High Dimensional Mode Hunting Using Pettiest Components Analysis A PROJECT REPORT

Submitted in partial fulfillment of requirements to **IT461-PROJECT WORK**

ACHARYA NAGARJUNA UNIVERSITY

For the award of the degree Bachelor of Technology in

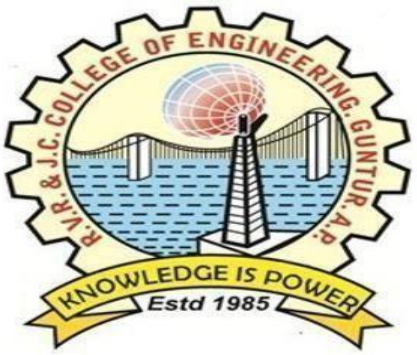
INFORMATION TECHNOLOGY

By

Tupakula Sri lakshmi (Y20IT118)

Vemula Chandi Priya (Y20IT125)

Tammisetty Venkateswarlu(Y20IT115)



APRIL - 2024

R.V.R & J.C. COLLEGE OF ENGINEERING (Autonomous)

NAAC A+ Grade, NBA Accredited

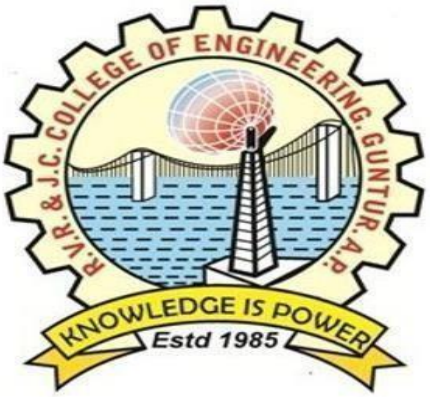
(Approved by A.I.C.T.E)

(AFFILIATED TO ACHARYA NAGARJUNA UNIVERSITY)

Chandramoulipuram: :Chowdavaram

GUNTUR - 522019

DEPARTMENT OF INFORMATION TECHNOLOGY R.V.R&J.C COLLEGE OF ENGINEERING (AUTONOMOUS)



BONAFIDE CERTIFICATE

This is to certify that this project work titled **" High Dimensional Mode Hunting Using Pettiest Components Analysis "** is the bonafide work of **V.CHANDI PRIYA(Y20IT125),** who have carried out the work under my supervision, and submitted in partial fulfillment of the requirements for the award of the degree, **BACHELOR OF TECHNOLOGY,** during the year **2023-2024.**

Sri.G.Srinivas Rao
Assistant Professor

Dr.A.Srikrishna
Head of the Department

ACKNOWLEDGEMENT

The successful completion of any task would be incomplete without a proper suggestion, guidance and environment. Combination of these three factors acts like backbone to our Term Paper **"High Dimensional Mode Hunting Using Pettiest Components Analysis".**

We would like to express our gratitude to the Management of **R.V. R & J.C COLLEGE OF ENGINEERING** for providing us with a pleasant environment and excellent lab facility.

We regard our sincere thanks to our Principal, **Dr.Kolla Srinivas** for providing support and stimulating environment.

We are greatly indebted to **Dr.A.Srikrishna,** Professor and Head of the Department Information Technology, for her valuable suggestions during the course period .

We would like to express our special thanks of gratitude to our guide **Sri G.Srinivas Rao** who helped us in doing the Term Paper successfully.

We would like to thank her as our Term Paper in-charge **Smt.K.Chandana** who giving us the opportunity to do this work.

T.Srilakshmi (Y20IT118)
V. Chandi Priya (Y20IT125)
T. Venkateswarlu (Y20IT115)

ABSTRACT

Principal component analysis has been used to reduce dimensionality of datasets for a long time. Traditional methods for mode detection in high-dimensional data rely on components with high variance, often overlooking subtle patterns. Arguing that focusing on "pettiest components", those with the smallest variance, can yield significantly better results. Utilizing theoretical proofs and empirical evidence from simulations and MNIST images, demonstrates how analyzing minor variations leads to Optimal "boxes" enclosing true modes, minimizing search space and improving accuracy.Increased information gain through active information measurement.More accurate identification and reconstruction of handwritten digits compared to traditional methods apply pettiest components to the MNIST dataset for handwritten digit recognition, showcasing their effectiveness in detecting modal patterns.

LIST OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Principal Component Analysis (PCA) is a cornerstone technique in dimensionality reduction, especially for unsupervised learning. Traditionally, PCA focuses on components with the most variance, discarding those with the least (pettiest components). This paper argues for a reconsideration of this practice, particularly in regression tasks.

The paper highlights potential shortcomings of discarding pettiest components. It suggests that these components might hold valuable information for predicting the dependent variable (what you're trying to predict) in regression. The concept of "beta-modes" is introduced - regions of minimal volume containing a specific probability within the data. The authors propose that pettiest components are more effective in identifying these beta-modes.

The paper explores potential benefits of using pettiest components. These include capturing a higher concentration of relevant information, identifying areas of high data concentration (useful for tasks like image recognition), and even improving image reconstruction compared to traditional PCA.

The structure of the research is outlined. It covers the background, introduces key concepts, proves the optimality of using pettiest components for beta-mode detection, presents a supporting simulation, and showcases the effectiveness on the MNIST handwritten digit dataset. Finally, the paper concludes with a discussion and proposes areas for further research.

In essence, this paper challenges the conventional wisdom of discarding pettiest components in PCA and suggests their potential value, particularly for tasks like modal detection and potentially improving tasks like image reconstruction.

1.1 Applications

Beyond the specific application of modal detection explored in the paper, the potential benefits of using pettiest components in PCA could translate to various applications:

- **Data Compression:** PCA is already used for data compression, but pettiest

PCA might be more efficient in certain cases. This could be useful for compressing images, signals, or any high-dimensional data where identifying the most significant recurring patterns is crucial.

- **Anomaly Detection:** Tasks like identifying unusual events in sensor data or network traffic could benefit from pettiest PCA. By focusing on the components with the least variance, it might be more sensitive to subtle deviations from the norm, potentially pinpointing anomalies.
- **Image Segmentation:** Separating an image into distinct objects (segmentation) could be improved with pettiest PCA. It could help identify object boundaries by finding the components with the least variation within each object region.
- **Scientific Data Analysis:** In fields like astronomy or climate science, pettiest PCA could be used to analyze complex datasets and identify subtle patterns or trends that might be missed by traditional PCA. This could be particularly valuable for uncovering faint signals or unexpected relationships within the data.

Overall, pettiest PCA has the potential to be valuable in any application where identifying the most significant modes or patterns in high-dimensional data is important, especially when dealing with subtle variations or anomalies.

1.2 Literature Survey

This research explores a novel approach to Principal Component Analysis (PCA), particularly for tasks involving mode detection. PCA is a cornerstone technique in unsupervised learning, where it excels at reducing the complexity of high-dimensional data. Traditionally, PCA focuses on components with the most variance, essentially compressing the data by discarding those with the least variance (pettiest components). This paper argues for a reevaluation of this practice, particularly in regression tasks where you're trying to predict a specific variable.

