Stochastic Target Encoder - A new categorical feature encoding applied to urban data regression problems

João Victor Araujo[1], **Gean da Silva Santos**[1], **Andre L L Aquino**[1], **Fabiane Queiroz**[2]

[1]Instituto de Computação - UFAL
Cidade Universitária 57072-970 — Maceió — AL — Brazil

[2]Campus de Engenharias e Ciências Agrárias - UFAL
BR-104 57100-000 - Rio Largo - AL - Brazil.

{joao.araujo, gean.santos, alla, fabiane.queiroz}@laccan.ufal.br

Abstract. *Regression problems are Machine Learning (ML) tasks often found in real world, with many attributes being categorical. Most ML algorithms works only with numerical data, so encoding these attributes tends to be necessary, but common encoding methods don't use data properties, which can lead to poor model performance on high cardinality data. Target Encoding methods address this, but encode each attribute into a discrete set of values of equal cardinality to the categorical attribute. We propose a Target Encoder that addresses both issues introducing variability to encoded data using target statistics, achieving results comparable with the existing Target Encoders. We test our method against existing Encoders, showing the robust performance of our method.*

1. Introduction

Handling categorical data constitutes a critical aspect of preparing datasets for ML tasks, especially for regression problems. In some cases, these features have high cardinality, and most ML algorithms require numerical input, so categorical features need to be transformed into some numerical representation or discarded. However, discarding such data could lead to information loss, thereby diminishing the performance of ML models. In this way, tackling the data transformation is usual when we have some categorical data in a dataset. The process of transforming categorical data into a numerical form is known as *encoding* with a taxonomy described by [Pargent et al. 2022] as *Target-Agnostic* and *Target-Based Encoding*.

The world around us has various phenomena monitored by devices provided with sensing, processing, and communication capabilities. Machine learning models are commonly used in the processing and make inferences over the data collected by different IoT (Internet of Things) sensors. These models play a significant role in attenuating problems in urban centers in various ways as Traffic Management, Public Transportation Optimization, Urban Planning and Development, Air Quality Monitoring, and Pollution control, among others. In this work, we analyze data related to urban problems, such as air pollutant levels and urban mobility services demand estimation, in a regression manner. Air pollutants such as CO^2, NO_3, NO_x and $PM_{2.5}$, PM_{10} bring many hazards to humans, with previous works citing that high levels of PM_{10} might cause cardiovascular and respiratory mortality [Leili et al. 2023], while [Su et al. 2023] say that the same pollutant may degrade kidney function. Urban mobility services like bike-sharing, would attenuate air pollutants emissions produced by cars and motorcycles [Cao et al. 2023]. Datasets related to these problems might include high cardinality attributes, such as spatial data (cities or regions) and temporal data (e.g. dates) or domain-specific attributes, so handling high cardinality attributes is an important aspect of modeling urban problems with data-driven approaches.

Categorical encoding remains an ongoing challenge in data science research. We propose a novel *Target-Based Encoder* tailored for regression problems in this context. Our encoder introduces consistent variations into the transformed data by utilizing data dispersion information for the categorical values, what, for the best of our knowledge, isn't an approach explored in existent *Target-Based* Encoders. Our approach yielded promising results in experimental analysis, consistently ranking among the top-performing encoders in tests with the KNN algorithm. This underscores its robust performance in diverse scenarios.

The rest of the paper is organized as follows. Section 2 presents the previous works on the Categorical Attributes Encoding problem. Section 3 presents our Categorical Encoding method. Section 4 presents the results of our conducted experiments. Finally, Section 5 concludes and presents some other gaps that may be addressed in future works.

2. Related works

The selection of the following papers began with the works cited in [Pargent et al. 2022], and after the preliminary selection we conduct backward and forward citations search on these selected works to gather more papers related to our method.

