Encoding Categorical Variables with Ambiguity

Koichi Takayama

Recruit Co., Ltd.,
Tokyo, Japan
k.takayama0902@gmail.com (main), ktakayama@r.recruit.co.jp (sub)

Abstract. Most of the supervised learning methods assume that the independent variables are provided without ambiguity. At the preprocessing stage of the learning methods, categorical variables are often transformed into numerical vectors by a mapping function from each category to a real number. In a real-world, however, there are several natural scenarios where categorical variables are collected with ambiguity, such as *the value of X is a or b*. In this paper, we show that the problem of encoding ambiguous categorical variables can be handled as missing value imputation problems. We extend existing One-Hot encoding methods to handle ambiguous categorical variables explicitly. We further propose two encoding methods based on missing value imputation algorithms, *Ambiguous Forests*. One is a naive extension of the MissForest algorithm, and the other is a novel application of MissForest and learning from partial labels to encoding methods. We evaluate the impact of encoding methods by masking two real-world datasets to contain categorical independent variables with ambiguity. We show that our encoding methods substantially outperforms existing methods when the ambiguity between the training set and test set is qualitatively different.

1 Introduction

Ordinary supervised learning algorithms require independent variables to be represented as a numerical feature. Numerical variables are often preprocessed by standardization, and categorical variables need to be transformed into numerical values by some encoding methods, such as One-Hot encoding [3] and Ordinal encoding [6]. Those classical encoding methods assume that the independent variables do not contain ambiguous information. Namely, the similarity between an instance that represents *the value of X is a or b* and another instance that represents *the value of X is a* is not explicitly considered. Therefore, classical encoding methods do not capture the semantic structure of such ambiguous categorical variables.

In real-world datasets, however, there are several scenarios where encoding methods that focus on the semantic structure of ambiguous categorical variables would be useful. For example, the definition of a database could be different depending on the types of data source or on the time when each instance is generated. The difference of the definition would result in generating ambiguous categorical variables in an undesirable way, where a feature of an instance could

be represented as *their major is math* and that of another instance could be represented as *their major is math, physics, or computer science*. Another example is the case where human beings annotate categorical variables. As with the case of annotating class labels, it would be easy and correct to annotate categorical variables with ambiguity. Such a situation often occurs in the field of psychology, where it is reasonable to annotate by ambiguous categorical variables.

We introduce four viewpoints to tackle the problems of the semantic structure of ambiguous categorical variables. We compare existing encoding methods and our methods in table 1 from the viewpoint. Ambiguity consideration is the main topic in this paper. As mentioned above, classical encoding methods do not consider the ambiguity. There is no method that considers the ambiguity explicitly except for our methods. The feature dimensionality (Maximum dimensionality) of categorical variables after encoding is crucial for the performance of final predictive model. One-Hot encoding would transform an ambiguous categorical variable with cardinality k into 2^k-dimensional vector, which would lead to the poor performance of the final predictive models. The pretraining cost cannot also be ignored to use encoders in real-world applications. Heterogeneous ambiguity denotes whether encoding methods refer to other independent variables to represent the heterogeneity of each instance. From our insight of encoding problems, encoders that satisfy the following properties would give useful feature representations: *(i) explicit consideration of ambiguity (ii) low dimensionality of a feature vector after encoding (iii) low pretraining cost (iv) consideration of heterogeneous ambiguity* . This paper is the first work that proposes encoding methods satisfying all of the desirable properties.

In this paper, we regard the encoding process as an extension of missing value imputation problems. We propose two types of encoding methods using the analogy of missing value imputations. One is based on One-Hot encoding, and the other is based on MissForest [14]. Experiments show that our methods outperform existing encoding methods. In particular, when the ambiguity between the training set and test set is different, our methods improve F1 (micro averaged) score by 0.05.

