Scalable and Near-Optimal ε-tube Clusterwise Regression

Aravinth Chembu[1], Scott Sanner[1], and Elias B. Khalil[1]

Department of Mechanical and Industrial Engineering, University of Toronto, Toronto
aravinth.chembu@mail.utoronto.ca, {ssanner,khalil}@mie.utoronto.ca

Abstract. Clusterwise Regression (CLR) methods that jointly optimize clustering and regression tasks are useful for partitioning data into disjoint subsets with distinct regression trends. Due to the inherent difficulty in simultaneously optimizing clustering and regression objectives, it is not surprising that existing optimal CLR approaches do not scale beyond 100s of data points. In an effort to provide more scalable and optimal CLR methods, we propose a novel formulation of the problem that takes inspiration from ε-tubes in Support Vector Regression (SVR). The advantage of this novel formulation, which aims to assign data points to clusters in order to minimize the largest ε-tube that encapsulates the regressed data, is that it admits an optimal MILP formulation. Furthermore, given that each constraint in our formulation corresponds to a single data point, we propose an efficient row generation solution that can optimally converge for the full dataset while only requiring optimization over a subset of the data. Our results on a variety of synthetic and benchmark real datasets show that our Clusterwise Regression MILP formulation provides near-optimal solutions up to 100,000 data points and the smallest data-encapsulating ε-tubes among CLR alternatives.

Keywords: Clusterwise Regression · Row Generation · Mixed-integer linear programming.

1 Introduction

Clusterwise Regression (CLR) is a fundamental task in Machine Learning that jointly optimizes for clustering and regression tasks, where the data is partitioned into several clusters, each group fit by a regression plane, such that the overall regression error is minimized. CLR models find applications in a plethora of fields such as social science [17], marketing analysis [9], and climate modeling [1].

Traditionally, CLR models entailed jointly optimizing for clustering with the squared error objective for regression, as proposed in seminal work on CLR [20]. Existing greedy algorithms for CLR are sensitive to initialization and provide only locally optimal results, thus limiting clustering quality and reproducibility. Moreover, the classical CLR model [20] was recently shown to be NP-hard [18] and considered a very difficult problem to solve [14]. Thus, optimally solving for 100s of data observations is challenging [4, 7, 5, 6], even with synthetic examples.

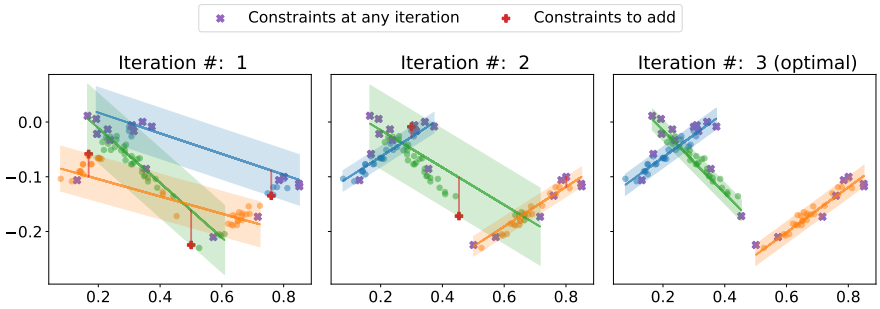


Fig. 1: We show the ε-tube CLR solution (right) with an illustrative example. Additionally, we demonstrate our row generation algorithm, where we start (left) with an initial set of points (denoted with $\times$) and run two iterations adding 3 constraints per iteration denoted with $+$ that are the farthest from the regression lines until we reach the optimal result in the third iteration. We observe that convergence to the optimal solution does not require ε to monotonically decrease.

