Component-based Feature Saliency for Clustering

Xin Hong, Hailin Li, Paul Miller, Jianjiang Zhou, Ling Li, Danny Crookes,
Yonggang Lu, Xuelong Li, Fellow, IEEE, and Huiyu Zhou

Abstract—Simultaneous feature selection and clustering is a major challenge in unsupervised learning. In particular, there has been significant research into saliency measures for features that result in good clustering. However, as datasets become larger and more complex, there is a need to adopt a finer-grained approach to saliency by measuring it in relation to a part of a model. Another issue is learning the feature saliency and advanced model parameters. We address the first by presenting a novel Gaussian mixture model, which explicitly models the dependency of individual mixture components on each feature giving a new component-based feature saliency measure. For the second, we use Markov Chain Monte Carlo sampling to estimate the model and hidden variables. Using a synthetic dataset, we demonstrate the superiority of our approach, in terms of clustering accuracy and model parameter estimation, over an approach using a model-based feature saliency with expectation maximisation. We performed an evaluation of our approach with six synthetic trajectory datasets obtaining an average clustering accuracy of 97%. To demonstrate the generality of our approach, we applied it to a network traffic flow dataset obtaining an accuracy of 93% for intrusion detection. Finally, we performed a comparison with state-of-the-art clustering techniques using three real-world trajectory datasets of vehicle traffic. Our approach achieved an average clustering accuracy of 96% compared to 77%-95% for the other techniques. In conclusion, for the datasets considered, component based feature saliency measures gave improved clustering over those based on whole models.

1 Introduction

Clustering is one of the most fundamental approaches in data analysis. It discovers structure in data by organising it into homogeneous groups where the within-group-object similarity is maximised and the between-group-object similarity is minimised [1]. There are two main directions one can adopt when developing a clustering approach. The vast majority of early approaches were distance-based algorithms in which some distance measure is defined to govern partitioning tasks. Challenges facing this group are that of uniform distances in high-dimensional data [2], as well as the curse of dimensionality. Recently, finite mixture models have been widely used to provide a formal framework for clustering. These methods take advantage of their natural capacity to represent heterogeneity. This latter group also face issues, including the choice of the statistical distribution, the learning algorithm for the mixture's parameters estimation, the number of clusters, and feature selection in high dimensional problems [3].

Multi-feature clustering has been a challenging problem for several reasons. Three of the main issues are: 1) feature selection for clustering is difficult; 2) it is difficult to find a clustering result consistent over all features; 3) the determination of the number of clusters is interrelated with feature selection [4]. Though it seems that using more features may help a clustering algorithm perform better, in practice some contain little information about data and can be viewed as "noise". These may not contribute to, and sometimes even degrade, the clustering process. This practical challenge has led to the widely studied topic of feature selection. By selecting the "best" feature subset, clustering performance is expected to be improved over that obtained by using them all. Thus, feature selection is a critical technique that helps prevent the curse of dimensionality and allows one to extract a compact representation of the original model [5], [6].

Feature selection in clustering can be roughly organised into two groups. The first group, including feature filter approaches, separate feature extraction from any particular clustering algorithm [7]. The second group can be further divided into two categories: wrapper and embedded approaches [8]. Though a clustering algorithm is involved, the wrapper approaches utilise the clustering algorithm to score the features first and then cluster on the selected feature subset. However, the embedded approaches perform the feature and model selections in a single learning paradigm [9]. Among the three, embedded approaches seem to provide a better solution as the feature and model selections are closely related to each other.

However, there are still several unresolved issues associated with feature selection that have received relatively little attention. Our research focuses on two of these. Firstly, in previous work feature saliency has tended to be related to complete models. As datasets become larger and increasingly complex there is a growing need to adopt a finer-grained approach to saliency such that it is measured in relation to a part of a model. For example, a feature could have high saliency to clustering a particular part of a model and low saliency in relation to another part. Therefore, rather than take the suboptimal average over the whole model we propose to consider the feature saliency in relation to each specific component of

- *Xin Hong and Paul Miller are with the Centre for Secure Information Technology, School of Electronics Electrical Engineering and Computer Science, Queen's University Belfast, UK. E-mail: {x.hong; p.miller}@qub.ac.uk.*
- *Hailin Li and Jianjiang Zhou are with the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu, China.*
- *Ling Li is with the School of Computing, University of Kent, UK.*
- *Danny Crookes is with the School of Electronics Electrical Engineering and Computer Science, Queen's University Belfast, UK.*
- *Yonggang Lu is with the School of Information Science and Engineering, Lanzhou University, China.*
- *Xuelong Li is with School of Computer Science and Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an 710072, P.R. China .*
- *Huiyu Zhou is with the Department of Informatics, University of Leicester, UK. E-mail: hz143@le.ac.uk.*

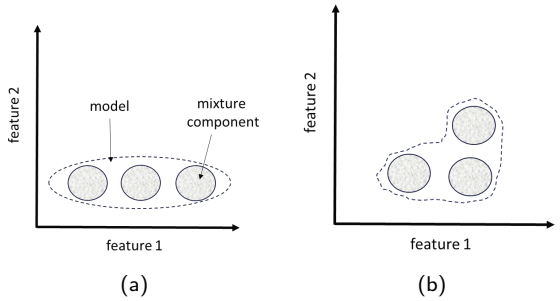

Fig. 1. Two dimensional feature space showing a rectilinear model (dotted ellipse) with three mixture components (filled circles), (a), and a more complex non-linear model, (b).

a model. This is illustrated in Fig. 1 which shows a rectilinear distribution consisting of three components parallel to the x-axis, (a), and a nonlinear model again consisting of three components, (b). It is clear from Fig. 1a that feature 1 along the x-axis has a higher saliency for clustering than that for feature 2 with respect to both the whole model and also each mixture component. In contrast, due to the non-linear complex shape of the model in Fig. 1b we can say that both features have equal saliency overall with respect to the whole model. However, we can also say that feature 1 has a higher saliency with respect to the first component, both features have equal saliency with respect to the second, whilst feature 2 has higher saliency with respect to the third component. Therefore, exploiting this component-based feature saliency should lead to improved clustering for more complex models. The other issue is how to learn the feature saliency and parameters of these more advanced models. Previous approaches to this have primarily employed the expectation-maximisation (EM) algorithm. One of the drawbacks of the EM algorithm is that it can get stuck in local maxima and produce models that generally overfit the data. The more complex the model is the greater the likelihood of this happening.

In this paper, we propose a novel solution to the multi-feature selection problem in clustering. We address the special characteristics of each feature in distinguishing a cluster rather than focusing on selecting a subset of features. For this, we quantitatively measure the clustering relevance of each feature to a cluster, which we refer to as component-based feature saliency. To achieve this we assume that the probability distribution of the features can be modelled as a Gaussian mixture model (GMM). To estimate the feature saliency, mixture models and to optimise the clustering results we use Bayesian parameter estimation with Markov Chain Monte Carlo (MCMC) sampling for those cases where no analytical solution is possible. Although the algorithm is presented with respect to Gaussian mixture-based clustering, it can be extended to other types of model-based clustering as well.

The main contributions of this paper are we:

- Propose a novel mixture-of-Gaussians model that explicitly models the distribution of each feature with respect to each component of the mixture model.
- Introduce a parameter to the model-based approach, which numerically measures relevance of a feature to a mixture component.
- Apply expectation maximisation to the model and derive novel update equations for the various model parameters.
- Present a Bayesian approach, that uses Gibbs simulation, to learn the complete set of model parameters, including mixture parameters, mixture probability and mixture-component based feature saliency.
- Produce a formal derivation of posterior distribution for each of the model parameters.
- Utilise states of the overall likelihood changes combined with the model parameters to determine the number of model components.
- Explore in experiments the properties of the proposed method, such as convergence and initialisation.
- Demonstrate the practical use of the proposed approach by evaluating its performance on two applications; trajectory clustering and network intrusion detection.

The remainder of this paper is organised as follows. In Section 2, we review approaches for feature selection with a focus on previous attempts concerning the feature weighting problem in the general area of clustering. Through the discussion of related work, we identify the contributions of this paper. The details of the proposed component-based feature saliency and Bayesian parameter estimation approach to mixture model-based clustering are presented in Sections 3 and 4 respectively. Experimental results are reported in Section 5, and analysis of results in Section 6. Finally, we conclude the paper in Section 7 and outline some directions for future work.

2 Related Work

In this section, we review work concerning feature selection in clustering in general. Within the context of the review, we highlight the main contributions of our work.

Feature selection is a critical technology that helps prevent the curse of dimensionality and extract a compact representation of the original variable model [5], [6]. When a clustering algorithm is applied to different representations, diverse partitions would be generated. One hopes to find out a consensus partition superior to any input partitions by reconciling diverse partitions (clustering fusion). However, partitions are unlikely to carry an equal amount of useful information due to their distributions being in different representation spaces [10]. Various schemes have been proposed to weight features in the multi-feature based clustering process. In [11] they propose a k-means type clustering algorithm that can automatically calculate feature weights. A new step is introduced to the k-means clustering process to iteratively update feature weights based on the current partition of data and a formula for weight calculation. A theoretical proof of convergence of the new clustering process is given. Experimental results on both synthetic and real data have shown that the new algorithm outperformed the standard k-means type algorithms.

Law et al. [4] define the concept of feature saliency to a GMM, under the assumption that the features are conditionally independent given the (hidden) component label. Using the minimum message length (MML) criterion with log-likelihood for model selection, model parameters and

feature weights are determined by the EM algorithm. Constantinopoulos et al. [12] utilise the same model proposed by [4], but present a Bayesian learning approach for estimating the feature weights and cluster parameters. Their method is based on the integration of a mixture model formulation that takes into account the saliency of the features and a Bayesian approach to mixture learning that can be used to estimate the number of mixture components. The proposed learning algorithm follows the variational framework and can simultaneously optimise over the number of components, the saliency of the features, and the parameters of the mixture model. Experimental results using high-dimensional artificial and real data illustrate the effectiveness of the method. Within the maximum weighted likelihood framework, a variant of the rival penalised EM (RPEM), namely, feature weighted RPEM is developed in [9]. The proposed algorithm differentiates redundant features whilst estimating the number of clusters automatically and simultaneously. Experiments conducted on both synthetic and real data show the efficacy of the proposed approach. In Li et al. [13], local feature saliency, together with other parameters of Gaussian mixtures, are estimated by Bayesian variational learning. Experiments performed on both synthetic and real-world data sets demonstrate that their approach is superior to both global feature selection and subspace clustering methods. Boutemedje et al. [14] present an unsupervised approach for feature selection and extraction in mixtures of generalised Dirichlet distributions. They define a new mixture model that is able to extract independent and non-Gaussian features without loss of accuracy. The proposed model is learned using the EM algorithm with MML. Experimental results show the merits of the proposed methodology in the categorisation of object images. [3] adopts the concept of feature salience and applies the RPEM algorithm in unsupervised non-Gaussian feature selection within the context of finite asymmetric generalised Gaussian mixture-based clustering.

Same as Law et al. [4], Raftery and Dean [15] treat feature selection as a model selection problem in the GMM context. In particular, they consider a collection of parsimonious and interpretable models based on a specific decomposition of the mixture component variance matrix. In their approach, the authors define two different sets of features: those that are relevant and those that are not. An interesting aspect of their approach is that they do not assume that the irrelevant variables are independent of the clustering variables, in contrast to Law et al. [4]. In particular, they define the irrelevant features as those which are independent of the clustering but which remain dependent of the set of relevant features. However, this strong assumption could be viewed as unrealistic in many practical cases. To overcome this limitation, Maugus et al. [16] developed a more generalised framework which combined the approach of Law et al. with that of Rafferty and Dean.