The *Target-Agnostic* Encoders transform the categorical data without using any information of the target feature, with some examples being the One-Hot Encoder [Wang et al. 2019, Poslavskaya and Korolev 2023] and Label Encoder [Hien et al. 2020, Poslavskaya and Korolev 2023], with the former adding more dimensions in the dataset (in a high cardinality dataset this may incur in the *curse of dimensionality* problem), and with the latter bringing some order on the categories that may or not exist in the data, and even in the presence of an order, the encoder might don't represent it either.

On the other hand, *Target-Based* Encoders, such as the most known Target Encoder (TE)[Micci-Barreca 2001], bring some kind of information about the target value to transform the categorical features in a dataset. This information might be the mean value of the target feature within a categorical value. It is the approach of many *Target-Based* Encoders, with some slight variations between one or another. Some drawbacks of this approach include: (i) a regularization term must be added to the categorical value target mean because if some category has few or even just one sample, the target mean value would be simply the target value itself, which would incur in the *Target Leakage* problem and (ii) these methods are prone to overfitting. To prevent that and other overfitting issues, a global mean of the categorical feature is weighted with the previous encoded value, being this a hyperparameter of the Encoder. The James-Stein Enconder (JSE) follows the work of [James and Stein 1992], adapting it to Categorical Encoding. JSE is similar to TE, but the weighting is done on behalf of the encoder, estimating the mean and variances of categorical values and categorical features. Quantile Encoder [Mougan et al. 2021] (QE) differs from others *Target-Based* approaches because it uses the quantile instead of the mean as target value information, and the authors points out that this leads to better results than mean-based Target Encoders. Utilizing a more different method, [Slakey et al. 2019] models the Encoding problem with a Bayesian approach. K-Encoding [Baboolal et al. 2023] is a encoding method only for regression problems, just like ours. It utilizes target statistics in the encoding, along with a distance based step, which uses euclidean distances of all features to compute the weighted average of target values. For classification tasks only, [Liu et al. 2024] encode categorical features in an multi-grained manner with Decision Forests (DF). The method first recursively split the dataset features in n subsets. For every subset, a DF estimate the probabilities of a given sample being of one of K classes. These estimates are concatenated, and, with a uncertainty-aware non-linear transformation of this numerical data, the Encoder produces the encoded data.

A lot of recent works have used Categorical Encoding for solve real problems, mostly because of the presence of high cardinality categorical features. [Hien et al. 2020] utilizes Categorical Encoding to address the student performance problem. The bayesian Encoder proposed in [Slakey et al. 2019] was used in a Customer Scoring System, to predict if a customer was more or less likely to use the service of the company. [Uyar et al. 2009] utilizes a Frequency Encoder to classify good gametes for *In Vitro Fertilization* problem. [Amihai et al. 2018] approach the prediction the failures of Medical Health Machines in two weeks timespan utilizing Categorical Encoding. [Jiun Hooi et al. 2022] encodes categorical features to solve the Customs Fraud Detection problem. [Wang et al. 2019] predicts traffic flow in a bycicle sharing service encoding categorical features like days of weather descriptions and site IDs and surroundings information). [Fernandez and Xu 2019] utilizes Categorical Encoding to address the Network Intrusion Detection problem encoding high cardinality features such as IP (Internet Protocol) addresses. These are some of the use cases that utilizes Categorical Encoding, demonstrating the usefulness and relevance of this technique in Data Science.

3. Method

In this work, we deal with the conventional scenario of supervised learning for regression problems. Then, we work with a tabular dataset of size N, partitioned into a training set containing n_{TR} records and a test set containing n_{TS} records. This dataset is drawn from the joint distribution $P(x, y)$, comprising a set of feature values x and a corresponding target continuous valued feature $y \in \mathcal{Y}$.

For any given Categorical Encoder we must split the Data in Train/Test Data before the Encoding step. This is specially important to *Target Based* Encoders, as to avoid *Target Leakage* problem, i.e. using target statistics of Test Data to encode Training Data. Thus, we utilize the Training Data to ”fit” the Encoder, i.e. acquiring the target statistics to encode the Train/Test splits. For the Test Data we need only to use the Encoder to transform it, without acquiring target statistics of this Data. The output of the Encoder is the Train/Test Data only in numerical form.

