

# Sufficient Representations for Categorical Variables

Jonathan Johannemann  
jonjoh@stanford.edu

Vitor Hadad  
vitorh@stanford.edu

Susan Athey  
athey@stanford.edu

Stefan Wager  
swager@stanford.edu

Stanford University

## Abstract

Many learning algorithms require that categorical data be transformed into real vectors before it can be used as input. Often, categorical variables are encoded as *one-hot* or *dummy* vectors. However, this mode of representation can be wasteful since it adds many low-signal regressors, especially when the number of categorical levels is large. In this paper, we investigate simple alternative solutions for universally consistent estimators that rely on lower-dimensional real-valued categorical representations that are *sufficient* in the sense that no predictive information is lost. We then compare different methods via simulations and an empirical application.

## 1 Introduction

Many regression problems involve data collected from a number groups that may be statistically relevant. For example, in a medical setting we may want to model health outcomes using data on patients from several hospitals, and acknowledge that different hospitals may have idiosyncratic effects on patients that are not explained by other covariates. Similar considerations arise when working with data on students from different schools, voters from different zip-codes, employees at different firms, etc.

One of the most wide-spread approaches to this problem is via fixed effect modeling, as follows. Suppose that we observe  $n$  samples  $(X_i, G_i, Y_i)$  for  $i = 1, \dots, n$ , where  $X_i \in \mathbb{R}^p$  is a set of subject-specific covariates,  $G_i \in \mathcal{G}$  is a categorical variable that records group membership and  $Y_i \in \mathbb{R}$  is the respond of in interest, and that we want to estimate

$$\mu(x, g) = \mathbb{E} [Y_i \mid X_i = x, G_i = g] . \quad (1)$$

Then, the simple fixed effects approach starts by positing a model

$$\mu(x, g) = \alpha_g + x\beta, \quad (2)$$

and then estimating the coefficients  $\beta$  and  $\alpha_g$  via ordinary least squares regression. More sophisticated extensions of this approach may involve considering non-linear transformations of  $x$ , interactions between group membership and the covariates  $x$ , and/or regularization [???].

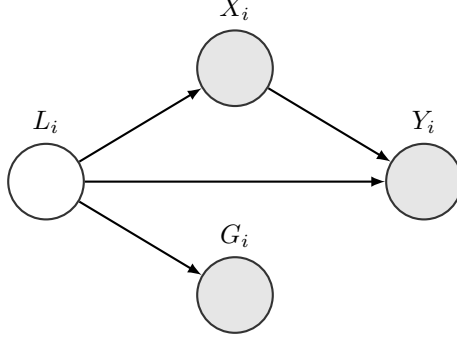


Figure 1: Causal graph depicting the key assumption that  $Y_i$  and  $X_i$  are independent of group membership  $G_i$  conditionally on latent state  $L_i$ . The grayed-out nodes are observed.

Fixed effects modeling, however, does not always perform well with complex non-linear signals or when the number of groups  $|\mathcal{G}|$  is large. The model (2) is quite rigid and may not be able to represent rich signals while; and, at the same time, the large number of  $\alpha_g$  parameters in the model (2) may result in problems for statistical inference [?]. In other words, the model (2) may have too many parameters to be stable all while lacking the degrees of freedom to fit the signal well.

The goal of this paper is to develop more parsimonious approach for representing group membership avoids the above problems. Specifically, we seek a mapping  $\psi$  that embeds group membership  $G_i$  into a  $k$ -dimensional space without losing any predictive information, i.e.,

$$\psi : \mathcal{G} \rightarrow \mathbb{R}^k, \quad \mu(x, g) = f(x, \psi(g)), \quad (3)$$

such that  $k$  is small (in particular,  $k \ll |\mathcal{G}|$ ) and the function  $f(\cdot, \cdot)$  is still easy to learn. Given such a mapping, the problem (1) becomes a routine regression problem with  $(p + k)$ -dimensional real-valued features  $(X_i, \psi(G_i))$ , and we can use out-of-the-box statistical learning software on it.

In order to obtain a useful representation of group membership  $G_i$ , we of course need to assume something about the relationship between  $G_i$  and the target outcome  $Y_i$ . The core assumption made in this paper is what we call the *sufficient latent state assumption* depicted in Figure 1: group membership  $G_i$  has no direct causal effect on  $Y_i$ , but may be associated with latent variables  $L_i$  that do have a direct effect on  $Y_i$ . For example, in the case of patients spread across hospitals, we assume that hospitals themselves do not directly *cause* health outcomes affect  $Y_i$ ; however, hospitals may still be *predictive* of  $Y_i$  through their association with latent causal variables. For example, patients may have unobserved characteristics, e.g., severity of disease or socioeconomic resources, that both affect  $Y_i$  and lead the patient to self-select into different hospitals. Our main result is that, under this sufficient latent state assumption, practical representations of the form (3) exist and can be learned from data.

The principle of representing high-cardinality categorical variables as real-valued vectors has played an important role in many different areas. For example, in natural language processing, it now common to start more complex analyses with a pre-processing step that represent words as vectors that capture the way in which the words are used in context [??]. Meanwhile, in the literature on panel data analysis, our approach is perhaps most

closely related to a proposal of ? where individual time series belong to discrete clusters and we have only one fixed effect per cluster (rather than one per time series). ? then fit this model via a  $k$ -means like algorithm that alternates clustering and estimation with per-cluster fixed effects.<sup>1</sup> The resulting framework provides a means for introducing lower dimensional, sufficient representations of categorical variables.

Our paper is structured as follows. We begin by reviewing similar problem settings in the fixed effects literature and the drawbacks of using existing methods in 1.1. In Section 2, we introduce the primary lemma which seeks to describe the true information we wish to extract from categorical variables. In Section 3, we expand on lemma 1 to develop methods that utilize this insight. In Sections 4 and 5, we run simulated and observational experiments with our proposed methods and follow up with discussion on how realized performance compared to our expectations.

## 1.1 Related Work

Traditionally, the discussion of how best to account group membership  $G_i$  in a non-parametric regression has focused on how to different ways to encode  $G_i$  in a way that can be given as an input to statistical software. One simple way to do so is via one-hot encoding:  $\iota : \mathcal{G} \rightarrow \{0, 1\}^M$  such that the  $j$ -th entry of  $\iota(g)$  is 1 if and only if  $g$  corresponds to the  $j$ -th element in  $\mathcal{G}$ , and where  $M := |\mathcal{G}|$ . Note that linear regression on one-hot encoded features  $(X_i, \iota(G_i))$  exactly recovers the standard fixed effects model (2).

