Pattern Recognition 47 (2014) 2582–2595

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Simultaneous feature selection and Gaussian mixture model estimation for supervised classification problems

Jens Kersten [1]

German Aerospace Center, German Remote Sensing Data Center, Muenchner Strasse 20, 82234 Wessling, Germany

ARTICLE INFO

Article history:
Received 27 November 2012
Received in revised form
14 February 2014
Accepted 26 February 2014
Available online 7 March 2014

Keywords:
Gaussian mixture models
Clustering
Feature selection
Feature saliency
Expectation maximization
Supervised learning
Remote sensing

ABSTRACT

A new expectation maximization (EM) algorithm for time-critical supervised classification tasks in remote sensing is proposed. Compared to standard EM and other approaches, it has the following advantages: (1) No knowledge about the class distributions is needed. (2) The number of components is estimated. (3) It does not require careful initialization. (4) Singular estimates are avoided due to the ability of pruning components. (5) The best discriminating features are identified simultaneously. (6) The features are identified by incorporating Mahalanobis distances.

Three experiments are carried out in order to demonstrate the relevance of the method. The main findings are the following: (1) Feature selection is very important in terms of prediction quality of models. (2) In the experiments the proposed method estimates better models than other state-of-the-art methods. (3) The incorporation of Mahalanobis distances is very valuable for the identification of relevant features. (4) The proposed method is more robust than the compared methods. (5) In case of complex data distributions the new approach is able to provide better results.

1. Introduction

The central limit theorem states that the mean of a sufficiently large number of independent and identically distributed (i.i.d.) random variables is approximately normally distributed. Therefore, the use of Gaussian models for the representation of uni- and multivariate data is justified. In many applications (for example remote sensing) a data distribution may be the sum of more than one normal distributions. Gaussian mixture models (GMMs) are able to approximate arbitrary data distributions and therefore are suitable for the parameterization of multivariate distributions. This paper deals with parameter estimation for Gaussian mixture models in the context of time-critical, supervised classification of optical aerial imagery. In this as well as in other pattern recognition applications, GMMs are often used to either represent thematic classes based on training data (e.g. supervised learning for maximum-likelihood (ML) classification [1]) or to identify and model spectral classes based on a complete dataset, e.g. a satellite image, without any prior knowledge (unsupervised learning, i.e., clustering). Due to the fact that semantic classes usually consist of more than one spectral class, the application of statistical clustering can be valuable in both cases. Furthermore, semisupervised approaches [2,3] make use of both labeled and unlabeled data, in order to avoid biased modeling.

A widely used approach to fit GMMs to data is the expectation maximization algorithm (EM) [4] or variants of it [5]. EM is a general approach of finding the ML estimate of mixture model parameters in applications dealing with incomplete data or with an analytically intractable optimization of the likelihood function. The latter application, which includes clustering, is more common in the computational pattern recognition community [6].

Besides the fact that EM is guaranteed to converge [7], the estimated local solution highly depends on parameter initialization including the usually predefined number of Gaussian components. By applying EM approaches in a supervised manner, for example for the classification of aerial imagery, a further problem arises: the selection of relevant features. This reduction of dimensionality is known to improve the predictive performance, saves computational costs and leads to simpler and more intelligible models. Especially when the initial number of features, i.e., the dimensionality of feature space, is high and the number of training samples may therefore be insufficient, the reduction of dimensionality is a very important task in statistical modeling. The following problems can arise and cause the need for dimensionality reduction. It is likely that many of the features are partially or even completely redundant or irrelevant to the underlying problem [8]. Another point is given with the usually limited number of training samples in supervised learning problems. Due to the so-called *curse of dimensions* [9], the target classes in high dimensional feature spaces are likely to be underrepresented.

E-mail address: jens.kersten@uni-weimar.de
[1] Permanent address: Bauhaus-Universität Weimar, Computer Vision, Bauhausstr. 11, 99423 Weimar, Germany.

http://dx.doi.org/10.1016/j.patcog.2014.02.015
0031-3203

Furthermore, feature reduction shall be done under consideration of multidimensional subspaces instead of evaluating single features, since the combination of two single weak features can lead to significant gain of the model quality [10].

For the classification of optical aerial imagery – especially when methods are embedded in operational and time-critical services – it is essential to automate parameter estimation and image processing procedures. In order to reduce the user interaction to the – itself difficult – task of providing training data representing the desired thematic classes, an appropriate automated feature selection and model detection method is required. The heuristic sequential floating forward search (SFFS) proposed by Pudil et al. [11] is widely used and has become the probably most preferred choice in many applications [12] since it finds near optimal solutions in tractable runtime [13–15]. Hence, in works of Kersten et al. [16] and Kersten [17], the SFFS approach was combined with a split-based EM [18] in order to additionally estimate the number of components (in the following denoted as SFFS-EM).

The EM approach proposed by Law et al. [19] elegantly tackles the problem of simultaneous feature selection, the estimation of GMM parameters and the number of clusters for unsupervised clustering tasks. Since the number of clusters and the best feature subset are inter-related [20], this simultaneous method seems to be very appropriate. The characteristics of the approach are as follows: (1) It requires no prior knowledge about the data distribution. (2) The number of components is estimated. (3) The algorithm does not require careful initialization. (4) Singular estimates are avoided due to the ability of pruning components. Furthermore, the use of a statistical method is likely to be more efficient regarding computational time than that of heuristic methods. These characteristics suggest a low amount of user interaction as well as a high robustness and therefore this method seems to be very valuable and relevant in context of a general framework for supervised classification of optical aerial imagery.

In this paper the approach of Law et al. [19], designed for unsupervised learning, is extended to a multiclass EM-algorithm for supervised classification tasks (Multiclass Feature Selection EM, MCFS-EM). It estimates a GMM and the number of components for each thematic class. The features that best discriminate between the classes are identified by incorporating the separability of the classes in the feature space domain via Mahalanobis distances. Hence, the best fitting model which simultaneously allows the best separation of the thematic classes is obtained. Besides the description of the algorithm, the paper focuses on its applicability in the context of supervised aerial imagery classification. The comparative results of three experiments, carried out in order to examine the characteristics of the method concerning its ability of modeling data, selecting the relevant features and its relevance in classification applications, are presented.

The rest of the paper is organized as follows. In Section 2 related work in the field of dimension reduction of feature spaces as well as simultaneous feature selection and GMM estimation using EM is reviewed briefly. In Section 3 existing methods of feature selection, model estimation and both are summarized. In Section 4 the proposed statistical MCFS-EM algorithm is described in detail. In Section 5 comparative experiments using six different datasets and different methods of feature selection and model estimation are described in detail. The experimental results are then discussed in Section 6, followed by concluding remarks drawn in Section 7.

2. Related work

2.1. Reduction of feature space dimensions

Methods of dimension reduction can be grouped into *feature extraction* and *feature selection* approaches [14]. A well known method for feature extraction is the principal component analysis [21]. Drawbacks of this and other methods of feature extraction (i.e., feature transformation) in context of supervised classification lie in the independence of the used classification method, the lost physical interpretation of the transformed features as well as the fact that all initial features have to be computed in order to execute the transformation (computational effort). An overview of feature selection methods is for example given by Dash and Liu [8]. The methods can be further divided into *filter* and *wrapper* methods [22]. Wrapper methods use the classification method itself to evaluate a certain model configuration. Usually x-fold cross-validation is used here leading to very high computational costs. Filter methods use defined measurements like inter- and intra-cluster similarities and are therefore usually independent of the used classifier.

Choosing a subset of features, i.e., identifying the redundant and irrelevant features with respect to the underlying classification problem, can be done using *complete*, *heuristic* and *random* approaches. In a complete search all possible $\binom{d}{n}$ feature combinations are evaluated, where d denotes the number of all initial features and n the number of features of the identified subspace. Leading to very high computational costs even for moderate dimensionalities, this approach ensures the identification of the optimal feature subspace. The branch-and-bound approach [23] enables a faster identification of optimal feature subsets. Due to the monotonicity property of the criterion function (i.e., the performance of a feature subset should improve whenever a feature is added), a complete search can be circumvented here. However, in case of insufficient large sets of training data the quality of selected feature subsets tends to be poor [14]. In order to reduce the computational costs of exhaustive approaches, various heuristics were proposed leading to suboptimal but good results. Based on different comparative studies [13–15], nearly optimal solutions can be found in a tractable runtime using the sequential forward search (SFS) [24] and the sequential forward floating search (SFFS) [11]. SFS starts with an empty set of features and iteratively adds the feature, which causes the highest gain of a defined evaluation function. This approach suffers from the so-called nesting effect, since a selected feature cannot be removed from the feature subset anymore. This effect is avoided in SFFS by optional removing of features after adding a single feature. A feature is removed if the evaluation criterion shows better results with respect to the number of dimensions.

Genetic algorithms (GAs) (see for example [25]) are based on the natural process of evolution and fall into the group of random approaches. In each iteration a defined number of solutions are generated by applying different genetic operators, like recombination and mutation, as a stochastic process. As pointed out in [13], GA is very useful for problems with a dimensionality of $d > 50$. However, the main drawback of genetic approaches lies in the lack of rules for the specification of algorithm-parameters. This forces an examination of several distinct parameter sets of GA at the same time. But even in that case better results than using SFFS are not guaranteed. This may be the reason for the comparably good results of SFFS and GA in [15].

2.2. Simultaneous model estimation and feature selection using EM

Gaussian mixture models can represent arbitrarily complex probability distributions. However, using EM for parameter estimation one has to deal with some drawbacks and difficulties, for example concerning the locality of the method leading to a sensitivity to initialization. A further problem is given with overfitting, which is related to the chosen or estimated number of clusters. Since different initializations may converge to different solutions using EM [5] a number of methods have been proposed to tackle this initialization problem. McLachlan and Peel [26] proposed to run EM multiple times with different randomly chosen parameters and select the best solution as the result. Random swap EM (RSEM) [27] searches the

solution space in a more efficient manner, since only a part of the model, i.e., a single component, is changed randomly after convergence. In contrast, Zhang [28] proposed a model modification using one split and one merge operation after convergence. However, the problem of estimating the number of components remains unresolved in these approaches. Figueireido and Jain [29] proposed an approach which addresses both aforementioned drawbacks of EM simultaneously. Careful initialization is not required here, since a large number of initial components can be introduced and subsequently pruned. Whereas the latter method starts with a large number of components which is subsequently reduced, Ververidis and Kotropoulos [18] proposed a split-based approach which starts with a unimodal model which is subsequently split using a statistical test based on Mahalanobis distances.

