Unsupervised Embeddings for Categorical Variables

Hannes De Meulemeester*, Bart De Moor† *Fellow, IEEE & SIAM*
ESAT, STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics, KU Leuven
Kasteelpark Arenberg 10, B-3001 Leuven, Belgium
Email: *hannes.demeulemeester@kuleuven.be, †bart.demoor@kuleuven.be

***Abstract*—Real-world data sets often contain both continuous and categorical variables yet most popular machine learning methods cannot by default handle both data types. This creates the need for researchers to transform their data into a continuous format. When no prior information is available, the most widely applied methods are simple ones such as one-hot encoding. However, they ignore many possible sources of information, in particular, categorical dependencies, which could enrich the vector representations. We investigate the effect of natural language processing techniques for learning continuous word-vector representations on categorical variables. We show empirically that the learned vector representations of the categorical variables capture information about the variables themselves and their dependencies with other variables similar to how word embeddings capture semantic and syntactic information. We also show that machine learning models using unsupervised categorical embeddings are competitive with supervised embeddings, and outperform them when fine-tuned, on various classification benchmark data sets.**

***Index Terms*—Machine Learning, Categorical Variables, Embedding Methods**

I. Introduction

Most machine learning models assume that their input values come from a continuous set (e.g. $\mathbb{R}^d$) or that, at least, a distance or similarity between inputs is defined. This is true for many applications but real-life data sets often contain both continuous and categorical variables. When presented with categorical values, it is often not obvious how to convert these to a continuous representation. Current categorical coding schemes may, very generally, be divided into nominal and ordinal.

Ordinal schemes, such as Helmert coding or polynomial coding [1], [2], assume an order and/or a structure in the variables. When such assumptions are not possible or sensible, the values may be treated as nominal and encoded using methods that replace the categorical values by certain calculated statistics, e.g. frequency encoding and target encoding [3]. Frequency encoding replaces the categorical value by its (normalized) frequency. Target encoding techniques replace the categorical values by certain statistics computed using the current variable value and their corresponding target variable values (e.g. mean encoding). The most popular technique, which does not calculate any statistics nor requires any assumptions, is one-hot encoding, also known as dummy encoding [4]. This technique replaces all values by equidistant indicator vectors. The disadvantage of this method is that the vector size increases with the cardinality of the variable. Furthermore, it does not capture any useful information about the variable itself or its interaction with other variables. As such, this encoding scheme completely leaves the interpretation and transformation of the variables and its values to the machine learning model.

In natural language processing (NLP), however, the standard is to encode the input, such as words, as continuous vector representations, called embeddings [5]. Recently, there has been some interest in learning embeddings [6], [7] for general categorical variables instead of using the standard encoding techniques. However, the focus has been on learning embeddings as a part of the supervised classification model, limiting their reusability. To our knowledge, few researchers investigated the general applicability of unsupervised NLP embeddings methods to categorical features.

This paper is organized as follows. In section II embeddings in natural language processing and the most popular methods to create them are introduced. In section III we discuss how these embedding methods may be applied to categorical variables. In section IV we show the ability of categorical embeddings to extract structure and dependencies and how they may improve classification performance compared to one-hot encoding and supervised neural embedding methods. We conclude the paper in section V and offer possible insights into future work in section VI.

II. Natural Language Processing

The data representation in natural language processing evolved from standard encoding techniques such as one-hot encoding and N-grams [8] to rich continuous vector space representations, called embeddings. Originally, neural embeddings were simply the continuous output of an intermediate neural network layer [9]. However, word embeddings only became widely accepted as the new standard after the introduction of the Skip-gram and Continuous Bag-of-words (CBOW) models [10]. These models allowed for efficient training due to their simple log-linear structure. Many extensions to these embeddings models have been made to allow them to work on i.a. character [11], sentence [12] or document [13] level.

This work was supported by KU Leuven: Research Fund (projects C16/15/059, C32/16/013, C24/18/022), Industrial Research Fund (Fellowship 13-0260) and several Leuven Research and Development bilateral industrial projects, Flemish Government Agencies: FWO (EOS Project no 30468160 (SeLMA), SBO project I013218N, PhD Grants (SB/1SA1319N, SB/1S93918, SB/151622)), EWI (PhD and postdoc grants Flanders AI Impulse Program), VLAIO (City of Things (COT.2018.018), PhD grants: Baekeland (HBC.20192204) and Innovation mandate (HBC.2019.2209), Industrial Projects (HBC.2018.0405)), European Commission (EU H2020-SC1-2016-2017 Grant Agreement No.727721: MIDAS)

The goal of embeddings is to capture useful semantic and syntactic information from the input data. One of the most noticeable properties about these embeddings is the ability to perform vector arithmetic in the embeddings space, that meaningfully translates back to the input space. The most famous example that illustrates this effect is that when one subtracts the vector for *Man* from *King* and adds instead the vector for *Woman*, one gets the vector for *Queen*: $vector(King) - vector(Man) + vector(Woman) \approx vector(Queen)$ [10]. It should be noted that recent work argued that this should be taken with a grain of salt [14], [15].

