



A Model-Based Approach to Mitigating Artificial Ordering, Dimensional Explosion, and Data Leakage in Categorical Encoding

Gaon Park_Grade 2, Email Address(2410211@jungdong.hs.kr)
Min Woo Lee_Grade 2, Email Address(alsdn0078@gmail.com)
Seungjun Lee_Grade 2, Email Address(iearthpaul@gmail.com)

Abstract

Categorical data encoding is a critical preprocessing step in machine learning and deep learning models, as it enables the effective handling of features with categorical values. Commonly used encoding methods, including one-hot encoding, label encoding, and target mean encoding, suffer from significant limitations such as artificial ordering, dimensional explosion, and data leakage. These issues become particularly severe when dealing with high-cardinality categorical features or limited sample sizes, often resulting in degraded model performance.

In this study, we propose a novel categorical encoding model that simultaneously addresses all three of these challenges. The proposed encoder represents each category as a vector positioned at equal distances from both the origin and all other category vectors, thereby eliminating artificial ordinal relationships inherent in traditional labeling-based encoders. To prevent the dimensional explosion commonly associated with one-hot encoding, the proposed method applies random projection to effectively reduce dimensionality while preserving distances across the vectors. Furthermore, since the encoding process is entirely independent of target labels, the proposed approach inherently prevents data leakage.

Experimental results across multiple machine learning models and datasets demonstrate that the proposed encoding model outperforms commonly used encoding methods in terms of predictive stability and accuracy.

Keywords: Categorical Data Encoding, Geometry Preserving Encoder, Random Projection



I . Introduction and Purpose of the Study

In machine learning and deep learning, the quality of data preprocessing plays a crucial role in determining overall model performance. Among various preprocessing steps, the encoding of categorical features is particularly important, as most machine learning algorithms are designed to operate on numerical inputs.

One-hot encoding is the most widely adopted method for handling categorical variables due to its simplicity and its ability to avoid artificial ordinal relationships between categories. However, it raises a critical drawback: when applied to high-cardinality categorical features, one-hot encoding causes a rapid increase in dimensionality, leading to excessive memory consumption, increased computational cost, and a higher risk of overfitting.

Label encoding offers a more compact representation by assigning integer values to categories, but this approach introduces artificial ordering among categorical values that do not possess inherent ordinal relationships. Such artificial ordering can mislead linear models and distance-based algorithms, resulting in reduced predictive accuracy.

Target mean encoding replaces each category with a value calculated from the target labels and can sometimes improve model performance. However, this method can easily cause data leakage since information from the target variable may be unintentionally used during training. In addition, when the dataset contains few samples or rare categories, the calculated values can be unreliable, which may lead the model to focus too strongly on specific categories.

Although each of these encoding methods has practical advantages, they inevitably suffer from at least one of the following issues: artificial ordering, dimensional explosion, or data leakage. As a result, no universally optimal categorical encoding method has yet been established. This limitation motivates the development of a new encoding model that can address all three issues simultaneously.

II . Theoretical Background and Methodology

Most machine learning algorithms require numerical input, which makes



categorical data encoding a necessary preprocessing step. However, three common encoding methods(which are represented by one-hot encoding, label encoding, and target mean encoding) inherently introduce at least one major limitation: artificial ordering, dimensional explosion, or data leakage.

Label encoding assigns numbers to categories, which creates artificial orderings that do not exist in the original data. One-hot encoding avoids this problem by treating each category separately, but the number of features increases as the number of categories grows. This can make computation inefficient and increase the risk of overfitting, especially when the data contain many categories. Target mean encoding replaces categories with values calculated from the target labels, which may improve performance in some cases. However, it can easily cause data leakage and produce unstable results when the dataset is small or when some categories appear infrequently.

As a result, no single encoding method can solve all of these problems at the same time. In addition, choosing different encoders for each feature in a dataset is often inefficient and difficult to manage. These limitations motivate the development of a unified categorical encoding model that avoids artificial ordering, keeps dimensionality low, and prevents data leakage.

To address these challenges, we propose the Geometry Preserving Categorical Encoder (GPCE), a model-based encoding method designed to represent categorical values without introducing artificial order, excessive dimensionality, or data leakage. GPCE is built on the idea that categorical values should be treated as equally distinct entities rather than ordered quantities.

The core concept of GPCE is to map each category to a numerical vector that lies on a geometric structure where all category vectors are equally distant from one another and from the center. By ensuring equal distances between the vectors, the proposed method removes any unintended ordinal relationships between categories while still allowing categorical information to be expressed numerically.

