Progressive Feature Upgrade in Semi-supervised Learning on Tabular Domain

1st Morteza Mohammady Gharasuie
Old Dominion University
mmoha014@odu.edu

2nd Fenjiao Wang
Old Dominion University
f1wang@odu.edu

Abstract—Recent semi-supervised and self-supervised methods have shown great success in the image and text domains by utilizing augmentation techniques. Despite such success, it is not easy to transfer this success to a tabular domain. The common transformations from image and language are not easily adaptable to tabular data containing different data types (continuous and categorical data). There are a few semi-supervised works on the tabular domain that have focused on proposing new augmentation techniques for tabular data. These approaches may have shown some improvement in datasets with low-cardinality in categorical data. However, the fundamental challenges have not been tackled. The proposed methods either do not apply to datasets with high-cardinality or do not use an efficient encoding of categorical data. We propose using conditional probability representation and an efficient progressively feature upgrading framework to effectively learn representations for tabular data in semi-supervised applications. The extensive experiments show the superior performance of the proposed framework and the potential application in semi-supervised settings.

Index Terms—Semi-supervised learning, Feature representation, Pseudo-label, Tabular domain

I. INTRODUCTION

Since the major breakthrough in the ImageNet Large Scale Visual Recognition Challenge, deep learning has attracted much attention due to its superior performance in many applications, e.g., Speech Recognition, Computer Vision, and Natural Language Processing. Such great progress is largely driven by enormous datasets. Collecting and labeling such an enormous dataset is expensive, time-consuming, and often impossible.

Recently, semi-supervised learning [1], [2] has gained a lot of attention due to superior performance in the image and language domain. Researchers have proposed different augmentation techniques and regularizations (consistency regularization) in semi-supervised learning approaches [2]. Often, these augmentation techniques are domain specific. Theoretically, the data augmentation technique in semi-supervised learning can help increase the generalization ability of the trained models by reducing the overfitting and expanding the decision boundary of the models.

However, the success of semi-supervised learning approaches from image and language domains heavily relies on the spatial or semantic structure of image or language data, which is absent in the tabular data (because of no explicit structure). The augmentation technique like Mixup [3] works well if the data manifold is likely convex, Whereas the tabular data is likely not convex. So, the known augmentation techniques in other domains can easily create out-of-distribution samples in tabular data, which may hurt the learning process [4].

Data augmentation and loss function development can be a straightforward ideas for solving SSL in the tabular domain. Data augmentation is more challenging in tabular data because of the lack of explicit structure and the mix of different data types. Some recent works on semi-supervised learning to tabular data have focused on proposing new augmentation operations; however, the new augmentation technique can not fundamentally solve the problem due to the high-cardinality of categorical data and inefficient representation.

We propose to use a conditional probability representation instead of one-hot encoding for categorical data to enjoy several unique benefits mentioned in the next paragraph. We take a step back and carefully examine how the categorical data can be represented in tabular data for the semi-supervised learning problem. Representation of the data is an important aspect that can easily be ignored. Table I shows that the choice of representation for categorical data may greatly impact the performance. Also, this impact may even be agnostic to what follow-up model is being used.

We propose utilizing conditional probability representation (CPR) for semi-supervised learning in the tabular domain. The CPR maps individual categorical values to the probability estimate or the expected value of the target attribute. In another word, it computes the likelihood of a specific categorical value leading to a particular label. It has many unique benefits compared to other representations (one-hot encoding, label encoding). Firstly, it is an efficient representation in terms of how many bits are used to represent the feature, especially for high-cardinality categorical data. The reason being that the number of dimensions of the CPR does not depend on the cardinality of the categorical feature. It only depends on the number of target labels. Secondly, label information has been baked into the representation. Label information is critical for a semi-supervised learning algorithm. If one can inject label information into the feature, it may be easier for the model to learn meaningful representations. More importantly, it opens the door for utilizing pseudo-labels (predicted labels) in a novel way (constructing the feature).

Thirdly, compared to other representations, the CPR is closer to the numerical features, since it uses conditional
probabilities as features. This property may open the door for better leverage of various existing augmentation techniques for tabular data. Instead of only using true labels to construct the CPR, we propose to use pseudo-labels to update the CPR during the model training process. Our initial study shows that even if we don’t change how much data is used for training, but only increase the amount of data used for constructing the CPR, the prediction accuracy can be hugely boosted (Table II). Pseudo-labels are defined as predicted class labels for unlabeled data. Self-training algorithm utilizes pseudo-labels of unlabeled data to continue improving the model. It is just one way of utilizing pseudo-labels which is treating pseudo-labels as if they are the ground-truth labels in training the model. However, pseudo-labels can also be used in another way which is to update the CPR and then influence the model training process. With the model being trained progressively, more accurate pseudo-labels will help generate more accurate representations for learning in the “feedback loop”.

