Learning Binary Latent Variable Models: A Tensor Eigenpair Approach

Ariel Jaffe [1] **Roi Weiss** [1] **Shai Carmi** [2] **Yuval Kluger** [3] **Boaz Nadler** [1]

Abstract

Latent variable models with hidden binary units appear in various applications. Learning such models, in particular in the presence of noise, is a challenging computational problem. In this paper we propose a novel spectral approach to this problem, based on the eigenvectors of both the second order moment matrix and third order moment tensor of the observed data. We prove that under mild non-degeneracy conditions, our method consistently estimates the model parameters at the optimal parametric rate. Our tensor-based method generalizes previous orthogonal tensor decomposition approaches, where the hidden units were assumed to be either statistically independent or mutually exclusive. We illustrate the consistency of our method on simulated data and demonstrate its usefulness in learning a common model for population mixtures in genetics.

1. Introduction

In this paper we propose a spectral method for learning the following binary latent variable model, shown in Figure 1. The hidden layer, $\boldsymbol{h} = (h_1, \ldots, h_d)$, consists of d binary random variables with an unknown joint distribution $P_{\boldsymbol{h}} : \{0,1\}^d \to [0,1]$. The observed vector $\boldsymbol{x} \in \mathbb{R}^m$ with $m \geq d$ features is modeled as

$$\boldsymbol{x} = W^\top \boldsymbol{h} + \sigma \boldsymbol{\xi}, \tag{1}$$

where $W \in \mathbb{R}^{d \times m}$ is an unknown weight matrix assumed to be full rank d. Here, $\sigma \geq 0$ is the noise level and $\boldsymbol{\xi}$ is an additive noise vector independent of $\boldsymbol{h}$, whose m coordinates are all i.i.d. zero mean and unit variance random variables. For simplicity we assume it is Gaussian, though our method can be modified to handle other noise distributions.

The model in (1) appears, for example, in overlapping clustering (Banerjee et al., 2005; Baadel et al., 2016), in various problems in bioinformatics (Segal et al., 2002; Becker et al., 2011; Slawski et al., 2013), and in blind source separation (Van der Veen, 1997). A special instance of model (1) is the Gaussian-Bernoulli restricted Boltzmann machine (G-RBM) where the distribution $P_{\boldsymbol{h}}$ is further assumed to have a parametric energy-based structure (Hinton & Salakhutdinov, 2006; Cho et al., 2011; Wang et al., 2012). G-RBMs were used, e.g., in modeling human motion (Taylor et al., 2007) and natural image patches (Melchior et al., 2017).

Given n i.i.d. samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ from model (1), the goal is to estimate the weight matrix W. A common approach for learning W is by maximum likelihood. As this function is non-convex, common optimization schemes include the EM algorithm and alternating least squares (ALS). In addition, several works developed iterative methods specialized to G-RBMs (Hinton, 2010; Cho et al., 2011). All these methods, however, often lack consistency guarantees and may not be well suited for large datasets due to their potential slow convergence. This is not surprising, as learning W under model (1) is believed to be computationally hard; see for example Mossel & Roch (2005).

Over the past years, several works considered variants and specific instances of model (1) under additional assumptions on the distribution $P_{\boldsymbol{h}}$ or on the weight matrix W. For example, when $P_{\boldsymbol{h}}$ is a product distribution, the learning problem becomes that of independent component analysis (ICA) with binary signals (Hyvärinen et al., 2004). In this case, several methods were derived for estimating W and under suitable non-degeneracy conditions were proven to be both computationally efficient and statistically consistent (Shalvi & Weinstein, 1993; Frieze et al., 1996; Regalia & Kofidis, 2003; Hyvärinen et al., 2004; Anandkumar et al., 2014; Jain & Oh, 2014). Similarly, when the hidden units are mutually exclusive, namely $P_{\boldsymbol{h}}$ has support $\boldsymbol{h} \in \{\boldsymbol{e}_i\}_{i=1}^d$, the model is a Gaussian mixture (GMM) with d spherical components with linearly independent means. Efficient and consistent algorithms were derived for this case as well (Moitra & Valiant, 2010; Anandkumar et al., 2012a;b; Hsu & Kakade, 2013). Among those, most relevant to this work

[1] Dept. of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot 7610001, Israel. [2] Braun School of Public Health and Community Medicine, The Hebrew University of Jerusalem, Jerusalem 9112102, Israel. [3] Program of Applied Mathematics, Yale University, New Haven, CT 06511, USA. Correspondence to: Ariel Jaffe <ariel.jaffe@weizmann.ac.il>, Roi Weiss <roi.weiss@weizmann.ac.il>.

Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

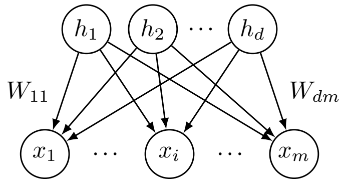

Figure 1. The binary latent variable model.

are orthogonal tensor decomposition methods (Anandkumar et al., 2014). Interestingly, these methods can learn some additional latent models, with hidden units that are not necessarily binary, such as Dirichlet allocation and other correlated topic models (Arabshahi & Anandkumar, 2017).

Learning W given the observed data $\{\boldsymbol{x}_j\}_{j=1}^n$ can also be viewed as a noisy *matrix factorization* problem. If W is known to be non-negative, then various non-negative matrix factorization methods can be used. Moreover, under appropriate conditions, some of these methods were proven to be computationally efficient and consistent (Donoho & Stodden, 2004; Arora et al., 2012). For general full rank W, the matrix factorization method in Slawski et al. (2013) (SHL) exactly recovers W when $\sigma = 0$ with a runtime exponential in d. This method, however, can handle only low levels of noise and has no consistency guarantees when $\sigma > 0$.

A tensor eigenpair approach In this paper we propose a novel spectral method for learning W which is based on the eigenvectors of both the second order moment matrix and the third order moment tensor of the observed data. We prove that our method is consistent under mild non-degeneracy conditions and achieves the parametric rate $O_P(n^{-\frac{1}{2}})$ for any noise level $\sigma \geq 0$.

The non-degeneracy conditions we pose are significantly weaker than those required by the previous tensor decomposition methods mentioned above. In particular, their assumptions and resulting methods can be viewed as specific cases of our more general approach.

Similarly to the matrix factorization method in Slawski et al. (2013), our algorithm has runtime linear in n, polynomial in m, and in general exponential in d. With our current Matlab implementation, most of the runtime is spent on computing the eigenpairs of a $d \times d \times d$ tensor. Practically, our method, implemented without any particular optimization, can learn a model with 12 hidden units in less than ten minutes on a standard PC. Furthermore, the overall runtime can be significantly reduced, since the step of computing the tensor eigenpairs can be embarrassingly parallelized.

Paper outline In the next section we give necessary background on tensor eigenpairs. In Section 3 we introduce our method in the case $\sigma = 0$. The case $\sigma \geq 0$ is treated in Section 4. Experiments with our method and comparison to other approaches appear in Section 5. All proofs are deferred to the supplementary material.

2. Preliminaries

Notation Denote $[d] = \{1, \dots, d\}$ and $\boldsymbol{e}_i$ as the i-th unit vector. We slightly abuse notation and view a matrix W also as the set of its columns, namely $\boldsymbol{w} \in W$ is some column of W and $\operatorname{span}(W)$ is the span of all its columns. The unit sphere is denoted by $\mathbb{S}_{d-1} = \{\boldsymbol{u} \in \mathbb{R}^d : \|\boldsymbol{u}\| = 1\}$.

