Tensor Analyzers

Yichuan Tang TANG@CS.TORONTO.EDU
Ruslan Salakhutdinov RSALAKHU@CS.TORONTO.EDU
Geoffrey Hinton HINTON@CS.TORONTO.EDU
Department of Computer Science, University of Toronto. Toronto, Ontario, Canada.

Abstract

Factor Analysis is a statistical method that seeks to explain linear variations in data by using unobserved latent variables. Due to its *additive* nature, it is not suitable for modeling data that is generated by multiple groups of latent factors which interact *multiplicatively*. In this paper, we introduce Tensor Analyzers which are a multilinear generalization of Factor Analyzers. We describe an efficient way of sampling from the posterior distribution over factor values and we demonstrate that these samples can be used in the EM algorithm for learning interesting mixture models of natural image patches. Tensor Analyzers can also accurately recognize a face under significant pose and illumination variations when given only one previous image of that face. We also show that Tensor Analyzers can be trained in an unsupervised, semi-supervised, or fully supervised settings.

1. Introduction

Exploratory Factor Analysis is widely used in statistics to identify underlying linear factors. Mixtures of Factor Analyzers have been used successfully for unsupervised learning (Yang et al., 1999; Verbeek, 2006). Factor Analyzers (FAs) model each observation vector as a weighted linear combination of the unobserved factor values plus additive uncorrelated noise. For many types of data, this additive generative process is less suitable than a generative process that also contains multiplicative interactions between latent factors.

An example of multiplicative interactions is the set of face images under varying illuminations. It is known that these images of a particular person approximately lie on a 3 dimensional linear subspace, making a FA with 3 factors a good model (Belhumeur & Kriegman, 1996). The linear subspace (and thus the factor loadings) will be *person-specific* due to the way facial structures interact with the light to create a face image. As a result of this person-specific property, the factor loadings need to be a *function* of the person identity when modeling face images of *multiple* people. Instead of modeling 100 individuals with 100 separate FAs, it is desirable to use person-identity variables to linearly combine a "basis" (a set of) factor loadings to compactly model all 100 faces, drastically reducing the number of parameters. This factorial representation, as shown in Fig. 1, naturally allows for generalization to new people required for one-shot face recognition. Fig. 1 provides an illustration and Sec. 5.3 provides experimental validations.

To this end, we introduce Tensor Analyzers (TAs), which generalize FAs to the multilinear setting by introducing a factor loading tensor and multiple groups of latent factors. Utilizing the loading tensor, a group can change how another group's factors interact with the observed variables. This allows latent factor groups in a TA to learn highly interpretable representations. In the faces example, one group could represent lighting direction while the other could represent the identity.

In the special case of a TA with only one group of factors, its loading tensor reduces to a loading matrix, and the model is exactly the same as an ordinary FA. In a TA, when conditioned on all but one group of factors, the model effectively becomes a FA where the factor loadings are a function of the factor values in the groups we are conditioning on. The posterior distribution of the factor values in a FA can be computed analytically, so by cycling through each group of factors, efficient alternating Gibbs sampling is therefore possible in the TA. A TA is a proper density model so the extension to a mixture of TAs (MTA) is straight-

Proceedings of the 30^{th} International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28.

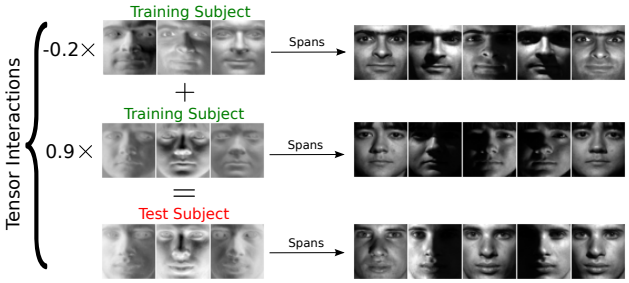


Figure 1. Illustration of why TA is needed. Three dimensional bases (left) span the pixel space of specific individuals. In order to properly generalize to a novel test subject, tensor interactions of the learned training bases are needed to form a new basis for modeling a test subject.

forward. When performing inference or learning in a TA, it is easy to make use of a supervisory signal that specifies the fact that 2 or more different observations are generated from the same factor values in some of the factor groups. This allows for TAs to seamlessly transition from an unsupervised density model to a semi and fully supervised model.

2. Related Works

Bilinear models with priors on the latent variables have been previously studied in the machine learning, statistics, and computer vision literatures. In (Grimes & Rao, 2005), sparsity is induced on the codes of a bilinear model to learn translational invariant representation from video. However, their model does not try to maximize $\log p(\mathbf{x})$, but instead finds the MAP estimate of the code activations, à la sparse coding. Culpepper et al. (2011) describe an outer-product factorization of the bilinear model that is trained as a density model using EM, but expensive Hamiltonian dynamics are required for sampling from the posterior. In addition, their model only admits an approximate M-step and does not make it easy to incorporate label information. Wang et al. (2007) proposed a multifactor GPLVM extension for modeling human motion (henceforth referred to as GPSC). In their model, the parameters are integrated out, and factors are kernalized. Optimization is needed to find the latent coordinates. Computationally, as in GPLVM, GPSC scales cubicly in the size of the training data.

While TAs and the above models take as input *i.i.d.* data vectors, there exists a plethora of tensor decomposition (TD) methods when the data comes in the form of **N-way tensors** (Tucker, 1963; Carroll & Chang, 1970; Lathauwer & Vandewalle, 2004; Wang & Ahuja, 2003; Sun et al., 2006). The SVD algorithm was used to learn a bilinear model to separate style and content (Tenenbaum & Freeman, 2000), which we will refer to as the S&C model. Tucker decomposition was applied to a 5-mode array of face images in (Vasilescu & Terzopoulos, 2002), finding multilinear bases called TensorFaces. Shashua & Hazan (2005) enforced non-negative constraints to PARAFAC decomposition. Chu & Ghahramani (2009) introduced a Bayesian probabilistic version of Tucker decomposition, while Xu et al. (2012) provided a nonparametric Bayesian extension. The main disadvantages of the tensor decomposition methods are that data must be arranged in a tensor and that inference given a *single* new test case is ill-posed and can be ad hoc[1].

