Robust Generative Subspace Modeling: The Subspace t Distribution

Zia Khan and Frank Dellaert
College of Computing
Georgia Institute of Technology
Atlanta, GA
{zkhan,dellaert}@cc.gatech.edu

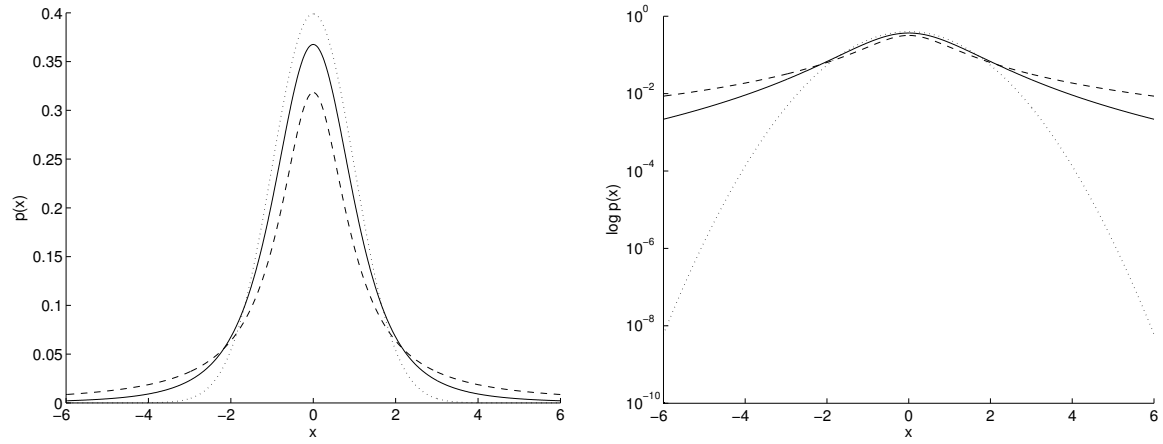
Technical Report number GIT-GVU-04-11
April 1, 2004

Abstract

Linear latent variable models such as statistical factor analysis (SFA) and probabilistic principal component analysis (PPCA) assume that the data are distributed according to a multivariate Gaussian. A drawback of this assumption is that parameter learning in these models is sensitive to outliers in the training data. Approaches that rely on M-estimation have been introduced to render principal component analysis (PCA) more robust to outliers. M-estimation approaches assume the data are distributed according to a density with heavier tails than a Gaussian. Yet, these methods are limited in that they fail to define a probability model for the data. Data cannot be generated from these models, and the normalized probability of new data cannot evaluated. To address these limitations, we describe a generative probability model that accounts for outliers. The model is a linear latent variable model in which the marginal density over the data is a multivariate t, a distribution with heavier tails than a Gaussian. We present a computationally efficient expectation maximization (EM) algorithm for estimating the model parameters, and compare our approach with that of PPCA on both synthetic and real data sets.

1 Introduction

In recent work, principal component analysis (PCA) has been expressed in a probabilistic formulation as a Gaussian latent variable model [18, 19]. The probabilistic formulation offers several advantages. The normalized probability of new data can be evaluated. Maximum likelihood (ML) parameters can be estimated efficiently using an expectation maximization (EM) algorithm. The



(a) (b)

Figure 1: The t distribution has heavier tails than a Gaussian. (a) A t distribution with mean $\mu = 0$, variance $\sigma^2 = 1$, and degrees of freedom $\nu = 3$ is shown with a solid line. When the degrees of freedom equals one $\nu = 1$, we obtain a Cauchy distribution shown by the dashed line. As $\nu \to \infty$ the t distribution approaches a Gaussian shown as a dotted line. (b) The heavy tails of the Cauchy and the t distribution can be seen more clearly on a log scale.

models also permit application of Bayesian methods. The probabilistic PCA (PPCA) model is closely related to statistical factor analysis (SFA) [1, 7]. In the factor analysis model, the noise is no longer constrained to be the same for each dimension of the data vector. Yet, SFA and PPCA are defined as Gaussian models which are known to be sensitive to the presence of outliers in training data [17, 15].

Several approaches have been proposed to render PCA and SFA robust to outliers. The methods rely on robust estimation, particularly M-estimation methods [8]. M-estimation departs from the assumption that the data are normally distributed. Rather, the approach assumes that the data are distributed according to a heavy tailed distribution. Consequently, maximum likelihood solutions need not be least squares solutions and are more robust to outliers. In [20], a robust version of PCA is introduced in the context of computer vision, using a Geman-McClure function as a robust error function. Implicit in this definition is the assumption that the image data is generated from a heavy tailed distribution. In [16], the calculation of the data covariance is replaced with with a minimum covariance determinant (MCD) estimator. MCD's theoretical influence function, which describes the effect of outliers on an estimator, has been shown to be more robust to outliers [15]. Even though M-estimation has a probabilistic interpretation and is effective in practice, it does not yield normalized probabilities or define a generative model for the data.

