Encoding Matters: Impact of Categorical Variable Encoding on Performance and Bias

Daniel Kopp[1] **Benjamin Maudet**[2] **Kristin P. Bennett**[1]

1) Rensselaer Polytechnic Institute 2) Universite of Paris-Saclay

With the availability of ML packages that automate preprocessing to the greatest extent possible, it is tempting to elude the problem of categorical variable encoding and rely on default settings, e.g. label encoding or one-hot encoding. However, performance and bias are influenced by the choice of encoding in conjunction with the choice of model, the specific type of categorical variable, the number of categories, the size of the dataset, and the particular problem context. In this paper, we empirically revisit—on synthetic and real data—the issue of variable encoding and its impact on modeling bias. We introduce a new metric of feature influence to quantitatively evaluate such effect. The results indicate that contrary to common practice, ordinal and nominal variables should not necessarily be coded differently. Indeed, differences in coding can lead to differences in variable availability to the model, which dominate other effects and lead to modeling bias (and thus potentially to unfairness). We remark that well regularized universal approximators can handle variables encoded identically (regardless of whether they are ordinal or nominal), without adverse effects on generalization, while reducing problems of bias. Various encodings may yield better results in various contexts, so the choice of encoding should be considered a hyper-parameter of the workflow.

In machine learning, it is common to encounter various data types that need to be properly encoded before being input into the learning model. Some models, such as linear models and neural networks, work well with quantitative data and require numerical inputs [2]. Other models, like decision trees and grid models, can also handle qualitative data, such as categorical variables [4]. Due to the increasing importance of neural networks in modern AI, it is crucial to correctly encode categorical variables to ensure these models function properly, achieve good generalization, and avoid bias.

Categorical variables are variables that can take one value from a set of possible values. Each possible value of a variable is referred to as a level. We consider two types of categorical variables: Ordinal variables with a meaningful ordering, *e.g.,*education levels, and nominal variables, which are qualitative with no meaningful ordering, *e.g.,*race. Although nominal variables may be numbered (*e.g.,*user ID's), these numbers do not indicate any particular order. Most textbooks [5, 7] recommend encoding ordinal and nominal variables differently. Typically, ordinal variables are encoded with a numeric value using their natural order. In contrast, nominal variables are encoded with one-hot encoding, converting k-level categorical variables into a k-dimensional binary vector representation where only one of the variables is "hot" (set to 1) for each observation, while all the others are "cold" (set to 0) [8]. This choice, while natural, can be far from neutral. We show that, everything else being equal, *i.e.,*given two variables with identical dependency to the target, mixing ordinal encoding and one-hot encoding introduces an asymmetry and can lead the model to prefer one variable over the other. Also, if one variable is more strongly relevant than the other (for instance, one variable belongs to the Markov blanket of the target and the other does not), an asymmetry in coding can lead a model to wrongly give more importance to the weaker predictor, which could be a confounder and lead to unfairness or bias in predictions.

Formally, we study supervised learning problems in which data samples $(\mathbf{x}, y) \in \mathcal{X}^d \times \mathcal{Y}$ are drawn from a joint distribution $P(\mathbf{x}, y) = P(\mathbf{x})P(y|\mathbf{x})$. The problem is to approximate $P(y|\mathbf{x})$ using machine learning (ML). To that end, training data are available, drawn *iid* from $P(\mathbf{x}, y)$:

Preprint. Under review.

$\{(\mathbf{x}^1, y^1), \cdots, (\mathbf{x}^n, y^n)\}$. We treat the learning procedure as a black box returning an estimator $\hat{P}(y|\mathbf{x})$. Our only degree of freedom is the choice of variable encoding. If a variable is used disproportionately over another in $\hat{P}(\mathbf{x}, y)$, not adequately reflecting statistical dependencies, there may be an introduction of bias in predictions. We are particularly interested in the case where the components of $\mathbf{x}$ are categorical variables and we study the effect of encoding on introduced modeling bias.

After experimenting with a wide variety of codes, we selected a few codes that illustrate our point well. Besides the Natural Ordinal encoding, used for ordinal variables, and One-Hot encoding, often used for nominal variables, we consider as a baseline Random Ordinal (assigning a random permutation of the level numbers to the categories) and two variants of Target Encoding: Target Continuous, which assigns to each level the average of the target values of training samples in that category, and Target Ordinal, which selects the ordinal encoding that correlates most to the target.

