A Novel Parameter Estimation Algorithm for the Multivariate t-Distribution and Its Application to Computer Vision

Chad Aeschliman, Johnny Park, and Avinash C. Kak

Purdue University
http://rvl.ecn.purdue.edu

Abstract. We present a novel algorithm for approximating the parameters of a multivariate t-distribution. At the expense of a slightly decreased accuracy in the estimates, the proposed algorithm is significantly faster and easier to implement compared to the maximum likelihood estimates computed using the expectation-maximization algorithm. The formulation of the proposed algorithm also provides theoretical guidance for solving problems that are intractable with the maximum likelihood equations. In particular, we show how the proposed algorithm can be modified to give an incremental solution for fast online parameter estimation. Finally, we validate the effectiveness of the proposed algorithm by using the approximated t-distribution as a drop in replacement for the conventional Gaussian distribution in two computer vision applications: object recognition and tracking. In both cases the t-distribution gives better performance with no increase in computation.

1 Introduction

Probability models are used in a wide range of applications in order to account for the uncertainty of processes and observations in a principled way. Often the true distribution underlying a process or observation is unknown or is difficult to use. In these cases one option is to use a nonparametric distribution. However, nonparametric distributions require a large amount of data to train, particularly in high-dimensional spaces. A common alternative is to fit a generic parametric probability model to the data.

By far the most commonly used parametric probability model is the multivariate Gaussian distribution. The Gaussian distribution is easy to use and has a number of nice properties. Parameter estimation for the Gaussian distribution is straightforward since its sufficient statistics are the parameters. Also, it is very easy to compute the marginal and conditional distributions from the joint distribution. However, for many applications the Gaussian distribution has tails which are too light; it tends to underestimate the probability of rare events occurring, which is unrealistic and can have a profound negative impact on performance. [10, 14, 7]. For example, in a tracking application a target may undergo a sudden change in illumination or may be partially occluded by another target. If these rare events are ignored the tracking algorithm will fail.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 594–607, 2010.

Several alternatives to the Gaussian distribution have been proposed in order to avoid this issue. One such alternative is the multivariate t-distribution [7]. The t-distribution has a similar shape as the Gaussian distribution but with much heavier tails. Because of the heavy tails, the t-distribution is a better model for situations in which rare events commonly occur. The t-distribution is particularly better suited for high-dimensional spaces where all events are expected to be rare. The heavy tails of the t-distribution also increase the robustness in parameter estimation, since the outliers in the data naturally have little overall impact on the parameters [5]. This is in stark contrast to the Gaussian for which a few outliers can dramatically change the parameter estimates of the distribution.

Despite these attractive properties of the t-distribution, it has not been widely used. We believe this can be attributed to the lack of good estimation techniques (in an engineering sense) for the parameters of the distribution. Numerous EM-based iterative algorithms have been developed to compute the maximum likelihood estimates for the parameters of the t-distribution [8, 9, 10]. However, because of their iterative nature, these algorithms are computationally expensive. Also, these algorithms work on the dataset as a whole and cannot be incrementally updated as new data becomes available. This deficiency severely limits their usefulness in real time applications.

This paper addresses the problem of parameter estimation for the multivariate t-distribution. We propose a new approximate algorithm which is both computationally efficient and incrementally updateable. The proposed algorithm provides comparable estimation accuracy compared to the EM-based algorithms while achieving a significant improvement in the computation time. Using the approximation formula, we then develop an approximate incremental probabilistic PCA (PPCA) for the t-distribution. Previous work has extended the idea of PPCA to the t-distribution [17], but with a focus on extending the EM-based maximum likelihood techniques. As we mentioned, these EM-based iterative estimators are computationally expensive and cannot be updated incrementally, posing severe limitations on the range of applications. We present an approximate incremental approach which has equivalent computational requirements as the incremental PPCA approaches for the Gaussian distribution [16, 11].

2 Multivariate t-Distribution

In this section, we will present some useful properties of the t-distribution, many of which come from the seminal work by Kotz and Nadarajah [6].

2.1 Basic Properties

The pdf of the p-variate t-distribution with ν degrees of freedom is given by

$$f(\mathbf{x}) = \frac{\Gamma((p+\nu)/2)}{\Gamma(\nu/2)(\pi\nu)^{p/2}|\mathrm{S}|^{1/2}} \left[1 + \frac{1}{\nu}(\mathbf{x}-\mathbf{c})^T \mathrm{S}^{-1}(\mathbf{x}-\mathbf{c})\right]^{-(p+\nu)/2} \tag{1}$$

where $\mathbf{c} \in \mathbb{R}^p$ is the location parameter and $\mathrm{S} \in \mathbb{R}^{p\times p}$ is the positive definite scale matrix. Notationally we will write $\mathbf{x} \sim t(\mathbf{c}, \mathrm{S}, \nu)$. The vector $\mathbf{c}$ specifies the location of the single mode of the distribution. The matrix S specifies the relative width of the central mode along each dimension and also the correlation between dimensions. The degrees of freedom ν controls the heaviness of the tails of the distribution. When $\nu = 1$ we have the Cauchy distribution which has very heavy tails while $\nu = \infty$ gives the Gaussian distribution.

