An EM-based Ensemble Learning Algorithm on Piecewise Surface Regression Problem

Juan Luo, Alexander Brodsky and Yuan Li

Department of Computer Science
George Mason University
Fairfax, VA 22030, U.S.A
jluo2@gmu.edu, brodsky@gmu.edu & ylif@gmu.edu

ABSTRACT

A multi-step Expectation-Maximization based (EM-based) algorithm is proposed to solve the piecewise surface regression problem which has typical applications in market segmentation research, identification of consumer behavior patterns, weather patterns in meteorological research, and so on. The multiple steps involved are local regression on each data point of the training data set and a small set of its closest neighbors, clustering on the feature vector space formed from the local regression, regression learning for each individual surface, and classification to determine the boundaries for each individual surface. An EM-based iteration process is introduced in the regression learning phase to improve the learning outcome. In this phase, ensemble learning plays an important role in the reassignment of the cluster index for each data point. The reassignment of cluster index is determined by the majority voting of predictive error of sub-models, the distance between the data point and regressed hyperplane, and the distance between the data point and centroid of each clustered surface. Classification is performed at the end to determine the boundaries for each individual surface. Clustering quality validity techniques are applied to the scenario in which the number of surfaces for the input domain is not known in advance. A set of experiments based on both artificial generated and benchmark data source are conducted to compare the proposed algorithm and widely-used regression learning packages to show that the proposed algorithm outperforms those packages in terms of root mean squared errors, especially after ensemble learning has been applied.

Keywords: piecewise surface regression, EM-based, cluster index reassignment, ensemble learning.

2000 Mathematics Subject Classification: 91C20, 62JXX.

1 Introduction

The solution to any learning problem involves the reconstruction of an unknown function $f : X \to Y$ from a finite set S of sample of f (training set), possibly affected by noise. Different approaches are usually adopted when the range of Y only contains a reduced number of disjoint elements, typically without a specific ordering among them (classification problems) or when Y is an interval of the real axis with the usual topology (regression problems). The real world

application areas can be the determination of market segments in a marketing research study, the identification of distinct spending patterns in studies of consumer behavior, the detection of different types (clusters) of documents in text mining or weather patterns in meteorological research.

Regression analysis attempts to build a model based on the relationship of several independent variables and a dependent variable (Draper and Smith, 1998). Let $x_1, ..., x_n$, be independent variables, and y, be dependent variable, both range over the set of $\mathbb{R}$. The latter is a random variable defined over the underlying distribution of sample tuples in I_n = $\mathbb{R} \times \mathbb{R} \times ... \times \mathbb{R}$. Suppose the learning set contains m tuples. Let us denote such a tuple as $x_h = (x_{h1}, ..., x_{hn})$ for $h = 1, ..., m$. The collection of data, $c = (x_h, y_h)$ for $h = 1, \ldots, m$, represent the available training data to estimate the values of the random variable $y = f(x_h, \beta) + N$ for $h = 1, \ldots, m$, where β represents a set of coefficients and N is a random noise. We assume that N is distributed as a Gaussian with 0 mean and variance σ such that: $E(y) = E(f(x_h, \beta) + N) = E(f(x_h, \beta)) = f(x_h, \beta)$, where E is the expected value. The standard least squares method is used to find coefficients β of f that minimize σ.

Application can be found, which lie on the borderline between classification and regression; these occur when the input space X can be subdivided into disjoint regions X_i characterized by different behaviors of the function f to be reconstructed. One of the simplest situation of such kind is piecewise surface regression: in this case X is a polyhedron in the n-dimensional space $\mathbb{R}^n$ and $\{X\}_{i=1}^k$ is a polyhedral partition of X, i.e. $X_i \cap X_j = \emptyset$ for every $i, j = 1, \ldots, k$ and $\bigcup_{i=1}^k X_i = X$. The target of a piecewise surface regression problem is to reconstruct an unknown function $f : X \to \mathbb{R}$ having a linear behavior in each region X_i

$$f^*(x) = f_i(x, \beta_i) \quad \text{if} \quad x \in X_i \tag{1.1}$$

when only a training set D containing m samples (x_h, y_h), $h = 1, \ldots, m$, is available. The output y_h gives a noisy evaluation of $f(x_h)$, being $x_h \in X$; the region X_i to which x_h belongs is not given in advance. The parameters set $\beta_1, \beta_2, \ldots, \beta_i$ for $i = 1, 2, \ldots, k$, characterizes the function set f_i and their estimate is a target of the piecewise surface regression problem. The regions X_i are polyhedral, i.e., they are defined by a set of l_i linear inequalities, which can be written in the following form:

$$A_i \begin{pmatrix} 1 \\ x \end{pmatrix} \geq 0 \tag{1.2}$$

where A_i is a matrix with l_i rows and $n + 1$ columns and their estimate is another target of learning process for every $i = 1, 2, \ldots, k$. According to 1.1 and 1.2, the target of the learning problem is actually two-fold: to generate both the regions X_i and the parameter set β_i for the unknown function set f_i, utilizing the information contained in the training set.