Our contributions are summarized as follows:

- We propose new encoding methods to tackle the difficulty of the ambiguity of categorical variables.
- Our methods named *Ambiguous Forests*, based on MissForest, contain a novel application of learning from partial labels to encoding methods.
- We empirically evaluate various types of encoding methods using real-world datasets and show the validity of our encoding methods.
- We further implement our methods as a publicly available module of open source software, named *Multi_hot encoder* in categorical encoders[1].

[1] http://contrib.scikit-learn.org/categorical-encoding/

Table 1: Overview of encoding methods (k: cardinality, d: embedding dimensionality)

Methods	Consideration of ambiguity	Maximum dimensionality	Pretraining	Heterogeneous ambiguity
One-Hot encoding [3]	No	2^k	No	Ignored
Ordinal encoding [6]	No	1	No	Ignored
Similarlity encoding [2]	Implicitly Yes	k	Light	Ignored
Entity embedding [7]	Implicitly Yes	d	Heavy	Ignored
This work (section 3.1)	Explicitly Yes	k	Light	Ignored
This work (section 3.2)	Explicitly Yes	k	Light	Considered

2 Problem settings and related work

In this section, we introduce fundamental notations of encoding categorical variables and review related work.

2.1 Notation

Let X^j be an independent categorical random variable whose domain is $\{C^{j,l}\}_{l=1}^{k_j}$, where $j \in \{1, 2, \cdots, D^C\}$ is an index of the variable and $k_j \geq 2$ is the cardinality of it. Here, $D^C \geq 2$ is the number of categorical variables.

We often encode the vector of categorical variable, $X^C = \{X^j\}_{j=1}^{D^C}$, into numerical values. The numerical values are concatenated with the vector of other independent variables of real-numbers, $X^N \in \mathbb{R}^{D^N}$, where D^N is the number of numerical variables. In supervised learning, algorithms learn functions that input $X = [X^N, X^C]$ and predict its corresponding output value, $Y \in \mathcal{Y}$.

In this work, we focus on the categorical variables with ambiguity. We define the ambiguity of categorical variables here.

Definition 1. *A categorical variable is ambiguous if the value of the variable could not be specified in its observation. Let the observed candidate set of X^j be O^j. We define an ambiguous indicator variable of j-th variable for each $l \in \{1, 2, \cdots, k_j\}$ as follows:*

$$AI^{j,l} = \mathbb{1}_{l \in O^j}, \tag{1}$$

where $\mathbb{1}$ is an indicator function.

Our definition is related to that of the epistemic view [8]. Although there exist various types of definitions of ambiguity in the context of fuzzy observation, we do not explain them in this paper because it is out of scope.

2.2 Encoding methods

One-Hot encoding One-Hot encoding transforms each category into a vector of binary variables [3]. While the categorical variable of k-cardinality is often

transformed into the $k-1$-dimensional vector in statistical inference, it is natural to adopt an encoder that outputs the k-dimensional vector to consider new categories that appear only in the test set. We formulate the encoder of j-th categorical variable as the following function:

$$enc_{OH}^{j} \colon \{C^{j,l}\}_{l=1}^{k_j} \to \Big\{ z \mid z \in \{0,1\}^{k_j}, \sum_{l=1}^{k_j} z_j \leq 1 \Big\}, \tag{2}$$

where $enc_{OH}^{j}(X^j) = [\mathbb{1}_{X^j=C^{j,1}}, \mathbb{1}_{X^j=C^{j,2}}, \cdots, \mathbb{1}_{X^j=C^{j,k_j}}]$.

Although One-Hot encoding is one of the most common methods to encode categorical variables, there are several drawbacks in the method. Cerda et al. [2] empirically showed that One-Hot encoding is inappropriate to transform high-cardinality categorical variables.

The curse of dimensionality due to high-cardinality can also appear in ambiguous categorical variables. When a categorical variable of cardinality k is observed with ambiguity, the observed cardinality of the variable would rise exponentially to 2^k. Furthermore, this method cannot reflect the latent semantic structure of ambiguous variables since each observed category is considered a completely different category. These drawbacks would lead to the poor performance of the final predictive model in ambiguous categorical data.

