MXN442 - Modern Computing Techniques

Student	Thanh Long Vu
Student ID	N11177071
Paper Title	Data-Driven Subgroup Identification for Linear Regression
Github	https://github.com/olivervu25/Assignment2_MXN442

1 Problem and research questions

1.1 Problem

In many scientific and medical studies, researchers aim to determine the relationships between various covariates and outcomes of interest, such as the effect of drug concentration on patient health metrics like blood pressure. Traditional parametric models, such as linear regression, are often employed due to their interpretability. However, these models operate under the implicit assumption that relationships between covariates and outcomes are uniform across the entire population. This assumption can lead to oversimplified interpretations and may obscure important variations in subpopulations, especially in heterogeneous datasets where different groups exhibit distinct trends.

For instance, a pharmaceutical company might want to study the effect of drug concentration on changes in blood pressure across patients of varying ages. Using a single, global model would assume that the drug's effect on blood pressure is the same for all age groups, which may not reflect the actual biological variations present. Different age groups may exhibit varying responses to the drug, with some subgroups showing a stronger or weaker response. In this context, a parametric model could miss these nuances, failing to capture significant subpopulation behaviours.

More complex models, such as neural networks, can be employed to capture intricate relationships within the data and improve predictive accuracy. However, this increased complexity often comes at the expense of losing interpretability, which is a critical consideration in scientific and medical contexts. In these fields, it is essential to understand the specific relationships between covariates and outcomes to ensure that clinical and scientific decisions are informed by transparent and actionable insights. While neural networks and other high-capacity models may provide better predictive performance, their lack of interpretability can limit their utility when the primary goal is to draw meaningful conclusions from the data.

1.2 Research Question

The objective of this study is to develop a method that identifies subgroups within a dataset where the relationship between covariates and outcomes can be more effectively modeled using simple, interpretable models. The aim is to discover regions where interpretable models, such as linear regression, perform well, thereby improving both interpretability and predictive performance compared to fitting a single model to the entire dataset.

The key research question is:

- Can we automatically identify and model subpopulations in a dataset while preserving the interpretability of the models used to describe the relationships between covariates and outcomes?

2 Proposed solution

2.1 Formal Problem Statement

The dataset $D = \{(x_i, y_i)\}_{i=1}^n$ consists of features $x_i \in \mathbb{R}^d$ and responses $y_i \in \mathbb{R}$. The region R is defined as $R = \prod_{i=1}^d [a_i, b_i]$, which represents an axis-aligned sub-region of the feature space. The model is characterized by the coefficients β, which define the linear relationship between the features and the response variable. The goal is to compute R and β such that the expected squared error

$$\mathbb{E}\left[\left(\beta^T x - y\right)^2 \mid x \in R\right]$$

is minimized, with the region R being as large as possible while still satisfying this constraint.

2.2 DDGroup Algorithm Description

The Data-Driven Group Discovery (DDGroup) algorithm is a novel approach for identifying interpretable regions within a dataset where a linear relationship between covariates and the response variable exists. The algorithm proceeds in three distinct phases:

1. **Core Group Discovery**: Identify a subset of points that can be well-explained by a local linear model.
2. **Point Rejection**: Eliminate points that do not conform to the model discovered in the core group.
3. **Region Expansion**: Grow a large axis-aligned region that contains no rejected points, starting from the core group.

Phase 1: Core Group Discovery In this phase, we aim to identify a core group of points that minimizes the training error. For each point x_i, we find its k-nearest neighbors and fit a local linear regression model using ordinary least squares (OLS) on the neighborhood. The mean squared error (MSE) is calculated for each local model. The group of points that achieves the lowest MSE is selected as the core group.