In this work, we propose a novel approach to CLR that is inspired by the ε-tubes (or margins) that correspond to absolute values of the regression residuals in Support Vector Regression (SVR) [22, 12]. In this formulation, we minimize the largest ε-tube across all clusters that encapsulates the regressed data. Such a formulation is inherently insensitive to cluster size imbalance since we only measure the worst-case residual. In addition, a core computational advantage of this formulation is that it can be expressed and optimally solved as a Mixed Integer Linear Program (MILP) that supports an efficient row (constraint) generation strategy. We illustrate this iterative row generation process in Fig 1 demonstrating the evolution of data point (re)assignments to three clusters and their corresponding shaded ε-tube at each iteration until optimality. It is important to note that this solution only generated the most-violated constraints for all data points (most often near the ε-tube boundaries, by definition) since the remaining data lie within tube boundaries and automatically satisfy the optimality criteria.

Leveraging our novel MILP formulation and row generation solution can thus solve ε-tube CLR using a subset of the data (while guaranteeing optimality w.r.t. all data), hence providing near-optimal results for up to 100,000 data points in comparison to other CLR formulations and solutions that cannot scale optimally beyond 100s of data points. We provide experiments on a variety of synthetic datasets (varying number of data points, dimensionality, clusters, and cluster imbalance) and 10 benchmark real datasets to demonstrate our algorithm's ability to reach the smallest ε-tube clusters when compared with several baselines.

2 Related Work

Several greedy algorithms have been proposed to solve the classical CLR problem, including exchange algorithms proposed in the pioneering works of Späth [20,

21], simulated annealing in [10], mathematical programming-based heuristics [3, 2, 13], and an Expectation-Maximization [8] type methodology in a recent work called k-plane clustering [16], which is analogous to the k-means algorithm [15]. In contrast, exact approaches involve the use of mixed-integer optimization [4, 14, 7], repetitive branch-and-bound methods [5], and column generation approaches [6, 18]. However, these algorithms only scale up to 100s of observations in low dimensions, even with synthetic datasets and typically less than 5 clusters. Moreover, numerous alternatives for the L_2 regression loss of CLR have been presented, like the more robust L_1 loss [4, 19, 2]. More recently, SVR for regression was used for the CLR problem [13]; however, the key difference with our approach is that we directly minimize the ε-tubes while they solve for pure SVRs in each cluster (by minimizing the slacks) with ε being a hyperparameter; further, they do not provide any optimality guarantees.

3 Optimal CLR with ε-tube Objective

3.1 Reduction of ε-tube CLR to a MILP

Our ε-tube objective for CLR minimizes the maximum regression residual for every point across all the clusters. More formally, consider that we have n observations $(\boldsymbol{x_i}, y_i)$ with d features in the dataset $(\boldsymbol{X}, y) \in \mathbb{R}^{n\times(d+1)}$ where $i \in N = \{1, ..., n\}$. The main goal in CLR is to find one regression plane for each of the k clusters (C_j), where the regression coefficients for the jth cluster are represented by weights $\boldsymbol{w_j} \in \mathbb{R}^d$ and bias $b_j \in \mathbb{R}$ for $j \in K = \{1, ..., k\}$. We will use $\boldsymbol{w}$ and b without cluster indices to refer to the collection of k regression plane parameters. Each data point is assigned to exactly one cluster, similar to a hard-partitioning setting in unsupervised clustering. We introduce binary variables c_{ij} that denote whether point i is assigned to cluster C_j ($c_{ij} = 1$) or not ($c_{ij} = 0$), thus enabling us to formulate a min-max mixed integer optimization problem. Using this notation, we define our **first key novel contribution of the ε-tube CLR objective**:

$$\min_{\boldsymbol{w},b,c} \max_{j\in K} \max_{i\in N} |y_i - \boldsymbol{w_j}^\intercal \boldsymbol{x_i} - b_j| \cdot c_{ij} \tag{1}$$

Here, we first observe that we can further reduce this ε-tube CLR objective to a bi-level optimization problem with the introduction of a new variable ε, which takes the value of the maximum residual from all points across the k regression planes (through the first constraint in (2)).