Many applications require the computation of the marginal distribution of one or more random variables for which the joint distribution is known. This is easily done with the multivariate t-distribution by simply partitioning the parameters $\mathbf{c}$ and S, i.e. if $\mathbf{x} \sim t(\mathbf{c}, \mathrm{S}, \nu)$ and we define

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \tag{2}$$

$$\mathbf{c} = \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix} \tag{3}$$

$$\mathrm{S} = \begin{bmatrix} \mathrm{S}_{11} & \mathrm{S}_{12} \\ \mathrm{S}_{21} & \mathrm{S}_{22} \end{bmatrix} \tag{4}$$

then $\mathbf{x}_1 \sim t(\mathbf{c}_1, \mathrm{S}_{11}, \nu)$ and $\mathbf{x}_2 \sim t(\mathbf{c}_2, \mathrm{S}_{22}, \nu)$. The conditional distribution $f(\mathbf{x}_2|\mathbf{x}_1)$ is unfortunately not a t-distribution and does not have a particularly clean form. However, the expectation of $\mathbf{x}_2$ given $\mathbf{x}_1$ does have a nice form

$$E\{\mathbf{x}_2|\mathbf{x}_1\} = \mathrm{S}_{21}\mathrm{S}_{11}^{-1}(\mathbf{x}_1 - \mathbf{c}_1) + \mathbf{c}_2 \tag{5}$$

2.2 Sampling from the Multivariate t-Distribution

Generating samples from a multivariate t-distribution is fairly straightforward. If $\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathrm{I})$ and $\gamma \sim \chi^2(\nu)$ then the random vector

$$\mathbf{x} = \sqrt{\frac{\nu}{\gamma}}\mathrm{T}^T\mathbf{y} + \mathbf{c} \tag{6}$$

is distributed as $\mathbf{x} \sim t(\mathbf{c}, \mathrm{T}^T\mathrm{T}, \nu)$. Note that every entry in the random vector $\mathbf{x}$ is scaled according to the same value γ. Because of this, even if the scale matrix is diagonal the entries in $\mathbf{x}$ will not be independent. This is an important limitation of the multivariate t-distribution.

3 Batch Parameter Estimation

3.1 Maximum Likelihood Estimator

The maximum likelihood estimates for the parameters of the t-distribution based on sample data $\mathrm{X} = [\mathbf{x}_1\, \mathbf{x}_2\, \ldots\, \mathbf{x}_n]$ must satisfy the following equations [10]

$$\mathbf{c} = \frac{\sum_{i=1}^{n} w_i \mathbf{x}_i}{\sum_{i=1}^{n} w_i} \tag{7}$$

$$\mathrm{S} = \frac{1}{n}\sum_{i=1}^{n} w_i \left(\mathbf{x}_i - \mathbf{c}\right)\left(\mathbf{x}_i - \mathbf{c}\right)^T \tag{8}$$

where

$$w_i = (p + \nu) \left(\nu + (\mathbf{x}_i - \mathbf{c})^T \mathrm{S}^{-1} (\mathbf{x}_i - \mathbf{c})\right)^{-1} \tag{9}$$

These equations cannot be solved to give closed form estimates for the parameters. An EM-based approach can be used to iteratively estimate $\mathbf{c}$, S, and ν which satisfy these constraints [8, 9, 12]. While some variations of the implementation may achieve a faster parameter estimation than others, fundamentally they are all iterative algorithms, thus computationally expensive. More importantly, none of these methods can be extended to efficiently update the estimates as new data becomes available. All of the algorithms are based on computing weighted means and covariances. Since the weight for each sample is a function of $\mathbf{c}$, S, and ν, the weights on old data will change as new data becomes available and hence the old data must be included in the computation.

3.2 Approximate Algorithm

Special Case. To develop an approximate algorithm for computing the parameters we begin by considering the special case $\mathbf{x} \sim t(\mathbf{0}, \alpha\mathrm{I}, \nu)$ for some constant $\alpha > 0$. In this special case the pdf of the norm of $\mathbf{x}$ is given by

$$f(\|\mathbf{x}\|) = \frac{2\|\mathbf{x}\|^{p-1}}{\mathrm{B}(\nu/2, p/2)(\alpha\nu)^{p/2}} \left(1 + \frac{1}{\alpha\nu}\|\mathbf{x}\|^2\right)^{-(\nu+p)/2} \tag{10}$$

where $\mathrm{B}(x, y) = \Gamma(x)\Gamma(y)\Gamma^{-1}(x+y)$ is the beta function. The goal is to estimate ν and α given sample data $\mathrm{X} = [\mathbf{x}_1\, \mathbf{x}_2\, \ldots\, \mathbf{x}_n]$. This can be done by considering the following results

$$E\{\log \|\mathbf{x}\|^2\} = \log \alpha + \log \nu + \psi_0\left(\frac{p}{2}\right) - \psi_0\left(\frac{\nu}{2}\right) \tag{11}$$

$$Var\{\log \|\mathbf{x}\|^2\} = \psi_1\left(\frac{\nu}{2}\right) + \psi_1\left(\frac{p}{2}\right) \tag{12}$$

where $\psi_0(x)$ is the digamma function and $\psi_1(x)$ is the trigamma function.

