Clustering using Monte Carlo Cross-Validation

Padhraic Smyth*
Department of Information and Computer Science
University of California, Irvine
CA 92717-3425
smyth@ics.uci.edu

This paper appeared in the Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, OR, Aug 1996, AAAI Press, pp.126--133.

Abstract

Finding the "right" number of clusters, k, for a data set is a difficult, and often ill-posed, problem. In a probabilistic clustering context, likelihood-ratios, penalized likelihoods, and Bayesian techniques are among the more popular techniques. In this paper a new cross-validated likelihood criterion is investigated for determining cluster structure. A practical clustering algorithm based on *Monte Carlo cross-validation* (MCCV) is introduced. The algorithm permits the data analyst to judge if there is strong evidence for a particular k, or perhaps weaker evidence over a sub-range of k values. Experimental results with Gaussian mixtures on real and simulated data suggest that MCCV provides genuine insight into cluster structure. v-fold cross-validation appears inferior to the penalized likelihood method (BIC), a Bayesian algorithm (AutoClass v2.0), and the new MCCV algorithm. Overall, MCCV and AutoClass appear the most reliable of the methods. MCCV provides the data-miner with a useful data-driven clustering tool which complements the fully Bayesian approach.

Introduction

Cluster analysis is the process of automatically searching for natural groupings in a data set and extracting characteristic descriptions of these groups. It is a fundamental knowledge discovery process. *Clustering algorithms* (of which there are many) typically consist of a specification of both (1) a *criterion* for judging the quality of a given grouping and (2) a *search method* for optimizing this criterion given data (see Jain and Dubes (1988) for an overview).

A particularly vexing question, which is often glossed over in published descriptions of clustering algorithms, is "how many clusters are there in the data ?". Formal methods for finding the "optimal" number of clusters are few. Furthermore, "optimality" can be difficult to pin down in this context without some assumptions being made. One viewpoint is that the problem of finding the best number of clusters is fundamentally ill-defined and best avoided (cf. Gelman et al , page 424, in a mixture modelling context). While we sympathize with this view we adopt a more pragmatic approach in this paper, namely, let the data tell us as much as possible about cluster structure, including the number of clusters in the data. If either the data are too few, or the measurement dimensions too noisy, then the data may not reveal much. However, when the data contain interesting structure one seeks an algorithmic technique which can reveal this structure. A fundamental point is that the process of structure discovery in data needs to be interactive, i.e., the data analyst must interpret the results as they see fit.

In this paper we limit our attention to Gaussian mixture models: however, any probabilistic clustering model for which a likelihood function can be defined is amenable to the proposed approach. The method could conceivably be extended to clustering algorithms which do not possess clear probabilistic semantics (such as the k-means family of algorithms), but this is not pursued here.

Probabilistic Clustering Using Mixture Models

Finite Mixture Models

The probabilistic mixture modelling approach to clustering is well-known: one assumes that the data are generated by a linear combination of component density functions resulting in a mixture probability density function of the form:

$$f_k(\underline{x}|\Phi_k) = \sum_{j=1}^{k} \alpha_j g_j(\underline{x}|\underline{\theta}_j) \qquad (1)$$

where $\underline{x}$ is a particular value of a d-dimensional feature vector $\mathbf{X}$, k is the number of components in the model, $\underline{\theta}_j$ are the parameters associated with density component g_j, the α_j are the "weights" for each component j, and $\Phi_k = \{\alpha_1, \ldots, \alpha_k, \underline{\theta}_1, \ldots, \underline{\theta}_k\}$ denotes the set of

*Also with the Jet Propulsion Laboratory 525-3660, California Institute of Technology, Pasadena, CA 91109.

parameters for the overall model. We will adopt the notation that $\hat{\Phi}_k$ denotes parameters which have been *estimated* from data. It is assumed that $\sum_j \alpha_j = 1$ and $\alpha_j > 0, 1 \leq j \leq k$.

The component density functions are often assumed to be multivariate Gaussian with parameters $\underline{\theta}_j = \{\underline{\mu}_j, \Sigma_j\}$ where $\underline{\mu}_j$ and Σ_j are the mean and covariance matrix, respectively. Thus the mean $\underline{\mu}_j$ specifies the *location* of the jth component density in feature space and the covariance matrix Σ_j prescribes how the data belonging to component j are typically dispersed or scattered around $\underline{\mu}_j$. The flexibility of this model has led to its widespread application, particularly in applied statistics (McLachlan and Basford 1988), and more recently in machine learning and knowledge discovery (Cheeseman and Stutz 1996).