A tensor $\mathcal{T} \in \mathbb{R}^{d\times d\times d}$ is symmetric if $\mathcal{T}_{ijk} = \mathcal{T}_{\pi(i,j,k)}$ for all permutations π of i, j, k. Here, we consider only symmetric tensors. $\mathcal{T}$ can also be seen as a *multi-linear* operator: for matrices W^1, W^2, W^3 with $W^i \in \mathbb{R}^{d\times d_i}$, the tensor-mode product, denoted $\mathcal{T}(W^1, W^2, W^3)$, is a $d_1 \times d_2 \times d_3$ tensor whose (i_1, i_2, i_3)-th entry is

$$\sum_{j_1,j_2,j_3\in[d]} W^1_{j_1 i_1} W^2_{j_2 i_2} W^3_{j_3 i_3} \mathcal{T}_{j_1 j_2 j_3}.$$

Tensor eigenpairs Several types of eigenpairs of a tensor have been proposed. Here, we consider the following definition, termed Z-eigenpairs by Qi (2005) and l_2-eigenpairs by Lim (2005). Henceforth we just call them eigenpairs.

Definition 1. *$(\boldsymbol{u}, \lambda) \in \mathbb{R}^d \times \mathbb{R}$ is an eigenpair of $\mathcal{T}$ if*

$$\mathcal{T}(I, \boldsymbol{u}, \boldsymbol{u}) = \lambda \boldsymbol{u} \quad \textit{and} \quad \|\boldsymbol{u}\| = 1. \tag{2}$$

Note that if $(\boldsymbol{u}, \lambda)$ is an eigenpair then the eigenvalue is simply $\lambda = \mathcal{T}(\boldsymbol{u}, \boldsymbol{u}, \boldsymbol{u})$. In addition, $(-\boldsymbol{u}, -\lambda)$ is also an eigenpair. Following common practice we treat these two pairs as one and make the convention that $\lambda \geq 0$.

In contrast to the matrix case, the number of eigenvalues $\{\lambda\}$ of a tensor $\mathcal{T} \in \mathbb{R}^{d\times d\times d}$ can be much larger than d. As shown by Cartwright & Sturmfels (2013), for a $d \times d \times d$ tensor, there can be at most $2^d - 1$ of them. With precise definitions appearing in Cartwright & Sturmfels (2013), for a *generic* tensor, all its eigenvalues have multiplicity one and the number of eigenpairs $\{(\boldsymbol{u}, \lambda)\}$ is at most $2^d - 1$.

In principle, enumerating the set of all eigenpairs of a general symmetric tensor is a #P problem (Hillar & Lim, 2013). Nevertheless, several methods have been proposed for computing at least *some* eigenpairs, including iterative higher-order power methods (Kolda & Mayo, 2011; 2014), homotopy continuation (Chen et al., 2016), semidefinite programming (Cui et al., 2014), and iterative Newton-based methods (Jaffe et al., 2017; Guo et al., 2017). We conclude this section with the definition of *Newton-stable* eigenpairs (Jaffe et al., 2017) which are most relevant to our work.

Newton-stable eigenpairs Equivalently to (2), eigenpairs of $\mathcal{T}$ can also be characterized by the function $\boldsymbol{g} : \mathbb{R}^d \to \mathbb{R}^d$,

$$\boldsymbol{g}(\boldsymbol{u}) = \mathcal{T}(I, \boldsymbol{u}, \boldsymbol{u}) - \mathcal{T}(\boldsymbol{u}, \boldsymbol{u}, \boldsymbol{u}) \cdot \boldsymbol{u}. \tag{3}$$

It is easy to verify that a pair $(\boldsymbol{u}, \lambda)$ with $\|\boldsymbol{u}\| = 1$ is an eigenpair of $\mathcal{T}$ if and only if $\boldsymbol{g}(\boldsymbol{u}) = \mathbf{0}$ and $\lambda = \mathcal{T}(\boldsymbol{u}, \boldsymbol{u}, \boldsymbol{u})$. The stability of an eigenpair is determined by its Jacobian matrix $\nabla \boldsymbol{g}(\boldsymbol{u}) \in \mathbb{R}^{d\times d}$, more precisely, by its projection into the $d-1$ dimensional subspace orthogonal to $\boldsymbol{u}$. Formally, let $L_{\boldsymbol{u}} \in \mathbb{R}^{d\times(d-1)}$ be a matrix with $d-1$ orthonormal columns that span the subspace orthogonal to $\boldsymbol{u}$ and define the $(d-1)\times(d-1)$ projected Jacobian matrix

$$J_p(\boldsymbol{u}) = L_{\boldsymbol{u}}^\top \nabla \boldsymbol{g}(\boldsymbol{u}) L_{\boldsymbol{u}}. \tag{4}$$

Definition 2. *An eigenpair $(\boldsymbol{u}, \lambda)$ of $\mathcal{T} \in \mathbb{R}^{d\times d\times d}$ is Newton-stable if the matrix $J_p(\boldsymbol{u})$ has full rank $d-1$.*

The homotopy continuation method in Chen et al. (2016) is guaranteed to compute all the Newton-stable eigenpairs of a tensor. Alternatively, all the Newton-stable eigenpairs can be computed by the iterative orthogonal Newton correction method (O–NCM) in Jaffe et al. (2017) as these are the attracting fixed points for this algorithm. Moreover, O–NCM converges to any Newton-stable eigenpair at a quadratic rate given a sufficiently close initial guess. Finally, for a generic tensor, all its eigenpairs are Newton-stable.

3. Learning in the noiseless case

To motivate our approach for estimating the matrix W it is instructive to first consider the ideal noiseless case where $\sigma = 0$. In this case, model (1) takes the form $\boldsymbol{x} = W^\top \boldsymbol{h}$. Our problem then becomes that of factorizing the observed matrix $X = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{m\times n}$ of n samples into a product of real and binary low-rank matrices,[1]

$$\text{Find} \quad W \in \mathbb{R}^{d\times m}, H \in \{0,1\}^{d\times n} \quad \text{s.t.} \quad X = W^\top H. \tag{5}$$

To be able to recover W we first need conditions under which the decomposition of X into W and H is unique. Clearly, such a factorization can be unique at most up to a permutation of its components; we henceforth ignore this degeneracy. A sufficient condition for uniqueness, similar to the one posed in Slawski et al. (2013), is that H is *rigid*. Formally, $H \in \{0,1\}^{d\times n}$ is rigid if any non-trivial linear combination of its rows yields a non-binary vector: $\forall \boldsymbol{u} \neq \mathbf{0}$,

$$\boldsymbol{u}^\top H \in \{0,1\}^n \quad \Leftrightarrow \quad \boldsymbol{u} \in \{\boldsymbol{e}_i\}_{i=1}^d. \tag{6}$$

Condition (6) is satisfied, for example, when the columns of H include $\boldsymbol{e}_i$ and $\boldsymbol{e}_i + \boldsymbol{e}_j$ for all $i \neq j \in [d]$. If there exists a positive constant $p_0 > 0$ such that $P_{\boldsymbol{h}}(\boldsymbol{e}_i) \geq p_0$ and $P_{\boldsymbol{h}}(\boldsymbol{e}_i + \boldsymbol{e}_j) \geq p_0$, then for a sample size $n > 2\log(d)/p_0$ the matrix H is rigid with high probability.

The following proposition, similar in nature to the (affine constrained) uniqueness guarantee in Slawski et al. (2013), shows that under condition (6) the factorization in (5) is unique and fully characterized by the binary constraints.

Proposition 1. *Let $X = W^\top H$ with $H \in \{0,1\}^{d\times n}$ rigid and $W \in \mathbb{R}^{d\times m}$ full rank with $m \geq d$. Let $W^\dagger \in \mathbb{R}^{m\times d}$ be the unique right pseudo-inverse of W so $WW^\dagger = I_d$. Then W and H are unique and for all $\boldsymbol{v} \in \operatorname{span}(X) \setminus \{\mathbf{0}\}$,*

$$\boldsymbol{v}^\top X \in \{0,1\}^n \quad \Leftrightarrow \quad \boldsymbol{v} \in W^\dagger. \tag{7}$$

Hence, under the rigidity condition (6), the matrix factorization problem in (5) is equivalent to the problem of finding the *unique* set $W^\dagger = \{\boldsymbol{v}_1^*, \ldots, \boldsymbol{v}_d^*\} \subseteq \operatorname{span}(X)$ of d non-zero vectors that satisfy the binary constraints $\boldsymbol{v}_i^{*\top} X \in \{0,1\}^n$. The weight matrix is then $W = (W^\dagger)^\dagger$.