Algorithm 1 COREGROUP(k, D)

```
Input: Core group size $k$, dataset $D$
$\mathrm{MSE}^* \leftarrow \infty$
for each $(x_i, y_i) \in D$ do
    $D_{\text{nbhd}} \leftarrow \mathrm{KNN}(x_i, k, D)$
    $\beta \leftarrow \mathrm{OLS}(X_{\text{nbhd}}, Y_{\text{nbhd}})$
    if $\mathrm{MSE}(\beta, D_{\text{nbhd}}) < \mathrm{MSE}^*$ then
        $D_{\text{core}} \leftarrow D_{\text{nbhd}}$
        $\mathrm{MSE}^* \leftarrow \mathrm{MSE}(\beta, D_{\text{nbhd}})$
    end if
end for
Output: $D_{\text{core}}$
```

Phase 2: Point Rejection Once the core group is identified, we proceed to reject points that cannot feasibly follow the same model as the core group. For each point x_i, the residual error $\ell_i = |y_i - \beta^T x_i|$ is computed, where β represents the core model parameters. Points are rejected if their residual is larger than the threshold $\rho_{\sigma,n}^{\text{grow}} = 2.1\sigma\sqrt{\frac{\log n}{n}}$, and these points are labeled for exclusion in the next phase.

Phase 3: Region Expansion In this final phase, we expand the region around the core group. Starting from the center point, we grow an axis-aligned box in each direction. The box expands outward until it either contacts a rejected point or reaches the boundary. This process continues until the largest region containing no rejected points is found. The final region is denoted as $\hat{R}$, an approximation of the true region of interest.

Algorithm 2 GROWBOX($\bar{x}$, X_{rej}, U)

```
Input: Starting point (center) $\bar{x}$, rejected points $X_{\text{rej}}$, normal vectors $U$
$R \leftarrow \emptyset$
while $X_{\text{rej}} \neq \emptyset$ do
    $x^* \leftarrow \arg\min_{x_i \in X_{\text{rej}}} \|x_i - \bar{x}\|_\infty$
    $a^* \leftarrow x^* - \bar{x}$
    $u^* \leftarrow \arg\max_{u \in U} a^{*T} u$
    Add $(u^*, a^*)$ to $R$
    Remove $x^*$ from $X_{\text{rej}}$
end while
Output: $R$
```

3 Main findings and results

Theoretical analysis findings: Theoretical analysis of DDGroup shows that, as the sample size $n \to \infty$, DDGroup accurately recovers the true region R^* with high probability. The algorithm selects a core group that lies mostly within R^*, and the points rejected by the thresholding procedure are unlikely to belong to R^*. With sufficient data, the algorithm returns a region $\hat{R}$ that approximates R^*, and the difference between the recovered region and the true region converges to zero as the sample size increases. This analysis can also be extended to cases with multiple disjoint subregions, where DDGroup can be applied iteratively to recover each subgroup accurately.

Experiment results: On synthetic datasets, DDGroup consistently outperforms baseline models like K-means clustering and Linear Model Trees (LMT). Table 2 shows F1 scores for DDGroup, LMT, and K-means clustering across different dataset sizes. DDGroup achieves significantly higher F1 scores even with smaller datasets ($n = 200$), whereas LMT and K-means struggle with lower performance. As the dataset size increases, DDGroup's performance approaches perfect accuracy, demonstrating its robustness and effectiveness in identifying the correct regions.

Model	200	400	800	1600	3200	6400	12800
DDGroup	0.73 ± 0.03	0.68 ± 0.07	0.93 ± 0.02	0.98 ± 0.00	0.99 ± 0.00	0.99 ± 0.00	1.00 ± 0.00
LMT	0.23 ± 0.09	0.32 ± 0.10	0.19 ± 0.09	0.28 ± 0.10	0.48 ± 0.11	0.92 ± 0.05	0.93 ± 0.05
Clustering	0.07 ± 0.04	0.18 ± 0.07	0.02 ± 0.01	0.12 ± 0.05	0.12 ± 0.06	0.14 ± 0.07	0.11 ± 0.06

Table 1: Performance on synthetic datasets comparing DDGroup, LMT, and K-means clustering, showing F1 scores across various sample sizes.