Estimating the Clusters from Data

Clustering (in this mixture model context) is as follows:

1. Assume that the data are generated by a mixture model, where each component is interpreted as a cluster or class ω_j and it assumed that each data point must have been generated by one and only one of the classes ω_j.
2. Given a data set where it is not known which data points came from which components, infer the characteristics (the parameters) of the underlying density functions (the clusters).

In particular, given an "unlabelled" data set $D = \{\underline{x}_1, \ldots, \underline{x}_N\}$, and assuming that the number of clusters k and the functional forms of the component densities g_j in Equation 1 are fixed, estimate the model parameters $\hat{\Phi}_k$. Given $\hat{\Phi}_k$, one can then calculate the probability that data point $\underline{x}$ belongs to class ω_j (by Bayes' rule):

$$\hat{p}(\omega_j|\underline{x}) = \frac{g_j(\underline{x}|\hat{\underline{\theta}}_j)\hat{\alpha_j}}{\sum_{l=1}^{k} g_l(\underline{x}|\hat{\underline{\theta}}_l)\hat{\alpha_l}} \quad 1 \leq j \leq k, \qquad (2)$$

where $\hat{\theta}$ denotes an estimate of the true parameter θ. Here, $\hat{\alpha_j} = \hat{p}(\omega_j)$, i.e., an estimate of the marginal or prior for each cluster. Since the mixture likelihood (or posterior) surface (as a function of the parameters) can have many local maxima, and no closed form solution for the global maximum exists, parameter estimation for mixtures is non-trivial. Much of the popularity of mixture models in recent years is due to the existence of efficient iterative estimation techniques (in particular, the expectation-maximization (EM) algorithm).

Choosing the Number of Clusters k

Above we have assumed that k, the number of clusters, is known *a priori*. While there may be situations where k is known, one would often like to determine k from the data if possible. Prior work on automatically finding k can roughly be divided into three categories.

The classical approach is based on hypothesis testing, where hypothesis k states that the underlying density is a mixture of k components. As discussed in Titterington, Smith and Makov (1985, Section 5.4), these techniques are largely unsatisfactory due to the "failure of standard regularity conditions" on the mixture likelihood function.

A second approach is the full Bayesian solution where the posterior probability of each value of k is calculated given the data, priors on the mixture parameters, and priors on k itself. A potential difficulty with this approach is the computational complexity of integrating over the parameter space to get the posterior probabilities on k. The AutoClass algorithm (Cheeseman and Stutz 1996) uses various approximations to get around the computational issues. Sampling techniques have also been applied to this problem with some success (cf. Diebolt and Robert 1994).

A third method (related to the Bayesian approach, see Chickering and Heckerman, 1996) is that of penalized likelihood (such as the Bayesian Information Criterion (BIC) and various coding-based (e.g., MDL/MML) criteria). A penalty term is added to the log-likelihood to penalize the number of parameters (e.g., Sclove 1983). A significant problem here is that the general assumptions underlying the asymptotic optimality of the penalized criteria do not hold in the mixture modelling context (Titterington, Smith and Makov, Section 5.4).

In theory, the full Bayesian approach is fully optimal and probably the most useful of the three methods listed above. However, in practice it is cumbersome to implement, it is not necessarily straightforward to extend to non-Gaussian problems with dependent samples, and the results will be dependent in a non-transparent manner on the quality of the underlying approximations or simulations. Thus, there is certainly room for exploring alternative methods.

Cross-Validated Likelihood for Choosing k

Let $f(\underline{x})$ be the "true" probability density function for $\underline{x}$. Let $D = \{\underline{x}_1, \ldots, \underline{x}_N\}$ be a random sample from f. Consider that we fit a set of finite mixture models with k components to D, where k ranges from 1 to $k_{\max}$. Thus, we have an indexed set of estimated models, $f_k(\underline{x}|\hat{\Phi}_k), 1 \leq k \leq k_{\max}$, where each $f_k(\underline{x}|\hat{\Phi}_k)$ has been fitted to data set D.

