Tensor Analyzers

Anonymous Author(s)
Affiliation
Address
email

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 a fairly 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 and of images containing a variety of simple shapes that vary in size and color. 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 mixtures of Tensor Analyzers outperform mixtures of Factor Analyzers at modeling natural image patches and artificial data produced using multiplicative interactions.

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 [20, 18]. 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.

In this paper, we introduce Tensor Analyzers (TAs) which are one way of generalizing FAs to include multiplicative interactions between latent variables from different groups. A TA generates data in the following way: First we sample all of the factor values in all of the groups independently from a zero-mean unit-variance Gaussian. Then, for each possible way of choosing one factor from each latent group, we take the product of the factor values and contribute this product times a learned parameter to each component of the generated data-vector. Finally, we add independent Gaussian noise to every component of the generated vector with the variance of the noise being learned separately for each component. The number of "tensor loading" parameters is the product of the sizes of all the factor groups and the dimensionality of the data-vector. We may additionally use lower order products that do not include a factor from every group, but in this paper the only lower-order products we use are the individual factors of every group. FAs are special cases of TAs that contain only one group of factors.

Conditioned on all but one group of factors, a TA reduces to an ordinary FA in which 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 straightforward. 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.

The related tensor decomposition has been widely applied to signal processing and computer vision [16, 2, 9, 19]. The SVD algorithm was used to learn a bilinear model to separate style and content [15] (henceforth referred to as the S&C model). Tucker decomposition was applied to a 5-mode array of face images in [17], finding multilinear bases called TensorFaces. However, tensor decomposition based methods lack a probabilistic formulation and are not density models. Inference given a *single* new test case is ill-posed and can be ad hoc[1]. Moreover, training data must be arranged into a tensor [11]. More recently, bilinear models with priors on the latent variables have been proposed. In [6], sparsity is induced in the codes of a bilinear model to learn translational invariant representation from video. However, the model does not try to maximize the $\log p(\mathbf{x})$ but instead finds the MAP estimate of the code activations à la sparse coding. [3] describe an outer-product factorization of the bilinear model that is trained as a density model using EM, but expensive Hamiltonian dynamics is 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.

In contrast, TAs do not have any of the above deficiencies. A TA is a density model that can be learned directly from data vectors in an entirely unsupervised manner, but 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. 3.4). 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 (Sec. 4.3). It can also be easily extended to a mixture model, provided we are willing to compute approximate densities as described in (Sec. 3.3). In addition, posterior inference for a *single* test case is simple and accurate, as demonstrated by one-shot face recognition experiments of Sec. 4.4.

2 Preliminaries

Following [8], 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, $\boldsymbol{t}_{(:,j,k)}$ 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 $D_1 \times \cdots \times D_{n-1} \times D_{n+1} \times \cdots \times D_N$.

2.1 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 a directed model, defined as by a prior and likelihood:

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

$\mathbf{I}$ 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; and 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:

$$p(\mathbf{x}) = \int_{\mathbf{z}} p(\mathbf{x}|\mathbf{z}) p(\mathbf{z}) d\mathbf{z} = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \Gamma), \quad \Gamma = \mathbf{\Lambda \Lambda}^{\mathsf{T}} + \mathbf{\Psi} \tag{2}$$

[1] E.g., the asymmetric model in (S&C) requires EM learning of a separate model during test time.

For inference, we are interested in the posterior, which is also a multivariate Gaussian:

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

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 [14]. During the E-step, Eqs. 3 are used to compute the posterior sufficient statistics given the current setting of the model parameters. 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 $\theta = \{\mathbf{\Lambda}, \boldsymbol{\mu}, \mathbf{\Psi}\}$.