A different approach to model selection for combining feature selection and clustering is to penalise the clustering criteria in order to yield sparsity in the features. This technique has been used, in particular, by penalizing the log-likelihood function to optimise. In the GMM context, Pan and Shen [17] proposed a penalised log-likelihood criterion using an l1-norm, and by assuming a GMM with the same diagonal covariance for each mixture component. Wang and Zhou [18] proposed two other penalty terms. The first one is based on L(1)-norm and the second penalty function is based on hierarchical penalties. Xie et al. [19] extended the model of Pan and Shen [17] by relaxing the equality constraint on the covariance matrices of the different mixture components. Indeed, they proposed an approach dealing with the case of cluster-specific diagonal covariance resulting in a different penalty function. Zhang et al. [20] proposed a novel penalization in the case of a constant covariance in GMM mode. This involved penalising large magnitudes of the inverse covariance matrix elements.

Of the papers reviewed above, the work that mostly motived us is that of Law et al.'s [4]. We effectively extend this work by defining a new feature saliency with respect to each component of a mixture model, rather than the whole model. In addition to the use of EM to determine the feature saliencies and model parameters, we also use Bayesian parameter estimation with MCMC sampling.

3 Component-based Feature Saliency

In this section, we present in detail the proposed component-based feature saliency in mixture model-based clustering. For clarity, Table I summarises the mathematical notation used.

3.1 Mixture Density

Suppose there is a set of N data points $\mathcal{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$ where each $\mathbf{y}_i \in \mathbb{R}^D$ is a vector of D features. We assume the following probability model for the data distribution

$$p(\mathbf{y}|\Theta) = \sum_{j=1}^{K} \alpha_j p(\mathbf{y}|\Theta_j) \tag{1}$$

where $\forall j,\ 0 < \alpha_j < 1,\ \sum_{j=1}^{K} \alpha_j = 1$; each Θ_j is the set of parameters of the jth component; and $\Theta = \{\alpha_1, \ldots, \alpha_K, \Theta_1, \ldots, \Theta_K\}$ denotes the full parameter set. Also, assume there is a set of missing labels, $Z = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$ where $\mathbf{z}_i = \{z_{i1}, \ldots, z_{iK}\}$, with $z_{ij} = 1$ and $z_{ik} = 0$, for $k \neq j$, meaning that $\mathbf{y}_i$ is a sample of $p(\mathbf{y}|\Theta_j)$. For future reference, i, j and l index the data sample number, mixture component and feature respectively.

3.2 Feature Saliency

Let us assume that the features are conditionally independent given the (hidden) component label, that is

$$p(\mathbf{y}|\Theta_j) = \prod_{l=1}^{D} p(\mathrm{y}_l|\theta_{jl}) \tag{2}$$

where y_l denotes the lth feature and θ_{jl} denotes the parameter of the lth feature in the jth component. Inserting (2) into (1) gives

$$p(\mathbf{y}|\Theta) = \sum_{j=1}^{K} \alpha_j \prod_{l=1}^{D} p(\mathrm{y}_l|\theta_{jl}). \tag{3}$$

To represent the relevance of the lth feature to the jth component of the mixture, we introduce a set of binary parameters $\mathcal{B} = \{\beta_{jl}\}$. If the lth feature is relevant to the jth mixture, then $\beta_{jl} = 1$, otherwise $\beta_{jl} = 0$. The lth feature is irrelevant if its distribution is independent of the component labels [21], [22], i.e., it follows a common density, denoted by $q(\cdot|\vartheta_l)$, ϑ_l is

Table I. Notation

i	data sample index	N	total amount of data points	$\mathcal{Z}$	the set of component labels	$\mathcal{P}$	the set of feature saliency probabilities
j	component index	K	total amount of mixture components	$\mathbf{z}_i$	the label vector of the ith data point	ρ_{jl}	the saliency probability of the lth feature in the jth component
l	feature index	D	total amount of features	z_{ij}	the label of the ith data point in the jth component	δ_j	the prior distribution parameter of α_j
$\mathcal{Y}$	data set	Θ	the full parameter set	α	the set of component probabilities	ν_{jl}, ζ_{jl}	the prior distribution parameters of ρ_{jl}
$\mathbf{y}_i$	the data vector at the ith point	Θ_j	the set of parameters of the jth component	α_j	the probability of the jth component	$\xi_{jl}, \tau_{jl}, \lambda_{jl}$	the prior Gaussian distribution parameters of μ_{jl}
y_{il}	the lth feature at the ith data point	θ_{jl}	the parameter of the lth feature in the jth component	μ, Σ	the set of Gaussian mixture parameters	$\hat{a}_{jl}, \hat{b}_{jl}$	the prior Gamma distribution parameters of λ_{jl}
$\mathbf{y}$	random variable	ϑ	the parameter set of common density	$\mu_{jl}, \Sigma_{jl}/\lambda_{jl}$	Gaussian parameters of the lth feature in the jth component	$\tilde{\xi}_l, \tilde{\tau}_l, \tilde{\lambda}_l$	the prior Gaussian distribution parameters of $\hat{\mu}_l$
y_l	the lth feature	ϑ_l	the parameter of the lth feature	$\hat{\mu}, \hat{\Sigma}/\hat{\lambda}$	the set of common Gaussian parameters	$\tilde{a}_l, \tilde{b}_l$	the prior Gamma distribution parameters of $\tilde{\lambda}_l$
$\mathcal{B}$	the set of feature relevance	β_{jl}	the relevance of the lth feature to the jth component	$\hat{\mu}_l, \hat{\Sigma}_l/\hat{\lambda}_l$	Gaussian parameters of the lth feature		

the parameter of the lth feature. Hence, the mixture density in (3) can be written as

$$p(\mathbf{y}|\mathcal{B}, \{\alpha_j\}, \{\theta_{jl}\}, \{\vartheta_l\})) = \sum_{j=1}^{K} \alpha_j \prod_{l=1}^{D} [p((\mathrm{y}_l|\theta_{jl})]^{\beta_{jl}} [q(\mathrm{y}_l|\vartheta_l)]^{1-\beta_{jl}} \tag{4}$$

Here we introduce another variable $\mathcal{P} = \{\rho_{jl}\}$, $\rho_{jl} = P(\beta_{jl} = 1)$, called the component-based feature saliency, which is the probability that the lth feature is relevant to the jth component. As $P(\beta_{jl} = 0) = 1 - \rho_{jl}$, we can write

$$P(\beta_{jl}|\rho_{jl}) = \rho_{jl}^{\beta_{jl}} (1 - \rho_{jl})^{1-\beta_{jl}}. \tag{5}$$

Again, we can write for the mixture density that (see the proof in Appendix A):

$$p(\mathbf{y}|\Theta) = \sum_{j=1}^{K} \alpha_j \prod_{l=1}^{D} (\rho_{jl} p(\mathrm{y}_l|\theta_{jl}) + (1 - \rho_{jl}) q(\mathrm{y}_l|\vartheta_l)) \tag{6}$$

where $\Theta = \{\{\alpha_i\}, \{\rho_{jl}\}, \{\theta_{jl}\}, \{\vartheta_l\}\}$ is the set of all the parameters of the model.

(6) has a generative interpretation. As in a standard finite mixture, we first select the component label j by sampling from a multinomial distribution with parameters $\alpha_1, \ldots, \alpha_K$. Then, for each feature $\mathrm{y}_1, \ldots, \mathrm{y}_D$, we flip a biased coin whose probability of getting a head is ρ_{jl}; if we get a head, we use the mixture component$p(\cdot|\theta_{jl})$ to generate the lth feature; otherwise, the common component $q(\cdot|\vartheta_l)$ is used. This differs from Law et al.'s approach in that the probability of their biased coin getting a head is ρ_l, i.e, there is no dependency on the mixture component.

Thus, let us suppose, for illustrative purposes, that we have a mixture with three components, as in Fig. 1, with $\alpha = [0.5\ 0.3\ 0.2]$. Assume we sample from this multinomial distribution and get $z_{i2} = 1$ indicating the sample comes from the $j = 2$ mixture component. Now let us assume that $D = 2$, also as in Fig. 1, and that $\rho_{21} = 0.8$. Therefore, y_1 will either be generated from $p(\cdot|\theta_{21})$ with a probability of 0.8, or from $q(\cdot|\vartheta_1)$ with probability 0.2. Similarly, assume that $\rho_{22} = 0.6$. In this case, y_2 will either generated from $p(\cdot|\theta_{22})$ with a probability of 0.6, or from $q(\cdot|\vartheta_2)$ with probability 0.4. In this way each feature vector sample, $\mathbf{y}_i$, can be generated feature by feature using $p(\cdot|\theta_{jl})$ and $q(\cdot|\vartheta_l)$.

3.3 Model-based Clustering

Let $(\mathcal{Y}, \mathcal{Z})$ be the completed data set. The density of $(\mathcal{Y}, \mathcal{Z})$ then is

$$P(\mathcal{Y}, \mathcal{Z}|\Theta) = \prod_{i=1}^{N} \prod_{j=1}^{K} \Big[\alpha_j \prod_{l=1}^{D} (\rho_{jl} p(y_{il}|\theta_{jl}) + (1-\rho_{jl}) q(y_{il}|\vartheta_l))\Big]^{z_{ij}} \tag{7}$$

and $P(z_{ij} = 1) = \alpha_j$, $\alpha = (\alpha_1, \ldots, \alpha_K)$, satisfying

$$P\big(\mathbf{z}_i\big|\alpha\big) = \prod_{j=1}^{K} (\alpha_j)^{z_{ij}}. \tag{8}$$

From now on, we shall limit $p(\cdot|\theta_{jl})$ and $q(\cdot|\vartheta_l)$ to be a Gaussian, such that $\theta_{jl} = (\mu_{jl}, \Sigma_{jl})$ and $\vartheta_l = (\hat{\mu}_l, \hat{\Sigma}_l)$. However, the methodology is generic and applies much more widely.

The model likelihood function can then be written as

$$P(\mathcal{Y}|\Theta) = \prod_{i=1}^{N} \Big\{ \sum_{j=1}^{K} \alpha_j \prod_{l=1}^{D} \Big[\rho_{jl} p\big(y_{il}\big|\mu_{jl}, \Sigma_{jl}\big) + \big(1 - \rho_{jl}\big) q\big(y_{il}|\hat{\mu}_l, \hat{\Sigma}_l\big)\Big]\Big\} \tag{9}$$

The objectives of mixture model-based clustering are twofold. One is to infer Θ from the data set $\mathcal{Y}$, i.e. model fitting. The other is to assign each data point to different components, that is to unveil the unobservable set $\mathcal{Z}$.