In contrast, TAs do not have any of the above deficiencies. Our main contribution is in the introduction of standard Gaussian priors on each latent groups, which allows us to utilize the efficient inference procedure of Factor Analysis as part of TA's inference procedure. We also provide the EM algorithm that allows TAs to learn directly from data vectors in an entirely unsupervised manner. It can also make use of supervision in the form of equality constraints that specify that one group of factors should have the same vector of values for a subset of the training cases (Sec. 4.4). As an extension to FA, TA inherits an efficient inference algorithm that is used in each step of alternating Gibbs sampling and a closed-form M-step during learning. Unlike bilinear models, it can handle multilinear cases with 3 or more groups of latent factors. It can also be easily extended to a mixture model, provided we are willing to compute approximate densities, as described in (Sec. 4.3). In addition, posterior inference for a *single* test case is simple and accurate, as demonstrated by our one-shot face recognition experiments of Sec. 5.3.

3. Preliminaries

Following (Kolda & Bader, 2009), we refer to the number of dimensions of the tensor as its *order* (also known as modes). We will use bold lowercase letters to denote vectors (tensors of order one), e.g. $\mathbf{x}$; bold uppercase letters for matrices (tensors of order two), e.g. $\mathbf{W}$. We use the notation $\mathbf{w}_{(i,:)}$ to denote the i-th row of matrix $\mathbf{W}$. Higher order tensors are denoted by Euler script letters, e.g. a third-order tensor with dimensions of I, J, and K: $\mathcal{T} \in \mathbb{R}^{I \times J \times K}$.

Fibers: Fibers are higher-order generalization of row/column vectors. Elements of a tensor fiber is found by fixing all but one index. Specifically, $\boldsymbol{t}_{(:,j,k)}$

[1]E.g., the asymmetric model in (S&C) requires EM learning of a separate model during test time. Probabilistic TD methods require new test cases to come with labels.

is the mode-1 fiber of the tensor $\mathcal{T}$. Row and column vectors are the mode-2 and mode-1 fiber of a 2nd-order tensor, respectively.

Matricization: Matricization is the process of "flattening" a tensor into a matrix, by reordering the elements of the tensor. It is denoted by $\mathbf{T}_{(n)}$, where the mode-n fibers of $\mathcal{T}$ are placed in the columns of the resulting matrix $\mathbf{T}_{(n)}$. For example, given $\mathcal{T} \in \mathbb{R}^{I\times J\times K}$, $\mathbf{T}_{(1)} \in \mathbb{R}^{I\times JK}$.

n-mode vector product: By multiplying a vector $\mathbf{y} \in \mathbb{R}^{D_n}$ with a tensor $\mathcal{T} \in \mathbb{R}^{D_1\times D_2\times\cdots\times D_N}$ along the mode-n, the *n-mode (vector) product* is denoted by $\mathcal{T} \bar{\times}_n \mathbf{y}$. The resulting tensor is of size $D_1 \times \cdots \times D_{n-1} \times D_{n+1} \times \cdots \times D_N$.

3.1. Factor Analyzers

Let $\mathbf{x} \in \mathbb{R}^D$ denote the D-dimensional data, let $\{\mathbf{z} \in \mathbb{R}^d : d \leq D\}$ denote d-dimensional latent factors. FA is defined by a prior and likelihood:

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I}), \quad p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}; \mathbf{\Lambda}\mathbf{z} + \boldsymbol{\mu}, \mathbf{\Psi}), \quad (1)$$

where $\mathbf{I}$ is the $d \times d$ identity matrix; $\mathbf{\Lambda} \in \mathbb{R}^{D\times d}$ is the factor loading matrix, $\boldsymbol{\mu}$ is the mean. A diagonal $\mathbf{\Psi} \in \mathbb{R}^{D\times D}$ represents the variance of the observation noise. By integrating out the latent variable $\mathbf{z}$, a FA model becomes a Gaussian with constrained covariance:

$$p(\mathbf{x}) = \int_{\mathbf{z}} p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z} = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \Gamma), \quad (2)$$

where $\Gamma = \mathbf{\Lambda}\mathbf{\Lambda}^\mathsf{T} + \mathbf{\Psi}$. For inference, we are interested in the posterior, which is also a multivariate Gaussian:

$$p(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \mathbf{m}, \mathbf{V}^{-1}), \quad (3)$$

where $\mathbf{V} = \mathbf{I}+\mathbf{\Lambda}^\mathsf{T}\mathbf{\Psi}^{-1}\mathbf{\Lambda}$, and $\mathbf{m} = \mathbf{V}^{-1}\mathbf{\Lambda}^\mathsf{T}\mathbf{\Psi}^{-1}(\mathbf{x}-\boldsymbol{\mu})$. Maximum likelihood estimation of the parameters is straightforward using the EM algorithm (Rubin & Thayer, 1982). During the E-step, Eq. 3 is used to compute the posterior sufficient statistics. During the M-step, the expected complete-data log-likelihood $\mathbb{E}_{p(\mathbf{z}|\mathbf{x};\theta_{old})}[\log p(\mathbf{x}, \mathbf{z}; \theta)]$ is maximized with respect to the model parameters $\theta = \{\mathbf{\Lambda}, \boldsymbol{\mu}, \mathbf{\Psi}\}$.

4. Tensor Analyzers

TA replaces FA's factor loading $\mathbf{\Lambda} \in \mathbb{R}^{D\times d}$ with a factor "loading" tensor $\mathcal{T} \in \mathbb{R}^{D\times d_1\times\cdots\times d_J}$. In addition, a TA has J groups of factors: $\{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_J\}$: $j = 1, \ldots, J$, $\mathbf{z}_j \in \mathbb{R}^{d_j}$. The key property of TAs is that the interactions between a particular factor group and the data is modified by the factor values in other groups. Given $\{\mathbf{z}_2, \ldots, \mathbf{z}_J\}$, $\mathbf{\Lambda}_{new}$, which is the factor loading matrix between $\mathbf{z}_1$ and $\mathbf{x}$, is given by:

$$\mathbf{\Lambda}_{new} = (((\mathcal{T} \bar{\times}_2 \mathbf{z}_2) \bar{\times}_3 \mathbf{z}_3)\bar{\times}\cdots) \bar{\times}_J \mathbf{z}_J).$$

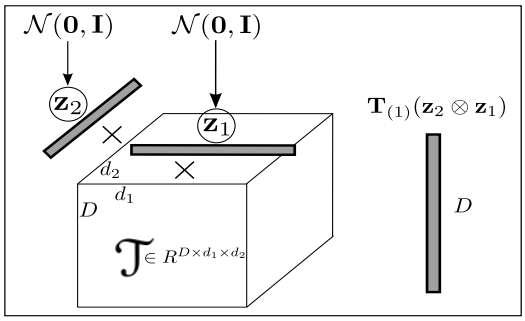


Figure 2. Diagram of TA's ($J = 2$) generative process. $\mathbf{\Lambda}_{new} = \mathcal{T} \bar{\times}_2 \mathbf{z}_2$ gives a new *factor loading matrix for $\mathbf{z}_1$. $\mathbf{\Lambda}_{new}\mathbf{z}_1 \triangleq \mathbf{T}_{(1)}(\mathbf{z}_2 \otimes \mathbf{z}_1)$ determines the mean of $p(\mathbf{x}|\mathbf{z}_1, \mathbf{z}_2)$.*

We will use the notation TA$\{D, d_1, d_2, \ldots, d_J\}$ to denote the aforementioned TA. By using a $(J+1)$-order tensor $\mathcal{T}$, a TA can model multiplicative interactions among its latent factors $\{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_J\}$. In contrast, FAs do not model multiplicative interactions involving terms such as $z_i z_j : \ i \neq j$.