As discussed above, however, one-hot encoding may lead to undesirably high-dimensional problems when  $M := |\mathcal{G}|$  is large.<sup>2</sup> In the Appendix (6.2), we present multiple encoding methods that similarly project the categories onto  $\mathbb{R}^M$ . These methods do not utilize information from the covariates  $X_i$  or response  $Y_i$  and suffer from the same pitfalls that come with high dimensional representation of the observed groups  $\mathcal{G}$ . The primary difference for these methods are the user’s interpretation of the encoded variables which are commonly constructed as the comparison of the mean effect of a subset  $\mathcal{G}' \in \mathcal{G}$  relative to the mean effect of the set  $\mathcal{G} \setminus \mathcal{G}'$  or one of its subsets.

The problem of fixed effects is especially challenging with sparsity-seeking methods such as the lasso [?] or decision trees [?], and related ensemble methods such as random forests [?] or gradient-boosted trees [?]. Sparsity-seeking methods will set the contribution of features to zero unless there is strong evidence that the features matters for prediction, and it is difficult for rare levels of  $G_i$  to produce sufficient evidence to get a non-zero contribution to the model via their one-hot features. The end result is that sparsity seeking methods may largely ignore high-cardinality one-hot encoded factors.

Another prevalent way of working with categorical variables with decision trees is to consider full factorial splits that allow for arbitrary grouping of the levels of the categorical variable. For a variable with  $M := |\mathcal{G}|$  levels, this allows for  $2^M - 1$  potential splits. ? showed that we can optimize over this exponential set of potential splits in time that scales

<sup>1</sup>Our approach is not directly comparable to either of these methods, as we do not focus on textual data, and do not assume that the latent state  $L_i$  can be consistently estimated (in contrast, ? assume that they have access to long enough time series that their clustering step is consistent which, in our setting, would be equivalent to assuming that  $L_i$  can be recovered).

<sup>2</sup>Another slightly more subtle difficulty is that when the categorical variable has many levels, the individual features  $\iota(G_i)_j$  become very sparse (i.e., they are usually 0 and only very rarely 1). Many approaches to statistical learning work better with features whose variance roughly captures their range than with such spiky features.

linearly in  $M$ ; however, from a statistical point of view, such factorial splits are prone to very strong overfitting when the number of levels is large.

Recently, ? consider a related problem of representing “dirty” categorical variables that might arise if, e.g., several categorical levels are just misspellings of each other, and propose using a low-dimensional embedding that exploits lexicographic similarity (i.e., factors with similar spellings are arranged close to each other). In this paper, we use information in the  $X_i$ , rather than lexicographic information, to construct an embedding; however, the high-level conclusion that we can achieve meaningful gains by using auxiliary information to embed categorical variables in a low-rank space remains.

We also note, it is sometimes possible to achieve strong results by randomly projecting a one-hot representation of the categorical variables into  $\mathbb{R}^k$  for relatively small  $k$  [?]. We also consider this approach but find that, at least in our experiments, we can achieve better performance using carefully crafted representations that leverage continuous covariates.

## 2 Representing Groups with Sufficient Latent State

Our *sufficient latent state* assumption presented in the introduction and depicted in the causal graph 1 implies that the distribution of the outcome  $Y_i$  only depends on the observable factor  $G_i$  through some unobservable latent variable  $L_i \in \{\text{good}, \text{poor}\}$ . In other words, if we knew the value of  $L_i$ , then also knowing  $G_i$  would give us no additional information about the outcome. For a simple example, one may posit that a patient’s underlying health status ( $L_i \in \{\text{good}, \text{poor}\}$ ) may simultaneously determine to which hospital they are admitted ( $G_i$ ), what symptoms ( $X_i$ ) they exhibit, and what health outcomes ( $Y_i$ ) they attain. Conditioned on the underlying health status, the hospital cannot provide any additional information about any of the other variables. Conversely, learning their hospital is only helpful inasmuch it allows us to infer something about their health status.

The following lemma states that we can say further characterize *how* the information about the categorical variable  $G_i$  enter the model: the conditional expectation function of the outcome depends only on the *conditional probabilities of the latent variable given the observable category*. This fact will be crucial when deriving the representation methods in future sections.

**Lemma 1.** *Suppose that the latent state  $L_i$  is discrete with  $k$  possible levels, and that the probabilistic structure require by the sufficient latent state assumption (Figure 1) holds. Then,*

$$\psi : \mathcal{G} \rightarrow \mathbb{R}^k, \quad \psi_l(g) = \mathbb{P} [L_i = l \mid G_i = g] \quad (4)$$

*provides a sufficient representation of  $G_i$  in the sense of (3):*

$$\mu(x, g) = \frac{\sum_{l=1}^k \mathbb{E} [Y_i \mid X_i = x, L_i = l] \mathbb{P} [X_i = x \mid L_i = l] \psi_l(g)}{\sum_{l=1}^k \mathbb{P} [X_i = x \mid L_i = l] \psi_l(g)}. \quad (5)$$

Expression (5) formalizes the intuition laid out in the previous paragraph. The information associated with the category only enters the conditional expectation via the set of probabilities  $\mathbb{P} [L_i = \ell \mid G_i = g]$ . If there are only  $k$  latent groups, then each category can be represented in a lossless manner by a  $k$  dimensional vector of probabilities. An immediate consequence of this result is that if we knew  $\psi$  and gave training examples  $((X_i, \psi(G_i)), Y_i)$  to any universally consistent learner, the learner would eventually recover the optimal prediction function  $\mu(\cdot)$ . To continue the example at the top of this section, the identity of the

hospital enters the model through the probability that a patient is in good or poor health given the hospital.

The dependence of the conditional expectation function  $\mu$  on the latent variable probabilities  $\psi$  via (5) is non-linear; however, we will retain consistency if we use an expressive enough method for learning on  $((X_i, \psi(G_i)), Y_i)$ . Methods known to be universally consistent include  $k$ -nearest neighbors [?], various tree-based ensembles [?], and neural networks [?].

The discussion above seems to imply that we need to estimate  $\psi(g) = \mathbb{P}[L_i|G_i = g]$  directly. However, because this quantity depends on the unobservable variable  $L_i$ , its identification is impossible without further assumptions and a more sophisticated approach. Instead we pursue a simpler approach by seeking different functions  $f(g)$  that depend only on observables (such as  $f(g) = E[X_i|G_i = g]$ ), and then proving that they are also sufficient representations because they can be written as invertible functions of  $\psi(g)$

### 3 Categorical variable encoding methods

Our methods proposed below take the form of removing the categorical column and replacing it with a set of columns that can be proven to encode all the categorical information. All of them exploiting the structure mentioned in the previous section. For an overview of other categorical encoding methods already in use, please see section 6.2 in the Appendix.