We propose a framework that can progressively upgrade the CPR representation and act as an add-on component to the existing semi-supervised learning frameworks. For this purpose, the SSL framework needs one component to produce pseudo-labels for unlabeled data, which is used for updating the CPR representation. Because pseudo-labels are not 100% correct, we propose some refinement mechanisms to select the pseudo-labels that are more likely correct.

The main contribution of our paper can be summarized as follows.

- We propose using conditional probability representation for high-cardinality categorical data for efficient representation. To the best of our knowledge, our work is the first work that uses an encoding different from one-hot encoding for the tabular domain in semi-supervised learning. The proposed framework can also be extended to other encoding methods such as target encoding ([5]), which also bakes the target label into the representation.

- We propose a novel feature upgrading framework by leveraging pseudo-labels. To the best of our knowledge, we are the first paper proposing to use pseudo-labels for updating CPR for categorical data in semi-supervised learning.

- The proposed framework is flexible and complementary which can be easily embedded into the existing semi-supervised learning algorithms to boost the learning performance.

- We demonstrate the superior performance of the proposed framework in extensive experiments. The robustness of the proposed framework has been testified by superior performance on two different semi-supervised algorithms and three different tabular datasets.

The paper is organized as follows. In Section 2, we provide a review of the related semi-supervised learning algorithms and works on the representation for categorical data, noting the importance of using and updating the conditional probability representation. In Section 3, we describe the details of the proposed framework. Then, we present the quantitative and qualitative experimental results in Section 4. Finally, Section 5 gives the conclusion.

II. RELATED WORKS

A. Semi-supervised Learning

Semi-supervised learning in general is attempting to improve the performance of the learning algorithms by utilizing both the labeled and unlabeled data, such that the resulted classifier is better than the trained classifier on just labeled data [6]. Recently, the semi-supervised learning has shown considerable progress in the language and image domains. Most of these works resulted from the consistency regularization and pseudo-labeling on the unlabeled data.

Consistency Regularization: The consistency regularization uses the different perturbations of an input sample and tries to enforce the same prediction for all the perturbations. These perturbations can be applied on either different epochs [7] or same epoch [8]. Also, the perturbation can be applied in the network (dropout, random max-pooling), the input space [9], and the latent space [10].

Pseudo-labeling and self-training: The goal of pseudo-labeling [11] and self-training [12] refers to a classical semi-supervised approach where the model is being trained on the labeled and unlabeled samples using labels and pseudo-labels associated with the unlabeled samples. The self-training [13] has recently shown improved performance over supervised counterpart. Some works [14] use calibration and uncertainty of predictions for the selection of samples to improve the pseudo-label selections.

B. Representation for tabular data

Unlike image and language domains, tabular domain is a combination of different data types (numeric and non-numeric data). General idea is to encode non-numeric data to numeric format and use it in machine learning algorithms. The classic approach is one-hot encoding, which is unsuitable for high-cardinality categories in large datasets due to generating high-dimensional vectors and posing computational problems [15]. There are some works that try to tackle the problem of high cardinality. Cerdar P. Varoquax G. [15] propose min-hash encoding and Gamma Poisson factorization. They also propose similarity encoding [16] to encode dirty, non-curated categorical data. Also, Slakey A. et al. [17] proposed a CBM encoding approach to represent categorical features in low dimensions.

Some works from NLP community have studied pre-training of some language models (LMs) over tables on the Web. These models convert the tables to row-wise sentences or table-wise sentences. For instance, Tapas [18] is a framework for question answering over tables, which extends BERT’s architecture to encode tables by converting the entire table to sentences. TaBert [19], a pre-trained LM that jointly learns representation for natural language sentences and tables in the question-answering domain. TURL [20] is a structure-aware Transformer encoder framework that learns contextual
representations on relational tables of the Web using the row-column structure.