Once the mixture model has been fitted, a probabilistic clustering of the data into K clusters can be obtained in terms of the fitted posterior probabilities of component membership for the data. Mixing proportions can be thought of as the prior probability that an observation originated from a specific mixing distribution. An outright assignment of the data into K clusters is achieved by assigning each data point to the component to which it has the highest estimated posterior probability of belonging. This is equivalent to finding the component index corresponding to the highest value of p_{ij},

$$\begin{aligned} p_{ij} &= P\big(\mathbf{y}_i,\ z_{ij} = 1,\ z_{ik,\ k\neq j} = 0\big|\alpha, \mathcal{P}, \mu, \Sigma\big) \\ &= \alpha_j \prod_{l=1}^{D} \Big[\rho_{jl} p\big(y_{il}\big|\mu_{jl}, \Sigma_{jl}\big) + \big(1 - \rho_{jl}\big) q\big(y_{il}|\hat{\mu}_l, \hat{\Sigma}_l\big)\Big] \end{aligned} \tag{10}$$

where p_{ij} is the probability that $\mathbf{y}_i$ is generated from the jth component of the mixture.

Therefore, $\mathbf{z}_i$ is estimated as follows:

$$z_{ij} = \begin{cases} 1 & j = \texttt{argmax}\{p_{ij}\} \\ 0 & \text{else} \end{cases} \tag{11}$$

When the mixture model is unknown, model-based clustering has to first estimate $\Theta = \{\{\alpha_i\}, \{\rho_{jl}\}, \{(\mu_{jl}, \Sigma_{jl})\}, \{(\hat{\mu}_l, \hat{\Sigma}_l)\}\}$ from the data set $\mathcal{Y}$, that maximises (9).

4 Model Parameter Estimation

Statistically one assumes that there are a collection of models which could have plausibly generated the data. In this section, we propose two approaches to unsupervised model learning: by the maximum likelihood estimate using the EM algorithm and by the maximum a posterior estimate using a Bayesian MCMC simulation respectively. The latter is the focus of this paper, however, the EM approach enables us to assess the relative contribution of the component-based feature saliency versus the later MCMC approach.

4.1 Model Learning using EM

In this section, we present our first approach to estimating the different parameters using EM. We suppose that the number of components K is known. In the following, we derive parameter update equations for our new model using the EM algorithm.

By treating $\mathcal{Z} = \{z_{ij}\}$ and $\mathcal{B} = \{\beta_{jl}\}$ as hidden variables, we derive (see details in Appendix B) the following EM algorithm for parameter estimation of our new model.

E-step: we compute the following quantities:

$$a_{ijl} = P(\beta_{jl} = 1, y_{il} | z_{ij} = 1) = \rho_{jl} p_{jl}(y_{il} | \mu_{jl}, \Sigma_{jl}) \tag{12}$$

$$b_{ijl} = P(\beta_{jl} = 0, y_{il} | z_{ij} = 1) = (1 - \rho_{jl}) q(y_{il} | \hat{\mu}_l, \hat{\Sigma}_l) \tag{13}$$

$$c_{ijl} = P(y_{il} | z_{ij} = 1) = a_{ijl} + b_{ijl} \tag{14}$$

$$w_{ij} = P(z_{ij} = 1 | \mathbf{y}_i) = \frac{\alpha_j \prod_{l=1}^{D} c_{ijl}}{\sum_{j=1}^{K} \alpha_j \prod_{l=1}^{D} c_{ijl}} \tag{15}$$

$$u_{ijl} = P(z_{ij} = 1, \beta_{jl} = 1 | \mathbf{y}_i) = \frac{a_{ijl}}{c_{ijl}} w_{ij} \tag{16}$$

$$v_{ijl} = P(z_{ij} = 1, \beta_{jl} = 0 | \mathbf{y}_i) = w_{ij} - u_{ijl} \tag{17}$$

M-step: we then reestimate the parameters according to:

$$\alpha_j = \frac{1}{N} \sum_{i=1}^{N} w_{ij} \tag{18}$$

$$\mu_{jl} = \frac{\sum_{i=1}^{N} u_{ijl} y_{il}}{\sum_{i=1}^{N} u_{ijl}} \tag{19}$$

$$\Sigma_{jl} = \frac{\sum_{i=1}^{N} u_{ijl} (y_{il} - \mu_{jl})^2}{\sum_{i=1}^{N} u_{ijl}} \tag{20}$$

$$\hat{\mu}_l = \frac{\sum_{i=1}^{N} \big(\sum_{j=1}^{K} v_{ijl}\big) y_{il}}{\sum_{i=1}^{N} \sum_{j=1}^{K} v_{ijl}} \tag{21}$$

$$\hat{\Sigma}_l = \frac{\sum_{i=1}^{N} \big(\sum_{j=1}^{K} v_{ijl}\big) (y_{il} - \hat{\mu}_l)^2}{\sum_{i=1}^{N} \sum_{j=1}^{K} v_{ijl}} \tag{22}$$

$$\rho_{jl} = \frac{\sum_{i=1}^{N} u_{ijl}}{\sum_{i=1}^{N} w_{ij}} \tag{23}$$

In Law et al.'s model-based feature saliency [4], the corresponding equation is given by

$$\rho_l = \frac{\sum_{i=1}^{N} \sum_{j=1}^{K} u_{ijl}}{\sum_{i=1}^{N} \sum_{j=1}^{K} w_{ij}} \tag{24}$$

In this case we can see that on the numerator there is an inner summation over the component index j. Hence the feature saliency is averaged over all of the mixture components of the model, whereas in our approach we calculate feature saliencies specific to each component of the model. This is the key difference of our component-based feature saliency from Law et al.'s.

4.2 Bayesian Parameter Estimation

In this section we describe how the Bayesian approach to parameter estimation using MCMC sampling is performed. Bayes rule is used to factorise the joint probability distribution of the model parameters, hidden variables and data into a product of distributions that are conditional on the data. For each of the parameters to be estimated we assume a suitable prior distribution and derive an expression for the posterior. These are presented in section 4.2.1 for our model. We then use MCMC sampling to generate samples from the posterior distribution such that we converge on parameter estimates that have a high probability in the posterior distribution. This is described in more detail in section 4.2.2.

4.2.1 Posteriors and Priors

For Bayesian estimation, we use posterior distributions of the model parameters. Below we give a detailed presentation for each of them (see Appendix C for proof of all).

[α]: To estimate α we need to obtain $P(\alpha|\mathcal{Z})$, which from Bayes $\propto P(\mathcal{Z}|\alpha)P(\alpha)$. The prior on α will always be taken as symmetric Dirichlet distribution [23], i.e. $P(\alpha) = Dir(\delta_1, \ldots, \delta_K)$, $\{\delta_j\}$ are the parameters, $\delta_j = \delta_k$ for $j \neq k$. From (8) and knowing $P(\alpha)$ we have the posterior probability for α as

$$P(\alpha|\mathcal{Z}) = Dir(\delta_1 + n_1, \ldots, \delta_K + n_K) \tag{25}$$

where $\delta_1 + n_1, \ldots, \delta_K + n_K$ are the parameters, $n_j = \sum_{i=1}^{N} z_{ij}$.

[$\mathcal{P}$]: An appealing flexible choice for its prior is $P(\rho_{jl}) \sim Beta(\nu_{jl}, \zeta_{jl})$ with $\{\nu_{jl}, \zeta_{jl}\}$ as the parameters, knowing that ρ_{jl} is defined as having compact support in the interval $[0, 1]$. With (5) we then have the posterior probability for $\mathcal{P}$ as

$$P(\mathcal{P}|\mathcal{Y}, \mathcal{Z}, \mathcal{B}) \sim \prod_{j=1}^{K} \prod_{l=1}^{D} Beta(n^*_{jl} + \nu_{jl}, n_j - n^*_{jl} + \zeta_{jl}) \tag{26}$$

where $\{n^*_{jl} + \nu_{jl}, n_j - n^*_{jl} + \zeta_{jl}\}$ are the parameters, $n^*_{jl} = \sum_{i=1}^{N} z_{ij} \phi_{ijl}$, $\phi_{ijl} \in \{0, 1\}$ satisfying

$$\phi_{ijl} = \begin{cases} 1 & r \geq 1 \\ 0 & \text{else,} \end{cases} \qquad r = \frac{\rho_{jl} p\big(y_{il} \big| \mu_{jl}, \Sigma_{jl}\big)}{\big(1 - \rho_{jl}\big) q(y_{il} | \hat{\mu}_l, \hat{\Sigma}_l)}.$$

[μ, Σ]: For the mixture means and variances [24], [25], we assign $P(\mu_{jl}) = N(\xi_{jl}, (\tau_{jl}\lambda_{jl})^{-1})$ and $P(\lambda_{jl}) =$

$Gamma(\hat{a}_{jl}, \hat{b}_{jl})$, where $\lambda_{jl} = \Sigma_{jl}^{-1}$. Hence we have the posterior probability for $\{\mu, \lambda\}$ as

$$P(\mu, \lambda|\mathcal{Y}, \mathcal{Z}) \sim \prod_{j=1}^{K}\prod_{l=1}^{D} N(\mu_{jl}|\xi'_{jl}, (\tau'_{jl}\lambda_{jl})^{-1}) Gamma(\lambda_{jl}|\hat{a}'_{jl}, \hat{b}'_{jl}) \quad (27)$$

where $\xi'_{jl} = \frac{\tau_{jl}\xi_{jl}+\gamma_1}{\tau'_{jl}}$, $\tau'_{jl} = \tau_{jl} + n^*_{jl}$, $\hat{a}'_{jl} = \hat{a}_{jl} + \frac{n^*_{jl}}{2}$, $\hat{b}'_{jl} = \hat{b}_{jl} + \frac{1}{2}\left[\tau_{jl}(\xi_{jl})^2 + \gamma_2 - \tau'_{jl}(\xi'_{jl})^2\right]$, $\gamma_1 = \sum_{i=1}^{N} z_{ij}\phi_{ijl}y_{il}$, and $\gamma_2 = \sum_{i=1}^{N} z_{ij}\phi_{ijl}(y_{il})^2$.

$[\hat{\mu}, \hat{\Sigma}]$: We assign $P(\hat{\mu}_l) = N(\tilde{\xi}_l, (\tilde{\tau}_l\hat{\lambda}_l)^{-1})$ and $P(\hat{\lambda}_l) = Gamma(\tilde{a}_l, \tilde{b}_l)$, where $\hat{\lambda}_l = \hat{\Sigma}_l^{-1}$. Hence we have the posterior probability for $\{\hat{\mu}, \hat{\lambda}\}$ as

$$P(\hat{\mu}, \hat{\lambda}|\mathcal{Y}, \mathcal{Z}) \sim \prod_{l=1}^{D} N(\hat{\mu}_l|\tilde{\xi}'_l, (\tilde{\tau}'_l\hat{\lambda}_l)^{-1}) Gamma(\hat{\lambda}_l|\tilde{a}'_l, \tilde{b}'_l) \quad (28)$$

where $\tilde{\xi}'_l = \frac{\tilde{\tau}_l\tilde{\xi}_l+\tilde{\gamma}_1}{\tilde{\tau}'_l}$, $\tilde{\tau}'_l = \tilde{\tau}_l + \tilde{n}_l$, $\tilde{a}'_l = \tilde{a}_l + \frac{\tilde{n}_l}{2}$, $\tilde{b}'_l = \tilde{b}_l + \frac{1}{2}\left[\tilde{\tau}_l(\tilde{\xi}_l)^2 + \gamma_2 - \tilde{\tau}'_l(\tilde{\xi}'_l)^2\right]$, $\tilde{n}_l = \sum_{j=1}^{K}\sum_{i=1}^{N} z_{ij}(1-\phi_{ijl})$, $\tilde{\gamma}_1 = \sum_{j=1}^{K}\sum_{i=1}^{N} z_{ij}(1-\phi_{ijl})y_{il}$, and $\tilde{\gamma}_2 = \sum_{j=1}^{K}\sum_{i=1}^{N} z_{ij}(1-\phi_{ijl})(y_{il})^2$.

