Projection based clustering for identifying piece-wise functions

Sourabh Roy*

November 4, 2024

1 Abstract

Discovering piece-wise functions is a crucial task in many scientific fields such as Signal Processing, Key-frame Animations, Option Pricing in modeling financial derivatives and so on. In this work, we propose a novel method for discovering the parts of piece-wise functions using clustering techniques, specifically for univariate functions. The method involves five key steps: sorting data based on (x, y) pairs, projecting the data into a higher dimension using a kernel, clustering in the higher-dimensional space, mapping clusters back to the original space, and applying symbolic regression to discover equations for each cluster. We evaluate our method on several synthetic datasets from [5] and SAT worksheets, demonstrating its success in identifying piece-wise components even with unclear discontinuities. We discuss potential improvements, including heuristic-based clustering, kernel selection, and clustering algorithm optimization, to enhance the robustness of the method.

2 Introduction

A **piecewise function** is a mathematical function defined by multiple sub-functions, each of which applies to a specific interval or condition in the domain. Formally, a piecewise function $f : R \to R$ can be defined as follows:

$$f(x) = \begin{cases} f_1(x) & \text{if } x \in I_1 \\ f_2(x) & \text{if } x \in I_2 \\ \vdots & \vdots \\ f_n(x) & \text{if } x \in I_n \end{cases}$$

where:

*Department of Computer Science, The University of Manchester
sourabh.roy@student.manchester.ac.uk

- Each $f_i(x)$ is a continuous or discontinuous function,
- The intervals $I_1, I_2, \ldots, I_n$ partition the domain of f such that $I_1 \cup I_2 \cup \ldots \cup I_n = R$ and $I_i \cap I_j = \emptyset$ for $i \neq j$.

The function may exhibit different behaviors or properties in each piece, making it suitable for modeling situations where relationships change abruptly or continuously across the domain. The points where the function transitions from one sub-function to another are called **breakpoints**.

In numerous scientific and engineering domains, the behavior of complex systems is often best captured using piecewise functions, where different equations govern distinct regions of the input space. Fields such as control systems engineering utilize piecewise functions to model systems that switch between different modes of operation, for example, depending on varying external conditions. In economics, tax systems or pricing models frequently change rates or strategies at defined thresholds. Similarly, in medical sciences, drug concentration in the bloodstream may follow different pharmacokinetic models during absorption and elimination phases.

Accurately identifying these regions and the corresponding equations is critical for understanding and modeling such systems. However, this task presents significant challenges. In many real-world applications, the discontinuities or transitions between different regimes are often not explicitly known in advance. For instance, in structural engineering, the stress distribution on a beam under load may follow different mechanical laws in different regions, but pinpointing where these changes occur can be highly complex due to irregularities in material properties. Likewise, in machine learning, algorithms like decision trees naturally approximate piecewise models, but they struggle with finding smooth transitions and hidden discontinuities in data.

Given a dataset with each entry of the form $\{x_1, x_2, ..., x_k, y\}$, where $y = f(x_1, x_2, ..., x_k)$, *Symbolic Regression* aims to discover the symbolic expression for f which best fits the dataset, both in terms of accuracy and simplicity [1]. While symbolic regression is a fundamental task in artificial intelligence, the state of the art methods are limited to discovering continuous functions.

Traditional regression techniques, such as linear regression or polynomial fitting, often fail in such contexts. They assume a continuous, smooth functional relationship, making it difficult to detect the points of transition between different behaviors. Moreover, when the functional regions overlap or the shifts between regimes are subtle (as seen in signal processing or biological systems modeling), the problem becomes even more complex. These methods also struggle when the data is noisy, as it becomes harder to distinguish actual discontinuities from random variations.

Thus, discovering piecewise functions requires specialized techniques capable of handling discontinuities, uneven data distribution for each piece, and abrupt shifts - something traditional regression is not equipped for without prior knowledge of the system.

Successfully discovering piecewise functions leads to practical benefits across fields. In control systems, it allows efficient switching between operating modes,

like in automated machines. In economics, it helps optimize tiered tax policies by clearly modeling income brackets. In medicine, it improves drug dosage recommendations by capturing distinct absorption phases. These examples show how piecewise discovery enhances predictions and decision-making in systems with complex, abrupt changes.

