Sparse encoding for more-interpretable feature-selecting representations in probabilistic matrix factorization

Joshua C. Chang, Patrick A. Fletcher, Jungmin Han
National Institutes of Health & med$\varepsilon_{\text{rrata}}$
{josh,patrick,jungmin}@mederrata.com

Ted L. Chang
med$\varepsilon_{\text{rrata}}$
ted@mederrata.com

Shashaank Vattikuti
Walter Reed Army Institute of Research
& med$\varepsilon_{\text{rrata}}$
shashaank@mederrata.com

Bart Desmet, Ayah Zirikly, Carson Chow
National Institutes of Health
{ bart.desmet, ayah.zirikly, carson.chow}@nih.gov

Abstract

Dimensionality reduction methods for count data are critical to a wide range of applications in medical informatics and other fields where model interpretability is paramount. For such data, hierarchical Poisson matrix factorization (HPF) and other sparse probabilistic non-negative matrix factorization (NMF) methods are considered to be interpretable generative models. They consist of sparse transformations for decoding their learned representations into predictions. However, sparsity in representation decoding does not necessarily imply sparsity in the encoding of representations from the original data features. HPF is often incorrectly interpreted in the literature as if it possesses encoder sparsity. The distinction between decoder sparsity and encoder sparsity is subtle but important. Due to the lack of encoder sparsity, HPF does not possess the column-clustering property of classical NMF – the factor loading matrix does not sufficiently define how each factor is formed from the original features. We address this deficiency by self-consistently enforcing encoder sparsity, using a generalized additive model (GAM), thereby allowing one to relate each representation coordinate to a subset of the original data features. In doing so, the method also gains the ability to perform feature selection. We demonstrate our method on simulated data and give an example of how encoder sparsity is of practical use in a concrete application of representing inpatient comorbidities in Medicare patients.

1 Introduction

For many inverse problems featuring high-dimensional count matrices, such as those found in healthcare, model interpretability is paramount. Building interpretable high-performing solutions is technically challenging and requires flexible frameworks. A general approach to these problems is to structure solutions into pipelines; if each step is interpretable, one can achieve interpretability of the overall larger model. A common first step in modeling high-dimensional data sets is to use dimensionality reduction to find tractable data representations (also called factors or embeddings), that are then fed into downstream analyses. Our goal is to develop a dimension reduction scheme for count matrices such that the reduced representation has an innate interpretation in terms of the original data features.

Interpretability versus explainability. We seek latent data representations that are not only post-hoc explainable (Laugel et al., 2019; Caruana et al., 2020), but also intrinsically interpretable (Rudin, 2019). Our definition of intrinsic interpretability requires clarity in the relationship between predictors and prediction, and meaningfulness of interactions and latent variables.

Post-hoc explanations are based on subjective examination of a solution through the lens of subject-matter expertise. For black-box models that lack intrinsic interpretability, these explanations are produced using inexact simpler approximating models (typically local linear regressions). These explanations can be misleading (Laugel et al., 2019).

Disentangled autoencoders. Disentangled variational autoencoders (Higgins et al., 2016; Tomczak & Welling, 2017; Deng et al., 2017) are deep learning models that are inherently mindful of post-hoc model explainability. Like other autoencoders, these models are encoder-decoder structured (see Definitions 1 and 2), where the encoder generates dimensionally reduced representations.

Definition 1. The **encoder** transformation maps input data features to latent representations

Definition 2. The **decoder** transformation maps latent representations to predictions

Disentangled autoencoders use a combination of penalties (Higgins et al., 2016; Hoffman et al., 2017) and structural constraints (Ainsworth et al., 2018) to encourage statistical independence in representations, facilitating explanation. These methods arose in computer vision and have demonstrated empirical utility in producing nonlinear factor models where the factors are conceptually sensible. Yet, due to the black-box nature of deep learning, explanations for how the factors are generated from the data, using local saliency maps for instance, are unreliable or imprecise (Laugel et al., 2019; Slack et al., 2020; Arun et al., 2020). In imaging applications, where the features are raw pixels, this type of interpretability is unnecessary. However, when modeling structured data problems, one often wishes to learn the effects of the individual data features.

Probabilistic matrix factorization. Probabilistic matrix factorization methods are related to autoencoders (Mnih & Salakhutdinov, 2008). These methods are often presented in the context of recommender systems. In these cases, rows of the input matrix are attributed to users, and columns (features) are attributed to items. Probabilistic matrix factorization methods are bi-linear in item- and user-specific effects, de-convolving them in a manner similar to item response theory (Chang et al., 2019). In applications with non-negative data, non-negative sparse matrix factorization methods further improve on interpretability by computing predictions using only additive terms (Lee & Seung, 1999).

For count matrices, Gopalan et al. (2014) introduced hierarchical Poisson matrix factorization (HPF). Suppose $\mathbf{Y} = (y_{ui})$ is a $U \times I$ matrix of non-negative integers, where each row corresponds to a *user* and each column corresponds to an *item* (feature). Adopting their notation, Gopalan et al. (2014) formulated their model as

$$y_{ui}|\boldsymbol{\Theta}, \mathbf{B} \sim \text{Poisson}\left(\sum_k \theta_{uk}\beta_{ki}\right) \qquad \begin{aligned} \theta_{uk}|\xi_u, a &\sim \text{Gamma}\left(a, \xi_u\right) \\ \beta_{ki}|\eta_i, c &\sim \text{Gamma}\left(c, \eta_i\right), \end{aligned} \tag{1}$$

where $\boldsymbol{\Theta} = (\theta_{uk})$ is a $U \times K$ matrix, and $\mathbf{B} = (\beta_{ki})$ is the representation decoder matrix. Additional priors $\eta_i \sim \text{Gamma}(c', c'/d')$ and $\xi_u \sim \text{Gamma}(a', a'/b')$ model item and user-specific variability in the dataset, and $a', b', c', d' \in \mathbb{R}^+$ are hyper-parameters. The row vector $\boldsymbol{\theta}_u = (\theta_{u1}, \ldots, \theta_{uK})$ constitutes a K-dimensional *representation* of the user, and the matrix $\mathbf{B} = (\beta_{ki})$ decodes the representation into predictions on the user's counts.

In HPF, the gamma priors on the decoder matrix $\mathbf{B} = (\beta_{ki})$ enforce non-negativity. Because the gamma distribution can have density at zero, these priors also allow for sparsity where only a few of the entries are far from zero. Sparsity, non-negativity, and the simple bi-linear structure of the likelihood in HPF combine to yield a simple interpretation of the model: in HPF, a predictive density for each matrix element is formed using a linear combination of a subset of representation elements, where the elements of $\mathbf{B}$ determine the relative additive contributions of each of the elements (Fig. 1a). However, the composition of each latent factor in terms of the original items is not explicitly determined but arises from Bayesian inference (Fig. 1c).

Limitations of HPF. Classical non-negative matrix factorization (NMF) is often touted for having a column-clustering property (Ding et al., 2005), where data features are grouped into coherent factors. The standard HPF of Eq. 1 lacks this property. In HPF, while each prediction is a linear combination of a subset of factors, each factor is not necessarily a linear combination of a subset of features (depicted in Fig. 1c).

