hyperplane. The confidence score is controlled by the weighted sum of input features. LR assumes that there is a monotonic relationship between $p(y = 1|\mathbf{x})$ and x_d, while in practice often does not exist. DLR was proposed to fix this problem by embedding each feature x_d into a nonlinear representation:

$$\Phi(\mathbf{x}) = (\phi_1(\mathbf{x}), ..., \phi_D(\mathbf{x})).$$

By assuming all the attributes of $\mathbf{x}$ are conditionally independent given the label y, each attribute was represented using logit transformation:

$$\phi_d(\mathbf{x}) = \ln \frac{p(y = 1|x_d)}{p(y = 0|x_d)}. \qquad (2)$$

DLR supports both numerical and categorical attributes. For categorical attributes, $p(y = 1|x_d)$ is estimated by calculating the proportion of positive samples among all the samples whose d^{th} attribute is x_d. We use $\mathcal{D}_{x_d}$ to denote the set of samples in $\mathcal{D}$. The DLR estimates is

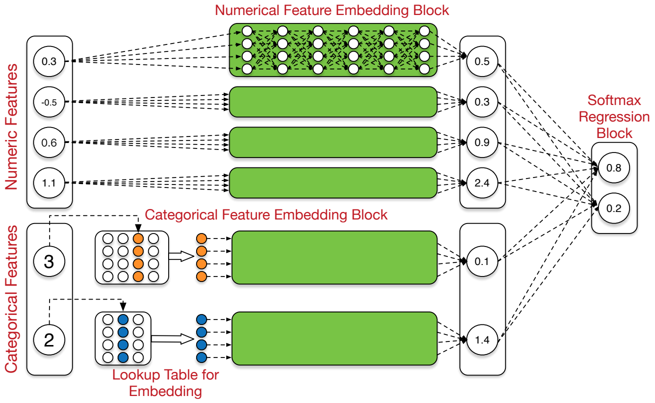

Fig. 1. Overall architecture of DELR

$$\phi_d(\mathbf{x}) = \ln \frac{|\mathcal{D}_1 \bigcap \mathcal{D}_{x_d}|}{|\mathcal{D}_0 \bigcap \mathcal{D}_{x_d}|} \qquad (3)$$

For continuous attributes, DLR uses a kernel density estimator with bandwidth h_d to estimate $\phi_d(\mathbf{x})$. The DLR estimate is

$$\phi_d(\mathbf{x}) = \ln \frac{\sum_{x' \in \mathcal{D}_1} \exp(-\frac{(x_d - x'_d)^2}{h_d^2})}{\sum_{x' \in \mathcal{D}_0} \exp(-\frac{(x_d - x'_d)^2}{h_d^2})} \qquad (4)$$

In DLR, the above nonlinear feature embedding is fed into an LR model for classification. Although DLR has nonlinearity and accountability, some serious drawbacks restrict its prevalence. First, DLR cannot be used in large scale datasets and real time applications due to its time complexity as discussed before. Second, DLR is an ad-hoc method, separating feature preprocessing and classification procedures. The expressing power of DLR is thus limited.

III. Deep Embedding Logistic Regression

In this paper, we propose a deep embedding logistic regression (DELR) framework, in which, no prior assumption for data distribution is needed. Instead of using kernel based estimator, we use deep neural networks for dimension-wise feature embedding. The overall architecture of DELR is depicted in Fig. 1. DELR contains three different blocks: numerical feature embedding block, categorical feature embedding block and logistic/softmax regression block.

A. Numerical Feature Embedding Block

For the numerical part, each feature is not necessarily in a monotonic relationship with class probability. To address this problem, we apply a multi-layer perceptron (MLP) for the numerical feature embedding as MLP has the ability to learn complex feature representation automatically. In order to reduce the total number of parameters to be learned, we design a deep and narrow MLP as shown in Fig. 1. Although the feature embedding block has 6 layers, each of which contains only 4 hidden neurons. Thus, the total number of parameters is only 80. Compared with classical MLP that has millions of parameters, our model greatly reduced the learning time in updating each parameters, being able to handle very high dimensional dataset. Rectified linear unit $A(x) = \max(x, 0)$ is used as activation function between two adjacent layers. Batch normalization [10] is also applied before every activation function. The input and output dimension of numerical feature embedding block are one. After the conversion, we have

$$f_d(\mathbf{x}) = A(b_d^{nh} + W_d^{nh} A(\cdots A(b_d^1 + W_d^1 x_d))) \tag{5}$$

where nh is a user defined number of hidden layers.

