

Unsupervised Generalised Gaussian
Mixture Model Classification
Using the EM Algorithm

A.M. Johansen and W. J. Fitzgerald
CUED/F-INFENG/TR 455 (2003)

Abstract

A generalised Gaussian mixture model unsupervised classifier has been developed. The EM algorithm is used to obtain an estimate of the maximum likelihood classification of data under this model. Results for simulated Cauchy distributed, Student-t distributed and Laplacian data indicate that this model gives significantly better performance than a Gaussian mixture model for non-Gaussian data; simulated data from Gaussian distributions is also well classified.

The algorithm has been applied to real data, including some previously unpublished laser scattering data, with positive results. The results presented here appear to suggest that care must be taken when assuming that data follows a Gaussian distribution. It is suggested that a generalised Gaussian mixture model is suitable for model-based classification of a wide range of real data.

This algorithm can also be used to obtain an estimate of the kurtosis of a data sample and hence how “close to Gaussian” that data is.

Contents

1	Introduction	1
1.1	Notation	2
2	Theoretical Background	3
2.1	The Update Equations	3
2.2	Updating B	4
2.3	Relationship between B and ρ_4	4
3	Classifier Implementation	6
3.1	Classifier Model	6
3.2	Initialisation	7
3.3	Iteration	8
3.4	Convergence & Speed	8
3.5	Outliers & Uncertain Assignments	8
4	Simulations	9
4.1	Gaussian and Laplacian Data	9
4.2	Positive Kurtosis Distributions	11
4.3	Comparison with Autoclass	12
5	Application to Real Data	14
5.1	Iris Data	14
5.2	Laser Scattering Data	15
6	Conclusions	17
A	Derivation of the Update Equations	19
A.1	Updating μ (approximately)	20
A.2	Updating σ (exactly)	21
B	The K-means Algorithm	22
	References	23

Chapter 1

Introduction

Model based clustering techniques are often based upon the assumption that clusters may be modelled as a Gaussian distribution of data points around a centroid. This is often well justified either experimentally, or on theoretical grounds – such as the central limit theorem. However, when applied inappropriately, this assumption is actually very strong, particularly when one is considering the tails of the distribution in which there is a quadratically exponential suppression of data probabilities. Consequently, outliers can have undue influence on the clustering procedure or otherwise be badly classified under a Gaussian noise assumption.

It has recently been shown [1] that much of the oligonucleotide microarray data which has been processed thus far is likely to follow a distribution which is closer to Cauchy than Gaussian. Additional [2] section 6.6 suggests that a Student-t distribution better fits the noise of a DNA microarray than a Gaussian distribution, and that substantially better results can be obtained if (an approximation to) such a noise model is adopted. Heavier than Gaussian tails have been observed [3] with the data of Hughes et al [4].

In this report, a classifier which makes use of the EM (Expectation Maximisation) method [5] to obtain an ML (Maximum Likelihood) classification assuming *generalised Gaussian* noise has been developed (see [6] page 157 for further details about these distributions). This was motivated both by the desire to investigate whether the assumption of Gaussianity causes significant problems when used to classify non-Gaussian data and by interest in the model for its own sake.

Univariate generalised Gaussian distributions have the form:

$$P(y|\mu, \sigma, B) = \frac{\omega(B)}{|\sigma|} \exp\left(-C(B) \left|\frac{y - \mu}{\sigma}\right|^{\frac{2}{1+B}}\right) \quad (1.1)$$

$$C(B) \triangleq \left(\frac{\Gamma(\frac{3}{2}(1+B))}{\Gamma(\frac{1}{2}(1+B))}\right)^{\frac{1}{1+B}} \quad (1.2)$$

$$\omega(B) \triangleq \frac{\Gamma(\frac{3}{2}(1+B))^{1/2}}{(1+B)\Gamma(\frac{1}{2}(1+B))^{3/2}} \quad (1.3)$$

Where y is the scalar feature and μ and σ correspond to the mean and gener-

alised deviation. This can be generalised to a diagonal multivariate case as:

$$P(y|\mu, \sigma, B) = \frac{\omega^d(B)}{|\sigma|} \exp\left(-C(B) \sum_{i=1}^d \left| \frac{(y^i - \mu^i)}{\sigma^{ii}} \right|^{\frac{2}{1+B}}\right) \quad (1.4)$$

Where d corresponds to the dimensionality of the feature vectors, y ; μ and σ have become a vector and a *diagonal* matrix, respectively and superscripts indicate specific elements of these vectors and matrices.

The value B provides a measure of the *kurtosis*¹ of the distribution.

For positive B , generalised Gaussian distributions have heavier tails than simple Gaussian distributions and so might offer a better description of the processes by which many data arise.

1.1 Notation

Inevitably, when it is necessary to refer to a mixture of models which have matrix and vector parameters, the notation becomes slightly opaque. The notation which has been used throughout this report is as follows: subscripts refer to the mixture element, superscripts to particular elements of a vector or matrix and bracketed superscripts to the iteration of the algorithm (where appropriate). As an example $(\mu_i^j)^{(n)}$ means the j -th element of the mean of mixture element (class) i during the n -th iteration of the algorithm.

¹Kurtosis is the difference between the fourth moment of a distribution and three times the square of the second moment; for a Gaussian distribution the kurtosis is zero. For our purposes, kurtosis has been defined in terms of the standardised variable $(x - \mu)/\sigma$ to provide scale invariance: $\rho_4 \triangleq \mathbb{E} \left[\left(\frac{y - \mu}{\sigma} \right)^4 \right] - 3 \mathbb{E} \left[\left(\frac{y - \mu}{\sigma} \right)^2 \right]^2$

Chapter 2

Theoretical Background

A generalised Gaussian mixture model is based upon the assumption that all data was generated by one of a set of k generalised Gaussians. We have assumed that B is the same for all of the generalised Gaussians which comprise such a mixture for simplicity, although this condition can easily be relaxed; the other parameters are assumed to be different for each cluster (i.e. for each component of the mixture).

2.1 The Update Equations

It can be shown [7] that for mixture models of the form:

$$P(y_i|\alpha, \Theta = \{\theta_1, \dots, \theta_k\}) = \sum_{j=1}^k \alpha_j p_j(y_i|\theta_j) \quad (2.1)$$

the complete data log-likelihood can be represented as:

$$Q(\Theta^{(n+1)}, \Theta^{(n)}) = \sum_{j=1}^k \sum_{i=1}^N \log(\alpha_j) p(j|y_i) + \sum_{j=1}^k \sum_{i=1}^N \log(p_j(y_i|\theta_j)) p(j|y_i, \Theta^{(n)}) \quad (2.2)$$

and maximised to obtain an estimate of the classification weights and assignments of each data point.

