Les Cahiers du GERAD ISSN: 0711–2440

Variable Neighborhood Search for Least Squares Clusterwise Regression

G. Caporossi
P. Hansen

G–2005–61

August 2005
Revised: December 2007

Variable Neighborhood Search for Least Squares Clusterwise Regression

Gilles Caporossi
Pierre Hansen

GERAD and Méthodes quantitatives de gestion
HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada, H3T 2A7

{gilles.caporossi; pierre.hansen}@gerad.ca

August 2005
Revised: December 2007

Les Cahiers du GERAD
G–2005–61

Abstract

Clusterwise regression is a technique for clustering data. Instead of using the classical homogeneity or separation criterion, clusterwise regression is based upon the accuracy of a linear regression model associated to each cluster. This model has many advantages, specially for the purpose of data mining, however, the underlying mathematical model is difficult to solve due to its large number of local optima. In this paper, we propose the use of the Variable Neighborhood Search metaheuristic (VNS) to improve the quality of the solution. Two perturbation strategies are described and one of them yields a substantial improvement if compared to multistart (the error is reduced by a factor of more than 1.5 on average for the 10 clusters problem).

Résumé

La régression par classes est une technique de classification des données. Au lieu d'utiliser les critères classiques d'homogénéité et de séparation, la régression par classes est basée sur l'adéquation d'un modèle de régression linéaire associé à chaque classe. Ce modèle a beaucoup d'avantages, particulièrement pour le data mining. Toutefois, le modèle mathématique sous-jacent est difficile à résoudre à cause de son grand nombre d'optima locaux. Dans cet article, nous proposons l'utilisation de la recherche à voisinages variables (VNS) pour améliorer la qualité de la solution. Deux stratégies de perturbation sont décrites et une d'elles donne des améliorations substantielles par rapport au multistart (l'erreur est réduite par un facteur de plus de 1,5 en moyenne pour le problème à 10 classes).

1 Introduction

Clustering helps researchers to understand the core information underlying data by grouping observations in clusters. Clustering may therefore be used to group similar observations (homogeneity criterion) and to separate different observations (separation criterion). These two different criterion are used in hierarchical clustering implemented with respectively the complete or single linkage. Most of the classical clustering algorithms, like $k-means$ [16] are based upon the measure of distance between objects and therefore need a special attention for the scaling part of the preparation task and the removal of irrelevant variables.

However, clustering may also be used to discover patterns in data. In this case, one aims at grouping observations according to their ability to fit the same model. This last application of clustering is becoming more important due to the expansion of data mining, the aim of which is to find useful information in a huge amount of data that was not collected for the purpose of the study. This last point indicates that the data preparation (cleaning, scaling and missing values treatment) part of the task is critical. Data mining requires the use of powerful algorithms to provide good solutions.

It would be surprising that the same data and the same treatment will provide useful informations for every purpose. For instance, if a company that wants to take advantage of the operational data collected during the years, clustering the transactions could be of interest for the marketing division, but also for the distribution division. There is no reason to believe the same clusters will provide useful information to both at the same time. Indeed, the marketing division will probably be interested in clustering the consumer according to the price they are ready to pay while the distribution division will be interested in the distribution method, which is related to the nature of the product bought and the geographical position of the customer in order to plan the implantation of a new depot for instance.

In this paper, we concentrate upon the clusterwise linear regression for which an algorithm was already proposed in 1979 by Späth [18]. By its structure, the clusterwise regression is useful for data mining: it tends to naturally assign a small weight to irrelevant variables and will automatically adapt the model to inappropriate scaling by the choice of the regression coefficients. Furthermore, if a variable is important for a given cluster, it will be considered while it will be assigned a small weight for the clusters where it is not. Indeed the clustering may still be affected by some inaccuracy in the data but the result will not be affected as much as it would be in case of $k-means$ or other similarity based technique. Being less sensitive to the quality of the input data, clusterwise regression is more suitable for the discovery of unknown structure in data without prior knowledge, which is a concern for data-mining. The technique is also well suited to market segmen-

tation and product pricing [1, 3, 4, 22, 23]. The regression models obtained may involve different variables from a segment to another. This variety suggests that a single regression model would not give precise insight on the customers concerns. If two customers buy the same product for different purposes or in different contexts (home use or professional use for example), there is no reason to think the price they are ready to pay will be defined by the same demand function or even use the same information.

