RUTCOR
RESEARCH
REPORT

Clustering Data with Measurement Errors

Mahesh Kumar [a] Nitin R. Patel [b]

RRR 12-2005, February, 2005

RUTCOR
Rutgers Center for
Operations Research
Rutgers University
640 Bartholomew Road
Piscataway, New Jersey
08854-8003
Telephone: 732-445-3804
Telefax: 732-445-5472
Email: rrr@rutcor.rutgers.edu
http://rutcor.rutgers.edu/∼rrr

[a]Rutgers Business School & RUTCOR, Rutgers University, 180 University Avenue, Newark, NJ 07102, maheshk@rutgers.edu

[b]Sloan School of Management, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Bldg. E40-111, Cambridge , MA 02139, nitinrp@mit.edu

Rutcor Research Report
RRR 12-2005, February, 2005

Clustering Data with Measurement Errors

Mahesh Kumar Nitin R. Patel

Abstract. Traditional clustering methods assume that there is no measurement error, or uncertainty, associated with data. Often, however, real world applications require treatment of data that have such errors. In the presence of measurement errors, well-known clustering methods like k-means and hierarchical clustering may not produce satisfactory results. The fundamental question addressed in this paper is: "What is an appropriate clustering method in the presence of errors associated with data?"

In the first part of this paper, we develop a statistical model and algorithms for clustering data in the presence of errors. We assume that the errors associated with data follow a multivariate Gaussian distribution and are independent between data points. The model uses the maximum likelihood principle and provides us with a new metric for clustering. This metric is used to develop two algorithms for error-based clustering, hError and kError, that are generalizations of Ward's hierarchical and k-means clustering algorithms, respectively.

In the second part of the paper, we discuss sets of clustering problems where error information associated with the data to be clustered is readily available and where error-based clustering is likely to be superior to clustering methods that ignore error. We give examples of the effectiveness of error-based clustering on data generated from the following statistical models: (1) sample averaging, (2) multiple linear regression, (3) ARIMA time series, and (4) Markov chain models. We present theoretical and empirical justifications for the value of error based clustering on these classes of problems.

1 Introduction

Clustering is a fundamental technique that divides data into groups (clusters) for the purpose of summarization or improved understanding. Clustering has been widely studied for over four decades across multiple disciplines including data mining, statistics, machine learning, and operations research, and across multiple application areas including taxonomy, medicine, astronomy, marketing, finance and e-commerce. In the last decade, clustering has become increasingly important due to the large amounts of data that are now being collected and stored electronically.

The problem of clustering is defined as follows: given n data points, $x_1, \ldots, x_n$, in a p-dimensional metric space, partition the data into $G \leq n$ clusters, $C_1, \ldots, C_G$, such that data points within a cluster are more similar to each other than data points in different clusters. Most clustering methods form clusters based on proximity between the data points in the x space. A commonly used measure of proximity between a pair of data points, $\{x_i, x_j\}$, is the *squared Euclidean distance* between the points as defined below.

$$d(x_i, x_j) = ||x_i - x_j||^2. \tag{1}$$

A popular criterion for clustering is to minimize the sum of the squared Euclidean distances given by

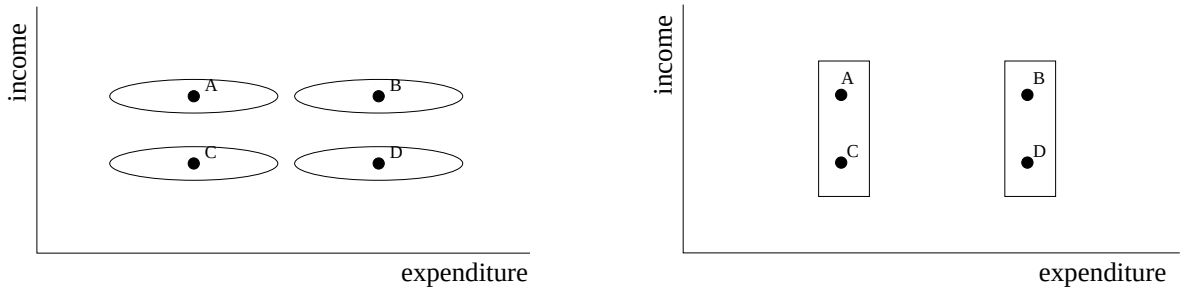
$$\min_{C_1,\ldots,C_G} \sum_{k=1}^{G} \sum_{x_i \in C_k} ||x_i - c_k||^2, \tag{2}$$

where c_k is the mean of the data points in cluster C_k. The well-known clustering methods, k-means and Ward's hierarchical clustering methods [15], for example, optimize this criterion.

A drawback of these and other traditional clustering methods is that they ignore measurement errors, or uncertainty, associated with the data. In certain applications, measurement errors associated with data are available and can play a significant role in making clustering decision. For example, for clustering of sample means in analysis of variance (ANOVA) [5], clustering of time series data [22], and clustering of online users based on Markov models [7] (we elaborate on these applications in Section 6), data to be clustered are not directly observable and a statistical method (sample averaging, time series analysis, or Markov modelling, respectively, for these examples) generates the data. A clustering method is then applied to the generated data to obtain the desired clusters. Most statistical methods, including the ones mentioned above, provide estimates of the errors associated with the generated data. Incorporating these errors in the clustering process can produce different and, often, better clustering results, as illustrated by the following example.

We wish to cluster four geographical regions into two clusters based on household income and expenditure. Suppose the data for each geographical region is estimated as the average of incomes and expenditures of households in the region. Points in Figure 1(a) represent the measurements of average income and expenditure for these four regions. The ellipses in this figure represent the standard errors associated with these measurements. Figure 1(b) shows clustering of these four regions into two clusters when k-means clustering method is applied on these four data points. The rectangles in this figure represent cluster membership.

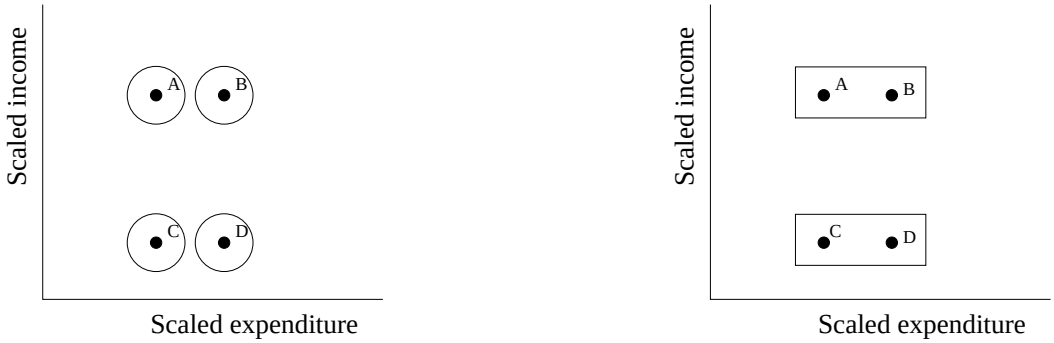
We notice that the standard errors associated with the expenditure measurements are much higher than the standard errors associated with the income measurements. Therefore, we may wish to give more weight to differences in income measurements than to differences in expenditure measurements. One way to do this is to scale each measurement to standard error units. The scaled data is shown in Figure 2(a). Different clusters are obtained when k-means clustering is applied to the transformed data as shown in Figure 2(b).



(a) Average income and expenditure of four regions along with standard deviation measurements.

(b) k-means clustering result based on average income and expenditure measurements.

Figure 1: Clustering of points without considering error information.



(a) Income and expenditure measurements scaled by standard errors

(b) k-means clustering result on scaled data.

Figure 2: Clustering of points by considering error information.

In general, the structure of errors could be more complex than in this example. In this paper, we develop theory and algorithms and present a range of illustrative applications for a new approach to clustering that we call *error-based clustering*. Error-based clustering explicitly incorporates errors associated with data into clustering algorithm.

The main contributions of this paper can be divided into two parts. In the first part, we develop a general model and algorithms for incorporating error information into cluster analysis. We assume that the errors associated with data follow multivariate Gaussian

distributions [1] and are independent between data points. Using this probability model for the data, we model the clustering problem as a likelihood maximization problem. The maximum likelihood procedure provides us with a new objective criterion for clustering that incorporates information about the error associated with data. The new objective criterion is then used to develop two algorithms for error-based clustering: (1) *hError*, a hierarchical clustering algorithm that produces a sequence of clusters at various levels in which clusters at any level are nested within the clusters at higher levels in the sequence and (2) *kError*, a partitioning algorithm that partitions the data into a specified number of clusters. These algorithms are generalizations of the popular hierarchical clustering algorithm of Ward [17] and the k-means clustering algorithm [15], respectively. We provide a heuristic method for selecting the number of clusters in a data set, which in itself is a challenging problem [23].

In the second part, we describe settings where error-based clustering is likely to outperform standard clustering methods. We focus on clustering of data generated from statistical models. In particular, we look at four statistical models: (1) sample averaging (2) multiple linear regression, (3) ARIMA models for time-series, and (4) Markov chain models. We show that, under certain assumptions, error-based clustering is approximately an optimal clustering method for clustering of data generated from such statistical models. We also conduct a series of empirical studies on real and simulated datasets where we have found that error-based clustering performs significantly better than traditional clustering methods on this application.

2 Related Work

Probability models have been used for quite some time as a basis for cluster analysis [3, 6, 7, 11, 27]. In these models, data are viewed as samples from mixtures of probability distributions, where each component in the mixture represents a cluster. The goal is to partition the data into clusters such that data points that come from the same probability distribution belong to the same cluster. The authors of [3] and [7] have shown effectiveness of such probability models in a number of practical applications including clustering of medical data, gene expression data, web-logs data, and image data. While these authors provide a general probability model that allows any probability distribution for data, they have found that a mixture of Gaussian distributions is applicable to many problems in practice.

The probability model used in error-based clustering is similar to the one used in model-based clustering [3, 11]. In model-based clustering, data points are modelled as arising from a mixture of multivariate Gaussian populations, where each population represents a cluster. The parameters of the mixture components are estimated by maximizing the likelihood of the observed data. A key advantage in model-based clustering is that it allows clusters of different shapes and sizes, which is not possible in the traditional clustering methods such as k-means and Ward's methods. We differ from standard model-based clustering in the sense that instead of modelling the *populations* as multivariate Gaussian, we model the *error*

[1] The Gaussian assumption is applicable to many problems in practice and is often (at least approximately) valid for estimates generated from a statistical model that uses the maximum likelihood principle.

associated with each data point as multivariate Gaussian. In other words, in the special case when it is assumed that all data points in the same cluster have the same error distribution, error-based clustering is equivalent to model-based clustering. By allowing different amounts of error for each data point, error-based clustering explicitly models error information for incorporation into clustering algorithms.