A diagonal generalised Gaussian mixture model can be written in the form of equation 2.1, with:

$$p_j(y_i|\theta_j) = \frac{\omega^d(B)}{|\sigma_j|} \exp\left(-C(B) \sum_{e=1}^d \left| \frac{(y_i^e - \mu_j^e)}{\sigma_j^{ee}} \right|^{\frac{2}{1+B}}\right) \quad (2.3)$$

In general, as α_l and θ_l are independent, the two terms can be maximised independently. In the case of the generalised Gaussian mixture model considered here, this leads to the set of update equations, where $\Theta = \{\mu, \sigma, B\}$:

$$\alpha_j^{(n+1)} = \frac{1}{N} \sum_{i=1}^N p(j|y_i, \Theta^{(n)}) \quad (2.4)$$

$$(\mu_j^m)^{(n+1)} \approx \frac{1}{N\alpha_j^{(n+1)}} \sum_{i=1}^N y_i^m p(j|y_i, \Theta^{(n)}) \quad (2.5)$$

$$(\sigma_j^{mm})^{(n+1)} = \left(\frac{1}{N\alpha_j^{(n+1)}} \sum_{i=1}^N \left(\frac{2C(B)}{1+B} \right) |y_i^m - (\mu_j^m)^{(n)}|^{\frac{2}{1+B}} \right)^{\frac{1+B}{2}} \quad (2.6)$$

The equations for α and σ are obtained exactly by analytical means. The equation for μ is only approximate, but provides good results and has a theoretical basis. The reader is referred to appendix A for more details of both the approximation and the derivations.

2.2 Updating B

It is not possible to obtain an analytical expression for the update of the kurtosis parameter, B . However, three approaches have been used to determine suitable values for B and all of them appear to converge rapidly. The most naïve approach is simply to run simulations for several fixed values of B and pick the maximum likelihood solution. A slightly more sophisticated approach is to update B at each step by scanning through the range of B and picking the maximum likelihood solution – this was implemented and found to converge rapidly, although it is strictly an implementation of the generalised EM algorithm. Finally, a numerical maximisation of the log likelihood with respect to B was carried out. This was found to converge rapidly and reliably.

2.3 Relationship between B and ρ_4

The relationship between B and kurtosis for a one dimensional distribution can be demonstrated to be monotonic and of the form shown in figure 2.1. This suggests that a generalised Gaussian classification model could be used to estimate the kurtosis of data and hence provide some information on whether a Gaussian cluster model is likely to be appropriate.

An analytical expression for this relationship has been obtained [6]:

$$\rho_4(B) = \frac{\Gamma[\frac{5}{2}(1+B)]\Gamma[\frac{1}{2}(1+B)]}{\{\Gamma[\frac{3}{2}(1+B)]\}^2} - 3 \quad (2.7)$$

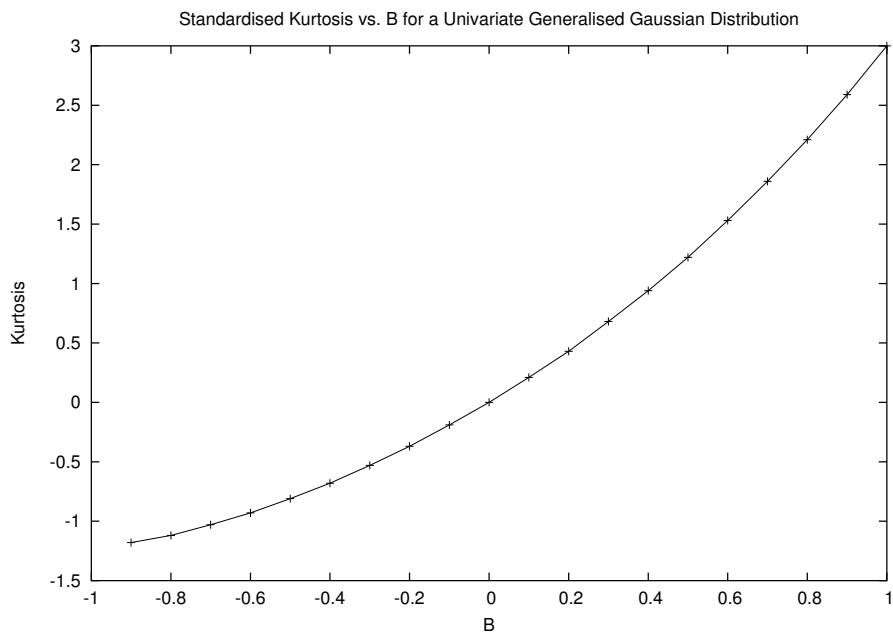


Figure 2.1: Variation of standardised kurtosis with B for a univariate generalised Gaussian.

Chapter 3

Classifier Implementation

A classifier which makes use of the generalised Gaussian model has been implemented using the EM algorithm for the sake of speed and simplicity. Such a classifier provides a simple mechanism for comparing the performance of Gaussian, Laplacian and generalised models when used for classification. It should be noted that this is a maximum likelihood classifier and, as such, will provide very limited information about the likely errors of the estimate and provides no mechanism for taking account of prior knowledge of the correct classification. Additionally, as a greedy algorithm, the EM algorithm can converge to a local rather than global maximum of the likelihood.

3.1 Classifier Model

As a model for this classifier, we took a mixture of axis-aligned generalised Gaussians which were assumed to have diagonal covariance matrices. The assumption of orthogonality coupled with that of axis alignment is a weakness of the method, but it renders analytical maximisation of the likelihood with respect to two of the four model parameters tractable. An analytical approximation to a third update equation (that for μ) is also used. The assumption is not as severe as it may at first appear as it is possible to use a suitable orthogonalisation procedure as a preprocessing step and good results have been obtained by using both principle component analysis and independent component analysis for this. See sections 5.1 and 5.2 for some examples.

This likelihood associated with this model has the form:

$$P(y|\alpha, \mu, \sigma, B) = \prod_{i=1}^N \sum_{j=1}^k \alpha_j p_j(y_i|\mu_j, \sigma_j, B) \quad (3.1)$$
$$p_j(y_i|\mu_j, \sigma_j, B) = \frac{\omega^d(B)}{|\sigma_j|} \exp\left(-C(B) \sum_{e=1}^d \left| \frac{(y_i^e - \mu_j^e)}{\sigma_j^{ee}} \right|^{\frac{2}{1+B}}\right) \cdot \mathbb{I}_{[1, \dots, k]}(j)$$

where d is the dimensionality of the feature vectors, N is the number of feature vectors and k is the model order. It would be simple to extend this model to permit B to take different values for different clusters and, in the diagonal case dealt with here, for different directions within a single cluster.