Each group of factors has a standard Normal prior:

$$p(\mathbf{z}_j) = \mathcal{N}(\mathbf{z}_j|0, \mathbf{I}), \quad j = 1, 2, \ldots, J. \quad (4)$$

For clarity of presentation, we assume $J = 3$ for the following equations. The likelihood $p(\mathbf{x}|\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3)$ is:

$$\mathcal{N}(\mathbf{x}|\mathbf{m} + \sum_j^3 \mathbf{W}_j\mathbf{z}_j + \mathbf{T}_{(1)}(\mathbf{z}_3 \otimes \mathbf{z}_2 \otimes \mathbf{z}_1), \mathbf{\Psi}), \quad (5)$$

where $\mathbf{x} \in \mathbb{R}^D$, $\mathbf{m}$, and $\mathbf{\Psi}$ are same as in FA. $\mathbf{W}_j \in \mathbb{R}^{D\times d_j}$ are the "biases" factor loadings, $\mathbf{T}_{(1)} \in \mathbb{R}^{D\times(d_1 d_2 d_3)}$ is the matricization of the tensor $\mathcal{T}$, and "$\otimes$" is the Kronecker product operator. Multiplicative interactions are due to the term: $\mathbf{z}_3 \otimes \mathbf{z}_2 \otimes \mathbf{z}_1$, which is a vector with dimensionality of $d_1 d_2 d_3$. We note that

$$\mathbf{T}_{(1)}(\mathbf{z}_3 \otimes \mathbf{z}_2 \otimes \mathbf{z}_1) = \sum_{i,j,k} \mathbf{t}_{(:,i,j,k)}\mathbf{z}_1(i)\mathbf{z}_2(j)\mathbf{z}_3(k),$$

where $\mathbf{z}_1(i)$ is the i-th element of vector $\mathbf{z}_1$, and $\mathbf{t}$ is the mode-1 fiber of $\mathcal{T}$. $\mathbf{T}_{(1)}(\mathbf{z}_3 \otimes \mathbf{z}_2 \otimes \mathbf{z}_1)$ is also equivalent to $\mathbf{\Lambda}_{new}\mathbf{z}_1$.

For clarity, we can concatenate the factors and loading matrices: let $\mathbf{y} \in \mathbb{R}^{d_1+d_2+d_3+1} \triangleq [\mathbf{z}_1; \mathbf{z}_2; \mathbf{z}_3; 1]$; $\mathbf{W} \in \mathbb{R}^{D\times(d_1+d_2+d_3+1)} \triangleq [\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3, \mathbf{m}]$; and $\mathbf{u} \in \mathbb{R}^{d_1 d_2 d_3} = \mathbf{z}_3 \otimes \mathbf{z}_2 \otimes \mathbf{z}_1$. The joint log-likelihood of the TA is:

$$\log p(\mathbf{x}, \mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3) = \sum_{j=1}^3 \left(-\frac{d_j}{2}\log(2\pi) - \frac{1}{2}\mathbf{z}_j^\mathsf{T}\mathbf{z}_j \right)$$
$$- \frac{D}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{\Psi}| - \frac{1}{2}(\mathbf{x}-\mathbf{e})^\mathsf{T}\mathbf{\Psi}^{-1}(\mathbf{x}-\mathbf{e}),$$

where $\mathbf{e} = \mathbf{W}\mathbf{y} + \mathbf{T}_{(1)}\mathbf{u}$. The last term of the above equation indicates that the TA models contain higher-order interactions (squared of the outer product of all factors). In comparison, FAs have only 2nd-order interactions among its latent factors. Fig. 2 displays a visual diagram of the TA's generative process.

Conditioned on any two of the three groups of factors, e.g. $\mathbf{z}_2$ and $\mathbf{z}_3$, the log-likelihood of $\mathbf{x}$ and $\mathbf{z}_1$ becomes:

$$\log\ p(\mathbf{x}, \mathbf{z}_1|\mathbf{z}_2, \mathbf{z}_3) = -\frac{1}{2}\big((d_1 + D)\log(2\pi) + \mathbf{z}_1^\mathsf{T}\mathbf{z}_1 + \log|\mathbf{\Psi}| + (\mathbf{x}-\mathbf{e})^\mathsf{T}\mathbf{\Psi}^{-1}(\mathbf{x}-\mathbf{e})\big). \quad (6)$$

Here, $\mathbf{e}$ can be re-written as $(\mathbf{m} + \mathbf{W}_2\mathbf{z}_2 + \mathbf{W}_3\mathbf{z}_3) + (\mathbf{W}_1 + \mathcal{T} \bar{\times}_3 \mathbf{z}_3 \bar{\times}_2 \mathbf{z}_2)\mathbf{z}_1$. We can see that conditioned on $\mathbf{z}_2$ and $\mathbf{z}_3$, we have a FA with parameters (c.f. Eq. 1):

$$\boldsymbol{\mu} = \mathbf{m} + \mathbf{W}_2\mathbf{z}_2 + \mathbf{W}_3\mathbf{z}_3,$$
$$\mathbf{\Lambda} = \mathbf{W}_1 + \mathcal{T} \bar{\times}_3 \mathbf{z}_3 \bar{\times}_2 \mathbf{z}_2. \quad (7)$$

The marginal probability density function is a Gaussian: $p(\mathbf{x}|\mathbf{z}_2, \mathbf{z}_3) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \mathbf{\Lambda}\mathbf{\Lambda}^\mathsf{T} + \mathbf{\Psi})$.