#### 3.1 Means encoding

For our first method, we drop the categorical variables  $G_i$  and substitute in the average value of the continuous regressors  $X_i$  given the categorical variable. Figure 2 shows an illustration.

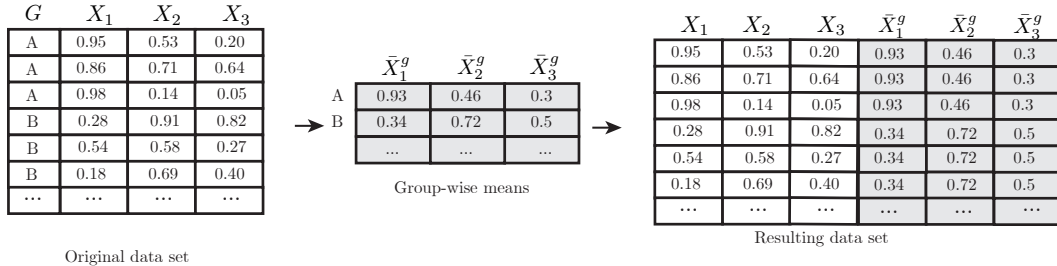


Figure 2: Implementation example of the *means* encoding.

This representation is easily interpretable, and it is simple to implement efficiently. This method may be particularly applicable in instances where the number of regressors  $p$  is small relative to the number of categories  $p \ll |\mathcal{G}|$ , since then the dimensionality reduction is more dramatic as compared to traditional encoding methods such as one-hot encoding. Figure 3 provides an intuitive explanation for why we should expect this to work: the group-wise averages of the continuous variables ( $X_1, X_2$ ) may reveal the dominant latent group in each category.

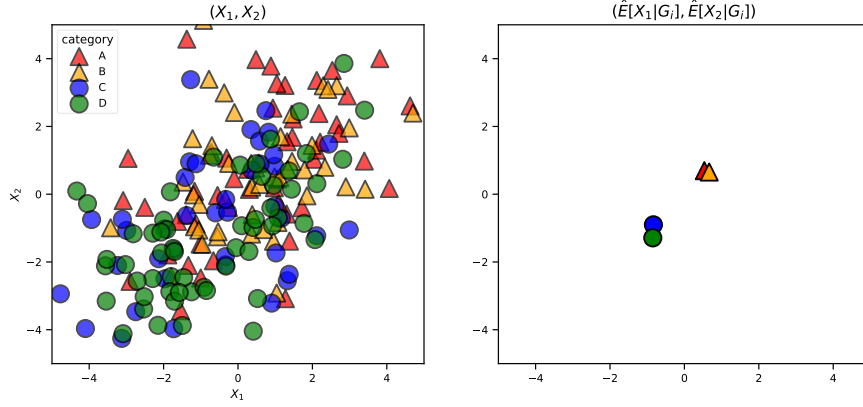


Figure 3: Intuition for the *means* encoding on an illustrative data. Here, categories  $(A, B)$  and  $(C, D)$  are more strongly associated with different latent groups.

The next lemma presents the conditions in which this representation provides a sufficient representation. All proofs are in the appendix.

**Lemma 2.** *Under the conditions of Lemma 1, suppose in addition that the matrix  $A$  defined by  $(A)_{tj} := \mathbb{E}[X_{it} | L_i = g]$  is left-invertible. Then, the  $p$ -dimensional vectors  $\omega(g) := \mathbb{E}[X_i | G_i = g]$  are sufficient representations of each category in the sense of (3):*

(6)

---

**Algorithm 1** Means Encoding Method

---

```

1: procedure GROUPAVERAGES( $X, G$ )
2:    $\hat{\Omega} \leftarrow 0_{M \times p}$  ▷ Compute group-wise averages of continuous covariates
3:   for  $g$  in  $1:M$  do
4:      $\hat{\Omega}_{\cdot, g} \leftarrow \frac{1}{|\{i: G_i = g\}|} \sum_{i: G_i = g} X_i$ 
5:   return  $\hat{\Omega}$ 
6:
7: procedure MEANSENCODING( $X, G$ )
8:    $\hat{\Omega} \leftarrow \text{GROUPAVERAGES}(X, G)$ 
9:    $S \leftarrow 0_{n \times p}$ 
10:  for  $i$  in  $1:n$  do ▷ Populate with group averages
11:     $S_{i, \cdot} \leftarrow \hat{\Omega}_{\cdot, G_i}$ 
12:  return  $S$ 

```

---

### 3.2 Low-rank encodings

The *means* encoding method may efficiently summarize the effect of the categorical variables if the continuous covariates are reasonably low-dimensional so that  $p \ll M$ . When  $p$  is large, it might be beneficial to use lower-dimensional representation of the conditional means. We

suggest two *low-rank encoding methos*, both involving applying factorization methods to the matrix of  $\Omega$  of group-wise means of the continuous variables  $(\Omega)_{jg} = E[X_{ij}|G = g]$ .

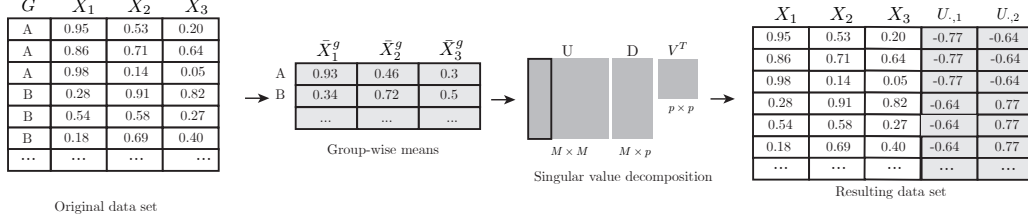


Figure 4: Implementation example of the *low-rank* encoding with singular value decomposition. Alternatively, we could also have used sparse PCA in place of SVD.

One alternative is to consider the factorization of the group-wise means matrix  $\Omega$  using singular value decomposition  $\Omega = UDV^T$ . Then we can use the first  $k$  columns of the  $g^{th}$  row of the left-singular vector matrix  $U$  as the representation for the  $g^{th}$  category. Note that in practice, we will be working with the empirical counterpart  $\hat{\Omega}$  and  $k$  is in general unknown, so we recommend using cross-validation. Figure 5 provides an illustration.