Run	Mean 1	Mean 2	Mean 3
0	0,0	0,9	12,0
1	0,0	0,6	8,0
2*	0,0	0,6	0,8
3	0,3	5,5	0,10
4	0,0	0,1	1,0
5	0,0	0,9	1,1
6	0.9, 4.4	5.4,9.0	-0.7, 7.0
7	0,0	-0.5, 1	0,5,1
8	0,0	0.5,-1	0.5,1
9	0,0	6,0	8,0

Table 3.1: Some initial conditions with which the algorithm was applied to Gaussian data with actual means corresponding to entry 0. Only run 2 failed to converge (with a misclassification of 0.33% in all other cases).

3.2 Initialisation

The EM algorithm is a greedy local maximisation algorithm and, as such, it is necessary to provide an initial “guess” at the parameter values. Any classification problem has a multi-modal solution as simple permutation of the class labels for any locally-optimal solution will produce an additional, equally likely solution. Problems will only arise with the classification of data if there is multi-modality beyond this label-switching phenomenon, and there are multiple distinct – even after arbitrary label-switching – maxima which can each be reached by the algorithm from a substantial number of initial assignments. It is necessary to provide an initial configuration which assigns more than one point to each cluster with a significant probability or the algorithm will simply shrink the singleton clusters to a point. Thus far, this has not proved to be a problem with real or simulated data.

In the present case, this initialisation phase is handled by running the K-means algorithm [8] (itself seeded by manually selecting points which appear to be roughly close to the centre of clusters of points – or, in fact, arbitrary points within the range of the data) on the data and using the resultant hard assignment to calculate the mean and generalised deviation of each cluster. The user is referred to appendix B for an outline of this algorithm. Whilst this involves an implicit assumption of Gaussianity, this stage is only used to provide some initial parameters and should not unduly influence the outcome of the process.

The algorithm converges to a stable state which is largely independent of the initial configuration. To illustrate this, Gaussian data was generated consisting of three clusters, each of 100 data points. These clusters were all spherical 2D Gaussians and had standard deviations 1, 2 and 3, respectively, with means (0, 0), (0, 9) and (12, 0). The algorithm was run with the “correct” centres and 9 other sets of initialisation values and only failed to converge in one case (Number 2 – a set of co-linear means which clearly do not describe the data). The configurations used are shown in table 3.1.

Whilst more sophisticated methods could be used to locate approximate

cluster centres, or approximate clusterings if the K-means algorithm is avoided entirely, it has not proved necessary to do so to obtain good clusterings of all the data which we have looked at thus far. One case in which a more subtle approach would be required is that in which there are isolated outliers as the K-means algorithm will lead to clusters in which these are the only points and there is no mechanism, within the EM classifier framework described here, by which a singleton “cluster” can expand to encompass other points.

3.3 Iteration

Each iteration was carried out by updating each of the parameters in turn to independently maximise the likelihood of the parameters given their previous value. This was done analytically using equations 2.4 to 2.6 for α , μ and σ and numerically, using the GSL [9] implementation of Brent’s [10] algorithm, for B .

We chose to restrict B to the range $(0 : 1)$ as this corresponds to the positive kurtosis subset of the generalised Gaussian distributions which we are interested in. It should be straightforward to extend B some way toward -1 , although numerical calculations become difficult close to $B = -1$ as the power within the exponent of the probability distribution becomes large and ultimately infinite.

3.4 Convergence & Speed

As this classifier implementation is entirely deterministic, it’s relatively simple to monitor convergence. In practice convergence always appear to occur in fewer than ten iterations with small data sets, and fewer than 50 iterations with 5,000 data vectors. With our C implementation of the algorithm, simulations running with 300 2d data using three classes, ten iterations plus the initialisation phase took approximately 5.5 s when running on a 2.4 GHz Pentium 4 under Linux kernel version 2.4.18-14. An implementation which uses a fixed value of B throughout and just updates α , σ and μ at each iteration takes less than 1.0 s in the same circumstances.

3.5 Outliers & Uncertain Assignments

As with any classification method, it is useful to have a mechanism for dealing with both outliers and points which cannot be assigned with any degree of confidence to a particular cluster.

One approach for dealing with outliers is to introduce an additional mixture into the model, an outlier class which has a uniform probability density over the sample space. Due to the exponential form of the generalised Gaussian this was not useful; it tended to classify far too many points as outliers.

The simplest method for dealing with uncertainty in classification is to identify those points with an assignment probability below some threshold as belonging to an unknown cluster. A threshold value of 0.6 produced promising results with a substantial fraction of points which were incorrectly assigned without this mechanism being correctly identified.

Chapter 4

Simulations

The classifier has been tested with a quantity of simulated data and the results appear to indicate that the assumed form of clusters does significantly influence the final classification.

The testing took the form of using the full EM algorithm classifier to determine a preferred value for B whilst also using a fixed- B classifier with otherwise identical parameters to determine the classification which would be obtained at each value of B , and considering the fraction of points which were incorrectly assigned. This provides both a measure of the performance of the full classifier, and some data about how sensitive the classification is to the value of B (and, of course, how this classification compares to that which might be obtained with a simple Gaussian classifier).

4.1 Gaussian and Laplacian Data

Initially, the model was tested with Gaussian and Laplacian data to determine how well it would perform in these cases. Illustrations have been provided for some representative distributions in figures 4.1 and 4.2. The parameters of the distributions which have been illustrated here are shown in table 4.1.

Distribution	k	N	μ	σ
L1	3	100 100 100	(0,0) (0,100) (100,0)	(1,1) (25,25) (25,25)
L2	3	100 100 100	(0,0) (0,90) (90,0)	(1,1) (25,25) (25,25)
L3	2	100 200	(0,0) (12,0)	(1,1) (3,3)
L4	3	100 100 100	(0,5) (-14,0) (3,-5)	(1,1) (3,3) (2,2)
G1	3	100 100 100	(0,0) (6,0) (0,8)	(1,1) (2,2) (3,3)
G2	3	100 100 100	(0,0) (8,0) (0,10)	(1,1) (2,2) (3,3)
G3	3	100 100 100	(0,0) (-50,0) (50,0)	(1,1) (22,22) (22,22)
G4	3	100 100 100	(0,0) (-50,0) (50,0)	(1,1) (25,25) (25,25)

Table 4.1: Parameters of the Laplacian and Gaussian distributions for which results are shown in figures 4.1 and 4.2. Note k refers to model order and N to the number of data generated in each cluster.