As an alternative to neural embeddings, we can use the matrix factorization methods. In general, matrix factorization or decomposition is the process of transforming a matrix into a multiplication of multiple, often structured, matrices with desired properties. In natural language processing, these methods do not work on the texts themselves, as the embedding methods, but on varying types of co-occurrence counts extracted from the corpus. Following the creation of such a co-occurrence matrix, the embeddings are created by applying low-rank matrix approximation methods, such as the Singular Value Decomposition (SVD). One of the most influential examples is Latent Semantic Analysis (LSA) [16], which works on the term-document matrix of a corpus. Another approach that instead uses a term-term co-occurrence matrix is the Hyperspace Analogue to Language (HAL) method [17]. However, a problem with count-based methods is that very frequent or infrequent words contribute a disproportionate amount. As such, recent work often focuses on how to properly transform the counts such that useful semantic information is extracted, while attempting to eliminate the side effects of these models [18]–[20].

A. CBOW and Skip-gram

The Continuous Bag-of-Words and Skip-gram models [10], commonly referred to as word2vec, are unsupervised methods for learning word vector representations. The CBOW model predicts a word based on the surrounding context words, while the skip-gram model predicts the context words based on the current word. Multiple improvements on these models have been made. It has been shown that for the Skip-gram model sub-sampling of frequent words and the use of negative sampling instead of hierarchical softmax allows for faster training and higher quality word vectors [21].

Words may also be split into multiple parts, N-grams, in order to capture subword information in N-gram embeddings. These embeddings may then be joined together in order to create the corresponding word embeddings [22], [23].

B. LSA

Latent Semantic Analysis (LSA) [16] produces word and documents vector representations using a term-document co-occurrence matrix. The rows of this matrix represent the terms and the columns represent the documents. Tf-idf [4] normalization is applied to the co-occurrence matrix to reduce the weight of uninformative high-frequency words. Finally, the dimensionality is reduced by performing the (truncated) Singular Value Decomposition (SVD). The resulting low rank approximation is used to obtain the word embeddings.

C. GloVe

One of the limitations of local context methods, such as CBOW and Skip-gram, is that they do not take global statistics of the data into account. GloVe [24] is a method for creating word embeddings that combines the local context methods and the global factorization methods. The algorithm first constructs the word-word co-occurrence matrix X, denoting X_{ij} as the number of times j occurs in context of word i. Using this matrix, the goal is to discover the relations between words, formulated as the ratio of their co-occurrence probabilities in some context. It can be shown that this goal can be structured as a weighted least squares regression problem,

$$\min_{w,b} \sum_{i,j=1}^{V} f\left(X_{ij}\right) \left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij}\right)^2. \quad (1)$$

With $w_i, \tilde{w}_i \in \mathbb{R}^d$ and $b_i, \tilde{b}_j$ respectively the (context) word vectors and biases for the words in vocabulary V. The weighting function f has as goal to limit the effect of frequent (and infrequent) co-occurrences and was selected to be,

$$f(x) = \begin{cases} (x/x_{\max})^{\alpha} & \text{if } x < x_{\max} \\ 1 & \text{otherwise.} \end{cases} \quad (2)$$

This function has two additional, positive-valued, parameters: $x_{\max}$ and α. The $x_{\max}$ determines at which point the function returns 1 and the α value is the power used to, possibly non-linearly, scale the fraction. It was found empirically that setting the values for $x_{\max}$ and α to be respectively 100 and $3/4$ offered the best performance for natural language processing tasks [24].

III. Methodology

Although categorical variables are very closely related to words, two important differences must be considered. The first is that the interpretation of a combination of categorical variables does not depend on the location of those variables. Put differently, the columns of the input data matrix may be permuted without loss of meaning. This is not the case in natural language, where the order of characters, words and sentences are a large part of what determines the meaning of the text.

The second difference lies in the relations between the values of the categorical variables. It is perfectly valid for a value, or even a complete category, to be independent or unrelated to the values of other variables. This is not the case in a meaningful sentence or text. No matter how unimportant a word, it is still needed to adhere to the syntax or to convey the correct meaning.

To translate these changes to the existing word embeddings methods, we need to make some adjustments. The first is the context window length. Since sentences have different

lengths and the assumption may be made that words closer together are more strongly related than those further away, it makes sense to limit the size of the context window to some reasonable number. In the case of categorical variables, the window should initially span all variables since the length is always the same and the meaning of these variables is location independent.

In the case of GloVe, there are two parameters that require extra attention: the weighing of occurrences and the α value (cf. Eqn. 2). By default, the values in the co-occurrence matrix are weighted by the distance to the focus word. Based on this distance, the co-occurrence count of a context word is update by $1/d$, with d the distance. In this way, occurrences that are further away are assumed to be less important than those closer. For categorical variables, this count should always be updated by one, independent of the distance.

The α parameter determines a transformation of the a co-occurrence counts, with a value 1 meaning the identity. Empirically, a value 0.75 was shown to perform well on natural language. In our experiments, we found that for certain data sets values $\alpha \geq 1$ performed better, while for other data sets, values even lower than 0.75 were needed to achieve the best performance. This indicates the need to treat the α as a tuning parameter.

In the next section, we show how these changes to the embeddings methods allow them to be applied to categorical variables. We apply them to an artificially created data set to clearly show their respective strengths and then show that we can make the same observations on a more complicated data set. Finally, we show empirically that these embeddings offer an increase in classification performance compared to when using one-hot encoding and are competitive with supervised neural embedding methods.