3.1. Target-based Encoders

For a particular categorical feature defined by the set $\mathcal{C}$, we can to establish that $\mathcal{C}$ is defined by an union of m distinct subsets (called *cells*) $\mathcal{X}_i$, that is $\mathcal{C} = \{\mathcal{X}_i\}_{i=1}^{m}$, in which every $\mathcal{X}_i = \{x_1, x_2, x_3, ..., x_{n_i}\}$ has n_i identical feature values. When we have large values for m, we can say that $\mathcal{C}$ is a high-cardinality categorical feature.

As stated in [Micci-Barreca 2001], a target-based encoding process is a transformation that maps individual values of a high-cadinality categorical feature into a scalar estimated value $\widetilde{X}_i$ (called encoded value), representing an estimated expected value. As estimating expected values becomes a crucial aspect of model training, it's essential to solely utilize the data within the training set. In this way, we define the continuous target by Equation 1.

$$\widetilde{X}_i = \lambda(n_i)\frac{\sum\limits_{k\in\mathcal{L}_i} Y_k}{n_i} + (1-\lambda(n_i))\frac{\sum\limits_{k=1}^{n_{TR}} Y_k}{n_{TR}} \tag{1}$$

where $\mathcal{L}_i$ refers to the set of records of size n_i, for which $\mathcal{X} = \mathcal{X}_i$. The regularization parametric function $\lambda(n_i)$ helps to reduce overfitting weighing the mixture of two expected values: the posterior expected value of $\mathcal{Y}$ given $\mathcal{X} = \mathcal{X}_i$ and the prior expected value of $\mathcal{Y}$. The rationale behind using it is that despite the high-cardinality scenario, in general, the records are unevenly distributed across possible values of $\mathcal{X}$. Then, we need to handle cells defined by a small n_i and need to mitigate the effect of these small cells. The regularization term formulation is shown in Equation 2.

$$\lambda(n_i) = \frac{n_i}{s+n_i} \tag{2}$$

the scalar s, called *smooth* term, is the single parameter of $\lambda(n_i)$. For our purposes, we defined that the condition for finding good values for s is to consider that $n_i << s$ for all $\mathcal{X}_i \in \mathcal{C}$. Experimental analysis can be seen in Section 4.

3.2. Stochastic Target-based Encoder

As we can observe, the target-base encoding generates a discrete set of continuous values, that is it produces equal encoded values $\tilde{x}$ for every encoded cell $\widetilde{\mathcal{X}}_i$, making the estimated numerical set static in some level, presenting several equal values. We noted that this can be an impediment factor to a good approximation of the target values from a given categorical value. One point that denotes this is the fact that the target-based encoder utilizes a regularization function $\lambda(n_i)$, because only relying on $\mathcal{X}_i$ mean would not be reliable. Another point is that we could aggregate other data descriptors to encode $\mathcal{X}_i$. We believe that by aggregating more statistical information belonging to $\mathcal{X}_i$ to encode it, we would be estimating more precisely the target values that would appear in a more diverse data set, bringing more generalization to the encoding procedure. To achieve this, we propose to aggregate the standard deviation of $\mathcal{X}_i$ to $\widetilde{X}_i$, more precisely, for every $\tilde{x}$ in $\widetilde{X}_i$, sample from an uniform distribution based on the standard deviation of $\mathcal{X}_i$ and sum it to $\tilde{x}$.

