Data Mining and Knowledge Discovery, 7, 215–232, 2003

Sampling and Subsampling for Cluster Analysis in Data Mining: With Applications to Sky Survey Data

DAVID M. ROCKE AND JIAN DAI
Center for Image Processing and Integrated Computing, University of California, Davis, CA 95616, USA

Editors: Fayyad, Mannila, Ramakrishnan

Received May 4, 1999; Revised March 8, 2000

Abstract. This paper describes a clustering method for unsupervised classification of objects in large data sets. The new methodology combines the mixture likelihood approach with a sampling and subsampling strategy in order to cluster large data sets efficiently. This sampling strategy can be applied to a large variety of data mining methods to allow them to be used on very large data sets. The method is applied to the problem of automated star/galaxy classification for digital sky data and is tested using a sample from the Digitized Palomar Sky Survey (DPOSS) data. The method is quick and reliable and produces classifications comparable to previous work on these data using supervised clustering.

Keywords: clustering algorithm, mixture likelihood, sampling, star/galaxy classification

1. Introduction

Development of algorithms for automated classification of objects in massive data sets is a research area of fundamental importance to data mining and knowledge discovery in databases (KDD) (Fayyad, 1997). In this paper we describe a clustering algorithm based on finite mixture models for unsupervised classification for large databases. This new method combines the mixture likelihood approach with a sampling and subsampling strategy in order to reduce model estimation time and increase search efficiency. Although the sampling strategy is paired here with a particular clustering algorithm, it is applicable to a broad range of data mining methods. The study was motivated by the problem of classifying astronomical objects detected from images of sky surveys; and our clustering algorithm will be tested using a sample data from the Digitized Palomar Sky Survey (DPOSS).

1.1. Surveying the cosmos

For a number of years, the Oschin Schmidt 48 inch telescope at Palomar was dedicated to work on the second Palomar Observatory Sky Survey (POSS II). POSS II will eventually cover 894 fields spaced 5 degree apart in three passbands: blue (IIIa-J+GG 395), red (IIIa-F+RG 610), and near-infrared (IV-N+RG 9). The survey covers the entire northern sky. The resulting data, the Palomar-STScI Digital Sky Survey (DPOSS), will consist of some 3 terabytes of data—enough information to fill 6 million books. The Palomar-Norris

Sky Catalog is expected to contain approximately 50 million galaxies, more than 2 billion stars, and over 100,000 quasars (Fayyad et al., 1996). Data from the sky survey will help scientists understand fundamental aspects of the universe, such as the large scale distribution of galaxies, the rate of their evolution, and the structure of our own galaxy. More sky surveys are under way or coming, including the Sloan Digital Sky Survey (SDSS), the 2 Micron All Sky Survey (2MASS), and the NRAO VLA Sky Surveys (NVSS/FIRST).

1.2. Automated cataloging of sky surveys

Sky surveys result in mountains of data. The first step in making the data useful is to identify, measure, and catalog all of the detected objects into their respective classes. Once the objects have been classified, scientific analysis can proceed. Historical methods of cataloging are not effective for classifying image features from sky surveys due to two severe limitations: (1) They cannot identify objects that are very faint. The fact is that the majority of objects in a DPOSS image are too faint for traditional recognition algorithms, or even object-by-object classification by eye; (2) They are too time consuming to be used for classifying massive data sets. New methods for automated classification of data can be generally classified as supervised or unsupervised methods. The supervised methods require that a human expert both has determined into what classes an object may be categorized and also has provided a set of sample objects with known classes (White, 1997). An example of supervised classification is the Sky Image Cataloging and Analysis Tool (SKICAT) (Fayyad et al., 1996), which is based on decision trees and associated methods of machine learning and uses a training set to derive the classification rules. SKICAT has been successfully applied to DPOSS. An example of unsupervised classification is AutoClass (Cheeseman and Stutz, 1996), which is based upon the classical mixture model, supplemented by a Bayesian method for determining the optimal classes.

1.3 Unsupervised classification: Clustering via mixture models

Unsupervised classification methods assume no or very limited prior knowledge about the underlying structure of the data and require no training set. Thus, those methods are often more difficult but can be used in a much wider set of circumstances, since one does not require a training set of already correctly classified data points in order to proceed. Furthermore, unsupervised clustering methods can reveal structure not previously known, whereas supervised methods can only classify based on a prior known structure. Both types of methods have their place. We address the problem of unsupervised classification. We only use prior results on supervised classification to provide a kind of upper bound on potential performance; since it would be unreasonable to expect an unsupervised method to achieve better results when the structure is known than supervised methods, which utilize much more information.

In the mixture model approach to cluster analysis, the data are assumed to come from a mixture of probability distributions, each representing a different group or cluster. By adopting some parametric form for the density function in each underlying cluster, a likelihood can be formed in terms of the mixture density, and unknown parameters estimated

by consideration of the likelihood. A probabilistic clustering of the objects can be obtained using the estimated posterior probabilities of group membership. Classification can be done by allocating each object to the group to which it has the highest estimated posterior probability of belonging. For multivariate data of a continuous nature, the multivariate normal (Gaussian) density is often a convenient choice due to its computational tractability.

There is an additional rationale for Gaussian likelihood methods, or at least elliptical likelihood methods. Although cluster shapes are known often to be non-elliptical, in high dimension a second-order approximation to the shape of the cluster (i.e., an ellipsoid) may be all that can reasonably be expected from even a very large sample of data. Consider the problem in dimension 10 (a relatively low dimension). An ellipsoidal model contains 10 parameters for the center and 55 for the shape matrix, for a total of 65 parameters. We can reasonably expect to estimate such a model with a few hundred to a few thousand points. But what if the shape is arbitrary? To determine shape in dimension 1 requires perhaps a minimum of 20 points so that the histogram is well determined. This means that in dimension 10, we will need about $20^{10} = 10{,}240{,}000{,}000{,}000$ (ten trillion!) points for a nonparametric shape estimate. Thus, Gaussian likelihood, while not a perfect reflection of reality in many cases, is an approximation about as good as can be accomplished. Of course, individual variables may need to be transformed such as by taking logarithms to improve the approximation, but such transformations can often be identified from the distribution variable-by-variable.