$$\begin{aligned}
&\min_{\boldsymbol{w},b,c} \ \varepsilon \\
&\text{s.t. } \varepsilon = \max_{j\in K} \max_{i\in N} |y_i - \boldsymbol{w_j}^\intercal \boldsymbol{x_i} - b_j| \cdot c_{ij} \\
&\quad \sum_{j=1}^{k} c_{ij} = 1, \ \ i \in N; \ \ w_{j,1} < w_{j+1,1}, \ j \in K\backslash\{k\}; \\
&\quad \boldsymbol{w_j} \in \mathbb{R}^d, \ \ b_j \in \mathbb{R}, \ \ j \in K; \ \ c_{ij} \in \{0, 1\}, \ \ i \in N, \ j \in K;
\end{aligned} \tag{2}$$

In this bi-level problem, indicator variables c_{ij} ensure we only capture residuals for the regression plane (cluster) a point is assigned to. Moreover, constraints $\sum_{j=1}^{k} c_{ij} = 1$ guarantee every point in the dataset is assigned to exactly one cluster. We also add symmetry-breaking constraints of the form $w_{j,1} < w_{j+1,1}$ that enforce the first dimension of the regression coefficients across clusters to be in increasing order. This guarantees that we choose exactly one solution out of the $k!$ possible permutations with the same optimal value. A key observation is that we can *remove both* max*'s* from the first constraint and rewrite it to $\varepsilon \geq |y_i - \boldsymbol{w_j}^\intercal \boldsymbol{x_i} - b_j| \cdot c_{ij},\ i \in N,\ j \in K$. While this constraint ensures that ε takes a value greater than or equal to the max prediction error, the minimization criteria in (2) enforces equality! Using this elegant transformation and indicator constraints to encode the product of regression residual and c_{ij} yields our **second key novel contribution of ε-tube CLR formulated as a MILP**:

$$
\begin{aligned}
\min_{\boldsymbol{w},b,c}\ & \varepsilon \\
\text{s.t.}\quad & c_{ij} = 1 \implies \varepsilon \geq |y_i - \boldsymbol{w_j}^\intercal \boldsymbol{x_i} - b_j|,\ \ i \in N,\ j \in K \\
& \sum_{j=1}^{k} c_{ij} = 1,\ \ i \in N;\ \ w_{j,1} < w_{j+1,1},\ j \in K \backslash \{k\}; \\
& \boldsymbol{w_j} \in \mathbb{R}^d,\ \ b_j \in \mathbb{R},\ \ j \in K;\ \ c_{ij} \in \{0,1\},\ \ i \in N,\ j \in K;
\end{aligned}
\tag{3}
$$

3.2 Row Generation Methodology

The final pure-MILP formulation for our ε-tube objective in (3) allows for the use of efficient branch-and-bound strategies through state-of-the-art MILP solvers. However, the large number of binary variables and constraints may present a challenge for MILP solvers. A possible solution would be to reduce the number of variables and constraints of the model without affecting its correctness.

In problem (3), we observe that we minimize a single variable ε whose value is governed through the $n \times k$ indicator constraints. If points that have large residuals w.r.t. the regression coefficients for the optimal result can be known a priori, the *indicator constraints corresponding to the points that have much smaller residuals can be neglected*. Neglecting these observations will not change the optimal value as ε is already larger than the residuals from these points.

This crucial insight can be leveraged in our **third novel contribution of an efficient row (constraint) generation MILP solution** by starting with a small subset of observations (with their associated variables and constraints) in a reduced version of (3) we term main problem (MP). Given an optimal solution to MP, we check whether it is optimal for the full problem (3). This check can be performed through a sub-problem (SP) that identifies points that have residuals larger than that of the current solution of the MP. In essence, the SP identifies the *most-violated constraints* corresponding to the points with largest residuals not yet included in the MP. These most-violated constraints can then be added

to the MP. This procedure can be iteratively executed until the SP ensures that all observations have residuals smaller than that of the current optimal solution. In such a case, we have found an optimum for the full problem (3). Convergence to optimality is guaranteed in finite time since in the (unlikely) worst case this happens when we generate all rows (constraints) and recover the full problem (3).