A second low-rank alternative is to use the sparse principal component analysis (PCA) method of ?. As the name suggests, this method extends the original PCA algorithm by applying a elastic-net-style penalty on the coefficients of the loadings matrix. The result is that the matrix  $\Omega$  is approximated by a sparse linear combination of vectors.<sup>3</sup> This sort of sparsity can be advantageous for two reasons. First, sparse PCA is deemed to be produce more interpretable results. Second, if our universal estimator is a tree-based models such as random forest ? or xgboost ?, which fit to data by considering singular covariates at any point in the model, it may have difficulty taking advantage of dense principal components due to the rotation of the original covariate space. That is, if a tree would have a been able to produce a good split by using each variable separately, it may not be able to do the same if the variables are combined linearly. On the other hand, sparse PCA requires additional tuning of the  $\lambda$  parameters via cross-validation.

Lemma 3 shows that if indeed there are only  $k$  latent groups, then a representation that uses only the first  $k$  columns of the left singular matrix or the first  $k$  sparse principal components is indeed sufficient. As with the *means* method, we rely on the universal consistency property of our estimator to learn a nonlinear mapping.

**Lemma 3.** *Under the conditions of Lemma 1, suppose in addition that the matrix  $A$  defined by  $(A)_{tj} := \mathbb{E}[X_{it} | L_i = g]$  is left-invertible. Then, the  $k$ -dimensional vectors  $\omega(g) := \mathbb{E}[X_i | G_i = g]$  are sufficient representations of each category in the sense of (3):*

$$\mu(x, g) = \frac{\sum_{l=1}^k \mathbb{E}[Y_i | X_i = x, L_i = l] \mathbb{P}[X_i = x | L_i = l] (A^\dagger \omega(g))_l}{\sum_{l=1}^k \mathbb{P}[X_i = x | L_i = l] (A^\dagger \omega(g))_l} \quad (9)$$

<sup>3</sup>Formally, for a given matrix  $M$ , the sparse PCA method solves the problem [?, eq. 3.12],

$$(\hat{A}, \hat{B}) = \arg \min_{A, B} \sum_{i=1}^n \|M_i - AB^T M_i\|^2 + \lambda \sum_{j=1}^k \|B_{:,j}\|_2^2 + \sum_{j=1}^k \lambda_{1,j} \|B_{:,j}\|_1 \quad (7)$$

$$\text{s.t. } A^T A = I_{k \times k} \quad (8)$$

---

**Algorithm 2** Low Rank Encoding Method

---

```

1: procedure LOWRANKENCODING( $X, G, k$ )
2:    $\hat{\Omega} \leftarrow \text{GROUPAVERAGES}(X, G)$ 
3:    $U, D, V^T \leftarrow \text{SVD}(\hat{\Omega})$  ▷ Singular value decomposition
4:    $S \leftarrow 0_{n \times k}$ 
5:   for  $i$  in  $1:n$  do ▷ Populate with left singular matrix truncated rows
6:      $S_{i,\cdot} \leftarrow U_{G_i,1:k}$ 
7:   return  $S$ 

```

---



---

**Algorithm 3** Sparse Low Rank Encoding Method

---

```

1: procedure SPARSELOWRANKENCODING( $X, G, k$ )
2:    $\hat{\Omega} \leftarrow \text{GROUPAVERAGES}(X, G)$ 
3:    $A, B \leftarrow \text{SPCA}(\hat{\Omega})$  ▷ Sparse principal component analysis
4:    $Z \leftarrow M \cdot B_{:,1:k}$  ▷ Projection on truncated principal components
5:    $S \leftarrow 0_{n \times k}$ 
6:   for  $i$  in  $1:n$  do ▷ Populate with sparse principal components rows
7:      $S_{i,\cdot} \leftarrow Z_{G_i,\cdot}$ 
8:   return  $S$ 

```

---

### 3.3 Encoding by multinomial logistic regression coefficients

Finally, we propose estimating the conditional probability of category membership by multinomial logistic regression parametrized by coefficients  $\{\theta_g\}_{g \in \mathcal{G}}$

$$P(G_i | X_i) = \Lambda_\theta(G_i = g | X_i) = \frac{\exp(X_i^T \theta_g)}{\sum_{g'} \exp(X_i^T \theta_{g'})} \quad (10)$$

and then use the  $p$ -dimensional vector of coefficients  $\theta_g$  associated with  $g^{th}$  category to represent it. The motivation for this method comes from the fact that the prediction model  $\mu(x, g)$  can be rewritten so that it only depends on the category  $g$  through  $P(G_i = g | X_i = x)$ , and under the multinomial logistic regression assumption above this boils down to dependence on the  $\theta_g$  coefficients.

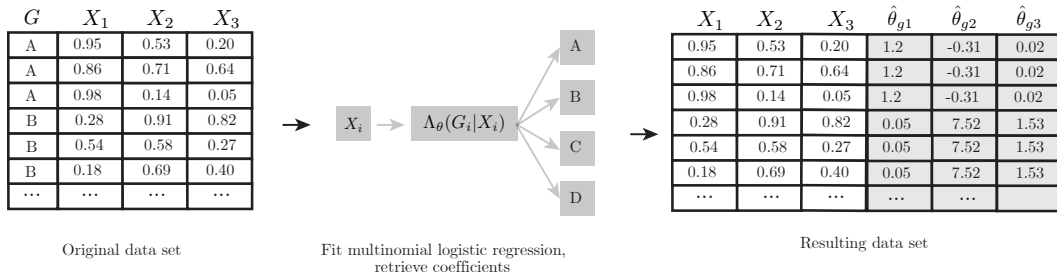


Figure 5: Implementation example of the *mnl* encoding.



A related work that also uses regression coefficients as categorical representations is the natural language processing *word2vec* model of ?. In that work, the authors propose two methods to represent words (categories) in a large corpus as relatively low-dimensional real-valued vectors. In one of these methods, each word is initially assigned two representations: as a center word  $v_w$ , and as a surrounding *context* word  $v_c$ . Then, the authors posit that the optimal representation is that maximizes the log-probability of the inner product of the two representations  $v_w^T v_c$  for all pair of words  $(w, c)$  that co-occur near each other. Our method works in an analogous way if we consider the continuous vectors  $X_i$  stand in as “contexts”, and let  $\theta_g \in \mathbb{R}^p$  represent each category, since then maximizing the log-probability of the inner product  $X_i^T \theta_g$  is the same as maximizing the multinomial logistic regression above.

