Nonparametric Estimation of Multi-View Latent Variable Models

Abstract

Spectral methods have greatly advanced the estimation of latent variable models, generating a sequence of novel and efficient algorithms with strong theoretical guarantees. However, current spectral algorithms are largely restricted to mixtures of discrete or Gaussian distributions. In this paper, we propose a kernel method for learning multi-view latent variable models, allowing each mixture component to be nonparametric. The key idea of the method is to embed the joint distribution of a multi-view latent variable into a reproducing kernel Hilbert space, and then the latent parameters are recovered using a robust tensor power method. We establish that the sample complexity for the proposed method is quadratic in the number of latent components and is a low order polynomial in the other relevant parameters. Thus, our non-parametric tensor approach to learning latent variable models enjoys good sample and computational efficiencies. Moreover, the non-parametric tensor power method compares favorably to EM algorithm and other existing spectral algorithms in our experiments.

1. Introduction

Latent variable models have been used to address various machine learning problems, ranging from modeling temporal dynamics, to text document analysis and to social network analysis (Rabiner & Juang, 1986; Clark, 1990; Hoff et al., 2002; Blei et al., 2003). Recently, there is a surge of interest in designing spectral algorithms for estimating the parameters of latent variable models (Hsu et al., 2009; Parikh et al., 2011; Song et al., 2011; Foster et al., 2012; Anandkumar et al., 2012a;b; Király, 2013). Compared to the Expectation-Maximization (EM) algorithm (Dempster et al., 1977) traditionally used for this task, spectral algorithms are better in terms of their computational efficiency and provable guarantees. Current spectral algorithms are largely restricted to mixture of discrete or Gaussian distributions, e.g. (Anandkumar et al., 2012a; Hsu & Kakade, 2013). When the mixture components are distributions other than these standard distributions, the theoretical guarantees for these algorithms are no longer applicable, and their empirical performance can be very poor.

In this paper, we propose a kernel method for estimating the parameters of multi-view latent variable models where the mixture components can be nonparametric. The key idea is to embed the joint distribution of such a model into a reproducing kernel Hilbert space, and exploit the low rank structure of the embedded distribution (or covariance operators). The key computation involves a kernel singular value decomposition of the two-view covariance operator, followed by a robust tensor power method on the three-view covariance operator. These standard matrix operations makes the algorithm very efficient and easy to deploy.

The kernel algorithm proposed in this paper is more general than the previous spectral algorithms which work only for distributions with parametric assumptions (Anandkumar et al., 2012a; Hsu & Kakade, 2013). When we use delta kernel, our algorithm reduces to the spectral algorithm for discrete mixture components analyzed in (Anandkumar et al., 2012a). When we use universal kernels, such as Gaussian RBF kernel, our algorithm can recover Gaussian mixture components as well as mixture components with other distributions. In this sense, our work also provides a unifying framework for previous spectral algorithms. We prove sample complexity bounds for the nonparametric tensor power method and establish that the sample complexity is quadratic in the number of latent components, and is a low order polynomial in the other relevant parameters such as the lower bound on mixing weights. Thus, we propose a computational and sample efficient nonparametric approach to learning latent variable models.

Kernel methods have been previously applied to learning latent variable models. However, none of the previous works explicitly recovers the actual parameters of the models (Song et al., 2011; Song & Dai, 2013; Sgouritsa et al., 2013). Most of them estimate an (unknown) invertible transformation of the latent parameters, and it is not clear how one can recover the actual parameters based on these estimates. Furthermore, these works focused on predictive task: recover the marginal distribution of the observed variables by making use of the low rank structure of the latent

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

variable models. It is significantly more challenging to design kernel algorithms for actual parameter recovery and analyze theoretical properties of these algorithms.

We compare our kernel algorithm to the EM algorithm and previous spectral algorithms. We show that when the model assumptions are correct for the EM algorithm and previous spectral algorithms, our algorithm converges in terms of estimation error to these competitors. In the opposite cases when the model assumptions are incorrect, our algorithm is able to adapt to the nonparametric mixture components and beating alternatives by a very large margin.

2. Notation

We denote by X a random variable with domain $\mathcal{X}$, and refer to instantiations of X by the lower case character, x. We endow $\mathcal{X}$ with some σ-algebra $\mathscr{A}$ and denote a distributions (with respect to $\mathscr{A}$) on $\mathcal{X}$ by $\mathbb{P}(X)$. We also deal with multiple random variables, $X_1, X_2, \ldots, X_\ell$, with joint distribution $\mathbb{P}(X_1, X_2, \ldots, X_\ell)$. For simplicity of notation, we assume that the domains of all $X_t, t \in [\ell]$ are the same, but the methodology applies to the cases where they have different domains. Furthermore, we denote by H a hidden variable with domain $\mathcal{H}$ and distribution $\mathbb{P}(H)$.

A *reproducing kernel Hilbert space (RKHS)* $\mathcal{F}$ on $\mathcal{X}$ with a kernel $k(x, x')$ is a Hilbert space of functions $f(\cdot) : \mathcal{X} \mapsto \mathbb{R}$ with inner product $\langle \cdot, \cdot \rangle_{\mathcal{F}}$. Its element $k(x, \cdot)$ satisfies the reproducing property: $\langle f(\cdot), k(x, \cdot) \rangle_{\mathcal{F}} = f(x)$, and consequently, $\langle k(x, \cdot), k(x', \cdot) \rangle_{\mathcal{F}} = k(x, x')$, meaning that we can view the evaluation of a function f at any point $x \in \mathcal{X}$ as an inner product. Alternatively, $k(x, \cdot)$ can be viewed as an implicit feature map $\phi(x)$ where $k(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{F}}$. Popular kernel functions on $\mathbb{R}^n$ include the Gaussian RBF kernel $k(x, x') = \exp(-s \left\| x - x' \right\|^2)$ and the Laplace kernel $\exp(-s \|x - x'\|)$. Kernel functions have also been defined on graphs, time series, dynamical systems, images and other structured objects (Schölkopf et al., 2004). Thus the methodology presented below can be readily generalized to a diverse range of data types as long as kernel functions are defined.

3. Kernel Embedding of Distributions

We begin by providing an overview of kernel embeddings of distributions, which are *implicit* mappings of distributions into potentially *infinite* dimensional RKHS. The kernel embedding approach represents a distribution by an element in the RKHS associated with a kernel function (Smola et al., 2007; Sriperumbudur et al., 2008),

$$\mu_X := \mathbb{E}_X \left[\phi(X) \right] = \int_{\mathcal{X}} \phi(x) \, \mathbb{P}(dx), \tag{1}$$

where the distribution is mapped to its expected feature map, *i.e.*, to a point in a potentially infinite-dimensional and implicit feature space. The kernel embedding μ_X has the property that the expectation of any RKHS function f can be evaluated as an inner product in $\mathcal{F}$, $\mathbb{E}_X[f(X)] = \langle \mu_X, f \rangle_{\mathcal{F}}, \forall f \in \mathcal{F}$.

