SPARSE ROBUST REGRESSION FOR EXPLAINING CLASSIFIERS

Anton Björklund
Andreas Henelius
Emilia Oikarinen
Kimmo Kallonen
Kai Puolamäki

Discovery Science (DS 2019), Lecture Notes in Computer Science, vol. 11828
https://doi.org/10.1007/978-3-030-33778-0_27

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

These notes (on the side) contain the spoken parts

This is a presentation of a novel algorithm that can be used for both robust regression, and to explain outcomes from black box models.

ORGANISATION

- Motivation
- Algorithm
- Robust Regression
- Explanations
- Conclusions

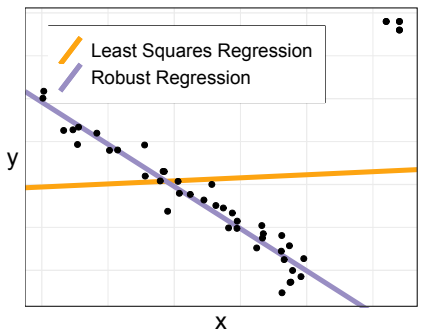
The presentation will begin with some motivating examples and background, followed by the definition of the algorithm. Then we will look at some results of using it for robust regression and explanations.

...

ORGANISATION

- **Motivation**
- Algorithm
- Robust Regression
- Explanations
- Conclusions

SPARSE ROBUST REGRESSION



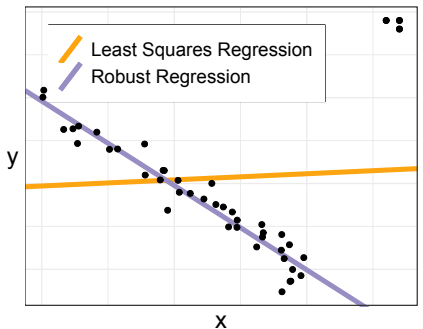
- Robust Regression handles outliers
- Least squares regression might break with just one outlier

So, what does "Sparse Robust Regression" mean?

The difference between normal regression and robust regression is that robust regression can handle outliers, data items that don't follow the same patttern as the majority of the data.

In the top corner of the plot there are three outliers. If we apply Ordinary Least Squares regression on this dataset we get a pretty useless result, because even one outlier is enough to potentially cause issues with Ordinary Least Squares. Meanwhile, robust regression is able to deal with the outliers.

Sparse Robust Regression

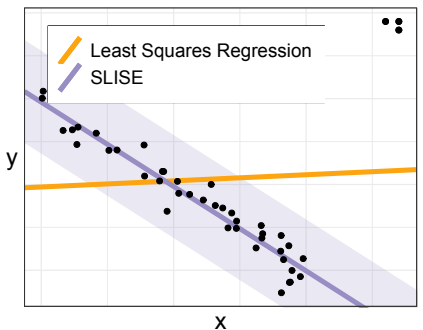


- Robust Regression handles outliers
- Least squares regression might break with just one outlier

- Sparse models are easier to interpret
- Sparsity through LASSO regularisation

Sparse models, on the other hand, are models where some of the weights are zero. This makes them easier to interpret, since we can ignore all zeroes, which is especially important with high-dimensional data. Sparsity is often achieved through LASSO regularisation, also known as L1 regularisation.

SPARSE LINEAR SUBSET



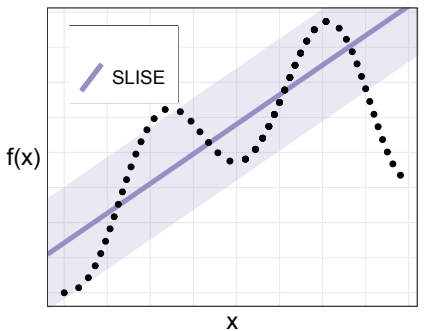
- Novel robust regression algorithm: SLISE
  - **S**parse **LI**near **S**ubset **E**xplanations
- Find the largest subset of data items that can be represented by a linear model to a given accuracy

First, a high-level overview of what we want to accomplish.

We propose a novel robust regression algorithm that finds the largest possible subset of data items that can be represented by a linear model to a given accuracy. We call this algorithm SLISE, an akronym for Sparse Linear Subset Explanations.