B. Categorical Feature Embedding Block

Suppose the categorical part $\mathbf{x}^C$ has D_2 features, $\mathbf{x}^C = (x_1^C, ..., x_{D_2}^C)$. The d^{th} categorical feature has K_d categories, $x_d^C \in \{1, 2, ..., K_d\}$.

The first component of the categorical feature embedding block is a lookup table that contains a numerical embedding for each category as shown in Fig. 1. The number of embeddings of each feature is equal to the number of categories of the corresponding categorical feature. At its core, each lookup table is a matrix $\mathbb{U}_d \in \mathcal{R}^{k \times K_d}$ where each column vector represents a k dimensional embedding for a corresponding category. $k > 0$ is a user defined integer. For example, in Fig. 1, we have $k = 4$. Each categorical feature would retrieve its corresponding embedding in the lookup table as its new feature representation.

Mathematically, we let $\mathbf{u}_i^d$ be the i^{th} column vector in lookup table $\mathbb{U}_d$. After embedding retrieve, we have the new feature representation $e_d(\mathbf{x}^C) = \mathbf{u}_i^d$.

This new representation is then fed into a new numerical feature embedding block to get the univariate output,

$$g_d(\mathbf{x}) = f_{D_1+d}(\mathbf{u}_i^d) \tag{6}$$

C. Classification

After obtaining the embedding for both numerical and categorical features, DELR concatenate them together to form a new representation, $\Phi(\mathbf{x})$, for the original input $\mathbf{x}$. Note that each feature in the new representation can be traced to its corresponding raw attribute. Now we have

$$\Phi(\mathbf{x}) = (f_1(\mathbf{x}), ...f_{D_1}(\mathbf{x}), g_1(\mathbf{x}), ..., g_{D_2}(\mathbf{x})) \tag{7}$$

We let $\phi_i(\mathbf{x})$ to denote the ith element of $\Phi(\mathbf{x})$.

LR is then used to estimate the probability that a given input is positive. We can extend our model to support multi-class classification by replacing LR with softmax regression.

Following the same training procedure as softmax regression, we minimize the cross entropy between true label distributions and prediction distribution. The embedding of each single dimension is learned jointly through stochastic back-propagation. To reduce the effect of over fitting, ℓ_2 normalization is applied when training the model.

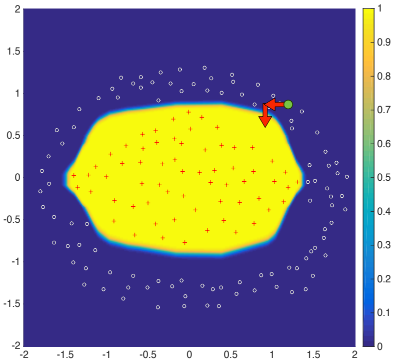

Fig. 2. Probability output of DELR on a toy example. The green dot represent an instance with the input value of $\{1.2, 0.8\}$. The red arrows are extracted actions to flip the label of greed dot.

IV. Discussions

In this section, we discuss the nonlinearity, interpretability, i.e. accountability and actionability of our proposed DELR model through a toy dataset shown in Fig. 2. We use red pluses and white circles to denote the positive and negative instances, respectively.

Nonlinearity. This dataset is not linearly separable as we cannot draw a single line that can perfectly split those two categories. LR obviously fails here. We train a DELR model using this toy dataset, and draw the probability output in this 2-D space. As we can see from Fig. 2, the decision boundary is a smooth circle, containing all the red pluses. The power of nonlinear separability granted by DELR yields a very reasonable separation curve on this 2-D space.

Accountability. Accountability refers to model's ability to reveal the significance of each feature in making the final prediction for some instance. In DELR, each attribute x_i is first transformed into $\phi_i(\mathbf{x})$ through numerical feature mapping layer. Final prediction is determined by the sign the following equation,

$$w_1\phi_1(\mathbf{x}) + w_2\phi_2(\mathbf{x}) + b \tag{8}$$

where b is the bias term. For this toy dataset, we find $b = 0$ after training. To discover the contribution of each input feature to the final prediction, we plot the value of $w_1\phi_1(\mathbf{x})$ and $w_2\phi_2(\mathbf{x})$ with the change of x_1 and x_2 in Fig. 3. These two figures clearly illustrate the correlation between the feature value and its corresponding contribution, which is not monotonic as in LR. We call them coordinate plots in the rest of the paper.