The data log-likelihood for the kth model is defined as

$$L_k(D) = \log\left(\prod_{i=1}^{N} f_k(\underline{x}_i|\hat{\Phi}_k)\right) = \sum_{i=1}^{N} \log f_k(\underline{x}_i|\hat{\Phi}_k). \qquad (3)$$

Assume that the parameters for the kth mixture model were estimated by maximizing this likelihood as a function of Φ_k, keeping the data D fixed (standard maximum likelihood estimation). We then get that $L_k(D)$ is

a non-decreasing function of k since the increased flexibility of more mixture components allows better fit to the data (increased likelihood). Thus, $L_k(D)$ can not directly provide any clue as to the *true* mixture structure in the data, if such structure exists[1].

Imagine that we had a large test data set D^{test} which is not used in fitting any of the models. Let $L_k(D^{\text{test}})$ be the log-likelihood as defined in Equation 3, where the models are fit to the training data D but the likelihood is evaluated on D^{test}. We can view this likelihood as a function of the "parameter" k, keeping all other parameters and D fixed. Intuitively, this "test likelihood" should be a more useful estimator (than the training data likelihood) for comparing mixture models with different numbers of components.

However, in practice, we can not afford, or do not have available, a large independent test set such as D^{test}. Let $\hat{L}_k^{cv}$ be a cross-validation estimate of $L_k(D^{\text{test}})$—we discuss in the next section the particulars of how $\hat{L}_k^{cv}$ is calculated. What can $\hat{L}_k^{cv}$ tell us about how close the model $f_k(\underline{x}|\hat{\Phi}_k)$ is to the true data-generating density f? Following Silverman (1986, p.53) and Chow, Geman, and Wu (1983) , it can be shown under appropriate assumptions that

$$E\left[\hat{L}_k^{cv}\right] = \int f(\underline{x}) \log \frac{f(\underline{x})}{f_k(\underline{x}|\hat{\Phi}_k)} d\underline{x} + C \qquad (4)$$

where C is a constant independent of k and $f_k(\underline{x}|\hat{\Phi}_k)$, and the expectation E is taken with respect to the true density $f(\underline{x})$.

The term in square brackets is the Kullback-Leibler (K-L) information distance between $f(\underline{x})$ and $f_k(\underline{x}|\hat{\Phi}_k)$, namely $I(f, f_k(\hat{\Phi}_k))$. $I(f, f_k(\hat{\Phi}_k))$ is strictly positive unless $f = f_k(\hat{\Phi}_k)$. Thus, the k which minimizes $I(f, f_k(\hat{\Phi}_k))$ tells us which of the mixture models is closest to the true density f. From Equation 4, $\hat{L}_k^{cv}$ is an approximately unbiased estimator (within a constant) of the expected value of the K-L distance $-I(f, f_k(\hat{\Phi}_k))$. Given that f (and $I(f, f_k(\hat{\Phi}_k))$) is unknown, maximizing $\hat{L}_k^{cv}$ over k is a reasonable estimation strategy and is the approach adopted in this paper.

A Monte Carlo Cross-Validated Clustering Algorithm

Choosing a Particular Cross-Validation Method

There are several possible cross-validation methods one could use to generate $\hat{L}_k^{cv}$. v-fold cross validation (vCV) consists of partitioning the data into v disjoint subsets. $v = 1$ yields the well-known "leave-one-out" cross validated estimator, but this is well-known to suffer from high variance. $v = 10$ has been a popular choice in practice (e.g., the CART algorithm for decision tree classification). In *Monte Carlo cross validation* (MCCV) the data are partitioned M times into disjoint train and test subsets where the test subset is a fraction β of the overall data (Burman 1989, Shao 1993). The key distinction between MCCV and vCV is that in MCCV the different test subsets are chosen randomly and need not be disjoint. Typically β can be quite large, e.g., 0.5 or larger, and hundreds or thousands of runs (M) can be averaged. In the regression context it was shown by Shao (1993) that keeping β relatively large reduces estimation variability in the test data (compared to vCV methods). Intuitively, the MCCV estimates should be unbiased (being an average of M individually unbiased estimates) and have desirable variance properties: however, there are few theoretical results available on MCCV in general and none on MCCV in a likelihood estimation context.

Specification of the MCCV Algorithm

The algorithm operates as follows. The outer loop consists of M cross-validation runs over M randomly-chosen train/test partitions. For each partition, k is varied from 1 to $k_{\max}$ and the EM algorithm is used to fit the k components to the training data.

