Fast Sample Generation with Variational Bayesian for Limited Data Hyperspectral Image Classification

AmirAbbas Davari, Hasan Can Özkan, Andreas Maier, Christian Riess
Pattern Recognition Lab, Department of Computer Science, Friendrich-Alexander University
Erlangen, Germany, Email: (amir.davari, hasan.can.oezkan, andreas.maier, christian.riess)@fau.de

***Abstract*—Labeling data for hyperspectral remote sensing image classification is a tedious and cost-intensive task. As a consequence, it is oftentimes necessary to perform classification when only very limited number of labeled training data is available. Several approaches have been proposed to address this problem. A recent proposal is to generate additional synthetic samples from a Gaussian Mixture Model for each class. One challenge with this approach lies in determining the number of components in the GMM.**

In this paper, we propose an approximation algorithm to select the number of components, namely Variational Bayesian (VB). The main advantage of VB is that it does not require multiple clustering computations in advance. Variational Bayesian not only greatly decreases the computational cost, but also generates comparable or better results in comparison to other methods.

***Index Terms*—Gaussian mixture model (GMM), Variational Bayesian (VB), synthetic data, hyperspectral remote sensing image classification, limited training data**

I. Introduction

Hyperspectral remote sensing (HSRS) is used in many application fields, including agriculture, mineralogy, surveillance, astronomy and environmental monitoring [1]. For these applications, one of the most common objectives is to identify objects or materials from their spectral and spatial signature. Identifying objects or materials for remote sensing can be done via classification. Two notorious challenges in HSRS image classification are the high dimensionality of feature data and the limited availability of training data. Both challenges together make HSRS image classification a difficult task.

Several approaches have been proposed to address this task which can be roughly categorized into two groups. The first category focuses more towards developing robust classifiers to limited training data, e.g., [2]–[4]. The second group, concentrates on reducing the feature dimensionality since the high dimensional features in the presence of limited data can be a more severe bottleneck, e.g. [5]–[7].

A recently proposed approach opens another direction to address the limited training data bottleneck [8] by adding synthetically generated training samples. The approach is to fit a Gaussian Mixture Model (GMM) to the few available training samples, and then to augment the trainingset with samples drawn from the GMM. While the overall idea is interesting, its processing pipeline is somewhat prototypical. A specific difficulty lies in finding a good GMM parameterization. Here, the authors compute four Gaussian mixture models (GMMs), consisting of one, two, three, and four components. Akaike information criterion (AIC) [9] is then used to decide for one of these models, or in other words, to determine the number of GMM components to use. This approach is somewhat heuristic in capping the maximum number of components at four, and inelegant in that it requires computation of all models to find the best model with respect to its AIC.

Variations of the proposed approach to determine the number of clusters might for example consider the Bayesian information criterion (BIC) [10], or more complex approaches. For example, Celeux et al. [11] determine the number of mixtures from an entropy criterion which is derived from a relation, combining the likelihood and the classification likelihood of a mixture. Laxhammar et al. [12] estimated the number of mixtures using a holdout method to detect overfitting. Other approaches to determine the number of mixtures aim at optimizing a function of inter-cluster and intra-cluster distances, like the elbow method or the average silhouette width [13], [14]. Tibshirani et al. [15] proposed the gap statistic which is based on the within intra-cluster variation for different numbers of clusters. However, the main drawback with all the aforementioned methods is, analogously to AIC, that the clustering algorithm needs to be computed several times a priori and the computed models need to be evaluated based on a certain criterion which makes this process expensive and time-consuming.

In this work, we propose to address this issue by implicitly adjusting the number of clusters with a Variational Bayesian (VB). Dirichlet Process (DP) have previously been used for this task for farm environmental estimation [16] and flame detection [17]. Williams et al. [18] used Variational Bayesian (VB) to estimate a GMM for small training sets. In a similar spirit, we investigate in this work VB for generation of GMMs and fast extraction of synthetic samples for hyperspectral remote sensing image classification. One particular benefit over the aforementioned methods is that the computational complexity is independent from the number of components to examine. This allows to not only consider four components, but an arbitrary number. While being more flexible, VB is almost twice as fast as AIC and BIC, eight times faster than the average silhouette width method and in the magnitude of hundreds times faster than the gap method, while yielding at least comparable, oftentimes better classification performance.

The rest of this paper is organized as follows. In Sec. II, we introduce the Variational Bayesian. In Sec. III, we present our workflow, and in Sec. IV the experimental setup and results. Section V concludes the paper.