In Sections 6 and 7, we present an application of error-based clustering for clustering of data generated from statistical models. The clustering problem presented in this application can be modelled as a special case of latent class models for clustering [7, 13], which is a very general framework encompassing most probability models for clustering. The algorithm that is commonly used for latent class models is the EM algorithm [7]. In this paper we show that while the EM algorithm deals with a large volume of data and can be computationally challenging, error-based clustering produces approximately the same clusters as EM algorithm, while working on a much smaller data set.

So far, we have come across only one publication [4] that explicitly considers error information in multivariate clustering. It provides a heuristic solution for the case of uniformly distributed spherical errors associated with data. We consider the case when errors follow multivariate Gaussian distributions and provide a formal statistical procedure to model them. Modelling errors as being Gaussian distributed has a long history (beginning with Gauss!) and is applicable to many problems in practice [28].

While there is almost no prior published work on incorporating error information in multivariate cluster analysis, there has been significant work done on this topic for one-dimensional data [5, 8, 9]. The authors of [5] applied their technique to clustering of the slope coefficients from a group of bivariate regressions and used it for predicting rainfall. We extend their work to clustering of multivariate regression coefficients in Section 7.2.

3 Error-based Clustering Model

The data to be clustered consists of n observations $x_1, \ldots, x_n$ (column vectors) in $\Re^p$ and n matrices $\Sigma_1, \ldots, \Sigma_n$ in $\Re^{p \times p}$, where x_i represents measurements on p characteristics and Σ_i represents the variance-covariance matrix associated with the observed measurements of x_i. Suppose that the data points are independent and that each arises from a p-variate Gaussian distribution with one of G possible means $\theta_1, \ldots, \theta_G$, $G \leq n$, that is, $x_i \sim N_p(\mu_i, \Sigma_i)$, where $\mu_i \in \{\theta_1, \ldots, \theta_G\}$ for $i = 1, \ldots, n$. Then the joint distribution of the observations is completely determined by the clusters $C_1, ..., C_G$ such that observations that have the same mean (μ_i) belong to the same cluster with $\mu_i = \theta_k$, the common value of μ_i for the observations in C_k, $k = 1, \ldots, G$. The goal is to find the clusters C_k and the values of θ_k, for $k = 1, \ldots, G$. We note that, while most traditional probability models for clustering [27] assume that both μ_i and Σ_i are unknown, here we assume that Σ_i is known and μ_i is unknown.

Let $S_k = \{i | x_i \in C_k\}$, $k = 1, \ldots, G$. Note that $S_1, \ldots, S_G$ consist of disjoint and non-empty subsets of $\{1, ..., n\} = \cup_{k=1}^{G} S_k$. Thus $\mu_i = \theta_k$ for $\forall i \in S_k$, $k = 1, \ldots, G$. Given data points $x_1, \ldots, x_n$ and the error matrices $\Sigma_1, \ldots, \Sigma_n$, the maximum likelihood procedure leads us to choose $S = (S_1, \ldots, S_G)$ and $\theta = (\theta_1, \ldots, \theta_G)$ so as to maximize the likelihood

$$L(x|S,\theta) = \prod_{k=1}^{G} \prod_{i \in S_k} \frac{1}{(2\pi)^{\frac{p}{2}}} |\mathbf{\Sigma}_i|^{-\frac{1}{2}} e^{-\frac{1}{2}(x_i-\theta_k)^t \mathbf{\Sigma}_i^{-1}(x_i-\theta_k)}, \tag{3}$$

where $|\Sigma_i|$ is the determinant of Σ_i for $i = 1, \ldots, n$.

Lemma 3.1. *The maximum likelihood estimate of $S_1, \ldots, S_G$ is a partition that solves*

$$\min_{S_1,\ldots,S_G} \sum_{k=1}^{G} \sum_{i \in S_k} (x_i - \hat{\theta}_k)^t \mathbf{\Sigma}_i^{-1} (x_i - \hat{\theta}_k), \tag{4}$$

where $\hat{\theta}_k$ is the maximum likelihood estimate of θ_k given by

$$\hat{\theta}_k = (\sum_{i \in S_k} \mathbf{\Sigma}_i^{-1})^{-1} (\sum_{i \in S_k} \mathbf{\Sigma}_i^{-1} x_i), \qquad k = 1, \ldots, G. \tag{5}$$

Proof. The log of the likelihood in Equation 3 is given by

$$\begin{aligned} \ell(x|S,\theta) &= \sum_{k=1}^{G} \sum_{i \in S_k} (-\frac{p}{2} \ln(2\pi) - \frac{1}{2} \ln |\Sigma_i| - \frac{1}{2}(x_i - \theta_k)^t \mathbf{\Sigma}_i^{-1} (x_i - \theta_k)) \\ &= -\frac{1}{2} \sum_{i=1}^{n} (p \ln(2\pi) + \ln |\Sigma_i|) - \frac{1}{2} \sum_{k=1}^{G} \sum_{i \in S_k} (x_i - \theta_k)^t \Sigma_i^{-1} (x_i - \theta_k) \\ &= \text{constant} - \frac{1}{2} \sum_{k=1}^{G} \sum_{i \in S_k} (x_i - \theta_k)^t \Sigma_i^{-1} (x_i - \theta_k). \end{aligned} \tag{6}$$

Maximizing the log likelihood is, therefore, equivalent to minimizing

$$\tilde{\ell}(x|S,\theta) = \sum_{k=1}^{G} \sum_{i \in S_k} (x_i - \theta_k)^t \Sigma_i^{-1} (x_i - \theta_k), \tag{7}$$

where minimization is over all possible partitions $S_1, \ldots S_G$ and all possible values of $\theta_1, \ldots, \theta_G$. For a given partitioning, $S_1, \ldots S_G$, the values of $\theta_1, \ldots, \theta_G$ that minimize $\tilde{l}(x|S,\theta)$ can be obtained by setting its patial derivative with respect to all θ_k equal to zero, i.e.,

$$\frac{\partial \tilde{\ell}(x|S,\theta)}{\partial \theta_k} = \sum_{i \in S_k} 2\Sigma_i^{-1}(x_i - \theta_k) = 0, \qquad k = 1, \ldots, G. \tag{8}$$

Solving this equation for each k, we find θ_k that minimizes Equation 7 is given by

$$\hat{\theta}_k = (\sum_{i \in S_k} \mathbf{\Sigma}_i^{-1})^{-1} (\sum_{i \in S_k} \mathbf{\Sigma}_i^{-1} x_i), \qquad k = 1, \ldots, G. \tag{9}$$

Substituting $\hat{\theta}_k$ for θ_k in Equation 7, it follows that the maximum likelihood clustering is the one that solves

$$\min_{S_1,\ldots,S_G} \sum_{k=1}^{G} \sum_{i \in S_k} (x_i - \hat{\theta}_k)^t \mathbf{\Sigma}_i^{-1} (x_i - \hat{\theta}_k). \tag{10}$$

□

This minimization makes intuitive sense because each data point is weighted by the inverse of its error, that is, data points with smaller error get higher weight and vice versa.

Notice that $\hat{\theta}_k$ is a weighted mean of the data points in C_k. We will refer to it as the *Mahalanobis mean* of cluster C_k. Let Ψ_k denote the error matrix associated with $\hat{\theta}_k$, then

$$\begin{aligned}
\Psi_k = Cov(\hat{\theta}_k) &= Cov((\sum_{i \in S_k} \Sigma_i^{-1})^{-1} (\sum_{i \in S_k} \Sigma_i^{-1} x_i)) \\
&= (\sum_{i \in S_k} \Sigma_i^{-1})^{-1} Cov(\sum_{i \in S_k} \Sigma_i^{-1} x_i) (\sum_{i \in S_k} \Sigma_i^{-1})^{-1} \\
&= (\sum_{i \in S_k} \Sigma_i^{-1})^{-1} (\sum_{i \in S_k} \Sigma_i^{-1} Cov(x_i) \Sigma_i^{-1}) (\sum_{i \in S_k} \Sigma_i^{-1})^{-1} \\
&= (\sum_{i \in S_k} \Sigma_i^{-1})^{-1} (\sum_{i \in S_k} \Sigma_i^{-1} \Sigma_i \Sigma_i^{-1}) (\sum_{i \in S_k} \Sigma_i^{-1})^{-1} \\
&= (\sum_{i \in S_k} \mathbf{\Sigma}_i^{-1})^{-1},
\end{aligned} \tag{11}$$

where $Cov(x)$ refers to the $p \times p$ variance-covariance matrix associated with x.

The clustering model described here is a generalization of the standard Euclidean distance based clustering model as described in Equation 2.

Proposition 3.1. *When $\Sigma_i = \sigma^2 * I$ for all $i = 1, \ldots, n$, where I denotes the identity matrix, the criterion of error-based clustering is the same as the minimum squared Euclidean distance criterion that minimizes the sum of the squared Euclidean distances of data points from their cluster centers.*

Proof. Substituting $\sigma^2 * I$ for Σ_i in Equation 5 gives

$$\hat{\theta}_k = \frac{\sum_{i \in S_k} x_i}{|S_k|} = c_k, \qquad k = 1, \ldots, G, \tag{12}$$

where c_k is the (usual) mean of the data points in C_k. Here $|S_k|$ refers to the size of S_k. The criterion of Equation 4 now becomes

$$\min_{S_1,\ldots,S_G} \sum_{k=1}^{G} \sum_{i \in S_k} \frac{1}{\sigma^2} ||x_i - c_k||^2, \tag{13}$$

which is equivalent to

$$\min_{S_1,\ldots,S_G} \sum_{k=1}^{G} \sum_{i \in S_k} ||x_i - c_k||^2. \tag{14}$$

□

Proposition 3.1 amounts to saying that if errors associated with all data point are same and if the errors of variables of a data point are same and uncorrelated, then error-based clustering is equivalent to standard clustering that optimizes the squared Euclidean distance criterion.

Another property of error-based clustering is that it is independent of scale, which is a useful property in many practical applications of clustering [10].

