7-1-1998

UNSUPERVISED CLASSIFICATION AND CHOICE OF CLASSES: BAYESIAN APPROACH

R. L. Kashyap
Purdue University School of Electrical and Computer Engineering

Srinivas Sista
Purdue University School of Electrical and Computer Engineering

UNSUPERVISED CLASSIFICATION AND CHOICE OF CLASSES: BAYESIAN APPROACH

R. L. KASHYAP
SRINIVAS SISTA

TR-ECE 98-12
JULY 1998

SCHOOL OF ELECTRICAL
AND COMPUTER ENGINEERING
PURDUE UNIVERSITY
WEST LAFAYETTE, INDIANA 47907-1285

UNSUPERVISED CLASSIFICATION AND CHOICE OF CLASSES: BAYESIAN APPROACH [1]

R. L. Kashyap and Srinivas Sista

School of Electrical & Computer Engineering
1285 Electrical Engineering Building
Purdue University
West Lafayette, IN 47907-1285

'This work has been partially supported by National Science Foundation under contract IRI 9619812 and the office of Naval Research under contract N00014-91-J-4126.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

We have given a solution to the problem of unsupervised classification of multidimensional data. Our approach is based on Bayesian estimation which regards the number of classes, the data partition and the parameter vectors that describe the density of classes as unknowns. We compute their MAP estimates simultaneously by maximizing their joint posterior probability density given the data,. The concept of partition as a variable to be estimated hasn't been considered. This formulation also solves the problem of validating clusters obtained from various methods. Our method can also incorporate any additional information about a class while assigning its probability density. It can also utilize any available training samples that arise from different classes.

We provide a. descent algorithm that starts with an arbitrary paatition of the data, and iteratively computes the MAP estimates. We also focus on robust regression which is a special case of unsupervised classification with two classes; inliers and outliers.

The problem of intensity image segmentation is posed as an unsupervised classification problem and solved using the Bayesian formulation in a multiscale set up. The proposed method is also applied to data set'sthat occur in statistical literature and target tracking. The results obtained demonstrate the power of Bayesian approach for unsupervised classification.

Keyworcls: Unsupervised classification, Bayesian estimation, cluster validation, robust regression, image segmentation.

1. INTRODUCTION

In unsupervised classification, the given data $Z = \{\boldsymbol{z}_i, i = 1, \cdots, N\}, \boldsymbol{z}_i \in \mathrm{R}^{\mathrm{m}}$ has to be partitioned into mutually exclusive and totally inclusive subsets of Z namely $\mathrm{c} = \{\boldsymbol{c}_1, \cdots, \boldsymbol{c}_s\}, \boldsymbol{c}_k \subseteq Z$ so that all the members belonging to a class are close to each other in some sense. The choice of s is itself a problem. The solution shoulcl include a compact description of each class so that a new unlabeled data point can be classified easily. Next the methodology shoulcl include the validation of the partition, i.e does the given partition adequately explain the data? Given two different partitions, which one of them gives a better explanation of the data?

When s is fixed, partitions are given by the so called clustering algorithms [2, 3, 4, 5]. It is clearly understood that these algorithms are designed to generate good partitions; there is no guarantee that the partition really explains the data. The criterion function used in the algorithm:; focus only on the centroid of clusters, not on the shape and orientation of the clusters. There is an extensive literature [2, 4, 7, 8, 27, 28, 30] on cluster validation; which includes the choice of s, the number of classes. Further, the criterion function usually has numerous local minima and many methods are satisfied with obtaining one arbitrary local minimum. Currently there is no method to compare different cluster sets derived for the same data obtained from different methods. We will discuss the literature review in a later section.

In the Bayes approach developed here the partition $\boldsymbol{c}$ is itself regarded as a variable to be chosen from the appropriate space. When s is known, $\mathrm{c} = \{\boldsymbol{c}_1, \ldots, \mathrm{c},)$, $\mathrm{c} \in \Omega_{s,s}$, the set of all partitions of set Z so that none of the sets $\boldsymbol{c}_k$ in $\boldsymbol{c}$ are null. When s is not specified, $s \leq$ *so*, then $\mathrm{c} = \{\boldsymbol{c}_1, \cdots, \boldsymbol{c}_{s_0}\}, \mathrm{c} \in \Omega_{s_0}$, the set of all partitions of Z into s_0 subsets. The members of class k are described by the probability density $p_k(\boldsymbol{z}_i \mid \boldsymbol{\theta}_k)$, p_k is a known function and $\boldsymbol{\theta}_k$ is a vector parameter whose values have to be determined, $\boldsymbol{\theta}_k \in R^{n_k}$. The unknowns are $\{c = \{\boldsymbol{c}_1, \cdots, \boldsymbol{c}_s\}, \boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_s\}\}$, when s is known. Using the Bayes formalism, the maximum aposteriori probability(MAP) estimates of {c,8) are obtained by maximizing the posterior probability of 8 and c given Z. Since the prior distribution of both c and 8 are assumed to be uniform, the MAP estimates (c'. 8') are given by

$$\{c^*,8') = \operatorname*{Arg\,min}_{\boldsymbol{c},\boldsymbol{\theta}} J(c,8) \tag{1.1}$$

where

$$J(\boldsymbol{c},\boldsymbol{\theta}) = -2\sum_{k=1} \sum_{\boldsymbol{z}_i \in \boldsymbol{C}_k} \ln p_k(\boldsymbol{z}_i \mid \boldsymbol{\theta}_k) \tag{1.2}$$

Even though the computation in (1.2) may appear to be time consuming because of the discrete variable $\boldsymbol{c}$, in fact the computation is much less than that with traditional clustering algorithms leading to much superior results. For illustration lei; $p_k(\boldsymbol{z}_i \mid \boldsymbol{\theta}_k) \sim$ Gauss$(\boldsymbol{\phi}_k, \boldsymbol{r}_k), \boldsymbol{\theta}_k = \{\boldsymbol{\phi}_k, \boldsymbol{r}_k\}$. Then

$$J(\boldsymbol{c},8) = \sum_{k=1}^{s} \sum_{\boldsymbol{z}_i \in \boldsymbol{C}_k} (\boldsymbol{z}_i - \boldsymbol{\phi}_k)^T \boldsymbol{r}_k^{-1} (\boldsymbol{z}_i - \boldsymbol{\phi}_k) + \ln |\det \boldsymbol{r}_k| + m \ln 2\pi \tag{1.3}$$

Minimizing $J(\boldsymbol{c},8)$ w.r.t 8 fixing c is straight forward and well known. Similarly minimizing $J(\boldsymbol{c},8)$ w.r.t c fixing 8 is again simple involving no more than data comparisons, i.e, we assign $\boldsymbol{z}_i$ to $\boldsymbol{c}_k$ if $(-2\ln p_u(\boldsymbol{z}_i \mid \boldsymbol{\theta}_u))$ has least value for $u =$ I;. By minimizing $J(\boldsymbol{c},8)$ alternately w.r.t c and w.r.t 8 we get a local minimum of $J(\boldsymbol{c},8)$ in a finite number of iterations. By starting with different initial partitions c generated with the aid of random numbers, we can obtain several local minima of $J(\boldsymbol{c},8)$ and choose the best local minimum.

The criterion in (1.3) has some similarity to the criterion used in some clustering algorithms. This aspect is discussed in section VII. One feature of the criterion function is that it has numerous local minima. In literature, the authors seem to be satisfied with one arbitrary local minimum without searching for the global minimum. The Bayes approach allows us to estimate s, the number of classes given that $s \leq s_0$. Correspondingly the best partition c has to be searched in the space $\boldsymbol{c} \in \Omega_{s_0}$. Bayes approach also solves the problem of partition comparison and partition or cluster validation. Two different partitions c and $\boldsymbol{c}'$ involving different values of s can be compared by computing the ratio of the corresponding posterior probabilities $P(\boldsymbol{c} \mid Z)$ and $P(\boldsymbol{c}' \mid Z)$.

In many classification problems, we may have additional information on the classes which are not often utilizecl. For instance, we may know that in a particular class, all the members $\boldsymbol{z_i}$ are clustered tightly around a straight line or a convex curve or a 2-D plane. This information can be incorporated in assigning the probability density function p_k associated with the class. Again our method can successfully utilize any available training samples i.e sets $Z^{(k)}, k = 1, \cdots, s$ where all the members of $Z^{(k)}$ are known to be from the k^{th} class. These are included in the data set Z, except that the class assignments art: not altered.

Next we formulate the segmentation of 2D image as an unsupervised classification problem. This problem has two distinguishing features. First of all, the members of each class may have markovian dependency. More importantly the number of points 'in the data set Z may he very large. A typical 256 x 256 image will have $256^2 = 65536$ data points. The segmentation has to be carried out at different scales, as described later. The key idea here also is the extensive search of the c-space. Currently available clustering algorithms cannot handle the image segmentation problems where we know only the intensities of all the pixels. In recent years, stochastic model-based image segmentation methods have gained importance. In these methods, the labels of the image pixels are modeled as a Markov random field and the segmentation is computed by estimating the best label for each pixel in an approximate MAP set up.

We also focus on a closely related problem namely robust estimation and robust regression [25, 26]. It is a special case of the unsupervised classification problem with two classes. The members of the two classes are labeled as *inliers* and *outliers;* outliers being the noise or corrupted data, the class label of each point being unknown. The aim is to estimate the parameters associated with the density of the inliers. Usually the parameters of the outliers or the classification problem itself are ignored. The robust regression methods are also used in image processing literature extensively [11, 13, 13]. M-estimators were applied in several important computer vision problems such as pose estimation, edge detection, hierarchical surface fitting etc. High breakdown point regression methods are desirable in computer vision and M-estimators have low breakdown points. The connection between the robust estimation and unsupervised classification problem has not been appreciated in the statistical literature. The popular method like Least Median Squares(LMedS)[24], fails when the number of data points in the outlier subset is greater than that of the inliers. In image processing applications, when a window is centered on a corner, or on a boundary between two regions, we cannot guarantee that 51% of the pixels belong to the surface in the window center. Our method does not suffer from such a restriction. Further the computation needed for the solution in our method is much less than that of the methods like Least Median Squares which need Monte-Carlo procedures since the function to he minimized has no continuity, let alone derivatives. Finally the Bayes method gives an expression for the covariance of estimates.

2. OPTIMAL PARTITION WITH A GIVEN NUMBER OF CLASSES

Let the data set be $Z = \{\boldsymbol{z}_i, \text{i} = 1.\cdots, N\}, \boldsymbol{z}_i \in \text{R}^{\text{m}}$ whose members are statistically independent. Let s be the number of distinct classes in Z, s is known to us. Let the s associated probability densities be $p_k(\boldsymbol{z}_i \mid \boldsymbol{\theta}_k), \boldsymbol{\theta}_k \in \text{R}^{\text{nk}}, k = 1.\cdots,\text{s}$. Let the set c $= \{\boldsymbol{c}_1, \cdots, \boldsymbol{c}_s\}$ be a partition of Z into s classes, each $\boldsymbol{c}_k$ being a subset of Z. the density p_k describing the members of the set $\boldsymbol{c}_k$

$$\begin{gathered} \boldsymbol{c}_k \subseteq Z,\ \forall k = 1, \cdots, s;\ \boldsymbol{c}_i \cap \boldsymbol{c}_j = \text{Null}, i \neq j \\ \bigcup_{k=1}^{s} \boldsymbol{c}_k = \text{Z}. \quad \boldsymbol{c}_k \neq \text{ Null } \forall k = 1, \cdots, s. \end{gathered} \tag{2.1}$$

Let $\Omega_{s,s}$ be the set of all possible distinct partitions of Z obeying (2.1). c and $\boldsymbol{c}'$ are different partitions if they are different sets. The number of distinct partitions, which is the cardinality of $\Omega_{s,s}$ is

$$\#\Omega_{s,s} = \frac{1}{s!} \sum_{i=0}^{s} (-1)^i \binom{s}{i} (s-i)^N \approx \frac{s^N}{s!} \tag{2.2}$$

c and $\boldsymbol{\theta}_k$, k $= 1, \cdot \cdot, s$ are the variables to be estimated. We regard c $\in \Omega_{s,s}, \boldsymbol{\theta}_k \in R^{n_k}$, k $=$ $1, \cdot . \cdot, s$ as independent random variables. $P(\boldsymbol{c})$, the prior probability associated with c is same for all c;

$$P(\boldsymbol{c}) = \frac{1}{\#\Omega_{s,s}}, \quad \forall \boldsymbol{c} \in \Omega_{s,s} \tag{2.3}$$

Let 8 $= \{\boldsymbol{\theta}_1, \cdot . ., \boldsymbol{\theta}_s\}$. Let $p(\boldsymbol{\theta}_k)$ he the prior probability density of $\boldsymbol{\theta}_k$, so that the probability density of each component of $\boldsymbol{\theta}_k$ is uniform for all its components.