The EM algorithm is initialized using a variant of the k-means algorithm, which is itself initialized randomly. To avoid local minima, the k-means algorithm is run t times (default value is $t = 10$) from different starting points and the highest likelihood solution used to begin the EM estimation. The EM estimation is constrained away from singular solutions in parameter space by limiting the diagonal elements of the component covariance matrices Σ_j to be greater than ϵ (default value is $\epsilon = 0.001\sigma$ where σ is the standard deviation of the unclustered data in the relevant dimension). The EM algorithm iterates until the change in likelihood is less than δ (default value is $\delta = 10^{-6}$), or up to a prespecified maximum number of iterations (default is 30), whichever occurs first. Keeping the maximum number of EM iterations small allows for quicker execution of the algorithm: the intuition is that since we are averaging multiple cross-validation runs, it is sufficient that the EM estimates be somewhere near a peak in the likelihood surface—this assumption warrants further investigation.

Each of the fitted models with k components are then applied to the unseen data in the test partition, and the test-data log-likelihood (Equation 3) is calculated for each. As indicated earlier, this is repeated M times, and the M cross-validated estimates are averaged for each k to arrive at $\hat{L}_k^{cv}, 1 \leq k \leq k_{\max}$. Similarly the standard deviation over the M runs can be calculated for each k, indicating the variability of the likelihood estimates.

The data analyst can plot the $\hat{L}_k^{cv}$ as a function of k along with the standard deviations to see what the data

[1] Traditionally this is the departure point for penalized likelihood and likelihood ratio testing methods.

says about the number of clusters. Another approach is to roughly calculate the posterior probabilities for each k, where one effectively assumes equal priors on the values of k:

$$p(k|D) \approx \frac{\exp(\hat{L}_k^{cv})}{\sum_{l=1}^{k_{\max}} \exp(\hat{L}_l^{cv})}, \qquad 1 \leq k \leq k_{\max}.$$

The distribution of $p(k|D)$ is essential to interpreting the results in the following sense. If one of the $p(k|D)$'s is near 1, then there is strong evidence for that particular number of clusters. If the $p(k|D)$'s are more spread out, then the data are not able to resolve the cluster structure, although "bunching" about a particular k value may allow one to focus on a sub-range of k. It is not recommended that the procedure be implemented as a "black-box" where simply the maximum k value is reported.

The complexity of the EM algorithm for fixed k is $O(kd^2NE)$ where d is the dimensionality of the data, and E denotes the average number of iterations of the EM algorithm. Thus, the overall computational complexity of the MCCV clustering algorithm is $O(Mk_{\max}^2 d^2 NE)$, i.e., *linear* in the number of samples N if one assumes that E does not depend on N.

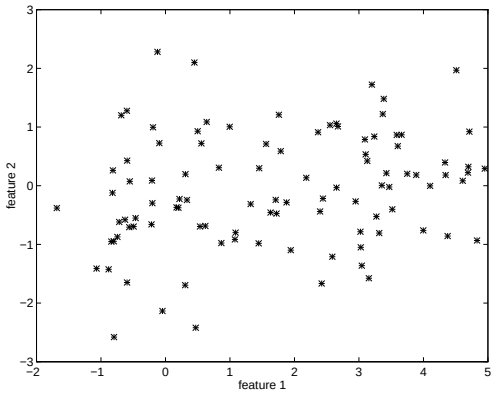
Experimental Results

Overall Experimental Methodology

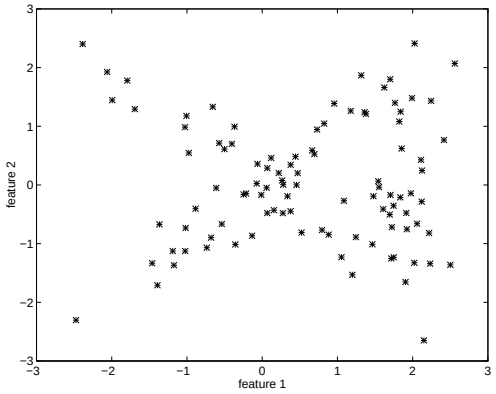
The MCCV algorithm was evaluated on both simulated and real data sets. Unless stated otherwise, the algorithm was run with $M = 20$ (the number of runs) and $\beta = 0.5$ (the fraction of data left out in each run). The value of $M = 20$ was chosen for pragmatic reasons to reduce simulation time (the MCCV procedure is currently coded in MATLAB which is not particularly efficient). The value of $\beta = 0.5$ was based on some initial experimentation which suggested that keeping the cross-validation train and test partitions roughly the same size gave better results than the more "traditional" 90/10 type partitions. Other details of these experiments are omitted here due to lack of space.