There has been work on the learning of piecewise surface regression problem. The quality of a piecewise regression algorithm heavily depends on the accuracy of the partition of input space. In the Local Linear Map (LLM) of (Ritter, 1991) and combinatorial regression learning of (Luo and Brodsky, 2010), only information about the input space is used for partition of the data. However, when data points can not be separated within the input space but a meaningful separation can still be obtained by considering the target variable together. Different

approaches have been proposed to solve the problem of partitioning data by incorporating the target variable. Some of them focuses on solving problems in two dimensional spaces as in (Cherkassky, 1991) and (Luo and Brodsky, 2011). In (Hathaway and Bezdek, 1993), the data is clustered based on the local model parameters learned from a set of small size local data set. In (Ferrari-Trecate and Muselli, 2002), a connectionist model, i.e., a three layer neural network is constructed to learn the parameter set in the regression problem. While neural networks show high accuracy on training data set, they typically do not perform well on testing data set which is caused by over-fitting problem. An approach is outlined in (McGee, 1963) that uses hierarchical clustering to cluster data points into segments that represent the individual regimes of the piecewise function and perform standard linear regression on them.

An EM-based algorithm (EMPRR) has been proposed proposed in (Arumugam and Scott, 2004) to solve general piecewise surface regression model. It is developed based on Levenberg-Marquardt (LM) (Levenberg, 1944) and (Marquardt, 1963) algorithm, an iterative technique which helps in locating the discrepancy between a given model and the corresponding data and has become a standard technique for nonlinear least-square problems. Since EMPRR is a nonlinear optimization technique, an absolute bound on the time complexity is not feasible since there is no way of knowing exactly how long it takes for the method to converge. At the same time, an initial guess need to be made for every unknown parameter in the piecewise surface regression model as inputs. This brings more uncertainty into the optimization process and increases the chance of getting trapped in a local minimum. Another EM-like piecewise linear regression algorithm has been proposed in (Nusser, Otte and Hauptmann, 2008) as well. It describes an EM-like piecewise linear regression algorithm that uses information about the target variable to determine a meaningful partitioning of the input space. The main goal of this approach is to incorporate information about the target variable in the prototype selection process of a piecewise regression approach. The drawback of this approach lies in the fact that it randomly assigns an initial cluster index for each data point in the data set. It makes the learning process hard and ineffective to converge and at the same time, the learning outcome is unpredictable.

Some algorithms which have been developed for learning decision trees are variations of the algorithms which employs a top-down, greedy search through the space of decision trees. The Classification and Regression Trees (CART) system (Breiman, Friedman, Olshen and Stone, 1984) is a tree learning technique that assigns constant values at the leaves of the tree. Consequently it can fit piecewise constant data well but fit piecewise linear data with errors. A package called M5P, which combines conventional decision trees with linear regression functions at the leaves, is developed by (Quinlan, 1992) and (Wang and Witten, 1997). These model trees are similar to piecewise linear functions. The M5P will be run as part of our experiment as a comparison approach.

Ensemble learning is a type of machine learning that studies algorithms and architectures that build collections, or ensembles, of statistical classifiers that are more accurate than a single classifier. An ensemble consists of a set of individual predictors (such as decision trees or neural network) whose predictions are combined when classifying a given example. The purpose of ensemble learning is to improve the performance of individual classifier by exploiting

knowledge derived form different sources. In (Jain, Duin and Mao, 2000), numerous reasons for combining multiple classifiers are listed: one may have different feature sets, different training sets, different classification methods or different training sessions, all resulting in a set of classifiers whose results may be combined with the hope of improved overall classification accuracy. The majority voting ensemble learning is applied in the regression learning phase of our learning algorithm.

The contribution of this paper can be summarized as: First, we propose an EM-based multi-step algorithm (EMMPSR) for the general piecewise surface regression problem described in 1.1 and 1.2, no matter what dimension the input space is and considering the target variable in the clustering process; Second, multiple steps involved are local regression, clustering, regression learning on each individual surface and classification to determine the boundaries of each surface. The clustering process is performed based on the feature vector space of input data set, instead of the data set itself. The feature vector for each data point is calculated by local regression on a small subset of data which are closest to that data point. The estimation of sub-models in 1.1 is learned by robust regression learning which can effectively detect "outliers". The estimation of boundaries for each region in 1.2 is performed by a multi-category classification algorithm; Third, an EM-based iteration process is introduced in the regression learning phase to improve the learning outcome. The clustering phase assigns a cluster index to each data point. However, the assignment may not be the correct and can be adjusted in the next iteration. A majority voting ensemble learning is applied to the decision of index reassignment; Fourth, in the scenario that the number of surfaces is not known in advance, a few clustering quality validity indexes are adopted to detect the optimal number of surfaces contained by the input data set; Finally, a set of experiments are conducted to show the performance of the proposed algorithm and the effects of ensemble learning.

The paper is organized as following. The problem definition is given and related literatures are discussed in section 1. The detailed algorithm is described in section 2. Experiment setups on both synthetic and benchmark data are discussed in section 3. Section 4 concludes the paper and points out possible future work.