Now suppose we are given a data point $(x_1, x_2) = (1.2, 0.8)$, which is plotted as a green dot in Fig. 2. The model predicts it as negative. We can get the contributions of the two dimensions from Fig. 3, which are -0.21 and -8.4. At this point, both features are making negative contributions to the final prediction. Note that such accountability is not offered in neural networks. This is because features are intricately interwoven in the hidden layers of neural networks, making

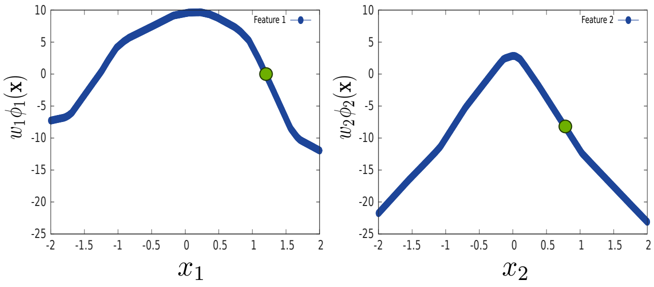

Fig. 3. The contribution of each input attribute for the final prediction. Left: The correlation between x_1 and the first component of Equation 8. Right: The correlation between x_2 and the second component of Equation 8.

it hard to account for each feature's contribution to the final outcome.

Actionability. Actionability is a much desired property in areas such as clinical prediction and market analysis. Classifiers need to not only identify critical features that lead to the prediction, but also make recommendations on how to change these features in order to clear the threats or achieve goals.

A practical constraint we should consider is that we can only change a limited number of input features. For example, in clinical predictions, it may be impractical to change all features of a patient simultaneously. Instead, we aim to make small changes to one or a few features, such as temperature, blood pressure or pulse, which are enough to revert the label.

In LR, each feature is independent, and has a constant slope in its contribution curve due to LR's linearity. Thus, we can make small changes on a few features with the largest absolute weights (slopes). In DELR, we can also look at the slope of each feature in the contribution curves as in Fig. 3. However, the slope is no longer a constant everywhere. Thus, we propose an iterative algorithm which selects features to change one by one. For an instance $\mathbf{x}$ that we want to revert its label, our iterative action extraction algorithm first selects the feature d with the maximum slope $\left|\frac{d\phi_d(\mathbf{x})}{dx_d}\right|$ and updates it by a small step. Then, the algorithm again selects the feature with the largest slope, but at the updated point. The process is iterated until reaching a budget. The selected features are returned for making changes.

The slope can be calculated through backpropagation from the second to last layer, or using the finite element method, both of which are computationally efficient. We use the same cost for changing each feature by a unit. The cost can also be non-uniform as we can use step sizes inversely proportional to the costs.

Let's again use the green dot in Fig. 2 for demonstration. Now that we know the label of this instance is negative, we want to make some changes on the input features to revert its label. Suppose we have a budget $B = 2$ and step size 0.25. At first, the slope of x_1 is larger, as seen from Fig. 3. Thus, the point moves left for 0.25. At the next point, the slope of x_2 becomes larger than that of x_1. Thus, the point moves down for 0.25 and reaches the positive region. In comparison, if we stick to one feature and do not iteratively update the slope calculation, it will cost more to reach the positive region either moving down or moving left.

V. Related Work

DELR is designed to preserve the interpretability with minimum sacrifice on the prediction performance.

Accountability has been studied for a long time. To reveal the feature importance, two different kind of algorithms have been proposed. The first category relies on analyzing the model directly. For example, the feature importance of LR and DLR is indicated by its corresponding weight parameters. When growing trees, RF can memorize the information gain for each feature and use this as an indicator of feature importance. Instead of identifying single feature importance, fast flux discriminant [11] first generates subsets of features and then use LR to identify the importance of subsets. The second category post-processes the well trained model when given an input instance. To check whether the prediction of DNNs is based on the right region of the image, guided back propagation [12] was proposed. Provided with an image and the well trained CNN, they run back propagation on the image space and visualize pixels corresponding to larger gradient values as important features. For DELR, we can not only show the feature importance using weight parameter, but also visualize the trending of each feature as it changes.

