A Presentation on Projection Pursuit

Statistical Methods IV: 2022-23

Adityarup Laha   Naman Bhaswar
Manas Sharma   Abdur Rahman

Indian Statistical Institute, Kolkata

March 19, 2023

Inspiration of Our Presentation

SLAC-PUB-1312
(M)
September 1973

A PROJECTION PURSUIT ALGORITHM FOR

EXPLORATORY DATA ANALYSIS

Jerome H. Friedman
Stanford Linear Accelerator Center*
Stanford, California 94305

and

John W. Tukey
Princeton University**
Princeton, New Jersey 08540
and
Bell Laboratories
Murray Hill, New Jersey 07974

Origin of an Idea: Friedman and Tukey's Experiment

- Friedman and Tukey used PRIM-9 in their 1974 paper "A Projection Pursuit Algorithm for Exploratory Data Analysis", which introduced the concept of projection pursuit and described a method for using PRIM-9 to identify interesting projections of high-dimensional data.
- PRIM-9 allows users to interactively explore high-dimensional data by visualizing subsets of the data that satisfy certain conditions. The program works by dividing the data into smaller subregions (called "primitives") and allowing users to interactively manipulate these primitives to identify interesting subsets of the data. The program also includes various statistical and graphical tools for analyzing the data, such as scatterplots, histograms, and regression analyses.

Principal Component Analysis

- Principal Component Analysis (PCA) is a statistical technique that is commonly used for dimensionality reduction and data visualization.
- It focuses on finding the direction where the "spread" (variance) of the data is maxmimised. So we maximize $x^T Ax$ where $A = \mathbb{V}\text{ar}(X) = XX^T$ is the covariance matrix. This is a quadratic form, and it is easy to find that $x^T Ax$ is maximized by the eigenvector of A corresponding to the largest eigenvalue. Successive maximas are attained by eigenvectors corresponding to progressively lower eigenvalues.

Linear Discriminant Analysis

- LDA stands for Linear Discriminant Analysis, which is a statistical technique used for dimensionality reduction and classification in machine learning and data science.The goal of LDA is to project the original high-dimensional data onto a lower-dimensional space while maximizing the separation between different classes.
- LDA works by computing the mean and covariance of each class in the data, and then finding a projection that maximizes the ratio of the between-class variance to the within-class variance.
- However, LDA operates on a fundamental assumption – **Data is divided into known Gaussian classes with identical covariance matrices.**

Projection Pursuit: Search of a more robust approach

None of the above techniques could help Friedman and Tukey achieve the goals of their 1973 experiment.

Thus, the limitations of the previous two techniques forced Friedman and Tukey to search for a technique which could have wider applications and get us interesting results.

An Introduction to Projection Pursuit

1. **Projection pursuit** is a statistical technique used in exploratory data analysis to identify interesting low-dimensional projections of high-dimensional data. The basic idea behind projection pursuit is to search for projections that maxmimize some index of "interestingness" or goodness-of-fit (called the *objective function*).

An Introduction to Projection Pursuit

1. **Projection pursuit** is a statistical technique used in exploratory data analysis to identify interesting low-dimensional projections of high-dimensional data. The basic idea behind projection pursuit is to search for projections that maxmimize some index of "interestingness" or goodness-of-fit (called the *objective function*).
2. The choice of objective function depends on the specific problem and the desired properties of the resulting projection. The goal is to optimise for projections that reveal the underlying structure of the data, such as clusters or patterns, deviations from randomness, etc., while suppressing noise and irrelevant features.

An Introduction to Projection Pursuit

1. **Projection pursuit** is a statistical technique used in exploratory data analysis to identify interesting low-dimensional projections of high-dimensional data. The basic idea behind projection pursuit is to search for projections that maxmimize some index of "interestingness" or goodness-of-fit (called the *objective function*).
2. The choice of objective function depends on the specific problem and the desired properties of the resulting projection. The goal is to optimise for projections that reveal the underlying structure of the data, such as clusters or patterns, deviations from randomness, etc., while suppressing noise and irrelevant features.
3. Projection pursuit has been used in various applications, including image analysis, signal processing, and pattern recognition. It can be a powerful tool for identifying important features or patterns in large and complex datasets that may be difficult to analyze using other methods.