Kernel embeddings can be readily generalized to joint distributions of two or more variables using tensor product feature maps. For instance, we can embed a joint distribution of two variables X_1 and X_2 into a tensor product feature space $\mathcal{F} \times \mathcal{F}$ by

$$\begin{aligned} \mathcal{C}_{X_1 X_2} &:= \mathbb{E}_{X_1 X_2}[\phi(X_1) \otimes \phi(X_2)] && (2) \\ &= \int_{\mathcal{X} \times \mathcal{X}} \phi(x_1) \otimes \phi(x_2) \, \mathbb{P}(dx_1 \times dx_2), && (3) \end{aligned}$$

where the reproducing kernel for the tensor product features satisfies $\langle \phi(x_1) \otimes \phi(x_2), \phi(x_1') \otimes \phi(x_2') \rangle_{\mathcal{F} \times \mathcal{F}} = k(x_1, x_1') \, k(x_2, x_2')$. By analogy, we can also define $\mathcal{C}_{X_1 X_2 X_3} := \mathbb{E}_{X_1 X_2 X_3}[\phi(X_1) \otimes \phi(X_2) \otimes \phi(X_3)]$.

Kernel embedding of distributions has rich representational power. The mapping is injective for characteristic kernels (Sriperumbudur et al., 2008). That is, if two distributions, $\mathbb{P}(X)$ and $\mathbb{Q}(X)$, are different, they are mapped to two distinct points in the RKHS. For domain $\mathbb{R}^d$, many commonly used kernels are characteristic, such as the Gaussian RBF kernel and Laplace kernel. This injective property of kernel embeddings has been exploited to design state-of-the-art two-sample tests (Gretton et al., 2012) and independence tests (Gretton et al., 2008).

3.1. Kernel Embedding as Multi-Linear Operator

The joint embeddings can also be viewed as an uncentered covariance operator $\mathcal{C}_{X_1 X_2} : \mathcal{F} \mapsto \mathcal{F}$ by the standard equivalence between a tensor product feature and a linear map. That is, given two functions $f_1, f_2 \in \mathcal{F}$, their covariance can be computed by $\mathbb{E}_{X_1 X_2}[f_1(X_1) f_2(X_2)] = \langle f_1, \mathcal{C}_{X_1 X_2} f_2 \rangle_{\mathcal{F}}$, or equivalently $\langle f_1 \otimes f_2, \mathcal{C}_{X_1 X_2} \rangle_{\mathcal{F} \times \mathcal{F}}$, where in the former we view $\mathcal{C}_{XY}$ as an operator while in the latter we view it as an element in tensor product feature space. By analogy, $\mathcal{C}_{X_1 X_2 X_3}$ can be regarded as a multi-linear operator from $\mathcal{F} \times \mathcal{F} \times \mathcal{F}$ to $\mathbb{R}$. It will be clear from the context whether we use $\mathcal{C}_{XY}$ as an operator between two spaces or as an element from a tensor product feature space. For generic introduction to tensor and tensor notation, please see (Kolda & Bader, 2009).

The operator $\mathcal{C}_{X_1 X_2 X_3}$ (with shorthand $\mathcal{C}_{X_{1:3}}$) is linear in each argument (mode) when fixing other arguments. Furthermore, the application of the operator to a set of elements $\{f_1, f_2, f_3 \in \mathcal{F}\}$ can be defined using the inner product from the tensor product feature space, *i.e.*,

$$\mathcal{C}_{X_{1:3}} \times_1 f_1 \times_2 \times_3 f_3 := \langle \mathcal{C}_{X_{1:3}},\ f_1 \otimes f_2 \otimes f_3 \rangle_{\mathcal{F}^3}$$
$$= \mathbb{E}_{X_1 X_2 X_3} \left[\prod_{i \in [3]} \langle \phi(X_i),\ f_i \rangle_{\mathcal{F}} \right],$$

where $\times_i$ means applying f_i to the i-th argument of $\mathcal{C}_{X_{1:3}}$. Furthermore, we can define the Hilbert-Schmidt norm $\|\cdot\|$ of $\mathcal{C}_{X_{1:3}}$ as

$$\|\mathcal{C}_{X_{1:3}}\|^2 = \sum_{i_1=1}^{\infty} \sum_{i_2=1}^{\infty} \sum_{i_3=1}^{\infty} (\mathcal{C}_{X_{1:3}} \times_1 u_{i_1} \times_2 u_{i_2} \times_3 u_{i_3})^2$$

using three collections of orthonormal bases $\{u_{i_1}\}_{i_1=1}^{\infty}$, $\{u_{i_2}\}_{i_2=1}^{\infty}$, and $\{u_{i_3}\}_{i_3=1}^{\infty}$. We can also define the inner product for the space of such operator that $\|\mathcal{C}_{X_{1:3}}\| < \infty$

$$\left\langle \mathcal{C}_{X_{1:3}},\ \widetilde{\mathcal{C}}_{X_{1:3}} \right\rangle = \sum_{i_1=1}^{\infty} \sum_{i_2=1}^{\infty} \sum_{i_\ell=1}^{\infty} (\mathcal{C}_{X_{1:\ell}} \times_1 u_{i_1} \times_2 u_{i_2} \times_3 u_{i_3})$$
$$\cdot\ (\widetilde{\mathcal{C}}_{X_{1:\ell}} \times_1 u_{i_1} \times_2 \ldots \times_\ell u_{i_\ell}).$$

The joint embedding, $\mathcal{C}_{X_1 X_2}$, is a 2nd order tensor, and we can essentially use notations and operations for matrices. For instance, we can perform singular value decomposition

$$\mathcal{C}_{X_1 X_2} = \sum_{i=1}^{\infty} \sigma_i \cdot u_{i_1} \otimes u_{i_2},$$

where $\sigma_i \in \mathbb{R}$ are singular values ordered in nonincreasing manner, and $\{u_{i_1}\}_{i_1=1}^{\infty} \subset \mathcal{F}, \{u_{i_2}\}_{i_2=1}^{\infty} \subset \mathcal{F}$ are singular vectors and orthonormal bases. The rank of $\mathcal{C}_{X_1 X_2}$ is the smallest k such that $\sigma_i = 0$ for $i > k$.