3 Tensor Analyzers

FA generates data by linear combining factors, as there are no multiplicative interactions involving terms such as $z_i z_j, i \neq j$. By using a $(J+1)$-order "loading" tensor $\mathcal{T} \in \mathbb{R}^{D \times d_1 \times \cdots \times d_J}$, a TA can model multiplicative interactions among its J groups of factors: $\{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_J\}$. For $j = 1, \ldots, J$, $\mathbf{z}_j \in \mathbb{R}^{d_j}$. We will use the notation TA$\{D, d_1, d_2, \ldots, d_J\}$ to denote the aforementioned TA with latent factors $\{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_J\}$ and interaction tensor $\mathcal{T}$. We give each group of factors a standard Normal prior:

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

For clarity and WLOG, we assume $J = 3$ for the following equations. The likelihood is defined as:

$$p(\mathbf{x}|\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3) = \mathcal{N}(\mathbf{x}|\mathbf{m} + \mathbf{W}_1\mathbf{z}_1 + \mathbf{W}_2\mathbf{z}_2 + \mathbf{W}_3\mathbf{z}_3 + \mathbf{T}_{(1)}(\mathbf{z}_3 \otimes \mathbf{z}_2 \otimes \mathbf{z}_1), \mathbf{\Psi}), \tag{5}$$

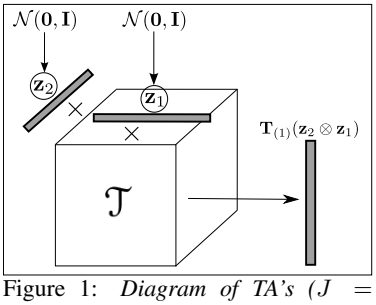


Figure 1: *Diagram of TA's ($J = 2$) generative process. Vector-tensor multiplications determines the mean of $p(\mathbf{x}|\mathbf{z}_1, \mathbf{z}_2)$.*

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 $\mathcal{T}$ as discussed in Sec. 2, 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 $d_1 \times d_2 \times d_3$.

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 $\mathbf{u} \in \mathbb{R}^{d_1 d_2 d_3} = \mathbf{z}_3 \otimes \mathbf{z}_2 \otimes \mathbf{z}_1$. We note that $\mathbf{T}_{(1)}(\mathbf{z}_3 \otimes \mathbf{z}_2 \otimes \mathbf{z}_1)$ is mathematically equivalent to $\sum_{i,j,k} \mathbf{t}_{(:,i,j,k)} \mathbf{z}_1(i) \mathbf{z}_2(j) \mathbf{z}_3(k)$, where $\mathbf{z}_1(i)$ is the i-th element of vector $\mathbf{z}_1$, and $\mathbf{t}$ is the mode-1 fiber of $\mathcal{T}$.

The joint/complete log-likelihood of the TA is:

$$\log p(\mathbf{x}, \mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3) = \sum_{j=1}^{3} \left(-\frac{d_j}{2} \log(2\pi) - \frac{1}{2}\mathbf{z}_j^\mathsf{T}\mathbf{z}_j \right) - \frac{D}{2} \log(2\pi) - \frac{1}{2} \log |\mathbf{\Psi}| - \frac{1}{2}(\mathbf{x} - \mathbf{e})^\mathsf{T}\mathbf{\Psi}^{-1}(\mathbf{x} - \mathbf{e}), \tag{6}$$

where $\mathbf{e} = \mathbf{W}\mathbf{y} + \mathbf{T}_{(1)}\mathbf{u}$. See Fig. 1 for a visual diagram of the TA's generative process. The last term in Eq. 6 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.

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:

$$\log p(\mathbf{x}, \mathbf{z}_1|\mathbf{z}_2, \mathbf{z}_3) = -\frac{d_1}{2} \log(2\pi) - \frac{1}{2}\mathbf{z}_1^\mathsf{T}\mathbf{z}_1 - \frac{D}{2} \log(2\pi) - \frac{1}{2} \log |\mathbf{\Psi}| - \frac{1}{2}(\mathbf{x} - \mathbf{e})^\mathsf{T}\mathbf{\Psi}^{-1}(\mathbf{x} - \mathbf{e}) \tag{7}$$

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):

$$\boldsymbol{\mu} = \mathbf{m} + \mathbf{W}_2\mathbf{z}_2 + \mathbf{W}_3\mathbf{z}_3, \;\; \mathbf{\Lambda} = \mathbf{W}_1 + \mathcal{T} \;\bar{\times}_3\; \mathbf{z}_3 \;\bar{\times}_2\; \mathbf{z}_2 \tag{8}$$