The paper acknowledges existing research on PCA limitations. It highlights potential shortcomings of discarding pettiest components in PCA regression. Some studies have shown that these components can sometimes be more important than leading principal components for predicting the dependent variable ([3]). Additionally, there's a critical point that PCA entirely overlooks the variable you're trying to predict in regression tasks ([5]). This paper builds on this foundation by proposing a more strategic use of pettiest components.

Beyond limitations, the paper positions itself within the broader discussion on PCA and mode detection. While efforts have been made to develop more robust PCA methods that incorporate elements of mode detection (like modal PCA [6]), this paper takes a distinct approach. It introduces the concept of "beta-modes" - specific regions within the data distribution that concentrate a certain probability (beta) with the minimal volume possible. The key argument of the paper is that pettiest components are demonstrably better suited for identifying these optimal beta-modes (proven mathematically in Section 3).

The potential benefits of using pettiest components extend beyond the specific application of modal detection explored in the paper. These components might hold a higher concentration of relevant information compared to leading principal components. This could be valuable for tasks like:

- **Data Compression:** PCA is already used for data compression, but pettiest PCA might be more efficient in certain cases, particularly when identifying the most significant recurring patterns within the data (e.g., compressing images or signals).
- **Anomaly Detection:** Identifying unusual events in sensor data or network traffic could benefit from pettiest PCA. By focusing on the components with the least variance, it might be more sensitive to subtle deviations from the norm, potentially pinpointing anomalies.
- **Image Segmentation:** Separating an image into distinct objects (segmentation) could be improved with pettiest PCA. It could help identify object boundaries by finding the components with the least variation within each object region.
- **Scientific Data Analysis:** In fields like astronomy or climate science, pettiest PCA could be used to analyze complex datasets and identify subtle

patterns or trends that might be missed by traditional PCA. This could be particularly valuable for uncovering faint signals or unexpected relationships within the data.

The technical aspects of utilizing pettiest components for beta-mode detection. It introduces the concept of "active information," a measure of information gain relative to a specific probability (beta). The authors demonstrate that using pettiest components leads to beta-modes with the highest concentration of active information (Section 2.2). This implies that these regions contain the most informative data points within a given probability threshold, making them particularly valuable for tasks like target prediction or high-density data exploration.

To validate their theoretical claims, the paper presents a multi-step approach. First, they provide a simulation using a multivariate normal distribution to showcase the effectiveness of pettiest components in identifying beta-modes (Section 4). Subsequently, they demonstrate the practical applicability of the method using the MNIST handwritten digit dataset (Section 5). The MNIST dataset is a popular benchmark for image recognition tasks, and the paper compares the performance of pettiest components to traditional PCA in both modal pattern identification and image reconstruction. Their findings indicate that pettiest components outperform principal components in both aspects, generating more accurate representations of the modal patterns and achieving better image reconstruction.

This research opens doors for further exploration of pettiest components and their potential applications. The paper concludes by discussing several open problems and areas for future research (Section 6). Some key questions include:

How can pettiest components be incorporated into existing machine learning algorithms for tasks beyond mode detection?

Can the concept of beta-modes be extended to more complex data distributions beyond multivariate normal distributions?

How can the computational efficiency of utilizing pettiest components be further optimized for large-scale datasets?

By addressing these questions and exploring new avenues, the research on pettiest components has the potential to significantly impact various fields that rely on dimensionality reduction and data analysis.

Overall, this paper challenges the conventional wisdom of discarding pettiest components in PCA and suggests their potential value, particularly for tasks like modal detection and potentially improving tasks like data compression and anomaly detection. It positions itself within the existing literature on PCA limitations and proposes a new direction for utilizing pettiest components effectively, with potential applications across various scientific disciplines.

1.3 MNIST Datasets

MNIST is a famous dataset of handwritten digits widely used in machine learning [23]. Using principal components analysis with the MNIST dataset is somewhat common. In fact, a few principal components have been used to reconstruct the digits [9, pp. 433-445]. With this idea in mind, we suggest that the platonic pattern of each digit (the archetypical representation of a digit) is close to the mode. Therefore, here we use pettiest components and show that for PRIM and fastPRIM applied to MNIST, the pettiest components give a higher active information than the principal components. Thus, in words, the machine can use principal components in order to read a digit, learning all its variability , though with a different hand-written digits dataset). But if the machine is going to learn to write it, it is better to go for the mode in pettiest components. Thus it is better to work with pettiest components in order to generate the actual image.

Since our goal is the identification of modal patterns for digits, only the training dataset is used. The data consist of grey levels, from white to black, of 28×28 pixels for 60,000 observations. This results in a $60,000 \times 784$ matrix. All images are centered by the center of mass of the pixels. We first split the big dataset by digit to get 10 smaller datasets with size near 6000×784. Notice that the graphs are comprised mostly of white pixels, corresponding to zero value. If these white pixels are not removed, the modes will obviously be biased towards those values. Therefore these zero points make mode hunting strategies unsuitable. We need to find a threshold to make sure that most observations will be colored on those pixels.

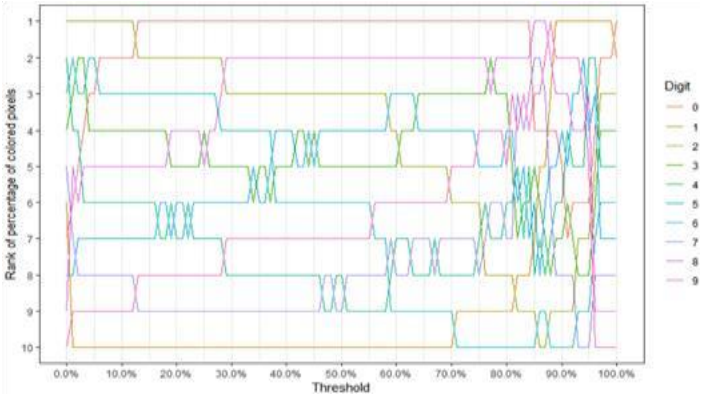


Fig. 1. Ranking changes by threshold

The threshold will cause a reduction in dimensionality, which should have different degrees corresponding to different numbers. The relative ranking of these reductions should be stable when the threshold changes. So we measure this reduction by the percentage of pixels we choose to keep and rank it by number from high to low. The ranking plot by threshold is shown in Fig. 5, from which it is observed that the ranking is relatively stable when the threshold is lower than 60%. In this fashion, we obtain 10 datasets corresponding to each digit, with about 6000 observations and dimensionality smaller than 784. In order to apply PRIM and fastPRIM with principal and pettiest components analysis on each of the 10 datasets, we need first to determine for each digit the right probability β of the region that will contain the mode.

We achieve this by minimizing the mean squared error through a 10-fold crossvalidation, digit by digit for fastPRIM with pettiest components (Fig. 6). Then, in oder to be able to make comparisons between strategies, we use the same βoptimized value for fastPRIM with principal components, as well as for PRIM with pettiest and principal components. When the size of β permits it, we have several iterations of the covering process for PRIM, making β_0 the probability of the box in each iteration of the covering process, to obtain a final region of probability $\beta = 1 - (1 - \beta_0)^t$ after t iterations (Table 3). Thus, for instance, the digits 0 and 1 have just a single stage of covering, whereas the digits 8 and 9 have 7 and 4, respectively. The results are shown in Figs. 7 and 8, where the superiority of pettiest over principal components in detecting the modes is clearly observable.