DDGroup's superior performance also extends to 5 real-world datasets, where it achieves the lowest test MSE compared to baseline linear regression, K-means clustering, and LMT. These datasets will be shown in the paper replication section to avoid repetition, but the evidence shows that DDGroup consistently outperforms alternative methods, making it a reliable tool for identifying interpretable subregions in both synthetic and real-world datasets.

4 Paper replication and new methods exploration

In this section, we replicate all three methods from the original paper (DDGroup, LMT, and Clustering) on synthetic datasets and implement two new approaches, Gradient Boosting Machines (GBM) and Support Vector Machines (SVM). For real-world datasets, we applied these methods to five datasets, selecting the two best-performing models, GBM and DDGroup, to compare against the baseline. This comparison aims to assess the effectiveness of each method in identifying regions where a linear model excels, both in simulated environments and in real-world scenarios.

4.1 Synthetic Data

The synthetic data simulates a scenario where a "good" region exists, within which a linear relationship between the features and the target variable holds. The good region is defined as:

$$R = \left[\left[-\frac{1}{3}, \frac{1}{3}\right], \left[-\frac{1}{3}, \frac{1}{3}\right]\right]$$

Inside this region, the target variable Y is generated with a linear model $Y = X \cdot \beta + \text{noise}$, where $\beta = [1.0, 1.0, 20.0]$ and the noise follows a small standard deviation (std_in = 0.3). Outside the region, the data is noisier with a larger standard deviation (std_out = 5.0).

The feature space B ranges from -1 to 1 for each dimension, and the dataset sizes $n = [200, 400, 800, 1600, 3200, 6400]$ are varied across multiple trials to study performance. The hyperparameters $g1s = [0]$ and $g2s = [2, 4, 8, 16, 32, 64]$ are used to adjust the expansion of the region during subgroup identification.

4.1.1 Paper Methods

The paper has implemented 3 methods, including DDGroup, in the experiment:

- **DDGroup:**
  - Identifies subgroups where a linear model can be effectively applied.
  - Starts with a core group of points fitting a linear model well, then grows a region by rejecting points that deviate.
  - Outputs a region where the linear model is expected to hold with low variance.
- **LMT (Linear Model Tree):**
  - Uses a decision tree to partition the data into regions where a linear model is fitted.

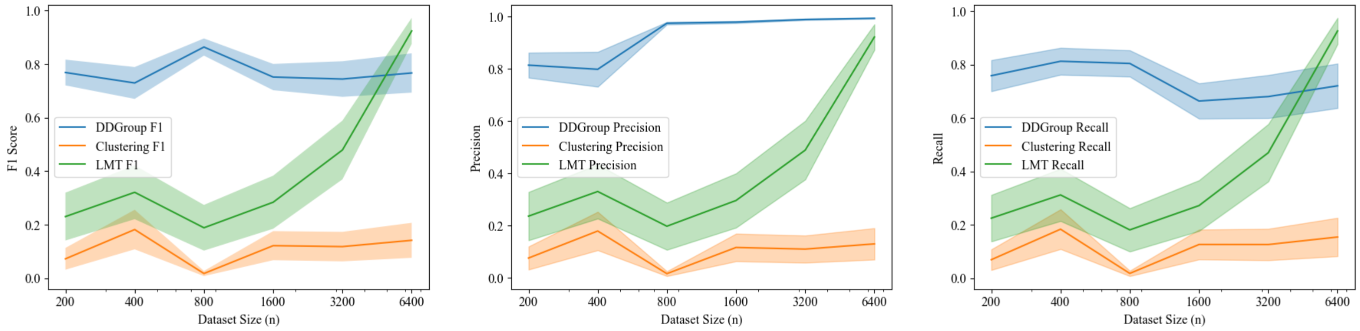
  - Varies the tree depth and selects the model with the smallest validation MSE.
  - The region corresponding to the best-performing tree depth is chosen for further analysis.

- **Clustering:**
  - Applies K-means clustering to partition the dataset into clusters.
  - For each cluster, a linear model is fitted, and the validation MSE is calculated.
  - The cluster with the lowest validation error is chosen as the final model.