Three other methods were compared to MCCV: AutoClass v2.0 (from the authors at NASA Ames), vCV (with $v = 10$), and BIC (using the standard $(q_k/2)\log N$ penalty term where q_k is the number of parameters in the mixture model with k components). The v-CV and BIC methods used the same version of the EM algorithm as MCCV. The maximum number of classes for each of the algorithms ($k_{\max}$) was set to 8 or 15, depending on the true number of classes in the data.

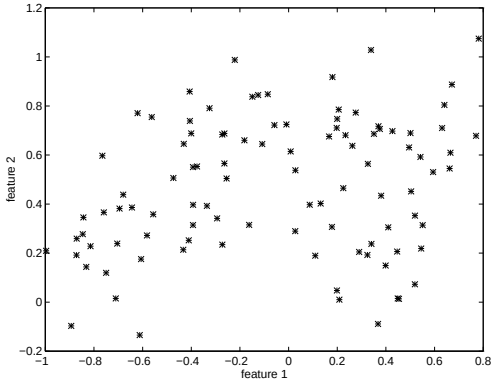
It is important to note that all of the algorithms have random components. The initialization of the EM algorithm (used by each of the clustering algorithms) for fixed k is based on randomly choosing k initial cluster means. The cross-validation algorithms contain further randomness in their choice of particular partitions of the data. Finally, the simulated data sets can



(a) 2-class data.



(b) 3-class data.



(c) 4-class data.

Figure 1: 2-d scatterplots of simulated data.

be regenerated randomly according to the probability model. An ideal set of experiments would average over all of these sources of randomness. Thus, although the experiments in principle could be more extensive (for example, by averaging results over multiple simulated realizations of the simulated problems for a given N) they nonetheless provide clear insight into the general behavior of the algorithms.

Experiments were run on relatively small-sample, low-dimensional data sets (Figures 1 and 2). From a data mining viewpoint this may seem uninteresting. In fact, the opposite is true. *Small sample sizes* are the most challenging problems for methods which seek to extract structure from data (since the MCCV algorithm scales roughly linearly with sample size, scaling up to massive data sets does not pose any particular problems). The focus on *low-dimensional* problems was driven by a desire to evaluate the method on problems where the structure can easily be visualized, the availability of well-known data sets, and the use (in this paper) of full-covariance Gaussian models (which scale poorly with dimensionality from an estimation viewpoint). For all data sets the classes are roughly equiprobable.

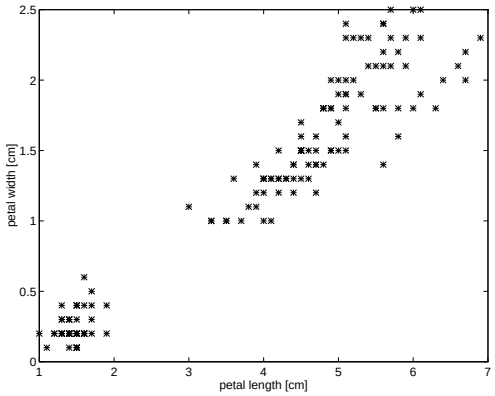
Finally due to space limitations we only summarize the experimental results obtained, namely, provide the value of k which maximizes the relevant criterion for each algorithm. Note that, as discussed in the previous section, this is *not* the recommended way to use the MCCV algorithm: rather the full posterior probabilities and variances for each k should be reported and interpreted by the user.

Details of Experiments on Simulated Data

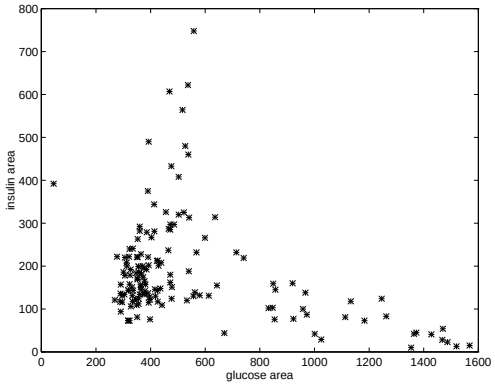
Table 1 contains a brief summary of the experimental results on simulated data.

