

# Regression Clustering

Bin Zhang

Hewlett-Packard Research Laboratories, Palo Alto, CA 94304

bzhang@hpl.hp.com

## Abstract

*Complex distribution in real-world data is often modeled by a mixture of simpler distributions. Clustering is one of the tools to reveal the structure of this mixture. The same is true to the datasets with chosen response variables that people run regression on. Without separating the clusters with very different response properties, the residue error of the regression is large. Input variable selection could also be misguided to a higher complexity by the mixture. In Regression Clustering (RC),  $K$  ( $>1$ ) regression functions are applied to the dataset simultaneously which guide the clustering of the dataset into  $K$  subsets each with a simpler distribution matching its guiding function. Each function is regressed on its own subset of data with a much smaller residue error. Both the regressions and the clustering optimize a common objective function. We present a RC algorithm based on K-Harmonic Means clustering algorithm and compare it with other existing RC algorithms based on K-Means and EM.*

## 1. Introduction

Two important data mining techniques are regression on the datasets with chosen response variables, and clustering on the datasets that do not have response information. The RC algorithm handles the case in between: the datasets that have response variables but they do not contain enough information to guarantee high quality learning, the missing part of the response is essential. Missing information is generally caused by insufficiently controlled data collection, due to a lack of means, a lack of understanding or other reasons. For example, sales or marketing data collected on all customers may not have the label on a proper segmentation of the customers.

Clustering algorithms partition (hard or soft) a dataset into a finite number of subsets each containing similar data points. Dissimilarity labeled by the index of the partitions provides additional supervision to the  $K$  regressions running in parallel so that each regression works on a subset of similar data. The  $K$  regressions in

turn provide the model of dissimilarity for clustering to partition the data. The linkage is a common objective function minimized by both the regressions and the clustering. Neither can be properly performed alone.

The concept of regression clustering is not new. A number of earlier papers are reviewed in the next section. This paper adds a new member, Regression-K-Harmonic Means clustering, to the family of RC algorithms and compares its performance with others.

### 1.1. Related Previous Work

Regression clustering has been studied under a number of different names: *Clusterwise Linear Regression* in Spath [14-17], DeSarbo and Cron [2], Hennig [6-8] and others; *Trajectory clustering using mixtures of regression models* by Gaffney and Smith [4]; *Fitting Regression Model to Finite Mixtures* by Williams [20]; *Clustered Partial Linear Regression* by Torgo [19]. We choose the name Regression-Clustering because a) RC is not limited to linear regressions; b) Comparing RC with center-based clustering algorithms, KM, KHM, and EM, the centers are replaced by regression functions -- RCs are just regression-function-centered clustering algorithms; c) By examining the computational structure, the clustering algorithm represents the main (outer) loop or the overall program structure, and the regression is called only as a subroutine to update the "centers".

Clusterwise Linear Regression by Spath [14-17] used linear regression and partition of the dataset in his algorithm that locally minimize the total mean square error over all  $K$ -regression (Eq. (2)). He also developed an incremental version to allow adding new observations into the dataset. Spath's algorithm is based on *K-means* clustering algorithm. DeSarbo [2] used maximum likelihood methodology for performing clusterwise linear regression, locally minimizing the objective function (Eq. (16)). A marketing application is presented in his paper. We will briefly introduce the details of his work in section 6 for comparison. Hennig continued the research of Clustered Linear Regression using the same linear mixing of Gaussian density functions. The number of clusters in his work is treated as unknown. Gaffney and

Smyth's work [4] is also based on EM clustering algorithm.

## 1.2. Contributions of This Paper

Previous work on RC used K-Means and EM in their algorithms, these RC algorithms will have the same well-known problem of being sensitive to the initialization of the regression functions as the K-Means and EM being sensitive to the initialization of the centers.

The author developed a center-based clustering algorithm, K-Harmonic Means, which is much less sensitive to initialization of centers. It is demonstrated through a large number of experiments on randomly generated datasets that KHM converges to better local optimum than K-Means and EM, measured by a common objective function of K-Means (Zhang [23][24]).

In this paper, we add a new algorithm RC-KHM to the family of RC algorithms (Section 4 and 5), provide performance comparisons of the three RC algorithms based on extensive experimental results (Section 11 and Fig. 5.), and give an interpretation of the  $K$ -regression functions as a predictor and its combination with a  $K$ -way classifier (Section 10).

The rest of the paper is organized in sections as: Section 2, defining the problem; Section 3 and 4, the RC-KM and its special case LinReg-KM. Section 5 and 6, the new RC-K-Harmonic Means and its special case LinReg-KHM; Section 7 and 8, the RC-Expectation Maximization algorithm and LinReg-EM; Section 9, computational costs; Section 10, probability interpretation of the  $K$ -regression functions as predictors; Section 11, experimental results and comparisons; Section 12, Conclusions.

