A Learning Scheme for Recognizing Sub-classes from Model Trained on Aggregate Classes

Ranga Raju Vatsavai[1], Shashi Shekhar[2], and Budhendra Bhaduri[1]

[1] Computational Sciences and Engineering Division,
Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA
vatsavairr@ornl.gov, bhaduribl@ornl.gov
[2] Dept. of Computer Science, University of Minnesota
4-192 EE/CS Bldg., 200 Union Street SE, MN 55455
shekhar@cs.umn.edu

Abstract. In many practical situations it is not feasible to collect labeled samples for all available classes in a domain. Especially in supervised classification of remotely sensed images it is impossible to collect ground truth information over large geographic regions for all thematic classes. As a result often analysts collect labels for aggregate classes. In this paper we present a novel learning scheme that automatically learns sub-classes from the user given aggregate classes. We model each aggregate class as finite Gaussian mixture instead of classical assumption of unimodal Gaussian per class. The number of components in each finite Gaussian mixture are automatically estimated. Experimental results on real remotely sensed image classification showed not only improved accuracy in aggregate class classification but the proposed method also recognized sub-classes.

Keywords: Semi-supervised learning, EM, GMM, Remote Sensing.

1 Introduction

Remote sensing, which provides inexpensive, synoptic-scale data with multi-temporal coverage, has proven to be very useful in land cover mapping, environmental monitoring, forest and crop inventory, urban studies, natural and man made object recognition, etc. Thematic information extracted from remote sensing imagery is also useful in a variety spatio-temporal applications. For example, land management organizations and the public have a need for more current regional land cover information to manage resources and monitor land use changes. Likewise, intelligence agencies, such as, National Geospatial Intelligence Agency (NGA), and Department of Homeland Security (DHS), utilizes pattern recognition and data mining techniques to classify both natural and man made objects from large volumes of high resolution imagery.

Image classification, i.e., assigning class labels to pixels, using some discriminant function, is one of the fundamental analysis technique used in remote sensing to generate thematic information. Image classification can be formally

N. da Vitora Lobo et al. (Eds.): SSPR&SPR 2008, LNCS 5342, pp. 967–976, 2008.

defined as finding a function $g(x)$ which maps the input patterns x onto output classes y_i (some times y_i's are also denoted as ω_i or c_i). The main objective is to assign a label (e.g. Water, Forest, Urban) to each pixel in the image to be classified, given corresponding feature vector x_j's in the input image. Depending on the type of supervised learning method used, the objective of a supervised learning could be finding a function $g(x)$ (also called a discriminant function), that divides the input $d-$dimensional feature space into several regions, where each region corresponds to a thematic class y. One such simple function is given by: $x \in y_i$ if $p(y_i|x) > p(y_j|x)\ \forall j \neq i$. That is, the feature vector x belongs to class y_i if $p(y_i|x)$ is the largest. Even though it sounds simple, this assignment problem is very difficult. There is no single algorithm which will correctly classify any given image. Multi-spectral image classification is still an open problem.

Collecting ground truth (labels) data over large geographic regions is costly, time consuming, and poses several other practical problems. Given these practical limitations, often the analyst groups the relevant classes and collects labels only for those aggregate (grouped) classes. Typical examples are forest, agriculture, urban, and other land-use classes. Usually, the analyst given forest class may contains samples from all types of forests, such as hardwoods, conifer, etc., each of which can be described by a unique statistical distribution. These distributions are clearly identifiable in many image classification problems (depending on the spectral resolution), though in some cases these finer classes may be highly overlapping (in low spectral resolution images).

Aggregating several classes into a single class such as forest poses two problems, first it violates the common assumption that each class is unimodal, secondly the estimated parameters could be wrong. Since we are combining distinctly identifiable classes, the estimated covariance matrix is large, as it accounts for both inter-class covariance and intra-class covariances (of component classes). It is very important to estimate covariance matrix accurately because even for a fixed means, it can be shown that increase in variance of any one class (keeping means fixed), leads to the increase in probability of error. This observation motivated us to develop a new classification scheme which relaxes the common assumption that the class has to be a unimodal distribution. Instead, we assume that each class is a finite mixture model.