Let $z_i = \log \|\mathbf{x}_i\|^2 = \log \mathbf{x}_i^T \mathbf{x}_i$. To estimate ν we need to solve for $\hat{\nu}$ which satisfies

$$\psi_1\left(\frac{\hat{\nu}}{2}\right) = \frac{1}{n}\sum_{i=1}^{n}(z_i - \bar{z})^2 - \psi_1\left(\frac{p}{2}\right) \tag{13}$$

where $\bar{z} = \frac{1}{n}\sum_{i=1}^{n} z_i$. Unfortunately we cannot directly solve Eq. (13). However, by using the approximation

$$\psi_1(x) \approx \frac{x+1}{x^2} \tag{14}$$

we can compute the estimate

$$\hat{\nu} = \frac{1 + \sqrt{1 + 4b}}{b} \tag{15}$$

with

$$b = \frac{1}{n}\sum_{i=1}^{n}(z_i - \bar{z})^2 - \psi_1\left(\frac{p}{2}\right) \tag{16}$$

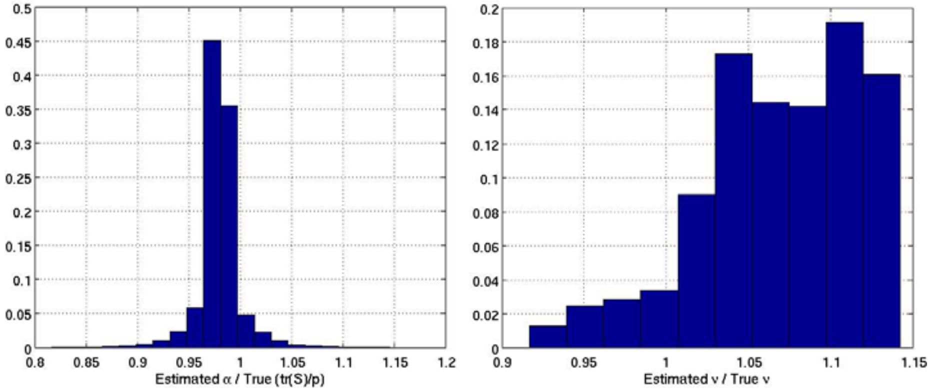

Fig. 1. An experimental evaluation of Eqs. 15 and 17 when applied to samples from general t-distributions. Each figure is a normalized histogram over 10000 trials. For each trial we set $\nu = 10^u$ where $u \sim U(-1,1)$ is a uniform random variable. The scale matrix for each trial was a random positive definite matrix drawn from a Wishart distribution and p was set to 50. The left figure compares $\hat{\nu}$ computed using Eq. 15 to the true value ν. The right figure compares $\hat{\alpha}$ computed using Eq. 17 to the mean of the diagonal entries of the scale matrix.

Finally, we use Eq. (11) to compute an estimate for the scaling

$$\hat{\alpha} = \exp\left\{\bar{z} - \log\hat{\nu} + \psi_0\left(\frac{\hat{\nu}}{2}\right) - \psi_0\left(\frac{p}{2}\right)\right\}. \tag{17}$$

General Case. We now consider the general case when $\mathbf{x} \sim t(\mathbf{c}, \mathrm{S}, \nu)$. The location vector $\mathbf{c}$ can be estimated by considering each dimension of the data separately and computing either the sample median or the mean of the center 25% of the data [13]. We will use $\hat{\mathbf{c}}$ to denote the estimate of the location vector.

Since our goal is a computationally efficient approximation rather than an exact solution to the parameters we begin by estimating ν and α using the equations of the preceding section, i. e. we assume for the purpose of approximation that $\mathrm{S} = \alpha\mathrm{I}$ for some α. This can be done by first computing $z_i = \log\|\mathbf{x}_i - \hat{\mathbf{c}}\|^2$ and then directly applying Eqs. (15) and (17). In practice, the estimate $\hat{\nu}$ is a good approximation to ν regardless of the structure of S as is shown in Fig. 1. The slight positive bias may be due to the error in the approximation for the trigamma function given in Eq. 14. The scaling estimate $\hat{\alpha}$ also provides a good estimate for the mean of the diagonal entries of S, as illustrated by the results shown in Fig. 1. Hence all that remains is to estimate the relative scaling of the elements of S.