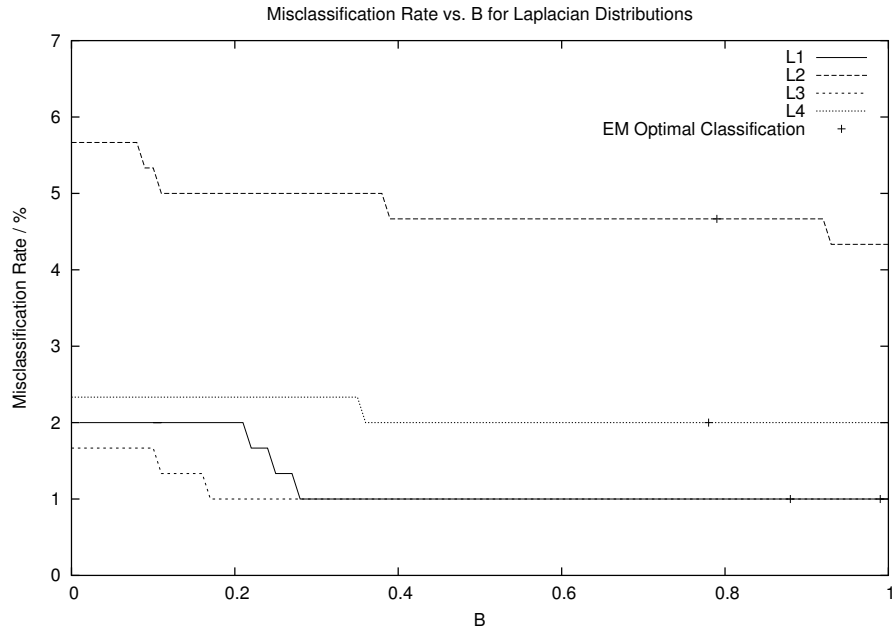


Figure 4.1: Classification error vs. B for simulated Laplacian data.

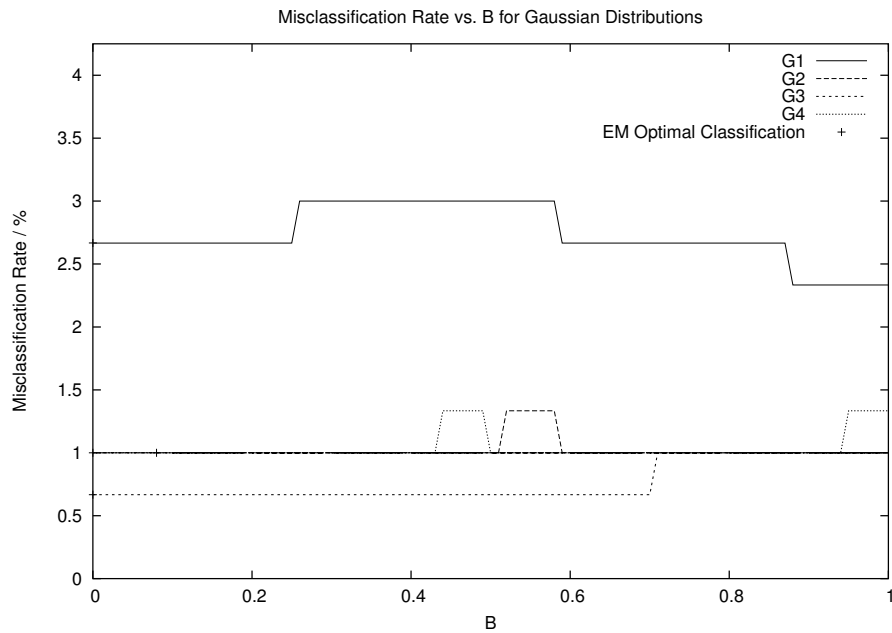


Figure 4.2: Classification error vs. B for simulated Gaussian data.

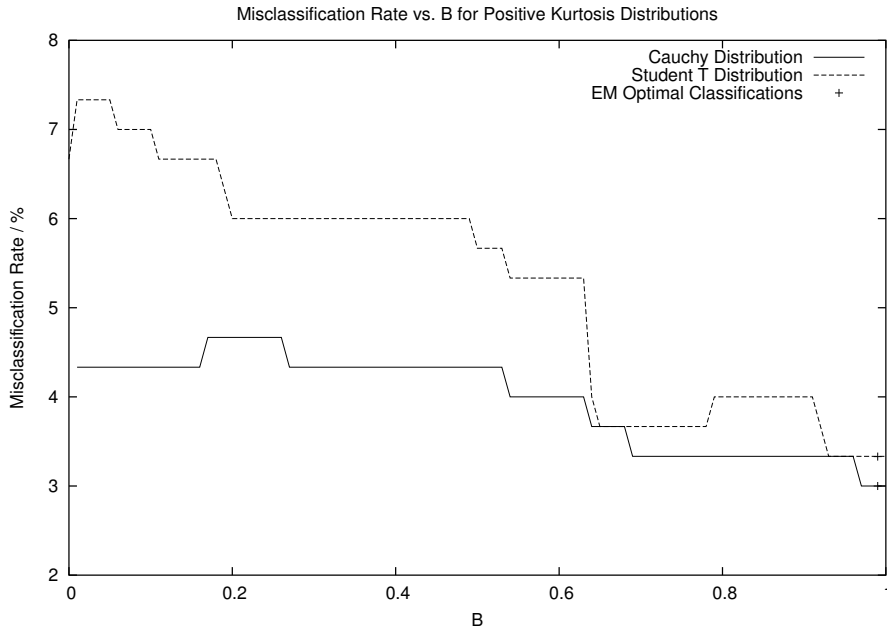


Figure 4.3: Classification error vs. B for data simulated as three Cauchy distributions with 100 members with means $(-100, -50)$, $(0, 0)$ and $(50, 100)$ and shape parameters 1, 2 and 3, respectively; and for data sampled as 100 points from each of three Student T distributions of shape parameter one centred at $(0, 0)$, $(12, 6)$ and $(-8, -16)$. $B = 0.99$ was preferred by the classifier in both cases.