Due to their discrete nature, categorical variables are drawn from a discrete joint categorical distribution. We call "cell" a particular combination of levels in the various input dimensions (*e.g.,*, if $\mathbf{x}$ is composed of the two variables "race" and "education level", all individuals with "graduate school" education level and whose race is "black" would constitute one cell.) While the training data are drawn using the natural distribution $P(\mathbf{x})$, test data are drawn from either the same distribution (in-domain generalization) or from a uniform distribution in which all cells have equal probability, even those not present in training data (out-of-domain generalization). This aligns with evaluating the predictor $\hat{P}(y|\mathbf{x})$ in a "fair" way, not privileging any cell, even for strongly underrepresented cells.

To evaluate the impact of variable encoding on modeling bias, we devise a method that tests $\hat{P}(y|\mathbf{x})$ using out-of-domain data, which does not require ground-truth knowledge of the target y, because, in real data, we may have no representatives of strongly under-represented cells. The method compares $\hat{P}(y|\mathbf{x})$ (reference predictions) and $\hat{P}_i(y|\mathbf{x})$ obtained by "neutralizing" the effect of variable x_i by randomly permuting values assigned to x_i, to obtain a coefficient c_i measuring the predictive importance of variable i. We study how the relative predictive importance of variables changes with different variable encodings. Details on the calculation of c_i are provided in supplemental material [1].

In numerical experiments on real data [1, 6] and synthetic data, we compare codes pairwise by encoding half of the variables with one code and half with another. We fix the ML model (using default hyper-parameters) to obtain $\hat{P}(y|\mathbf{x})$. For each encoding combination, we evaluate the difference in predictive importance of the two sets of variables under study $R = c_1 - c_2$, where c_i is the predictive variable importance obtained by the permutation method. R is influenced both by the statistical dependency between x_1, x_2 and y and by the "availability" of variables x_1 and x_2 to the given model. Here, "availability" is the ease with which the model exploits x_1 or x_2 to make predictions [3], which depends on variable encoding. Let e_k be the encodings of variables x_1 and x_2 ($e_k \in \{Target\ Continuous, Target\ Ordinal, Random\ Ordinal, One-Hot\}$). We compute $R(e_i, e_j)$ for each pair of encodings, e_i encoding variable x_1 and e_j variable x_2, and compare values.

For didactic purposes, we designed a synthetic example in which x_1 and x_2 are identical and equivalently statistically dependent upon y. Hence, R solely depends on variable availability. If both variables are equally favored, we expect that $R(e_i, e_j) = 0$. If $R(e_i, e_j) > 0$ then it means the choice of encoding favors variable x_1 over x_2 while if $R(e_i, e_j) < 0$, x_2 is favored over x_1. In either case, $R(e_i, e_j) \neq 0$ indicates bias introduced by coding. This is indeed what we confirm experimentally for $i \neq j$. We also implemented a synthetic example in which the two variables are not identical and are identically related to the target. The (causal) relationship between y and x_2 is deterministic, but that between x_1 and x_2 is noisy and there is no direct relationship between x_1 and y ($x_1 = x_2 + noise \leftarrow x_2 \rightarrow y = \theta(x_2)$). Hence, x_2 should be more predictive of y. However, depending on the coding, some models can learn to make x_1 more predictive. This could affect the fairness of decision making. Consider, for example, the case where x_2 is a person's age, y is whether the person should be admitted to a nightclub serving alcohol, and x_1 is the person's height.

In conclusion, our findings indicate that encoding ordinal and nominal variables differently may introduce bias and we advocate using the same code for all categorical variables to reduce out-of-distribution bias. The optimal code to be chosen depends on the predictive model, the amount of available data, and the specific problem, and can be selected *e.g.,* by cross-validation.

[1] https://github.com/danielkopp4/BayLearn2024Appendix

Acknowledgements

Many thanks to Isabelle Guyon for advising this research. This work was partially supported by Chalearn.

References

[1] Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

[2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[3] Katherine L Hermann, Hossein Mobahi, Thomas Fel, and Michael C Mozer. On the foundations of shortcut learning. *arXiv preprint arXiv:2310.16228*, 2023.

[4] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, et al. *An introduction to statistical learning*, volume 112, page 315. Springer, 2013.

[5] Max Kuhn and Kjell Johnson. *Feature engineering and selection: A practical approach for predictive models*. Chapman and Hall/CRC, 2019.

[6] Cook County State's Attorney Office. Sentencing courts dataset, Updated April 5, 2023.

[7] scikit-learn developers. Encoding of categorical variables. `https://inria.github.io/scikit-learn-mooc/python_scripts/03_categorical_pipeline.html`.

[8] Alice Zheng and Amanda Casari. *Feature engineering for machine learning: principles and techniques for data scientists*. " O'Reilly Media, Inc.", 2018.