To address these limitations, we describe a robust generative subspace model in which the marginal density over the data is a multivariate t, a distribution with heavier tails than a Gaus-

sian (see Figure 1). t distributions are commonly used in robust regression [4, 17]. They have also been shown to be effective in modeling constraints found in images [22]. ML estimation of the parameters of a t distribution using EM has been detailed in [12, 13], and ML estimation of mixtures of t distributions is described in [14]. Variational methods have also been applied to mixtures of t distributions [2]. However, these methods do not provide a probabilistic mechanism for dimensionality reduction.

In this paper, we introduce a generative subspace model which we call the Subspace t distribution. The model provides a robust probabilistic mechanism for dimensionality reduction and can be utilized to efficiently model high dimensional data. The model is more general than PPCA and SFA. Both PPCA and SFA can be shown to be limiting cases of the Subspace t distribution, obtained when the the degrees of freedom approach infinity (as in Figure 1). We present an efficient EM algorithm for learning model parameters [6]. Finally, we show the robustness of the Subspace t distribution on a simulated data set with background noise and set of images in which several have been corrupted by noise.

2 The Model

2.1 Linear Latent Variable Models

A linear latent variable model for d-dimensional data vectors t is given by

$$t = \mu + Wx + n$$

where μ is the mean of the data, x is a q -dimensional vector of latent variables, the columns of W contain d-dimensional factors, and n is a vector of additive noise. In SFA the noise n is assumed normally distributed with a diagonal covariance matrix Ψ, which makes the observed variables t_i conditionally independent given the latent variables x [19]. In PPCA, the residual variances Ψ_i are constrained to be equal.

In many applications the data may contain outliers. Normally distributed noise is an inappropriate choice because outliers are typically not normally distributed. A learning algorithm that estimates the model parameters must either eliminate outliers from the data, or the outliers must be modeled explicitly. We take the latter approach because it yields a generative probabilistic model.

2.2 The Subspace t Distribution

To obtain robustness to outliers, the generative subspace model we propose includes an additional random variable called a *scaling* u, as shown in Figure 2. The scaling randomly expands the noise variance and the effect of the factor loadings. Specifically, the model for a d-dimensional data vector t is

$$t = \mu + Wx + n$$

(a) (b)

Figure 2: (a) shows the Bayes net for the SFA and PPCA models. (b) Subspace t distribution introduces a single random variable, the scaling u, to account for outliers. The scaling randomly expands the noise variance and the effect of the factor loadings x.

where μ is a robust mean of data, and where the latent variables x and the noise n are distributed as

$$x|u \sim N\left(x; 0, \frac{I_q}{u}\right) \tag{1}$$

$$n|u \sim N\left(n; 0, \frac{\Sigma}{u}\right) \tag{2}$$

with Σ a diagonal covariance. Additionally, the variance in the Subspace t distribution may be constrained to be identical along each dimension that is $\Sigma = I_d\sigma^2$.

The robustness to outliers arises when we additionally assume that the scaling u is distributed according to a Gamma distribution. Specifically,

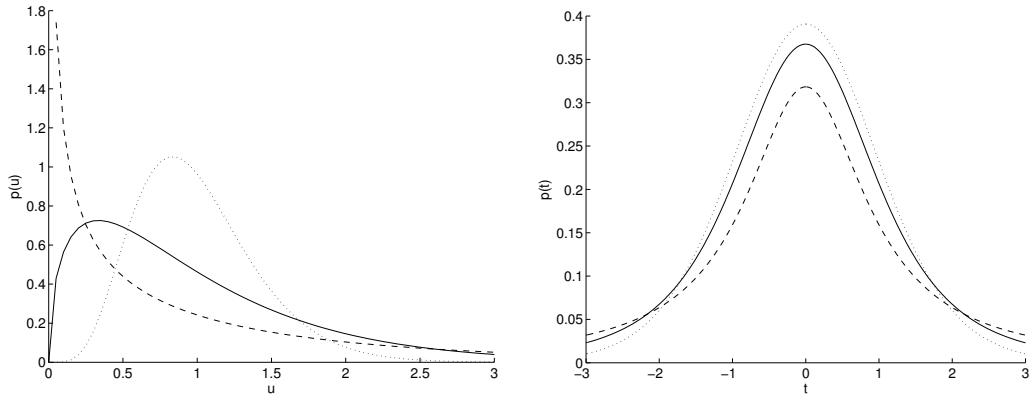
$$u \sim Gamma(\nu/2, \nu/2)$$

$$P(u) = \frac{(\nu/2)^{\nu/2}}{\Gamma(\nu/2)} u^{\nu/2-1} \exp(-\frac{\nu}{2}u) \tag{3}$$