4.2.2 Model Learning by Gibbs Sampling

In practice, the posterior probabilities required to determine the Bayesian model are almost invariably not available analytically, because the parameters of interest usually impose complex non-linear relationships. However, MCMC offers a powerful and flexible method that can produce 'exact' results without imposing an overly burdensome computational overhead. There are two main approaches using MCMC for model determination problems: across-model simulation, in which there is a single MCMC simulation with states of the number of components K and the parameter set $\{\alpha, \mathcal{P}, \mu, \Sigma, \hat{\mu}, \hat{\Sigma}\}$. The second approach is within-model simulation, in which there are separate simulations of $\{\alpha, \mathcal{P}, \mu, \Sigma, \hat{\mu}, \hat{\Sigma}\}$ for each K [26]. In this paper, we choose the latter option. Among many MCMC algorithms, the Gibbs sampler is the most commonly used approach in Bayesian mixture estimation [27].

In general, Gibbs simulation is an iterative process where, at each iteration, parameters are simulated alternatively conditional on one another and on the data $\mathcal{Y}$. For the mixture model with component-based feature saliency proposed in Section 3, Fig. 2 outlines the procedure of Gibbs sampling for model parameter estimation. To begin with, the model parameters are initialised $\{\alpha^{(0)}, \mathcal{P}^{(0)}, \mu^{(0)}, \Sigma^{(0)}, \hat{\mu}^{(0)}, \hat{\Sigma}^{(0)}\}$ is made. These, along with the data $\mathcal{Y}$, are then used to provide an estimate of $\mathcal{Z}$ using (10). The posterior distributions in (25) and (26) are then sampled to give new estimates for α and $\mathcal{P}$. Similarly, $\mathcal{Y}$ is used to calculate parameters for the posterior distribution in (27) and (28) which is sampled to give new estimates of (μ, Σ) and $(\hat{\mu}, \hat{\Sigma})$. The updated parameter estimates, along with $\mathcal{Y}$, are then used to update $\mathcal{Z}$ and so on. In this way the algorithm iteratively estimates new values of the parameters until the maximum a posterior likelihood has been reached. The intuition here is that the estimates over all the iterations effectively comprise a Markov chain which explores the posterior distribution space. When we say the estimates have converged what we are effectively saying is that they have found a high probability region of the posterior distribution. This is what we are looking for as we want to find posterior parameter estimates that are most likely conditional on the data $\mathcal{Y}$.

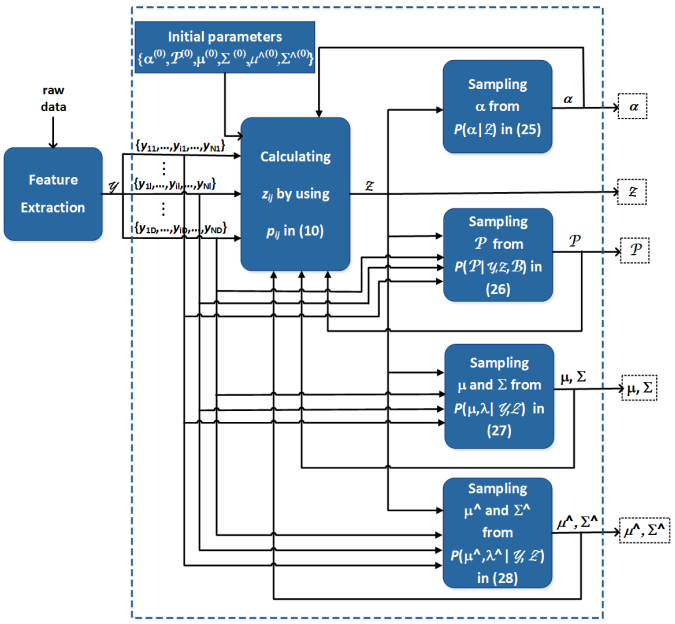

Fig. 2. Block diagram of the proposed algorithm.

5 Experiments

To evaluate the performance of our approach, we used eleven test datasets, including one two-dimensional dataset, nine trajectory datasets, and a dataset of network traffic. For an overall performance measure we used clustering accuracy

$$acc = \frac{\sum_{j=1}^{K} N_j}{N} \times 100 \quad (29)$$

where K is the total number of the clusters in a dataset, N_j is the number of samples that are correctly clustered to the jth cluster and N is the total amount of samples in the dataset.

5.1 Synthetic Two-Dimensional Dataset

The first evaluation focuses on illustrating the two main contributions of the proposed method. Firstly, that feature relevance to each mixture component is considered for clustering, secondly, that an MCMC approach is used to estimate some of the model parameters for clustering optimisation.

To illustrate the first point, we generated a dataset to approximate a distribution that was not rectilinear in the feature space, i.e., it curved, Fig. 3. This was achieved by sampling one hundred data points from each of three two-dimensional Gaussians. The "curve" characteristic of the dataset is obtained by ensuring the second and third Gaussians distributions lie along a line parallel to the first dimension, whilst the first and second Gaussians lie along a line parallel to, and orthogonal to the first line, the second dimension.

We compared the clustering accuracy of our algorithm with Law et al.'s method [4]. They previously proposed the

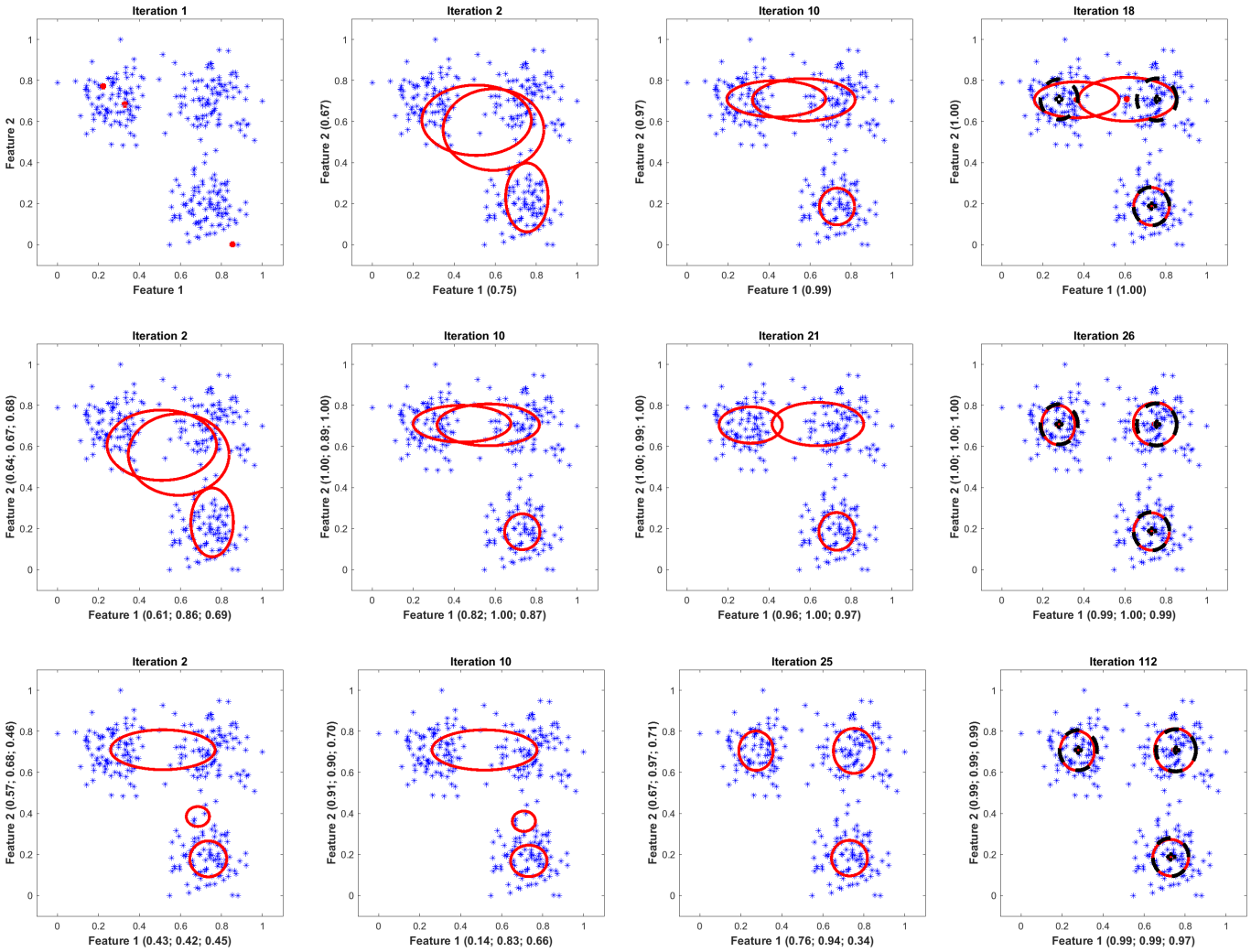

Fig. 3. Plots showing the evolution of the mixture-based model using EM for learning [4] (top row), and our component-based model using EM (middle row) and using Bayesian parameter estimation with MCMC (bottom row). The first at the top row shows the initialisations of three clusters for all three algorithms, the final clustering results are shown in the right-hand column and the other columns show intermediate iterations. The ellipses represent the Gaussian mixture components calculated by the clustering (red) and ground-truth (black dashed). The numbers in parenthesis along the axes are the feature saliencies to the mixture (top row) and components (middle and bottom row). - *All the figures in this paper are best viewed in colour.*

concept of feature relevance, but, in contrast to our approach, they only considered relevance to a mixture model.

To demonstrate our first contribution of using a component-based feature saliency, we replaced the MCMC algorithm of our method with the EM algorithm for model learning and compared it with Law et al.'s approach. Thus, essentially we have modified Law et al.'s approach by replacing their model-based feature saliency with a component-based feature saliency. Both of these approaches were then compared with the approach proposed here of having a component-based feature saliency but with the model parameters estimated using Bayesian inference and MCMC.

For the evaluation, the same initialisation is applied to all three algorithms. They all start with the same set of 3 clusters, the centres of which are randomly chosen with a large variance. The initial clusters are given a random probability weighting, with the initial value for feature saliency fixed at 0.5. Each algorithm was run for a hundred times. The results are given in Table II, along with the ground truth values for μ and Σ, with $acc \pm std$ representing the average clustering accuracy and variance over the hundred runs. The results given for α, $\mathcal{P}$, μ and Σ are taken from a single exemplar run. The three rows for each of α, $\mathcal{P}$, μ and Σ correspond to each of the three clusters respectively. As Law et al.'s approach only has ρ_l, there is only one row. The two columns for each of α, $\mathcal{P}$, μ and Σ correspond to each of the two dimensions.

From Table II, we observe that our component-based feature saliency approach with MCMC is significantly superior to using it with EM and also Law et al.'s approach, in terms of the clustering accuracy and estimated mixture model parameters. Clustering accuracy is significantly improved for our approach with MCMC a bigger factor than component-based feature saliency. The α estimate is improved by using both MCMC sampling and component-based feature saliency. Component

Table II. Clustering Results of our proposed method with MCMC, our proposed method with EM and Law et al.'s EM.

