Clustering Data with Measurement Errors

Mahesh Kumar, Nitin R. Patel, James B. Orlin

Operations Research Center, MIT

Working Draft Paper
September, 2002

Clustering Data with Measurement Errors

Working Draft Paper

Abstract

Most traditional clustering work assumes that the data is provided without measurement error. Often, however, real world data sets have such errors. Often one can obtain estimate of these errors. In the presence of such errors, popular clustering methods like k-means and hierarchical clustering may produce un-intuitive results.

The fundamental question that this paper addresses is: "What is an appropriate clustering method in the presence of measurement errors associated with data?" In the first half of the paper we propose using maximum likelihood principle to obtain an objective criterion for clustering that incorporates information about the errors associated with data. The objective criterion provides a basis for several clustering algorithms that are generalizations of the k-means algorithm and Ward's hierarchical clustering. The objective criterion has scale-invariance property so that the clustering results are independent of units of measuring data. We also provide a heuristic solution to obtain the correct number of clusters which in itself is a challenging problem. In the second half, we focus on two applications of error-based clustering: (a) regression coefficients clustering and (b) seasonality estimation in retail merchandizing, where it outperforms the k-means and Ward's hierarchical clustering.

1 Introduction

1.1 Motivation

Clustering is a fundamental and widely applied methodology in understanding structure in large data sets. General assumption is that the data is provided with no measurement error. However, in certain applications such as clustering of regression coefficients, and seasonality estimation in retail merchandizing (we elaborate on these applications in Sections 5 and 6, respectively), data to be clustered is not directly observable and a statistical method generates the data. For example, if one wishes to cluster various geographical regions based on household income and expenditure, the data for geographical regions could be estimates of average household income and average expenditure. A sample average by itself is inadequate and can be misleading unless the sampling error for each region is negligible. Sampling error, which can be estimated as the standard deviation of the sample average, may be very different for different regions.

In this paper we show that these errors are an important part of data that should be used in clustering. In contrast to standard clustering methods, in a clustering method that considers error information, two points that differ significantly in their observed values might belong to the same cluster if the points have large errors whereas two points that do not differ much in their observed values might belong to different clusters if they have small errors. The above argument is illustrated in Figure 1.

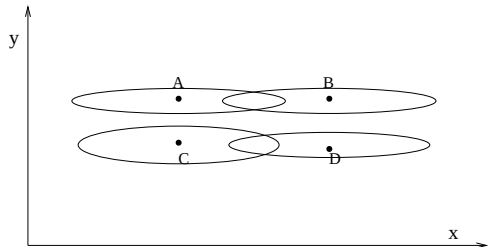

Figure 1: Data points along with errors

Four points A, B, C and D are the observed data points and the ellipses represent Gaussian errors associated with the points. A clustering method that does not consider errors will put A and C in one cluster and B and D in another, whereas a clustering method that recognizes errors

will cluster A and B together and C and D together. In this example, error-based clustering makes more sense because the x values have large error in their measurement, whereas the y value measurements are accurate and should therefore dominate the clustering decision.

The following simulation experiment further illustrates the importance of error information in clustering. Four points in one-dimension were obtained as sample means for four samples, two from one distribution, and two from another as shown in Figure 2. Our goal is to cluster the four points into two groups of two each so as to maximize the probability that each cluster represents two samples from the same distribution.

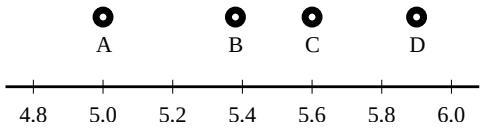

Figure 2: Four sample means in one dimension

Any clustering method would put A and B in one cluster, and C and D in another. Now we obtain additional information that the two samples on the left were samples with 10,000 points each, and the samples on the right were two samples with 100 points each. Figure 3 shows the sample means along with standard errors around them. Note that small circles on the left means larger data sets and more certainty in the sample means. Using the error information we get the following likelihood table [1] for three possible clusterings in this case.

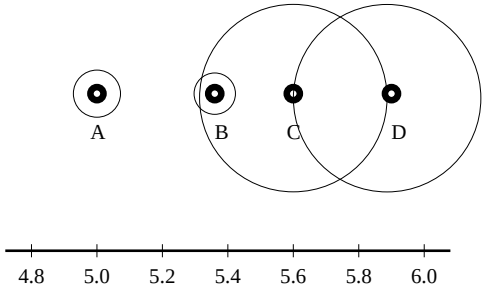

Figure 3: Four sample means along with errors

[1] The likelihood is calculated using equation (3).

Table 1: Clustering likelihood

Clustering	Likelihood
{A,B}, {C,D}	551.4
{A,C}, {B,D}	8812.2
{A,D}, {B,C}	7524.9

In the simulation study, A and C were drawn from the same distribution, and B and D were drawn from another distribution, which is the maximum likelihood clusters in Table 1. There is no way to discover the true clustering unless the error information is considered.

In practice, the structure of errors could be far more complex than the ones shown in the above examples. In this paper we model errors as coming from multivariate Gaussian distributions, which is sufficiently general and works well in practice.

1.2 Contributions Of This Paper

The fundamental question addressed in this paper is: "What is an appropriate clustering in the presence of errors associated with data ?" The traditional clustering methods, like the k-means method and the hierarchical clustering methods, are inadequate in handling such errors. We make the following contributions in this paper.

- Assuming Gaussian errors, we define an appropriate clustering model and derive an objective criterion that incorporates the information in data as well as the information in error associated with data.

- The objective criterion provides a basis for new distance functions that incorporate error information and are generalizations of the Euclidean distance function. The distance functions are used to develop clustering methods for hierarchical clustering as well as for partitioning into a specified number of clusters. The methods are generalization of the popular Ward's hierarchical clustering [17] and the k-means algorithm [10]. We also provide a heuristic method to obtain the correct number of clusters.