Ordinal encoding Ordinal encoding assigns an integer to each category [6]. This method is known to work well with tree-based predictive models [12]. Unlike One-Hot encoding, the dimensionality does not increase after encoding in Ordinal encoding. However, the problem of considering the latent semantic structure of ambiguous variables remains unresolved.

Other encoding methods We review other encoding methods for categorical variables. None of them consider the latent semantic structure of ambiguous variables and the heterogeneous ambiguity explicitly. Several methods could be enhanced by introducing our methods as preprocessing.

There exist several encoding methods that assign statistical information of each category to the representation of the category. Count encoding assigns the number of observations of each category and target encoding assigns the sample mean of the target variable Y conditioned on each category [10]. These methods can represent the characteristics of rare categories combined with smoothing methods.

Hash encoding uses a feature hashing trick [16] to reduce the dimensionality. The latent semantic structure is not captured because the hash function does not take into account it.

One of the most related studies of this paper is similarity encoding for dirty categorical variables [2]. They focused on the problems where categorical variables are dirty, and showed that encoding categorical variables by similarity vector achieved high predictive performance. Their proposed method can be seen as

disambiguation of spelling variants, and the ambiguity of categorical variables is partially represented by spelling variants. Although the similarity encoding partially resolve the ambiguity problems, the goal of the method is quite different from our goal to encode ambiguous categorical variables explicitly.

Embedding-based encoding methods also partially resolve the ambiguity problems. Guo [7] proposed an entity embedding method based on neural networks. The entity embedding implicitly encodes the semantic structure of categorical variables. As pointed out in Cerda et al. [2], the cost of training these models and tuning their structure is heavy, and the initial encoding scheme could be combined with their similarity encoding methods. Our encoding methods can also be used as well as the similarity encoding methods.

2.3 Imputation methods

The difference between ambiguous categorical variables and missing values is the degree of ambiguity. In classical missing value imputation approach, the definition of X^j *is missing* can be formulated as $O^j = \{C^{j,l}\}_{l=1}^{k_j}$, where unknown category is out of scope. Such a similarity implies that we can handle the problem of encoding ambiguous categorical variables using missing value imputation methods.

MissForest MissForest [14] is an non-parametric missing value imputation method for mixed-type data based on Random Forests [1]. This method repeats two steps until convergence: (i) *create predictive models of Random Forests for each independent variable* and (ii) *impute data from the model*. After convergence, the imputed data are considered as the training set and used to create final predictive models. The test set with missing values are imputed by the models trained in the step (i) to make a final prediction. We need to modify the two steps because the original method did not take into account ambiguous categorical variables.

Multiple imputation In statistics, multiple imputation methods are often used to analyze data with missing values. The goal of those methods is to estimate the parameters of statistical models by generating multiple datasets. Although there exist various types of imputation methods [13, 15], they cannot be used to make predictions for new data.

3 Proposed methods

We propose two types of encoding methods to tackle the difficulty of encoding ambiguous information explicitly. In this paper, we assume that only one variable contains ambiguous information for simplicity. We encode other categorical variables by either One-Hot encoding or Ordinal encoding, and numerical variables by standardization.

3.1 Naive Extensions of One-Hot encoding

We first extend One-Hot encoding methods to incorporate ambiguous information explicitly. In One-Hot encoding, the output vector can be interpreted as the probability of each category. Let Z be the encoded vector by $enc^j_{OH}(X^j)$ as shown in equation (2). If the Z_l equals 1, we know that X^j equals $C^{j,l}$ with probability one. We use this probabilistic view to represent ambiguous information, which can be regarded as missing value imputation without considering heterogeneous ambiguity. If we observe that X^j is either $C^{j,1}$ or $C^{j,2}$, natural encoding methods should output a vector where $Z_1 + Z_2 = 1$, $0 \leq Z_1,\ Z_2 \leq 1$ and $Z_l = 0$ $(l > 2)$. There are several variations of how to output Z_1 and Z_2. We propose three methods and compare the output of Z_1 and Z_2.