In this paper, we focus on discovering piece-wise functions in uni-variate datasets, which present unique challenges due to the potential overlap of functional outputs across regions.

Clustering is essential when discovering piecewise functions[9],[13] because it helps identify distinct regions in the data where different equations apply. By grouping data points with similar behaviors or patterns, clustering isolates discontinuities and nonlinear transitions that are not easily detectable through traditional regression methods. Standard clustering algorithms, when applied directly to the data, tend to struggle with separating regions where outputs are closely aligned, making it difficult to detect transitions or discontinuities. Additionally, simple data transformations, such as scaling or translating, often fail to improve clustering performance.

To address these challenges, we propose a method that leverages kernel-based projection techniques, transformations and clustering algorithms to identify distinct regions in piece-wise functions. Our approach begins by sorting the data based on (x, y) value pairs to ensure that the clustering algorithm respects the natural order of the data. We then project the data into a higher-dimensional space using kernels, inspired by Support Vector Machines' approach to handling non-linear separability. This projection amplifies discontinuities in the data, making them more distinguishable for clustering algorithms. After clustering in the higher-dimensional space, we re-project the results back into the original 2D plane and use symbolic regression to derive equations for each cluster.

Our method is tested on a set of synthetic functions, with mostly positive results. Even in cases with unclear discontinuities or point discontinuities our approach successfully identifies the distinct regions.

We explore these limitations and propose several areas for improvement, including developing an informed heuristic for clustering, determining the optimal kernel for different datasets, and selecting the most appropriate clustering algorithm for various types of piece-wise functions.

3 Related Work

There is little work in the discovery of piece-wise or discontinuous functions. The earliest work [5] introduced HVES, a genetic programming (GP) based approach for discovering discontinuous functions. It exploited the error exhibited by the best model to partition the data space to identify the sub-functions. However, this method was extremely costly computationally, with certain functions taking more than 24 hours to discover.

Although not directly related, Ly et al [9], introduced MSMR to learn symbolic models of discrete dynamical mappings which has a component named

Clustered Symbolic Regression (CSR). It uses clustering, Symbolic Regression (SR) and Expectation Maximisation (EM). However, this method can be computationally expensive with larger datasets. EM assumes that the data follows a specific distribution. If the actual underlying distribution differs significantly, it can lead to inaccurate parameter estimates and misclassification of data points.

[13] uses a tree based GP method along with an adaptive space partition strategy to discover piece-wise equations. However, in case of functions with very subtle or unclear discontinuities, there might be incorrect partitioning leading to poor results. In these cases, a clustering approach is better suited.

4 Method

Our method follows three steps to discover the parts of a piece-wise function. Note that we have explored cases in uni-variate functions only. Code is available in the GitHub repository of the project.[1]

Data Processing Sort the data according to increasing (x, y) value pairs: this is useful for the clustering algorithm since the data is now in the same order as how one would plot it.

Data Projection Use a kernel to project the data to a higher dimension (see figure 1): this step was motivated by how Support Vector Machines use kernels to project linearly inseparable data to a higher dimension. in our case, this is from 2D to 3D. The functional outputs are extremely close in the real plane, often hiding discontinuities. Vanilla clustering methods such as K-Means, DBSCAN, Hierarchical Clustering perform poorly in such cases.

There are 3 kernels that we used to project our data points from 2D to 3D for enhanced separation of points of discontinuities. Given a data point (x, y) and Kernel K, we have:

1. **Linear Kernel**: $K(x, y) = c.x.y$
2. **Radial Basis Function (RBF) Kernel**: $K(x, y) = exp(-\gamma * ||x - y||^2)$
3. **Sigmoid Kernel**: $K(x, y) = tanh(\gamma * x.y + c)$

In some of the more difficult cases, we employed a composition of Sigmoid and RBF kernels i.e $K_{sigmoid}(K_{RBF}(x, y))$. Simple transformations such as scaling or translating also fail to make any noticeable improvements. Combining transformations with projections to a higher dimension, improves the clustering by creating significant gaps between the discontinuities.