	Ground Truth		Ours MCMC		Ours EM		Law et al.'s EM	
acc	–		$98.72 \pm 0.44\%$		$82.38 \pm 19.99\%$		$80.80 \pm 20.62\%$	
α	–		0.3305 0.3265 0.3430		0.3213 0.3318 0.3469		0.2551 0.3527 0.4122	
$\mathcal{P}$	–		0.9946 0.9930 0.9854 0.9940 0.9728 0.9933		0.9876 1.0 1.0 0.9991 0.9857 1.0		0.9994	0.9987
μ	0.2792 0.7298 0.7553	0.7082 0.1873 0.7075	0.2740 0.7294 0.7512	0.7054 0.1846 0.7077	0.2748 0.7295 0.7476	0.7049 0.1860 0.7093	0.3653 0.7296 0.6119	0.7061 0.1867 0.7085
Σ	0.0911 0.0872 0.0970	0.0975 0.0933 0.1025	0.0070 0.0076 0.0100	0.0091 0.0079 0.0111	0.0066 0.0075 0.0114	0.0091 0.0083 0.0109	0.0425 0.0075 0.0565	0.0077 0.0084 0.0114

saliency measures, ρ_{jl}, for both features and for each component are very high. The model saliency measures for both features used by Law et al.'s, ρ_l, are also very high. Estimates of μ for both component-based approaches are significantly better than those obtained with Law et al.'s approach. Component-based μ estimates are slightly better with MCMC than with EM. Variance estimates for our component-based approach with MCMC are similar to with EM.

On the sample of the exemplar run, ρ_l results of Law et al.'s approach clearly show that only considering feature relevance to a mixture cannot distinguish between individual clusters. However, our ρ_{jl} looks at feature relevance to an individual cluster, therefore, it is able to separate overlapped mixtures by using the two features simultaneously. Furthermore, MCMC learning of mixture model parameters enables the proposed method to achieve more accurate models and improved separation of overlapping mixtures.

Fig. 3 shows sample iterations in a typical run of the proposed component-based feature saliency with both MCMC sampling and an EM implementation, and also Law et al.'s [4] approach. The first plot of the top row illustrates the cluster samples and initial mean parameters (red dots) used by all three algorithms. The plots in the right-most column of Fig. 3 show the converged solutions of the three algorithms. Plots in columns 2 and 3 show the evolution of the parameter estimates of the various approaches with iteration number (the centre of the ellipse is the mean and the ellipse indicates the variance). The plots along the bottom row of Fig. 3 demonstrate that the proposed approach with MCMC dynamically evolves in such a way to capture the mixture components. Similarly, the plots along the middle row show successful convergence for the proposed approach with EM. In both cases, high features saliencies are obtained at convergence. In addition, we can see that the intermediate clusters, represented by the middle two columns, show increasingly better fitting to the clusters with increase in iteration number, giving a more convincing sense of convergence taking place. In contrast, for Law et al.'s approach good matching between the estimated parameters and the samples is only obtained for one cluster at convergence, with mixing occurring for the other two. Thus, we can see that the final convergence of our component-based model with both EM and MCMC is superior to that of Law et al.'s [4] mixture model.

In summary, the bottom right-hand plot of Fig. 3, along with the results of μ and Σ in Table II, clearly show that the estimated components match the ground truth well for our approach with MCMC, with a clustering accuracy of 99% for this exemplar run. Our approach using EM was also able to achieve a high clustering accuracy of 99%, above the 94.67% achieved by Law et al.'s approach. Together with the results of μ and Σ in Table II, we can claim that the component-based feature saliency has contributed to our high performance in comparison to Law et al.'s approach using a model-based feature saliency. The high *acc* over 100 runs achieved by our approach with MCMC in Table II also demonstrates the global nature of the proposed approach with MCMC, thereby achieving a global optimum. In contrast both approaches using EM are local in nature, hence final results with some initialisations only reach a local optimum.

5.2 Synthetic Trajectory Datasets

Six synthetic trajectory datasets were used to demonstrate the capability of the proposed method. These were generated using Piciarelli *et al.*'s simulation programme [28]. For the six datasets the number of clusters was varied from five to ten in steps of one, with each cluster containing one hundred trajectories, Fig. 4.

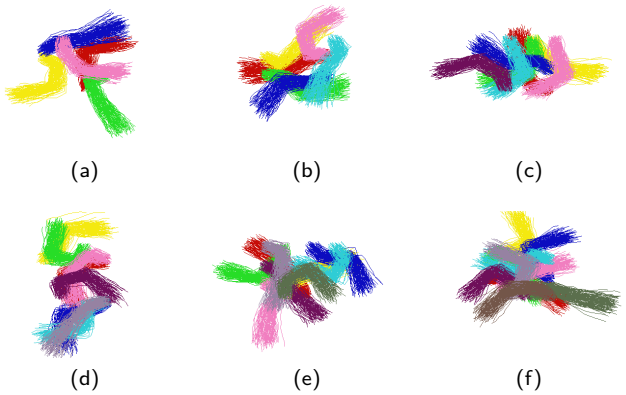

Fig. 4. Plots of the synthetic trajectory datasets, with five (a), six (b), seven (c), eight (d), nine (e), and ten clusters (f). The colour of trajectory indicates the cluster it belongs to.

The first step in trajectory analysis is trajectory representation. Various structural properties of trajectories can be presented by different features. For this evaluation, nineteen features (those listed in Appendix D) were used to represent trajectories. So a trajectory dataset can be transformed to a set of nineteen features. As each feature is represented in two dimensions, each feature set has thirty-eight dimensions in total. Once the trajectories have been transformed into feature spaces, the second step is to cluster trajectories in the feature spaces.

Fig. 5 shows a graph of clustering accuracy versus no. of clusters. These show the average taken over thirty different runs of the algorithm for each cluster number. The initialisation scheme involved randomly selecting a single example for each initial cluster mean. For five of the six datasets the proposed method has achieved a clustering accuracy higher than 99%, the other one being 86% for seven clusters. Analysis showed that in the case of seven clusters our approach was merging two of the clusters into one, whilst separating the other single cluster into two.

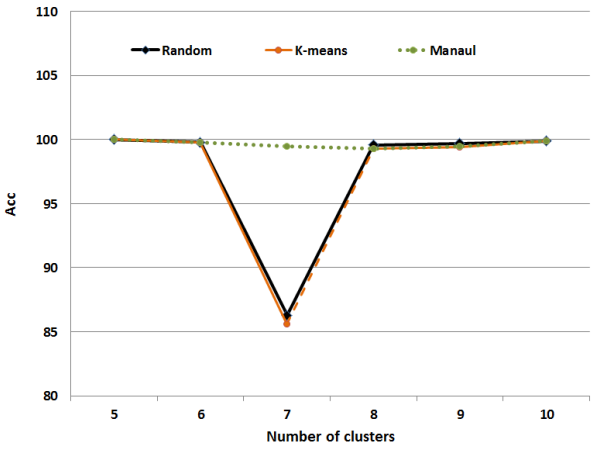

Fig. 5. Plots of clustering accuracy versus number of clusters with random initialisation, (solid line), k-means, (dashed line), and manual (dotted line), for our component-based feature saliency approach with MCMC.

Sensitivity to Initialisation. Based on our previous analysis our suspicion was that the poor results obtained with seven clusters was being caused by poor initialisation of the parameters. The proposed component-based feature saliency approach with MCMC has six parameters: α, $\mathcal{P}$, μ, Σ, $\hat{\mu}$, $\hat{\Sigma}$. To help improve performance we ran k-means on the data as a preprocessing step to try and provide better initial parameter estimates. In addition, we also initialised our algorithm by manually selecting correct samples from each cluster, as opposed to randomly selecting them from the whole dataset. The results for both approaches are shown in Fig. 5. Comparison shows that the use of k-means as a preprocessing step does not improve the clustering accuracy. However, the manual initialisation has improved this obtaining an accuracy of 99.5% significantly higher than 86% with seven clusters. This clearly demonstrates that the poor results with seven clusters were due to random initialisation.

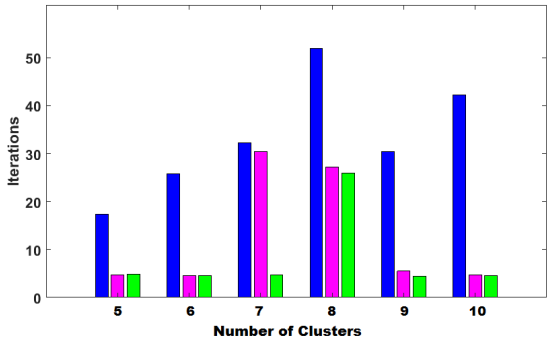

Fig. 6. Computational iteration results on the six synthetic dataset with random initialisation (blue bars), k-means (pink) and manual (green).

An investigation was then performed to determine whether initialisation reduces computation time. Fig. 6 shows that initialisation by k-means, or manually, has significantly reduced the number of iterations in the case of five, six, eight, nine and ten clusters. For seven clusters k-means has not significantly reduced the number of iterations which is consistent with no increase in clustering accuracy. However, manual initialisation has produced a significant reduction in the no. of iterations, which is again consistent with a corresponding significant increase in clustering accuracy.

In summary, these tests reveal that the proposed approach in general works well without initialisation. However, a form of initialisation can benefit clustering performance, in particular reducing computational cost.

Size and Quality of the Feature Set. In this section we investigate the performance with respect to sets of different feature combinations. Firstly, we rank the nineteen features by applying k-means to cluster the samples on each individual feature. For this we used the dataset with ten clusters. Following clustering the accuracy is then calculated and the features ranked from highest to lowest. We then investigated the clustering performance of our approach against a set of fifteen different feature combinations in which we varied the number of features and also the clustering accuracy of the individual features from high (H), to medium (M), to low (L). For example, the fourteenth combination (indexed as S14, containing features of H2M1L1) takes two features with the highest two accuracies, one feature with a medium accuracy, and one feature with the lowest accuracy. Three combinations are also generated to include two features each. The last one uses all nineteen features. These combinations are summarised in Table 2.

Table III. List of feature combinations.

Index	Features	Index	Features	Indx	Features
		Combination			
S1	H4	S2	H3L1	S3	H2L2
S4	H1L3	S5	L4	S6	M4
S7	H1M3	S8	M3L1	S9	H2M2
S10	M2L2	S11	H1M2L1	S12	H3M1
S13	M1L3	S14	H2M1L1	S15	H1M1L2
S16	H1M1	S17	H1L1	S18	M1L1
S19	all nineteen features				

Results for clustering accuracy of the feature combinations using the dataset with ten clusters are given in Fig. 7. In addition, to the results of our approach applied to each feature combination, we have also plotted the lowest clustering accuracy obtained from a single feature, the highest clustering accuracy obtained from single feature and the average clustering accuracy of the features in the combination. In general, feature sets consisting of greater numbers of higher quality features perform better.