**Lemma 4.** *Under the conditions of Lemma 1, suppose in addition that  $A$  in (22) is left-invertible, and that  $\mathbb{P}[G_i = g | X_i]$  is the multinomial logit distribution with coefficients  $\{\theta_g\}_{g \in \mathcal{G}}$  containing an intercept. Then, the vector  $\theta_g \in \mathbb{R}^p$  is sufficient in the sense of (3):*

$$\mu(x, g) = \frac{\sum_{l=1}^k \mathbb{E}[Y_i | X_i = x, L_i = l] \mathbb{P}[X_i = x | L_i = l] (A^\dagger f(\theta_g))_l}{\sum_{l=1}^k \mathbb{P}[X_i = x | L_i = l] (A^\dagger f(\theta_g))_l} \quad (11)$$

$$\text{where } f(\theta_g) := \frac{\mathbb{E}_X[X_i \Lambda_\theta(g | X_i)]}{\mathbb{E}_X[\Lambda_\theta(g | X_i)]} \quad (12)$$

---

**Algorithm 4** Multinomial logistic regression method (MNL)

---

```

1: procedure MNL( $X, G$ )
2:    $\hat{\theta} \leftarrow \arg \min_{\theta} \sum_i \log \Lambda_{\theta}(G_i | X_i)$  ▷ Multinomial logistic regression
3:    $S \leftarrow 0_{n \times p}$ 
4:   for  $i$  in  $1:n$  do ▷ Populate with left singular matrix truncated rows
5:      $S_{i,:} \leftarrow \hat{\theta}_{G_i}$ 
6:   return  $S$ 

```

---

## 4 Experiments

In the following section, we explore each method’s effectiveness relative to one hot encoding across simulated and real world data sets. We apply two typically used methods random forests and xgboost.<sup>4</sup>

### 4.1 Simulations

We consider two simulations designs that share the distributions of latent groups  $L_i$ , observable groups  $G_i$  and covariates  $X_i$ , but whose outcome models for  $Y_i$  differ.

**Latent groups, observable groups and continuous covariates** Latent groups  $L_i$  is drawn uniformly from the set of available groups, which we identify with integers.

$$L_i \sim \text{Uniform}(\{1, \dots, |\mathcal{L}|\}) \quad (13)$$

---

<sup>4</sup>Simulation code can be found at: [repo url](#).

Next, observable groups  $G_i$  are drawn according to the following rule. First, we partition the set of possible observable groups  $\mathbb{G}$  into equally-sized sets  $\{\mathbb{G}_\ell\}_{\ell=1}^{|\mathcal{L}|}$ . Then, we draw the observable group  $G_i$  so that observations that were assigned latent group  $L_1$  have higher probability of falling into observable group  $\mathbb{G}_1$ , those in  $L_2$  likely belong to  $\mathbb{G}_2$ , and so on. In symbols,

$$P(G_i = g \mid L_i) = \begin{cases} \frac{p_{L_i}}{|\mathbb{G}_{L_i}|} & \text{if } g \in \mathbb{G}_{L_i} \\ \frac{1-p_{L_i}}{|\mathbb{G}_{L_i}^c|} & \text{otherwise} \end{cases} \quad \text{where } p_{L_i} > 0.5 \quad (14)$$

Covariates associated with latent group  $L_i = \ell$  are normally distributed as  $X_i \sim \mathcal{N}(\mu_\ell, \Sigma)$ . The mean is zero except for a randomly drawn set of entries  $\mathcal{J}$  that are  $-1, +1$ , with  $|\mathcal{J}| = 3$ .

$$(\mu_\ell)_j = \begin{cases} 0 & \text{if } j \in \mathcal{J} \\ \text{Uniform}(\{-1, 1\}) & \text{otherwise} \end{cases} \quad (\Sigma)_{kj} = \left(\frac{1}{2}\right)^{|k-j|} \quad (15)$$

**Outcomes** In the *latent linear* setup, the outcome model is linear in regressors conditional on coefficients that are specific to each latent group.

$$Y_i = \alpha_\ell + X_i^T \beta_\ell + \epsilon_i \quad (16)$$

where the intercept and slopes are created as follows. The slope normalization ensures that the signal from the intercept, regressors and noise is roughly comparable.

$$\alpha_\ell \sim \text{Laplace}(1) \quad \text{for each } \ell \in \mathcal{L} \quad (17)$$

$$\tilde{\beta}_{\ell j} \sim \text{Uniform}(\{0, 1, -1\}) \quad \beta_\ell = \frac{\tilde{\beta}_\ell}{\|\tilde{\beta}_\ell\|_2} \quad (18)$$

$$\epsilon_i \sim \mathcal{N}(0, 1) \quad (19)$$

In the *global linear* setup, only the intercept changes and the slopes are kept fixed across latent groups, so  $\beta_\ell = \beta_{\ell'}$  for all  $\ell, \ell'$ .

In the *hermite* setup, we compute a third-order Hermite polynomial basis  $\phi(X_i)$ , then select a subset  $\mathcal{J}'$  of “active” regressors, with  $|\mathcal{J}'| = p$ . The outcome is linear in this sparse basis.

$$Y_i = \sum_{j \in \mathcal{J}'} \phi_j(X_i)^T \beta_{\ell j} + \epsilon_i \quad \text{where } \beta_{\ell j} \sim \text{Uniform}\left(\left\{-\frac{1}{\sqrt{p}}, +\frac{1}{\sqrt{p}}\right\}\right) \quad (20)$$

As in the *latent linear* setup, the slopes are specific for each latent group.

## 4.2 Simulation Results

We simulated one hundred datasets for each combination of parameters on [add something here](#). For each simulated dataset, we estimated the outcome using the various methods described in Section 3, and then evaluated the predictions by their mean squared error.