C. Semi-supervised Learning in tabular domain

Recent advances in semi-supervised learning using deep neural networks have been applied to the tabular domain. Darabi S. et al. [21] proposed a semi-supervised framework for the tabular domain, called "Contrastive Mixup". Yoon J. et al. [4] used consistency loss among perturbed versions of the one-hot encoded input samples (swap noise), called VIME. These two frameworks are discussed in more detail in the methodology section. Also, Ucar T. et al. [22] introduced a new framework of multi-view representation learning for tabular data. It reconstructed the input from the subsets of features rather than the corrupted input in an autoencoder. Furthermore, the proposed framework also included the idea of contrastive loss to further improve the performance.

Recent works only use one-hot encoding and experiment on small datasets containing low-cardinality categorical features. One-hot encoding usually works fine on these small datasets because the one-hot encoded features are not big. But in the case of big datasets with high cardinality, the one-hot encoding is not feasible. There are many different encodings for categorical data and several Python libraries offer implementation of the encoding methods \(^1\) and \(^2\). Usually, the different encoding methods may lead to different performance on different datasets. Study the impact of different encodings in the context of self-supervised learning is interesting yet challenging problem. In this paper, we propose using a conditional probability representation (CPR) that uses label of data for creating a numerical and continuous representation of the categorical data. We also propose a new framework that uses the CPR and progressively updates the representation for categorical features. Intuitively, this representation is more friendly to existing augmentation techniques than other representations (one-hot encoding or label encoding).

To the best of our knowledge, our work is the first work that uses an encoding different from one-hot encoding for the tabular domain in semi-supervised learning. In this regard, we considered big datasets in experimental sections to show the effectiveness of our work.

III. METHODOLOGY

In this section, we describe our proposed framework and the conditional probability representation (CPR), Update Policy and Refinement methods for the proposed framework. Then, we introduce progressive VIME and Progressive Contrastive Mixup.

A. Preliminaries

We formulate the semi-supervised problem for presenting the method. Consider a dataset with N samples. There are a small subset of labeled samples \(D_L = \{(X_n, y_n)\}_{n=1}^{N_L}\) and a large set of the unlabeled samples \(D_U = \{(x_n)\}_{n=L+1}^{N}\)

\(^1\)https://contrib.scikit-learn.org/category_encoders/
\(^2\)https://dirty-cat.github.io

TABLE I

| Representation                | Test Accuracy |
|------------------------------|---------------|
| Label Encoding               | 55.3%         |
| One-hot encoding             | 69.23%        |
| Conditional probability encoding | 73.33%        |

where \(N = N_U + N_L\). We consider the setting where \(N_U >> N_L\). The supervised training on labeled samples without learning from unlabeled samples most likely causes overfit. The unlabeled samples can be used to improve the generalization of the model to get better accuracy on unseen test samples. For this purpose, we use pseudo-labels and an Update Policy for better generalization in the training of the neural network, and also a better representation of data that helps boost the performance.

B. Conditional Probability Representation

In case of big datasets containing high-cardinality categorical data, the one-hot encoding is not the best choice because of the space consumption and curse of dimensionality problems [23]. In contrast, the CPR of the categorical data has a fixed representation w.r.t the number of targets in the classification problem, which creates a compact representation because usually the number of labels is much smaller than the cardinality of the features. Table I compares the performance of a Multi-layer perceptron (MLP) network on the CPR and two other encodings using Traffic Violations dataset. This experiment shows that the MLP using CPR can outperform the same model while using one-hot encoding or label encoding. Please note that different encoding methods may perform differently across different datasets/applications.

To define the problem mathematically, let \(X\) be an \(N \times M\) data matrix with row vectors \(X_n\) and column vectors \(X_m^T\). Let \(Y\) be an \(N\)-dimensional target vector and \(Y_n\) is the observed value correspond to \(X_n\). Then \(D = (X, Y)\) is a dataset with \(N\) samples where \(D_n = (X_n, Y_n)\) is the \(n\)th sample with label \(Y_n \in \{0, 1, ..., C\}\). Let feature column \(X_m^T\) be a categorical feature column with cardinality \(K_m\). It has a domain \(V(m) = \{X_{n,m}\}_n\) containing unique nominal values \(V_m \in 1, ..., K_m\). Let \(C\) be the number of the values in the target variable. The CPR measures given each category value in a categorical feature, how likely this category value lead to different target labels within the dataset. Therefore, for each categorical feature, a \(C\)-dimensional vector representation will be produced, where \(C\) is the number of target labels in the dataset. The following equation computes the CPR.

\[
X_{n,m} = [N_{n,m,1}, N_{n,m,2}, ..., N_{n,m,C}]
\]

where \(N_{n,m,c}\) is the number of observation of categorical value \(X_{n,m}\) that belongs to the label target \(c \in C\), and \(N_{n,m}\) is the number of observation of categorical value \(X_{n,m}\) in \(X_m^T\).