GPCE first constructs vectors in high-dimensional space such that the distance between every pair of vectors is the same. However, such high dimensionality is impractical for real-world datasets with many categories. To resolve this issue, GPCE applies random projection to reduce the dimensionality to a predefined



lower dimension while approximately preserving the distances between vectors. This dimensionality reduction step is based on the Johnson-Lindenstrauss lemma, which guarantees that pairwise distances are maintained with high probability.

After projection, the resulting vector components are used as numerical features. Because GPCE relies only on the structure of the categorical values and does not use target labels at any stage, the encoding process is completely free from data leakage.

Through this geometric and projection-based approach, GPCE provides a unified solution to the major limitations of conventional categorical encoding methods.

```
# Import
import numpy as np
import pandas as pd

# Define functions
def make_random_projection_matrix(K: int, d: int, rng: np.random.Generator) -> np.ndarray:
    """
    Create a random projection matrix R of shape (K, d) to project
    K-dimensional vectors into d-dimensional space.

    The matrix entries are sampled from a Gaussian distribution:
        N(0, 1) / sqrt(d)

    Parameters
    -----
    K : int
        Original dimensionality (number of categories).
    d : int
        Target embedding dimension.
    rng : np.random.Generator
        NumPy random number generator for reproducibility.

    Returns
    -----
    np.ndarray
        Random projection matrix of shape (K, d).
    """
    R = rng.normal(0.0, 1.0, size=(K, d)) / np.sqrt(d)
    return R

# Define GPC Encoder (Geometry Preserving Category Encoder)
def gpc_encoder(X_train: pd.DataFrame, X_test: pd.DataFrame, cat_cols: list[str], d: int = 16, r: float = 1.0, seed: int
= 42,) -> tuple[pd.DataFrame, pd.DataFrame]:
    """
    Geometry Preserving Category Encoder (GPCE).
    """
```



For each categorical column:

- Categories are mapped (based on TRAIN data only) to vectors lying on a hypersphere of radius r .
- Category geometry is preserved by:
 - 1) Centering one-hot vectors
 - 2) Scaling to fixed norm
 - 3) Applying random projection
 - 4) Re-normalizing after projection
- The same mapping is applied to both train and test sets.

The resulting d -dimensional vectors are expanded into d numerical columns and concatenated with the original numerical features.

Parameters

`X_train` : `pd.DataFrame`

Training feature matrix.

`X_test` : `pd.DataFrame`

Test feature matrix.

`cat_cols` : `list[str]`

List of categorical column names to encode.

`d` : `int`, `default=16`

Target embedding dimension for each categorical feature.

`r` : `float`, `default=1.0`

Radius of the hypersphere on which category vectors lie.

`seed` : `int`, `default=42`

Random seed for reproducibility.

Returns

`(pd.DataFrame, pd.DataFrame)`

Encoded training and test feature matrices.