- We show the effectiveness of our technique in two applications: (a) regression coefficients clustering and (b) seasonality estimation in retail merchandizing. For both the applications, first we provide background on how to obtain error estimates from data and then present empirical results for various clustering methods. Although we examine these two applications in this paper, our approach is very general and can be applied in many clustering applications where measurement errors are significant.

1.3 Related Work

Approaches to clustering include statistical, machine learning, optimization and data mining perspectives. See [10, 11] for a review. In recent years probability models have been proposed as a basis for cluster analysis [1, 4, 7, 9, 15]. Methods of this type have shown promise in a number of practical applications [1, 4, 7]. In this approach, the data are viewed as coming from a mixture of probability distributions, each representing a different cluster. Our work is similar to the work by Banfield and Raftery [1], and Fraley [7] on model-based clustering. Their approach is based on maximizing the likelihood when data comes from G different multivariate Gaussian populations. We extend their model by explicitly modelling error information in the clustering technique. We also differ from their model in that instead of modelling the *populations* as multivariate Gaussian, we model the *errors* as multivariate Gaussian. This leads to a different objective function that provides a basis for various error-based clustering algorithms that are developed in this paper. We have come across only one publication [5] that explicitly considers error in data. Their method considers uniformly distributed spherical errors and is a modification of the k-means algorithm. We consider multivariate Gaussian errors and provide a formal statistical procedure to model them.

The rest of the paper is organized as follows. In Section 2, we define a maximum likelihood model for error-based clustering. In Section 3, we develop a hierarchical clustering algorithm using the above model. In Section 4, we present a generalization of the k-means algorithm for data with measurement errors. In Section 5, we describe the application of clustering in regression and present experimental results on both simulated data as well as real data. In Section 6, we provide a brief background on seasonality estimation problem and present experimental results

on real data from a major retail chain. Finally in Section 7, we present concluding remarks along with future research directions.

2 Error-based Clustering Model

We assume that n points, $x_1, ..., x_n$, are given in R^p and there is an observation error associated with each data point. We assume that the errors are Gaussian so that $x_i \sim N_p(\mu_i, \Sigma_i)$ where μ_i is a vector in R^p and Σ_i is a $p \times p$ covariance matrix. Further we assume that while μ_i is not known, Σ_i is known for all i. We wish to partition the data into G clusters, $C_1, C_2, ..., C_G$, such that all data points that have the same true mean (same μ_i) belong to the same cluster.

Let $S_l = \{i | x_i \in C_l\}$. Note that $S_i \cap S_j = \phi, i \neq j$ and $S_1 \cup S_2 \cup ... \cup S_G = \{1, 2, ..., n\}$.

Lemma 1 *Given a cluster C_l, the maximum likelihood estimate (MLE) of the mean of the cluster is given by*

$$\hat{x}_l = [\sum_{i \in S_l} \Sigma_i^{-1}]^{-1} [\sum_{i \in S_l} \Sigma_i^{-1} x_i] \tag{1}$$

and the covariance matrix of $\hat{x}$ is given by

$$\hat{\Sigma}_l = [\sum_{i \in S_l} \Sigma_i^{-1}]^{-1}. \tag{2}$$

We refer to $\hat{x}$, which is a weighted mean of a set of point, as Mahalanobis mean of the points.

Proof : Let us denote the common mean for the observations in C_l by a p-dimensional vector θ_l so that $\mu_i = \theta_l$ for $\forall i \in S_l$.

Given a clustering C along with sets S, the likelihood of the observed data is

$$\prod_{l=1}^{G} \prod_{i \in S_l} (2\Pi)^{-\frac{p}{2}} |\Sigma_i|^{-\frac{1}{2}} e^{-\frac{1}{2}(x_i - \theta_l)^t \Sigma_i^{-1} (x_i - \theta_l)} \tag{3}$$

The log likelihood is

$$\sum_{l=1}^{G} \sum_{i \in S_l} -\frac{p}{2} \ln(2\Pi) - \frac{1}{2} \ln |\Sigma_i| - \frac{1}{2}(x_i - \theta_l)^t \Sigma_i^{-1} (x_i - \theta_l)$$

$$= -\frac{1}{2} \{np \ln(2\Pi) + \sum_{i=1}^{n} \ln |\Sigma_i| + \sum_{l=1}^{G} \sum_{i \in S_l} (x_i - \theta_l)^t \Sigma_i^{-1} (x_i - \theta_l)\}$$

Since the first two terms are constant, maximizing the log likelihood is equivalent to

$$\min_{S_1,S_2,...,S_G} \sum_{l=1}^{G} \sum_{i \in S_l} (x_i - \theta_l)^t \Sigma_i^{-1} (x_i - \theta_l) \tag{4}$$

where minimization is over all possible partitions of $S = \{1, 2, ..., n\}$ into G parts. Let us denote the function being minimized in (4) by $f(S_1, S_2, ..., S_G)$. For any $l \in \{1, 2, ..., G\}$, setting $\frac{df(S_1,S_2,...,S_G)}{d\theta_l} = 0$ gives

$$\sum_{i \in S_l} \Sigma_i^{-1} (x_i - \theta_l) = 0$$

which provides the MLE of θ_l as

$$\hat{x}_l = [\sum_{i \in S_l} \Sigma_i^{-1}]^{-1} [\sum_{i \in S_l} \Sigma_i^{-1} x_i]. \tag{5}$$