The Gaussian mixture likelihood approach to clustering is easy to implement and program (McLachlan and Basford, 1988). It has a number of appealing features and has shown promise in a number of practical applications such as character recognition, diabetes diagnosis, identification of textile flaws from images, minefield and seismic fault detection, and tissue segmentation (see Fraley and Raftery (1998) for a list of recent applications).

1.4. Research objectives

In clustering via mixture models, the values of unknown parameters are estimated using the expectation-maximization (EM) algorithm (Dempster et al., 1977). The EM algorithm is a general approach to the iterative computation of maximum likelihood estimates and is useful for a variety of incomplete-data problems (McLachlan and Krishnan, 1997). A problem with the mixture likelihood approach is that the likelihood equation usually has multiple roots, and therefore, we need to search for those local maxima. The approach also requires the specification of an initial value for the unknown parameters, or equivalently, of an initial classification of the data with respect to the components of the mixture model being fitted. In other words, a good initial value is needed to get a good solution. However, in data-mining applications, we usually do not have prior knowledge about what the initial values ought to be and, therefore, have to search for them. In clustering massive data sets, an effective search strategy is critically important due to constraints on computation time.

Generally, one cannot easily search directly for the multiple maxima of the likelihood function. Descent methods of various kinds (such as the EM algorithm) can locate a local maximum, but there is no known method for searching in the parameter space in finite time and locating the global maximum.

In this paper we propose to use a sampling and subsampling strategy for searching for local (and potentially global) maxima of the likelihood function and also for locating suitable initial values for starting the algorithm. This strategy will allow us to save significant computation time while almost guaranteeing getting a good solution. We use extensive computational experiments to evaluate the strategy and its parameter values. We test the new algorithm using a DPOSS sample data and simulated data.

The rest of paper is organized as follows. Section 2 discusses the methodology for this study and presents our approach to the problem. Section 3 describes the DPOSS sample data and presents the results of clustering this digital sky data. Section 4 provides the results of experiments based on simulated data. Section 5 discusses the limitations of this study and issues for further research.

2. Methodology

2.1. *Mixture likelihood approach to clustering*

Methods of cluster analysis seek to separate data into its constituent groups or clusters. Some well-known methods such as discriminant analysis assume that the classification scheme is known a priori. We consider the more difficult techniques of cluster analysis which can apply in situations where there is no or very limited knowledge about the underlying group structure. Furthermore, we are interested in clustering methods that are practical in analyzing large databases. A promising approach to the problems we consider is clustering via finite mixture models. These models assume that the data are from a mixture of probability distributions, each representing a different cluster. The models can be fitted by the well-established statistical methods of maximum likelihood. As will be seen later, this approach can be extended to classification of large data sets.

To define the mixture likelihood, let $\mathbf{x}_1, \ldots, \mathbf{x}_n$ be the observed values of a random sample of p-dimensions and size n. Under finite mixture models, each $\mathbf{x}_i$ is assumed to come from a super population G which is a mixture of a finite number, g, of populations $G_1, \ldots, G_g$ in some proportions $\pi_1, \ldots, \pi_g$, respectively, where the mixing proportions π_j are nonnegative and sum to 1. The probability density function of (p.d.f.) an observation x in G is given by

$$f(x;\lambda) = \sum_{j=1}^{g} \pi_j f_j(x;\theta),$$

where $f_j(x;\theta)$ is the p.d.f. corresponding to the jth component in the mixture, θ is a vector of unknown parameters associated with the parametric forms that may be chosen for the density functions, and $\lambda = (\pi_1, \ldots, \pi_g, \theta')'$. Multivariate normal components are a convenient choice in practice due to their computational tractability. The normal mixture probability density function can be represented as

$$f(x;\lambda) = \sum_{j=1}^{g} \pi_j \phi(x;\mu_j, \Sigma_j), \tag{1}$$

where $\phi(x; \mu_j, \Sigma_j)$ denotes the multivariate normal p.d.f. with mean vector μ and covariance matrix Σ. With the normal mixture approach the log likelihood function for λ can be easily formed

$$\ell(\lambda) = \log L(\lambda) = \sum_{i=1}^{n} \log\left[\sum_{j=1}^{g} \pi_j \phi(x_i; \mu_j, \Sigma_j)\right], \tag{2}$$

and an estimate of $\hat{\lambda}$ of λ can be obtained as a solution of the likelihood equation,

$$\partial \ell(\lambda)/\partial \lambda = 0, \tag{3}$$

(the solution with the highest value of the likelihood is a candidate for the MLE, but there is never a guarantee that the likelihood obtained is the global maximum). Once $\hat{\lambda}$ has been obtained, the posterior probability that observation i belongs to the jth underlying group G_j can be estimated. Thus, the mixture approach gives a probabilistic clustering in terms of these estimated posterior probabilities of group membership. A partitioning of $\mathbf{x}_1, \ldots, \mathbf{x}_n$ into g nonoverlapping clusters can be effected by assigning each identity to the group to which it has the highest probability of belonging.

2.2. *Application of EM algorithm*

Mixture models can be fitted using the Expectation-Maximization (EM) algorithm (Dempster et al., 1977), which is a general method of iterative computation of maximum likelihood (ML) estimates when the data can be viewed as incomplete (Everitt and Hand, 1981; McLachlan and Basford, 1988; McLachlan and Krishnan, 1997). As with all known methods for this problem, the EM algorithm guarantees only a local maximum to the likelihood; it gives the local maximum for which the starting point of the iterations lies in the domain of attraction. The EM algorithm is applied to the mixture model by treating the observed data (unclassified data) as incomplete. The "complete" data would be $y_i = (x_i, z_i)$, where $z_i = (z_{i1}, \ldots, z_{ig})$ is a vector of indicator variables defined by

$$z_{ij} = \begin{cases} 1, & x_i \in G_j, \\ 0, & \text{otherwise.} \end{cases}$$

Each z_{ij} is independently and identically distributed according to a multinomial distribution of one draw on G categories with probabilities $\pi_1, \ldots, \pi_g$. This leads to the complete-data log likelihood

$$\log L^*(\lambda) = \sum_{i=1}^{n} \sum_{j=1}^{g} z_{ij} \log[\pi_j \phi(x_i; \mu_j, \Sigma_j)]. \tag{4}$$