Related Work and Our Contributions: Supervised methods are extensively used in remote sensing imagery classification [9,4]. Finite mixture model parameter estimation [5] for unlabeled samples is also investigated extensively under various disguises. For example, semi-supervised learning approaches are very close to the approach we presented in Section 3. Well-known studies in this area include, but not limited to [7,3,8,1,11]. Model selection approaches have also been extensively studied [12,6] and used in finding the number of clusters [2].

In this work we clearly identified an important practical problem in supervised classification of remotely sensed imagery. We developed a novel learning algorithm which takes user defined aggregate classes and automatically discovers sub-classes within each aggregate class. Each aggregate class is modeled as Gaussian mixture and is split into sub-classes (components) using a Gaussian splitting criteria. The resulting classifier showed not only improvement in overall classification accuracy but also recognized finer classes which analyst is always interested in, but sufficient ground truth data cannot be collected for the finer classes.

The rest of this paper is organized as follows. In Section 2, we provide a basic statistical framework for Bayesian classification and maximum likelihood based parameter estimation. We present our proposed learning scheme and parameter estimation in Section 3. Experimental results are given in Section 4. Related work is presented in Section 5, followed by conclusions and future directions in Section 6.

2 Statistical Classification Framework

In the classification of a remote sensing images, our objective is to assign a class label (y) to each pixel (x) based on certain decision criterion. Maximum likelihood classification (MLC) and maximum a posteriori (MAP) classification are two of the most widely used classifiers in remote sensing. Assuming the training samples were generated by a multivariate normal or Gaussian density, we can write the decision rule for maximum a posterior (MAP) classifier as following:

$$g_i(x) \;=\; \ln\; P(y_i) - \frac{1}{2}\ln|\Sigma_i| - \frac{-1}{2}(x-\mu_i)^t|\Sigma_i|^{-1}(x-\mu_i) \qquad (1)$$

If we don't have *a priori* knowledge about the classes then we can drop $P(y_i)$ term in the above equation and the resulting decision rule is known as maximum likelihood classification (MLC). The covariance matrix Σ plays a key role in discriminant analysis. Covariance accounts for the shape (size and orientation) of classes in the feature space. The effectiveness of ML/MAP classification depends on the quality of the estimated parameter vector Θ (i.e., mean vector μ and the covariance matrix Σ for each class) from the training samples. Using a well-known parameter estimation technique, maximum likelihood estimation (MLE), we can obtain the parameters μ and Σ as following:

$$\hat{\mu} = \frac{1}{n}\sum_{k=1}^{n} x_k; \quad \hat{\Sigma} = \frac{1}{n}\sum_{k=1}^{n}(x_k-\hat{\mu})(x_k-\hat{\mu})^t. \qquad (2)$$

2.1 Limitations of MLC and MAP

One of the classical assumptions in supervised (statistical) classification is that the classes are unimodal. We now test the impact of violation of this constraint through a simulated example. We generated bivariate Normal samples (150 per class) for three distinctly identifiable classes using the parameters given in the Table 1 and 2. Maximum likelihood estimates from the simulated samples were also summarized along with the original parameters.

Table 1. Simulation Parameters (Means)

	Given		No of	Estimated (MLE)	
Classes	X1	X2	Samples	X1	X2
C_1	55.00	25.00	150.00	54.93	24.50
C_2	80.00	50.00	150.00	79.58	50.23
C_3	50.00	40.00	150.00	49.74	39.55

Table 2. Simulation Parameters (Covariance)

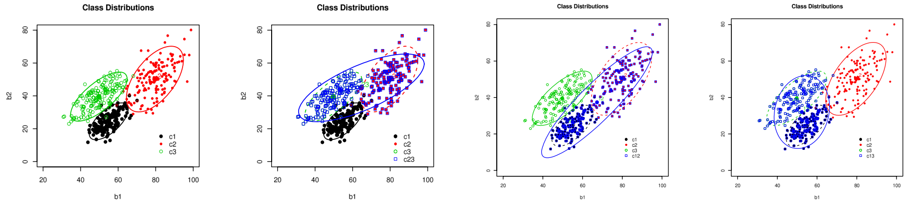
	C_1		C_2		C_3	
Features	X1	X2	X1	X2	X1	X2
X_1	30.00	25.00	60.00	40.00	60.00	50.00
X_2	25.00	40.00	40.00	90.00	50.00	70.00
Estimated Parameters (MLE)						
X_1	27.03	19.10	57.66	43.43	56.81	40.60
X_2	19.10	31.58	43.43	100.03	40.60	55.84