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

3.1 Inference

Higher order interaction in the TA means that inference is more complicated as 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: Step 1: $p(\mathbf{z}_1|\mathbf{x}, \mathbf{z}_2, \mathbf{z}_3)$; Step 2: $p(\mathbf{z}_2|\mathbf{x}, \mathbf{z}_1, \mathbf{z}_3)$; Step 3: $p(\mathbf{z}_3|\mathbf{x}, \mathbf{z}_1, \mathbf{z}_2)$.

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

$$p(\mathbf{z}_1|\mathbf{x}, \mathbf{z}_2, \mathbf{z}_3) = \mathcal{N}(\mathbf{z}_1|\mathbf{V}^{-1}\mathbf{\Lambda}^\top\mathbf{\Psi}^{-1}(\mathbf{x} - \boldsymbol{\mu}), \mathbf{V}^{-1}), \tag{9}$$

where $\mathbf{V} = \mathbf{I} + \mathbf{\Lambda}^\top\mathbf{\Psi}^{-1}\mathbf{\Lambda}$. $\boldsymbol{\mu}$, and $\mathbf{\Lambda}$ are defined by Eq. 8.

3.2 Learning

Maximum likelihood learning of a TA is similar to FA and is straightforward using a variant of the EM algorithm. 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, $\theta = \{\mathbf{W}, \mathbf{T}_{(1)}, \mathbf{\Psi}\}$. The expected joint log-likelihood function is:

$$Q = E\Big[\log \prod_i^N (2\pi)^{-D/2}|\mathbf{\Psi}|^{-1/2} \exp\{-\frac{1}{2}(\mathbf{x}_i - \mathbf{e}_i)^\top\mathbf{\Psi}^{-1}(\mathbf{x}_i - \mathbf{e}_i)\}\Big] \tag{10}$$

Setting $\frac{\partial Q}{\partial \theta} = 0$, we have update equations:

$$\mathbf{W} = \Big(\sum_i^N \mathbf{x}_i E[\mathbf{y}_i^\top] - \mathbf{T}_{(1)} \sum_i^N E[\mathbf{u}_i\mathbf{y}_i^\top]\Big)\Big(\sum_i^N E[\mathbf{y}_i\mathbf{y}_i^\top]\Big)^{-1} \tag{11}$$

$$\mathbf{T}_{(1)} = \Big(\sum_i^N \mathbf{x}_i E[\mathbf{u}_i^\top] - \mathbf{W} \sum_i^N E[\mathbf{y}_i\mathbf{u}_i^\top]\Big)\Big(\sum_i^N E[\mathbf{u}_i\mathbf{u}_i^\top]\Big)^{-1} \tag{12}$$

$$\mathbf{\Psi} = \frac{1}{N} diag\Big\{\sum_i^N \Big(\mathbf{x}_i\mathbf{x}_i^\top - 2\mathbf{T}_{(1)}(E[\mathbf{u}_i]\mathbf{x}_i^\top - E[\mathbf{u}_i\mathbf{y}_i^\top]\mathbf{W}^\top - \frac{1}{2}E[\mathbf{u}\mathbf{u}^\top]\mathbf{T}_{(1)}^\top) - 2\mathbf{W}(E[\mathbf{y}_i]\mathbf{x}_i^\top - \frac{1}{2}E[\mathbf{y}_i\mathbf{y}_i^\top]\mathbf{W}^\top)\Big)\Big\} \tag{13}$$

(See Supplementary Materials for the derivation.)