The EM algorithm proceeds iteratively in two steps, E(for expectation) and M(for maximization). The E-step requires the computation of the expectation of the complete data log likelihood, conditional on the observed data and the initial value of the unknown parameters. In this step each indicator variable z_{ij} is replaced by its conditional expectation, that

is, the posterior probability that x_i is in the jth cluster ($i = 1, \ldots, n$; $j = 1, \ldots, g$). In the M-step, maximum likelihood estimates of the parameters, λ, are obtained by maximizing the estimated complete data log-likelihood. The E and M steps are alternated repeatedly. Under mild conditions, the iteration converges to a local maximum of the log-likelihood (Wu, 1983). A nice feature of the EM algorithm is that the log likelihood for the incomplete data specification can never decrease after an EM sequence. The maximum likelihood estimate is usually taken to be the root corresponding to the largest of the local maxima located.

2.3. *Research issues*

The mixture likelihood approach via the EM algorithm is easy to implement and program, and has several appealing properties. However, it also has limitations. We consider three of the problems here. First, the likelihood function tends to have multiple local maxima. Second, the EM algorithm requires the specification of an initial value. The estimates obtained and the asymptotic rate of convergence depend on the choice of the initial values. In data mining applications we usually do not know what suitable initial values are and have to search for them. We also need to search the solution space to find the multiple local maxima. In clustering large data sets, these searches can be very costly in terms of computation time, and a good search strategy is very much needed. Third, a straightforward implementation of the EM algorithm with multiple starting points has unacceptable complexity for large data sets. If $\boldsymbol{R_r}$ initial starting points are used in a sample of size N in dimension $\boldsymbol{p}$, the complexity (fixing $\boldsymbol{p}$) must be at least $O(N * R_r * i)$, where i is the number of iterations needed for convergence. If N is in the millions or billions, this is likely to be unacceptably large.

2.4. *A strategy for clustering large data sets*

Our basic purpose is to conduct an unsupervised classification of a very large data set. To do this classification, we need at the end to pass the entire data set through a classification program, but otherwise we need to insure that the computations are sublinear in the size of the data set. We reduce the complexity of the computations in a number of ways.

First, we perform all estimation on a random sample of the full data set. If this random sample is capped at a level that depends on the dimension, but not on N, then with respect to N, the computations are $O(1)$. Since our examples are relatively small (60,000–130,000 points), we use the entire data set to perform the calculations, but we would essentially never need to exceed this sample size no matter how large the data base was so long as we are searching for large clusters (and identifying small ones by their deviation from the large clusters).

The second cost saving comes from the fact that multiple starting points are needed to insure that the best chance for a global (rather than merely a local) maximum is obtained. If our evaluation sample is 130,000 points, we might select a few thousand points for trying out various starting points and iterate on the whole data set only the best result on the smaller sample. This reduces the computation essentially by the factor of the number of starting points, since the computations on the smaller sample of a few thousand are

negligible compared to the final iteration on the whole sample. Finally, and less quantifiable in a theoretical sense, we speed up the process by using highly variable starting points so as to maximize the diversity, and thus maximize the chance of hitting the global maximum in a reasonable number of starts.

In short, we propose a sampling and subsampling strategy for clustering large data sets using the mixture likelihood approach. In this strategy, most computations are done on a sample of data large enough to reflect the whole sample structure, but small enough to reduce computation time to a fraction of what would have been with the entire data set. After extensive searching in a sample, one can iterate to convergence once using the entire data set. Subsamples are used as starting points for searching within a sample. These subsamples should be small in size to generate diversity in starting points, which allows difficult local optima to be located. Generally, many subsamples will be used with each sample, but only a few samples will be used because of computation time constraints, and because use of multiple samples will not be needed if the sample size is carefully chosen. This strategy has four parameters and is implemented in a three-step estimation procedure:

Given data set of size N in dimension p, which we wish to have clustered by EM into g groups. In our first example given below, we cluster $N = 132{,}402$ points in dimension $p = 6$ into $g = 2$ groups.

Step 1: Search for a suitable initial value

1.1 Take a random sample of size N_s from the whole sample. We try several values for N_s in our examples, but this may be 1,000 out of the 132,402 total points.

1.2 Take R_r random starts consisting of N_r points randomly chosen from the sample for each of the g groups. For each start, we initialize the mean and covariance of each group with the mean and covariance of the N_r randomly selected points. Then we iterate EM from that start to convergence on the sample of N_s points. For example, our random starts in the example might consist of 10 points in dimension 6 for each of the two groups. These 10 points determine a mean and covariance matrix for each group that are used to begin the EM iteration. How many random starts are necessary depends on the problem, as we shall see below.

1.3 Keep the estimates of cluster location and shape with the highest likelihood value on the sample. We iterate to convergence from each starting point on the sample of N_r points (perhaps 1000 out of 132,402). We keep for further use, only the solution with the highest likelihood on these N_s points.

Step 2: Get a solution based on the whole data set
Perform the EM algorithm on the whole data set using the best parameter estimates obtained from Step 1.

Step 3: Repeat above steps R_s times. Report all solutions from step 2. Thus we may select the sample of N_s points (1,000 out of 132,402) several times and do all the computations. The idea is that by using multiple samples, we may be able to use smaller samples, and reduce the computational time considerably.

Some variations on this scheme are possible for extremely large data sets. For model fitting (identifying the clusters by their centers and shapes), it is hardly necessary to use

the entire data set. Therefore, step 2 would be conducted only on a supersample, which is much larger than the sample, but may be only a small fraction of the data set. The whole data set is then processed only to classify all the points according to the already developed model, so that the total processing time remains reasonable.

2.5. *Design issues*

First, we summarize the four data groupings that we use in our approach:

Data Set is the entire collection of data to be analyzed. For inferential purposes, this may be treated as having been sampled from a population. All of the data set items will be classified by the process.

Supersample is a subset of the data set chosen by simple random sampling. In our examples, it is the entire data set, but for larger data sets it will be considerably smaller. All computations prior to the final classification are performed on the supersample. For problems in moderate dimension (up to 50), the supersample will never need to be larger than 100,000–1,000,000 points, since the estimation error in a sample of this size is already too small to matter.