2 The EM-based Multi-Step Piecewise Surface Regression Algorithm

The Expectation Maximization (EM) algorithm (Dempster, Laird and Rubin, 1977) has been adapted in our algorithm. The input space is partitioned by applying a double-fold k-means clustering algorithm, incorporating the value of target variable. After the clustering of polyhedral regions, a multi-category SVM library (Chang and Lin, 2001) is called to calculate the boundary matrix A_i (see 1.2) which represents a polyhedral region. For each polyhedral region, its surface regression model can be learned by robustfit (Huber and Ronchetti, 1981). Similar to EM algorithm, an iteration process is involved in our approach. First, the surface models are learned the resulted clusters of clustering process. Then all data points in are re-assigned to the surface model which has the best predictive performance. The surface models are further updated based on the newly created clusters of polyhedral regions. The iteration process is repeated until termination criterion has been reached. Details are described in Algorithm 1.

Algorithm 1: The EM-based Multi-step Piecewise Surface Regression Algorithm

Input: Data set D with size m, number of clusters k

Output: Surface function model f_i and boundary matrix A_i for i = 1, ..., k

(Local regression) **foreach** *h = 1, ..., m* **do**

1.1 Build the local dataset E_h containing the sample (x_h, y_h) and the pairs $(x, y) \in D$, together with the $e-1$ closest neighbors x to x_h.

1.2 Perform a linear regression to obtain the feature vector v_h of a linear unit fitting the samples in E_h.

(Clustering) Perform clustering process in the feature vector space.

2.1 Run regular k-means on feature vector space $\mathbb{R}_{n+1}$ with an assigned feature vector centroid set CV to subdivide the set of feature vectors v_h into k groups $U_i, i = 1, ..., k$.

2.2 Build a new training set D' containing m pairs (x_h, i_h) being U_{i_h} the cluster including v_h

repeat

(Regression) For every j = 1,..., k, run a linear regression on the samples $(x, y) \epsilon D$ with $x \epsilon X_i$. The parameter set β_i returned represents the i_{th} surface function f_i.

Update cluster index of each data point, further the training set D', according to the minimal predictive error among surface models f_i for i = 1, ..., k.

until *Maximum number of iterations has been reached or no cluster index is reassigned*;

Multi-category classification on training set D' to compute the boundary matrix A_i for every surface X_i.

2.1 Local Regression

As discussed in the introduction, the learning effect of piecewise regression problem depends on the accuracy of the partition of input space. Given a training set D, the intuitive way to do partitioning is classical K-means clustering (MacKay, 2003). However, k-means clustering results are sensitive to the initial centroid which are randomly picked, when clusters are of differing sizes, densities or non-globular shapes. Instead of clustering on training set, we proposes a different way of clustering, i.e. clustering on the feature (parameter) vector space of training set. The feature vector for each data point is learned by local linear regressor based on small subset of the whole training set D. It is observed that points close to one another are more likely to belong to the same region X_i than those are not. For each sample (x_h, y_h), with h = 1, ..., m, a local data set E_h is built to contain (x_h, y_h) and its $e-1$ nearest neighbors $(\hat{x}, \hat{y})$ that satisfy

$$\|x(h) - \hat{x}\|^2 \leq \|x(h) - \tilde{x}\|^2 \qquad \forall (\tilde{x}, \tilde{y}) \in D \quad \setminus \quad E_h, \tag{2.1}$$

The distance between points is calculated by $\|x_h - x\|$ where $\|.\|$ is the Euclidean norm. Note that each E_h can be labeled by the point (x_h, y_h). This way a bijective map between data points and local data sets is formed. Most sets E_h contain data points belonging to the same region X_i, while the rest, called mixed, include data points from different regions X_j. The local regression step is trying to obtain a first estimate of the parameter set β_i set which characterize the functional models f_i. Local linear regression is run on small subsets of the whole training

set D based on the fact that points x_h which are close to one another are more possible to belong to the same region X_j than those not close. The feature vector v_h (with dimension $n+1$) learned from pure local data set E_h is a good estimate of parameter β_i which represents the region function f_i, while the feature vectors learned from mixed local set lead to the wrong estimate of β_i so their number need to be kept at the lowest possible level.

The number of mixed local data set depends both on the sampling schedule of input space and choice of parameter e. As to the sampling schedule, an implicit assumption for good results from our algorithm is that the sampling is fair, i.e., the data points drawn are not all concentrated around the boundary of the sets X_i. The parameter e should be chosen well in order to obtain non-overlapping clusters of feature spaces and minimize, at the same time, the number of outliers. If the parameter e is low, the ratio between the number of mixed and non-mixed local data sets is low. However, when the noise level is not negligible, a low e produces poor estimates of the feature vectors, i.e. estimates with high variance, thus preventing a good partitioning of the feature vectors. So the value of e can not be assigned too low. On the other hand, if e is too high, a large percent of mixed local data sets (further outliers in the feature space) will be generated. In the extreme case is that when e is equal to the number of sample data set D, all local sets built are mixed and every feature vector collapse in a single hyperplane fitting all the data. In order to have a well-defined clusters, the value of the parameter e need to be tuned in experiments with cross-validation techniques.