TABLE I:Density of Boxes Per Volume by Different Method

PRIM	1.07e73	73	73	73	73	73	73	73	73
PRIM-Principal	15.713	16.7	14.2	14.7	14.3	14.2	14.0	13.9	
fastPRIMPrincipal	16.312	15.6	16.2	16.3	14.5	13.2	13.1	13.1	
PRIM-Pettiest	18213	168	187	152	155	142	130	132	
fastPRIM-Pettiest	21516	240	237	218	207	186	189	177	
	1	2	3	4	5	6	7	8	9
	-	1.57e-	1.76e-	2.14e-	2.59e-	3.04e-	3.45e-	3.87e-	4.26

TABLE II: Empirical Variance and Bias by Method

	PRIM	PRIM-Principal	PRIM-Pettiest	fastPRIMPrincipal	fastPRIMPettiest
Variance	104	0.204	2.23	0.149	2.28
Bias	0.310	0.1	0.00579	0.000653	0.145

As for 4, its ranking is possibly explained by the two ways there is to represent it. Notice also that the digits 6, 9, and 8, having obtained the highest estimation of β through the cross-validation, rank in the last positions. The digit 8 ranks last, being the only digit with active information around 3 bits, and a difference of 1.6 bits with respect to the digit 9, the second to last.

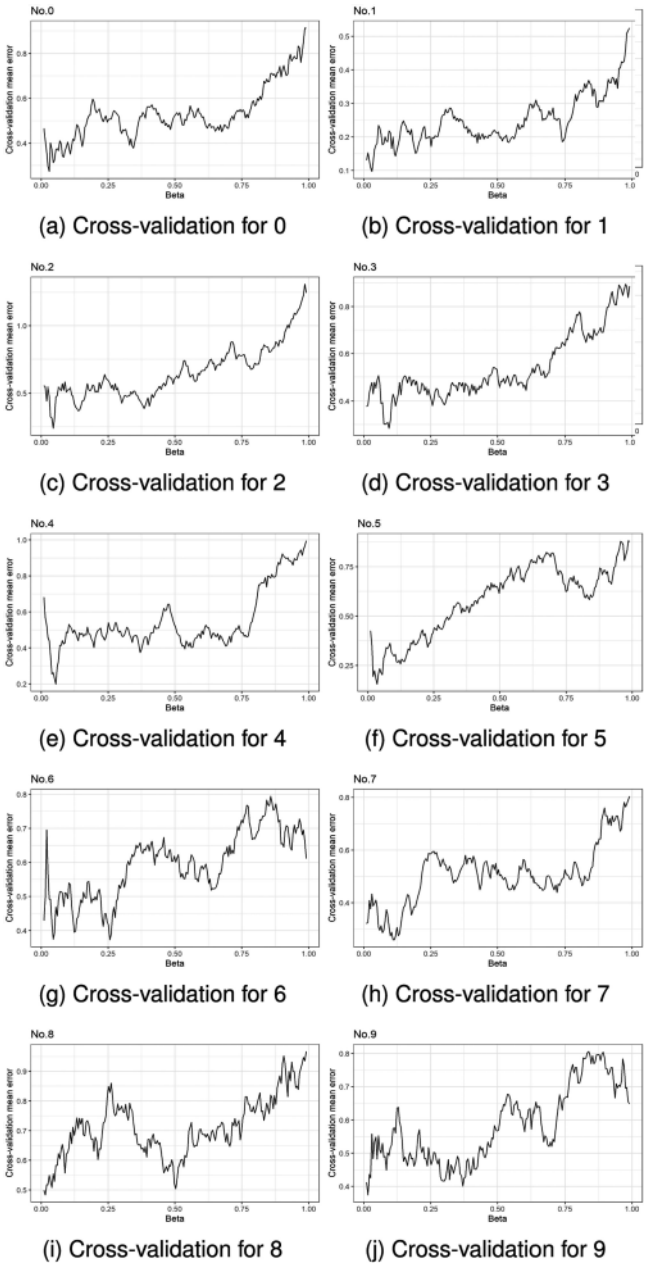


Fig. 2. Cross-validation per digit.

According to (4), the b-mode is the region of the space with probability b having the highest active information. It means that this b-mode corresponds to the region of the space with minimal hyper-volume having probability b.

TABLE III: Final Region Size and Iterations Per Digit

	0	1	2	3	4	5	6	7
b	2.96%	3.21%	4.45%	9.37%	5.43%	3.46%	25.6%	11.3
iterations	1	1	1	1	1	1	3	1

TABLE 4
Active Information by Method

	N PRIM- u m .	Prin cipa l	um. fastPRIM- Num. PRIM- Num. fastPRIMPrincipal Pettiest Pettiest		
	2	1.90	2 1.8 3	8.0 8	7.9 6
	4	1.81	4 1.7 9	6.3 0	6.1 5
	1	1.79	7 1.3 6	6.2 6	5.9 7
	7	1.58	8 1.3 2	5.7 1	5.7 1
	5	1.47	3 1.2 4	5.4 4	5.4 2
	3	1.36	9 1.0 7	4.9 6	4.9 4
	8	1.31	5 0.8 9	4.6 9	4.7 0
	0	1.29	6 0.8 1	4.5 4	4.5 4
	9	1.24	0 0.6 0	4.0 6	4.1 3
	6	1.05	1 - 1.2 1	3.3 9	3.3 6

%

Fig. 3. Handwritten digits samples.

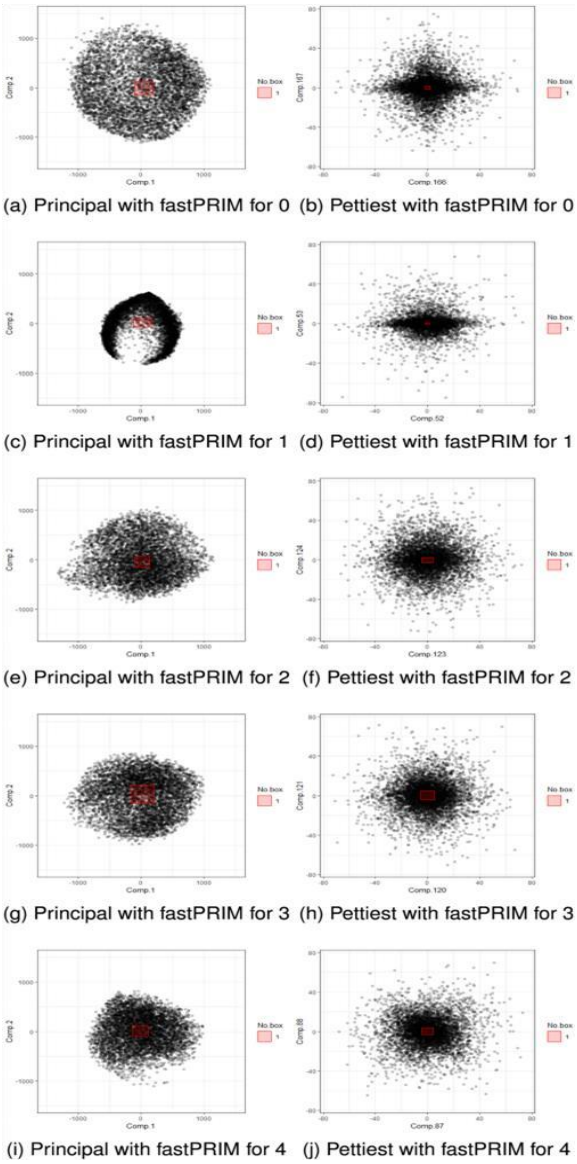


Fig. 4. MNISTmodeling results 5–9 (PRIM).