Among all classifiers, DELR resembles to gradient boosted decision stumps (GBDS) [13] most in terms of individual feature transformation. Different from gradient boosted decision trees (DBDT) [14], GBDS is an ensemble of one level decision trees. For each decision stump, only a single feature is used for classification. After training the GBDS, we can group decision stumps with same features together and draw plots similar to Fig. 3. However, the coordinate plot of DELR is much smoother than GBDS, leading to better generalizability. We will compare the model performance in the experimental section.

Compared with accountability, extracting knowledge from machine learning model is an even harder task. Rule based algorithms can be post-analyzed through pruning and summarization [15], [16]. In [17], the authors use a greedy algorithm to provide actions that can maximize the expected profit from the decision tree. Despite their actionability, decision trees cannot achieve high accuracy. Reference [18] further proposed an integer linear programming (ILP) algorithm to extract optimal actionable knowledge from random forests. However, ILP is a NP-Complete problem, restricting its usage in large scale datasets. For deep neural networks, meaningfully actionable knowledge is hard to extracted due to model complexity [19]. DELR inherits all the advantages of LR, especially efficiency, accountability, and actionability, making it capable of extracting knowledge from various kinds of datasets.

VI. Experiments

We conduct extensive experiments to evaluate DELR on several benchmark datasets. We first test the performance of

TABLE I

PERFORMANCE OF VARIOUS METHODS ON VARIOUS UCI DATASETS. NOTE THAT LR, DT, DLR AND DELR HAVE ACCOUNTABILITY. LR, DT, DLR, GBDT AND DELR HAVE ACTIONABILITY. "N/A" INDICATES MEMORY OVERFLOW.

Dataset	breast_cancer	splice	musk	mushroom	mnist38	nursery	adult	census-income
N	683	1,000	6,598	8,124	11,982	12,960	30,162	299,285
P	9	60	166	22	784	8	14	41
Data-type	Numerical	Categorical	Numerical	Mixed	Numerical	Categorical	Mixed	Mixed
LR	89.70±5.33	81.50±3.05	95.13±0.61	100±0.00	96.87±0.36	91.93±0.85	84.01±0.20	94.97±0.07
DT	90.70±1.27	87.80±1.94	96.49±0.66	100±0.00	96.10±0.26	98.61±0.14	79.56±0.41	93.06±0.07
DLR	96.42±1.61	92.40±2.68	95.20±0.74	100±0.00	N/A	N/A	N/A	N/A
GBDS	92.99±1.89	92.20±1.36	95.60±0.41	99.95±0.41	96.12±0.24	92.32±0.39	84.20±0.34	95.00±0.03
DELR	**97.57±1.16**	**93.00±0.55**	**97.57±1.16**	**100.00±0.00**	96.05±0.38	92.83±0.38	**84.57±0.28**	**95.43±0.14**
SVM-rbf	93.27±1.95	91.80±1.72	95.08±5.99	100±0.00	97.83±1.23	98.01±0.33	83.12±0.46	94.26±0.14
RF	94.27±1.62	94.50±2.11	97.39±0.25	100±0.00	98.73±0.25	99.06±0.21	83.82±0.21	95.25±0.09

DELR on UCI datasets[1], which contain both numerical feature datasets and mixed-type feature datasets. Then we exhibit the convenience and interpretability on classifying a large-scale dataset with a huge amount of categories. All tested datasets are publicly available.

A. Experimental setup

We compare DELR with the following four interpretable methods, 1) Logistic regression with ℓ_2 regularization (LR). 2) Decision Tree (DT) 3) Density-based Logistic Regression (DLR). 4) Gradient boosted decision stumps. We also add the performance of Support Vector Machines with RBF kernel (SVM-rbf) and Random Forest (RF) for reference. We implement DELR using PyTorch [20] and will release the code after publication. LR, DT, GBDS and RF are implemented with the $scikit-learn$ package [21]. The SVM-rbf is implemented by *libSVM* [22]. DLR is downloaded from the authors' homepage [3].

All the experiments are run on an off-the-shelve desktop with two 8-core Intel(R) Xeon(R) processor of 2.67 GHz, 128*GB* RAM and a Tesla P100 GPU. For each dataset, we run 5 fold cross validation and report the average performance and standard derivation. We further hold out 1/3 from the training set as the validation set. All algorithms are trained on the training set, choosing hyper-parameters based on the validation set and evaluated on the test set in each fold. Hyper-parameters are selected for all algorithms with Bayesian optimization [23] implemented in the $spearmint$[2] package.