Under that assumption, it can be shown (see Appendix) that the marginal density over the data vectors t is a multivariate t distribution [12, 14, 13]:

$$t \sim \mathcal{T}\left(t; \mu, WW^T + \Sigma, \nu\right) \tag{4}$$

where $\mathcal{T}(\mu, \Sigma, \nu)$ has the density function



(a) (b)

Figure 3: By adjusting the gamma distribution over the scaling u through the degrees of freedom parameter ν, the thickness of the tails of the corresponding Student's t distribution increases or decreases. In (a) we plot several gamma distributions. In (b) we plot the corresponding Student's t distribution. The dashed line shows $\nu = 1$, the solid line shows $\nu = 3$, and the dotted line shows $\nu = 12$.

$$f(t;\mu,\Sigma,\nu) = (\nu\pi)^{-d/2}|\Sigma|^{-1/2}\frac{\Gamma\left(\frac{\nu+d}{2}\right)}{\Gamma(\nu/2)}\left(\frac{\|t-\mu\|_\Sigma^2}{\nu}+1\right)^{-\frac{\nu+d}{2}}$$

where we use the following notation for the squared Mahalanobis distance:

$$\|t-\mu\|_\Sigma^2 \stackrel{\Delta}{=} (t-\mu)^T\Sigma^{-1}(t-\mu)$$

The plot of the Gamma distribution over the scaling u is for several values of the degrees of freedom ν is shown in Figure 3(a). The corresponding t distribution with the same degrees of freedom is shown in Figure 3(b). As the the probability over scalings less than one decreases, the tails of the the t distribution begin to flatten.

Given the scaling u, it is easily seen that the conditional density on t is a normal density with a scaled covariance matrix

$$t|u \sim \mathcal{N}\left(t;\mu,\frac{WW^T+\Sigma}{u}\right)$$

If additionally the values for the latent variables x are known, the conditional density $P(t|x,u)$ is also normal:

$$t|x,u \sim \mathcal{N}\left(t;\mu+Wx,\frac{\Sigma}{u}\right)$$

Sampling in this hierarchical model can be done most efficiently by first sampling from u, then from $x|u$ and finally from $t|x,u$.

2.3 Inference

As with PPCA, it is of interest to infer the joint posterior density $P(x,u|t)$ over the latent variables x and u, given an observed data vector t. This will also be used below to learn the model parameters through EM.

It is easily seen from (1) and (2) that, given u, the posterior density $P(x|u,t)$ on x is normal

$$x|t,u \sim \mathcal{N}\left(s, \frac{R}{u}\right)$$

where

$$s = RW^T\Sigma^{-1}(t-\mu) \tag{5}$$

is the projected data, and

$$R = (W^T\Sigma^{-1}W + I_q)^{-1} \tag{6}$$

is the unscaled covariance of the factor loadings.

It can be shown that, given the model (4), the marginal posterior density $P(u|t)$ over the scaling [12] is given by

$$u|t \sim Gamma\left(\frac{\nu+d}{2}, \frac{\nu+m}{2}\right) \tag{7}$$

where we define m to be the squared Mahalanobis distance for the data point t:

$$m \stackrel{\Delta}{=} \|t-\mu\|^2_{WW^T+\Sigma} \tag{8}$$

Note that m can be computed efficiently by applying an inversion lemma

$$y^T(WW^T+\Sigma)^{-1}y = y^T\Sigma^{-1}y - y^T\Sigma^{-1}WRW^T\Sigma^{-1}y$$

where we define $y \stackrel{\Delta}{=} t-\mu$.

In summary, the joint posterior density $P(x,u|t)$ over the latent variables x and u is given by

$$P(x,u|t) = P(x|u,t)P(u|t) = \mathcal{N}\left(x; s, \frac{R}{u}\right) Gamma\left(u; \frac{\nu+d}{2}, \frac{\nu+m}{2}\right) \tag{9}$$

3 Learning

We can learn the model parameters $\theta = \{\mu, W, \Sigma, \nu\}$ from data using the expectation-maximization (EM) algorithm. The complete log-likelihood $\mathcal{L}_\theta$ is defined as follows

$$\mathcal{L}_\theta = \log \prod_{n=1}^{N} P(x_n, t_n, u_n|\theta) = \log \prod_{n=1}^{N} P(t_n|x_n, u_n, \mu, W, \Sigma) P(x_n|u_n) P(u_n|\nu)$$

After some manipulation and omitting any terms that do not depend on θ we obtain

$$\begin{aligned} \mathcal{L}_\theta &= -\frac{1}{2}\sum_{n=1}^{N} -\frac{N}{2}\log|\Sigma| - \frac{1}{2}\sum_{n=1}^{N} u_n \|t_n - (\mu + W x_n)\|_\Sigma^2 \\ &\quad + \frac{N\nu}{2}\log\frac{\nu}{2} - N\log\Gamma\left(\frac{\nu}{2}\right) + \left(\frac{\nu}{2} - 1\right)\sum_{n=1}^{N}\log u_n - \frac{\nu}{2}\sum_{n=1}^{N} u_n \end{aligned}$$