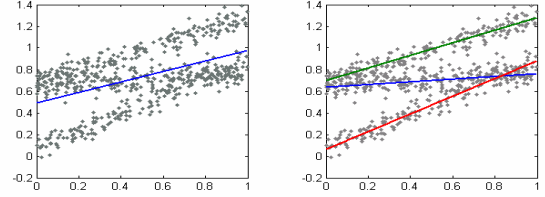
## 2. The Problem

Given a dataset with supervising responses,  $Z = (X, Y) = \{(x_i, y_i) \mid i = 1, \dots, N\}$ , a (constrained) family of functions  $\Phi = \{f\}$  and an loss function  $e() \geq 0$ , regression solves the following minimization problem,

$$f^{opt} = \arg \min_{f \in \Phi} \sum_{i=1}^N e(f(x_i), y_i) \quad (1)$$

Usually,  $\Phi = \{\sum_{l=1}^m \beta_l h(x, a_l) \mid \beta_l \in R, a_l \in R^n\}$ , linear expansions of simple parametric functions such as polynomials of degree up to  $m$ , Fourier series of bounded frequency, neural networks, RBF, ..... Usually,  $e(f(x), y) = \|f(x) - y\|^p$ , with  $p=1,2$  most widely used. (1) is not effective when the data set contains a mixture of very different response characteristics as

shown in Fig. 1a. It is much better to find the partitions in the data and learn a separate function on each partition as shown in Fig. 1b.



**Fig. 1: a) Left: a single function is regressed on all training data which is a mixture of three different distributions. b) Right: three regression functions, each regressed on a subset found by RC. The residue errors are much smaller.**

We assume that there are  $K$  partitions in the data. Determining the right  $K$  has been discussed in the clustering context [5][18], which still applies under our new setting.  $K$  can also be determined (or bounded) by other aspects of the original problem.

In RC algorithms,  $K$  regression functions  $M = \{f_1, \dots, f_K\} \subset \Phi$  are applied to the data, each of which finds a partition  $Z_k$  and regress on it. Both parts of the process -- the  $K$  regressions and the partitioning of the dataset -- optimize a common objective function. The partition of the dataset can be a "soft" partition given by  $K$  density functions defined on the dataset.

## 3. The RC-KM Algorithm

Clusterwise Linear Regression [14] is the simplest RC algorithm. We review it as an introduction to RC. The  $K$  regressions do not have to be linear.

RC-KM solves the following optimization problem,

$$\min_{\{f_k\} \subset \Phi, \{Z_k\}} Perf_{RC-KM} = \sum_{k=1}^K \sum_{(x_i, y_i) \in Z_k} e(f_k(x_i), y_i), \quad (2)$$

where  $Z = \bigcup_{k=1}^K Z_k$  ( $Z_k \cap Z_{k'} = \emptyset, k \neq k'$ ). The optimization is over both the  $K$  regression functions and the partition. The optimal partition will satisfy

$$Z_k = \{(x, y) \in Z \mid e(f_k^{opt}(x), y) \leq e(f_{k'}^{opt}(x), y) \quad \forall k' \neq k\}, \quad (3)$$

which allows us to replace the function in (2) by

$$Perf_{RC-KM}(Z, \{f_k\}_{k=1}^K) = \sum_{i=1}^N \min\{e(f_k(x_i), y_i) \mid k=1, \dots, K\} \quad (4)$$

*RC-KM Algorithm*, a monotone-convergent algorithm to find a local optimum of (2):

*Step1:* Pick  $K$  functions  $f_1^{(0)}, \dots, f_K^{(0)} \in \Phi$  randomly, or by any heuristics that are believed to give a good start.

*Step2: Clustering Phase:* In the  $r$ -th iteration,  $r=1, 2, \dots$ , repartition the dataset as

$$Z_k^{(r)} = \{(x, y) \in Z \mid e(f_k^{(r-1)}(x), y) \leq e(f_{k'}^{(r-1)}(x), y) \quad \forall k' \neq k\}. \quad (5)$$

(A tie can be resolved randomly among the winners.)

Intuitively, each data point is associated with the regression function that gives the smallest approximation error on it. Algorithmically, a data point in  $Z_k^{(r-1)}$  is moved to  $Z_{k'}^{(r)}$  iff  $e(f_{k'}^{(r-1)}(x), y) < e(f_k^{(r-1)}(x), y)$  and  $e(f_{k'}^{(r-1)}(x), y) \leq e(f_{k''}^{(r-1)}(x), y)$  for all  $k'' \neq k, k'$ .  $Z_k^{(r)}$  gets all the data points in  $Z_k^{(r-1)}$  that are not moved.

*Step3: Regression Phase:* Run any regression optimization algorithm that gives the following

$$f_k^{(r)} = \arg \min_{f \in \Phi} \sum_{(x_i, y_i) \in Z_k} e(f(x_i), y_i) \quad \text{for } k = 1, \dots, K. \quad (6)$$

(The regression algorithm is selected by the nature of the original problem or other criteria. RC adds no additional constraint on its selection.)

*Step4: Stopping Rule:* Run Step 2 and Step 3 repeatedly until there is no more data points changing its membership.

Step 2 and Step 3 never increase the value of the objective function in (2). If any data changes its membership in Step 2, the objective function is strictly decreased. Therefore, the algorithm stops in finite number of iterations.