Results for the simulations are provided in 7. We find that the methods described above which seek to estimate the latent groups consistently outperform methods which require

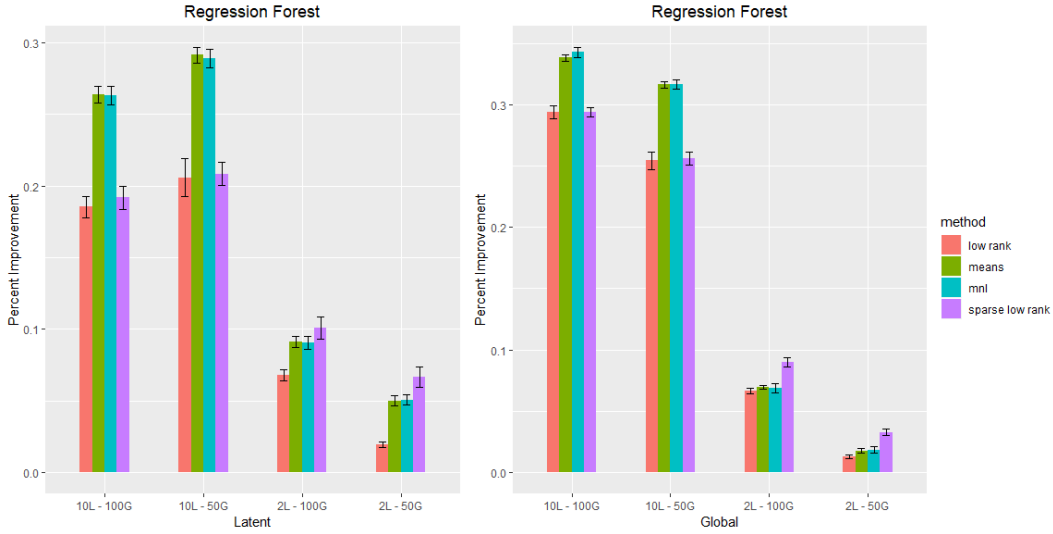


Figure 6: Percent Improvement over One Hot Encoding for Regression Forests.

adding  $|\mathcal{G}|$  additional columns to our input matrix  $X$  for both the Regression Forest and XGBoost. In particular, it appears that the sparse low rank approach tends to do well when the number of latent groups is very small. For a larger number of latent groups, we see that low rank approaches underperform and the multinomial and means encoding perform better. We also take note on how the multinomial weight approach does well in this case possibly because  $n$  is large and the number of observations per group is high enough to satisfy this data hungry approach.

For the methods that do take advantage of the low rank structure, we notice that the main improvement in performance for Regression Forests and XGBoost occurs due to the immense reduction in dimensionality. We find that the permutation, fisher, and multiple permutation methods are on average much better than the methods that add  $|\mathcal{G}|$  columns but still slightly fall behind the methods that estimate the latent groups.

While the performance improvements over one hot encoding for 2 latent groups ranges from 1-10%, performance improvement can approach 27-33% for 10 latent groups. Intuitively, we find that this benefit is generally less prevalent for 2 latent groups for Regression Forests and XGBoost due to the lesser complexity of the underlying relationship as defined by the conditional independence graph in 1. Furthermore, we find that a more expressive method like XGBoost benefits slightly less from these encoding methods.

### 4.3 Empirical Applications

We also evaluate these methods on publicly available datasets that are accessible on Kaggle. We run 4 fold, stratified cross validation on the datasets to avoid the case where there are categorical variables in the test set which are not contained in the training set. Since we are throwing out what could potentially be a sizable amount of information with each fold, we also conduct a paired t-test to further validate or deny results seen in the cross validation process.

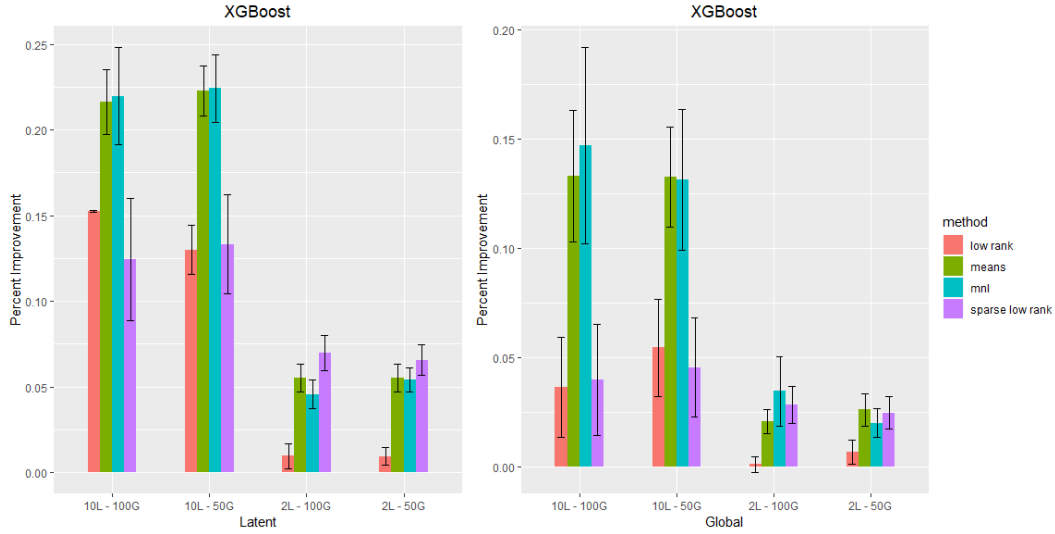


Figure 7: Percent Improvement over One Hot Encoding for XGBoost.

**Pakistan Educational Performance** In ?, Hemani consolidated a series of surveys from Alif Ailaan, a nonprofit organization in Pakistan that focuses on improving education across the country. The objective of the surveys was to provide an objective means of comparing school systems across cities and provinces in order to spark competition between local governments to spur educational reform.

The dataset used in the analysis below contains  $n = 580$  and 504 after removing null valued rows which can be broken down into  $|\mathcal{G}| = 127$  cities from 2013 to 2016. The number of additional covariates  $p$  is 20 and our estimation methodology is further elaborated on in Appendix 6.3.

**Ames Housing** The objective of the Ames Housing Dataset ? was to act as a more complex alternative to the Boston Housing Dataset ?. De Cock’s aim was to use this dataset for the final project in his regression course which would allow students to more extensively showcase what they had learned.

The Ames Housing Dataset has  $n = 2,930$  individual home sales in Ames, Iowa from 2006 to 2010. The dataset has 80 covariates and our categorical variable “neighborhood” has  $|\mathcal{G}| = 25$ .

**King County House Sales** The King County House Sales Dataset from ? is a data set containing the record of 21,613 home sales in King County, Washington between May 2014 to May 2015. The author does not provide much additional information aside from it being a good dataset to test regression models. The dataset is relatively popular with over 169,000 views and 28,000 downloads at the time of this paper.

The data itself came with 21 covariates including the sale price of the house. We treat the “zipcode” covariate, which has  $|\mathcal{G}| = 70$ , as the categorical variable.