3.1 Expectation Maximization

In the E-step we compute the expected log-likelihood

$$Q(\theta|\theta^i) \stackrel{\Delta}{=} E[\mathcal{L}_\theta|\{t_n\}, \theta^i]$$

with respect to the joint distribution $\prod_{n=1}^{N} p(u_n, x_n|t_n, \theta^i)$ over the latent variables. In the M-step, we re-estimate all parameters by maximizing $Q(\theta)$:

$$\theta^{i+1} = \operatorname*{argmax}_\theta \; Q(\theta|\theta^i)$$

This is explained in detail below for each of the parameters in turn.

Mean μ

Setting the derivative of $Q(\theta|\theta^i)$ with respect to μ to zero

$$\frac{\partial Q(\theta|\theta^i)}{\partial \mu} = \sum_{n=1}^{N} \left\langle u_n \Sigma^{-1}(t_n - \mu - W x_n) \right\rangle = 0$$

we readily obtain

$$\mu^{i+1} = \frac{\sum_{n=1}^{N} \langle u_n \rangle t_n - W^i \sum_{n=1}^{N} \langle u_n x_n \rangle}{\sum_{n=1}^{N} \langle u_n \rangle}$$

The expectations $\langle u_n \rangle$ and $\langle u_n x_n \rangle$ are computed in the E-step using the parameters θ^i. As shown in Section 2.3, the scaling parameter $u_n|t_n$ follows the Gamma distribution (7), and its mean is

$$\langle u_n \rangle \stackrel{\Delta}{=} \frac{\nu^i + d}{\nu^i + m_n^i}$$

where m_n^i is defined as in (8). To compute $\langle u_n x_n \rangle$ we make use of (9) and obtain

$$\begin{aligned} \langle u_n x_n \rangle &\stackrel{\Delta}{=} \mathrm{E}\left[u_n x_n | t_n, \theta^i\right] \\ &= \mathrm{E}\left[u_n \mathrm{E}\left[x_n | u_n, t_n, \theta^i\right] | t_n, \theta^i\right] \\ &= \mathrm{E}\left[u_n s_n^i | t_n, \theta^i\right] \\ &= \langle u_n \rangle s_n^i \end{aligned} \tag{10}$$

where s_n^i is defined as in (5), and the last equality follows because s_n^i does not depend on u_n.

Factor Loading Matrix W

In order to re-estimate W, it is convenient to write the Mahalanobis distance as follows:

$$\|t_n - (\mu + W x_n)\|_\Sigma^2 = \mathrm{tr}\left(\Sigma^{-1} y_n y_n^T\right) - 2 x_n^T W^T \Sigma^{-1} y_n + \mathrm{tr}\left(W^T \Sigma^{-1} W x_n x_n^T\right)$$

where $y_n = t_n - \mu$. If we then compute and set the derivative equal to zero, we have

$$\begin{aligned} \frac{\partial Q(\theta|\theta^i)}{\partial W} &= \sum_{n=1}^N \left[\frac{d}{dW}\left\langle u_n x_n^T W^T \Sigma^{-1} y_n - \frac{1}{2} u_n \mathrm{tr}\left(W^T \Sigma^{-1} W x_n x_n^T\right) + C \right\rangle\right] \\ 0 &= \sum_{n=1}^N \left[\Sigma^{-1} y_n \langle u_n x_n \rangle^T - \Sigma^{-1} W \left\langle u_n x_n x_n^T \right\rangle\right] \end{aligned}$$

where we made use of the trace derivative [5]:

$$\frac{\partial tr(X^T A X B)}{\partial X} = 2AXB$$

We solve for W and obtain an update that closely resembles the one used in [19, 7] respectively for PPCA and factor analysis:

$$W^{i+1} = \left(\sum_{n=1}^N y_n \langle u_n x_n \rangle^T\right)\left(\sum_{n=1}^N \left\langle u_n x_n x_n^T \right\rangle\right)^{-1}$$