II. Variational Bayesian Inference for GMM

Variational Bayesian (VB) can be considered as a family of methods that makes the computation of probability distributions tractable. VB methods are an extension of the EM algorithm that maximizes a lower bound on model evidence $p(\boldsymbol{X})$ where $\boldsymbol{X}$ denotes the set of observations. Variational methods and EM are both iterative algorithms which alternate between a) determining the probabilities for a data point to belong to a mixture component and b) to fit the mixture to the corresponding data. However, variational methods add regularization by integrating information from prior distributions. A particularly nice property of VB over maximum likelihood GMM is that VB methods avoid over-fitting and singularities [19].

Given an observation x, a Gaussian mixture model can be written as

$$p(x|\boldsymbol{\mu}, \boldsymbol{\Lambda}, \boldsymbol{\pi}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) \tag{1}$$

where $\boldsymbol{\mu}_k$ and $\boldsymbol{\Lambda}_k$ are the mean and covariance matrix of the k-th Gaussian component, $\boldsymbol{\pi}_k$ is the mixing coefficient, and K is the number of mixture components.

Assume that the N observations are introduced as $\boldsymbol{X} = \{x_1, ..., x_N\}$, and the latent variables as $\boldsymbol{Z} = \{z_1, ..., z_N\}$. Probabilistic formulation of VB becomes easier when the membership of the GMM components is made explicit. To this end, each observation x_i has an associated latent indicator variable z_i. Then, $p(\boldsymbol{X})$ is the marginal distribution of $p(\boldsymbol{X}, \boldsymbol{Z})$, i.e.,

$$p(x) = \sum_{z} p(z) p(x|z) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Lambda_k) \ , \tag{2}$$

where we omitted for clarity of notation the dependency on the model parameters $\boldsymbol{\mu}$, $\boldsymbol{\Lambda}$, and $\boldsymbol{\pi}$.

Consider a variational distribution which factorizes into latent variables and model parameters as

$$q(\boldsymbol{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = q(\boldsymbol{Z}) q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) \ . \tag{3}$$

This factorization is the only assumption required in order to acquire a tractable and useful result for the mixture model. Considering the expectation maximization (EM), $q(\boldsymbol{Z})$ is estimated in the expectation and $q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})$ in the maximization step. Both can be determined automatically by optimizing the variational distribution. For the full theoretical derivation we refer to [19, Chap. 10] due to space constraints. We will restrict the exposition here to the required EM update equations. For the expectation, the update is

$$q^*(\mathbf{Z}) = \mathbb{E}[\mathbf{z_{nk}}] = \mathbf{r_{nk}} \ , \tag{4}$$

where r_{nk} denotes the "responsibility" of component k to sample n, which will be defined in Eqn. 15 further below. Let furthermore

$$N_k = \sum_{n=1}^{N} r_{nk} \tag{5}$$

$$\bar{x}_k = \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} x_n \tag{6}$$

$$S_k = \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} (x_n - \bar{x}_k)(x_n - \bar{x}_k)^T \tag{7}$$

denote three auxiliary statistics derived from r_{nk}, namely the number of assigned samples, average and covariance. The update equations for the maximization step are based on the factorization

$$q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = q(\boldsymbol{\pi}) \prod_{k=1}^{K} q(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) \ . \tag{8}$$

The individual terms are

$$q^\star(\boldsymbol{\pi}) = \mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}) \ , \tag{9}$$

where $\mathbf{Dir}$ denotes the Dirichlet distribution as a prior for the mixture weights, and $\alpha_k = \alpha_0 + N_k$.

The second factor of Eqn. 8 is represented as a product of a Gaussian distribution $\mathcal{N}$ and a Wishart distribution $\mathcal{W}$,

$$q^*(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) = \mathcal{N}(\boldsymbol{\mu}_k|\boldsymbol{m}_k, (\beta_k \boldsymbol{\Lambda}_k)^{-1}) \mathcal{W}(\boldsymbol{\Lambda}_k|\boldsymbol{W}_k, \boldsymbol{\nu}_k) \tag{10}$$

where

$$\beta_k = \beta_0 + N_k \tag{11}$$

$$\boldsymbol{m}_k = \frac{1}{\beta_k}(\beta_0 \boldsymbol{m}_0 + N_k \bar{\boldsymbol{x}}_k) \tag{12}$$

$$\boldsymbol{W}_k^{-1} = \boldsymbol{W}_0^{-1} + N_k \boldsymbol{S}_k + \frac{\beta_0 N_k}{\beta_0 + N_k}(\bar{\boldsymbol{x}}_k - \boldsymbol{m}_0)(\bar{\boldsymbol{x}}_k - \boldsymbol{m}_0)^T \tag{13}$$

$$\nu_k = \nu_0 + N_k \tag{14}$$

denote the remaining parameters for the maximization step.