Experiment 1 consisted of a "control" experiment: data from a single 2-dimensional Gaussian (with $\Sigma = I$ (the identity matrix)). BIC, AutoClass, and MCCV correctly determined the presence of only a single class. vCV on the other hand exhibited considerable variability and incorrectly detected multiple clusters. In general, across different experiments, the vCV method (with v=10) was found to be an unreliable estimator compared to the other methods.

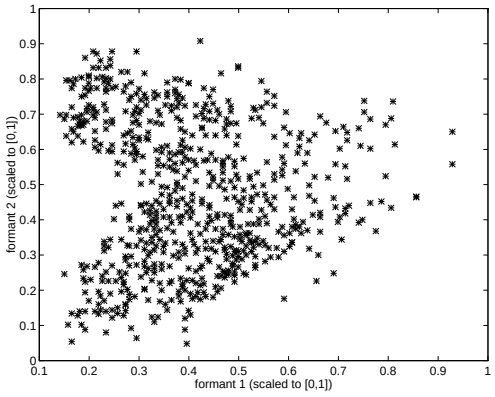
The second simulation problem consists of 2 Gaussian clusters in 2 dimensions, both with covariance matrices $\Sigma_1 = \Sigma_2 = I$ and means $\mu_1 = (0,0), \mu_2 = (0,3)$. There is considerable overlap of the clusters (Figure 1(a)) making this a non-trivial problem. MCCV finds evidence of only one cluster with $N = 100$, but for $N = 600, 1200$ it correctly finds both clusters. This conservative tendency of the MCCV algorithm (whereby it finds evidence to support fewer clusters given less data) is pleasing and was noted to occur across different data sets. BIC and AutoClass detected the same number of clusters as MCCV: vCV was consistently incorrect on this problem.



(a) 2d iris data.



(b) 2d diabetes data.



(c) Vowel data.

Figure 2: 2-d scatterplots of real data.

Table 1: Experimental results on simulated data.

Problem	Sample Size	BIC	AC	vCV	MCCV	Truth
1-Class	50	1	1	2	1	1
	200	1	1	1	1	1
	800	1	1	5	1	1
2-Class	100	1	1	4	1	2
	600	2	2	3	2	2
	1200	2	2	3	2	2
3-Class	100	3	3	4	3	3
	600	3	3	3	3	3
	1200	3	3	4	3	3
4-Class	100	2	3	5	3	4
	500	3	4	5	4	4
	1000	4	4	6	4	4

The 3-Class problem (Figure 1(b)) follows the simulated Gaussian structure (in two dimensions) used in Banfield and Raftery (1993). Two of the clusters are centred at $(0, 0)$ but are oriented in "orthogonal" directions. The third cluster overlaps the "tails" of the other two and is centred to the right. MCCV, BIC and AutoClass each correctly detected 3 clusters: once again, vCV was not reliable.

The final simulation problem (Figure 1(c)) contains 4 classes and is taken from Ripley (1994) where it was used in a supervised learning context (here, the original class labels were removed). vCV is again unreliable. Each of BIC, MCCV and AutoClass "zero-in" on the correct structure given enough data, with BIC appearing to require more data to find the correct structure.

Real Data Sets

Below we discuss the application of MCCV (and the comparison methods) to several real data sets. Table 2 contains a brief summary of the results. Each of these data sets has a known classification: the clustering algorithms were run on the data with the class label information removed. Note that unlike the simulated examples, the number of classes in the original classified data set is not necessarily the "correct" answer for the clustering procedure, e.g., it is possible that the a particular class in the original data is best described by two or more "sub-clusters," or that the clusters are in fact non-Gaussian.

Iris Data The classic "iris" data set contains 150 4-dimensional measurements of plants classified into 3 species. It is well known that 2 of the species are overlapped in the feature-space and the other is well-separated from these 2 (Figure 2(a)). MCCV indicates 2 clusters with some evidence of 3. AutoClass found 4 clusters and BIC found 2 (the fact that vCV found 3 clusters is probably a fluke). It is quite possible that the clusters are in fact non-Gaussian and, thus, do not match the clustering model (e.g., the measurements have limited precision, somewhat invalidating the Gaussian assumption). Given these caveats, and the relatively small sample size, each of the methods are both providing useful insight into the data.