In [16,17] a filter approach for feature selection and supervised learning of mixture models was proposed. Here, the heuristic SFFS approach is combined with the split-based EM approach proposed by Ververidis and Kotropoulos [18]. Each cluster configuration is evaluated using the Mahalanobis distances between the components from different classes.

Due to the fact that EM introduces hidden random variables and estimates unknown but fixed model parameters, this approach can be understood as partially Bayesian. A (fully) Bayesian approach for simultaneous feature selection and model estimation is proposed by Constantinopoulos et al. [30]. Here, the so-called *feature saliencies*, introduced by Pudil et al. [31], are used as weights indicating the usefulness of features with respect to the underlying problem. Furthermore, the features are considered to be independent given a component. A variational approach, which suggests the maximization of a lower bound, is employed for parameter estimation. A further Bayesian approach proposed by Li et al. [32] uses a localized Bayesian inference method of Gaussian mixtures dealing with *local* feature saliencies, i.e., different clusters can be represented in different subspaces of the initial feature space. One drawback of these approaches is the introduction of several variational parameters which have to be initialized in a proper way and which have to be estimated additionally to the desired model parameters. Slightly different initializations may converge to different solutions, leading to the need for multiple restarts of the method. Hence, additional expert knowledge or experience concerning the initialization of the variational parameters is required.

3. Methods of feature selection, clustering and both of these tasks

In this section, existing methods that are used for comparison to the proposed method in Section 5 are described. The focus of each method, i.e., feature selection, model estimation or both, is noted in the title of each subsection.

3.1. Split-based EM clustering [18]

This clustering method estimates GMMs including the number of clusters and their parameters. It is based on the well known standard EM-algorithm [4]. The mixture model parameters, i.e., the weight α_j, mean vector $\boldsymbol{\mu}_j$ and covariance matrix $\boldsymbol{\Sigma}_j$ of a component $j=1,\ldots,K$, are estimated as follows:

E-step: Computation of probabilities of the membership $p_j^r(y_i)$ to a component j for all observed y_i (with $i=1,\ldots,N$) in iteration r:

$$p_j^r(y_i)=\frac{\alpha_j^{r-1}p(y_i|\mu_j^{r-1},\Sigma_j^{r-1})}{\sum_{j=1}^{K}\alpha_j^{r-1}p(y_i|\mu_j^{r-1},\Sigma_j^{r-1})}. \tag{1}$$

M-step: Re-estimation of model parameters using the results of E-step:

$$\alpha_j^r=\frac{1}{N}\sum_{i=1}^{N}p_j^r(y_i) \tag{2}$$

$$\mu_j^r=\frac{\sum_{i=1}^{N}p_j^r(y_i)y_i}{\sum_{i=1}^{N}p_j^r(y_i)} \tag{3}$$

$$\Sigma_j^r=\frac{\sum_{i=1}^{N}p_j^r(y_i)(y_i-\mu_j^r)(y_i-\mu_j^r)^T}{\sum_{i=1}^{N}p_j^r(y_i)}. \tag{4}$$

These two steps are performed alternating till the expectation of the log-likelihood function of the model reaches a local maximum. In order to additionally estimate the number of clusters for each class c including their parameters, the following algorithm is used (see Fig. 1):

1. Assign each vector y to a component j.
2. Identification of a single component to split due to a multivariate test using the distribution of the Mahalanobis distances to the according center of component.
3. Application of a multivariate kurtosis test with respect to the identified component in order to determine the type of initialization of the two new components.
4. Initialization of the new components and application of the standard EM-algorithm.

For more detailed information concerning the derivation of the complete algorithm, compare [11,16–18].

3.2. Random-swap EM clustering [27]

The idea of RSEM is to run standard EM multiple times and to select the best solution, in terms of log-likelihood, as the result. The algorithm is presented in Fig. 2. The initial model parameters are estimated using k-means. t random swap iterations are then performed. Instead of initializing the complete model after each

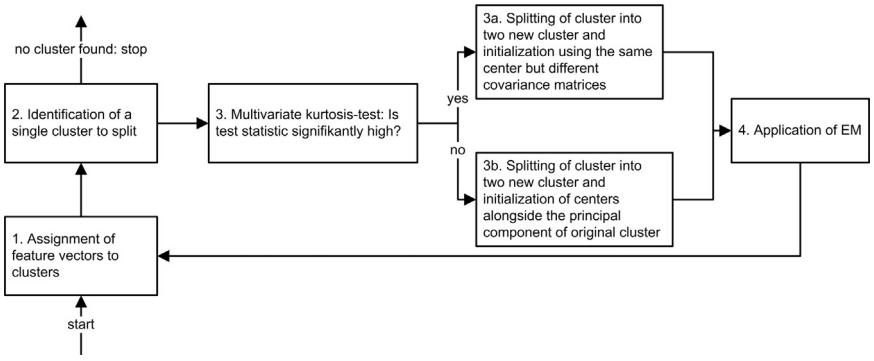

Fig. 1. Workflow of split-based clustering following [18].

Input: Training data $\mathbf{y} = \{y_1, \ldots, y_N\}$

Output: Mixture model parameters $\Theta = \{\{\alpha\}, \{\mu\}, \{\Sigma\}\}$ and log-likelihood $L(\Theta)$

Initialization: Estimate initial parameters $[\Theta_0,\ L(\Theta_0)]$ using k-means clustering

for RS-iteration = 1 to t **do**

$r = U(1, K)$, remove rth component

$p = U(1, N)$, add component at pth position

Normalize weights α to sum to one

New parameters $\Theta^s = \{\{\alpha^s\}, \{\mu^s\}, \{\Sigma^s\}\}$

$\left[\Theta^{st},\ L(\Theta)^{st}\right] \leftarrow EM(\mathbf{y}, \Theta^s)$

if $L(\Theta)^{st} > L(\Theta)$ **then**

$\Theta = \Theta^{st}$

$L(\Theta) = L(\Theta^{st})$

end if

end for

return $\Theta, L(\Theta)$

Fig. 2. RSEM algorithm following [27].

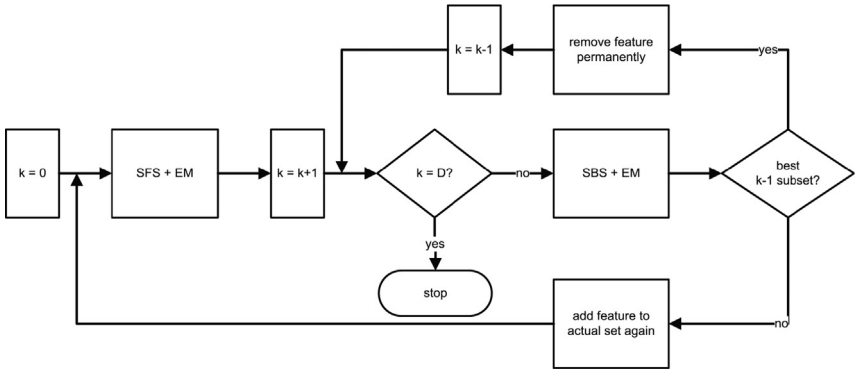

Fig. 3. Workflow of SFFS-EM following [35].

EM run, only one part, i.e., one component, of the model is randomly changed. For the removal one component is selected randomly from uniform distribution. The mean of the new component is initialized by randomly selecting a training data point. If an estimated solution is better, in terms of log-likelihood, than the best solution so far it is used as a starting point for the subsequent iteration.

If the EM estimation of a swap t produces no better solution than the starting model, this solution is discarded. RSEM is able to diminish the negative effect of bad initializations. On the other hand, the number of initial clusters has to be defined manually and remains fixed. Instead of trying out all possible swap pairs to improve the solution, the number of swaps is restricted to a user defined number t for reasons of practicability. Since RSEM is used for supervised classification problems here, it is applied on each class subsequently. The number of swaps was set to $t=30$ in all experiments.

3.3. Feature selection: recursive feature elimination using Support Vector Machines [33]

The recursive feature elimination (RFE) algorithm utilizes Support Vector Machine (SVM) methods to generate a ranking of features. It was proposed by Guyon et al. [33] for the selection of small, critically important subsets of genes from broad patterns of gene expression data. The basic idea of this wrapper method is to eliminate redundant genes, i.e., features, according a criterion related to their support to the discrimination function of the SVM. The algorithm can be summarized as follows:

1. Train an SVM using the training dataset.
2. Rank the features using the weights of the trained classifier.
3. Eliminate the feature with the smallest weight.
4. Iteration of steps 1–3 using the selected features.

For more detailed insights concerning the derivation of this algorithm see [33].

3.4. SFFS-EM clustering and feature selection [17]

The sequential floating forward selection (SFFS) uses the sequential forward as well as backward selections (SFS and SBS) in order to avoid the aforementioned nesting effect (see Section 2). SFS starts with an empty set of features and identifies each iteration of the (previously unselected) feature which affects the highest gain of the evaluation function in combination with the actual set of selected features. As a top-down pendant to SFS, the SBS approach [34] starts with a complete set of features and iteratively reduces the set of selected features as long as a gain of the evaluation function w.r.t. the number of features is obtained. In Fig. 3 the workflow of the SFFS algorithm [11] is shown.

SFFS-EM is a combination of SFFS and standard EM. It starts with a single SFS step in which the split-based EM method (see Section 3.1) is used for model estimation. The nesting effect is avoided due to the backtracking-phase in which the removing of one or more features from the set of selected features is allowed. In each SFS and SBS step the Split-based EM-algorithm is applied. SFFS-EM stops when the desired number of features D is reached.

3.5. Ranking-based feature selection [36]

This simple method of feature ranking was developed at the German Aerospace Center (DLR) for the estimation of parking space occupancy based on aerial imagery. Due to the fact that parking areas usually consist of concrete, i.e., one spectral class, a single Gaussian is used to model this class. Separation of free space (concrete) and occupied space (e.g. parking cars) can be carried out by applying a threshold on the computed likelihoods. In order to circumvent the manual definition of a threshold, both classes were modeled using a single Gaussian in the experiments. Hence, classification is done using the ML criterion. The ranking of features concerning a certain classification problem is done as follows:

1. Compute for each feature the normalized cross-correlation with respect to class x (occupied space) and y (free space) according to
$$corr = \frac{\sum_{i=1}^{N}(x_i \cdot y_i)}{\sum_{i=1}^{N} x_i^2 \cdot \sum_{i=1}^{N} y_i^2}, \tag{5}$$
where N is the number of bins of the histograms x and y.
2. Compute a coverage measure, i.e., a measure how good a single Gaussian can represent the data, for class y in each dimension by
$$cov = \frac{\sum_{i=1}^{N}(G_i^y \cdot y_i)}{h^2 \cdot \sum_{i=1}^{N} y_i}, \tag{6}$$
with the histogram G_i^y of a single Gaussian representing class y and the height h of this Gaussian.
3. Compute a quality measurement q for each feature according to
$$q = (1.0 - corr) \cdot cov. \tag{7}$$
4. Sort all features according to the quality measurement q and select the first n features.