Let $\{\hat{c}, \boldsymbol{\theta}_k, k = 1, \cdot . \cdot, s\}$ be estimates of c and $\boldsymbol{\theta}_k$, k $= 1, \cdot . ., s$, based on Z. Let the loss function $L(\boldsymbol{c}, \hat{\boldsymbol{c}}, \boldsymbol{\theta}, \boldsymbol{\theta})$ be zero-one. zero if c $= \hat{\boldsymbol{c}}$ and $\boldsymbol{\theta} = \boldsymbol{\theta}$, one otherwise. Then the risk function is

$$R(\hat{c}, \hat{\boldsymbol{\theta}}) = \sum_{\boldsymbol{c} \in \Omega_s} \int L(\boldsymbol{c}, \hat{\boldsymbol{c}}, \boldsymbol{\theta}, \hat{\boldsymbol{\theta}})\ p(Z \mid \boldsymbol{c}, \boldsymbol{\theta})\ P(\boldsymbol{c}) \left(\prod_{k=1}^{s} p(\boldsymbol{\theta}_k) \right) dZ\ d\boldsymbol{\theta}_1 \cdots d\boldsymbol{\theta}_s. \tag{2.4}$$

The MAP estimates $(c*\beta^*)$ minimize $R(\mathbf{c},\boldsymbol{\theta})$,

$$(c*\beta'') = \text{Arg} \max_{\mathbf{c},\boldsymbol{\theta}} P(\mathbf{c}, 8 \mid Z). \tag{2.5}$$

Since the priors of 8 and $\mathbf{c}$ are uniform

$$(c'',\boldsymbol{\theta}^*) = \text{Arg} \max_{\mathbf{c},\boldsymbol{\theta}} P(Z \mid \mathbf{c}, 8). \tag{2.6}$$

Since the data Z is independent, the joint density of Z has the following form:

$$P(Z \mid \mathbf{c},\boldsymbol{\theta}) = \prod_{k=1}^{s} \left(\prod_{\boldsymbol{z_i} \in \boldsymbol{C}_k} p_k(\boldsymbol{z_i} \mid \boldsymbol{\theta}_k) \right). \tag{2.7}$$

Let,

$$\begin{aligned} f_k(\boldsymbol{z_i},\boldsymbol{\theta}_k) &= -2 \ln p_k(\boldsymbol{z_i} \mid \boldsymbol{\theta}_k) & (2.8)\\ J(\mathbf{c} = \{\mathbf{c}_1,\cdots,\mathbf{c}_s\},\boldsymbol{\theta} = \{\boldsymbol{\theta}_1,\cdots,\boldsymbol{\theta}_s\}) &= \sum_{k=1}^{s} \sum_{\boldsymbol{z_i} \in \mathbf{C}_k} f_k(\boldsymbol{z_i},\boldsymbol{\theta}_k). & (2.9) \end{aligned}$$

Then

$$(\mathbf{c}^*,\boldsymbol{\theta}^*) = \text{Arg} \min_{\mathbf{c},\boldsymbol{\theta}} J(\mathbf{c}, 8) \tag{2.10}$$

$J(\mathbf{c},\boldsymbol{\theta})$ has interesting extremal properties.

For a fixed $8 = \{\boldsymbol{\theta}_1,\cdots,\boldsymbol{\theta}_s\}$ the value of $\mathbf{c}$ which minimizes $J(\mathbf{c}, 8)$ w.r.t $\mathbf{c}$ can be written easily:

$$\hat{\mathbf{c}}_{\boldsymbol{\theta}} = \{\hat{\mathbf{c}}_{\boldsymbol{\theta},1},\cdots,\hat{\mathbf{c}}_{\boldsymbol{\theta},s}\} \tag{2.11}$$

where

$$\hat{\mathbf{c}}_{\boldsymbol{\theta}.k} = \{\boldsymbol{z_i} : f_k(\boldsymbol{z_i},\boldsymbol{\theta}_k) \leq f_u(\boldsymbol{z_i},\boldsymbol{\theta}_u),\ \forall k \neq u, u = 1,\cdots,s\}, k = 1,\cdots,s \tag{2.12}$$

The above expression involves only data comparisons.

Similarly for a fixed $\mathbf{c} = \{\mathbf{c}_1,\cdots,\mathbf{c}_s\}$, the minimizing value of $\boldsymbol{\theta}$ is unique and it can he given easily:

$$\hat{\boldsymbol{\theta}}_{\mathbf{c}} = \{\hat{\boldsymbol{\theta}}_{\mathbf{c},1},\cdots,\hat{\boldsymbol{\theta}}_{\mathbf{c},s}\} \tag{2.13}$$

where

$$\hat{\boldsymbol{\theta}}_{\boldsymbol{c},k} = \min_{\boldsymbol{\theta}_k \in R^{n_k}} \sum_{\boldsymbol{z}_i \in \boldsymbol{c}_k} f_k(\boldsymbol{z}_i, \boldsymbol{\theta}_k), \quad k = 1, \cdots, s. \tag{2.14}$$

Moreover, an explicit expression for $\boldsymbol{\theta}_{\boldsymbol{c},k}$ can be given because of the structure of f_k.

We indicate how the computation required in (2.14) can be done in a simple manner. Let $\boldsymbol{\theta}_k = (\boldsymbol{\phi}_k, \boldsymbol{r}_k)$, where $\boldsymbol{\phi}_k$ is an m-vector and $\boldsymbol{r}_k$ is an m x m covariance matrix, and

$$\begin{aligned} f_k(\boldsymbol{z}_i, \boldsymbol{\theta}_k) &= -2 \ln p_k(\boldsymbol{z}_i \mid \boldsymbol{\theta}_k) \\ &= (\boldsymbol{z}_i - \boldsymbol{\phi}_k)^T \boldsymbol{r}_k^{-1} (\boldsymbol{z}_i - \boldsymbol{\phi}_k) + \ln \mid \det \boldsymbol{r}_k \mid + m \ln 2\pi \end{aligned} \tag{2.15}$$

For the given c, let the minimizing value of $\boldsymbol{\theta}_{\boldsymbol{c}.k}$ be $(\boldsymbol{\phi}_{\boldsymbol{c},k}, \hat{\boldsymbol{r}}_{\boldsymbol{c},k})$

$$\begin{aligned} \hat{\boldsymbol{\phi}}_{\boldsymbol{c},k} &= \frac{1}{N_{1k}} \sum_{i:\boldsymbol{z}_i \in \boldsymbol{c}_k} \boldsymbol{z}_i \\ \hat{\boldsymbol{r}}_{\boldsymbol{c},k} &= \frac{1}{N_{1k}} \sum_{i:\boldsymbol{z}_i \in \boldsymbol{c}_k} (\boldsymbol{z}_i - \hat{\boldsymbol{\phi}}_{\boldsymbol{c},k})(\boldsymbol{z}_i - \hat{\boldsymbol{\phi}}_{\boldsymbol{c},k})^T \\ N_{1k} &= \# \boldsymbol{c}_k \\ J(\boldsymbol{c}, \hat{\boldsymbol{\theta}}_{\boldsymbol{c}}) &= Nm(1 + \ln 2\pi) + \sum_{k=1}^{s} N_{1k} \ln \mid \det \hat{\boldsymbol{r}}_{\boldsymbol{c},k} \mid \end{aligned} \tag{2.16}$$

<u>Definition 1</u>: (c*,$\boldsymbol{\theta}^*$) is a local minimum of $J(\boldsymbol{c}, 8)$ if

$$J(\boldsymbol{c}^*, \boldsymbol{\theta}^*) \ \leq \ J(\boldsymbol{c}, \boldsymbol{\theta}^*) \ \forall \boldsymbol{c} \in \Omega_s \tag{2.17}$$

$$J(\boldsymbol{c}^*, \boldsymbol{\theta}^*) \ \leq \ J(\boldsymbol{c}^*, \boldsymbol{\theta}) \ \forall \boldsymbol{\theta} \in R^{n_1} \times R^{n_2} \cdots R^{n_s} \tag{2.18}$$

<u>Definition 2</u>: (Global minimum)

(c,$\bar{\boldsymbol{\theta}}$) is a global minimum if

$$J(\bar{\boldsymbol{c}}, \bar{\boldsymbol{\theta}}) \ \leq \ J(\boldsymbol{c}, \boldsymbol{\theta}) \ \forall \ \boldsymbol{c} \in \Omega_s, \ \boldsymbol{\theta} \in R^{n_1} \times R^{n_2} \cdots R^{n_s} \tag{2.19}$$

Note that a local minimum need not be a global minimum, since in (2.17) and (2.18) we perturb only c or 8 at one time, not simultaneously.

A simple descent algorithm is given for finding a local minimum of $J(\boldsymbol{c}, 8)$. It is done by changing 8 and c alternatively using expressions (2.12) and (2.14), each time having a reduction in $J(\boldsymbol{c}, 8)$.

Since the determination of $\boldsymbol{\theta}_k$ utilizing all $\boldsymbol{z}_i$ in $\boldsymbol{c}_k$ involves the inversion of a matrix, assume that the number of members in $\boldsymbol{c}_k$ must be greater than $2n_k$. Let us call this assumption (Al). (Al) will be relaxed later.

Descent Algorithm(with assumption Al)

1. Let $\boldsymbol{c}^j = (\boldsymbol{c}_1^j, \ldots, \boldsymbol{c}_s^j)$ and $\boldsymbol{\theta}^j = (\boldsymbol{\theta}_1^j, \cdots, \boldsymbol{\theta}_s^j)$ be estimates at the end of j^{th} iteration. Choose $\boldsymbol{c}^1$ arbitrarily, perhaps from a solution of a clustering algorithm with random seeds.
2. Given $\boldsymbol{c}^{(j)}$, compute $\boldsymbol{\theta}^{(j)}$ using the formula in (2.14).
3. Given $\boldsymbol{\theta}^{(j)}$, compute $\boldsymbol{c}^{(j+1)}$ using (2.12).
4. Stop if $\boldsymbol{c}^{(j)} = \boldsymbol{c}^{(j+1)}$; otherwise goto 2.

End.

Note that the computational effort for finding a local minimum is very little. It involves the inversion of a matrix of relatively small dimension and data comparisons.

Theorem 1:

Given data Z, the algorithm converges to a local minimum in a finite number of steps.

Proof:

Since $J(\boldsymbol{c}^{(j+1)}, \boldsymbol{\theta}^{(j+1)}) \leq J(\boldsymbol{c}^{(j)}, \boldsymbol{\theta}^{(j)})$, convergence of the algorithm is assured. Recall that the algorithm stops when $\boldsymbol{c}^{(j)} = \boldsymbol{c}^{(j+1)}$. This is a fixed point for the algorithm. Before reaching a fixed point, there is a finite and non-zero reduction in the **J** value at each iteration namely $\left\{J(\boldsymbol{c}^{(i)}, \boldsymbol{\theta}^{(i)}) - J(\boldsymbol{c}^{(i-1)}, \boldsymbol{\theta}^{(i-1)})\right\}$. Since **J** is bounded below. the convergence has to be in a finite number of steps.

Comments:

1. Relaxing of the assumption Al:

Suppose at some iteration, say j, one of the subsets in $\boldsymbol{c}^j$, say $\boldsymbol{c}_k^j$ has less than $2n_k$ members. Then we cannot carry out step 2 completely, since $\boldsymbol{\theta}_k^j$ cannot be computed. Then we set $\boldsymbol{\theta}_k^j = \boldsymbol{\theta}_k^{j-1}$ and complete step 2. If this condition persists, i.e if one of the subsets of c has fewer members than required, the implication is that the chosen value of s is too high. Then redo the algorithm with the reduced value of **s**.

2. Computation of the best local minimum

By beginning with different partitions, we get different local minima. The best among them, i.e one with the least value gives the desired solution. Of course, there is no guarantee that it is the global minimum. In all the examples, the computation of at most six local minima was sufficient to give the desired solution.

Since the computational requirements for each local minimum is very little, the overall computation needecl for the best local minimum is not much.

3. Recovery of the original partition

In a simulation experiment: we obviously know the label of each data point. Will they be the same as the estimated ones for all $i = 1, \cdots, N$, assuming s is known'? The answer is, yes under certain conditions. Let us denote $Z^{(k)}$ as the generated data set for the k^{th} class. $Z = \bigcup_{k=1}^{s} Z^{(k)}$. If the convex closures of each $Z^{(k)}$ does not. overlap with that of another, then one can prove that the estimated data partition is same as the partition of the original generated data.

4. **Accuracy of estimates**

We give only an example: Suppose $s = 2$. Let $p_k(\boldsymbol{z}_i \mid \boldsymbol{\theta}_k) \sim \text{Gauss}(\mu_k, \boldsymbol{r}_k)$, k = 1, 2 and $\boldsymbol{\theta}_k = (\mu_k, \boldsymbol{r}_k)$.

The posterior density of μ_k given $\boldsymbol{c}_k^*$ is

$$P_k(\mu_k \mid \boldsymbol{c}_k^*) = \text{Gauss}(\mu_k^*, \boldsymbol{r}_k^*/N_{1k})$$

where N_{1k} = number of elements in the set $\boldsymbol{c}_k^*$.

The covariance matrix of μ_k given $\boldsymbol{c}_k^*$ is

$$\text{Cov}\,[\mu_k \mid \boldsymbol{c}_k^*] = \boldsymbol{r}_k^*/N_{1k}$$

4. Maximum Likelihood estimate

In the approach of classical statistics, there are s probability distributions $p_k(\boldsymbol{z}_i \mid \boldsymbol{\theta}_k)$, the parameters $\boldsymbol{\theta}_k, k = 1, \ldots, s$ being unknown. Every data point in Z has a definite class label which is unknown to us or equivalently the ideal partition is an unknown constant. The likelihood expression of the data Z is in (2.7). where both c and $\boldsymbol{\theta}$ are unknown constants. Thus the value of c and 8 which maximize P(Z | c,8) is the maximum likelihood estimate of c and 8. Because of the use of uniform priors, the ML estimate is same a.s the MAP estimates Found earlier.