Diabetes Data Reaven and Miller (1979) analyzed 3-dimensional plasma measurement data for 145 subjects who were clinically diagnosed into three groups: normal, chemically diabetic, or overtly diabetic. This data set has since been analyzed in the statistical clustering literature by Symons (1981) and Banfield and Raftery (1993). When viewed in any of the 2-dimensional projections along the measurement axes, the data are not separated into obvious groupings: however, some structure is discernible (Figure 2(b)). The MCCV algorithm detected 3 clusters in the data (as did vCV and AutoClass). BIC incorrectly detected 4 classes. It is encouraging that the MCCV algorithm found the same number of classes as that of the original clinical classification. We note that the "approximate weight of evidence" clustering criterion of Banfield and Raftery (1993) (based on a Bayesian approximation) was maximized at $k = 4$ clusters and indicated evidence of between 3 to 6 clusters.

Speech Vowel Data Peterson and Barney (1952) measured the location of the first and second prominent peaks (formants) in 671 estimated spectra from subjects speaking various vowel sounds. They classified the spectra into 10 different vowel sounds based on acoustic considerations. This data set has since been used in the neural network literature with the best cross-validated classification accuracies being around 80%. As can be seen in Figure 2(c) the classes are heavily overlapped. Thus, this is a relatively difficult problem for methods which automatically try to find the correct number of clusters. AutoClass detected 5 clusters, while MCCV detected 7, with some evidence between 6 and 9 clusters. BIC completely underestimated the number of clusters at 2. Given the relatively

Table 2: Experimental results on non-simulated data.

Problem	Sample Size	BIC	AC	vCV	MCCV	"Truth"
Iris	150	2	4	3	2	3
Diabetes	145	4	3	3	3	3
Vowels	671	2	5	8	7	10

small sample size (one is fitting each cluster using only roughly 30 sample points), the MCCV algorithm is doing well to indicate the possibility of a large number of clusters.

Discussion

The MCCV method performed roughly as well as AutoClass on the particular data sets used in this paper. The BIC method performed quite well, but overall was not as reliable as MCCV or AutoClass. vCV (with v=10) was found to be largely unreliable. Further experimentation on more complex problems may reveal some systematic differences between the Bayesian (AutoClass) and cross-validation (MCCV) approaches, but both produced clear insights into the structure of the data sets used in the experiments in this paper.

We note in passing that Chickering and Heckerman (1996) have investigated a problem which is equivalent to that addressed in this paper, except that the feature vector $\underline{x}$ is now discrete-valued and and the individual features are assumed independent of the (hidden) class variable. Chickering and Heckerman empirically evaluated the performance of AutoClass, BIC, and other approximations to the full Bayesian solution on a wide range of simulated problems. Their results include conclusions which are qualitatively similar to those reported here: namely that the AutoClass approximation to the full Bayesian solution outperforms BIC in general and that BIC tends to be overly conservative in terms of the number of components it prefers.

The theoretical basis of the MCCV algorithm warrants further investigation. For example, MCCV is somewhat similar to the bootstrap method. A useful result would be a characterization (under appropriate assumptions) of the basic bias-variance properties of the MCCV likelihood estimate (in the mixture context) as a function of the leave-out fraction β, the number of cross-validation runs M, the sample size N, and some measure of the problem complexity. Prescriptive results for choosing β automatically (as developed in Shao (1993) in a particular regression context) would be useful. For example, it would be useful to justify in theory the apparent practical utility of $\beta = 0.5$.

On the practical front, clearly there is room for improvement over the basic algorithm described in this paper. The probabilistic cluster models can easily be extended beyond the full-covariance model to incorporate, for example, the geometric shape and Poisson "outlier" models of Banfield and Raftery (1993) and Celeux and Govaert (1995), and the discrete variable models in AutoClass. Diagnostic tests for detecting non-Gaussianity could also easily be included (cf. McLachlan and Basford (1988), Section 2.5) and would be a useful practical safeguard.

Some obvious improvements could also be made to the search strategy. Instead of "blindly" searching over each possible value of k from 1 to $k_{\max}$ a more intelligent search could be carried out which "zeros-in" on the range of k which has appreciable posterior probability mass (the current implementation of the AutoClass algorithm includes such a search algorithm).