Here $\langle u_n x_n \rangle$ is given by (10), whereas in order to calculate $\left\langle u_n x_n x_n^T \right\rangle$ we first compute

$$\begin{aligned} \mathrm{E}\left[x_n x_n^T | u_n, t_n, \theta^i\right] &= \mathrm{COV}\left[x_n | u_n, t_n, \theta^i\right] + \mathrm{E}\left[x_n | u_n, t_n, \theta^i\right] \mathrm{E}\left[x_n | u_n, t_n, \theta^i\right]^T \\ &= \frac{R^i}{u_n} + \langle u_n x_n \rangle \langle u_n x_n \rangle^T \end{aligned}$$

and then

$$\begin{aligned} \mathrm{E}\left[u_n x_n x_n^T | t_n, \theta^i\right] &= \mathrm{E}\left[u_n \mathrm{E}\left[x_n x_n^T | u_n, t_n, \theta^i\right] | t_n, \theta^i\right] \\ &= \mathrm{E}\left[u_n | t_n, \theta^i\right] s_n s_n^T + \mathrm{E}\left[u_n \frac{R}{u_n} | t_n, \theta^i\right] \\ &= \langle u_n x_n \rangle \left(s_n^i\right)^T + R^i \end{aligned}$$

where R^i is computed according to (6) and s_n^i is computed as in (5) using the current parameters θ^i.

Covariance Matrix Σ

We use a similar approach for Σ and collect all of the terms containing this variable

$$\begin{aligned} \frac{\partial Q(\theta|\theta^i)}{\partial \Sigma} &= \sum_{n=1}^{N} \left[\frac{d}{d\Sigma} \left\langle -\frac{1}{2}\log|\Sigma| - \frac{1}{2} u_n \mathrm{tr}\left(\Sigma^{-1} y_n y_n^T\right) + u_n x_n^T W^T \Sigma^{-1} y_n - \frac{1}{2}\mathrm{tr}\left(W^T \Sigma^{-1} W u_n x_n x_n^T\right) \right\rangle \right] \\ 0 &= \sum_{n=1}^{N} \left[-\Sigma^{-1} + \langle u_n \rangle \Sigma^{-1} y_n y_n^T \Sigma^{-1} - 2\Sigma^{-1} W \langle u_n x_n \rangle y_n^T \Sigma^{-1} + \Sigma^{-1} W \left\langle u_n x_n x_n^T \right\rangle W^T \Sigma^{-1} \right] \end{aligned}$$

where we made use of the trace derivatives [5]:

$$\frac{\partial tr(X^{-1}A)}{\partial X} = -X^{-1}A^T X^{-1}$$

$$\frac{\partial tr(A^T X^{-1} B)}{\partial X} = -X^{-T} A B^T X^{-T}$$

Consequently, we can update the diagonal covariance according to

$$\Sigma^{i+1} = \frac{1}{N} \sum_{n=1}^{N} \mathrm{diag}\left[\langle u_n \rangle y_n y_n^T - 2W^i \langle u_n x_n \rangle y_n^T + W^i \left\langle u_n x_n x_n^T \right\rangle \left(W^i\right)^T \right]$$

The diag indicates that the computation of the outer products need only be performed along the the diagonal. When the noise variance is assumed to be identical along each dimension, we average all of the variance terms computed along the diagonal

$$\left(\sigma^2\right)^{i+1} = \frac{1}{d}\mathrm{tr}\left[\Sigma^{i+1}\right]$$

Degrees of Freedom ν

The update for the degrees of freedom ν follows

$$\nu^{i+1} = \underset{\nu}{\operatorname{argmax}} \left[\frac{N\nu}{2} \log \frac{\nu}{2} - N \log \Gamma\left(\frac{\nu}{2}\right) + \left(\frac{\nu}{2} - 1\right) \sum_{i=1}^{N} \langle \log u_n \rangle - \frac{\nu}{2} \sum_{i=1}^{N} \langle u_n \rangle \right]$$

In this we need (see [12, 14])

$$\langle \log u_n \rangle \triangleq \mathrm{E}\left[\log u_n | t_n, \theta^i\right] = \Psi\left(\frac{\nu^i + d}{2}\right) - \log\left(\frac{1}{2}\left(\nu^i + m_n^i\right)\right)$$

where $\Psi(x)$ is the Digamma function. In the M-step, we find ν^{i+1} using 1-d non-linear maximization.

3.2 Summary

Below we describe the principal steps of the EM learning algorithm:

1. **E-step:**
   Using the current set of parameters $\theta^i = \{\mu^i, W^i, \Sigma^i, \nu^i\}$, compute the following expected values for all data points $n = 1 \ldots N$.

   (a) $\langle \log u_n \rangle = \Psi\left(\frac{\nu^i + d}{2}\right) - \log\left(\frac{1}{2}\left(\nu^i + m_n^i\right)\right)$

   (b) $\langle u_n \rangle = \frac{\nu^i + d}{\nu^i + m_n^i}$

   (c) $\langle u_n x_n \rangle = \langle u_n \rangle s_n^i$

   (d) $\left\langle u_n x_n x_n^T \right\rangle = \langle u_n x_n \rangle \left(s_n^i\right)^T + R^i$

2. **M-step:**
   Re-estimate the parameters as

   (a) $\mu^{i+1} = \frac{\sum_{i=1}^{N} \langle u_n \rangle t_n - W^i \langle u_n x_n \rangle}{\sum_{i=1}^{N} \langle u_n \rangle}$