5. Comparison of partitions c and $\boldsymbol{c}'$

Let $P(\boldsymbol{c} \mid Z)$ denote the probability of the partition c being valid for Z. Given two different partitions c and $\boldsymbol{c}'$, $\boldsymbol{c}, \boldsymbol{c}' \in \Omega_{s,s}$, with same s, we can compute the ratio $\frac{P(\boldsymbol{C}|Z)}{P(\boldsymbol{C}'|Z)}$. We provide the expressions in the 1D Gaussian case. The class densities are given by $p_k(z_i \mid \boldsymbol{\theta}_k) \sim$ Gauss(μ_k, ρ_k). Define the vectors $8 = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_s\}$ and $\boldsymbol{\theta}_k = (\mu_k, \rho_k)$.

$$\begin{aligned} P(\boldsymbol{c} \mid Z) &= \int_{\boldsymbol{\theta}} P(\boldsymbol{c}, \boldsymbol{\theta} \mid Z)\, d\boldsymbol{\theta} \\ &= K(Z,s) \int P(Z \mid \boldsymbol{c}, \boldsymbol{\mu}, \boldsymbol{\rho})\, d\mu_1 \cdots d\mu_s\, d\rho_1 \cdots d\rho_s \\ &= K(Z,s) \prod_{k=1}^{s} \int \left(\prod_{z_i \in \boldsymbol{C}_k} p_k(z_i \mid \mu_k, \rho_k) \right) d\mu_k\, d\rho_k \end{aligned} \tag{2.20}$$

where $K(Z,s) = P(\boldsymbol{c})\ P(\boldsymbol{\theta})/P(Z)$. Following our choice of prior probabilities $P(\boldsymbol{\theta})$ does not depend on 8. Denote the integrand by Q_k. Let $\#\boldsymbol{c}_k = N_k$.

$$Q_k = \prod_{z_i \in \boldsymbol{C}_k} p_k(z_i \mid \mu_k, \rho_k) = \exp\left\{ -\frac{1}{2} \sum_{z_i \in \boldsymbol{C}_k} \frac{(z_i - \mu_k)^2}{\rho_k} - \frac{N_k}{2} \ln 2\pi\rho_k \right\} \tag{2.21}$$

Writing $(z_i - \mu_k)$ a.s $(z_i - \hat{\mu}_k + \hat{\mu}_k - \mu_k)$, where $\hat{\mu}_k = \sum_{z_i \in \boldsymbol{C}_k} z_i / N_k$, simplifying the expression and integrating w.r.t μ_k gives

$$\int Q_k\, d\mu_k = \xi_k\, \rho_k^{-\beta_k}\ \exp\left(-\frac{\alpha_k}{\rho_k} \right) \tag{2.22}$$

where $\alpha_k = N_k \hat{\rho}_k / 2$, $\beta_k = (N_k - 1)/2$, $\xi_k = (2\pi)^{-\beta_k} N_k^{-1/2}$, and $\hat{\rho}_k = \sum_{z_i \in \boldsymbol{C}_k} (z_i - \hat{\mu}_k)^2 / N_k$. Setting $\gamma_k = \rho_k^{-1}$ and using the definition of Γ-function[1].

$$\begin{aligned} \int Q_k\, d\mu_k\, d\rho_k &= \xi_k \int \gamma_k^{\beta_k - 2}\, e^{-\alpha_k \gamma_k}\, d\gamma_k \\ &= \xi_k \frac{\Gamma(\beta_k - 1)}{\alpha_k^{\beta_k - 1}}, \quad \beta_k > 1 \end{aligned} \tag{2.23}$$

Simplifying further, the final expression for $P(\boldsymbol{c} \mid Z)$ is given by

$$P(\boldsymbol{c} \mid Z) = K(Z,s) \prod_{k=1}^{s} a_k\ \hat{\rho}_k^{-(\beta_k - 1)} \tag{2.24}$$

[1] $\int_0^\infty \gamma^{\beta-1} \epsilon^{-\alpha\gamma}\, d\gamma = \frac{\Gamma(\beta)}{\alpha^\beta}$

where $a_k = \xi_k \cdot \Gamma(\beta_k - 1) \cdot \left(\frac{N_k}{2}\right)^{-(\beta_k-1)}$. We can compare $\boldsymbol{c}$ and $\boldsymbol{c}'$ via the log likelihood ratio

$$\begin{aligned} \ln \frac{P(\boldsymbol{c} \mid Z)}{P(\boldsymbol{c}' \mid Z)} &= \ln P(\boldsymbol{c} \mid Z) - \ln P(\boldsymbol{c}' \mid Z) \\ &= \sum_{k=1}^{s} \left[\ln \left(\frac{a_k}{a'_k} \right) - \{(\beta_k - 1) \ln \hat{\rho}_k - (\beta'_k - 1) \ln \hat{\rho}'_k\} \right] \qquad (2.25) \end{aligned}$$

3. CHOICE OF S, THE NUMBER OF DISTINCT CLASSES

The problem of choosing the value of s is known as model order identification or cluster validation. A popular method is to use the Akaike's information criterion. However, it has been shown in [10] that this criterion does not yield consistent estimates. In our method we obtain the estimate of s via Bayesian esimation by considering s also as a random variable. The value of s is intimately related to the partition. So we combine it along with the MAP estimation of the partition and parameters.

Using the Bayes formalism we will compare all the partitions c of Z in $\Omega_{s,s}$ for 5, $1 \leq s \leq$ so, s_0 being known and find the best partition, and incidentally the best value of s.

Reasoning as before, include s also as an unknown to be estimated, $1 \leq s \leq s_0$. The optimal Bayes estimator of (s,c,8) is given below:

$$
\begin{aligned}
&(s^*, \boldsymbol{c}^*_{s*}, \boldsymbol{\theta}^*_1, \cdots, \boldsymbol{\theta}^*_s) && (3.1)\\
&= \operatorname{Arg} \max_{s:1\leq s\leq s_0} \left\{ \max_{\boldsymbol{c}:\boldsymbol{c}\in\Omega_{s,s}} \operatorname{mas}_{\boldsymbol{\theta}_k\in R^{n_k}} p(Z \mid s, \mathrm{c}, \boldsymbol{\theta}) P(\boldsymbol{c} \mid \mathrm{s}) \left(\prod_{k=1}^{s} p(\boldsymbol{\theta}_k) \right) \right\} && (3.2)\\
&= \operatorname{Arg} \min_{s} \left[\left\{ \min_{\boldsymbol{c},\boldsymbol{\theta}_k, k=1,\cdots,s} J_s(\boldsymbol{c}, \boldsymbol{\theta}_k, k=1,\cdots,s) \right\} - 2\ln P(\boldsymbol{c} \mid s) - 2\ln(\prod_{k=1}^{s} p(\boldsymbol{\theta}_k) \right]
\end{aligned}
$$

We need the prior probabilities. Let all s have same prior value,

$$P(s) = 1/s_0, \; s = 1, \cdots, s_0 \qquad (3.3)$$

$$
\begin{aligned}
P(\boldsymbol{c} \mid \mathrm{s}) &= \text{Prior probability of a partition c given that it has } s \text{ non-zero subsets}\\
&= \frac{1}{\#\Omega_{s,s}}, \quad \sum_{\boldsymbol{c}:\boldsymbol{c}\in\Omega_{s,s}} P(\boldsymbol{c} \mid s) = 1 \qquad (3.4)
\end{aligned}
$$

Prior probability of each component in $\boldsymbol{\theta}_k$ is uniform and equals $1/L_k$. Let

$$\operatorname{Arg} \min_{\boldsymbol{c},\boldsymbol{\theta}_k} J_s(\boldsymbol{c}, \boldsymbol{\theta}_k, k=1,\cdots,s) = \left\{ \boldsymbol{c}^*_s, \boldsymbol{\theta}^*_{s,1}, \cdots, \boldsymbol{\theta}^*_{s,s} \right\} \qquad (3.5)$$

where $\boldsymbol{c} \in \Omega_{s,s}$. For $1 \leq s \leq s_0$, $\boldsymbol{c}_s^*, \boldsymbol{\theta}_{s,1}^*, \cdot . \cdot, \boldsymbol{\theta}_{s,s}^*$; $\boldsymbol{c}_s^* = \{\boldsymbol{c}_{s,1}^*, \ldots, \boldsymbol{c}_{s,s}^*\}$ can be determined.

$$s^* = \min_{s:1 \leq s \leq s_0} \left[J_s(\boldsymbol{c}_s^*, \boldsymbol{\theta}_{s,1}^*, \cdots, \boldsymbol{\theta}_{s,s}^*) + 2\ln(\#\Omega_{s,s}) + 2\sum_{k=1}^{s} n_k \ln L_k \right] \tag{3.6}$$

Since L_k is the prior density of $\boldsymbol{\theta}_k$, it should cover the total range of all the components of $\boldsymbol{\theta}_k$. Different choices of L_k may give different optimal s. We set $L_k = \boldsymbol{10}, \boldsymbol{k} = 1, \cdot, ., s$ in the first three numerical examples and $L_k = 256, k = 1, \cdot .., s$ in the image segmentation examples.

Example:

Let all the s densities p_k be multivariate Gaussian. $p_k(\boldsymbol{z}_i \mid \boldsymbol{\theta}_k = \boldsymbol{\phi}_k, \boldsymbol{r}_k) \sim \text{Gauss}(\boldsymbol{\phi}_k, \boldsymbol{r}_k)$ then

$$\begin{aligned} J_s(\boldsymbol{c}_s^*, \boldsymbol{\theta}_{s,1}^*, \cdots, \boldsymbol{\theta}_{s,s}^*) &= Nm(1 + \ln 2\pi) + \sum_{k=1}^{s} N_{s,k} \ln \det \hat{\boldsymbol{r}}_{s,k} \qquad (3.7) \\ &\triangleq G_s \end{aligned}$$

where $N_{s,k} = \#\boldsymbol{c}_{s.k}^*$ and

$$\hat{\boldsymbol{r}}_{s,k} = \text{the covariance matrix computed from the subset } \boldsymbol{c}_{s,k}^* \qquad \textbf{(3.8)}$$

Let $n_k = n_0 = m + m(m + 1)/2, k = 1, \cdot .., \text{s}$. The argument to be minimized in equation (3.6) be denoted by $\boldsymbol{H},$. Then

$$\begin{aligned} H_s &= G_s + 2(N \ln s - \ln s!) + 2\ s\ n_0\ \ln 10, \quad 1 \leq s \leq s_0 &(3.9) \\ s^* &= \text{Arg} \min_{1 \leq s \leq s_0} H_s &(3.10) \end{aligned}$$

Note G_s usually goes down as s increases and $\{2(N \ln s - \text{In}\, \boldsymbol{s!)} + 2\ s\ n_0\ \ln \boldsymbol{10)}$ increases with s. Thus a minimizing $\boldsymbol{H},$ w.r.t s yields a finite value for s.

Comparison of two partitions *c* and *c′* with different values of s

Suppose we have *2* partitions $\boldsymbol{c}^1 \in \Omega_{s_1,s_1}$ and $\boldsymbol{c}^2 \in \Omega_{s_2,s_2}$ with the number of classes s_1 and s_2 respectively. We can compare the probabilities $P(s_k, \boldsymbol{c}^k, \boldsymbol{\theta}^{*k} \mid \boldsymbol{Z}), \boldsymbol{k} = \boldsymbol{1}, 2$ to decide which partition is better. We compute the log likelihood ratio $\ln\left(\frac{P(s_1, \boldsymbol{c}^1, \boldsymbol{\theta}^{*1} \mid Z)}{P(s_2, \boldsymbol{c}^2, \boldsymbol{\theta}^{*2} \mid Z)}\right)$. Let the class densities be multivariate Gaussians given by $p_k(z_i \mid \boldsymbol{\theta}_k) \sim \text{Gauss}(\boldsymbol{\phi}_k, \boldsymbol{r}_k)$ as in the preceding example. Then $P(s, \boldsymbol{c}, \boldsymbol{8} \mid \boldsymbol{Z})$ is given by H_s defined in (3.9), with $\boldsymbol{G},$ given by ***(3.8)***. The log likelihood ratio can be computed as follows

$$\ln P(s_1, \boldsymbol{c}^1, \boldsymbol{\theta}^{*1} \mid Z) - \ln P(s_2, \boldsymbol{c}^2, \boldsymbol{\theta}^{*2} \mid Z) = -\frac{1}{2} H_{s_1} + \frac{1}{2} H_{s_2} \tag{3.11}$$

4. ROBUST REGRESSION

Ordinary regression by least squares is widely used in many disciplines. However it is well known that the estimate of the fitted line or plane to the data is very sensitive to the presence of extraneous data. The addition of even five percent to ten percent of the noise data–the so called leveraged data points–can drastically alter the fitted line; sometimes it. is even roughly orthogonal to the original line.

Robust regression is a special case of the unsupervised classification problem with two classes. The given data set. Z has to he partitioned into 2 classes c $= \{\boldsymbol{c}_1, \boldsymbol{c}_2\}$, $\boldsymbol{c}_1$ being the subset of all the inliers or good points and $\boldsymbol{c}_2$ being the subset of all the outliers or noise points. It is closely related to the generalized maximum likelihood order statistic (GMLOS) method given in [13]. Let the data vector be $\boldsymbol{z_i} = (y_i, \boldsymbol{x}_i)$. where y_i is the dependent variable and $\boldsymbol{x}_i$ the $(m-$ I) dimensional independent vector. We need to regress y on. x for the inliers. Fit multivariate Gaussians for both inliers and outliers. For the inliers, the fitted Gaussian model is

$$p_1(\boldsymbol{z_i} \mid \boldsymbol{\theta}_1) = p_{11}(y_i \mid \boldsymbol{x}_i, \boldsymbol{\phi}_{11}, \rho_1) p_{12}(\boldsymbol{x}_i \mid \boldsymbol{\phi}_{12}, \boldsymbol{r}_1) \tag{4.1}$$

where

$$\begin{aligned} p_{11}(y_i \mid \boldsymbol{x}_i, \boldsymbol{\phi}_{11}, \rho_1) &\sim \text{Gauss}(\boldsymbol{\phi}_{11}^T(1, \boldsymbol{x}_i), \rho_1) \\ p_{12}(\boldsymbol{x}_i \mid \boldsymbol{\phi}_{12}, \boldsymbol{r}_1) &\sim \text{Gauss}(\boldsymbol{\phi}_{12}, \boldsymbol{r}_1) \end{aligned}$$

and $\boldsymbol{\theta}_1 = \{\boldsymbol{\phi}_{11}, \boldsymbol{\phi}_{12}, \rho_1, \boldsymbol{r}_1\}$, $\boldsymbol{\phi}_{11}$ is m-vector, $\boldsymbol{\phi}_{12}$ is (m – 1)-vector, $\boldsymbol{r}$ is $(m-1)$ x $(m-1)$-matrix. The outliers can he fitted by $p_2(\boldsymbol{z_i} \mid \boldsymbol{\theta}_2)$ where

$$p(\boldsymbol{z_i} \mid \boldsymbol{\phi}_2, \boldsymbol{r}_2) \sim \text{Gauss}(\boldsymbol{\phi}_2, \boldsymbol{r}_2)$$

The optimal data partition c, for $s = 1, 2, 3$ etc can be computed and the optimal s can be estimated as before.