The transformation matrix $\mathbf{B}$ defines a decoder (Def. 2) like a classic autoencoder. A corresponding encoding transformation (Def. 1) does not explicitly appear in the formulation of Eq. 1. Determining the composition of factors is not simply a matter of reading the decoding matrix row-wise. Mathemat-

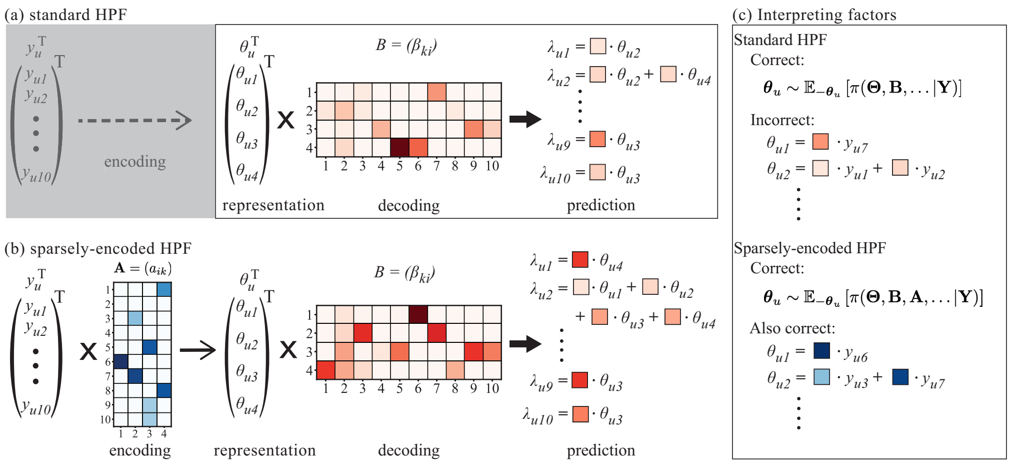
(a) standard HPF
encoding
representation
decoding
prediction
$B = (\beta_{ki})$
(b) sparsely-encoded HPF
$\mathbf{A} = (a_{ik})$
(c) Interpreting factors
Standard HPF
Correct:
$\boldsymbol{\theta}_u \sim \mathbb{E}_{-\boldsymbol{\theta}_u}[\pi(\boldsymbol{\Theta}, \mathbf{B}, \ldots|\mathbf{Y})]$
Incorrect:
Sparsely-encoded HPF
Correct:
$\boldsymbol{\theta}_u \sim \mathbb{E}_{-\boldsymbol{\theta}_u}[\pi(\boldsymbol{\Theta}, \mathbf{B}, \mathbf{A}, \ldots|\mathbf{Y})]$
Also correct:

Figure 1: **Interpreting hierarchical sparse probabilistic matrix factorization (HPF). (a) Standard HPF:** Rates λ_{ui} for Poisson-distributed predictions are sparse linear combinations of the learned representation as defined by the decoding matrix; this matrix does not define how representations are derived from the input data. **(b) Sparsely-encoded HPF (proposed method):** the mapping from input data to representation is given explicitly by a sparse encoding matrix. **(c) Interpreting representations:** It is tempting but misleading to read the decoding matrices row-wise in determining the feature subsets that contribute to forming a representation coordinate. Representations $\boldsymbol{\theta}_u$ are computed by inferring the statistics of an associated joint posterior distribution $\pi(\ldots|\mathbf{Y})$ – the sets of non-sparse entries in rows of the decoding matrices do not necessarily correspond to feature sets that determine the representation components. However, for sparsely-encoded HPF, the representations are explicit functions of subsets of features.

ically, sparsity in decoding does not imply sparsity in encoding analogous to how pseudo-inverses of sparse matrices are not necessarily sparse.

HPF is also unable to perform feature selection. By Eq. 1, predictions are formed by weighting representations using values from columns of the decoding matrix – a feature's corresponding terms in the decoding matrix will be near zero if and only if that feature's mean is near zero. Exclusion of a feature column from the decoding matrix yields no information on whether that feature plays a part in generating representations. Also, this deficiency can cause HPF to erroneously imply structure when none is present, as demonstrated in Fig. 3a) on a factorization of pure Poisson noise.

Our contributions. We propose a method to self-consistently constrain HPF so that its corresponding encoding transformation is explicit. In doing so, we improve interpretability of HPF, and give it the ability to perform feature selection. Constraining HPF in this manner also makes it more suitable to training with large datasets because the representation matrix does not need to be stored in memory. Using a medical claims case study, we demonstrate how our method facilitates reparameterization of decision rules in representation space into corresponding rules on the original data features.

2 Methods

In this section we describe our extension to HPF that resolves its limitations. Our method yields representations that have explicit sparse dependence on relevant features in the input data.

2.1 Improving HPF by constraining it

Our augmented HPF model takes the form

$$y_{ui}|\boldsymbol{\Theta}, \mathbf{B}, \boldsymbol{\varphi} \sim \text{Poisson}\left(f_i\left(\sum_k \theta_{uk}\beta_{ki}\right) + \varphi_i\right) \qquad \begin{aligned} \theta_{uk}|\mathbf{A}, \mathbf{y}_u, \xi_u &= \xi_u \sum_i g_i(y_{ui})\alpha_{ik} \\ \beta_{ki} &\sim \text{Normal}^+(0, 1/4K), \end{aligned} \tag{2}$$

where prior distributions for the model parameters are defined later in this section.

The key point is that the encoder function (that computes θ_{uk}) is an explicit function of the input data, formulated using a generalized additive model (GAM) (Rigby & Stasinopoulos, 2005; Hastie & Tibshirani, 1987; Klein et al., 2015). The encoding matrix $\mathbf{A} = (\alpha_{ik})$ controls how features map into the representation.

To allow the model to perform automatic feature selection, we also incorporate a non-negative item-specific gain term φ_i as a background Poisson rate for item i that is intended to be independent of the factor model. We also slightly generalize the likelihood of Eq. 1 by giving each feature an associated link function f_i, which models nonlinearities without sacrificing interpretability.

The distributions of the parameters of the encoder are learned self-consistently with other model parameters. In the process, one is training not only the generative model, but also the subsequent Bayesian inference of mapping data to representation by learning the statistics of the posterior distribution,

$$\boldsymbol{\theta}_u|\mathbf{y}_u, \mathbf{Y} \sim \iint \pi(\boldsymbol{\theta}_u, \mathbf{B}, \varphi|\mathbf{y}_u, \mathbf{Y})\mathrm{d}\mathbf{B}\mathrm{d}\varphi, \tag{3}$$

where the generative process has been marginalized. In short, the model of Eq. 2 uses the marginal posterior distribution of the encoding matrix $\mathbf{A}$ to reparameterize this Bayesian inference. Doing so amortizes this inference, making it trivial to apply the model to new data in order to compute new representations. It also allows us to impose desirable constraints on the representations themselves.

In the original HPF, the parameters ξ_u are used to account for variability in user activity (row sums). Similarly, the η_i parameters account for variability in item popularity (in column sums). To simplify the method, we pre-set these parameters (based on some training data) to $\xi_u = 1$ and $\eta_i = \frac{1}{U}\sum_u y_{ui}$, where η_i is absorbed into the function f_i. Doing so de-scales the encoder parameters α_{ik} so we can generalize weakly-informative and other scale-dependent priors within the model to disparate datasets, as is common in preprocessing for Bayesian statistical inference problems (Gelman et al., 2017). One may also model over-dispersed data using $\xi_u = {}^{U\sum_i y_{ui}}/_{\sum_u \sum_i y_{ui}}$, to account for document-size variability.