The summation of all the conditional probabilities is 1.
We design \textit{Update Policy} that incorporates pseudo-labels of the unlabeled samples to update the CPR. More samples help generate more efficient representation. Our initial study in Table II shows that the more labeled samples used for generating representation indeed drastically improve the model performance. This study sheds a light on our approach that using more “labeled” samples may boost the model performance. We get better accuracy using more samples to generate the CPR representation. The difference in this study is that we used the ground truth labels for statistics, however, in reality, pseudo-labels are being used.

In this approach, we use labeled samples $D_L$ at first to calculate the conditional probability on categorical features and generate the initial representation of all samples in the dataset ($D$). This representation will be updated using both labels for $D_L$ and pseudo-labels for $D_U$, and keep on training the semi-supervised model on the updated representation by updating the representation using more samples ($D_L + D_U$), the model can obtain better generalization since the representation contains more information from the dataset.

\textbf{D. Refinement}

We propose \textit{refinement} mechanism for handling noise in pseudo-labels. It works by filtering out likely incorrect pseudo-labels. It helps generate a more accurate representation in \textit{Update Policy} and improve the performance of the trained model. Some methods are introduced to choose more accurate pseudo-labels. Note that, though how to find more accurate pseudo-labels have been discussed in these papers, our main idea of utilizing CPR and keep updating the representation progressively is different from the methods proposed in these papers. These works use weight of pseudo-labels in the graph-based label propagation [24], the confidence of pseudo-labels in a classifier [9], uncertainty weight for each sample [25], [26] or using all pseudo-labels without \textit{refinement} [27].

If the label-propagation method is a component in the method, we use measured weights in the label-propagation method for filtering. If the classifier is available, we can use the confidence of pseudo-labels in the classifier for filtering. When both label-propagation and classifier are used in the architecture, we propose a mechanism to leverage both components for filtering. Two steps of filtering are used in the proposed mechanism. First, we keep only those pseudo-labels agreed between classifier and label propagation methods. Then, the final pseudo-labels are selected based on a threshold on the measured weights by the label-propagation method. After the second step, the final pseudo-label and its corresponding sample are used by \textit{Update Policy} to update the CPR on $D$. When one of the components is available in the architecture, the agreement can not be used. Instead, a threshold on the confidence of prediction in the classifier or the weight of pseudo-label in label-propagation is used for filtering.

\textbf{E. Progressive Framework}

Progressive training refers to updating the input feature representation to train the model. We believe that by updating representation using pseudo-labels, it is possible to train a more effective model. This section shows how \textit{Update Policy} and \textit{Refinement} are used in the progressive architecture. In this regard, the progressive architecture is plugged into two existing semi-supervised learning architectures [4], [21] that have extended the recent advances in semi-supervised learning to the tabular domain. In the following, we shortly describe the previous works, then expand them to the proposed progressive approach. We utilize \textit{Update Policy} and \textit{Refinement} for expansion.

\textit{1) VIME:}

Before introducing the proposed Progressive VIME, in this section, we first describe VIME [4]. VIME has two training steps. A self-supervised task that learns the data representation in step 1 using a denoising autoencoder with corrupted inputs (an augmentation method through mask and pretext generator components). This autoencoder has two decoders. One decoder (feature vector estimator) reconstructs the original input sample, while the other decoder (mask vector generator) learns to identify the inconsistency between feature values. Step 2 uses a consistency regularization for semi-supervised learning. The predictor in the second step uses the pre-trained encoder and augmentation from step 1 to train the predictor using labeled and unlabeled samples. Several corrupted samples (augmented inputs) are used to compute the consistency loss for training the predictor.

\textit{2) Progressive VIME:}

We describe how the progressive architecture is added to VIME. The black box in Figure ?? shows the added \textit{Update Policy} to generate more efficient representation. The representation using more data in Table II shows that the more labeled samples used for generating representation indeed drastically improve the model performance.
The first run stands for training the encoder and the predictor in both steps. Refinement, and representation generation. Added to the architecture for training the encoder: classifier, pseudo-labels, and representation generation.

**Policy Component** which is used to generate the CPR before feeding samples to the model. We introduce the term **run** that stands for training the encoder and the predictor in both steps. So the first run is equivalent to the original VIME.