4. The proposed multiclass feature selection EM-algorithm (MCFS-EM)

In generative modeling using Gaussian mixtures each datapoint y^c of a class c is assumed to be generated by a probabilistic mixture-based model with K_c components, defined by

$$p(\mathbf{y}^c|c) = \sum_{j=1}^{K_c} \alpha_j^c p(\mathbf{y}^c|\theta_j^c), \tag{8}$$

where α^c are the mixing coefficients for class c satisfying $\forall_j : \alpha_j^c > 0$, $\sum_j \alpha_j^c = 1$. θ_j^c represents the set of parameters of the jth component of class c and $\mathbf{y}^c$ is a set of N_c datapoints. Introducing the assumption of $l = 1,\ldots,D$ conditionally independent features given the hidden component labels, the mixture density for each class c is

$$p(\mathbf{y}^c|\theta^c) = \sum_{j=1}^{K_c} \alpha_j^c \prod_{l=1}^{D} p(y_l^c|\theta_{jl}^c). \tag{9}$$

This assumption enables the derivation of a closed form solution of the problem, i.e., to utilize the power of the EM algorithm [37]. In the particular case of Gaussian mixtures, the conditional independence assumption is equivalent to adopting diagonal covariance matrices, which is a common choice for high-dimensional data, such as in naive Bayes classifiers and latent class models, as well as in the emission densities of continuous hidden Markov models [37].

Following the definition of feature irrelevancy in [31,38] and [39], the lth feature is irrelevant to a certain problem if its distribution is independent of the class labels, i.e., if it follows a common density denoted by $q(y_l^c|\lambda_l)$ [19]. Given a set of binary parameters $\Phi = (\phi_1, \ldots, \phi_D)$, where $\phi_l = 1$ if the lth feature is relevant and $\phi_l = 0$ otherwise, the mixture density (9) can be rewritten as

$$p(\mathbf{y}^c|\Phi, \{\alpha_j^c\}, \{\phi_{jl}^c\}, \{\lambda_l^c\}) = \sum_{j=1}^{K_c} \alpha_j^c \prod_{l=1}^{D} [p(y_l^c|\theta_{jl}^c)]^{\phi_l} [q(y_l^c|\lambda_l^c)]^{(1-\phi_l^c)}. \tag{10}$$

In [19] the saliencies of features are treated as missing variables and are defined as $\rho_l = P(\phi_l = 1)$. Hence, ρ_l describes the probability that the lth feature is relevant, leading to the expression

$$p(\mathbf{y}^c|\phi) = \sum_{j=1}^{K_c} \alpha_j^c \prod_{l=1}^{D} [\rho_l p(y_l^c|\theta_{jl}^c) + (1-\rho_l) q(y_l^c|\lambda_l^c)], \tag{11}$$

with the set of all unknown model parameters $\phi = \{\{\alpha_j^c\}, \{\phi_{jl}^c\}, \{\lambda_l^c\}, \{\rho_l\}\}$. By introducing the membership z of each datapoint i to a component as well as Φ as hidden variables [19], the following multiclass EM-algorithm for parameter estimation can be derived.

E-step: Computation of the following quantities for each class c:

$$a_{ijl}^c = P(\phi_l^c = 1, y_{il}^c | z_i = j) = \rho_l \cdot p(y_{il}^c|\theta_{jl}^c) \tag{12}$$

$$b_{ijl}^c = P(\phi_l^c = 0, y_{il}^c | z_i = j) = (1-\rho_l) \cdot p(y_{il}^c|\theta_{jl}^c) \tag{13}$$

$$e_{ijl}^c = P(y_{il}^c | z_i = j) = a_{ijl}^c + b_{ijl}^c \tag{14}$$

$$w_{ij}^c = P(z_i = j|\mathbf{y}_i^c) = \frac{\alpha_j^c \prod_l e_{ijl}^c}{\sum_j \alpha_j^c \prod_l e_{ijl}^c} \tag{15}$$

$$u_{ijl}^c = P(\phi_l^c = 1, z_i = j|\mathbf{y}_i^c) = \frac{a_{ijl}^c}{e_{ijl}^c} w_{ij}^c \tag{16}$$

$$v_{ijl}^c = P(\phi_l^c = 0, z_i = j|\mathbf{y}_i^c) = w_{ij}^c - u_{ijl}^c \tag{17}$$

M-step: Re-estimation of model parameters:

$$\hat{\alpha}_j^c = \frac{\max\left(\sum_i w_{ij}^c - \frac{RD}{2}, 0\right)}{\sum_c \sum_j \max\left(\sum_i w_{ij}^c - \frac{RD}{2}, 0\right)} \tag{18}$$

$$\text{Mean in } \hat{\theta}_{jl}^c = \frac{\sum_i u_{ijl}^c y_{il}^c}{\sum_i u_{ijl}^c} \tag{19}$$

$$\text{var in } \hat{\theta}_{jl}^c = \frac{\sum_i u_{ijl}^c (y_{il}^c - (\text{Mean in } \hat{\theta}_{jl}^c))^2}{\sum_i u_{ijl}^c} \tag{20}$$

$$\text{Mean in } \hat{\lambda}_l^c = \frac{\sum_i (\sum_j v_{ijl}^c) y_{il}^c}{\sum_{ij} v_{ijl}^c} \tag{21}$$

$$\text{var in } \hat{\lambda}_l^c = \frac{\sum_i (\sum_j v_{ijl}^c)(y_{il}^c - (\text{Mean in } \hat{\lambda}_l^c))^2}{\sum_{ij} v_{ijl}^c} \tag{22}$$

$$\hat{\rho}_l = \frac{\sum_c \max\left(\sum_{ij} u_{ijl}^c - \frac{K_c R}{2}, 0\right)}{\sum_c \left[\max\left(\sum_{ij} u_{ijl}^c - \frac{K_c R}{2}, 0\right) + \max\left(\sum_{ij} v_{ijl}^c - \frac{S}{2}, 0\right)\right]} \tag{23}$$

where R and S are the number of parameters in θ_{jl}^c and λ_l^c, respectively. Hence, if p and q are modeled using univariate Gaussians it follows $R = S = 2$. The variable u_{ijl}^c expresses the influence of the ith datapoint to the jth component of class c using feature l. u_{ijl}^c is therefore used as weight for the estimation of

the mean and variance of θ_{jl}^c. The same interpretation is valid for $\sum_j v_{ijl}^c$ and λ_l^c. The sum $\sum_{ij} u_{ijl}^c$ expresses how likely θ_l equals one, which leads the proportionality to ρ_l.

As suggested in [29,19], a minimum message length (MML) model criterion can be used to formulate the minimization problem, leading to the following criterion for multiclass problems:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\theta}\left\{\sum_c\left(-\log[p(\mathbf{y}_c|\theta_c)]+\frac{1}{2}(K_c+D+K_cDR+DS)\cdot\log[N_c]\right.\right.$$
$$\left.\left.+\frac{R}{2}\sum_{j=1}^{K_c}\sum_{l=1}^{D}\log[\alpha_j^c\rho_l]\right)+\frac{S}{2}\sum_{l=1}^{D}\log[1-\rho_l]\right\}. \quad (24)$$

The benefit of this criterion lies in the ability of pruning components and features when α_j^c or ρ_l goes to zero (term (18) and (23)). Hence, no knowledge about the number of components K_c for each class c is required and the algorithm can be initialized with large values for all K_c circumventing the need for careful initialization. Furthermore, a component-wise EM-algorithm as proposed in [40] can be adopted to this n-class problem. The full algorithm is summarized in Fig. 4.

Training data for each thematic class, i.e., feature vectors, as well as the initial maximum (e.g. 30) and minimum (e.g. 1) number of components for each class has to be provided. The algorithm is initialized using standard k-means clustering as well as by computing the common distributions for each class. Furthermore, the saliencies are initialized by $\rho_l = 0.9$.

The number of components for each class is successively reduced as long as $K^c \geq K_{min}$. The component-wise algorithm minimizes the MML-criterion by estimating the parameters of each component j of each class c individually, keeping all other parameters fixed. After a local minimum is reached, the model parameters as well as the corresponding message length are recorded. The new iteration is started after removing the component with the smallest global weight α.

The described EM-algorithm estimates the best fitting model with respect to the given training data. In order to identify the features that best discriminate between the components in unsupervised clustering, Law et al. [19] introduced an additional post-processing step, which involves the maximization of the sum of logarithms of the posterior probabilities of the data. Here, all parameters estimated by the EM-algorithm except ρ_l are kept fixed. In contrast, the following simple modification of term (23) is proposed to achieve simultaneous estimation of all parameters by finding a model that additionally incorporates the separability between the classes

$$\overline{\rho}_l = \frac{\overline{m}_l + \hat{\rho}_l}{2}, \quad (25)$$

where $\overline{m}_l$ is a vector of all normalized mean inter-class Mahalanobis distances in each dimension l. The impact of this weighting criterion on feature selection and model detection is demonstrated in the following example. As shown in Fig. 5, two classes are modeled in a two-dimensional feature space using a mixture of three Gaussian functions. Furthermore, eight noisy features, i.e., sampled from a standard normal density, are added to the data leading to a ten-dimensional feature space ($l = 0, \ldots, 9$). The algorithm should identify the two relevant features. In the first run, the proposed multiclass EM-algorithm without the modified computation of saliencies is carried out (see Fig. 6, left). The estimated saliencies are $\rho = (0.95, 1.0, 0.64, 1.0, 0.03, 0.11, 1.0, 1.0, 0.62, 1.0)$ which means that the two relevant features could not be identified clearly. Since the training data is sampled from single Gaussians in the last eight dimensions, the result clarifies that this algorithm finds the best fitting model instead of finding the model that best discriminates between the classes. The application of the proposed modified calculation of feature saliencies (see Fig. 6, right) according to Eq. (25) leads to clearly better results regarding the feature saliencies. The following saliencies are obtained $\rho = (0.82, 0.86, 0.45, 0.75, 0.32, 0.50, 0.54, 0.60, 0.78, 0.24)$. This result allows to