Algorithm 1 Row generation

Input $(\boldsymbol{x_i}, y_i)$, k

1: $\varepsilon^* \leftarrow 0,\ \hat{\varepsilon} \leftarrow \infty$
2: $I \neq \emptyset,\ \text{CONS} \neq \emptyset,\ \boldsymbol{w_j}, b_j \leftarrow$ Initialize ▷ *Initial constraints* CONS *for points* I
3: $\varepsilon^*, \boldsymbol{w_j^*}, b_j^*, c_{ij}^* \leftarrow$ Solve MILP with CONS
4: $\hat{c_{ij}} \leftarrow \{\mathbb{1}_{j=\hat{j}} | \hat{j} = \arg\min_j |y_i - \boldsymbol{w_j^{*\intercal}} \boldsymbol{x_i} - b_j^*|\},$ ▷ *Assign points* $i \in N$
5: $\hat{\varepsilon}, I, \text{CONS} \leftarrow$ Add-Constraints()
6: **if** $\hat{\varepsilon} > \varepsilon^*$ **then**
7: **go to** line 3 ▷ *Re-solve MILP with augmented constraints set* CONS
8: **end if**
9: **return** $\varepsilon^*, \boldsymbol{w_j^*}, b_j^*, \hat{c_{ij}}$ ▷ *Optimal*

Algorithm 2 Add-Constraints

Input $(\boldsymbol{x_i}, y_i), \boldsymbol{w_j^*}, b_j^*, \hat{c_{ij}}, \text{CONS}, \varepsilon^*$

$\hat{\varepsilon} = \max_{i \in N,\ j \in K} \{|y_i - \boldsymbol{w_j^{*\intercal}} \boldsymbol{x_i} - b_j^*| \cdot \hat{c_{ij}}\}$
▷ *Check if* ε^* *is the max residual*
if $\hat{\varepsilon} > \varepsilon^*$ **then**
for $j \in K$ **do**
$I_{add} \leftarrow \{\arg\max_i |y_i - \boldsymbol{w_j^{*\intercal}} \boldsymbol{x_i} - b_j^*| \cdot \hat{c_{ij}}\}$ ▷ *Find largest residual*
$I \leftarrow I \ \cup \ I_{add}$
end for
CONS $\leftarrow$ CONS $\cup \ \{c_{ij} = 1 \implies \varepsilon \geq |y_i - \boldsymbol{w_j^\intercal} \boldsymbol{x_i} - b_j|, i \in I_{add}, j \in K\}$
end if
return $\hat{\varepsilon}, I, \text{CONS}$

We formalize the above row (or constraint) generation procedure through Algorithms 1 and 2 that primarily perform the MP and SP tasks, respectively. In Algorithm 1, we initialize our model with a small subset of observations in $I \subset N$, and their corresponding variables c_{ij} and constraints in C. We solve the MP with the reduced formulation of (3) using a MILP solver to obtain the optimal value ε^*. The coefficients for the k regression planes are used to assign a point $i \in N$ to the cluster to which it has the lowest prediction error (line 4 in Algorithm 1). This is a crucial step in our algorithm as we use the cluster assignment information to then compute the maximum residual stored in $\hat{\varepsilon}$ (line 1 in Algorithm 2) for all points w.r.t. the coefficients obtained from the MP. If $\hat{\varepsilon} > \varepsilon^*$, we are yet to reach the optimal solution. Hence, we identify the most violating constraints (if one exists) for each of the k clusters through the SP (line 7 in Algorithm 2) and add them to the MP. We stop iterating between the MP and SP when $\hat{\varepsilon} = \varepsilon^*$, i.e., when no more points $i \in N$ incur residuals larger than the current objective, implying optimality w.r.t. all constraints.