#### 4.4 Empirical Results

We can see that for Regression Forests on average there is an improvement over one hot encoding and XGBoost stands to benefit less from using these encoding methods over one hot encoding. For Regression Forests, the primary case that does not benefit much from these approaches is the Ames data set which follows naturally since the number of covariates  $p$  is much larger than the number of observed groups  $|\mathcal{G}|$ . Therefore, methods such as means and MNL are adding 80 dimensions to the prediction problem while one hot encoding only adds 25. The Low Rank and Sparse Low Rank approaches benefit in these cases and appear to maintain potentially promising results. Contrary to the Regression Forest results, most of the output was not statistically significantly different than one hot encoding and the Ames data set was the closest evidence to any benefit.

Dataset	Metric	Means	Low Rank	Sparse Low Rank	MNL
Pakistan	MSE	9.963	8.228	8.868	8.656
Pakistan	p-val	0.00402	0.04333	0.00089	0.01132
Ames	MSE	1.349	1.798	3.987	-2.120
Ames	p-val	0.73221	0.00930	0.06932	0.81650
Kingcounty	MSE	8.405	8.671	7.062	8.054
Kingcounty	p-val	0.00445	0.01267	0.03102	0.00364

Table 1: Observational Dataset Results for Regression Forests.

For Regression Forests, on average, it looks like low rank approaches to generating encodings were most robust across data sets. This could be due to the reduction in dimensionality which may be beneficial for two reasons. First, the underlying relationships were much lower rank than the number of covariates and these methods were able to capture this information. Second, if there was no signal in the categorical variable to begin with, the low rank approaches which utilize K-fold to determine the dimensionality of the encoding are able to pick small  $K$  to reduce the potential noise covariates one would be adding.

Dataset	Metric	Means	Low Rank	Sparse Low Rank	MNL
Pakistan	MSE	2.904	0.668	2.528	-3.391
Pakistan	p-val	0.52714	0.88127	0.29955	0.23304
Ames	MSE	7.382	9.736	14.889	1.890
Ames	p-val	0.31597	0.07341	0.13348	0.59210
Kingcounty	MSE	0.773	-3.243	2.468	-0.471
Kingcounty	p-val	0.67990	0.62293	0.49389	0.87640

Table 2: Observational Dataset Results for XGBoost.

## 5 Conclusion

In this paper, we explore the task of mapping high-cardinality categorical variables  $G_i$  to a lower-dimensional real space without loss of information relevant to our response  $Y_i$ . To do this, we make an assumption about the relationship between  $G_i$  and  $Y_i$  which we call the *sufficient latent state assumption*. This assumption provides us with the basis for creating encoding methods which can be used by universally consistent estimators to extract

sufficient representations of  $G_i$ . Among our recommendations for encoding methods, we provide encoding methods which are interpretable or focus more on reducing the size of the  $\mathbb{R}^k$  representation. We find that these methods tend to outperform one hot encoding and other traditional approaches to modeling with categorical variables as the number of categorical levels increases.

## 6 Appendix

### 6.1 Proofs

**Definitions** The following matrices that will be used below.

$$\Omega = \begin{bmatrix} \mathbb{E}[X_1 | G = g_1] & \cdots & \mathbb{E}[X_1 | G = g_M] \\ \vdots & \ddots & \vdots \\ \mathbb{E}[X_p | G = g_1] & \cdots & \mathbb{E}[X_p | G = g_M] \end{bmatrix}_{p \times M} \quad (21)$$

$$A = \begin{bmatrix} \mathbb{E}[X_1 | L = l_1] & \cdots & \mathbb{E}[X_1 | L = l_K] \\ \vdots & \ddots & \vdots \\ \mathbb{E}[X_p | L = l_1] & \cdots & \mathbb{E}[X_p | L = l_K] \end{bmatrix}_{p \times K} \quad (22)$$

$$\Psi = \begin{bmatrix} \mathbb{P}[L = l_1 | G = g_1] & \cdots & \mathbb{P}[L = l_1 | G = g_M] \\ \vdots & \ddots & \vdots \\ \mathbb{P}[L = l_K | G = g_1] & \cdots & \mathbb{P}[L = l_K | G = g_M] \end{bmatrix}_{K \times M} \quad (23)$$

We denote the columns of  $\Omega$  as  $\omega(g)$  and the columns of  $\Psi$  as  $\psi(g)$ .

**Overview** Proof 6.1.1 shows that categories  $G_i$  only enter the conditional expectation function  $\mu(x, g)$  through the latent state probabilities  $\psi(g) = \mathbb{P}[L_i | G_i = g]$ . Proofs 6.1.2-6.1.3 rely on strategies for writing  $\psi(g) = f(h(g))$  then showing that  $h(g)$  is also a sufficient representation.

#### 6.1.1 Proof of Lemma 1

*Proof.* To show the equivalence of (1) and (5), we begin by expanding (1) as

$$\mu(x, g) = \sum_{l=1}^L \mathbb{E}[Y_i | X_i = x, G_i = g, L_i = l] \mathbb{P}[L_i = l | X_i = x, G_i = g] \quad (24)$$

Now, the conditional independence assumptions encoded in our graph imply that the expectation term simplifies to

$$\mathbb{E}[Y_i | X_i = x, G_i = g, L_i = l] = \mathbb{E}[Y_i | X_i = x, L_i = l] \quad (25)$$

while the second term can be rewritten using Bayes rule as

$$\mathbb{P}[L_i = l | X_i = x, G_i = g] = \frac{\mathbb{P}[X_i = x | L_i = l] \mathbb{P}[L = l | G_i = g]}{\sum_{l'=1}^k \mathbb{P}[X_i = x | L_i = l'] \mathbb{P}[L = l' | G_i = g]} \quad (26)$$

Combining the above, we see that the mapping  $\mu$  only depends on the categorical variable through the multivariable function  $\psi(g) = \mathbb{P}[L_i | G_i = g]$ . Therefore,  $\psi(g)$  is a sufficient representation as defined in (3).  $\square$

### 6.1.2 Proof of Lemma 2

*Proof.* Begin by noting that conditionl expectations can be computed as a linear combination of the sufficient statistics discussed in Lemma 1.

$$\mathbb{E} [X_i \mid G_i = g] = \sum_{l=1}^K \mathbb{E} [X_i \mid L_i = l] \mathbb{P} [L_i = l \mid G_i = g] \quad (27)$$

$$= \sum_{l=1}^K \mathbb{E} [X_i \mid L_i = l] \psi_l(g) \quad (28)$$

or, in matrix form,

$$\Omega = A\Psi \quad (29)$$

where these matrices are defined as in the top of this section. The sufficient representation for the category  $\psi(g) = \Psi_g$  lies on the linear span of the set of columns of  $A$ . Since  $A$  has a left-inverse  $A^\dagger$  such that  $A^\dagger A = I$ , we can retrieve the representations by matrix multiplication.