Table 3. ML Estimates of Aggregated Class

C_{23}		X1	X2
Mean		64.66	44.89
Covariance	X1	280.49	121.80
	X2	121.80	106.26

Let us now assume that classes 2 and 3 are sub-classes of an aggregate class C_{23}, i.e., analyst gave a single label to all the samples generated from the classes C_2 and C_3. The new estimates of the aggregate class C_3 are given in the Table 3. For understanding the distribution (and interaction) of original classes (C_1, C_2, C_3) and aggregate classes (C_1, C_{23}), we have generated the bivariate density plots which are shown in Figure 1.

The overlap between classes (C_1, C_2, C_3) is almost negligible (see Figure 1(a)). However, aggregation of classes C_2, C_3 into C_{23} has greatly increased its overlap with class C_1 (see Figure 1(b)). We have seen previously that this overlap directly accounts for the probability of error, p_E. One can expect to improve the classification accuracy, if somehow the original classes (C_2, C_3) which gave raise to aggregate class C_{23} can be automatically discovered. Also, there is a great need for finer class discovery from remotely sensed images, and precisely this is what our proposed algorithm tries to accomplish.



(a) Original Distribution

(b) C_2, C_3 aggregated into C_{23}

(c) C_1, C_2 aggregated into C_{12}

(d) C_1, C_3 aggregated into C_{13}

Fig. 1. Interaction Between Finer and Aggregate Classes

3 Learning To Discover Sub-classes

Basic idea behind the proposed algorithm is very simple. Instead of assuming that each class is a unimodal multivariate Gaussian, we assume that the samples from each class are generated by finite Gaussian mixture. There are two subproblems associated with this assumption: First, we don't have labels for any of the component (sub-class) so that we can employ regular MLE technique to estimate the parameters of each component; second, we don't know how many components (sub-classes) are there in this finite mixture model. We address these two problems in the following two sub-sections.

3.1 Estimating Finite Mixture Parameters

First problem is solved by assuming that the training samples in each (aggregate) class were generated a mixture density and then estimate parameters of mixture density for arbitrary number of components using the expectation maximization (EM) algorithm. The EM algorithm consists of two steps, called the E-step and and M-step as given below.

E-Step: For multivariate normal distribution, the expectation $E[.]$, which is denoted by p_{ij}, is the probability that Gaussian mixture j generated the data point i, and is given by:

$$p_{ij} = \frac{\left|\hat{\Sigma}_j\right|^{-1/2} e^{\left\{-\frac{1}{2}(x_i-\hat{\mu}_j)^t \hat{\Sigma}_j^{-1}(x_i-\hat{\mu}_j)\right\}}}{\sum_{l=1}^{M} \left|\hat{\Sigma}_l\right|^{-1/2} e^{\left\{-\frac{1}{2}(x_i-\hat{\mu}_l)^t \hat{\Sigma}_l^{-1}(x_i-\hat{\mu}_l)\right\}}} \tag{3}$$

M-Step: The new estimates (at the k^{th} iteration) of the model parameters in terms of the old parameters are computed using the following update equations:

$$\hat{\alpha}_j^k = \frac{1}{n}\sum_{i=1}^{n} p_{ij} \tag{4}$$

$$\hat{\mu}_j^k = \frac{\sum_{i=1}^{n} x_i p_{ij}}{\sum_{i=1}^{n} p_{ij}} \tag{5}$$

$$\hat{\Sigma}_j^k = \frac{\sum_{i=1}^{n} p_{ij}(x_i - \hat{\mu}_j^k)(x_i - \hat{\mu}_j^k)^t}{\sum_{i=1}^{n} p_{ij}} \tag{6}$$

The EM algorithm iterates over these two steps until convergence is reached. We can now put together these individual pieces into the following algorithm (Table 4) which computes the parameters for each component in the finite Gaussian mixture model that generated our training data D (without any labels).

3.2 Estimating the Number of Component of a Finite Gaussian Mixture

We now address our second problem, i.e., we don't know how many components (sub-classes) are there in each (aggregate) class. As with estimating model parameter for finite Gaussian mixture model, we assume that the training dataset D is generated by a finite Gaussian mixture model, but we don't know either the number of components or the labels for any of the sub-component. In previous

Table 4. Algorithm for Computing Parameter of Finite Gaussian Mixture Model Over Unlabeled Training Data

Inputs: D_j, training dataset (no labels for sub-classes) for any aggregate class y_j; M, the number of sub-classes in the corresponding aggregate class.