Proposition 3.2. *The objective criterion of error-based clustering is invariant under an affine transformation of the data space.*

Proof. Let $f : x \to Ax + u$ be an affine transformation of the data space, where A is an invertible matrix. Let $x'_i, \hat{\theta}'_k$, and Σ'_i denote the new values of $x_i, \hat{\theta}_k$, and Σ_i, respectively, in the transformed space. Since $x_i \sim N(\mu_i, \Sigma_i)$, we have $x'_i = Ax_i + u \sim N(A\mu_i + u, A\Sigma_i A^t)$. Thus, $\Sigma'_i = A\Sigma_i A^t$. The *Mahalanobis means* of the clusters in the transformed space are given by

$$\begin{aligned}
\hat{\theta}'_k &= (\sum_{i \in S_k} \Sigma_i'^{-1})^{-1} (\sum_{i \in S_k} \Sigma_i'^{-1} x'_i) \\
&= (\sum_{i \in S_k} (A^t)^{-1} \Sigma_i^{-1} A^{-1})^{-1} (\sum_{i \in S_k} (A^t)^{-1} \Sigma_i^{-1} A^{-1} (Ax_i + u)) \\
&= [(A^t)^{-1} (\sum_{i \in S_k} \Sigma_i^{-1}) A^{-1}]^{-1} (A^t)^{-1} (\sum_{i \in S_k} \Sigma_i^{-1} x_i + \sum_{i \in S_k} \Sigma_i^{-1} A^{-1} u) \\
&= A(\sum_{i \in S_k} \Sigma_i^{-1})^{-1} (\sum_{i \in S_k} \Sigma_i^{-1} x_i + \sum_{i \in S_k} \Sigma_i^{-1} A^{-1} u) \\
&= A(\sum_{i \in S_k} \Sigma_i^{-1})^{-1} (\sum_{i \in S_k} \Sigma_i^{-1} x_i) + A(\sum_{i \in S_k} \Sigma_i^{-1})^{-1} (\sum_{i \in S_k} \Sigma_i^{-1}) A^{-1} u \\
&= A\hat{\theta}_k + u, \qquad k = 1, \ldots, G.
\end{aligned} \tag{15}$$

Next we show that each term of the criterion in Equation 4 is invariant under this transformation of the data space.

$$\begin{aligned}
(x'_i - \hat{\theta}'_k)^t \Sigma_i'^{-1} (x'_i - \hat{\theta}'_k) &= (Ax_i + u - A\hat{\theta}_k - u)^t (A^t)^{-1} \Sigma_i^{-1} A^{-1} (Ax_i + u - A\hat{\theta}_k - u) \\
&= (x_i - \hat{\theta}_k)^t \Sigma_i^{-1} (x_i - \hat{\theta}_k).
\end{aligned} \tag{16}$$

Therefore, the criterion of error-based clustering is invariant under an affine transformation of the data space. □

4 The *hError* Clustering Algorithm

4.1 A Hierarchical Greedy Heuristic for *hError*

The formulation in Equation 4 is a nonlinear discrete optimization problem for which we do not know of any polynomial time algorithm [2]. We develop a greedy heuristic to optimize the objective criterion of error-based clustering, which we call as *hError* algorithm. The *hError* algorithm is similar to the agglomerative hierarchical clustering algorithm in [17].

The *hError* algorithm starts with n singleton clusters, each corresponding to a data point. At each stage of the algorithm, we merge a pair of clusters that leads to the minimum increase in the objective function of error-based clustering. Thus, each stage of the algorithm decreases the number of clusters by one and, therefore, corresponds to a unique partition of the data. The merging process can either continue until all data points are merged into a single cluster or stop when the desired number of clusters is obtained.

In the next subsection, we will show that the greedy heuristic at each stage of the *hError* algorithm is equivalent to combining the closest pair of clusters according to a distance function that is easy to compute. Although, the resulting clustering may be suboptimal, hierarchical clustering methods are common in practice because they often yield reasonable results and are easy to compute. Another advantage of hierarchical clustering is that it provides nested clusters for all possible values of $G = 1, \ldots, n$. The user can then pick the best value of G based on her needs. In Subsection 4.3, we will present a heuristic for selecting the number of clusters in a data set.

4.2 The Distance Function in *hError*

Theorem 4.1. *At each step of the hError algorithm, we merge a pair of clusters C_u and C_v for which the distance d_{uv} is minimized, where*

$$d_{uv} = (\hat{\theta}_u - \hat{\theta}_v)^t (\Psi_u + \Psi_v)^{-1} (\hat{\theta}_u - \hat{\theta}_v), \tag{17}$$

$\hat{\theta}_u$ and $\hat{\theta}_v$ are the Mahalanobis means of the clusters C_u and C_v, respectively, and Ψ_u and Ψ_v are the associated error matrices as defined in Equation 11.

Proof. Let E_k denote the contribution of cluster C_k to the objective function, i.e.,

$$E_k = \sum_{i \in S_k} (x_i - \hat{\theta}_k)^t \Sigma_i^{-1} (x_i - \hat{\theta}_k). \tag{18}$$

Then the objective function of error-based clustering can be rewritten as

$$E = \sum_{k=1}^{G} E_k. \tag{19}$$

[2] An optimal clustering can be obtained in polynomial time for one-dimensional data using dynamic programming [9].

Suppose we chose to merge clusters C_u and C_v during an iteration of *hError*, and let the resulting cluster be C_w. Then the net increase in the value of E is given by

$$\triangle E_{uv} = E_w - E_u - E_v. \tag{20}$$

We can rewrite E_k as

$$\begin{aligned} E_k &= \sum_{i \in S_k} x_i^t \mathbf{\Sigma}_i^{-1} x_i + \hat{\theta}_k^t (\sum_{i \in S_k} \mathbf{\Sigma}_i^{-1}) \hat{\theta}_k - 2\hat{\theta}_k^t (\sum_{i \in S_k} \mathbf{\Sigma}_i^{-1} x_i) \\ &= \sum_{i \in S_k} x_i^t \mathbf{\Sigma}_i^{-1} x_i + \hat{\theta}_k^t \Psi_k^{-1} \hat{\theta}_k - 2\hat{\theta}_k^t \Psi_k^{-1} \hat{\theta}_k \\ &= \sum_{i \in S_k} x_i^t \mathbf{\Sigma}_i^{-1} x_i - \hat{\theta}_k^t \Psi_k^{-1} \hat{\theta}_k, \qquad k = 1, \ldots, G. \end{aligned} \tag{21}$$

The second line in Equation 21 follows from Equations 5 and 11.

From Equations 5 and 11, and the fact that $S_w = S_u \cup S_v$, it follows that

$$\Psi_w = (\Psi_u^{-1} + \Psi_v^{-1})^{-1}, \tag{22}$$

$$\hat{\theta}_w = (\Psi_u^{-1} + \Psi_v^{-1})^{-1} (\Psi_u^{-1} \hat{\theta}_u + \Psi_v^{-1} \hat{\theta}_v). \tag{23}$$

Substituting Equations 21 and 22 into Equation 20 gives

$$\begin{aligned} \triangle E_{uv} &= (\sum_{i \in S_u \cup S_v} x_i^t \Sigma_i^{-1} x_i - \hat{\theta}_w^t (\Psi_u^{-1} + \Psi_v^{-1}) \hat{\theta}_w) \\ &\quad -(\sum_{i \in S_u} x_i^t \Sigma_i^{-1} x_i - \hat{\theta}_u^t \Psi_u^{-1} \hat{\theta}_u) - (\sum_{i \in S_v} x_i^t \Sigma_i^{-1} x_i - \hat{\theta}_v^t \Psi_v^{-1} \hat{\theta}_v) \\ &= -\hat{\theta}_w^t (\Psi_u^{-1} + \Psi_v^{-1}) \hat{\theta}_w + \hat{\theta}_u^t \Psi_u^{-1} \hat{\theta}_u + \hat{\theta}_v^t \Psi_v^{-1} \hat{\theta}_v \end{aligned} \tag{24}$$

Equation 23 gives

$$(\Psi_u^{-1} + \Psi_v^{-1})^{\frac{1}{2}} \hat{\theta}_w = (\Psi_u^{-1} + \Psi_v^{-1})^{-\frac{1}{2}} (\Psi_u^{-1} \hat{\theta}_u + \Psi_v^{-1} \hat{\theta}_v). \tag{25}$$

Taking dot product with itself on both sides of Equation 25 gives

$$\begin{aligned} \hat{\theta}_w^t (\Psi_u^{-1} + \Psi_v^{-1}) \hat{\theta}_w &= (\Psi_u^{-1} \hat{\theta}_u + \Psi_v^{-1} \hat{\theta}_v)^t (\Psi_u^{-1} + \Psi_v^{-1})^{-1} (\Psi_u^{-1} \hat{\theta}_u + \Psi_v^{-1} \hat{\theta}_v) \\ &= \hat{\theta}_u^t \Psi_u^{-1} (\Psi_u^{-1} + \Psi_v^{-1})^{-1} \Psi_u^{-1} \hat{\theta}_u \\ &\quad + \hat{\theta}_v^t \Psi_v^{-1} (\Psi_u^{-1} + \Psi_v^{-1})^{-1} \Psi_v^{-1} \hat{\theta}_v \\ &\quad + 2\hat{\theta}_u^t \Psi_u^{-1} (\Psi_u^{-1} + \Psi_v^{-1})^{-1} \Psi_v^{-1} \hat{\theta}_v. \end{aligned} \tag{26}$$

The last term of Equation 26 can be rewritten as

$$\begin{aligned} 2\hat{\theta}_u^t \Psi_u^{-1} (\Psi_u^{-1} + \Psi_v^{-1})^{-1} \Psi_v^{-1} \hat{\theta}_v &= \hat{\theta}_u^t \Psi_u^{-1} (\Psi_u^{-1} + \Psi_v^{-1})^{-1} \Psi_v^{-1} \hat{\theta}_u \\ &\quad + \hat{\theta}_v^t \Psi_u^{-1} (\Psi_u^{-1} + \Psi_v^{-1})^{-1} \Psi_v^{-1} \hat{\theta}_v \\ &\quad - (\hat{\theta}_u - \hat{\theta}_v)^t \Psi_u^{-1} (\Psi_u^{-1} + \Psi_v^{-1})^{-1} \Psi_v^{-1} (\hat{\theta}_u - \hat{\theta}_v). \end{aligned} \tag{27}$$

Substituting Equation 27 in Equation 26 gives

$$\begin{aligned}\hat{\theta}_w^t(\Psi_u^{-1}+\Psi_v^{-1})\hat{\theta}_w &= \hat{\theta}_u^t\Psi_u^{-1}(\Psi_u^{-1}+\Psi_v^{-1})^{-1}(\Psi_u^{-1}+\Psi_v^{-1})\hat{\theta}_u \\ &\quad +\hat{\theta}_v^t(\Psi_u^{-1}+\Psi_v^{-1})(\Psi_u^{-1}+\Psi_v^{-1})^{-1}\Psi_v^{-1}\hat{\theta}_v \\ &\quad -(\hat{\theta}_u-\hat{\theta}_v)^t\Psi_u^{-1}(\Psi_u^{-1}+\Psi_v^{-1})^{-1}\Psi_v^{-1}(\hat{\theta}_u-\hat{\theta}_v) \\ &= \hat{\theta}_u^t\Psi_u^{-1}\hat{\theta}_u+\hat{\theta}_v^t\Psi_v^{-1}\hat{\theta}_v-(\hat{\theta}_u-\hat{\theta}_v)^t\Psi_u^{-1}(\Psi_u^{-1}+\Psi_v^{-1})^{-1}\Psi_v^{-1}(\hat{\theta}_u-\hat{\theta}_v) \quad (28)\end{aligned}$$