#### 4. MSE Linear Regression with K-Means Clustering -- LinReg-KM

With  $\bar{D}$  functions  $h_1(x), \dots, h_{\bar{D}}(x)$  chosen as the basis, we consider the function class  $\Phi = \{\sum_{l=1}^{\bar{D}} c_l h_l(x) \mid c_l \in R\}$ . To simplify the notations, let  $\bar{x} = (h_1(x), \dots, h_{\bar{D}}(x))$  and  $\bar{X} = [\bar{x}_i]_{N \times \bar{D}}$ . As an example, for the set of two-variable ( $D=2$ ) polynomials up to degree 2, the basis functions are  $h_1(x) = 1$ ,  $h_2(x) = x_1$ ,  $h_3(x) = x_2$ ,  $h_4(x) = x_1^2$ ,  $h_5(x) = x_1 x_2$ ,  $h_6(x) = x_2^2$ . We have

$$\bar{X} = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & x_{1,1}^2 & x_{1,1}x_{1,2} & x_{1,2}^2 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & x_{N,1} & x_{N,2} & x_{N,1}^2 & x_{N,1}x_{N,2} & x_{N,2}^2 \end{bmatrix}.$$

With the MSE  $e(f(x), y) = |f(x) - y|^2$ , LinReg-KM minimizes the objective function

$$Perf_{LinReg-KM}(Z, \{f_k\}_{k=1}^K) = \sum_{i=1}^N \min_{1 \leq k \leq K} \{\|\bar{x}_i * c_k - y_i\|^2\}.$$

With row-partition of  $Z$  into  $K$  subsets  $Z_1, \dots, Z_K$ , matrices  $\bar{X}$  and  $Y$  are row-partitioned accordingly,  $\bar{X} \rightarrow \bar{X}_1, \dots, \bar{X}_K$  and  $Y \rightarrow Y_1, \dots, Y_K$ , the coefficients of the optimal function on the  $k$ -th subset is (Step 3 of the RC-KM)

$$c_k = (\bar{X}_k^T * \bar{X}_k)^{-1} \bar{X}_k^T * Y_k. \quad (7)$$

The matrix of losses used for the comparisons in Step 2 of RC-KM is

$$E = [e(f_k(x_i), y_i)]_{N \times K} = abs(\bar{X} * [c_1, \dots, c_K] - [Y, \dots, Y]). \quad (8)$$

(squaring is monotone and not necessary.)

#### 5. RC-K-Harmonic Means Algorithm (RC-KHM)

K-Means clustering algorithm is known to be sensitive to the initialization of its centers. The same is true for RC-KM. Convergence to a poor local optimum has been observed quite frequently (See Fig 5).

K-Harmonic Means clustering algorithm showed very strong insensitivity to initialization due to its dynamic weighting of the data points (Zhang 2001, 2003). The regression clustering algorithm RC-KHM<sub>p</sub> is presented in this section. It is shown experimentally that it out-performs RC-KM and RC-EM.

RC-KHM<sub>p</sub>'s objective function is defined by replacing the  $MIN()$  function in (4) by the harmonic average  $HA()$ . The error function is  $e(f_k(x_i), y_i) = \|f_k(x_i) - y_i\|^p$ ,  $p \geq 2$ ,

$$Perf_{RC-KHM_p}(Z, M) = \sum_{i=1}^N HA\{\|f_k(x_i) - y_i\|^p\} = \sum_{i=1}^N \frac{K}{\sum_{k=1}^K \frac{1}{\|f_k(x_i) - y_i\|^p}} \quad (9)$$

An iterative algorithm (see Zhang 2001) is available for finding a local optimum of (9).

*RC-KHM Algorithm:*

*Step 1:* Pick  $K$  functions  $f_1^{(0)}, \dots, f_K^{(0)} \in \Phi$  randomly.

*Step 2: Clustering Phase:* In the  $r$ -th iteration, let

$$d_{i,k} = \|f_k^{(r-1)}(x_i) - y_i\|. \quad (10)$$

a) The hard partition  $Z = \bigcup_{k=1}^K Z_k$ , in RC-KM, is replaced by a "soft" membership function – the  $i$ -th data point is associated with the  $k$ -th regression function with probability

$$p(Z_k \mid z_i) = d_{i,k}^{p+q} / \sum_{l=1}^K d_{i,l}^{p+q}. \quad (11)$$

The choice of  $q$  ( $\geq 1$ ) will put the regression's error function in  $L^q$ -space. See (13). (This is more general than the K-Harmonic Means clustering algorithm

presented before, which had  $q = 2$ .) For simpler notations, we do not index  $p(Z_k | z_i)$  and  $a_p(z_i)$  in (12)

by  $q$ . Quantities  $d_{i,k}$ ,  $p(Z_k | z_i)$ , and  $a_p(z_i)$  should be indexed by the iteration  $r$ , which is also dropped.

b) In RC-KHM, not all data points fully participate in all iterations like in RC-KM. Each data point's participation is weighted by

$$a_p(z_i) = \sum_{l=1}^K d_{i,l}^{p+q} / \sum_{l=1}^K d_{i,l}^p. \quad (12)$$

$a_p(z_i)$  is small if and only if  $z_i$  is close to one of the functions (i.e. done for it). Weighting function  $a_p(z_i)$  changes in each iteration as the regression functions are updated. If all functions drifted away from a point  $z_i$  in the last iteration,  $a_p(z_i)$  goes up. More details on this weighting function are in (Zhang 2001).