Input: Training data $\mathbf{y}^c = \{y_1, \ldots, y_{N_c}\}$,
initial and minimum number of components K_{max}^c, K_{min}^c

Output: Number of components K^c for each class, mixture model parameters $\left\{\theta_{jl}^c\right\}$, $\{\alpha_j^c\}$, parameters of common distribution $\{\lambda_l\}$, feature saliencies $\{\rho_l\}$

Initialization: Estimate initial parameters for K_{max}^c components using k-means clustering
Set the common distributions using all data for each class
Set feature saliencies, e.g. $\rho_l = 0.9$

while all $K^c > K_{min}$ **do**

while local minimum is not reached do

loop over all classes c

loop over all components of class c

Perform E-step according to equations (12) to (17)
Perform M-step according to equations (18) to (23) and (25)
If $\alpha_j^c = 0$, prune corresponding component j
If $\rho_l = 1$, prune $q(y_l|\lambda_l)$
If $\rho_l = 0$, prune $p(y_l|\theta_{jl})$ for all classes c

end loop

end loop

end while

Record model parameters and corresponding message length according to equation (24)
Remove the component with the smalles global weight

end while

Return model parameters according to the smallest global message length

Fig. 4. The component-wise MCFS-EM-algorithm for feature selection and model detection.

differentiate between relevant and irrelevant features, for example by applying a threshold. The weighting of $\hat{\rho}_l$ using the mean of the Mahalanobis distances in each dimension leads to an even better pruning behavior for this example. All irrelevant features are pruned here and the two relevant features are selected automatically avoiding the application of a threshold. However, in further experiments this approach has shown a crude feature pruning behavior for real datasets and therefore usually leads to poorer classification results.

One drawback of the proposed algorithm may be the assumed independency of features leading to diagonal covariance matrices. Hence, in the experiments described in Section 5, a simple post-processing step is introduced and evaluated. Here, a standard EM-algorithm is applied on the result of the component-wise multiclass algorithm.

5. Experimental results

Three experiments were carried out in order to demonstrate the relevance of the proposed method using six different datasets. The first experiment focuses on the quality of model estimation using all introduced features. In the second experiment the quality of the selection of relevant features using different approaches is addressed. In the third experiment the simultaneous feature selection and model detection for different classification problems is examined. In the following the datasets and quality measures used in the experiments are described. Subsequently, the experiments and their results are presented.

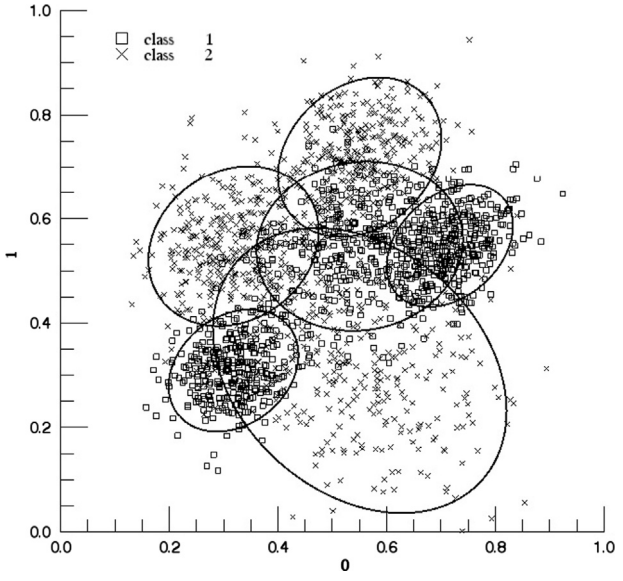

Fig. 5. Sampled distribution of the artificial dataset in the two relevant dimensions.

5.1. Datasets

The six datasets used in the experiments had to be normalized due to the usage of Mahalanobis distances in term (25). Furthermore, SFFS-EM (see Section 3.4) involves the comparison of mean Mahalanobis distances between components with respect to feature spaces with an equal number of dimensions but different features. Normalization of the data y in each dimension was carried out according to [41]

$$\tilde{y} = \frac{(y-\mu)/3\sigma + 1}{2}, \tag{26}$$

with the global mean μ and standard deviation σ of the complete dataset, in order to ensure comparability of the aforementioned distances. The following datasets were used.

5.1.1. Synthetic dataset

This dataset represents two classes, each of them consisting of 1000 datapoints sampled from three Gaussians $\mathcal{N}(\mathbf{m}_j^c, \mathbf{\Sigma}_j^c)$, with the classes $c=\{1,2\}$ and the components $j=\{1,2,3\}$ and the parameters

$$m_1^1 = \begin{pmatrix} 1.0 \\ 1.0 \end{pmatrix}, \quad \mathbf{\Sigma}_1^1 = \begin{pmatrix} 3.0 & 0.9 \\ 0.9 & 3.0 \end{pmatrix}, \quad m_2^1 = \begin{pmatrix} 8.0 \\ 8.0 \end{pmatrix}, \quad \mathbf{\Sigma}_2^1 = \begin{pmatrix} 8.0 & 0.0 \\ 0.0 & 6.0 \end{pmatrix}$$

$$m_3^1 = \begin{pmatrix} 13.0 \\ 8.0 \end{pmatrix}, \quad \mathbf{\Sigma}_3^1 = \begin{pmatrix} 3.0 & 0.9 \\ 0.9 & 3.0 \end{pmatrix}, \quad m_1^2 = \begin{pmatrix} 8.0 \\ 1.0 \end{pmatrix}, \quad \mathbf{\Sigma}_1^2 = \begin{pmatrix} 16.0 & -5.0 \\ -5.0 & 16.0 \end{pmatrix}$$

$$m_2^2 = \begin{pmatrix} 1.0 \\ 8.0 \end{pmatrix}, \quad \mathbf{\Sigma}_2^2 = \begin{pmatrix} 1.5 & 0.9 \\ 0.9 & 5.0 \end{pmatrix}, \quad m_3^2 = \begin{pmatrix} 8.0 \\ 13.0 \end{pmatrix}, \quad \mathbf{\Sigma}_3^2 = \begin{pmatrix} 5.0 & 0.9 \\ 0.9 & 5.0 \end{pmatrix}.$$

As described in Section 4, eight additional noisy features sampled from a $\mathcal{N}(0,1)$ density were added. The first two dimensions of the sampled data as well as the defined mixture components are visualized in Fig. 5. 1000 datapoints per class were additionally sampled randomly for testing.

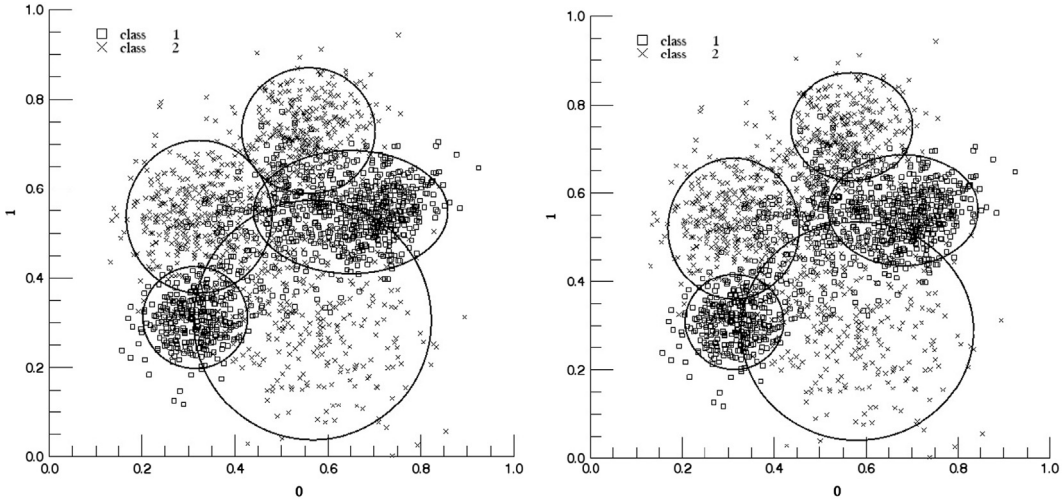

Fig. 6. Feature space representation of the relevant features and the corresponding estimated GMMs. Left: Result of the proposed multiclass EM-algorithm. Right: Result of the proposed modified multiclass EM-algorithm.

5.1.2. MiniBooNE dataset [42] (UCI Machine Learning Repository)

This dataset consists of 130,065 instances with 50 real-valued attributes divided into two classes (signal and background). For each run of the experiment 1200 datapoints per class were randomly chosen for training and 200 for testing.

5.1.3. Image segmentation dataset [42] (UCI Machine Learning Repository)

This dataset contains image data described by 19 attributes, e.g. hue and saturation mean, with respect to 3×3 pixel image regions. Seven different classes, each represented by 300 training and 30 test datapoints, are available.

5.1.4. 3K aerial imagery dataset 1

This real dataset was acquired in context of the project VABENE [43] in 2010 using the airborne optical wide angle 3K camera system [44]. The aim of this exercise was the derivation of occupancy of parking areas during a fair in Munich, Germany, i.e., the separation of occupied and free pixels (2 class problem). Based on three image channels (red, green, blue), 52 features including several first order statistics and color transformations are computed for each image pixel. Training data regions were extracted interactively for both classes. For each run of the experiments, 1600 samples per class were randomly chosen from a set of 2000 samples. The remaining 400 samples were used for testing.

5.1.5. 3K aerial imagery dataset 2

The second 3K dataset was acquired in October 2011 around Bonn, Germany with the same setting and thematically background as the first real dataset. For each run of the experiments, 2400 samples per class were randomly chosen from a set of 3000 data vectors. The remaining 600 samples were used for testing.

5.1.6. 3K aerial imagery dataset 3

This 3K dataset was acquired with the same setting as the first real dataset. The images show the extents of flooded areas around several cities in Germany during the heavy floods in June 2013. The two thematic classes represent water and the complete remaining image content. 89 features including first order statistics, color transformations as well as texture measurements were used. For each run of the experiments, 1500 samples per class were randomly chosen from a set of 3000 data vectors. The remaining 1500 samples were used for testing.

5.2. Quality measurements

There exist two categories of measurements to evaluate the quality of clusterings: internal and external (see for example [45]). Internal measurements only use the training data or the estimated model, i.e., they do not use any external information. The following internal measurements were used in this study:

The sum of squared errors (SSE) is computed by

$$SSE=\sum_{k=1}^{K}\sum_{\forall y_i\in C_k}\|y_i-\mu_k\|^2, \tag{27}$$

where C_k is the set of data points in cluster k and μ_k is the mean vector of all data points $\mathbf{y}\in C_k$.