$\hat{x}_l$ is a linear combination of x_i's, therefore it follows a Gaussian distribution as below

$$\hat{x}_l \sim N_p(\theta_l, \hat{\Sigma}_l). \tag{6}$$

where

$$\hat{\Sigma}_l = [\sum_{i \in S_l} \Sigma_i^{-1}]^{-1}. \tag{7}$$

■

Lemma 2 *An optimal clustering that maximizes the likelihood of the observed data is given by*

$$\arg \max_{S_1,S_2,...,S_G} \sum_{l=1}^{G} [\sum_{i \in S_l} \Sigma_i^{-1} x_i]^t [\sum_{i \in S_l} \Sigma_i^{-1}]^{-1} [\sum_{i \in S_l} \Sigma_i^{-1} x_i] \tag{8}$$

Proof : Replacing θ_l by its MLE, $\hat{x}_l = [\sum_{i \in S_l} \Sigma_i^{-1}]^{-1} [\sum_{i \in S_l} \Sigma_i^{-1} x_i]$, the optimal clustering criterion of equation (4) becomes

$$\sum_{i=1}^{n} x_i^t \Sigma_i^{-1} x_i - \sum_{l=1}^{G} [\sum_{i \in S_l} \Sigma_i^{-1} x_i]^t [\sum_{i \in S_l} \Sigma_i^{-1}]^{-1} [\sum_{i \in S_l} \Sigma_i^{-1} x_i]$$

Since the first term is constant, minimizing it over all partitions is equivalent to

$$\max_{S_1,S_2,...,S_G} \sum_{l=1}^{G} [\sum_{i \in S_l} \Sigma_i^{-1} x_i]^t [\sum_{i \in S_l} \Sigma_i^{-1}]^{-1} [\sum_{i \in S_l} \Sigma_i^{-1} x_i]$$

■

3 The *hError* Clustering Algorithm

3.1 Distance Function

The formulation in equation (8) is a combinatorial problem that cannot be solved in polynomial time, thus we use a greedy heuristic. The heuristic involves a tree hierarchy where the lowest level of the tree consists of n clusters, each corresponding to one data point. At successive levels, a pair of clusters is merged into a single cluster so that there is maximum increase in the value of the objective function in equation (8). We stop merging clusters when a desired number of clusters is obtained.

Theorem 3 *At an intermediate stage the greedy heuristic combines a pair of clusters C_i and C_j for which the following distance is minimized*

$$d_{ij} = (\hat{x}_i - \hat{x}_j)^t [\hat{\Sigma}_i + \hat{\Sigma}_j]^{-1} (\hat{x}_i - \hat{x}_j). \tag{9}$$

where $\hat{x}_i$ and $\hat{x}_j$ are the MLEs of the centers of C_i and C_j respectively, and $\hat{\Sigma}_i$ and $\hat{\Sigma}_j$ are the associated covariance matrices as defined in Lemma 1.

Proof : Let us denote the objective function in equation (8) by z. If we combine clusters C_i and C_j then the increase in the value of z is given by

$$\begin{aligned} \triangle z = [\sum_{l \in S_i \bigcup S_j} \Sigma_l^{-1} x_l]^t [\sum_{l \in S_i \bigcup S_j} \Sigma_l^{-1}]^{-1} [\sum_{l \in S_i \bigcup S_j} \Sigma_l^{-1} x_l] - [\sum_{l \in S_i} \Sigma_l^{-1} x_l]^t [\sum_{l \in S_i} \Sigma_l^{-1}]^{-1} [\sum_{l \in S_i} \Sigma_l^{-1} x_l] \\ - [\sum_{l \in S_j} \Sigma_l^{-1} x_l]^t [\sum_{l \in S_j} \Sigma_l^{-1}]^{-1} [\sum_{l \in S_j} \Sigma_l^{-1} x_l] \end{aligned} \tag{10}$$

Let us define the following for the ease of handling notations

$$y_u = \sum_{l \in S_u} \Sigma_l^{-1} x_l \qquad u = i, j. \tag{11}$$

$$T_u = \sum_{l \in S_u} \Sigma_l^{-1} \qquad u = i, j. \tag{12}$$

Note that $T_u^{-1} y_u = \hat{x}_u$ and $T_u^{-1} = \hat{\Sigma}_u$ for $u = i, j$. Also note that T_i and T_j are symmetric covariance matrices. Using the above notation, we get the following

$$
\begin{aligned}
-\triangle z &= -(y_i + y_j)^t (T_i + T_j)^{-1} (y_i + y_j) + y_i^t T_i^{-1} y_i + y_j^t T_j^{-1} y_j && (13)\\
&= y_i^t [T_i^{-1} - (T_i + T_j)^{-1}] y_i + y_j^t [T_j^{-1} - (T_i + T_j)^{-1}] y_j - 2 y_i^t (T_i + T_j)^{-1} y_j && (14)\\
&= y_i^t [(T_i + T_j)^{-1} T_j T_i^{-1}] y_i + y_j^t [T_j^{-1} T_i (T_i + T_j)^{-1}] y_j - 2 y_i^t (T_i + T_j)^{-1} y_j && (15)\\
&= (T_i^{-1} y_i - T_j^{-1} y_j)^t [T_i (T_i + T_j)^{-1} T_j] (T_i^{-1} y_i - T_j^{-1} y_j) && (16)\\
&= (T_i^{-1} y_i - T_j^{-1} y_j)^t [T_i^{-1} + T_j^{-1}]^{-1} (T_i^{-1} y_i - T_j^{-1} y_j) && (17)\\
&= (\hat{x}_i - \hat{x}_j)^t [\hat{\Sigma}_i + \hat{\Sigma}_j]^{-1} (\hat{x}_i - \hat{x}_j) && (18)
\end{aligned}
$$

Equations (15), (16) and (17) can be derived using simple matrix algebra. Hence maximizing $\triangle z$ is same as minimizing the distance, $d_{ij} = (\hat{x}_i - \hat{x}_j)^t [\hat{\Sigma}_i + \hat{\Sigma}_j]^{-1} (\hat{x}_i - \hat{x}_j)$, among all possible cluster pairs C_i and C_j. ■

This distance function satisfies the standard properties for a dissimilarity measure, namely.

$$
\begin{aligned}
dist(x, y) &= dist(y, x)\\
dist(x, y) &\geq 0\\
dist(x, x) &= 0\\
dist(x, y) = 0 &\Leftrightarrow x = y
\end{aligned}
$$

An interesting property of the proposed distance function is that it is independent of scale. When we change units of measurement of data, the observed values x's and corresponding errors Σ's are multiplied by the same factor. Therefore, d_{ij} is unit-free and scale invariant.