B. Evaluation on UCI datasets

We evaluate the performance of DELR on several UCI benchmark datasets as shown in Table I. We show the dataset statistics on the left part of the table and the test accuracy on the right part. Here, *N*, *P* and *Data-Type* represents the number of instances, the number of features and data type in each datasets, respectively. We try to include datasets of different scales, covering small sized dataset such as "breast-cancer" to large scale datasets such as "census-income". Among all these datasets, "adult", "mushroom" and "census-income" contain both categorical feature and numerical features. "splice" and "nursery" only contain categorical features while the rest datasets only have numerical features.

TABLE II

SUMMARY OF THE TRIPTYPE DATASET

Feature Name	#Categories
VisitNumer	95,674
Weekday	7
UPC	97,715
ScanCount	39
adultDepartment Description	69
Fineline Number	5,196

TABLE III

TEST ACCURACY ON THE TRIPTYPE DATASET

Dataset	SFT	DT	DELR
Triptype	71.41%	61.86%	**72.59%**

From Table I, we can make the following observations. First, LR, the linear classifier performs the worst for the most of the time. Second, DELR performs the best among all classifiers on six out of eight datasets, demonstrating its superior nonlinear separability. For dataset nursery, DT outperform the rest interpretable models by a large margin, indicating the highly complex distribution of this dataset. Rule based algorithms handle extreme cases better in this case. DLR come across memory overflow on four largest datasets and DELR outperform DLR on all datasets, demonstrating the superior of deep feature embedding compared with kernel density estimator. In addition, DELR is time efficient. The running time on the smallest dataset is less than one minute and it takes about 3 hours to train the largest dataset. As we are using stochastic gradient descent, training time is proportional to the number of instances.

We further perform action extraction on adult datasets, determining whether a person makes over 50K a year. Given a negative instance, the algorithm suggests the person to switch his work class to Self-emp-inc or achieve doctoral education level. Further, if we set the cost of changing categorical features to a large number, then the algorithm suggests increasing working hour, which is quite reasonable.

[1] https://archive.ics.uci.edu/ml/datasets.html

[2] https://github.com/JasperSnoek/spearmint

C. *Evaluation on the Walmart dataset*

We evaluate DELR on triptype[3], a large real-world market analysis related dataset with many categories to demonstrate the distinctive advantages. This dataset is a transactional dataset of items purchased at Walmart. The goal of the task is to predict the type of each customer trip, which would help Warlmart's decision making in business and improve customers' shopping experiences. There are 38 trip types in total including a small daily dinner trip, a weekly large grocery trip, and so on. This dataset contains 647,054 instances, each of which contains 6 categorical features[4] .

The difficulty of mining this dataset lies in the huge amount of categories as shown in Table II. There are in total 198,700 distinct categories, leading to a 198,700-dimensional feature vector if one-hot encoding is used. Few classifiers can handle this dataset directly.

We only show the results of softmax regression (SFT), DT and DELR as RF, SVM-rbf and DLR cannot achieve any meaningful results within one day on this dataset even with state-of-the-art packages. For DELR, we adopt the same architecture as the previous experiment. We intentionally use the raw feature without any feature engineering. We show the test performance of each method on Table III. We can see that DELR again outperforms all the other classifiers.

Another advantage of DELR is that it can learn meaningful feature embedding and quantify categorical features. We interpret the category correlation through visualizing the output of the nonlinear feature embedding block. We set the embedding dimensionality to 2 and train DELR from scratch till it converges. Next, we plot the categories using their embeddings as coordinates. As ”Department Description” is the only feature whose semantic meaning of each category is released, we only present the visualization of this feature, which contains 69 categories as shown by blue circles in Figure 4. Due to space limit, we are not allowed to show all the category names in the figure. We can observe that similar categories are located closely while dissimilar categories are far away from each other. We can find many surprisingly meaningful clusters. In Fig. 4, the red cycle covers clothes related features, such as ”ladies' wear”, ”mens' wear”, etc. There are also some clusters relating to food, home decoration, horticulture, and so on. These results indicate that DELR learns meaningful embeddings, which helps knowledge extraction and further augments interpretability.