Finally, the responsibilities r_{nk} are computed as

$$r_{nk} \propto \widetilde{\pi}_k \widetilde{\Lambda}_k^{1/2} \exp\{-\frac{D}{2\beta_k} - \frac{\nu_k}{2}(\boldsymbol{x_n} - \boldsymbol{m}_k)^T \boldsymbol{W}_k (\boldsymbol{x_n} - \boldsymbol{m}_k)\} \ , \tag{15}$$

where D denotes the feature dimensionality. Eqn. 15 makes use of the expectation

$$\begin{aligned} &\mathbb{E}_{\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k}[(\boldsymbol{x}_n - \mu_k)^{\mathrm{T}} \boldsymbol{\lambda}_k (\boldsymbol{x}_n - \boldsymbol{\mu}_k)] \\ &= D\beta_k^{-1} + \nu_k (\boldsymbol{x}_n - \boldsymbol{m}_k)^{\mathrm{T}} \boldsymbol{W}_k (\boldsymbol{x}_n - \boldsymbol{m}_k) \end{aligned} \tag{16}$$

and the expectations

$$\ln \widetilde{\Lambda}_k \equiv \mathbb{E}[ln\, |\boldsymbol{\Lambda}_k|] \tag{17}$$

$$\ln \widetilde{\pi}_k \equiv \mathbb{E}[ln\, \pi_k] \tag{18}$$

with

$$\ln \widetilde{\Lambda}_k = \sum_{i=1}^{D} \psi \left(\frac{\nu_k + 1 - i}{2} \right) + D \ln 2 + \ln |\boldsymbol{W}_k| \qquad (19)$$

$$\ln \widetilde{\pi}_k = \psi(\alpha_k) - \psi(\sum_k (\alpha_k)) \ , \qquad (20)$$

where $\psi(\cdot)$ denotes the digamma function. The EM equations are iteratively evaluated analogously to the standard EM algorithm [19].

III. Workflow

The main processing and classification pipeline in this paper is similar to previous work [8]. First, a standard dimensionality reduction method, namely principal component analysis (PCA) is used to reduce the spectral dimensionality of the HSRS images. Then, extended morphological attribute profiles (EMAP) is computed as a feature vector. To reduce the dimensionality again, PCA and non-parametric weighted feature extraction (NWFE) [20] are applied as two variants. Next, we fit a GMM to each class and sample from it so that we can generate the synthetic data. Then, the low-dimensional samples are classified with random forests.

The main contribution of this work comes in the GMM fitting block. In order to speed up the GMM computation and make it memory efficient, we use Variational Bayesian method to determine the number of GMM components automatically. As mentioned before, the main advantage of this method is that it does not need many GMMs to be generated a posteriori.

IV. Experimental Setup and Results

Similar to [8], we used two commonly used hyperspectral datasets, namely Pavia Centre and Salinas, which are acquired by ROSIS and AVIRIS sensors, respectively. These datasets are high dimensional. In order to reduce their dimensionality, first, PCA is performed on input data. Then, for both datasets, EMAP is computed on the principal components. For the Pavia Centre dataset, the threshold values of the attributes are similar to [21]. For Salinas dataset, the threshold values are selected similar to [22]. Later, the dimensionality of EMAP data is reduced in one variant with PCA and in another with NWFE to test the effect of both unsupervised and supervised dimensionality reduction algorithms on the results. PCA and NWFE are performed such that 99.9% of the variance is preserved.

For the GMMs, the covariance matrix is constrained to be diagonal. As for the AIC, BIC, average silhouette width and gap methods, we constrain K between 1 and 4 and let these algorithms choose the best model. AIC, BIC, average silhouette width and gap methods are implemented using the Statistics and Machine Learning Toolbox in MATLAB. VB results are obtained via Pattern Recognition and Machine Learning Toolbox (PRMLT) [23] in MATLAB. As mentioned in Section II, we prefer to have a large initial number of components (K). Therefore, K is selected to be 25. Number of synthetic samples, generated to populate the limited training data is 5000 samples per each class for all the experiments.