   (b) $W^{i+1} = \left(\sum_{n=1}^{N} y_n \langle u_n x_n \rangle^T\right) \left(\sum_{i=1}^{N} \left\langle u_n x_n x_n^T \right\rangle\right)^{-1}$

   (c) $\Sigma^{i+1} = \frac{1}{N} \sum_{n=1}^{N} \mathrm{diag}\left[\langle u_n \rangle y_n y_n^T - 2W^i \langle u_n x_n \rangle y_n^T + W^i \left\langle u_n x_n x_n^T \right\rangle \left(W^i\right)^T\right]$
   If the noise variance is identical for each dimension, update it according to:
   $\left(\sigma^2\right)^{i+1} = \frac{1}{d} \mathrm{tr}\left[\Sigma^{i+1}\right]$

(d) Find ν^{i+1} such that

$$\nu^{i+1} = \max_{\nu} \left[\frac{N\nu}{2} \log \frac{\nu}{2} - N \log \Gamma\left(\frac{\nu}{2}\right) + \left(\frac{\nu}{2} - 1\right) \sum_{i=1}^{N} \langle \log u_n \rangle - \frac{\nu}{2} \sum_{i=1}^{N} \langle u_n \rangle \right]$$

using 1-d non-linear maximization.

3. Repeat until convergence criteria are met.

4. Resolve any rotational ambiguity in the estimated W by computing the singular value decomposition (SVD) of $WW^T = RSV^T$ and rotating W according to RW.

The EM algorithm has the following intuitive interpretation: in the E-step we fix the subspace spanned by columns of W and compute the moments of hidden factor loadings x by projecting all of the data t into subspace and weighting according the the robustness parameter ν and the distance the current guess of the mean μ. In the M-step the we fix the distributions over the hidden factor loadings and update the subspace to minimize robust reconstruction error of the data points. The robustness parameter ν is adjusted to account for the distances of the data points from the mean.

4 Examples

In experimental studies it is often difficult to assure that data is free of background noise due to irregularities in the measurement process. Consequently, we considered a simulated 2-d data set in which we sampled 100 data points from from a Gaussian with mean $\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ and covariance $\Sigma = \begin{bmatrix} 10 & 7 \\ 7 & 3 \end{bmatrix}$ with background noise generated by sampling 30 data points from uniform distributions over the range of -30 to +30 along each dimension. It is clear from Figure 4 that PPCA attempts to model the background noise and incorrectly selects the first principal component. In this comparison, it is possible to improve the performance of PPCA by modeling the data as a mixture between PPCA and a uniform model. However, this approach makes a strong assumption about the background noise and cannot be expected to work as well as the Subspace t distribution in situations where the noise is not uniform.

Similarly, in real world applications of computer vision training data may include artifacts due to occlusions, illumination, noise, and errors underlying collection of the data [20]. Problems in erroneous data collection are prevalent in tracking methodologies where a model of a target's appearance is updated in an EM algorithm [10, 21]. Temporary failures in tracking may introduce images that do not include the tracked target into the training data set. To simulate this effect we compiled an image set containing several corrupted images (see Figure 5(a)). As shown in Figure 5(b) and 5(c) the effect the corrupted images on the eigen-images of the learned Subspace