4 Empirical Evaluation

We first study the properties of our ε-tube CLR objective and comparatively evaluate our solution on synthetic and real datasets. We use a hyphenated three-part naming convention: (1) clustering criteria (k-means or direct point-to-cluster assignment), (2) regression loss (least squares, SVR, or ε-tube), and (3) optimization method (independent, iterative, or MILP). We compare the following methods: (i) **km-ls-indep**: k-means (km) followed by least-squares

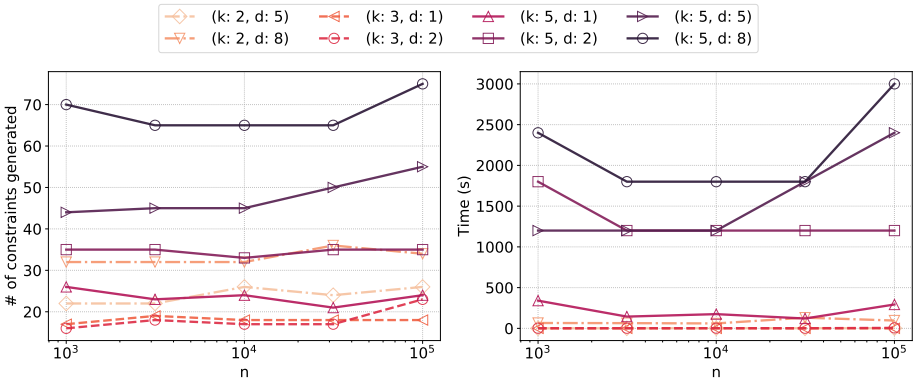


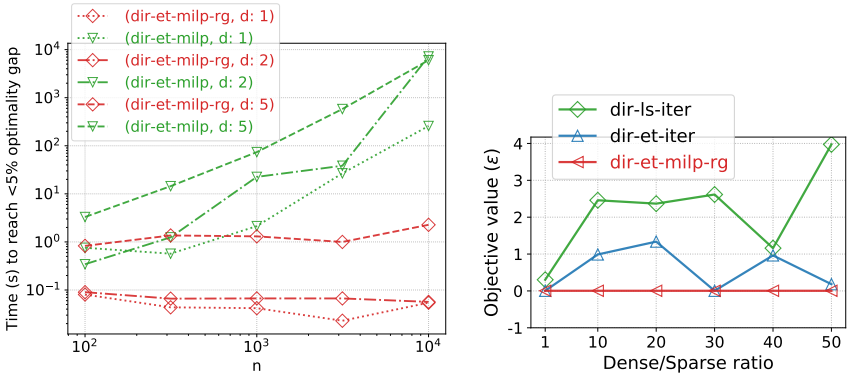
Fig. 2: # Constraints generated and solution time to 5% optimality gap for **dir-et-milp-rg** with number of clusters k and dimensionality d vs. amount of data n for the *klr-data*. # Constraints and time only increase marginally as n increases.

(ls) regression where the clustering and regression take place independently; (ii) **km-svr-indep**: k-means followed by SVR [22]; (iii) **km-et-indep**: k-means followed by optimizing our ε-tube objective; (iv) **dir-ls-iter**: k-planes [16] algorithm with least squares regression where the clustering is directly (dir) assigned by an iterative procedure (iter); (v) **dir-et-iter**: a novel k-means inspired approximate iterative algorithm to optimize for our ε-tube (et) objective — here we iterate between (a) finding the best set of hyperplanes (compute $\boldsymbol{w_j}, b_j, j \in K$) per cluster given a cluster assignment for all points and (b) re-assigning points to clusters (update c_{ij}) such that each point has the lowest prediction error when assigned to that cluster; (vi) **dir-et-milp**: our novel full MILP from Eq (3); (vii) **dir-et-milp-rg**: our novel full MILP with *row generation* (rg).