We use sparsity to achieve feature selection. We encourage the elements α_{ik} and φ_i to be mutually exclusive by using the decomposition

$$\begin{aligned} \alpha_{ik} &= u_{ik}\frac{s_i^+}{s_i^+ + s_i^-} & [s_i^+\ s_i^-]^\intercal &\sim \text{Horseshoe}^+(1,1) \\ & & [u_{1k}\ u_{2k}\ \ldots\ u_{Ik}]^\intercal &\sim \text{Horseshoe}^+(1, {}^1/\sqrt{UI}) \\ \varphi_i|\eta_i, w_i, s_i^\pm &= \eta_i w_i \frac{s_i^-}{s_i^+ + s_i^-} & w_i &\sim \text{Normal}^+(0, 10), \end{aligned} \tag{4}$$

where the non-negative version of the Horseshoe$^+$ prior (Carvalho et al., 2009; 2010; Polson & Scott, 2011) is the hierarchical Bayesian model

$$\mathbf{x} \sim \text{Horseshoe}^+(\lambda_0, \tau_0) \quad \Longleftrightarrow \quad \begin{aligned} x_j|\lambda_j, \tau &\sim \text{Normal}^+(0, \lambda_j\tau) \\ \lambda_j &\sim \text{Cauchy}^+(0, \lambda_0) \\ \tau &\sim \text{Cauchy}^+(0, \tau_0). \end{aligned} \tag{5}$$

This concentrates marginal distributions of vector components near zero. Additionally, it minimally shrinks large components, resulting in lower bias compared to lasso and other alternatives (Bhadra et al., 2015a;b; 2019; Piironen & Vehtari, 2017b). The horseshoe has previously been applied in other factorization methods, including autoencoders (Ghosh & Doshi-Velez, 2017b) and item response theory (Chang et al., 2019), but not to probabilistic matrix factorization.

Applied to the parameters $s_i^\pm$, sparsity discourages variables that load into the factor model from leaking into the corresponding background rate term φ_i. Conversely, variables that load into φ_i are discouraged from appearing in α_{ik}.

Finally, as is often done in variational autoencoders, we regularize the representation by placing unit half normal priors on its components

$$\theta_{uk}|g, \mathbf{Y}, \mathbf{A} = \xi_u \sum_i g_i(y_{ui})\alpha_{ik} \sim \text{Normal}^+_{\theta_{uk}}(0, 1). \tag{6}$$

The choices of the encoding function f_i and decoding functions g_i are application-specific, and may be learned (Rigby & Stasinopoulos, 2005). So as not to distract from our focus on improving the interpretability of pre-existing matrix factorization approaches, we fix these functions here. In standard Poisson matrix factorization approaches, $f_i(x) = g_i(x) = x,\ \forall i$. Equivalently, we choose to rescale the inputs so that

$$f_i(x) = \eta_i x \qquad \text{and} \qquad g_i(x) = f_i^{-1}(x) = {}^{x}\!/_{\eta_i}. \tag{7}$$

Another choice for these functions can be motivated by Poisson regression with a logarithmic link function, where $f_i(x) = e^{\eta_i x} - 1$, and $g_i(x) = f_i^{-1}(x) = \log\left({}^{x}\!/_{\eta_i} + 1\right)$. For maximum interpretability, restricting f_i and g_i to monotonically increasing functions where $g_i(0) = f_i(0) = 0$ results in order-preserving representations that are zero when the corresponding feature counts are zero.

2.2 Interpreting representations and derived quantities

In constraining the encoder mapping using the generalized additive model of Eq. 6, we regain the column-clustering property of classical non-negative matrix factorization methods: each representation component is explicitly determined from a well-defined subset (cluster) of the data features. In Fig. 1c), we demonstrate how the encoding matrix can be read to determine the composition of the factors. Consequently, decision rules over the representation can be easily expressed as decision rules over the original features,

$$\theta_{uk} \in (a, b) \iff \sum_{j \in \Omega_k} g_j(y_{uj}) \alpha_{jk} \in \left({}^{a}\!/_{\xi_u}, {}^{b}\!/_{\xi_u}\right), \tag{8}$$

where Ω_k is the subset of features that determines factor k. As we will demonstrate in our main case study, Eq. 8 is useful for inverting clustering rules defined over the representations.

2.3 Inference

The model of Eq. 2 is a generalized linear factor model that we have mathematically related to a probabilistic autoencoder. When augmenting HPF with explicit encoder inference, as we have done, one obtains a probabilistic autoencoder. This suggests that previous work can serve as a guide for training, especially work done on using the horseshoe prior in Bayesian neural networks (Ghosh & Doshi-Velez, 2017a; Ghosh et al., 2018; Louizos et al., 2017).

In particular, Ghosh et al. (2018) investigated structured variational approximations of inference of Bayesian neural networks that use the horseshoe prior and found them to have similar predictive power as mean-field variational approximations. The disadvantage of structured approximations is the extra computational cost of inferring covariance matrices. For these reasons, we focus on mean-field black-box variational inference, using Ghosh et al. (2018) as a guide, noting consistency of their scheme with other works that have investigated variational inference on problems using the horseshoe prior (Wand et al., 2011; Louizos et al., 2017).

As in Ghosh & Doshi-Velez (2017a); Ghosh et al. (2018); Chang et al. (2019), for numerical stability, we reparameterize the Cauchy distributions in terms of the auxiliary inverse Gamma representation (Makalic & Schmidt, 2016),

$$x \sim \text{Cauchy}^+(0, \sigma) \qquad \iff \qquad \begin{array}{l} x^2 \sim \text{Inverse-Gamma}\left({}^{1}\!/_{2},\ {}^{1}\!/_{\lambda}\right) \\ \lambda \sim \text{Inverse-Gamma}\left({}^{1}\!/_{2},\ {}^{1}\!/_{\sigma^2}\right). \end{array} \tag{9}$$

We perform approximate Bayesian inference using fully-factorized mean-field Automatic Differentiation Variational Inference (ADVI) (Kucukelbir et al., 2017). For all matrix elements, we utilized softplus-transformed Gaussians, and coupled these to inverse-Gamma distributions for the scale parameters, as investigated in Wand et al. (2011). For our use cases, we implemented a minibatch training regimen common to machine learning, with stepping given by the Adam optimizer (Kingma & Ba, 2017) combined with the Lookahead algorithm (Zhang et al., 2019) for stabilization.

Bayesian sparsity methods concentrate marginal distributions for parameters near zero. To further refine the Bayesian parameter densities so that some parameters are identically zero in distribution, one may use projection-based sparsification (Piironen & Vehtari, 2017b). Finally, one can assess predictive power without model refitting using approximate leave-one-out cross validation (LOO) using the Widely Applicable Information Criterion (WAIC) (Watanabe, 2010; Gelman et al., 2014; Piironen & Vehtari, 2017a; Vehtari et al., 2017; Chang, 2019) or Pareto smoothed importance sampling LOO (PSIS-LOO) (Vehtari et al., 2017).

3 Experiments

We implemented the method in tensorflow-probability (Dillon et al., 2017), modifying the inference routines for implementing our variational approximation. Our implementation can be found at `github:mederrata/spmf`, along with notebooks reproducing our simulation results. We present here simulation results and an application to medical claims data.

3.1 Simulations

To demonstrate the properties of our method, we factorized synthetic datasets of: a) completely random noise with no underlying structure, b) a mixture of random noise and linear structure, and c) a mixture of random noise and nonlinear structure.