Furthermore, different scalar scatter criteria can be derived from the scatter matrices. The within-cluster scatter matrix S_W, the between-scatter matrix S_B, the total scatter matrix S_T and the scatter matrix S_k for each cluster k can be calculated by

$$S_k=\sum_{y\in C_k}(y-\mu_k)(y-\mu_k)^T \tag{28}$$

$$S_W=\sum_{k=1}^{K}S_k \tag{29}$$

$$S_B=\sum_{k=1}^{K}N_k(\mu_k-\mu)(\mu_k-\mu)^T \tag{30}$$

$$S_T=\sum_{y\in C_1,\ldots,C_k}(y-\mu_k)(y-\mu_k)^T, \tag{31}$$

where μ is the total mean vector and N_k denotes the number of all data points representing cluster k. Minimizing the trace of S_W, i.e., the within-cluster scatter, is similar to minimizing the SSE

$$T_w=\mathrm{tr}[S_W]=\sum_{k=1}^{K}\sum_{y\in C_k}\|y-\mu_k\|^2. \tag{32}$$

Another criterion, which may be maximized, is the between cluster criterion

$$T_b=\mathrm{tr}[S_B]=\sum_{k=1}^{K}N_k\|\mu_k-\mu\|^2. \tag{33}$$

The eigenvalues $\lambda_1,\ldots,\lambda_d$ of $S_W^{-1}S_B$ are the basic linear invariants of the scatter matrices. Large nonzero eigenvalues indicate good models. The following two criteria are based on the eigenvalues:

$$I_1=\mathrm{tr}[S_W^{-1}S_B]=\sum_{i=1}^{l}\lambda_i \tag{34}$$

$$I_2=\mathrm{tr}[S_T^{-1}S_W]=\sum_{i=1}^{l}\frac{1}{1+\lambda_i}. \tag{35}$$

External measures use N unseen data samples to examine the quality of models. In this study the well known overall classification accuracy

$$oa=\frac{\mathrm{tr}[\Sigma]}{N} \tag{36}$$

as well as the kappa value

$$\kappa=\frac{N\cdot\mathrm{tr}[\Sigma]-\sum_{i=1}^{l}(\Sigma_{i+}\cdot\Sigma_{+i})}{N^2-\sum_{i=1}^{l}(\Sigma_{i+}\cdot\Sigma_{+i})} \tag{37}$$

are used. The number of dimensions of the error matrix Σ is denoted as l and the marginal totals of each row and column of Σ are denoted as Σ_{i+} and Σ_{+i}, respectively.

Since also the runtime rt is relevant for the application of the methods, it is used as an additional criterion.

5.3. Experiment 1: model estimation

The first experiment focuses on the quality of estimated models from different clustering methods without feature selection, i.e., by using all introduced features. The estimated GMMs resulting from split-based EM clustering (Section 3.1) and RSEM clustering (Section 3.2) are compared to the results of the proposed MCFS-EM. The model for MCFS-EM was initialized using k-means with 30 components for each class in all experiments. Whereas the number of components is also estimated by split-based EM, it was set to $K=5$ for each class in all runs of RSEM. The number of RS-iterations was set to $t=30$ in all experiments. In order to relax the assumption of independent features, the standard EM-algorithm is applied to the results of MCFS-EM for post-processing (see Section 4, in the following denoted as MCFS−EM′). In order to provide information concerning the robustness of the methods, 20 runs with each dataset, randomly divided into training and test data, were performed. The mean quality measurement values of experiment 1 are presented in Tables 1–6. For each dataset the number of features l as well as the number of classes c is noted.

Table 1
Results of experiment 1 for the Synthetic dataset.

Criterion	Dataset: Synthetic, $l=10$, $c=2$			
	Split-based EM	RSEM	MCFS-EM	MCFS−EM′
$SSE=T_w$	511.21	**465.46**	468.36	469.92
T_b	**42.89**	85.97	84.28	81.94
I_1	1.80	4.84	**6.60**	5.97
I_2	**9.24**	8.41	8.47	8.50
oa (%)	84.30	84.49	**85.93**	85.40
κ	0.69	0.69	**0.72**	0.71
rt (*s*)	**2.20**	403.70	8.45	16.70

Table 2
Results of experiment 1 for dataset MiniBooNE.

Criterion	Dataset: MiniBooNE, $l=50$, $c=2$			
	Split-based EM	RSEM	MCFS-EM	MCFS−EM′
$SSE=T_w$	2407.24	2445.32	**2211.55**	2295.74
T_b	375.87	**337.72**	658.49	487.29
I_1	3.60	3.18	**5.80**	4.54
I_2	48.58	**48.80**	47.95	48.22
oa (%)	81.32	81.08	**84.96**	80.57
κ	0.63	0.62	**0.70**	0.61
rt (*s*)	**6.55**	23.60	17.85	23.65

Table 3
Mean results of experiment 1 for the Segmentation dataset.

Criterion	Dataset: Segmentation, $l=13$, $c=7$			
	Split-based EM	RSEM	MCFS-EM	MCFS−EM′
$SSE=T_w$	200.40	200.40	**150.97**	200.40
T_b	**497.13**	**497.13**	539.87	**497.13**
I_1	48.95	48.98	**73.00**	48.96
I_2	**8.53**	8.52	7.58	8.53
oa (%)	91.90	**92.90**	89.90	**92.90**
κ	**0.91**	**0.91**	0.88	**0.91**
rt (*s*)	**0.01**	5.55	2.50	2.51

Table 4
Results of experiment 1 for the 3K dataset 1.

Criterion	Dataset: 3K aerial imagery 1, $l=54$, $c=2$			
	Split-based EM	RSEM	MCFS-EM	MCFS−EM′
$SSE=T_w$	2132.12	2068.83	16 903.50	**1961.50**
T_b	**2105.46**	2168.76	19 727.96	2276.22
I_1	20.05	20.69	**50.14**	21.03
I_2	53.02	52.82	**4 465 547.20**	52.83
oa (%)	49.77	50.04	**85.26**	49.77
κ	0.00	0.00	**0.70**	0.00
rt (*s*)	**3.25**	101.20	17.10	7.50

The best of each quality measurement value is highlighted in bold. The tables show that approximately the half of all best quality measurements were reached by the proposed method. Especially in terms of overall classification accuracy as well as kappa statistic, which are important indicators in classification tasks, the proposed method outperformed the other methods for all datasets except Segmentation. For the 3K aerial imagery datasets 1 and 3 very poor results were achieved by the methods that estimate non-diagonal covariance matrices, i.e., that do not assume independence of the features. This was obviously effected by redundant, perfectly correlated features, which lead to non-invertible covariance matrices. This finding further strengthens the general need for feature selection. Due to the fact that the feature independence assumption avoids this behavior, MCFS-EM estimates valid models in that case.

Table 5
Results of experiment 1 for the 3K dataset 2.

Criterion	Dataset: 3K aerial imagery 2, $l=54$, $c=2$			
	Split-based EM	RSEM	MCFS-EM	MCFS−EM′
$SSE=T_w$	4709.21	3939.66	**2499.41**	3782.97
T_b	**1598.53**	2368.07	3550.13	2524.76
I_1	7.37	12.31	**31.64**	14.59
I_2	**53.07**	52.28	49.83	51.85
oa (%)	83.36	89.03	**94.36**	88.20
κ	0.67	0.78	**0.89**	0.76
rt (*s*)	**14.10**	1203.30	44.20	140.95

Table 6
Results of experiment 1 for the 3K dataset 3.

Criterion	Dataset: 3K aerial imagery 3, $l=89$, $c=2$			
	Split-based EM	RSEM	MCFS-EM	MCFS−EM′
$SSE=T_w$	6376.11	6376.11	**3072.41**	6376.11
T_b	**479.40**	**479.40**	3772.02	**479.40**
I_1	–	–	–	–
I_2	–	–	–	–
oa (%)	46.82	48.79	**84.48**	50.11
κ	−0.06	−0.02	**0.69**	0.00
rt (*s*)	**0.01**	33.25	67.70	67.71

The subsequent post-processing step after MCFS-EM (MCFS−EM′) did not result in better models in most cases. This may be an effect of a differing optimal number of components for models with and without the feature independence assumption.

5.4. Experiment 2: feature selection

This experiment focuses on the selection of relevant features for underlying classification problems. The proposed method was compared to the SVM based recursive feature elimination algorithm (RFE-SVM, Section 3.3), the Ranking method (Section 3.5) as well as the heuristic SFFS-EM (Section 3.4). For each of the five binary datasets the selected features were subsequently used to train a GMM using RSEM. This experimental setting was chosen in order to use an estimation method which is independent of the feature selection approach. The evaluation results are shown in Table 7. Since only one result (and not the mean of several runs) is shown, the overall accuracy and kappa of the randomly sampled datasets using RSEM with all features are shown as a benchmark. The features identified by all three methods as well as the best result for each dataset are highlighted in bold.

The identified subsets of relevant features only have few overlap. This may be a result of the different chosen approaches. For the Synthetic dataset – the only dataset, where the relevant features are known – all four methods identified the correct two features as the most relevant ones.

Compared to the benchmark method, the results confirm the positive impact of feature selection, in terms of overall accuracy and kappa. The models estimated with features identified by RFE-SVM lead to the best results for the 3K datasets 1 and 2. The models obtained from Ranking and SFFS-EM yielded the best results for the datasets Synthetic as well as MiniBooNE and 3K dataset 3, respectively.

Table 7
Results of experiment 2 for the binary datasets.

