Feature Representation and Discrimination Based on Gaussian Mixture Model Probability Densities – Practices and Algorithms

Pekka Paalanen, Joni-Kristian Kamarainen [*], Jarmo Ilonen, Heikki Kälviäinen

Department of Information Technology, Lappeenranta University of Technology, P.O.Box 20, FI-53851 Lappeenranta, Finland

Abstract

This study promotes the use of statistical methods in specific classification tasks since statistical methods have certain advantages which advocate their use in pattern recognition. One central problem in statistical methods is estimation of class conditional probability density functions based on examples in a training set. In this study maximum likelihood estimation methods for Gaussian mixture models are reviewed and discussed from a practical point of view. In addition, good practices for utilizing probability densities in feature classification and selection are discussed for Bayesian, and more importantly, for non-Bayesian tasks. As a result, the use of confidence information in the classification is proposed and a method for confidence estimation is presented. The propositions are tested experimentally.

Key words: Gaussian mixture model, EM, classifier, confidence, highest density region

[*] Corresponding author. Tel.: +358-5-6212844; fax.:+358-5-6212899.
Email addresses: `paalane@lut.fi` (Pekka Paalanen), `jkamarai@lut.fi`

1 Introduction

Recently, black box and gray box pattern recognition (PR) and feature classification methods have proved to be very powerful and methods such as multi-layer perceptron neural networks [1] and support vector machines [2] are frequently applied with a great success. Furthermore, other novel methods seem to embed feature selection into a classifier synthesis as, for example, in the AdaBoost boosting algorithm [3]. These powerful methods are also state-of-the-art methods in practice and it is justifiable to ask whether structural and statistical PR approaches are still relevant.

Drawbacks in black and gray box PR methods are often their incapability to provide confidence information for their decision or difficulty in incorporating risk and cost models into the recognition process. In many applications it is not sufficient just to assign one predefined class to new observations; for example, in face detection facial evidence, such as eye centers and nostrils, should be detected from a scene and provided to the next processing level in ranked order (best candidates first) in order to perform detection computationally efficiently [4,5]. Gray box methods may include the confidence information as an ad hoc solution. Statistical methods, on the other hand, usually provide the information in an interpretable form along with sufficient mathematical foundations. Statistical methods thus provide some advantages over black box methods; the decision making is based on an interpretable basis from which the most probable or lowest risk (expected cost) option can be chosen (e.g. Bayesian decision making [6]) and different observations can be compared based on their statistical properties.

(Joni-Kristian Kamarainen), `ilonen@lut.fi` (Jarmo Ilonen), `kalviai@lut.fi` (Heikki Kälviäinen).

In a typical PR problem, features from known observations, a training set, are provided and necessary statistics must be established for recognition of unknown observations and estimation of confidence. A class of patterns is typically represented as a probability density function (pdf) of features. Selection of proper features is a distinct and application specific problem, but as a more general consideration, finding a proper pdf estimate has a crucial impact on successful recognition. Typically, the form of the pdf is somehow restricted and the search is reduced to a problem of fitting the restricted model to observed features. Often already simple models such as a single Gaussian distribution (normal distributed random variable) can efficiently represent patterns but a more general model, such as a finite mixture model, must be used to approximate more complex pdf's; arbitrarily complex probability density functions can be approximated using finite mixture models. The finite mixture representation is a natural choice for certain kinds of observations: observations which are produced by a randomly selected source from a set of alternative sources belonging to a same main class. This kind of task occurs when object categories are identified instead of object classes. For example, features from eye centers are partitioned into closed eye and open eye, or Caucasian and Asian eye sub classes. The problem arises how probability densities should be approximated with finite mixture models and how the model parameters should be estimated. Equally important is to define correct practices for the use of pdf's in pattern recognition and classification tasks.

In this study Gaussian mixture model (GMM) pdf's are studied as finite mixture models. The two main considerations with the GMM are estimation of number of Gaussian components and robustness of the algorithm to tolerate singularities occurring due to a small sample size. It cannot be assumed that the user knows all necessary details, and thus, the estimation should be unsupervised and utilize existing approximation and statistical theories. Several estimation methods have been proposed in literature and the most prominent ones are experimentally evaluated in this study. The methods are extended to $\mathbb{C}^n$ since complex domain features, such as Gabor filter responses, seem to be convenient for some applications [4,7]. Correct classification practices are analyzed and defined based on problem characteristics: i) classifying an unknown observation into one of predefined classes, ii) finding best candidates from a set of observations, iii) deciding class association to a single known class when other classes are unknown or their samples are insufficient, and iv) concluding what useful statistical information should be provided to the next processing level. Finally, by providing implementations ([8]) for the described methods, we aim to encourage good practices when using GMM pdf's for representation and discrimination of patterns.

2 Gaussian mixture probability density function

Finite mixture models and their typical parameter estimation methods can approximate a wide variety of pdf's and are thus attractive solutions for cases where single function forms, such as a single normal distribution, fail. However, from a practical point of view it is often sound to form the mixture using one predefined distribution type, a basic distribution. Generally the basic distribution function can be of any type but the multivariate normal distribution, the Gaussian distribution, is undoubtedly one of the most well-known and useful distributions in statistics, playing a predominant role in many areas [9]. For example, in multivariate analysis most of the existing inference procedures have been developed under the assumption of normality and in linear model problems the error vector is often assumed to be normally distributed. The multivariate normal distribution also appears in multiple comparisons, in studies of the dependence of random variables, and in many other related fields. If no prior knowledge of a pdf of a phenomenon exists, only a general model can be used and the Gaussian distribution is a good candidate. For a more detailed discussion on the theory, properties and analytical results of multivariate normal distributions we refer to [9].

2.1 Multivariate normal distribution

A non-singular multivariate normal distribution of a D dimensional random variable $\boldsymbol{X} \mapsto \boldsymbol{x}$ can be defined as

$$\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x}-\boldsymbol{\mu})\right] \tag{1}$$

where $\boldsymbol{\mu}$ is the mean vector and Σ the covariance matrix of the normally distributed random variable $\boldsymbol{X}$. Multivariate Gaussian pdf's belong to the class of elliptically contoured distributions, and thus, for example equiprobability surfaces of the Gaussian are $\boldsymbol{\mu}$-centered hyperellipsoids [9].

The Gaussian distribution in Eq. 1 can be used to describe a pdf of a real valued random vector ($\boldsymbol{x} \in \mathbb{R}^D$). A similar form can be derived for complex random vectors ($\boldsymbol{x} \in \mathbb{C}^D$) as [10]

$$\mathcal{N}^{\mathbb{C}}(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma) = \frac{1}{\pi^D |\Sigma|} \exp\left[-(\boldsymbol{x}-\boldsymbol{\mu})^* \Sigma^{-1} (\boldsymbol{x}-\boldsymbol{\mu})\right] \tag{2}$$

where * denotes the adjoint matrix (transpose and complex conjugate).

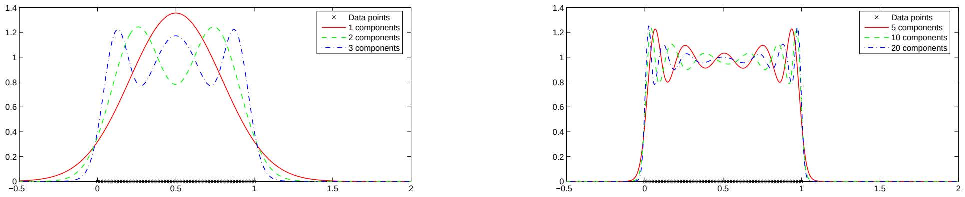
2.2 Finite mixture model

Despite the fact that multivariate Gaussian pdf's have been successfully used to represent features and discriminate between different classes in many practical problems (e.g., [11,12]) the assumption of single component leads to strict requirements for characteristics of the phenomenon: a single basic class which smoothly varies around the class mean. The most significant problem is not typically the smooth behavior but the assumption of unimodality. For multimodally distributed features the unimodality assumption may cause an intolerable error in the estimated pdf and consequently in the discrimination between classes. Errors are not only in the limited representation power but also in completely wrong interpretations (e.g. a class represented by two Gaussian distributions and another class between them). In object recognition this can be the case for such a simple thing as a human eye, which is actually an object category instead of a class since visual presence of the eye may include open eyes, closed eyes, Caucasian eyes, Asian eyes, eyes with glasses, and so on.