4.1. Inference

Higher order interaction in the TA means that inference is more complicated, since the joint posterior $p(\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_3|\mathbf{x})$ has no closed-form solution. We resort to alternating Gibbs sampling by cycling through $p(\mathbf{z}_1|\mathbf{x}, \mathbf{z}_2, \mathbf{z}_3)$; $p(\mathbf{z}_2|\mathbf{x}, \mathbf{z}_1, \mathbf{z}_3)$; $p(\mathbf{z}_3|\mathbf{x}, \mathbf{z}_1, \mathbf{z}_2)$.

Conditioned on two groups of factors, the posterior of the third is simple as the model reduces to a FA:

$$p(\mathbf{z}_1|\mathbf{x}, \mathbf{z}_2, \mathbf{z}_3) = \mathcal{N}(\mathbf{z}_1|\mathbf{V}^{-1}\mathbf{\Lambda}^\mathsf{T}\mathbf{\Psi}^{-1}(\mathbf{x} - \boldsymbol{\mu}), \mathbf{V}^{-1}), \quad (8)$$

where $\mathbf{V} = \mathbf{I} + \mathbf{\Lambda}^\mathsf{T}\mathbf{\Psi}^{-1}\mathbf{\Lambda}$. $\boldsymbol{\mu}$, $\mathbf{\Lambda}$ are defined by Eq. 7.

Although inference involves a matrix inverse, it only has cost of $O(d^3)$, where $d << D$, is the dimension of a latent factor group. d can be small since the data is assumed to be explained by a low dimensional manifold. We provide detailed timing evaluations in Sec. 5.5.

4.2. Learning

Maximum likelihood learning of a TA is similar to FA and is straightforward using a stochastic variant of the EM algorithm (Levine & Casella, 2001). During the E-step, MCMC samples are drawn from the posterior distribution using alternating Gibbs sampling. In the M-step, the samples are used to approximate the sufficient statistics involving $\mathbf{u}$ and $\mathbf{y}$, followed by closed-form updates of the model parameters, $\theta = \{\mathbf{W}, \mathbf{T}_{(1)}, \mathbf{\Psi}\}$.

Algorithm 1 EM Learning for TA

1: Given training data with N samples: $\mathbf{X} \in \mathbb{R}^{D\times N}$
2: Initialize θ: $\{\mathbf{W}, \mathbf{T}_{(1)}\} \sim \mathcal{N}(0, .01^2)$, $\mathbf{\Psi} \leftarrow 10 * \text{std}(\mathbf{X})$.
repeat
//Approximate E-step:
for $n = 1$ **to** N **do**
3: Sample $\{\mathbf{z}_1^{(n)}, \mathbf{z}_2^{(n)}, \mathbf{z}_3^{(n)}\}$ from $p(\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3|\mathbf{x}^{(n)})$ using Eq. 8, and alternating between $\mathbf{z}_1, \mathbf{z}_2, \&\ \mathbf{z}_3$.
end for
//M-step:
4: Concatenate samples $\{\mathbf{z}_1^{(n)}, \mathbf{z}_2^{(n)}, \mathbf{z}_3^{(n)}\}$ into $\{\mathbf{y}^n, \mathbf{u}^n\}$
5: Approximate posterior expectations using samples: $E[\mathbf{y}_n] \simeq \mathbf{y}^n$, $E[\mathbf{u}_n\mathbf{y}_n{}^\mathsf{T}] \simeq \mathbf{u}^n\mathbf{y}^{n\mathsf{T}}$, etc.
6: Update $\{\mathbf{W}, \mathbf{T}_{(1)}, \mathbf{\Psi}\}$ using Eqs. 9, 10, and 11.
until convergence

The expected joint log-likelihood function is:

$$Q = E\Big[\log \prod_i^N (2\pi)^{-\frac{D}{2}}|\mathbf{\Psi}|^{-\frac{1}{2}} \exp\{-\frac{1}{2}(\mathbf{x}_i - \mathbf{e}_i)^\mathsf{T}\mathbf{\Psi}^{-1}(\mathbf{x}_i - \mathbf{e}_i)\}\Big]$$

Setting $\frac{\partial Q}{\partial \theta} = 0$, we have update equations (see Supp. Materials for the derivation):

$$\mathbf{W} = \Big(\sum_i^N \mathbf{x}_i E[\mathbf{y}_i^\mathsf{T}] - \mathbf{T}_{(1)}\sum_i^N E[\mathbf{u}_i\mathbf{y}_i^\mathsf{T}]\Big)\Big(\sum_i^N E[\mathbf{y}_i\mathbf{y}_i^\mathsf{T}]\Big)^{-1}, \quad (9)$$

$$\mathbf{T}_{(1)} = \Big(\sum_i^N \mathbf{x}_i E[\mathbf{u}_i^\mathsf{T}] - \mathbf{W}\sum_i^N E[\mathbf{y}_i\mathbf{u}_i^\mathsf{T}]\Big)\Big(\sum_i^N E[\mathbf{u}_i\mathbf{u}_i^\mathsf{T}]\Big)^{-1}, \quad (10)$$

$$\mathbf{\Psi} = \frac{1}{N} diag\Bigg\{\sum_i^N \Big(\mathbf{x}_i\mathbf{x}_i^\mathsf{T} - 2\mathbf{T}_{(1)}\big(E[\mathbf{u}_i]\mathbf{x}_i^\mathsf{T} - E[\mathbf{u}_i\mathbf{y}_i^\mathsf{T}]\mathbf{W}^\mathsf{T} - \frac{1}{2}E[\mathbf{u}\mathbf{u}^\mathsf{T}]\mathbf{T}_{(1)}^\mathsf{T}\big) - 2\mathbf{W}\big(E[\mathbf{y}_i]\mathbf{x}_i^\mathsf{T} - \frac{1}{2}E[\mathbf{y}_i\mathbf{y}_i^\mathsf{T}]\mathbf{W}^\mathsf{T}\big)\Big)\Bigg\}. \quad (11)$$

During learning, we have found that the Gibbs sampler mixes very fast and that a relatively small number (20 to 100) are needed to achieve good performance. It is important to stress that Gibbs sampling is efficient in TAs because the posterior for each group is exact when conditioned on all other groups. It also mixes quickly because even though the variables of the posterior are dependent, the posterior itself is likely to be unimodal: an image of the face is explained by 1 lighting code and 1 subject code (See Sec. 5.3). We provide trace plots of the latent variables to demonstrate the fast converegnce property in the Supp. Materials.