Across our experiments, similar to the approach followed in [16, 5, 6, 18], we primarily focus on comparing the ε value, i.e., the maximum residual among all clusters, since providing an optimal algorithm for the ε-tube objective is our key contribution; we only include non-"et" methods for relative comparison with other common CLR methods. All experiments were run on Google Colab in the standard CPU setting (at 2.3 GHz and 32 GB memory) with Gurobi 9.5.2. All code is available at https://github.com/Aravinthck/CLR-epsTube.

4.1 Synthetic Dataset Experiments

Scalability Our **dir-et-milp-rg** algorithm depends on solving a MILP at every iteration, which leads us to ask how well it scales vs. the dimensionality, number of clusters, and amount of data. To this end, we evaluate the scalability of **dir-et-milp-rg** when the ground truth clusters are recovered by constructing synthetic datasets (called *klr-data*) similar in spirit to [5, 6, 16] where we choose $k \in \{2, 3, 5\}$ regression parameters uniformly at random with $d \in \{1, 2, 5, 8\}$ features and normal error in the regression. The feature vectors are extracted



(a) We show that run-times for the row generation method are a fraction of dir-et-milp (as we increase n, d with $k = 2$) where at $n = 10^4$, the difference was 2s to 6030s.

(b) We show that ε-tube objective is agnostic to cluster imbalance and finds the true ground-truth clusters, unlike the least squares objective.

Fig. 3: Comparative experiments of different algorithms for synthetic datasets.

from Gaussian clusters with observations n varied from 10^3 to 10^5 points. Fig 2 shows that the number of constraints generated to reach an optimality gap of 5% only increases marginally with the number of data points n. Similar trends are observed for the reported run-time for these experiments (cf. Fig 2, right). These empirical results further suggest that only a small fraction of the observations are needed. For example, only ≈ 75 observations are needed to solve the problem to 5% optimality gap with 10^5 observations (top-right point in Fig 2, left).

Performance gain for row generation We now compare the run-times of our row generation method **dir-et-milp-rg** and full-MILP solution **dir-et-milp** in Fig 3a where we terminate on reaching the 5% optimality gap with both methods. With the *klr-data*, we experiment with n ranging from 100 to 10^4, $d \in \{1, 2, 5\}$ and $k = 2$ to ensure that **dir-et-milp** finishes in under 10^4 seconds. Here, **dir-et-milp-rg** strictly dominates its **dir-et-milp** counterpart in all cases and by more than 3 orders of magnitude for $n = 10^4$. What is more remarkable is that **dir-et-milp-rg** remains relatively flat as n increases in contrast to **dir-et-milp** that grows exponentially with n (i.e., as evidenced by the superlinear trend on this log-log plot).

Robustness to cluster imbalance The optimal solution of our ε-tube CLR should be insensitive to imbalance in the number of points in the clusters because we only measure the worst-case residual. This alleviates the need for a higher concentration of clusters in areas where the data is denser. We validate this claim with imbalanced clusters with $n = 10^4$, $d = 1$, and $k = 3$. One of the clusters was designed to be dense, and the others were sparse, with the ratio

Table 1: Comparative evaluation of objective values ε from (3) on 10 datasets shown in the columns and ordered by number of data points (n).

index	Iris	Autompg	Ceosalaries	Boston	Airfoil	Redwine	Abalone	Whitewine	Powerplant	Protein
# data (n)	150	392	500	506	1503	1599	4177	4898	9568	45730
# dimension (d)	4	7	1	13	5	11	7	11	4	9
# clusters (k)	3	3	6	2	4	3	3	3	5	2
km-ls-indep	0.8	1.5	10.64	2.34	2.77	3.04	4.56	3.98	2.71	3.94
km-svr-indep	0.94	1.38	5.69	2.02	2.3	2.81	3.57	3.39	1.99	1.72
km-et-indep	0.63	1.06	5.68	1.27	2.2	2.24	2.83	2.98	1.62	1.72
dir-ls-iter	0.48	1.0	2.5	1.76	1.21	2.34	2.82	3.05	1.76	2.16
dir-et-iter	0.32	0.59	0.7	0.99	0.86	1.55	1.75	1.86	0.53	1.67
dir-et-milp	0.24	0.42	0.64	0.78	0.77	1.13	1.75	1.82	1.45	1.72
dir-et-milp-rg	**0.22**	**0.38**	**0.48**	**0.67**	**0.59**	**0.62**	**0.88**	**1.03**	**0.3**	**0.82**