Algorithm 1 EM Learning for TA

Given training data with N samples: $\mathbf{X} \in \mathbb{R}^{D\times N}$
Initialize θ: $\{\mathbf{W}, \mathbf{T}_{(1)}\} \sim \mathcal{N}(0, .01^2)$, $\mathbf{\Psi} \leftarrow 10 * \text{stddev}(\mathbf{X})$.
repeat
  //Approximate E-step:
  for $n = 1$ **to** N **do**
    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. 9, and alternating between $\mathbf{z}_1, \mathbf{z}_2, \& \mathbf{z}_3$.
  end for
  //M-step:
  Concatenate samples $\{\mathbf{z}_1^{(n)}, \mathbf{z}_2^{(n)}, \mathbf{z}_3^{(n)}\}$ into $\{\mathbf{y}^n, \mathbf{u}^n\}$
  Approximate posterior expectations using samples: $E[\mathbf{y}_n] \simeq \mathbf{y}^n$, $E[\mathbf{u}_n\mathbf{y}_n{}^\top] \simeq \mathbf{u}^n\mathbf{y}^{n\top}$, etc.
  Update $\{\mathbf{W}, \mathbf{T}_{(1)}, \mathbf{\Psi}\}$ using Eqs. 11, 12, and 13.
until convergence

3.3 Likelihood Computation

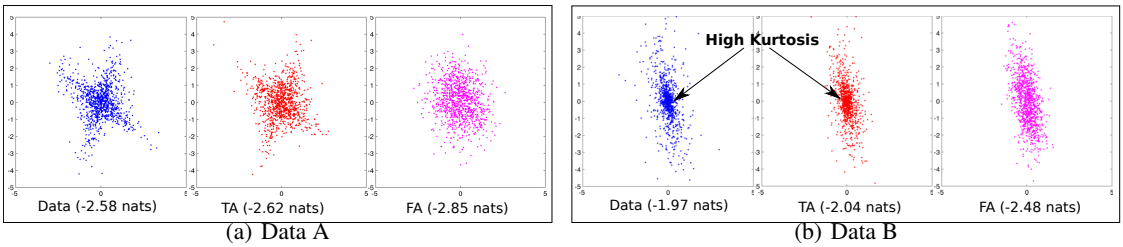
For model comparison, we are interested in evaluating the data log-likelihood $\log p(\mathbf{x}|\theta)$. As noted in Sec. 3, a TA with J groups of factors reduces to a FA when conditioned on $J - 1$ factor groups. Utilizing the fact that $\log p(\mathbf{x}|\theta)$ can be easily computed (Eq. 2), a 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$\{D, d_1, d_2\}$, $J = 2$:

$$\log p(\mathbf{x}) = \log \int_{\mathbf{z}_2} p(\mathbf{x}|\mathbf{z}_2)p(\mathbf{z}_2)d\mathbf{z}_2 \simeq \log \frac{1}{K}\sum_{k=1}^K p(\mathbf{x}|\mathbf{z}_2^{(k)}), \qquad \mathbf{z}_2^{(k)} \sim \mathcal{N}(0, I) \tag{14}$$

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.

AIS estimation of likelihood

For large d_j, however, simple Monte Carlo estimation is very inefficient as random samples from the prior will mostly yield close to zero log-likelihoods, therefore giving an estimator with large variance. In this situation, Annealed Importance Sampling [13] is a much better alternative. We can treat the problem of estimating $\log p(\mathbf{x})$ as calculating the partition function of unnormalized



(a) Data A (b) Data B

Figure 2: *TA vs. FA on 2D synthetic datasets. Training data log-likelihood of each model is in parenthesis. TA better models more complex densities.*

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:

$$p(\mathbf{x}) = \int_{\mathbf{z}} d\mathbf{z} \, \frac{p^*(\mathbf{z}|\mathbf{x})}{q(\mathbf{z})} q(\mathbf{z}) \simeq \frac{1}{M} \sum_{i}^{M} w^{(i)}, \quad \text{where } w^{(i)} = \frac{p^*(\mathbf{z}^{(i)}|\mathbf{x})}{q(\mathbf{z}^{(i)})}, \quad z^{(i)} \sim q(\mathbf{z}) \tag{15}$$

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:

$$p_\beta(\{\mathbf{z}_j\}) \propto q(\{\mathbf{z}_j\})^{1-\beta} p^*(\{\mathbf{z}_j\}|\mathbf{x})^\beta = p(\{\mathbf{z}_j\}) p^\beta(\mathbf{x}|\{\mathbf{z}_j\}), \tag{16}$$

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 Supplementary Materials.