The interest in clusterwise linear regression is demonstrated by the publications on the topic. Some of them were application oriented either considering the minimum sum of squares criterion [1, 22], the minimum sum of absolute deviations [20] or the more probabilistic fuzzy clustering approach [21].

Due to a its combinatorial structure and the regression model, clusterwise regression models are hard to optimize. For this reason, researchers have been working on the algorithmic aspect of the problem by the use of metaheuristics as simulated annealing [7] or a bio-mimetic approach (close to genetic algorithms) [1] or other heuristics as a relaxation resolution based upon a mathematical programming approach [15]. A slightly different approach to clusterwise regression, called fixed points clusters was also proposed by Hennig [12, 13].

2 Problem formulation

In this paper, we consider the minimum sum of squares model of Späth; [18]. This problem was mathematically formulated by Lau *et al.* as follows:

$$Min\ e = \sum_{i=1}^{n}\sum_{k=1}^{K} z_{ik}(y_i - \sum_{j=1}^{m} b_{jk}x_{ij} + b_{0k})^2 \tag{1}$$

$$Subject\ To$$

$$\sum_{k=1}^{K} z_{ik} = 1\ \forall i = 1 \dots n \tag{2}$$

$$z_{ik} \in \{0,1\}\ \forall i = 1 \dots n\ \forall k = 1 \dots K. \tag{3}$$

where x_{ij} and y_i are respectively the value of the variable j and the value of the dependent variable for the observation i. b_{jk} is the j^{th} regression coefficients for the cluster k and z_{ij} is a binary variable that equals 1 if and only if the observation i belongs to cluster k. n is the number of observations, m the number of independent variables considered and K the number of clusters. The objective (1) is to minimize the sum of squared prediction errors computed for each observation using the equation of the corresponding cluster. Constraints (2) ensures each observation is assigned to exactly one cluster. From the optimization point

of view, this problem has real valued variables corresponding to the regression aspect of the problem and binary variables corresponding to its clustering aspect.

3 Resolution approach: the Variable Neighborhood Search

The resolution approach we propose here is based upon the Variable Neighborhood Search (VNS) metaheuristic [10, 17] as it proves a good capability to handle combinatorial optimization problems. It has been successfully applied to various kind of combinatorial optimization problems such as vehicle routing [2], graph theory [5] or, more related to clustering, location problems [8]. In clustering, it was applied to the minimum sum of squared errors problem, similar to $k-means$, global $k-means$ [11] or $J-means$ [9].

Except the local search LS and the perturbation P^k of magnitude k, VNS only needs the parameters k_{max} and the stopping criterion to be defined. For this study, we used $k_{max} = K - 1$ and the stopping criterion was the total CPU time elapsed.

The algorithm is the following:

1. Initialization:
   - construct an initial solution S by randomly assigning each observation to a cluster
   - for each cluster k, compute coefficients b_{jk} and b_{0k} of the regression model fitting its data
   - set the perturbation magnitude k to 1
   - set the best solution $S^* \leftarrow S$ to the initial solution and the best objective value e^* to the corresponding total error.
2. Repeat until the stopping condition is met
   (a) let $S' = LS(S)$ be the solution obtained by applying a local search to the current solution S, e and e' being the errors respectively corresponding to solution S and S'.
   (b) If $e' < e^*$ (the best known solution is improved)
   $k \leftarrow 1$, $S^* \leftarrow S'$ and $e^* \leftarrow e'$
   else
   $k \leftarrow k\ mod(k_{max}) + 1$
   (c) $S \leftarrow S^*$
   (d) $S \leftarrow P^k(S)$ (apply a perturbation of magnitude k to S).