Friedman and Tukey's Projection Index

The projection index is a special kind of objective function used in projection pursuit. It is a measure of the "interestingness" of a projection. The idea is to find projections that have high values of the projection index, as these projections are likely to reveal interesting structures in the data.

Friedman and Tukey's Projection Index

The projection index is a special kind of objective function used in projection pursuit. It is a measure of the "interestingness" of a projection. The idea is to find projections that have high values of the projection index, as these projections are likely to reveal interesting structures in the data.

The P-indexes we use to express quantitatively such properties of a projection axis, 2, can be written as a product of two functions :

$$I(\hat{k}) = s(\hat{k}) \cdot d(\hat{k})$$

Where $s(\hat{k})$ measures the spread of the data, and $d(\hat{k})$ describes the "local density" of the points after projection onto $\hat{k}$.

Spread of Data

For $s(\hat{k})$, we take the trimmed standard deviation of the data from the mean as projected onto $\hat{k}$;

$$s_{trim}(\hat{k}) = \sqrt{\frac{1}{N(1-2p)} \sum_{i=pN}^{N(1-p)} (\vec{X}_i \cdot \hat{k} - \bar{X}_{trim})^2}$$

where

$$\bar{X}_{trim} = \sum_{i=pN}^{N(1-p)} \frac{\vec{X}_i \cdot \hat{k}}{(1-2p)N}$$

Here N is the total number of data points. A small fraction, p, of the points that lie at each of the extremes of the projection are omitted. Thus, extreme values of $\vec{X}_i \cdot \hat{k}$ do not contribute to $s(\hat{k})$, which is therefore robust against outliers.

Local Density

For $d(\hat{k})$, we use an average nearness function of the form :

$$d(\hat{k}) = \sum_{1 \leq i < j \leq n} f(r_{ij}) \mathbf{1}(R - r_{ij})$$

where,

$$r_{ij} = |\vec{X}_i \cdot \hat{k} - \vec{X}_j \cdot \hat{k}|$$

The indicator is unity for positive valued arguments and zero for negative values. Thus, the double sum is confined to pairs with $0 \leq r_{ij} < R$. The function $f(r)$ should be monotonically decreasing for increasing r in the range $r \leq R$, reducing to zero at $r = R$. This continuity assures maximum smoothness of the objective function, $I(\hat{k})$.

1 Why?
- History of Projection Pursuit
- Some Existing Projection Techniques
- Overcoming Limitations

2 What?
- Idea of Projection Pursuit
- Friedman and Tukey's Projection Index

3 How?
- Simplifying the Problem
- Optimisation
- Extending to 2D

4 Where?
- Examples given in the paper
  - Artificially generated data
  - Some Real Data
- Exploring More .....

The Problem

To start with, we shall look at 1-dimensional projections of our data. Hence, we denote our objective function as:

$$H_X(\hat{u}) = g(X \cdot \hat{u})$$

where $\hat{u}$ is our $(1 \times d)$ projection vector, and g is some objective function on the projection $X \cdot \hat{u} = X\hat{u}^T$ of our $(n \times d)$ data matrix X onto $\hat{u}$.

The Problem

To start with, we shall look at 1-dimensional projections of our data. Hence, we denote our objective function as:

$$H_X(\hat{u}) = g(X \cdot \hat{u})$$

where $\hat{u}$ is our $(1 \times d)$ projection vector, and g is some objective function on the projection $X \cdot \hat{u} = X\hat{u}^T$ of our $(n \times d)$ data matrix X onto $\hat{u}$.