4.3. Likelihood Computation

For model comparison, we are interested in evaluating the data log-likelihood $\log p(\mathbf{x}|\theta)$. As noted in

Sec. 4, a TA with J groups of factors reduces to a FA when conditioned on $J-1$ factor groups. Utilizing the fact that $\log p(\mathbf{x}|\theta)$ can be easily computed (Eq. 2), a Monte Carlo estimation of data log-likelihood in TA can be performed by sampling from the prior of the $J-1$ groups of factors. For example, in a model TA$\{D, d_1, d_2\}$, $J=2$:

$$\log p(\mathbf{x}) = \log \int_{\mathbf{z}_2} p(\mathbf{x}|\mathbf{z}_2)p(\mathbf{z}_2)d\mathbf{z}_2 \simeq \log \frac{1}{K}\sum_{k=1}^{K} p(\mathbf{x}|\mathbf{z}_2^{(k)}), \quad \mathbf{z}_2^{(k)} \sim \mathcal{N}(0, I). \quad (12)$$

This simple estimator is asymptotically unbiased but has high variance unless the dimensionality of $\mathbf{z}_2$, or d_2, is very small. Since $\mathbf{z}_1$ can be analytically integrated out, the Monte Carlo technique can be accurate when only one factor group has large dimensionality.

For large d_j, however, simple Monte Carlo estimation is very inefficient, giving an estimator with large variance. In this situation, Annealed Importance Sampling (Neal, 2001) is a much better alternative. We can treat the problem of estimating $\log p(\mathbf{x})$ as calculating the partition function of unnormalized posterior distribution $p^*(\mathbf{z}|\mathbf{x}) \triangleq p(\mathbf{x}, \mathbf{z})$, where $p^*(\cdot)$ denotes an unnormalized distribution. The basic Importance Sampling gives:

$$p(\mathbf{x}) = \int_{\mathbf{z}} d\mathbf{z}\ \frac{p^*(\mathbf{z}|\mathbf{x})}{q(\mathbf{z})}q(\mathbf{z}) \simeq \frac{1}{M}\sum_{i}^{M} w^{(i)}, \quad w^{(i)} = p^*(\mathbf{z}^{(i)}|\mathbf{x})/q(\mathbf{z}^{(i)}), \quad z^{(i)} \sim q(\mathbf{z}). \quad (13)$$

AIS provides a better estimate by first sampling from a tractable base distribution $q(\mathbf{z})$. Subsequent MCMC steps are taken in a set of intermediate distribution, annealing to the distribution of interest: $p(\mathbf{z}|\mathbf{x})$. Annealing allows for a much better estimate of $w^{(i)}$. For TAs, we assume the base distribution is the prior over the factors: $q(\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3) = \prod_j^3 p(\mathbf{z}_j)$. An intermediate distribution is defined as:

$$p_\beta(\{\mathbf{z}_j\}) \propto q(\{\mathbf{z}_j\})^{1-\beta} p^*(\{\mathbf{z}_j\}|\mathbf{x})^\beta = p(\{\mathbf{z}_j\})p^\beta(\mathbf{x}|\{\mathbf{z}_j\}),$$

where β is a scalar which varies from 0.0 to 1.0, as we anneal from the prior to the posterior. Derivations and experiments with the AIS estimator is provided in Supp. Materials.

4.4. Equality Constraints

An equality constraint indicates that a subset $\{\mathbf{x}^{(k)}\}_{k=1}^K$ of the training data have the same factor values for the j-th factor group $\mathbf{z}_j$. For example, if group $j=1$ represents the identity of a person and group $j=2$ represents lighting directions, an equality constraint on group 1 indicates that all images of the subset are from the same person, while an equality constraint on group 2 indicates that images of this subset are from the same lighting conditions.

During learning, the availability of equality constraints will only change the inference step. Assuming we have constraints for the factor group $j=1$, the posterior for $\mathbf{z}_j$ (Eq. 8) will be modified as follows:

$$p(\mathbf{z}_1|\{\mathbf{x}^{(k)}\}, \{\mathbf{z}_2^{(k)}\}, \{\mathbf{z}_3^{(k)}\}) = \mathcal{N}(\mathbf{z}_1|\tilde{\mathbf{V}}^{-1}\sum_{k=1}^{K}\{\mathbf{\Lambda}^{(k)\mathsf{T}}\mathbf{\Psi}^{-1}(\mathbf{x}^{(k)} - \boldsymbol{\mu}^{(k)})\}, \tilde{\mathbf{V}}^{-1}), \quad (14)$$

where $\tilde{\mathbf{V}} = \mathbf{I} + \sum_{k=1}^K \mathbf{\Lambda}^{(k)\mathsf{T}}\mathbf{\Psi}^{-1}\mathbf{\Lambda}^{(k)}$; $\mathbf{\Lambda}^{(k)} = \mathbf{W}_1 + \mathcal{T} \bar{\times}_3 \mathbf{z}_3^{(k)} \bar{\times}_2 \mathbf{z}_2^{(k)}$; and $\boldsymbol{\mu}^{(k)} = \mathbf{m} + \mathbf{W}_2\mathbf{z}_2^{(k)} + \mathbf{W}_3\mathbf{z}_3^{(k)}$. The M-step is not affected by the presence of equality constraints, so TAs can learn when equality constraints are provided for arbitrary subsets of the data.

4.5. Mixture of Tensor Analyzers

Extending TAs to Mixture of Tensor Analyzers (MTAs) is straightforward, as Sec. 4.3 showed how $p(\mathbf{x}|c)$ can be efficiently approximated. Each component c will have its own parameters $\theta_c = \{\mathbf{W}_c, \mathbf{T}_{(1),c}, \mathbf{\Psi}_c\}$. Posterior distribution over the factors and components can be decomposed as: $p(\{\mathbf{z}_j\}, c|\mathbf{x}) = p(\{\mathbf{z}_j\}|\mathbf{x}, c)p(c|\mathbf{x})$, where $p(\{\mathbf{z}_j\}|\mathbf{x}, c)$ can be sampled using Eq. 8.