Apparently there is no Bayesian method of robust estimation in the literature as mentioned in)]. The estimates presented here have several important advantages over traditional methods of robust regression like the Least Median Squares(LMedS) [24].

The LMedS uses the following criterion:

$$\min_{\alpha,\beta}\left[\mathrm{Median}_{(y_i,x_i)\in Z}\{(y_i-\alpha-\beta x_i)^2\}\right]. \tag{4.2}$$

Note that the median is taken over the entire data. In image processing applications, whenever two surfaces meet in a processing window, the pixels belonging to one of the surfaces can be regarded as outliers for the fit to the other one. Clearly if the number of outlier points is greater than number of inliers. the median gives information about outliers than the inliers, i.e, LMedS is fitting the best regression line to the outliers. Obviously LMedS result has no relevance.

Secondly the Bayesian theory also yields the covariance matrices of the estimates of parameters. It is well known that LMedS estimates are biased and their variances or covariance? are nowhere near the possible minimum values.

5. NUMERICAL EXAMPLES

5.1 Example 1: (Star Data)

This example comes from astronomy. It has been used earlier in [25]. The data set (y_i, x_i) consists of 47 points. x_i is the logarithm of the effective temperature at the surface of the star and y_i is the logarithm of its light intensity. The raw data is shown in Figure 5.1a. Majority of the stars which constitute the main sequence follow a steep band and four stars in a vertical line standing apart are the red giants.

We will apply our method, compare the results with $s = 1, 2, 3$ and 4. We will distinguish between the two problems namely clustering and regression. The goals of regression and clustering are entirely different. In regression we look for a single line that explains the entire data except for a few outliers. This line is used later for forecasting purposes. Where as in clustering we look for different distinct groups of data with their associated distribution parameters. A solution for robust regression, where we first identify the outliers ancl find the regression line for only the inliers, is clearly provided by clustering with $s = 2$.

We will compare the results given by our clustering algorithm with s = 2 and the regression estimate given by the least median squares method(LMedS). The mean square value of the error residuals will be our criterion for comparison.

In classification for clustering, we fit s bivariate Gaussians to the data, $s = 1, 2, 3$ and **3** and get the **best** local minimum in each case whose highlights are given below.

Notice that G_s value falls as s increases. Note (s = 3) has the lowest value of H_s indicating that the optimal number of classes is **3** according to our criterion. The results of clustering using our method are given in Figures 5.1a-5.1c for s = 1,2 and 3 respectively. For s = 2 and 3 we show the best local minimum along with a fitted line to each class. We note that visual inspection of the data does show **3** distinct classes. Hence H_s seems to be a useful statistic in determining the number of classes needed for classification or clustering.

s	Members in optimal partition	G_s	H_s
1	{47}	93.88	116.91
2	{41, 6}	-38.02	71.80
3	{26, 4, 17}	-105.48	63.28
4	{4, 14, 12, 17}	-138.99	77.07

Table 5.1 Optimal data partitions and their corresponding costs for the Star data; $N = 47$

Regression

With $s = 2$, about half a dozen local minima were found with G_s value:; {-38.02, -34.03, -21.36, -21.34, 6.69, 1.41). The one given in Table. 5.1 is the best local minimum. Except for local minima with relatively large values of G,. all other local minima give similar values.

Figure 5.1a is raw data and also indicates the regression line obtained from the least squares method to entire clata. Figure 5.1b is the partition corresponding to the best local minimum with $G_s = -38.02$; the associated starting partition is in Figure 5.1d. Our method picks up all the 4 red giants as outliers. It also picks two stars from the main sequence as outliers. We note here that this partition was declared better than the one with only the 4 red giants as the outliers. Even the mean square value of the residuals for the inlier set is lower for the configuration given in Figure 5.1b.

The regression lines derived for the fitted bivariate densities are shown for each class. Note that the regression line is nearly orthogonal to the one obtained with $s = 1$. Also the mean square value of residuals of the inlier class with $s = 2$ is 0.1365, which is less than one half of the corresponding value obtained with $s = 1$ namely 0.3052. This feature again indicates the advantage of detecting outliers and eliminating them from consideration.

The variances of the estimated parameters of the regression line for $s = 1$ is $\{(0.0784, 1.4637)\}$. For $s = 2$ the parameters of the regression line fitted to the class with 6 points have variances $\{(0.1041, 1.3785)\}$ and those of the line fitted to the class with 41 points have $\{(0.2894, 5.6296)\}$.

Since we need not be restricted to choosing gaussian distributions for the class densities, we have also conducted experiments using a Gaussian distribution and a uniform distribution for the regression case. We note that when the outliers form another cluster: which happens when there are leveraged points, using a gaussian distribution to model them gives better regression results. However, if they do not, then a uniform distribution captures the outlier distribution well.

Figure 5.2 shows two partitions each for $s = 2$ and $s = 3$ that are fixed points of the algorithm. These correspond to local minima of the functional G_s other than the best,. Figures 5.2a and 5.2b are the starting partitions with $s = 2$ that converged to the partitions in Figures 5.2c and 5.2d respectively. Similarly, Figures 5.2e and 5.2f are starting partitions with $s = 3$ that converged to the partitions shown in Figures 5.2g and 5.2h. The costs of the respective partitions are shown along with each figure.

Clustering Algorithm

Figures 5.1b and 5.1c give the results obtained by applying the clustering algorithms with number of classes s equal to 2 and 3 respectively. The results of clustering algorithm are relatively robust to the starting points. The clustering result of Figure 5.1b with $s = 2$ is counter intuitive. It makes a horizontal cut of data, including the outliers in one of the classes. With $s = 3$, the 4 outliers are placed in a. separate class. We could use the results of clustering algorithm as the initial partition.

LMedS methocl

As discussed in [25], the line fit given by LMedS method, $\hat{y} = 3.898x - 12.298$, nicely fits the main sequence of stars. There is no apriori recognition of outliers in the LMedS method. The outliers, if needed to be picked, are identified by their large residual errors after the regression line is computed. For the parameters estimated using LMedS, there is no available method to compute the accuracy of estimates.

5.2 Example 2: (Simulated data)

This data set is a variant of the one used extensively in statistical literature in connection with robust regression [24]. Most robust regression estimation techniques except LMedS fail in this case. The data set, (y_i, x_i), i $= 1, \ldots, N$, consists of a cluster around a straight line

[1] with 20 points, the so called inliers and a blob cluster. [2] consisting of outlier data with 80 points. The original data had 30 inliers and 20 outliers. We essentially changed the fracton of outliers in the data from 0.4 to 0.8.

The aim in the statistical literature is to recover the regression line associated with inliers. Since the number of outliers is much greater than that of inliers, all the standard statistical methods including LMedS fail. They assume that the fraction of outliers is less than 0.5, consequently the outlier set is treated as the inlier set.

We will use our approach with $s = 1, 2$ and 3, fitting bivariate Gaussians in all the cases. The results for all the cases are shown in Figure 5.3. We tabulate the results in Table. 5.2. Figure 5.3a shows the raw data and the regression line for the entire data.

s	Members in optimal partition	G_s	H_s
1	{100}	610.52	633.55
2	{80, 20}	257.73	441.03
3	{20, 42, 38}	165.29	450.51

Table 5.2 Optimal data partitions and their corresponding costs for the Simulated data.

$\underline{s = 2}$

Figures 5.3b and 5.3d show the best local minimum, G_s = 257.73, and the initial partition that led to the best local mininimum respectively for s = 2. There was only one other minimum for this data set with G_s = 489.92. Note from Table. 5.2 that s = 2 has the least value of H_s indicating that the best value of s is s = 2.

$\underline{s = 3}$

Figure 5.3c shows the best local minimum with s = 3. With s = **3,** there is only a finer partition of the partition with s = 2. Note the regression line for the inliers is captured well with s = 2. The G, values corresponding to six other local minima are {165.29, 171.82, 173.94, 15'7.25, 182.01, 225.76). We have observed that there are numerous local minima

[1] $y_i = x_i + 2 + \text{Gauss}(0, 0.04)$, where x_i is uniformly distributed on $(1, 4)$

[2] 2D Gaussian distribution with mean (7.2) and covariance $0.5 * \mathbf{I}$.

around $G_s = 170$. This configuration corresponds to the splitting of the blob cluster into two and the line cluster.

Clustering algorithm Figures 5.3e and 5.3f give the results of the clustering algorithm with s = 2 and $s = \mathbf{3}$ respectively. Both the clustering results are reasonable. As before. when we get 3 clusters, we have no way of deciding whether the data contains 2 or 3 clusters. Our method clearly indicates that $s = 2$ is sufficient,.

The variances of the estimated parameters of the regression line for s = 1 is $\{(0.0013, 0.0525)\}$. For $s = 2$ the parameters of the regression line fitted to the class with 80 points have variances $\{(0.0142, 0.7077)\}$ and those of the line fitted to the class with 20 points have $\{(0.0065, 0.0459)\}$.

5.3 Example 3: (Multi-sensor target tracking)

Multiple sensors send observations $\boldsymbol{z_i} = (y_i, x_i), i = 1, \cdots, N$ to the central station [20, 21]. There could be multiple targets in the atmosphere and their number could be variable at any given time. The raw data of 120 points is shown in Figure 5.4a. The x-coordinate is related to time. The figure shows all the observations collected upto a time t_1. We have not shown time in the graph. As time progresses there is more data,. There is no target label attached to each observation. It is known apriori that the trajectory of a target obeys some parametric curve in the $X - Y$ plane; straight line: parabola etc. For simplicity we consider a straight line. There are also observations caused purely by noise, the clutter. Note that one trajectory is completely inside the clutter. Moreover the range of this trajectory is much less than that of others. The problem is to identify the number of targets, their tracks and the clutter points. Intersection of the trajectories in the figure indicates intersection in feature space, not in real time.

There are many methods for assigning labels to these targets. One of the methods is clustering. The principle difficulty in some of the procedures is as follows: Consider two intersecting targets say AB and CD intersecting at the point O, so that AO and OB belong to one trajectory, CO and OD to the other. Many methods (including the clustering algorithm) take two parts of two different trajectories, i.e the part AO from one trajectory and OC from the other and declare AOC as one trajectory, similarly BOD as the other. This happens because the clustering algorithms do not use the available information that the trajectories are straig11.tlines or parabolas etc.

Each trajectory is parametrized by a line $L(\beta, \gamma, \rho)$ and obeys the equation

$$y_i = \beta x_i + \gamma + \text{Gauss}(0, \rho), \quad i = 1, \cdots, N \tag{5.1}$$

x_i are uniformly distributed in the range $[0, 10]$. The three line trajectories are $L_1(0.4, 5, 0.01)$, $L_2(-0.3, 9, 0.01)$, $L_3(0.1, 2.1, 0.0025)$. The clutter is modeled by a Gaussian distribution given by Gauss $\left(\begin{bmatrix} 5 \\ 4 \end{bmatrix}, \begin{bmatrix} 0.4 & 0.2 \\ 0.2 & 1.2 \end{bmatrix}\right)$. There are 30 points in each trajectory class as well a.s the clutter class, a total of 120 points.

Results with clustering algorithm

$s = 4$

The result with $s = 4$ is given in Figure 5.4e. The clustering captures only one of the three line trajectories. One cluster combines parts of the 2 lines of the data, before the intersection. The other cluster captures the other two halves of the line clusters in the data.

$s = 5$

The result with $s = 5$ is given in Figure 5.4f. This clustering is also erroneous. It doesn't identify any line trajectories correctly. The clusters corresponding to the clutter and the trajectory within it are subdivided into two clusters without the trajectory being identified.

Results with our method

$s = 4$

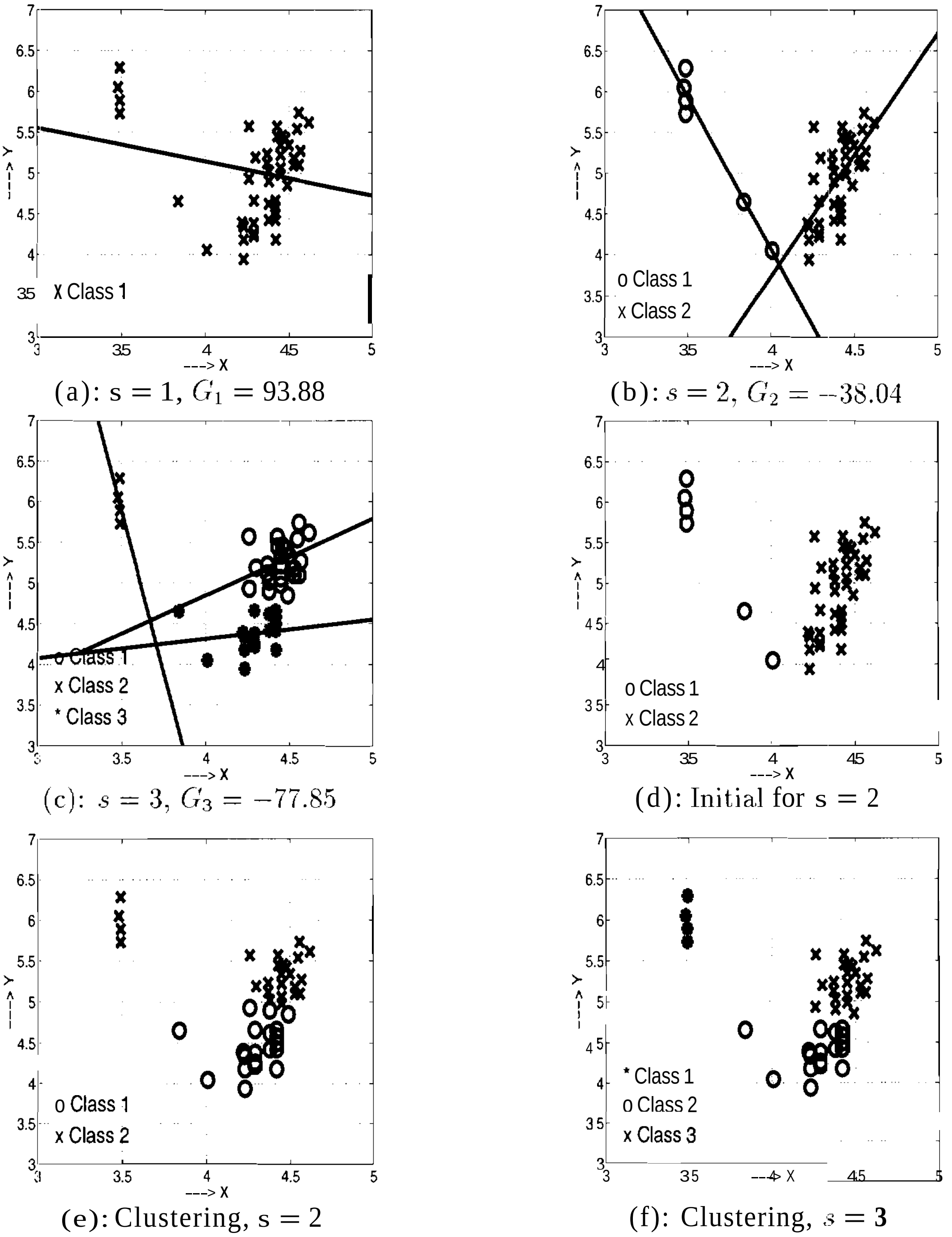
All density families p_k are multivariate densities. The best local minimum has G_s = 219.62 and the corresponding plot is in Figure 5.4c.