For a multimodal random variable, whose values are generated by one of several randomly occurring independent sources instead of a single source, a finite mixture model can be used to approximate the true pdf. If the Gaussian form is sufficient for single sources, then a Gaussian mixture model (GMM) can be used in the approximation. It should be noted that this does not necessarily need to be the case as GMMs can also approximate many other types of distributions (see Fig. 1).

The GMM probability density function can be defined as a weighted sum of Gaussians

$$p(\boldsymbol{x};\boldsymbol{\theta}) = \sum_{c=1}^{C} \alpha_c \mathcal{N}(\boldsymbol{x};\boldsymbol{\mu}_c,\Sigma_c) \tag{3}$$



(a)

(b)

Fig. 1. A uniform distribution $\mathcal{U}(0,1)$ is represented by 100 evenly spaced data points. The distribution is approximated using a Gaussian mixture model and EM parameter estimation.

where α_c is the weight of the cth component. The weight can be interpreted as *a priori* probability that a value of the random variable is generated by the cth source, and thus, $0 \leq \alpha_c \leq 1$ and $\sum_{c=1}^{C} \alpha_c = 1$. Now, a Gaussian mixture model probability density function can be completely defined by a parameter list [13]

$$\boldsymbol{\theta} = \{\alpha_1, \boldsymbol{\mu}_1, \Sigma_1, \ldots, \alpha_C, \boldsymbol{\mu}_C, \Sigma_C\} \quad . \tag{4}$$

An example of a Gaussian mixture model pdf is shown in Fig. 2.

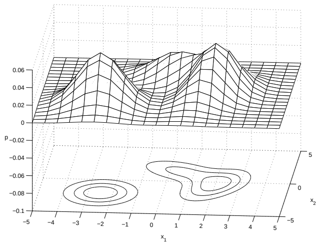


Fig. 2. Surface of a two-dimensional Gaussian mixture model pdf with three components and contour plots (equiprobability surfaces).

2.3 Estimating mixture model parameters

A vital question with GMM pdf's is how to estimate the model parameters in $\boldsymbol{\theta}$. For a mixture of C components and a D dimensional random variable $\boldsymbol{X}$ ($\boldsymbol{X} \mapsto \boldsymbol{x} \in \mathcal{R}^D$) the total number of parameters to be estimated is presented in Table 1.

The number of free parameters is lower for pure complex data than for data where real and imaginary parts are concatenated to form a real vector of double length. However, if computing with real numbers is necessary, the mapping from $\mathbb{C}$ to $\mathbb{R}^2$ should be chosen according to data. If magnitude and phase representation is used the phase may introduce a discontinuity into features [14]. Nevertheless, using purely complex representation may be advantageous, requiring less training examples in parameter estimation. In the calculations the degree of freedom for a single real variable is 1 and for a single complex variable 2. The same identity has been used in the degrees of freedom for the complex covariance matrix since it applies as an upper bound.

Table 1

Number of free parameters in a Gaussian mixture model.

Type	α_c	μ_c	Σ_c	Tot.
$\boldsymbol{x} \in \mathbb{R}^D$	1	D	$\frac{1}{2}D^2 + \frac{1}{2}D$	$C(\frac{1}{2}D^2 + \frac{3}{2}D) + C - 1$
$\boldsymbol{x} \in \mathbb{C}^D$	1	$2D$	D^2	$C(D^2 + 2D) + C - 1$
$\boldsymbol{x} \in \mathbb{C}^D \rightarrow \mathbb{R}^{2D}$	1	$2D$	$2D^2 + D$	$C(2D^2 + 3D) + C - 1$

In the literature two principal approaches exist for estimating the parameters: maximum-likelihood estimation and Bayesian estimation. While there are strong theoretical and methodological arguments supporting Bayesian estimation, in practice the maximum-likelihood estimation is simpler and, when used for designing classifiers, can lead to classifiers nearly as accurate; many implementation issues support the selection of maximum-likelihood estimation. In this study the maximum-likelihood estimation is selected purely for practical reasons.

2.3.1 Maximum-likelihood estimation

Assume that there is a set of independent samples $\mathsf{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ drawn from a single distribution described by a probability density function $p(\boldsymbol{x};\boldsymbol{\theta})$ where $\boldsymbol{\theta}$ is the pdf parameter list. The likelihood function

$$\mathcal{L}(\mathsf{X};\boldsymbol{\theta}) = \prod_{n=1}^{N} p(\boldsymbol{x}_n;\boldsymbol{\theta}) \tag{5}$$

tells the likelihood of the data X given the distribution or, more specifically, given the distribution parameters in $\boldsymbol{\theta}$. The goal is to find $\hat{\boldsymbol{\theta}}$ that maximizes the likelihood:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \mathcal{L}(\mathsf{X};\boldsymbol{\theta}) \quad . \tag{6}$$

Usually this function is not maximized directly but the logarithm

$$L(\mathsf{X};\boldsymbol{\theta}) = \ln \mathcal{L}(\mathsf{X};\boldsymbol{\theta}) = \sum_{n=1}^{N} \ln p(\boldsymbol{x}_n;\boldsymbol{\theta}) \tag{7}$$

called the log-likelihood function, which is analytically easier to handle. Because of the monotonicity of the logarithm function, the solution to Eq. (6) is the same using $\mathcal{L}(\mathsf{X};\boldsymbol{\theta})$ or $L(\mathsf{X};\boldsymbol{\theta})$.

Depending on $p(\boldsymbol{x};\boldsymbol{\theta})$ it might be possible to find the maximum analytically by setting the derivatives of the log-likelihood function to zero and by solving $\boldsymbol{\theta}$. For a Gaussian pdf the analytical solution leads to well-known estimates of mean and variance but usually the analytical approach is intractable. In practice an iterative method such as the expectation maximization (EM) algorithm is used. Maximizing the likelihood may in some cases lead to singular estimates, which is the fundamental problem of maximum likelihood methods with Gaussian mixture models [13].

If the parameters of the Gaussian mixture model pdf must be estimated for K different classes it is typical to assume independence, i.e., instances belonging to one class do not reveal anything about other classes. In the case of independent classes, the estimation problem of K class-conditional pdf's can be divided into K separate estimation problems.

2.3.2 Basic EM estimation

The expectation maximization (EM) algorithm is an iterative method for calculating maximum likelihood distribution parameter estimates from incomplete data (elements missing in feature vectors) [15]. The algorithm can also be used to handle cases where an analytical approach for maximum likelihood estimation is not feasible, such as Gaussian mixtures with unknown and unrestricted covariance matrices and means.

In the following discussion of the estimation the notation and derivations correspond to the ones used by Duda et al. [16] and Figueiredo and Jain [17].