3.2. Finite Sample Estimate

While we rarely have access to the true underlying distribution, $\mathbb{P}(X)$, we can readily estimate its embedding using a finite sample average. Given a sample $\mathcal{D}_X = \{x^1, \ldots, x^m\}$ of size m drawn *i.i.d.* from $\mathbb{P}(X)$, the empirical kernel embedding is

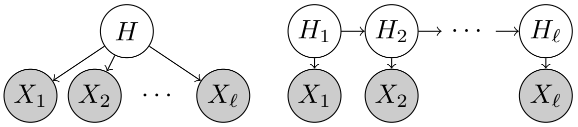
$$\widehat{\mu}_X := \frac{1}{m} \sum\nolimits_{i=1}^{m} \phi(x^i). \tag{4}$$

This empirical estimate converges to its population counterpart in RKHS norm, $\|\widehat{\mu}_X - \mu_X\|_{\mathcal{F}}$, with a rate of $O_p(m^{-\frac{1}{2}})$ (Smola et al., 2007).

The covariance operator can be estimated similarly using finite sample average. Given m pairs of training examples $\mathcal{D}_{XY} = \{(x_1^i, x_2^i)\}_{i \in [m]}$ drawn *i.i.d.* from $\mathbb{P}(X_1, X_2)$,

$$\widehat{\mathcal{C}}_{X_1 X_2} = \frac{1}{m} \sum_{i=1}^{m} \phi(x_1^i) \otimes \phi(x_2^i). \tag{5}$$

Similarly, given sample from distribution $\mathbb{P}(X_1, X_2, X_3)$, one can estimate $\widehat{\mathcal{C}}_{X_{1:3}} = \frac{1}{m} \sum_{i=1}^{m} \phi(x_1^i) \otimes \phi(x_2^i) \otimes \phi(x_3^i)$.



(a) Naïve Bayes model (b) Hidden Markov model

Figure 1. Examples of multi-view latent variable models.

By virtue of the kernel trick, most of the computation required for subsequent statistical inference using kernel embeddings can be reduced to the Gram matrix manipulation. The entries in the Gram matrix K correspond to the kernel value between data points x^i and x^j, *i.e.*, $K_{ij} = k(x^i, x^j)$, and therefore its size is determined by the number of data points in the sample. The size of the Gram matrix is in general much smaller than the dimension of the feature spaces (which can be infinite). This enables efficient nonparametric methods using the kernel embedding representation. If the sample size is large, the computation in kernel embedding methods may be expensive. In this case, a popular solution is to use a low-rank approximation of the Gram matrix, such as incomplete Cholesky factorization (Fine & Scheinberg, 2001), which is known to work very effectively in reducing computational cost of kernel methods, while maintaining the approximation accuracy.

4. Multi-View Latent Variable Models

Multi-view latent variable models studied in this paper are a special class of Bayesian networks in which

- observed variables $X_1, X_2, \ldots, X_\ell$ are conditionally independent given a **discrete** latent variable H, and
- the conditional distributions, $\mathbb{P}(X_t|H)$, of the $X_t, t \in [\ell]$ given the hidden variable H can be different.

The conditional independent structure of a multi-view latent variable model is illustrated in Figure 1(a), and many complicated graphical models, such as the hidden Markov model in Figure 1(b), can be reduced to a multi-view latent variable model. **For simplicity of exposition, we will explain our method using the model with symmetric view**. That is the conditional distribution are the same for each view, *i.e.*, $\mathbb{P}(X|h) = \mathbb{P}(X_1|h) = \mathbb{P}(X_2|h) = \mathbb{P}(X_3|h)$. In Appendix 8, we will show that multi-view models with different views can be reduced to ones with symmetric view.

4.1. Conditional Embedding Operator

For simplicity of exposition, we focus on a simple model with three observed variables, *i.e.*, $\ell = 3$. Suppose $H \in [k]$, then we can embed each conditional distribution $\mathbb{P}(X|h)$ corresponding to a particular value of $H = h$ into the RKHS as

$$\mu_{X|h} = \int_{\mathcal{X}} \phi(x)\, \mathbb{P}(dx|h). \tag{6}$$

If we vary the value of H, we obtain the kernel embedding for different $\mathbb{P}(X|h)$. Conceptually, we can collect these embeddings into a matrix (with potentially infinite number of rows)

$$\mathcal{C}_{X|H} = \left(\mu_{X|h=1}, \mu_{X|h=2}, \ldots, \mu_{X|h=k}\right), \tag{7}$$

which is called the conditional embedding operator. If we use the standard basis e_h in $\mathbb{R}^k$ to represent each value of h, we can retrieve each $\mu_{X|h}$ from $\mathcal{C}_{X|H}$ by

$$\mu_{X|h} = \mathcal{C}_{X|H} e_h \tag{8}$$

Once we have the conditional embedding $\mu_{X|h}$, we can estimate the density $p(x|h)$ by performing an inner product $p(x|h) = \langle \phi(x), \mu_{X|h} \rangle$.

4.2. Factorized Kernel Embedding

Then the distributions, $\mathbb{P}(X_1, X_2)$ and $\mathbb{P}(X_1, X_2, X_3)$, can be factorized respectively

$$\mathbb{P}(dx_1, dx_2) = \int_{\mathcal{H}} \mathbb{P}(dx_1|h)\, \mathbb{P}(dx_2|h)\, \mathbb{P}(dh), \text{ and}$$

$$\mathbb{P}(dx_1, dx_2, dx_3) = \int_{\mathcal{H}} \mathbb{P}(dx_1|h)\, \mathbb{P}(dx_2|h)\, \mathbb{P}(dx_3|h)\, \mathbb{P}(dh).$$

Since we assume the hidden variable $H \in [k]$ is discrete, we let $\pi_h := \mathbb{P}(h)$. Furthermore, if we apply Kronecker delta kernel $\delta(h, h')$ with feature map e_h, then the embeddings for $\mathbb{P}(H)$

$$\mathcal{C}_{HH} = \mathbb{E}_H[e_H \otimes e_H] = \begin{pmatrix} \pi_1 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & \pi_k \end{pmatrix}, \text{ and}$$

$$\begin{aligned} \mathcal{C}_{HHH} &= \mathbb{E}_H[e_H \otimes e_H \otimes e_H] \\ &= \Big(\pi_h\, \delta(h, h')\, \delta(h', h'') \Big)_{h, h', h'' \in [k]} \end{aligned}$$

are diagonal tensors. Making use of $\mathcal{C}_{HH}$ and $\mathcal{C}_{HHH}$, and the factorization of the distributions $\mathbb{P}(X_1, X_2)$ and $\mathbb{P}(X_1, X_2, X_3)$, we obtain the factorization of the embedding of $\mathbb{P}(X_1, X_2)$ (second order embedding)

$$\begin{aligned} &\mathcal{C}_{X_1 X_2} \\ &= \int_{\mathcal{H}} \left(\int_{\mathcal{X}} \phi(x_1)\, \mathbb{P}(dx_1|h) \right) \otimes \left(\int_{\mathcal{X}} \phi(x_2)\, \mathbb{P}(dx_2|h) \right) \mathbb{P}(dh) \\ &= \int_{\mathcal{H}} \left(\mathcal{C}_{X|H} e_h \right) \otimes \left(\mathcal{C}_{X|H} e_h \right)\, \mathbb{P}(dh) \\ &= \mathcal{C}_{X|H} \left(\int_{\mathcal{H}} e_h \otimes e_h\, \mathbb{P}(dh) \right) \mathcal{C}_{X|H}^{\top} \\ &= \mathcal{C}_{X|H}\, \mathcal{C}_{HH}\, \mathcal{C}_{X|H}^{\top}, \end{aligned} \tag{9}$$

and that of $\mathbb{P}(X_1, X_2, X3)$ (third order embedding)

$$\mathcal{C}_{X_1 X_2 X_3} = \mathcal{C}_{HHH} \times_1 \mathcal{C}_{X|H} \times_2 \mathcal{C}_{X|H} \times_3 \mathcal{C}_{X|H}. \tag{10}$$