The hierarchical merging of clusters using the above distance function leads to the *hError* algorithm. The algorithm turns out to be a generalization of Ward's method for hierarchical clustering [17]. When $\Sigma_i = \sigma^2 I$ for all i, the method specializes to Ward's method.

3.2 Number Of Clusters

In the *hError* algorithm, two clusters are combined when we believe that they have the same true mean. Consider the hypothesis $H_0 : \theta_i = \theta_j$, i.e., the true means of clusters C_i and C_j are

the same. In other words, we combine C_i and C_j if H_0 is true. For a fixed i, j it is easy to show that the statistic

$$d_{ij} = (\hat{x}_i - \hat{x}_j)^t [\hat{\Sigma}_i + \hat{\Sigma}_j]^{-1} (\hat{x}_i - \hat{x}_j)$$

follows a Chi-Square distribution with p degrees of freedom [14]. However, the minimum d_{ij} over all i, j pairs does not follow a Chi-Square distribution. Nevertheless, we heuristically use the Chi-square distribution in the same spirit that the F-distribution is used in step-wise regression [6]. If we denote the cumulative distribution function of a Chi-Square distribution with p degrees of freedom at point t by $\chi_p(t)$, then $1 - \chi_p(d_{ij})$ gives the p-value for accepting the hypothesis. At 95% confidence, we can stop merging clusters when minimum d_{ij} is greater than $\chi_p^{-1}(0.95)$.

The clustering algorithm is formally described below.

3.3 *hError* Algorithm

Algorithm 1: *hError*

Input: $(x_i, \Sigma_i), i = 1, 2, ...n$

Output: $Cluster(i), i = 1, 2, ..., G$.

for $i = 1$ to n **do**

  $Cluster(i) = \{i\}$

$NumClust = n$

loop

  for $1 \leq i < j \leq NumClust$ **do**

    calculate $d_{ij} = dist(Cluster(i), Cluster(j))$ using equation (9)

  $(I, J) = \arg\min_{ij} d_{ij}$

  if $d_{IJ} > \chi_p^{-1}(0.95)$ **then**

    break

  $Cluster(I) = Cluster(I) \cup Cluster(J)$

  $Cluster(J) = Cluster(NumClust)$

  $NumClust = NumClust - 1$

return $Cluster(i), i = 1, 2, ..., G$

4 The *kError* Clustering Algorithm

In this section we present an algorithm that is appropriate when we know G, the number of clusters. It turns out to be a generalization of the k-means algorithm that considers errors associated with data. Similar to k-means, *kError* is an iterative algorithm that cycles through two steps. Step 1 computes centers given a clustering. The center of each cluster is its *Mahalanobis mean*. Step 2 reassigns points to clusters. Each point, x_i, is reassigned to the cluster, C_l, whose center, c_l, is the closest to x_i according to the following formula.

$$l = \arg\min_m d_{im},$$

where

$$d_{im} = (x_i - c_m)^t \Sigma_i^{-1} (x_i - c_m) \tag{19}$$

The algorithm is formally described here.

Algorithm 2: *kError*

Input: $(x_i, \Sigma_i), i = 1, 2, ...n$

G = number of clusters.

Output: $Cluster(i), i = 1, 2, ..., G$.

Find initial clusters randomly

Step 1:

for $i = 1$ to G **do**

c_i = *Mahalanobis mean* of $Cluster(i)$

Step 2:

for $i = 1$ to n **do**

for $m = 1$ to G **do**

calculate d_{im} using equation (19)

$l = \arg\min_m d_{im}$

Reassign x_i to $Cluster(l)$.

if Clusters change **then**

Go to Step 1.

return $Cluster(i), i = 1, 2, ..., G$

It is easy to show that the objective function of equation (8) improves in both steps of the algorithm in each iteration. Finite convergence of *kError* follows.

5 Application In Regression

We consider the problem of clustering the points in a regression. The standard multiple linear regression model is [6]

$$Y = X\beta + \epsilon$$

where Y is a vector of n observations, X is a known $n \times p$ matrix of n p-dimensional vectors, β is a vector of p unknown parameters and ϵ is a vector of n zero-mean independent random variables with variance σ^2. Then the usual least squares estimator for β is given by

$$b = (X'X)^{-1}X'Y \tag{20}$$

The covariance error matrix associated with b is

$$\Sigma = \sigma^2(X'X)^{-1}. \tag{21}$$

Here σ^2 is unknown but can be approximated as below.

$$\sigma^2 \sim \frac{1}{n-p}(Y'Y - Y'X(X'X)^{-1}X'Y)$$

The standard linear regression model assumes that β is same for all data points. On a reasonably complex domain, it may not be true that all data points have the same regression coefficients. Consider a simple *humidity example*.

$$Humidity = \begin{cases} 10 + 5 * temp + \epsilon & \text{in winter} \\ 20 + 3 * temp + \epsilon & \text{in summer} \end{cases}$$