**Step 3: Regression Phase:** Run any regression optimization algorithm that gives the following

$$f_k^{(r)} = \arg \min_{f \in \Phi} \sum_{i=1}^N a_p(z_i) p(Z_k | z_i) \| f(x_i) - y_i \|^q \quad (13)$$

for  $k = 1, \dots, K$ .

**Step 4:** Since there is no discrete membership change in RC-KHM, the stopping rule is replaced by measuring the changes to its objective function (9), when the change is smaller than a threshold, the iteration is stopped.

## 6. Linear Regression with K-Harmonic Means Clustering -- LinReg-KHM

For linear regression, we choose  $q=2$ . Writing (13) in matrix form, we have

$$c_k^{(r)} = \arg \min_c (\bar{X}^* c - Y)^T * \text{diag}(a_p(z_i) p(Z_k | z_i)) * (\bar{X}^* c - Y) \quad (14)$$

$1 \leq i \leq N$

and its solution is

$$c_k^{(r)} = \left( \bar{X}^T * \left[ \bar{x}_i / d_{i,k}^{p+2} \left( \sum_{l=1}^K \frac{1}{d_{i,l}^p} \right)^2 \right]_{N \times \bar{D}} \right)^{-1} * \bar{X}^T * \left[ y_i / d_{i,k}^{p+2} \left( \sum_{l=1}^K \frac{1}{d_{i,l}^p} \right)^2 \right]_{N \times \bar{D}} \quad (15)$$

where  $d_{i,k} = \|\bar{x}_i * c_k^{(r-1)} - y_i\|$ . ( $[\alpha]_{N \times \bar{D}}$  is a matrix of size  $N \times \bar{D}$  with entries  $\alpha$  being one of three possibilities: row vectors, column vectors or scalars.) The inversion in (15) is on a  $\bar{D} \times \bar{D}$  matrix.

## 7. The RC-EM Algorithm

One of the applications of the general EM algorithm (McLachlan and Krishnan [11]) is on probability density estimation or clustering. The best of the linear mixing of

Gaussian EM clustering algorithm is the natural probability interpretation of its linear mixing (superposition). We include a brief presentation of RC-EM for comparing the performance of all three algorithms in Section 11. The objective function for RC-EM is defined as

$$Perf_{RC-EM}(Z, M) = -\log \left\{ \prod_{i=1}^N \sum_{k=1}^K \frac{P_k}{\sqrt{(2\pi)^d |\Sigma_k|}} \exp \left( -\frac{1}{2} (f_k(x_i) - y_i) \Sigma_k^{-1} (f_k(x_i) - y_i)^T \right) \right\} \quad (16)$$

where  $d = \dim(Y)$ . In case  $d=1$ ,  $(f_k(x_i) - y_i)$  is just a real number and  $\Sigma_k^{-1} = 1/\sigma_k^2$ . In higher dimensions, restriction to the covariance matrix  $\Sigma_k$  is necessary for EM to work properly.  $\Sigma_k = \text{diagonal matrix}$  is often used.

The RC-EM recursion is given by

**E-Step:**

$$p(Z_k^{(r)} | z_i) = \frac{\frac{P_k^{(r-1)}}{\sqrt{|\Sigma_k|}} \exp \left( -\frac{1}{2} (f_k^{(r-1)}(x_i) - y_i) \Sigma_{r-1,k}^{-1} (f_k^{(r-1)}(x_i) - y_i)^T \right)}{\sum_{k=1}^K \frac{P_k^{(r-1)}}{\sqrt{|\Sigma_k|}} \exp \left( -\frac{1}{2} (f_k^{(r-1)}(x_i) - y_i) \Sigma_{r-1,k}^{-1} (f_k^{(r-1)}(x_i) - y_i)^T \right)} \quad (17)$$

$$\text{M-Step: } p_k^{(r)} = \frac{1}{N} \sum_{i=1}^N p(Z_k^{(r)} | z_i) \quad (18)$$

$$f_k^{(r)} = \arg \min_{f \in \Phi} \sum_{i=1}^N p(Z_k^{(r)} | z_i) \| f(x_i) - y_i \|^2 \quad (19)$$

$$\Sigma_{r,k} = \frac{\sum_{i=1}^N p(Z_k^{(r)} | z_i) (f_k^{(r)}(x_i) - y_i)^T (f_k^{(r)}(x_i) - y_i)}{N * p_k^{(r)}} \quad (20)$$

## 8. MSE Linear Regression with EM Clustering -- LinReg-EM

When MSE linear regression is used, (19) can be solved and takes the following special form, while all other formulas (16)-(18) and (20) remain the same.

$$c_k^{(r)} = \left( \bar{X}^T * [p(Z_k^{(r)} | z_i) \bar{x}_i]_{N \times \bar{D}} \right)^{-1} * \bar{X}^T * [p(Z_k^{(r)} | z_i) y_i]_{N \times 1} \quad (21)$$

Very strong similarity between (21) and LinReg-KHM's (15), or between (21) and LinReg-KM' (7) can be observed.

## 9. Computational Costs for RCs with MSE Linear Regression

We compare the cost of one iteration of RC with the cost of single function linear regression on the whole

dataset without clustering for all three examples LinReg-KM, LinReg-KHM and LinReg-EM. This comparison shows the cost ratio of switching from single function regression to RC.