4.3. Identifiability of Parameters

We note that $\mathcal{C}_{X|H} = \left(\mu_{X|h=1},\ \mu_{X|h=2},\ \ldots,\ \mu_{X|h=k}\right)$, and the kernel embeddings for $\mathcal{C}_{X_1 X_2}$ and $\mathcal{C}_{X_1 X_2 X_3}$ can be alternatively written as

$$\mathcal{C}_{X_1 X_2} = \sum_{h \in [k]} \pi_h \cdot \mu_{X|h} \otimes \mu_{X|h}, \tag{11}$$

$$\mathcal{C}_{X_1 X_2 X_3} = \sum_{h \in [k]} \pi_h \cdot \mu_{X|h} \otimes \mu_{X|h} \otimes \mu_{X|h} \tag{12}$$

Allman et al. (Allman et al., 2009) showed that, under mild conditions, a finite mixture of nonparametric product distributions is identifiable. The multi-view latent variable model in (12) has the same form as a finite mixture of nonparametric product distribution, and therefore we can adapt Allman's results to the current setting.

Proposition 1 (Identifiability) *Let $\mathbb{P}(X_1, X_2, X_3)$ be a multi-view latent variable model, such that the conditional distributions $\{\mathbb{P}(X|h)\}_{h \in [k]}$ are linearly independent. Then, the set of parameters $\left\{\pi_h, \mu_{X|h}\right\}_{h \in [k]}$ are identifiable from $\mathcal{C}_{X_1 X_2 X_3}$, up to label swapping of the hidden variable H.*

Example 1. The probability vector of a discrete variable $X \in [n]$, and the joint probability table of two discrete variables $X_1 \in [n]$ and $X_2 \in [n]$, are both kernel embeddings. To see this, let the kernel be the Kronecker delta kernel $k(x, x') = \delta(x, x')$ whose feature map $\phi(x)$ is the standard basis of e_x in $\mathbb{R}^n$. The x-th dimension of e_x is 1 and 0 otherwise. Then

$$\mu_X = \begin{pmatrix} \mathbb{P}(x = 1) & \ldots & \mathbb{P}(x = n) \end{pmatrix}^{\top},$$

$$\mathcal{C}_{X_1 X_2} = \Big(\mathbb{P}(x_1 = s, x_2 = t) \Big)_{s, t \in [n]}.$$

We require that the conditional probability table $\{P(X|h)\}_{h \in [k]}$ to have full column rank for identifiability in this case.

Example 2. Suppose we have a k-component mixture of one dimensional spherical Gaussian distributions. The Gaussian components have identical covariance σ^2, but their mean values are distinct. Note that this model is not identifiable under the framework of (Hsu & Kakade, 2013) since the mean values are just scalars and therefore, rank deficient. However, if we embed the density functions using universal kernels such as Gaussian RBF kernel, it can be shown that the mixture model becomes identifiable. This is because we are working with the entire density function which are linear independent from each other in this case. Thus, the non-parametric framework allows us to incorporate a wider range of latent variable models.

Finally, we remark that the identifiability result in Proposition 1 can be extended to cases where the conditional distributions do not satisfy linear independence, *i.e.*, they are overcomplete, e.g. (Kruskal, 1977; De Lathauwer et al., 2007; Anandkumar et al., 2013b). However, in general, it is not tractable to learn such overcomplete models and we do not consider them here.

5. Kernel Algorithm

We first design a kernel algorithm to recover the parameters, $\{\pi_h, \mu_{X|h}\}_{h\in[k]}$, of the multi-view latent variable model based on $\mathcal{C}_{X_1X_2}$ and $\mathcal{C}_{X_1X_2X_3}$. This can be easily extended to the sample versions and this is discussed in Section 5.2. Again for simplicity of exposition, the algorithm is explained for symmetric view case. The more general version is presented in Appendix 8.

5.1. Population Case

We first derive the algorithm for the population case as if we could access the true operator $\mathcal{C}_{X_1X_2}$ and $\mathcal{C}_{X_1X_2X_3}$. Its finite sample counterpart will be presented in the next section. The algorithm can be thought of as a kernel generalization of the algorithm in (Anandkumar et al., 2013a) using embedding representations.

Step 1. We perform eigen-decomposition of $\mathcal{C}_{X_1X_2}$,

$$\mathcal{C}_{X_1X_2} = \sum_{i=1}^{\infty} \sigma_i \cdot u_i \otimes u_i$$

where the eigen-values are ordered in non-decreasing manner. According to the factorization in Eq. (9), $\mathcal{C}_{X_1X_2}$ has rank k. Let the leading eigenvectors corresponding to the largest k eigen-value be $\mathcal{U}_k := (u_1, u_2, \ldots, u_k)$, and the eigen-value matrix be $S_k := \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_k)$. We define the whitening operator $\mathcal{W} := \mathcal{U}_k S_k^{-1/2}$ which satisfies

$$\mathcal{W}^\top \mathcal{C}_{X_1X_2} \mathcal{W} = (\mathcal{W}^\top \mathcal{C}_{X|H} \mathcal{C}_{HH}^{1/2})(\mathcal{C}_{HH}^{1/2} \mathcal{C}_{X|H}^\top \mathcal{W}) = I,$$

and $M := \mathcal{W}^\top \mathcal{C}_{X|H} \mathcal{C}_{HH}^{1/2}$ is an orthogonal matrix.

Step 2. We apply the whiten operator to the 3rd order kernel embedding $\mathcal{C}_{X_1X_2X_3}$

$$\mathcal{T} := \mathcal{C}_{X_1X_2X_3} \times_1 (\mathcal{W}^\top) \times_2 (\mathcal{W}^\top) \times_3 (\mathcal{W}^\top).$$

According to the factorization in Eq. (10),

$$\mathcal{T} = \mathcal{C}_{HHH}^{-1/2} \times_1 M \times_2 M \times_3 M,$$

which is a tensor with orthogonal factors. Essentially, each column v_i of M is an eigenvector of the tensor $\mathcal{T}$.