Clustering in higher dimensions Perform clustering on 3D data: for simplicity, we used K-Means clustering and Gaussian mixture models separately and picked the algorithm with the best performance.

[1] `https://github.com/bublaiSAURUS/Piecewise-Symbolic-Regression`

Clustering in original dimension Re-project back to the 2D plane and find the clusters: We use the clustering labels from 3D clustering to map back the clusters on the actual 2D data.

Symbolic Regression Use Symbolic Regression to discover the equations from the clusters. We used the package PySR [2], available online. It uses genetic programming. The intervals could be found by extracting the minimum and maximum x values from each cluster. In the context of piece-wise functions, this step is of little importance as compared to identifying the parts of the function correctly. We will **not** present any further discussions on this.

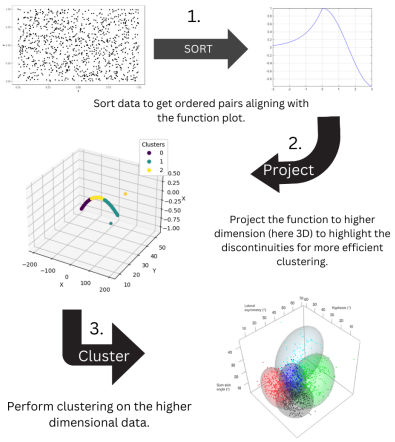

Figure 1: Figure showing key workflow steps prior Symbolic Regression.

Algorithm 1 Projection based clustering using Sigmoid Kernel and GMM

INPUT: A dataset of (x, y) pairs.
OUTPUT: A set of clusters corresponding to the data points.
$data \leftarrow$ input dataset containing (x, y) pairs
$x \leftarrow 10 \cdot data[:, 0]$ ▷ Transform x coordinate.
$y \leftarrow 10 + data[:, 1]$ ▷ Transform y coordinate.
$z \leftarrow \tanh(100 \cdot x \cdot y)$ ▷ Project to 3D.
$data\_3d \leftarrow$ concatenate (x, y, z) into a 3D array
Use GMM to fit $data\_3d$ and obtain cluster labels
Return clusters from the fitted GMM

5 Experimental Setup

We experimented on a set of 6 functions, denoted $F_1, F_2, ..., F_6$. All functions are uni-variate. F_1, F_2 are those used in [5]. All functions are shown in the

Appendix. We remark that we defined the functions F_4, F_6 with point discontinuities so as to test the robustness of our algorithm. We executed K-Means, Gaussian Mixture Models (GMMs), DBSCAN and hierarchical clustering on all the functions and finally selected the one with best performance.

6 Results

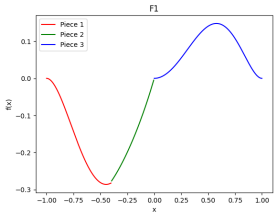

Figure 2: Plot of piecewise function F_1 with 3 pieces

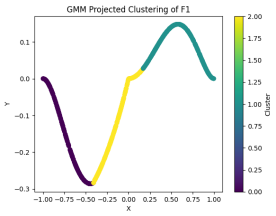

Figure 3: Projection based clustering of pieces of F_1

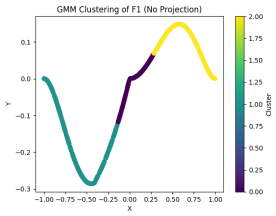

Figure 4: Without Projection clustering of pieces of F_1

We first demonstrate above that projecting is an essential step by comparing the results of clustering with and without projection on F_1.

We next present the outcomes of our experiments on the full set of functions.