VII. Evaluation on real world clinical dataset

In this section, we apply DELR onto a real-world clinical dataset, performing 30-day postoperative mortality prediction. This work is done in partnership with Barnes-Jewish Hospital (BJH), one of the largest hospitals in the United States. Our data includes all preoperative, intraoperative and postoperative data combined with other inpatient and outpatient EMR data. More than 110,000 surgeries' data is collected between 2012 and 2016, each of which contains 44 preoperative EMR features and 49 vital signs. Thirty-day postoperative mortality is used as the output label. After data screening, we randomly split the dataset into training set (70,000 patients), validation set (10,000 patients), and testing set (19,791 patients), at the ratio of roughly 7:1:2.

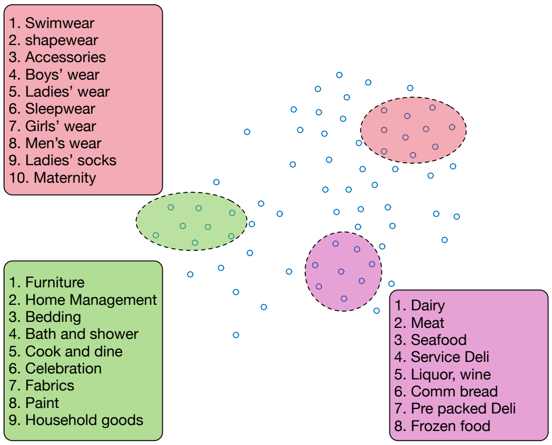

Fig. 4. The embedding for the ”Department Description” feature. Each point in this figure represents a unique category. We show three representative clusters.

TABLE IV
Experimental results on 30-day mortality prediction. DELR outperform all the baseline methods.

Method	AUROC	AUPRC	Specificity	Sensitivity
DT	0.6513	0.0598	0.95	0.3398
LR	0.8455	0.0749	0.95	0.4175
GBDS	0.8658	0.0911	0.95	0.4439
SVM	0.8609	0.0823	0.95	0.4417
RF	0.8536	0.0750	0.95	0.4175
DELR	**0.8725**	**0.0981**	0.95	**0.4515**

Preoperative data are static data collected from patients before the operations. 15 numerical features and 32 categorical features are includes in the Pre-op data. Intraoperative data are in the form of multi-variate time series of patients' vital signs and general signals monitored throughout patients' operations, in which, 10 vital signs are selected. We calculate its mean and standard deviation.

Our target outcome is 30-day mortality, which has a positive-negative ratio of approximately 1:100. We have tried two different methods to deal with this imbalance. The first method is to use class weights inversely proportional to their proportion to multiply the loss of positive training examples 100 times larger than that of negative training examples. The second method is to upsample positive training examples by 100 times each. We tested all baseline methods and DELR and all of them perform better using the second upsampling method. All the following experiments use upsampling method.

We evaluate all the forecasting models using well-accepted criteria including: Area Under the curve of Receiver Operating

[3] https://www.kaggle.com/c/walmart-recruiting-trip-type-classification

[4] Note that only the training set is available online, we randomly select 70% as the training set and the rest as the test set.

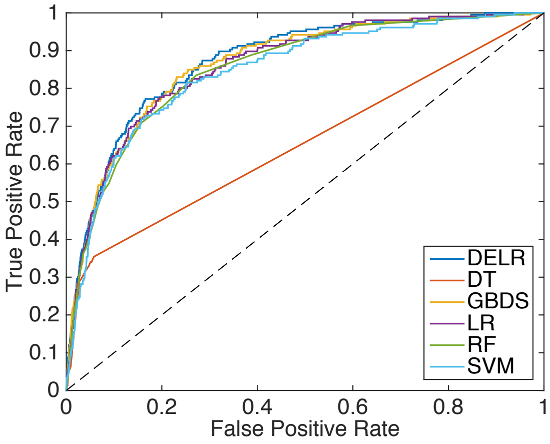

Fig. 5. Receiver Operating Characteristic (ROC) of the model performance on the test set.