3.1 Local Search

The local search used is based upon the principle of the alternate search used in the *K-means* algorithm and involves two main steps:

1. Compute the regression model corresponding to each cluster.
2. Move observations to the cluster that best fits them; if no observation needs to be moved, the local search is complete; otherwise go back to 1.

This so called alternate local search alternatively optimizes the variables b_{jk} (step 1) and z_{ik} (step 2). It is very fast but unfortunately leads to a solution that strongly depends upon the initial solution. It therefore could not be used alone if one needs correct results.

3.2 Perturbation schemes

The VNS algorithm was implemented using the two perturbations $P^k(S)$ described below.

- **Merge :** Apply k times the following:
  1. Choose randomly 2 clusters,
  2. assign randomly all the observations in these 2 clusters to one or the other cluster.
  3. Apply the alternate local search to a restricted version of the problem consisting in only these 2 clusters.
- **Split :** Apply k times the following:
  1. Choose randomly a cluster,
  2. Assign randomly all its observations to other clusters.
  3. Choose randomly another cluster,
  4. Split this cluster in two using a random solution and a local search.

The last step of each perturbation scheme consists in optimizing a restricted problem, *i.e.,* the objects not bellonging to any of the two considered clusters are not affected.

As a benchmark, the *multistart* algorithm which consists in replacing VNS's perturbation by a new random solution was also implemented.

4 Numerical experiments

In order to test the VNS algorithm, we compared the implementations using the *split* and the *merge* perturbation schemes to the *multistart* using the same local search.

4.1 The data used to test the algorithm

Unfortunately, the data used for previous experiments on clusterwise linear regression are not publicly available. It was therefore not possible to compare our results with the other algorithms. We thus decided to use Monte Carlo simulation as a first way to evaluate the algorithm and then to apply it to a real data that was suitable for clusterwise regression and freely available. The so called "housing" dataset from UCI repository [14] was used. This dataset consists in 506 observations with 14 continuous variables.

4.2 Monte Carlo simulation

The goal of this simulation is to explore the importance of each characteristic of the data such as the number of observations n, the dimension m, the number of clusters K and the noise which is related to the parameter s in the data generation.

The data generation (x_{ij}, y_i) process was the following given n, K, m and s the standard deviation of residual.

1. Generate the models
   - for $k = 1$ to K do: let $c_{jk}\, \forall j = 1 \dots m$ be a uniform 0-10 variable.
     Let c_{0k} be a uniform 0-50 variable.
2. Generate the data
   - For $i = 1$ to n do:
     - Let $x_{ij}\, \forall j = 1 \dots m$ be a uniform 0-10 random variable.
     - Choose randomly k the cluster to which the observation i belongs.
     - Let $y_i = c_{0k} + \sum_{j=1}^{m} c_{jk} x_{ij} + e$ where e is a $N(0, s)$ random variable.

Each algorithm was run on 10 data sets for each set of parameters (the same data was used for all algorithms) for a maximum CPU time of 5 minutes on a sun computer with opteron 250 processor (2.4 GHz) and 8Go RAM running the solaris 10 operating system. The average SSE among 10 runs for each perturbation type as well as multistart is recorded and displayed in the following sections in order to show the sensitivity of each strategy to different parameters. For each dataset generated, the minimum SSE obtained among the *merge, split* and *multistart* strategies is recorded and the average of these minimums is referred to as *minimum*. This average minimum is used as reference when drawing the graphics.

4.2.1 Impact of the number of observations

In order to evaluate the importance of the number of observations, the problems were generated with the following parameters: $K = 10$, $m = 10$ and $s = 50$ and n varying from 5000 to 30000. Each configuration was replicated and the average value obtained from the 10 runs is displayed on Table 1 and represented on Figure 1.