3.4 Equality Constraints

An equality constraint indicates that a set of observed vectors $\{\mathbf{x}^{(k)}\}_{k=1}^K$ have the same factor values for the j-th factor group $\mathbf{z}_j$. During learning, the availability of equlity constraints will only change the inference step. Assuming we have constraints for the factor group $j = 1$, the posterior for $\mathbf{z}_j$ (Eq. 9) will be modified as follows:

$$p(\mathbf{z}_1|\{\mathbf{x}^{(k)}\}, \{\mathbf{z}_2^{(k)}\}, \{\mathbf{z}_3^{(k)}\}) = \mathcal{N}(\mathbf{z}_1|\tilde{\mathbf{V}}^{-1} \sum_{k=1}^{K} \{\mathbf{\Lambda}^{(k)\mathsf{T}} \mathbf{\Psi}^{-1} (\mathbf{x}^{(k)} - \boldsymbol{\mu}^{(k)})\}, \tilde{\mathbf{V}}^{-1}) \tag{17}$$

$$\tilde{\mathbf{V}} = \mathbf{I} + \sum_{k=1}^{K} \mathbf{\Lambda}^{(k)\mathsf{T}} \mathbf{\Psi}^{-1} \mathbf{\Lambda}^{(k)}, \qquad \mathbf{\Lambda}^{(k)} = \mathbf{W}_1 + \mathcal{T} \bar{\times}_3 \mathbf{z}_3^{(k)} \bar{\times}_2 \mathbf{z}_2^{(k)}, \qquad \boldsymbol{\mu}^{(k)} = \mathbf{m} + \mathbf{W}_2 \mathbf{z}_2^{(k)} + \mathbf{W}_3 \mathbf{z}_3^{(k)}$$

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.

3.5 Mixture of Tensor Analyzers

Extending TAs to Mixture of Tensor Analyzers (MTAs) is straightforward. Each component c will have its own parameters $\theta_c = \{\mathbf{W}_c, \mathbf{T}_{(1),c}, \mathbf{\Psi}_c\}$. The data likelihood is marginalized over the C components: $p(\mathbf{x}) = \sum_{c=1}^{C} p(\mathbf{x}|c)p(c)$. Posterior distribution over the factors and components can be decomposed as:

$$p(\{\mathbf{z}_j\}, c|\mathbf{x}) = p(\{\mathbf{z}_j\}|\mathbf{x}, c)p(c|\mathbf{x}) \tag{18}$$

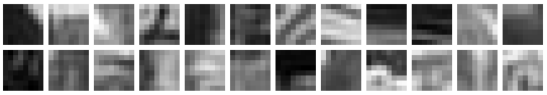
where $p(\{\mathbf{z}_j\}|\mathbf{x}, c)$ can be sampled using Eq. 9 with θ_c and $p(c|\mathbf{x}) \propto p(\mathbf{x}|c)p(c)$. Sec. 3.3 showed how $p(\mathbf{x}|c)$ can be efficiently approximated, thereby making it feasible to train MTAs.

4 Experiments

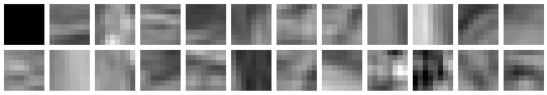
We demonstrate the usefulness of TA on 2 synthetic and 3 real-life datasets. We plan to release source code with implementation details of the experiments in the future.