Assume that each training sample contains known features and missing or unknown features. Existing features are represented by X and all unknown features by Y. The expectation step (E-step) for the EM algorithm is to form the function

$$Q(\boldsymbol{\theta}; \boldsymbol{\theta}^i) \equiv \mathrm{E}_{\mathsf{Y}}[\,\ln \mathcal{L}(\mathsf{X}, \mathsf{Y}; \boldsymbol{\theta}) \,|\, \mathsf{X};\, \boldsymbol{\theta}^i\,] \tag{8}$$

where $\boldsymbol{\theta}^i$ is the previous estimate for the distribution parameters and $\boldsymbol{\theta}$ is the variable for a new estimate describing the (full) distribution. $\mathcal{L}$ is the likelihood function in Eq. (5). The function calculates the likelihood of the data, including the unknown feature Y marginalized with respect to the current estimate of the distribution described by $\boldsymbol{\theta}^i$. The maximization step (M-step) is to maximize $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^i)$ with respect to $\boldsymbol{\theta}$ and set

$$\boldsymbol{\theta}^{i+1} \leftarrow \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}; \boldsymbol{\theta}^i). \tag{9}$$

The steps are repeated until a convergence criterion is met.

For the convergence criterion it is suggested (e.g. [16]) that

$$Q(\boldsymbol{\theta}^{i+1};\boldsymbol{\theta}^{i}) - Q(\boldsymbol{\theta}^{i};\boldsymbol{\theta}^{i-1}) \leq T \tag{10}$$

with a suitably selected T and (e.g. [18]) that

$$||\boldsymbol{\theta}^{i+1} - \boldsymbol{\theta}^{i}|| \leq \epsilon \tag{11}$$

for an appropriately chosen vector norm and ϵ. Common for both these criteria is that iterations are stopped when the change in the values falls below a threshold. A more sophisticated criterion can be derived from Eq. (10) by using a relative instead of an absolute rate of change.

The initialization is one of the problems with the EM algorithm. The selection of the initial value $\boldsymbol{\theta}^0$ (partly) determines where the algorithm converges or hits the boundary of the parameter space producing singular, meaningless results. Singularities occur especially with arbitrary (not restricted) covariance matrices. Some solutions use multiple random starts or a clustering algorithm for initialization [17].

In the case of Gaussian mixture models the known data X is interpreted as incomplete data. The missing part Y is the knowledge of which component produced each sample $\boldsymbol{x}_n$. For each $\boldsymbol{x}_n$ there is a binary vector $\boldsymbol{y}_n = \{y_{n,1}, \ldots, y_{n,C}\}$, where $y_{n,c} = 1$, if the sample was produced by the component c, or zero otherwise. The complete data log-likelihood is

$$\ln \mathcal{L}(\mathsf{X}, \mathsf{Y}; \boldsymbol{\theta}) = \sum_{n=1}^{N} \sum_{c=1}^{C} y_{n,c} \ln \left(\alpha_c \, p(\boldsymbol{x}_n | c; \boldsymbol{\theta}) \right) \quad . \tag{12}$$

The E-step contains computation of the conditional expectation of the complete data log-likelihood, the Q-function, given X and the current estimate $\boldsymbol{\theta}^i$ of the parameters. Since the complete data log-likelihood $\ln \mathcal{L}(\mathsf{X}, \mathsf{Y}; \boldsymbol{\theta})$ is linear with respect to the missing Y, the conditional expectation $\mathsf{W} \equiv \mathrm{E}[\mathsf{Y}\,|\,\mathsf{X}, \boldsymbol{\theta}]$ has simply to be computed and put into $\ln \mathcal{L}(\mathsf{X}, \mathsf{Y}; \boldsymbol{\theta})$. Therefore

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^i) \equiv \mathrm{E}\left[\ln \mathcal{L}(\mathsf{X}, \mathsf{Y}; \boldsymbol{\theta})\,|\,\mathsf{X}, \boldsymbol{\theta}^i\right] = \ln \mathcal{L}(\mathsf{X}, \mathsf{W}; \boldsymbol{\theta}) \tag{13}$$

where the elements of W are defined as

$$w_{n,c} \equiv \mathrm{E}\left[y_{n,c}\,|\,\mathsf{X}, \boldsymbol{\theta}^i\right] = \mathrm{Pr}\left[y_{n,c} = 1\,|\,\boldsymbol{x}_n, \boldsymbol{\theta}^i\right]. \tag{14}$$

The probability can be calculated with the Bayes law ([17])

$$w_{n,c} = \frac{\alpha_c^i\, p(\boldsymbol{x}_n | c; \boldsymbol{\theta}^i)}{\sum_{j=1}^{C} \alpha_j^i\, p(\boldsymbol{x}_n | j; \boldsymbol{\theta}^i)} \tag{15}$$

where α_c^i is the a priori probability (of estimate $\boldsymbol{\theta}^i$) and $w_{n,c}$ is the a posteriori probability that $y_{n,c} = 1$ after observing $\boldsymbol{x}_n$. In other words, $w_{n,c}$ is the probability that $\boldsymbol{x}_n$ was produced by component c.

Applying the M-step to the problem of estimating the distribution parameters for the C-component Gaussian mixture with arbitrary covariance matrices, the resulting iteration formulas are as follows:

$$\alpha_c^{i+1} = \frac{1}{N} \sum_{n=1}^{N} w_{n,c} \tag{16}$$

$$\boldsymbol{\mu}_c^{i+1} = \frac{\sum_{n=1}^{N} \boldsymbol{x}_n w_{n,c}}{\sum_{n=1}^{N} w_{n,c}} \tag{17}$$

$$\Sigma_c^{i+1} = \frac{\sum_{n=1}^{N} w_{n,c} (\boldsymbol{x}_n - \boldsymbol{\mu}_c^{i+1})(\boldsymbol{x}_n - \boldsymbol{\mu}_c^{i+1})^T}{\sum_{n=1}^{N} w_{n,c}} \quad . \tag{18}$$

The new estimates are gathered to $\boldsymbol{\theta}^{i+1}$ (Eq. 4). If the convergence criterion (Eqs. 10 or 11) is not satisfied, $i \leftarrow i + 1$ and Eqs. (15)–(18) are evaluated again with new estimates.

The interpretation of the Eqs. (16)–(18) is actually quite intuitive. The weight α_c of a component is the portion of samples belonging to that component. The weight is computed by approximating the component-conditional pdf with the previous parameter estimates and taking the posterior probability of each sample point belonging to the component c (Eq. 15). The component mean $\boldsymbol{\mu}_c$ and covariance matrix Σ_c are estimated in the same way. The samples are weighted with their probabilities of belonging to the component, and then the sample mean and sample covariance matrix are computed.

It is worthwhile noting that hitherto the number of components C was assumed to be known. Clustering techniques try to find the true clusters and components from a training set, but our task of training a classifier only needs a good enough approximation of the distribution of each class. Therefore, C does not need to be guessed accurately, it is just a parameter defining the complexity of the approximating distribution. Too small C prevents the classifier from learning the sample distributions well enough and too large C may lead to an overfitted classifier. More importantly, too large C will definitely lead to singularities if the amount of training data becomes insufficient.

2.3.3 Figueiredo-Jain Algorithm

The Figueiredo-Jain (FJ) algorithm tries to overcome three major weaknesses of the basic EM algorithm [17]. The EM algorithm presented in Section 2.3.2 requires the user to set the number of components and the number remains fixed during the estimation process. The FJ algorithm adjusts the number of components during estimation by annihilating components that are not supported by the data. The other EM failure point is the boundary of the parameter space. FJ avoids the boundary when it annihilates components that are becoming singular. FJ also allows starting with an arbitrarily large number of components, which tackles the initialization issue with the EM algorithm. The initial guesses for means of components can be distributed into the whole space occupied by training samples, even setting one component for every single training sample.