As we can see, the plot from the previous slide actually uses SLISE for the robust regression, and in 2D there is an intuitive interpretation of SLISE: Namely, find the "corridor", marked with a purple background, that covers as many points as possible.

LOCAL APPROXIMATION

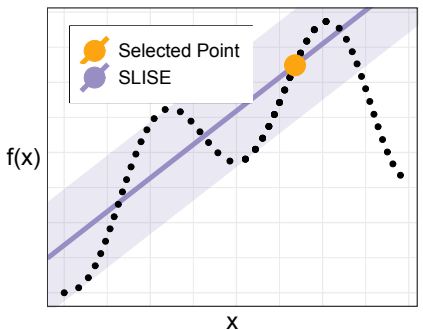


1. Complex function instead of y-values

Now we take the sane idea and apply it on a slightly different domain.

First qe replace the y-values with a complex function, such as a neural network.

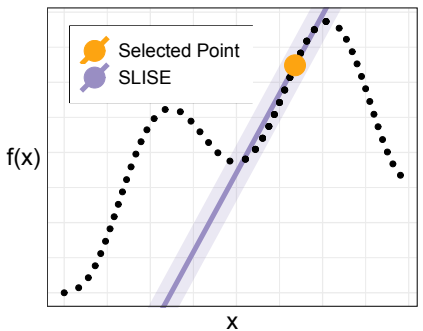
Local Approximation



1. Complex function instead of y-values
2. Force it to go through a point

Then we force the regression line to pass through a selected point.

Local Approximation



1. Complex function instead of y-values
2. Force it to go through a point
3. Slightly reduce the width of the corridor

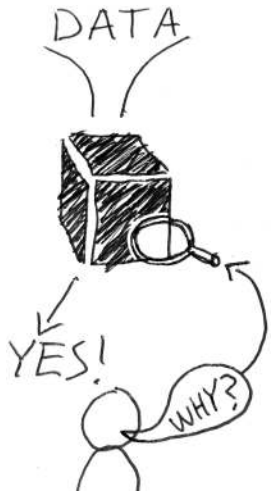
The linear model locally approximates the complex function

Finally we slightly narrow down the corridor, and the result is a local approximation of the complex function with a simple function.

This result is especially interesting because local approximations are a common way of explaining outcomes from complex machine learning models.

WHY EXPLANATIONS

- The best performing models are often “black box” models
- Good accuracy is not enough



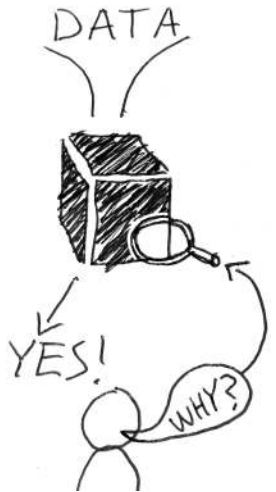
So, why do we need explanations?

The best performing machine learning models are often black box models, where we do not have any intuition of what they are doing internally. These models sacrifice interpretability for accuracy, but sometimes a good accuracy is not enough.

WHY EXPLANATIONS

- The best performing models are often “black box” models
- Good accuracy is not enough

- Building trust
- Communication with domain experts
- Finding and eliminating biases
- New insights about data or the model
- GDPR requires explanations



For example if the black box model is supposed to collaborate with a domain expert a binary yes or no is not sufficient, more information is needed.

Antoher reason, that is very relevant for our geographic location, the European Union, is the General Data Protection Regulation, which requires explanations for "algorithmic descisions".

The Importance of Data

- The interaction between the model and the data is important
- Example:

$$\mathrm{p}(\mathrm{Income} = \mathrm{High}) = \sigma(\ 2 \times \mathrm{Education} + 1 \times \mathrm{Age} - 2\ )$$

The goal of machine learning is to find and exploit structures in the data. With structures we mean any kind of constraints, or dependencies between variables, et.c. the data might have.

Thus, the interaction between the model and the data is important, which I'll demonstrate with a simple example.

Imagine we hava a simple classifier predicting high or low income based on education level and age. In this case it is a logistic regression.

The Importance of Data

- The interaction between the model and the data is important
- Example:

$$\mathrm{p}(\mathrm{Income} = \mathrm{High}) = \sigma(\ 2 \times \mathrm{Education} + 1 \times \mathrm{Age} - 2\ )$$

- Most informative variable:
  - Without data

$$\mathrm{I}(\mathrm{Education}) \ > \ \mathrm{I}(\mathrm{Age})$$

We wan't to know the most informative variable, and if we only look at the model, the education seems like the most important variable.