4.1 Synthetic Data

As a proof of concept, we compared TA to FA on two synthetic datasets (Fig. 2 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 until convergence. The TAs



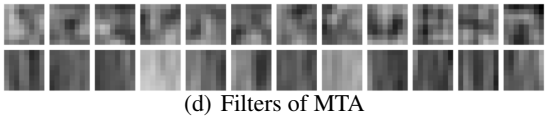
(a) Training Patches



(b) Samples from MTA

ICA	GRBM	MFA	MTA
-38.7±0.8	-44.9±1.2	-25.5±0.3	**-24.7±0.2**

(c) Average test log-likelihood (nats)



(d) Filters of MTA

Figure 3: *Learning MTA on natural image patches. (a) Training data. (b) Samples from MTA look realistic. (c) Average test log-likelihood comparisons. ICA: Independent Component Analysis [1]. GRBM: Gaussian Restricted Boltzmann Machine [7]. (d) Each row contains filters from a different MTA component.*

performed model recovery nicely. The left panel of Fig. 2(a) plots training points. The data log-likelihood of the true model is -2.58 nats. The middle panel plots the samples of a TA which achieved -2.62 nats on the training data. The right panel plots samples drawn from a FA. Likewise in Fig. 2(b), TA is a significantly better model than the FA: -2.04 ± 0.05 to -2.48 ± 0.09 nats. 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.

4.2 Natural Images

Learning a good density model of natural images is useful for image denoising and inpainting. We compared MTAs to MFAs and other models on image patches. 100,000 8×8 patches were extracted from the training set of the Berkeley Natural Images database [12] for training, while 20,000 patches from the test set were extracted for testing. PCA is performed to preprocess the data to 30 dimensions, preserving 98% of the variance. For both MTA and MFA[2], we used 10 mixture components and selected the number of factors per component using a validation set. For MFA, 40 factors per component were used, resulting in an overcomplete representation. For MTA, each component is a TA$\{30,8,2\}$. Convergence is achieved when the objective does not improve by more than 0.01 percent. The average test log-likelihood of MTA is better than that of the MFA by **0.8 nats**[3]. Fig. 3(b) shows that samples of a MTA matches closely to the training patches in Fig. 3(a). Each row of Fig. 3(d) 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.

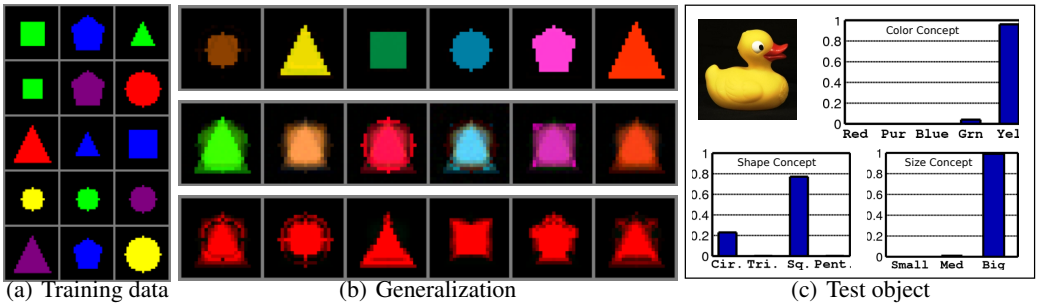
4.3 Concept Learning with equality constraints

Human intelligence is characterized by the ability to form abstract concepts which allows for generalization across widely differing percepts. The color concept of "red" is perceived from both the images of an small red apple and a red car. Concepts can be learned using TAs. We use images of synthetic shapes for training. There are 4 different shapes: circle, triangle, square, and pentagon; 5 different colors: red, green, blue, yellow, and purple; 3 different sizes: small, medium, and big. In total, there are 60 images of resolution 24×24 with 3 color channels per image. We learned a TA$\{3 \times 24^2, 5, 3, 4\}$ using 30 iterations of EM. 200 alternating Gibbs steps were taken to estimate the posterior during each E-step. Equality constraints were used during training, implicitly assigning meanings to the latent factor groups, i.e. $\mathbf{z}_1 \triangleq \mathbf{z}_{shape}$, $\mathbf{z}_2 \triangleq \mathbf{z}_{color}$, and $\mathbf{z}_3 \triangleq \mathbf{z}_{size}$.