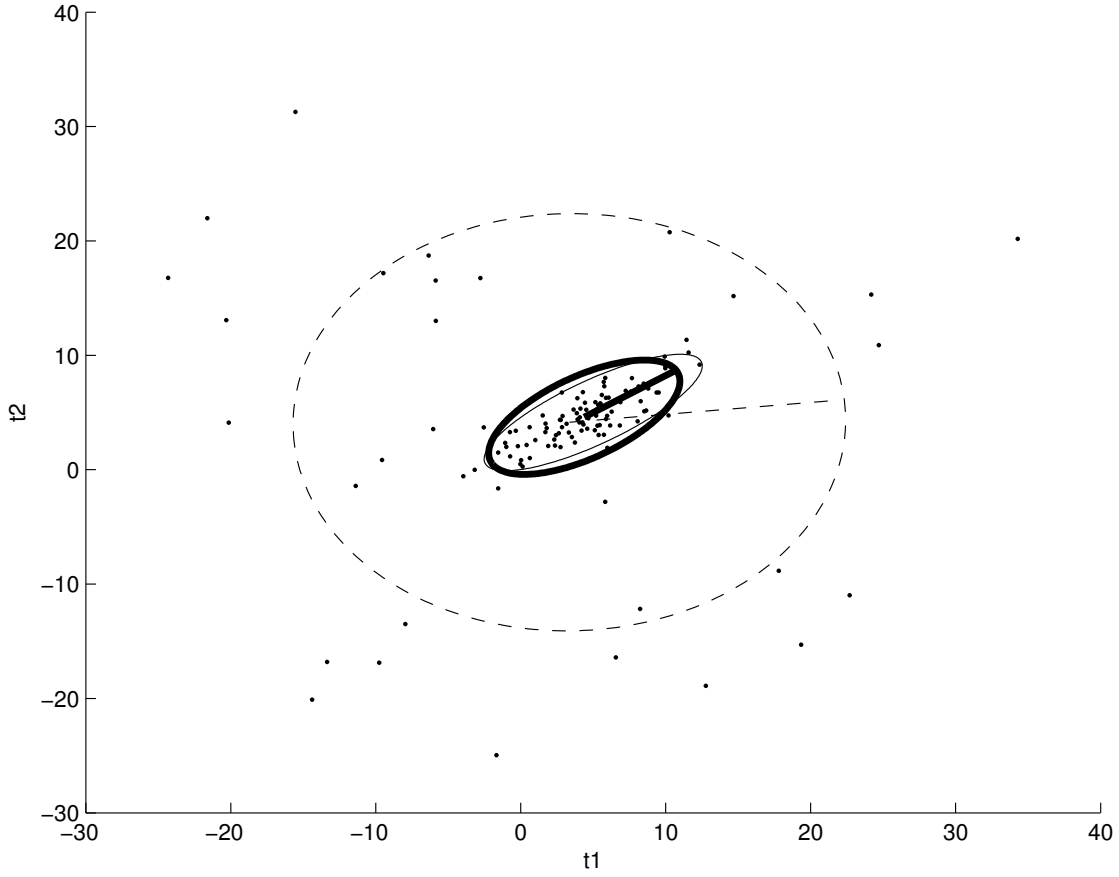


Figure 4: The PPCA model is sensitive to outliers. A data set was generated from a uniform distribution and a 2 component Gaussian for which the 2.296 standard deviation contour is shown as a solid oval. The dashed oval shows the 2.296 standard deviation contour of the Gaussian estimated by a 1-component PPCA model and the dashed line shows the incorrectly estimated first principal component. The thick oval shows the 2.296 standard deviation contour of the student *t* estimated by a 1 component Subspace *t* distribution with identical noise along each dimension and the thick line shows the correctly estimated first principal component. The model correctly down weights the outliers producing a better estimate of the covariance and first principal component.

(a)

(b)

(c)

Figure 5: (a) We compiled a training image set containing 100 images of the ant *Leptothorax albipennis*. Ten corrupted images were also added to the training data set. (b) shows the mean image (left) and the four first principal components estimated by the Subspace *t* distribution with identical variance for each dimension from the training image set. (c) shows the mean image and the first four principal components estimated by PPCA. The Subspace *t* distribution estimates a more robust mean and set of principal components. The estimated components obtained for PPCA incorrectly model the corrupted images.

t distribution is substantially less than that observed in the PPCA learned model. In an adaptive tracking framework, the Subspace *t* distribution might substantially improve tracking in approaches that use subspace representations in modeling target shape or appearance [3, 11, 9].

5 Conclusions

The Subspace *t* distribution offers a robust alternative to PPCA and SFA. It retains the advantages of the SFA and PPCA models. We list some of the important advantages of the Subspace *t* distribution below:

- The model yields a normalized probability. An experimenter can correctly ask whether or not a new data point came from a Subspace *t* distribution or another probability model.
- Bayesian methods can be applied in such a model. Priors on parameters can be included in learning. The Subspace *t* distribution may be included in Bayesian model selection methods. Variational methods may also be applied to inference in such models.
- The Subspace *t* distribution is robust. Strong assumptions about the distribution of the noise are not necessary.
- A single Subspace *t* distribution can be used in a mixture model.
- SFA and PPCA are limiting cases of the Subspace *t* distribution obtained when the the degrees of freedom approach infinity $\nu \to \infty$.

Acknowledgments

We would like to thank Stephan Pratt at the Princeton University Department of Ecology and Evolutionary Biology for the *Leptothorax albipenni*s images. This work was funded under NSF Award IIS-0219850.

References

[1] D. J. Bartholomew. *Latent Variable Models and Factor Analysis*. Oxford University Press, New York, 1987.

[2] C. M. Bishop and M. Svensen. Robust bayesian mixture modelling. In *Proceedings of the European Symposium on Artificial Neural Networks*, 2004.

[3] M.J. Black and A.D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. In *Eur. Conf. on Computer Vision (ECCV)*, 1996.

[4] T. Briegel and V. Tresp. Robust neural network regression for offline and online learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2000.

[5] M. Brookes. *Matrix Reference Manual*. http://www.ee.ic.ac.uk/hp/staff/dmb/matrix/, Accessed March 2004.

[6] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[7] B. J. Frey. Factor analysis using batch and online EM. Technical Report TR-99-2, Internal UW/CS Adaptive Computation, 1999.