The Importance of Data

- The interaction between the model and the data is important
- Example:

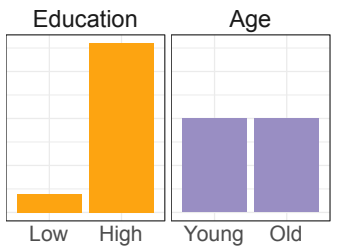
$$\mathrm{p}(\mathrm{Income} = \mathrm{High}) = \sigma(\ 2 \times \mathrm{Education} + 1 \times \mathrm{Age} - 2\ )$$

- Most informative variable:
  - Without data

    $\mathrm{I}(\mathrm{Education}) \ > \ \mathrm{I}(\mathrm{Age})$
  - With data

    $\mathrm{I}(\mathrm{Age}) \ > \ \mathrm{I}(\mathrm{Education} = \mathrm{High})$

    $\mathrm{I}(\mathrm{Education} = \mathrm{Low}) \ > \ \mathrm{I}(\mathrm{Age})$



However, if we take into account the data distribution, that might change. For example this dataset is from a university faculty, where most people have a high education, so learning that somebody has a high education gives little information, in that case age is more important. But, if somebody has a low education, then that is still the most important variable.

ORGANISATION

Moving on to the definition of the SLISE algorithm

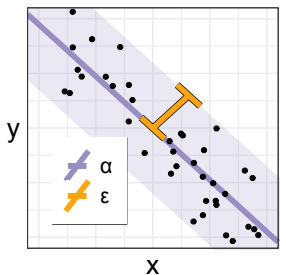
- Motivation
- **Algorithm**
- Robust Regression
- Explanations
- Conclusions

Problem Definition

- **Find the linear model α that minimises:**

$$\text{Loss} = \sum_{i=1}^{n} H\left(\varepsilon^2 - r_i^2\right)\left(r_i^2/n - \varepsilon^2\right) + \lambda \sum_{j=1}^{d} |\alpha_j|$$

- Residuals: $r_i = x_i{}^{\mathsf{T}}\alpha - y_i$
- Error tolerance (corridor radius): $\varepsilon \geq 0$
- Regularisation coefficient: $\lambda \geq 0$



Starting with the mathematical definition of the problem.

Here's the loss function we want to minimise, but don't worry about it, I'll give tou the intuitive description of what it does.

The width of the corridor specified by a parameter we call epsilon, and the corridor consists of all items that have residuals that are smaller than this epsilon.

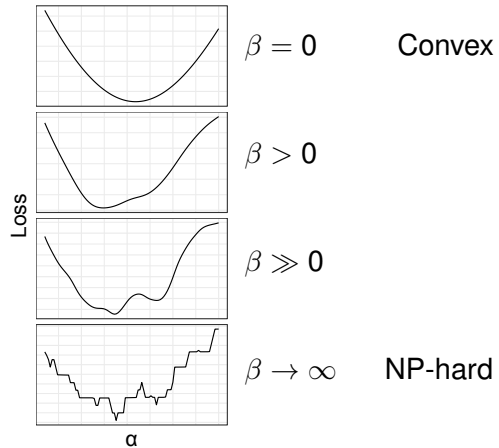
The main goal is to find a linear model, called alpha, that maximises the number of items within this corridor, and as a secondary thing minimises the residuals of the of the items inside the corridor.

Finally, we also add LASSO regularisation at the end to be able to provide sparse solutions.

However, finding the optimum to this loss function is NP-hard (the proof is in the paper). Thus, we need to find a good approximation.

THE SLISE ALGORITHM

- Graduated Optimisation



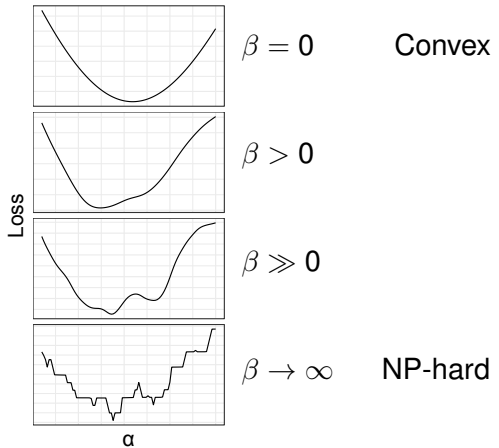
For the approximation we relax the loss function in such a way that we can control the complexity through a parameter we call beta.