Here β is different during winter and summer. In this case the least squares estimator (equation (20)) on one year's data would produce a wrong answer. On such complex domains, it would be best to partition the data points into clusters so that points within the same cluster have the same β. Then a separate regression estimator could be obtained for each cluster. [3] proposed a

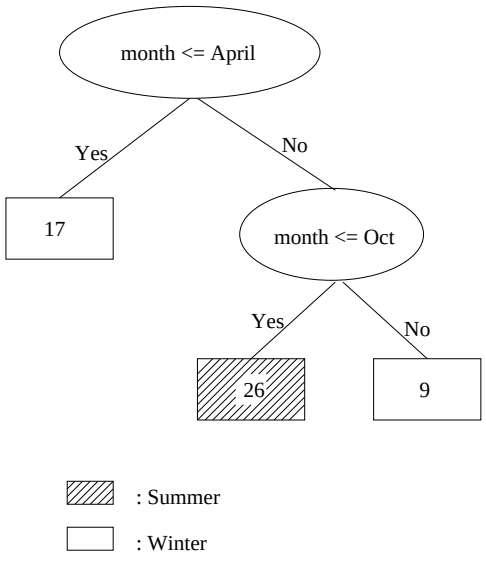

Figure 4: Regression Tree for Humidity Example

recursive partitioning of data into a regression tree such that data at each leaf node of the tree is likely to have common β. Figure 4 shows a regression tree for the humidity example.

The number inside each leaf node is the number of data points at the node. Separate regression estimator is obtained at each leaf node using equation (20). When an unseen example comes, first it is assigned to one of the leaf nodes using the test criteria at internal nodes, and then its value is predicted using the regression estimator at the leaf node.

In the humidity example, if the data is available once every week, then the rightmost leaf has only 9 data points. A regression estimate using only 9 data points would have large error. At the same time we notice that there are two leaves that correspond to winter. If we merge the two leaves the combined data from two leaves would produce better estimate. In general regression trees have many leaves, some of them might have common β. If we can cluster data from leaves that have same β, the estimator for a cluster of leaves would have smaller error. We propose using error-based clustering to form cluster of leaves based on their least squares regression estimates. Empirically we show that error-based clustering performs better than the k-means and Ward's hierarchical clustering in this application.

5.1 Simulation Data Study

5.1.1 Data Generation

[3] reported regression tree results for an artificial data set affected by noise. In this study there are ten variables, $X_1, ..., X_{10}$; X_1 takes values -1 or 1 and the others uniformly distributed continuous values from interval $[-1, 1]$. The generating function for the dependent variable, Y, is

$$Y = \begin{cases} 3 + 3X_2 + 2X_3 + X_4 + \epsilon & \text{if } X_1 = 1 \\ -3 + 3X_5 + 2X_6 + X_7 + \epsilon & \text{if } X_1 = -1. \end{cases}$$

where ϵ is a zero-mean Gaussian random variable with variance 2, representing noise. From 500 training cases, [3] constructed a regression tree with 13 leaves. Using equations (20) and (21), respectively, we calculate regression coefficient estimate, b_i, and associated error matrix, Σ_i, for each leaf, $i = 1, ..., 13$. This constitutes our training data that would be used for clustering.

5.1.2 Clustering Results

It is important to note that the clustering is done on regression estimates, so the input to a clustering algorithm consists of b_i's and Σ_i's. Regression estimates are recomputed for each cluster of data, which are used for predicting value of an unseen example. First we studied the effect of standard clustering methods, the k-means and Ward's hierarchical clustering, that do not use error information. We found that these clustering methods result in substantial misclassification (number of data points that are assigned to a wrong cluster). Using *hError* and *kError* the number of misclassifications were significantly smaller and therefore the regression coefficient estimates were more accurate.

We conducted validation experiment with out-of-sample test data that is also generated using the method described in section 5.1.1. Training data is used to learn the regression function and then it is validated on the test data. We repeated this experiment ten times. Table 2 gives average MSE (Mean Square Error) with different clustering techniques. We also show the results when no clustering is used (separate regression is used for data at each leaf node, i.e., same as the method in [3]), and when a single regression is used on entire data. Note that if clusters

were correct and all regression coefficients were known, the expected MSE would be 2.0, which is the variance of ϵ. The third column shows the percentage improvement against using a single regression on entire data.

Table 2: Clustering result

Clustering Method	Average MSE	% improvement
hError	3.61	27
kError	3.55	28
k-means	5.54	-12
Ward	5.09	-2
No Clustering	4.59	7
Single regression on entire data	4.93	

We see that *hError* and *kError* improve the results while other clustering methods make it worse. In this experiment, using the technique described in section 3.2, *hError* was able to find the right number of clusters in every run of the experiment.

5.2 Real Data Study

In this section we apply the error-based clustering to cluster regression coefficients on Boston Housing data available from the UCI Machine Learning Repository [2]. The data reports the median value of owner-occupied homes in 506 U.S. census tracts in the Boston area, together with several variables which might help to explain the variation in median value across tracts. After removing outliers and correlated attributes we have 490 data points and four explanatory variables (average number of rooms, pupil/teacher ratio, % of lower-income status population, and average distance from five major business districts). We conducted 10-fold cross validation experiment on the data by randomly splitting the data into training and test data. Regression tree method is used to divide the training data in several subsets. Then various clustering methods are used to combine subsets of data that have similar regression coefficients. The results are

validated on the test data. Table 3 reports average MSE and percentage improvement in ten repetitions of this experiment.

Table 3: Clustering result

Clustering Method	Average MSE	% improvement
hError	13.81	23
kError	13.59	24
k-means	23.98	-34
Ward	22.61	-26
No Clustering	16.50	8
Single regression on entire data	17.94	

We get some improvement in MSE by dividing training data into subsets using regression tree but clustering of these subsets using error-based clustering improves the result further. Note that clustering methods that do not consider error information performed worse than using a single regression on entire training data.