Table 1: Average and Minimum SSE for 10 runs according to the number of observations and perturbation mode.

n	merge	split	multistart	minimum
5000	307328	248931	436981	248931
10000	676801	518502	804222	518502
20000	1548192	1113761	1659555	1113761
30000	2385738	1813867	2421368	1809383

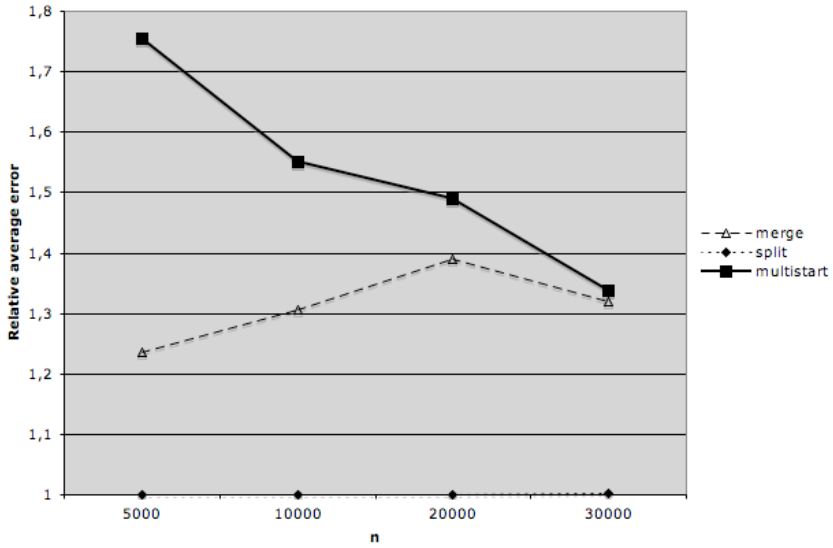

Figure 1: Relative SSE using *minimum* as reference when varying n.

4.2.2 Impact of the dimension

In order to evaluate the importance of the dimension, the problems were generated with the following parameters: $n = 5000$, $K = 10$, $s = 50$ and m varying from 5 to 30. Each

configuration was replicated and the average value obtained from the 10 runs is displayed on Table 2 and represented on Figure 2.

Table 2: Average and Minimum SSE for 10 runs according to the dimension m and perturbation mode.

m	merge	split	multistart	minimum
5	307328	248931	436981	248931
10	502750	314573	676736	314573
20	599359	423232	691113	417074
30	571127	560251	601441	523695

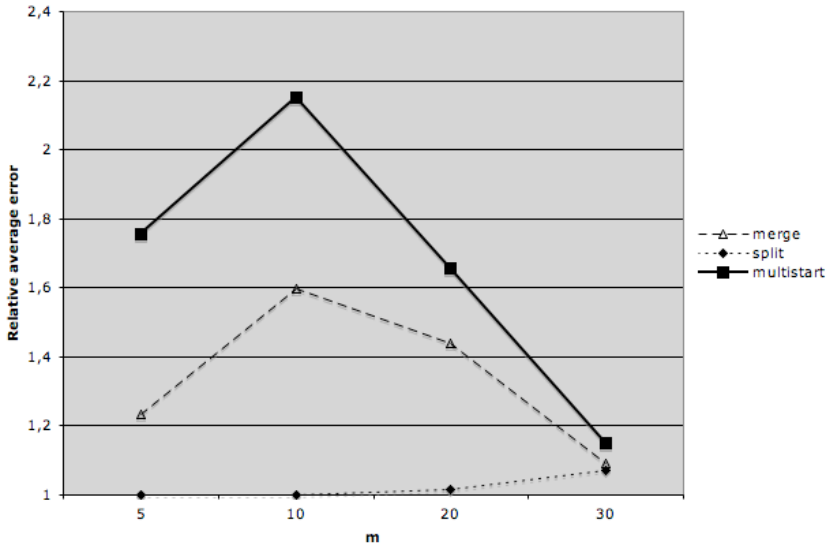

Figure 2: Relative SSE using *minimum* as reference when varying m.