To illustrate how our method avails of the characteristics of each feature to improve the clustering accuracy, one may consider combinations S1 (H4) and S2 (H3L1). From Fig. 7 we can see that the clustering accuracy using our approach is approximately the same, yet S2 contains a feature of low quality for clustering, whereas the features of S1 are all high quality. To explain why this is, Fig. 8 shows the confusion matrices for clustering on a single feature and our component-based feature saliency approach with MCMC. Fig. 8(a) - 8(d) shows the confusion matrix for features 6, 13, 15 and 2 individually, which is combination S1 (H4). Analysis of Fig. 8(c)

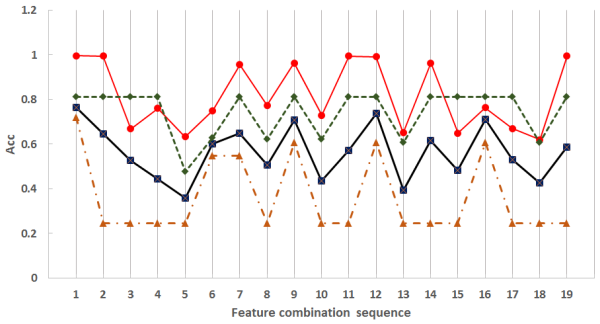

Fig. 7. Accuracy of clustering the ten-cluster dataset on a set of feature combinations using our approach (red), average weighting (blue), lowest single feature (yellow), and highest single feature (green).

shows that feature 15 provides good clustering performance for clusters 1 and 2. Fig. 8(b) shows that feature 13 enables good clustering for cluster 3. Good clustering for cluster 4 is provided by both features 13 and 15, cluster 5 by feature 15 and cluster 6 and 7 by all four features. Fig. 8(d) shows that good performance for clusters 8 and 9 is provided by feature 2, as well as feature 13 Fig. 8(b). Finally, features 2 and 15 provide good clustering for cluster 10. Combining all four features using our component-based feature saliency with MCMC gives an overall clustering accuracy of 99% compared to single feature accuracies ranging from 62% to 71%.

To illustrate the robustness of our approach, we then deliberately selected a feature, feature 16, whose clustering performance is very poor as illustrated by its confusion matrix, Fig. 8(f). As before we then combined features 6, 13 and 15, only this time with feature 16 instead of 2, using our clustering approach. This corresponds to combination S2 (H3L1). The resulting confusion matrix, Fig. 8(g), shows comparable clustering performance with Fig. 8(e), even though we have replaced feature 2 with 16. This shows the ability of our approach to adaptively select the most important features and disregard those that provide little benefit for clustering.

5.3 Network Traffic

To demonstrate the generality of our approach we evaluated its ability to perform clustering on network traffic. The dataset used was the UNSW-NB15 dataset containing both normal traffic and traffic corresponding to nine different types of network attacks including Analysis, Generic, DoS, and Backdoors. Our goal was to develop a network intrusion detection algorithm by first unsupervised learning mixture models for normal traffic. For model estimation, we used 37,000 records of normal network traffic flows. Each flow was characterised by a feature vector consisting of 49 elements (see [29], [30] for details). Based on our learnt model we then estimate the lower probability densities as a threshold, below which traffic was deemed to be not normal, i.e. an attack. To test the accuracy of the model we then measured its performance in detecting attack traffic using a test dataset consisting of 3,700 normal records and 3,585 attack records. The system detected 3,220 of the attacks, whilst detecting 148 normal flows as attacks, corresponding to a classification accuracy of 93%.

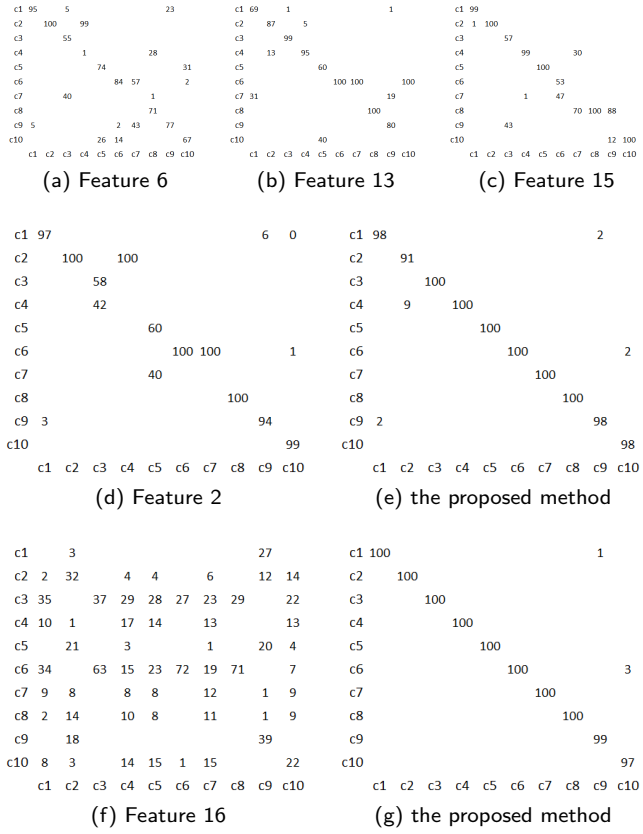

Fig. 8. Confusion matrix of clustering results for features 6 (a), 13 (b), 15 (c), and 2, (d), individually, and their combination S1 (H4), (e), feature 16 (f) and combination S2 (H3L1), (g). The dataset used contained ten clusters.

5.4 Real Traffic Trajectory Datasets

In this section we compared the performance of the proposed approach with state-of-art unsupervised clustering methods in applying to motion trajectory clustering. These include: the mean shift clustering "MS" in [31], the manifold blurring mean shift algorithm "MBMS" in [32], the adaptive multikernel-based shrinkage + K-means "AMKS" in [33], the hidden Markov model based method "HMM"in [34], the sorting potential values based clustering "CSPV" in [35], and the incremental Dirichlet process mixture model-based algorithm "DPMM" in [36].

Three different vehicle motion trajectory datasets [33] are used for the evaluation. *T11:* the traffic trajectory data set in [37], containing 220 trajectories clustered into evelven groups. Each of the clusters contains twenty trajectories. *T15:* contains 1,500 trajectories which were collected by tracking vehicles in a real traffic scene and labelled manually to produce fifteen clusters [36] . *T19:* the traffic trajectory data set in [34], which contains nineteen clusters of 100 trajectories each.

Fig. 9 illustrates the trajectories in the three datasets. Traffic in T15 and T19 can be grouped together by paths and directions. For example, in T15, traffic moving from top to bottom and vice versa comprise two main paths. Fig. 9(b) and 9(d) show the trajectories grouped into main paths, each of which is represented by a different colour. However, note that traffic in each main path may cover more than one

road lane. Fig. 9(c) and 9(e) show the trajectories clustered into road lanes. A main path represented by one colour in Fig. 9(b) and 9(d) is divided into more than one road lane and is represented by different colours in Fig. 9(c) and 9(e). Traffic in T11, Fig. 9(a), consists of main paths only.

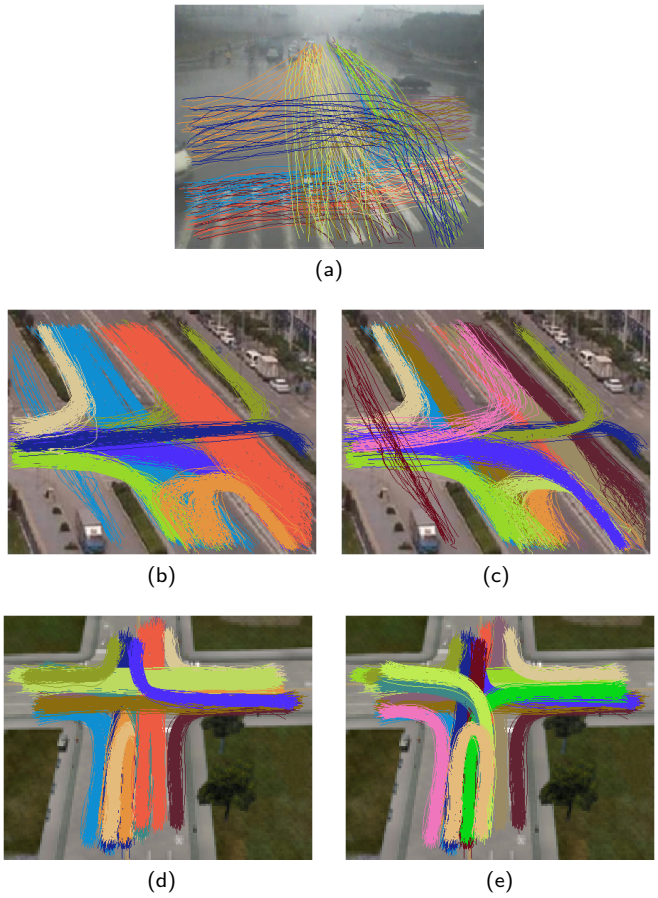

Fig. 9. Visualisation of the three vehicle motion trajectory datasets. Main paths for T11, (a), main paths for T15, (b), with lanes (c), main paths for T19, (d) and with lanes (e). The colour of trajectory indicates the cluster it belongs to.

The experimental results for clustering accuracy are shown in Table IV. To implement our proposed approach, we initialised $\alpha_j = 1/K$ for each component, with all $\rho_{jl} = 0.5$. All μ_{jl} were randomly chosen from 0 to 1, and each σ_{jl} was assigned a small value.

Our approach was implemented in a two-layer hierarchy. In the first layer, a coarse clustering approach is used to extract dominant paths/routes. Then a second layer of clustering is applied to the result of the first layer to achieve a finer result. As a consequence, trajectories in road lanes are mostly separated after the second layer clustering.

The last column represents the results of our method. The values before '/' are the results from the first layer, those after '/' are the results of the second layer clustering, which are the final ones. There is only one result for dataset T11 as it doesn't have lanes to separate, hence, only the first layer of clustering is needed. The results by MS, MBMS, AMKS, HMM and DPMM are taken from [33], where the number of clusters is fixed. The feature set we used was the same as that of CSPV and Law et al.'s, however, the feature sets used by the other baseline techniques were different. The T11 result for DPMM is missing and is marked by a '-'. All of these results are of the final (second layer) clustering. For CSPV, the number of clusters is not fixed and the default parameters are used. CSPV only has results on the main paths for T15 and T19, therefore, the values are given before '/'. A '-' means that a value is not available.

Table IV. Comparison of our approach, in terms of clustering accuracy (%), with other state-of-the-art methods for the T11 (top row), T15 (middle) and T19 (bottom) vehicle trajectory datasets.

MS	MBMS	AMKS	HMM	DPMM	CSPV	Law et al.'s	Our method
95.5	97.7	99.1	86.3	-	36.4	86.82	99.5
85.3	86.6	87.4	84.4	86.6	96.4/-	79.6/55.73	99.9/87.3
98.4	98.6	99.5	96.8	98.0	97.3/-	66.89/64.58	100/99.7

' - ' indicates no result available.

From Table IV, we observe that the results of our approach are better compared to the SOTA methods on all three datasets, apart from our result on T15 being similar to that of AMKS. The results of HMM and DPMM are highly dependent on the clustering initialization. The clustering accuracy of these methods may decrease when they are poorly initialised [33]. However, our approach with MCMC sampling ensures a global optimum is achieved, and is more robust to variation in initialization. Our approach also outperforms both MS and MBMS and achieves better results at the first layer than that of CSPV when both do not fix the number of clusters. Moreover, our method is able to achieve fine clustering of trajectories, i.e., it can separate parallel lanes in a coarse cluster from the first layer.