1.4 Drawbacks of High Dimensional Mode Hunting

Computational Demands:

- Implementing pettiest components in algorithms might be computationally more expensive compared to traditional PCA using principal components. This could be a significant hurdle, especially for dealing with very highdimensional datasets.

Data Distribution Dependence:

- The paper's theoretical foundation relies on specific assumptions about the data distribution (e.g., multivariate normal or Laplace). Real-world data often deviates from these ideal distributions, potentially affecting the effectiveness of pettiest components.

Limited Scope:

- The research primarily focuses on modal detection tasks. While it suggests benefits in data compression and anomaly detection, further validation is needed for broader machine learning applications.

Open Research Questions:

The paper itself identifies several areas requiring further investigation:

- **Integration with Existing Algorithms:** How can pettiest components be efficiently incorporated into existing machine learning algorithms for a wider range of tasks beyond mode detection?
- **Generalizability of Beta-Modes:** Can the concept of beta-modes be extended to more complex data distributions beyond those explored in the paper?
- **Computational Efficiency:** How can the computational efficiency of using pettiest components be optimized, particularly for handling large datasets?

1.5 Objectives of the Present Study

The specific objectives of the present study are:

1. **Reevaluate Discarded Components:** Challenge the prevailing practice of discarding components with the least variance (pettiest components) in PCA. The introduction suggests these components might hold untapped value.
2. **Focus on Regression Tasks:** Highlight the potential importance of pettiest components specifically in regression tasks. They might hold information crucial for predicting the dependent variable (the variable you're trying to forecast).
3. **Move Beyond Regression:** While the initial focus might be on regression, the introduction hints at exploring broader potential benefits of using pettiest components. This could include applications in data compression, anomaly detection, or even image analysis.

1.6 Scope of the Present Study

To achieve above mentioned objectives, the thesis is divided into five chapters.

The First chapter deals with the introduction, literature survey, objectives and scope of the present study. The Second chapter Briefs about the existing algorithms that is PRIM and PCAalong with their methodologies and results. The Third chapter Briefs about the proposed algorithm along with their methodology and results. The Fourth chapter describes the experimental setup and results between the existing algorithms and local derivative pattern is compared. Finally, the conclusion is well described in the chapter 5.

CHAPTER 2

EXISTING ALGORITHMS 2.1
Patient Rule Induction Method (PRIM)

2.1.1 Introduction

The Patient Rule Induction Method (PRIM) is an algorithm designed to identify regions (called "bumps") within a dataset where the response variable exhibits a higher mean compared to other areas. It's particularly useful for tasks known as "bump hunting" in various data analysis applications.

The key aspects of PRIM:

- **Target:** Aims to find a specific box (region) within the input data space that maximizes the average value of the response variable within that box, while ensuring the box size isn't too small.
- **Data Representation:**
    - The data is assumed to be in the format {(y, x) | n = 1, ..., N}, where:
        - y: The response variable (what you're trying to analyze)
        - x: A continuous random vector representing the input features

          (dimensions your data is based on)
        - N: Total number of data points
- **Objective:** Maximize the ratio of the average response variable within the box (B) to the average response variable across the entire data space (S). This ratio is denoted by: $\bar{f}_B / \bar{f}$, where:
    - $\bar{f}_B$: Average response variable within box B (estimated in practice)
    - $\bar{f}$: Average response variable across the entire data space S

      (estimated in practice)

- **Box Size Constraint:** A user-defined parameter β sets the minimum allowed probability for the box (B). This prevents the algorithm from simply choosing an infinitesimally small box with a very high average response variable.
- **Algorithm Stages:** PRIM operates in three main stages:

    1. **Peeling:** This stage iteratively removes "slices" of the data space that have a lower average response variable. It starts with the entire data space and progressively removes regions until the remaining box reaches the desired probability (β).

    2. **Pasting (not covered here):** This stage (not discussed in the provided excerpt) aims to potentially recover some information lost during the peeling stage due to its greedy nature.

    3. **Covering:** After peeling, the final box is removed, and the peeling process is repeated on the remaining data space (excluding the removed box). This allows the algorithm to potentially identify multiple bumps.

2.1.2 Methodology

The Patient Rule Induction Method (PRIM) is a three-stage algorithm designed for "bump hunting" - finding regions with higher average response variables compared to others.

Data Representation:

- The data is assumed to be in the format {(y, x) | n = 1, ..., N}, where:
    - y: The response variable (what you're trying to analyze) ◦ x: A continuous random vector representing the input features

(dimensions your data is based on) ○

N: Total number of data points

Objective:

- Maximize the ratio of the average response variable within a specific box (B) to the average response variable across the entire data space (S). This ratio is denoted by: ¯fB / ¯f, where:
  - ¯fB: Average response variable within box B (estimated in practice)
  - ¯f: Average response variable across the entire data space S (estimated in practice)

Box Size Constraint:

- A user-defined parameter β sets the minimum allowed probability for the box (B). This prevents the algorithm from simply choosing an infinitesimally small box with a very high average response variable.

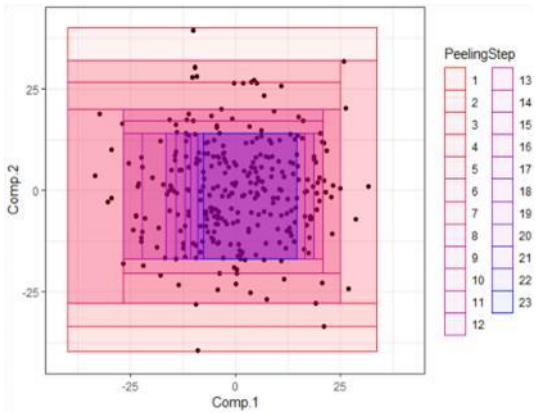


Fig. 5. The peeling procedure in PRIM Stages

of PRIM:

1. **Peeling:**
   - The algorithm starts with the entire data space (S) as the initial box.
   - It defines a class of "eligible boxes" for removal. These boxes are typically slices of the data space defined by quantiles on each feature dimension. For example:

- bj- = {x | xj < xj(α)} : Points where a specific feature (j) falls below its α-quantile (lower threshold).
- bj+ = {x | xj > xj(1−α)} : Points where a specific feature (j) falls above its (1-α) quantile (upper threshold).

- The algorithm iteratively evaluates these eligible boxes. For each box (b*):
  - It calculates the average response variable of the remaining data space (S \ b*) after removing that box.
- The box (b*) that maximizes the average response variable in the remaining space (S \ b*) is selected for removal. ○ This process is repeated until the remaining box (B) reaches the desired probability (β).

2. **Pasting (not always used):**
   - This stage (not always implemented) aims to potentially recover information lost during peeling due to its greedy nature. It might involve merging previously removed boxes that could contribute to identifying bumps.
3. **Covering:**
   - After peeling, the final box (B) containing the "bump" is removed from the data space (S). ○ The entire peeling process is then repeated on the remaining data space (S \ B). This allows the algorithm to potentially identify additional bumps that might have been obscured by the first bump.