The classical way to select the number of mixture components is to adopt the ”model-class/model” hierarchy, where some candidate models (mixture pdf’s) are computed for each model-class (number of components), and then select the ”best” model. The idea behind the FJ algorithm is to abandon such a hierarchy and to find the ”best” overall model directly. Using the minimum message length criterion and applying it to mixture models leads to the objective function [17]

$$\Lambda(\boldsymbol{\theta}, \mathsf{X}) = \frac{V}{2} \sum_{c:\, \alpha_c > 0} \ln(\frac{N\alpha_c}{12}) + \frac{C_{\mathrm{nz}}}{2} \ln \frac{N}{12} + \frac{C_{\mathrm{nz}}(V+1)}{2} - \ln \mathcal{L}(\mathsf{X}, \boldsymbol{\theta}) \qquad (19)$$

where N is the number of training points, V is the number of free parameters specifying a component, and C_{nz} is the number of components with nonzero weight in the mixture ($\alpha_c > 0$). $\boldsymbol{\theta}$ in the case of Gaussian mixture is the same as in Eq. (4). The last term $\ln \mathcal{L}(\mathsf{X}, \boldsymbol{\theta})$ is the log-likelihood of the training data given the distribution parameters $\boldsymbol{\theta}$ (Eq. 7).

The EM algorithm can be used to minimize Eq. (19) with a fixed C_{nz} [17]. It leads to the M-step with the component weight updating formula

$$\alpha_c^{i+1} = \frac{\max\left\{0, \left(\sum\limits_{n=1}^{N} w_{n,c}\right) - \frac{V}{2}\right\}}{\sum\limits_{j=1}^{C} \max\left\{0, \left(\sum\limits_{n=1}^{N} w_{n,j}\right) - \frac{V}{2}\right\}} \,. \qquad (20)$$

This formula contains an explicit rule for annihilating components by setting their weights to zero. Other distribution parameters are updated as in Eqs. (17) and (18).

The above M-step is, however, not suitable for the basic EM algorithm. When initial C is high, it can happen that all weights become zero because none of the components have enough support from the data. Therefore a component-wise EM algorithm (CEM) is adopted. CEM updates the components one by one, computing the E-step (updating W) after each component update, whereas the basic EM updates all components ”simultaneously”. When a component is annihilated its probability mass is immediately redistributed strengthening the remaining components [17].

When CEM converges, it is not guaranteed that the minimum of $\Lambda(\boldsymbol{\theta}, \mathsf{X})$ is found, because the annihilation rule (Eq. 20) does not take into account the decrease caused by decreasing C_{nz}. After convergence the component with the smallest weight is removed and the CEM is run again, repeating until $C_{\mathrm{nz}} = 1$. Then the estimate with the smallest $\Lambda(\boldsymbol{\theta}, \mathsf{X})$ is chosen. The implementation of the FJ algorithm uses a modified cost function instead of $\Lambda(\boldsymbol{\theta}, \mathsf{X})$ [17].

Hitherto the only assumptions are that the EM algorithm can be written for the mixture distribution and all components are parameterized in the same way (number of parameters V for a component). With the Gaussian mixture model the number of parameters per component is $V = D + \frac{1}{2}D^2 + \frac{1}{2}D$ in the case of real valued data and arbitrary covariance matrices. With complex valued data the number of real free parameters $V = 2D + D^2$ where D is the dimensionality of the (complex) data. For complex data the number of free parameters should be replaced by the new value for V instead of the values from Figueiredo and Jain, who derived the rule for real valued data [17]. As can be seen in Eq. (20) the new value amplifies the annihilation, which makes sense because there are more degrees of freedom in a component. On the other hand there are more training data with the same number of training points, because the data is complex (two values in one variable as opposed to one value).

2.3.4 Greedy EM algorithm

The greedy algorithm starts with a single component and then adds components into the mixture one by one [19]. The optimal starting component for a Gaussian mixture is trivially computed with optimal meaning the highest training data likelihood. The algorithm repeats two steps: insert a component into the mixture, and run EM until convergence. Inserting the component that increases the likelihood the most is thought to be an easier problem than initializing a whole near-optimal distribution. Component insertion involves searching for the parameters for only one component at a time. Recall that EM finds a local optimum for the distribution parameters, not necessarily the global optimum, which makes it an initialization dependent method.

Let p_C denote a C-component mixture with parameters $\boldsymbol{\theta}_C$. The general greedy algorithm for the Gaussian mixture is as follows [19]:

Algorithm 1 *Greedy EM*

```
1: Compute the optimal (in the ML sense) one-component mixture $p_1$ and set $C \leftarrow 1$.
2: Find a new component $\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}', \Sigma')$ and corresponding mixing weight $\alpha'$ that increase the likelihood the most:
       $\{\boldsymbol{\mu}', \Sigma', \alpha'\} = \arg \max_{\{\boldsymbol{\mu}, \Sigma, \alpha\}} \sum_{n=1}^{N} \ln[(1-\alpha) p_C(\boldsymbol{x}_n) + \alpha \mathcal{N}(\boldsymbol{x}_n; \boldsymbol{\mu}, \Sigma)]$    (21)
   while keeping $p_C$ fixed.
3: Set $p_{C+1}(\boldsymbol{x}) \leftarrow (1-\alpha') p_C(\boldsymbol{x}) + \alpha' \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}', \Sigma')$ and then $C \leftarrow C + 1$.
4: Update $p_C$ using EM (or some other method) until convergence. [optional]
5: Evaluate some stopping criterion; go to Step 2 or quit.
```

The stopping criterion in Step 5 can be, for example, any kind of model selection criterion, wanted number of components, or the minimum message length criterion.

The crucial point is of course Step 2. Finding the optimal new component requires a global search, which is performed by creating CN_{cand} candidate components. The number of candidates will increase linearly with the number of components C, having N_{cand} candidates per each existing component. The candidate resulting in the highest likelihood when inserted into the (previous) mixture is selected. The parameters and weight of the best candidate are then used in Step 3 instead of the truly optimal values.

The candidates for executing Step 2 are initialized as follows: the training data set X is partitioned into C disjoint data sets $\{\mathsf{A}_c\}$, $c = 1 \ldots C$, according to the posterior probabilities of individual components; the data set is Bayesian classified by the mixture components. From each A_c the number of N_{cand} candidates is initialized by picking uniformly randomly two data points $\boldsymbol{x}_l$ and $\boldsymbol{x}_r$ in A_c. The set A_c is then partitioned into two using smallest distance selection with respect to $\boldsymbol{x}_l$ and $\boldsymbol{x}_r$. The mean and covariance of these two new subsets are the parameters for two new candidates. The candidate weights are set to half of the weight of the component that produced the set A_c. Then new $\boldsymbol{x}_l$ and $\boldsymbol{x}_r$ are drawn until N_{cand} candidates are initialized with A_c. The partial EM algorithm is then used on each of the candidates. The partial EM differs from the EM and CEM algorithms by optimizing (updating) only one component of a mixture; it does not change any other components. In order to reduce the time complexity of the algorithm, a lower bound on the log-likelihood is used instead of the true log-likelihood. The lower-bound log-likelihood is calculated with only the points in the respective set A_c. The partial EM update equations are as follows:

$$w_{n,C+1} = \frac{\alpha \mathcal{N}(\boldsymbol{x}_n; \boldsymbol{\mu}, \Sigma)}{(1-\alpha)p_C(\boldsymbol{x}) + \alpha \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma)} \tag{22}$$

$$\alpha = \frac{1}{\aleph(\mathsf{A}_c)} \sum_{n \in \mathsf{A}_c} w_{n,C+1} \tag{23}$$

$$\boldsymbol{\mu} = \frac{\sum_{n \in \mathsf{A}_c} w_{n,C+1} \boldsymbol{x}_n}{\sum_{n \in \mathsf{A}_c} w_{n,C+1}} \tag{24}$$

$$\Sigma = \frac{\sum_{n \in \mathsf{A}_c} w_{n,C+1} (\boldsymbol{x}_n - \boldsymbol{\mu})(\boldsymbol{x}_n - \boldsymbol{\mu})^T}{\sum_{n \in \mathsf{A}_c} w_{n,C+1}} \tag{25}$$

where $\aleph(\mathsf{A}_c)$ is the number of training samples in the set A_c. These equations are much like the basic EM update equations in Eqs. (16)–(18). The partial EM iterations are stopped when the relative change in log-likelihood of the resulting $C + 1$ -component mixture drops below a threshold or a maximum number of iterations is reached. When the partial EM has converged, the candidate is ready to be evaluated.