IV. Experiments

A. Simple Example

In this section we use a handcrafted example data set to illustrate how the embedding methods may be used to extract dependencies and structure out of categorical variables. The data set contains three categorical variables: Activity, Animal and Disposition. By default, all values of a variable are equally likely to appear. In addition to this, three relations, in the form of conditional probabilities have been created to simulate the dependency between variables. For example, given that the value *Barks* appears in the Activity column, the second column will contain *Dog* with a probability of 0.8.

On figure 1 the word vectors are depicted using t-Distributed Stochastic Neighbor Embedding (t-SNE) [25] as a dimensionality reduction and visualisation technique. Perplexity values of 5 to 10 often provided the clearest figures. The choice was made to use t-SNE for all embedding visualisations because this technique generates clearer and more interpretable figures than alternatives such as principal component analysis (PCA). However, it is important to note that, for all experiments, the exact same observations can be made on figures where PCA was used as the dimensionality reduction technique. To show this, the PCA visualisations of the two best models are additionally provided on figure 2. All three neural embedding methods were set up to generate embeddings of size 50. Additionally, for GloVe, the number of epochs and α were set to be 50 and 1.5, respectively. For LSA, the truncated SVD was performed using 2 components.

Activity	**Animal**	**Disposition**
Barks	Dog	Playful
Roars	Cat	Lazy
Talks	Bird	Aggressive
Sings	Penguin	Calm
Dances	Mouse	

Relation	**Probability**
P(Dog | Barks)	0.8
P(Playful | Dog)	0.85
P(Lazy | Cat)	0.8

TABLE I
LEFT: THE CATEGORICAL VARIABLES WITH THEIR CORRESPONDING VALUES OF THE ARTIFICIAL DATA SET. RIGHT: THE CORRESPONDING MANUALLY CREATED RELATIONS PRESENT IN THE DATA SET.

This simple example was created in order to show the relative strengths and weaknesses of the studied methods. As can be seen on figure 1, both CBOW and Skip-gram appear to perform well in discovering which values belong to which variables, visualised by the distinct clusters. The methods are less effective at discovering the relations between the values themselves: they often fail to detect more difficult relations such as the one with Dog, Barks and Playful. Skip-gram performs slightly better in that aspect.

Making the link with natural language processing, one could say that these methods are better at distinguishing syntax over semantic relations.

LSA is able to extract the relations between the values of different categorical variables, but is often worse at differentiating between the categorical variables themselves. This is indicated by the less clearly defined clusters of variables.

True to its original goal, GloVe performs equally well at extracting relations and recognizing different categorical variables while respecting the distances between the embedded variable clusters.

In general GloVe performs the best at extracting the structure and relations out of this simple data set. Although, it should be said that all techniques are able to represent more information than would be the case when using popular transformations such as one-hot encoding.

B. Nursery Data

The UCI nursery data[1] [26] is a data set containing only categorical variables. It is extracted from a hierarchical decision model originally developed to rank applications for nursery schools [27]. The structure derived from a decision model makes it a suitable candidate to illustrate the use and advantages of categorical embeddings on a more complicated data set. The data consists of 12 960 samples, with eight categorical variables that represent the hierarchical structure (see table II) and a categorical variable that represents the final decision of the model.