Sample is one of several (R_s) of size N_s chosen by simple random sampling from the supersample. All intensive search operations are conducted in the sample so that the supersample is only used for one iteration from the best solution found in the sample. The sample size N_s should be chosen to be large enough to reflect the essential structure of the data, while being small enough to keep the computations feasible. In our examples, 1000 is a reasonable choice for N_s, as we will see below.

Subsample is one of several (R_r) of size N_r chosen by simple random sampling from the sample that is used to begin iterations on the sample. As will be shown below, this number should be very small (perhaps 5–10 in dimension 4 for example) because great diversity in starting points generates diversity in solutions, and increases the chance of finding the best local maximum of the likelihood.

There are four elements in the sampling and subsampling strategy: R_s, N_s, R_r, and N_r, whose values must be jointly chosen. Small subsample size (N_r) is desirable since it would generate diversity in starting points and thus help locate difficult local maxima (Rocke, 1998). N_r can be as small as $(p + 1)$ and still generate a nonsingular covariance matrix. The sample size (N_s) needs to be large enough to reflect the whole data structure and avoid pathologies, and small enough to reduce computation time. In any case it should be at least $g * N_r$ (since that is the number of points for the starting iteration which must be a subset of the N_s sample points). Large number of random starts (R_r) would increase the chance of getting a good initial value, leading to a good solution within the sample. Since subsampling is relatively inexpensive, many subsamples can be used. Number of samples (R_s) should be small since computations involving the whole supersample are very time-consuming, and each sample in the end requires an iteration on the whole supersample.

We can characterize the success of a strategy by some criterion of merit. If a particular strategy has certain chance of getting a good solution for given values of the parameters

(R_s, N_s, R_r, N_r), then we can estimate how many samples are needed to get a good answer with high probability. In this study, we do 20 experiments for each data set (the DPOSS data set or a simulation data set). In each experiment, we try 3 sample sizes, 3 numbers of random starts, and 4 sizes of starting groups. There are 36 possibilities for each experiment and 720 in total for a data set.

3. Clustering the DPOSS sample data

3.1 Data and specification

The DPOSS sample data is a sample of JFN catalog data from a single DPOSS field for testing new data clustering algorithms (Odewahn et al., 1998). The data set consists of 648,291 objects and 27 parameters (such as magnitude, area, image moments), 9 for each of three bands: blue, red, and near-infrared (referred to as F, J, and N). An additional parameter is called CLASS, which gives the astronomers' current best guess to the type of each object (=1 for a star, =2 for a galaxy). We clustered the data without using the classification variable, but we did use it for evaluating the accuracy of our clustering results. The testing data set we used is a subset of the original data set and contains 132,402 data points, all of which have measurements on all three bands with values within the expected ranges given in the data document. We started with preliminary data analysis such as correlation analysis and Q-Q plots. It turned out that most of the 27 parameters were highly correlated. We selected following six attributes with little correlations out of total available attributes: MtotF, csfF, EllipF, csfJ, EllipJ, EllipN. The suffix capital letter in the variable names indicates the color band from which the measurement is taken. The meanings of the variables are:

Mtot	total instrument magnitude
csf	combined stellar function (measure how stellar-like the object is)
Ellip	the ellipticity

As a data mining exercise, we did not assume much prior knowledge about the data, and did not create new physically meaningful variables such as color index and concentration index by combining the basic attributes. The purpose was simply to see what the clustering algorithm could achieve with limited information.

The experiment specifications are given in Table 1. For the clustering, we assumed that the data were composed of two groups of objects (stars and galaxies). We did 20 experiments. In each experiment, three sample sizes were used: 500, 1000, and 10,000. For each sample, we tried 1, 5, or 10 random starts. A subsample of data points was used for each start. The number of points were 10, 20, 40, or N_s/g per group. The last case was designed to show the problematic nature of using large subsamples.

3.2 Classification accuracy

Three local likelihood maximum solutions resulted from the 20 experiments, which are shown in Table 2. Astronomers use completeness (fraction of galaxies classified as galaxies)

Table 1. Experiment specification (DPOSS).

Total data points (N)	132402
Variables (p)	6
Groups (g)	2
Experiments (R_s)	20
Sample size (N_s)	500, 1000, 10000
Random starts (R_r)	1, 5, 10
Starting points/group (N_r)	10, 20, 40, N_s/g

Table 2. EM solutions and classification accuracy (DPOSS sample data).

	Solution I		Solution II		Solution III	
Magnitude in F	Completeness	Contamination	Completeness	Contamination	Completeness	Contamination
⇐16.5	0.955	0.045	0.787	0.213	0.954	0.046
(16.5, 17.0]	0.972	0.028	0.914	0.086	0.954	0.046
(17.0, 17.5]	0.970	0.030	0.903	0.097	0.945	0.055
(17.5, 18.0]	0.954	0.046	0.881	0.119	0.870	0.130
(18.0, 18.5]	0.914	0.086	0.859	0.141	0.736	0.264
(18.5, 19.0]	0.899	0.101	0.876	0.124	0.677	0.323
(19.0, 19.5]	0.811	0.189	0.808	0.192	0.597	0.403
(19.5, 20.0]	0.754	0.246	0.768	0.232	0.593	0.407
(20.0, 20.5]	0.744	0.256	0.779	0.221	0.642	0.358
>20.5	0.805	0.195	0.817	0.183	0.714	0.286
All	0.844	0.156	0.826	0.174	0.684	0.316

and contamination (fraction of nongalaxies classified as galaxies) to measure the accuracy of star/galaxy separation (Weir et al., 1995). For much astronomical and cosmological research, accuracy level at approximately 90% of completeness and 10% contamination is needed.

In Table 2, the first column lists the total magnitude in the F band, second column gives completeness, and third column shows contamination. These measures of classification accuracy were obtained by comparing the classification results from our algorithm with the known classification. It is obvious from the table that the fainter the objects are, the harder to classify the objects correctly. It should be noted that the classification given in the data is the best guess by the astronomers. For obvious reasons, the original classification of very faint objects is more likely to be erroneous than for bright objects, since there is less information on which to base the classification.