Note that our method captures the four classes correctly. Even the trajectory within the clutter is identified correctly. Another local minimum is given in Figure 5.5d. The corresponding H_s = 638.09.

s = 5

The result associated with best local minimum is in Figure 5.4d. The result with another local minimum is in Figure 5.5e. Note the result divides the data of smaller trajectory and the clutter into 3 clusters, correctly finding the clusters of two big lines. G_s = 193.69 and

$H_s = 685.51$. Notice that H_5, the H-statistic with $s = \mathbf{5}$ is much larger than H_4 indicating that the correct value of s is 4.



(a): s = 1, $G_1 = 93.88$

(b): $s = 2$, $G_2 = -38.04$

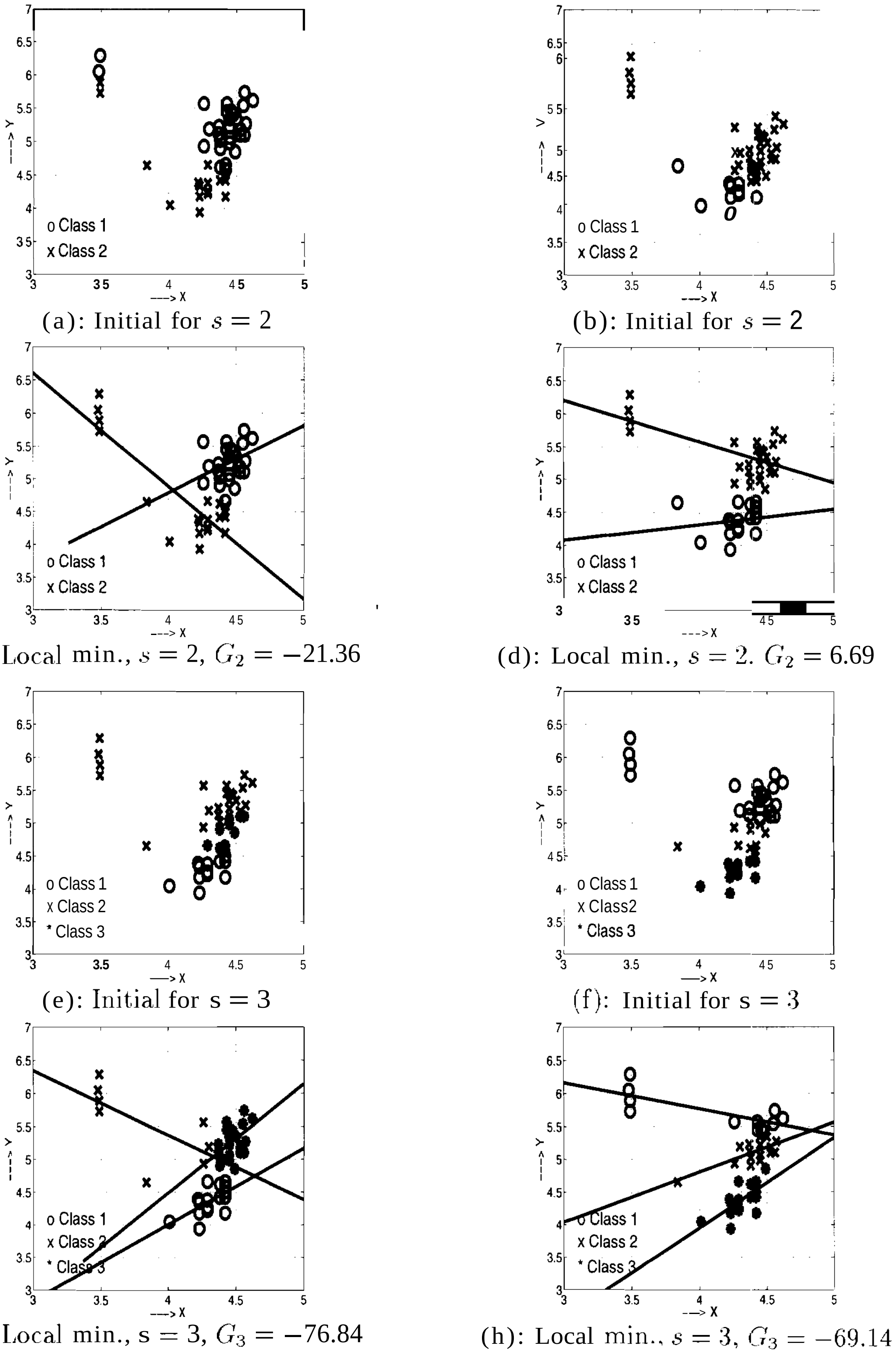
(c): $s = 3$, $G_3 = -77.85$

(d): Initial for s = 2

(e): Clustering, s = 2

(f): Clustering, $s = 3$

Fig. 5.1. Best local minima. Star clata, $N = 47$, for $s = 1, 2, 3$. (e,f) Fuzzy clustering results for $s = 2$ ancl 3. (d) Starting partition for (b).



(a): Initial for $s = 2$

(b): Initial for $s = 2$

(c): Local min., $s = 2$, $G_2 = -21.36$

(d): Local min., $s = 2$. $G_2 = 6.69$

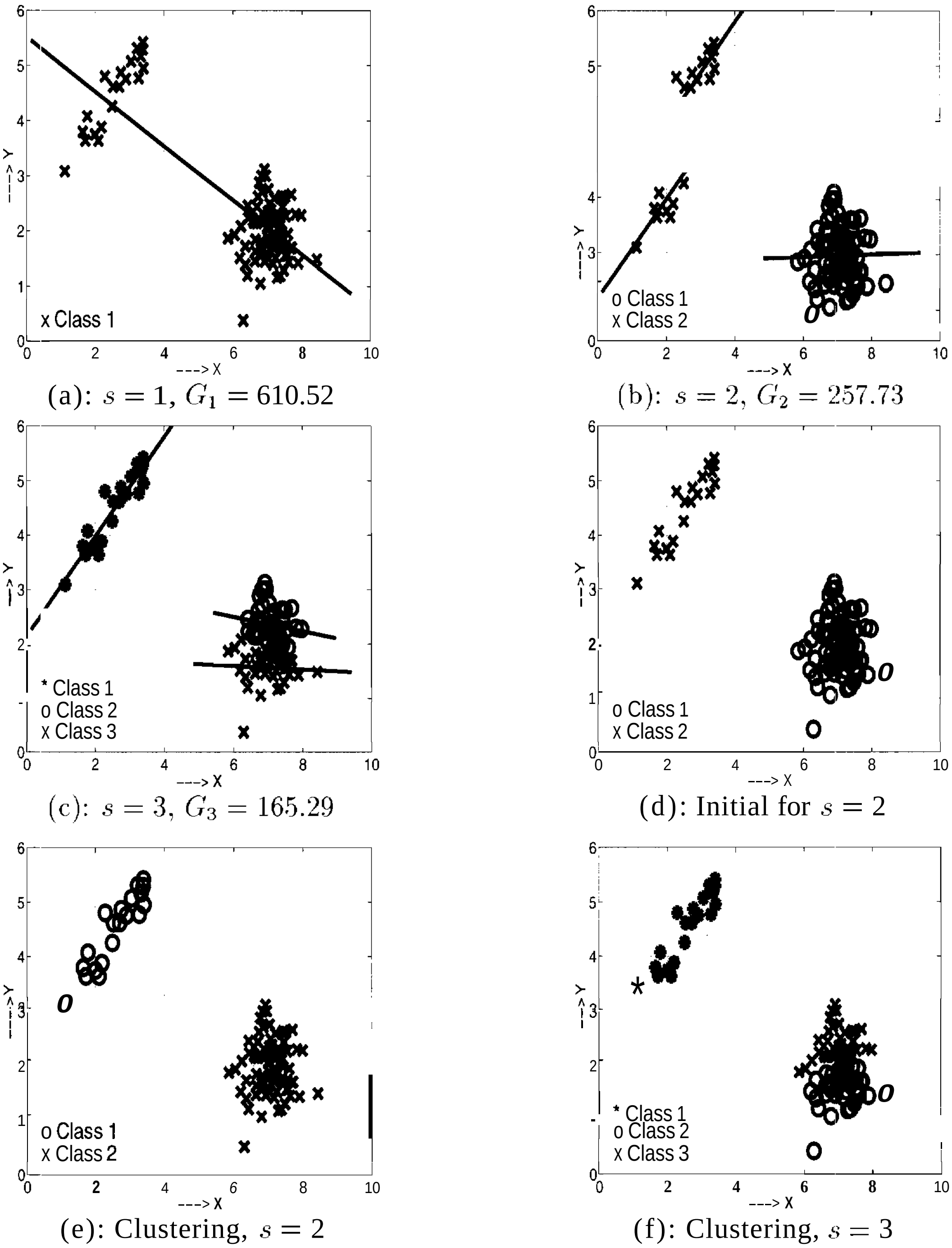
(e): Initial for s = 3

(f): Initial for s = 3

(g): Local min., s = 3, $G_3 = -76.84$

(h): Local min., $s = 3$, $G_3 = -69.14$

Fig. 5.2. Other local minima. Star data, $N = 47$, for s = 2, 3. (a,b) Starting partitions for (c,d) respectively. (e,f) Starting partitions for (g,h) respectively.



(a): $s = 1$, $G_1 = 610.52$

(b): $s = 2$, $G_2 = 257.73$

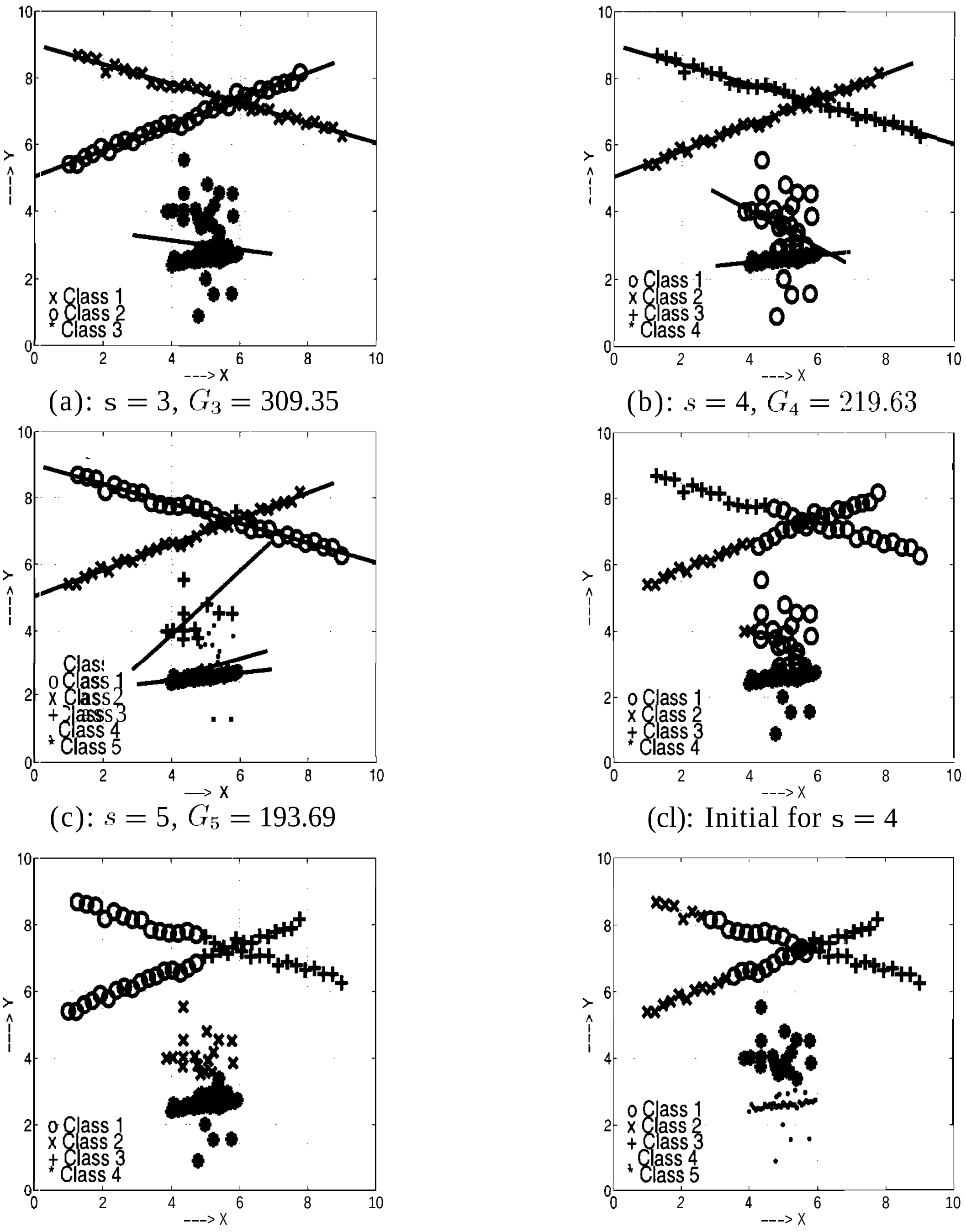
(c): $s = 3$, $G_3 = 165.29$

(d): Initial for $s = 2$

(e): Clustering, $s = 2$

(f): Clustering, $s = 3$

Fig. 5.3. Best local minima. Simulated data, $N = 100$, for $s = 1, 2, 3$. d) Starting partition for (b), $s = 2$. e,f) Fuzzy clustering results for $s = 2$ and 3.