Uniform encoding Uniform encoding assings the same value to each candidate category, where $Z_1 = Z_2$. This method assumes that we have no prior information about the distribution of categorical variables.

Empirical encoding Empirical encoding assigns the value of each candidate category proportional to the empirical distribution. Let the count vector of the categorical variable be $c_1, c_2, \cdots, c_{k_j}$, calculated from non-ambiguous data in the training set. The output of this method is as follows: $Z_1 = (c_1{+}1)/(c_1{+}1{+}c_2{+}1)$, $Z_2 = (c_2 + 1)/(c_1 + 1 + c_2 + 1)$. We add the value *+1* for smoothing parameter.

Unnormalized encoding We also propose Unnormalized encoding, where $Z_1 = Z_2 = 1$. Note that this method does not reflect the ambiguous information precisely, but reflect the situation that X^j is $C^{j,1}$ and $C^{j,2}$.

3.2 Ambiguous Forests

The naive encoding methods in section 3.1 assign the same numerical values for all instances that have the same observation of the variable X^j. Although this restriction results in simple encoding rules, encoding methods that consider other independent variables of each instance are expected to give better feature representations because they consider the heterogeneous ambiguity. To represent the heterogeneous ambiguity, we propose to construct predictive models for ambiguous categorical variables. The predictive models reduce the ambiguity of ambiguous categorical variables. We call the methods *Ambiguous Forests* inspired by MissForest[2].

Simple MissForest-based encoding We first introduce the simple encoding method based on MissForest. The difference between this method and the original MissForest algorithm is that (i) *the algorithm outputs probabilites* and

[2] Obviously, other classification models can be used as predictive models

(ii) *probabilities are normalized by ambiguous information.* As the original MissForest algorithm, we fit predictive models of ambiguous variable X^j using only non-ambiguous labels.

The transformation process is slightly complicated. First, for each instance $i \in \{1, 2, \cdots, n\}$, we encode all independent variables except for the ambiguous variable x^j using an encoder E as follows: $x_i^E \leftarrow E([x_i^N, x_i^{C\setminus j}])$, where $x_i^{C\setminus j}$ is defined by dropping x_i^j from x_i^C. Next, we define non-ambiguous instances na as follows: $na \leftarrow \{i \mid 1 \leq i \leq n,\ \sum_{j=1}^{k_j} AI_i^j = 1\}$. Then, for each instance $i \in na$, we get a feature vector using ambiguous indicator: $z_i^{j,l} \leftarrow AI_i^{j,l}$. Alternatively, for each instance $i \notin na$, we calculate a probability vector $z_i^j\ (\in \{[0,1]\}^{k_j}) \leftarrow M_{ce}(x_i^E)$, where M_{ce} is a predictive model of ambiguous categorical variable that outputs probabilites. Next, we normalize the probability vector to reflect the ambiguous indicator as follows: $z_i^{j,l} \leftarrow \frac{AI_i^{j,l} z_i^{j,l}}{\sum_{l'} AI_i^{j,l'} z_i^{j,l'}}$. Lastly, the concatenation step of x_i^E and z_i^j provides each feature vector.

Learning from partial labels Our *Simple MissForest-based encoding* method discards ambiguous information since ordinary multiclass learning uses only non-ambiguous labels. To overcome such a limitation, we propose a novel predictive method for ambiguous categorical variables based on learning from partial labels.