2.3.5 Avoiding covariance matrix singularities

All the above estimation methods may fail due to singularities appearing during computation of new estimates. Several heuristic explicit covariance matrix fixing procedures can be applied in order to prevent singularities. Nagy et al. [20] also present an algorithm using factorized covariance matrices that avoids singularities altogether. The method used with the EM and FJ algorithms is described next.

Computation numerics may introduce inaccuracies so that the covariance matrix is no longer strictly Hermitian (complex conjugate symmetric, $\Sigma = \Sigma^*$). The matrix is forced to Hermitian by

$$\Sigma \leftarrow \Sigma - \frac{\Sigma - \Sigma^*}{2} \tag{26}$$

and removing an imaginary part from the diagonal.

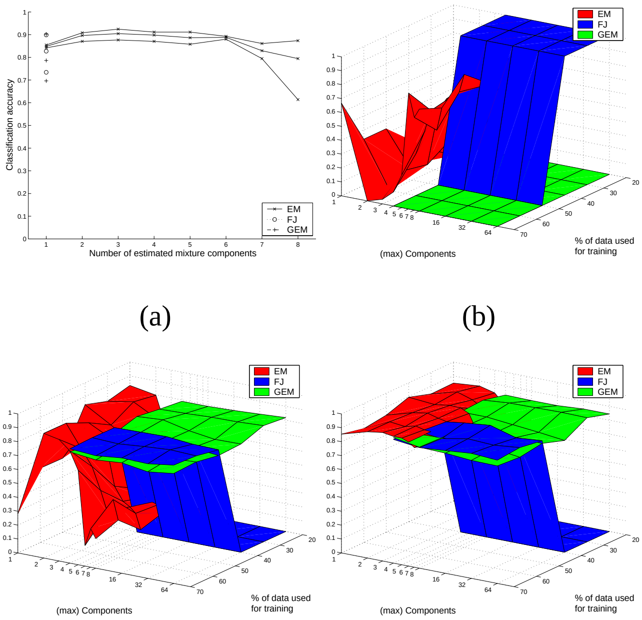
The matrix is tested for positive definiteness with Cholesky factorization (Chol). If the test is negative, the diagonal of the matrix is modified. If the diagonal contains elements that are less than 10ε, the diagonal values are grown by an amount relative to the largest absolute value on the diagonal, ensuring also that negative diagonal elements become positive enough. If the diagonal elements are larger the diagonal is grown by 1.0 percent.

2.4 Experiments

All the above methods are publicly available in the Matlab software package provided by the authors [8]. In the experiments discussed next the methods were applied in classification tasks with several publicly available data sets. A comparison between the three methods, EM, FJ and GEM, was performed to evaluate their accuracy and robustness. The experiments were conducted by varying both the parameter related to the number of components and the amount of data used for training. For FJ and GEM the number of components actually denoted the maximum number of components. Mean and maximum accuracies and algorithm crash rate were inspected. A crash is a situation where the algorithm does not produce a final mixture estimate and cannot continue, e.g., in EM a covariance matrix becomes undefined or in FJ all mixture components are annihilated.

Forest spectral data first introduced by Jaaskelainen et al. [21] was used in the first experiment. Since the original data were of high dimensionality they were projected to fewer dimensions as proposed in [22]. For training, varying amounts of the data (20–70%) were randomly selected and 30% were used for classification by Bayes rule. Fig. 3(a) shows that all methods performed equally well and minimum and maximum accuracies were both near the mean accuracy. It should be noted that in Fig. 3(a) FJ and GEM always converged to the same number of components, that is, a single Gaussian. GEM was the most stable in terms of crash rate while EM frequently crashed for a non-optimal number of components and the FJ algorithm did not succeed until enough training examples were included (Fig. 3(b)).

The crash rate is represented as a proportion of the total number of re-executions of the algorithm and re-executions with reselected data. There were no significant differences in maximum accuracies of the different methods when they succeeded in estimation as shown in Figs. 3(c) and 3(d). However, for a good result EM must be executed several times.



(a) (b)

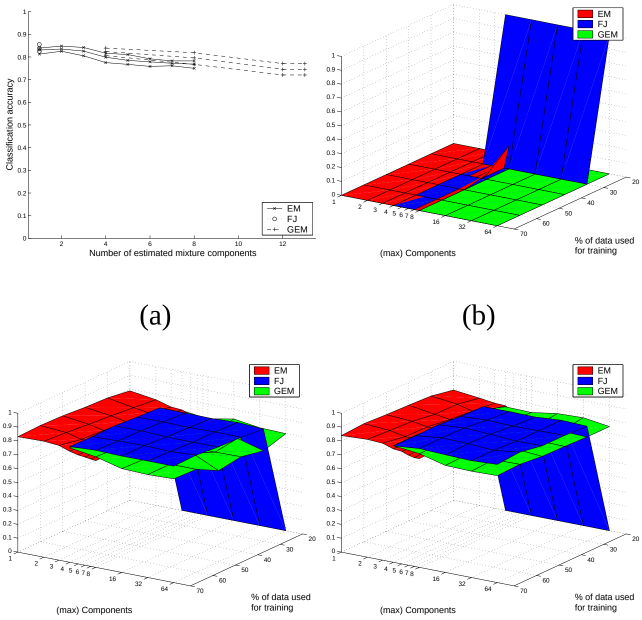
(c) (d)

Fig. 3. Results for the 23-d forest data as functions of amount of training data: (a) classification accuracy (70% of data used for training and min., max., and mean curves plotted); (b) algorithm crash rate; (c) mean accuracy; (d) maximum accuracy.

By projecting the data on fewer dimensions the data becomes smoother and all methods perform more reliably. Due to the smoothness the FJ did not crash either [23]. However, more discriminating information is lost and consequently the maximal accuracy decreases.

In the second experiment well-known waveform data were used. The waveform data consisted of 5000 40-element vectors, where the first attributes are meaningful attributes with added noise and the last 19 elements are pure noise. The optimal

Bayes classification accuracy is 86% [24]. FJ reached the optimum with one Gaussian component and the other methods came very close (Fig. 4). In Fig. 4(a) it seems that there is enough data for slight overfitting. The Greedy EM (GEM) shows that it adds components very greedily. Otherwise all methods performed equally.



(a)

(b)

(c)

(d)

Fig. 4. Results for the waveform data as functions of amount of training data: (a) classification accuracy (70% of data used for training and min., max., and mean curves plotted); (b) algorithm crash rate; (c) mean accuracy; (d) maximum accuracy.

Based on the experiments it can be said that the standard EM algorithm outperforms FJ and GEM if good prior knowledge exists about the number of components. Both FJ and GEM automatically estimate the number of components and it seems that FJ produces more accurate results, but also requires more training samples than GEM. As an implementation GEM is the most stable while the standard EM is the most unstable. The unsupervised nature of FJ and GEM, however, still motivates their use in many practical applications.

3 Feature discrimination and classification

Bayesian decision making is the most standard way to classify observations if class conditional pdf's are known but the classification can be applied only to one of the four items mentioned in the introduction: i) classification of an unknown observation into one of predefined classes. It should be noted that the plain use of the Bayes formula does not guarantee Bayesian optimal decision making and there are numerous pattern recognition problems which cannot be formulated as a Bayesian task [6].