After learning, to probe into the representation learned by the TA, we first infer the latent factors $\{\mathbf{z}_{shape}, \mathbf{z}_{color}, \mathbf{z}_{size}\}$ for the training shapes by sampling from $p(\mathbf{z}_{shape}, \mathbf{z}_{color}, \mathbf{z}_{size}|\mathbf{x})$. To see if $\mathbf{z}_{color}$ truly represents the color concept, we fix the inferred factors $\{\mathbf{z}_{shape}, \mathbf{z}_{size}\}$ of a small blue circle, while drawing a new $\mathbf{z}_{color}^{new}$ from its standard Normal prior. The top row of Fig. 4(b) shows the generated images from the factors $\{\mathbf{z}_{shape}, \mathbf{z}_{color}^{new}, \mathbf{z}_{size}\}$. Old shapes with the same size but novel (and more importantly) homogeneous colors are generated. This type of generalization is possible if and only if $\mathbf{z}_{color}$ models color. We further sample from the priors of both the color and shape factors. Generations mix and match novel colors and novel shapes, shown in the middle row of Fig. 4(b). Due to the fact that TA is a multilinear model, it can only linearly combine the 4 training shapes to synthesize new shapes. The bottom row of Fig. 4(b) are filters/detectors for the concept red - tuned to red shapes of differing size and shape. They are the fibers of the tensor of the TA conditioned on $\mathbf{z}_{color} = red$.

[2]MFA code [18] is downloaded from http://lear.inrialpes.fr/∼verbeek/software.php

[3]The gain is found to be statistically significant using the paired t-test at $p = 0.05$.



(a) Training data (b) Generalization (c) Test object

Figure 4: *Learning color, shape, and size concepts. (a) Data contains 4 shapes, 5 colors, and 3 sizes. (b) Each row demonstrates a particular generalization achieved by the TA. (c) Posterior distribution over the concept mixture of Gaussians. the model is able to correctly activate previously learned concepts for a novel test object. See text for details, best viewed in color and on-screen.*

The aggregated posterior $\sum_{n=1}^{N} p(\mathbf{z}_{shape}^{(n)}, \mathbf{z}_{color}^{(n)}, \mathbf{z}_{size}^{(n)} | \mathbf{x}^{(n)})$ is tightly clustered in distinct modes. Since training data has 5 colors, 5 clusters will form in the space of $\mathbf{z}_{color}$. Likewise, there are 3 and 4 clusters for the size and shape factors. We train three second layer mixture of Gaussians (MoG) models using the samples from the TA posterior as training data, e.g. $\mathbf{z}_{color}^{(n)} \sim p(\mathbf{z}_{color} | \mathbf{x}^{(n)})$. Using prior knowledge, we set the number of components of the "color" MoG to be 5, "size" MoG to be 3, and the "shape" MoG to be 4. Learning using EM and initializing the means of MoG components to small values, the components of the MoGs become semantically meaningful, each representing a separate concept. Using this hierarchical model, we demonstrate strong generalization by using a 24×24 image of a toy duck as the observed test image Fig. 4(c). A two stage inference step first infers the factors using the parameters of the TA. We then compute the posterior probabilities over the components of all 3 MoGs, conditioned on the inferred factors from the first stage. Interestingly, our model perceive a big, yellow object which is mostly square but slightly circular.

4.4 Face Recognition with equality constraints

In a one-shot learning setting, classifiers must be trained from only one example per class. For face recognition, only one example per test subject is used for training. We use the Yale B database and its Extended version in this experiment [10]. 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. Equality constraints specifying which images have the same identity or lighting type are used during training.

The learned TA allows for strong generalization to new people under new lighting conditions. It achieved an average log-likelihood of $\mathbf{836 \pm 7}$ nats on the images of the 10 held-out subjects. As a comparison, the best MFA model achieved only 791 ± 10 nats. The number of components and factors of the MFA are optimally 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, exactly as in Sec. 4.3. One factor group's activation is fixed, while we sample the other group of factors from its standard Normal prior. Results are in Fig. 5. A row in panel (b) shows the same person under different lighting. A row in panel (c) shows a different person but under the same lighting. We emphasize that only a *single* test image from *novel* subjects is used for inference (panel (a)).