Data-driven methods for automatically selecting the leave-out fraction method β are also a possibility, or possibly averaging results over multiple values of β. When the data are few relative to the number of clusters present, it is possible for the cross-validation partitioning to produce partitions where no data from a particular cluster is present in the training partition. This will bias the estimate towards lower k values (since the more fractured the data becomes, the more likely this "pathology" is to occur). A possible solution is some form of data-driven *stratified* cross-validation (Kohavi (1995) contains a supervised learning implementation of this idea).

Both the EM and MCCV techniques are amenable to very efficient parallel implementations. For large sample sizes N, it is straightforward to assign $1/p$ of the data to p processors working in parallel which communicate via a central processor. For large numbers of cross-validation runs M, each of p processors can independently run M/p runs.

Finally we note that our attention was initially drawn to this problem in a time-series clustering context using hidden Markov models. In this context, the general MCCV methodology still applies but because of sample dependence the cross-validation partitioning strategy must be modified—this is currently under development. The MCCV approach to mixtures described here can also be applied to supervised learning with mixtures: the MCCV procedure provides an automatic method for determining how many components to use to model each class. Other extensions to learning of graphical models (Bayesian networks) and image segmentation are also possible.

Conclusions

MCCV clustering appears to be a useful idea for determining cluster structure in a probabilistic clustering context. Experimental results indicate that

the method has significant practical potential. The method complements Bayesian solutions by being simpler to implement and conceptually easier to extend to more complex clustering models than Gaussian mixtures.

Acknowledgments

The author would like to thank Alex Gray for assistance in using the AutoClass software. The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

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

Burman, P. 1989 'A comparative study of ordinary cross-validation, v-fold cross-validation, and the repeated learning-testing methods,' *Biometrika*, 76(3), 503–514.

Celeux, G., and Govaert, G. 1995. 'Gaussian parsimonious clustering models,' *Pattern Recognition*, 28(5), 781–793.

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, R. Uthurusamy (eds.), Cambridge, MA: AAAI/MIT Press, pp. 153–180.

Chickering, D. M., and Heckerman, D. 1996. 'Efficient approximations for the marginal likelihood of incomplete data given a Bayesian network,' MSR-TR-96-08 Technical Report, Microsoft Research, Redmond, WA.

Chow, Y. S., Geman, S. and Wu, L. D. 1983. 'Consistent cross-validated density estimation,' *The Annals of Statistics*, 11(1), 25–38.

Diebolt, J. and Robert, C. P. 1994. 'Bayesian estimation of finite mixture distributions,' *J. R. Stat. Soc. B*, 56, 363–375.

Gelman, A., J. B. Carlin, H. S. Stern, D. B. Rubin. 1995. *Bayesian Data Analysis*, London, UK: Chapman and Hall.

Jain, A. K., and R. C. Dubes. 1988. *Algorithms for Clustering Data*, Englewood Cliffs, NJ: Prentice Hall.

Kohavi, R. 1995. 'A study of cross-validation and bootstrap for accuracy estimation and model selection,' *Proc. Int. Joint. Conf. AI*, Montreal.

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

Peterson, G. and Barney, H. 1952. 'Control methods used in the study of vowels,' . *J. Acoust. Soc. Am.*, 24, 175–184.

Reaven, G. M., and Miller, R. G. 1979. 'An attempt to define the nature of chemical diabetes using a multi-dimensional analysis,' *Diabetologia*, 16, 17–24.

Ripley, B. D. 1994. 'Neural networks and related methods for classification (with discussion),' *J. Roy. Stat. Soc. B*, 56, 409–456.

Sclove, S. C. 1983. 'Application of the conditional population mixture model to image segmentation,' *IEEE Trans. Patt. Anal. Mach. Intell.*, PAMI-5, 428–433.

Shao, J. 1993. 'Linear model selection by cross-validation,' *J. Am. Stat. Assoc.*, 88(422), 486–494.

Silverman, B. W. 1986. *Density Estimation for Statistics and Data Analysis*, Chapman and Hall.

Symons, M. 1981. 'Clustering criteria and multivariate normal mixtures,' *Biometrics*, 37, 35–43.

Titterington, D. M., A. F. M. Smith, U. E. Makov. 1985. *Statistical Analysis of Finite Mixture Distributions*, Chichester, UK: John Wiley and Sons.