In order to get the most useful projections, we have to **maximise** the objective function. Hence, this becomes a multivariate optimisation problem.

The Problem

To start with, we shall look at 1-dimensional projections of our data. Hence, we denote our objective function as:

$$H_X(\hat{u}) = g(X \cdot \hat{u})$$

where $\hat{u}$ is our $(1 \times d)$ projection vector, and g is some objective function on the projection $X \cdot \hat{u} = X\hat{u}^T$ of our $(n \times d)$ data matrix X onto $\hat{u}$.

In order to get the most useful projections, we have to **maximise** the objective function. Hence, this becomes a multivariate optimisation problem.

However, it is worth noting that this optimisation is not unconstrained. Our projection vectors (say $\vec{u}$) are not arbitrary, they lie on the unit hypersphere in d dimensions ($S^{d-1} \subset \mathbb{R}^d$).

The Problem

To start with, we shall look at 1-dimensional projections of our data. Hence, we denote our objective function as:

$$H_X(\hat{u}) = g(X \cdot \hat{u})$$

where $\hat{u}$ is our $(1 \times d)$ projection vector, and g is some objective function on the projection $X \cdot \hat{u} = X\hat{u}^T$ of our $(n \times d)$ data matrix X onto $\hat{u}$.

In order to get the most useful projections, we have to **maximise** the objective function. Hence, this becomes a multivariate optimisation problem.

However, it is worth noting that this optimisation is not unconstrained. Our projection vectors (say $\vec{u}$) are not arbitrary, they lie on the unit hypersphere in d dimensions ($S^{d-1} \subset \mathbb{R}^d$).

Turns out, optimising directly on the hypersphere is **hard**. We need a way to work around this.

The Solution

To get rid of the issues with optimising on the hypersphere, we use **stereographic projection**, to convert our d-dimensional optimisation problem on the surface of hypersphere to an easier, $(d-1)$-dimensional optimisation problem on a Euclidean space.

The Solution

To get rid of the issues with optimising on the hypersphere, we use **stereographic projection**, to convert our d-dimensional optimisation problem on the surface of hypersphere to an easier, $(d-1)$-dimensional optimisation problem on a Euclidean space.

$$\psi : S^{d-1} \setminus \{(1, 0, \cdots, 0)\} \subset \mathbb{R}^d \to \mathbb{E}^{d-1} \cong \mathbb{R}^{d-1}$$

defined as

$$\psi : (x_0, x_1, \cdots, x_d) \mapsto \left(\frac{x_1}{1 - x_0}, \cdots, \frac{x_d}{1 - x_0} \right)$$

The Solution

To get rid of the issues with optimising on the hypersphere, we use **stereographic projection**, to convert our d-dimensional optimisation problem on the surface of hypersphere to an easier, $(d-1)$-dimensional optimisation problem on a Euclidean space.

$$\psi : S^{d-1} \setminus \{(1, 0, \cdots, 0)\} \subset \mathbb{R}^d \to \mathbb{E}^{d-1} \cong \mathbb{R}^{d-1}$$

defined as

$$\psi : (x_0, x_1, \cdots, x_d) \mapsto \left(\frac{x_1}{1 - x_0}, \cdots, \frac{x_d}{1 - x_0} \right)$$

Turns out, this is a $\mathcal{C}^\infty$-diffeomorphism with inverse

$$\phi \equiv \psi^{-1} : (y_1, \cdots, y_d) \mapsto \left(\frac{1 - \sum y_j^2}{1 + \sum y_j^2}, \frac{y_1}{1 + \sum y_j^2}, \cdots, \frac{y_d}{1 + \sum y_j^2} \right)$$

Finding the optimal projection

Now, instead of optimising $H_X(\hat{u})$ over S^{d-1}, we optimise $H_X \circ \phi(\vec{v})$ over $\mathbb{E}^{d-1}$. For this, we an algorithm given by Rosenbrock in his 1960 paper *"An Automatic Method for Finding the Greatest or Least Value of a Function"*. This algorithm has the advantage of not needing to compute gradients, which is great as our objective function isn't necessarily differentiable.