The cost of forming  $\bar{X}$  is common to both RC and single linear regression. In single linear regression, the cost of calculating  $c = (\bar{X}^T * \bar{X})^{-1} \bar{X}^T * Y$  is the sum of (an unit of calculation here is multiplying two numbers and adding the result to another number):  $\bar{D}^2 * N$  units for forming  $\bar{X}^T * \bar{X}$ ,  $\bar{D}^2 + \bar{D} * N$  units for forming  $\bar{X}^T * Y$  and  $\beta \bar{D}^3$  for solving  $(\bar{X}^T * \bar{X}) * c = \bar{X}^T * Y$ ,  $\beta$  is a small constant, where  $\bar{D} = m + 1$  if  $D = I$ , or  $\bar{D} = \frac{D^{m+1} - 1}{D - 1}$  for  $D > I$ .  $D = \dim(X)$ .  $N \geq \bar{D}$ , otherwise the regression has infinite solutions. We assume that  $N \gg \bar{D}$ , otherwise the potential of over fitting (and/or over shooting) is high. In any case the dominate term is  $O(\bar{D}^2 * N)$ . Let  $N_k$  be the size of the  $k$ th cluster, the costs of  $K$  regressions are  $\sum_{k=1}^K \bar{D}^2 * N_k = \bar{D}^2 * N$  units for all  $\bar{X}_k^T * \bar{X}_k$ ,  $k = 1, \dots, K$ ,  $K\bar{D}^2 + \bar{D} * N$  units for all  $\bar{X}_k^T * Y_k$  and  $K\beta\bar{D}^3$  for solving  $K$  linear equations,  $(\bar{X}_k^T * \bar{X}_k) * c_k = \bar{X}_k^T * Y_k$ .  $K$  is very small and we do not expect it ever to be large (say  $> 50$ ). The repartition cost for LinReg-KM is  $O(\bar{D} * N * K)$  due to the number of error function evaluations and comparisons. Therefore, the cost of each iteration of LinReg-KM is at the same order of complexity as the simple single function regression.

We observed a quick convergence at start in all experiments but some of them had a long tail. (See Section 11.2)

The cost of calculating the repartition probabilities in LinReg-KHM and LinReg-EM are in the same order as the repartition cost in LinReg-KM.

With input variable selection, not all the variables selected for the single function regression need to appear in the selected variables for each subset. Therefore, the dimensionality of the regression problem on each subset may become lower.

## 10. Probability Interpretation of RC's $K$ Regression Functions

Regression results are most often used for predictions,  $y = f(x)$  is taken as a prediction of the response at a new  $x \notin X$ . With  $K$  regression functions

returned by RC, we get  $K$  predictions  $\{f_k(x)\}_{k=1}^K$  on the same input  $x$ , which is interpreted in this section.

Assuming that dataset  $X$  is iid sampled from a hidden density distribution  $P()$ . Kernel density estimation on the  $K$   $X$ -projections of  $Z_k = \{p(Z_k | z) \mid z = (x, y) \in Z\}$  (for KHM and EM see (11) & (17), for KM they are the real subsets) gives

$$\hat{P}(x | X_k) = \frac{\frac{1}{N} \sum_{i=1}^N p(Z_k | z_i) H\left(\frac{x_i - x}{h}\right)}{\hat{P}(X_k)} \quad (22)$$

$$\text{with } \hat{P}(X_k) = \frac{1}{N} \sum_{i=1}^N p(Z_k | z_i). \quad (23)$$

$H()$  in (22) is a symmetric kernel and  $h$  the bandwidth (See [13]). If we add the density estimation of each subset, we get the kernel density estimation on the whole dataset,

$$\hat{P}(x) = \sum_{k=1}^K \hat{P}(x | X_k) \hat{P}(X_k) = \frac{1}{N} \sum_{i=1}^N H\left(\frac{x_i - x}{h}\right). \quad (24)$$

Bayes' inversion gives the probability of  $x$  belongs to each subset,

$$\hat{P}(X_k | x) = \frac{\hat{P}(x | X_k) \hat{P}(X_k)}{\hat{P}(x)} = \frac{\sum_{i=1}^N p(Z_k | z_i) H\left(\frac{x_i - x}{h}\right)}{\sum_{i=1}^N H\left(\frac{x_i - x}{h}\right)} \quad (25)$$

Let  $\tilde{f}(x)$  be the random variable prediction which equals  $f_k(x)$  with probability  $P(X_k | x)$ , and the expected value of this prediction is estimated by

$$E(\tilde{f}(x) | x) \approx \sum_{k=1}^K f_k(x) \hat{P}(X_k | x) = \frac{\sum_{i=1}^N \left[ \sum_{k=1}^K f_k(x) p(Z_k | z_i) \right] H\left(\frac{x_i - x}{h}\right)}{\sum_{i=1}^N H\left(\frac{x_i - x}{h}\right)} \quad (26)$$

A random variable contains more information than its expectation; therefore, RC's prediction  $\tilde{f}(x) | x$ , a random variable, gives more information than its expectation  $E(\tilde{f}(x) | x)$ . Instead of giving a single valued prediction with a large uncertainty,  $\tilde{f}(x) | x$  gives  $K$  possible values each with a much smaller uncertainty. The significant part of the uncertainty is described by the probability distribution  $\{P(X_k | x), k = 1, \dots, K\}$ .