(a): s = 3, G_3 = 309.35

(b): s = 4, G_4 = 219.63

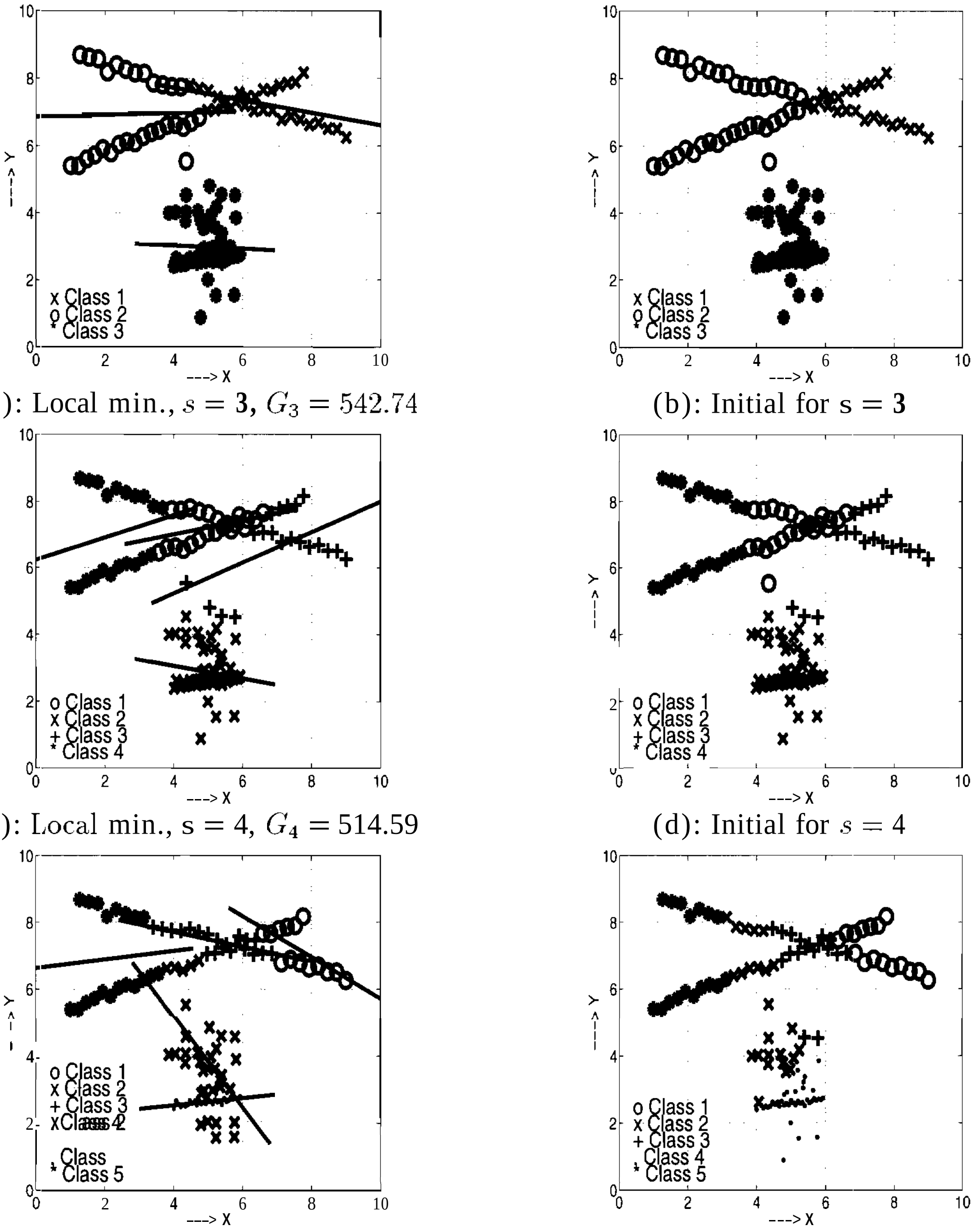
(c): s = 5, G_5 = 193.69

(cl): Initial for s = 4

(e): Clustering, s = 4

(f): Clustering, s = 5

Fig. 5.4. Best local minima. Target tracking data, $N = 120$, for s = $3, 4, 5$. (d) Starting partition for (b), s = 4. (e,f) Fuzzy clustering results for s = 4 and 5.



(a): Local min., $s = 3$, $G_3 = 542.74$

(b): Initial for s = 3

(c): Local min., s = 4, $G_4 = 514.59$

(d): Initial for $s = 4$

(e): Local min., s = 5, $G_5 = 330.11$

(f): Initial for s = 5

Fig. 5.5. Other local minima. Target tracking data, $N = 120$, for $s = 3, 4, 5$. (b,d,f) Starting partitions for (a,c,e) respectively.

6. BAYESIAN APPROACH FOR IMAGE SEGMENTATION

Consider an M x M image with intensities given by {y„ ,$\imath,\jmath$ = $0,\cdots,M-1$). Our aim is to partition the image into b segments such that the segments are non-overlapping except for border pixels. All the pixels corresponding to the same segment represent the same artifact like road, water, house. etc. However the number of distinct classes, s, is less than or equal to b, since two or more segments which are separated from each other by some other segment may belong to the same artifact like water. Thus we have only to deal with s densities $p_k(y_{i,j} \mid \boldsymbol{\theta}_k)$

Secondly the number of pixels in an image is a large number. For a 2.56 x 256 image we are dealing with $N = 256^2 = 65536$ pixels. Carrying out the segmentation at the finest level may involve very extensive computation. Instead, the computation is formed at several scales. The image is divided into blocks of size 10 x 10 so that all pixels of the same block are assigned the same class. Effectively the number N for the algorithm becomes $(N/100)$. Based on the segmentation, we go to the next coarser scale say 5 x 5 etc. Repeat till we get the lowest level.

We assume that the $y_{i,j}$ are clustered around polynomials specifiable through facet models.

$$y_{i,j} = \alpha_0 + \alpha_1 i + \alpha_2 j + \alpha_3 ij + \eta_{i,j} \tag{6.1}$$

$\eta_{i,j} \sim \text{Gauss}(0, \rho_k)$ is the white noise with variance ρ_k. Then the density of $y_{i,j}$ belonging to the k^{th} segment is given by

$$p_k(y_{i,j} \mid \boldsymbol{\theta}_k) = \text{Gauss}\left(\alpha_0 + \alpha_1 i + \alpha_2 j + \alpha_3 ij,\ \rho_k\right) \tag{6.2}$$

Note all the pixels belonging to class k have the same variance but not the same parameters $\boldsymbol{\theta}_k = (\boldsymbol{\alpha}_k, \rho_k)$.

For a fixed s, to obtain the MAP estimates of c and 8 we have to minimize the function

$$J_s(\boldsymbol{c},\boldsymbol{\theta}) = -2\sum_{k=1}^{s}\sum_{(i,j),y_{ij}\in c_k} \ln p_k(y_{i,j} \mid \boldsymbol{\theta}_k) \tag{6.3}$$

Segmentation with multiple scales

We want a partition of the image c $= \{\boldsymbol{c}_1,\cdots,\boldsymbol{c}_s\}$ where $\boldsymbol{c}_k$ is a subset of Z, and all members in $\boldsymbol{c}_k$ may not be contiguous. Since the number of pixels N is large we have to carry out the partition at several scales.

In the beginning let us deal with blocks of pixels say 4×4. Let the block be denoted by the leading pixel. For instance the hlock $\{(i+k, j+l), k,1 = 0,1,2,3\}$ will he denoted by b,,,. We assign the entire pixel hlock to one subset $\boldsymbol{c}_k$ in the partition. Note we are not averaging the intensities in the block. Each pixel retains its identity. Thus we have $(N/16) = N_1$ blocks. The class assignment of the block is given by

$$\begin{array}{l}\text{Assign all pixels } \in b_{i,j} \text{ to } \boldsymbol{c}_k\\ \text{if } \left\{\sum_{i+u}\sum_{j+v} f_k(y_{i,j};\boldsymbol{\theta}_k) \le \sum_{i+u}\sum_{j+v} f_u(y_{i,j};\boldsymbol{\theta}_k) \quad \forall u\right\}\end{array} \tag{6.4}$$

Partition at the coarse level

Since in the example we cleal with 80 x 80 image, $N = 6400$. We carry out segmentation at 3 levels: 4 x 4, 2 x 2 and the finest level. Consider the coarsest level. Let

$$Z_1 = \{y_{i,j}, 0 \le i,j \le 79\} \tag{6.5}$$

$$Z_2 = \{b_2^{2i,2j}, 0 \le i,j \le 39\} \tag{6.6}$$

$$Z_4 = \{b_4^{4i,4j}, 0 \le i,j \le 19\} \tag{6.7}$$

where $b_k^{u,v} = \{y_{u+i,v+j}, 0 \le i, \mathbf{j} \le$ k – 1). Let the corresponding partition be $\boldsymbol{c}_4$,

$$\boldsymbol{c}_4 = \{\boldsymbol{c}_{4,1},\cdots,\boldsymbol{c}_{4,s}\}$$

where $\boldsymbol{c}_{4,s} \subseteq Z_4$. All the 1 x 1 pixels in the same block $b_4^{u,v}$ will have the same class assignment, i.e the same density p_k is assigned to all the 1 x 1 pixels in the same block.

$\boldsymbol{\theta}_k =$ parameter associated with the density of class k

For a given partition $\boldsymbol{c}_4$

$\boldsymbol{\theta}_k$ is computed for the 1 x 1 pixel intensities $y_{i,j}$ in all the blocks $b_4^{u,v}$ assigned to $\boldsymbol{c}_{4,k}$ as indicated.

For a given $\boldsymbol{\theta}_k, k = 1, \cdots, s$ the partition is updated as follows

Assign $b^{u,v}$ to $\boldsymbol{c}_{4,k}$ if

$$\left[\sum_{0 \leq i,j \leq 3} f_k(y_{u+i,v+j}; \boldsymbol{\theta}_k) \leq \sum_{0 \leq i,j \leq 3} f_l(y_{u+i,v+j}; \boldsymbol{\theta}_l) \quad \forall l = 1, \cdots, s \right] \tag{6.8}$$

Thus we get the best partition

$$\boldsymbol{c}_4^* = \{\boldsymbol{c}_{4,1}^*, \cdots, \boldsymbol{c}_{4,s}^*\}$$

Partition at next coarse level 2 x 2

We divide the 4 x 4 blocks into 2 groups, the boundary or B blocks and non-boundary or NB blocks. Each block has 4 immediate neighbors: top, bottom, left, right. Two neighboring blocks are labeled NB if their class labels are different.

$$\begin{aligned} \boldsymbol{c}_{4,k}^{NB} &= \{b_4^{u,v}, b_4^{u,v} \text{ is } NB, b_4^{u,v} \in \boldsymbol{c}_{4,k}\} \\ Z_4' &= \bigcup_{k=1}^{s} \boldsymbol{c}_{4,k}^{B} \end{aligned}$$

The innportant idea. here is the class assignments made to the 1 x 1 pixels in the blocks of $\boldsymbol{c}_{4,k}^{NB}$ are fixed and not altered in subsequent iterations. Only the assignments of pixels in Z_4' are altered. Consider a partition of $\boldsymbol{c}_2$ at the level 2 x 2

$$\boldsymbol{c}_2 = \{\boldsymbol{c}_{2,1}, \cdots, \boldsymbol{c}_{2,s}\}$$

All the 2×2 blocks derived from $\boldsymbol{c}_{4,k}^{NB}$ are permanently assigned to $\boldsymbol{c}_{2,k}$ and their class labels are not altered in iteration.

$$Z_2'' = \{b_2^{u,v}, b_2^{u,v} \text{ derived from } Z_4'\}$$

At every iteration every member of Z_2'' is assigned to $\boldsymbol{c}_{2,1}, \cdots, \boldsymbol{c}_{2,s}$ as the case may be. Computation of $\boldsymbol{\theta}$ and updating of $\boldsymbol{c}_2$ is similar to the earlier case of 4 x 4. After arriving at 1×1 level, the final result is cleaned by averaging over a 5 x 5 window.

Choice of s

The best value of s is that which minimizes H,. However, in our multiscale scheme, it is more robust to decide on the value of s at a coarser scale itself. So the value of s is decided at the scale where all the pixels in blocks of size 4 x 4 have the same class assigned to them.

6.1 Example 1: (Synthetic Image)

We consider a synthetic image made up of three textures from the Brodatz album. The image is 80×80 made of 5 segments and **3** classes. The original image is in Figure 6.1a.