Algorithm outline We recover $W^\dagger$ via a two step procedure. First, a finite set $V = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots\} \subseteq \operatorname{span}(X)$ of *candidate* vectors is computed with a guarantee that $W^\dagger \subseteq V$. Specifically, V is computed from the set of eigenpairs of a $d \times d \times d$ tensor, constructed from the low order moments of X. Typically, the size of V will be much larger than d, so in the second step V is *filtered* by selecting all $\boldsymbol{v} \in V$ that satisfy $\boldsymbol{v}^\top X \in \{0,1\}^n$.

Before describing the two steps in more detail we first state the additional non-degeneracy conditions we pose. To this end, denote the unknown first, second, and third order moments of the latent binary vector $\boldsymbol{h}$ by

$$\begin{aligned} &\boldsymbol{p} = \mathbb{E}[\boldsymbol{h}] \in \mathbb{R}^d, \qquad C = \mathbb{E}[\boldsymbol{h} \otimes \boldsymbol{h}] \in \mathbb{R}^{d\times d}, \\ &\mathcal{C} = \mathbb{E}[\boldsymbol{h} \otimes \boldsymbol{h} \otimes \boldsymbol{h}] \in \mathbb{R}^{d\times d\times d}. \end{aligned} \tag{8}$$

Non-degeneracy conditions We assume the following:

(I) H is rigid.

(II) $\operatorname{rank}(2\mathcal{C}(I, I, \boldsymbol{e}_i) - C) = d$ for all $i \in [d]$.

Condition *(I)* implies that both $\operatorname{rank}(HH^\top) = d$ and $\operatorname{rank}(C) = d$. This in turn implies $p_i = \mathbb{E}[h_i] > 0$ for all $i \in [d]$ and that at most one variable h_i has $p_i = 1$. Such an "always on" variable can model a fixed bias to $\boldsymbol{x}$. As far as we know, condition *(II)* is new and its nature will become clear shortly.

We now describe each step of our algorithm in more detail.

Computing the candidate set To compute a set V that is guaranteed to include the columns of $W^\dagger$ we make use of

[1] Note that this is different from the problem known as "Boolean matrix factorization", where X and W are assumed to be binary as well; see Miettinen & Vreeken (2014) and references therein.

the second and third order moments of $\boldsymbol{x}$,

$$\begin{aligned} M &= \mathbb{E}[\boldsymbol{x} \otimes \boldsymbol{x}] \in \mathbb{R}^{m \times m}, \\ \mathcal{M} &= \mathbb{E}[\boldsymbol{x} \otimes \boldsymbol{x} \otimes \boldsymbol{x}] \in \mathbb{R}^{m \times m \times m}. \end{aligned} \tag{9}$$

Given a large number of samples $n \gg 1$, these can be easily and accurately estimated from the sample X. For simplicity, in this section we consider the population setting where $n \to \infty$, so M and $\mathcal{M}$ are known exactly. M and $\mathcal{M}$ are related to the unknown second and third order moments of $\boldsymbol{h}$ in (8) via (Anandkumar et al., 2014)

$$M = W^\top C W, \qquad \mathcal{M} = \mathcal{C}(W, W, W). \tag{10}$$

Since both C and W are full rank, the number of latent units can be deduced by $\operatorname{rank}(M) = d$. Since C is positive definite, there is a whitening matrix $K \in \mathbb{R}^{m \times d}$ such that

$$K^\top M K = I_d. \tag{11}$$

Such a K can be computed, for example, by an eigendecomposition of M. Although K is not unique, any $K \subseteq \operatorname{span}(M)$ that satisfies (11) suffices for our purpose. Define the $d \times d \times d$ lower dimensional whitened tensor

$$\mathcal{W} = \mathcal{M}(K, K, K). \tag{12}$$

Denote the set of *eigenpairs* of $\mathcal{W}$ by

$$U = \{(\boldsymbol{u}, \lambda) \in \mathbb{S}_{d-1} \times \mathbb{R}_+ : \mathcal{W}(I, \boldsymbol{u}, \boldsymbol{u}) = \lambda \boldsymbol{u}\}. \tag{13}$$

Our set of candidates is then

$$V = \{K\boldsymbol{u}/\lambda : (\boldsymbol{u}, \lambda) \in U \text{ with } \lambda \geq 1\} \subseteq \mathbb{R}^m. \tag{14}$$

The following lemma shows that under condition *(I)* the set V is guaranteed to contain the d columns of $W^\dagger$.

Lemma 1. *Let $\mathcal{W}$ be the tensor in (12) corresponding to model (1) with $\sigma = 0$ and let V be as in (14). If condition (I) holds then $W^\dagger \subseteq V$. In particular, each $(\boldsymbol{u}_i, \lambda_i)$ in the set of d relevant eigenpairs*

$$U^* = \{(\boldsymbol{u}, \lambda) \in U : K\boldsymbol{u}/\lambda \in W^\dagger\} \tag{15}$$

has the eigenvalue $\lambda_i = 1/\sqrt{p_i} \geq 1$ where $p_i = \mathbb{E}[h_i] > 0$.

Computing the tensor eigenpairs By Lemma 1, we may construct a candidate set V that contains $W^\dagger$ by first calculating the set U of eigenpairs of $\mathcal{W}$. Unfortunately, computing the set of all eigenpairs of a general symmetric tensor is computationally hard (Hillar & Lim, 2013). Moreover, besides the d columns of $W^\dagger$, the set V in (14) may contain many spurious candidates, as the number of eigenpairs of $\mathcal{W}$ is typically $O(2^d) \gg d$ (Cartwright & Sturmfels, 2013).

Nevertheless, as discussed in Section 2, several methods have been proposed for computing *some* eigenpairs of a tensor under appropriate stability conditions. The following lemma highlights the importance of condition *(II)* for the stability of the eigenpairs in U^*. Note that conditions *(I)-(II)* do not depend on W, but only on the distribution of $\boldsymbol{h}$.

Lemma 2. *Let $\mathcal{W}$ be the whitened tensor in (12) corresponding to model (1) with $\sigma = 0$. If conditions (I)-(II) hold, then all $(\boldsymbol{u}, \lambda) \in U^*$ are Newton-stable eigenpairs of $\mathcal{W}$.*

Hence, under conditions *(I)-(II)*, the homotopy method in Chen et al. (2016), or alternatively the O–NCM with a sufficiently large number of random initializations (Jaffe et al., 2017), are guaranteed to compute a candidate set V which includes all the columns of $W^\dagger$. The next step is to extract $W^\dagger$ out of V.

Filtering As suggested by Eq. (7) we select the subset of vectors $\bar{V} \subseteq V$ that satisfy the binary constraints,

$$\bar{V} = \{\boldsymbol{v} \in V : \boldsymbol{v}^T X \in \{0, 1\}^n\}. \tag{16}$$

Indeed, under condition (I), Proposition 1 implies that $\bar{V} = W^\dagger$ and the weight matrix is thus $W = \bar{V}^\dagger$.

Algorithm 2 in Appendix C summarizes our method for the noiseless case and has the following recovery guarantee.