TABLE I

Performance for different cases using Pavia Centre dataset.

Algorithm	pix per class	AA% (±SD)	OA% (±SD)	Kappa (±SD)	Run time (s) (±SD)
PCA					
AIC	13	85.17 (±1.21)	93.39 (±1.44)	0.9072 (±0.0197)	0.0877 (±0.0115)
	30	88.52 (±0.71)	94.68 (±0.51)	0.9252 (±0.0070)	0.0978 (±0.0054)
BIC	13	85.50 (±1.14)	93.67 (±0.82)	0.9110 (±0.0114)	0.0781 (±0.0017)
	30	88.23 (±1.06)	94.81 (±0.46)	0.9270 (±0.0064)	0.0856 (±0.0025)
avg. silhouette	13	85.10 (±1.27)	93.64 (±1.03)	0.9105 (±0.0142)	0.2013 (±0.0082)
	30	87.61 (±1.14)	94.68 (±0.32)	0.9251 (±0.0045)	0.3521 (±0.0223)
gap	13	83.14 (±1.87)	92.23 (±1.00)	0.8911 (±0.0138)	16.4766 (±0.1400)
	30	85.87 (±2.62)	93.54 (±1.03)	0.9091 (±0.0145)	35.4273 (±0.2511)
VB	13	85.60 (±0.62)	93.52 (±0.40)	0.9090 (±0.0055)	0.0324 (±0.0030)
	30	89.14 (±0.46)	95.15 (±0.42)	0.9317 (±0.0059)	0.0450 (±0.0029)
NWFE					
AIC	13	87.72 (±1.96)	94.75 (±0.96)	0.9260 (±0.0133)	0.0788 (±0.0043)
	30	91.86 (±1.05)	96.41 (±0.53)	0.9493 (±0.0074)	0.0895 (±0.0037)
BIC	13	89.84 (±0.90)	95.65 (±0.66)	0.9387 (±0.0092)	0.0785 (±0.0044)
	30	91.98 (±0.53)	96.50 (±0.45)	0.9506 (±0.0063)	0.0884 (±0.0038)
avg. silhouette	13	88.83 (±0.99)	95.00 (±0.61)	0.9297 (±0.0084)	0.2136 (±0.0103)
	30	91.30 (±0.78)	96.28 (±0.64)	0.9476 (±0.0089)	0.3817 (±0.0377)
gap	13	89.08 (±0.83)	95.30 (±0.61)	0.9338 (±0.0084)	17.5375 (±0.1385)
	30	90.75 (±1.16)	96.01 (±0.61)	0.9437 (±0.0084)	39.1785 (±0.5053)
VB	13	89.60 (±1.37)	96.11 (±0.53)	0.9404 (±0.0075)	0.0328 (±0.0023)
	30	91.55 (±0.62)	96.43 (±0.47)	0.9469 (±0.0065)	0.0471 (±0.0026)

TABLE II

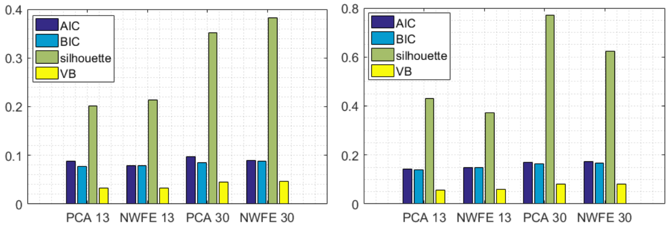
Performance for different cases using Salinas dataset.