This allows us to use graduated optimisation, where you solve a complex problem by gradually increasing the complexity, as illustrated here.

THE SLISE ALGORITHM

- Graduated Optimisation
- Pseudocode:

```
α ← initialise()
β ← 0
while β < β_max:
    α ← minimise(Loss, α, β)
    β ← increase(β)
return α
```



The high-level pseudocode for this is quite simple. It's a loop where we alternate between finding the optimum for the current loss function and increasing the complexity.

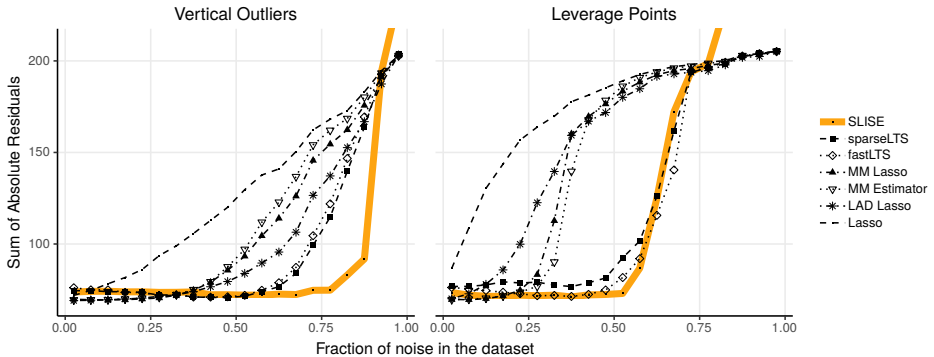
In the paper we describe a clever way of *dynamically* selecting how much the complexity should be increased at each step.

ORGANISATION

- Motivation
- Algorithm
- **Robust Regression**
- Explanations
- Conclusions

Now onto the part of the presentation where we use the algorithm.

Robustness Experiment



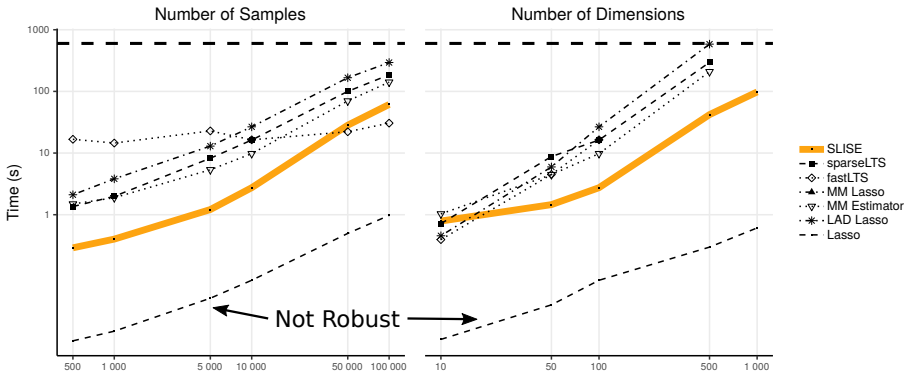
Lower is better

Here we want to compare SLISE to a couple of other, prominent, robust regression methods, and we also include LASSO regression as a non-robust baseline.

The first experiment takes a dataset and slowly corrupts the data items according to two types of outliers, vertical outliers where y-values are changed and leverage points where x-values are changed. We are looking for the point where the curves start to trend upwards, this is called the breakdown point. The larger the breakdown point the more robust a method is.

As we can see, SLISE is able to, at least, match the robustness of all other methods.

Scalability Experiment



Lower is better

In the second experiment we investigate how much time the algorithms need, with a focus on large datasets. Note that we are using a logarithmic scale so the differences are actually larger than they might seem.

In the left plot we are increasing the numbers of items in the dataset, and SLISE is faster than the rest, except for LASSO, and the one method that uses subsampling.