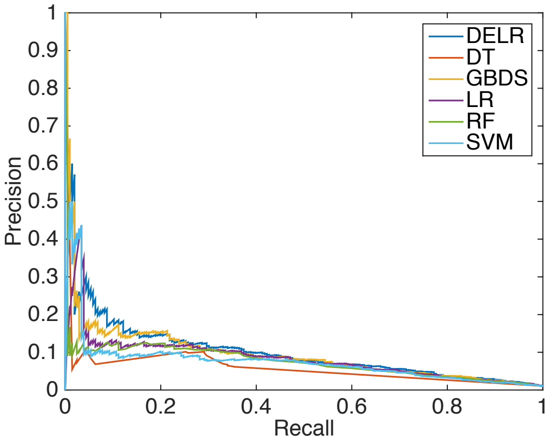

Fig. 6. Precision-Recall curve of the model performance on test set.

Characteristic (AUROC), sensitivity and specificity.

In addition, as our data is extremely imbalanced, Receiver Operating Characteristic (ROC) can be sometimes deceptive on evaluating the model performance [24], [25]. Therefore, we also show the Precision-Recall curve of each model and adopt Area Under Precision-Recall Curve (AUPRC), which measures the average precision as an additional evaluation criterion. All the models are tuned based on the AUPRC performance on the validation set and then report the model performances evaluated on the testing set.

Another key metric we evaluate and compare is the sensitivity at 95% specificity since it is important to maintain a high specificity (i.e., low false alarm rate) for meaningful clinical decision support.

A. Baseline Methods

We compare DELR with the most wildly used classifiers, decision tree (DT), logistic regression (LR), support vector machine (SVM) and random forest (RF) and gradient boosted stumps (GDBS). Categorical features are represented as one-hot encoding vectors for baseline methods.

B. Experimental Results

Table IV shows the model performances. We observe that DELR consistently outperforms both interpretable models and non-interpretable models in terms of AUROC and AUPRC. The reason DELR can beat SVM and RF is that this clinical dataset is very easy to overfit. Our model regularizes very well in this scenario. We will discuss more about it in the next paragraph. Sensitivity at 95% specificity level is included in the performance chart as well. Even under this high specificity, DELR achieves 0.4515 sensitivity, which is much higher than rest models. We also plot out all the ROC curves in Figure 5. DELR achieves the highest AUROC of 0.8725. As our dataset is extremely imbalance, the Precision-Recall curve shown in Figure 6 contains more meaningful information. The positive-negative ratio is approximately 1:100, indicating that random guess can get only 0.01 AUPRC. DELR achieves an AUPRC of 0.0981, which is nearly ten times better than random guess and more than 30% gain compared with the LR model.

Next, we check the interpretability of DELR on this clinical dataset. As we've talked in the discussions section, coordinate plot of each feature can be drawn to visualize its relationship with 30-day mortality. We plot out eight features, shown in Fig. 7 and Fig. 8. The x-axis is the feature value and y-axis is the feature contribution to the final prediction. Positive value enhance the probability of 30-day mortality. Among these eight plots, the first four plots in Fig. 7 show nonlinear relationship between the input feature and final contribution to the prediction. Classifiers such as logistic regression cannot handle this situation. In addition, the nonlinear transformations are also in line with our expectation. We also discover some features remain linear after transformation as shown in Fig. 8. Take SpO2 as an example, the higher the value of SpO2, the less likely the patient will die within 30 days. We've checked SVM-rbf and RF, no such linear relationship is found. Without mapping all features to complex nonlinear representation, DELR can generalize better. We will further analyze DELR on this clinical dataset with domain experts in the future based on coordinate plots and extract reliable suggestions from this model.

VIII. Conclusions

Complex models such as DNNs have strong learning ability and high accuracy. However, in many tasks such as marketing and clinical prediction, LR is still a preferred choice as it provides good interpretability and scales well. In this paper, we propose a novel DELR model, inheriting those nice properties of LR while overcoming its drawbacks of linearity and inability to handle categorical features. DELR incorporates dimension-wise nonlinear feature embedding using deep neural networks and feeds the embeddings into LR for classification. Extensive analysis and experimental results demonstrate that DELR is a compact yet powerful model, achieving both high accuracy and excellent interpretability.

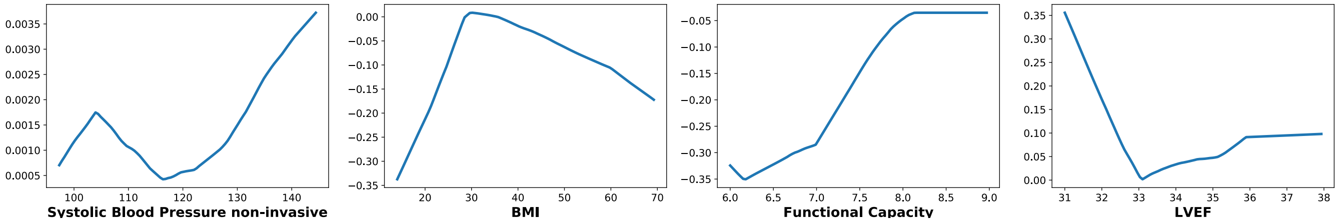

Fig. 7. We select four features that is not in linear relationship between the input value and 30-day mortality rate.