The *one-versus-all* (OVA) loss of ordinal multiclass learning is often defined as follows:

$$L_{OVA}(f(X), Y) = l(g_Y(X)) + \sum_{Y' \neq Y} l(-g_{Y'}(X)), \tag{3}$$

where X is an independent variable, Y is the corresponding true class label, g_Y is a binary classifier for class Y versus the rest classes and l is some loss function for binary classification [17]. Cour et al. focused on the learning from partial labels, where the labels are given as a candidate set, only one of which is correct [4]. We propose to apply their learning methods to encoding categorical variables with ambiguity. We use the "naive" partial loss represented by equation (3) in their paper:

$$\tilde{L}_{OVA}(f(X), O^j) = \frac{1}{k_j - S} \sum_{l \in O^j} l(g_l(X)) + \sum_{l \notin O^j} l(-g_l(X)), \tag{4}$$

where $S = \sum_{l=1}^{k_j} AI^{j,l}$. There are two rational reasons to use the loss of equation (4). One is that the "naive" partial loss is a natural extension of the loss of learning with complementary labels [9], and the other is that the "naive" partial loss is easier to implement than other losses of learning from partial labels [4]. This method eliminates the drawback of ordinary multiclass learning since it uses label information as much as possible. We optimize the empirical loss of equation (4). The transformation process is shared in *Ambiguous Forests.*

4 Experiments

We use two real-world datasets [5] for experiments. We preprocess them to generate ambiguous categorical independent variables. We focus on the relationship between the dataset's ambiguity and the performance of encoding methods. The details of the datasets and parameters are summarized in section 4.3.

4.1 Experimental settings

Generating ambiguous dataset First of all, we train Random Forests to predict a target class Y using original training set. Hyperparameters are tuned by 5-fold cross-validation and feature importances are calculated by the tuned model. A categorical variable that has the highest feature importance is then selected as a variable that would contain ambiguous values. When we give ambiguity to the variable, we set the maximum ratio of ambiguous instances and define mapping functions to translate from original non-ambiguous values to ambiguous values. Next, a generator of the ambiguous variable picks each mapping function one by one, samples an instance from candidate instances that could be input data of the function at random and ambiguate the value by the mapping function. This procedure repeats until the ratio of ambiguous instances gets the maximum ratio or no candidate instance exists for all mapping functions.

We set various types of maximum ratio for the training set and test set separately. For each ratio, we generate ten ambiguous sets by running the generator of ambiguous variables to consider random variability. In figure 1b, for example, we visualize boxplots of the F1 (micro averaged) score of the test sets for 100 pairs of the (training, test) sets where the train ratio of ambiguity and the test ratio of that are both 0.5. Note that, in figure 1a, the boxplots are calculated by 10 pairs of the (training, test) sets because there is no random variability of ambiguity when the train ratio of ambiguity is 0. Therefore, we consider only the variability of the test set when the train ratio of ambiguity is 0.

Implementation details We standardize numerical variables as a preprocessing stage of the final supervised learning. We use four modules (HashingEncoder, TargetEncoder, OneHotEncoder and OrdinalEncoder) of categorical encoders as existing encoding methods[3]. We evaluate the difference of impact between One-Hot encoding and Ordinal encoding for categorical variables except for an ambiguous variable. We also compare two types of feature representations of proposed Ambiguous Forests: (i) *a probability vector itself* and (ii) *an ordinal label of the class with the highest probability*.

We extend the OneVsRestClassifier module in scikit-learn [11] to incorporate multilabel learning with sample weights to implement our Ambiguous Forests based on learning from partial labels as equation (4). The implementation of our *Simple MissForest-based encoding* method is based on Random Forests implemented in scikit-learn. Hyperparameters of these Ambiguous Forests are fixed for

[3] http://contrib.scikit-learn.org/categorical-encoding/

simplicity. We use Linear LogisticRegression (**LR**) and Random Forests (**RF**) implemented in scikit-learn as final predictive models. Hyperparameters of the final models are tuned by cross-validation as mentioned above.