In this study Gaussian mixture models are considered to represent class conditional pdf's. The Bayesian rule can be used to select which class to assign for an unclassified observation, but the weakness of the approach is that a posteriori probabilities can be used only to make a decision between known classes for a single observation. It is important to note that posteriors cannot be used to compare different observations, for example, to decide which one of them is more reliably from a certain class. Comparison between instances must be based on some other statistical reasoning.

The Bayesian decision making was useful only in the first problem defined in the introduction, but the other two problems: ii) finding best candidates from a set of observations and iii) deciding a class association to a single known class when other classes are unknown or their samples are insufficient must be formulated in another way. In this study use of confidence information is proposed and confidence additionally covers the last problem, iv) concluding what useful statistical information should be provided to the next processing level.

3.1 Classification using the Bayesian decision rule

Bayesian classification and decision making are based on probability theory and the principle of choosing the most probable or the lowest risk (expected cost) option [6,18]. Assume that there is a classification task to classify feature vectors (observations) to K different classes. A feature vector is denoted as $\boldsymbol{x} = [x_1, x_2, \ldots, x_D]^T$ where D is the dimension of a vector. The probability that a feature vector $\boldsymbol{x}$ belongs to a class ω_k is $P(\omega_k|\boldsymbol{x})$, and it is often referred to as a posteriori probability. The classification of the vector is done according to posterior probabilities or decision risks calculated from the probabilities [6].

The posterior probabilities can be computed with the Bayes formula

$$P(\omega_k|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\omega_k)P(\omega_k)}{p(\boldsymbol{x})} \tag{27}$$

where $p(\boldsymbol{x}|\omega_k)$ is the probability density function of class ω_k in the feature space and $P(\omega_k)$ is the a priori probability, which gives the probability of the class before measuring any features. If a priori probabilities are not actually known, they must be explicitly defined or estimated from the training set. A fatal error exists if a prioria probabilities do not exist [6]. The divisor

$$p(\boldsymbol{x}) = \sum_{i=1}^{K} p(\boldsymbol{x}|\omega_i)P(\omega_i) \tag{28}$$

is merely a scaling factor to assure that posterior probabilities are really probabilities, i.e., their sum is one. It can be shown that choosing the class of the highest posterior probability produces the minimum error probability [16,6,18]. However, if the cost of making different kinds of errors is not uniform, the decision can be made with a risk function that computes the expected cost using the posterior probabilities, and the class with minimum risk is chosen [6].

3.2 Classification using confidence

The term confidence may have various different meanings and formulations, but in our case confidence is used to measure reliability of a classification result where a class label is assigned to an observation. If confidence is low, it is more probable that a wrong decision has been made. Intuitively, the value of a class conditional pdf at an observation corresponds to confidence for a specific class: the higher the pdf value, the more instances in a corresponding class appear similar to the observation.

A posteriori is a between-class measure for a single observation, but pdf values can be used as an inter-class measure to select the best representative of a class. For example, in object detection it is computationally efficient to process best evidence candidates representing parts of an object first [5]. Unlikely evidence candidates can be pruned as outliers. In such tasks the use of confidence is beneficial.

3.2.1 Interpretation of confidence

A posteriori values can be used to select the most probable class for a single observation, but confidence values can be used to select the most reliable class representative among many observations. In certain tasks, confidence can be used to discard observations which cannot be sufficiently reliably assigned to any of known classes, that is, their pdf values are too low for reliable decision making. The same approach can be used in two class problems where training examples are available only for one class and observations which do not pass a predefined confidence level are assigned to the unknown class (e.g. in detection of motor failure conditions [12]). In that sense the confidence does not refer to a single value limit but to a certain allowed region in the feature space.

Definition 1 *Confidence value $\kappa \in [0, 1]$ and confidence region $\mathcal{R} \subseteq \Omega$ for a probability distribution function $0 \leq p(\boldsymbol{x}) < \infty, \forall \boldsymbol{x} \in \Omega$. κ is a confidence value related to a non-unique confidence region $\mathcal{R}$ such that*

$$\int_{\Omega \setminus \mathcal{R}} p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = \kappa \tag{29}$$

The definition of confidence in Definition 1 is easily interpretable via the confidence region $\mathcal{R}$. It is a region which covers a proportion $1 - \kappa$ of the probability mass of $p(\boldsymbol{x})$, because for probability distributions $\int_{\Omega} p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = 1$. It is clear that $\kappa = 1$ for $\mathcal{R}$ contains only a finite number of individual points and $\kappa = 0$ for $\mathcal{R} = \Omega$. It should be noted that the confidence value has no use until the region $\mathcal{R}$ is defined as a minimal volume region which satisfies Definition 1. The minimal volume region is called the highest density region (HDR) [25].

A class specific confidence value κ_k can be defined for each $k = 1, \ldots, K$ different classes, but intuitively the same value is good for all classes. A confidence value corresponds to the proportion of a probability mass that remains in an area $\mathcal{R}_k$. In a classification task where certain confidence for decision making is required, the confidence value itself is not used but the confidence region $\mathcal{R}_k$ itself is important since a sample vector $\boldsymbol{x}$ is allowed to enter the class ω_k only if $\boldsymbol{x} \in \mathcal{R}_k$. If a sample is not within the confidence region of any of the classes, it must be classified to a garbage class. The garbage class is a special class and samples assigned to the class need special attention; for example, more information is needed for observations in the garbage class or in a two-class problem where data is available only from one class the garbage class may represent the other class with an unknown distribution.

There are various uses for the confidence value in classification. For example, in the search for the best evidence sets from a large set of samples the use of the confidence value may decrease the work done in further processing steps. Confidence may also be directly used to guarantee the reliability of the classification. In condition monitoring, mechanical failures typically evolve during the process and too early warning may lead to too pessimistic repair procedures. It is often easy to collect samples from normal system conditions as with condition monitoring, but all failures can never be comprehensively recorded, and thus, two class classification to normal or failure condition can be performed utilizing the confidence value: samples within the confidence region are assigned to the normal class and outside that to the failure condition, whose probability distribution is unknown. Correct use and the advantages of using confidence in classification will be demonstrated and further discussed in the experimental part of this study.

3.2.2 Rank-order statistics based confidence estimation

An analytical solution for finding the confidence region (minimal volume region) would be very difficult to implement in practice. Formulation can be based on integration rules of multidimensional Gaussians and using gradient information. The formulation would be simple for a single multi-dimensional Gaussian, but becomes more difficult when there are several non-overlapping Gaussians and very complex when overlapping is allowed: integration must be done over all components in every ellipsoid. The required theory for the computations exists [26–28], but the optimization may become computationally infeasible. Thus, efficient and sufficiently accurate approximating methods are needed. One such method will be described next.

To find the confidence region, a reverse approach can be used to find a pdf value τ which is at the border of the confidence region. It is assumed that the pdf is continuous and the gradient of the pdf is never zero in a whole neighborhood of any point where the pdf value is nonzero. τ must be equal everywhere in the border, otherwise the region cannot be the minimal volume region [25]. τ can be computed by rank-order statistics using the density quantile ([25]) $F(\tau)$ and by generating data according to the pdf function. Density quantile is the opposite of confidence $\kappa = 1 - F(\tau)$.

$\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ distributed data can be generated by generating D-dimensional vectors $\boldsymbol{y}_i$, each element i.i.d. $\mathcal{N}(0, 1)$, and computing $\boldsymbol{x}_i = \boldsymbol{y}_i \ \mathrm{Chol}\ \Sigma + \boldsymbol{\mu}$ [9]. Complex valued data can be generated by generating D-dimensional vectors $\boldsymbol{z}_i$ with each element i.i.d. $\mathcal{U}(0, \pi)$ and computing $\boldsymbol{x}_i = (\boldsymbol{y}_i \cos \boldsymbol{z}_i + \boldsymbol{y}_i \mathrm{i} \sin \boldsymbol{z}_i)\ \mathrm{Chol}\ \Sigma + \boldsymbol{\mu}$. Gaussian mixture data can be generated by first randomly selecting one of the components using component weights as probabilities and then generating a point with the parameters of the selected component.