Initial Estimates: Do clustering by K-Means, and estimate initial parameter using MLE, to find $\hat{\theta}$ (see Equations 2)

Loop: While the complete data *log-likelihood* improves:

  E-step: Use current classifier to estimate the class membership of each unlabeled sample, i.e., the probability that each Gaussian mixture component generated the given sample point, p_{ij} (see Equation 3).

  M-step: Re-estimate the parameter, $\hat{\theta}$, given the estimated Gaussian mixture component membership of each unlabeled sample (see Equations 4, 5, 6)

Output: Parameter vector Θ.

section, we devised an algorithm to find parameters by assuming a M-component finite Gaussian mixture model. In general, we can estimate parameters for any arbitrary M-component model, as long as there are sufficient number of samples available for each component and the covariance matrix does not become singular. Then the question remains, which M-component model is better? This question is addressed in the area of model selection, where the objective is to chose a model that maximizes a cost function. There are several cost functions available in the literature, most commonly used measures are Akaike's information criterion (AIC), Bayesian information criteria (BIC), and minimum description length (MDL). The common criteria behind these models is to penalize the models with additional parameters, so BIC and AIC based model selection criteria follows the principal of parsimony. In this study we considered BIC as a model selection criteria, which is also takes the same form as MDL. We also chose BIC, as it is defined in terms of maximized log-likelihood which any way we are computing in our parameter estimation procedure defined in the previous section. BIC can be defined as

$$BIC = MDL = -2 \log L(\Theta) + m \log(N) \tag{7}$$

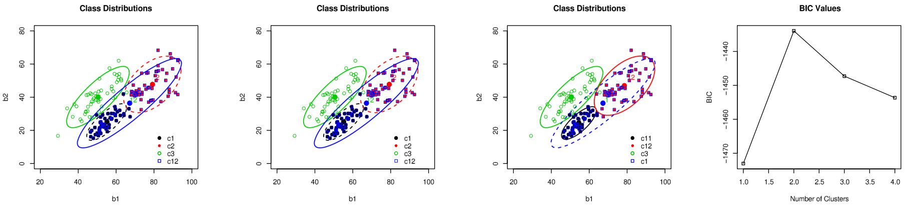
where N is the number of samples and m is the number of parameters. We now describe our BIC based model selection criteria to determine the number components in each aggregate class. First, we take the aggregate class and split it into two Gaussians at a time using the Gaussian splitting criteria specified in [10]. Then the parameters of this new mixture model are estimated using the algorithm 4. This process is recursively applied for a fixed number times or BIC is minimized. We then repeat the algorithm for each (aggregate) class in the original classification problem. At the end of each iteration we have parameters for each sub-class within an aggregate class. We can now apply MLC/MAP in two ways. First, we modified MLC/MAP to output both aggregate classes (original analyst given classes) and as well sub-classes which were discovered automatically using the procedure just described. We can combine the finer classes (predicted) into the corresponding aggregate class in order to find the aggregate class classification accuracy.

4 Experimental Results

We have conducted several experiments using simulated and as well as the real dataset.

Dataset 1: The objective of first experiment on simulated data is to see performance of proposed method in aggregate class classification and as well as finer class classification. We used the parameters listed in Table 1 to generate two distinct datasets. First dataset consisted of 150 samples at 50 samples per class, and second dataset consisted of 450 samples at 150 samples per class. We used first dataset for training and the second dataset for testing. We conducted following three experiments.

Experiment 1: MAP classification was carried out using all three classes, whose distribution is shown in Figure 2(a). Test accuracy of MAP is shown in Table 5.