"""

```
rng = np.random.default_rng(seed)
```

```
# Separate numerical features
```

```
Xtr_num = X_train.drop(columns=cat_cols)
```

```
Xte_num = X_test.drop(columns=cat_cols)
```

```
tr_blocks = [Xtr_num]
```

```
te_blocks = [Xte_num]
```

```
for col in cat_cols:
```

```
    # Treat missing values as a separate category to ensure consistency
```

```
    tr = X_train[col].astype("object")
```

```
    te = X_test[col].astype("object")
```

```
    # 1. Determine category set from TRAIN data only
```

```
    cats = pd.unique(tr)
```

```
    K = len(cats)
```

```
    if K < 2:
```

```
        # If there is only one category, no discriminative information exists.
```

```
        # Assign a constant vector on the hypersphere.
```

```
        const_vec = np.zeros(d, dtype=float)
```

```
        const_vec[0] = r
```

```
        mapping = {cats[0]: const_vec}
```

```
        fallback = const_vec
```



```

else:
    # 2. Construct standard basis vectors e_i (identity matrix)
    I = np.eye(K, dtype=float)

    # 3. Compute centroid (1/K) * 1 vector
    ones = np.ones((K,), dtype=float) / K

    # 4. Center the basis vectors: u_i = e_i - (1/K) * 1
    U = I - ones[None, :] # shape (K, K)

    # 5. Scale vectors so that ||v_i|| = r
    # ||u_i|| = sqrt((K - 1) / K)
    # v_i = r * sqrt(K/(K-1)) * u_i
    scale = r * np.sqrt(K / (K - 1))
    V = scale * U # shape (K, K)

    # 6. Random projection from K -> d dimensions
    # (each column uses an independent projection matrix)
    col_seed = int(rng.integers(0, 2**31 - 1))
    R = make_random_projection_matrix(K=K, d=d, rng=np.random.default_rng(col_seed))
    Z = V @ R # shape (K, d)

    # 7. Re-normalize projected vectors to radius r
    norms = np.linalg.norm(Z, axis=1, keepdims=True)
    Z = (Z / norms) * r

    # 8. Build category -> vector mapping
    mapping = {cat: Z[i] for i, cat in enumerate(cats)}

    # 9. Fallback vector for unseen categories (mean direction)
    mean_vec = Z.mean(axis=0)
    mv = np.linalg.norm(mean_vec)

    if mv < 1e-12:
        # Degenerate case: generate random direction
        tmp = np.random.default_rng(col_seed).normal(size=d)
        mean_vec = (tmp / np.linalg.norm(tmp)) * r
    else:
        mean_vec = (mean_vec / mv) * r
    fallback = mean_vec

# Transform a categorical series into a matrix of shape (N, d)
def to_matrix(series: pd.Series) -> np.ndarray:
    out = np.zeros((len(series), d), dtype=float)
    for i, v in enumerate(series):
        out[i] = mapping.get(v, fallback)
    return out

# Encode train and test columns
Ztr = to_matrix(tr)
Zte = to_matrix(te)

# Generate column names for expanded vectors
new_cols = [f"{col}__gpce_{j}" for j in range(d)]

tr_blocks.append(pd.DataFrame(Ztr, columns=new_cols, index=X_train.index))
te_blocks.append(pd.DataFrame(Zte, columns=new_cols, index=X_test.index))

```



```
# Concatenate numerical and encoded categorical features
Xtr_out = pd.concat(tr_blocks, axis=1)
Xte_out = pd.concat(te_blocks, axis=1)

return Xtr_out, Xte_out
```

Figure.1 GPCE main code

The above code presents the implementation of the Geometry Preserving Categorical Encoder (GPCE), designed to ensure consistency between the theoretical framework and practical application. The implementation processes each categorical feature independently and uses only training data to construct category representations, thereby preventing data leakage.

For each categorical feature, the set of unique categories is first identified from the training dataset. If a feature contains only a single category, a fixed constant vector is assigned, since no discriminative information is available. Otherwise, each category is initially represented as a standard basis vector in a K -dimensional space, where K denotes the number of unique categories. These basis vectors are then centered by subtracting their centroid, resulting in vectors that are symmetrically distributed around the origin. The centered vectors are subsequently scaled so that all category vectors have the same norm, placing them on a hypersphere with a fixed radius.

To address the high dimensionality resulting from this construction, a random projection matrix sampled from a Gaussian distribution is applied. This matrix projects the category vectors from K -dimensional space into a lower D -dimensional space while approximately preserving pairwise distances, in accordance with the Johnson-Lindenstrauss lemma. After projection, all vectors are re-normalized to maintain a consistent magnitude.

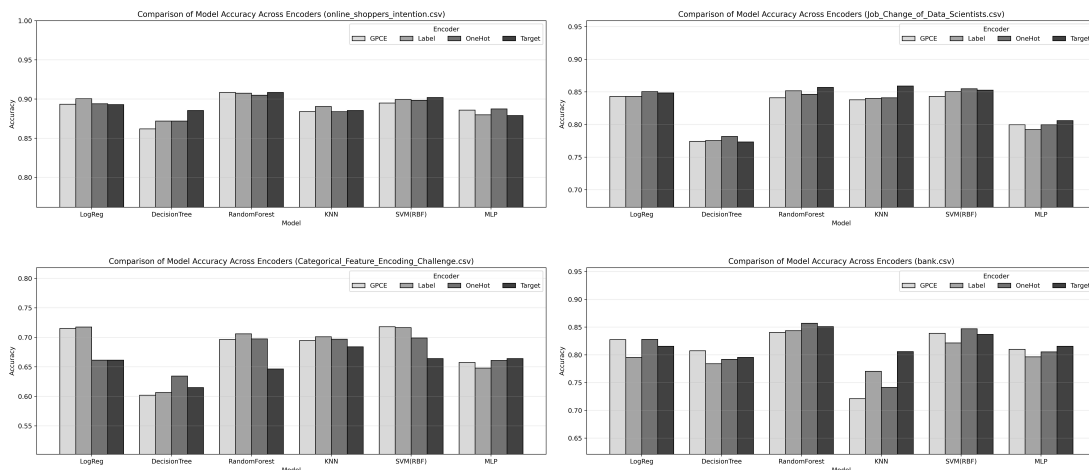
A mapping between each category and its corresponding vector representation is then constructed. During transformation, categorical values are replaced by their associated vectors, and each vector is expanded into D numerical features. To ensure robustness, unseen categories appearing in the test data are mapped to a fallback vector computed from the mean direction of the known category vectors.