4.2.3 Impact of the number of clusters

In order to evaluate the importance of the number of clusters K, the problems were generated with the following parameters: $n = 5000$, $m = 5$, $s = 50$ and K varying from 5 to 20. Each configuration was replicated and the average value obtained from the 10 runs is displayed on Table 3 and represented on Figure 3.

Table 3: Average and Minimum SSE for 10 runs according to the number of clusters and perturbation mode.

K	merge	split	multistart	minimum
5	710068	710039	806867	710013
10	303748	248930	436981	248931
15	221448	135403	296804	135403
20	158968	85506	196035	85506

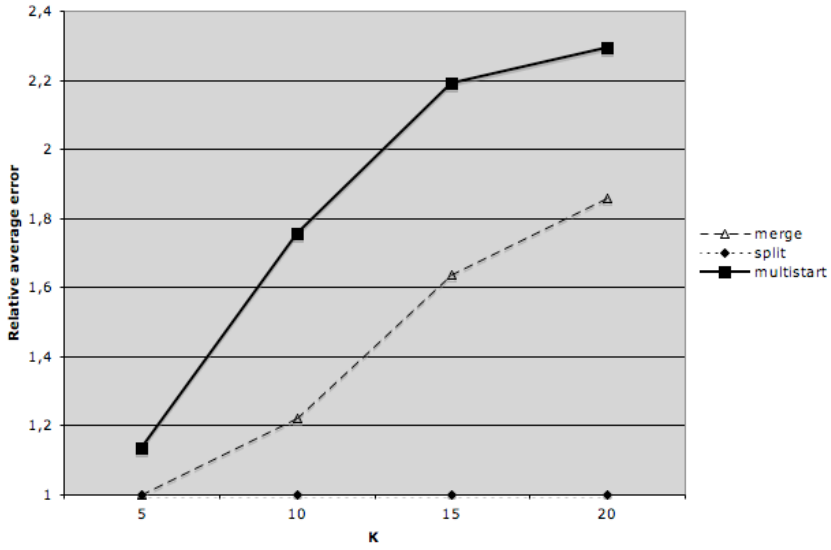

Figure 3: Relative SSE using *minimum* as reference when varying K.

4.2.4 Impact of the noise

In order to evaluate the importance of the noise in data, the problems were generated with the following parameters: $n = 5000$, $m = 5$, $K = 10$ and the noise level s varying from 0 to 100. Each configuration was replicated and the average value obtained from the 5 runs is displayed on Table 4 and represented on Figure 4.

4.3 The "housing" data

Table 5 shows the average SSE among the 10 successive runs for each perturbation used in VNS as well as the multistart implementation. In order to better compare the performance

Table 4: Average and Minimum SSE for 10 runs according to the noise level s and perturbation mode.

s	merge	split	multistart	minimum
0	0	0	831	0
10	87488	80317	106488	80316
20	148554	130133	204989	130133
50	307374	248931	436981	248931
100	676142	543922	959074	543922

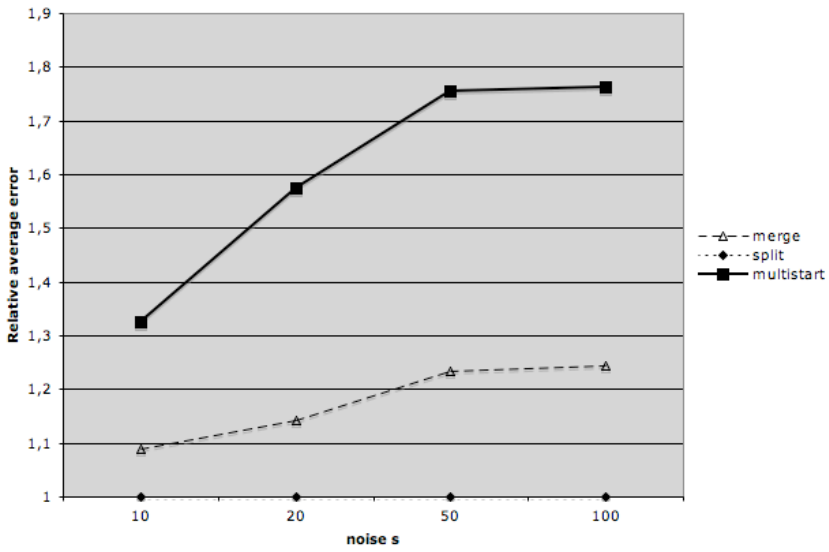

Figure 4: Relative SSE using *minimum* as reference when varying s.