To compare with what a standard statistical method is able to

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

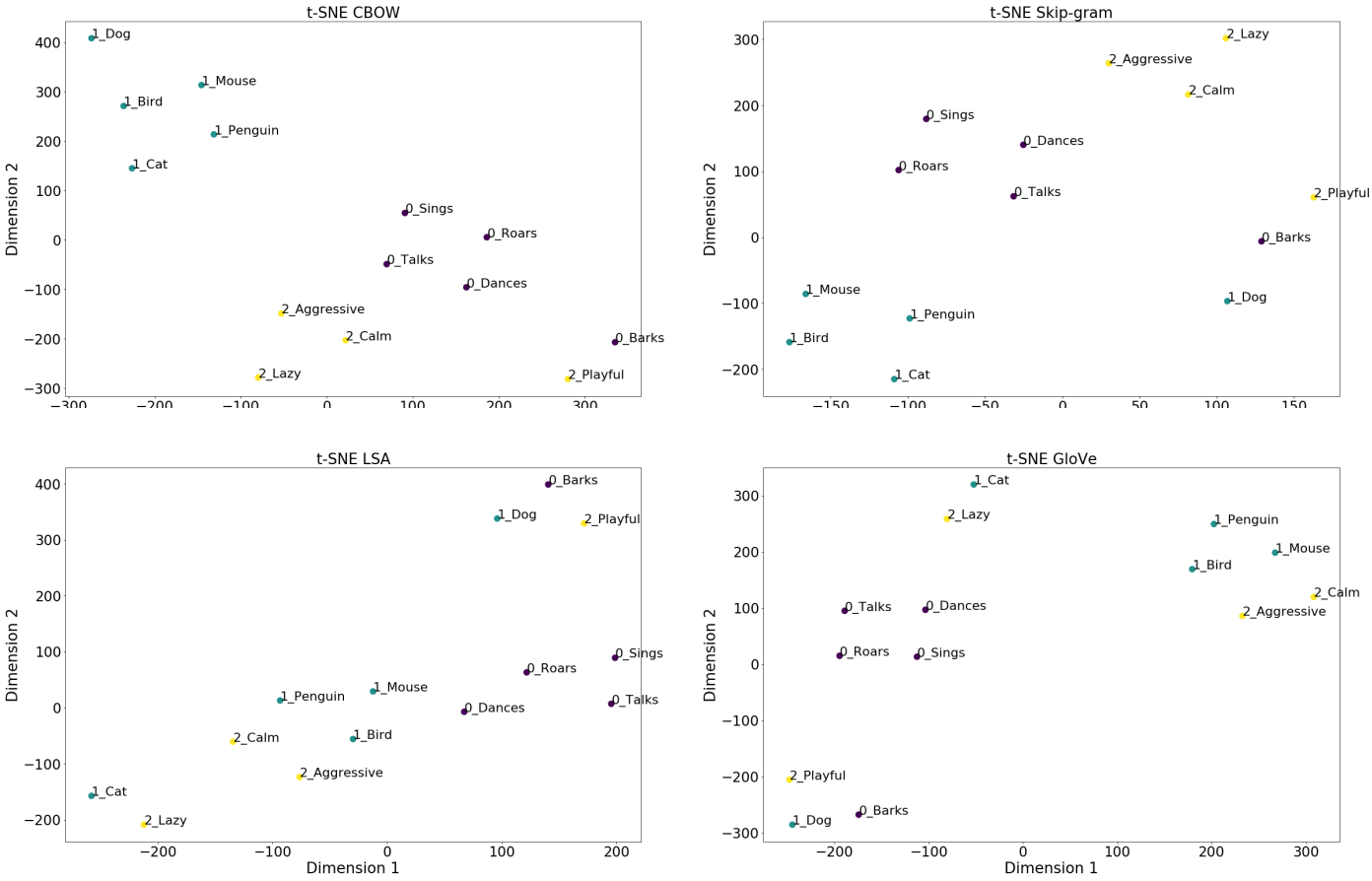

Fig. 1. The t-SNE visualisations of respectively CBOW, Skip-gram, LSA and GloVe for the simple example. CBOW and Skip-gram perform well at representing the categorical variables, shown by the clear variable clusters. LSA is able to extract all dependencies, as indicated by the closeness of the related values in embedding space, but has less distinct clusters. GloVe performs well at both tasks.

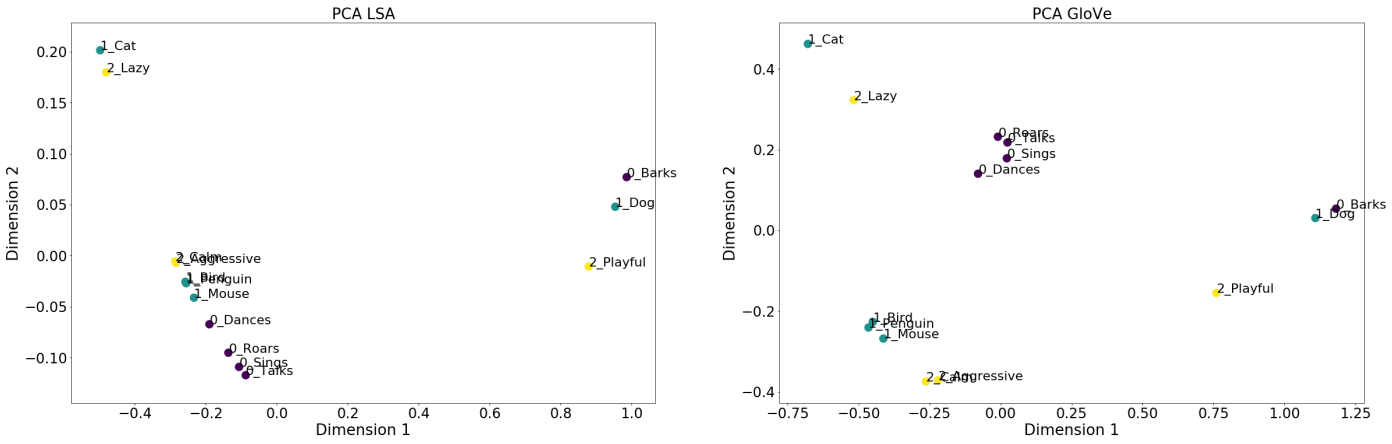

Fig. 2. The PCA visualisations of the two best performing models, LSA and GloVe, for the simple example. The visualisations show the same relations as those that can be observed on the t-SNE plots: Both LSA and GloVE are able to extract the relations but GloVe is better at constructing clusters of the categorical values.

extract, Cramer's V score [28], [29] for all attribute pairs was calculated. It is a measure of dependency between nominal variables with 0 indicating no dependency and 1 indicating complete dependency. The score was 0 for all variable pairs except for the pairs including the target variable, which are presented in table II. These scores show that a strong relation exists between the 7th attribute (the health conditions) and the target variable. However, the test does not detect any other relations, such as the hierarchical structure.

The results of the embeddings methods can be found in figure 3. As in the simple example, CBOW and Skip-gram perform exceedingly well at creating representations of the categorical variables themselves. To a lesser degree, the strong relation between the health conditions and the target variable (variables 7 and 8, respectively) can also be observed.