4.2 Positive Kurtosis Distributions

The classifier was also tested on some more general positive kurtosis distributions, the Student-t and Cauchy (Lorentzian) distributions. These results suggest that the generalised Gaussian model may be better suited to classifying data from this type of source than a simple Gaussian method. However, it also suggests that Gaussian methods which have been supplied to such data are likely to have suffered from a significant error rate.

It has been shown [1] that data extracted from DNA oligonucleotide microarrays by the ratio of medians method are likely to be distributed according to a Cauchy, rather than normal distribution. There is also a theoretical argument that supports this: data obtained as the ratio of two correlated normal distributions might be expected to have a non-Gaussian distribution with heavier tails in the presence of significant noise[11]. [2] suggests that a Student-t distribution is more suitable than a Gaussian distribution. Given the results obtained here, it seems likely that a generalised Gaussian classifier model will give better results than a Gaussian classifier for this type of data.

The results for moderately well separated clusters of these distributions are shown in figure 4.3.

4.3 Comparison with Autoclass

The Autoclass-C program (version 3.3.4) [12] was used to classify some of the data sets illustrated in the previous sections. This particular classifier was chosen because it performs extremely well for data with a Gaussian distribution and uses an algorithm which is significantly more sophisticated than that employed by the general Gaussian classifier described above. In spite of this, however, Autoclass might be expected to perform poorly on data which is not Gaussian.

The program's preferred classification, including model order, as judged by its MAP (maximum a posteriori) estimate was used in each case and in some cases this lead to incorrect model orders for data generated by widely non-Gaussian data. This is neither surprising nor the fault of the classifier, but an inevitable consequence of ignoring the deviation of data from an assumed model and applying inappropriate algorithms to data.

Initially, the program was run using the same model structure (independent Gaussian distributions on all parameters, which were assumed to be real scalar parameters) and search parameters (assumed measurement error of 0.001, maximum of 50 iterations) for all of these data.

The results (in terms of misclassification error, relative to the preferred classification) are as follows:

Dataset	Preferred K	Misclassification / %
L4	5	N/A
G4	3	1.333
Cauchy	13	N/A
Student-t	10	N/A

The failure in the case of the Cauchy distribution illustrates the damage that can be done by assuming normally distributed errors in cases where this is not the case: the probability of a 3 class clustering for this data was given as being $\exp(-6752.560)$ relative to that of 13 classes which was $\exp(-6257.453)$. The situation with the Student-t distribution is similar; no classification with fewer than four classes was present in the "10 Best Results" generated with these parameters. In the Laplacian case, the probability of a 3 class clustering for the data was only a factor of fifty lower than the preferred classification.

It seems likely that if real data with the properties of the simulated data mentioned here were generated then a rather larger allowance for measurement error would be made. To test the hypothesis that this might improve the chance of obtaining reasonable numbers of clusters, the measurement error was set to one for the Laplacian and Gaussian distributions and 10 for the Cauchy and Student distributions and the simulations repeated.

Dataset	Preferred K	Misclassification / %
L4	2	N/A
G4	3	1.667
Cauchy	3	> 30
Student-t	3	> 30

The probability of the Laplacian data having a classification with three clusters is now 50 times lower than the probability that it has two clusters. The "3 cluster" classifications of the Cauchy and Student-t data both have one cluster

of just two outlying values. It is clear from these results that it is important to treat non-Gaussian data carefully; the assumption of Gaussianity is a very strong one due to the extremely light tails of the distribution. It is clear that even a sophisticated classification scheme will fail if the data which it is presented with differs markedly and qualitatively from the model which it uses to classify that data.

The fact that the truly Gaussian data, G4, is well classified for two different assumptions of measurement noise which differ by a factor of 1,000 suggests that the precise value of this quantity is of much less import than the qualitative form of the data distribution.

Chapter 5

Application to Real Data

In order to demonstrate that this method works for real data obtained in the presence of measurement errors and potentially significant measurement overlap, we applied the classifier described above to some real data.

5.1 Iris Data

The first data which we considered were the measurements of Fisher’s irises [13]. This data consists of four measurements of physical components of fifty irises from each of three species (Iris Sertosa, Iris Verginica and Iris Versicolor).

The generalised Gaussian classifier produced an optimal classification with $B = 0.02$ with 6.67% of the data misclassified. As these data are clearly correlated and this model assumes axis alignment, some kind of orthogonalising transformation seems likely to improve classification. A PCA (Principal Component Analysis, or Hotelling Transform [14]) approach will allow dimension reduction as well as aligning the maximum variance directions of the distribution with the axes. Using principal components makes the assumption that the data has a Gaussian distribution, but in this case such an assumption is justified by both the central limits theorem and the results of the initial classification ($B = 0.02$ is strong support for a close to Gaussian distribution).

Performing PCA and using the first two principal components (PC) as a new feature leads to a classification in which the likelihood is optimised by $B = 0.17$ but the misclassification rate is uniformly 2.0% for $B \in (0 : 1)$. The uniformly similar performance is an artifact of the good separation provided by the initial transformation.

This performance matches that of the “non-Gaussian ICA mixture model” proposed in [15], which cites a misclassification error of 3.3% for Autoclass and 4.7% for K-means clustering when each was applied to the *raw* data. This non-Gaussian classification method makes use of a Laplacian approximation for all super-Gaussian densities, unlike that presented here. Other full-covariance Gaussian mixture model classifiers have also been found to misclassify 3.3% of the iris data points (5 of the 150 data): [16]¹, [17].

¹However, Alpaym demonstrated that better classification (a 2% misclassification rate) could be obtained by constraining the three covariance matrices to be identical.