In the right plot we increase the number of dimensions. With high-dimensional data SLISE is clearly the fastest robust method. And with a thousand dimensions SLISE finishes whithin a hundred seconds, while all the other robust methods exceede ten minutes, wich we use as a cutoff for practical reasons

ORGANISATION

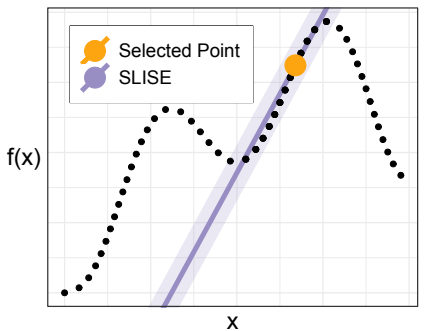
- Motivation
- Algorithm
- Robust Regression
- **Explanations**
- Conclusions

Then it is time to go deeper into the explanations.

SLISE AS AN EXPLAINER

- Model-agnostic explanations
- No modification of the model
- Data items as vectors
- No distance measure needed
- Explains individual outcomes

- Center the data on the selected item to make sure the approximation passes through it

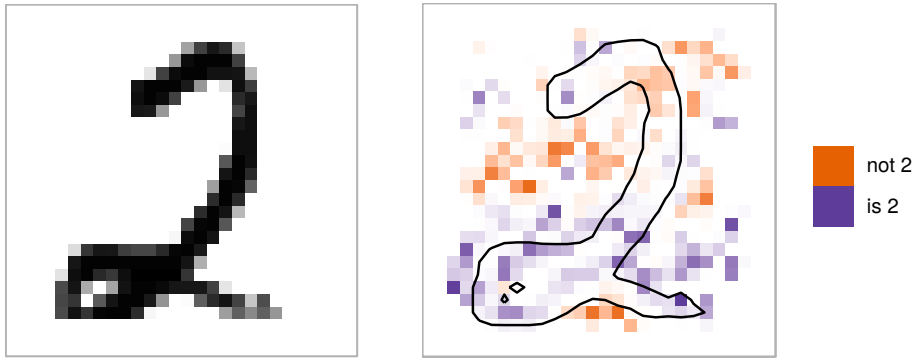


SLISE can be used to explain any kind of model, from neural networks to random forests, or human decisions, as opposed to only consider a specific type of model. SLISE also doesn't require any modifications to the model and only requires the data items to be turned into vectors, with no distance measure needed.

Also note that SLISE gives explanations for individual outcomes, rather than trying to explain the whole black box.

We don't need to modify the SLISE algorithm in order to use it for explanations. However, we need to make sure the regression line passes through the item we want to explain, but this requirement is easy to satisfy by centring the data on that item, and using no intercept.

IMAGE EXPLANATION



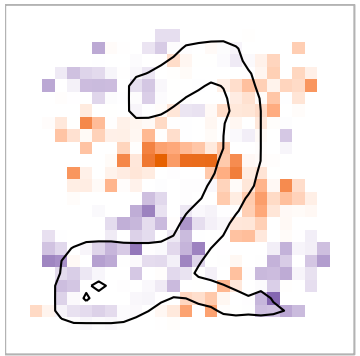
Now for an example of using SLISE for explanations.

This is a two from everybody's favourite, the MNIST dataset, which is a good example because the data is high-dimensional and complex, yet easy to interpret, as handwritten digits.

The explanation highlights two major features: the horizontal line at the bottom, and the empty space to the left. At least to me, it makes sense that the classifier would use these two features.

Data Dependent Explanations

- The explanations are dependent on the data
- Take into account the interaction between the model and the data
- Modify the data to get specific answers

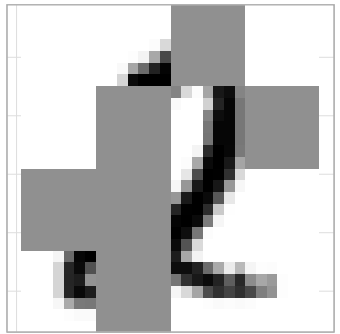


Using real data for the explanations makes the explanations dependent on the data, which allows us to take the interaction between the model and the data into account.

This dependeny can also be used to extract more information from the black box model. To the left is an explanation for the same classifier and the same digit, but using a dataset of only twos and threes. So the explanation shows why the digit is classified as a two and not a three.