LSA performs well at picking up the relation between variables 7 and 8 but has trouble identifying the structure present in the data. GloVe again shows the best results; in fact the method is able to almost fully capture the hierarchical structure of the decision model. The categorical variables which are aggregated together in the tree structure are located close in the embedding space. For example, variables 0-1 and 2-3 are aggregated into "Employment of parents and child's nursery" and "Family structure" respectively and are clustered together in the embedding space. The relationship extracted by Cramer's V, between variable 7 and 8, is also captured, demonstrated by the closeness of their corresponding embeddings.

As in the simple example described in the previous section, GloVe is able to extract the most information out of the data set. Additionally, the experiments show that even on a more difficult, real, data set, all methods are able to capture the structure and dependencies, albeit to varying degrees.

C. Classification Performance

We compare the performance of the embeddings methods with the one-hot encoding baseline on the Breast cancer[2], CMC[3], Adult[4] and Credit[5] data sets from the UCI repository. The first two data sets do not contain any continuous features and are completely transformed using one-hot encoding or one of the embedding methods. For the last two data sets, the categorical variables are embedded, and the continuous features are left as is. For the CMC data set, the two minority classes are treated as a single class to create a binary classification task.

Classification models. All experiments were performed using the same two simple models: a logistic regression and neural network model, implemented using respectively scikit-learn [30] and Tensorflow [31]. The neural network consists of a single hidden layer with 512 neurons, a RELu activation function and a dropout layer with a rate of 0.1 as regularization. The neural embedding architecture has a separate input, followed by an embedding layer, for each categorical variable of which the outputs are then concatenated. These concatenated embeddings are then used as input for one or more hidden layers. The experiments have been performed with one hidden layer and a second hidden layer as a reference to reflect the original architecture [6]. No architecture tuning has been performed as the goal is to compare the embeddings methods as unbiased as possible. For the logistic regression model, the SAGA optimizer [32] was used while the neural network model was trained using the Adam optimizer [33]. For the embeddings methods, the embeddings size and the frequency scaling parameter, the α, were manually tuned. All models have been trained for at least 10 times. The average and best performance of the models, measured by the area under the ROC curve (AUC), is given in table III. For the Adult data set, the AUC was calculated on the provided test data set and for the other three the 5-fold cross-validation score is reported.

Only categorical variables. For the Breast cancer and CMC data sets, all embeddings methods significantly outperform one-hot encoding. On the breast cancer data, Skip-gram and CBOW performed the best on average, with Skip-gram and GloVe obtaining the best performance for the logistic regression and the neural network model respectively. The best results were observed using an embedding size of 50 to 100 and an $\alpha <= 0.75$. For LSA, the number of components was set to be 3 to 5.

On the CMC data set, GloVe is the best performing embedding method consistently for both models. The optimal embedding size and α values were found to be respectively 20 to 50 and 0.25. For LSA, we achieved the best results using 5 components.

Mixed variables. For the two data sets containing both continuous and categorical variables, Adult and Credit, categorical embeddings again lead to the best results. However, the performance gain by the embeddings methods was less pronounced. On the Adult data, LSA performed best in combination with the logistic regression model, while GloVe outperformed the other methods when used in the neural network model. For both models Skip-gram and CBOW benefited from an embedding size of 20 and GloVe performed better with an embedding size of 100 and an α value of 0.75 or lower. For both models, 7 components were selected for the LSA embeddings.

On the credit data set, CBOW and and Skip-gram perform slightly worse on average than one-hot encoding. Nevertheless, the higher variability in their performance allows their best achieved results to outperform the baseline. The best performing embedding methods are GloVe and LSA for the logistic regression and neural network models respectively. These results were obtained with embedding sizes of 25 for the Skip-gram and CBOW models. GloVe performed better with an α value of 0.4 and an embedding size of 50-100. For the LSA method, 6 components were selected.

In general, the best performing model is consistently one with categorical embeddings as input, indicating that the discussed embeddings methods are able to extract useful information, such as structure and dependencies, that are

[2]https://archive.ics.uci.edu/ml/datasets/breast+cancer

[3]https://archive.ics.uci.edu/ml/datasets/Contraceptive+Method+Choice

[4]https://archive.ics.uci.edu/ml/datasets/Adult

[5]https://archive.ics.uci.edu/ml/datasets/credit+approval

```
8: Final evaluation of applications for nursery schools
  . Employment of parents and child's nursery
  . . 0: Parents' occupation
  . . 1: Child's nursery
  . Family structure and financial standings
  . . Family structure
  . . . 2: Form of the family
  . . . 3: Number of children
  . . 4: Housing conditions
  . . 5: Financial standing of the family
  . Social and health picture of the family
  . . 6: Social conditions
  . . 7: Health conditions
```

Attribute	Cramer's V
0	0.214
1	0.246
2	0.047
3	0.072
4	0.112
5	0.075
6	0.106
7	0.731

TABLE II

LEFT: HIERARCHICAL STRUCTURE OF THE VARIABLES OF THE NURSERY DATA. RIGHT: THE CRAMER'S V SCORE, A MEASURE FOR ASSOCIATION BETWEEN CATEGORICAL VARIABLES, FOR THE CORRESPONDING VARIABLES WITH REGARDS TO THE FINAL EVALUATION.