Definition 2 *The density quantile $F(\tau)$ corresponding to a pdf value τ is*

$$F(\tau) = \int_{p(\boldsymbol{x}) \geq \tau} p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \ . \tag{30}$$

The density quantile corresponds to HDR probability mass, an integral over a region where pdf values remain above the given minimal level τ. Clearly $F(\tau) \in [0, 1]$ since it is similar to a cumulative pdf. The pdf value τ has no interpretation which could be used, but it is easy to define a density quantile, e.g., a quantile of 0.9 would accept the most typical 90% of instances generated by the pdf. The quantile is a non-increasing function, and thus, a reverse mapping $h(F) = \tau$ can be defined, where $F \in [0, 1]$ is the desired quantile.

The density quantile has a well-known connection to the confidence bounds of a normal distribution and its pdf. When extended to mixtures of normal distributions, the confidence bounds may become arbitrarily complex as the number of mixture components increases. An example of a one-dimensional case is illustrated in Figure 5. HDR regions become even more complex in multidimensional complex spaces $\mathbb{C}^D$.

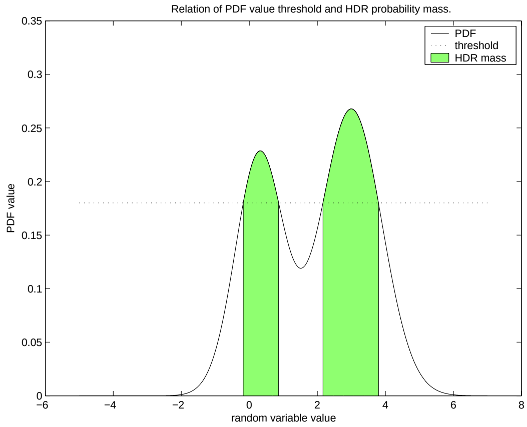


Fig. 5. A highest density region (HDR) of a two-component GMM pdf and the corresponding threshold in one dimension. The confidence region is no longer a simple connected set.

An analytical solution for $F(\tau)$ or $h(F)$ can be very difficult, but a Monte Carlo method can be used to approximate the functions. Computation utilizes random sampling by generating N random points $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$ from a distribution described by $p(\boldsymbol{x})$. The pdf value is computed for every point $p_i = p(\boldsymbol{x}_i)$ and all p_i's are ordered into a sequence $Y = (y_1, \ldots, y_N) = \mathrm{sort}\ \{p_i\}$ in an ascending order. Y represents now a non-decreasing finite series which can be used to estimate $F(\tau)$ and $h(F)$. We have now reduced the original space $\mathbb{C}^D$ into (a discrete approximation of) $\mathbb{R}^+$. A similar method was proposed by Hyndman [25].

Now, the value of $F(\tau)$ can be approximated by

$$i = \arg\max_i \{y_i \mid y_i \leq \tau\} \tag{31}$$

and, with linear interpolation,

$$F(\tau) \approx \begin{cases} 1 & \text{if } \tau < y_1 \\ 0 & \text{if } \tau \geq y_N \\ 1 - \frac{i-1+l(i,\tau)}{N-1} & \text{otherwise} \end{cases} \tag{32}$$

where the linear interpolation term

$$l(i,\tau) = \begin{cases} 0.5 & \text{if } y_{i+1} - y_i = 0 \\ \frac{\tau - y_i}{y_{i+1} - y_i} & \text{otherwise} \end{cases} . \tag{33}$$

The reverse mapping $h(F)$ can be approximated by

$$\tau = h(F) \approx \begin{cases} y_N & \text{if } i = N \\ y_i + ((N-1)(1-F) + 1 - i)(y_{i+1} - y_i) & \text{otherwise} \end{cases} \tag{34}$$

where $i = \lfloor (N-1)(1-F) + 1 \rfloor$.

Now, the estimated pdf value τ can be used as a limit pdf value where observations falling below are directed to the “garbage class”. Hitherto, only Gaussian mixture models have been applied, but the proposed estimation approach applies to any continuous pdf’s that fulfill the gradient condition given in this section.

3.3 Experiments

Bayesian decision making was already demonstrated in the experiments where different estimation algorithms were compared and in the literature there is an enormous number of experimental studies utilizing Bayesian decision making. Here two examples are presented in order to demonstrate the use of confidence.

In the first experiment the use of confidence information was studied in an application where its usability is apparent. In face detection methods where detection of a whole face is based on smaller facial parts, evidence, features are extracted from every spatial location (e.g., [4,5]) and ranked as possible evidence from different classes. The detection of the whole face is based on inspection of spatial connections of different evidence. In Fig. 6(a) an example facial image is shown and the 10 ground truth evidence keypoints are marked [4]. In the training phase Gabor features were extracted from training set images and class conditional pdf's were estimated for each evidence class by the FJ method. In Fig. 6(b) a pdf surface is shown for the evidence class number 7 corresponding to the left (left on the image) nostril. Figs. 6(c) and 6(d) display the confidence regions of density quantiles 0.5 and 0.05 corresponding to 0.5 and 0.95 confidence values respectively. It is clear that the correct evidence location, the left nostril, was already included in the 0.95 confidence region, and thus, the evidence search space for the next processing level was reduced dramatically. Using the confidence information, it is possible to rank evidence from each class and provide at most a requested number of best candidates to a next processing level [4] but also allow the possibility that evidence is not found on the image.

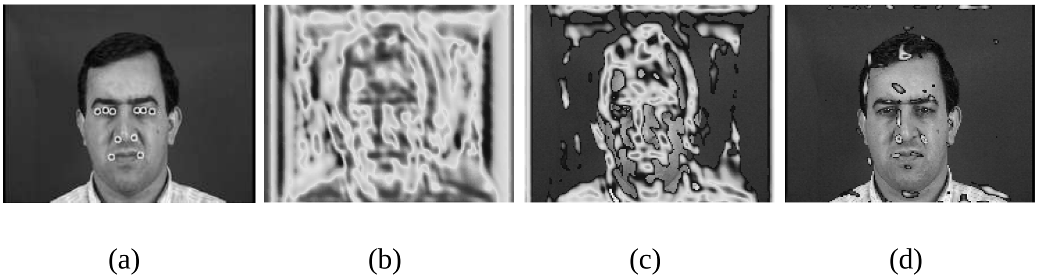


Fig. 6. Example of using density quantile to definine confidence regions : (a) face image and 10 marked evidence classes; (b) pdf value surface for the left (in image) nostril class; (c) confidence threshold 0.5 ($F(\tau) = 0.5$); (d) confidence threshold 0.95 ($F(\tau) = 0.05$).

Lindh et al. [12] have proposed a statistical solution for detecting motor bearing failures from stator current signals of electric motors, but they pointed out the drawback that not only normal condition but also failure condition measurements are needed in their solution. In practice, providing failure condition measurements is often economically impossible and it is difficult to cover all failure situations while normal condition data is produced continuously. This experiment used the same features as proposed in the original study [12], but only normal condition measurements were used to form a pdf and confidence was used to decide between normal and failure conditions. In Fig. 7 a ROC curve is shown for the conducted experiment. From the curve it can be seen how more normal condition measurements were correctly identified (true positives) by decreasing the confidence, but also an increasing number of failure conditions were considered as normal (false positives). The optimal trade-off depends on the application. On the other hand, there was only a minor difference when comparing to the result where also failure condition pdf was used in Bayesian classification (a prioria probabilities were estimated from the training set, which does not correspond to the situation in practice).