First, consider a set $\widetilde{\mathcal{U}} = \{\tilde{u}_{\tilde{\mathcal{A}}_i}\}_{i=1}^{m}$, in which m is the cardinality of the categorical set and $\tilde{u}_{\tilde{\mathcal{A}}_i}$ refers to the unique value extracted of each encoded set $\widetilde{\mathcal{A}}_i$. To achieve the desired variability in the encoded data, we use the standard deviation σ of $\widetilde{\mathcal{U}}$ to map all values into each $\widetilde{\mathcal{A}}_i$ into new different values. To achieve it, we formulate this transformation in Equation 3.

$$\widetilde{\mathcal{V}}_i = \widetilde{X}_i + [r \sim U(\frac{-\sigma}{s}, \frac{\sigma}{s})] \tag{3}$$

where $\mathcal{V}_i$ refers to the new variable encoded values for each cell, r is a random scalar uniformly sampled from an interval limited by σ and the smooth factor s. In other words, every equal value a into each encoded set $\widetilde{\mathcal{A}}_i$ is transformed by addition of a different random component r. This limited interval is needed because $\widetilde{\mathcal{U}}$ belongs to a different distribution from the weighted value $\mathcal{X}_i$, so dividing the interval by s would constrain the interval to not diverge so much the $\mathcal{X}_i$ distribution. Its important observe that if σ is zero the encoded value remains as the same of $\widetilde{X}_i$, because a zero σ means that $\mathcal{X}_i$ has no variability.

3.2.1. Handle missing data

As we cite in Section 3.2, we need also address the case of handling categorical values, that exists only on the training record or haven't been observed on the training records during the encoding process. To accomplish that we utilize the already computed global mean and sum it to an uniformly sampled scalar value from the standard deviation over $\mathcal{Y}$, as formulated in 4.

$$\widetilde{X}_{new} = \frac{\sum\limits_{k=1}^{n_{TR}} Y_k}{n_{TR}} + [r \sim U(\frac{-\sigma_{TR}}{s}, \frac{\sigma_{TR}}{s})] \tag{4}$$

4. Experiments and results

We evaluate our method performance over 4 regression datasets related to urban problems. We either compared our proposal with the state of art Categorical Encoders.

4.1. Datasets

The first dataset is about CO^2 Emissions by Vehicles in chassis dynamometer testings, which was made by the Canada Government and compiled and published in Kaggle platform[1]. The second dataset displays data about a Bike Sharing Demand [Fanaee-T and Gama 2014][2]. The third dataset presents Air Pollution data gathered throughout the years of 2015 and 2017 by 87 measurements stations scattered on the State of São Paulo, Brazil[3]. This dataset includes various pollutants measures, but we choosed to analyze only PM_{10} measurements. The fourth dataset denote AirBnB data[4] related to the accommodations listings, which we use as target value the users reviews attribute. Because of computational constraints, we select a subset of 20000 entries of each dataset (except for the bike sharing dataset, which has approximately 17000 entries) through random sampling. These subsets datasets have a maximum cardinality count of 10479, 11413, 19981 and 11227, respectively, which denotes a high count of cardinality

[1] https://www.kaggle.com/datasets/ahmettyilmazz/fuel-consumption

[2] https://www.openml.org/search?type=datastatus=activeid=42712

[3] https://www.kaggle.com/datasets/samirnunesdasilva/sao-paulo-pollution-data

[4] https://www.kaggle.com/datasets/lovishbansal123/airbnb-data

values. That would promote a valuable test on the usage of categorical encoders to handle high cardinality datasets.

4.2. Experiment settings

Along our encoding method we utilized the following Categorical Encoders: Target Encoder (TE), M-Estimate Encoder (MEE), James Stein Encoder (JSE), Quantile Encoder (QE), Leave One Out Encoder (LOOE) and Ordinal Encoder (OE). We used the KNN regression algorithm in our experiments because its a non-parametric model, that makes estimates based on distance of data samples, so feature importance is absent in this model, what makes it very suitable to investigate if any preprocessing by itself may produce better ML models. The metrics used to evaluate the encoders in our experiments are the Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE).

To have a fair test between the Encoders, we set the Encoders parameters equally between all of them. In the case that an Encoder has an unique parameter, we've set it to the default value provided by the Scikit Learn module responsible for Categorical Encoders, the Category Encoders package. All datasets are normalized with Min-Max Scaling to bring all the attributes to the same scale, to facilitate the learning of the algorithm, given that KNN is sensitive to different scales, because of the distance-based learning it has.