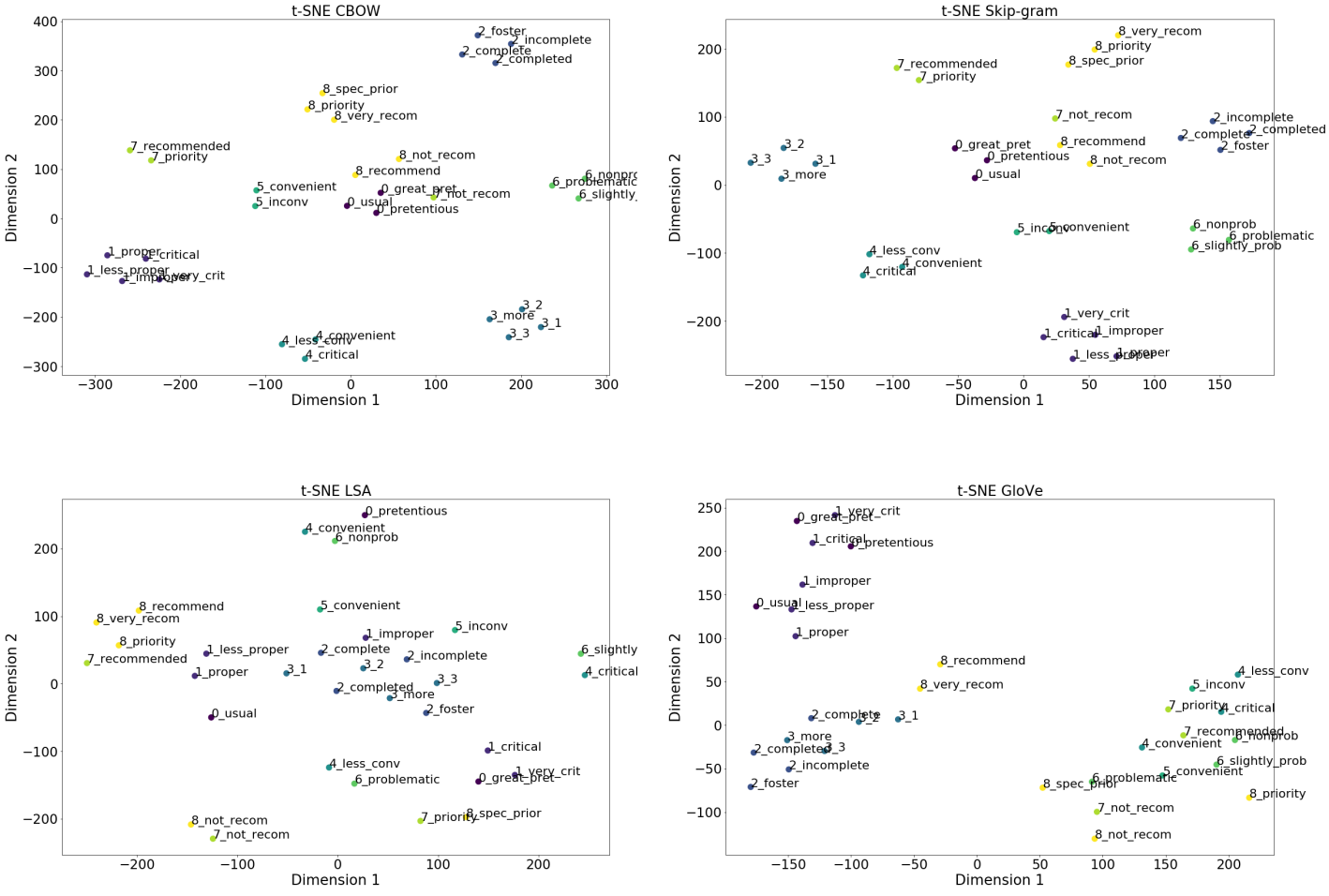

Fig. 3. The t-SNE visualisations of respectively CBOW, Skip-gram, LSA and GloVe for the nursery data set. CBOW and Skip-gram do well at representing the categorical variables, indicated by their corresponding distinct clusters. LSA is able to extract more relations than CBOW and Skip-gram: some attributes of the same hierarchical level are closer together in the embeddings space. GloVe performs well at both.

Method	Breast cancer				CMC			
	LR		NN		LR		NN	
	Avg.	Best	Avg.	Best	Avg.	Best	Avg.	Best
One-hot	62.03 ± 1.61	64.42	70.38 ± 1.36	71.75	73.76 ± 1.41	76.12	82.80 ± 0.77	83.97
CBOW	66.86 ± 0.87	68.08	**71.96** ± 0.99	73.51	74.15 ± 1.72	77.47	85.16 ± 0.66	86.57
Skip-gram	**67.19** ± 1.57	**68.75**	71.76 ± 1.46	73.04	76.20 ± 1.12	78.33	84.93 ± 0.53	85.91
LSA	65.30 ± 1.12	67.77	71.70± 0.94	72.91	77.92 ± 1.08	80.32	84.33 ± 0.74	85.60
GloVe	**67.19** ± 1.14	68.30	71.50 ± 1.26	**73.78**	**79.10** ± 0.91	**80.58**	**86.05** ± 0.36	**86.68**
	Adult				Credit			
	LR		NN		LR		NN	
	Avg.	Best	Avg.	Best	Avg.	Best	Avg.	Best
One-hot	81.88 ± 0.0	81.88	90.30 ± 0.28	90.60	86.85 ± 0.35	87.67	92.50 ± 0.17	92.71
CBOW	81.98 ± 0.0	81.98	90.49 ± 0.23	90.73	86.71 ± 0.41	87.68	92.37 ± 0.31	93.06
Skip-gram	81.98 ± 0.0	81.98	90.45 ± 0.22	90.79	86.72 ± 0.40	87.98	92.32 ± 0.35	92.81
LSA	**82.03** ± 0.0	**82.03**	90.41 ± 0.45	90.81	86.92 ± 0.35	87.72	**93.05** ± 0.20	**93.29**
GloVe	81.92 ± 0.0	81.92	**90.58** ± 0.37	**90.96**	**87.22** ± 0.36	**88.22**	92.66 ± 0.24	93.01