Further,

$$\begin{aligned}\Psi_u^{-1}(\Psi_u^{-1}+\Psi_v^{-1})^{-1}\Psi_v^{-1} &= [\Psi_u^{-1}(\Psi_u^{-1}+\Psi_v^{-1})^{-1}\Psi_v^{-1}(\Psi_u+\Psi_v)](\Psi_u+\Psi_v)^{-1} \\ &= [\Psi_u^{-1}(\Psi_u^{-1}+\Psi_v^{-1})^{-1}\Psi_v^{-1}\Psi_u+\Psi_u^{-1}(\Psi_u^{-1}+\Psi_v^{-1})^{-1}](\Psi_u+\Psi_v)^{-1} \\ &= [\Psi_u^{-1}(\Psi_u^{-1}+\Psi_v^{-1})^{-1}\Psi_v^{-1}\Psi_u+\Psi_u^{-1}(\Psi_u^{-1}+\Psi_v^{-1})^{-1}\Psi_u^{-1}\Psi_u](\Psi_u+\Psi_v)^{-1} \\ &= [\Psi_u^{-1}(\Psi_u^{-1}+\Psi_v^{-1})^{-1}(\Psi_v^{-1}+\Psi_u^{-1})\Psi_u](\Psi_u+\Psi_v)^{-1} \\ &= [\Psi_u^{-1}\Psi_u](\Psi_u+\Psi_v)^{-1}=(\Psi_u+\Psi_v)^{-1}. \quad (29)\end{aligned}$$

Substituting Equations 28 and 29 into Equation 24 gives

$$\triangle E_{uv}=(\hat{\theta}_u-\hat{\theta}_v)^t(\Psi_u+\Psi_v)^{-1}(\hat{\theta}_u-\hat{\theta}_v). \quad (30)$$

Minimizing $\triangle E_{uv}$ is, therefore, the same as minimizing the distance $d_{uv}=(\hat{\theta}_u-\hat{\theta}_v)^t(\Psi_u+\Psi_v)^{-1}(\hat{\theta}_u-\hat{\theta}_v)$ among all possible pairs of clusters, C_u and C_v. □

The distance function in Equation 17 is similar to the *Mahalanobis distance* function [21]. It becomes equivalent to *Mahalanobis distance* when $\Psi_u=\Psi_v$.

4.3 Number Of Clusters

The problem of finding the number of clusters in a data set has been studied by several researchers. There is no single best method; the best method depends on the clustering method being used and the application at hand [23]. A common practice in most hierarchical clustering methods is to subjectively choose a threshold value for the distance function to be used as a stopping criterion in the merging process of the hierarchical method. The value of such threshold usually depends on the data being clustered [15, 23]. We provide a new method for estimating the number of clusters where the value of the threshold does not depend on data.

The new method for estimating the number of clusters involves testing a series of hypotheses, one at each stage of the *hError* algorithm. The first and the simplest hypothesis is the one where all $\mu_i, i=1,\ldots,n$, are different. At any intermediate stage of the *hError* algorithm, we measure consistency of the clusters produced by the algorithm by testing the significance level of the hypothesis that the means (μ_i) are equal within clusters. If the hypothesis is rejected, we discard the last merge operation and stop the algorithm. Otherwise, the algorithm continues with the next merge operation followed by test of a new hypothesis.

Let the clusters at an intermediate stage of *hError* be $S_1, \ldots, S_G$, then we need to evaluate consistency with hypothesis of the form

$$H_G : \mu_i = \theta_k, \qquad \forall i \in S_k, \;\; k = 1, ..., G. \tag{31}$$

We test consistency with hypothesis H_G using the following statistic.

$$Z_G^2 = \sum_{k=1}^{G} \sum_{i \in S_k} (x_i - \hat{\theta}_k)^t \mathbf{\Sigma}_i^{-1} (x_i - \hat{\theta}_k). \tag{32}$$

From standard multivariate theory, we know that Z_G^2 follows a chi-square distribution with $(n - G)p$ degrees of freedom, and we therefore reject H_G at significance level of $(1 - \alpha)$ if $Z_G^2 > \chi^2_{\alpha,(n-G)p}$.

The hypothesis is clearly consistent at the first stage of the *hError* algorithm, when there are n clusters ($Z_G^2 = 0$ at the first stage). The algorithm stops merging clusters when it encounters the first hypothesis that is rejected, because further merging of the clusters will generally give a less consistent clustering [8]. In our implementation of *hError*, we have used $\alpha = 0.01$.

We must note that the hypothesis testing proposed here is a heuristic and provides only a rough measure of the quality of clustering for the following reason. The merging process of *hError* selectively puts data points that are close-by in the same cluster, whereas the Z_G^2 statistic assumes that the clusters are random sets of points. Therefore, the value of the Z_G^2 statistic will generally be underestimated according to $\chi^2_{\alpha,(n-G)p}$ measure. This makes the proposed measure a liberal measure of quality of clustering in the sense that the proposed measure will accept a clustering even if it should be rejected with probability α. This implies that *hError* will tend to produce fewer number of clusters than the true number of clusters in the data [3]. We found this to be true in the simulation study presented in Section 7.

4.4 *hError* Algorithm

The *hError* algorithm is formally described in Algorithm 1, which has time complexity of $O(n^2)$. The algorithm is a generalization of Ward's method for hierarchical clustering [1, 17]. In the special case when $\Sigma_i = \sigma^2 * I$ for all $i = 1, \ldots, n$, the *hError* algorithm specializes to Ward's algorithm. The proof follows from Proposition 3.1.

5 The *kError* Clustering Algorithm

We next present another heuristic algorithm, *kError*, that is appropriate when the number of clusters, G, is given. *kError* is similar to the well-known k-means algorithm. It is an iterative algorithm that cycles through the following two steps.

- Step 1: For a given a clustering, compute the cluster centers as the *Mahalanobis means* of the clusters.

[3] This problem is analogous to the problem of using F test in step-wise multiple linear regression.

Algorithm 1 : $hError(x, \Sigma)$

Input: $(x_i, \mathbf{\Sigma}_i), i = 1, ..., n$.
Output: Clusters $C_1, ..., C_G$.
initialization:
  $C_i = \{x_i\}, i = 1, \ldots, n$;
  $G = n$;
  $p_{\chi^2_G} = 1$;
  Calculate pairwise distances between all pairs of clusters using Equation 17;
  For each cluster, record its closest cluster;
end initialization
while $Z^2_G \leq \chi^2_{\alpha,(n-G)p}$ **do**
  Find the closest pair of clusters $\{C_u, C_v\}, u < v$;
  Merge C_u and C_v into one cluster, $C_u \cup C_v$;
  Calculate distances of $C_u \cup C_v$ to all other clusters;
  For each cluster, update its closest cluster (in case $C_u \cup C_v$ is closer to it than its previous closest cluster);
  $G = G - 1$;
  Calculate Z^2_G using Equation 32;
end while
Discard the last merge operation if there was any;
return $C_1, \ldots, C_G$;

- Step 2: Reassign each data point to the closest cluster center using the distance formula in Equation 33.

The distance of a point, x_i, from a cluster center, $\hat{\theta}_k$, of the cluster C_k is given by

$$d_{ik} = (x_i - \hat{\theta}_k)^t \mathbf{\Sigma}_i^{-1} (x_i - \hat{\theta}_k). \tag{33}$$

We must note that the distance functions in Equation 33 is different from the one in Equation 17. While the latter one is used for calculating distance between two intermediate clusters in *hError*, the former one is used for calculating the distance of a data point from a cluster center in *kError*. We should also note that the distance function in Equation 33 does not contain the error term, Ψ_k, associated with $\hat{\theta}_k$. We have chosen this distance function because it guarantees a decrease in the value of the objective function in each iteration of *kError*, as shown in Lemma 5.1. The difference between these distance functions is analogous to the difference in the distance functions used in Ward's and k-means methods [1].

Lemma 5.1. *The kError algorithm converges in a finite number of iterations.*

Proof. We will show that the value of the objective function in Equation 4 decreases strictly at each iteration of the *kError* algorithm. Since there are only a finite number of different clusterings, finite convergence of *kError* follows.

We showed in the proof of Lemma 3.1 that, for a given clustering of data, the value of the objective criterion of error-based clustering is minimized when the cluster centers (θ_k's) are chosen to be the *Mahalanobis means* of the clusters. Thus, Step 1 never increases the value of the objective function.

In Step 2, a data point is reassigned from its current cluster to a new cluster only if it is nearer to the new cluster according to the distance function in Equation 33. Thus, for each data point reassigned, the value of the objective function decreases more for the losing cluster than it increases for the gaining cluster, thereby giving an overall decrease in the value of the objective function. □

The *kError* algorithm is formally described in Algorithm 2. The time complexity of the *kError* algorithm is linear in the number of data points, n, and the number of iterations the algorithm makes. In our empirical study, we found that the algorithm generally converges after only a few iterations; therefore, *kError* is generally faster than the *hError* algorithm. The tradeoff is that *kError* requires a priori knowledge of the number of clusters.

The *kError* algorithm is a generalization of the k-means algorithm for clustering. In the special case when $\Sigma_i = \sigma^2 * I$ for all $i = 1, \ldots, n$, the *kError* algorithm specializes to k-means algorithm. The proof follows from Proposition 3.1.

Algorithm 2 : $kError(x, \Sigma, G)$

Input: $(x_i, \mathbf{\Sigma}_i), i = 1, ..., n$; and G.
Output: Clusters $C_1, ..., C_G$.
initialization:
  Find an initial random partition of the data into G clusters;
end initialization
Step 1:
  Compute G cluster centers using Equation 5;
end Step 1
Step 2:
  Reassign each data point to its closest cluster using Equation 33;
end Step 2
if Clusters change **then**
  go to Step 1;
end if
return $C_1, ..., C_G$;

A drawback of the *kError* algorithm is that the final clusters may depend on the initial partition. It can also produce empty clusters if all points in a cluster are reassigned to other clusters, thereby reducing the number of clusters. k-means also suffers from these problems [25]. We propose the following solution that is similar to the one that is often used in k-means [25]. Run the *kError* algorithm a large number of times with different random initial partitions and pick the one that has the least value of the objective function. We ignore those solutions that contain one or more empty clusters. The authors of [25] show that, if

the k-means algorithm is run a large number of times, the resulting clusters will be close to optimal and insensitive to the initial partition. In our empirical studies, we have found that this is also true for the *kError* algorithm.