6 Application In Seasonality Estimation In Retail Merchandize

6.1 Seasonality Background [13]

In retail merchandizing it is very important to understand the seasonal behavior in the sales of different items to correctly forecast demand and make appropriate business decisions. We model the seasonality estimation problem as a clustering problem in the presence of errors and present the experimental results when applied to point-of-sale retail data. We were able to discover meaningful clusters of seasonality whereas classical methods which do not take account of errors did not obtain good clusters. In the rest of this subsection we provide brief background on

seasonality in retail industry. We also provide how to obtain initial seasonality estimates and associated error matrices that would be input to an error-based clustering algorithm.

Seasonality is defined as the normalized underlying demand of a group of similar merchandize as a function of time of the year *after taking into account other factors that impact sales such as discounts, inventory, promotions and random effects.* Seasonality is a weekly numeric index of seasonal buying behavior that is consistent from year to year. For example, a Christmas item will have high seasonality indices during the month of December, whereas shorts will have consistently high seasonality indices during summer and low indices during winter. In a retail merchandize, there are many different possible seasonal patterns. Practical concerns regarding logistic complexity require that we handle no more than a few (5-15) seasonal patterns. Therefore, our goal is to identify a small set of seasonal patterns that model the items sold by the retailer and relate each item to one of these seasonal patterns.

Considerable work has been done on how to account for the effect of price, promotions inventory and random effects[12, 16]. In our retail application, weekly sales of an item i are modelled as products of several factors that affect sales as described in equation (22).

$$Sale_{it} = f_I(I_{it}) * f_P(P_{it}) * f_Q(Q_{it}) * f_R(R_{it}) * PLC_i(t - t_0^i) * Seas_{it} \tag{22}$$

Here, I_{it}, P_{it}, Q_{it} and R_{it} are the quantitative measures of inventory, price, promotion and random effect, respectively, for an item i during week t. f_I, f_P, f_Q and f_R model the impact of inventory, price, promotion and random effect on sales, respectively. PLC is the Product Life Cycle coefficient which is defined as the sale of an item in the absence of seasonality as well as the factors discussed above. The shape and duration of the PLC curve depends on the nature of the item. For example, a fashion item will sell out very fast compared to a non-fashion item as shown in Figure 5. $Sale_{it}, Seas_{it}$ and $PLC_i(t - t_0^i)$ are the sale value, seasonality coefficient and PLC coefficient of item i during week t where t_0^i is the week when the item is introduced. For convenience we define the PLC value to be zero during weeks before the item is introduced and after it is removed. Seasonality coefficients are relative. To compare seasonality coefficients of different items on the same scale, we assume that sum of all seasonality coefficients for an item over a year is constant, say equal to the total number of weeks, which is 52 in this case.

In this paper we assume that our data has been pre-processed by using equation (22) to remove

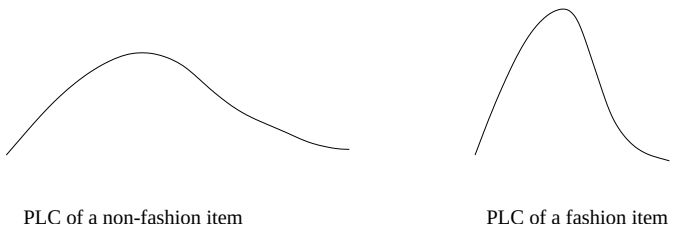

Figure 5: PLCs for non-fashion and fashion items

the effects of all these non-seasonal factors. We also assume that the data has been normalized to enable comparison of sales of different items on the same scale. After pre-processing and normalization, the adjusted sale of an item, $Sale_{it}$, is determined by the PLC and seasonality as described below.

$$Sale_{it} = PLC_i(t - t_0^i) * Seas_{it}. \tag{23}$$

Since adjusted sales of an item is the product of its PLC and seasonality, it is not possible to determine seasonality just by looking at the sale values of the item. The fact that items having the same seasonality pattern might have different PLCs complicates the analysis.

Initially, based on domain knowledge from merchants we group items that are believed to follow similar seasonality over the entire year. For example, one could group together a specific set of items that are known to be selling during Christmas, all items that are known to be selling during summer and not during winter, etc. The idea is to get a set of items following similar seasonality that are introduced and removed at different points of time during the year. This set, say S, consists of items having a variety of PLCs differing in their shape and time duration. If we take the weekly average of all PLCs in S then we would have a somewhat flat curve as shown in Figure 6. This implies that weekly average of PLCs for all items in S can be assumed to be constant as shown in Theorem 4.

Figure 6: Averaging effect on a set of uniformly distributed PLCs

Theorem 4 *For a large number of PLCs that have their introduction dates uniformly spread over different weeks of year, the weekly average of PLCs is approximately constant, i.e.,*

$$\frac{1}{|S|}\sum_{i\in S} PLC_i(t-t_0^i) \approx c \qquad \forall t = 1, ..., 52 \tag{24}$$

Proof : Let us consider a given week, say week t. Since only those PLCs that have starting time between week $t-51$ and week t will contribute to the weekly average for week t, we consider only those PLCs that have t_0^i between week $t-51$ and week t. Let p_l be the probability of t_0^i being l. Because of equally likely starting times, $p_l = \frac{1}{52}$ for $l = t-51, t-50, ..., t$.