To estimate the relative scaling of the elements of S we use the auxiliary matrix

$$\bar{\mathrm{S}} = \frac{1}{n}\sum_{i=1}^{n}\frac{(\mathbf{x}_i - \hat{\mathbf{c}})(\mathbf{x}_i - \hat{\mathbf{c}})^T}{\|\mathbf{x}_i - \hat{\mathbf{c}}\|^{\beta}}. \tag{18}$$

which is similar to the sample covariance except that each sample is first scaled by the norm raised to a constant power β. We have experimentally validated that a good choice for β can be given by

$$\beta = \frac{2\log_2 p}{\hat{\nu}^2 + \log_2 p} \tag{19}$$

Note that for many applications p is large and $\hat{\nu}$ is small so we can directly use $\beta = 2$. The scaling term in the denominator of Eq. 18 is necessary in order to give a good approximation when ν is small. We can now apply the estimated mean of the diagonal entries $\hat{\alpha}$ to obtain an estimate for S

$$\hat{\mathrm{S}} = \frac{\hat{\alpha} p}{\mathrm{tr}\left(\bar{\mathrm{S}}\right)} \bar{\mathrm{S}} \tag{20}$$

This completes the development of the approximation algorithm which is given in succinct form in Fig. 2.

$\hat{\mathbf{c}}$ = median of each dimension of the data

$$z_i = \log \|\mathbf{x}_i - \hat{\mathbf{c}}\|^2 \quad \bar{z} = \frac{1}{n}\sum_{i=1}^{n} z_i$$

$$b = \frac{1}{n}\sum_{i=1}^{n} (z_i - \bar{z})^2 - \psi_1\left(\frac{p}{2}\right)$$

$$\hat{\nu} = \frac{1 + \sqrt{1 + 4b}}{b}$$

$$\hat{\alpha} = \exp\left\{\bar{z} - \log \hat{\nu} + \psi_0\left(\frac{\hat{\nu}}{2}\right) - \psi_0\left(\frac{p}{2}\right)\right\}$$

$$\beta = \frac{2\log_2 p}{\hat{\nu}^2 + \log_2 p}$$

$$\bar{\mathrm{S}} = \frac{1}{n}\sum_{i=1}^{n} \frac{(\mathbf{x}_i - \hat{\mathbf{c}})(\mathbf{x}_i - \hat{\mathbf{c}})^T}{\|\mathbf{x}_i - \hat{\mathbf{c}}\|^{\beta}}.$$

$$\hat{\mathrm{S}} = \frac{\hat{\alpha} p}{\mathrm{tr}\left(\bar{\mathrm{S}}\right)} \bar{\mathrm{S}}$$

Fig. 2. Batch Approximation Algorithm

3.3 Comparative Evaluation of Maximum Likelihood and Approximation Algorithms

To evaluate the accuracy of the approximation algorithm we performed several experiments on synthetic data. In three experiments we varied separately the

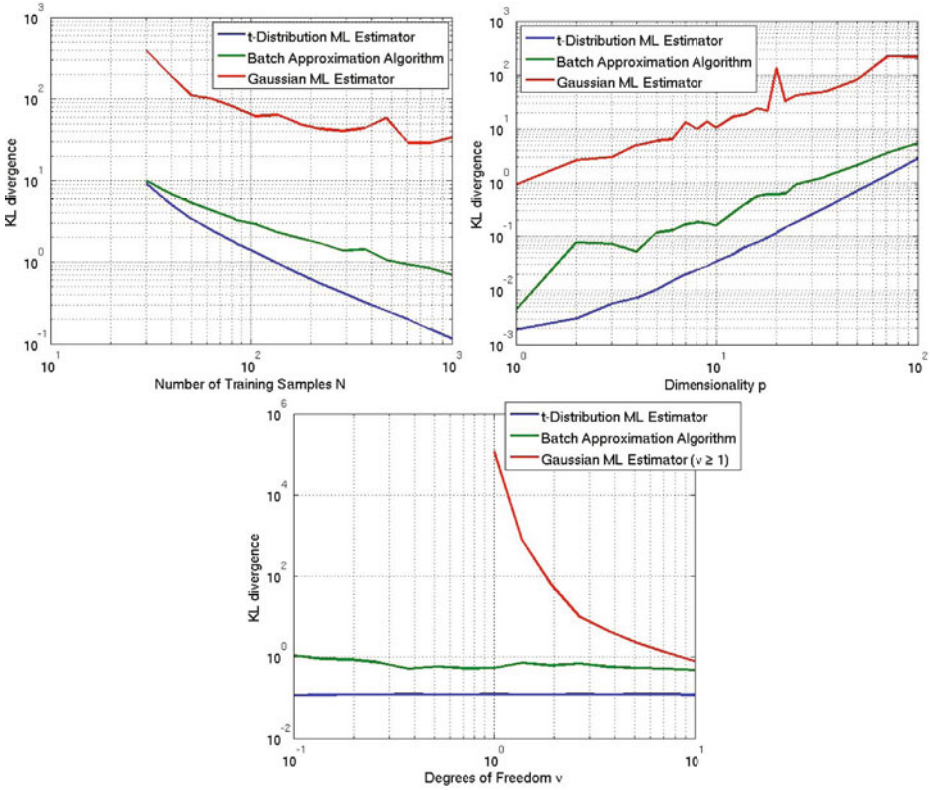

Fig. 3. Comparison of the accuracy of the maximum likelihood and approximation algorithms for estimating the parameters of a multivariate t-distribution. The accuracy of the maximum likelihood Gaussian distribution is provided for comparison.

dimensionality p, the degree of freedom ν, and the number of training samples N. In each case we generated synthetic data using the sampling technique described in section 2.2 and then computed the KL divergence from the true distribution for both the maximum likelihood parameter estimates (computed using the method in [9]) and the approximate parameter estimates. We also computed the KL divergence from the true distribution for the maximum likelihood Gaussian distribution in order to give a basis for comparison. The results are shown in Fig. 3. As expected, the KL divergence of the approximation algorithm is higher than that of the maximum likelihood algorithm. However, the approximation algorithm is nearly as good across a broad range of parameter settings and in particular it is significantly better than the maximum likelihood Gaussian in every case.