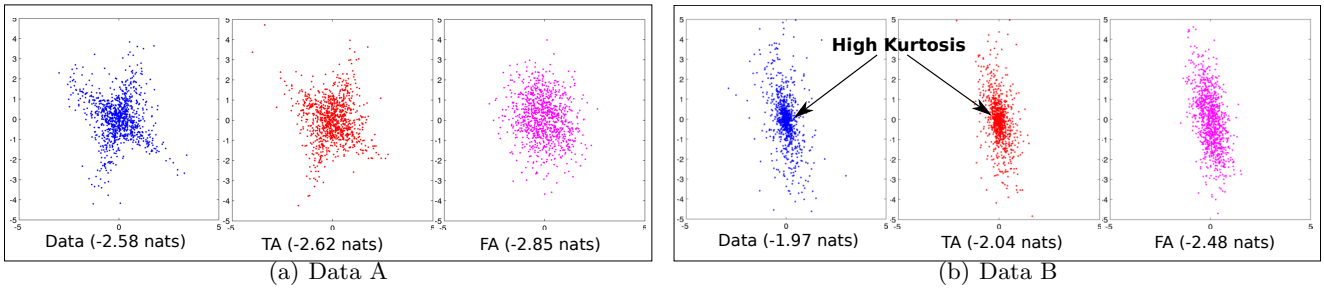
MTAs should be used instead of TAs when modeling highly multimodal data with multiplicative interactions such as multiple types of objects under varying illumination.

For our experiments with MTAs, $p(c|\mathbf{x})$ is approximated with Eq. 12 using 1000 samples per mixture component. For our natural images experiment, it means only 180 ms per mixture components is required, see sec 5.2.

5. Experiments

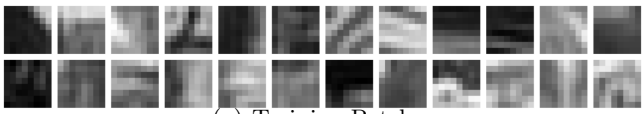
5.1. Synthetic Data

As a proof of concept, we compared TA to FA on two synthetic datasets (Fig. 3 A & B). Data A is highly structured and is generated using a TA with random parameters. Data B has high kurtosis, with density concentrated at the origin. For both datasets, we learned using a TA$\{D=2, d_1=2, d_2=2\}$ and a FA with the same number of parameters as the TA. The TAs performed model recovery nicely. The left panel of Fig. 3(a) displays training points. The data

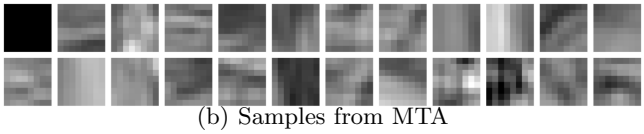


(a) Data A (b) Data B

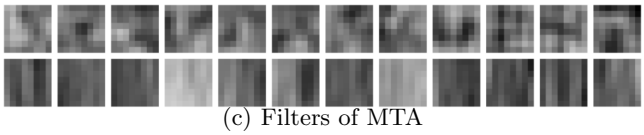
Figure 3. TA vs. FA on 2D synthetic datasets. TAs can better model complex densities.



(a) Training Patches



(b) Samples from MTA



(c) Filters of MTA

Indp. Pixel	ICA	GRBM	DBN	Fac. Bilinear
78.3	135.7±1.2	137.8±1.0	144.4±1.1	145.9±2.7
Single FA	**MFA**	**Deep MFA**	**GMM**	**MTA**
130.1±0.6	166.5±1.8	169.3±1.5	167.2	158.2±3.3

(d) Average test log-probability (measured in nats)

Figure 4. Natural image patches. (a) Training data. (b) Samples from MTA. (c) Each row contains filters from a different MTA component. (d) Average test log-probability comparisons. ICA: Independent Component Analysis. GRBM: Gaussian Restricted Boltzmann Machine (Hinton & Salakhutdinov, 2006). DBN: Deep Belief Nets (Hinton et al., 2006). Fac. Bilinear: (Culpepper et al., 2011).

log-likelihood of the true model is -2.58. The middle panel plots the samples of a TA, which achieved the log-probability of -2.62 on the training data. The right panel plots samples drawn from a FA. Likewise in Fig. 3(b), TA is a significantly better model than the FA: -2.04 ± 0.05 to -2.48 ± 0.09. We also tested mixtures of TAs vs mixtures of FA (MFA) on data generated by randomly initialized MFA models. The performance of MTA and MFA were very similar, demonstrating that (M)TAs can also efficiently emulate (M)FAs when necessary.

5.2. Natural Images

Learning a good density model of natural images is useful for image denoising and inpainting. Following (Zoran & Weiss, 2011), we compared MTAs to MFAs and other models on modeling image patches. Two million 8×8 patches were extracted from the training set of the Berkeley Natural Images database (Martin et al., 2001) for training, while 50,000 patches from the test set were extracted for testing. The DC component of each image patches were set to 0 by subtracting the patch mean form every pixel. For MTAs and MFAs[2], we used 200 components and selected the number of latent factors based on cross validation. For MTAs, 64 factors were used while each component of the MTA is a TA{64, 64, 5}. For the Factorized Bilinear model, we used code from the authors[3]. Indp. Pixel and GMM results are from (Zoran & Weiss, 2011), while the Deep MFA result is from (Tang et al., 2012).

After training, Fig. 4(b) shows that samples of a MTA matches closely to the training patches in Fig. 4(a). Each row of Fig. 4(c) shows filters (fibers of the tensor) of one of the MTA components. Components of MTA specialize to model patches of different spatial frequencies and orientations. Fig. 4(d) lists quantitative evaluations of the different models. Although MTAs achieve good results, they do not outperform MFAs or GMMs with 200 mixture components[4].

We also compared the performance of MFAs vs. MTAs when the number of mixture components is smaller. Fig. 5 shows the average log-likelihoods on the test set. Due to the fact that MTAs can model higher-order interaction, it is able to outperform MFAs with 50 or less mixture components. However, as the number of components increases, the advantage of MTAs disappears.

5.3. Face Recognition with equality constraints

In a one-shot learning setting, classifiers must be trained from only one example per class. For face

[2]MFA code (Verbeek, 2006) is downloaded from http://lear.inrialpes.fr/∼verbeek/software.php

[3]https://github.com/jackculpepper/dir-linear.

[4]The gap (less than 10 nats) in performance can be attributed to the fact that the E-step in MTA is not exact and requires sampling, while MFA has a closed-form posterior. Sampling introduces noise which adversely affects the M-step updates.

(a) Test (b) Fixing Identity (c) Fixing Lighting

Tasks	Human	NN	XCORR	SVM	FA	DLN	CP	NTF	Tucker	pTucker	S&C	GPSC	TA
One-shot recognition	9.3	51.1	33.6	35.5	19.6	19	N/A	N/A	N/A	N/A	24.0	N/A	**6.4**
Using lighting labels							18.3	20.4	37.3	23.6	21.7	14.5	

(d) Recognition Error in %

Figure 6. Tensor Analyzer is able to simultaneously decompose a test image into separate identity and lighting factors. (a) 4 test images with frontal, left, right and top light sources. (b) Random samples with identity factor fixed to the inferred values from the test faces in (a). (c) Random samples with lighting factor fixed to the inferred values. (d) Comparison to other methods. Tensor decomposition methods require additional labels of lighting direction during the test phase.