$$\psi(g) = (\Psi)_{\cdot, g} = A^\dagger (\Omega)_{\cdot, g} =: A^\dagger \omega(g) \quad (30)$$

Since  $\psi(g)$  only depends on  $g$  through  $\omega(g)$ , it follows that  $\omega(g)$  is also a sufficient representation for the category  $g$ .  $\square$

### 6.1.3 Proof of Lemma 4

*Proof.* We begin by noting that we can use Bayes' theorem to express  $\omega(g) = \mathbb{E} [X_i \mid G = g]$  as a function of  $\mathbb{P} [G = g \mid X_i]$ , here is assumed to be multinomial logit.

$$\mathbb{E}_{X|G} [X_i \mid G_i = g] = \mathbb{E}_X [X_i \mathbb{P} [X_i \mid G_i = g]] \quad (31)$$

$$= \frac{\mathbb{E}_X [X_i \mathbb{P} [G_i = g \mid X_i]]}{\mathbb{P} [G_i = g]} \quad (32)$$

$$= \frac{\mathbb{E}_X [X_i \mathbb{P} [G_i = g \mid X_i]]}{\mathbb{E}_X [\mathbb{P} [G_i = g \mid X_i]]} \quad (33)$$

$$= \frac{\mathbb{E}_X [X_i \Lambda_\theta(g \mid X_i)]}{\mathbb{E}_X [\Lambda_\theta(g \mid X_i)]} \quad (34)$$

However, note that expression (34) only depends on the category through the multinomial logit coefficients  $\theta_g$  that are associate with category  $g$ . Therefore, under this assumption we can write  $\omega(g) = f(\theta_g) =: E[X_i \mid G = g]$ . However, recall from (30) that if the matrix  $(A)_{j\ell} := E[X_{ij} \mid L_i = \ell]$  has a left-inverse  $A^\dagger A = I$ , we can write

$$\psi(g) = A^\dagger \omega(g) = A^\dagger f(\theta_g) \quad (35)$$

Since  $\psi(g)$  only depends on  $g$  through  $\theta(g)$ , it follows that  $\theta(g)$  is also a sufficient representation for the category  $g$ .  $\square$



## 6.2 Additional Encoding Methods

For a more in-depth treatment, see ?. Note that several of the methods below are simple linear transformations of each other and should yield equivalent levels of performance in theory. However, as we will see in sections 4.2 and 4.3, in practice the resulting performance can differ substantially.

**One-hot or dummy** This is the most common categorical encoding, and it is the method we take to be our main baseline, against which we will compare all other methods. It expands out the categorical column into  $k - 1$  columns where  $k$  is the number of unique elements in the set of categorical levels in the column. Each column is binary 1 or 0 depending on whether the corresponding level was observed in the original categorical column. [?, sec 2.3.2]

	b	c	d	e
a	0	0	0	0
b	1	0	0	0
c	0	1	0	0
d	0	0	1	0
e	0	0	0	1

**Deviation** Similar to one-hot encoding except that the  $k^{th}$  unique element's row that is the reference level is now set to all values of  $-1$ . This means that categorical levels are being compared to the grand mean of all of the levels instead of the mean of a given level with respect to the reference level.

	b	c	d	e
a	1	0	0	0
b	0	1	0	0
c	0	0	1	0
d	0	0	0	1
e	-1	-1	-1	-1

**Difference** Compares a given level to the mean of the levels that precede it.

	b	c	d	e
a	-0.5	-0.333	-0.25	-0.2
b	0.5	-0.333	-0.25	-0.2
c	0.0	0.667	-0.25	-0.2
d	0.0	0.000	0.75	-0.2
e	0.0	0.000	0.00	0.8

**Helmert** Compares levels of a chosen categorical variable to the mean of the subsequent levels uniquely observed thus far.

	b	c	d	e
a	0.80	0.00	0.00	0.00
b	-0.20	0.75	0.00	0.00
c	-0.20	-0.25	0.67	0.00
d	-0.20	-0.25	-0.33	0.50
e	-0.20	-0.25	-0.33	-0.50

**Repeated Effect** Columns are encoded to represent a cumulative comparison of subsequent levels with previous ones.

	b	c	d	e
a	0.8	0.6	0.4	0.2
b	-0.2	0.6	0.4	0.2
c	-0.2	-0.4	0.4	0.2
d	-0.2	-0.4	-0.6	0.2
e	-0.2	-0.4	-0.6	-0.8

**Permutation** Assigns a unique integer to each category. Note that even when the categories do not possess an intrinsic ordering, some mappings may yield better results if they happen to be aligned with the true average effect the category has on the outcome variable.

	perm
a	5
b	3
c	4
d	1
e	2

**Multi-Permutation (Multi-Perm)** Following the intuition above, with a larger number of columns we might find more interesting permutations. Hence, we also experiment with four random integer mappings at once.

	perm1	perm2	perm3	perm4
a	1	5	4	2
b	2	3	5	3
c	3	1	2	4
d	4	4	1	1
e	5	2	3	5

**Fisher** taken from ?, we order the categories by increasing mean of the response.

For the following five methods, we use information about the continuous covariates to construct the mapping  $\psi$ .

### 6.3 Estimation details

Below we provide additional details to better clarify how new methods related to Means were estimated and remove basic problems in the data that complicate training models.

For PCA Means, we select the first  $k$  principal components which generate 95% of the variance in  $X$ , group by the unique categorical levels, and take the means of those principal components  $z_{1:k}$ . For SPCA Means, we use the default hyperparameters in the “sparsepca” R package.

In the Ames dataset, we remove the features “PoolQC”, “GarageQual”, and “GarageYrBlt” due to almost being perfectly correlated with other features and remove the missing data rows as well. In the King County House Sales dataset, we remove the ID and date of the house sale as the covariates. Finally, in the Pakistan dataset, many of the covariates are almost perfectly correlated such as “% girls enrolled”, “% boys enrolled”, and “gender parity score.” As a result, we removed such covariates resulting in the following list of covariates: Education score, Toilet, Province, Population, School infrastructure score, Total number of schools, Primary Schools with single teacher, Primary Schools with single classroom, Pakistan Economic Growth, Number of secondary schools, Electricity, No Facility, City, Global Terrorism Index - Pakistan, Complete Primary Schools, Building condition satisfactory, Drone attacks in Pakistan, Drinking water, Boundary wall, Bomb Blasts Occurred, % Complete Primary Schools, % Boys Enrolled.