TABLE III

OVERVIEW OF THE CLASSIFICATION PERFORMANCE OF THE BASELINE ONE-HOT ENCODING AND THE CATEGORICAL EMBEDDING METHODS ON THE FOUR UCI DATA SETS. THE REPORTED VALUES ARE THE AVERAGE AUC WITH STANDARD DEVIATION AND THE BEST OBSERVED AUC FOR A LOGISTIC REGRESSION MODEL (LR) AND A NEURAL NETWORK WITH ONE HIDDEN LAYER (NN). THE BEST PERFORMING MODEL IS CONSISTENTLY ONE WITH AS INPUT CATEGORICAL EMBEDDINGS.

Architecture	Breast cancer		CMC		Adult		Credit	
	Avg.	Best	Avg.	Best	Avg.	Best	Avg.	Best
One Hidden Layer	68.78 ± 1.52	71.78	83.25 ± 0.80	84.68	90.69 ± 0.13	90.88	92.25 ± 0.28	92.70
Two Hidden Layers	68.75 ± 1.34	71.52	83.25 ± 0.85	85.07	90.74 ± 0.20	90.99	92.50 ± 0.26	92.89
One Hidden Layer (Pretrained)	71.56 ± 1.33	74.19	**86.38** ± 0.37	87.22	91.08 ± 0.03	91.12	93.14 ± 0.19	93.43
Two Hidden Layers (Pretrained)	**72.30** ± 1.50	**74.81**	86.19 ± 0.72	**87.64**	**91.11** ± 0.06	**91.18**	**93.42** ± 0.26	**93.87**

TABLE IV

OVERVIEW OF THE CLASSIFICATION PERFORMANCE OF THE NEURAL EMBEDDINGS WITH AND WITHOUT UNSUPERVISED PRETRAINING. THE REPORTED VALUES ARE THE AVERAGE AUC WITH STANDARD DEVIATION AND THE BEST OBSERVED AUC. FOR THE BREAST CANCER, CMC AND CREDIT DATA SET THE UNSUPERVISED METHODS OUTPERFORM THE NEURAL EMBEDDINGS WITHOUT PRETRAINING. IN ALL CASES, PRETRAINED NEURAL EMBEDDINGS WITH FINE-TUNING OUTPERFORM THE STANDARD SUPERVISED NEURAL EMBEDDINGS AND ACHIEVE THE BEST RESULT ON ALL DATA SETS.

helpful for the classification process. Among the embeddings methods, there is no technique that consistently outperforms the others. However, for the majority of our experiments GloVe, more frequently, offered the largest classification performance increase. This, in combination with the visualisation experiments, where it was shown that GloVe was the best at i.a. extracting relations, makes it the preferred initial choice to create unsupervised categorical embeddings.

Supervised neural embeddings. The performance of supervised neural embeddings and unsupervised embeddings with fine-tuning is reported in table IV. The pretraining is performed using the best reported method in table III, which is GloVe for all data sets except for the Credit data set where LSA performed better. For the standard supervised embeddings (without pretraining), small embedding sizes of 5-20 performed better than large sizes. The supervised neural embeddings also perform better than one-hot encoding. However, even without fine-tuning, the unsupervised embeddings outperformed the supervised embeddings on all data sets except on the Adult data. Finally, the unsupervised embeddings with fine-tuning achieve the best results on all data sets.

We explain the competitive, and often better, performance of the unsupervised methods by their ability to take global statistics (e.g. LSA, GloVe) into account. In addition, it is known that unsupervised pretraining, as in the case of the NLP embeddings, is beneficial to the performance and generalization capabilities of neural network models [34]. The experiments show that this is also the case for categorical variables.

V. CONCLUSION

In this work we studied the applicability of unsupervised NLP embedding methods to categorical variables. We observed that the learned continuous vectors do not only represent the categorical variables efficiently, but also capture dependencies and structure in an unsupervised way. Drawing the link with natural language processing, the ability for word embeddings to capture semantic and syntactic relations translates well to categorical variables.

Additionally, we have empirically shown that pretrained categorical embeddings outperform one-hot encoding on various popular classification benchmark data sets. Out of the discussed embedding methods, GloVe offered the most consistent improvements.

When comparing the unsupervised methods to the supervised neural embeddings, it was shown that they are at least competitive with each other. But more often than not, the unsupervised

embeddings outperformed the supervised neural embeddings, especially when the data set consisted only of categorical variables. Finally, we showed that fine-tuned unsupervised embeddings consistently outperformed any other embedding method.