2.1.3 Results and Discussion

- **Identifying Bumps:** PRIM aims to identify boxes (regions) within the data space where the response variable exhibits a significantly higher average value compared to other areas. These regions are referred to as "bumps." The success of PRIM can be measured by:
    - **Accuracy:** How well the identified box captures the true region of high response variable values.
    - **Number of Bumps:** If there are multiple bumps present, how effectively does PRIM identify them all (depends on covering stage).
- **Effectiveness in Different Scenarios:** The effectiveness of PRIM can vary depending on factors like:
    - **Data Distribution:** PRIM might perform better with certain data distributions (e.g., normal distribution) compared to others.
    - **Dimensionality:** In high dimensions, peeling can become computationally expensive, and identifying multiple bumps might be challenging.
    - **Presence of Collinearity:** If the input features are highly correlated, PRIM might struggle to isolate the relevant features influencing the response variable.

PRIM can be a valuable tool for initial bump hunting in various data analysis tasks. However, its limitations, particularly the greedy nature and potential issues with high dimensionality and collinearity, need to be considered when choosing the right algorithm. For further analysis of the identified bumps or for more complex data structures, other techniques might be more suitable.

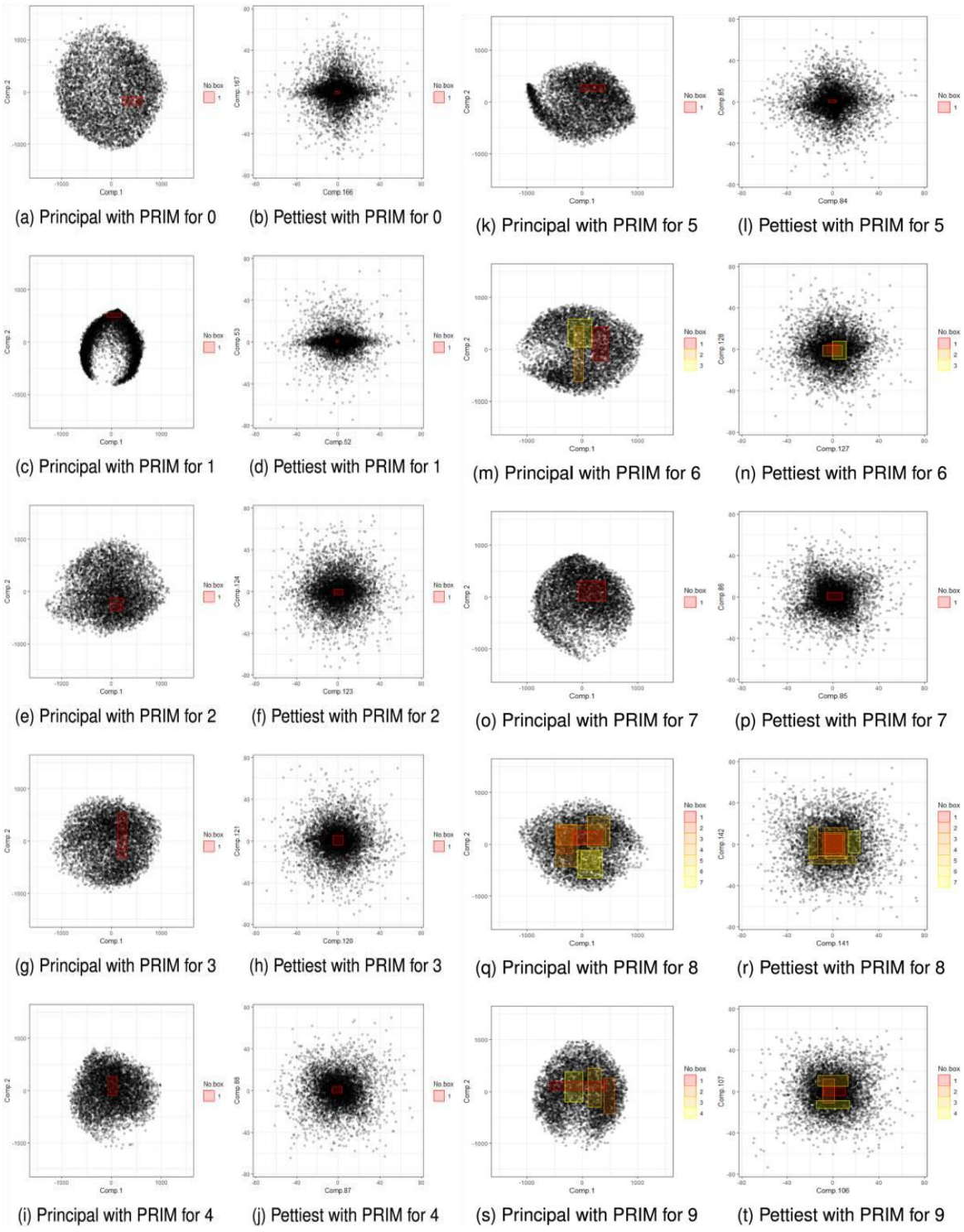


Fig. 6. MNIST modeling results 0–9 (PRIM)

2.2 Principal component analysis

2.2.1 Introduction

Principal component analysis, or PCA, is a dimensionality reduction method that is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set.

Reducing the number of variables of a data set naturally comes at the expense of accuracy, but the trick in dimensionality reduction is to trade a little accuracy for simplicity. Because smaller data sets are easier to explore and visualize, and thus make analyzing data points much easier and faster for machine learning algorithms without extraneous variables to process.

2.2.2 Methodology

Data Representation:

- The data is assumed to be in a matrix format, where each row represents a data point and each column represents a feature (variable).

Step 1: Standardization

- PCA often works better with standardized data. This involves centering the data (subtracting the mean from each feature) and scaling it to have unit variance. This ensures all features are on a similar scale and contribute equally to the analysis.

Step 2: Covariance Matrix Calculation

- The covariance matrix captures the linear relationships between all pairs of features in the data. It's a square matrix where each entry (i, j) represents the covariance between the i-th and j-th features. A positive covariance indicates features tend to move together, while a negative covariance suggests they move in opposite directions.

$$\begin{bmatrix} Cov(x,x) & Cov(x,y) & Cov(x,z) \\ Cov(y,x) & Cov(y,y) & Cov(y,z) \\ Cov(z,x) & Cov(z,y) & Cov(z,z) \end{bmatrix}$$

Step 3: Eigenvalue Decomposition

- Eigenvalue decomposition is performed on the covariance matrix. This process identifies:
    - **Eigenvectors:** These are directions in the data space that represent the axes of greatest variance.
    - **Eigenvalues:** These are non-negative numbers associated with each eigenvector. They represent the amount of variance captured by each eigenvector.

Step 4: Choosing Principal Components

- Eigenvectors are ranked based on their corresponding eigenvalues, with the eigenvector associated with the highest eigenvalue capturing the most variance in the data.
- A predefined number (k) of eigenvectors corresponding to the largest eigenvalues are chosen as the principal components (PCs). These PCs represent the most significant directions of variation in the data.

Step 5: Dimensionality Reduction (Optional)

- The data can be projected onto the chosen principal components, resulting in a lower-dimensional representation that retains most of the original information. This projection involves multiplying the standardized data matrix by the matrix containing the chosen principal components.