4.1.2 Evaluation Metrics

- **Precision:** Measures the proportion of correctly identified points in the "good" region out of all points predicted to be in the good region. A higher precision means fewer points outside the good region are included.
- **Recall:** Measures the proportion of true "good" points correctly identified by the model. A higher recall means most points in the good region are captured by the model.
- **F1 Score:** The harmonic mean of precision and recall, balancing the trade-off between them for an overall performance measure.

For each method, the precision, recall, and F1 score are calculated over a range of dataset sizes. The best region for each trial is selected based on the F1 score, which balances the model's ability to both correctly identify the "good" points (recall) and exclude the "bad" points (precision).



(a) F1 Scores for DDGroup, LMT, and Clustering

(b) Precision for DDGroup, LMT, and Clustering

(c) Recall for DDGroup, LMT, and Clustering

Figure 1: Comparison of the three models (DDGroup, LMT, and Clustering) across three performance metrics: (a) F1 Score, (b) Precision, and (c) Recall. DDGroup consistently performs better in F1 score across different sample sizes, while the other methods show varied performance.

Model	200	400	800	1600	3200	6400	12800
DDGroup	0.73 ± 0.03	0.68 ± 0.07	0.93 ± 0.02	0.98 ± 0.00	0.99 ± 0.00	0.99 ± 0.00	1.00 ± 0.00
LMT	0.23 ± 0.09	0.32 ± 0.10	0.19 ± 0.09	0.28 ± 0.10	0.48 ± 0.11	0.92 ± 0.05	0.93 ± 0.05
Clustering	0.07 ± 0.04	0.18 ± 0.07	0.02 ± 0.01	0.12 ± 0.05	0.12 ± 0.06	0.14 ± 0.07	0.11 ± 0.06

Table 2: Performance on synthetic datasets comparing DDGroup, LMT, and K-means clustering, showing F1 scores across various sample sizes.

4.1.3 New methods

We explore Gradient Boosting Machines (GBM) and Support Vector Machines (SVM) to identify the "good" region in the synthetic dataset. GBM, an ensemble of decision trees, excels at modeling complex patterns but may overfit in noisy data or struggle with multiple subregions. SVM, known for creating clear decision boundaries, can effectively handle non-linear patterns but may produce overly large regions, including outliers, due to its reliance on support vectors. We will assess their effectiveness in the synthetic data below and select one for the real-world data experiment.

4.1.3.1. **Gradient Boosting Machine (GBM):**

For GBM, we selected the model parameters through a combination of trial and error, optimizing for performance on the synthetic dataset. The dataset is relatively simple, which allowed us to use a smaller number of estimators than is typical for more complex data. After experimenting with higher numbers of estimators, we found that using a smaller number actually resulted in better performance for our dataset. Below are the key parameters we adjusted:

- **n_estimators**: We set this to 30, as higher values did not significantly improve performance due to the simplicity of the data.
- **max_depth**: We tuned this parameter from 1 to 5, selecting the depth that minimized the Mean Squared Error (MSE) on the validation set. A smaller depth helps prevent overfitting while still capturing the core structure of the data.
- **learning_rate, subsample, random_state, and loss**: We set the learning rate to 0.05 to ensure gradual learning, and used a subsample of 0.9 to introduce randomness and reduce overfitting. A fixed random seed of 40 was used to ensure reproducibility, while the Huber loss was chosen for its robustness to outliers, given the presence of noise in the synthetic dataset.

After training the model, we extract regions based on the leaf structure of the decision trees. We evaluated several tree depths, and the model with the lowest MSE on the validation set was selected as the best model. The corresponding region is stored in `gbm_res_dict`, and this region is used to compute Precision, Recall, and F1 scores.