For the recognition task, we first use the 28 training subjects to learn the parameters for the TA$\{$576,80,4$\}$ in the training phase. During the testing phase, a single image (under frontal lighting) for each of the 10 test subjects is used to compute the TA's posterior mean using 200 alternating Gibbs steps. For each one of the 10 $\mathbf{z}_{identity}$ factors, we use it to collapse the TA into a FA (See Eqs. 7, 8). Each FA has 4 factors and is equivalent to a rank 4 Gaussian in pixel space. During testing, an image is classified to be the same class as the FA which assigns it the highest likelihood.

We compared TA with Nearest Neighbor (NN), normalized Cross Correlation (CC), Support Vector Machines (SVM), S&C bilinear model, Factor Analyzers, and Human. The CC method first subtract the image mean from all pixels of the image. The image is then normalized to have unit norm. The cosine similarity between test and training images is used for classification. Multiclass linear SVM from the LIBLINEAR package [4] is used, where the hyperparameter C was chosen using

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

(d) FA Transfer

NN	CC	SVM	Bilinear	FA	TA	Human
51.1	33.6	35.5	30.8	19.6	**12.9**	9.3

(e) Recognition Error in %

Figure 5: *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 from (a). (d) Factor loading transferring in a FA model creates artifacts. (e) Recognition error on the one-shot face recognition task.*

validation. For the S&C bilinear method, we implemented the exact algorithm as stated in Sec. 3.2 of [15]. During one-shot learning, adapting the style and content codes is performed using the algorithm described in Sec. 6.2 of [15]. Recognition using the FA 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 transfered from the matched FA. In essence, the transfered loadings models 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. Human error is the average of testing several human subjects on the same one-shot recognition task. Recognition errors are listed in Fig. 5(e). TA outperforms the rest of the field by a significant margin and approaches human performance.

FA do not work as well as TA because the transferred factor loadings are not accurate. Fig. 5(d) shows random samples from a transfered FA of a test subject. This is due to the fact that lighting variations of faces must be a *function* of identity, and can not be simply transfered additively.

4.5 Learning with incomplete equality constraints

We demonstrate the advantage of the TA in a semi-supervised setting on the UMIST face database [5]. 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 one-shot face recognition. 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 using equality constraints and 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. 4.4. Fig. 6 plots recognition errors as 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%**.

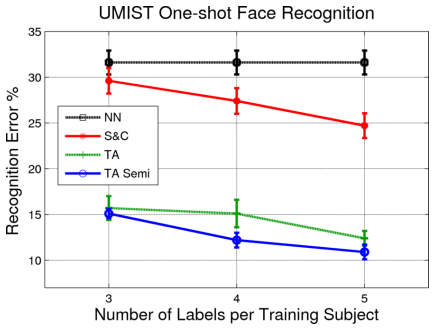


Figure 6: UMIST Recognition errors.

5 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, better model natural image patches, 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

[1] A. Bell and T.J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159, 1995.

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

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

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

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

[6] David B. Grimes and Rajesh P. Rao. Bilinear sparse coding for invariant vision. *Neural computation*, 17(1):47–73, January 2005.

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

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

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

[10] K. C. Lee, Jeffrey Ho, and David Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:684–698, 2005.

[11] Haiping Lu, K. N. Plataniotis, and A. N. Venetsanopoulos. Multilinear principal component analysis of tensor objects for recognition. In *in Proc. Int. Conf. on Pattern Recognition*, pages 776–779, 2006.

[12] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

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

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

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

[16] L. R. Tucker. Implications of factor analysis of three-way matrices for measurement of change. In C. W. Harris, editor, *Problems in measuring change.*, pages 122–137. University of Wisconsin Press, Madison WI, 1963.

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

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

[19] Hongcheng Wang and Narendra Ahuja. Facial expression decomposition. In *ICCV*, pages 958–965, 2003.

[20] Ming-Hsuan Yang, Narendra Ahuja, and David Kriegman. Face detection using a mixture of factor analyzers. In *IEEE International Conference on Image Processing (ICIP)*, Kobe, Japan, 1999.