Theorem 1. *Let X be a matrix of n samples from model (1) with $\sigma = 0$. If conditions (I)-(II) hold, then the above method recovers W exactly.*

We note that when $\sigma = 0$ and conditions *(I)-(II)* hold for the *empirical* latent moments $\hat{C}$ and $\hat{\mathcal{C}}$ (rather than C and $\mathcal{C}$), the above procedure *exactly* recovers W when M and $\mathcal{M}$ are replaced by their finite sample estimates. The matrix factorization method SHL in Slawski et al. (2013) also exactly recovers W in the case $\sigma = 0$. While its runtime is also exponential in d, practically it may be much faster than our proposed tensor based approach. This is because SHL constructs a candidate set of size 2^d that can be computed by a suitable linear transformation of the *fixed* set $\{0, 1\}^d$, as opposed to our candidate set which is constructed by eigenpairs of a $d \times d \times d$ tensor. However, SHL does not take advantage of the large number of samples n, since only $m \times d$ sub-matrices of the $m \times n$ sample matrix X are used for constructing its candidate set. Indeed, in the noisy case where $\sigma > 0$, SHL has no consistency guarantees and as demonstrated by the simulation results in Section 5 it may fail at high levels of noise. In the next section we derive a modified version of our method that consistently estimates W for any noise level $\sigma \geq 0$.

4. Learning in the presence of noise

The method in Section 3 to estimate W is clearly inadequate when $\sigma > 0$. However, we now show that by making several adjustments, the two steps of computing the candidate set and its filtering can be both made robust to noise, yielding a consistent estimator of W for any $\sigma \geq 0$.

Computing the candidate set As in the case $\sigma = 0$, our goal in the first step is to compute a finite candidate set

$V_\sigma \subseteq \mathbb{R}^m$ that is guaranteed to contain accurate estimates for the d columns of $W^\dagger$. To this end, in addition to the second and third order moments M and $\mathcal{M}$ in (9), we also consider the first order moment $\boldsymbol{\mu} = \mathbb{E}[\boldsymbol{x}]$ and define the following noise corrected moments,

$$\begin{aligned} M_\sigma &= M - \sigma^2 I_m, \\ \mathcal{M}_\sigma &= \mathcal{M} - \sigma^2 \sum_{i=1}^m \Big(\boldsymbol{\mu} \otimes \boldsymbol{e}_i \otimes \boldsymbol{e}_i \\ &\qquad + \boldsymbol{e}_i \otimes \boldsymbol{\mu} \otimes \boldsymbol{e}_i + \boldsymbol{e}_i \otimes \boldsymbol{e}_i \otimes \boldsymbol{\mu} \Big). \end{aligned} \tag{17}$$

By assumption, the noise satisfies $\mathbb{E}[\xi_i^3] = 0$. Thus, similarly to the moment equations in (10), the modified moments in (17) are related to these of $\boldsymbol{h}$ by (Anandkumar et al., 2014)

$$M_\sigma = W^\top C W, \qquad \mathcal{M}_\sigma = \mathcal{C}(W, W, W). \tag{18}$$

Hence, if M_σ and $\mathcal{M}_\sigma$ were known exactly, a candidate set V_σ that *contains* $W^\dagger$ could be obtained exactly as in the noiseless case, but with M and $\mathcal{M}$ replaced with M_σ and $\mathcal{M}_\sigma$; namely, first calculate the whitening matrix K_σ such that $K_\sigma^\top M_\sigma K_\sigma = I_d$ and then compute the eigenpairs of the population whitened tensor

$$\mathcal{W}_\sigma = \mathcal{M}_\sigma(K_\sigma, K_\sigma, K_\sigma). \tag{19}$$

In practice, σ^2, d, $\boldsymbol{\mu}$, M and $\mathcal{M}$ are all unknown and need to be estimated from the sample matrix X. Assuming $m > d$, the parameters σ^2 and d can be consistently estimated, for example, by the methods in Kritchman & Nadler (2009). For simplicity, we assume they are known exactly. Similarly, $\boldsymbol{\mu}$, M, $\mathcal{M}$ are consistently estimated by their empirical means, $\hat{\boldsymbol{\mu}}$, $\hat{M}$, and $\hat{\mathcal{M}}$. So, after computing the plugin estimates $\hat{K}_\sigma$ such that $\hat{K}_\sigma^\top \hat{M}_\sigma \hat{K}_\sigma = I_d$ and $\hat{\mathcal{W}}_\sigma = \hat{\mathcal{M}}_\sigma(\hat{K}_\sigma, \hat{K}_\sigma, \hat{K}_\sigma)$, we compute the set $\hat{U}_\sigma$ of eigenpairs of $\hat{\mathcal{W}}_\sigma$ and for some small $0 < \tau = O(n^{-\frac{1}{2}})$ take our candidate set as

$$\hat{V}_\sigma = \{\hat{K}_\sigma \boldsymbol{u}/\lambda : (\boldsymbol{u}, \lambda) \in \hat{U}_\sigma \text{ with } \lambda \geq 1 - \tau\}. \tag{20}$$

The following lemma shows that under conditions *(I)-(II)* the above procedure is stable to small perturbations. Namely, for perturbations of order $\delta \ll 1$ in $\mathcal{W}_\sigma$ and K_σ, the method computes a candidate set $\hat{V}_\sigma$ that contains a subset of d vectors that are $O(\delta)$ close to the columns of $W^\dagger$. Furthermore, these d vectors all correspond to Newton-stable eigenpairs of the perturbed tensor and are $\Omega(1)$ separated from the other candidates in $\hat{V}_\sigma$.

Lemma 3. *Let $K_\sigma, \mathcal{W}_\sigma$ be the population quantities in (19) and let $\hat{K}_\sigma, \hat{\mathcal{W}}_\sigma$ be their perturbed versions, inducing the candidate set $\hat{V}_\sigma$ in (20). If conditions (I)-(II) hold, then there are $c, \delta_0, \delta_1 > 0$ such that for all $0 \leq \delta \leq \delta_0$ the following holds: If the perturbed versions satisfy*

$$\max\{\|\hat{\mathcal{W}}_\sigma - \mathcal{W}_\sigma\|_F, \|\hat{K}_\sigma - K_\sigma\|_F\} \leq \delta, \tag{21}$$

then any $\boldsymbol{v}^* \in W^\dagger$ has a unique $\hat{\boldsymbol{v}} \in \hat{V}_\sigma$ such that

$$\|\hat{\boldsymbol{v}} - \boldsymbol{v}^*\| \leq c\delta. \tag{22}$$

Moreover, $\hat{\boldsymbol{v}}$ corresponds to a Newton-stable eigenpair of $\hat{\mathcal{W}}_\sigma$ with eigenvalue $\lambda \geq 1 - c\delta$ and for all $\tilde{\boldsymbol{v}} \in \hat{V}_\sigma \setminus \{\hat{\boldsymbol{v}}\}$,

$$\|\tilde{\boldsymbol{v}} - \boldsymbol{v}^*\| \geq \delta_1 > 2c\delta. \tag{23}$$

The proof is based on the implicit function theorem (Hubbard & Hubbard, 2015); small perturbations to a tensor result in small perturbations to its Newton-stable eigenpairs.

Now, by the delta method, the plugin estimates $\hat{K}_\sigma$ and $\hat{\mathcal{W}}_\sigma$ are both $O_P(n^{-\frac{1}{2}})$ close to their population quantities,

$$\begin{aligned} \|\hat{K}_\sigma - K_\sigma\|_F &= O_P(n^{-\frac{1}{2}}), \\ \|\hat{\mathcal{W}}_\sigma - \mathcal{W}_\sigma\|_F &= O_P(n^{-\frac{1}{2}}). \end{aligned} \tag{24}$$

By (24), we have that (21) holds with $\delta = O_P(n^{-\frac{1}{2}})$. Hence, by Lemma 3, the eigenpairs of $\hat{\mathcal{W}}_\sigma$ provide a candidate set $\hat{V}_\sigma$ that contains d vectors that are $O_P(n^{-\frac{1}{2}})$ close to the columns of $W^\dagger$. In addition, any irrelevant candidate is $\Omega_P(1)$ far away from $W^\dagger$. As we show next, these properties ensure that with high probability the d relevant candidates can be identified in $\hat{V}_\sigma$.