A classifier,  $k = C(x)$ , can be trained using the labels provided by the clustering phase of the RC algorithm. In case the false classification rate of  $C$  is low, which may not be true for some datasets, a prediction on  $x$  can be  $f_{C(x)}(x)$ .

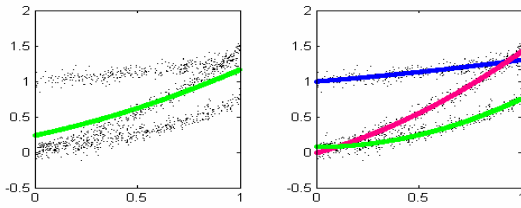
## 11. Experimental Results

We conducted three sets of experiments: Set 1 for visualization of RC, and Set 2 for statistical comparisons of LinReg-KM, LinReg-KHM and LinReg-EM.

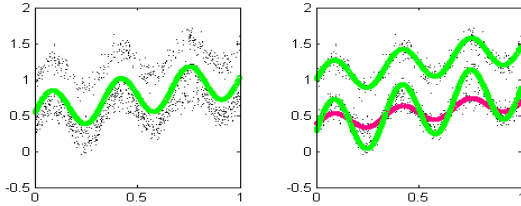
### 11.1. Visualization Experiments

This section visually demonstrates RC. Statistical performance analysis and comparison of different variations of RCs are in the next section.

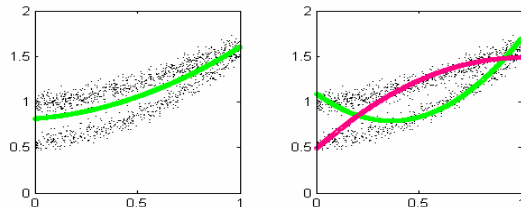
Dimensionality of  $X$  is 1, so that 2-dimensional visualization can be presented. Linear regression RC is already demonstrated in **Fig.1b**. We do both quadratic (**Fig. 2**) and trigonometric (**Fig. 3**) regressions in this section.



**Fig. 2.**  $N=600, D=1, K=3$ . On the left is the result of simple quadratic regression on the whole dataset. On the right is LinReg-KHM.



**Fig. 3.**  $N=1200, D=1, K=3$ .  $\Phi = \{a_1 \sin(6\pi x) + a_2 x + a_3 \mid a_i \in R\}$  and the data set is a mixture of three subsets generated by three functions in  $\Phi$  with added Gaussian noise. Left: one regression function is applied to the whole dataset. Right: three regression functions are used. Each of them found a very good approximation of the original functions used to generate the dataset.



**Fig. 4.** A local optimum. It happens to all three RC algorithms, RC-KM, RC-KHM, and RC-EM.

Ploy-KHM, with the version KHM presented in Zhang et al [22] which is better for one and two dimensional spaces, is used in this section.

A local optimum is shown in **Fig. 4**. This tells us how the algorithms may fail to reach the global optimum. Knowing this helps to manually correct it, by providing a special initialization after seeing a suspected result.

### 11.2 Statistical Comparisons of LinReg-KM, LinReg-KHM and LinReg-EM

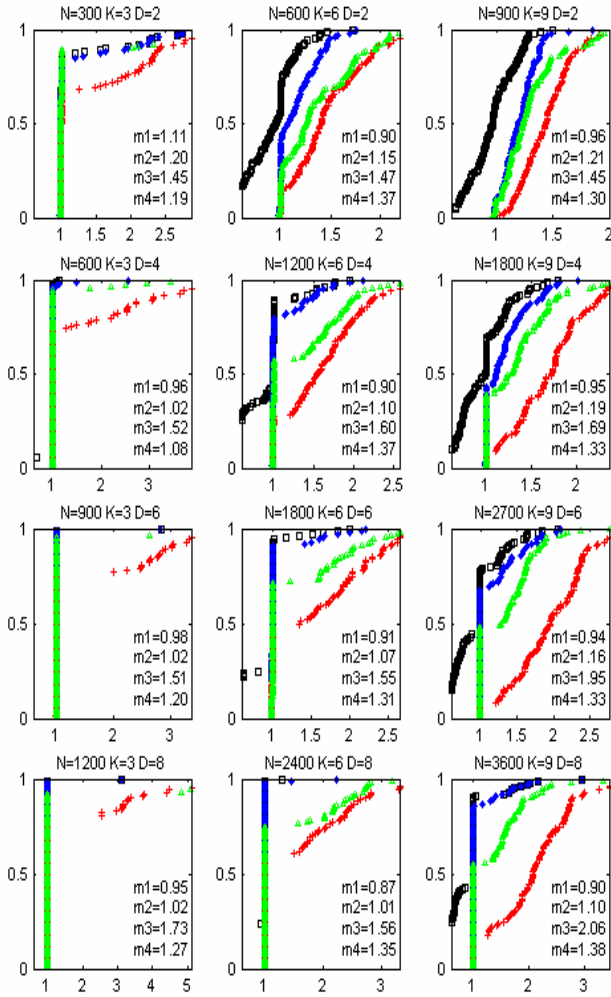
Twelve sets of experiments, with  $D = 2, 4, 6, 8$  and  $K = 3, 6, 9$ , are conducted. In each set, 60 datasets with  $N = 50 \cdot D \cdot K$  are generated by randomly picking  $N$  points on  $K$  randomly generated hyperplanes and then adding Gaussian noise to the  $y$ -components. The regression functions are linear (hyperplanes). For each dataset, a common initialization of the regression functions is used for all three different algorithms.