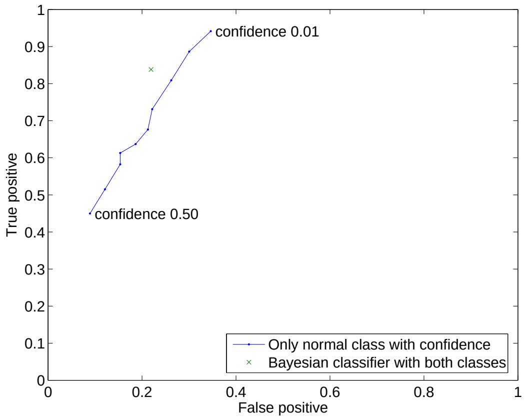


Fig. 7. Receiver operating characteristic (ROC) curve for using confidence value in two class classification of electric motors. A positive test result corresponds to normal operating condition of a motor.

4 Conclusions

Our motivation for this study was to show that statistical methods, especially methods based on class conditional probability density functions of features, are suitable in certain classification tasks. The study was aimed to be useful in feature classification and evidence based object detection, which were not considered in depth in this study but only briefly demonstrated in the experiments.

The estimation of unknown pdf's is a common problem and in this study Gaussian mixture models (GMM) were analyzed as a model, and expectation maximization (EM), Figueiredo-Jain (FJ) and greedy EM (GEM) algorithms were studied for estimating GMM parameters. The study also covered estimation of complex data in multiple dimensions for which case EM and FJ were used. It seems that EM is the most suitable approach if the number of components is known or can reliably be estimated. The optimal number of components can also be estimated using a separate evaluation set and by pruning. However, the two unsupervised methods FJ and greedy EM seemed to be robust and reliable algorithms for GMM parameter estimation. In practice, singularities however occur, and thus, algorithms must tolerate them, for example by enforcing covariance matrices to non-singular form.

Estimated pdf's can be used in classification and the second part of this study concentrated on three different classification problems and what kind of information is needed in the next levels of processing. The first classification problem was solved in a traditional way by utilizing Bayes formula, but the next two problems required a non-Bayesian approach. The use of confidence, as defined in this study, was proposed and a method to establish confidence information was introduced. Finally the use of confidence information was demonstrated experimentally.

This work presents a view of using feature pdf's in classification and ranking of extracted features, observations. The work consisted of a study of how pdf's can be estimated and a study of how the classification should be performed. In future research, the results will be applied to an invariant detection of object evidence by utilizing simple Gabor features and statistical ranking.

References

[1] C. Bishop, Neural Networks for Pattern Recognition, Oxford Univ. Press, 1995.

[2] N. Cristiani, J. Shawe-Taylor, An introduction to Support Vector Machines and other Kernel-based learning methods, Cambridge Univ. Press, 2000.

[3] Y. Freund, R. E. Schapire, A decision theoretic generalization of on-line learning and an application to boosting, J. of Computer and System Sciences 55 (1) (1997) 119–139.

[4] M. Hamouz, J. Kittler, J.-K. Kamarainen, P. Paalanen, H. Kälviäinen, Affine-invariant face detection and localization using GMM-based feature detector and enhanced appearance model, in: Proc. of the 6th Int. Conf. on Automatic Face and Gesture Recognition, Seoul, Korea, 2004, pp. 67–72.

[5] M. Hamouz, J. Kittler, J.-K. Kamarainen, P. Paalanen, H. Kälviäinen, J. Matas, Feature-based affine-invariant detection and localization of faces, IEEE Transactions on Pattern Analysis and Machine Intelligence(to be published).

[6] M. Schlesinger, V. Hlavac, Ten Lectures on Statistical and Structural Pattern Recognition, Computational Imaging and Vision, Kluwer Academic Publishers, 2002.

[7] V. Kyrki, J.-K. Kamarainen, H. Kälviäinen, Simple Gabor feature space for invariant object recognition, Pattern Recognition Letters 25 (3) (2003) 311–318.

[8] GMMBAYES Matlab Toolbox, http://www.it.lut.fi/project/gmmbayes.

[9] Y. Tong, The Multivariate Normal Distribution, Springer Series in Statistics, Springer-Verlag, 1990.

[10] N. Goodman, Statistical analysis based on a certain multivariate complex Gaussian distribution (an introduction), The Annals of Mathematical Statistics 34 (1) (1963) 152–177.

[11] J. Ilonen, J.-K. Kamarainen, H. Kälviäinen, O. Anttalainen, Automatic detection and recognition of hazardous chemical agents, in: Proc. of the 14th Int. Conf. on Digital Signal Processing, Vol. 2, 2002, pp. 1345–1348.

[12] T. Lindh, J. Ahola, J.-K. Kamarainen, V. Kyrki, J. Partanen, Bearing damage detection based on statistical discrimination of stator current, in: Proc. of the 4th IEEE Int. Symp. on Diagnostics for Electric Machines, Power Electronics and Drives, Atlanta, Georgia, USA, 2003, pp. 177–181.

[13] B. Everitt, D. Hand, Finite Mixture Distributions, Monographs on Applied Probability and Statistics, Chapman and Hall, 1981.

[14] J.-K. Kämäräinen, Feature extraction using Gabor filters, Ph.D. thesis, Lappeenranta University of Technology (2003).

[15] J. Bilmes, A gentle tutorial on the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models (1997).
URL `citeseer.nj.nec.com/bilmes98gentle.html`

[16] R. Duda, P. Hart, D. Stork, Pattern Classification, 2nd Edition, John Wiley & Sons, Inc., 2001.

[17] M. Figueiredo, A. Jain, Unsupervised learning of finite mixture models, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (3) (2002) 381–396.

[18] S. Theodoridis, K. Koutroumbas, Pattern Recognition, Academic Press, 1999.

[19] J. J. Verbeek, N. Vlassis, B. Kröse, Efficient greedy learning of Gaussian mixture models, Neural Computation 5 (2) (2003) 469–485.

[20] I. Nagy, P. Nedoma, M. Karny, Factorized EM algorithm for mixture estimation, in: Proc. of the Int. Conf. on Artificial Neural Networks and Genetic Algorithms, Wien, 2001.

[21] T. Jaaskelainen, R. Silvennoinen, J. Hiltunen, J. P. S. Parkkinen, Classification of the reflectance spectra of pine, spruce, and birch, Applied Optics 33 (12) (1994) 2356–2362.

[22] J.-K. Kamarainen, V. Kyrki, J. Ilonen, H. Kälviäinen, Improving similarity measures of histograms using smoothing projections, Pattern Recogn. Lett. 24 (12) (2003) 2009–2019.

[23] P. Paalanen, J.-K. Kamarainen, J. Ilonen, H. Kälviäinen, Feature representation and discrimination based on gaussian mixture model probability densities - practices and algorithms, Research report 95, Dept. of Information Technology, Lappeenranta University of Technology (2005).

[24] L. Breiman, J. Friedman, R. Olshen, C. Stone, Classification and Regression Trees, Wadsworth International Group, Belmont, California, 1984, pp. 43–49.

[25] R. J. Hyndman, Computing and graphing highest density regions, The American Statistician 50 (2) (1996) 120–126.

[26] P. N. Somerville, Numerical computation of multivariate normal and multivariate-t probabilities over convex regions, J. of Computational and Graphical Statistics 7 (4) (1998) 529–544.

[27] P. N. Somerville, Numerical computation of multivariate normal and multivariate t probabilities over ellipsoidal regions, Journal of Statistical Software.

[28] G. J. Tee, Surface area and capacity of ellipsoids in n dimensions, Tech. rep., Department of Mathematics, University of Auckland, use Tee:2005 instead. (2004).