Related Explainers

- Create new data instead of using existing data
- Might break data constraints
- Ignore the interaction between the data and the model
- Require a distance measure
- E.g., LIME (Ribeiro et al. 2016, KDD '16)



The explanation methods that are most closely related to SLISE are those that also create local approximations in a model-agnostic way.

The big difference is that the other methods usually create new data instead of using existing data, and desigining these, data-specific, data generating functions can be difficult.

This is a potential source of problems since the generated data might not follow the same constraints as the real data, and the interaction between the model and the generated data might be different from the interaction between the model and real data.

They also need some kind of distance measure, which also might be difficult to design.

The most well known related method is LIME, by Ribeiro et al.

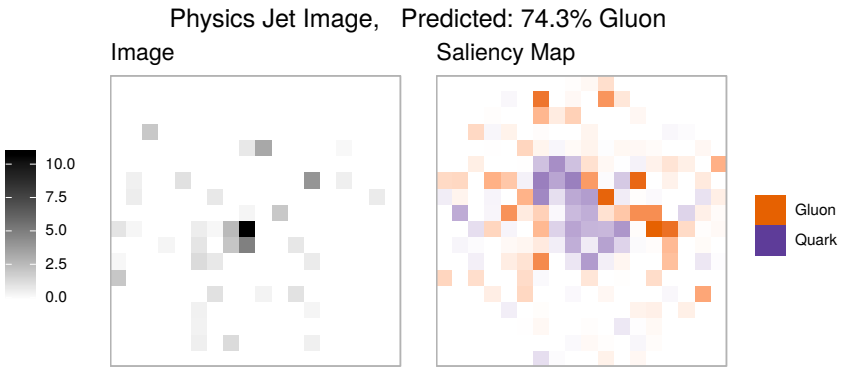
TEXT EXPLANATION

LIME Although it might seem a bit bizarre to see a ... simply enjoy the fun. Mary is a street kid ... older William ... at the time & just on the cusp ... too much of the plot ... great fun to watch ... are very good ... street scenes ...

SLISE Although it might seem a bit bizarre to see a ... simply enjoy the fun. Mary is a street kid ... older William ... at the time & just on the cusp ... too much of the plot ... great fun to watch ... are very good ... street scenes ...

This is a rea world example of the faculty example from the motivation. LIME surprisingly thinks that street is an important word, while SLISE uses the data to realise that it is not.

Physics Example



We also applied SLISE on an particle physics classifier, since this is a domain where it is especially important to adhere to the data constraints. Also known as the laws of physics, for example the conservation of energy.

...

ORGANISATION

- Motivation
- Algorithm
- Robust Regression
- Explanations
- **Conclusions**

CONCLUSIONS

- Novel sparse robust regression approach
- Algorithm, SLISE, that scales well to large datasets
- Can be used for explanations that preserve constraints in the data
- The interaction between the model and the data is important

`https://github.com/edahelsinki/slise`

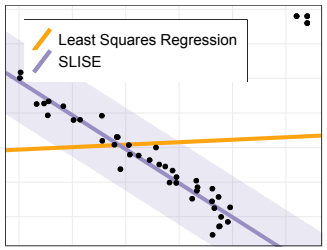
So, in conclusion, we propose a novel approach for robust regression and an algorithm, named SLISE, for finding the solution in a timely manner, even on large datasets.

SLISE can also be used to find explanations for outcomes from black box models, in a way that preserves constraints in the data. The interaction between the model and the data is also important, and SLISE is able to take that into account.

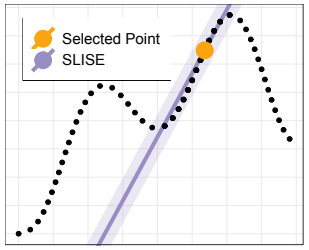
Finally, saving the best part for last, SLISE is open source, and available from Github.

Sparse Robust Regression for Explaining Classifiers

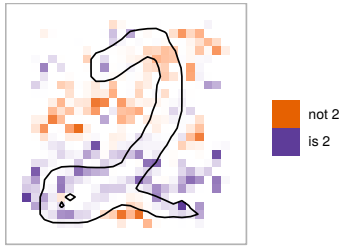
Paper: `https://rdcu.be/bVbda`



Robust Regression



Local Approximation



Outcome Explanation

Code: `https://github.com/edahelsinki/slise`

Thank you for reading!