The least-squared estimation is used to compute the feature (parameter) vector of every local data set E_h which contains data points $(x_h^1, y_h^1), (x_h^2, y_h^2), ..., (x_h^e, y_h^e)$. We can define ϕ_h and ψ_h as

$$\phi_h = \begin{bmatrix} x_h^1 & x_h^2 & ... & x_h^e \\ 1 & 1 & ... & 1 \end{bmatrix}', \quad \psi_h = \begin{bmatrix} y_h^1 & y_h^2 & ... & y_h^e \end{bmatrix}' \tag{2.2}$$

The $'$ is the transpose operator for matrix. The feature vector v_h can be computed by the formula

$$v_h = (\phi_h' \phi_h)^{-1} \phi_h' \psi_h \tag{2.3}$$

Another bijective map can be formed between feature vectors V computed for each data point and each local data set. Given the bijective map between data points and local data sets, a new bijective map between each data point and each feature vectors is formed as well.

2.2 Clustering

After the feature vector space has been generated, the next step of the algorithm is to cluster the feature vectors into k disjoint subsets U_i. Principally, any clustering algorithm can be used but the performance of classical clustering algorithms like k-means is often spoiled by poor initialization of centroid which are randomly picked, when clusters are of differing sizes, densities or non-globular shapes (MacKay, 2003). In our case, we propose a two fold k-means clustering algorithm which can decrease the misclassification rate of k-means and at the same time, the computational efficiency of K-means will still be kept. The clustering process is described in Algorithm 2.

Algorithm 2: Two-fold k-means clustering algorithm

Input: data set D with size m, feature vector set V with size m, number of clusters k

Output: Feature vector set V with cluster index assigned for every feature vector

1 Do regular k-means clustering on the data points with randomly picked initial centroid
2 For each cluster $X_1, X_2, ..., X_k$ returned by 1, calculate its mean $\bar{X}_1, \bar{X}_2, ..., \bar{X}_k$
3 For each cluster mean $\bar{X}_i$, $i = 1, ..., k$, among data points in cluster X_i, find the data point which is most close to $\bar{X}_i$ and save it to the centroid set C with size k
4 For each $c_i \in$ centroid set C, i=1,...,k, find the corresponding feature vector cv_i in the feature vector set V having the same index as centroid in the data set to form a feature vector centroid set CV.
5 Run k-means clustering on the feature vector set V with initially assigned centroid set CV to subdivide feature vector set into k groups U_i for $i = 1, ..., k$.

The difference between Algorithm 2 and the classical k-means is the initial picking of centroid. In Algorithm 2, the k-means clustering is run the first time to estimate the centroid of clusters of data point. Due to the bijective mapping between each data point and each feature vector, a corresponding feature vector can be found for each centroid resulted from the first step. This centroid set works as a much better initial input for the second k-means clustering which will be run on the feature vector space, compared to the randomly picked centriod among feature vectors. The clustering quality for the feature vector sets are obviously improved by decreasing the misclassification rate of each feature vector. Finally, by using the bijective maps between feature vectors and data points, the original data can now be classified. In fact, each data point (x_h, y_h), h=1,...,m is assigned a cluster index i_h being U_{i_h} the cluster including v_h. A new data set D' is formed as m pairs (x_h, i_h), h=1,...,m.

2.3 EM-based iteration process

The EM-based iteration process consists of two main steps: First, surface regression models f_i, i = 1,..., k for each region are determined according to the cluster assignment i_h of each data point (x_h, y_h), h=1,...,m; Second, each data point is re-assigned to one of the clusters X_i, i=1,...,k where the predictive error of the corresponding regression model is minimal. The assignment can be defined as a mapping:

$$CI(h) = i, \qquad \text{with} \quad 1 \leq h \leq m, 1 \leq i \leq k \tag{2.4}$$

which assigns the h^{th} data point to the i^{th} cluster. Each surface model is represented by function f_i, i = 1,..., k. The surface regression model of the i^{th} cluster is trained on all data points that are assigned by the mapping CI to the i^{th} cluster:

$$f_i(x_h) = y_i \quad \text{where} \quad CI(h) = i \tag{2.5}$$

2.3.1 Estimation of sub-models

To learning the surface regression models f_i, i = 1,..., k, least squares can accomplish this task. However, one of the main drawbacks of least squares lies in the sensitivity of the method to outliers (Huber and Ronchetti, 1981) that may be present due to classification errors. The robust regression techniques (Huber and Ronchetti, 1981) is less sensitive to outliers than least squares, especially when the number of outliers is a small fraction of the data points. That is why the estimation of each sub-model is solved by the robust regression learning.