Solution one has the highest likelihood value and provides the best result. It achieves the accuracy level of 90% completeness and 10% contamination in galaxy catalogs down to a magnitude (F) limit of 19.0. Even with all data including those of the faintest objects (down

to about the 24th magnitude), the solution achieves an accuracy level of 85% completeness and 15% contamination. These results are pretty close to those obtained by the astronomers using supervised classification methods.[1] Although there is no method of proving that this is the global maximum of the likelihood, from a practical point of view there is no alternative but to treat it as the correct solution.

3.3. *Evaluation of sampling strategy*

The strategy is (R_s, N_s, R_r, N_r). Given the number of experiments ($R_s = 20$), we will see how the other parameters would affect the outcome of clustering. For this purpose, we classified solutions on the whole data set as good or bad. We consider it to be a success if the best solution (the largest of the local maxima located) is obtained. Figure 1 consists of three graphs and shows the relationships between percent success and the parameters: the number of random starts R_r, the number of starting points (subsample size) N_r, and the size of sample N_s. The effects of sampling on computation time are shown in figure 2, in which estimation time is seconds per sample. In the figures, the three graphs show the results based on sample sizes of $N_s = 500, 1000, 10000$, respectively. The results are based on 20 experiments. The following observations are made from the figures.

(1) The smaller the number of starting points, the better the results. On the graphs, results obtained using the smallest subsamples are shown in squares, those with largest subsamples in circles. In figure 1, the squares always show higher percent of success than the other symbols as long as more than one subsample is taken. This is because smaller subsamples give higher diversity in starting points, and thus a better chance of finding solutions that may be more difficult to locate (and may be better).
(2) More random starts result in better outcomes. The best solution can be achieved 100% of the time by using enough random starts (e.g. 10) and small number of starting points

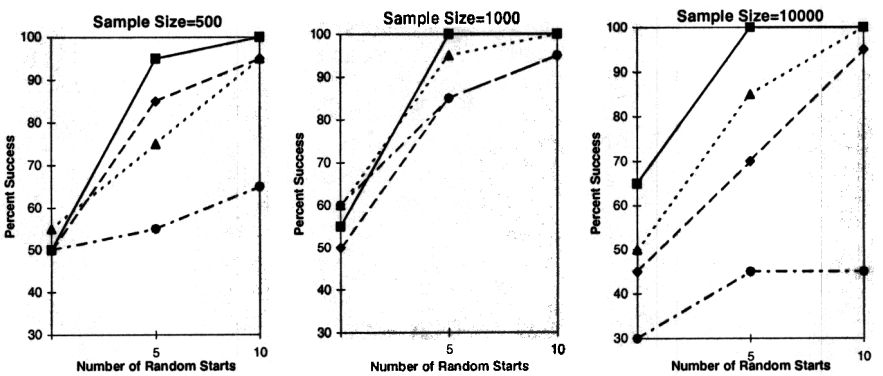

Figure 1. Percent of success (DPOSS sample data). Each plotted point shows the frequency of obtaining the best solution given a sampling and subsampling strategy. Squares correspond to subsample size $N_r = 10$, triangles to $N_r = 20$, diamonds to $N_r = 40$, and circles to $N_r = N_s/g$.

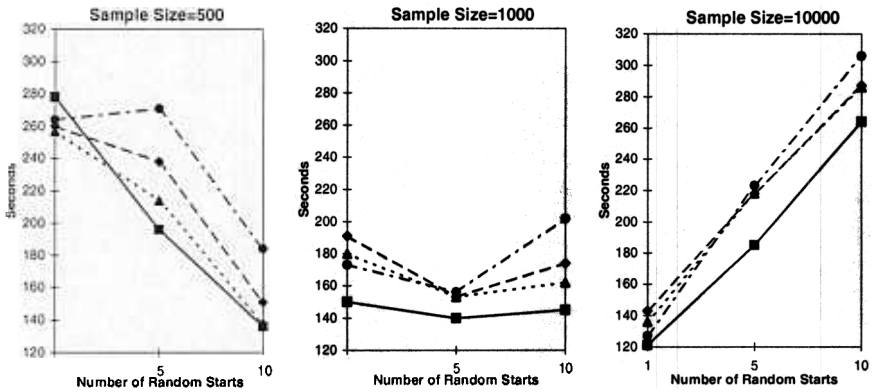

Figure 2. Computation time in seconds (DPOSS sample data). Each plotted point shows the average computation time in seconds used given a sampling and subsampling strategy. Squares correspond to subsample size $N_r = 10$, triangles to $N_r = 20$, diamonds to $N_r = 40$, and circles to $N_r = N_s/g$.

(e.g. 2p to 3p). In this case, a few runs would suffice to pretty much guarantee getting a good solution. For example, when $N_s = 1000$ and $N_r = 10$, a single trial succeeds in finding the best solution about 50% of the time. If 20 samples are used with only one random start per sample, the chance of failing to find the best solution is only about $(0.5)^{20} = .00001$. If more than one random start is used, the chance is even greater. The most conservative interpretation that is consistent with this experiment is as follows. In 5 random starts, the best solution was obtained in 20 out of 20 samples. This is consistent with binomial variation and a chance of success on each sample of between .832 and .999 (95% confidence interval). With and 83% chance of success on each sample, the chance of missing with only 5 samples is .0001.

(3) Sample size should be large enough to avoid pathology. But it does not need to be very large. Sample size around 1000 seems to be sufficient. The worst case is using large sample and large subsample at the same time. The use of too many data points would not only increase computation time but also degrade the performance of the estimator.

(4) Sample size has significant impact on computation time. Sample size around 1000 seems to be optimal. When it is smaller (e.g. 500), increase of the number of random starts (up to a point) tends to reduce time, most likely due to improvement of starting values, which reduces the iteration time on the whole data set. When it is larger (e.g. 10000), more random starts would rapidly increase computation time because each random start requires iteration on the whole sample, which at 10,000 is a significant fraction of the data set.

Those results are consistent with our expectations. It is suggested that our sampling and subsampling strategy works well in clustering large data sets. With the best strategy, a good solution can be obtained in less than two and half minutes. Our algorithm seems to be fast and reliable.