Filtering Given the candidate set $\hat{V}_\sigma$ computed in the first step, our goal now is to find a set $\bar{V}_\sigma \subseteq \hat{V}_\sigma$ of d vectors that accurately estimate the d columns of $W^\dagger$. To simplify the theoretical analysis, we assume the filtering step is done using a sample X of size n that is independent of $\hat{V}_\sigma$. This can be achieved by first splitting a given sample of size $2n$ into two sets of size n, one for each step.

Recall that for $\boldsymbol{x}$ from model (1) and any $\boldsymbol{v} \in \mathbb{R}^m$,

$$\boldsymbol{v}^\top \boldsymbol{x} = \boldsymbol{v}^\top W^\top \boldsymbol{h} + \sigma \boldsymbol{v}^\top \boldsymbol{\xi}. \tag{25}$$

Obviously, when $\sigma > 0$, the filtering procedure in (16) for the noiseless case is inadequate, as typically no $\boldsymbol{v}^* \in W^\dagger$ will exactly satisfy $\boldsymbol{v}^{*\top} X \in \{0,1\}^n$. Nevertheless, we expect that for a sufficiently small noise level σ, any $\boldsymbol{v} \in \hat{V}_\sigma$ that is close to some $\boldsymbol{v}^* \in W^\dagger$ will result in $\boldsymbol{v}^\top X$ that is close to being binary, while any $\boldsymbol{v}$ sufficiently far from $W^\dagger$ will result in $\boldsymbol{v}^\top X$ that is far from being binary. A natural measure for how $\boldsymbol{v}^\top X$ is "far from being binary", similar to the one used for filtering in Slawski et al. (2013), is simply its deviation from its binary rounding,

$$\min_{\boldsymbol{b} \in \{0,1\}^n} \frac{\|\boldsymbol{v}^T X - \boldsymbol{b}\|^2}{n\|\boldsymbol{v}\|^2}. \tag{26}$$

Eq. (26) works extremely well for small σ, but fails for high noise levels. Here we instead propose a filtering procedure based on the classical Kolmogorov-Smirnov goodness of fit test (Lehmann & Romano, 2006). As we show below, this approach gives consistent estimates of W for any $\sigma > 0$.

Before describing the test, we first introduce the probabilistic analogue of the rigidity condition (6). For any $\boldsymbol{u} \in \mathbb{R}^d$, define its corresponding expected binary rounding error,

$$r(\boldsymbol{u}) = \mathbb{E}_{\boldsymbol{h}\sim P_{\boldsymbol{h}}}\left[\min_{b\in\{0,1\}} (\boldsymbol{u}^\top \boldsymbol{h} - b)^2\right].$$

Clearly, $r(\boldsymbol{0}) = 0$ and $r(\boldsymbol{e}_i) = 0$ for all $i \in [d]$. We pose the following *expected rigidity* condition: for all $\boldsymbol{u} \neq \boldsymbol{0}$,

$$r(\boldsymbol{u}) = 0 \quad \Leftrightarrow \quad \boldsymbol{u} \in \{\boldsymbol{e}_i\}_{i=1}^d. \tag{27}$$

Analogously to the deterministic rigidity condition in (6), condition (27) is satisfied, for example, when $P_{\boldsymbol{h}}(\boldsymbol{e}_i) > 0$ and $P_{\boldsymbol{h}}(\boldsymbol{e}_i + \boldsymbol{e}_j) > 0$ for all $i \neq j \in [d]$.

To introduce our filtering test, recall that under model (1), $\boldsymbol{\xi} \sim \mathcal{N}(\boldsymbol{0}, I_m)$. Hence, for any fixed $\boldsymbol{v}$, the random variable $\boldsymbol{v}^\top \boldsymbol{x}$ in (25) is distributed according to the following univariate Gaussian mixture model (GMM),

$$\boldsymbol{v}^\top \boldsymbol{x} \sim \sum_{\boldsymbol{h}\in\{0,1\}^d} P_{\boldsymbol{h}}(\boldsymbol{h}) \cdot \mathcal{N}(\boldsymbol{v}^\top W^\top \boldsymbol{h}, \sigma^2 \|\boldsymbol{v}\|^2). \tag{28}$$

Denote the cumulative distribution function of $\boldsymbol{v}^\top \boldsymbol{x}$ by $F_{\boldsymbol{v}}$. For general $\boldsymbol{v}$, this mixture may have up to 2^d distinct components. However, for $\boldsymbol{v}^* \in W^\dagger$, it reduces to a mixture of *two* components with means at 0 and 1. More precisely, for any candidate $\boldsymbol{v}$ with corresponding eigenvalue $\lambda(\boldsymbol{v}) \geq 1$, define the GMM with two components

$$(1 - \tfrac{1}{\lambda(\boldsymbol{v})^2}) \cdot \mathcal{N}(0, \sigma^2 \|\boldsymbol{v}\|^2) + \tfrac{1}{\lambda(\boldsymbol{v})^2} \cdot \mathcal{N}(1, \sigma^2 \|\boldsymbol{v}\|^2). \tag{29}$$

Denote its cumulative distribution function by $G_{\boldsymbol{v}}$. The following lemma shows that under condition (27), $G_{\boldsymbol{v}}$ fully characterizes the columns of $W^\dagger$.

Lemma 4. *Let $K_\sigma, \mathcal{W}_\sigma$ be the population quantities in (19) and let V_σ be the set of population candidates as computed from the eigenpairs of $\mathcal{W}_\sigma$. If conditions (I)-(II) and the expected rigidity condition (27) hold, then for any $\boldsymbol{v} \in V_\sigma$ with corresponding eigenvalue $\lambda(\boldsymbol{v})$,*

$$F_{\boldsymbol{v}} = G_{\boldsymbol{v}} \quad \Leftrightarrow \quad \boldsymbol{v} \in W^\dagger.$$

Given the empirical candidate set $\hat{V}_\sigma$, Lemma 4 suggests ranking all $\hat{\boldsymbol{v}} \in \hat{V}_\sigma$ according to their goodness of fit to $G_{\hat{\boldsymbol{v}}}$ and taking the d candidates with the best fit. More precisely, given a sample $X = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$ that is independent of $\hat{V}_\sigma$, for each candidate $\hat{\boldsymbol{v}} \in \hat{V}_\sigma$ we compute the empirical cumulative distribution function, $\hat{F}_{\hat{\boldsymbol{v}}}(t) = \frac{1}{n}\sum_{j=1}^n \mathbb{1}\{\hat{\boldsymbol{v}}^\top \boldsymbol{x}_j \leq t\}, t \in \mathbb{R}$, and calculate its Kolmogorov-Smirnov score

$$\Delta_n(\hat{\boldsymbol{v}}) = \sup_{t\in\mathbb{R}} |\hat{F}_{\hat{\boldsymbol{v}}}(t) - G_{\hat{\boldsymbol{v}}}(t)|. \tag{30}$$

Algorithm 1 Estimate W when $\sigma > 0$ and $n < \infty$

```
Input: sample matrix $X \in \mathbb{R}^{m\times n}$ and $0 < \tau \ll 1$
estimate number of hidden units $d$ and noise level $\sigma^2$
compute empirical moments $\hat{\boldsymbol{\mu}}$, $\hat{M}$ and $\hat{\mathcal{M}}$ and plugin moments $\hat{M}_\sigma$ and $\hat{\mathcal{M}}_\sigma$ of (17)
compute $\hat{K}_\sigma$ such that $\hat{K}_\sigma^\top \hat{M}_\sigma \hat{K}_\sigma = I_d$
construct $\hat{\mathcal{W}}_\sigma = \hat{\mathcal{M}}_\sigma(\hat{K}_\sigma, \hat{K}_\sigma, \hat{K}_\sigma)$
compute the set $\hat{U}_\sigma$ of eigenpairs of $\hat{\mathcal{W}}_\sigma$
compute the candidate set $\hat{V}_\sigma$ in (20)
for each $\hat{\boldsymbol{v}} \in \hat{V}_\sigma$ compute its KS score $\Delta_n(\hat{\boldsymbol{v}})$ in (30)
select $\bar{V}_\sigma \subseteq \hat{V}_\sigma$ of $d$ vectors with smallest $\Delta_n(\hat{\boldsymbol{v}})$
return the pseudo-inverse $\hat{W} = \bar{V}_\sigma^\dagger$
```

Our estimator $\bar{V}_\sigma \subseteq \hat{V}_\sigma$ for $W^\dagger$ is then the set of d vectors with the smallest scores $\Delta_n(\hat{\boldsymbol{v}})$. The estimator for W is the pseudo-inverse, $\hat{W} = \bar{V}_\sigma^\dagger$.