The maximum likelihood estimator has slightly better accuracy but in many other ways the approximate algorithm is superior. The primary advantages of the approximate algorithm are

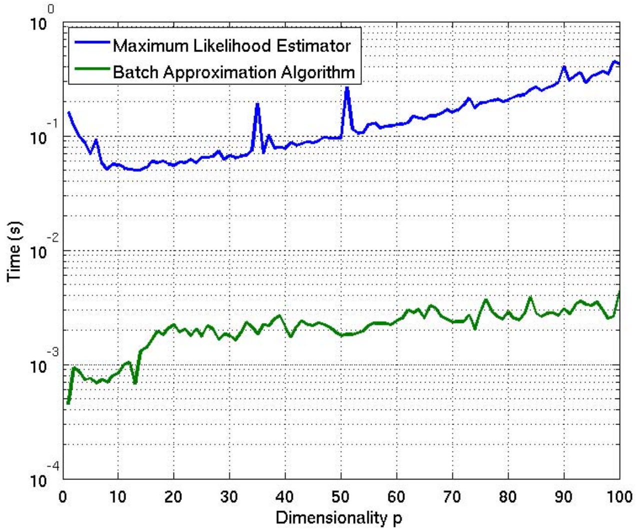

Fig. 4. Training time on 200 samples as a function of p for the maximum likelihood and approximate algorithms

- **Computational Efficiency:** Fig. 4 shows the running time for both methods as a function of the dimensionality p of the data based on a MATLAB implementation. The approximate algorithm is consistently 50-100 times faster.
- **Easy Implementation:** Because the approximate algorithm is directly computed there is no need for iterative looping in the code. This also eliminates the need to check for convergence.
- **Useful Theoretical Tool:** We can use the approximate parameter estimation equations as a basis for developing additional algorithms which would not be possible with the maximum likelihood estimator, e.g. incremental algorithms.

4 Incremental Parameter Estimation

Many real-time applications require online updating of the parameters of the distribution. To handle this situation we present two incremental approaches which can be used with the t-distribution based on the batch approximation algorithm of the preceding section. The first approach is essentially a direct extension of the batch algorithm. The second approach uses PPCA to estimate the parameters under the assumption that the underlying dimensionality of the model is much lower than the true dimensionality. Note that for both algorithms, we can incrementally estimate $\hat{\mathbf{c}}$ without needing to store previously seen data by using an online quantile estimator [15, 2].

4.1 Direct Incremental Algorithm

In order to convert the batch algorithm to an incremental algorithm we need to rewrite Eqs. (15), (17), and (18) to be incremental. To compute $\hat{\nu}$ and $\hat{\alpha}$ we need to incrementally update estimates for the mean and variance of $z = \log \|\mathbf{x} - \hat{\mathbf{c}}\|^2$. After the kth sample, the mean $\bar{z}$ and variance v_z are updated by

$$\bar{z}^{(k)} = \frac{k-1}{k}\bar{z}^{(k-1)} + \frac{1}{k}z_k \tag{21}$$

$$v_z^{(k)} = \frac{k-1}{k}v_z^{(k-1)} + \frac{k-1}{k^2}\left(z_k - \bar{z}^{(k-1)}\right)^2 \tag{22}$$

where $z_k = \log \|\mathbf{x}_k - \hat{\mathbf{c}}^{(k)}\|^2$, i.e. we use the best available estimate for $\mathbf{c}$ for each incremental update. Because the estimate for $\mathbf{c}$ changes with each sample these incremental update formulas will not give exactly the same results as the batch algorithm. In practice this is typically not a problem. However, when k is very small we must be careful to ensure that $\|\mathbf{x}_k - \hat{\mathbf{c}}^{(k)}\|^2 \neq 0$. One way to do this is to store the first few samples and use these to compute a batch estimate before switching to the incremental estimator.