Finding the optimal projection

Now, instead of optimising $H_X(\hat{u})$ over S^{d-1}, we optimise $H_X \circ \phi(\vec{v})$ over $\mathbb{E}^{d-1}$. For this, we an algorithm given by Rosenbrock in his 1960 paper *"An Automatic Method for Finding the Greatest or Least Value of a Function"*. This algorithm has the advantage of not needing to compute gradients, which is great as our objective function isn't necessarily differentiable.

The popular Python library `scipy`, which we will be using, implements a variation of this, as the "Nelder-Mead" optimisation method. Once we have the optimal $\vec{v} \in \mathbb{R}^{d-1}$, we get back the optimal $\hat{u}$ as $\hat{u} = \phi(\vec{v})$.

This works because ψ is a $\mathcal{C}^{\infty}$-diffeomorphism, and hence a homeomorphism.

Finding the optimal projection

Now, instead of optimising $H_X(\hat{u})$ over S^{d-1}, we optimise $H_X \circ \phi(\vec{v})$ over $\mathbb{E}^{d-1}$. For this, we an algorithm given by Rosenbrock in his 1960 paper *"An Automatic Method for Finding the Greatest or Least Value of a Function"*. This algorithm has the advantage of not needing to compute gradients, which is great as our objective function isn't necessarily differentiable.

The popular Python library `scipy`, which we will be using, implements a variation of this, as the "Nelder-Mead" optimisation method. Once we have the optimal $\vec{v} \in \mathbb{R}^{d-1}$, we get back the optimal $\hat{u}$ as $\hat{u} = \phi(\vec{v})$.

This works because ψ is a $\mathcal{C}^\infty$-diffeomorphism, and hence a homeomorphism.

To demonstrate that this works, we shall use the variance as the objective function. If all goes right, we should get PCA.

Objective functions in 2D

2-dimensional projections are characterised by 2 unit orthogonal (or simply linearly independent) vectors that determine a 2-plane in $\mathbb{R}^d$. We then try to find the "best" projection by changing $\hat{u}$ and $\hat{v}$.

So, now our objective function looks like $H_X(\hat{u}, \hat{v})$.

Objective functions in 2D

2-dimensional projections are characterised by 2 unit orthogonal (or simply linearly independent) vectors that determine a 2-plane in $\mathbb{R}^d$. We then try to find the "best" projection by changing $\hat{u}$ and $\hat{v}$.

So, now our objective function looks like $H_X(\hat{u}, \hat{v})$.

Of course, it is not possible to now maximise this directly. So we use iterative profiled maximisation.
In this algorithm, we start off with some $\hat{u}_0$ and $\hat{v}_0$, take projections to get $\vec{k_0}$ and $\vec{l_0}$, now iterate as follows:

$$\vec{l_n} = \arg\max_{\vec{l}} H_X(\phi(\vec{k_{n-1}}), \phi(\vec{l})) \qquad \vec{k_n} = \arg\max_{\vec{k}} H_X(\phi(\vec{k}), \phi(\vec{l_n}))$$

We get a $\vec{k}$ and $\vec{l}$ which are (hopefully) linearly independent. From that, we recover our best estimates of $\hat{u}, \hat{v}$.

Applications of Projection Pursuit

To illustrate the application of the algorithm, Friedman and Tukey described its effect upon several data sets. The first two are artificially generated so that the result scan be compared with the known data structure. The third is the well known Iris data used by Fisher and the fourth is a data set taken from a particle physics experiment.