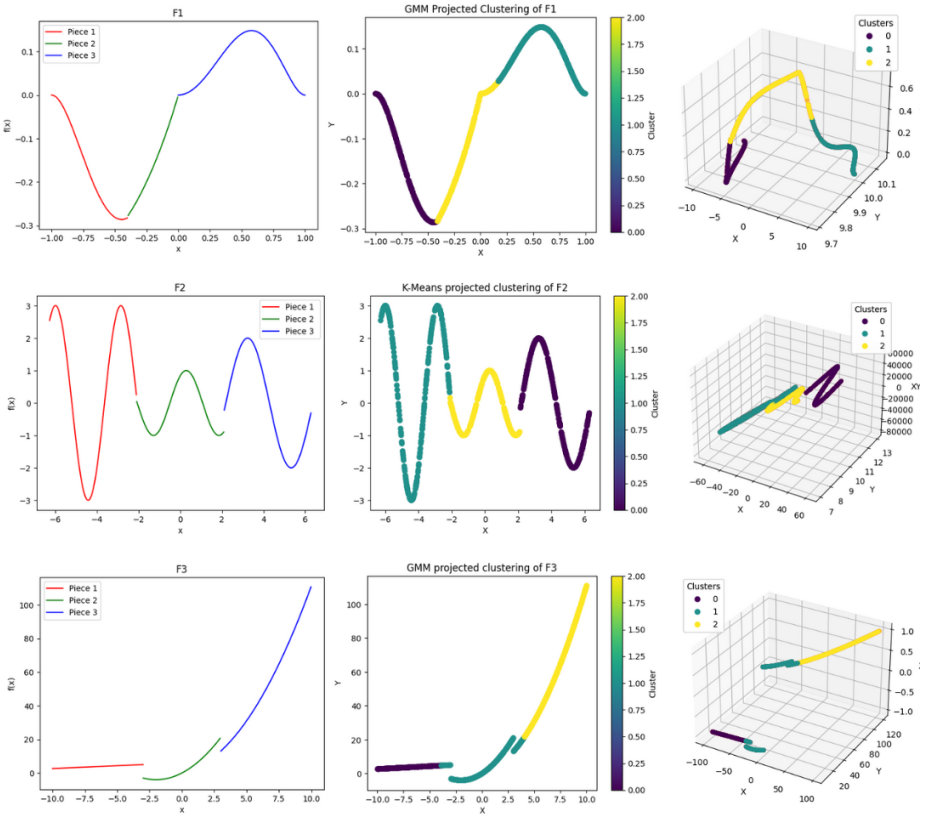

Figure 5: Results for $F_1 - F_3$. First column shows the ground truth plot of the function. Second column shows the results of clustering of the pieces of the function done by the projection based clustering method. Third column shows the 3D plot of projected data for the function.

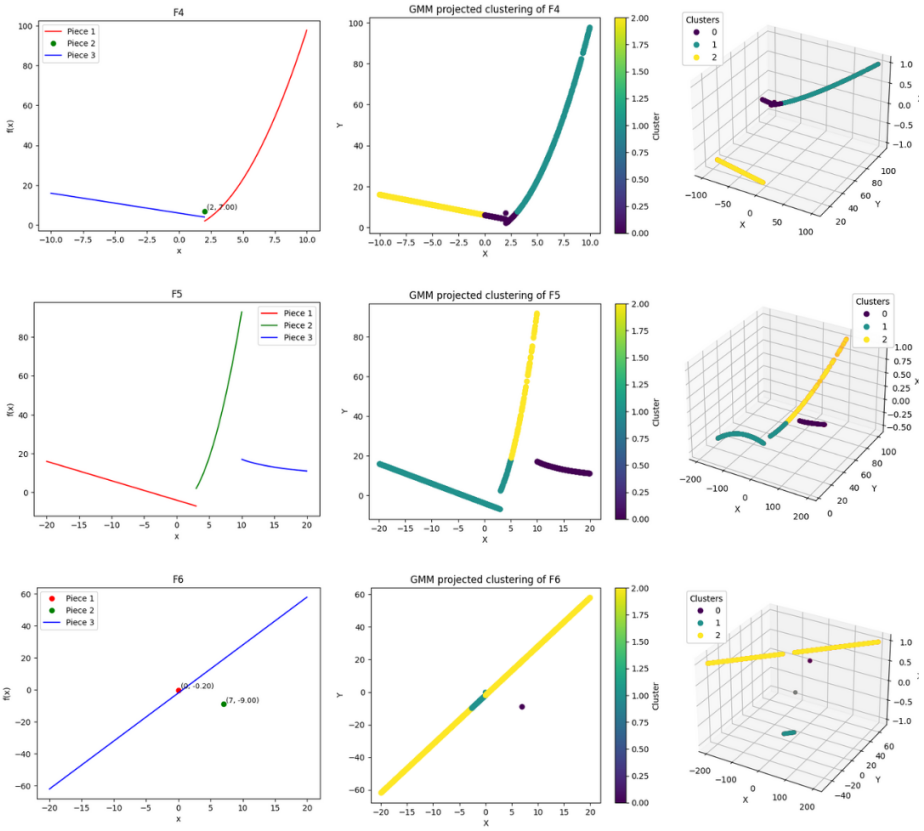

Figure 6: Results for F_4, F_5, F_6. Third column shows the 3D plot of projected data for each function.