[8] P. Huber. *Robust Statistics*. John Wiley & Sons, New York, NY, 1981.

[9] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *Eur. Conf. on Computer Vision (ECCV)*, pages 343–356, 1996.

[10] Allan D. Jepson, David J. Fleet, and Thomas F. El-Maraghi. Robust online appearance models for visual tracking. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 415–422, 2001.

[11] Z. Khan, T. Balch, and F. Dellaert. A Rao-Blackwellized particle filter for EigenTracking. In press, CVPR 2004, 2004.

[12] K. L. Lange, R. J. A. Little, and J. M. G. Taylor. Robust statistical modeling using the *t* distribution. *Journal of the American Statistical Association*, 84(408):881–896, 1989.

[13] C. Liu. ML estimation of the multivariate *t* distribution and the EM algorithm. *Journal of Multivariate Analysis*, 63:296–312, 1997.

[14] D. Peel and G. J. McLachlan. Robust mixture modelling using the *t* distribution. *Statistics and Computing*, 10:339–348, 2000.

[15] G. Pison., P. J. Rousseeuw, P. Filzmoser, and C. Croux. Robust factor analysis. *Journal of Multivariate Analysis*, 84(1):145–172, 2003.

[16] P. J. Rousseeuw. Least median of squares regression. *Journal of the American Statistical Association*, 79:871–880, 1984.

[17] P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection*. John Wiley & Sons, New York, NY, 1981.

[18] S. Roweis. EM algorithms for PCA and SPCA. In Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10. The MIT Press, 1998.

[19] M.E. Tipping and C.M. Bishop. Probabilistic principal component analysis. Technical Report NCRG/97/010, Neural Computing Research Group, Aston University, September, 1997.

[20] F. Torre and M. J. Black. Robust principal component analysis for computer vision. In *Intl. Conf. on Computer Vision (ICCV)*, pages 362–369, 2001.

[21] J. Vermaak, P. Perez, M. Gangnet, and A. Blake. Towards improved observation models for visual tracking: Selective adaptation. In *Eur. Conf. on Computer Vision (ECCV)*, pages 645–660, 2002.

[22] M. Welling and G. E. Hinton S. Osindero. Learning sparse topographic representations with products of student-t distributions. In *Advances in Neural Information Processing Systems (NIPS)*, 2003.

Appendix

The Student's t distribution in (4) can be derived by noting this useful definition of a gamma function

$$\int_0^\infty e^{-(\alpha t)^\beta} t^\gamma dt = \frac{\Gamma\left(\frac{\gamma+1}{\beta}\right)}{\beta\alpha^{\gamma+1}}$$

We assume that the data t given the the covariance scaling u is distributed according to a Gaussian

$$t|u \sim \mathcal{N}\left(t; \mu, \frac{\Sigma}{u}\right)$$

The scaling has the effect of "broadening" the covariance when $u < 1$. Hence, it models possible outliers. To derive the t distribution we assume that u is distributed according to a

$$u \sim Gamma\left(\frac{\nu}{2}, \frac{\nu}{2}\right)$$

where $Gamma(\alpha, \beta)$ has the density function

$$f(u; \alpha, \beta) = \frac{\beta^\alpha u^{\alpha-1}}{\Gamma(\alpha)} \exp(-\beta u)$$

for $u > 0$ and $\alpha, \beta > 0$.

The t distribution can be thought of an infinite mixture of Gaussian distributions centered at μ where $p(u)$ is the weight of a mixture component. We can obtain a student t by marginalizing over

all values of u (with m defined as before in Equation 8)

$$\begin{aligned}
p(t) &= \int_0^\infty p(t|u)p(u)du \\
&= k\int_0^\infty u^{d/2}\exp\left[-\frac{u}{2}m\right]\times u^{\frac{\nu}{2}-1}\exp\left(-\frac{\nu}{2}u\right)du \\
&= k\int_0^\infty u^{\frac{\nu+d}{2}-1}\exp\left[-u\left(\frac{1}{2}m+\frac{\nu}{2}\right)\right]du \\
&= k\Gamma\left(\frac{\nu+d}{2}\right)\left(\frac{1}{2}m+\frac{\nu}{2}\right)^{-\frac{\nu+d}{2}} \\
&= k\Gamma\left(\frac{\nu+d}{2}\right)\frac{\nu}{2}^{-\frac{\nu+d}{2}}\left(\frac{m}{\nu}+1\right)^{-\frac{\nu+d}{2}} \\
&= k'\left(\frac{m}{\nu}+1\right)^{-\frac{\nu+d}{2}}
\end{aligned}$$

which has the form of a t distribution with ν degrees of freedom, with

$$k' = (\nu\pi)^{-d/2}|\Sigma|^{-1/2}\frac{\Gamma\left(\frac{\nu+d}{2}\right)}{\Gamma(\nu/2)}$$