As the smoothing parameter is important to reduce overfitting and target leakage, we empirically select 3 distinct values for the smoothing factor: 100 (as a baseline), 50% and 2 times higher than the most frequent $\mathcal{X}_i \in \mathcal{C}$, for every dataset. These values for the smoothing parameter are used on our method and in TE, MEE and QE (as the others encoders don't have the smoothing parameter). The experiments settings are shared for all datasets. The experiment for a given dataset works as follows:

1. We perform basic preprocessing on the dataset (filling/removing missing data, etc).
2. We get the smoothing parameters s, as stated above.
3. For every encoder E, we perform 10-fold Cross-Validation (CV) on the dataset in the following way:
    (a) We split the CV fold data in train and test sets, being $\frac{9}{10}$ of the data belonging to the train set, and $\frac{1}{10}$ to the test set.
    (b) We encode the train set with E and transform the test set with it.
    (c) We normalize the train and test sets.
    (d) We train the given model on the train set, and test it on the test set.
4. Finally we get the CV results.

4.3. Results and Discussion

To facilitate the visualization of the results, we used the logarithmic scale on the metrics analyzed (y-axis). Although we have made experiments with 3 distinct smooth values, we noticed that the *2x higher* scheme achieved better results overall on all encoders. Figure 1 illustrate that the best results were achieved with *2x higher* scheme, that has the best metrics in 3 out of 4 datasets.