2.3.2 Cluster index reassignment

The second main step of EM-based iteration process is to update cluster index of each data point. The mapping defined in 2.4 is updated according to the minimum predictive error among all sub-models f_i, i = 1,..., k

$$CI(h) = argmin_{i=1,...,k}|y_h - f_i(x_h)| \tag{2.6}$$

If any data point which is misclassified at the beginning, it is possible that its cluster index will be re-assigned by Equation 2.6. The reassignment process changes region in terms of data point, furthermore, the estimation of the sub-models and boundary matrices will be re-estimated as well. The iteration process is repeated until no more cluster reassignment occurs or the maximum number of iterations has been reached.

2.3.2.1 Majority voting ensemble learning

The minimum predictive error of sub-models is one measure to reassign the cluster index. However, another two measures can be adopted to do reassignment as well. The minimum distance between data point and centroid of each sub-model is one natural choice. Given the centroid CE_i for sub-model f_i, i = 1,...,k, the mapping defined in Equation 2.4 is updated according to the following equation

$$CI(h) = argmin_{i=1,...,k}||x - CE_i|| \tag{2.7}$$

The third measure is the distance between data point and hyperplane set $coef_i$, which represents the sub-model f_i, for i = 1,...,k. The mapping defined in Equation 2.4 is updated according to the following equation

$$CI(h) = argmin_{i=1,...,k}(\frac{x * coef_i}{sqrt||coef_i||}) \tag{2.8}$$

Any single measure may not be always right, however, the majority voting combines three measures together and leads to performance improvement.

2.4 Estimation of boundary matrices

After the surface models have been obtained, the next step is to obtain an approximation of the unknown polyhedral regions which are specified by a set of matrices A_i, i = 1,..., k in Equation 1.2. The matrices are solved by multi-category classification technique derived from the support vector machine (Chang and Lin, 2001).

2.5 Detection of the number of regions

So far we assume that the number of clusters for data set is known in advance. However for some real data sets, this is not known a priori and, in fact, there might be no definite or unique answer as to what value k should take. In other words, k is a nuisance parameter of the clustering model. Numerous techniques can be applied to determine this k value. Among them, v-fold cross validation (Hill and Lewicki, 2007) and a few clustering validity indexes (silhouette index (Rousseeuw, 1987), Davis-Bouldin index (Davies and Bouldin, 1979), Calinski-Harabasz index (Calinski and Harabasz, 1974) and Dunn index (Dunn, 1974)) are computed and compared.

2.5.1 V-fold cross-validation

The v-fold cross-validation algorithm is applied to clustering. The general idea of this method is to divide the overall sample into a number of v folds. The same type of analysis is then successively applied to the observations belonging to the v-1 folds (training sample), and the results of the analysis are applied to sample v (the sample or fold that was not used to estimate the parameters, i.e, the testing sample) to compute some index of predictive validity. The results for the v replications are averaged to yield a single measure of the stability of the respective model, i.e., the validity of the model for predicting new observations. We can apply the v-fold cross-validation method to a range of numbers of clusters in k-means, observe the resulting average distance of the observations (in the testing samples) from their cluster centers.

2.5.2 Clustering Quality Validity Indexes

Four different indexes are calculated to estimate the optimal number of clusters in benchmark data sets. The Silhouette index calculates the silhouette width for each sample, average silhouette width for each cluster and overall average silhouette width for a total data set. It uses average dissimilarities between points to identify the structure of the data and highlights possible clusters. It is suitable for estimating the first choice or the best partition. The value range of Silhouette index is between [-1, 1]. If silhouette value is close to 1, it means that sample data set is well-clustered and it was assigned to a very appropriate cluster. If silhouette value is close to -1, it means that sample data set is misclassified and is merely somewhere in between the clusters. The Davis-Bouldin index is a function of the ratio of the sum of within-cluster scatter to between-cluster separation. The ratio is small if the clusters are compact and far from each other. Consequently, Davis-Bouldin index will have a small value for a good clustering. Calinski-Harabasz index is the pseudo F statistic calculating the quotient between the intracluster average squared distance and intercluster average squared distance. The higher the Calinski-Harabasz index, the better the clustering quality. Dunn's index is based on geometrical considerations for hard clustering. This index is designed to identify sets of clusters that are compact and well separated. The main goal of this measure is to maximize the intercluster distances and minimize the intracluster distances. Therefore, the number of clusters that maximize the Dunn's index is taken as the optimal number of clusters.

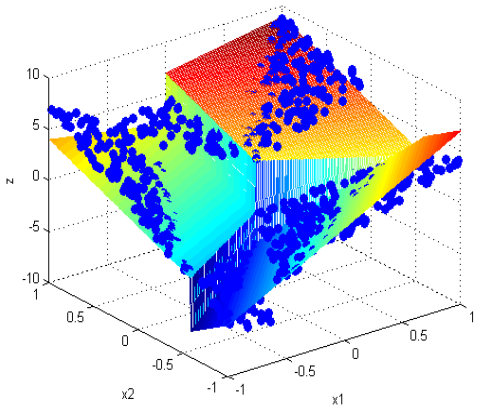

Figure 1: Synthetic Data Set Generated in Model 2

3 Experiments

To evaluate our EM-based multi-step piecewise surface regression algorithm EMMPSR, we generate synthetic high-dimensional data which is piecewise-defined. We compare the performance of EMMPSR with those of M5P (WEKA package) (Hall, Frank, Holmes, Pfahringer, Reutemann and Witten, 2009), classregtree (Matlab statistical toolbox) (MATLAB, 2010), and MultilayerPerceptron (three layer neural network) (Hall et al., 2009) on a set of experimental data set. These three packages are widely-used regression learning tools.The data set includes three synthetic data set and four benchmark data set.