Step 3. We use tensor power method to find eigenvectors M for $\mathcal{T}$ (Anandkumar et al., 2013a). We provide the method in the Appendix in Algorithm 2 for completeness.

Step 4. We recover the conditional embedding operator by undoing the whitening step

$$\mathcal{C}_{X|H} = (\mu_{X|h=1}, \mu_{X|h=1}, \ldots, \mu_{X|h=k}) = (\mathcal{W})^\dagger M.$$

Algorithm 1 KernelSVD(K, L, k)

```
Out: $\widehat{S}_k$ and $(\beta_1, \ldots, \beta_k)$
1: Cholesky decomposition: $K = R^\top R$
2: Eigen-decomposition: $\frac{1}{4m^2} RLR^\top \widetilde{\beta} = \widehat{\sigma}^2\, \widetilde{\beta}$
3: Use $k$ leading eigenvalues: $\widehat{S}_k = \mathrm{diag}(\widehat{\sigma}_1, \ldots, \widehat{\sigma}_k)$
4: Use $k$ leading eigenvectors: $(\widetilde{\beta}_1, \ldots, \widetilde{\beta}_k)$ to compute:
   $(\beta_1, \ldots, \beta_k) = R^\dagger (\widetilde{\beta}_1, \ldots, \widetilde{\beta}_k)$
```

5.2. Finite Sample Case

Given m observation $\mathcal{D}_{X_1X_2X_3} = \{(x_1^i, x_2^i, x_3^i)\}_{i\in[m]}$ drawn *i.i.d.* from a multi-view latent variable model $\mathbb{P}(X_1, X_2, X_3)$, we now design a kernel algorithm to estimate the latent parameters from data. Although the empirical kernel embeddings can be infinite dimensional, we can carry out the decomposition using just the kernel matrices. We denote the implicit feature matrix by

$$\Phi := (\phi(x_1^1), \ldots, \phi(x_1^m), \phi(x_2^1), \ldots, \phi(x_2^m)),$$
$$\Psi := (\phi(x_2^1), \ldots, \phi(x_2^m), \phi(x_1^1), \ldots, \phi(x_1^m)),$$

and the corresponding kernel matrix by $K = \Phi^\top \Phi$ and $L = \Psi^\top \Psi$ respectively. Then the steps in the population case can be mapped one-by-one into kernel operations.

Step 1. We perform a kernel eigenvalue decomposition of the empirical 2nd order embedding

$$\widehat{\mathcal{C}}_{X_1X_2} := \frac{1}{2m} \sum_{i=1}^{m} \left(\phi(x_1^i) \otimes \phi(x_2^i) + \phi(x_2^i) \otimes \phi(x_1^i)\right),$$

which can be expressed succinctly as $\widehat{\mathcal{C}}_{X_1X_2} = \frac{1}{2m} \Phi \Psi^\top$. Its leading k eigenvectors $\widehat{\mathcal{U}}_k = (\widehat{u}_1, \ldots, \widehat{u}_k)$ lie in the span of the column of Φ, *i.e.*, $\widehat{\mathcal{U}}_k = \Phi(\beta_1, \ldots, \beta_k)$ with $\beta \in \mathbb{R}^{2m}$. Then we can transform the eigen-value decomposition problem for an infinite dimensional matrix to a problem involving finite dimensional kernel matrices,

$$\widehat{\mathcal{C}}_{X_1X_2} \widehat{\mathcal{C}}_{X_1X_2}^\top u = \widehat{\sigma}^2\, u \;\Rightarrow\; \frac{1}{4m^2} \Phi \Psi^\top \Psi \Phi^\top \Phi \beta = \widehat{\sigma}^2\, \Phi \beta$$
$$\Rightarrow\; \frac{1}{4m^2} KLK\beta = \widehat{\sigma}^2\, K\beta.$$

Let the Cholesky decomposition of K be $R^\top R$. Then by redefining $\widetilde{\beta} = R\beta$, and solving an eigenvalue problem

$$\frac{1}{4m^2} RLR^\top \widetilde{\beta} = \widehat{\sigma}^2\, \widetilde{\beta}, \text{ and obtain } \beta = R^\dagger \widetilde{\beta}. \qquad (13)$$

The resulting eigenvectors satisfy $u_i^\top u_{i'} = \beta_i^\top \Phi^\top \Phi \beta_{i'} = \beta_i^\top K \beta_{i'} = \widetilde{\beta}_i^\top \widetilde{\beta}_{i'} = \delta_{ii'}$. This step is summarized in Algorithm 1.

Step 2. We whiten the empirical 3rd order embedding

$$\widehat{\mathcal{C}}_{X_1X_2X_3} := \frac{1}{3m} \sum_{i=1}^{m} (\phi(x_1^i) \otimes \phi(x_2^i) \otimes \phi(x_3^i)$$
$$+ \phi(x_3^i) \otimes \phi(x_1^i) \otimes \phi(x_2^i) + \phi(x_2^i) \otimes \phi(x_3^i) \otimes \phi(x_1^i))$$

using $\widehat{\mathcal{W}} := \widehat{\mathcal{U}}_k \widehat{S}_k^{-1/2}$, and obtain

$$\widehat{\mathcal{T}} := \frac{1}{3m} \sum_{i=1}^{m} (\xi(x_1^i) \otimes \xi(x_2^i) \otimes \xi(x_3^i)$$
$$+ \xi(x_3^i) \otimes \xi(x_1^i) \otimes \xi(x_2^i) + \xi(x_2^i) \otimes \xi(x_3^i) \otimes \xi(x_1^i)),$$

where

$$\xi(x_1^i) := \widehat{S}_k^{-1/2} (\beta_1, \ldots, \beta_k)^\top \Phi^\top \phi(x_1^i) \in \mathbb{R}^k.$$

Step 3. We run tensor power method (Anandkumar et al., 2013a) on the finite dimension tensor $\widehat{\mathcal{T}}$ to obtain its leading k eigenvectors $\widehat{M} := (\widehat{v}_1, \ldots, \widehat{v}_k)$.

Step 4. The estimates of the conditional embeddings are $\widehat{\mathcal{C}}_{X|H} = (\widehat{\mu}_{X|h=1}, \ldots, \widehat{\mu}_{X|h=k}) = \Phi(\beta_1, \ldots, \beta_k) \widehat{S}_k^{1/2} \widehat{M}$.

6. Sample Complexity

Let $\rho := \sup_{x \in \mathcal{X}} k(x, x)$, $\| \cdot \|$ be the Hilbert-Schmidt norm, $\pi_{\min} := \min_{i \in [k]} \pi_i$ and $\sigma_k(\mathcal{C}_{X_1 X_2})$ be the k-th singular value of $\mathcal{C}_{X_1 X_2}$.