4.2 Results and Discussion

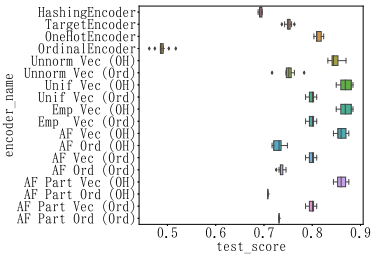
Our results of experiments are plotted in figure 1 and figure 2. The abbreviations we used in y-axis are as follows: Unnorm: *Unnormalized encoding*, Unif: *Uniform encoding*, Emp: *Empirical encoding*, AF: *Ambiguous Forests based on simple MissForest*, AF Part: *Ambiguous Forests with partial label approach*, (OH): *One-Hot encoding for non-ambiguous categorical variables*, (Ord): *Ordinal encoding for non-ambiguous categorical variables*, Vec: *Vector representation for ambiguous categorical variables*, Ord: *Ordinal representation for ambiguous categorical variables*.

The experimental results of *car evaluation* dataset in figure 1 and 2 show that our methods that focus on encoding ambiguous categorical variables achieve extremely high performance when the train ratio of ambiguity is 0 and the test ratio of that is 0.5. In paricular, our methods improve F1 (micro average) score by 0.05 in 1a. This would be because existing methods encode all ambiguous values into an unknown category. In such a situation where the ambiguity of the train set and test set is qualitatively different, our methods work well.

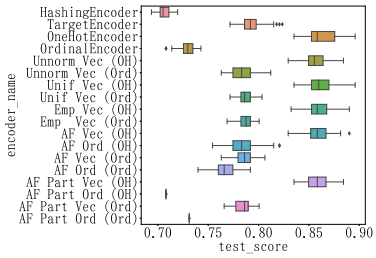
When the train ratio is 0.5, our methods slightly outperform existing methods. This would be because our methods overcome the drawbacks of existing methods that do not consider the semantic structure of ambiguity.

The experiments also show that Random Forests tend to work well with encoders that do not increase the dimensionality and Logistic Regression tends to work well with encoders that are based on vectorization. This result would add empirical evidence that tree-based algorithms and ordinal encoders are good compatibility. There is an exception of the tendency that Ambiguous Forests do not work with encoders that output an ordinal label of the class with the highest probability, which would be because they result in the point estimation of the ambiguous categorical variable. To improve the predictive performance, sampling and ensembling like ordinary imputation algorithms would be useful.

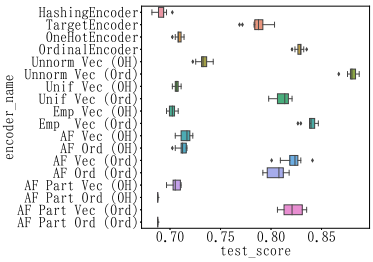
Ambiguous Forests do not outperform out simple methods proposed in section 3.1 in the experiments of *car evaluation* dataset. However, Ambiguous Forests based on learning from partial labels, *AF Part Vec (ord)*, outperform other encoders in *adult* dataset. These observations would be because *adult* dataset is enough large and informative to create accurate imputation models for the ambiguous categorical variable. From these observations, if we use a more complicated dataset that contains multiple ambiguous variables and construct more sophisticated algorithms such as sampling and ensembling, Ambiguous Forests could achieve much higher performance. It would be a useful insight that the naive encoding methods perform in small and simple datasets and Ambiguous Forests perform in large and complex datasets.



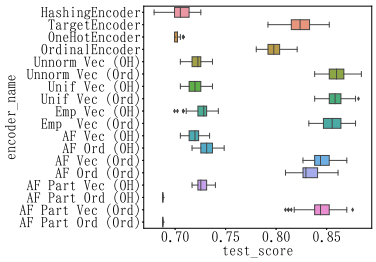
(a) LR (train ratio: 0, test ratio: 0.5)



(b) LR (train ratio: 0.5, test ratio: 0.5)