4.1.3.2. **Support Vector Machine (SVM)**

For SVM, we used the Radial Basis Function (RBF) kernel to model nonlinear relationships within the data. We experimented with different values of the regularization parameter (C) to balance margin size and misclassification. The key parameter selections and their rationale are as follows:

In addition to the RBF kernel, we also tested the linear and polynomial (poly) kernels. However, the linear kernel failed to capture the complex relationships in the data, as it is too simple to account for nonlinearity. The polynomial kernel also did not perform well, likely due to its inability to effectively capture the structure of the "good region" within the data. The RBF kernel, on the other hand, provided the best performance as it is well-suited for modeling nonlinear patterns, which was crucial for this dataset.

- **C**: This parameter controls the trade-off between maximizing the margin and minimizing classification error. We tested a range of values ({0.01, 0.1, 1, 10, 100}) and selected the best-performing one based on MSE on the validation set. A higher C value reduces the margin but increases the model's focus on classifying all training points correctly.
- **kernel**: We used the RBF kernel to capture nonlinearity in the data, as we expected the relationship between features and labels to be nonlinear.
- **gamma**: This was set to 'auto', which automatically scales the gamma parameter based on the number of features in the data.

For each C value, we computed the MSE on the validation set, selecting the best-performing C value. After identifying the optimal model, we extracted regions based on the support vectors—points that define the SVM decision boundary.

The region corresponding to the best C value is stored in `svm_res_dict`. Similar to GBM, the precision, recall, and F1 scores for each region are calculated based on the true "good" points in the dataset.

4.1.4 Performance Comparison and Evaluation

In this section, we compare the performance of the five models: DDGroup, LMT, Clustering, GBM, and SVM, based on the key evaluation metrics: Precision, Recall, and F1 Score. These metrics help assess how effectively each model identifies the "good region" in the data.

To further illustrate the model's ability to identify the 'good region', we provide visualizations comparing the true region with the regions identified by the models. These visualizations show how well each method approximates the underlying structure of the data. For instance, the precision of GBM and SVM in identifying the "good region" increases as the dataset size grows, with each model adapting to the data.

4.1.4.1. **Metrics**

The performance of **GBM** demonstrates strong potential, as it scales effectively with increasing sample sizes. As the data size grows, GBM's F1 scores consistently improve, suggesting that it efficiently captures the structure of the dataset. Notably, when the sample size reaches 1600 and above, GBM nearly identifies the exact regions, showcasing its accuracy. This makes GBM a reliable and scalable method for identifying regions with high precision and recall. It outperforms DDGroup in many instances, making it a compelling candidate for further exploration.

In contrast, **SVM** shows limited effectiveness. While it performs slightly better than Clustering and LMT in some cases, its overall performance remains relatively low. This suggests that SVM may struggle to model the underlying structure of the data effectively, rendering it less suitable for this specific task.

Model	**200**	**400**	**800**	**1600**	**3200**	**6400**
DDGroup F1	0.77 ± 0.05	0.73 ± 0.06	0.86 ± 0.03	0.75 ± 0.05	0.74 ± 0.07	0.77 ± 0.07
Clustering F1	0.07 ± 0.04	0.18 ± 0.07	0.02 ± 0.01	0.12 ± 0.05	0.12 ± 0.06	0.14 ± 0.07
LMT F1	0.23 ± 0.09	0.32 ± 0.10	0.19 ± 0.09	0.28 ± 0.10	0.48 ± 0.11	0.92 ± 0.05
GBM F1	**0.76 ± 0.04**	**0.85 ± 0.03**	**0.96 ± 0.01**	**0.98 ± 0.00**	**0.99 ± 0.00**	**0.99 ± 0.00**
SVM F1	**0.27 ± 0.01**	**0.25 ± 0.01**	**0.21 ± 0.00**	**0.21 ± 0.00**	**0.20 ± 0.00**	**0.20 ± 0.00**

Table 3: F1 Scores for Different Models Across Various Sample Sizes

Upon further analysis of the precision and recall scores in Table 4, we can see that **GBM** consistently outperforms the other methods across both metrics. This performance trend reinforces the effectiveness of GBM in identifying regions with high precision and recall, even as the sample size increases.