6 Applications of Error-Based Clustering

In this section, we present clustering applications where error information about data to be clustered is readily available and where error-based clustering results are typically superior to the standard clustering methods that ignore error information. We focus on clustering problems where the objects to be clustered are modelled using statistical models. Each object in this case is identified by the parameters of a statistical model. A commonly used method for estimating the parameters of any statistical model is the maximum likelihood method, which also provides the covariance matrix associated with the estimates of the model parameters. If we wish to cluster these objects on the basis of the model parameter estimates, then error based clustering can be applied on the estimates of the parameters and their covariance matrices. The central result of this paper is that, in this application, the objective criterion of error based clustering is approximately the optimal clustering criterion if the goal is to maximize the likelihood of the clustering based on the observed data for each object. We illustrate this application of error based clustering by examining four statistical models: (1) sample averaging (2) multiple linear regression, (3) time series models, and (4) Markov chain models.

6.1 Motivation for Clustering of Model Parameters

Clustering algorithms typically assume that the data to be clustered is available in a vector form in fixed dimensions. For example, given p measurements on a set of individuals, we represent each individual by a p-dimensional vector consisting of these measurements. However, there are many clustering problems when individuals to be clustered do not have an obvious vector representation in a fixed dimension. Examples include clustering of time series data or clustering of individuals based on web browsing behavior (see [7] for more examples). Tables 1 and 2 show two examples of such data.

User 1	session 1	1	2	2	2	5	4	2	6				
	session 2	4	1	3	1	1	4	1	5	6	2		
	session 3	2	3	4	7	2	3						
User 2	session 1	5	2	1	4	5	2	3	6				
	session 2	2	7	9	3	3	4	4	4	1	2	6	7	5
User 3	session 1	4	2	7	2	3	2	5	4	8	3	2		

Table 1: An example of web navigation data

The first table contains sequences of web page requests made by three users on a web site, where each web page is identified by a unique page number. The second table contains sales

	Week #	1	2	3	4	5	6	7	8	9	10	11	12	13
Item 1	Sales	14	23	32	29	35	41	27	46					
Item 2	Sales			17	21	38	42	56	44	58	37	23		
Item 3	Sales	24	27	29	35	32	45	48	42	61	57	68	70	57

Table 2: An example of weekly sales data for a quarter

data for three items in a store (each item is sold for different number of weeks). We would like to cluster these users (or items) based on their web navigation patterns (or sales data), but it is not clear how to obtain such clusters using the standard clustering techniques.

[7, 22] and references therein have shown that the data to be clustered in these examples can be transformed to a fixed dimension vector representation via a preprocessing step. By modelling the data for each object as coming from a statistical model (for example, Markov chain model for the web navigation example and time series model for the sales data example), the preprocessing step generates a set of vectors of model parameters of fixed dimension (one vector for each object) that can be clustered using the standard clustering techniques, as shown in Figure 3. In the above example, when online users are modelled as mixture of Markov chains, each user is represented by a vector consisting of the transition probabilities of the underlying Markov chain, and then the clusters can be obtained based on similarity between the estimates of transition probabilities for different users [2, 7]. Similarly, when the sales pattern of an item is modelled as a mixture of time series, the clusters of items can be obtained based on similarity between the estimates of the time series parameters for each item [22, 24].

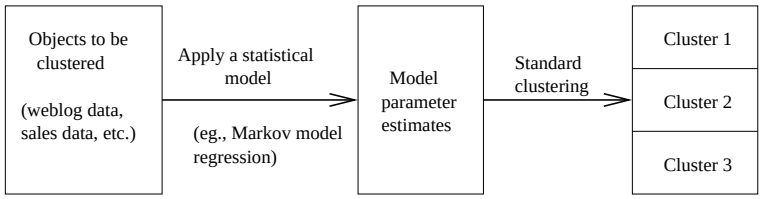


Figure 3: Clustering of model parameters using standard clustering.

Another example is clustering of similar stocks based on their β values that can be obtained by using ordinary linear regression (the Capital Asset Pricing Model) on the observed stock prices [16]. More details on this example are given in Section 7.2. A sample data is shown below.

A commonly used method for estimating model parameters is the maximum likelihood method, which also provides us with the covariance matrix associated with the estimate of the parameters. The estimated parameters follow approximately multivariate Gaussian distribution. This makes it natural to use error-based clustering for clustering of the model parameters, as shown in Figure 4.

In the next subsection we show that if our goal is to maximize the likelihood of the observed data under the specified statistical model, then error based clustering of the estimated model parameters is approximately the optimal clustering method. In other words, solution

	Quarter #	1	2	3	4	5	6	7	8	9	10
Stock 1	Stock return	8.2%	6.7%	9.1%	9.7%	5.4%	4.1%	7.2%	6.4%	8.1%	5.9%
	Market return	6.0%	4.7%	8.7%	7.9%	4.5%	4.4%	5.2%	6.0%	7.5%	5.0%
Stock 2	Stock return	6.1%	4.6%	7.8%	8.3%	4.2%	5.1%	5.0%	6.4%	7.1%	4.3%
	Market return	6.0%	4.7%	8.7%	7.9%	4.5%	4.4%	5.2%	6.0%	7.5%	5.0%
Stock 3	Stock return	7.1%	4.5%	9.0%	9.7%	4.0%	3.9%	5.5%	7.0%	9.8%	5.0%
	Market return	6.0%	4.7%	8.7%	7.9%	4.5%	4.4%	5.2%	6.0%	7.5%	5.0%

Table 3: An example of stock price returns for 10 quarters

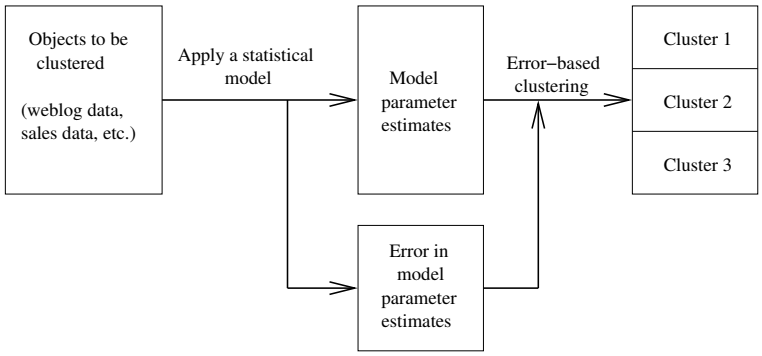


Figure 4: Clustering of model parameters using error-based clustering.

obtained by the two step approach as in Figure 4 is approximately same as the solution obtained by solving a single likelihood problem as in Figure 5.

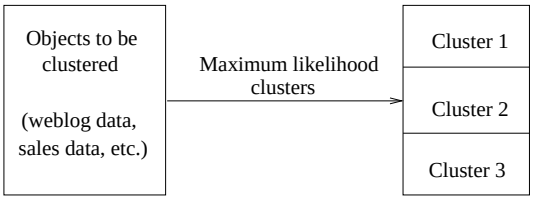


Figure 5: Maximum likelihood clustering of the observed data.

6.2 Approximate Optimality of Error-based Clustering of Model Parameters

Let there be n objects having m_i observation points for the i^{th} object, $X_i = (x_{i1}, x_{i2}, \ldots, x_{im_i})$, for $i = 1, ..., n$. For example, in Table 2, there are three objects with 8, 9 and 13 observation points corresponding to the number of weeks these items were sold. We assume that the observed data for the i^{th} object comes from a statistical model based on a set of parameters $\theta_i = (\theta_{i1}, ..., \theta_{ip})$. The log-likelihood of the observed data for the i^{th} object is given by

$$\ell(X_i|\theta_i) = \ell(x_{i1}, x_{i2}, ..., x_{im_i}|\theta_i) = \ell(x_{i1}|\theta_i) + \ell(x_{i2}|\theta_i) + ... + \ell(x_{im_i}|\theta_i). \tag{34}$$

We assume that the standard regularity conditions about the density function hold [26], that is, the integral of the density function can be twice differentiated and the third derivative of log-likelihood satisfies

$$|(\frac{\partial^3 \ell(X_i|\theta)}{\partial\theta\partial\theta'\partial\theta})| \leq M(x) \tag{35}$$

for all x and $\theta_i - c \leq \theta \leq \theta_i + c$ with $E_{\theta_i}[M(X)] < \infty$.

The maximum likelihood estimate of θ_i is obtained by solving the following set of equations.

$$\frac{\partial \ell(X_i|\theta)}{\partial\theta} = 0 \tag{36}$$

Let the solution to this set of equations be $\hat{\theta}_i$, then an estimate of the covariance matrix associated with $\hat{\theta}_i$ is given by

$$\Sigma_i = (-[\frac{\partial^2 \ell(X_i|\theta)}{\partial\theta\partial\theta'}])^{-1}_{\hat{\theta}_i} \tag{37}$$

Let us assume that the objects to be clustered are characterized well by the estimates of the model parameters so that similarity in the model parameters space corresponds to similarity in the original objects. Then the objects could be clustered based on their model parameter estimates. Here, the input to clustering consists of n sets of model parameter estimates, $\hat{\theta}_i$, and associated error matrices, Σ_i, for $i = 1, 2, ..., n$.

Consider a cluster C_k that contains n_k objects with indices $S_k = \{i_1, i_2, ..., i_{n_k}\}$. Let us denote the entire observed data in cluster C_k by $X_{S_k} = (X_{i_1}, X_{i_2}, ..., X_{i_{n_k}})$. If we assume that all data in this cluster have a common model parameter, θ_{S_k}, then it can be estimated by solving the following set of equations

$$\frac{\partial \ell(X_{S_k}|\theta)}{\partial\theta} = 0, \tag{38}$$

where

$$\ell(X_{S_k}|\theta) = \ell(X_{i_1}, X_{i_2}, ..., X_{i_{n_k}}|\theta) = \sum_{i\in S_k} \ell(X_i|\theta). \tag{39}$$

Let the solution to this set of equations be $\hat{\theta}_{S_k}$, then an estimate of the covariance matrix associated with $\hat{\theta}_{S_k}$ is given by

$$\Sigma_{S_k} = (-[\frac{\partial^2 \ell(X_{S_k}|\theta)}{\partial\theta\partial\theta'}])^{-1}_{\hat{\theta}_{S_k}} \tag{40}$$

Definition 6.1. *We call a cluster C_k a true cluster if all objects in the cluster have the same parameters for their underling statistical models, i.e., $\theta_i = \theta_{S_k}$ for all $i \in S_k$.*