The following lemma shows that for sufficiently large n, $\Delta_n(\hat{\boldsymbol{v}})$ accurately distinguishes between $\hat{\boldsymbol{v}} \in \hat{V}_\sigma$ that are close to the columns of $W^\dagger$ from these that are not.

Lemma 5. *Let $\boldsymbol{v}^* \in W^\dagger$ and $\hat{\boldsymbol{v}}_{(1)}, \hat{\boldsymbol{v}}_{(2)}, \ldots$ a sequence of random vectors such that $\|\hat{\boldsymbol{v}}_{(n)} - \boldsymbol{v}^*\| = O_P(n^{-\frac{1}{2}})$. Then, $\Delta_n(\hat{\boldsymbol{v}}_{(n)}) = o_P(1)$. In contrast, if $\min_{\boldsymbol{v}^*\in W^\dagger} \|\hat{\boldsymbol{v}}_{(n)} - \boldsymbol{v}^*\| = \Omega_P(1)$, then $\Delta_n(\hat{\boldsymbol{v}}_{(n)}) = \Omega_P(1)$, provided the expected rigidity condition (27) holds.*

Lemma 5 follows from classical and well studied properties of the Kolmogorov-Smirnov test, see for example Lehmann & Romano (2006); Billingsley (2013).

Algorithm 1 summarizes our method for estimating W in the general case where $\sigma > 0$ and $n < \infty$. The following theorem establishes its consistency.

Theorem 2. *Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ be n i.i.d. samples from model (1). If conditions (I)-(II) and the expected rigidity condition (27) hold, then the estimator $\hat{W}$ computed by Algorithm 1 is consistent, achieving the parametric rate,*

$$\hat{W} = W + O_P(n^{-\frac{1}{2}}).$$

Runtime The runtime of Algorithm 1 is composed of three main parts. First, $O(nm^3)$ operations are needed to compute all the relevant moments from the data and to construct the $d \times d \times d$ whitened tensor $\hat{\mathcal{W}}_\sigma$. The most time consuming task is computing the eigenpairs of $\hat{\mathcal{W}}_\sigma$, which can be done by either the homotopy method or O–NCM. Currently, no runtime guarantees are available for either of these methods. In practice, since there are $O(2^d)$ eigenpairs, these methods spend $O(2^d \cdot poly(d))$ operations in total. Finally, since there are $O(2^d)$ candidates and each KS test takes $O(dn)$ operations (Gonzalez et al., 1977), the filtering procedure runtime is $O(d2^d n)$.

Power-stability and orthogonal decomposition The exponential runtime of our algorithm stems from the fact that the set U_N of Newton-stable eigenpairs of $\mathcal{W}_\sigma$ is typically $O(2^d)$. However, in some cases, the set U^* of d relevant eigenpairs has additional structure so that a smaller candidate set may be computed instead of U_N. Consider the subset $U_P \subseteq U_N$ of *power-stable* eigenpairs of $\mathcal{W}_\sigma$:

Definition 3. *An eigenpair $(\boldsymbol{u}, \lambda)$ is power-stable if its projected Jacobian $J_p(\boldsymbol{u})$ is either positive or negative definite.*

Typically, the number of power-stable eigenpairs is significantly smaller than the number of Newton-stable eigenpairs.[2] In addition, U_P can be computed by the shifted higher-order power method (Kolda & Mayo, 2011; 2014).

Similarly to Lemma 2, one can show that U_P is guaranteed to contain U^* whenever the following stronger version of condition *(II)* holds: for all $(\boldsymbol{u}_i, \lambda_i) \in U^*$, the matrix

$$(WKL_{\boldsymbol{u}_i})^\top (2\mathcal{C}(I, I, \boldsymbol{e}_i) - C)(WKL_{\boldsymbol{u}_i}) \tag{31}$$

is either positive-definite or negative-definite.

As an example, consider the case where $P_{\boldsymbol{h}}$ has the support $\boldsymbol{h} \in I_d$. Then model (1) corresponds to a GMM with d spherical components with linearly independent means. In this case, both C and $\mathcal{C}$ are diagonal with $\boldsymbol{p}$ on their diagonal. Thus, the matrices in (31) take the form $-L_{\boldsymbol{e}_i}^\top \operatorname{diag}(\boldsymbol{p}) L_{\boldsymbol{e}_i}$, which are all negative-definite when $\boldsymbol{p} > 0$. In fact, in this case, $\mathcal{W}_\sigma$ has an orthogonal CP decomposition and the d orthogonal eigenpairs in U^* are the *only* negative-definite power-stable eigenpairs of $\mathcal{W}_\sigma$ (Anandkumar et al., 2014). Similarly, when $P_{\boldsymbol{h}}$ is a product distribution, the same orthogonal structure appears if the *centered* moments of $\boldsymbol{x}$ are used instead of M and $\mathcal{M}$. As shown in Anandkumar et al. (2014), the power method, accompanied with a deflation procedure, decomposes an orthogonal tensor in polynomial time, thus implying an efficient algorithm in these cases. However, under the much weaker conditions we pose on $P_{\boldsymbol{h}}$, the relevant eigenpairs in U^* are not necessarily power-stable and the CP decomposition of $\mathcal{W}_\sigma$ does not necessarily include U^*.

5. Experiments

We demonstrate our method in three scenarios: (I) simulations from the exact binary model (1); (II) learning a common population genetic admixture model; (III) learning the proportion matrix of a cell mixture from DNA methylation levels. Due to lack of space, (III) is deferred to Appendix N. Code to reproduce the simulation results can be found at https://github.com/arJaffe/BinaryLatentVariables.

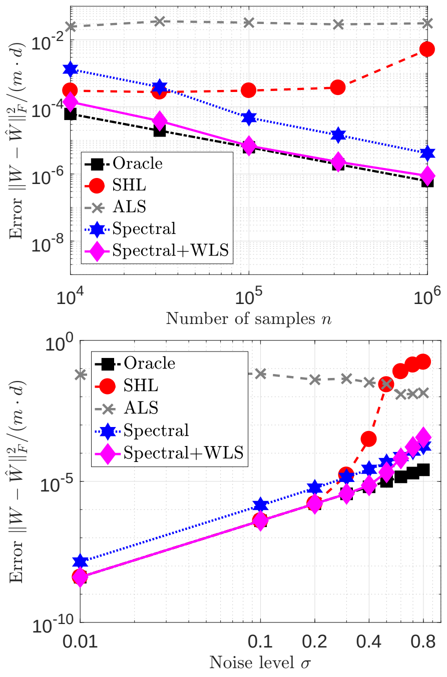

Figure 2. Upper panel: Error in W vs. sample size n with $\sigma = 0.4$. lower panel: Error in W vs. noise level σ with $n = 10^5$.

5.1. Simulations

We generated n samples from model (1) with $d = 6$ hidden units, $m = 30$ observable features, and Gaussian noise $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, I_m)$. The m columns of W were drawn uniformly from the unit sphere $\mathbb{S}_{d-1}$. Fixing a mean vector $\boldsymbol{a} \in \mathbb{R}^d$ and a covariance matrix $R \in \mathbb{R}^{d\times d}$, each hidden vector $\boldsymbol{h}$ was generated independently by first drawing $\boldsymbol{r} \sim \mathcal{N}(\boldsymbol{a}, R)$ and then taking its binary rounding.