To make direct comparisons of three algorithms possible, we have to measure them by a common performance measure, which is chosen to be the LinReg-KM's objective function in (2). After LinReg-KHM and LinReg-EM converged, we discard its own performance measure, and re-measure its result by the LinReg-KM's. Doing so is slightly in favor of LinReg-KM. We use the notations  $Perf_{KHM/KM}$  and  $Perf_{EM/KM}$  for these re-measurements.

Taking advantage of the known partitions of the synthetic datasets, we calculated a  $Perf_{baseline}$ , by running regression on each of the  $K$  subsets and add them up, for comparing against the performance of LinReg-KM and LinReg-KHM.  $Perf_{baseline}$  is close to the global optimum.

The results are in **Fig 5**. Each curve has 60 points from the 60 runs of RC, without interpolation. Four curves in each plot, which are frequency-estimations of the accumulative distributions in (22)-(25), with  $v$ -axis horizontal and  $prob$ -axis vertical,

$$\begin{aligned} & \Pr\left(\frac{Perf_{KHM/KM}}{Perf_{EM/KM}} < v\right), \quad \Pr\left(\frac{Perf_{KHM/KM}}{Perf_{baseline}} < v\right), \\ & \Pr\left(\frac{Perf_{RC-KM}}{Perf_{baseline}} < v\right), \quad \Pr\left(\frac{Perf_{EM/KM}}{Perf_{baseline}} < v\right) \end{aligned} \quad (22-25)$$



**Fig 5. The accumulative distribution of the performance ratios. Icons and the text in each plot: black squares: LinReg-KHM over LinReg-EM; blue (\*)'s: LinReg-KHM over the baseline; red (+)'s: LinReg-KM over the baseline and green triangles: LinReg-EM over the baseline.  $m1$  = mean of the ratios of LinReg-KHM over LinReg-EM,  $m2$  = mean of the ratios of LinReg-KHM over the baseline,  $m3$  = mean of the ratios of LinReg-KM over the baseline, and  $m4$  = mean of the ratios of LinReg-EM over the baseline.**

The plot of (22), in black squares, shows how often LinReg-KHM performed better than LinReg-EM, with equal performance when the ratio is 1.

The plot of (23), in blue (\*)'s, shows how well LinReg-KHM performed against the  $Perf_{baseline}$ , which should be very close to the true optimum. When the value is close to 1, a very good approximation of the global optimum was found.

The plot of (24) in red (+)'s and (25) in green triangles shows how well LinReg-KM and LinReg-EM performed against the  $Perf_{baseline}$ .

We truncated the  $x$ -axis to make the interesting part of the plot (near 1) more readable.

In addition to the plotted distributions in (22)-(25), the expectation is also given on each plot,

$$m1 \approx E\left(\frac{Perf_{KHM/KM}}{Perf_{EM/KM}}\right), m2 \approx E\left(\frac{Perf_{KHM/KM}}{Perf_{baseline}}\right), \quad (26)$$

$$m3 \approx E\left(\frac{Perf_{KM}}{Perf_{baseline}}\right), m4 \approx E\left(\frac{Perf_{EM/KM}}{Perf_{baseline}}\right).$$

**Observations:** A) Except for  $K=3$  and  $D=2$ , LinReg-KHM performed the best among the three. As  $K$  and  $D$  increase, the performance gaps become larger; B) LinReg-EM performed better than LinReg-KM on average for all  $K$  and  $D$ . This is due to the low dimensionality of the  $Y$ -space ( $dim(Y)=1$ ), where the clustering algorithms are applied; C) In my previous comparisons on the performance of center-based clustering algorithms (Zhang 2003),  $K$ -means performed better than EM on average on datasets with dimensionality  $> 1$ . The higher the dimensionality of the data, the more  $K$ -Means outperform EM.

## 12. Conclusions

Clustering recovers a discrete estimation of the missing part of the responses and provides each regression function with the right subset of data. A new regression clustering algorithm RC-KHM is presented. LinReg-KHM outperforms both LinReg-EM and LinReg-KM.

In the general form of RCs, the regression part of the algorithm is completely general, no requirements is added to it by the RC algorithm. This implies that a) RC algorithms work with any type of regression; b) RC can be built on top of existing regression libraries and call the existing regression program as a subroutine.

We give two other advantages of using RC. Regression helps with understanding the data by replacing it with an analytical function plus a residue noise. When the noise is small, the function describes the data well. RC does a much better job requirements is added to it by the RC algorithm. This implies that a) RC algorithms work with any type of regression; b) RC can be built on top of existing regression libraries and call the existing regression program as a subroutine.