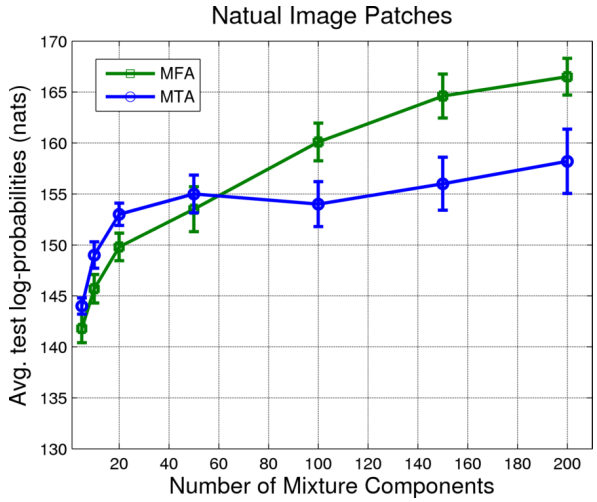


Figure 5. Performance of MTA vs. MFA with different number of mixture components.

recognition, only one example per test subject is used for training. We use the Yale B database and its Extended version (Lee et al., 2005). The database contains 38 subjects under 45 different lighting conditions. We use 28 subjects for training and test on the 10 subjects from the original Yale B database. The images are first downsampled to 24×24, and we used a TA$\{576, 80, 4\}$, which contains 2 groups of factors.

The learned TA allows for strong generalization to new people under new lighting conditions. It achieves an average test log-probability of $\mathbf{836 \pm 7}$ on the images of the 10 held-out subjects. As a comparison, the best MFA model achieved only 791 ± 10. The number of components and factors of the MFA are selected using grid search. The gain of 45 nats demonstrates a significant win for the TA.

Qualitatively, to see how well the TA is able to factor out identity from lighting, we first sample from the posterior distribution conditioned on a single test image using step 3 of Alg. 1. We then fix a factor group's sampled values and sample the factors of the other group using its Normal prior. Results are shown in Fig. 6. A row in panel (b) shows the same person under different sampled lighting conditions. A row in panel (c) shows sampled people, but under the same lighting condition. We emphasize that only a *single* test image from *novel* subjects is used for inference (panel (a)).

The one-shot recognition task consists of using a *single* image of each of the 10 test subjects for training during the *testing* phase. The rest of the images of the test subjects are then classified into 1 of 10 classes. Labels indicating lighting directions of the test images can be optionally provided. TAs can operate with or without the labels, while tensor decomposition methods require these labels. We first use the 28 training subjects to learn the parameters for the TA$\{576,80,4\}$ in the training phase. Equality constraints specifying which images have the same identity or lighting type are used during training. During the testing phase, a single image for each of the 10 test subjects is used to compute the mean of the posterior $p(\mathbf{z}_{identity}, \mathbf{z}_{light}|\mathbf{x})$, using 200 Gibbs steps. For each one of the 10 sampled $\mathbf{z}_{identity}$, TA$\{576,80,4\}$ is transformed into a person-specific FA using Eqs. 6, 7. The resulting 10 FAs define the class-conditionals of a generative classifier, which we use to classify test images of the 10 test subjects.

We compared TA to standard classifiers and tensor decomposition methods. Human error is the average of

testing several human subjects on the same one-shot recognition task. NN is the nearest neighbour classifier, and XCORR represents the normalized Cross Correlation. Multiclass linear SVM from the LIBLINEAR package (Fan et al., 2008) is used, where the hyperparameter C was chosen using validation. Factor analyzers method requires the transfer of the factor loadings from the training phase. We first learned 28 FAs, one for each training subject. During the testing phase, each one of the 10 training images of the test subject is matched with the FA which gives it the largest likelihood. A new FA is created, centered at the training image. The rest of the parameters are transferred from the matched FA. In essence, the transferred loadings model the lighting variations of the training subjects (1 of 28), which is most similar to the test phase training image. After the creation of these 10 new FAs, classification is same as in the TA method.

We also compared TA to various Tensor Decomposition methods, including CP: CANDECOMP/PARAFAC (Carroll & Chang, 1970); NTF: Nonnegative Tensor Factorization (Shashua & Hazan, 2005); Tucker (Tucker, 1963); and Probabilistic Tucker decomposition (Chu & Ghahramani, 2009). For CP, Tucker, and NTF, we used the N-way toolbox (Bro, 1998). For the S&C bilinear method, we implemented the exact algorithm as stated in Sec. 3.2 of (Tenenbaum & Freeman, 2000). The style and content codes are adapted for new test images during one-shot learning. For GPSC, the code provided by the authors was used. In all experiments, hyper-parameters were selected by cross-validation. Recognition errors are shown in Fig. 6(d). Observe that the TA not only outperforms these methods by a significant margin, but it even achieves better than human performance.

5.4. Learning with incomplete equality constraints

We now demonstrate the advantage of the TA in a semi-supervised setting on the UMIST face database (Graham & Allinson, 1998). It contains 20 subjects with 20 to 40 training images per subject. The variation consists of in-depth head rotations. We compared the TA to S&C model and NN on the one-shot face recognition task. Out of the 20 subjects, 15 were used for training and 5 for testing. The split was randomized over 10 different trials. The images are downsampled to the resolution of 24×24. We experimented with equality constraints, using 3, 4, or 5 images per training subject. During the training phase, a TA{576,3,5} model was trained using 30 EM iterations. The algorithms for classification are exactly the same as in Sec. 5.3. Fig. 7 plots recognition errors as

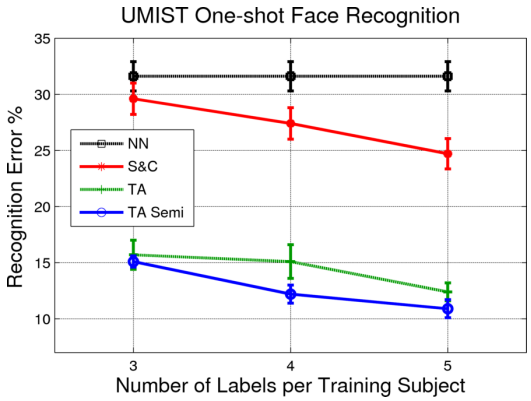


Figure 7. UMIST Recognition errors.

a function of the number of images per subject used during the *training phase*. For the case where 5 images per training subjects are available, TA achieves significantly lower errors at 12.4% compared to 24.9% for S&C. If we add an equal number of images without equality constraints during training, the error is further reduced to **10.9%**. (An additional experiment on concept learning from colorful shapes is presented in the Supp. Materials).