VI. Future Work

Categorical embeddings provide an attractive alternative to commonly used methods such as one-hot encoding. However, for the NLP techniques to be applicable to categorical variables, some changes to the treatment of the context window and distance weighting needed to be made. Additionally, it was often necessary to tune the frequency weighting parameter α in order to obtain the best results. This offers future research directions into specialized categorical methods and weighting schemes. Additionally, real life data often contains both categorical and continuous data while the current unsupervised embedding methods only take the categorical part into account. Techniques may be developed that allow the training of the categorical embeddings to take these continuous values into account as well.

References

[1] M. J. Davis, "Contrast coding in multiple regression analysis: Strengths, weaknesses, and utility of popular coding structures," *Journal of Data Science*, vol. 8, no. 1, pp. 61–73, Jan 2010.

[2] C. A. W. C. A. Wendorf, "Primer on multiple regression coding: Common forms and the additional case of repeated contrasts," *Understanding Statistics*, vol. 3, no. 1, pp. 47–57, 2004.

[3] D. Micci-Barreca, "A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems," *SIGKDD Explorations*, vol. 3, no. 1, pp. 27–32, 2001.

[4] T. G. Kolda, "Limited-memory matrix methods with applications," Ph.D. dissertation, University of Maryland, 1997.

[5] F. Almeida and G. Xexéo, "Word embeddings: A survey," *CoRR*, vol. abs/1901.09069, 2019. [Online]. Available: http://arxiv.org/abs/1901.09069

[6] C. Guo and F. Berkhahn, "Entity embeddings of categorical variables," *CoRR*, vol. abs/1604.06737, 2016. [Online]. Available: http://arxiv.org/abs/1604.06737

[7] Y. Russac, O. Caelen, and L. He-Guelton, "Embeddings of categorical variables for sequential data in fraud context," in *The International Conference on Advanced Machine Learning Technologies and Applications, AMLTA 2018, Cairo, Egypt, February 22-24, 2018*, ser. Advances in Intelligent Systems and Computing, vol. 723. Springer, 2018, pp. 542–552.

[8] D. Jurafsky and J. H. Martin, *Speech and Language Processing (2Nd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2009.

[9] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.

[10] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013. [Online]. Available: http://arxiv.org/abs/1301.3781

[11] Y. Kim, Y. Jernite, D. A. Sontag, and A. M. Rush, "Character-aware neural language models," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, 2016, pp. 2741–2749.

[12] Z. Lin, M. Feng, C. N. dos Santos, M. Yu, B. Xiang *et al.*, "A structured self-attentive sentence embedding," *CoRR*, vol. abs/1703.03130, 2017. [Online]. Available: http://arxiv.org/abs/1703.03130

[13] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, 2014, pp. 1188–1196.

[14] T. Bolukbasi, K. Chang, J. Y. Zou, V. Saligrama, and A. T. Kalai, "Man is to computer programmer as woman is to homemaker? debiasing word embeddings," in *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, 2016, pp. 4349–4357.

[15] M. Nissim, R. van Noord, and R. van der Goot, "Fair is better than sensational: Man is to doctor as woman is to doctor," *CoRR*, vol. abs/1905.09866, 2019. [Online]. Available: http://arxiv.org/abs/1905.09866

[16] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman, "Indexing by latent semantic analysis," *JASIS*, vol. 41, no. 6, pp. 391–407, 1990.

[17] K. Lund and C. Burgess, "Producing high-dimensional semantic spaces from lexical co-occurrence," *Behavior Research Methods, Instruments, & Computers*, vol. 28, no. 2, pp. 203–208, Jun 1996.

[18] D. L. T. Rohde, L. M. Gonnerman, and D. C. Plaut, "An improved model of semantic similarity based on lexical co-occurence," *COMMUNICATIONS OF THE ACM*, vol. 8, pp. 627–633, 2006.

[19] J. A. Bullinaria and J. P. Levy, "Extracting semantic representations from word co-occurrence statistics: A computational study," *Behavior Research Methods*, vol. 39, no. 3, pp. 510–526, Aug 2007.

[20] R. Lebret and R. Collobert, "Word embeddings through hellinger PCA," in *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014, April 26-30, 2014, Gothenburg, Sweden*. The Association for Computer Linguistics, 2014, pp. 482–490.

[21] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, 2013, pp. 3111–3119.

[22] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *TACL*, vol. 5, pp. 135–146, 2017.

[23] E. Grave, T. Mikolov, A. Joulin, and P. Bojanowski, "Bag of tricks for efficient text classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*. Association for Computational Linguistics, 2017, pp. 427–431.

[24] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. ACL, 2014, pp. 1532–1543.

[25] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[26] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[27] M. Olave, V. Rajkovic, and M. Bohanec, "An application for admission in public school systems," in *Expert Systems in Public Administration,*, 1989, pp. 145–160.

[28] H. Cramer, *Mathematical methods of statistics*. Princeton University Press Princeton, 1946.

[29] W. Bergsma, "A bias-correction for cramér's v and tschuprow's t," *Journal of the Korean Statistical Society*, vol. 42, no. 3, pp. 323 – 328, 2013.

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[31] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. [Online]. Available: https://www.tensorflow.org/

[32] A. Defazio, F. R. Bach, and S. Lacoste-Julien, "SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives," in *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, 2014, pp. 1646–1654.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[34] D. Erhan, Y. Bengio, A. C. Courville, P. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?" *Journal of Machine Learning Research*, vol. 11, pp. 625–660, 2010.