3.1 Synthetic Data Sets

Three data sets are generated using four different piecewise models. Each model has linear boundaries between regions and linear functions within each region. Model 1 and model 2 each has three regions and two independent variables. Model 3 has five regions and nine independent variables with linear boundaries and linear functions as well. Data in each model are generated with additive Gaussian noise with zero mean and 0.1 variance. We generated 300 sample points for model 1, 900 data points for model 2 and 1500 data points for model 3. The second data set is generated from the following piecewise functions:

$$f(x_1,x_2)=\begin{cases} 3+4x_1+2x_2, & \text{if} \quad 0.5x_1+0.29x_2\geq 0 \quad \text{and} \quad x_2\geq 0 \\ -5-6x_1+6x_2, & \text{if} \quad 0.5x_1+0.29x_2<0 \quad \text{and} \quad 0.5x_1-0.29x_2<0 \\ -2+4x_1-2x_2, & \text{if} \quad 0.5x_1-0.29x_2\geq 0 \quad \text{and} \quad x_2<0 \end{cases} \tag{3.1}$$

This target function is depicted in Figure 1. Total 900 samples are drawn uniformly from I_2 = $[-1,1]\times[-1,1]$ and y is determined as $y=f^*(x_1,x_2)+\varepsilon$, where $\varepsilon\sim N(0,0.1)$. In this setting, the target value need to combined to determine the appropriate cluster prototypes.

The following function estimate is yielded by the EMMPSR algorithm:

$$f(x_1,x_2)=\begin{cases} 3.0067+3.9940x_1+1.9977x_2, & \text{if} \quad 0.5x_1+0.32x_2\geq 0.005 \quad \text{and} \quad x_2\geq 0 \\ -5.0217-6.0201x_1+6.0056x_2, & \text{if } 0.5x_1+0.32x_2<0.005 \text{and} 0.5x_1-0.31x_2<0.01 \\ -2.0035+3.9793x_1-2.0330x_2, & \text{if} \quad 0.5x_1-0.31x_2\geq 0.01 \quad \text{and} \quad x_2<0 \end{cases} \tag{3.2}$$

As noted, the generated model is a good approximation of the unknown function to learn in 3.1. Five-fold cross validation is adopted to evaluate the learning performance by randomly dividing the data set into 5 equal parts. Each part is held out in turn and the remaining four is trained for the learning method. The root mean squared error (RMSE) (Alpaydin, 2004) will be calculated on the unseen data. The results are summarized in Table 1.

Table 1: RMSE values for performance comparison experiments on synthetic data sets

Model	M5P	MultilayerPerceptron	Classregtree	EMMPSR
Model1	1.0925	3.0657	2.8899	0.3759
Model2	0.7599	1.8773	0.4995	0.2538
Model3	37.6910	47.8030	33.3755	30.8755

Another matrix to be compared among different methods is average number of rules generated by each model for a data set. In EMMPSR it is the number of regions, while in M5P and Classregtree it is the number of rules generated during the process of building the tree. EMMPSR only uses a fraction of the rules that are generated by M5P and Classregtree. It is obvious that EMMPSR outperforms other methods as to RMSE as well.

3.2 Benchmark Data Set

Benchmark data sets are obtained from the Repository of Regression Problems at (LIACC, 2006). This repository is actually a collection of data from other sources, however we still choose it because the data sets have been preprocessed to meet our specifications – nominal attributes and samples with missing attributes have been removed. Five-fold cross validation is adopted to evaluate the learning performance as well.

Auto MPG Data Set: The task of this data set is to predict the fuel consumption in miles per gallon (MPG) of different cars. Five attributes of the original data set are used as input dimensions acceleration', displacement', horsepower', model-year, and weight. The data set consists of 398 instances. Six instances with missing values are ignored within the experiments.

Delta Ailerons Data Set: This data set is also obtained from the task of controlling the ailerons of an F16 aircraft, although the target variable and attributes are different from the ailerons domain. The target variable here is a variation instead of an absolute value, and there is some pre-selection of the attributes. 7129 cases with 6 continuous attributes.

California Housing: This data set contains information on block groups in California from the 1990 Census. The target variable is median house value. Independent attributes are median income, housing median age, total rooms, total bedrooms, population, households, latitude, and longitude. 20640 cases with 8 continuous attributes.