Lemma 6.1. *We can make the following approximation for a true cluster.*

$$(\frac{\partial^2 \ell(X_i|\theta)}{\partial\theta\partial\theta'})_{\hat{\theta}_i} + (\frac{\partial^3 \ell(X_i|\theta)}{\partial\theta\partial\theta'\partial\theta})_{\theta^*}(\hat{\theta}_{S_k} - \hat{\theta}_i) \approx (\frac{\partial^2 \ell(X_i|\theta)}{\partial\theta\partial\theta'})_{\hat{\theta}_i} \qquad \forall i \in S_k \tag{41}$$

where θ^* lies between $\hat{\theta}_i$ and $\hat{\theta}_{S_k}$

Proof. From the asymptotic properties of maximum likelihood [26], we know that $\hat{\theta}_i - \hat{\theta}_{S_k} \to 0$ for all $i \in S_k$. Further, according to assumption in Equation 35, the third derivative of $\ell(X_i|\theta)$ is bounded; therefore, it is reasonable to make the above approximation. □

Lemma 6.2. *The maximum likelihood estimate of the common parameter in a true cluster can be approximated to be the Mahalanobis mean of the model parameter estimates of individual objects in the cluster, i.e.,*

$$\hat{\theta}_{S_k} \approx (\sum_{i \in S_k} \Sigma_i^{-1})^{-1} (\sum_{i \in S_k} \Sigma_i^{-1} \hat{\theta}_i). \tag{42}$$

Proof. Using Taylor series, we get the following

$$(\frac{\partial \ell(X_i|\theta)}{\partial \theta})_{\hat{\theta}_{S_k}} = (\frac{\partial \ell(X_i|\theta)}{\partial \theta})_{\hat{\theta}_i} + (\frac{\partial^2 \ell(X_i|\theta)}{\partial \theta \partial \theta'})_{\hat{\theta}_i} (\hat{\theta}_{S_k} - \hat{\theta}_i) + \frac{1}{2} (\hat{\theta}_{S_k} - \hat{\theta}_i)' (\frac{\partial^3 \ell(X_i|\theta)}{\partial \theta \partial \theta' \partial \theta})_{\theta^*} (\hat{\theta}_{S_k} - \hat{\theta}_i), \tag{43}$$

where θ^* lies between $\hat{\theta}_i$ and $\hat{\theta}_{S_k}$. Using approximation in Lemma 6.1, we get

$$(\frac{\partial \ell(X_i|\theta)}{\partial \theta})_{\hat{\theta}_{S_k}} \approx (\frac{\partial \ell(X_i|\theta)}{\partial \theta})_{\hat{\theta}_i} + (\frac{\partial^2 \ell(X_i|\theta)}{\partial \theta \partial \theta'})_{\hat{\theta}_i} (\hat{\theta}_{S_k} - \hat{\theta}_i) = -\Sigma_i^{-1} (\hat{\theta}_{S_k} - \hat{\theta}_i). \tag{44}$$

Since $\hat{\theta}_{S_k}$ is the MLE of the common parameter for the cluster, we have

$$0 = (\frac{\partial \ell(X_{S_k}|\theta)}{\partial \theta})_{\hat{\theta}_{S_k}} = \sum_{i \in S_k} (\frac{\partial \ell(X_i|\theta)}{\partial \theta})_{\hat{\theta}_{S_k}} \approx - \sum_{i \in S_k} \Sigma_i^{-1} (\hat{\theta}_{S_k} - \hat{\theta}_i). \tag{45}$$

Rearranging the terms in the above equation we get

$$\hat{\theta}_{S_k} \approx (\sum_{i \in S_k} \Sigma_i^{-1})^{-1} (\sum_{i \in S_k} \Sigma_i^{-1} \hat{\theta}_i). \tag{46}$$

□

Lemma 6.3. *The error matrix associated with $\hat{\theta}_{S_k}$ in a true cluster is approximately same as the error matrix associated with the Mahalanobis mean of the model parameter estimates of individual objects in the cluster, i.e.,*

$$\Sigma_{S_k} \approx (\sum_{i \in S_k} \Sigma_i^{-1})^{-1}. \tag{47}$$

Proof. From the Taylor series expansion and Lemma 6.1 we get

$$(\frac{\partial^2 \ell(X_i|\theta)}{\partial \theta \partial \theta'})_{\hat{\theta}_{S_k}} = (\frac{\partial^2 \ell(X_i|\theta)}{\partial \theta \partial \theta'})_{\hat{\theta}_i} + (\frac{\partial^3 \ell(X_i|\theta)}{\partial \theta \partial \theta' \partial \theta})_{\theta^*} (\hat{\theta}_{S_k} - \hat{\theta}_i) \approx (\frac{\partial^2 \ell(X_i|\theta)}{\partial \theta \partial \theta'})_{\hat{\theta}_i} = -\Sigma_i^{-1} \tag{48}$$

Therefore,

$$\Sigma_{S_k} = (-[\frac{\partial^2 \ell(X_{S_k}|\theta)}{\partial \theta \partial \theta'}])^{-1}_{\hat{\theta}_{S_k}} = (\sum_{i \in S_k} (-[\frac{\partial^2 \ell(X_i|\theta)}{\partial \theta \partial \theta'}])_{\hat{\theta}_{S_k}})^{-1} \approx (\sum_{i \in S_k} \Sigma_i^{-1})^{-1}. \tag{49}$$

□

Theorem 6.1. *Error-based clustering of the model parameters estimates produces approximately the same clusters as the maximum likelihood clusters of the observed data.*

Proof. Maximum likelihood clusters of the observed data are given by

$$\max_{S_1,\ldots,S_G;\theta_{S_1},\ldots,\theta_{S_G}} \sum_{k=1}^{G}\sum_{i\in S_k} \ell(X_i|\theta_{S_k}) = \max_{S_1,\ldots,S_G} \sum_{k=1}^{G}(\max_{\theta_{S_k}} \sum_{i\in S_k} \ell(X_i|\theta_{S_k})) = \max_{S_1,\ldots,S_G} \sum_{k=1}^{G}\sum_{i\in S_k} \ell(X_i|\hat{\theta}_{S_k}) \tag{50}$$

Each term on the right side of Equation 50 can be expanded using Taylor series as

$$\begin{aligned}
\ell(X_i|\hat{\theta}_{S_k}) &= \ell(X_i|\hat{\theta}_i) + (\hat{\theta}_{S_k}-\hat{\theta}_i)'(\frac{\partial \ell(X_i|\theta)}{\partial\theta})_{\hat{\theta}_i} + \frac{1}{2}(\hat{\theta}_{S_k}-\hat{\theta}_i)'(\frac{\partial^2 \ell(X_i|\theta)}{\partial\theta\partial\theta'})_{\hat{\theta}_i}(\hat{\theta}_{S_k}-\hat{\theta}_i) \\
&\quad + \frac{1}{6}(\hat{\theta}_{S_k}-\hat{\theta}_i)(\hat{\theta}_{S_k}-\hat{\theta}_i)'(\frac{\partial^3 \ell(X_i|\theta)}{\partial\theta\partial\theta'\partial\theta})_{\theta^*}(\hat{\theta}_{S_k}-\hat{\theta}_i) \\
&\approx \ell(X_i|\hat{\theta}_i) + 0 + \frac{1}{2}(\hat{\theta}_{S_k}-\hat{\theta}_i)'(-\Sigma_i^{-1})(\hat{\theta}_{S_k}-\hat{\theta}_i) \\
&= \ell(X_i|\hat{\theta}_i) - \frac{1}{2}(\hat{\theta}_{S_k}-\hat{\theta}_i)'\Sigma_i^{-1}(\hat{\theta}_{S_k}-\hat{\theta}_i)
\end{aligned} \tag{51}$$

Replacing Equation 51 in Equation 50, we get

$$\begin{aligned}
\max_{S_1,\ldots,S_G;\theta_{S_1},\ldots,\theta_{S_G}} \sum_{k=1}^{G}\sum_{i\in S_k} \ell(X_i|\theta_{S_k}) &\approx \max_{S_1,\ldots,S_G} \sum_{k=1}^{G}\sum_{i\in S_k} [\ell(X_i|\hat{\theta}_i) - \frac{1}{2}(\hat{\theta}_{S_k}-\hat{\theta}_i)'\Sigma_i^{-1}(\hat{\theta}_{S_k}-\hat{\theta}_i)] \\
&= \min_{S_1,\ldots,S_G} \sum_{k=1}^{G}\sum_{i\in S_k} (\hat{\theta}_{S_k}-\hat{\theta}_i)'\Sigma_i^{-1}(\hat{\theta}_{S_k}-\hat{\theta}_i),
\end{aligned} \tag{52}$$

which is error-based clustering of the model parameter estimates. The last equality follows because $\sum_{k=1}^{G}\sum_{i\in S_k} \ell(X_i|\hat{\theta}_i)$ is a constant. □

Corollary 6.1. *In a special case, when the log-likelihood function is a quadratic function (for example, in linear regression models under Gaussian assumption), error-based clustering of the model parameters estimates produces exactly the same clusters as the maximum likelihood clusters of the observed data.*

Proof. When the log-likelihood function is a quadratic function, the third derivative of log-likelihood is identically zero; therefore, the approximate results in Lemma 6.1, 6.2, 6.3 and Theorem 6.1 become exact results. □

Theorem 6.1 establishes that an optimal error-based clustering of the model parameter estimates is approximately the same as the maximum likelihood clusters of the observed data. This means that the clusters obtained in Figure 4 using the two-step decomposition method are approximately the same as the clusters obtained from a single maximization method, as in Figure 5. It is important to note that this would not be true if we clustered the model parameters estimates ignoring the errors, as in Figure 3.

In the remainder of this paper, we will present results from a series of empirical studies that suggest that error-based clustering is more appropriate than the traditional clustering methods for clustering of model parameters. While we focus our empirical study on four statistical models in this paper, the concept of using error-based clustering for clustering of model parameters is very general and can be applied to a large class of statistical models where maximum likelihood method is used to estimate the model parameters.

7 Empirical Study

In this section, we present results from empirical studies on simulated data on four statistical models: (1) sample averaging, (2) multiple linear regression, (3) time series, and (4) Markov chain models. We also present empirical study results on a real data set on time series. We experimented with six different clustering methods: kError, hError, k-means, Ward, k-means with normalization, and Ward with normalization. Clustering results from Ward and k-means methods depend on the units of measurement, whereas kError and hError results are unit-independent, therefore we applied k-means and Ward after normalizing the data to unit variance on each variable. We evaluate a clustering method by its misclassification error, i.e., the number of data points assigned to an incorrect cluster. In our study, we found that the misclassification error is significantly smaller for hError and kError than the other methods. We also found that kError and k-means perform better than hError and Ward's methods, respectively. The reason for this is as follows. In our implementation, we run the kError and k-means algorithms with 50 different random initial solutions and pick the one that achieves the best objective value. This helps kError and k-means achieve better solution than hError and Ward's methods, respectively.

7.1 Clustering of Sample Means

There are many situations when one has access to only aggregated data. This may happen because of the need to simplify data management, for example, in census data or transportation traffic data, or due to confidentiality issues, for example, in data from clinical trials or surveys. The aggregated data is often represented by its sample mean and variance statistics (or the error information) associated with the sample mean. Our goal is to cluster samples of such data based on their sample means and variance statistics (this is similar to the ANOVA and MANOVA approaches to data analysis [5].)