Dataset	Method	Selected features (descending quality)	Runtime (s)	Results of RSEM	
				oa (%)	κ
Synthetic	Benchmark	–	–	85.30	0.71
	SFFS-EM	**2**, **1**, 4, 6, 9	693	87.85	0.76
	RFE-SVM	**2**, **1**, 5, 4,8	1	87.65	0.75
	MCFS-EM	**2**, **1**, 6, 8, 7	33	87.50	0.75
	Ranking	**1**, **2**, 6, 3, 8	**0**	**88.05**	**0.76**
MiniBooNE	Benchmark	–	–	90.00	0.80
	SFFS-EM	**1**, 13, 32, 26, 16, 12, 18, 6, 19, 28	6979	**91.75**	**0.83**
	RFE-SVM	**1**, 32, 16, 13, 28, 30, 4, 21, 42, 23	34	86.50	0.73
	MCFS-EM	17, 16, **1**, 38, 7, 15, 3, 24, 6, 2	636	90.00	0.80
	Ranking	**1**, 2, 7, 35, 32, 38, 4, 28, 17, 8	**0**	86.65	0.73
3K dataset 1	Benchmark	–	–	51.37	0.03
	SFFS-EM	6, 41, 19, 40, 24, 31, 50, 39, 37, 21	6755	97.25	0.94
	RFE-SVM	48, 51, 39, 38, 33, 41, 27, 24, 42, 30	3	**98.00**	**0.96**
	MCFS-EM	38, 41, 48, 51, 44, 54, 45, 40, 52, 49	639	97.87	0.96
	Ranking	52, 44, 49, 46, 48, 51, 47, 45, 40, 54	**0**	97.37	0.95
3K dataset 2	Benchmark	–	–	92.67	0.85
	SFFS-EM	**38**, 25, 52, 12, 24, 48, 21, 22, 28, 11	7246	96.33	0.93
	RFE-SVM	41, 26, **38**, 20, 27, 39, 50, 28, 24, 6	8	**96.92**	**0.94**
	MCFS-EM	41, **38**, 39, 40, 33, 47, 30, 22, 49, 51	2591	96.67	0.93
	Ranking	44, 45, 46, 39, 40, 48, 43, 41, **38**, 54	**0**	95.83	0.92
3K dataset 3	Benchmark	–	–	50.13	0.00
	SFFS-EM	56, 48, 33, 45, 40, 75, 61, 49, 18, 14	5756	**91.13**	**0.82**
	RFE-SVM	35, 78, 64, 70, 31, 30, 56, 37, 59, 58	443	87.77	0.75
	MCFS-EM	39, 43, 51, 46, 82, 15, 20, 12, 40, 16	941	89.33	0.79
	Ranking	4, 82, 81, 86, 62, 83, 65, 29, 10, 84	**0**	84.67	0.69

Hence, MCFS-EM did not provide the best solutions here. However, the highest difference between the overall accuracy achieved using MCFS-EM and the best method is 1.8% (3K dataset 3).

SFFS-EM and MCFS-EM significantly reached the highest values of runtime, whereas MCFS-EM was faster than SFFS-EM up to a factor of about 10. In contrast, the overall accuracies suggest us to choose one of these methods for feature selection, since good models were obtained here for all datasets. Due to this important indicator of robustness, MCFS-EM may be an appropriate method of feature selection in context of time-critical applications.

5.5. Experiment 3: simultaneous model estimation and feature selection

In this experiment, the application of the proposed method using all its components, i.e., simultaneous feature selection and model estimation, was conducted. The aim was to estimate models using the six best features for each dataset. The models of MCFS-EM were compared to the results of the Ranking approach and SFFS-EM. Post-processing of MCFS-EM was not applied here (MCFS−EM′), since this step did not lead to better results in experiment 1. In order to obtain information about the impact of incorporating Mahalanobis distances into the computation of the feature saliencies ρ, the results of the proposed algorithm without Mahalanobis weighting (in the following denoted as MCFS-EM*) are also provided. Furthermore, RSEM was applied for parameter estimation using the features selected by MCFS-EM for a comparison of the estimated models (similar to experiment 1). Note that RSEM is not able to select features. The results of MCFS-EM in experiment 1 using all features are used as a benchmark. The mean results of 20 independent runs with randomly sampled training and test datasets are shown in Tables 8–13. Due to the fact that the internal measurements are not comparable for models with different dimensionalities, only the external measurements are shown for the benchmark method. The best achieved values of the feature selection and model detection methods are highlighted in bold.

Table 8
Results of experiment 3 for the synthetic dataset.

Criterion	Dataset: Synthetic, $l=10$, $c=2$					
	Benchmark	Ranking	SFFS-EM	MCFS-EM*	MCFS-EM	RSEM
$SSE=T_w$	–	329.80	266.41	**263.93**	271.28	244.34
T_b	–	**6.74**	64.49	116.87	118.21	85.10
I_1	–	**0.17**	3.23	6.06	6.91	6.37
I_2	–	5.92	4.79	**4.78**	4.92	4.43
oa (%)	85.93	83.70	**85.59**	80.24	83.73	85.87
κ	0.72	0.67	**0.71**	0.60	0.67	0.72
rt (*s*)	8.45	**0.00**	940.18	28.94	28.65	222.65

The results demonstrate that more suitable and discriminative models can be estimated using a subset of features in general. But not all tested methods lead to better results than that obtained by the benchmark method. In fact, the results obtained by MCFS-EM were poorer than using all features, except for the aerial imagery dataset 1, where a gain of 11% in *oa* was achieved. However, especially in terms of *oa* and κ, MCFS-EM estimated significantly better models than MCFS-EM* for the datasets Synthetic, MiniBooNE as well as for the 3K aerial imagery datasets 1 and 2. Comparably good results were obtained by MCFS-EM and MCFS-EM* for the 3K dataset 3.

For all datasets, except the aforementioned, the best overall accuracies and kappa values were obtained by SFFS-EM – the method with the significantly highest computational costs. Compared to MCFS-EM, the runtime of SFFS-EM was higher up to a factor of about 32. Furthermore, the best runtimes (always lower than 0.00 s) were obtained by the Ranking approach. This simple

Table 9
Results of experiment 3 for dataset MiniBooNE.

Criterion	Dataset: MiniBooNE, $l=50$, $c=2$					
	Benchmark	Ranking	SFFS-EM	MCFS-EM*	MCFS-EM	RSEM
$SSE=T_w$	–	300.53	232.26	183.06	**161.99**	163.04
T_b	–	173.32	**90.43**	121.76	228.74	212.22
I_1	–	**1.89**	2.26	3.71	8.16	5.93
I_2	–	5.51	4.86	4.48	**4.35**	4.11
oa (%)	84.96	83.21	**86.43**	80.79	83.07	84.57
κ	0.70	0.66	**0.73**	0.62	0.66	0.69
rt (*s*)	17.85	**0.00**	2334.12	293.24	510.88	327.94

Table 10
Results of experiment 3 for dataset Segmentation.

Criterion	Dataset: Segmentation, $l=13$, $c=7$					
	Benchmark	Ranking	SFFS-EM	MCFS-EM*	MCFS-EM	RSEM
$SSE=T_w$	–	**38.09**	81.54	197.31	148.12	56.16
T_b	–	2182.09	**237.74**	470.26	418.09	291.69
I_1	–	169.55	**33.22**	946.85	4567.44	31.80
I_2	–	3.30	**2.64**	93.63×10^8	11.68×10^{10}	2.83
oa (%)	89.90	85.71	**94.43**	77.76	75.24	86.92
κ	0.88	0.83	**0.93**	0.74	0.71	0.85
rt (*s*)	2.50	**0.00**	347.29	43.18	339.41	19.18

Table 11
Results of experiment 3 for the aerial imagery dataset 1.

Criterion	Dataset: 3K aerial imagery 1, $l=54$, $c=2$					
	Benchmark	Ranking	SFFS-EM	MCFS-EM*	MCFS-EM	RSEM
$SSE=T_w$	–	**132.25**	150.74	1523.76	137.92	82.40
T_b	–	726.87	**346.10**	2526.49	763.45	439.44
I_1	–	**7.76**	17.00	38.84	1068.33	12.23
I_2	–	5.20	**4.22**	67.36	2394.75	4.47
oa (%)	85.26	98.07	**99.07**	86.46	97.51	98.57
κ	0.70	0.96	**0.98**	0.73	0.95	0.97
rt (*s*)	17.10	**0.00**	4851.12	191.00	1022.18	64.94

Table 12
Results of experiment 3 for the aerial imagery dataset 2.

Criterion	Dataset: 3K aerial imagery 2, $l=54$, $c=2$					
	Benchmark	Ranking	SFFS-EM	MCFS-EM*	MCFS-EM	RSEM
$SSE=T_w$	–	424.53	330.80	1350.24	**190.56**	152.77
T_b	–	648.00	**397.52**	1925.44	1182.46	616.03
I_1	–	**4.66**	10.62	20.68	81.97	15.99
I_2	–	5.30	**4.16**	352.26	25.07	3.96
oa (%)	94.36	91.32	**97.02**	76.59	88.55	96.36
κ	0.89	0.83	**0.94**	0.53	0.77	0.93
rt (*s*)	44.20	**0.00**	6498.47	1001.06	2907.41	226.71

method surprisingly leads to comparably good results for the datasets Synthetic, MiniBooNE, 3K aerial imagery datasets 1 and 2.

The results of RSEM show that this method is able to estimate better models (for all datasets except 3K dataset 3), in terms of external measurements, than the proposed method when the feature subsets are already known.

Table 13
Results of experiment 3 for the aerial imagery dataset 3.

Criterion	Dataset: 3K aerial imagery 3, $l=89$, $c=2$					
	Benchmark	Ranking	SFFS-EM	MCFS-EM*	MCFS-EM	RSEM
$SSE=T_w$	–	444.51	308.61	94.08	**81.50**	233.17
T_b	–	**67.73**	163.49	380.04	397.57	262.22
I_1	–	0.51	**3.81**	14.84	13.80	5.21
I_2	–	5.91	4.63	**3.39**	46.66×10^8	4.26
oa (%)	84.48	71.80	84.89	**84.91**	83.34	82.72
κ	0.69	0.44	0.70	**0.70**	0.67	0.65
rt (*s*)	67.70	**0.00**	4392.82	1293.65	1412.65	291.29

6. Discussion

The results of experiment 1 demonstrate that the proposed MCFS-EM is able to estimate Gaussian mixture models of better quality than other state-of-the-art methods. In terms of computational runtime the method was significantly faster than RSEM for five datasets. In terms of the internal indicator SSE as well as the external overall accuracy and kappa statistic, MCFS-EM outperformed the compared methods for four, respectively five datasets. Furthermore, the standard deviations of the overall accuracies (0.77–3.51%) and the kappa values (0.02–0.15%) were smaller for four of the six datasets, indicating a stronger robustness towards different initializations. On the other hand, the standard deviation of SSE is lower using the other methods for all datasets but one. However, due to the original aim of aerial image classification, the overall accuracy and kappa value are the most important indicators. The split-based EM leads to comparably good results in terms of the measurements T_b, I_2 and especially of runtime. However, SSE, the overall accuracy and the kappa value have shown to be poorer compared to the other methods in the most cases.

The results for the aerial imagery datasets 1 and 3 demonstrate the need of dimension reduction when redundant features are introduced. In that case only MCFS-EM is able to estimate valid models due to the independence assumption of features. Hence, this supposed drawback seems to have positive effects. It has furthermore shown to be very valuable, since it allows a component-wise estimation of model parameters and therefore an initialization using a large number of components. The manual definition of the number of desired components is a crucial drawback of RSEM. This may affect the resulting model significantly when no prior knowledge about the data distribution is available.