of the various strategies, Figure 5 represent the average error value among the 10 runs for each strategy compared to the best solution obtained for the same problem regardless the method used.

This figure is directly drawn from data on Table 5. From Figure 5, we notice that on 10 clusters, on average, the objective value obtained by *multistart* is about twice the value obtained using VNS and the *merge* perturbation.

Table 5: Average SSE for 10 runs of 300 sec each according to the number of clusters and the perturbation method used. The minimum found is provided as reference.

Number of clusters	Merge	Split	Multistart	Min
2	3246.07	3248.14	3248.33	3232.24
3	1404.18	1404.82	1433.2	1374.17
4	658.59	676.31	769.37	625.44
5	349.2	376.7	457.13	300.16
7	126.49	159.37	198.86	104.24
10	37.76	52.69	70.98	32.71

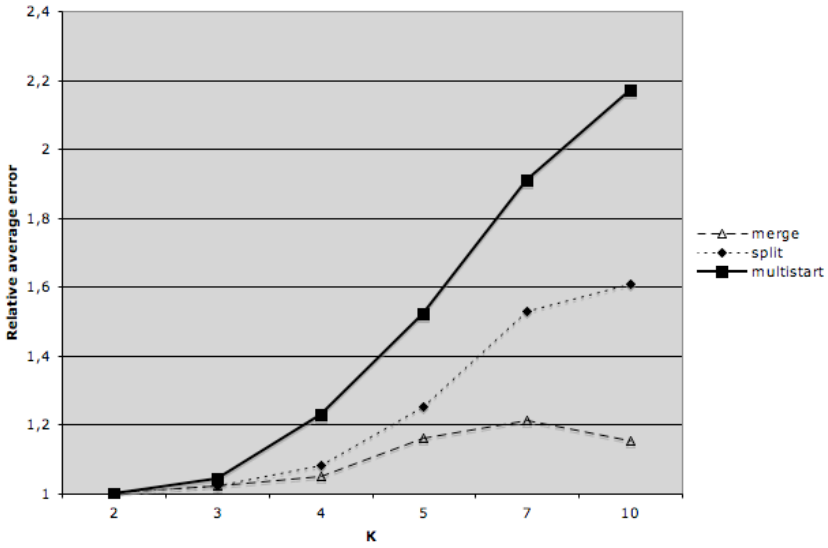

Figure 5: Relative average (among 10 runs) errors obtained with each searching strategy. The "Merge" strategy is used as reference

5 Conclusion

The first remark from these numerical experiments is that VNS systematically performs better than multistart. For some parameters or instances, the *multistart* is more than 2 times worse than VNS and the *split* perturbation scheme systematically performs much better than *merge*.

Unexpectedly, the monte carlo simulation shows that clusterwise regression is to be easier when the number of observations increases. This phenomenon is probably due to the robustness of the regression model that is less likely to change when moving objects from a cluster to another. We notice also, which is not so surprising, that increasing the dimension make the problem easier. This remark would probably not hold if the error level was increased together with the dimension. Indeed, increasing the dimension while leaving the noise level constant makes more separated clusters and the problem is then easier to solve. On the contrary, increasing either the noise level or the number of clusters increases the difficulty of the problem. Indeed, increasing the noise level or the number of clusters makes the optimization more difficult since observations from different clusters are not as well separated. It is not a surprise that more the clusters are separated, more the problem is easy to solve. In the case of the number of clusters, this phenomenon is even stronger due to the combinatorial complexity which is also increased.