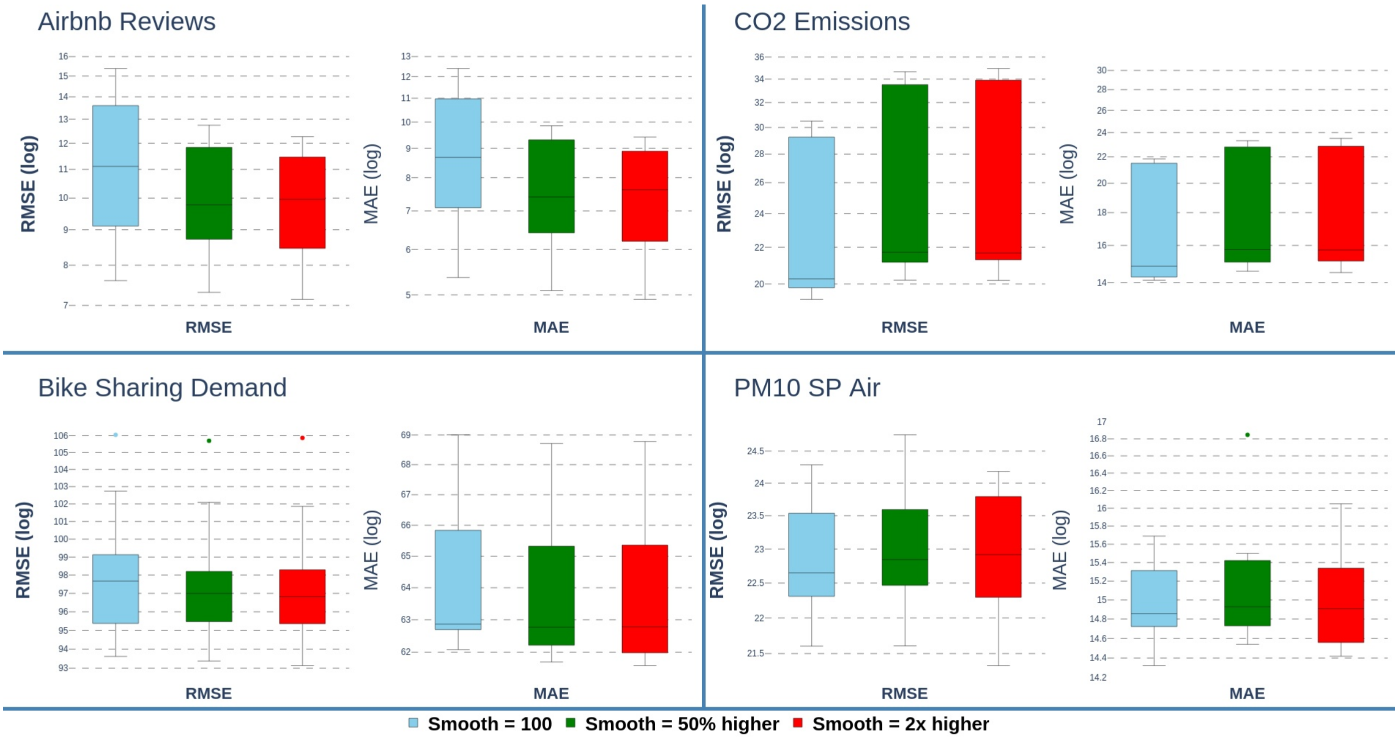

Figure 1. CV results of MAE and RMSE of our method, varying the smooth parameter, for all datasets. Y-axis in log scale. Lower is better.

Guided by that, we present the results only of the encoded values with *2x higher* smooth values. Besides the results based on the smooth values, we notice in our experiments that our method is par with the existent encoders, but has a more stable performance, presenting few outliers, in gereral. One point to note is the smaller boxplots of our method, compared to others encoders, which demonstrate that our method has a more consistent performance through various experiments runs. Figure 2 shows the CVs results of all encoders for every dataset. We can see that our method achieves better results than TE and JSE in the Airbnb Reviews Dataset, while having a smaller deviation. Interestingly, OE achieved very competitive results in this dataset.

On the CO^2 Emissions results the *MAE* has extremely high values on QE and JSE, with the other encoders (except ours) having high outlier values on the CVs. The *RMSE* metric does not have that issue, and like in the Airbnb Dataset, our encoder has similar values to the MEE and QE.

The Bike Sharing Dataset results shows an almost equally performance between all encoders (except LOOE), both on *MAE* and *RMSE* metrics. In this dataset, the OE has achieved comparable performance to the *Target-Based* Encoders.

The SP Air Pollutants Dataset has the same issue on the *MAE* high values like CO^2 Dataset. Interestingly, is the fact that JSE has a poorer performance on both metrics. Another point is that our method achieves a comparable performance to the other encoders, having a smaller deviation. The OE has a worse performance than the *Target-Based* Encoders, except in comparison to JSE.