7.1.1 Data Generation

Thirty samples of data were generated from a mixture of three bivariate Gaussian populations (one population for each cluster) with means μ_1, μ_2 and μ_3 and randomly generated covariance matrices $\Sigma_1, \ldots, \Sigma_{30}$ (ten samples were generated corresponding to each mean, but the covariance was different for each sample). The value of μ are the same for samples in the same cluster but differ between samples in different clusters. For each sample, we generated 10 data points from the corresponding Gaussian distribution. We represent each

sample by the estimated sample mean and sample covariance on these 10 data points. Given sample means and covariance matrix estimates for thirty samples (without the knowledge of their cluster membership), our goal is to partition them into three clusters that correspond to the true clusters. We used the following values of parameters for this experiment.

$$\mu_1 = \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \qquad \mu_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \qquad \mu_3 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

Random covariance matrices for thirty samples were generated as follows. We generated a 2×1 vector v_i of uniformly distributed random numbers between 0 and 3, and then $\Sigma_i = v_i * v_i'$ for $i = 1, \ldots, 30$.

7.1.2 Clustering Results

The average misclassification errors using various clustering methods in 100 replications of the above experiment are reported in Table 4. We found that the misclassification errors are much smaller for *kError* and *hError* than for the other methods.

Clustering Method	Misclassification Error (Average)
hError	5.46
Ward	13.53
Ward (normalized)	14.00

Clustering Method	Misclassification Error (Average)
kError	3.59
k-means	7.14
k-means (normalized)	7.08

Table 4: Average Misclassification Error

On the above experiment, *hError* was able to find the correct number of clusters (three in this case) 86 times in 100 runs of the experiment. On the remaining 14 runs it found two clusters. This is consistent with the theory we presented in Section 4.3. We repeated the above experiment with different number of clusters and different values of μ and obtained similar results.

In many practical situations it is not only infeasible to obtain $p \times p$ variance-covariance matrix associated with each sample mean, but also it is an over-parametrization of the samples, especially for high-dimensional data. Often, we only have access to the variance on each variable of the samples. In such cases, we can approximate the variance-covariance matrix by a diagonal matrix that has variance terms on diagonal and zero on off-diagonal. We studied the effect of this approximation in the above experiment and found that kError and hError still produced better clusters than k-means and Ward's methods, but the results were slightly worse than when we used entire covariance matrix for each sample mean.

7.2 Clustering of Stocks: An Example of Clustering Multiple Linear Regression Models

Suppose we want to cluster a set of stocks based on their performance against the overall market performance. A commonly used model to measure the performance of a stock against

the market performance is via the Capital Asset Pricing Model (CAPM) described below. [4]

$$R_{it} - R_f = \alpha_i + \beta_i(R_{mt} - R_f) + \epsilon_{it} \qquad i = 1, \ldots, N, \qquad t = 1, \ldots, T, \tag{53}$$

where R_{it} is the return on stock i in time period t, R_f is the return on a "risk-free" investment, i.e., fixed deposit in a bank, and R_{mt} is the return on a market benchmark, like the S&P 500. Here β_i for a stock i is a measure of risk profile of the stock, that is, higher the β more risky is the stock, and α is a measure of how much better (or worse) the stock did than CAPM predicted.

We can rewrite the above equation as

$$R_{it} = \alpha_i + \beta_i R_{mt} + \epsilon_{it} \qquad i = 1, \ldots, N, \qquad t = 1, \ldots, T, \tag{54}$$

where new $\alpha_i = \alpha_i + R_f - \beta_i * R_f$. Given observed data on the market return and each stock's return for a range of time periods, we can estimate its α and β using a simple linear regression.

In many applications it is useful to cluster stocks that have similar values of α and β. Through a study on simulated datasets, we illustrate that error-based clustering produces better clusters than standard clustering methods on this example.

7.2.1 Data Generation

We generated data for thirty stocks from three clusters, ten from each cluster. The values of (α, β) are the same for stocks in the same cluster but differ between stocks in different clusters. For the purpose of this experiment we used the following three values of $(\alpha, \beta) = (0.0, 1.0)$, $(-1.0, 1.5)$, and $(1.0, 0.5)$, one for each cluster. Once the values of (α, β) are chosen for a stock, we generate its return in ten quarters using Equation 54, where R_{mt}, market return for each quarter, is randomly generated for each stock from a uniform distribution between 3 % and 8 %. ϵ_{it} is generated from a normal distribution $N(0, 0.25)$. Given stock return and market return data for thirty stocks (without the knowledge of their cluster membership), our goal is to partition them into three clusters that correspond to the true clusters.

The maximum likelihood estimates of the regression parameters and associated covariance matrix for each stock can be obtained using the ordinary least square method on the return data during ten quarters for each stock. Let these estimates be $(\hat{\alpha}_i, \hat{\beta}_i)$ and Σ_i, for $i = 1, \ldots, 30$. This constitutes the data to be clustered.

7.2.2 Clustering Results

Table 5 presents the average misclassification error for various clustering methods in 100 replications of the above experiment.

[4] Another popular model is Fama and French Three Factor Model that uses three independent variables in regression: (1) risk-free market return, same as in CAPM, (2) SMB that stands for "small minus big cap" and (3) HML that stands for "high minus low book value".

Clustering Method	Misclassification Error (Average)
hError	0.00
Ward	11.33
Ward (normalized)	8.51

Clustering Method	Misclassification Error (Average)
kError	0.00
k-means	8.53
k-means (normalized)	5.31

Table 5: Average Misclassification Error

We see that on this example *kError* and *hError* always found the correct clusters with no misclassification error, whereas other methods made significant number of misclassifications. Using the method described in section 4.3, *hError* was able to find the right number of clusters (three in this case) 92 times in 100 runs of the above experiment. On the remaining 8 runs it found two clusters. We repeated the above experiment with different number of clusters and different values of (α, β) and obtained similar results.

7.3 Clustering of Time Series Models

Clustering of time series has been widely studied in the literature [12, 18, 22, 24]. The method in [24] is similar to ours except that the author does not account for the errors associated with the estimated parameters for each time series, that is, he applies standard clustering to the estimated parameters as shown in Figure 3. [12] and [22] have proposed probability model approach to clustering of time series data. In this approach, series that arise from the same model are classified in the same cluster. The method in [12] is an integrated approach based on the EM algorithm where the model parameters and cluster membership are estimated iteratively (this approach is similar to one shown in Figure 5). Our approach is a two step process: (a) estimate the parameters in the first step and then (b) cluster time series on the basis of the estimated parameters in the second step, as shown in Figure 4. Our method is very similar to the one in [22], except that here the author uses a hypothesis testing approach for measuring proximity between a pair of time series, whereas we use the Mahalanobis distance between estimated parameters for a pair of time series.

7.3.1 Data Generation

We illustrate the proposed methodology using a mixture of autoregressive (AR) models of order p,

$$y_{it} = \phi_{i1} y_{it-1} + \cdots + \phi_{ip} y_{it-p} + \epsilon_{it}, \qquad i = 1, \ldots, n, \tag{55}$$

where y_{it} is the value of the i^{th} time series at time t; $\phi_{i1}, \ldots, \phi_{ip}$ are the model parameters for this time series; and ϵ_{it} are independent and Gaussian distributed random errors.

The maximum likelihood estimates $\hat{\phi}_i = (\hat{\phi}_{i1}, \ldots, \hat{\phi}_{ip})$ and the associated covariance matrices Σ_i are obtained for each time series using the System Identification Toolbox in MATLAB. We cluster n p-dimensional vectors, $\hat{\phi}_1, \ldots, \hat{\phi}_n$, using error-based clustering. The resulting clusters correspond to clustering of the observed time series data.

For this experiment we generated a set of time series from AR(2) models. We considered three sets of values for (ϕ_{i1}, ϕ_{i2}) = (0.2, 0.1), (0.4, 0.5) and (0.6, 0.2), one for each of the three clusters. Ten time series were generated from each cluster, giving a total of thirty time series. For each time series we generated data for 50 time points. ϵ_{it} was chosen to be from Gaussian distribution $N(0, 0.01)$.

7.3.2 Clustering Results

The average misclassification error using various clustering methods in 100 replications of the above experiment is reported in Table 6. Here again we find that the misclassification error is smaller for *kError* and *hError* than for the other methods. On this application, the *hError* algorithm was able to find the correct number of clusters (three in this case) 84 times in the 100 runs of the experiment. Of the remaining 16 runs, it found two clusters on 15 runs and one cluster on one run. We repeated the above experiment with different number of clusters and different values of ϕ and obtained similar results.

Clustering Method	Misclassification Error (Average)
hError	5.25
Ward	8.18
Ward (normalized)	8.22

Clustering Method	Misclassification Error (Average)
kError	4.51
k-means	4.72
k-means (normalized)	4.76

Table 6: Average Misclassification Error

7.4 Clustering Online Shoppers: An Example of Clustering Markov Chain Models

Clustering of online users based on their navigation pattern has been studied in [2] and [7]. Similar to the work in [7], we assume that a user's online behavior can be described by an underlying Markov chain where page clicks correspond to different sates of the Markov chain. Given the observed page click data for each user, we can estimate the transition probabilities across different states of the underlying Markov chains. This gives us a set of transition probability vectors (in a fixed dimension), one for each user. The online users are then clustered based on similarity in their transition probability vectors.

7.4.1 Data Generation

We simulated a four-state online browsing behavior model as shown in Figure 6. A user starts at the *Start* page and moves to the *Shopping Cart* page and then to the *Place Order* page. The user can leave the site either at the *Start* page or at the *Shopping Cart* page. She can also go back to the *Start* page from the *Shopping Cart* page. Here, *Exit* and *Place Order* pages are the absorbing states. The transition probabilities between these states are

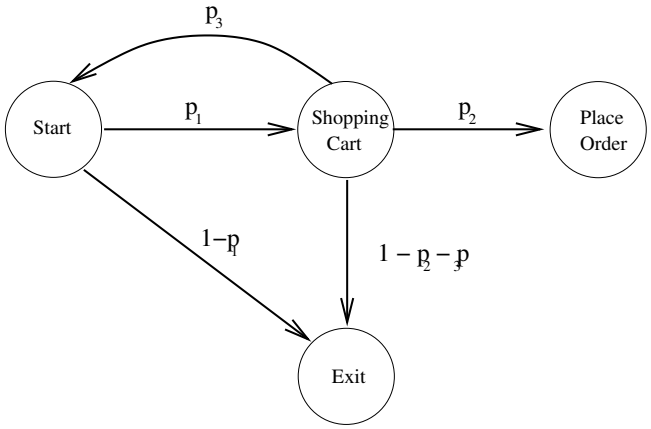


Figure 6: Markov chain model for online users.

shown in the figure. We do not allow self-transitions in this case (self-transition is equivalent to refreshing of a webpage, which can be ignored).