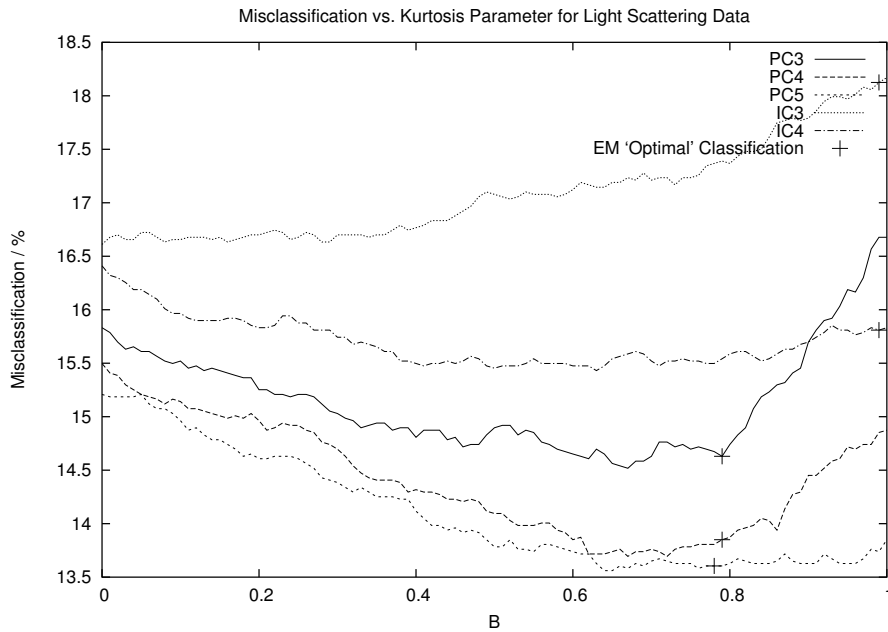


Figure 5.1: The proportion of points misclassified among the light scattering data after various transformations plotted against the kurtosis parameter, B . PC n denotes classification on the basis of n principal components; IC n denotes classification on the basis of n independent components. It should be noted that the variation with B is complicated due to the large, complex data set; failure to find the global maximum is an inevitable consequence of the use of a local maximisation algorithm.

This example demonstrates that the algorithm works and also indicates that the PC of a dataset which follows a Gaussian distribution can provide a good, orthogonal basis for classification. However, this is nothing new as a simple Gaussian classifier would produce equally good results.

5.2 Laser Scattering Data

Some light scattering data from water contaminated with *e. coli* and plain tap water was obtained. This consisted of 16 element feature vectors: 2581 from contaminated water and 1910 from clean water. In order to orthogonalise the data as much as possible, and to facilitate dimension reduction, both PCA and ICA (independent component analysis [18], using the Fast-ICA algorithm [19]) were used and several different degrees of dimension reduction were considered. With a dataset of this size, we found it necessary to use a larger number of iterations to ensure convergence of the EM algorithm: a value of 50 was found to be suitable.

The proportion of points correctly identified using the full generalised mixture model after the data had been so processed were:

# Components Used	PCA	ICA
2		0.80
3	0.85	0.82
4	0.86	0.84
5	0.86	

The preferred values of B in each of these cases were:

# Components Used	PCA	ICA
2		0.99
3	0.794	0.99
4	0.789	0.99
5	0.778	

Whilst these rates are far from ideal, they do show a reasonable performance for a naïve classification program applied to a reduced dimensional form of complex, cluttered data. Figure 5.1 shows that, with the exception of the IC3 case, improved classification is obtained by using the generalised Gaussian formulation, rather than relying upon an assumption of normality within the transformed data vectors. The exception can be explained by the fact that too much data is discarded if just three independent components are relied upon to perform the classification, and the most likely classification of this data set does not correspond to the “true” classification.

The large values of B preferred by the classifier for this data set are significant. Although the improvement in classification obtained by using this value of B rather than a standard Gaussian distribution are small (but finite, and positive), the most important point is the predictions which will be made for new data not close to the centres of existing clusters will be substantially different under the model described by large values of B and as such, in the case of supervised classification, one might expect to get substantially better results with new data. This is in contrast to the common problem of over-fitting which can occur when too many free parameters are tuned to a data set. We suggest that this may be the case because the variation of B is very important when describing non-Gaussian data.

It can be seen that for this data set, better results appear to be obtained with PCA than with ICA, and the clusters which exist after the PCA dimension reduction are non-Gaussian to a significant extent.

In order to demonstrate that the improved classification obtained with $B > 0$ is not a consequence of over-fitting, cross validation was performed using the PC4 data set. A classification was applied to both 50% and 90% of the data, and the resulting values of B , σ and μ were used to classify the full data set. The variation in the misclassification rate was less than 0.1% (corresponding to four data points in the full sample) and, thus, everywhere considerably better (more than 1%) than the classification provided by a Gaussian clustering of the full data set.

Chapter 6

Conclusions

The classification method developed here is rather crude and has the limitations which might be expected with any maximum likelihood method. However, it has been demonstrated that it might perform better than a Gaussian mixture model method of comparable complexity. The use of axis-aligned generalised Gaussians within the mixture model is not ideal, but relaxing the alignment condition complicates the form of the equations to the point where analytic maximisation of the model parameters is unlikely to be tractable. We have found that using PCA or ICA as a preprocessing step allows the classifier to be applied to non-axis-aligned clusters (such as Fisher’s irises, [13]) and provide good classification.

More significant is the illustration of the effect of non-Gaussianity on the performance of a Gaussian classifier compared with a more general model. It can be seen that the generalised classifier performs better than a Gaussian classifier for super-Gaussian (i.e. positive kurtosis) distributions even if those distributions differ in form substantial from the Laplacian case. It is also clear that the generalised model contains the Gaussian formulation within it, and that classification results do not suffer if the data has a genuinely Gaussian distribution¹.

It has been demonstrated that this classifier can provide good classification of Fisher’s iris dataset, and furthermore that it can better classify some previously unpublished light scattering data than a naïve Gaussian classifier; an improvement $\sim 1-2\%$ is observed for a number of different preprocessing conditions (between 3 and 5 components obtained from 16 dimensional data by PCA and ICA). This is a substantial fraction of those points which are not clearly members of a particular cluster. Holdout cross validation indicates that this is a genuine improvement, not simple overfitting.

The failure of AutoClass C – undoubtedly a more sophisticated classifier, but one which it is only appropriate to apply to data which is distributed according to a Gaussian distribution – to produce good classification of non-Gaussian

¹Except through the simplifying assumption made with this particular implementation that clusters may be assumed to be axis-aligned. Although it is more difficult to deal with a general covariance matrix in the general case than it is in the simple Gaussian case, it should be possible to develop a generalised Gaussian method for general covariance matrices. Suitable preprocessing of data, such as PCA, has been found to alleviate the difficulties associated with this assumption.

simulated data demonstrates that more work is required in the development of non-Gaussian model based classifiers before the assumption of normality which is so commonly used can be justified. The classification algorithm developed here provides an estimate of B which is monotonically related to the standardised kurtosis of a distribution. This estimate can be used to give a qualitative indication of how “close to Gaussian” a given data set is.