We can now directly use Eq. (15) to conclude that the estimate for ν after the kth sample is given by

$$\hat{\nu}^{(k)} = \frac{1 + \sqrt{1 + 4b^{(k)}}}{b^{(k)}} \tag{23}$$

where

$$b^{(k)} = v_z^{(k)} - \psi_1\left(\frac{p}{2}\right) \tag{24}$$

Similarly, the estimate for α is given by

$$\hat{\alpha}^{(k)} = \exp\left\{\bar{z}^{(k)} - \log \hat{\nu}^{(k)} + \psi_0\left(\frac{\hat{\nu}^{(k)}}{2}\right) - \psi_0\left(\frac{p}{2}\right)\right\}. \tag{25}$$

The last step is to compute an estimate for $\bar{\mathrm{S}}$. Under the assumption that p is large and ν is small (and hence $\beta = 2$ in Eq. (19)) we use the estimate

$$\bar{\mathrm{S}}^{(k)} = \frac{k-1}{k}\bar{\mathrm{S}}^{(k-1)} + \frac{1}{k}\left[\frac{(\mathbf{x}_k - \hat{\mathbf{c}}^{(k)})(\mathbf{x}_k - \hat{\mathbf{c}}^{(k)})^T}{\|\mathbf{x}_k - \hat{\mathbf{c}}^{(k)}\|^2}\right] \tag{26}$$

where again we use the best available estimate for $\mathbf{c}$ for each update. Again we must be careful to ensure that $\|\mathbf{x}_k - \hat{\mathbf{c}}^{(k)}\|^2 \neq 0$. This is most likely to occur when k is very small and as a solution, as already stated, we use the first few samples to compute a batch estimate of $\bar{\mathrm{S}}$ before switching to the incremental algorithm.

4.2 PPCA for t-Distribution

Although PPCA was originally developed in the context of the multivariate Gaussian distribution the idea has been extended to the t-distribution [16, 17]. The idea behind PPCA is to model the scale matrix in the following way

$$\mathrm{S} = s\mathrm{I} + \mathrm{W}\mathrm{W}^T \tag{27}$$

where $s > 0$ captures the general level of uncertainty in the random variable while $\mathrm{W} \in \mathbb{R}^{p\times q}$, $q < p$, captures the correlation between dimensions. Since we typically have $q \ll p$, this model for S can be trained with significantly fewer data samples while still providing a powerful model.

The maximum likelihood estimates for W and s can be obtained through an iterative EM-based approach [17]. Once again, this approach is too slow for practical use in many computer vision problems. As an alternative, we present an incremental algorithm based on the approximate incremental estimator of the preceding section. The key is to note that the incremental equation for $\hat{\alpha}$ given in the preceding section is still applicable and so instead of directly modeling S as in Eq. 27 we can instead model $\bar{\mathrm{S}}$. Specifically, the goal is to find estimates for $\hat{s}$ and $\hat{\mathrm{W}}$ such that

$$\bar{\mathrm{S}}^{(k)} \approx \hat{s}^{(k)}\mathrm{I} + \hat{\mathrm{W}}^{(k)}\left(\hat{\mathrm{W}}^{(k)}\right)^T \tag{28}$$

Since $\bar{\mathrm{S}}$ is in essence a weighted covariance matrix the incremental update formulas for PPCA with the multivariate Gaussian distribution can be used as a template for how to estimate $\hat{s}$ and $\hat{\mathrm{W}}$ [11]. The idea is to use

$$\hat{\mathrm{W}}^{(k)} = \mathrm{V}^{(k)}(\Lambda^{(k)} - \hat{s}^{(k)}\mathrm{I})^{1/2} \tag{29}$$

where $\Lambda^{(k)}$ is a diagonal matrix of the q largest eigenvalues of $\bar{\mathrm{S}}^{(k)}$ and the columns of $\mathrm{V}^{(k)} \in \mathbb{R}^{p\times q}$ are the corresponding eigenvectors.

The first step is to rewrite the incremental update equation for $\bar{\mathrm{S}}$ as

$$\bar{\mathrm{S}}^{(k)} = \frac{k-1}{k}\left(\bar{\mathrm{S}}^{(k-1)} + \mathbf{y}\mathbf{y}^T\right) \tag{30}$$

where

$$\mathbf{y} = \frac{1}{\sqrt{k-1}}\frac{\mathbf{x}_k - \hat{\mathbf{c}}^{(k)}}{\|\mathbf{x}_k - \hat{\mathbf{c}}^{(k)}\|} \tag{31}$$

Let $\mathrm{L} = \left[\hat{\mathrm{W}}^{(k-1)}\,\mathbf{y}\right]$ and let $\mathrm{Q} = \mathrm{L}^T\mathrm{L}$. Compute an eigen decomposition of $\mathrm{Q} \in \mathbb{R}^{q\times q}$ s.t.

$$\mathrm{Q} = \mathrm{U}\Gamma\mathrm{U}^T \tag{32}$$

where $\Gamma = \mathrm{diag}(\gamma_1, \ldots, \gamma_{q+1})$. Then the first $q+1$ eigenvalues of $\bar{\mathrm{S}}^{(k)}$ are given by

$$\lambda_i = \frac{n}{n+1}[\hat{s}^{(k-1)} + \gamma_i] \tag{33}$$

and the corresponding eigenvectors are given by the columns of

$$\hat{\mathrm{V}} = \mathrm{L}\mathrm{U}\Gamma^{-1/2} \tag{34}$$

Note that we keep only the first q eigenvalues and eigenvectors in order to compute $\hat{\mathrm{W}}^{(k)}$. Finally, we update $\hat{s}$

$$\hat{s}^{(k)} = \frac{n}{n+1}\left[\frac{\gamma_{q+1}}{p-q} + \hat{s}^{(k-1)}\right] \tag{35}$$

5 Application to Computer Vision

5.1 Classification

A common task in computer vision is to determine which object from a set of possible choices is visible in a small subsection of the image. One way to solve this problem is to first train a probability model for each possible choice based on training data. The best estimate for which object is visible in a small subsection of the image is then given by the probability model which assigns the highest probability to the subsection. This method of classification is known as the generative approach.