5.5. Computation Time

Our experiments used Matlab on a standard multicore workstation, we report the computation time required for inference and learning. MTA on face images in Sec. 5.3: inference per image per Gibbs update takes 14 ms (milliseconds), while learning took a total of 587 secs. MFA learning took around 108 secs. In practice, matrix inversion is not very expensive in our model. For the face recognition model TA{576,80,4}, while having the same number of parameters as a FA with 320 latent factors (which requires inverting 320x320 matrix), only require inversion of 80x80 and 4x4 matrices, which takes 0.4 ms. MTA on natural images: inference per image per Gibbs update takes 47 ms, while the entire learning algorithm took 33 hours. In comparison, inference per image in Mixture of Factor Analyzers (MFA) takes 18 ms, while learning took 9.5 hours.

6. Conclusions

We have introduced a new density model which extends Factor Analysis to modeling multilinear interactions. Using efficient alternating sampling and the EM algorithm, we have shown that (M)TAs can learn more complex densities and separate factors of variation, leading to the learning of simple concepts. Moreover, at the important task of one-shot face recognition, TAs outperform a variety of other models.

References

Belhumeur, P. N. and Kriegman, D. J. What is the set of images of an object under all possible lighting conditions. In *CVPR*, pp. 270–277, 1996.

Bro, Rasmus. *Multi-way Analysis in the Food Industry. Models, Algorithms and Applications.* PhD thesis, University of Amsterdam, The Netherlands, 1998. URL `www.models.kvl.dk/users/rasmus/thesis/thesis.html`.

Carroll, J. and Chang, Jih J. Analysis of individual differences in multidimensional scaling via an n-way generalization of 'Eckart-Young' decomposition. *Psychometrika*, 35(3):283–319, September 1970.

Chu, Wei and Ghahramani, Zoubin. Probabilistic models for incomplete multi-dimensional arrays. *Journal of Machine Learning Research - Proceedings Track*, 5:89–96, 2009.

Culpepper, Benjamin J., Sohl-Dickstein, Jascha, and Olshausen, Bruno A. Building a better probabilistic model of images by factorization. In *ICCV*, pp. 2011–2017, 2011.

Fan, Rong-En, Chang, Kai-Wei, Hsieh, Cho-Jui, Wang, Xiang-Rui, and Lin, Chih-Jen. LIBLINEAR: A library for large linear classification, August 2008.

Graham, Daniel B and Allinson, Nigel M. Characterizing virtual eigensignatures for general purpose face recognition. In *Face Recognition: From Theory to Applications*, pp. 446–456. 1998.

Grimes, David B. and Rao, Rajesh P. Bilinear sparse coding for invariant vision. *Neural Computation*, 17(1), January 2005.

Hinton, G. E. and Salakhutdinov, R. Reducing the dimensionality of data with neural networks. *Science*, 313: 504–507, 2006.

Hinton, G. E., Osindero, S., and Teh, Y. W. A fast learning algorithm for deep belief nets. *Neural Computation*, 18 (7):1527–1554, 2006.

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

Lathauwer, Lieven De and Vandewalle, Joos. Dimensionality reduction in higher-order signal processing and rank-$(r_1,r_2,...,r_n)$ reduction in multilinear algebra, 2004.

Lee, K. C., Ho, Jeffrey, and Kriegman, David. Acquiring linear subspaces for face recognition under variable lighting. *IEEE PAMI*, 27:684–698, 2005.

Levine, Richard A. and Casella, George. Implementations of the Monte Carlo EM Algorithm. *Journal of Computational and Graphical Statistics*, 10(3):422–439, 2001.

Martin, D., Fowlkes, C., Tal, D., and Malik, J. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, volume 2, pp. 416–423, July 2001.

Neal, R. M. Annealed importance sampling. *Statistics and Computing*, 11:125–139, 2001.

Rubin, Donald B. and Thayer, Dorothy T. EM algorithms for the ML factor analysis. *Psychometrika*, 47(1):69–76, 1982.

Shashua, Amnon and Hazan, Tamir. Non-negative tensor factorization with applications to statistics and computer vision. In *Proc. of ICML*, pp. 792–799. ICML, 2005.

Sun, Jimeng, Tao, Dacheng, and Faloutsos, Christos. Beyond streams and graphs: dynamic tensor analysis. In *KDD*, 2006.

Tang, Yichuan, Salakhutdinov, Ruslan, and Hinton, Geoffrey. Deep mixtures of factor analysers. In *Proc. of ICML 2012, Edinburgh, Scotland*, 2012.

Tenenbaum, Joshua B. and Freeman, William T. Separating style and content with bilinear models. *Neural Computation*, pp. 1247–1283, 2000.

Tucker, L. R. Implications of factor analysis of three-way matrices for measurement of change. In Harris, C. W. (ed.), *Problems in measuring change.*, pp. 122–137. University of Wisconsin Press, Madison WI, 1963.

Vasilescu, M. A. O. and Terzopoulos, D. Multilinear analysis of image ensembles: Tensorfaces. In *ECCV*, pp. 447–460, 2002.

Verbeek, Jakob. Learning nonlinear image manifolds by global alignment of local linear models. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28:14, 2006.

Wang, Hongcheng and Ahuja, Narendra. Facial expression decomposition. In *ICCV*, pp. 958–965, 2003.

Wang, Jack M., Fleet, David J., and Hertzmann, Aaron. Multifactor gaussian process models for style-content separation. In *ICML*, pp. 975–982, 2007.

Xu, Zenglin, Yan, Feng, and Qi, Yuan. Infinite tucker decomposition: Nonparametric bayesian models for multiway data analysis. In *Proc. of ICML, 2012, Edinburgh, Scotland*, 2012.

Yang, Ming-Hsuan, Ahuja, Narendra, and Kriegman, David. Face detection using a mixture of factor analyzers. In *ICIP*, Kobe, Japan, 1999.

Zoran, D. and Weiss, Y. From learning models of natural image patches to whole image restoration. ICCV, 2011.