In the beginning, the **Update Policy** uses labeled samples to generate the CPR representation because pseudo-labels for unlabeled samples are not available. At the end of each run, the latest pseudo-labels are available which are used in Update Policy (black box in Figure 1) for updating the CPR before continuing the next run. In case of using the refinement inside the Update Policy, the samples with unreliable pseudo-labels are filtered out and the remaining samples are used for updating CPR. Otherwise, all samples are used for regenerating the CPR. The Representation Regeneration in Update Policy uses the labeled and unlabeled samples to generate the new representation for categorical features. After updating the CPR, the next run is started on the latest CPR representation.

For **progressive VIME**, we propose a **two-step** refinement mechanism. A pseudo-label agreement is checked first between the label-propagation and the classifier. Then, further refinement is based on a threshold on the measured weights from the label propagation component. For the ablation study in terms of the effectiveness of different components in the proposed architecture, there may be only one component available, we filter out samples based on the confidence of predictions in the classifier or weights in the label propagation component.

3) **Contrastive Mixup:**

Before describing the progressive architecture, we review the **Contrastive Mixup** [21] first. Contrastive Mixup leverages mixup-based augmentation in latent space for contrastive learning. The **Contrastive Mixup** has two training steps like VIME. The first step is to train an autoencoder using labeled and unlabeled data by contrastive learning and reconstruction. The mixup operations are applied in the latent space on the samples with the same labels for contrastive learning. Note, unlike most mixup-based augmentation methods that they randomly select samples for mixup. **Contrastive Mixup** select samples with the same labels for mixup operation. Supervised contrastive learning is applied between the original and mixed samples. The label-propagation [24] is used after warm-up to generate or update pseudo-labels on unlabelled samples. Pseudo-labels will eventually be used for contrastive learning. In the second step of the training, a predictor is trained using consistency loss as a regularization. The transparent components in Figure 2 show the original architecture of **Contrastive Mixup**. The highlighted black box and red box are components proposed in the progressive architecture.

4) **Progressive Contrastive Mixup:**

We introduce **Progressive Contrastive Mixup** where **Update Policy** and **Refinement** are plugged into the **Contrastive Mixup** [21]. All components in Figure 2 show the proposed progressive architecture. The highlighted components in Figure 2 show the changes compared to the original architecture. We propose adding a classifier connected to the encoder to offer an alternative way to produce pseudo-labels as well as confidence on the pseudo-labels.

Like **Progressive VIME**, we need to update the representation using **Update Policy**. In **Update Policy**, we can use all pseudo-labels (named with update in experiments) or a subset of them using **Refinement** component (named with refinement in experiments). We believe that the **Refinement** may improve the accuracy because of using more accurate pseudo-labels to generate the representation.

There are different strategies for performing **Refinement** in **Progressive Contrastive Mixup** based on the availability of both label-propagation and the proposed classifier components in the architecture. We propose a **two-step** filtering mechanism (illustrated in **Pseudo-label Refinement** component in Figure 2) when both of components are available. Firstly, the pseudo-label agreement between the label-propagation and classifier is used to select more likely correct pseudo-labels. Then, the pseudo-labels are filtered based on a threshold on the measured weights by the label-propagation method. Based on our study, we find that label-propagation weights are more robust than the classifier’s confidence scores. If the classifier is not available, the refinement is **one-step** based on a threshold on the measured weights by the label-propagation. We found 0.9 as a threshold for most of the experiments.

**IV. EXPERIMENTAL STUDY**

This section shows the results of progressive architectures on different tabular datasets with the high-cardinality categorical data. Also, the progressive architecture is compared with **VIME** [4] and **Contrastive Mixup** [21] on three datasets. A detailed study of the performance of using different components in progressive architectures and the original architectures is presented.

**A. Tabular Datasets**

To demonstrate the efficacy of the CPR, **Update Policy** and **Refinement** on the progressive architectures, we conduct a series of experiments on three datasets: Traffic Violations 3, Drug Directory 4 and Display Advertising Challenge 5. All

3[https://catalog.data.gov/dataset/traffic-violations-5fdda](https://catalog.data.gov/dataset/traffic-violations-5fdda)

4[https://www.fda.gov/drugs/drug-approvals-and-databases/national-drug-code-directory](https://www.fda.gov/drugs/drug-approvals-and-databases/national-drug-code-directory)

5[https://www.kaggle.com/c/criteo-display-ad-challenge](https://www.kaggle.com/c/criteo-display-ad-challenge)
datasets have multiple high-cardinality categorical data. We shortly introduce each dataset and describe the cardinality of some features in the following. More details of the datasets are shown in Table V.