6 Analysis

6.1 Convergence

Our proposed approach stops iterating if the distance between two consecutive mode estimates becomes less than 0.001 or the iterations have reached the maximum number, in this case 200. To evaluate the convergence of the proposed solution, Fig. 10 shows the variation in log likelihood and the model parameters, α_j, ρ_{jl}, μ_{jl}, and σ_{jl} with iteration for a single trajectory feature with dataset T11. As can be seen, the log likelihood increases sharply then flattens, corresponding to the maximum optimisation of the log likelihood (the likelihood is given by Eq.(9). It is interesting to notice that, although μ_{jl} and σ_{jl} converge after twenty iterations, α_j and ρ_{jl} take more iterations. The converged α_j and ρ_{jl} contribute to a higher log likelihood value. This verifies the effectiveness of our proposed algorithm.

6.2 Number of Clusters

Determining "the right number of clusters" in clustering has always been an important issue. Our approach proposed a solution to automatically select the cluster number at each stage of the clustering hierarchy. We consider the convergence status of the log likelihood to determine the correct value for K. Specifically, the K with the highest converged log likelihood is chosen as the number of clusters in the mixture.

Fig. 11 shows the log likelihood versus iteration with different numbers of clusters for the T11 dataset. For dataset

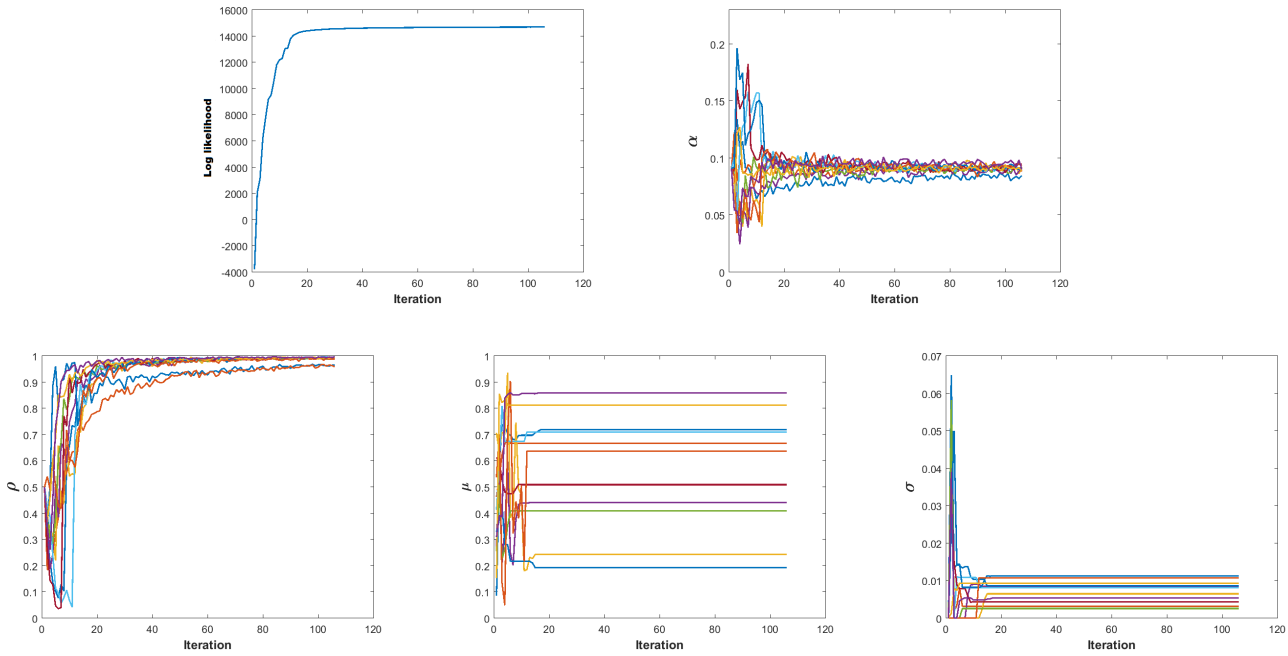

Fig. 10. Plots of log likelihood (a), α, (b), ρ_{jl}, (c), μ_{jl}, (d) and σ_{jl}, (e) with iteration number for a single trajectory feature. The dataset used is T11 and the different colours of (b)-(e) correspond to different clusters.

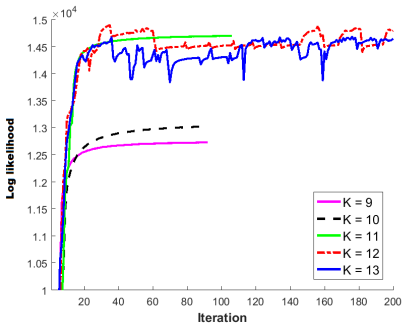

Fig. 11. Log likelihood versus iteration for different number of clusters with the T11 dataset.

T11, $K = 12$ and $K = 13$ overfit and do not converge, whilst $K = 11$ achieves the highest likelihood amongst $K = 9, 10$, and 11. Similar results are obtained for the T15 and T19 datasets.

Cluster overlap appears to be the factor most affecting the clustering results [38]. On dataset T11, Fig. 12 reveals the basic idea of selecting the cluster number by monitoring the changes in $\mathcal{P}$ and μ over time for a single feature. From Figs. 12, all ρ_{jl} and μ_{jl} converge when the cluster number is chosen as ten or eleven. Therefore, the log likelihood values for both cases converge too. As the log likelihood with eleven clusters is higher than that of ten clusters, it indicates that eleven is the more optimal cluster number for the dataset. But on choosing twelve as the cluster number, some of the ρ_{jl} and μ_{jl} do not converge, resulting in the log likelihood. Consequently, we can say that twelve cannot be the correct cluster number.

There is one other thing we feel worth mentioning here. Whilst T11 and T19 have evenly distributed numbers of trajectories in all clusters, T15 has a very uneven number of trajectories for each cluster (smallest one is 19, largest is 271). Hence, it can be observed that our solution is not affected by the cluster size.

6.3 Feature Selection

Let us consider two of the trajectory clusters from the T11 dataset. Each cluster in feature space corresponds to vehicles moving in a straight line along parallel lanes in the road, Fig. 9(a)(yellow and green trajectories). The top row of Fig. 13 shows the distributions of two features, feature 2 and 11, over the two clusters (one shown in red, the other in blue). If we consider feature 2, we can see that the two cluster distributions for this feature are completely overlapping. Hence, one might assume that the feature saliency for this feature with respect to these cluster distributions is low. The bottom row in Fig. 13 show how the feature saliency measure, with respect to both clusters, converges with iteration for different feature distributions. As we can see, the feature saliency measures for feature 2 both converge to a very low value, which is in line with our intuition. For feature 11, we can see that in this case the cluster distributions are well separated. Hence, intuitively one would expect that in this case feature 11 would be very useful for clustering and that its saliency with respect to both clusters would be high. The corresponding graph on the bottom row indeed shows that the saliency measures for feature 11 with respect to both cluster distributions converges to high values, which again is in line with our intuition. Hence, this shows that our approach learns meaningful saliency measures for each feature such that those with high values are effectively"selected" over those with low measures by dint of weighting each feature with the saliency measure.

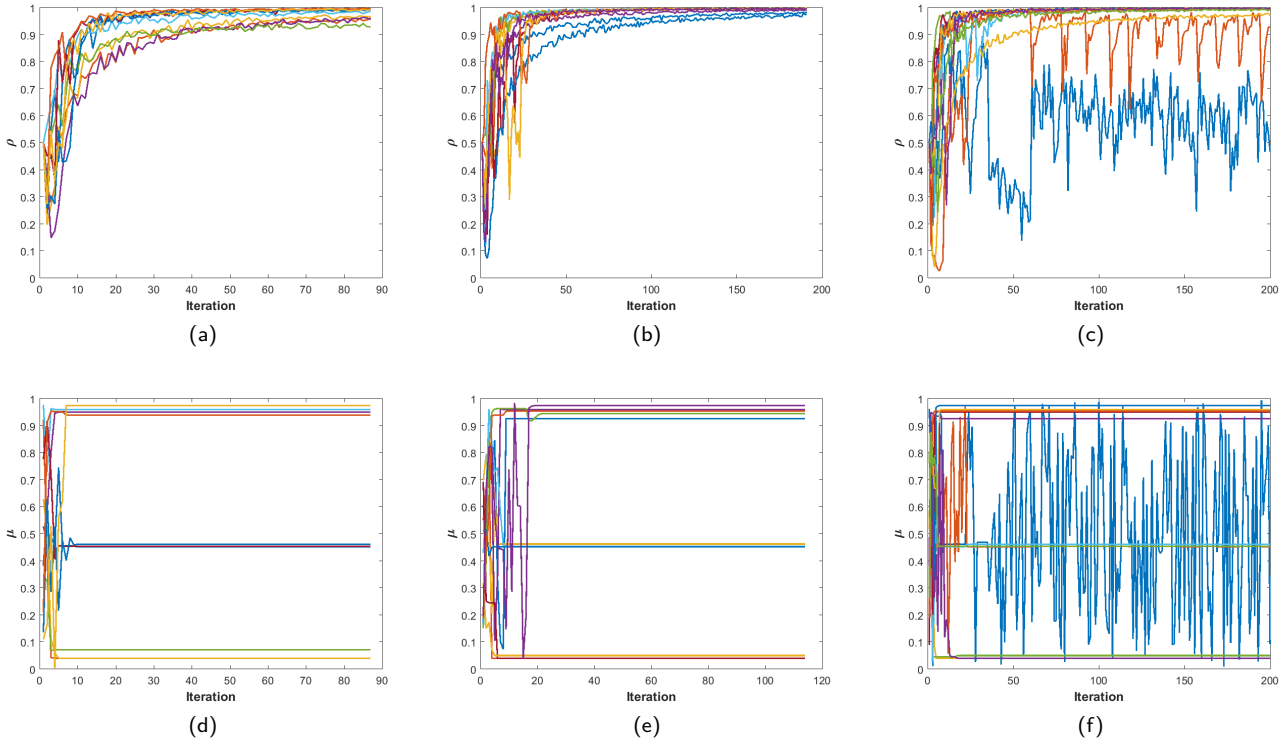

Fig. 12. Plots of ρ_{jl} and μ_{jl} versus iteration number with K=10, (a) and (d), K=11, (b) and (e), and K=12, (c) and (f), respectively, for a single feature with the T11 dataset. (Different colours correspond to different clusters).

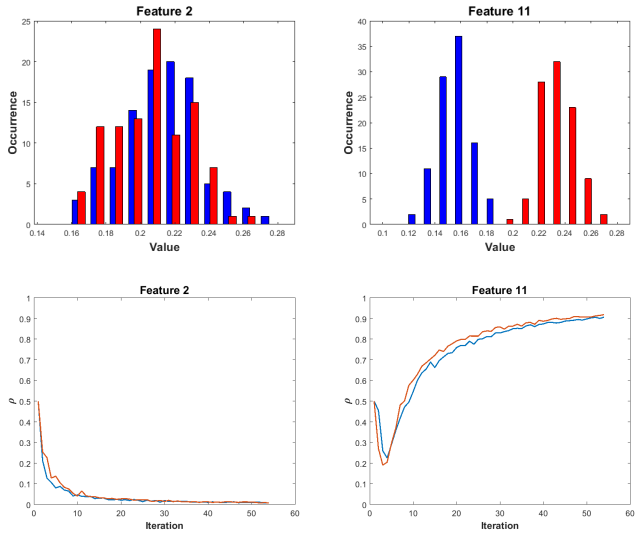

Fig. 13. Example of feature selection: top row - histogram of data, bottom row - extracted ρ_{jl}. Different colours represent different clusters.