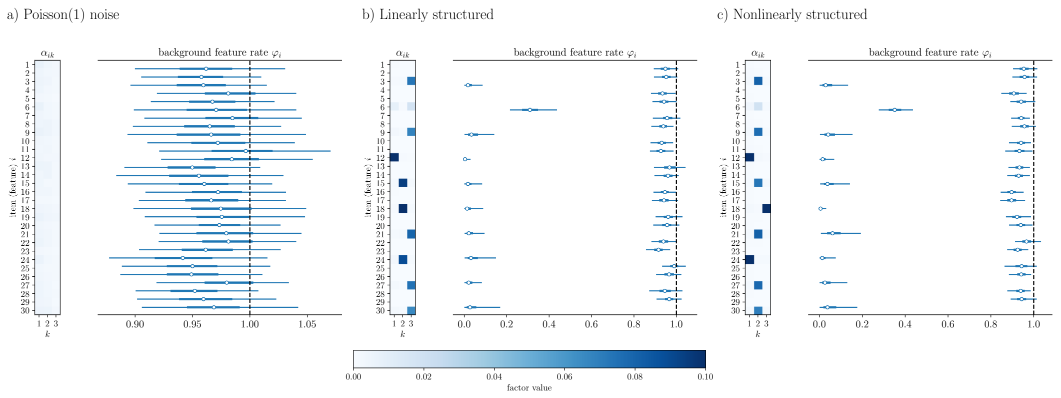

Figure 2: **Factorization of simulated datasets**. The (mean) effective encoding matrix $\mathbf{A} = (\alpha_{ik})$ for each factor process, placed on a common color scale, and the posterior distribution of the background process rate φ_i by item for a) Poisson(1) noise, where there is no relationship between the features, b) linear factor model where every third variable is generated from a dense factor model and the other variables are Poisson(1) noise, c) nonlinear factor model where every third variable is generated from a dense nonlinear factor model and the other variables are Poisson(1) noise. See Fig. 3 for standard HPF on these datasets for comparison.

For a), we sampled a $50,000 \times 30$ Poisson(1) random matrix. Figure 2a) shows the inferred mean encoder matrix $\mathbf{A}$ along with the posterior distributions for each of the background components φ_i. We see that all features are excluded from the encoding matrix, showing up instead as background noise.

Next, we created a test system where there is underlying linear structure mixed with noise. For this system, we again used $I = 30$ features and put every third feature into a dense system by generating a random 10×10 decoding matrix $\mathbf{B}$, sampling representations from a non-negative truncated normal distribution, and sampling counts according to the generative process of Eq. 1. For the remaining features, we used Poisson(1) noise. After simulating $50,000$ records by this process, we performed factorization again into $K = 3$ dimensions. The results of this factorization are shown in Figure 2b), where it is clear that every third feature falls into the overall factor model and the remaining features show up as background noise.

As an example of factorization under model mismatch, we generated random data with underlying nonlinear structure. Here again, we used every third feature, simulating $\mathbf{B}$ and $\boldsymbol{\Theta}$ as before. However, we simulated counts for these features using the model $y_{ui} \sim \text{Poisson}\left((\sum_k {}^{\theta_{uk}\beta_{ki}}/2) \exp(-\sum_k {}^{\theta_{uk}\beta_{ki}}/2) + (\sum_k {}^{\theta_{uk}\beta_{ki}}/2)^2 \right)$. Factorization of this dataset is shown in Figure 2c). Again, it is clear that every third feature falls into the overall factor model and the remaining features are load into the background process with rates near 1, indicating that even when the model is mis-specified, it can successfully separate structure from noise. In Supplemental Fig. S1, we present the same factorizations while using the logarithmic link function described in Sec. 2.1, demonstrating robustness to mis-specification of the link functions as measured using the WAIC.

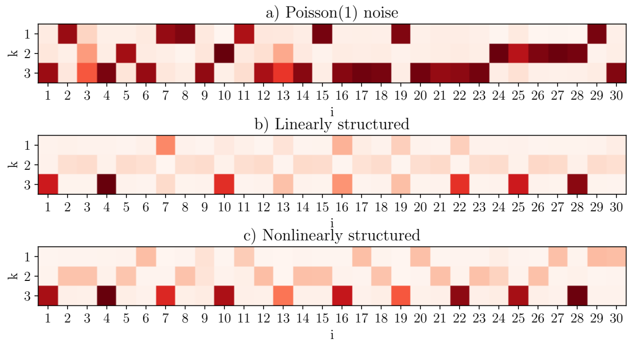

Figure 3: Decoder matrices $\mathbf{B} = (\beta_{ki})$ for standard HPF factorization of the synthetic datasets of Fig. 2 using the python package `hpfrec`. Shown are posterior means.

Comparison to standard HPF. In standard HPF (Gopalan et al., 2014) only decoder matrices are inferred, and encoders are not explicitly reconstructed. Fig. 3 demonstrates factorization of the same synthetic datasets using standard hierarchical Poisson matrix factorization (Gopalan et al., 2014) found in the `hpfrec` package. In all three examples, the standard HPF fails to remove independent noise items from the factor model. In contrast, our method excludes all irrelevant features from the factor model (Fig. 2). In this case, the encoder is not explicitly solved. It is incorrect to read the decoder matrix $\mathbf{B}$ row-wise, to say for instance that the first factor in Fig. 3a is determined from items $\{2, 3, 7, \ldots, 29\}$. However, results from standard HPF are often erroneously interpreted in this manner, suggesting that there is structure to the dataset even when none is present. Additionally, for this reason, the generative process fails to adequately fit the data in that it correlates sources of independent noise.

3.2 Comorbidities from billing codes

As a real-world case study, we used a 5% sample of the Medicare Limited Data Set (LDS) over the years 2009–2011 to discover a representation of inpatient comorbidity during a hospital visit. The LDS consists of de-identified medical claims data for Medicare and Medicaid recipients across the United States. Pursuant to a data use agreement, the Center for Medicare and Medicaid Services (CMS) provides a 5% sample of this dataset for research purposes. A single hospital visit consists of multiple claims across providers and types of services. No standard method for grouping claims into hospital visits within claims data exists. A heuristic algorithm (often called a *grouper* algorithm) is used to reconstruct medical and billing events during a visit or type of service from claims. Our grouper algorithm collapsed the claims into $U = 1,949,788$ presumptive inpatient visits. Diagnostic codes were then made coarser-grained by backing off from the original $\approx$ 13,000 ICD-9-CM to 136 clinically-relevant categories using the top two levels of the CCS multilevel classification. Within each visit, we counted the number of codes that fell into each of the CCS categories. Fig. 4 presents the encoding matrix $\mathbf{A} = (\alpha_{ik})$ for a factorization of comorbidities into four dimensions, the transpose of the decoding matrix $\mathbf{B} = (\beta_{ki})$, and the vector of background process rates $\boldsymbol{\varphi} = (\varphi_i)$ for the same model.

The values in the encoding matrix provide coefficients that are used in Eq. 6 to produce a weighted sum of billing code counts, which is then used to formulate a representation. One may read the encoding matrix column-wise in order to determine the feature composition of each of the representation factors. Being able to do so facilitates interpretation of the factor model, by allowing one to understand a single factor at a time by focusing on subsets of the original features. For example, conceptually, it is easy to see what factor 2 represents. It is computed by tallying up various billing codes that pertain to lung ailments – lung cancer (CCS 2.3), several broad respiratory disorders (CCS 8.x), and heart disease (CCS 7.2). The relative weights of these codes are depicted by the color of the shading.