- Uniformly distributed random data
- Gaussian data distributed at the vertices of a simplex
- IRIS data
- Particle physics data

Applications of Projection Pursuit

- **Uniformly distributed random data** – To test the effect of projection pursuit on artificial data having no preferred projection axes, we generated 975 data points, randomly, from a uniform distribution inside a 14-dimensional sphere, and repeatedly applied applied one and two dimensional projection pursuit to the sample with different starting directions.
- **Gaussian data distributed at the vertices of a simplex** – A simplex in n-dimensional space is defined as the convex hull of a set of $n + 1$ affinely independent points. In other words, it is the smallest convex polytope containing those $n + 1$ points as vertices. Centred at every point, we generate spherically symmetric Gaussian data.

Applications of Projection Pursuit

- **IRIS data** – This is a classical data set first used by Fisher and subsequently by many other researchers for testing statistical procedures.The Data consists of measurements made on 50 observations from each of three species of IRIS flowers (one separable from the other two).
- **Particle physics data** – It is a data set taken from a high energy particle physics scattering experiment. In this experiment, a beam of positively charged pi mesons, with an energy of 16 GeV, was used to bombard a stationary target of protons contained in hydrogen nuclei. Five hundred examples were recorded of those nuclear reactions in which the final products were a proton, two positively charged pi mesons, and a negatively charged pi meson.

Different Index (Interestingness)

$$\hat{I}(\alpha) = \frac{1}{2} \cdot \sum_{j=1}^{J} (2j+1) \left[\frac{1}{N} \sum_{i=1}^{N} P_j(2\phi(\alpha^T z_i) - 1) \right]^2$$

where,

- $\hat{I}$: A different projection index
- α : Unit vector specifying direction
- P_j : j^{th} Legendre polynomial
- ϕ : CDF of $N(0,1)$

Projection Pursuit Regression

Projection pursuit regression (PPR) is a non parametric regression technique that aims to model the relationship between a set of predictor variables X and a response variable Y. The PPR algorithm involves searching for a low-dimensional projection of the predictor variables that best captures the relationship with the response variable.

Projection Pursuit Regression

Projection pursuit regression (PPR) is a non parametric regression technique that aims to model the relationship between a set of predictor variables X and a response variable Y. The PPR algorithm involves searching for a low-dimensional projection of the predictor variables that best captures the relationship with the response variable.

The basic PPR model can be expressed as follows:

$$y_i = \beta_0 + \sum_{j=1}^{r} f_j(\beta_j^T x_i) + \epsilon_i$$

Remark: This is just for information and we do not intend to study it further in this presentation, for the sake of brevity.

References

[1] H. H. Rosenbrock, "An automatic method for finding the greatest or least value of a function," *The Computer Journal*, vol. 3, no. 3, pp. 175–184, 1960. DOI: 10.1093/comjnl/3.3.175. [Online]. Available: https://academic.oup.com/comjnl/article/3/3/175/345501.

[2] J. Tukey and J. Friedman, "A Projection Pursuit Algorithm for Exploratory Data Analysis," *IEEE Transactions on Computers*, vol. C-23, no. 9, pp. 881–890, 1974. DOI: 10.1109/T-C.1974.224051. [Online]. Available: https://ieeexplore.ieee.org/document/1672644.

[3] J. H. Friedman and W. Stuetzle, "Projection pursuit regression," *Journal of the American Statistical Association*, vol. 76, no. 376, pp. 817–823, 1981. DOI: 10.1080/01621459.1981.10477727. [Online]. Available: https://www.jstor.org/stable/2287576.

[4] P. J. Huber, "Projection pursuit," *The Annals of Statistics*, vol. 13, no. 2, pp. 435–475, 1985. DOI: 10.1214/aos/1176349519. [Online]. Available: https://www.jstor.org/stable/2241838.

[5] J. Friedman and J. Tukey, "Exploratory projection pursuit," *Journal of the American Statistical Association*, vol. 82, no. 397, pp. 249–266, 1987. DOI: 10.2307/2289468. [Online]. Available: https://www.jstor.org/stable/2289468.

THANK YOU