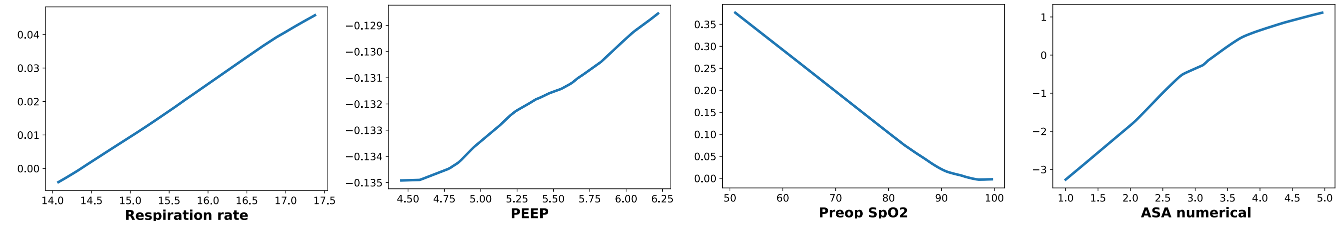

Fig. 8. We select four features that remains linear after feature embedding. This indicates that DELR regularizes very well, not memorizing the input data through learning complex decision rules.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[2] J. Zhu and T. Hastie, "Kernel logistic regression and the import vector machine," *Journal of Computational and Graphical Statistics*, vol. 14, no. 1, pp. 185–205, 2005.

[3] W. Chen, Y. Chen, Y. Mao, and B. Guo, "Density-based logistic regression," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013, pp. 140–148.

[4] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, vol. 1, no. 2, 2017, p. 3.

[5] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4960–4964.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[7] G. Alain and Y. Bengio, "Understanding intermediate layers using linear classifier probes," *arXiv preprint arXiv:1610.01644*, 2016.

[8] Z. Che, S. Purushotham, R. Khemani, and Y. Liu, "Interpretable deep models for icu outcome prediction," in *AMIA Annual Symposium Proceedings*, vol. 2016. American Medical Informatics Association, 2016, p. 371.

[9] H. Chen, R. H. Chiang, and V. C. Storey, "Business intelligence and analytics: from big data to big impact," *MIS quarterly*, pp. 1165–1188, 2012.

[10] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[11] W. Chen, Y. Chen, and K. Q. Weinberger, "Fast flux discriminant for large-scale sparse nonlinear classification," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 621–630.

[12] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv preprint arXiv:1412.6806*, 2014.

[13] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, ser. Springer Series in Statistics. New York, NY, USA: Springer New York Inc., 2001.

[14] T. Chen, "Introduction to boosted trees," *University of Washington Computer Science*, 2014.

[15] B. Liu and W. Hsu, "Post-analysis of learned rules," in *AAAI/IAAI, Vol. 1*, 1996, pp. 828–834.

[16] B. Liu, W. Hsu, and Y. Ma, "Pruning and summarizing the discovered associations," in *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1999, pp. 125–134.

[17] Q. Yang, J. Yin, C. Ling, and R. Pan, "Extracting actionable knowledge from decision trees," *IEEE Transactions on Knowledge and data Engineering*, vol. 19, no. 1, pp. 43–56, 2007.

[18] Z. Cui, W. Chen, Y. He, and Y. Chen, "Optimal action extraction for random forests and boosted trees," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 179–188.

[19] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[20] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[22] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM transactions on intelligent systems and technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[23] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Advances in neural information processing systems*, 2012, pp. 2951–2959.

[24] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 233–240.

[25] K. Boyd, V. S. Costa, J. Davis, and C. D. Page, "Unachievable region in precision-recall space and its effect on empirical evaluation," in *Proceedings of the... International Conference on Machine Learning. International Conference on Machine Learning*, vol. 2012. NIH Public Access, 2012, p. 349.