The interpretation of matrix factorization is not provided by the decoding matrix, which is the sole output of standard HPF. The decoding matrix provides only an incomplete picture of the structure of the data. Recall from Fig. 1 that the decoding matrix is not to be read row-wise (or column-wise in the transpose depiction of Fig. 4). Looking solely at the decoding, and reading it in this incorrect way,

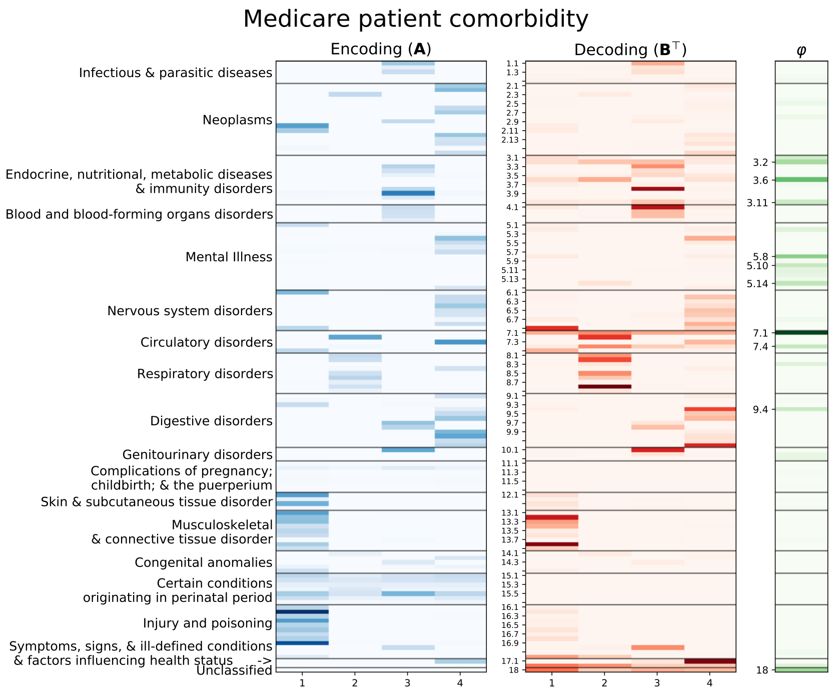

Figure 4: **Medicare comorbidity factorization** for inpatient visits based on medical claims from a 5% sample of the Medicare Limited Dataset (LDS), in four factor dimensions. Prior to factorization, we mapped each raw ICD diagnostic code into the second tier of the Clinical Classification Software (CCS), counting the number of codes present within each broad category. Shown are posterior means. **Left:** encoding $\mathbf{A} = (\alpha_{ik})$, **middle:** decoding $\mathbf{B}^\top = (\beta_{ki})^\top$, **right:** background $\boldsymbol{\varphi} = (\varphi_i)$

one might erroneously conclude that lower respiratory disorders (CCS 8.8) are the main determinant of factor 2. However, this conclusion is incorrect – diagnoses of heart disease (CCS 7.2) are the main determinant.

The decoding matrix also provides misleading insights on feature selection. For example, lung cancer (CCS 2.3) appears only very faintly in the decoding. For this reason, one might come to the erroneous conclusion that it is not important as a feature. However, recall that by Eq. 1, relatively rare features will only faintly register in the decoder matrix. The rate of lung cancer diagnoses is low, yet lung cancer diagnoses are predictive of - and coincide with - other respiratory issues. Hence, lung cancer (CCS 2.3) appears strongly in the encoder matrix.

After computing representations for the entire dataset, one may cluster patients into diagnostic groups. One way of doing so is through stratification, for instance into low, medium, and high groups for each of the four factors in Fig. 4. Doing so based on quantiles yields thresholds between the groups, defined over representations. Using Eq. 8 one can easily convert these thresholds into decision rules on the counts. For example, a patient presenting with one or more lung cancer-related diagnoses would generally be placed in the medium or high strata for factor 2, depending on the number of other respiratory billing codes in that visit. Recall that the input data is sparse so in general the low strata for each representation would encompass people who had no or very few of the associated diagnoses. As a first step in a modeling pipeline, one could use the strata to segment a larger overall model, so that the model has both local and global behavior, while making it easy to interpret how individual or collective billing codes contribute to an overall prediction.

4 Discussion and Conclusion

We introduced a constrained HPF where an encoder transformation is learned self-consistently with the matrix factorization. By imposing sparsity on the encoder, rather than on the decoder, we improve interpretability of HPF. We demonstrated the approach on simulated data, showing that the method can successfully separate structure from noise, even when the model is mis-specified. We also presented a comorbidity factorization as a case study.

Although we focused on Poisson factorization, our central argument holds for other sparse matrix factorization methods. Sparse decoding matrices (loading matrices) inferred using these methods are generally not orthogonal. Unlike in classical or orthogonally-rotated PCA, the transpose of these decoding matrices does not correspond to their pseudo-inverse. Hence, decoding matrices should never be interpreted row-wise (Fig. 1c).

LIMITATIONS AND EXTENSIONS

Our method relies on the horseshoe prior for sparsification. The horseshoe prior relies on scaling hyperparameters, which control the effective sparsity of the method. In order to make these priors scale asymptotically with data size (Piironen & Vehtari, 2017b), we chose to scale this prior using $1/\sqrt{UI}$. Empirically, this choice, along with the scaling of the priors on β_{ki}, led to desirable behavior in simulations like those in Fig. 2 under several combinations of U and I. In effect, we have taken the liberty of formulating our method based on these considerations so that it is usable without needing to manually choose hyperparameters. One may wish to rescale the horseshoe prior in order to control sparsity. Further guidance to the regularization scale will require analysis outside of the scope of this manuscript.

We note that one could also formulate our method using the sparsifying priors found in Gopalan et al. (2014). The chief advantage of the original HPF formulation is in how it yields explicit variational inference updates. However, our method yields well to ADVI, achieving convergence with a learning rate of 0.05 in approximately 100 epochs in all included examples.

A limitation of our method, shared by standard HPF, is that a generalized linear model does not have the expressivity of nonlinear paradigms such as deep learning. For some applications, with sufficient data, nonlinear models may be more performant. We note that one could place non-linearity in either f_i or g_i, without compromising interpretability of the representation. These functions may be learned using Gaussian processes (Chang et al., 2014) splines, or even neural networks, making the method more like other probabilistic autoencoders. So long as the conditions of monotonicity and a fixed point at $y = 0$ are maintained, the overall method remains interpretable. However, the simplicity of HPF offers statistical advantages that help it generalize better than deep learning except when there is enough data to learn any true nonlinearities in the true generating process. Additionally, while we do not explore this, a strength of Bayesian modeling is that it provides a principled approach to incorporating prior information. One could encourage or discourage features from co-factoring by setting suitable priors on the encoding and decoding matrices.

ACKNOWLEDGEMENTS

We thank the Innovation Center of the Center for Medicare and Medicaid services for providing access to the CMS Limited Dataset. We also thank the med$\varepsilon_{\text{rrata}}$ team (particularly Joe Maisog) for their support in various data pre-processing tasks. JCC, AZ, and BD were supported in-part by the US Social Security Administration. PF, JH, CCC, and SV were supported by the Intramural Research Program of the NIH, NIDDK. This work used the Extreme Science and Engineering Discovery Environment (XSEDE) (Towns et al., 2014), which is supported by National Science Foundation grant number ACI-1548562 through allocation TG-DMS190042. We also thank Amazon Web Services for providing computational resources and Boost Labs LLC for helping with visualization.

REFERENCES

Samuel Ainsworth, Nicholas Foti, Adrian KC Lee, and Emily Fox. Interpretable VAEs for nonlinear group factor analysis. *arXiv:1802.06765 [cs, stat]*, February 2018. URL `http://arxiv.org/abs/1802.06765`. arXiv: 1802.06765.

Nishanth Arun, Nathan Gaw, Praveer Singh, Ken Chang, Mehak Aggarwal, Bryan Chen, Katharina Hoebel, Sharut Gupta, Jay Patel, Mishka Gidwani, Julius Adebayo, Matthew D. Li, and Jayashree Kalpathy-Cramer. Assessing the (Un)Trustworthiness of Saliency Maps for Localizing Abnormalities in Medical Imaging. *arXiv:2008.02766 [cs]*, August 2020. URL http://arxiv.org/abs/2008.02766. arXiv: 2008.02766.