Appendix A

Derivation of the Update Equations

The following is a brief derivation of the EM update equations for a generalised Gaussian mixture model with a likelihood of the form:

$$L = \prod_{i=1}^N \sum_{j=1}^k \frac{\omega^d(B)}{|\sigma_j|} \exp \left(-C(B) \sum_{e=1}^d \left| \frac{(y_i^e - \mu_j^e)}{\sigma_j^{ee}} \right|^{\frac{2}{1+B}} \right) p(j|y_i, \Theta) \quad (\text{A.1})$$

Such a mixture model has a log-likelihood which consists of a sum of logarithms of sums. It has been shown [7] that this rather complicated form can be reduced, for general mixture models, to a complete data log-likelihood of the form:

$$\begin{aligned} \mathcal{L}(\Theta^{(n+1)}|\Theta^{(n)}) &= \sum_{i=1}^n \sum_{j=1}^k \log(\alpha_j) p(j|y_i, \Theta^{(n)}) \\ &+ \sum_{i=1}^n \sum_{j=1}^k \log(p_j(y_i|\theta_j^{(n)})) p(j|y_i, \Theta^{(n)}) \end{aligned} \quad (\text{A.2})$$

And as logarithms are monotonic functions, this may be maximised in place of the likelihood itself. It can be seen that the term containing α and that containing θ are independent and hence the full expression is maximised by the values of these parameters which maximise the two expressions independently.

The first can be dealt with generally, and is independent of the form of the probability densities of the mixture members. Using the method of Lagrange multipliers to constrain the assignment probabilities, $\{\alpha\}$ to sum to one and maximising the first term in this expression with respect to α_l , it can be seen that:

$$\alpha_j^{(n+1)} = \frac{1}{N} \sum_{i=1}^N p(j|y_i, \Theta^{(n)}) \quad (\text{A.3})$$

The more interesting second term can also be maximised with respect to two of the three parameters by analytical means as this is (to the authors' best knowledge) novel, this derivation is given in more detail than those above.

The full form of the second term of equation A.2 is:

$$l = \sum_{j=1}^k \sum_{i=1}^N \left(d \log(\omega(B)) - \log(|\sigma_j|) - C(B) \sum_{m=1}^d \left| \frac{y_i^m - \mu_j^m}{\sigma_j^m} \right|^{\frac{2}{1+B}} \right) p(j|y_i, \Theta^{(n)}) \quad (\text{A.4})$$

where μ_j^m refers to the m th element of μ_j and σ_j^m refers to the m th element of σ_j and $\Theta^{(n)}$ refers to the value of the parameters after n iterations of the EM algorithm. Where it is necessary to refer to the value of the m th element of μ_j after n iterations, the notation $(\mu_j^m)^{(n)}$ has been used.

A.1 Updating μ (approximately)

In order to update μ it is necessary to maximise

$$l = \sum_{j=1}^k \sum_{i=1}^N \left(d \log(\omega(B)) - \log(|\sigma_j|) - C(B) \sum_{m=1}^d \left| \frac{y_i^m - \mu_j^m}{\sigma_j^m} \right|^{\frac{2}{1+B}} \right) p(j|y_i, \Theta^{(n)}) \quad (\text{A.5})$$

with respect to μ_j^m . In order to do this analytically, it is necessary to solve the equation:

$$\sum_{i=1}^N p(j|y_i, \Theta^{(n)}) \operatorname{sgn}(y_i^m - \mu_j^m) |y_i^m - \mu_j^m|^{\frac{1-B}{1+B}} = 0 \quad (\text{A.6})$$

This is a non-trivial exercise; it is useful to consider some special cases. First, the Gaussian case, $B = 0$ leads to the rather simpler equation:

$$\sum_{i=1}^N p(j|y_i, \Theta^{(n)}) (y_i^m - \mu_j^m) = 0 \quad (\text{A.7})$$

with the obvious solution that:

$$(\mu_j^m)^{(n+1)} = \frac{\sum_{i=1}^N p(j|y_i, \Theta^{(n)}) y_i^m}{\sum_{i=1}^N p(j|y_i, \Theta^{(n)})} \quad (\text{A.8})$$

Now consider the general $B \in (0 : 1]$ case with $N = 2$:

$$p(j|y_1, \Theta^{(n)}) \operatorname{sgn}(y_1^m - \mu_j^m) |y_1^m - \mu_j^m|^{\frac{1-B}{1+B}} = -p(j|y_2, \Theta^{(n)}) \operatorname{sgn}(y_2^m - \mu_j^m) |y_2^m - \mu_j^m|^{\frac{1-B}{1+B}} \quad (\text{A.9})$$

Which again has a simple solution:

$$(\mu_j^m)^{(n+1)} = \frac{\sum_{i=1}^N p(j|y_i, \Theta^{(n)})^{\frac{1+B}{1-B}} y_i^m}{\sum_{i=1}^N p(j|y_i, \Theta^{(n)})^{\frac{1+B}{1-B}}} \quad (\text{A.10})$$

Requiring consistency of the general result with both of these special cases, and adopting as an axiom that the form of μ must be a weighted average of the data vectors with the weighting provided by some power of the assignment probabilities of those data vectors, justified in part by the symmetry of the system and in part by pragmatism, leads to:

$$(\mu_j^m)^{(n+1)} = \frac{\sum_{i=1}^N p(j|y_i, \Theta^{(n)})^{\gamma(N,B)} y_i^m}{\sum_{i=1}^N p(j|y_i, \Theta^{(n)})^{\gamma(N,B)}} \quad (\text{A.11})$$

where $\gamma(N, B)$ is some function which must equal unity for $B = 0$ and which must equal $(1 + B)/(1 - B) \forall B \in (0 : 1]$ in the $N = 2$ case. We also have from the $N = 2$ case that γ must be an increasing function of B which does not fall below unity at any point.

Thus, if we adopt the pragmatic approach that this is a reasonable form for the update equation, and neglect the influence of N on γ (which seems reasonable as it has no effect in the $B = 0$ case) then the difference between the general update equation and that which can be analytically determined for the Gaussian case is simple. The super-unity power γ of the probabilities in the weighted sum will further reduce the influence of points which do not have a high probability of belonging to a given cluster on determining the location of the centre of that cluster. As the influence of “distant” points on a cluster’s mean position is already exponentially inhibited by the assignment probability, this might be expected to have a limited effect upon the ultimate position of the cluster centres – their positioning is already dominated by those points which almost certainly belong to the cluster which they represent.

In practice, the assignment probabilities of the vast majority of data are either very close to 0 or very close to 1 as the exponential variation of the individual probability densities penalises even small displacements from the contours of equal probability of any pair of clusters very heavily.