The algorithm could successfully identify the distinct parts of out functions in 6 cases. For few cases such as F_5 there were "leakages" to other parts. This can be rectified if we provide more prior information to the algorithm.

In almost all of the cases, GMMs outperformed all other clustering algorithm. Only in F_2, K-Means produced a slightly better outcome.

F_1 used a composition of Sigmoid and RBF kernels. F_2, F_5 used linear kernel. F_3, F_4, F_6 used Sigmoid kernel.

7 Discussion and Future Work

Our method is promising. However there are clear improvements that need to be made:

Efficient heuristic for clustering : An efficient heuristic that guides the clustering method can significantly improve correct identification of parts of piece-wise functions. We used the standard metrics such as BIC score, Akaike information, CH score, Silhouette score and Davis Bouldin score. Unfortunately, these metrics do not agree with the actual correct number of clusters. Developing an informed clustering should yield the solution to this problem.

Designing an informed clustering heuristic In some cases, the clustering is not exactly the same as the ground truth, highlighting the need for more sophisticated clustering heuristics and kernel selection strategies.

Predetermining the kernel For different datasets, different kernels are effective. Two of the most effective kernels are: linear kernel and $tanh()$ kernel. For function 1, a composition of $tan()$ and RBF kernel was found to be effective. For a robust algorithm, a fixed kernel or some method to determine the appropriate kernel is necessary.

Predetermining the clustering algorithm: We tested both hard clustering using K-Means as well as soft-clustering using Gaussian mixture models. Again, for a robust algorithm, a fixed clustering method or some process to determine the appropriate method is necessary.

Extension to Multivariate functions Extending the algorithm to Multivariate cases is necessary because most of the piece-wise functions found in the real world, are much more complex.

Experimenting on noisy data To simulate real world scenario more closely, experiments with noisy data can be conducted.

8 Conclusion

In this report, we introduced a novel approach for discovering piece-wise functions in uni-variate datasets using a combination of kernel-based projections, clustering techniques, and symbolic regression. By leveraging projections into higher-dimensional spaces, our method successfully identified distinct regions and corresponding equations in datasets that exhibited both clear and subtle discontinuities. The experimental results demonstrated that the approach effectively recognized the parts of the functions in most cases, with successful identification of distinct regions in 6 out of the 7 test functions.

However, there are challenges that remain, particularly with overlapping outputs and point discontinuities, as evidenced by the failure in one of the test cases. These limitations highlight the need for further refinement of the method. Potential improvements include developing more sophisticated heuristics for clustering, enhancing kernel selection, and experimenting with alternative clustering algorithms that may be better suited for complex piece-wise structures.

Overall, the proposed method provides a solid foundation for further exploration in the field of piece-wise function discovery, with promising applications in scientific and engineering contexts where such functions are prevalent. Future work will focus on addressing the limitations identified and optimizing the approach to handle a wider variety of function types more robustly.

9 Acknowledgements

This work is supported by the **School of Engineering (SoE), The University of Manchester (UoM)**. Its contents are solely the responsibility of the authors and do not necessarily represent the official views of UoM or SoE. We are especially thankful to our supervisor, Dr. **Hongpeng Zhou**, for his invaluable guidance, support, and mentorship throughout the project. His insights and encouragement were instrumental in the successful completion of this work. We also appreciate the resources and collaborative environment provided by the university, which greatly enhanced our research experience.

References

[1] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. "Discovering governing equations from data by sparse identification of nonlinear dynamical systems". In: *Proceedings of the national academy of sciences* 113.15 (2016), pp. 3932–3937.