First consider the coarse level segmentation at level 3 x 3 involving $N = 300$ blocks. We begin with arhitrary class assignments for the initial partition and derive the associated local minimum. Several different local minima are derived. There are many which are close to each other visually as well as in J-values. The best local minimum is displayed in Figure 6.1b and the associated initial partition in Figure 6.1c. Next we explore segmentation at 2×2 level. Note there is no need for arbitrary initial choice of partition since the $\{\boldsymbol{\theta}_k, k = 1, \cdot \cdot \cdot, s\}$ obtained from 4 x 4 level can serve as the starting point. The final result is given in Figure 6.1d. Note that class assignments for the nonboundary blocks in Figure 6.11) are not altered. Similarly the result of segmentation at the lowest level is display in Figure 6.1e and the cleaned image in Figure 6.1f. The number of errors in the final segmentation at the pixel level is 63 which corresponds to 1% misclassification error. We note that the boundaries are visually perfect and the actual error at the pixel level is only 1%.

Choice of s

The value:; of H_s for $s = 2, 3$ and 4 are $63032.97, 61550.89$ and 61707.41 respectively. The value of $\boldsymbol{H_s}$ is minimum for $s = 3$ which is the actual number of distinct textures present in the image

6.2 Example 2: (Real Intensity Image)

We consider an aerial image made up of intensities in the range $[0, 255]$. The image is of size 80 x 80. The original image is in Figure 6.2a [1]. We apply our multiscale segmentation method to this image. The **best** local minimum at the coarse level 4 x 4 involving N = 400

[1] the authors acknowledge J. M. H. duBuf for providing the image

blocks for s = 3 is displayed in Figure 6.2b. The final result after cleaning is given in Figure 6.2c. Each iteration at level 4 x 4 took about 1 to **3** sec on a Sparc machine and the number of' iterations for obtaining each local minimum varied between 20 to 50 iterations. The time per iteration increases with the number of classes but not significantly. Note that the central circular region as well as the boundaries of the regions with different textures are captured reasonably well. The patches appearing within the seemingly uniform regions of the cleaned image correspond to regions of a different texture. This can be observed on closer examination of the original image.

Choice of s

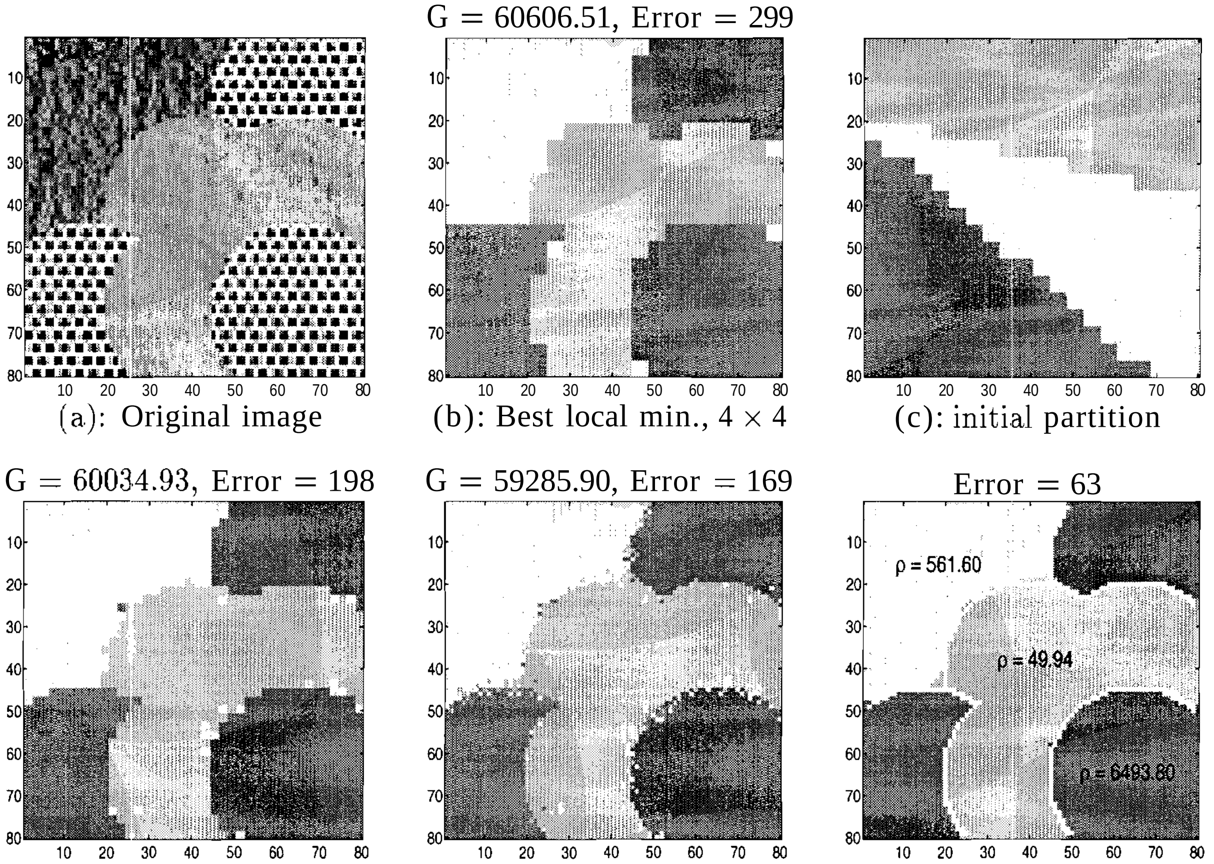
The values of H_s for s = 2.3 and 4 are $57239.3, 55863.8$ and 56052.9 respectively. The value of H_s is minimum for s = 3 and the corresponding segmentation result looks quite reasonable. We chose $L_k = 256, k = 1, \cdots,$ s to he the range of the parameters of each class.

6.3 Example 3: (Real Intensity Image)

We consider a Baboon image made up of intensities in the range [0, 255]. The image is of size 256 x 256. The original image is in Figure 6.3a We apply our multiscale segmentation method to this image. The **best** local minimum at the coarse level 4 x 4 intolving $N = 4096$ blocks for s = 3 is displayed in Figure 6.3b. The segmentation result at the pixel level 1 x 1 is given in Figure 6.3c. The final result after cleaning is given in Figure 6.3d. Each iteration at level 4 × 4 took about 19 sec in this case and the number of iterations for obtaining each local minimum varied between 30 to 60 iterations.

Note that the segmentation is quite good capturing the eyes, the two highlights on the nose and the hair regions well. The hair regions on the sides are both classified to the same class. In Figure 6.3c, the hairs were captured clearly before getting smoothed in Figure 6.3d. The segmentation seems reasonable with the visually *equivalent* regions being classified to the same class.

We show the results of unsupervised segmentation from [30] in Figures 6.3e and 6.3f. Figure 6.3e is the segmentation with 4 classes and Figure 6.3f is the segmentation with 6 classes. Our result clearly is much superior to those segmentations. Oui method is fully unsupervised with no arbitrary parameters. We also haven't used any random field models for the class label distributions.



(a): Original image

(b): Best local min., 4 × 4

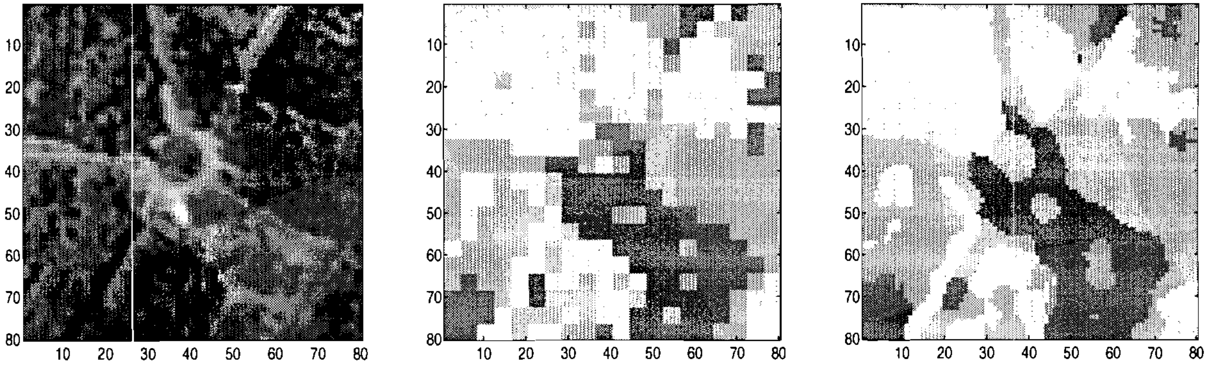
(c): initial partition

(d): Best local min., 2 x 2

(e): Best local min., 1 × 1

(f): Cleaned image

Fig. 6.1. [mage segmentation on texture image, N = 6400, for $s = 3$. (b) Classification at scale 4 x 4 (c) Initial partition that gave (b). (d) Classification at scale 2 × 2 starting from (b). (e) Classification at scale 1 x 1 starting from (d). (f) Cleaned version of (e).

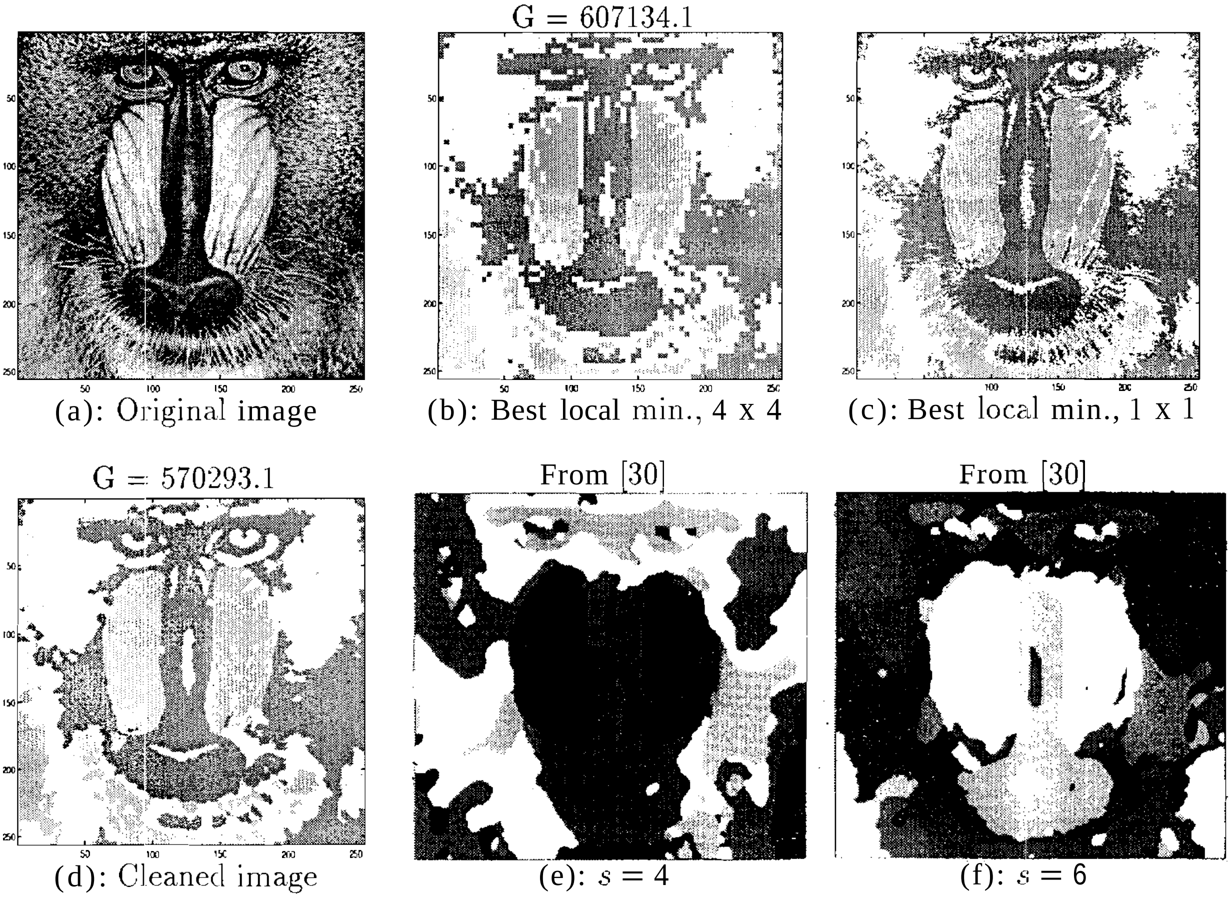


(a): Original image

(b): Rest local min., 4 × 4

(c): Cleaned image

Fig. 6.2. Segmentation of aerial image, N = 6400, for $s = 3$. a) Original image. b) Classification at scale 4 × 4. c) Cleaned final result.



G = 607134.1

G = 570293.1

From [30]

From [30]

(a): Original image

(b): Best local min., 4 x 4

(c): Best local min., 1 x 1

(d): Cleaned image

(e): $s = 4$

(f): $s = 6$

Fig. 6.3. Segmentation of Baboon image, N = 65536, $s = 3$. (a) Original image. (b) Classification at scale 4 x 4. (c) Classification at scale 1 x 1 starting from (b). (d) Cleaned version of (c). (e) Results given in [30] for $s = 4$. (f) Results given in [30] for s = 6.

7. COMPARISON AND DISCUSSION

7.1 Discussion

There are several clustering algorithms [3, 4, 5, 6, 28, 29]. We will focus only on one or two of them and relate them to the algorithm developed in the paper. The choice of s, the number of' classes, has received some attention [29, 19], but there are no definitive results.

The self-organizing feature map proposed by Kohonen is quite popular for unsupervised classification in the neural network literature. This method is essentially a clustering technique using a Euclidean metric. For a given data sample, only the neuron closest to it and those in a small neighborhood around it are allowed to change. The neighborhood needs to be specified which introduces an arbitrary parameter that has to be chosen by the user.