Both Advertising Challenge dataset (one million samples are used for experiments) and Traffic Violation dataset (1,578,154 samples) have cardinality higher than 200,000. Moreover, Drug Directory dataset, the smallest in our experiments, has 19,764 samples with the cardinality 5,032. All datasets have multiple categorical features. The Display Advertising Challenge dataset has 11 numerical and 26 categorical features. Two other datasets have a combination of DateTime, numerical, boolean, and categorical features.

For preprocessing, the DateTime feature in Traffic Violations is excluded and in Drug Directory dataset is converted to numerical values. All numerical features are normalized using standard scaler in scikit-learn library⁶. Also, categorical features are converted to numerical features using the CPR.

In the experiments, we use 80% of the data as the train set and 20% as the test set. Also, 10% of the train set is chosen as the labeled set and the rest as the unlabeled set. This division is used in all experiments. The prediction accuracy on the test set is used as the metric for evaluation. We use division is used in all experiments. The prediction accuracy chosen as the labeled set and the rest as the unlabeled set. This process is repeated over 5 runs.

We report the performance of original VIME method along with Progressive VIME. This evaluation shows how VIME works on the CPR and how the Progressive VIME performs versus VIME.

### B. Experiments

This section evaluates the proposed progressive architectures on all the aforementioned datasets. To explore the efficacy of our progressive semi-supervised framework on limited labeled data in practical setting, we compare the accuracy with the state-of-the-art methods by varying components in both proposed progressive architectures.

1) **Progressive VIME:**

We report the performance of original VIME method along with Progressive VIME in Table III. We evaluate the performance gain of each component in both VIME and Contrastive Mixup. This evaluation shows how VIME works on the CPR and how the Progressive VIME performs versus VIME.

The components and the training methods that are shown in the Table III for evaluation are described in the following:

- **Supervised:** Train the predictor in the second step using the labeled data.
- **VIME:Self-Supervised:** The encoder is trained in step 1 and the predictor is trained in step 2. Only labeled data is used in the second step to train the predictor.
- **VIME:Semi-Supervised:** Similar to self-supervised method, there are 2 steps in training. Difference is that in the second step, the data augmentation and consistency loss are used for training the predictor.
- **Progressive VIME:Self-Supervised with Update:** Adding Update Policy and several runs of training to

The summary of datasets. The number of categorical and the number of non-categorical features in each dataset are shown. Most of the features are categorical. More than one feature in each dataset has high cardinality.
the VIME:Self-Supervised method. All pseudo-labels are used for updating the representation (CPR).

- **Progressive VIME:Semi-supervised with Update:** Adding Update Policy and several runs of training to the VIME:semi-supervised method. All pseudo-labels are used for updating the representation.

- **Progressive VIME:Self-Supervised with Refinement:** Adding Update Policy and several runs of training to the VIME:semi-supervised method. Updating the representation in Update Policy component using more confident pseudo-labels in the predictor.

- **Progressive VIME:Semi-Supervised with Refinement:** Adding Update Policy and several runs of training to the VIME:semi-supervised method. Selection of pseudo-labels based on the confidence of the predictor to update the representation in Update Policy component.

Table III shows the proposed Progressive VIME with refinement outperforms the other methods, resulting in the best prediction performance. The VIME underperforms on Drug directory dataset in comparison with supervised method, while progressive method is more robust and provide consistent improvement on all datasets. In other words, the progressive methods with update and refinement always achieve better performance compared with the original VIME and supervised methods. The higher model’s predictive power shows the advantage of the progressive approach in leveraging the unlabeled data and learning better representations. The progressive training with refinement performs slightly better than progressive training with update because it only keeps likely correct pseudo-labels, which results in better representation regeneration.

2) **Progressive Contrastive Mixup:**
In this section, we compare our progressive architecture with the Contrastive Mixup using the same aforementioned datasets. We compare the performance of different models. All the models used for comparison are introduced as follows.