We give two other advantages of using RC. Regression helps with understanding the data by replacing it with an analytical function plus a residue noise. When the noise is small, the function describes the data well. RC does a much better job on this. The compact representation of data by a regression function can also be considered as (or part of) data compression. With a significantly smaller mean residue noise, RC does a much better job on this too.

EM's linear mixing of simple distributions has the most natural probability interpretation. To benefit from both the EM's probability model and the KHM algorithm's robust convergence, we recommend running RC-KHM first and use its converged results to initialize RC-EM. RC-KHM does not supply the initial values for  $p_k^{(r)}$  and  $\Sigma_{r,k}$ . To solve this problem, keep the initial function-centers fixed at the RC-KHM's output for a number of iterations to let the probabilities  $p_k^{(r)}$  and  $\Sigma_{r,k}$  to converge under RC-EM before setting the function-centers free.

## References

- [1] 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
- [2] DeSarbo, W. S., Corn, L. W. (1988), "A Maximum Likelihood Methodology for Clusterwise Linear Regression," *J. of Classification*, 5:249-282
- [3] Duda, R., Hart, P. (1972), "Pattern Classification and Scene Analysis", John Wiley & Sons
- [4] Gaffney, S., and P. Smyth, 'Trajectory clustering using mixtures of regression models,' in Proceedings of the ACM 1999 Conference on Knowledge Discovery and Data Mining, S. Chaudhuri and D. Madigan (eds.), New York, NY: ACM, 63--72, August 1999.
- [5] Hamerly, G. and Elkan C., **Learning the  $k$  in  $k$ -means**. To appear in the Seventeenth Annual Conference on Neural Information Processing Systems ([NIPS 2003](#))
- [6] Hennig, C. (1997), "Datenanalyse mit Modellen Fur Cluster Linear Regression." Dissertation, Institut Fur Mathmatische Stochastik, Universitat Hamburg
- [7] Hennig, C. (1999): *Models and Methods for Clusterwise Linear Regression in Gaul*, W. and Locarek-Junge, H. (Eds.): Classification in the Information Age, Springer, Berlin, p. 179-187.
- [8] Hennig, C. (2002): Fixed point clusters for linear regression: computation and comparison (Part of Preprint 2000-02) *Journal of Classification* 19, 249-276.
- [9] Lazarevic A. Xu X., Fietz T. and Obradovic Z. (1999): "Clustering-Regression-Ordering Steps for Knowledge Discovery in Spatial Databases", International Joint Conference on Neural Networks (IJCNN'99), July 10-16, Washington, DC. [Paper \(pdf 221k\)](#)
- [10] MacQueen, J. (1967), "Some Methods for Classification and Analysis of Multivariate Observations". Pp. 281-297 in: L. M. Le Cam & J. Neyman [eds.] Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, Vol. 1. University of California Press, Berkeley. xvii + 666 p
- [11] McLachlan, G. J. and Krishnan, T. (1997), "The EM Algorithm and Extensions.", John Wiley & Sons
- [12] Montgomery, D. C., Peck, E. A., Vining, G. G. (2001), "Introduction to Linear Regression Analysis", John Wiley & Sons; 3rd edition, April
- [13] Silverman, B. W. (1998), "Density Estimation for Statistics and Data Analysis," Chapman & Hall/CRC.
- [14] Spath, H. (1979), Algorithm 39: Clusterwise Linear Regression, *Computing*, 22, 367-73.
- [15] Spath, H. (1981), "Correction to Algorithm 39: Clusterwise Linear Regression," *Computing*, 26, 275.
- [16] Spath, H. (1982), "Algorithm 48: A Fast Algorithm for Clusterwise Linear Regression," *Computing*, 29, 175-181.
- [17] Spath, H. (1985), "Cluster Dissection and Analysis," New York: Wiley.
- [18] Tibshirani, R., Walther, G., and Hastie, T. (2000), "Estimating the Number of Clusters in a Dataset via the Gap Statistic", Available at <http://www-stat.stanford.edu/~tibs/research.html>.
- [19] Torgo, L., and Pinto da Costa, J. (2000): "Clustered Partial Linear Regression," *Machine Learning*, 50 (3), pp. 303-319. Kluwer Academic Publishers.
- [20] Williams, J. (2000), "Fitting Regression Models to Finite Mixtures," ANZMAC Visionary Marketing for the 21st Century: Facing the Challenge, 1409-1414.
- [21] Wedel, M. and Steenkamp, J. B. (1991) 'A clusterwise regression method for simultaneous fuzzy market structuring and benefit segmentation,' *Journal of Marketing Research*, 28, pp.385--96.
- [22] Zhang, B., Hsu, M., Dayal, U. (2000), "K-Harmonic Means", Intl. Workshop on Temporal, Spatial and Spatio-Temporal Data Mining, Lyon, France Sept. 12.
- [23] Zhang, B. (2001), "Generalized K-Harmonic Means--Dynamic Weighting of Data in Unsupervised Learning," the First SIAM International Conference on Data Mining (SDM'2001), Chicago, USA, April 5-7.
- [24] Zhang, B. (2003), "Comparison of the Performance of Center-based Clustering Algorithms", the proceedings of PAKDD-03, Seoul, South Korea, April.