Figure 2 shows the error, in Frobenius norm, averaged over 50 independent realizations of X as a function of n (upper panel) and σ (lower panel) for 5 methods: (i) our spectral approach, Algorithm 1 (Spectral); (ii) Algorithm 1 followed by a single weighted least squares step (Appendix K) (Spectral+WLS); (iii) SHL, the matrix decomposition method of Slawski et al. (2013)[3]; (iv) ALS with a random initialization (Appendix L); and (v) an oracle estimator that is given the exact matrix H and computes W via least squares.

As one can see, as opposed to SHL, our method is consistent for $\sigma > 0$ and achieves an error rate $O(n^{-\frac{1}{2}})$ corresponding to a slope of -1 in the upper panel of Fig. 2. In addition, as seen in the lower panel of Fig. 2, at low levels of noise our method is comparable to SHL, whereas at high levels it is far more accurate. Finally, adding a weighted least squares step reduces the error for low noise levels, but increases the error

[2] We currently do not know whether the number of power-stable eigenpairs of a generic tensor is polynomial or exponential in d.

[3] Code from https://sites.google.com/site/slawskimartin/code. For each realization X, we made 50 runs of SHL and chose H, W minimizing $\|X - W^\top H\|_F$.

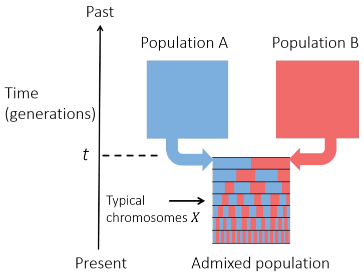

Figure 3. Illustration of the admixture model.

for high noise levels. A comparison between the runtime of SHL and the spectral method appears in Appendix I.

5.2. Population genetic admixture

We present an application of our method to a fundamental problem in population genetics, known as admixture (see Fig. 3). Admixture refers to the mixing of $d \geq 2$ ancestral populations that were long separated, e.g., due to geographical or cultural barriers (Pritchard et al., 2000; Alexander et al., 2009; Li et al., 2008). The observed data X is an $m \times n$ matrix where m is the number of modern "admixed" individuals and n is the number of relevant locations in their DNA, known as SNPs. Each SNP corresponds to two alleles and individuals may have different alleles. Fixing a reference allele for each location, X_{ij} takes values in $\{0, \frac{1}{2}, 1\}$ according to the number of reference alleles appearing in the genotype of individual $i \in [m]$ at locus $j \in [n]$.

Given the genotypes X, an important problem in population genetics is to estimate the following two quantities. The allele frequency matrix $H \in [0,1]^{d \times n}$ whose entry H_{kj} is the frequency of the reference allele at locus $j \in [n]$ in ancestral population $k \in [d]$; and the admixture proportion matrix $W \in [0,1]^{d \times m}$ whose columns sum to 1 and its entry W_{ki} is the proportion of individual i's genome that was inherited from population k.

A common model for X in terms of W and H is to assume that the number of alleles $2X_{ij} \in \{0,1,2\}$ is the sum of two i.i.d. Bernoulli random variables with success probability $F_{ij} = \sum_{k=1}^{d} W_{ki} H_{kj}$, namely, $X_{ij}|H \sim \frac{1}{2} \cdot \text{Binomial}(2, F_{ij})$. Note that under this model

$$\mathbb{E}[X|H] = F = W^\top H. \qquad (32)$$

Although (32) has similar form to model (1), there are two main differences; the noise is not normally distributed and the matrix H is non-binary. Yet, the binary model (1) serves as a good approximation whenever various alleles are rare in some populations but abundant in others. Specifically, for ancestral populations that have been long separated, some alleles may become *fixed* in one population (i.e., reach frequency of 1) while being totally absent in others.

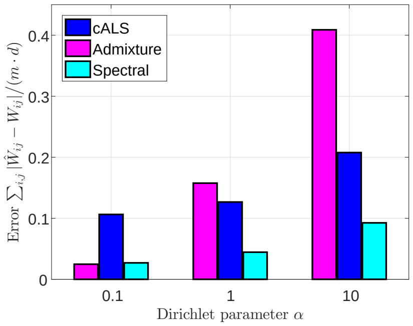

Figure 4. Average absolute error in $\hat{W}$ vs. Dirichlet parameter α for $d = 3$ ancestral populations and $m = 50$ admixed individuals.

Simulating genetic admixture We followed a standard simulation scheme apllied, for example, in Xue et al. (2017); Gravel (2012); Price et al. (2009). First, using SCRM (Staab et al., 2015), we simulated $d = 3$ ancestral populations separated for 4000 generations and generated the genomes of 40 individuals for each. H was then computed as the frequency of the reference alleles in each population. Next, the columns of W were sampled from a symmetric Dirichlet distribution with parameter $\alpha \geq 0$. Finally, the genomes of $m = 50$ admixed individuals were generated as mosaics of genomic segments of individuals from the ancestral populations with proportions W. The mosaic nature of the admixed genomes is an important realistic detail, due to the *linkage* (correlation) between SNPs (Xue et al., 2017). A detailed description is in Appendix M.

We compare our algorithm to two methods. The first is Admixture (Alexander et al., 2009), one of the most widely used algorithms in population genetics, which aims to maximize the likelihood of X. The second is the recently proposed spectral method ALStructure (Cabreros & Storey, 2017), where an estimation of $\text{span}(W^\top)$ via Chen & Storey (2015) is followed by constrained ALS iterations of W and H. For our method, two modification are needed for Algorithm 1. First, since the distribution of $X_{ij} - \boldsymbol{w}_i^T \boldsymbol{h}_j$ is not Gaussian, the corrected moments $\hat{M}_\sigma, \hat{\mathcal{M}}_\sigma$ as calculated by (17) do not satisfy (18). Instead, we implemented a matrix completion algorithm derived in (Jain & Oh, 2014) for a similar setup, see Appendix J for more details. In addition, the filtering process described in Section 4 is no longer valid. However, as d is relatively small, we performed exhaustive search over all candidate subsets of size d and choose the one that maximized the likelihood.

Figure 4 compares the results of the 3 methods for $\alpha = 0.1, 1, 10$. The spectral method outperforms Admixture and ALStructure for $\alpha = 1, 10$ and performs similarly to Admixture for $\alpha = 0.1$.

Acknowledgements This research was funded in part by NIH Grant 1R01HG008383-01A1.

References

Alexander, D., Novembre, J., and Lange, K. Fast model-based estimation of ancestry in unrelated individuals. *Genome research*, 19(9):1655–1664, 2009.

Anandkumar, A., Foster, D., Hsu, D., Kakade, S., and Liu, Y. A spectral algorithm for latent dirichlet allocation. In *Advances in Neural Information Processing Systems*, pp. 917–925, 2012a.

Anandkumar, A., Hsu, D., and Kakade, S. A method of moments for mixture models and hidden markov models. In *COLT*, volume 1, pp. 4, 2012b.

Anandkumar, A., Ge, R., Hsu, D., Kakade, S., and Telgarsky, M. Tensor decompositions for learning latent variable models. *Journal of Machine Learning Research*, 15(1): 2773–2832, 2014.

Arabshahi, F. and Anandkumar, A. Spectral methods for correlated topic models. In *Artificial Intelligence and Statistics*, pp. 1439–1447, 2017.

Arora, S., Ge, R., Kannan, R., and Moitra, A. Computing a nonnegative matrix factorization–provably. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pp. 145–162. ACM, 2012.

Baadel, S., Thabtah, F., and Lu, J. Overlapping clustering: A review. In *SAI Computing Conference (SAI), 2016*, pp. 233–237. IEEE, 2016.