References

[1] J.M. Aurifeille. A bio-mimetic clusterwise regression algorithm for consumer segmentation. In *Advances in Computational Management*, pages 145–163, 1998.

[2] I. Braysy and M. Gendreau. Vehicle routing problem with time window, part ii: Metaheuristics. *Transportation Science*, 39:119–139, 2005.

[3] M.J. Brusco, J. D Cradit, and S. Stahl. A simulated annealing heuristic for a bicriterion partitioning problem in market segmentation. *Journal of Marketing Research*, 39:99–109, 2002.

[4] M.J. Brusco, J. D Cradit, and A. Tashchian. Multicriterion clusterwise regression for joint segmentation settings: An application to customer value. *Journal of Marketing Research*, 40:225–234, 2003.

[5] G. Caporossi and P. Hansen. Variable Neighborhood Search for Extremal Graphs. 1. The Autographix System. *Discrete Mathematics*, 212:29–44, 2000.

[6] W.S. DeSarbo. A maximum likelyhood methodology for clusterwise linear regression. *Journal of Classification*, 5:249–282, 1988.

[7] W.S. DeSarbo, R.L. Olivier, and A. Rangaswamy. A simulated annealing methodology for clusterwise linear regression. *Psychometrika*, 54:707–736, 1989.

[8] P. Hansen and N. Maldenović. Variable neighborhood search for the p-median. *Location Science*, 5:207–226, 1997.

[9] P. Hansen and N. Mladenović. J-means: a new local search heuristic for minimum sum of squares clustering. *Pattern Recognition*, 34:405–413, 2001.

[10] P. Hansen and N. Mladenović. Variable neighborhood search: Principles and applications. *European Journal of Operations Research*, 130:449–467, 2001.

[11] P. Hansen, E. Ngai, B.K. Cheung, and N. Mladenović. Analysis of global k-means, an incremental heuristic for minimum sum-of-squares clustering. *Les cahiers du GERAD*, pages 1–23, 2002. Submitted.

[12] C. Hennig. Fixed Point Clusters for linear regression: computation and comparison. *Journal of Classification*, 19:249–276, 2002.

[13] C. Hennig. Clusters, outliers, and regression: fixed point clusters. *Journal of Multivariate Analysis*, 86:183–212, 2003.

[14] S. Hettich, C.L. Blake, and C.J. Merz. Uci repository of machine learning databases, 1998. URL: http://www.ics.uci.edu/∼mlearn/MLRepository.html, University of California, Irvine, Dept. of Information and Computer Sciences.

[15] Kin-Nam Lau, Pui-Iam Leung, and Ka-Kit Tse. A mathematical programming approach to clusterwise regression. *European Journal of Operations Research*, 116:640–652, 1999.

[16] J.B. MacQueen. Some Methods for classification and Analysis of Multivariate Observations. *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press* 1:281–297, 1967.

[17] N. Mladenović and P. Hansen. Variable neighborhood search. *Computers and Operations Research*, 24:1097–1100, 1997.

[18] H. Späth. Clusterwise linear regression. *Computing*, 22:367–373, 1979.

[19] H. Späth. A fast algorithm for clusterwise linear regression. *Computing*, 29:175–181, 1981.

[20] H. Späth. Clusterwise linear least absolute deviations regression. *Computing*, 37:371–378, 1986.

[21] J,-B. Steenkamp and M. Wedel Fuzzy Clusterwise Regression in Benefit Segmentation: Application and Investigation into its Validity *Journal of Business Research*, 26:237–249, 1993.

[22] M. Wedel and C. Kistemaker. Consumer benefit segmentation using clusterwise linear regression. *International Journal of Research in Marketing*, 6:45-49, 1989.

[23] M. Wedel and J.-B. Steenkamp. A clusterwise regression method for simultaneous fuzzy market structuring and benefit segmentation. *Journal of Marketing Research*, 27:385–396, 1991.