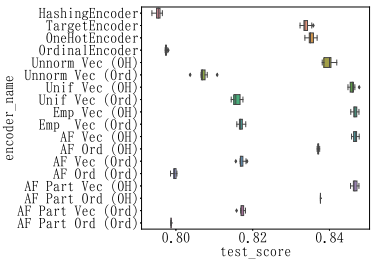


(c) RF (train ratio: 0, test ratio: 0.5)

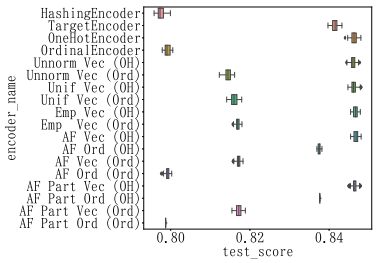


(d) RF (train : 0.5, test : 0.5)

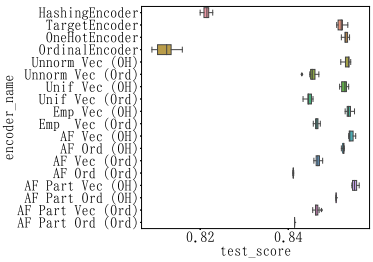
Fig. 1: F1 scores of car evaluation dataset



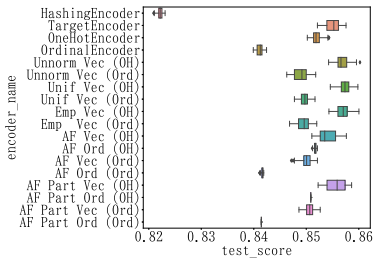
(a) LR (train ratio: 0, test ratio: 0.5)



(b) LR (train ratio: 0.5, test ratio: 0.5)



(c) RF (train ratio: 0, test ratio: 0.5)



(d) RF (train : 0.5, test : 0.5)

Fig. 2: F1 scores of adult dataset

4.3 Details of experiments

We use two public-available datasets of UCI [5]. One is the *car evaluation* dataset that contains 1728 data. We make *safety* column to be ambiguated. We split the original *car evaluation* dataset to make the size of the test set 20% of the original dataset using a stratified split. Another dataset is the *adult* dataset, where the training set contains 30162 data and the test set contains 15060 data, where we remove missing values by listwise deletion. We make *education* column to be ambiguated. We remove the *education-num* variable since it is completely correlated with the *education* variable. We visualize boxplots of the F1 (micro averaged) score of the test sets.

Hyperparameters for car evaluation dataset is tuned as follows: *Random Forests:* $\{$*n estimators:* $[30, 50, \mathbf{100}]$*, max features:* $[0.6, \mathbf{1}]$*, min samples leaf:* $[\mathbf{5}, 10]$*, max depth:* $[3, 5, \mathbf{7}]\}$ and *Logistic Regression:* $\{$*C:* $[0.01, 0.1, 1, \mathbf{10}]$ $\}$, and that for adult dataset is tuned as follows: *Random Forests:* $\{$*n estimators:* $[50, \mathbf{100}, 300]$*, max features:* $[\mathbf{0.6}, 1]$*, min samples leaf:* $[\mathbf{5}, 10]$*, max depth:* $[3, 5, \mathbf{7}]\}$ and *Logistic Regression:* $\{$*C:* $[0.01, \mathbf{0.1}, 1, 10]\}$. The bold-font parameters are selected by grid search. Hyperparameters of Ambiguous Forests are fixed as follows: $\{$*n estimators:* 30*, max features:* 1*, min samples leaf:* 5*, max depth:* 7$\}$. The description of the parameters conforms to that of scikit-learn.