In our schema, the supersample size does not rise indefinitely with the data set size, so the computation time before the final classification is capped at $O(1)$. With respect to the supersample size N, for a fixed dimension the sample size which may be determined to be 1000 will stay constant, and will become a decreasing fraction of the supersample size down to a limit. If the computation time on a sample from one random start is T_s, and the computation time on the supersample is $T_d(R_r)$, a decreasing function of the number of random starts (because a better starting point on the sample reduces the number of iterations on the supersample), then the total computation time is $T_s * R_r + T_d(R_r) * R_s$. As R_r rises from 1, the computation time decreases since $T_d(R_r)$ can fall rapidly with R_r and is much larger than T_s. After a point, further increases in R_r do not reduce T_d much, but $T_s * R_r$ still increases.

4. Further experiments

There are certain limitations in experiments with real world data due to factors such as errors in the data, measurement bias, and variable selection. Therefore, we conducted further tests using simulated data. We generated 60,000 data points with three clusters and four variables. Each cluster consisted of 20,000 data points following a multivariate normal distribution. The mean vectors for the three groups of data were (10,0,0,0), (20,0,0,0), (30,0,0,0), and the identity covariance matrix was used for all three clusters. One advantage of using simulated data is that we know the true membership of each data point. This information is useful for evaluating the accuracy of the clustering. The specifications of the experiments are shown in Table 3. The design is similar to that used in clustering the DPOSS sample data.

The results from 20 experiments (and hundreds of attempts to locate a maximum of the likelihood) show that there are 8 solutions to the EM problem. The solution with the highest log likelihood value correctly classifies all data points. We call this solution the best solution. Fractions of correct classification of the rest solutions are 76 percent or less. For the purpose of evaluation, we consider it to be a success if the best solution is obtained. The results are summarized in figures 3 and 4. Figure 3 shows how the sampling strategy affects the rate of success. Figure 4 shows how it affects computation time (seconds per sample). The results are very similar to those obtained using the DPOSS sample data.

(1) As demonstrated using the DPOSS data, the best strategy is to use small number of starting points and relatively large number of random starts. The best solution is pretty much guaranteed when sufficient number of random starts and small subsamples are

Table 3. Experiment specification (simulated data).

Total data points (N)	60000
Variables (p)	4
Groups (g)	3
Experiments (R_s)	20
Sample size (N_s)	500, 1000, 60000
Random Starts (R_r)	1, 5, 10
Starting points/group (N_r)	5, 10, 100, N_s/g

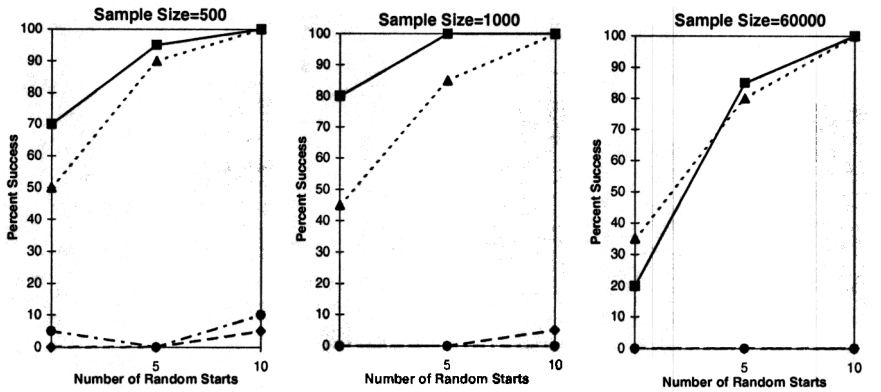

Figure 3. Percent of success (simulated data). Each plotted point shows the frequency of obtaining the best solution given a sampling and subsampling strategy. Squares correspond to subsample size $N_r = 5$, triangles to $N_r = 10$, diamonds to $N_r = 100$, and circles to $N_r = N_s/g$.

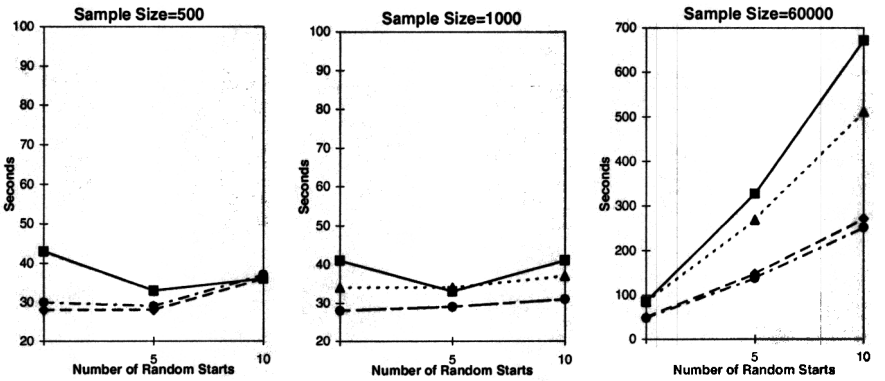

Figure 4. Computation time in seconds (simulated data). Each plotted point shows the average computation time in seconds used given a sampling and subsampling strategy. Squares correspond to subsample size $N_r = 5$, triangles to $N_r = 10$, diamonds to $N_r = 100$, and circles to $N_r = N_s/g$. Note that the plot for sample size 60000 is in different scale.

used. With this particular data set, the use of 10 random starts and 5 to 10 starting points per cluster results in 100 percent success. On the other hand, use of large subsamples has little chance of getting the best solution.

(2) Sample size should be adequate, sufficiently large to avoid pathology but not too large. Figure 4 shows that the use of whole data set (the largest sample possible) is very expensive in terms of computation time, especially with large number of random starts, and it does not necessarily make the results better. In the experiments sample sizes around 500 or 1000 worked very well.