Anindya Bhadra, Jyotishka Datta, Nicholas G. Polson, and Brandon Willard. The Horseshoe+ Estimator of Ultra-Sparse Signals. *arXiv:1502.00560 [math, stat]*, February 2015a. URL http://arxiv.org/abs/1502.00560. tex.ids: bhadraHorseshoeEstimatorUltraSparse2015 arXiv: 1502.00560.

Anindya Bhadra, Jyotishka Datta, Nicholas G. Polson, and Brandon T. Willard. Default Bayesian analysis with global-local shrinkage priors. *arXiv:1510.03516 [stat]*, October 2015b. URL http://arxiv.org/abs/1510.03516. arXiv: 1510.03516.

Anindya Bhadra, Jyotishka Datta, Nicholas G. Polson, and Brandon Willard. Lasso Meets Horseshoe: A Survey. *Statist. Sci.*, 34(3):405–427, August 2019. ISSN 0883-4237, 2168-8745. doi: 10.1214/19-STS700. URL https://projecteuclid.org/euclid.ss/1570780977. Publisher: Institute of Mathematical Statistics.

Rich Caruana, Scott Lundberg, Marco Tulio Ribeiro, Harsha Nori, and Samuel Jenkins. Intelligible and Explainable Machine Learning: Best Practices and Practical Challenges. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, pp. 3511–3512, New York, NY, USA, August 2020. Association for Computing Machinery. ISBN 978-1-4503-7998-4. doi: 10.1145/3394486.3406707. URL https://doi.org/10.1145/3394486.3406707.

Carlos M. Carvalho, Nicholas G. Polson, and James G. Scott. Handling Sparsity via the Horseshoe. In *Artificial Intelligence and Statistics*, pp. 73–80, April 2009. URL http://proceedings.mlr.press/v5/carvalho09a.html. ISSN: 1938-7228 Section: Machine Learning.

Carlos M. Carvalho, Nicholas G. Polson, and James G. Scottt. The horseshoe estimator for sparse signals. *Biometrika*, 97(2):465–480, 2010. ISSN 0006-3444. URL https://www.jstor.org/stable/25734098. Publisher: [Oxford University Press, Biometrika Trust].

Joshua C. Chang. Predictive Bayesian selection of multistep Markov chains, applied to the detection of the hot hand and other statistical dependencies in free throws. *Royal Society Open Science*, 6(3):182174, March 2019. doi: 10.1098/rsos.182174. URL https://royalsocietypublishing.org/doi/full/10.1098/rsos.182174.

Joshua C. Chang, Van M. Savage, and Tom Chou. A Path-Integral Approach to Bayesian Inference for Inverse Problems Using the Semiclassical Approximation. *J Stat Phys*, 157(3):582–602, November 2014. ISSN 1572-9613. doi: 10.1007/s10955-014-1059-y. URL https://doi.org/10.1007/s10955-014-1059-y.

Joshua C. Chang, Shashaank Vattikuti, and Carson C. Chow. Probabilistically-autoencoded horseshoe-disentangled multidomain item-response theory models. *arXiv:1912.02351 [cs, stat]*, December 2019. URL http://arxiv.org/abs/1912.02351. arXiv: 1912.02351.

Zhiwei Deng, Rajitha Navarathna, Peter Carr, Stephan Mandt, Yisong Yue, Iain Matthews, and Greg Mori. Factorized Variational Autoencoders for Modeling Audience Reactions to Movies. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6014–6023, July 2017. doi: 10.1109/CVPR.2017.637. ISSN: 1063-6919.

Joshua V. Dillon, Ian Langmore, Dustin Tran, Eugene Brevdo, Srinivas Vasudevan, Dave Moore, Brian Patton, Alex Alemi, Matt Hoffman, and Rif A. Saurous. TensorFlow Distributions. *arXiv:1711.10604 [cs, stat]*, November 2017. URL http://arxiv.org/abs/1711.10604. arXiv: 1711.10604.

Chris Ding, Xiaofeng He, and Horst D. Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *in SIAM International Conference on Data Mining*, 2005.

Andrew Gelman, Jessica Hwang, and Aki Vehtari. Understanding predictive information criteria for Bayesian models. *Stat Comput*, 24(6):997–1016, November 2014. ISSN 1573-1375. doi: 10.1007/s11222-013-9416-2. URL https://doi.org/10.1007/s11222-013-9416-2.

Andrew Gelman, Daniel Simpson, and Michael Betancourt. The Prior Can Often Only Be Understood in the Context of the Likelihood. *Entropy*, 19(10):555, October 2017. doi: 10.3390/e19100555. URL `https://www.mdpi.com/1099-4300/19/10/555`. Number: 10 Publisher: Multidisciplinary Digital Publishing Institute.

Soumya Ghosh and Finale Doshi-Velez. Model Selection in Bayesian Neural Networks via Horseshoe Priors. *arXiv:1705.10388 [stat]*, May 2017a. URL `http://arxiv.org/abs/1705.10388`. arXiv: 1705.10388.

Soumya Ghosh and Finale Doshi-Velez. Model Selection in Bayesian Neural Networks via Horseshoe Priors. May 2017b. URL `https://arxiv.org/abs/1705.10388v1`.

Soumya Ghosh, Jiayu Yao, and Finale Doshi-Velez. Structured Variational Learning of Bayesian Neural Networks with Horseshoe Priors. *arXiv:1806.05975 [cs, stat]*, June 2018. URL `http://arxiv.org/abs/1806.05975`. tex.ids: ghoshStructuredVariationalLearning2018 arXiv: 1806.05975.

Prem Gopalan, Jake M. Hofman, and David M. Blei. Scalable Recommendation with Poisson Factorization. *arXiv:1311.1704 [cs, stat]*, May 2014. URL `http://arxiv.org/abs/1311.1704`. arXiv: 1311.1704.

Trevor Hastie and Robert Tibshirani. Generalized Additive Models: Some Applications. *Journal of the American Statistical Association*, 82(398):371–386, June 1987. ISSN 0162-1459. doi: 10.1080/01621459.1987.10478440. URL `https://www.tandfonline.com/doi/abs/10.1080/01621459.1987.10478440`. Publisher: Taylor & Francis _eprint: https://www.tandfonline.com/doi/pdf/10.1080/01621459.1987.10478440.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. November 2016. URL `https://openreview.net/forum?id=Sy2fzU9gl`.

Matt Hoffman, Carlos Riquelme, and Matthew Johnson. The Beta VAE's Implicit Prior. 2017. URL `http://bayesiandeeplearning.org/2017/papers/66.pdf`.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, January 2017. URL `http://arxiv.org/abs/1412.6980`. arXiv: 1412.6980.

Nadja Klein, Thomas Kneib, and Stefan Lang. Bayesian Generalized Additive Models for Location, Scale, and Shape for Zero-Inflated and Overdispersed Count Data. *Journal of the American Statistical Association*, 110(509):405–419, January 2015. ISSN 0162-1459. doi: 10.1080/01621459.2014.912955. URL `https://doi.org/10.1080/01621459.2014.912955`. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/01621459.2014.912955.

Alp Kucukelbir, Dustin Tran, Rajesh Ranganath, Andrew Gelman, and David M. Blei. Automatic differentiation variational inference. *J. Mach. Learn. Res.*, 18(1):430–474, January 2017. ISSN 1532-4435.

Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. The Dangers of Post-hoc Interpretability: Unjustified Counterfactual Explanations. *arXiv:1907.09294 [cs, stat]*, July 2019. URL `http://arxiv.org/abs/1907.09294`. arXiv: 1907.09294.

Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, October 1999. ISSN 1476-4687. doi: 10.1038/44565. URL `https://www.nature.com/articles/44565`. Number: 6755 Publisher: Nature Publishing Group.

Christos Louizos, Karen Ullrich, and Max Welling. Bayesian Compression for Deep Learning. *arXiv:1705.08665 [cs, stat]*, November 2017. URL `http://arxiv.org/abs/1705.08665`. arXiv: 1705.08665.

Enes Makalic and Daniel F. Schmidt. A simple sampler for the horseshoe estimator. *IEEE Signal Process. Lett.*, 23(1):179–182, January 2016. ISSN 1070-9908, 1558-2361. doi: 10.1109/LSP.2015.2503725. URL http://arxiv.org/abs/1508.03884. arXiv: 1508.03884.

Andriy Mnih and Russ R Salakhutdinov. Probabilistic Matrix Factorization. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis (eds.), *Advances in Neural Information Processing Systems 20*, pp. 1257–1264. Curran Associates, Inc., 2008. URL http://papers.nips.cc/paper/3208-probabilistic-matrix-factorization.pdf.

Juho Piironen and Aki Vehtari. Comparison of Bayesian predictive methods for model selection. *Stat Comput*, 27(3):711–735, May 2017a. ISSN 1573-1375. doi: 10.1007/s11222-016-9649-y. URL https://doi.org/10.1007/s11222-016-9649-y.

Juho Piironen and Aki Vehtari. Sparsity information and regularization in the horseshoe and other shrinkage priors. *Electron. J. Statist.*, 11(2):5018–5051, 2017b. ISSN 1935-7524. doi: 10.1214/17-EJS1337SI. URL http://arxiv.org/abs/1707.01694. arXiv: 1707.01694.

Nicholas G. Polson and James G. Scott. On the half-Cauchy prior for a global scale parameter. *arXiv:1104.4937 [stat]*, April 2011. URL http://arxiv.org/abs/1104.4937. arXiv: 1104.4937.

R. A. Rigby and D. M. Stasinopoulos. Generalized additive models for location, scale and shape. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 54 (3):507–554, 2005. ISSN 1467-9876. doi: 10.1111/j.1467-9876.2005.00510.x. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9876.2005.00510.x%4010.1111/%28ISSN%291467-9876.TOP_SERIES_C_RESEARCH. _eprint: https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-9876.2005.00510.x.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, May 2019. ISSN 2522-5839. doi: 10.1038/s42256-019-0048-x. URL https://www.nature.com/articles/s42256-019-0048-x. Number: 5 Publisher: Nature Publishing Group.

Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods. *arXiv:1911.02508 [cs, stat]*, February 2020. URL http://arxiv.org/abs/1911.02508. arXiv: 1911.02508.

Jakub M. Tomczak and Max Welling. VAE with a VampPrior. *arXiv:1705.07120 [cs, stat]*, May 2017. URL http://arxiv.org/abs/1705.07120. arXiv: 1705.07120.

J. Towns, T. Cockerill, M. Dahan, I. Foster, K. Gaither, A. Grimshaw, V. Hazlewood, S. Lathrop, D. Lifka, G. D. Peterson, R. Roskies, J. R. Scott, and N. Wilkins-Diehr. XSEDE: Accelerating scientific discovery. *Computing in Science & Engineering*, 16(5):62–74, October 2014. ISSN 1521-9615. doi: 10.1109/MCSE.2014.80. URL doi.ieeecomputersociety.org/10.1109/MCSE.2014.80.

Aki Vehtari, Andrew Gelman, and Jonah Gabry. Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Stat Comput*, 27(5):1413–1432, September 2017. ISSN 1573-1375. doi: 10.1007/s11222-016-9696-4. URL https://doi.org/10.1007/s11222-016-9696-4.

Matthew P. Wand, John T. Ormerod, Simone A. Padoan, and Rudolf Frühwirth. Mean Field Variational Bayes for Elaborate Distributions. *Bayesian Anal.*, 6(4):847–900, December 2011. ISSN 1936-0975, 1931-6690. doi: 10.1214/11-BA631. URL https://projecteuclid.org/euclid.ba/1339616546.

Sumio Watanabe. Asymptotic Equivalence of Bayes Cross Validation and Widely Applicable Information Criterion in Singular Learning Theory. *Journal of Machine Learning Research*, 11 (Dec):3571–3594, 2010. ISSN ISSN 1533-7928. URL http://www.jmlr.org/papers/v11/watanabe10a.html.

Michael R. Zhang, James Lucas, Geoffrey Hinton, and Jimmy Ba. Lookahead Optimizer: k steps forward, 1 step back. *arXiv:1907.08610 [cs, stat]*, December 2019. URL http://arxiv.org/abs/1907.08610. arXiv: 1907.08610.

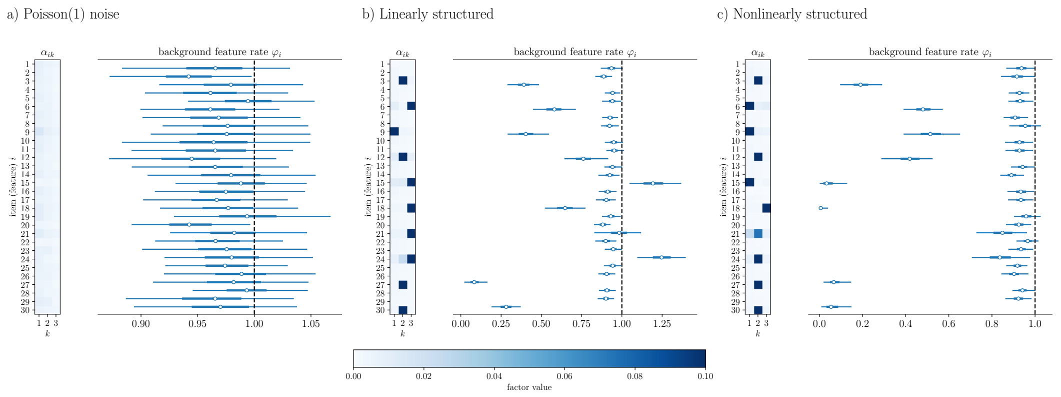

Figure S1: **Factorization based on the logarithmic link function** of Section 2.1 of the synthetic dataset of Fig. 2

.

	$f_i(x) = x/\eta_i$	$f_i(x) = \log(x/\eta_i + 1)$
Poisson(1) noise	$(3.54 \pm 0.02) \times 10^5$	$(3.54 \pm 0.02) \times 10^5$
Linearly structured	$(4.45 \pm 0.03) \times 10^5$	$(4.43 \pm 0.03) \times 10^5$
Nonlinearly structured	$(4.13 \pm 0.03) \times 10^5$	$(4.13 \pm 0.03) \times 10^5$

Table 1: **Model comparison using WAIC** ($\pm$ standard error) for factorizatons of the synthetic data. Lower is better.

Here we provide additional experiments. We note that the code for reproducing these experiments and those in the main manuscript can be found at github:mederrata/spmf. These supplemental examples can be found at Google Collaboratory. Please refer to the notebooks therein, where one can also find the details behind hyperparameters and optimization. In general, we did little tuning of the method beyond tuning the learning rate for stable inference.