- **Supervised Model:** using predictor in the second step and training it on the labeled samples.
- **Contrastive Mixup:** training the original Contrastive Mixup without updating the CPR. Training the encoder in the first step, then training predictor in the second step.
- **Progressive Contrastive Mixup with Update:** This method adds Update Policy to Contrastive Mixup and updates the representation using all pseudo-labels. When the classifier is used, it just effects in training of the representation and do not participate in updating the CPR.
- **Progressive Contrastive Mixup with Refinement:** This method adds Update Policy and the pseudo-label refinement components to the Contrastive Mixup. If the classifier is available, the two-step refinement is used. The availability of classifier is shown in the first column of Table IV.

Table IV shows that the progressive training outperforms the original Contrastive Mixup [21] and supervised models. On Display Advertising Challenge and Drug Directory datasets, **Refinement** using the classifier in the architecture performs the best, while on Traffic Violations dataset the best performance obtained by the proposed progressive training without a classifier and **Refinement**.

The results on the Display Advertising Challenge dataset show that Contrastive Mixup does not always perform better than the supervised method, but progressive training outperforms the supervised method.

Finally, both Tables III and IV show that VIME and Contrastive Mixup can not consistently improve the accuracy compared to the supervised approach on all datasets, while the progressive training outperforms the supervised and non-progressive methods on all datasets.

3) **Ablation Study:**
We evaluate the Update Policy, Refinement and other components in Progressive Contrastive Mixup. The effect of each component (Classifier, Projection Network, Decoder, and Label-propagation) is studied to examine how they affect final performance. The change of Components are only made in the first training step, and the second step stays the same. Because the label-propagation is an important part of the Contrastive Mixup, it is used in all evaluations.

We introduced two terms in Progressive Contrastive Mixup subsection: with **Update** and with **Refinement**. When we do not use these terms, it means there is no regenerating the CPR representation in the architecture. A two-step refinement mechanism is used when both label-propagation and classifier components are available.

We want to study the robustness of the proposed framework under various combinations of components. Mainly three different cases are compared: 1. without updating CPR; 2. updating CPR with all pseudo-labels; 3. refining pseudo-labels for updating CPR. Table VI shows the results of this study in Traffic Violations, Display Advertising Challenge and Drug Directory datasets. Note that, **Projection+decoder** architecture is the architecture proposed in Contrastive Mixup.

Table VI shows that updating the conditional probability representation consistently outperforms the case when conditional probability representation is not updated across different models and datasets. It demonstrates the robustness of the proposed framework. In all three datasets, when refinement is added, Training with Refinement outperforms the Training without Update and Training with Update in majority cases. This shows that Training with Refinement uses more accurate pseudo-labels and improves the CPR representation. In Traffic Violations dataset, adding Training with update to the Contrastive Mixup performs best. In Display Advertising Challenge dataset, adding the proposed Classifier and Training with Refinement achieves the best result, which improves Contrastive Mixup by 5.28%. The proposed framework achieves the highest improvement in Display Advertising Challenge dataset because the cardinality of categorical features is really high. In Drug Directory dataset, the best performing method adds the proposed Classifier and Training with Refinement but removes the contrastive learning component. In both Display Advertising Challenge...
and Drug Directory datasets, the proposed classifier brings positive effective to the model architecture. In Traffic Violation and Drug Directory dataset, we observe that Training with Update underperforms Training without Update probability because the noise introduced by wrong pseudo-labels.

V. CONCLUSION

In this paper, we rethink semi-supervised learning from the feature representation perspective for high-cardinality categorical data. We show the potential of using the conditional probability representation in semi-supervised settings for categorical data. We propose using the CPR representation and upgrading it during training using labels and pseudo-labels from labeled samples and unlabeled samples respectively. Also, we proposed Refinement mechanism to reduce the effect of noise in the pseudo-labels that affect the representation of data. We demonstrate the effectiveness of the proposed framework in two recent algorithms and evaluate it on different datasets.