Behavior of a user is completely determined by the probability vector (p_1, p_2, p_3). In this experiment, we consider two sets of transition probabilities $(p_1, p_2, p_3) = (0.4, 0.6, 0.2)$ and $(0.6, 0.4, 0.3)$. Thirty users are generated for each set of probabilities, giving us a total of sixty users. We generate twenty visits (sessions) for each user. Based on these twenty visits, the maximum likelihood estimate of the transition probabilities $(\hat{p}_1, \hat{p}_2, \hat{p}_3)$ for each user was obtained using the following equations.

$$\hat{p}_1 = \frac{\text{Total number of transitions from Start to Shopping Cart page}}{\text{Total number of transitions from Start to any other page}} \tag{56}$$

$$\hat{p}_2 = \frac{\text{Total number of transitions from Shopping Cart to Place Order page}}{\text{Total number of transitions from Shopping Cart to any other page}} \tag{57}$$

$$\hat{p}_3 = \frac{\text{Total number of transitions from Shopping Cart to Start page}}{\text{Total number of transitions from Shopping Cart to any other page}} \tag{58}$$

The associated covariance matrix is given by

$$\Sigma = \begin{pmatrix} \hat{p}_1(1-\hat{p}_1)/n_1 & 0 & 0 \\ 0 & \hat{p}_2(1-\hat{p}_2)/n_2 & -\hat{p}_2\hat{p}_3/n_2 \\ 0 & -\hat{p}_2\hat{p}_3/n_2 & \hat{p}_3(1-\hat{p}_3)/n_2 \end{pmatrix},$$

where n_1 and n_2 are the total number of times the user visited the *Start* and *Shopping Cart* pages, respectively, in her twenty visits. We cluster these sixty users based on the values of $(\hat{p}_1, \hat{p}_2, \hat{p}_3)$ and Σ.

7.4.2 Clustering Result

The average misclassification error using various clustering methods in 100 replications of the above experiment is reported in Table 7. Here again we find that the misclassification

error is much smaller for *kError* and *hError* than for the other methods. On this application, the *hError* algorithm was able to find the correct number of clusters (two in this case) 89 times in the 100 runs of the experiment. On the remaining 11 runs, it found one cluster. We repeated the above experiment with different values of probabilities p and obtained similar results.

Clustering Method	Misclassification Error (Average)
hError	12.70
Ward	25.21
Ward (normalized)	25.63

Clustering Method	Misclassification Error (Average)
kError	9.83
k-means	13.84
k-means (normalized)	14.04

Table 7: Average Misclassification Error

7.5 Real Data Study

We studied the effectiveness of using error-based clustering on the personal income dataset [14], which is a collection of 25 time series representing the per capita personal income during 1929-1999 in 25 states of the USA [5]. This dataset was first studied by Kalpanis et. al. [18], where they used the Euclidean distance between the Linear Predictive Coding (LPC) cepstrum as a basis for clustering of ARIMA time-series. The authors believe that the dataset has two groups: one consists of the east coast states, CA, and IL where there is high growth rate in personal income, and the other consists of the mid-west states where there is low growth rate in personal income [6].

The per capita income time-series are non-stationary in mean as well as variance. To remove this non-stationarity, we applied the pre-processing steps used in [18]. We smoothed the original series by taking a window average over a window of size 2 and then took logarithm of the smoothed time-series. We fitted ARIMA(1,1,0) models to the resulting time-series. The ARIMA(1,1,0) model has only one parameter, ϕ_1. Clusters of time-series were formed on the basis of the values of the estimated parameter of the ARIMA(1,1,0) model and the associated error matrices. The resulting clusters were compared against the true clusters and the misclassification error is reported in Table 8. We also compared against the CEP method proposed in [18]. We found that *kError* and *kError* were able to discover the true clusters, whereas other methods did not.

We also conducted an out-of sample study as follows. First 55 years of data from each state was used to estimate the time series parameters, and then income for the last 15 years was predicted using the estimated common parameter for each cluster using various clustering methods. The accuracy of the predicted income was measured by Sum of Square Errors (SSE). Table 8 presents SSE for each clustering method. We also computed the SSE for the case when no clustering was used. We found that the standard clustering methods

[5] The 25 states included were: CT, DC, DE, FL, MA, ME, MD, NC, NJ, NY, PA, RI, VA, VT, WV, CA, IL, ID, IA, IN, KS, ND, NE, OK, and SD.

[6] The first 17 states form the first group and the last 8 states form the second group.

improved the forecast error over no clustering by a small amount, and error-based clustering improved it further by a significant amount.

Clustering Method	Misclassification Error	Sum of Square Error (SSE)
kError	0	0.0597
hError	0	0.0597
k-means	3	0.0623
Ward	3	0.0623
k-means with normalization	3	0.0623
Ward with normalization	5	0.0631
CEP	3	0.0623
No clustering	-	0.0636

Table 8: Misclassification Error and SSE measure on Personal Income data set

8 Summary and Future Research

In this paper, we have developed a new clustering technique that recognizes errors associated with data. We developed a probability model for incorporating error information in the clustering process and provided algorithms for error-based clustering. We have shown that the new clustering algorithms are generalizations of the popular Ward's and k-means clustering algorithms. We also showed that error-based clustering is invariant under an affine transformation of the data space, which is a valuable property in many applications of clustering. Finally, we studied application of error-based clustering on a large class of problems where data are clustered on the basis of similarity in the parameters of the underlying data generating statistical models. In this application, we showed that error-based clustering of the model parameter estimates is approximately same as the maximum likelihood clusters of the observed data. We also demonstrated the effectiveness of error-based clustering through a series of empirical studies on clustering of model parameters for four statistical models: sample averaging, linear regression, time series, and Markov chain models. While we studied these four statistical models in this paper, the concept of using error-based clustering for clustering of model parameters is very general and can be applied to a large class of statistical models where maximum likelihood method is used to estimate the model parameters.

Here are a few future directions for this research that we are exploring and wish to report the results at a later stage. The empirical study is done mostly on simulated data, except for one real data for time series models. We wish to do a detailed study with real data on clustering of other statistical models. In our empirical study we have found that *kError* is fast and gives better solution than *hError*, while *hError* has the advantage of automatically finding the number of clusters in a data set. One could combine the two algorithms to develop a new algorithm that is fast and finds the number of clusters automatically.

9 Acknowledgement

We would like to thank James B. Orlin, MIT Sloan e-Business Center and ProfitLogic Inc. for their valuable suggestions and for helping us with funding support to carry out this research.

References

[1] M.R. Anderberg, *Cluster Analysis for Applications*, Academic Press, 1973

[2] D. Bertsimas, A. J. Mersearau, and N. R. Patel, Dynamic Classification of Online Customers, In *Proceedings of the third SIAM International Conference on Data Mining*, 107 – 118, 2003

[3] J.D. Banfield and A.E. Raftery, Model-based Gaussian and Non-Gaussian Clustering, *Biometrics*, 49, 803 – 821, 1993

[4] B. B. Chaudhuri and P. R. Bhowmik, An Approach of Clustering Data with Noisy or Imprecise Feature Measurement, *Pattern Recognition Letters*, 19, 1307 – 1317, 1998

[5] P.S.P. Cowpertwait and T.F. Cox, Clustering Population Means under Heterogeneity of Variance with an Application to a Rainfall Time Series Problem, *The Statistician*, 41, 113 – 121, 1992

[6] G. Celeux and G. Govaert, Gaussian Parsimonious Clustering Models, *Pattern Recognition*, 28, 781 – 793, 1995

[7] I. V. Cadez, S. Gaffney, and P. Smyth, A general probabilistic framework for clustering individuals, In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 140 – 149, 2000

[8] D.R. Cox and E. Spjotvoll, On Partitioning Means into Groups, *Scandinavian Journal of Statistics*, 9, 147 – 152, 1982

[9] W.D. Fisher, On Grouping for Maximum Homogeneity, *Journal of the American Statistical Association*, 53, 789 – 798, 1958

[10] H. P. Friedman and J. Rubin, On Some Invariant Criteria for Grouping Data, *Journal of the American Statistical Association*, 62, 1159 – 1178, 1967

[11] C. Fraley and A.E. Raftery, How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis, *The Computer Journal*, 41(8), 578 – 588, 1998

[12] S. Gaffney and P. Smyth, Trajectory Clustering Using Mixtures of Regression Models, In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 63 – 72, 1999

[13] C. Hennig, Identifiability of Models for Clusterwise Linear Regression, *Journal of Classification*, 17, 273 – 296, 2000

[14] http://www.bea.gov/bea/regional/spi

[15] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*, Prentice-Hall, 1988

[16] P. Jones, *Investment Analysis and Management*, John Wiley & Sons Inc., 1991

[17] J. H. Ward Jr., Hierarchical Grouping to Optimize an Objective Function, *Journal of the American Statistical Association*, 58, 236 – 244, 1963

[18] K. Kalpakis, D. Gada, and V. Puttagunta, Distance Measures for Effective Clustering of ARIMA Time-Series, In *Proceedings of the 2001 IEEE International Conference on Data Mining*, 273 – 280, 2001

[19] M. Kumar, *Error-Based Clustering and Its Application to Sales Forecasting in Retail Merchandising*, Ph.D. Thesis, MIT, 2003

[20] M. Kumar, N.R. Patel, and J. Woo, Clustering Seasonality Patterns in the Presence of Errors, In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 557 – 563, 2002

[21] P.C. Mahalanobis, On the Generalized Distance in Statistics, In *Proceedings of the National Institute of Science of India*, 2, 49 – 55, 1936

[22] E.A. Maharaj, Clusters of Time Series, *Journal of Classification*, 17, 297 – 314, 2000

[23] G.W. Milligan and M.C. Cooper, An Examination of Procedures for Determining the Number of Clusters in a Data Set, *Psychometrika*, 50(2), 159 – 179, 1985

[24] D. Piccolo, A Distance Measure for Classifying Arima Models, *Journal of Time Series Analysis*, 11(2), 153 – 164, 1990

[25] J. Pena, J. Lozano, and P. Larranaga, An Empirical Comparison of Four Initialization Methods for the k-means Algorithm, *Pattern Recognition Letters*, 50, 1027 – 1040, 1999

[26] C.R. Rao and C.R. Radhakrishna, *Linear Statistical Inference and Its Application*, John Wiley & Sons Inc., 2002

[27] A. J. Scott and M. J. Symons, Clustering Methods Based on Likelihood Ratio Criteria, *Biometrics*, 27, 387 – 397, 1971

[28] Y.L. Tong, *The Multivariate Normal Distribution*, Springer-Verlag, 1990