S1 Comparing choices of f, g

In our method, we are free to choose functions f_i, g_i. We evaluate models for predictive power without refitting by using the WAIC. Here we provide an example of factorizations under different functions f, g, and compare the models. In Fig. S1, we performed factorization of the synthetic datasets of Fig. 2 using logarithmic link function of Section 2.1. Although the model is mis-specified, the key structure of the data is still exposed and irrelevant features are removed.

We then used WAIC to compare the use of the log link function versus the identity function. On the basis of predictive accuracy, the two models are similar as shown in Table 1, so the method is not sensitive to this choice.

S2 Sample sizes

For a systematic exploration of how sample size affects results, we used the nonlinearly generated synthetic dataset of Fig. 2 and examined factorization as we varied N. Fig. S2 presents examples of these factorizations using the standard HPF link functions of Eq. 7 and Fig. S3 presents factorizations using the logarithmic link of Section 2.1.

For U sufficiently large, the factorizations successfully remove the irrelevant background features. However, the structure of the factors is inconsistent as U changes. Examining the correlation matrix of this dataset (Fig. S4) sheds light on this behavior. Since the true generating data for this example is dense in every third feature, these features are highly correlated. Hence, without a spare substructure to select, the factorization settles on one of the many sparse approximations to the truly dense process.

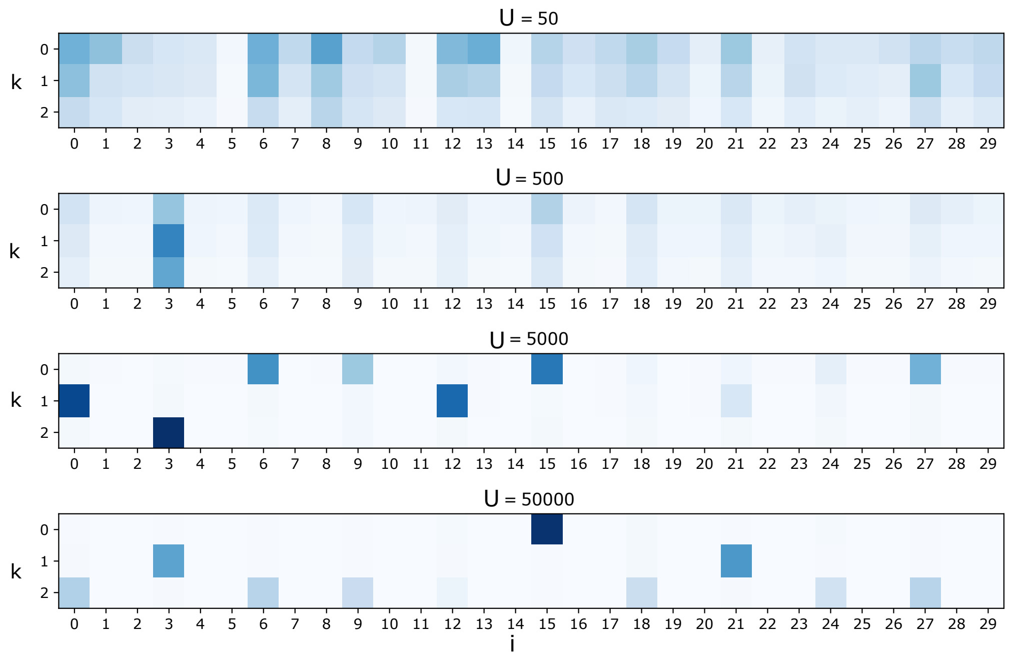

Figure S2: **Factorization under different data set sizes** of the nonlinearly generated data of Fig. 2

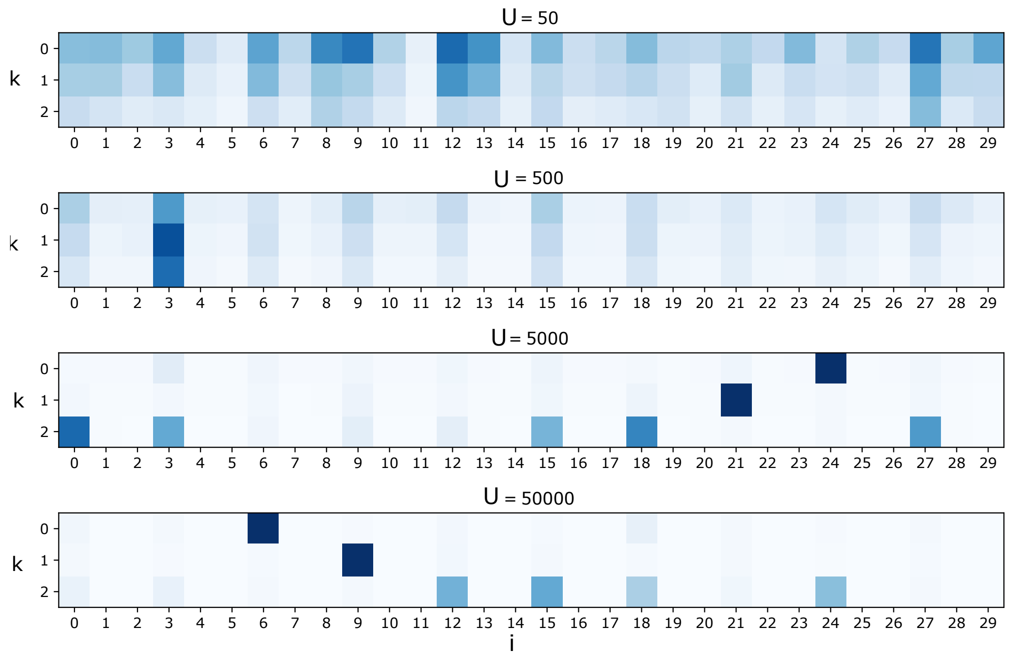

Figure S3: **Factorization under different data set sizes** of the nonlinearly generated data of Fig. 2 using the logarithmic link function.

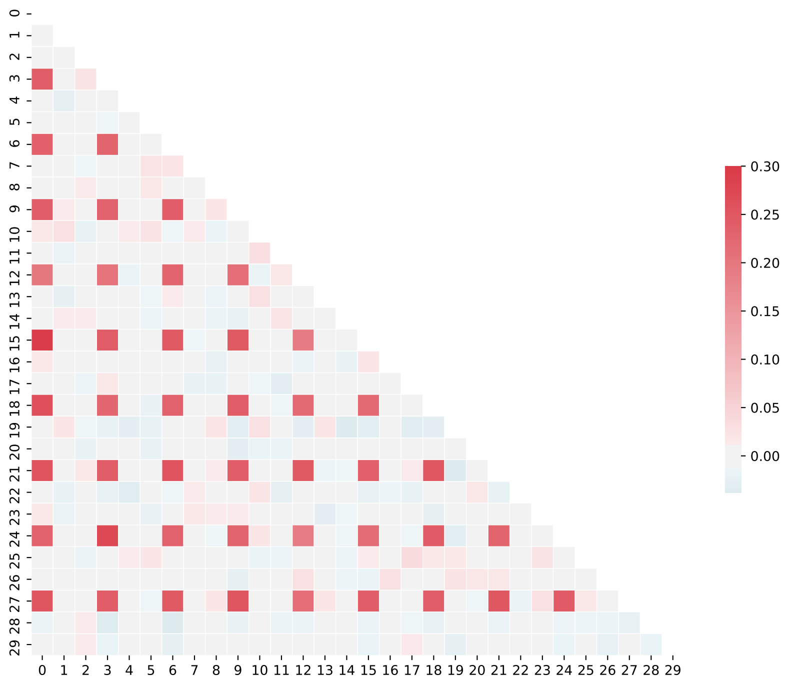

Figure S4: Correlation of the nonlinear synthetic dataset features