Algorithm	pix per class	AA% (±SD)	OA% (±SD)	Kappa (±SD)	Runtime (s) (±SD)
PCA					
AIC	13	91.01 (±0.87)	83.90 (±1.61)	0.8214 (±0.0175)	0.1430 (±0.0060)
	30	92.55 (±0.31)	85.80 (±0.91)	0.8425 (±0.0098)	0.1686 (±0.0069)
BIC	13	90.40 (±0.85)	83.00 (±2.02)	0.8115 (±0.0222)	0.1391 (±0.0071)
	30	92.68 (±0.55)	85.93 (±1.45)	0.8440 (±0.0158)	0.1646 (±0.0065)
avg. silhouette	13	90.50 (±0.72)	82.76 (±1.31)	0.8092 (±0.0141)	0.4293 (±0.0159)
	30	92.14 (±0.42)	85.35 (±1.29)	0.8374 (±0.0140)	0.7702 (±0.0418)
gap	13	90.01 (±1.08)	81.66 (±1.69)	0.7973 (±0.0183)	38.9433 (±0.6351)
	30	91.49 (±0.87)	84.27 (±1.82)	0.8258 (±0.0199)	81.9563 (±0.8802)
VB	13	91.02 (±0.87)	84.07 (±1.60)	0.8235 (±0.0175)	0.0579 (±0.0044)
	30	92.59 (±0.55)	86.00 (±1.01)	0.8447 (±0.0110)	0.0802 (±0.0051)
NWFE					
AIC	13	92.46 (±1.08)	85.93 (±2.33)	0.8435 (±0.0254)	0.1491 (±0.0049)
	30	94.38 (±0.51)	88.42 (±1.30)	0.8715 (±0.0142)	0.1734 (±0.0096)
BIC	13	93.22 (±0.60)	86.99 (±1.28)	0.8557 (±0.0140)	0.1493 (±0.0051)
	30	94.33 (±0.30)	88.90 (±0.64)	0.8767 (±0.0070)	0.1660 (±0.0058)
avg. silhouette	13	93.02 (±0.57)	85.84 (±1.96)	0.8430 (±0.0213)	0.3710 (±0.0184)
	30	93.95 (±0.44)	87.30 (±1.65)	0.8591 (±0.0179)	0.6242 (±0.0246)
gap	13	92.98 (±0.62)	86.60 (±1.05)	0.8514 (±0.0115)	30.4463 (±0.5223)
	30	94.05 (±0.42)	87.93 (±1.13)	0.8659 (±0.0124)	65.7990 (±1.5273)
VB	13	93.26 (±0.67)	87.05 (±1.03)	0.8562 (±0.0112)	0.0610 (±0.0070)
	30	94.09 (±0.55)	88.30 (±1.31)	0.8700 (±0.0144)	0.0818 (±0.0024)

For the classification purpose, we employed the random forest classifier. As suggested by Breiman [24], the number of trees are set to 100 and the number of variable is adjusted to be the square root of the number of data variables. The training samples are selected randomly as 13 and 30 pixels per class. Each experiment is repeated 10 times. The quantitative evaluation is reported using the mean of the Overall Accuracy (OA%), Average Accuracy (AA%), Kappa [25], the average runtime and their standard deviations in the 10 repetitions. The runtime demonstrates the average duration to estimate the optimum number of components in ten iterations and fitting GMM to the training data.

The quantitative evaluation results are demonstrated in Table I and Table II for Pavia Centre and Salinas data sets, respectively. Considering the classification performance, in all the cases using VB generates similar, if not better, results. Besides, in most cases the standard deviation is generally lower for Variational Bayesian, which indicates the more accurate underlying data distribution approximation by VB.

Focusing on the runtime, it can be observed that VB is in average almost two times faster than the AIC and BIC and eight times faster than the average silhouette width method.



(a) (b)

Fig. 1. Runtimes in seconds for (a) Pavia Centre and (b) Salinas.

These timing differences are visualized in the diagram in Fig. 1. The gap method is by about two orders of magnitude slower than the other methods, and therefore is not shown in the plot. Furthermore, we conducted the statistical Wilcoxon signed rank test on both the classification accuracies and the runtimes. Based on this test, the runtime of VB is significantly different (lower) than the other methods while the classification accuracies using all these methods are not statistically significantly different. Since AIC, BIC, silhouette and gap methods select among different models, there is a need to create multiple GMMs, which is not the case for VB. This is the main reason for the big runtime advantage of the Variational Bayesian.

V. Conclusion

One of the most challenging problems in HSRS image classification is the limited availability of labeled data which is tackled by various methods in the literature. In previous work [8], it was proposed to populate the small training data with GMM-based synthetic samples. However, determining the optimal number of components in a GMM is challenging.

In this work, we used Variational Bayesian, which determines the number of mixtures in a GMM by an iterative procedure. We compared the performance of VB with other known methods for determining the number of components, i.e. AIC, BIC, average silhouette width and gap. The quantitative results show that VB yields similar, if not better, performance compared to the other methods. The results using VB are generally more consistent as well. More importantly, Variational Bayesian does not need the clustering algorithm to be executed in advance. This makes the VB memory efficient and drastically reduces the computational cost.

References

[1] S. Valero, P. Salembier, and J. Chanussot, “Hyperspectral image representation and processing with binary partition trees,” *Image Processing, IEEE Transactions on*, vol. 22, no. 4, pp. 1430–1443, 2013.