Post-processing of the results of MCFS-EM using standard EM (MCFS−EM′) did not lead to better quality measures. This may be a result of the fact that the optimal estimated number of components can differ significantly for models with and without the assumption of independent features.

The results of experiment 2 clarify the general positive impact of feature selection. MCFS-EM did not provide the best results in terms of external measurements. However, the highest difference between the overall accuracy achieved using MCFS-EM and the best method is 1.8%. Additionally, MCFS-EM has shown to be more robust, since good results could be achieved for all tested datasets.

In experiment 3, where feature selection and model estimation was conducted simultaneously, the positive impact of dimension reduction is further demonstrated. Compared to the results of experiment 1, more discriminative models could be estimated using a subset of features. The proposed MCFS-EM outperformed MCFS-EM* for four of the six datasets, demonstrating the positive impact of incorporating Mahalanobis distances into the estimation process. However, the best results were obtained by the heuristic SFFS-EM as well as the ranking method. These two methods do

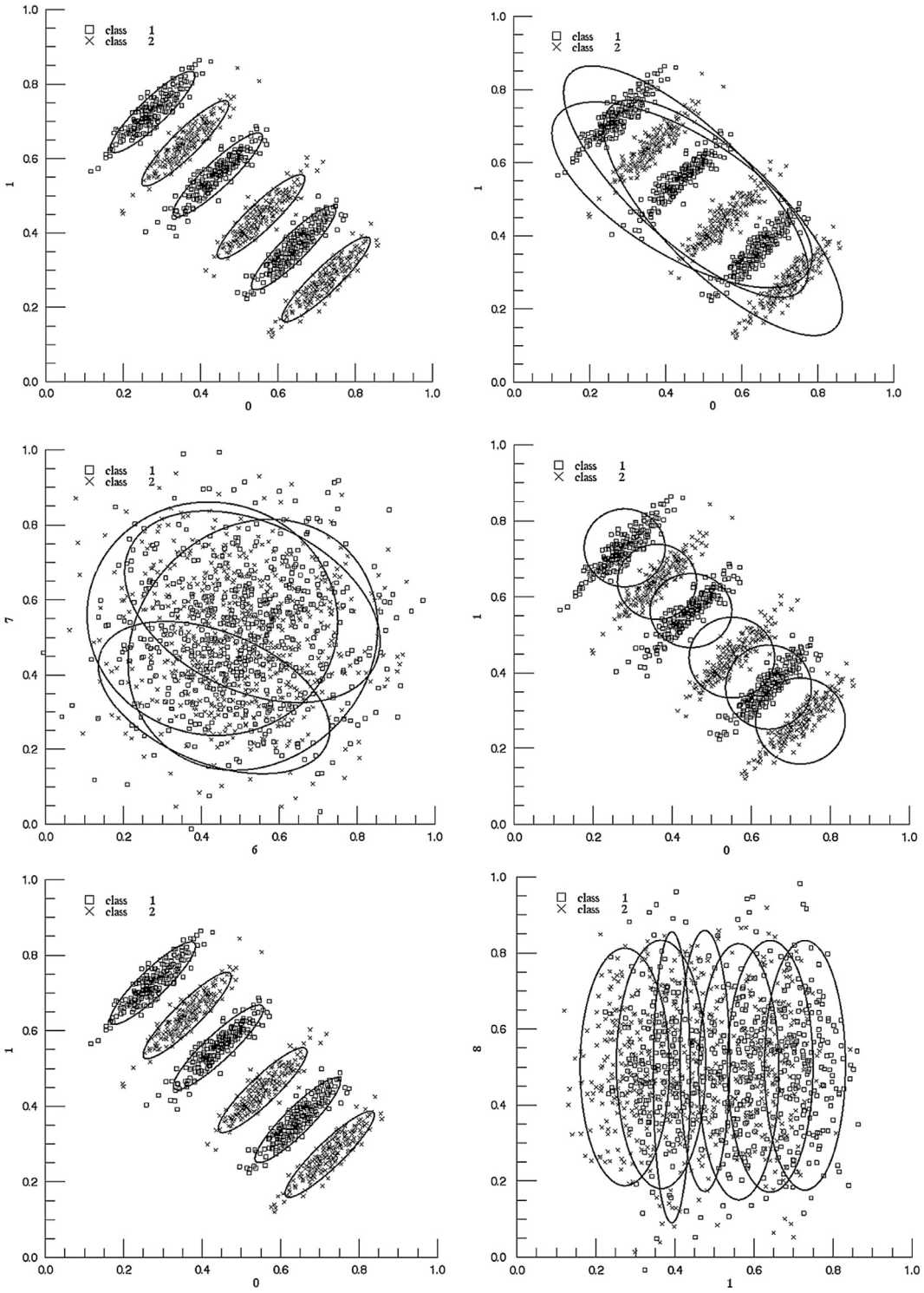

Fig. 7. Two-dimensional feature space representation for the second artificial dataset. Upper left: Sampled distribution. Upper right: Split-EM. Center left: SFFS-EM. Center right: MCFS-EM. Lower left: MCFS−EM′. Lower right: MCFS-EM*.

not estimate the parameters simultaneously. Whereas ranking uses single Gaussians and computes a ranking measurement for each feature individually, SFFS-EM is a filter approach, which compares different feature combinations subsequently using a defined evaluation criterion. Both of these approaches seem to be more suitable than the proposed method in the presented experiments. However, focusing on the original aim of feature selection and model detection for time-critical applications, SFFS-EM may not be appropriate due to the high computational costs. Ranking was the significantly fastest method and obtained better models than MCFS-EM for the datasets Segmentation as well as 3K dataset 2. For the other datasets MCFS-EM estimated comparably good or better models. Hence, in this application-oriented context either ranking or MCFS-EM would be the appropriate choice.

The results of the ranking method confirm an often discussed issue concerning generative modeling, i.e., that solving a classification problem is not necessarily related to finding a complex model that for example consists of multiple Gaussians. According to this

Table 14
Results with the above-described artificial dataset: Overall classification accuracy obtained by application of the estimated models on randomly chosen and unseen testdata.

SPLIT-EM	Ranking	SFFS-EM	MCFS-EM	MCFS−EM′	MCFS-EM*
61.7%	63.7%	60.3%	88.5%	99.9%	61.4%

consideration, the classes of all datasets except 3K aerial imagery dataset 3 are obviously well modeled using a single Gaussian for each class and of course those features, for which this simple model fits well and provides a good separability of the classes. In contrast, 3K aerial imagery dataset 3 is a more complex example, due to obviously inhomogeneous spectral classes (*water* and *rest*). Hence, the results of ranking are significantly poorer than using MCFS-EM. Another more complex artificial dataset is given with the following two-class problem (see Fig. 7, upper left), each class consisting of 600 datapoints with the following means

$$m_1^1 = \begin{pmatrix} 1.0 \\ 9.0 \end{pmatrix}, \quad m_2^1 = \begin{pmatrix} 4.0 \\ 6.0 \end{pmatrix}, \quad m_3^1 = \begin{pmatrix} 7.5 \\ 2.5 \end{pmatrix}$$
$$m_1^2 = \begin{pmatrix} 2.5 \\ 7.5 \end{pmatrix}, \quad m_2^2 = \begin{pmatrix} 6.0 \\ 4.0 \end{pmatrix}, \quad m_3^2 = \begin{pmatrix} 9.0 \\ 1.0 \end{pmatrix},$$

with $m_{j=1,2,3}^{c=1,2}$, the component index j and the class index c. Furthermore, all components have the same covariance matrix:

$$\mathbf{\Sigma} = \begin{pmatrix} 1.0 & 0.9 \\ 0.9 & 1.0 \end{pmatrix}.$$

The three components of each class are located on the diagonal of the subspace of the first two features in an alternating order. Similar to the example described in Section 4, eight noisy features were added to the feature space. In Table 14 the corresponding classification results are shown. As expected, the model estimated by the Ranking method is comparably poor (63.7%) since the classes are not well separable using a single Gaussian for each class.

The models estimated by the split-based EM and SFFS-EM approaches also fail to represent the data in a useful way (see Fig. 7, upper right and center left). The split-based EM-algorithm obviously is not able to find representative models for the given distributions. In contrast, the application of MCFS-EM leads to a more suitable model (see Fig. 7, middle right) and better classification results (88.5%). Furthermore, MCFS-EM was able to significantly identify the two relevant features with the saliencies $\rho = (0.99, 0.99, 0.69, 0.30, 0.48, 0.740.62, 0.63, 0.51, 0.36)$, whereas the saliencies estimated by MCFS-EM* (without Mahalanobis incorporation) are $\rho = (0.97, 1.0, 0.0, 1.0, 1.0, 1.01.0, 0.0, 1.0, 0.0)$. Due to the assumption of independent features, the post-processing step was also tested (MCFS−EM′). The obtained model also incorporates the correlation of features (see Fig. 7, lower left). This model leads to the best result here (99.9% overall accuracy). However, as demonstrated in experiment 1, this post-processing usually does not lead to better models in terms of separability of classes, since the models are optimized with respect to the independence assumption. In other words, the selected features would probably differ by incorporating the correlation of features. All feature space representations of the estimated models (except for ranking) are shown in Fig. 7.

7. Conclusion

In this paper, a new EM approach for simultaneous feature selection and model estimation is proposed and compared to other state-of-the-art methods of clustering, feature selection and both of these tasks. The results of the experiments demonstrate its power and usefulness in the context of supervised aerial imagery classification. The main findings of the presented experiments can be summarized as follows:

1. The selection of relevant features is needed, especially when the number of introduced features is high and redundant features occur.
2. MCFS-EM is able to estimate GMMs which are more suitable for the task of aerial imagery classification than other state-of-the-art methods, for example RSEM.
3. The incorporation of Mahalanobis distances is very valuable for the identification of relevant features.
4. Compared to other state-of-the-art feature selection methods, MCFS-EM did not provide the best solutions in terms of internal and external measurements. However, the highest difference between the overall accuracy achieved using MCFS-EM and the best method was 1.8%.
5. MCFS-EM has shown to be more robust than the compared methods.
6. The application of very fast and simple feature selection methods, like the ranking approach, provides comparably good results for well-behaved data distributions.
7. In case of more complex data distributions, MCFS-EM provided better results in the experiments than the compared methods.
8. Due to the assumption of independent features in MCFS-EM, valid models can be estimated when dealing with perfectly correlated features.