What we can see throughout the experiments is that results obtained by other

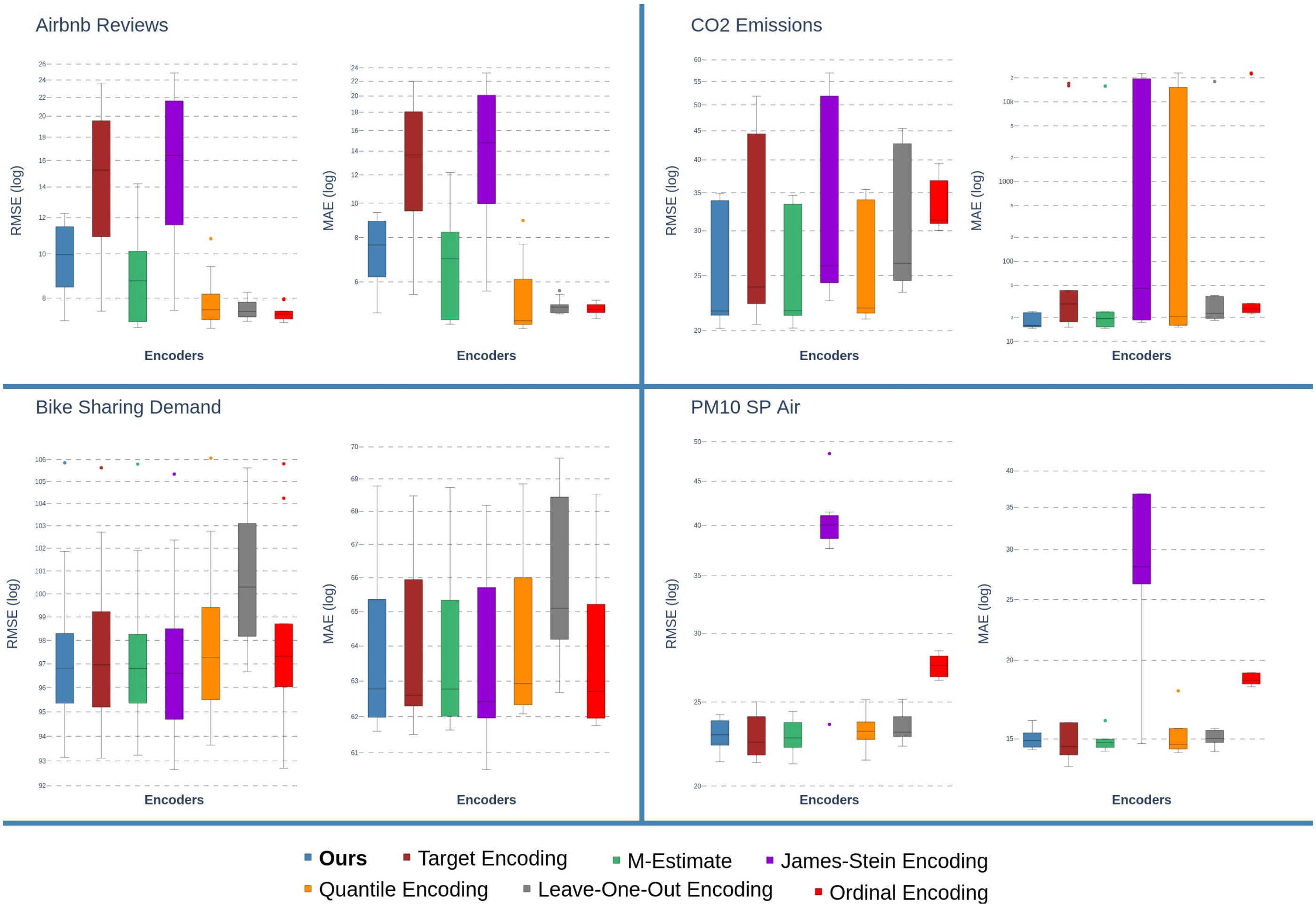

Figure 2. CV results of MAE and RMSE of all encoders, in all datasets. Y-axis in log scale. Lower is better.

encoders vary from dataset to dataset, i.e. an encoder shows good performance in one dataset, but poor results in another dataset. On the other hand, our method shows strong performance across all datasets, being consistently very close to the best result for a given dataset and with a small variation across the CV. This denotes that bringing more variability to encoded data through target statistics can provide better results in regression problems.

Our encoder has achieved a strong performance along the existing encoders, having a small deviation on the CVs, which denotes that it can encode the data bringing consistent results in regression problems, even with multiples training/testing data subsets. The other *Target-Based* encoders have inconsistent results along the CVs, especially TE, JSE, and QE, even though in some cases the last did perform well. An intriguing fact is that *Target-Agnostic* Encoders achieved results close to the *Target-Based* Encoders, as we expected that the results between the two approaches would be more drastically different.

Although, the smoothing parameter is a hyperparameter of our method, as is for the *Target-Based* encoders, as stated in section 3. This parameter depends on the dataset itself, as it has no *one-size-fits-all* value for it, so an *Target-Based* Encoder may lead to poor performance if the smooth parameter is not choose wisely. The dependence on this parameter is a limitation by itself for *Target Based* encoders.

5. Conclusion and future works

As showed in the previous section, our method achieves the same level of performance as the existent encoders, having surpassed some of them. Along this, our method has a more constrained deviation on the CVs tests, with this by itself denoting that our method brings a more reliable encoding to the datasets. Such feature of our method should compensate the fact that our Encoder hasn't achieved the best results, despite being very close.

Our method could be used in data driven solutions towards urban, medical, financial or any other domain with high cardinality categorical features, to tackle real world regression problems through data.