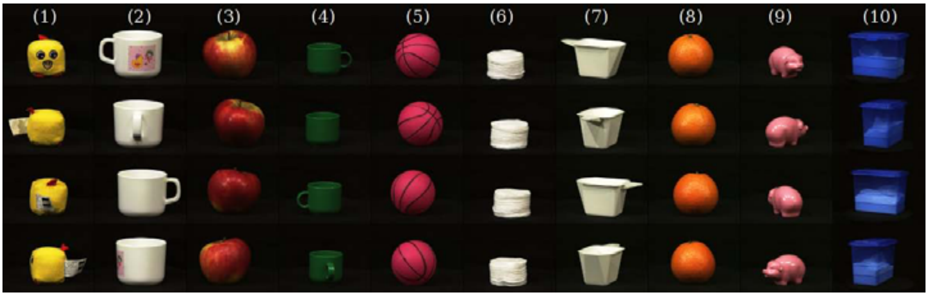

Fig. 5. Objects from the Amsterdam Library of Object Images (ALOI)[3]

In order to compare the power of the Gaussian and t-distributions for solving the classification problem, we analyzed ten objects (shown in Fig. 5) from the Amsterdam Library of Object Images [3]. For each object, there are 72 images taken in 5° increments around the object. We randomly split these images into 36 training images and 36 testing images for each object. For each image, we then extracted the brightness of the pixels from 100 non-overlapping 10×10 squares and used these as the data samples. The data samples from the training images were used to obtain the maximum likelihood Gaussian distribution and the approximate t-distribution using the proposed batch algorithm. The probability models that had been learned for all of the objects were then used to classify the samples from the testing images.

Under these conditions, the Gaussian distribution led to a classification accuracy of 51% while using the t-distribution significantly improved the accuracy to 68%. The reason for this can be seen by considering Table 1 which gives individual results for each object. The Gaussian distribution gives very poor results for objects 2, 8, and 9; each of which has substantial changes in brightness due to the design, specular highlights, and shadows. These changes represent outliers and are poorly handled by the Gaussian model, resulting in a very broad distribution with poor discrimination. Objects 4 and 6 on the other hand, which give good results with a Gaussian distribution, are mostly uniform in brightness and do not undergo significant changes from frame to frame.

Table 1. Object classification rates in %. Each entry gives the percentage of samples that were correctly classified for that object.

	1	2	3	4	5	6	7	8	9	10
t-distribution	74	66	66	90	61	97	47	62	43	71
Gaussian	74	25	44	91	43	80	63	24	8	55

The parameter estimation algorithm for the t-distribution automatically includes robustness against outliers and so large changes in brightness have little effect on the overall parameter estimation. The result is a tighter distribution compared to the Gaussian. Because of this the t-distribution more effectively models each object and hence gives better discrimination. Note that the algorithm also performs very well when no outliers are present, giving excellent results for objects 4 and 6. It is this flexibility to handle a wide range of data types which makes the t-distribution an ideal choice for many applications.

5.2 Tracking

Tracking is another very important application in computer vision. The goal in tracking is to identify which pixels in each frame of a video sequence were generated by one or more targets. This can be done by training a probability distribution over the brightness of the pixels making up each target. The joint