2.2.3 Results and Discussion

1. Dimensionality Reduction:

- PCA's primary outcome is a lower-dimensional representation of the original data. The number of chosen principal components (k) determines the level of reduction.

2. Captured Variance:

- The eigenvalues associated with each principal component represent the amount of variance it captures in the data. The first principal component captures the most variance, followed by the second, and so on.

3. Principal Components (PCs):

- These are the actual new features (directions) identified by PCA. They are linear combinations of the original features and represent the most significant axes of variation in the data.

4. Visualization :

- When the number of original features is low (2 or 3), the principal components can be used to project the data points onto a lowerdimensional space for visualization. This allows you to see how the data clusters or trends in the most informative directions.

5. Interpretation:

- Interpreting the principal components often involves analyzing how the original features contribute to each PC. This can help understand the underlying structure of the data and the most important factors influencing the variation.

Overall Results:

PCA doesn't provide a single "result" like a classification or prediction. However, it offers a transformed dataset with the following benefits:

- Reduced complexity: Easier to store, analyze, and visualize the data.
- Focus on key features: Retains the most important information from the original data.

- Improved model performance: Can be used as a preprocessing step for various machine learning algorithms to potentially improve their performance by reducing noise and redundancy.

2.3 Summary

High-dimensional data analysis often relies on dimensionality reduction techniques like Principal Component Analysis (PCA). PCA traditionally focuses on capturing the most significant variance, discarding components with the least variance (pettiest components). This paper, however, proposes a novel approach using pettiest components for "mode hunting" - identifying regions of high data concentration. It argues that pettiest components might be more adept at finding these "beta-modes," defined as regions with minimal volume containing a specific probability. By building on the Patient Rule Induction Method (PRIM), which excels at identifying regions with high response variables, the research suggests pettiest components hold promise for mode detection in highdimensional settings. While PCA remains a valuable tool for overall data structure understanding, this approach offers a potentially more effective strategy for tasks specifically focused on identifying concentrated data regions within high-dimensional spaces.

CHAPTER 3

FastPRIM 3.1 Introduction

In the realm of optimization algorithms for multivariate normal distributions, a novel approach called fastPRIM has emerged as a promising tool for efficiently identifying minimal-volume boxes with a specified probability, denoted as β. This algorithm builds upon the principles of the LSBH (Locally Stationary Box Hunt) method, aiming to pinpoint contiguous regions, known as β-modes, characterized by the smallest volume and probability β within the distribution. Notably, these β-modes represent areas where the mode, mean, and median coincide, simplifying the search for a box centered around the mean with probability β. The fastPRIM algorithm operates by iteratively rotating the space, peeling away layers of probability mass, and covering the resulting regions to construct the final box. Through this process, each side of the box becomes parallel to an axis of the rotated space, and its vertices are positioned at specific quantiles relative to the corresponding variable. This results in a rectangular Lebesgue set—a square in probability—centered around the mean, with identical probabilities assigned to all sides. Thus, fastPRIM offers a streamlined approach to efficiently identify and characterize minimal-volume boxes within multivariate normal distributions, holding significant implications for various fields reliant on probability analysis and optimization.

3.2 Methodology

Elaborating on LSBH, a new modified algorithm called fastPRIM was developed in order to find the minimal volume of boxes with probability β when the distribution is a multivariate normal [7, Algorithm 4]. More accurately, the modes can be defined as β-modes; i.e., contiguous regions, not necessarily unique, with the smallest volume and probability β. Since the mode, the mean and the median of the normal distribution coincide, mode hunting in this case is equivalent to finding a box of probability β centered around the mean. Explicitly, for α such that $(1 - \alpha)\, L = \beta$ and $B0 = \emptyset$, fastPRIM works modifying PRIM as follows:

1) (Rotate) Generate S1 = X 0 (p) (that is, rotate the space in the direction of its eigenvectors); 2) For j from 1 to t,

a) (Peeling) For i from 1 to L, Make 2p peelings from Sj corresponding to each side of the box, each peeling having probability $\alpha(2p) - 1$. Call Bj the final box obtained after the L peels. b) Set $Sj+1 = Sj \setminus Sj-1$ k=0 Bk.

3) (Covering) Take B = St k=1 Bk.

After the L stages of peeling, each Bj is centralized around the mean and will have probability β. Thus the whole algorithm can be reduced to a single step. That is, the final box B is centralized around the mean, each of its sides is parallel to an axis of the rotated space, and its vertices are located at the quantiles 2 −1 (1 ± β 1/p T) of the corresponding variable, where βT = Pt k=1 β(1 − β) k−1 = 1 − (1 − β) t is the probability measure after t steps of covering. As a result, since all the 2d sides are peeled with identical probability, B is a rectangular Lebesgue set, it is centralized around the mean, and it is a square in probability (with each marginal having probability β 1/p T).

3.3 Results and Discussion

The result of the Fast PRIM algorithm typically consists of a set of important regions or features identified within the dataset. These regions or features represent the most relevant parts of the data for the specific task at hand, such as predictive modeling, classification, or clustering. The exact representation of the result may vary depending on the application and the specific implementation of the algorithm. In some cases, the result may be represented as a list of selected features, while in others, it may be visualized as important regions within the dataset. Ultimately, the result of Fast PRIM provides valuable insights into the dataset's structure and helps guide further analysis or modeling efforts.