Theorem 2 (Sample Bounds) *Pick any $\delta \in (0, 1)$. When the number of samples m satisfies*

$$m > \frac{\theta \rho^2 \log \frac{\delta}{2}}{\sigma_k^2(\mathcal{C}_{X_1, X_2})}, \quad \theta := \max \left(\frac{C_3 k^2}{\sigma_k(\mathcal{C}_{X_1, X_2})}, \frac{C_4 k^{2/3}}{\pi_{\min}^{1/3}} \right),$$

for some constants $C_3, C_4 > 0$, and the number of iterations N and the number of random initialization vectors L (drawn uniformly on the sphere $\mathcal{S}^{k-1}$) satisfy

$$N \geq C_2 \cdot \left(\log(k) + \log \log \left(\frac{1}{\sqrt{\pi}_{\min} \epsilon_T} \right) \right),$$

for constant $C_2 > 0$ and $L = \text{poly}(k) \log(1/\delta)$, the robust power method in (Anandkumar et al., 2013a) yields eigenpairs $(\widehat{\lambda}_i, \widehat{\phi}_i)$ such that there exists a permutation η, with probability $1 - 4\delta$, we have

$$\| \pi_j^{-1/2} \mu_{X|h=j} - \widehat{\phi}_{\eta(j)} \| \leq 8 \epsilon_T \cdot \pi_j^{-1/2},$$
$$| \pi_j^{-1/2} - \widehat{\lambda}_{\eta(j)} | \leq 5 \epsilon_T, \quad \forall j \in [k],$$

and

$$\left\| T - \sum_{j=1}^{k} \hat{\lambda}_j \hat{\phi}_j^{\otimes 3} \right\| \leq 55 \epsilon_T,$$

where ϵ_T is the tensor perturbation bound

$$\epsilon_T := \| \widehat{\mathcal{T}} - \mathcal{T} \| \leq \frac{12 \rho \sqrt{\log \frac{\delta}{2}}}{\sqrt{m}\, \sigma_k^{1.5}(\mathcal{C}_{X_1, X_2})} + \frac{512 \sqrt{2} \rho^3 \left(\log \frac{\delta}{2} \right)^{1.5}}{m^{1.5}\, \sigma_k^3(\mathcal{C}_{X_1, X_2}) \sqrt{\pi}_{\min}}$$

Thus, the above result provides bounds on the estimated eigen-pairs using the robust tensor power method. The proof is in Appendix 10.

Remarks: We note that the sample complexity is $\text{poly}(k, \rho, 1/\pi_{\min}, 1/\sigma_k(\mathcal{C}_{X_1, X_2}))$ of a low order, and in particular, it is $O(k^2)$, when the other parameters are fixed. For the special case of discrete measurements, where the kernel $k(x, x') = \delta(x, x')$, we have $\rho = 1$. Note that the sample complexity depends in this case only on the number of components k and not on the dimensionality of the observed state space. Thus, the robust tensor method has efficient sample and computational complexities for nonparametric latent variable estimation.

7. Experiments

Methods. We compared our kernel nonparametric algorithm with three alternatives

1. The EM algorithm for mixture of Gaussians. The EM algorithm is not guaranteed to find the global solution in each trial. Thus we randomly initialize it 10 times.
2. The spectral algorithm for mixture of spherical Gaussians (Hsu & Kakade, 2013). The assumption in Hsu & Kakade (2013) is very restrictive: the collection of spherical Gaussian centers need to span a k-dimension subpsace.
3. A discretization based spectral algorithm (Kasahara & Shimotsu, 2010). This algorithm approximates the joint distribution of the observed variables with histogram and then applies the spectral algorithm to recover the discretized conditional density. It is well-known that density estimation using histogram suffers from poor performance even for 3-dimension data. The error of this algorithm is typically 10 times larger than alternatives. To make the curves for other methods clearer, we did not plot the performance of Kasahara & Shimotsu (2010) algorithm in the figures.

Our method has a hyper-parameter, kernel bandwidth, which we selected for each view separately using cross-validation.

7.1. Synthetic Data

We generated three-dimensional synthetic data from various mixture models. The variables corresponding to the dimensions are independent given the latent component indicator. More specifically, we explored two settings

1. Gaussian conditional densities with different variances;
2. Mixture of Gaussian and shifted Gamma conditional densities.

The shifted Gamma distribution has density

$$p(x - \mu) = \frac{(x - \mu)^{(d-1)} e^{-x/\theta}}{\theta^d \Gamma(d)}, \; x \geq \mu$$

where we chose the shape parameter $d \leq 1$ such that density is very skewed. Furthermore, we chose the mean and variance parameters of the Gaussian/Gamma density such that component pair-wise overlap is relatively small according to the fisher ratio $\frac{(\mu_1-\mu_2)^2}{\sigma_1^2+\sigma_2^2}$.

We also varied the number of samples m for observed variables, from 50 to $10,000$, and experimented with $k = 2, 3, 4$ or 8 mixture components. The mixture proportion for the h-th component is set to be $\pi_h = \frac{2h}{k(k+1)}$, $\forall h \in [k]$ (unbalanced). It is worth noting that as k becomes larger, it is more difficult to recover parameters. This is because only a small number of data will be generated for the first several clusters. For each n, k with each setting, we randomly generated 10 sets of samples. The average results are reported.

Error measure. We measured the performance of algorithms by the following weighted ℓ_2 norm difference

$$MSE := \sum_{h=1}^{k} \pi_h \sqrt{\sum_{j=1}^{m'} (p(x^j|h) - \widehat{p}(x^j|h))^2}$$

where $\{x^j\}_{j\in[m]}$ is a set of uniformly-spaced test points.

Results. The results are plotted in Figure 2. It is clear that the kernel spectral method converges rapidly with the data increment in all experiments setting.

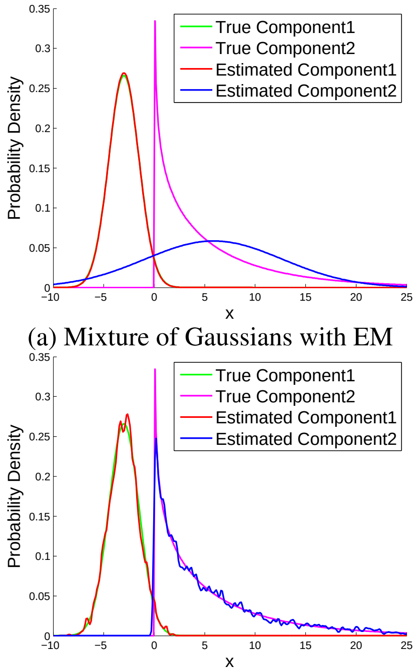
In mixture of Gaussians setting, the EM algorithm is best since the model is correctly specified in this case. The spectral learning algorithm for spherical Gaussians did not perform well since the assumption for the method is too restricted. The performance our kernel method converges to that of EM algorithm.