We define the mapping functions that translate original non-ambiguous values into ambiguous values as follows. First, we prepare candidate sets composed of arrays of categories for each dataset: *car evaluation dataset:* { *[med, low], [med, high]* }*, adult dataset:* { *[Some-college, Assoc-voc, Assoc-acdm, Bachelors], [Prof-school, Doctorate, Masters], [Some-college, Assoc-voc, Assoc-acdm, HS-grad], [1st-4th, 5th-6th, 7th-8th], [9th, 10th, 11th, 12th], [Preschool, 1st-4th]* } . Next, for each array a, we choose b $(2 \leq b \leq |a|)$ categories as a mapping function where a is the length of an array a. For example, when we choose *[Some-college, Assoc-voc, Assoc-acdm]* from an array *[Some-college, Assoc-voc, Assoc-acdm, HS-grad]*, the mapping function transforms a category that is included in *[Some-college, Assoc-voc, Assoc-acdm]* into the new category, *Some-college or Assoc-voc or Assoc-acdm*. The number of mapping functions for each array a is $\sum_{b=2}^{|a|} {}_{|a|}\mathrm{C}_b$, where C means a combination.

5 Conclusion

We propose two types of encoding methods to tackle the difficulty of preprocessing categorical variables that contain the semantic structure of ambiguity. We confirm that our methods show higher performance than existing methods by various types of empirical evaluations. Naive extensions of One-Hot encoding would perform in small and simple datasets. Ambiguous Forests contain a novel application of learning from partial labels to MissForest-based encoding methods, and they would perform in large and complex datasets.

Despite the significance of encoding categorical variables with ambiguity, few studies have focused on the problems. We hope that our formulation and

proposition would accelerate the research in the field of encoding such categorical variables. Our future work is to study more sophisticated algorithms and evaluate their performance using more complex datasets.

References

1. Breiman, L.: Random forests. Machine learning **45**(1), 5–32 (2001)
2. Cerda, P., Varoquaux, G., Kégl, B.: Similarity encoding for learning with dirty categorical variables. Machine Learning **107**(8-10), 1477–1494 (2018)
3. Cohen, P., West, S.G., Aiken, L.S.: Applied multiple regression/correlation analysis for the behavioral sciences. Psychology Press (2014)
4. Cour, T., Sapp, B., Taskar, B.: Learning from partial labels. Journal of Machine Learning Research **12**(May), 1501–1536 (2011)
5. Dua, D., Graff, C.: UCI machine learning repository (2017), http://archive.ics.uci.edu/ml
6. Eye, A.v., Clogg, C.C.: Categorical variables in developmental research: methods of analysis. Academic Press, New York; London (1996)
7. Guo, C., Berkhahn, F.: Entity embeddings of categorical variables. arXiv preprint arXiv:1604.06737 (2016)
8. Hüllermeier, E.: Learning from imprecise and fuzzy observations: Data disambiguation through generalized loss minimization. International Journal of Approximate Reasoning **55**(7), 1519–1534 (2014)
9. Ishida, T., Niu, G., Hu, W., Sugiyama, M.: Learning from complementary labels. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. pp. 5644–5654. Curran Associates Inc. (2017)
10. Micci-Barreca, D.: A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems. ACM SIGKDD Explorations Newsletter **3**(1), 27–32 (2001)
11. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. Journal of machine learning research **12**(Oct), 2825–2830 (2011)
12. Prettenhofer, P., Louppe, G.: Gradient boosted regression trees in scikit-learn. PyData 2014 (2014)
13. Schafer, J.L.: Analysis of incomplete multivariate data. Chapman and Hall/CRC (1997)
14. Stekhoven, D.J., Bühlmann, P.: Missforest–non-parametric missing value imputation for mixed-type data. Bioinformatics **28**(1), 112–118 (2011)
15. Van Buuren, S., Oudshoorn, K.: Flexible multivariate imputation by MICE. Leiden: TNO (1999)
16. Weinberger, K., Dasgupta, A., Langford, J., Smola, A., Attenberg, J.: Feature hashing for large scale multitask learning. In: Proceedings of the 26th Annual International Conference on Machine Learning. pp. 1113–1120. ACM (2009)
17. Zhang, T.: Statistical analysis of some multi-category large margin classification methods. Journal of Machine Learning Research **5**(Oct), 1225–1251 (2004)