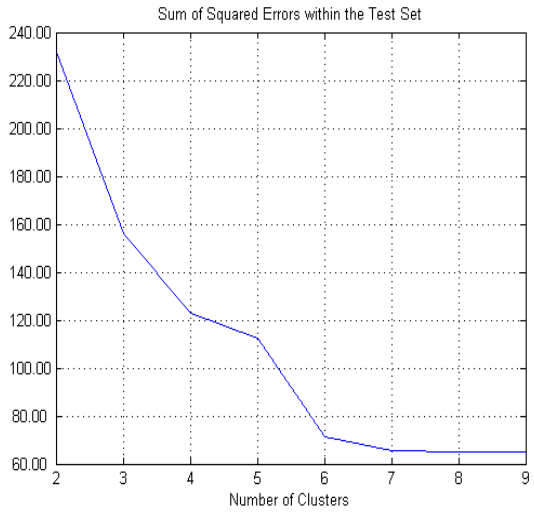

Figure 2: Sum of squared errors within the test data set vs. number of clusters

Stock Data Set: Daily stock prices from January 1988 through October 1991, for ten aerospace companies. 950 cases with 10 continuous attributes.

The number of clusters for the real data set is not known in advance so first the v-fold cross-validation algorithm described in section 2.5.1 is adopted to determine the number of piecewise surfaces which are involved in the piecewise regression problem. The optimal number of clusters in Stock Data Set is calculated to describe the determination process. From Figure 2, it is observed that as the number of clusters increases, the sum of squared errors within the test set goes down fast until the number of cluster is equal to 7. Then the sum of squared errors goes down very slightly and gets stable when the number of clusters is equal to 8, 9 or even more. Consequently we can set the tentative number of clusters to value either 8 or 9. However, the v-fold cross-validation is very time consuming due to the fact that the EMMPSR algorithm is involved in each run of the v-fold cross-validation process. The clustering validity indexes are calculated as well to determine the optimal number of clusters in the Stock data set. Each index value is plotted in Figure 3 against the number of clusters. The star which represents the optimal number of clusters in the figure is circled with a small rectangle box. For Silhouette index, when the number of cluster is equal to 8, its value reaches the peak point, 0.52. Davis-Boudin index and Dunn index both display the optimal number of clusters as 8. The Calinski-Harabasz index shows that the optimal number is 9. We observe very similar result to what is observed in the cross validation.

After we determine the optimal number of clusters for every benchmark data set, the learning outcome is summarized in Table 2.

It is observed from the Table 2 that the EMMPSR algorithm can mostly achieve the best learning outcome among the four algorithms and at the same time, has the simplest format of representation for the piecewise regression problem.

Figure 4 displays the effect of ensemble learning by majority voting which is described in section 2.3.2.1. The measure to evaluate its learning effect is the number of cluster index adjustments during each iteration of the learning process. From Figure 4, we can observe that

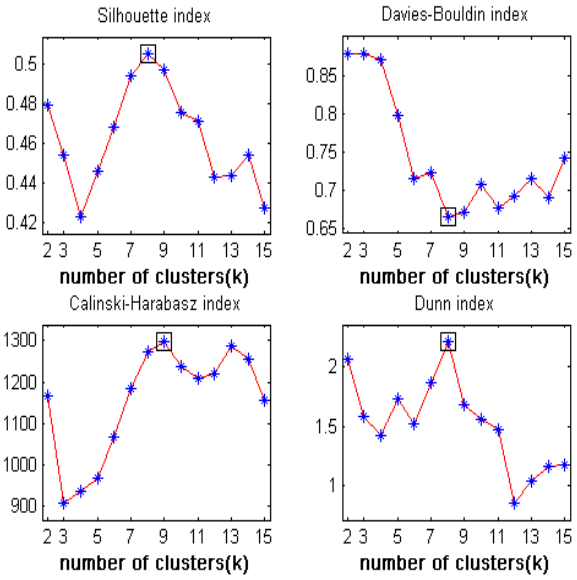

Figure 3: Clustering Quality Validity Index vs. The Number of clusters

Table 2: RMSE values for performance comparison experiments on benchmark data sets

Model	M5P	MultilayerPerceptron	Classregtree	EMMPSR
Auto	3.8419	4.6238	4.0455	2.5332
Delta Ailerons	0.0002	0.0002	0.0003	0.0002
California Housing	0.4838	0.6022	0.5998	0.1617
Stock	1.0151	1.3441	0.9746	0.4876

the number of cluster index adjustments after applying the majority voting ensemble learning decreases faster than the learning process by applying Equation 2.6. Furthermore, the number of iterations required for the ensemble learning is less than what is required for learning process by applying Equation 2.6. At the same time, the learning outcome (in terms of RMSE) of ensemble learning is slightly smaller as well.