$$\begin{aligned} E(\frac{1}{|S|}\sum_{i\in S} PLC_i(t-t_0^i)) &= \frac{1}{|S|}\sum_{i\in S} E(PLC_i(t-t_0^i)) \\ &= \frac{1}{|S|}\sum_{i\in S}\sum_{l=t-51}^{t} p_l * PLC_i(t-l) \\ &= \frac{1}{52*|S|}\sum_{i\in S}\sum_{l=0}^{51} PLC_i(l) \\ &= c \end{aligned}$$

where c is a constant that does not depend on t. The variance of $\frac{1}{|S|}\sum_{i\in S} PLC_i(t-t_0^i)$ is inversely proportional to $|S|$ as in equation (28). If $|S|$ is large, the variance will be small and the weekly observed values of $\frac{1}{|S|}\sum_{i\in S} PLC_i(t-t_0^i)$ will be approximately constant and hence the result. ■

If we take the average of weekly sales of all items in S then it would nullify the effect of PLCs as suggested by equations 25-27. Let $Sale_t$ be the average sale during week t for items in S, then

$$Sale_t = \frac{1}{|S|}\sum_{i\in S} Sale_{it} = \frac{1}{|S|}\sum_{i\in S} Seas_{it} * PLC_i(t-t_0^i). \tag{25}$$

Since all items in S are assumed to have the same seasonality, $Seas_{it}$ is the same for all items in S, say equal to $Seas_t$, i.e.,

$$Seas_{it} = Seas_t \quad \forall i \in S, \quad t = 1, 2, .., 52. \tag{26}$$

Therefore,

$$Sale_t = Seas_t * \frac{1}{|S|}\sum_{i\in S} PLC_i(t-t_0^i) \approx Seas_t * c \quad t = 1, ..., 52. \tag{27}$$

The last equality follows from Theorem 4. Thus seasonality values, $Seas_t$, can be estimated by appropriate scaling of weekly sales average, $Sale_t$.

The average values obtained above will have errors associated with them. An estimate of the variance in $Sale_t$ is given by the following equation.

$$\sigma_t^2 = \frac{1}{|S|} \sum_{i \in S} (Sale_{it} - Sale_t)^2 \tag{28}$$

The above procedure provides us with a large number of seasonal pattern estimates, one for each set S, along with estimate of associated errors. Note that each seasonality pattern estimate is a 52×1 vector. The covariance error matrix associated with each seasonality pattern is a 52×52 diagonal matrix with diagonal entries of $\sigma_t^2, t = 1, 2, ..., 52$. The goal is to form clusters of these seasonal patterns based on their estimated values and errors. Each cluster of seasonal patterns is finally used to estimate seasonality of the cluster. This estimate will have smaller error than the estimate of each seasonal pattern obtained above.

6.2 Retailer Data Study

In order to investigate the usefulness of our technique in practice, we carried out comparative analysis on real data from a major retail chain. A retail merchandize is divided into several departments (for example: shoes, shirts, etc.) which are further classified into several classes (for example: men's winter shoes, formal shirts, etc.). Each class has a varying number of items for which sales data is available. For this experiment we considered only those classes that have sales data for at least 20 items. The data used consisted of two years of sales data. One year of data was used to estimate seasonalities. Using these estimated seasonalities, we forecast sales for the next year and compare it against the actual sales data available for the next year. We considered 6 different departments (greeting cards, books, music and video, toys, automotive, and sporting goods). Each department has 4-15 classes and we used data from a total of 45 classes across all 6 departments. First we estimated seasonalities and associated errors for each class based on the method described in section 6.1. Having estimated seasonalities, we applied Algorithms *hError* and *kError* to reconstruct seasonalities for each class. Using these new seasonality estimates, we predicted sales for the items in the books department. We chose the books department because

the effects such as price, promotions and inventory were small for this department, thereby, weekly change in sales for the books department was mainly because of seasonality. We assessed the quality of forecast by calculating average *Forecast Error*, which is the ratio of total difference between actual sale and forecast sale to total actual sale, as defined below.

$$Forecast\ Error = \frac{\sum_t |Actual\ Sale_t - Forecast\ Sale_t|}{\sum_t Actual\ Sale_t} \tag{29}$$

We compared our result against k-means and Ward's method. We also compared our forecast when no clustering was used, i.e., when the forecast was based on the seasonality estimate for each class using average of weekly sales data as described in section 6.1. We found that forecasts using *hError* and *kError* were substantially better than forecasts using k-means or Ward's method or forecasts without using clustering. Table 4 compares average *Forecast Error* in these five situations for 17 different items in the books department.

Table 4: Average Forecast Error

Clustering Method	Average Forecast Error %
hError	18.7
kError	18.3
Ward's	23.9
k-means	24.2
No clustering	31.5

Figure 7 shows graphs comparing these forecasts for one item in the books department. Graphs of *kError* and k-means are similar to that of *hError* and Ward's method respectively and therefore removed from the figure. This item was sold for a total of 33 weeks during January through September 1997. The price for the item was constant during this period and there was no promotion on this item, therefore we ignored all external factors and made our forecast using only PLC and seasonality coefficients. Seasonality of the class that contains this item is estimated using past year's sales data of all the items in the class. The first 18 weeks of sales data of this item is used to estimate the PLC. PLC is estimated by simple curve fitting from a