In the fuzzy c-means clustering [3], the problem is posed, not as one of chosing a partition, but of using related variables u_{ik} defined as the degree of belonging of z_i to class k,

$$J_2(\mathcal{U},\phi) = \sum_{k=1}^{s}\sum_{i=1}^{N} u_{ik}^{\alpha}\|z_i - \phi_k\|^2, \quad \sum_{k=1}^{s} u_{ik} = 1 \tag{7.1}$$

where $\phi = \{\phi_1, \cdots, \phi_s\}, \phi_k \in R^m$ and $\mathcal{U} = \{u_{ik}, i = 1, \cdots, N; k = 1, \cdots, s\}, u_{ik} > 0$ and $2 \leq \alpha < \infty$. $J_2(\mathcal{U},\phi)$ is not convex. Hence it has numerous local minima in $\mathcal{U}$-ϕ space. The authors [4] use a gradient approach which yields only a stationary point at which the first, derivative:; of J_2 w.r.t $\mathcal{U}$ and ϕ are zero. The authors claim that it is a strict local minimum and not just a stationary point.

Further. the structure of the cost criterion in (7.1) is such that there is no explicit reference to a cluster. The shape and orientation of a cluster are not modeled by it. Extensions to the above cost criterion involve using matrix norms. But they too suffer from the same criticism as above namely lack of explicit clusters. Also the matrix norm has to be defined for the entire data which automatically becomes insensitive to shape and orientation of individual clusters. Also, it does not exploit the prior information of y and x coordinates, for instance if the point's are clustered around a line. This was evident in the target tracking example. The criterion in (1.3) accounts for the shape of the cluster by employing the covariance matrix, which is data dependent: in the cost criterion.

Gray et al [5] consider a criterion function which implicitly recognizes the partition.

$$J_3(\boldsymbol{c} = \{\boldsymbol{c}_1, \cdots, \boldsymbol{c}_s\}, \phi) \; = \; \sum_{i=1}^{N} \sum_{k=1}^{s} \|\boldsymbol{z}_i - \phi_k\|^2 \tag{7.2}$$

They get *it* local minimum. There has been no attempt in their paper to begin with several different initial partitions. They instead propose an annealing like technique and show convergence to the global minimum in the 1D case.

Because of all these limitations, the results of these algorithms are mixed, i.e there are many cases especially with 2D data where the suggested solutions do not appear to be appropriate.

The methods for solving unsupervised classification based on statistical theory begin with the mixture density [22, 23]. Then. the unsupervised learning problem is equivalent to identification of finite mixtures. Although Gaussian mixtures are shown to be identifiable, the Bayesian method has been applied only to 1D data restricted to only two classes; typically in digital communications. Each $\boldsymbol{z}_i$ is assumed to obey the mixture density

$$p(\boldsymbol{z}_i; \boldsymbol{\theta}) = \sum_{k=1} p_k(\boldsymbol{z}_i \mid \boldsymbol{\theta}_k) \; \alpha_k \tag{7.3}$$

where 8 = $(\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_s)$, α_k = fraction of the k^t class in the entire sample, $0 \leq \alpha_k \leq 1$, $\boldsymbol{k} = 1, \cdots, s$, $\sum_{k=1}^{s}$ a; $= 1$. Since all $\boldsymbol{z}_i$ are independent, the joint density is

$$H(Z; \boldsymbol{\theta}, \boldsymbol{\alpha}) = \prod_{i=1}^{N} \left(\sum_{k=1}^{s} p_k(\boldsymbol{z}_i \mid \boldsymbol{\theta}_k) \; \alpha_k \right) \tag{7.4}$$

The unknowns are the paramters $\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_s$ in the densities p_k and α_k, k $= 1, \cdots, s$. The estimate which maximizes $H(Z; \boldsymbol{\theta}, \boldsymbol{\alpha})$ w.r.t 8 and $\boldsymbol{\alpha}$ is the desired estimate. If each $\boldsymbol{\theta}_k$ is of dimension n, we are dealing with the unknowns of dimension $(s + 1)n$. $(-\ln H)$ has numerous local minima. Computing the various local minima is itself formidable.

Even solutions when the dimension of $\boldsymbol{z}_i$ is one have not been very successful. The meaning of density (7.4) is not clear. If $\alpha_1, \cdots, \alpha_s$ are the prior probabilities of the s classes(fixed before getting the data) then the mixture density is the density of $\boldsymbol{z}_i$ in Bayesian sense; i.e the class label of each observation as a random variable with prior probabilities $\alpha_1, \cdots$, a,. But in the strict likelihood reasoning. we should regard all the class labels of the N points z_i as unknown. For example in the robust estimation. each data point is a good data or noise data, nothing in between. The correct ML estimate is the estimate given in this paper, as mentioned in section II.

There are several methods for image segmentation [14, 15, 16, 17, 18. 19]. Some of the clustering techniques have been applied for image segmentation, with the associatecl disadvantages discussed earlier. In the stochastic model-based methods the different classes are modeled as random fields and the segmentation problem is posed as a statistical optimization problem. Some of the existing methods employ Gauss-Markov models or Gaussian mixture models for the class densities P(Z | $s, \boldsymbol{c}. \boldsymbol{\theta}$), discrete Markov random fields for the class label distribution $P(\boldsymbol{c} \mid s)$ and use the expectation maximization(EM) algorithm to obtain the maximum likelihood estimates of the unknown distribution parameters. Since the EM algorithm involves the generation of a sample realization from the class label field at every iteration, it is essentially Monte-Carlo in nature and also computationally expensive. The aim of most of the existing statistical techniques is to obtain the parameters of the class clistributions and the class label field distribution. Using these parameters a final hard segmentation is derived. The problem of multiple local minima in the estimation of the parameters exists and a systematic search for best local minima hasn't been investigated. One of the key requirements for such a procedure is a method of validating the partitions obtained. We provide a method based on the Bayes formulation for comparing and validating partitions. An attractive feature of our method is that the partitions being compared need not contain the same number of classes.

We also employ a multiscale method that reduces computation significantly and gives good results at the pixel level. Our method differs from the existing methods in its approach. In the multiscale methods currently employed the parameters at different scales are computed using appropriately decimated statistics at the finest level. In our method, since the parameter estimation step is a direct computation, we compute them at the pixel level. Only the class labels are assigned jointly to blocks of varying sizes at various scales.

The number of classes are usually assumed to be known. Otherwise it is estimated based on information theoretic criteria such as Akaike's information criterion. These are mostly based on likelihood of the data with an additional penalty term. A comparison of various such criteria is given in [30]. In our case, we have the data partition, class parameters and the number of classes combined jointly into the Bayesian formulation. The optimal number of classes can be infered from the MAP estimate of the number of classes.

Another very attractive feature of our method is its ability to take an existing segmentation and iteratively improve it by computing the MAP estimates of the partition and the associated parameters. This can be applied to video segmentation. Typically video is segmented for various applications such as compression, tracking objects for content characterization and content based retrieval etc. In video, the difference between two successive frames is small when there is no scene change. Hence their segmentation:; will be close to each other. Our method can take a good segmentation that is already available ,that of the previous frame. as an initial condition and find the nearest local minimum.

7.2 Conclusion

We proposed a solution to the problem of unsupervised classification of multidimensional data based on Bayesian estimation. The new feature of our method is, we regard the data partition as a variable to be estimated. We developed a Bayesian framework to estimate the number of classes, the class parameters and the data partition simultaneously. The cluster validation problem was formally addressed. We addressed the robust regression problem treating it as a two class unsupervised classification problem. The breakdown point obtained was as high as 80%. We investigated several examples including the image segmentation problem. The advantage of our method is a single formulation can handle data clustering, data cleaning and image segmentation. The examples on natural images illustrate the power of our method. It is worthwhile noting that we haven't used any Markov random field models for class labels. We used only a facet model for intensity values which amounts to assuming that they are independently distributed random variables with varying mean.

Future work will include applications of the current method for video segmentation to detect objects and facilitate content based retrieval. Using our method we can segment the first frame in a shot and use that segmentation as starting partition for the second frame. Similarly the segmentation obtained for frame 2 can he used as starting partition for frame **3** and so on. Since the number of classes remains almost the same when there is no scene change we can gain significantly in terms of computation time.

LIST OF REFERENCES

[1] D. F. Andrews, P. J. Bickel and I. D. Know, "Robust Estimates of Location: Survey and Advances," Princeton Univ. Press, 1972.

[2] R. O. Duda and P. E. Hart, *Pattern Classification* and Scene *Analysis.* New York: Wiley, 1973.

[3] E. Ruspini, ..Numerical Methods for Fuzzy Clustering," *Inf. Sci.*, Vol. 2, pp. 319–350. 1970.

[4] J. C. Bezdek, Pattern recognition *with fuzzy objective* function *algorithms.* New York: Plenum Press. 1981.

[5] Y. Lincle, A. Buzo and R. M. Gray, "An algorithm for vector quantizer design," IEEE Trans. on Comm., COM-28, pp. 84–95, January 1980.

[6] T. Kohonen, "The self-organizing map," *Proceedings* of *the* IEEE, Vol. 78. pp. 1464–1480, 1990.

[7] R. Sokal ancl C. D. Michner, "A Statistical Method for Evaluating Systematic Relationships," *Univ. Kansas* Sci. Bull., Vol. 38, pp. 1409–1438, 1958.

[8] J. Wolfe, "Pattern Clustering by Multivariate Mixture Analysis," *Multivar. Behav.* Res., Vol. 5, pp. 329–350, 1970.

[9] R. L. Kashyap and K. B. Eom, "Robust image models ancl their applications," Advances in Electronics and Electron Physics, Vol. 70, Academic Press, San Diego, CA. pp 80–157

[10] R. L. Kashyap, "Inconsistency of the AIC rule for estimating the Order of Autoregressive Moving Average Models," IEEE *Trans. Automat. Contr.*, Vol. AC-25, No. 5, October 1980.

[11] P. Meer, D. Mintz. D. Y. Kim and A. Rosenfelcl, "Robust Regression Methods for Computer Vision: A Review," *Int'l. Jl.* of *Comp.* Vision, Vol. 6, No. 1, pp. 59–70, April 1991.

[12] R. L. Kashyap and K. B. Eom, "Robust Image hlodeling Techniques With an Image Restoration Application," IEEE *Trans. Acoustics,* Speech, *and* Signal Processing, Vol. 36, No. 8, pp. 1313–1325, August 1988.

[13] H. R. Rabiee and R. L. Kashyap, "Adaptive Multiresolution Image and Video Compression and Pre/Post-Processing of Image and Video Streams," PhD thesis, Purdue University, December 1996.

[14] R. M. Haralick, "Statistical and structural approaches to texture," *Proc.* IEEE, Vol. 67, pp. 786–804, May 1979.

[15] H. Derin and H. Elliot, "Modeling and segmentation of noisy and textured images using gibbs random fields," IEEE *Trans.* Pattern *Analysis and Machine* Intelligence, Vol. 9. No. 1, pp. 39–55, January 1987.

[16] J. Zhang, J. Mi. Modestino and D. A. Langan, "hlaximum-likelihood parameter estimation for unsupervised stochastic model-based image segmentation." IEEE *Trans.* Image Processing, Vol. 3, pp. 403–419, July 1994.

[17] S. Lakshmanan and H. Derin, "Simultaneous parameter estimation and segmentation of Gibbs random fields using simulated annealing," IEEE Trans. Pattern *Analysis* and *Machine Intelligence*, Vol. 11, No. 8, pp. 799–8113, 1989

[18] S. L. Sclove, "Application of the conditional population-mixture model to image segmentation," IEEE *Trans.* Pattern Analysis and *Machine* Intelligence. Vol. 5, pp. 428–433. July 1983.

[19] C. A. Bouman and B. Liu, "Multiple Resolution Segmentation of Textured Images," IEEE Trans. *Pattern Analysis* and *Machine* Intelligence, Vol. 13, No. 2, pp. 99–113, February 1991.

[20] A. Satish and R. L. Kashyap, "Estimation of Singularities for Intercept Point Forecasting," *IEEE* Trans. on Aerospace *and* Electronic *Systems*, Vol. 32, No. -!, pp. 1301–1309, October 1996.

[21] K. Wang, "An assessment of tactical surface-to-air missile midcourse guidance technology," In *Proceedings of the* 19.91 American *Control* Conference, Vol. 1, pp. 854–855, 1991.

[22] J. Spragins, "Learning without a teacher," IEEE Trans. *Information Theory*, Vol. IT-12, pp. 223–230, April 1966.

[23] S. Yakowitz, "Unsupervised learning and the identification of finite mixtures," IEEE *Trans.* Information Theory, Vol. IT-16, pp. 330–338, May 1970.

[24] P. J. Rousseeuw, "Least Median of Squares Regression," *Jl. Amer.* Stat. Assoc., Vol. 79, No. 388, pp. 871–880, December 1984.

[25] P. J. Rousseeuw and A. M. Leroy, Robust *Regression* and Outlier *Detection.* New York: John Wiley, 1987.

[26] F. R. Hampel, P. J. Rousseeuw, E. M. Ronchetti and W. A. Stahel, *Robust* Statistics: The Approach Based on *Influence* Functions. New York: John Wiley, 1986.

[27] A. Jain and R. Dubes, *Algorithms for Clustering Data.* New Jersey, Prentice Hall, 1988.

[28] I. Gath and A. Geva, "Unsupervised optimal fuzzy clustering," *IEEE Trans. Pattern Analysis and Machine Intelligence,* Vol. 11, pp. 773–781, 1989.

[29] G. Beni and X. Liu "A least biased fuzzy clustering method," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 16. No. 9, pp. 954–960, September 1994.

[30] D. A. Langan, J. W. Modestino and J. Zhang, "Cluster validation for unsupervised stochastic model-based image segmentation," *IEEE Trans. on Image Processing*, Vol. 7, No. 2, pp. 180–195, February 1998.