It will never be possible to provide complete assurance that the best solution has been found. It is also not possible to recommend with certainty parameter choices for unknown situations. Nevertheless, some general guidelines emerge. First, subsample sizes should be very small to insure diversity of starting points. Perhaps as small as $p + 1$, but certainly no larger than $2p$, where p is the dimension of the data. Second, the sample on which most of the computation occurs should be a function of the dimension and the number of parameters to be estimated, but no larger than necessary. The simulation was in dimension 4, where there are 4 mean parameters and $4 + (4)(3)/2 = 10$ covariance parameters per group, for a total of 28 parameters. 500 point samples are sufficient. For the DPOSS data in dimension 6, the number of parameters is 54 and 1000 points seems sufficient. Further research is needed to determine levels required for high dimension. Finally, the number of random starts per sample can be fairly large, since it has only a small effect on computation time (and may even reduce it). Thus for data in dimensions below about 10, we can recommend sample sizes of 500–2,000, subsample size of $2p$, and 10–20 random starts per sample.

5. Summary and discussions

In this paper we have introduced a new clustering algorithm for automated classification for large data sets such as those from digital sky surveys. It is an unsupervised method based on finite mixture models and can be estimated using maximum likelihood via the EM algorithm. The proposed method uses a sampling and subsampling strategy to search for multiple local maxima and good initial values. Using this strategy, most computations are done on a sample of data so that computation time can be reduced to a fraction of what it would have been with the entire data set.

We have applied this clustering method to star/galaxy classification. The testing data set is a subset of DPOSS sample data and consists of 132,402 objects and 6 variables. Though the results are preliminary, we were able to achieve the accuracy level of 90% completeness and 10% contamination down to a magnitude limit of $\sim$19.0 in F, using the catalog data as the reference. The computation can be done in as little as less than two and half minutes in a SGI Origin 2000 server. Computation time can be further reduced by using parallel processing techniques. The simulated data contain 60,000 data points, 4 variables and 3 clusters. In clustering the simulated data, 100% correct classification is achieved and the computation can be done in 40 seconds. The results of experiments using the DPOSS sample data and the simulated data show that the new method is promising: it can produce good classification; and it is fast and reliable.

This study is our first step in an ongoing research aimed at developing clustering algorithms that can be applied to mining massive data sets. There are a number of limitations in this study, which may also be opportunities for further research. Some of the issues are discussed below. Assessing the number of clusters in a given data set is an important but very difficult problem. In clustering via mixture models, the number of clusters corresponds to the number of components in the fitted mixture model. A straightforward approach is to use the likelihood ratio statistic to test for the smallest number of components in the mixture model. Unfortunately, this test statistic does not have the usual asymptotic null distribution

of chi-square in mixture models (Wolfe, 1971; Titterington et al., 1985). One way to assess the null distribution is to use a resampling method in application of the general bootstrap approach, that is, to bootstrap the log likelihood ratio statistic (McLachlan, 1987). This method has been applied to assessing the number of clusters in small data sets. But it may be difficult to use in clustering massive data sets due to constraints on computation time. A similar method is proposed in Smyth (1996). An alternative approach is to use approximate Bayes factors to compare models. This approach has the advantage of having a means of selecting not only the parameterization of the model, but also the number of components or clusters (Banfield and Raftery, 1993). With the mixture likelihood method via the EM algorithm, an approximation to twice the log Bayes factor called Bayesian Information Criterion (BIC) can be derived (Fraley and Raftery, 1998). The BIC can be used to compare mixture models with different number of components. The larger the value of the BIC, the stronger the evidence for the model. The problem is that the regularity conditions for BIC do not hold for mixture models either (Fraley and Raftery, 1998), though there is theoretical and practical support for its use in mixture models. Another potential problem with the Bayesian approach is the "Occam Factor" (Cheeseman and Stutz, 1996), which implies that Bayesian parameters priors always favor classification with smaller numbers of classes and do so overwhelmingly once the number of classes exceeds some small fraction of the size of the data.

In this paper, we have used random sampling, a simple method, for search in the solution space. This method has the merit of simplicity and fastness, but it could not guarantee finding the best possible solution. There are more sophisticated search methods available such as simulated annealing (Kirkpaterick et al., 1983) and reactive tabu search (Battiti and Tecchiolli, 1994). Another possible improvement is to extend the clustering method described in this paper to modeling noise and outliers. For example, noise can be modeled as a constant rate Poisson process in certain applications. It has been shown that the mixture likelihood approach can handle data with noise and clusters (Dasgupta and Raftery, 1995). Identification of outliers in multivariate data has been studied extensively (e.g. Rocke, 1996; Rocke and Woodruff, 1996; Rousseeuw and Driessen, 1999; Woodruff and Rocke, 1994). The methodology for detecting outliers can be used to improve the robustness of the clustering methods. The capability of detecting outliers and separating noise and true objects of interests is useful in many applications such as mining digital sky data, where detecting unusual objects and finding unexpected patterns are important research objectives.

There are also other ways to use sampling to improve the performance of method such as the EM algorithm for clustering. Bradley and Fayyad (1998) and Fayyad et al. (1998) use a method of clustering clusters to achieve a similar objective to that of our paper. One advantage of the schema we describe is that it is completely general and applies well beyond clustering. For example, if a regression type model is to be used to provide predictions, the exact same schema can be used to conduct the analysis in reduced computation time, whereas the Bradley and Fayyad schema is more particularly aimed at clustering. The ideas of sampling proposed here and in the work of Bradley and Fayyad (1998) and Fayyad et al. (1998) are fundamental to the efficient analysis of massive data sets.

Acknowledgments

Research reported in this paper was supported by the National Science Foundation (DMS 95-10511, DMS 96-26843, ACI 96-19020, and DMS 98-70172) and the National Institute for Environmental Health Sciences, National Institutes of Health (P42 ES04699). The authors are grateful to two referees for suggestions that substantially improved the paper.

Note

1. Our results and those of Weir et al. (1995) are not completely comparable. They do not give results for objects brighter than the 16th magnitude. For objects in the range of about the 16th to the 19th magnitudes, our methods gave a completeness of 94.2% and contamination of 5.8%, so that the total classification error is 11.6%. Their results for these objects had a completeness of 88.4% and contamination of 5.4% for a total classification error of 17.0%, which is actually worse. However, for dimmer objects of magnitude 19 or greater, our method had a completeness of 77.9% and contamination of 22.2% for a total classification error of 44.3%, while the supervised method had a completeness of 90.8% and contamination of 17.6% for a total classification error of 26.8%, which is better. Overall, one might give the supervised method a slight edge, which would only be expected given the larger amount of information they employed.