However, **SVM** shows an interesting pattern with recall consistently being 1.0 across all sample sizes. Upon investigation, we discovered that the regions identified by SVM tend to be large, often encompassing most of the dataset, including all the points in the good region. This leads to perfect recall but also highlights a limitation of the SVM model in this context. The large regions result in low precision, as the model includes many points outside the good region, failing to balance precision and recall effectively. This reinforces the observation that SVM struggles to identify compact, well-specified regions when applied to this dataset.

Sample Size	**Precision**				
	DDGroup Precision	**Clustering Precision**	**LMT Precision**	**GBM Precision**	**SVM Precision**
200	0.81 ± 0.05	0.08 ± 0.04	0.24 ± 0.09	**0.82 ± 0.05**	**0.16 ± 0.01**
400	0.80 ± 0.07	0.18 ± 0.07	0.33 ± 0.10	**0.91 ± 0.03**	**0.14 ± 0.01**
800	0.97 ± 0.00	0.02 ± 0.01	0.20 ± 0.09	**0.98 ± 0.01**	**0.12 ± 0.00**
1600	0.98 ± 0.00	0.12 ± 0.05	0.30 ± 0.10	**0.99 ± 0.00**	**0.11 ± 0.00**
3200	0.99 ± 0.00	0.11 ± 0.05	0.49 ± 0.11	**1.00 ± 0.00**	**0.11 ± 0.00**
6400	0.99 ± 0.00	0.13 ± 0.06	0.92 ± 0.05	**1.00 ± 0.00**	**0.11 ± 0.00**
Sample Size	**Recall**				
	DDGroup Recall	**Clustering Recall**	**LMT Recall**	**GBM Recall**	**SVM Recall**
200	0.76 ± 0.06	0.07 ± 0.04	0.22 ± 0.09	**0.75 ± 0.03**	**1.00 ± 0.00**
400	0.81 ± 0.05	0.18 ± 0.07	0.31 ± 0.10	**0.82 ± 0.03**	**1.00 ± 0.00**
800	0.80 ± 0.05	0.02 ± 0.01	0.18 ± 0.08	**0.94 ± 0.01**	**1.00 ± 0.00**
1600	0.66 ± 0.07	0.13 ± 0.06	0.27 ± 0.09	**0.97 ± 0.01**	**1.00 ± 0.00**
3200	0.68 ± 0.08	0.13 ± 0.06	0.47 ± 0.11	**0.98 ± 0.00**	**1.00 ± 0.00**
6400	0.72 ± 0.08	0.15 ± 0.07	0.93 ± 0.05	**0.99 ± 0.00**	**1.00 ± 0.00**

Table 4: Precision and Recall Scores for Different Models Across Various Sample Sizes

4.1.4.2. **Visual Performance**

Figure 2 shows the visual performance of good region identification by the five models.

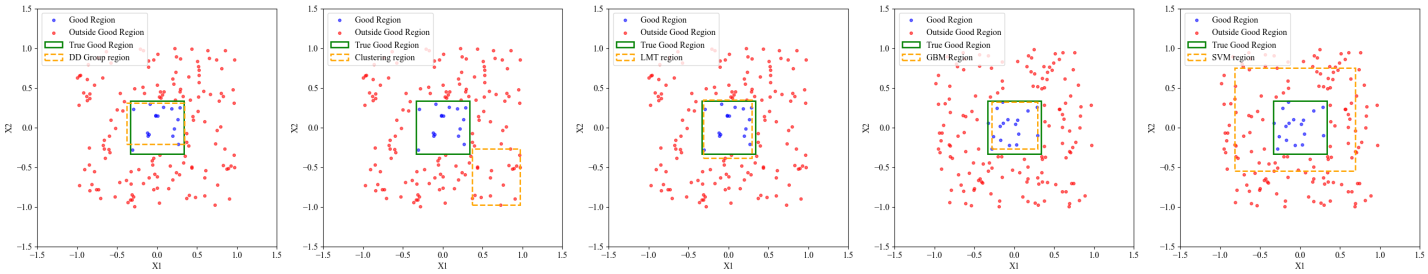
In this figure, we used $n = 200$ data points and replication $rep = 5$ for visualization. Based on the visual performance, GBM shows the best identification of the region, closely matching the actual good region. DDGroup also performs well, though it exhibits some misclassifications, particularly around the boundaries. The SVM model identifies a much larger region that contains many points outside the actual good region, leading to high recall but low precision.