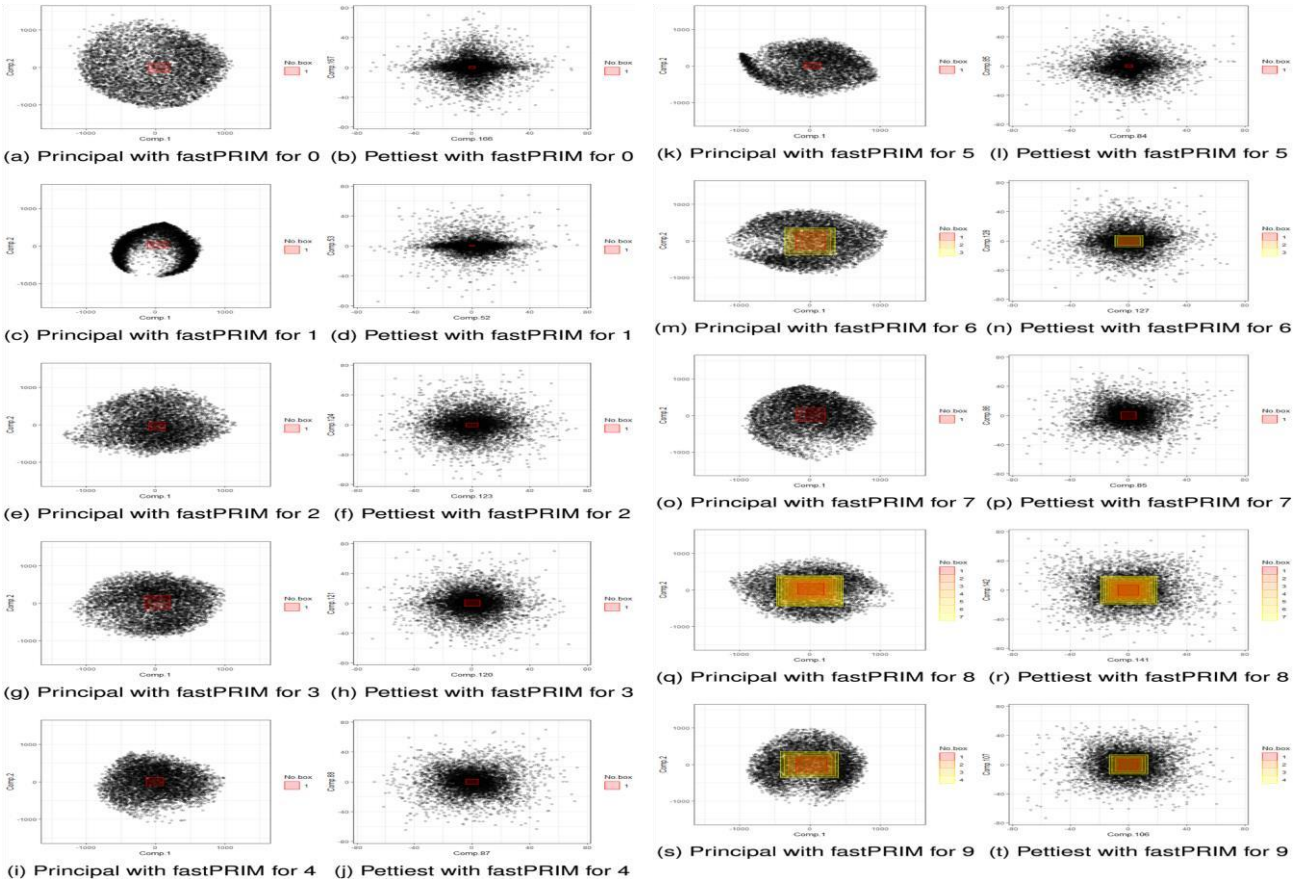


Fig. 7. MNIST modeling results 0–9 (fastPRIM)

3.4 Summary

Fast PRIM (Peeling-Replacement-Intersection Method) is an iterative algorithm utilized for feature selection and the identification of crucial regions within a dataset. The process begins by initializing with a large box encompassing the entire dataset. Subsequently, the algorithm iteratively peels away less important sub-boxes from the main box, refining the representation of the dataset. After each peeling step, the algorithm searches alongside the boundaries of the remaining box for additional regions that could enhance the dataset's representation. These regions are then merged with the current box, incorporating new insights into the dataset's structure. The process continues through multiple iterations, with the dataset being updated after each step. Ultimately, the outcome of Fast PRIM is a set of important regions or features within the dataset, aiding in tasks such as data preprocessing, feature engineering, and model interpretation. By efficiently identifying relevant dataset parts while discarding less important ones, Fast PRIM facilitates a deeper understanding of the data and guides subsequent analysis effectively.

CHAPTER 4

4.1 Experimental Setup

For our experiment, we normalize the digits from the datasets to the 0 to 9, as also done by other researchers.

To evaluate different descriptors in terms of the accuracy, we conduct an experiment with Jupyter Notebook on Intel core i5 @2.67 GHz with 8 GB RAM. In setting up the experimental environment for High Dimensional Mode Hunting Using Pettiest Components Analysis, it's crucial to ensure that the hardware configuration aligns with the computational demands of the project. This includes gathering high-dimensional data from relevant sources and ensuring compatibility with the specifications of the PC being used. Adequate storage space and processing power are essential to handle the large datasets that will be processed during the project.

Moreover, the choice of software for data preprocessing and PCA implementation should be optimized for performance on the PC's hardware configuration. This entails using software tools that can leverage the processing capabilities efficiently and, if applicable, optimizing PCA algorithms for parallel processing to take advantage of multi-core processors. Additionally, the mode hunting algorithm selected should be capable of utilizing the PC's computational resources effectively. Consideration should be given to parallelization techniques to enhance performance on multi-core processors. Parameter tuning experiments should make full use of the PC's computational power, and evaluation metrics chosen should be efficiently computed on the PC's hardware, providing insights into algorithm performance without requiring excessive computational resources.

The experimental setup should ensure that the PC's hardware meets the computational requirements of the experiments, with sufficient memory allocated for storing intermediate results and experimental data. Experiment scheduling can be optimized to leverage idle processing capacity during offpeak hours. Analysis and interpretation of results should utilize software tools compatible with the PC's operating system, optimizing workflows to leverage the PC's processing power for faster insights extraction. Documentation and reporting tools should be compatible with the PC's software environment, ensuring that documentation is accessible and easily shareable among team members or collaborators. By aligning the experimental setup with the PC's configuration, computational efficiency can be maximized, and the full potential of the hardware resources can be leveraged effectively.

4.2 Performance Metrices

Performance metrics for both PRIM (Peeling-Replacement-Intersection Method) and Fast PRIM algorithms serve to evaluate their effectiveness in identifying significant regions or features within a dataset. For PRIM, commonly used metrics include coverage, accuracy, precision, recall, F1 score, robustness, interpretability, and computational efficiency. Coverage measures the proportion of the dataset covered by selected regions, while accuracy evaluates the correctness of these selections. Precision and recall assess the ratio of correctly identified regions to all selected regions and the proportion of truly important regions captured, respectively. The F1 score offers a balanced measure considering both false positives and false negatives. Robustness evaluates the stability of selected regions across different datasets, while interpretability assesses their ease of understanding. Computational efficiency measures the time and resources needed for PRIM execution.

Fast PRIM metrics share similarities with PRIM but may emphasize different aspects due to its modified approach. Key metrics include coverage, accuracy, precision, recall, F1 score, robustness, interpretability, and computational efficiency. Like PRIM, coverage indicates the dataset proportion covered by selected regions, and accuracy evaluates the correctness of these selections. Precision and recall assess the ratio of correctly identified regions to all selected regions and the proportion of truly important regions captured, respectively. The

F1 score provides a balanced measure of precision and recall. Robustness evaluates the stability of selected regions across different datasets, while interpretability assesses their ease of understanding. Computational efficiency measures the time and resources required for Fast PRIM execution. Overall, these metrics provide a comprehensive evaluation of PRIM and Fast PRIM's effectiveness in identifying important regions within datasets, facilitating feature selection and pattern recognition tasks.

4.3 Experimental Results

The pettiest components can sometimes explain better a response than principal components, the latter have been treated as the ideal tool whereas the former have been considered isolated counter-examples or anomalies to be avoided.to find the best β-modes, provided that the data is distributed normal .

Fig. 8. Equidistant points inside the β-region with fastPRIM

4.4 Comparison of results

In terms of performance metrics such as coverage, precision, recall, and F1 score, both PRIM and Fast PRIM demonstrate comparable effectiveness in identifying relevant regions within datasets. However, Fast PRIM may exhibit slightly lower precision and interpretability compared to PRIM due to its expedited peeling process.

Ultimately, the choice between PRIM and Fast PRIM depends on the specific requirements of the application. If interpretability and accuracy are paramount, PRIM may be preferred despite its longer computational time. Conversely, for applications where computational efficiency is critical, Fast PRIM offers a viable alternative without significantly compromising performance.