[2] L. Bruzzone, M. Chi, and M. Marconcini, “A novel transductive SVM for semisupervised classification of remote-sensing images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 11, pp. 3363–3373, 2006.

[3] J. Xia, J. Chanussot, P. Du, and X. He, “Rotation-based support vector machine ensemble in classification of hyperspectral data with limited training samples,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 3, pp. 1519–1531, 2016.

[4] J. Li, X. Huang, P. Gamba, J. M. Bioucas-Dias, L. Zhang, J. A. Benediktsson, and A. Plaza, “Multiple feature learning for hyperspectral image classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 3, pp. 1592–1606, 2015.

[5] M. Sofolahan and O. Ersoy, “Summed component analysis for dimensionality reduction and classification,” Purdue University, Tech. Rep. 445, 2013.

[6] T. Castaings, B. Waske, J. Atli Benediktsson, and J. Chanussot, “On the influence of feature reduction for the classification of hyperspectral images based on the extended morphological profile,” *International Journal of Remote Sensing*, vol. 31, no. 22, pp. 5921–5939, 2010.

[7] A. Kianisarkaleh and H. Ghassemian, “Nonparametric feature extraction for classification of hyperspectral images with limited training samples,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 119, pp. 64–78, 2016.

[8] A. Davari, E. Aptoula, B. Yanikoglu, A. Maier, and C. Riess, “GMM-based synthetic samples for classification of hyperspectral images with limited training data,” *IEEE Geoscience and Remote Sensing Letters*, pp. 1–5, 2018.

[9] H. Akaike, “Information theory and an extension of the maximum likelihood principle,” in *Selected Papers of Hirotugu Akaike*. Springer, 1998, pp. 199–213.

[10] G. Schwarz *et al.*, “Estimating the dimension of a model,” *The annals of statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[11] G. Celeux and G. Soromenho, “An entropy criterion for assessing the number of clusters in a mixture model,” *Journal of classification*, vol. 13, no. 2, pp. 195–212, 1996.

[12] R. Laxhammar, G. Falkman, and E. Sviestins, “Anomaly detection in sea traffic-a comparison of the gaussian mixture model and the kernel density estimator,” *Information Fusion, 2009. FUSION’09. 12th International Conference on*, pp. 756–763, 2009.

[13] T. M. Kodinariya and P. R. Makwana, “Review on determining number of cluster in k-means clustering,” *International Journal of Advance Research in Computer Science and Management Studies*, vol. 1, no. 6, pp. 90–95, 2013.

[14] P. J. Rousseeuw and L. Kaufman, *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley Online Library, 1990.

[15] R. Tibshirani, G. Walther, and T. Hastie, “Estimating the number of clusters in a data set via the gap statistic,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 2, pp. 411–423, 2001.

[16] H. Pang, L. Deng, L. Wang, and M. Fei, “The application of spark-based gaussian mixture model for farm environmental data analysis,” in *Asian Simulation Conference*. Springer, 2016, pp. 164–173.

[17] Z. Li, L. Mihaylova, O. Isupova, and L. Rossi, “Autonomous Flame Detection in Videos with a Dirichlet Process Gaussian Mixture Color Model,” *IEEE Transactions on Industrial Informatics*, vol. 3203, no. c, pp. 1–9, 2017.

[18] D. Williams, X. Liao, Y. Xue, L. Carin, and B. Krishnapuram, “On classification with incomplete data,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 3, pp. 427–436, 2007.

[19] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[20] B.-C. Kuo and D. A. Landgrebe, “Nonparametric weighted feature extraction for classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 5, pp. 1096–1105, 2004.

[21] M. Dalla Mura, J. Atli Benediktsson, B. Waske, and L. Bruzzone, “Extended profiles with morphological attribute filters for the analysis of hyperspectral data,” *International Journal of Remote Sensing*, vol. 31, no. 22, pp. 5975–5991, 2010.

[22] T. Liu, Y. Gu, X. Jia, J. A. Benediktsson, and J. Chanussot, “Class-specific sparse multiple kernel learning for spectral–spatial hyperspectral image classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 12, pp. 7351–7365, 2016.

[23] “Pattern recognition and machine learning toolbox.” [Online]. Available: http://github.com/PRML/PRMLT/

[24] L. Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[25] J. Cohen, “A coefficient of agreement for nominal scales,” *Educational and psychological measurement*, vol. 20, no. 1, pp. 37–46, 1960.