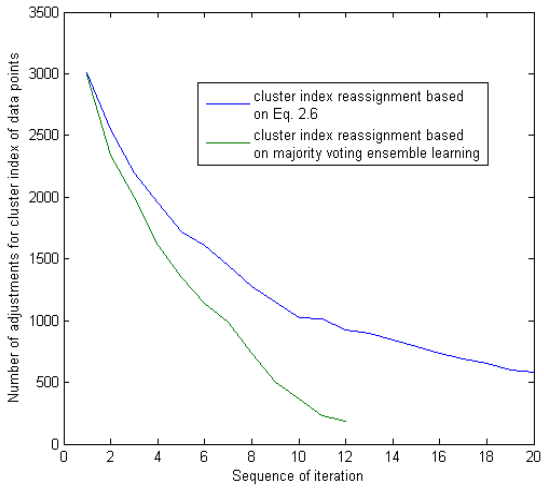

Figure 4: Learning Effect of Ensemble Learning by Majority Voting

4 Conclusion and Future Work

A EM-based multi-step piecewise linear regression learning algorithm is proposed in this paper. A set of experiments are compared to show in most cases the EMMPSR algorithm outperforms other popular packages used for classification and regression. The ensemble learning is involved in the cluster index reassignment phase and proves to perform better than single learning strategy. Future research topic will be main feature selections which will used for regression, and how to improve the EMMPSR algorithm to solve more general form of piecewise surface regression problems.

References

Alpaydin 2004. *Introduction to Machine Learning*, MIT Press.

Arumugam, M. and Scott, S. 2004. Emprr: A high-dimensional em-based piecewise regression algorithm, *The 2004 International Conference on Machine Learning and Applications*, pp. 264–271.

Breiman, L., Friedman, J. H., Olshen, R. A. and Stone, C. J. 1984. *Classification and Regression Trees*, Statistics/Probability Series, Wadsworth Publishing Company, Belmont, California, U.S.A.

Calinski, R. and Harabasz, J. 1974. A dendrite method for cluster analysis, *Communications in Statistics - Theory and Methods* **3**(1): 1–27.

Chang, C.-C. and Lin, C.-J. 2001. *LIBSVM: a library for support vector machines*. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Cherkassky, L.-N. 1991. Constrained topological mapping from non-parametric regression analysis, *Neural Networks* **4**: 27–40.

Davies, D. L. and Bouldin, D. W. 1979. A cluster separation measure, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-1**(2): 224–227.

Dempster, A. P., Laird, N. M. and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the em algorithm, *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B* **39**(1): 1–38.

Draper, N. and Smith, H. 1998. *Applied Regression Analysis Wiley Series in Probability and Statistics*.

Dunn, J. C. 1974. Well separated clusters and optimal fuzzy-partitions, *Journal of Cybernetics* **4**: 95–104.

Ferrari-Trecate, G. and Muselli, M. 2002. A new learning method for piecewise linear regression, *Artificial Neural Networks ICANN 2002*, Vol. 2415 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 135–135.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I. H. 2009. The weka data mining software: An update, *SIGKDD Explorations* **11**.

Hathaway, R. and Bezdek, J. 1993. Switching regression models and fuzzy clustering, *Fuzzy Systems, IEEE Transactions on* **1**(3): 195–204.

Hill, T. and Lewicki, P. 2007. *STATISTICS Methods and Applications*, StatSoft.

Huber, P. and Ronchetti, E. 1981. *Robust statistics*, Wiley series in probability and mathematical statistics, Wiley, New York, NY, U.S.A.

Jain, A. K., Duin, R. P. W. and Mao, J. 2000. Statistical pattern recognition: A review, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**: 4–37.

Levenberg, K. 1944. A method for the solution of certain non-linear problems in least squares, *Quarterly Journal of Applied Mathmatics* **II**(2): 164–168.

LIACC 2006. *Data Set*. Website at `http://www.liacc.up.pt/~ltorgo/Regression/`.

Luo, J. and Brodsky, A. 2010. An optimal regression algorithm for piecewise functions expressed as object-oriented programs, *Machine Learning and Applications, 2010. ICMLA '10. Ninth International Conference on*, pp. 937–942.

Luo, J. and Brodsky, A. 2011. A heaviside-based regression of piecewise functions expressed as object-oriented programs, *Machine Learning and Computing, 2011. ICMLC '11. Third International Conference on*, Vol. 1, pp. 296–301.

MacKay, D. 2003. An example inference task: Clustering, *Information Theory, Inference and Learning Algorithms* (284).

Marquardt, D. W. 1963. An algorithm for least-squares estimation of nonlinear parameters, *Journal of the Society for Industrial and Applied Mathematics* **11**(2): 431–441.

MATLAB 2010. *version 7.10.0 (R2010a)*, The MathWorks Inc., Natick, Massachusetts.

McGee, C. 1963. Piecewise regression, *Journal of the Society for Industrial and Applied Mathematics* **11(2)**: 431–441.

Nusser, S., Otte, C. and Hauptmann, W. 2008. An em-based piecewise linear regression algorithm, *Proceedings of the 3rd international workshop on Hybrid Artificial Intelligence Systems*, pp. 466–474.

Quinlan 1992. Learning with continuous classes, *Proceedings of the Second Australian Conference on Aritificial Intelligance*, pp. 343–348.

Ritter 1991. Learning with the self-organizing map, *Artificial Neural Networks* pp. 379–384.

Rousseeuw, P. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, *J. Comput. Appl. Math.* **20**: 53–65.

Wang, Y. and Witten, I. H. 1997. Inducing model trees for continuous classes, *In Proc. of the 9th European Conf. on Machine Learning Poster Papers*, pp. 128–137.