(a) Original Distribution

(b) C_1, C_2 aggregated into C_{12}

(c) Sub-classes C_{11}, C_{12} discovered from C_{12}

(d) BIC Values For Different Number of Clusters

Fig. 2. Interaction Between Finer, Aggregate, and Newly Discovered Classes

Table 5. Accuracy (All Classes)

G.Truth	C_1	C_2	C_3	P.Acc
C_1	141.00	4.00	5.00	94.00
C_2	1.00	147.00	2.00	98.00
C_3	1.00	1.00	148.00	98.67
U.Acc	98.60	96.71	95.48	(OA) 96.89

Table 6. Accuracy ($C_1, C_2 \rightarrow C_1$)

G.Truth	C_1	C_3	P.Acc
C_1	281.00	19.00	93.67
C_3	2.00	148.00	98.67
U.Acc	99.29	88.62	(OA) 95.33

Table 7. Accuracy ($C_1 \rightarrow C_1, C_2$)

G.Truth	C_1	C_2	C_3	P.Acc
C_1	132.00	11.00	7.00	88.00
C_2	1.00	147.00	2.00	98.00
C_3	1.00	1.00	148.00	98.67
U.Acc	98.51	92.45	94.27	(OA) 94.89

Table 8. Accuracy ($C_1 \rightarrow C_1, C_2 \rightarrow C_1$)

G.Truth	C_1	C_3	P.Acc
C_1	291.00	9.00	97.00
C_3	2.00	148.00	98.67
U.Acc	99.32	94.27	(OA) 97.56

Table 9. Accuracy (Aggregate Classes)

G. Truth	1	2	3	4	P. Acc.
Forest(1)	1475.00	9.00	28.00	0.00	97.55
Ag.(2)	90.00	142.00	2.00	0.00	60.68
Urban(3)	0.00	0.00	45.00	0.00	100.00
Wetlands(4)	18.00	0.00	2.00	34.00	62.96
Users Acc.	93.18	94.04	58.44	100.00	91.92

Table 10. Accuracy (Sub-Classes)

GT	1	2	3	4	P. Acc.
(1)	1448.00	13.00	51.00	0.00	95.77
(2)	14.00	214.00	6.00	0.00	91.45
(3)	0.00	0.00	45.00	0.00	100.00
(4)	3.00	0.00	13.00	38.00	70.37
UA.	98.84	94.27	39.13	100.00	94.58

Experiment 2: Classes C_1, C_2 were combined into one aggregate class and class C_3 remained untouched. Resulting new class distributions were shown in Figure 2(b). Test accuracy of MAP using aggregated class is shown in Table 6.

Experiment 3: Our new algorithm was applied on the dataset generated in Experiment 2. We tested both aggregate classification performance and as well as the finer class performance. Newly discovered classes were shown in Figure 2(c). Test accuracy of MAP using newly discovered classes is shown in Table 7, and the corresponding aggregated class accuracy is shown in Table 8.

Dataset 2: In this experiment we used a spring Landsat 7 scene, taken on May 31, 2000 over the Cloquet town located in Carlton County of Minnesota state. The training dataset consisted of sixty plots and four aggregate classes, namely, Forest(1), Agriculture(2), Urban(3), and Wetlands(4). We have an independent test dataset consisting of 205 plots. Feature vectors were extracted from the Landsat image (6-dimensional) by placing a 3×3 window at each of these plots. Maximum likelihood classification is carried out using the conventional approach and as well as the proposed approach. The results were summarized in the following contingency tables (or error matrices).

Table 9 gives MLC accuracy using the standard approach and Table 10 provides the classification accuracy obtained by our proposed method. We regrouped the sub-classes into the corresponding aggregate classes for testing the accuracy using same test dataset (consisting of four aggregate classes).

4.1 Analysis

Accuracy assessment on simulated data shows interesting results. Bivariate density plot shown in Figure 2(a) and as well as hight test accuracy (Table 5) shows that the three classes were clearly separable. Aggregation of classes $C_1, C_2 \rightarrow C_1$ has increased the overlap between the aggregate class C_1 and C_3 (see Figure 2(b)), and this overlap has resulted in more classification error (Table 6) as compared to finer class classification error. Our proposed algorithm on this aggregate data has discovered two sub-classes in the aggregate class C_1 (see Figure 2(c)) and the corresponding BIC value (for number of clusters = 2) is maximum (Figure 2(d)). The test accuracy of MAP classifier trained on newly discovered classes is given in the Table 7 and the corresponding aggregated class accuracy is shown in Table 8. First, comparison of this accuracy table with original classification accuracy (Table 5) reveals that the sub-classes discovered closely corresponds to the original classes. Second, the aggregate (i.e., predicted sub-classes were merged) classification accuracy with our new scheme is higher than the MAP on original aggregate class classification (compare with Table 6). This study revels that our new algorithm not only discovered sub-classes that are close to the original fine classes (without providing any labeled training data) but also improved classification accuracy of original aggregate classes.