In the mixture of Gaussian and Gamma setting, our kernel spectral algorithm performs clearly much better than other algorithms. These results show that our algorithm is able to automatically adapt to the shape of the density.

We also plotted the actual recovered conditional density in Figure 3. It can be seen that kernel spectral algorithm recover pretty nicely both the Gaussian and Gamma components, while EM algorithm is able to fit only one component.

7.2. Flow Cytometry Data

Flow cytometry (FCM) data are multivariate measurements from flow cytometers that record light scatter and fluorescence emission properties of hundreds of thousands of individual cells. They are important to the studying of the cell structures of normal and abnormal cells and the diagnosing of human disease. Aghaeepour et al. (2013) introduced the FlowCAP-challenge whose main task is grouping the flow cytometry data automatically. Clustering on the FCM data is a difficult task because the distribution of the data is non-Gaussian and heavily skewed.



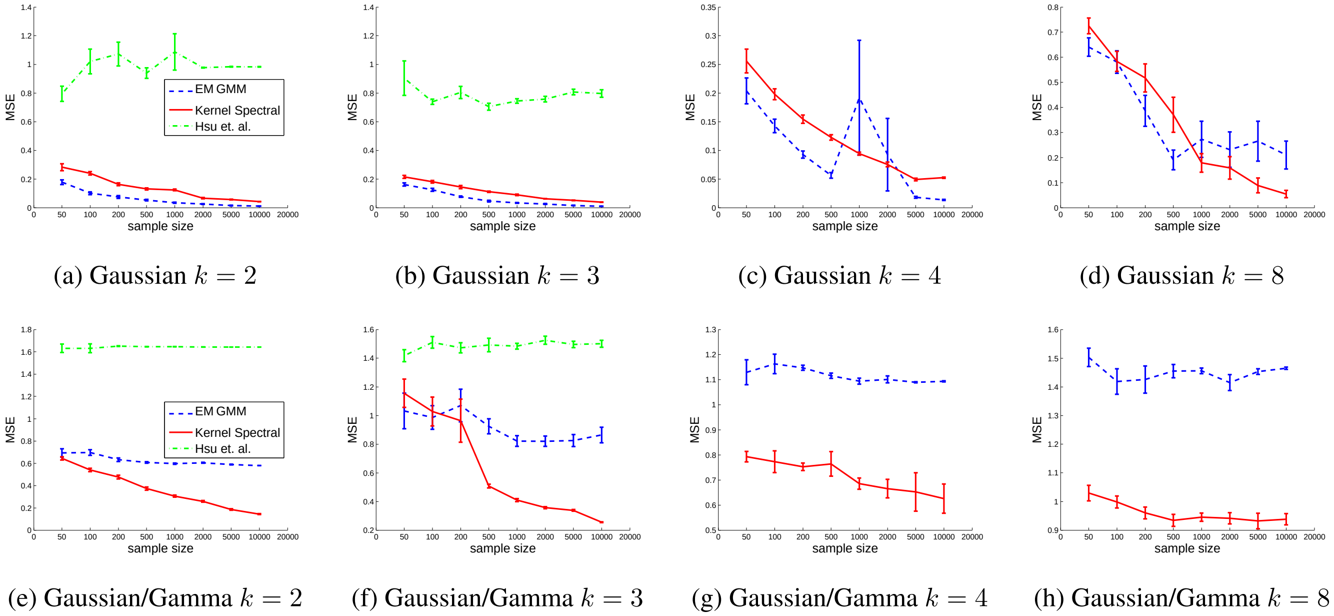
(a) Mixture of Gaussians with EM

(b) Kernel Spectral

Figure 3. Kernel spectral algorithm is able to adapt to the shape of the mixture components, while EM algorithm for mixture of Gaussians misfit the Gamma distribution.

We used the DLBCL Lymphoma dataset collection from (Aghaeepour et al., 2013) to compare our kernel algorithm with multi-view mixture of Gaussian model. This collection of datasets contain 30 datasets, and each consists of tens of thousands of cells measurements in 5 dimensions. Each dataset is a separate clustering task, and we fit a multi-view model to each dataset separately and use the maximum-a-posteriori assignment for obtaining the cluster labels. In each dataset, each data point is already manually labeled, and therefore we can evaluate the clustering performance using the F-score.

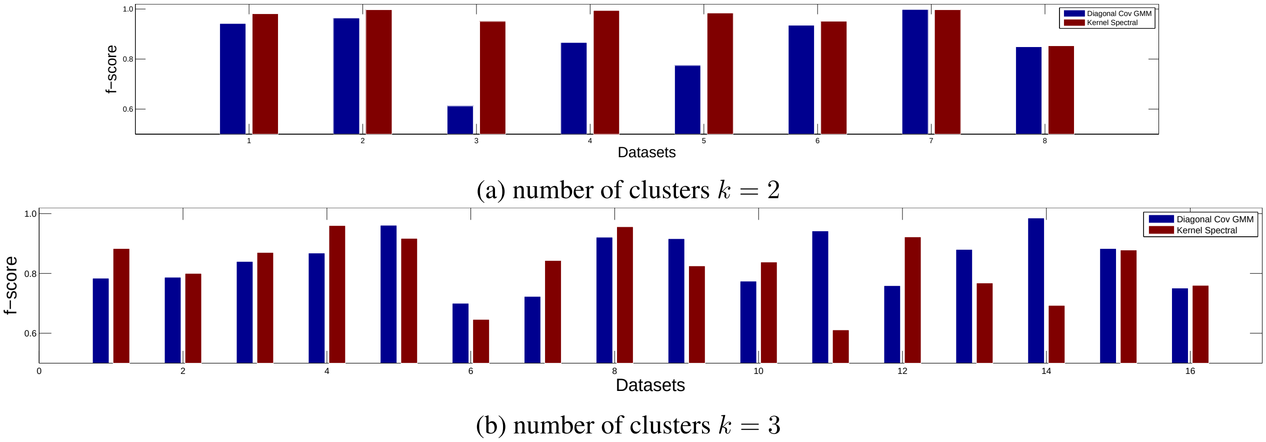
We splitted the 5 dimensions of the data into three views: dimension 1 and 2 as the first view, 3 and 4 the second and 5 the third view. For each dataset, we selected the best kernel bandwidth by 5-fold cross validation using log-likelihood. For EM algorithm for mixture of Gaussians (GMM) with diagonal covariances, we used a very generous 20 restarts. Figure 4 presents the results sorted by the number of clusters. Our method (kernel spectral) outperforms EM-GMM in a majority of datasets. However, there are also datasets where kernel spectral algorithm has a large gap in performance compared to GMM. These are the datasets where the multi-view assumptions are heavily violated. Obtaining improved performance in these datasets will be a subject of our future study where we plan to develop even more robustness kernel spectral algorithms.