Although LMT shows decent performance in this instance, further investigation reveals that in some cases, it completely misidentifies the region, contributing to its instability and resulting in lower F1 scores across trials.

4.2 Real-world Data

For the real-world data, we used five datasets given in the paper: Brazil Health on heart failure and strokes, China Glucose on glucose and serum uric acid levels, China HIV on HIV-related stigma, Dutch Drinking on alcohol use and cognitive functions in adolescents, and Korea Grip on grip strength and osteoarthritis. These datasets helped us compare two new models, GBM and DDGroup, with the baseline linear regression model.

In this comparison, we evaluate the baseline model (linear regression on the entire dataset) against the two best-performing models from the synthetic data experiments: DDGroup and GBM. We focus on comparing the Test MSE across these models to assess their effectiveness in identifying regions where a linear model performs well. This allows us to gauge how well each model generalizes to the real-world datasets.



(a) DDGroup (b) Clustering (c) LMT (d) GBM (e) SVM

Figure 2: Visual performance of good region identification by five models: (a) DDGroup, (b) Clustering, (c) LMT, (d) GBM, and (e) SVM. The green line represents the true good region, the dashed yellow line indicates the region identified by the model, red points represent the data points outside the good region, and blue points represent those inside.

In this experiment, we conducted a single run for each dataset, which differs from the original paper, where the authors performed 10 runs due to computational limitations. Despite this difference, the results we obtained align closely with the outcomes presented in the original study, confirming the consistency and reliability of the methods tested.

Dataset	Task	Baseline Test MSE	DDGroup Test MSE	GBM Test MSE
Brazil Health	HF	0.5944	0.5045	**0.3751**
	Stroke	0.4318	**0.0180**	0.3580
China Glucose	SUA-F	1.0999	0.5422	**0.4804**
	SUA-M	0.9698	**0.8931**	0.9788
China HIV	Stigma	0.9364	**0.5489**	0.6489
Dutch Drinking	Inh	**0.6121**	0.5319	0.6233
	Wm	0.8090	**0.7211**	0.7820
	Sha	0.5585	**0.4657**	0.5712
Korea Grip	Strength	**0.8053**	0.8125	0.8394

Table 5: Test MSE Comparison across Baseline, DDGroup, and GBM approaches

In this comparison, DDGroup demonstrates superior performance in most of the tests, with GBM outperforming it in only two cases: Brazil HF and China SUA-F. One possible reason for this is that while GBM excels in synthetic data where there is typically only one well-defined ”good” region, it struggles with real-world data, which often contains multiple regions. GBM may have difficulty identifying the largest or most representative region, and its performance can be negatively affected by outliers.

On the other hand, the growing box algorithm used by DDGroup appears to handle these complexities more effectively. It performs better in the majority of real-world cases, indicating that it generalizes well to problems with multiple regions or areas of interest, making it a more robust choice in diverse, real-world settings.

5 Conclusion

We successfully replicated all key results from the original paper using a combination of the provided code and our own implementations. Additionally, we introduced two new methods (GBM and SVM) taught in MXN442. GBM outperformed the original methods on synthetic datasets, showing promising results.

Our experiments on both synthetic and real-world datasets align with the paper's findings, with DDGroup consistently demonstrating better performance, particularly on real-world data. While GBM performed well on synthetic data, further tuning is needed to optimize its performance on more complex real-world datasets.

References

- Izzo, Z., Liu, R., & Zou, J. (2023). Data-Driven Subgroup Identification for Linear Regression. In *Proceedings of the 40th International Conference on Machine Learning (ICML'23)*, Vol. 202, JMLR.org, pp. 14531–14552.