It has also been asserted that equation A.8 is the correct update equation to use for Laplacian clustering [20]. This suggests that it is the correct update equation at extremal values of B and a reasonable approximation at intermediate values. The results which we have obtained using this update rule for clustering highly non-Gaussian data appear to support this hypothesis.

A.2 Updating σ (exactly)

Consider the maximisation of l with respect to σ_j^m :

$$\begin{aligned} \frac{\partial l}{\partial \sigma_j^m} &= \sum_{i=1}^N p(j|y_i, \Theta^{(n)}) \left[-\frac{\partial \log \sigma_j^m}{\sigma_j^m} - C(B) \frac{\partial}{\partial \sigma_j^m} \left(\frac{y_i^m - \mu_j^m}{\sigma_j^m} \right)^{\frac{2}{1+B}} \right] = 0 \\ &\Rightarrow \sum_{i=1}^N p(j|y_i, \Theta^{(n)}) \left[\frac{2C(B)}{1+B} \frac{|y_i^m - \mu_j^m|^{2/(1+B)}}{\sigma_j^m |\sigma_j^m|^{2/(1+B)}} - \frac{1}{\sigma_j^m} \right] = 0 \\ &\Rightarrow \sum_{i=1}^N p(j|y_i, \Theta^{(n)}) \left[2C(B) |y_i^m - \mu_j^m|^{2/(1+B)} - (1+B) |\sigma_j^m|^{2/(1+B)} \right] = 0 \end{aligned}$$

Which makes it clear that, if one assumes that each σ_j is independent, then the update equation is:

$$(\sigma_j^m)^{(n+1)} = \left[\frac{\sum_{i=1}^N p(j|y_i, \Theta^{(n)}) \left(\frac{2C(B^{(n)})}{1+B^{(n)}} \right) |y_i^m - (\mu_j^m)^{(n)}|^{2/1+B^{(n)}}}{\sum_{i=1}^N p(j|y_i, \Theta^{(n)})} \right]^{\frac{1+B^{(n)}}{2}} \quad (\text{A.12})$$

Appendix B

The K-means Algorithm

The K-means algorithm, initially described in [8], is a clustering method which is equivalent to the local maximisation of the cluster likelihood subject to an assumption of spherical Gaussianity (it minimises the Euclidean distance between each data point and the centroid with which it is associated). It is necessary to specify the number of clusters which are required and a starting point for the centroid of each cluster. The algorithm is straightforward to implement and generally converges rapidly, but it is subject to a certain amount of sensitivity to both the data structure and the initial centroid locations.

The algorithm is an iterative run which consists of the following steps:

1. Decide upon a number of clusters, K .
2. Select a centroid, $\mu_k \forall k \in \{1, \dots, K\}$, for each cluster.
3. Determine the closest centroid to each point, y_i , in the dataset (of N members) and set the allocation variable z_i to a label which identifies the point as a member of the cluster associated with that centroid:

$$z_i = \arg \min_k \sqrt{(y_i - \mu_k)^2} \quad \forall i \in \{1, \dots, N\} \quad (\text{B.1})$$

4. Update the positions of the cluster centroids by setting them to the means of all those points which have been identified as members of their associated clusters:

$$\mu_k = \frac{\sum_{i=1}^N \delta_{z_i, k} \cdot y_i}{\sum_{i=1}^N \delta_{z_i, k}} \quad \forall k \in \{1, \dots, K\} \quad (\text{B.2})$$

where and $\delta_{z_i, k}$ is the Dirac delta function defined as:

$$\delta_{i, j} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \quad (\text{B.3})$$

5. If no data points have been reassigned since the last iteration, then the algorithm has converged. Otherwise, return to step 3 and iterate as necessary.

References

- [1] James P. Brody, Brian A. Williams, Barbara J. Wold, and Stephen R. Quake. Significance and statistical errors in the analysis of DNA microarray data. *Proceedings of the National Academy of Science*, 99(20):12975–12978, October 2002.
- [2] James W. Miskin. *Ensemble Learning for Independent Component Analysis*. Ph.D. thesis, University of Cambridge, Department of Astrophysics, Cavendish Laboratory, 2000.
- [3] K. P. Kreil and D. J. C. MacKay. Reproducibility assessment of independent component analysis of expression ratios from dna microarrays. Technical report, University of Cambridge, Inference Group, Department of Astrophysics, Cavendish Laboratory, October 2002.
- [4] T. R. Hughes et al. Functional discovery via a compendium of expression profiles. *Cell*, 102(1):109–126, July 2000.
- [5] 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*, 39:2–38, 1976.
- [6] G. E. P Box and G. C. Tiao. *Bayesian Inference in Statistical Analysis*. Addison Wesley, 1973.
- [7] Jeff A. Bilmes. A gentle tutorial on the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Technical report, University of Berkeley, ICSI-TR-97-021, 1997.
- [8] J. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. University of California Press, Berkeley, 1967.
- [9] Mark Galassi et al. *GNU Scientific Library Reference Manual*. Network Theory Ltd., 1.3 edition, December 2002.
- [10] Richard P. Brent. *Algorithms for Minimization without Derivatives*. Prentice Hall, 1973.
- [11] D. V. Hinkley. On the ratio of two correlated normal random variables. *Biometrika*, 56:635–639, 1969.

- [12] P. Cheeseman et al. Autoclass: A bayesian classification system. *Proceedings of the Fifth International Conference on Machine Learning*, pages 54–64, 1988.
- [13] R.A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.
- [14] H. Hotelling. Analysis of complex statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441; 498–520, 1933.
- [15] T.-W. Lee, M. S. Lewicki, and T. J. Sejnowski. ICA mixture models for unsupervised classification and automatic context switching. In *International Workshop on Independent Component Analysis*, 1999.
- [16] Ethem Alpaydm. Soft vector quantisation and the EM algorithm. *Neural Networks*, 11:467–477, 1998.
- [17] Sirajudeen Gulam Razul. *Bayesian Methods for Unified Models*. Ph.D. thesis, University of Cambridge, Signal Processing Group, Department of Engineering, 2003.
- [18] Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. *Independent Component Analysis*. Wiley Interscience, 2001.
- [19] Aapo Hyvärinen and Erkki Oja. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9(7):1483–1492, 1997.
- [20] Steven Armstrong, Peter J. W. Rayner, and Anil C. Kokaram. Restoring images taken from scratched 2-inch tape. In Stephen Marshall, Neal Harvey, and Druti Shah, editors, *Workshop on Non-Linear Model Based Image Analysis (NMBIA'98)*, pages 83–88. Springer-Verlag, July 1998.