6.4 Computational Cost

The computational load in the proposed algorithm is mainly in updating the model parameters (α, $\mathcal{P}$, μ, Σ, $\hat{\mu}$, $\hat{\Sigma}$) and clustering results $\mathcal{Z}$. Computing α requires $O(K)$ calculations, $\mathcal{P}$, μ and Σ all require $O(KD)$. Computing $\mathcal{Z}$ is proportional to $N \times K$. The total computation for each iteration is approximately $O(N \times K \times D)$. The overall amount of computation depends on the number of iterations required for convergence. Although the number of iterations is difficult to estimate, we put an upper bound of two hundred on the number of iterations. However, experiments have shown that less iterations are usually needed for convergence. In most cases, irrespective of whether convergence has occured, optimal results can be obtained with less than a hundred iterations.

7 Conclusion

Our main conclusions are two-fold. Firstly, for complex data models the use of both a component-based feature saliency and Bayesian parameter estimation with MCMC improve feature selection and clustering compared to prior art. Secondly, the improvement is greatest due to the later contribution. We also conclude that, based on our evaluation using synthetic, vehicle trajectory and network traffic flow datasets, our approach has general applicability to clustering.

Our new model currently assumes that the features are independent, however, in many practical scenarios that may not be the case. In future we hope to address this issue formally. Also, for many applications the data is streamed as opposed to batch, we therefore intend to extend our current approach to an online version capable of dealing with streaming data. Finally, we also intend to further investigate the applicability of our clustering approach to the related problem of anomaly detection.

Acknowledgements

We thank the anonymous reviewers and the associate editor for their helpful comments and constructive suggestions. This

work has been in part supported by UK EPSRC under Grants EP/G034303/1 and EP/N508664/1.

References

[1] T. W. Liao, "Clustering of time series data - a survey," *Pattern Recognition*, vol. 38, pp. 1857–1874, 2005.

[2] M. Steinbach, L. Ertöz, and V. Kumar, "The challenges of clustering high-dimensional data," in *New Vistas in Statistical Physics: Applications in Econophysics, Bioinformatics, and Pattern Recognition*. Springer-Verlag, 2003.

[3] T. Elguebaly and N. Bouguila, "Simultaneous high-dimensional clustering and feature selection using asymmetric gaussian mixture models," *Image and Vision Computing*, vol. 34, pp. 27–41, 2015.

[4] M. H. Law, M. A. Figueiredo, and A. K. Jain, "Simultaneous feature selection and clustering using mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, pp. 1154–1166, 2004.

[5] J. Wang, J.-M. Wei, Z. Yang, and S.-Q. Wang, "Feature selection by maximizing independent classification information," *IEEE Trans. Knowl. Data Eng.*, vol. 29, pp. 828–843, 2017.

[6] X. Zhu, X. Li, S. Zhang, C. Ju, and X. Wu, "Robust joint graph sparse coding for unsupervised spectral feature selection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, pp. 1263–1275, 2017.

[7] Q. Gu, Z. Li, and J. Han, "Generalized fisher score for feature selection," in *Proc. UAI*, 2011, pp. 266–273.

[8] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[9] Y. ming Cheung and H. Zeng, "Feature weighted rival penalized em for gaussian mixture clustering: automatic feature and model selections in a single paradigm," in *Proc. CIS 2006*. springer-Verlag Berlin Heidelberg, 2007, pp. 1018–11 028.

[10] Y. Yang and K. Chen, "Temporal data clustering via weighted clustering ensemble with different representations," *IEEE Trans. Knowledge and Data Engineering*, vol. 23, pp. 307–320, 2011.

[11] J. Huang, M. Ng, H. Rong, and Z. Li, "Automated variable weighting in k-means type clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, pp. 657–668, 2005.

[12] C. Constantinopoulos, M. Titsias, and A. Likas, "Bayesian feature and model selection for gaussian mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, pp. 1013–1018, 2006.

[13] M. D. Y. Li and J. Hua, "Simultaneous localized feature selection and model detection for gaussian mixtures," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, pp. 953–960, 2009.

[14] S. Boutemedjet, N. Bouguila, and D. Ziou, "A hybrid feature extraction selection approach for high-dimensional non-gaussian data clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, pp. 1429–1443, 2009.

[15] A. Raftery and N. Dean, "Variable selection for model-based clustering," *Journal of the American Statistical Association*, vol. 101, pp. 168–178, 2006.

[16] G. C. C. Maugis and M.-L. Martin-Magniette, "Variable selection for clustering with gaussian mixture models," *Biometrics*, vol. 65, pp. 701–709, 2009.

[17] W. Pan and X. Shen, "Penalized model-based clustering with application to variable selection," *Journal of Machine Learning Research*, vol. 8, pp. 1145–1164, 2007.

[18] S. Wang and J. Zhou, "Variable selection for model-based high dimensional clustering and its application to microarray data," *Biometrics*, vol. 64, pp. 440–448, 2008.

[19] W. P. B. Xie and X. Shen, "Penalized model-based clustering with cluster-specific diagonal covariance matrices and grouped variables," *Electrical Journal of Statistics*, vol. 2, pp. 168–212, 2008.

[20] D. G. J. M. Zhang, Z., "A flexible and efficient algorithm for regularized fisher discriminant analysis," in *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*, 2009, pp. 632–647.

[21] P. Pudil, J. Novovicová, and J. Kittler, "Feature selection based on the approximation of class densities by finite mixtures of the special type," *Pattern Recognition*, vol. 28, pp. 1389–1398, 1995.

[22] S. Vaithyanathan and B. Dom, "Generalized model selection for unsupervised learning in high dimensions," *Advances in Neural Information Processing Systems*, vol. 12, pp. 970–976, 1999.

[23] C.P.Robert and G. Casella, *Monte Carlo Statistical Methods*. Springer-Verlag.

[24] T. Elguebaly and N. Bouguila, "Bayesian learning of finite generalized gaussian mixture models on images," *Signal Processing*, vol. 91, pp. 801–820, 2011.

[25] C. P.Robert and G. Casella, *Monte Carlo Statistical Methods*, 2nd ed. Springer-Verlag.

[26] D. I. Hastie and P. J. Green, "Model choice using reversible jump markov chain monte carlo," *Statistica Neerlandica*, vol. 66, pp. 309–338, 2012.

[27] J. Diebolt and C. P. Robert, "Estimation of finite mixture distributions through bayesian sampling," *J. R. Statist. Soc. B*, vol. 56, pp. 363–375, 1994.

[28] C. Piciarelli, C. Micheloni, and G. L. Foresti, "Trajectory-based anomalous event detection," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 18, no. 11, pp. 1544–1554, 2008.

[29] N. Moustafa and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *Military Communications and Information Systems Conference (MilCIS), IEEE*, 2015.

[30] ——, "The evaluation of network anomaly detection systems: statistical analysis of the unsw-nb15 data set and the comparison with the kdd99 data set," *Information Security Journal: A Global Perspective*, vol. 25, pp. 18–31, 2016.

[31] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, pp. 603–619, 2002.

[32] W. Wang and M. Carreira-Perpinán, "Manifold blurring mean shift algorithms for manifold denoising," in *Proc. CVPR*. IEEE, 2010, pp. 1759–1766.

[33] H. Xu, Y. Zhou, W. Lin, and H. Zha, "Unsupervised trajectory clustering via adaptive multi-kernel-based shrinkage," in *Proc. ICCV*, 2015, pp. 4328–4336.

[34] B. T. Morris and M. M. Trivedi, "Trajectory learning for activity understanding: Unsupervised, multilevel, and long-term adaptive approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, pp. 2287–2301, 2011.

[35] Y. Lu and Y. Wan, "Clustering by sorting potential values (cspv): a novel potential-based clustering method," *Pattern Recognition*, vol. 45, pp. 3512–3522, 2012.

[36] W. Hu, X. Li, G. Tian, S. Maybank, and Z. Zhang, "An incremental dpmm-based method for trajectory clustering, modeling, and retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, pp. 1051–1065, 2013.

[37] W. Wang, W. Lin, Y. Chen, J. Wu, J. Wang, and B. Sheng, "Finding coherent motions and semantic regions in crowd scenes: A diffusion and clustering approach," in *Proc. ECCV*. Springer, 2014, pp. 756–771.

[38] M. M.-T. Chiang and B. Mirkin, "Intelligent choice of the number of clusters in k-means clustering: an experimental study with different cluster spreads," *Journal of Classification*, vol. 27, pp. 3–40, 2010.

Xin Hong received the B.Sc. degree (Hons.) from Fudan University, China, and the Ph.D. degree in Artificial Intelligence from the University of Ulster, UK. She is currently a Research Fellow with the Centre for Secure Information Technologies, Queen's University Belfast. Her research interests include reasoning under uncertainty, pattern recognition, information fusion and artificial intelligence applications in video surveillance and cyber-physical security.

PLACE PHOTO HERE

Hailin Li is an associate professor with the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu, P. R. China.

Paul Miller received the B.Sc. (Hons.) and Ph.D. degrees from Queen's University Belfast. He is a Senior Lecturer at the School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast. His main research interests include image restoration, segmentation, multi-camera tracking, gender/age profiling in video, deep learning. Dr. Miller was a recipient of the IPRCS International Machine Vision and Image Processing Conference Best Paper Award in 2008.

PLACE PHOTO HERE

Jianjiang Zhou is a full professor with the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu, P. R. China.

Ling Li is the Director of Internationalisation at the School of Computing and the founding coordinator of BC2 Lab. She had six-year research experience at Imperial College London with a focus to understand body sensor data (EEG, EMG, ECG, eAR-sensor, and etc.). She participated in large scale projects, and also involved in projects from government and industry. She now serves at the editorial board of Brain Informatics and the secretary of IEEE Computing Society in UK and Ireland.

Danny Crookes was appointed Professor of Computer Engineering in 1993 at Queen's University Belfast, and was Head of Computer Science from 1993-2002. His research interests include the use of acceleration technologies for high performance image, video and speech processing. He is currently involved in projects in medical imaging for cancer diagnosis, Speech separation and enhancement, Capital Markets software using GPUs, and high level tools for FPGA-based image and video processing.

Yonggang Lu is a professor in the School of Information Science and Engineering, Lanzhou University, China. He received the B.S. and M.S. Degrees in Physics from Lanzhou University of China, and the M.S. and Ph.D. Degrees in Computer Science from New Mexico State University, USA. He finished some of the Ph.D. work at Los Alamos National Lab, of USA. His main research interests include artificial Intelligence, machine learning, pattern recognition, image processing and bioinformatics.

PLACE PHOTO HERE

Xuelong Li (M'02-SM'07-F'12) is a full professor with School of Computer Science and Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an 710072, P.R. China.

Huiyu Zhou received a B.E. degree in Radio Technology from Huazhong University of Science and Technology of China, and a Master of Science degree in Biomedical Engineering from University of Dundee of UK. He was then awarded the PhD degree in Computer Vision from Heriot-Watt University, UK. Dr. Zhou presently is a Reader at Department of Informatics, University of Leicester, UK.