Banerjee, A., Krumpelman, C., Ghosh, J., Basu, S., and Mooney, R. Model-based overlapping clustering. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pp. 532–537. ACM, 2005.

Becker, E., Robisson, B., Chapple, C., Guénoche, A., and Brun, C. Multifunctional proteins revealed by overlapping clustering in protein interaction network. *Bioinformatics*, 28(1):84–90, 2011.

Billingsley, P. *Convergence of probability measures*. John Wiley & Sons, 2013.

Cabreros, I. and Storey, J. A nonparametric estimator of population structure unifying admixture models and principal components analysis. *bioRxiv*, pp. 240812, 2017.

Cartwright, D. and Sturmfels, B. The number of eigenvalues of a tensor. *Linear algebra and its applications*, 438(2): 942–952, 2013.

Chen, L., Han, L., and Zhou, L. Computing tensor eigenvalues via homotopy methods. *SIAM Journal on Matrix Analysis and Applications*, 37(1):290–319, 2016.

Chen, X. and Storey, J. Consistent estimation of low-dimensional latent structure in high-dimensional data. *arXiv preprint arXiv:1510.03497*, 2015.

Cho, Ilin, and Raiko. Improved learning of Gaussian-Bernoulli restricted Boltzmann machines. *Artificial Neural Networks and Machine Learning–ICANN 2011*, pp. 10–17, 2011.

Cui, C., Dai, Y., and Nie, J. All real eigenvalues of symmetric tensors. *SIAM Journal on Matrix Analysis and Applications*, 35(4):1582–1601, 2014.

Donoho, D. and Stodden, V. When does non-negative matrix factorization give a correct decomposition into parts? In *Advances in neural information processing systems*, pp. 1141–1148, 2004.

Frieze, A., Jerrum, M., and Kannan, R. Learning linear transformations. In *Foundations of Computer Science, 1996. Proceedings., 37th Annual Symposium on*, pp. 359–368. IEEE, 1996.

Golub, G. and Van Loan, C. *Matrix computations*, volume 3. JHU Press, 2012.

Gonzalez, T., Sahni, S., and Franta, W. An efficient algorithm for the Kolmogorov-Smirnov and Lilliefors tests. *ACM Transactions on Mathematical Software (TOMS)*, 3 (1):60–64, 1977.

Gravel, S. Population genetics models of local ancestry. *Genetics*, 191(2):607–619, 2012.

Guo, C., Lin, W., and Liu, C. A modified Newton iteration for finding nonnegative Z-eigenpairs of a nonnegative tensor. *arXiv preprint arXiv:1705.07487*, 2017.

Hillar, C. and Lim, L. Most tensor problems are np-hard. *Journal of the ACM (JACM)*, 60(6):45, 2013.

Hinton, G. A practical guide to training restricted Boltzmann machines. *Momentum*, 9(1):926, 2010.

Hinton, G. and Salakhutdinov, R. Reducing the dimensionality of data with neural networks. *science*, 313(5786): 504–507, 2006.

Houseman, E., Accomando, P., Koestler, D., Christensen, B., Marsit, C., Nelson, H., Wiencke, J., and Kelsey, K. Dna methylation arrays as surrogate measures of cell mixture distribution. *BMC bioinformatics*, 13(1):86, 2012.

Hsu, D. and Kakade, S. Learning mixtures of spherical gaussians: moment methods and spectral decompositions. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, pp. 11–20. ACM, 2013.

Hubbard, J. and Hubbard, B. *Vector calculus, linear algebra, and differential forms: a unified approach*. Matrix Editions, 2015.

Hyvärinen, A., Karhunen, J., and Oja, E. *Independent component analysis*, volume 46. John Wiley & Sons, 2004.

Jaffe, A., Weiss, R., and Nadler, B. Newton correction methods for computing real eigenpairs of symmetric tensors. *arXiv preprint arXiv:1706.02132*, 2017.

Jain, P. and Oh, S. Learning mixtures of discrete product distributions using spectral decompositions. In *Conference on Learning Theory*, pp. 824–856, 2014.

Kolda, T. and Mayo, J. Shifted power method for computing tensor eigenpairs. *SIAM Journal on Matrix Analysis and Applications*, 32(4):1095–1124, 2011.

Kolda, T. and Mayo, J. An adaptive shifted power method for computing generalized tensor eigenpairs. *SIAM Journal on Matrix Analysis and Applications*, 35(4):1563–1581, 2014.

Kritchman, S. and Nadler, B. Non-parametric detection of the number of signals: Hypothesis testing and random matrix theory. *IEEE Transactions on Signal Processing*, 57(10):3930–3941, 2009.

Lehmann, E. and Romano, J. *Testing statistical hypotheses*. Springer Science & Business Media, 2006.

Li, J., Absher, D., Tang, H., Southwick, A., Casto, A., Ramachandran, S., Cann, H., Barsh, G., Feldman, M., Cavalli-Sforza, L., et al. Worldwide human relationships inferred from genome-wide patterns of variation. *science*, 319(5866):1100–1104, 2008.

Lim, L. Singular values and eigenvalues of tensors: a variational approach. In *Computational Advances in Multi-Sensor Adaptive Processing, 2005 1st IEEE International Workshop on*, pp. 129–132. IEEE, 2005.

Melchior, J., Wang, N., and Wiskott, L. Gaussian-binary restricted Boltzmann machines for modeling natural image statistics. *PloS one*, 12(2):e0171015, 2017.

Miettinen, P. and Vreeken, J. Mdl4bmf: Minimum description length for boolean matrix factorization. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 8 (4):18, 2014.

Moitra, A. and Valiant, G. Settling the polynomial learnability of mixtures of gaussians. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pp. 93–102. IEEE, 2010.

Mossel, E. and Roch, S. Learning nonsingular phylogenies and hidden markov models. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pp. 366–375. ACM, 2005.

Price, A., Tandon, A., Patterson, N., Barnes, K., Rafaels, N., Ruczinski, I., Beaty, T., Mathias, R., Reich, D., and Myers, S. Sensitive detection of chromosomal segments of distinct ancestry in admixed populations. *PLoS genetics*, 5(6):e1000519, 2009.

Pritchard, J., Stephens, M., and Donnelly, P. Inference of population structure using multilocus genotype data. *Genetics*, 155(2):945–959, 2000.

Qi, L. Eigenvalues of a real supersymmetric tensor. *Journal of Symbolic Computation*, 40(6):1302–1324, 2005.

Regalia, P. and Kofidis, E. Monotonic convergence of fixed-point algorithms for ica. *IEEE Transactions on Neural Networks*, 14(4):943–949, 2003.

Segal, E., Battle, A., and Koller, D. Decomposing gene expression into cellular processes. In *Proceedings of the Pacific Symposium on Biocomputing*, volume 8, pp. 89–100, 2002.

Shalvi, O. and Weinstein, E. Super-exponential methods for blind deconvolution. *IEEE Transactions on Information Theory*, 39(2):504–519, 1993.

Slawski, M., Hein, M., and Lutsik, P. Matrix factorization with binary components. In *Advances in Neural Information Processing Systems*, pp. 3210–3218, 2013.

Staab, Zhu, Metzler, and Lunter. scrm: efficiently simulating long sequences using the approximated coalescent with recombination. *Bioinformatics*, 31(10):1680–1682, 2015.

Taylor, G., Hinton, G., and Roweis, S. Modeling human motion using binary latent variables. In *Advances in neural information processing systems*, pp. 1345–1352, 2007.

Van der Veen, A.-J. Analytical method for blind binary signal separation. *IEEE Transactions on Signal Processing*, 45(4):1078–1082, 1997.

Wang, N., Melchior, J., and Wiskott, L. An analysis of gaussian-binary restricted Boltzmann machines for natural images. In *ESANN*, 2012.

Xue, J., Lencz, T., Darvasi, A., Peer, I., and Carmi, S. The time and place of european admixture in ashkenazi jewish history. *PLoS genetics*, 13(4):e1006644, 2017.