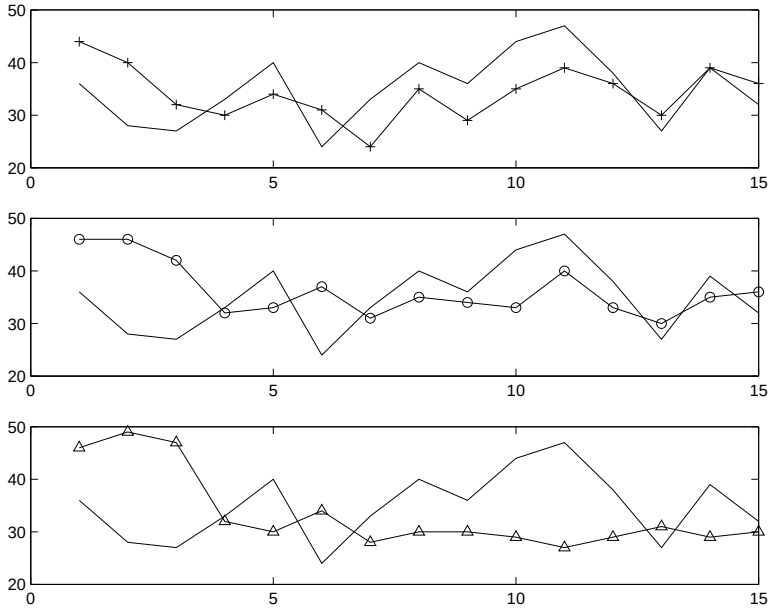

Figure 7: Sales forecast against actual sales

——— : actual sales

—+— : forecast using *hError*

—o—- : forecast using Ward's method

—△— : forecast without clustering

set of predefined PLCs. Using the seasonality estimates from past year's data and PLC estimate from the first 18 weeks of data, we forecast sales for the remaining 15 weeks. The graphs show that forecast using *hError* is significantly better than the others.

In Figure 7, we observe that seasonality estimate without clustering failed to capture the sales pattern. Standard clustering succeeded in making a better forecast but error-based clustering was even better. The reason is that the books department has 5 classes. Because very few items are used to estimate seasonality for each class, seasonality estimate for each class has large errors and therefore the forecast based on this seasonality (without clustering) does not match actual sales. On close inspection of the data we found that there are two groups of 3 and 2 classes having similar seasonalities. Clustering identifies the right clusters of 3 and 2 seasonalities. The combined seasonality of each cluster has higher accuracy because more items are used to estimate it. Error-based clustering does better than standard clustering because it gives more weight to

seasonality with smaller errors obtained by using larger number of items.

We restricted our forecast analysis to only a small section of books department that had very small fluctuation in price over their selling period. This helped us eliminate effects due to price or promotion. With the help of appropriate pricing models we could have analyzed the remaining items as well.

7 Summary And Future Research

The traditional clustering methods are inadequate when different data points have very different errors. In this paper we have developed a clustering method that incorporates information about error associated with data. We developed a new objective function which is based on Gaussian distribution of errors. The objective function provides a basis for *hError* and *kError* clustering algorithms that are generalization of Ward's hierarchical clustering and k-means algorithms, respectively. Finally, we demonstrated the utility of our method in getting good estimate of regression coefficients both on simulated data as well as on real data. We also demonstrated its utility in obtaining good estimate of seasonality in retail merchandizing. *kError* is fast and performs slightly better than *hError*, but *hError* has the advantage of automatically finding the right number of clusters which in itself is a challenging problem. Finally, we feel that errors contain very useful information about data and a clustering method using the information contained in errors is an important conceptual step in the field of cluster analysis.

In our formulation we assumed that we have estimate of error matrix for each data point. Sometimes only partial information about error matrices might be available, e.g., only the diagonal entries are available (as in the case of seasonality estimation). The next step in this research would be to explore how error-based clustering performs when we have only restricted information about error matrices. Another future research direction would be to analyze theoretically the effect of error-based clustering on regression application. We have already developed a theoretical model that justifies error-based clustering of regression coefficients. We will report the results at a later stage. We are also exploring other applications where error-based clustering would be useful.

Acknowledgement

The work described in this paper was supported by the e-business Center at MIT Sloan, and ProfitLogic Inc.

References

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

[2] C.L. Blake, C.J. Merz. UCI repository of machine learning databases, 1998. *http://www.ics.uci.edu/ mlearn/MLRepository.html*.

[3] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone. Classification and Regression Trees. *Wadsworth Int. Group, Belmon, California* 1984.

[4] G. Celeux, G. Govaert. Gaussian parsimonious clustering models. *Pattern Recognition*, Volume 28, 781-793, 1995.

[5] B. B. Chaudhuri, P. R. Bhowmik. An approach of clustering data with noisy or imprecise feature measurement. *Pattern Recognition Letters*, Volume 19, 1307-1317, 1998.

[6] N.R. Draper. Applied Regression Analysis. *Wiley*, 1998.

[7] C. Fraley. Algorithms for model-based Gaussian hierarchical clustering. *SIAM Journal on Scientific Computing*, Volume 20(1), 270-281, 1998.

[8] H. P. Friedman, J. Rubin. On some invariant criteria for Grouping data. *Journal of the American Statistical Association*, Volume 62, 1159-1178 1967.

[9] Scott Gafney, Padhraic Smyth. Trajectory clustering with mixtures of regression models. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1999.

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

[11] A. K. Jain, M. N. Murty, P. J. Flynn. Data Clustering: A Review. *ACM Computing Surveys*, Volume 31, No. 3, 264-323, 1999.

[12] Praveen K. Kopalle, Carl F. Mela, Lawrence Marsh. The dynamic effect of discounting on sales: Empirical analysis and normative pricing implications. *Marketing Science* , 317-332, 1999.

[13] M. Kumar, N. R. Patel, J. Woo. Clustering seasonality patterns in the presence of errors. *The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 557-563, 2002.

[14] John A. Rice. Mathematical Statistics and Data Analysis. Second Edition. *Duxbury Press*.

[15] A. J. Scott, M. J. Symons. Clustering methods based on likelihood ratio criteria. *Biometrics*, Volume 27, 387-97, 1971.

[16] Jorge M. Silva-Risso, Randolph E. Bucklin, Donald G. Morrison. A decision support system for planning manufacturers' sales promotion calendars. *Marketing Science*, 274-300, 1999.

[17] J. H. Ward Jr. Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association* Volume 58, Issue 301, 236-244, 1963.