Let us now compare the classification error matrices (Table 9 and Table 10) obtained on the real dataset. From these two tables, we can see that our new procedure improved overall classification accuracy (OA) for the same training dataset without any additional (sub-class related training) information. In addition the new procedure automatically discovered sub-classes within each aggregate class. In this (aggregate class) training dataset, our new procedure discovered four additional component in the forest class, two additional components in the agriculture class, and two additional components in the urban class. Our preliminary investigation into the newly discovered sub-classes reveled very interesting information. The four classes discovered roughly corresponds to the following information classes: upland hardwood, lowland hardwood, upland conifer, and lowland conifer forests. There is a great demand for such additional information in many real world applications.

5 Conclusions

We identified an important practical classification problem that requires knowledge discovery approaches for automatically discovering the sub-classes from the aggregate classes. We developed a new classification scheme that automatically discovers the sub-classes from the user given aggregate classes, without any additional labeled training data for sub-classes. In addition, the procedure showed improvement in the classification of aggregate classes as well. This improvement can be attributed to the fact that the aggregate classes tend to increase the overlap between class distributions. Our preliminary investigation also showed a strong correspondence between sub-classes and true information classes. Further research is needed to automatically (or with minimal efforts) label these sub-classes. We are investigating the semantic relationships between various information classes that are common in this domain which might help to automatically label these sub-classes.

Acknowledgments

We would like to thank our former collaborators Thomas E. Burk, Jamie Smedsmo, Ryan Kirk and Tim Mack at the University of Minnesota for useful comments and inputs into this research. The comments of Eddie Bright, Phil Coleman, and Veeraraghavan Vijayraj, have greatly improved the technical accuracy and readability of this paper. Prepared by Oak Ridge National Laboratory, P.O. Box 2008, Oak Ridge, Tennessee 37831-6285, managed by UT-Battelle, LLC for the U. S. Department of Energy under contract no. DEAC05-00OR22725.

References

1. Cozman, F.G., Cohen, I., Cirelo, M.C.: Semi-supervised learning of mixture models. In: Twentieth International Conference on Machine Learning, ICML (2003)
2. Figueiredo, M.A.T., Jain, A.K.: Unsupervised selection and estimation of finite mixture models. In: Pattern Recognition, 2000. Proceedings. 15th International Conference on 2000, vol. 2, pp. 87–90 (2000)
3. Goldman, S., Zhou, Y.: Enhancing supervised learning with unlabeled data. In: Proc. 17th International Conf. on Machine Learning, pp. 327–334. Morgan Kaufmann, San Francisco (2000)
4. Jensen, J.R.: Introductory Digital Image Processing, A Remote Sensing Perspective. Prentice Hall, Upper Saddle River (1996)
5. Mclachlan: Mixture Models: Inference and Applications to Clustering. CRC, New York (1987)
6. Miloslavsky, M., van der Laan, M.J.: Fitting of mixtures with unspecified number of components using cross validation distance estimate. Comput. Stat. Data Anal. 41(3-4), 413–428 (2003)
7. Mitchell, T.: The role of unlabeled data in supervised learning. In: Proceedings of the Sixth International Colloquium on Cognitive Science, San Sebastian, Spain (1999)
8. Nigam, K., McCallum, A.K., Thrun, S., Mitchell, T.M.: Text classification from labeled and unlabeled documents using EM. Machine Learning 39(2/3), 103–134 (2000)
9. Richards, J.A., Jia, X.: Remote Sensing Digital Image Analysis. Springer, New York (1999)
10. Sankar, A.: Experiments with a gaussian merging-splitting algorithm for hmm training for speech recognition. In: Proceedings of the Broadcast News Transcription and Understanding Workshop, pp. 99–104 (1998)
11. Shahshahani, B., Landgrebe, D.: The effect of unlabeled samples in reducing the small sample size problem and mitigating the hughes phenomenon. IEEE Trans. on Geoscience and Remote Sensing 32(5) (1994)
12. Xuelei, Lei, X.U.: Investigation on several model selection criteria for determining the number of clusters. Neural Information Processing - Letters and Reviews 4(1), 139–148 (2004)