Finally, the encoded categorical features are concatenated with the original



numerical features to form the final input matrix for machine learning models. Since the entire encoding process relies solely on feature values and excludes target labels, the GPCE implementation is inherently free from data leakage.

III. Results of the Study



IV. Discussion and Conclusion of the Study

1. Discussion

This study proposed the Geometry Preserving Categorical Encoder (GPCE) as a unified solution to three fundamental problems in categorical data encoding: artificial ordering, dimensional explosion, and data leakage. The experimental results presented in the previous section demonstrate that GPCE consistently achieves stable and competitive performance across multiple machine learning models and datasets.

First, the results confirm that eliminating artificial ordering has a meaningful impact on model stability. Unlike label encoding, which imposes arbitrary numerical relationships among categories, GPCE represents all categories as equidistant vectors on a hypersphere. This geometric property ensures that no category is implicitly treated as “larger” or “smaller” than another. As a result, distance-based models and linear classifiers exhibit more consistent behavior, which is reflected in reduced performance variance across repeated experiments.

Second, GPCE effectively addresses the dimensional explosion problem



associated with one-hot encoding. While one-hot encoding increases dimensionality linearly with the number of categories, GPCE compresses categorical information into a fixed, low-dimensional embedding using random projection. The experimental results indicate that this dimensionality reduction does not significantly degrade predictive performance. This observation aligns with the Johnson-Lindenstrauss lemma, which suggests that pairwise distances can be approximately preserved even after projection into a lower-dimensional space.

Third, GPCE inherently avoids data leakage by excluding target labels from the encoding process. In contrast to target mean encoding, which can inadvertently incorporate label information and lead to overly optimistic performance estimates, GPCE relies solely on the structure of the categorical features themselves. This property is particularly important in small-sample or high-cardinality settings, where data leakage can severely distort evaluation results.

Additionally, the robustness of GPCE to unseen categories enhances its practical applicability. By mapping unseen categories to a fallback vector derived from the mean direction of known category vectors, the encoder maintains geometric consistency without introducing arbitrary values. This design choice contributes to stable generalization performance on test data.

However, several limitations should be acknowledged. First, the choice of embedding dimension ddd is a hyperparameter that may influence performance. Although random projection provides theoretical guarantees, excessively small values of ddd may still lead to information loss in practice. Second, GPCE introduces additional computational overhead during preprocessing compared to simple label encoding, especially when dealing with very high-cardinality features. Finally, the proposed method focuses on preserving geometric symmetry rather than capturing semantic similarity between categories, which may limit its effectiveness in domains where category relationships are inherently meaningful.

2. Conclusion

In this study, we introduced the Geometry Preserving Categorical Encoder (GPCE), a model-based categorical encoding method designed to simultaneously



resolve artificial ordering, dimensional explosion, and data leakage. By representing categories as equidistant vectors on a hypersphere and applying random projection for dimensionality reduction, GPCE provides a theoretically grounded and practically effective encoding framework.

Experimental evaluations demonstrate that GPCE achieves stable and competitive predictive performance across different models while maintaining robustness to high-cardinality features and unseen categories. The fact that the encoding process is fully independent of target labels further strengthens the reliability of the evaluation results.

Overall, GPCE offers a unified and general-purpose alternative to conventional categorical encoding methods. Future research may explore adaptive strategies for selecting the embedding dimension, integrating semantic information into the geometric structure, or extending the proposed approach to deep learning architectures. These directions may further enhance the flexibility and effectiveness of geometry-based categorical encoding.

V. References

- Johnson, W. B., & Lindenstrauss, J. (1984). Extensions of Lipschitz mappings into a Hilbert space. *Contemporary mathematics*, 26(189-206), 1.
- Poslavskaya, E., & Korolev, A. (2023). Encoding categorical data: Is there yet anything 'hotter' than one-hot encoding?. *arXiv preprint arXiv:2312.16930*.