The runtime of MCFS-EM has shown to be much faster than the runtime of the heuristic SFFS-EM approach. Note that all used methods except the RFE-SVM are implemented in high level and therefore slower programming languages. Hence, the runtime is likely to be much faster using optimized implementations.

A further advantage of MCFS-EM lies in the ability of pruning components. The approach of initializing a large number of components has shown to yield more robust models in the experiments than the compared methods. The supposed drawback of independent features given the components seems to be irrelevant in the presented experiments. In fact this assumption has shown to be very valuable, since it allows a component-wise estimation of model parameters and therefore an initialization using a large number of components.

Another important fact is that the desired number of features has to be predefined in SFFS-EM, whereas MCFS-EM allows the selection of the most relevant features, for example by the application of a threshold on the estimated feature saliencies. Finding an appropriate threshold may be a difficult task on the other hand. Therefore, the desired number of features was also predefined in the experiments.

An interesting general outcome concerning generative models is the fact that there is no need to invest time on complex model estimation when it has no relevance for the original classification problem. Because of this, the simple ranking method, which uses a single Gaussian for each class, also achieved good results in the experiments. However, this and other simple approaches are not useful in more complex problems. Discriminative approaches, like the well known SVM, are a promising alternative to generative models, since no assumptions about the distribution of data are introduced. However, the challenge of dimension reduction remains an important issue in all domains.

Conflict of interest

None declared.

References

[1] R.O. Duda, P.E. Hart, Pattern Classification and Scene Analysis, John Wiley & Sons, New York, 1973.
[2] J. Ratsaby, S.S. Venkatesh, Learning from a mixture of labeled and unlabeled examples with parametric side information, in: Proceedings of the Eighth Annual Conference on Computational Learning Theory, COLT '95, ACM, New York, NY, USA, 1995, pp. 412–417.
[3] X. Zhu, Semi-Supervised Learning Literature Survey, Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.
[4] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM-algorithm, J. R. Stat. Soc. Ser. B 39 (1) (1977) 1–38.
[5] G.J. McLachlan, T. Krishnan, The EM Algorithm and Extensions, Wiley Series in Probability and Statistics, 2nd ed., Wiley-Interscience, Hoboken, New Jersey, 2008.
[6] J.A. Bilmes, A Gentle Tutorial on the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models, Technical Report, International Computer Science Institute, Department of Electrical Engineering and Computer Science, 1997.
[7] C.F.J. Wu, On the convergence properties of the EM algorithm, Ann. Stat. 11 (1) (1983) 95–103.
[8] M. Dash, H. Liu, Feature selection for classification, Intell. Data Anal. 1 (1997) 131–156.
[9] M. Koeppen, The curse of dimensionality, in: 5th Online World Conference on Soft Computing in Industrial Applications (WSC5), 2000.
[10] I. Guyon, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (2003) 1157–1182.
[11] P. Pudil, J. Novovicova, J. Kittler, Floating search methods in feature selection, Pattern Recognit. Lett. 15 (11) (1994) 1119–1125.
[12] Y. Peng, Z. Wu, J. Jiang, A novel feature selection approach for biomedical data classification, J. Biomed. Inf. 43 (1) (2010) 15–23.
[13] M. Kudo, J. Sklansky, Comparison of algorithms that select features for pattern classifiers, Pattern Recognit. 33 (2000) 25–41.
[14] A.K. Jain, R.P.W. Duin, J. Mao, Statistical pattern recognition: a review, IEEE Trans. Pattern Anal. Mach. Intell. 22 (1) (2000) 4–37.
[15] F. Ferri, P. Pudil, M. Hatef, J. Kittler, Comparative study of techniques for large scale feature selection, in: E. Gelsema, L. Kanal (Eds.), Pattern Recognition in Practice IV, 1994, pp. 403–413.
[16] J. Kersten, M. Gaehler, S. Voigt, A general framework for fast and interactive classification of optical VHR satellite imagery using hierarchical and planar Markov random fields, PFG Photogramm. Fernerkund. Geoinformation. 6 (2010) 439–449.
[17] J. Kersten, Ein Rahmenwerk zur interaktiven Klassifikation hochaufloesender optischer Satellitenbilder mittels graphenbasierter Bildmodellierung. (Dissertation), Technical University Berlin, Faculty of Electrical Engineering and Computer Science, Department of Computer Engineering and Microelectronics: Computer Vision and Remote Sensing, 2011.
[18] D. Ververidis, C. Kotropoulos, Gaussian mixture modeling by exploiting the Mahalanobis-distance, IEEE Trans. Signal Process. 56 (2008) 2797–2811.
[19] M.H.C. Law, M.A.T. Figueiredo, A.K. Jain, Simultaneous feature selection and clustering using mixture models, IEEE Trans. Pattern Anal. Mach. Intell 26 (2004) 1154–1166.
[20] J.G. Dy, C.E. Brodley, Feature subset selection and order identification for unsupervised learning, in: Proceedings of the 17th International Conference on Machine Learning, Morgan Kaufmann, 2000, pp. 247–254.
[21] K. Mardia, J. Kent, J. Bibby, Multivariate Analysis, Probability and Mathematical Statistics, Academic Press, London, 1979.
[22] P. Langley, Selection of relevant features in machine learning, in: Proceedings of the AAAI Fall Symposium on Relevance, 1994, pp. 1–5.
[23] P.M. Narendra, K. Fukunaga, A branch and bound algorithm for feature subset selection, IEEE Trans. Comput. 26 (1977) 917–922.
[24] A.W. Whitney, A direct method of nonparametric measurement selection, IEEE Trans. Comput. 20 (9) (1971) 1100–1103.
[25] W. Siedlecki, J. Sklansky, A note on genetic algorithm for large-scale feature selection, Pattern Recognit. Lett. 10 (5) (1989) 335–347.
[26] G.J. McLachlan, D. Peel, Finite Mixture Models, Wiley Series in Probability and Statistics, New York, 2000.
[27] Q. Zhao, V. Hautamäki, I. Kärkkäinen, P. Fränti, Random swap {EM} algorithm for gaussian mixture models, Pattern Recognit. Lett. 33 (16) (2012) 2120–2126.
[28] Z. Zhang, EM algorithms for Gaussian mixtures with split-and-merge operation, Pattern Recognit. 36 (9) (2003) 1973–1983.
[29] M.A.T. Figueiredo, S. Member, A.K. Jain, Unsupervised learning of finite mixture models, IEEE Trans. Pattern Anal. Mach. Intell. 24 (3) (2002) 381–396.
[30] C. Constantinopoulos, M.K. Titsias, A. Likas, Bayesian feature and model selection for Gaussian mixture models, IEEE Trans. Pattern Anal. Mach. Intell. 28 (2006) 1013–1018.
[31] P. Pudil, J. Novovicová, N. Choakjarernwanit, J. Kittler, Feature selection based on the approximation of class densities by finite mixtures of special type, Pattern Recognit. 28 (9) (1995) 1389–1398.
[32] Y. Li, M. Dong, J. Hua, Simultaneous localized feature selection and model detection for Gaussian mixtures, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2009) 953–960.
[33] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, Mach. Learn. 46 (1–3) (2002) 389–422.
[34] T. Marill, D.M. Green, On the effectiveness of receptors in recognition system, IEEE Trans. Inf. Theory 9 (1963) 11–17.
[35] P. Somol, P. Pudil, J. Novovicová, P. Paclík, Adaptive floating search methods in feature selection, Pattern Recognit. Lett. 20 (1999) 1157–1163.
[36] H. Roemer, J. Kersten, R. Kiefl, S. Plattner, A. Mager, S. Voigt, Airborne near-real-time monitoring of assembly and parking areas in case of large-scale public events and natural disasters, Int. J. Geogr. Inf. Sci. (2013) 1–18.
[37] H.C. Law, Clustering, dimensionality reduction, and side information (Ph.D. thesis), East Lansing, MI, USA, 2006.
[38] J. Novovicova, P. Pudil, J. Kittler, Divergence based feature selection for multimodal class densities, IEEE Trans. Pattern Anal. Mach. Intell. 18 (2) (1996) 218–223.
[39] S. Vaithyanathan, B. Dom, Generalized model selection for unsupervised learning in high dimensions, in: Proceedings of Neural Information Processing Systems, MIT Press, 1999, pp. 970–976.
[40] G. Celeux, S. Chretien, F. Forbes, A. Mkhadri, A Component-wise EM-Algorithm for Mixtures, Technical Report RR-3746, INRIA, Rhone-Alpes, Montbonnot St. Martin, France, August 1999.
[41] S. Aksoy, R.M. Haralick, Feature normalization and likelihood-based similarity measures for image retrieval, Pattern Recognit. Lett. 22 (2000) 563–582.
[42] A. Asuncion, D.J. Newman, UCI Machine Learning Repository, 2007. URL: ⟨http://www.ics.uci.edu/~mlearn/MLRepository.html⟩.
[43] ⟨http://vabene.dlr.de/vabene⟩, VABENE homepage, 2012 (accessed 23 November 2012).
[44] F. Kurz, R. Mueller, M. Stephani, P. Reinartz, M. Schroeder, Calibration of a wide-angle digital camera system for near real time scenarios, in: High Resolution Earth Imaging for Geospatial Information, ISPRS Workshop 2007, Hannover, Germany, 2007.
[45] L. Rokach, A survey of clustering algorithms, in: O. Maimon, L. Rokach (Eds.), Data Mining and Knowledge Discovery Handbook, Springer, USA, 2010, pp. 269–298.

Jens Kersten was born in Potsdam, Germany, in 1981. He received a Diploma degree in Geodesy from the Technical University of Berlin, Germany, in 2008. In his Diploma thesis he analyzed new approaches in the field of adjustment calculus.

In 2008, he joined the German Remote Sensing Data Center (DFD) of the German Aerospace Center (DLR) where he worked in different research projects focused on operational methods for information extraction from optical and SAR imagery in the context of emergency and crisis mapping as well as traffic monitoring. In 2011 he received the Ph.D. degree in Remote Sensing and Computer Vision from the Technical University of Berlin. His areas of research were interactive classification of aerial and satellite imagery, context-based image classification, graph-based image modeling, generative modeling, feature selection and statistical optimization. Furthermore, he was involved in the development of methods and algorithms at DLR/DFD's Center for satellite-based Crisis Information (ZKI) which is contributing to several national and international activities and projects in the field of disaster mitigation, humanitarian relief, as well as civil security.

Since 2014, he is a research associate at the Bauhaus-Universität Weimar, Germany. His current research topic is the 3D reconstruction from multiple near-range images.