of points given by *dense/sparse ratio*. From Fig 3b, it is evident that all three methods came close to recovering the optimal *balanced* clusters (dense/sparse ratio = 1). However, for the *imbalanced* cases (ratio > 1), only **dir-et-milp-rg** identified the true ground truth cluster in all cases, while **dir-et-iter** struggled with local optimality of its greedy method (reaching optimality once) and **dir-ls-iter** appears to further suffer from the sensitivity of its least sum-of-squared loss to imbalanced clusters.

4.2 Real Dataset Experiments

We now benchmark our model with 10 commonly used CLR datasets found in [13, 16, 11, 2]. In Table 1, we report the mean objective value ε obtained after 10 independent runs for the various greedy baselines compared with the full-MILP solution **dir-et-milp**, which is run for the same amount of time that it takes **dir-et-milp-rg** to reach the 5% optimality gap. We use the best value for k reported in the mentioned peer-reviewed works for each dataset. The ε values obtained for **dir-et-milp-rg** are better than all other methods, often substantially. Interestingly, we note that the greedy **dir-et-iter** that we have proposed outperforms **dir-et-milp** on several datasets. Moreover, we observed that the lower bound returned by Gurobi for **dir-et-milp** was zero and did not increase within the restricted time limit for these experiments. Improved formulation for **dir-et-milp** (and **-rg**) and tightening of lower bounds is an interesting direction for potential future work. However, the strictly dominant performance of **dir-et-milp-rg** underscores its ability to scale to large real datasets.

5 Conclusion

We provided a novel formulation for the ε-tube CLR problem that reduces to a MILP and further admits an efficient row generation solution. Our results on benchmark datasets make it evident that our row generation solution is much faster than solving the full MILP and that we outperform other CLR methods in terms of maximum ε-tube loss and different levels of cluster imbalance.