In future works, we would like to investigate further the performance of our method, utilizing more datasets, with various sizes and in various domains. The usage of more regression algorithms it's another thing we'll like to investigate, even performing hyperpameter tuning on the algorithms, to see, for example, if the usage of some encoder might influence the hyperparameter search space of the algorithms. Another element we want to investigate is the smooth parameter, as we haven't enough time nor compute resources to search for this parameter in our experiments.

References

Amihai, I., Chioua, M., Gitzel, R., Kotriwala, A. M., Pareschi, D., Sosale, G., and Subbiah, S. (2018). Modeling machine health using gated recurrent units with entity embeddings and k-means clustering. In *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*, pages 212–217.

Baboolal, K., Gooljar, S., and Hosein, P. (2023). A novel approach to feature encoding. In *2023 IEEE International Conference on Technology Management, Operations and Decisions (ICTMOD)*, pages 1–6.

Cao, G., Zhou, L.-A., Liu, C., and Zhou, J. (2023). The effects of the entries by bike-sharing platforms on urban air quality. *China Economic Quarterly International*, 3(3):213–224.

Fanaee-T, H. and Gama, J. (2014). Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence*, 2:113–127.

Fernandez, G. C. and Xu, S. (2019). A case study on using deep learning for network intrusion detection.

Hien, D. T. T., Thuy, C. T. T., Anh, T. K., Son, D. T., and Giap, C. N. (2020). Optimize the combination of categorical variable encoding and deep learning technique for the problem of prediction of vietnamese student academic performance. *International Journal of Advanced Computer Science and Applications*, 11(11).

James, W. and Stein, C. (1992). *Estimation with Quadratic Loss*, pages 443–460. Springer New York, New York, NY.

Jiun Hooi, E. K., Zainal, A., Kassim, M. N., and Ayub, Z. (2022). Feature encoding for high cardinality categorical variables using entity embeddings: A case study in customs fraud detection. In *2022 International Conference on Cyber Resilience (ICCR)*, pages 1–5.

Leili, M., Bahrami Asl, F., Jamshidi, R., and Dehdar, A. (2023). Mortality and morbidity due to exposure to ambient air pm10 in zahedan city, iran: The airq model approach. *Urban Climate*, 49:101493.

Liu, H., Qiu, Q., and Zhang, Q. (2024). End-to-end approach of multi-grained embedding of categorical features in tabular data. *Information Processing Management*, 61(3):103645.

Micci-Barreca, D. (2001). A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems. *SIGKDD Explor. Newsl.*, 3(1):27–32.

Mougan, C., Masip, D., Nin, J., and Pujol, O. (2021). Quantile encoder: Tackling high cardinality categorical features in regression problems.

Pargent, F., Pfisterer, F., Thomas, J., and Bischl, B. (2022). Regularized target encoding outperforms traditional methods in supervised machine learning with high cardinality features. *Comput. Stat.*, 37(5):2671–2692.

Poslavskaya, E. and Korolev, A. (2023). Encoding categorical data: Is there yet anything 'hotter' than one-hot encoding?

Slakey, A., Salas, D., and Schamroth, Y. (2019). Encoding categorical variables with conjugate bayesian models for wework lead scoring engine.

Su, W.-Y., Wu, D.-W., Tu, H.-P., Chen, S.-C., Hung, C.-H., and Kuo, C.-H. (2023). Association between ambient air pollutant interaction with kidney function in a large taiwanese population study. *Environmental science and pollution research international*, 30(34):82341—82352.

Uyar, A., Bener, A., Ciray, H. N., and Bahceci, M. (2009). A frequency based encoding technique for transformation of categorical variables in mixed ivf dataset. In *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 6214–6217.

Wang, B., Shaaban, K., and Kim, I. (2019). Reveal the hidden layer via entity embedding in traffic prediction. *Procedia Computer Science*, 151:163–170. The 10th International Conference on Ambient Systems, Networks and Technologies (ANT 2019) / The 2nd International Conference on Emerging Data and Industry 4.0 (EDI40 2019) / Affiliated Workshops.