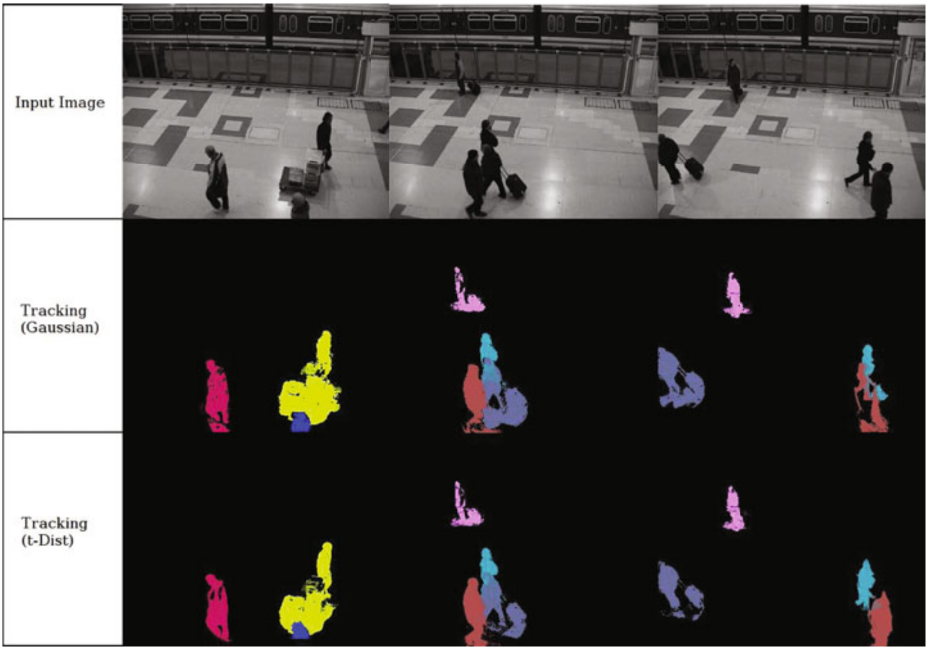

Fig. 6. Tracking results using the Gaussian distribution and the t-distribution

distribution is used to identify where a target is located in a given frame. The marginal distributions can then be used to determine for each pixel if it was generated by the target or something else, effectively segmenting out the target from its surroundings.

Using a tracking algorithm based on PPCA for the Gaussian distribution as a basis we modified the algorithm to use the t-distribution instead [1]. Both algorithms were tested on a video sequence from PETS2006 [4]. The results for three frames of the video sequence are shown in Fig. 6. The complete video sequence is included with the supplementary material. Although the overall results are similar regardless of which distribution is used, the t-distribution does show improved performance. The t-distribution is much less susceptible to shadows which can be seen by looking at the gray target in the second and third frames. The t-distribution also handles overlapping targets more cleanly. Because of this it is able to properly distinguish between the orange and cyan targets in the final frame while the Gaussian distribution confuses them.

6 Conclusions

The Gaussian distribution is by far the most commonly used parametric probability model mainly because it is simple to use and computationally tractable even for high dimensional data. The light tails of the Gaussian distribution, however, make it a poor model for the randomness present in many sources of data. We believe the t-distribution represents a viable replacement for the Gaussian. By developing an approximate algorithm to compute the parameters, we have shown that the t-distribution can be made as computationally efficient as the Gaussian. Furthermore, we show that the proposed algorithm can be updated online for real time applications. Even though the parameter estimation is only approximate, the results show that the t-distribution outperforms the Gaussian for two important applications in computer vision. We expect future research along these lines to touch a large spectrum of domains in computer vision.

Acknowledgment

The authors want to thank Sierra Nevada Corporation for their support.

References

[1] Aeschliman, C., Park, J., Kak, A.C.: A Probabilistic Framework for Joint Segmentation and Tracking. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2010)

[2] Chen, F., Lambert, D., Pinheiro, J.C.: Incremental quantile estimation for massive tracking. In: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 516–522. ACM, New York (2000)

[3] Geusebroek, J.M., Burghouts, G.J., Smeulders, A.W.M.: The Amsterdam library of object images. International Journal of Computer Vision 61(1), 103–112 (2005)

[4] Iscaps, C.: Pets2006 (2006), `http://www.cvg.rdg.ac.uk/pets2006/data.html`

[5] Khan, Z., Balch, T., Dellaert, F.: MCMC-based particle filtering for tracking a variable number of interacting targets. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1805–1918 (2005)

[6] Kotz, S., Nadarajah, S.: Multivariate t distributions and their applications. Cambridge Univ. Pr., Cambridge (2004)

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

[8] Liu, C., Rubin, D.B.: ML estimation of the t distribution using EM and its extensions, ECM and ECME. Statistica Sinica 5(1), 19–39 (1995)

[9] Meng, X.L., van Dyk, D.: The EM algorithm–an old folk-song sung to a fast new tune. Journal of the Royal Statistical Society. Series B (Methodological), 511–567 (1997)

[10] Nadarajah, S., Kotz, S.: Estimation Methods for the Multivariate t Distribution. Acta Applicandae Mathematicae: An International Survey Journal on Applying Mathematics and Mathematical Applications 102(1), 99–118 (2008)

[11] Nguyen, H.T., Ji, Q., Smeulders, A.W.M.: Spatio-temporal context for robust multitarget tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(1), 52 (2007)

[12] Peel, D., McLachlan, G.: Robust mixture modelling using the t distribution. Statistics and Computing 10(4), 339–348 (2000)

[13] Rothenberg, T.J., Fisher, F.M., Tilanus, C.B.: A note on estimation from a Cauchy sample. Journal of the American Statistical Association 59(306), 460–463 (1964)

[14] Simoncelli, E.P.: Statistical modeling of photographic images. In: Handbook of Image and Video Processing, pp. 431–441 (2005)

[15] Tierney, L.: A space-efficient recursive procedure for estimating a quantile of an unknown distribution. SIAM Journal on Scientific and Statistical Computing 4, 706 (1983)

[16] Tipping, M., Bishop, C.M.: Probabilistic principal component analysis. Journal of the Royal Statistical Society. Series B (Statistical Methodology) 61(3), 611–622 (1999)

[17] Zhao, J., Jiang, Q.: Probabilistic PCA for t distributions. Neurocomputing 69(16-18), 2217–2226 (2006)