REFERENCES

[1] B. Liu, H. Li, H. Kang, N. Vasconcelos, and G. Hua, “Semi-supervised long-tailed recognition using alternate sampling,” arXiv preprint arXiv:2105.00133, 2021.
[2] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in International conference on machine learning. PMLR, 2020, pp. 1597–1607.
[3] H. Zhang, M. Cisso, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” CoRR, vol. abs/1710.09412, 2017. [Online]. Available: http://arxiv.org/abs/1710.09412
[4] J. Voon, Y. Zhang, J. Jordon, and M. van der Schaar, “Vime: Extending the success of self- and semi-supervised learning to tabular domain,” Advances in Neural Information Processing Systems, vol. 33, 2020.
[5] D. Micci-Barreca, “A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems,” ACM SIGKDD Explorations Newsletter, vol. 3, no. 1, pp. 27–32, 2001.
[6] J. E. Van Engelen and H. H. Hoos, “A survey on semi-supervised learning,” Machine Learning, vol. 109, no. 2, pp. 373–440, 2020.
[7] S. Laine and T. Aila, “Temporal ensembling for semi-supervised learning,” arXiv preprint arXiv:1610.02242, 2016.
[8] K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel, “Fixmatch: Simplifying semi-supervised learning with consistency and confidence,” arXiv preprint arXiv:2001.07685, 2020.
[9] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. Raffel, “Mixmatch: A holistic approach to semi-supervised learning,” arXiv preprint arXiv:1905.02249, 2019.
[10] T.-H. Cheung and D.-Y. Yeung, “Modals: Modality-agnostic automatic data augmentation in the latent space,” in International Conference on Learning Representations, 2020.
[11] W. Shi, Y. Gong, C. Ding, Z. M. Tao, and N. Zheng, “Transductive semi-supervised deep learning using min-max features,” in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 299–315.
[12] D. McClosky, E. Charniak, and M. Johnson, “Effective self-training for parsing,” in Proceedings of the Human Language Technology Conference of the NAACL, Main Conference, 2006, pp. 152–159.
[13] C. Haase-Schütz, R. Stal, H. Hertlein, and B. Sick, “Iterative label improvement: Robust training by confidence based filtering and dataset partitioning,” in 2020 25th International Conference on Pattern Recognition (ICPR). IEEE, 2021, pp. 9493–9490.
[14] M. N. Rizve, K. Duarte, Y. S. Rawat, and M. Shah, “In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning,” arXiv preprint arXiv:2103.06329, 2021.
[15] P. Cerda and G. Varoquaux, “Encoding high-cardinality string categorical variables,” IEEE Transactions on Knowledge and Data Engineering, 2020.
[16] P. Cerda, G. Varoquaux, and B. Kégl, “Similarity encoding for learning with dirty categorical variables,” Machine Learning, vol. 107, no. 8, pp. 1477–1494, 2018.
[17] A. Slaney, D. Salas, and Y. S. Rawat, “Encoding categorical variables with conjunctive bayesian models for wweed lead scoring engine,” arXiv preprint arXiv:1904.13001, 2019.
[18] J. Herzig, P. K. Nowak, T. Müller, F. Piccinno, and J. M. Eisenschlos, “Tapas: Weakly supervised table parsing via pre-training,” arXiv preprint arXiv:2004.02349, 2020.
[19] P. Yin, G. Neubig, W.-t. Yih, and S. Riedel, “Tabert: Pretraining for joint understanding of textual and tabular data,” arXiv preprint arXiv:2005.08314, 2020.
[20] X. Deng, H. Sun, A. Lees, Y. Wu, and C. Yu, “Turl: Table understanding through representation learning,” ACM SIGMOD Record, vol. 51, no. 1, pp. 33–40, 2022.
[21] S. Darabi, S. Fazeli, A. Pazoki, S. Sankararaman, and M. Sarrafa-zadeh, “Contrastive mixup: Self- and semi-supervised learning for tabular domain,” 2021.
[22] T. Ucar, E. Hajiramezanali, and L. Edwards, “Subtab: Subsetting features of tabular data for semi-supervised representation learning,” Advances in Neural Information Processing Systems, vol. 34, 2021.
[23] M. Verleysen and D. François, “The curse of dimensionality in data mining and time series prediction,” in International work-conference on artificial neural networks. Springer, 2005, pp. 758–770.
[24] A. Iscen, G. Tolias, Y. Avrithis, and O. Chum, “Label propagation for deep semi-supervised learning,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5070–5079.
[25] E. Arazo, D. Ortego, P. Albert, N. E. O’Connor, and K. McGuinness, “Pseudo-labeling and confirmation bias in deep semi-supervised learning,” in 2020 International Joint Conference on Neural Networks (IJCNN). IEEE, 2020, pp. 1–8.
[26] M. N. Rizve, K. Duarte, Y. S. Rawat, and M. Shah, “In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning,” arXiv preprint arXiv:2103.06329, 2021.
[27] B. Liu, H. Li, H. Kang, N. Vasconcelos, and G. Hua, “Semi-supervised long-tailed recognition using alternate sampling,” arXiv preprint arXiv:2105.00133, 2021.