(a) Gaussian $k = 2$ (b) Gaussian $k = 3$ (c) Gaussian $k = 4$ (d) Gaussian $k = 8$

(e) Gaussian/Gamma $k = 2$ (f) Gaussian/Gamma $k = 3$ (g) Gaussian/Gamma $k = 4$ (h) Gaussian/Gamma $k = 8$

Figure 2. (a)-(d) Mixture of Gaussian distributions with $k = 2, 3, 4, 8$ components. (e)-(h) Mixture of Gaussian/Gamma distribution with $k = 2, 3, 4, 8$. For the former case, the performance of kernel spectral algorithm converge to those of EM algorithm for mixture of Gaussian model. For the latter case, the performance of kernel spectral algorithm are consistently much better than EM algorithm for mixture of Gaussian model. Spherical Gaussian spectral algorithm does not work for $k = 4, 8$, and hence not plotted.



(a) number of clusters $k = 2$

(b) number of clusters $k = 3$

Figure 4. Clustering results on DLBCL flow cytometry data. Each group of bars represents F-scores from EM-GMM with diagonal covariances (blue) and kernel spectral method (red). The datasets are ordered by increasing sample size.

References

Aghaeepour, Nima, Finak, Greg, Consortium, The FlowCAP, Consortium, The DREAM, Hoos, Holger, Mosmann, Tim R, Brinkman, Ryan, Gottardo, Raphael, and Scheuermann, Richard H. Critical assessment of automated flow cytometry data analysis techniques. *Nature Methods*, 10(3):228–238, 2013.

Allman, Elizabeth, Matias, Catherine, and Rhodes, John. Identifiability of parameters in latent structure models with many observed variables. *The Annals of Statistics*, 37(6A):3099–3132, 2009.

Anandkumar, A., Ge, R., Hsu, D., Kakade, S. M., and Telgarsky, M. Tensor Methods for Learning Latent Variable Models. *Available at arXiv:1210.7559*, Oct. 2012a.

Anandkumar, A., Ge, R., Hsu, D., and Kakade, S. M. A Tensor Spectral Approach to Learning Mixed Membership Community Models. *ArXiv 1302.2684*, Feb. 2013a.

Anandkumar, A., Hsu, D., Janzamin, M., and Kakade, S. M. When are Overcomplete Topic Models Identifiable? Uniqueness of Tensor Tucker Decompositions with Structured Sparsity. *ArXiv 1308.2853*, Aug. 2013b.

Anandkumar, Animashree, Foster, Dean P., Hsu, Daniel, Kakade, Sham M., and Liu, Yi-Kai. A spectral algorithm for latent dirichlet allocation. *Available at arXiv:1204.6703*, 2012b.

Blei, D., Ng, A., and Jordan, M. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, January 2003.

Clark, A. Inference of haplotypes from PCR-amplified samples of diploid populations. *Molecular Biology and Evolution*, 7(2):111–122, 1990.

De Lathauwer, L., Castaing, J., and Cardoso, J.-F. Fourth-order cumulant-based blind identification of underdetermined mixtures. *IEEE Tran. on Signal Processing*, 55: 2965–2973, June 2007.

Dempster, A. P., Laird, N. M., and Rubin, D. B. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39(1):1–22, 1977.

Fine, S. and Scheinberg, K. Efficient SVM training using low-rank kernel representations. *Journal of Machine Learning Research*, 2:243–264, 2001.

Foster, D.P., Rodu, J., and Ungar, L.H. Spectral dimensionality reduction for hmms. *Arxiv preprint arXiv:1203.6130*, 2012.

Gretton, A., Fukumizu, K., Teo, C.-H., Song, L., Schölkopf, B., and Smola, A. J. A kernel statistical test of independence. In *Advances in Neural Information Processing Systems 20*, pp. 585–592, Cambridge, MA, 2008. MIT Press.

Gretton, A., Borgwardt, K., Rasch, M., Schoelkopf, B., and Smola, A. A kernel two-sample test. *JMLR*, 13:723–773, 2012.

Hoff, Peter D., Raftery, Adrian E., and Handcock, Mark S. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97 (460):1090–1098, 2002.

Hsu, D., Kakade, S., and Zhang, T. A spectral algorithm for learning hidden markov models. In *Proc. Annual Conf. Computational Learning Theory*, 2009.

Hsu, Daniel and Kakade, Sham M. Learning mixtures of spherical gaussians: moment methods and spectral decompositions. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, ITCS '13, pp. 11–20, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1859-4.

Kasahara, Hiroyuki and Shimotsu, Katsumi. Nonparametric identification of multivariate mixtures. *Journal of the Royal Statistical Society - Series B*, 2010.

Király, Franz. Efficient orthogonal tensor decomposition, with an application to latent variable model learning. *Available at arXiv:1309.3233*, 2013.

Kolda, Tamara. G. and Bader, Brett W. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

Kruskal, J.B. Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear algebra and its applications*, 18(2):95–138, 1977.

Parikh, A., Song, L., and Xing, E. P. A spectral algorithm for latent tree graphical models. In *Proceedings of the International Conference on Machine Learning*, 2011.

Rabiner, L. R. and Juang, B. H. An introduction to hidden Markov models. *IEEE ASSP Magazine*, 3(1):4–16, January 1986.

Rosasco, L., Belkin, M., and Vito, E.D. On learning with integral operators. *Journal of Machine Learning Research*, 11:905–934, 2010.

Schölkopf, B., Tsuda, K., and Vert, J.-P. *Kernel Methods in Computational Biology*. MIT Press, Cambridge, MA, 2004.

Sgouritsa, Eleni, Janzing, Dominik, Peters, Jonas, and Schölkopf, Bernhard. Identifying finite mixtures of nonparametric product distributions and causal inference of confounders. In *Conference on Uncertainty on Artificial Intelligence (UAI)*, 2013.

Smola, A. J., Gretton, A., Song, L., and Schölkopf, B. A Hilbert space embedding for distributions. In *Proceedings of the International Conference on Algorithmic Learning Theory*, volume 4754, pp. 13–31. Springer, 2007.

Song, L. and Dai, B. Robust low rank kernel embedding of multivariate distributions. In *Neural Information Processing Systems (NIPS)*, 2013.

Song, L., Parikh, A., and Xing, E.P. Kernel embeddings of latent tree graphical models. In *Advances in Neural Information Processing Systems*, volume 25, 2011.

Sriperumbudur, B., Gretton, A., Fukumizu, K., Lanckriet, G., and Schölkopf, B. Injective Hilbert space embeddings of probability measures. In *Proc. Annual Conf. Computational Learning Theory*, pp. 111–122, 2008.