References

1. Bagirov, A.M., Mahmood, A., Barton, A.: Prediction of monthly rainfall in victoria, australia: Clusterwise linear regression approach. Atmospheric Research **188**, 20–29 (2017). https://doi.org/https://doi.org/10.1016/j.atmosres.2017.01.003, https://www.sciencedirect.com/science/article/pii/S0169809517300285
2. Bagirov, A.M., Taheri, S.: Dc programming algorithm for clusterwise linear l_1 regression. Journal of the Operations Research Society of China **5**(2), 233–256 (2017)
3. Bagirov, A.M., Ugon, J., Mirzayeva, H.: Nonsmooth nonconvex optimization approach to clusterwise linear regression problems. European Journal of Operational Research **229**(1), 132–142 (2013). https://doi.org/https://doi.org/10.1016/j.ejor.2013.02.059, https://www.sciencedirect.com/science/article/pii/S0377221713002087
4. Bertsimas, D., Shioda, R.: Classification and regression via integer optimization. Operations Research **55**(2), 252–271 (2007). https://doi.org/10.1287/opre.1060.0360, https://doi.org/10.1287/opre.1060.0360
5. Carbonneau, R.A., Caporossi, G., Hansen, P.: Extensions to the repetitive branch and bound algorithm for globally optimal clusterwise regression. Comput. Oper. Res. **39**(11), 2748–2762 (nov 2012). https://doi.org/10.1016/j.cor.2012.02.007, https://doi.org/10.1016/j.cor.2012.02.007
6. Carbonneau, R.A., Caporossi, G., Hansen, P.: Globally optimal clusterwise regression by column generation enhanced with heuristics, sequencing and ending subset optimization. J. Classif. **31**(2), 219–241 (jul 2014). https://doi.org/10.1007/s00357-014-9155-x, https://doi.org/10.1007/s00357-014-9155-x
7. Carbonneau, R.A., Caporossi, G., Hansen, P.: Globally optimal clusterwise regression by mixed logical-quadratic programming. European Journal of Operational Research **212**(1), 213–222 (2011). https://doi.org/https://doi.org/10.1016/j.ejor.2011.01.016, https://www.sciencedirect.com/science/article/pii/S0377221711000191
8. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society. Series B (Methodological) **39**(1), 1–38 (1977), http://www.jstor.org/stable/2984875
9. DeSarbo, W.S., Cron, W.L.: A maximum likelihood methodology for clusterwise linear regression. Journal of classification **5**(2), 249–282 (1988). https://doi.org/10.1007/BF01897167, https://doi.org/10.1007/BF01897167
10. DeSarbo, W.S., Oliver, R.L., Rangaswamy, A.: A simulated annealing methodology for clusterwise linear regression. Psychometrika **54**(4), 707–736 (1989)
11. Di Mari, R., Rocci, R., Gattone, S.A.: Clusterwise linear regression modeling with soft scale constraints. International Journal of Approximate Reasoning **91**, 160–178 (2017). https://doi.org/https://doi.org/10.1016/j.ijar.2017.09.006, https://www.sciencedirect.com/science/article/pii/S0888613X17305686
12. Drucker, H., Burges, C.J.C., Kaufman, L., Smola, A., Vapnik, V.: Support vector regression machines. In: Mozer, M., Jordan, M., Petsche, T. (eds.) Advances in Neural Information Processing Systems. vol. 9. MIT Press (1996), https://proceedings.neurips.cc/paper/1996/file/d38901788c533e8286cb6400b40b386d-Paper.pdf
13. Joki, K., Bagirov, A.M., Karmitsa, N., Mäkelä, M.M., Taheri, S.: Clusterwise support vector linear regression. European Journal of Operational Research

287(1), 19–35 (2020). https://doi.org/https://doi.org/10.1016/j.ejor.2020.04.032, https://www.sciencedirect.com/science/article/pii/S0377221720303830
14. Lau, K.n., Leung, P.l., Tse, K.k.: A mathematical programming approach to clusterwise regression model and its extensions. European Journal of Operational Research **116**(3), 640–652 (1999), https://EconPapers.repec.org/RePEc:eee:ejores:v:116:y:1999:i:3:p:640-652
15. Lloyd, S.: Least squares quantization in pcm. IEEE Transactions on Information Theory **28**(2), 129–137 (1982). https://doi.org/10.1109/TIT.1982.1056489
16. Manwani, N., Sastry, P.: K-plane regression. Inf. Sci. **292**(C), 39–56 (Jan 2015). https://doi.org/10.1016/j.ins.2014.08.058, https://doi.org/10.1016/j.ins.2014.08.058
17. Olson, A.W., Zhang, K., Calderon-Figueroa, F., Yakubov, R., Sanner, S., Silver, D., Arribas-Bel, D.: Classification and regression via integer optimization for neighborhood change. Geographical Analysis **53**(2), 192–212 (2021)
18. Park, Y.W., Jiang, Y., Klabjan, D., Williams, L.: Algorithms for generalized clusterwise linear regression. INFORMS Journal on Computing **29**(2), 301–317 (2017)
19. Späth, H.: Clusterwise linear least absolute deviations regression. Computing **37**(4), 371–377 (1986)
20. Späth, H.: Algorithm 39 clusterwise linear regression. Computing **22**(4), 367–373 (1979). https://doi.org/10.1007/BF02265317, https://doi.org/10.1007/BF02265317
21. Späth, H.: A fast algorithm for clusterwise linear regression. Computing **29**(2), 175–181 (1982). https://doi.org/10.1007/BF02249940, https://doi.org/10.1007/BF02249940
22. Vapnik, V.: The nature of statistical learning theory. Springer science & business media (1999)