References

Banfield, J.D. and Raftery, A.E. 1993. Model-based Gaussian and non-Gaussian clustering. Biometrics, 49:803–821.

Battiti, R. and Tecchiolli, G. 1994. The reactive tabu search. ORSA Journal on Computing, 6:126–140.

Bradley, P.S. and Fayyad, U.M. 1998. Refining initial points for k-means clustering. In Proc. 15th Int. Conf. on Machine Learning, J. Shavlik (Ed). San Francisco: Morgan Kaufman, pp. 91–99.

Cheeseman, P. and Stutz, J. 1996. Bayesian classification (AutoClass): Theory and results. In Advances in Knowledge Discovery and Data Mining, U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy (Eds.). Cambridge, MA: The MIT Press, pp. 153–180.

Dasgupta, A. and Raftery, A. 1995. Detecting features in spatial point processes with clutter via model-based clustering. Technical Report No. 195, Department of Statistics, University of Washington.

Dempster, A.P., Laird, N.M., and Robin, D.B. 1977. Maximum likelihood from incomplete data via the EM algorithm (with discussion). Journal of the Royal Statistical Society B, 39:1–38.

Everitt, B.S. and Hand, D.J. 1981. Finite Mixture Distributions. Monographs on Applied Probability and Statistics. New York, NY: Chapman and Hall Ltd.

Fayyad, U.M. 1997. "Editorial", Data Mining and Knowledge Discovery, 1:5–10.

Fayyad, U.M. 1991. On the induction of decision trees for multiple concept learning. Ph.D. Thesis, EECS Department. The University of Michigan, Ann Arbor.

Fayyad, U.M., Djorgovski, S.G., and Weir, N. 1996. Automating the analysis and cataloging of sky surveys. In Advances in Knowledge Discovery and Data Mining, U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy (Eds.). Cambridge, MA: The MIT Press, pp. 471–493.

Fayyad, U.M., Reina, C., and Bradley, P.S. 1998. Initialization of iterative refinement clustering algorithms. In Proc 4th Int. Conf. on Knowledge Discovery and Data Mining KDD-98, R. Agrawal, P. Stolorz, and G. Piatetsky-Shapiro (Eds.). Menlo Park, CA: AAAI Press, pp. 193–198.

Fraley, C. and Raftery, A.E. 1998. How many clusters? Which clustering method? Answers via model-based cluster analysis. Technical Report No. 329, Department of Statistics, University of Washington, Box 354322, Seattle, QA 98195-4322.

Hawkins, D.M. 1981. A new test for multivariate normality and homoscedasticity. Technometrics, 23:105–110.

Hawkins, D.M., Muller, M.W., and ten Krooden, J.A. 1982. Cluster analysis. In Topics in Applied Multivariate Analysis, D.M. Hawkins (Ed.). Cambridge: Cambridge University Press, pp. 303–356.

Kaufman, L. and Rousseeuw, P.J. 1990. Finding Groups in Data: An Introduction to Cluster Analysis. New York: John Wiley & Sons.

Kendall, M.G. and Stuart, A. 1963. The Advanced Theory of Statistics III. Griffin, London.

Kirkpaterick, S., Gelatt, J., and Vecchi, M.P. 1983. Optimization by simulated annealing. Science, 220:671–680.

McLachlan, G.J. 1987. On bootstrapping the likelihood ratio test statistic for the number of components in a normal mixture. Applied Statistics, 36:318–324.

McLachlan, G.J. and Basford, K. 1988. Mixture Models: Inference and Applications to Clustering. New York: Marcel Dekker.

McLachlan, G.J. and Krishnan, T. 1997. The EM Algorithm and Extensions. New York: John Wiley & Sons.

McLachlan, G.J. and Peel, D. 1998. MIXFIT: An algorithm for the automatic fitting and testing of normal mixture models. Center for Statistics, Department of Mathematics, University of Queensland, St. Lucia, Queensland 4072, Australia.

Odewahn, S.C., Djorgovski, S.G., Brunner, R.J., and Gal, R. 1998. Data from the digitized palomar sky survey. Department of Astronomy, California Institute of Technology, Pasadena, CA 91125.

Rocke, D.M. 1996. "Robustness properties of S-estimators of multivariate location and shape in high dimension. The Annals of Statistics, 24:1327–1345.

Rocke, D.M. 1998. Constructive Statistics: Estimators, Algorithms, and Asymptotics. Center for Image Processing and Integrated Computing, University of California, Davis, CA 95616.

Rocke, D.M. and Woodruff, D.L. 1996. Identification of outliers in multivariate data. Journal of the American Statistical Association, 91:1047–1061.

Rousseeuw, P.J. and Van Driessen, K. 1999. A fast algorithm for the minimum covariance determinant estimator. Techometrics, 41:212–223.

Smyth, P. 1996. Clustering using Monte Carlo cross-validation. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. Menlo Park, CA: AAAI Press.

Titterington, D.M., Smith, A.F.M., and Makov, U.E. 1985. Statistical Analysis of Finite Mixture Distributions.

Weir, N., Fayyad, U.M., and Djorgovski, S.G. 1995. Automated star/galaxy classification for digitized POSS-II. The Astronomical Journal, 109:2401–2414.

White, R.L. 1997. Object classification in astronomical images. In Statistical Challenges in Modern Astronomy II, G.J. Babu and E.D. Feigelson (Ed.). New York: Springer-Verlag, pp. 135–148.

Wolf, J. 1971. A Monte Carlo study of the sampling distribution of the likelihood ratio for mixtures of multinormal distributions. Technical Report STB 72-2, San Diego: U.S. Naval Personnel and Training Research Laboratory.

Woodruff, D.L. and Rocke, D.M. 1994. Computable robust estimation of multivariate location and shape in high dimension using compound estimators. Journal of the American Statistical Association, 89:888–896.

Wu, C.F.J. 1983. On convergence properties of the EM algorithm for Gaussian mixtures. Annals of Statistics, 11:95–103.