[2] Miles Cranmer. "Interpretable machine learning for science with PySR and SymbolicRegression. jl". In: *arXiv preprint arXiv:2305.01582* (2023).

[3] Miles Cranmer et al. "Discovering symbolic models from deep learning with inductive biases". In: *Advances in neural information processing systems* 33 (2020), pp. 17429–17442.

[4] Miles D Cranmer et al. "Learning symbolic physics with graph networks". In: *arXiv preprint arXiv:1909.05862* (2019).

[5] Cyril Fillon and Alberto Bartoli. "Symbolic regression of discontinuous and multivariate functions by Hyper-Volume Error Separation (HVES)". In: *2007 IEEE Congress on Evolutionary Computation*. IEEE. 2007, pp. 23–30.

[6] Dongni Jia et al. “Governing equation discovery based on causal graph for nonlinear dynamic systems”. In: *Machine Learning: Science and Technology* 4.4 (2023), p. 045008.

[7] Ying Jin and Weilin Fu. “Bayesian Symbolic Regression”. In: *Association for the Advancement of Artificial Intelligence*. 2020. URL: `https://arxiv.org/pdf/1910.08892`.

[8] Samuel Kim et al. “Integration of neural network-based symbolic regression in deep learning for scientific discovery”. In: *IEEE transactions on neural networks and learning systems* 32.9 (2020), pp. 4166–4177.

[9] Daniel L Ly and Hod Lipson. “Learning symbolic representations of hybrid dynamical systems”. In: *The Journal of Machine Learning Research* 13.1 (2012), pp. 3585–3618.

[10] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. “Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations”. In: *arXiv preprint arXiv:1711.10561* (2017).

[11] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. “Physics informed deep learning (part ii): Data-driven solutions of nonlinear partial differential equations”. In: *arXiv preprint arXiv:1711.10566* (2017).

[12] Silviu-Marian Udrescu et al. “AI Feynman 2.0: Pareto-optimal symbolic regression exploiting graph modularity”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 4860–4871.

[13] Hengzhe Zhang et al. “PS-Tree: A piecewise symbolic regression tree”. In: *Swarm and Evolutionary Computation* 71 (2022), p. 101061.

[14] Hongpeng Zhou and Wei Pan. “Bayesian learning to discover mathematical operations in governing equations of dynamic systems”. In: *arXiv preprint arXiv:2206.00669* (2022).

A Piece-wise functions for evaluation

F_1

$$f(x) = \begin{cases} x^5 - 2x^3 + x & \text{if } -1 \leq x < -0.4 \\ x^4 + x^3 + x^2 + x & \text{if } -0.4 \leq x < 0 \\ x^6 - 2x^4 + x^2 & \text{if } 0 \leq x \leq 1 \end{cases}$$

F_2

$$f(x) = \begin{cases} 3\sin(2x+1) & \text{if } -2\pi \leq x < -\frac{2}{3}\pi \\ \sin(2x+1) & \text{if } -\frac{2}{3}\pi \leq x \leq \frac{2}{3}\pi \\ 2\sin\left(\frac{3x}{2} + 3\right) & \text{if } \frac{2}{3}\pi < x \leq 2\pi \end{cases}$$

F_3

$$f(x) = \begin{cases} \frac{x}{3} + 6 & \text{if } x < -3 \\ x(x+4) & \text{if } -3 \leq x < 3 \\ x^2 + x + 1 & \text{if } x \geq 3 \end{cases}$$

F_4

$$f(x) = \begin{cases} x^2 - 2 & \text{if } x > 2 \\ 7 & \text{if } x = 2 \\ 6 - x & \text{if } x < 2 \end{cases}$$

F_5

$$f(x) = \begin{cases} -x - 4 & \text{if } x < 3 \\ x^2 - 7 & \text{if } 3 \leq x \leq 10 \\ \frac{120}{x} + 5 & \text{if } x > 10 \end{cases}$$

F_6

$$f(x) = \begin{cases} \frac{1}{x-5} & \text{if } x = 0 \\ -(x-4)^2 & \text{if } x = 7 \\ 3x - 2 & \text{if } x \neq 0 \text{ and } x \neq 7 \end{cases}$$