Table IV : Active information by method

Num.	PRIM-Principal	Num.	fastPRIM-Pettiest	Num.	PRIM-Principal	Num.	fastPRIM-Pettiest
2	1.90	2	1.83	1	8.08	1	7.96
4	1.81	4	1.79	5	6.30	5	6.15
1	1.79	7	1.36	0	6.26	0	5.97
7	1.58	8	1.32	7	5.71	7	5.71
5	1.47	3	1.24	2	5.44	2	5.42
3	1.36	9	1.07	3	4.96	3	4.94
8	1.31	5	0.89	4	4.69	4	4.70
0	1.29	6	0.81	6	4.54	6	4.54
9	1.24	0	0.60	9	4.06	9	4.13
6	1.05	1	-1.21	8	3.39	8	3.36

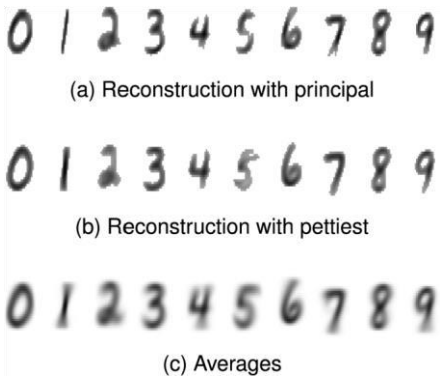


Fig. 9. Reconstructed digits with fastPRIM.

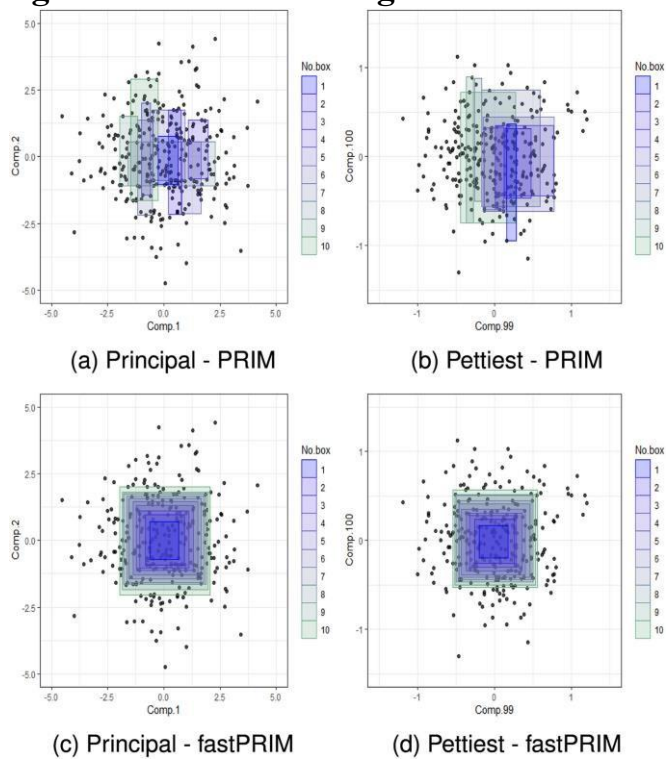


Fig. 10. Simulation results by method.

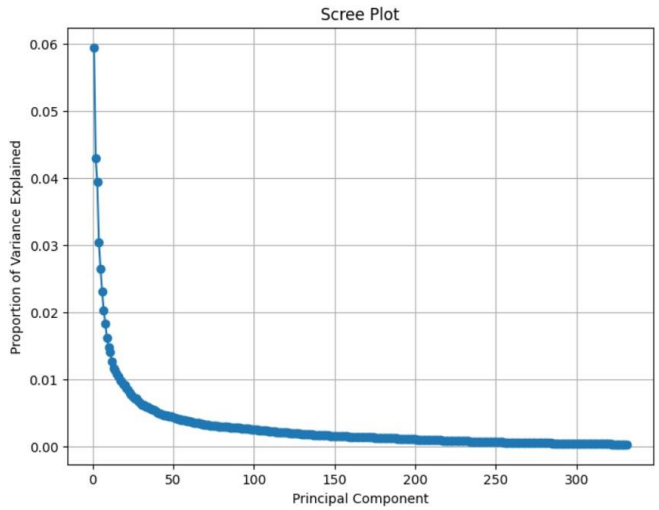


Fig.5: ploting principal components

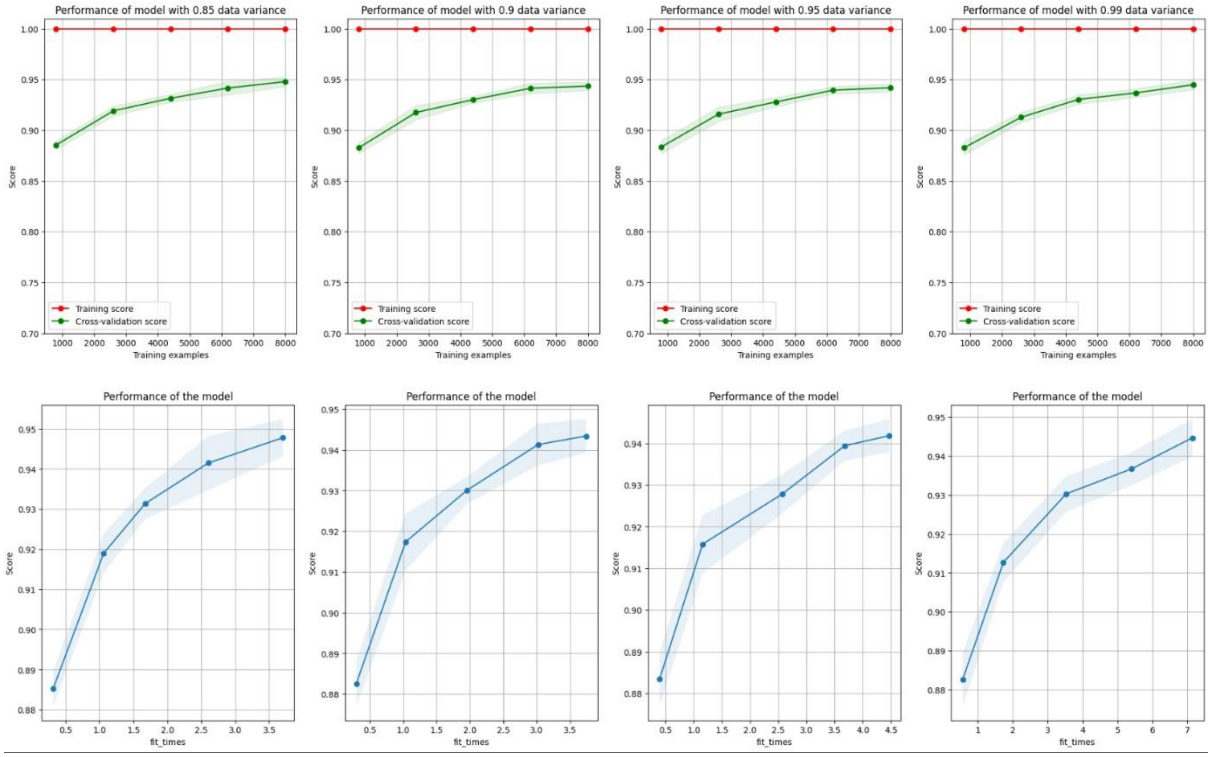


Fig6:comparision between pca and pettiest components

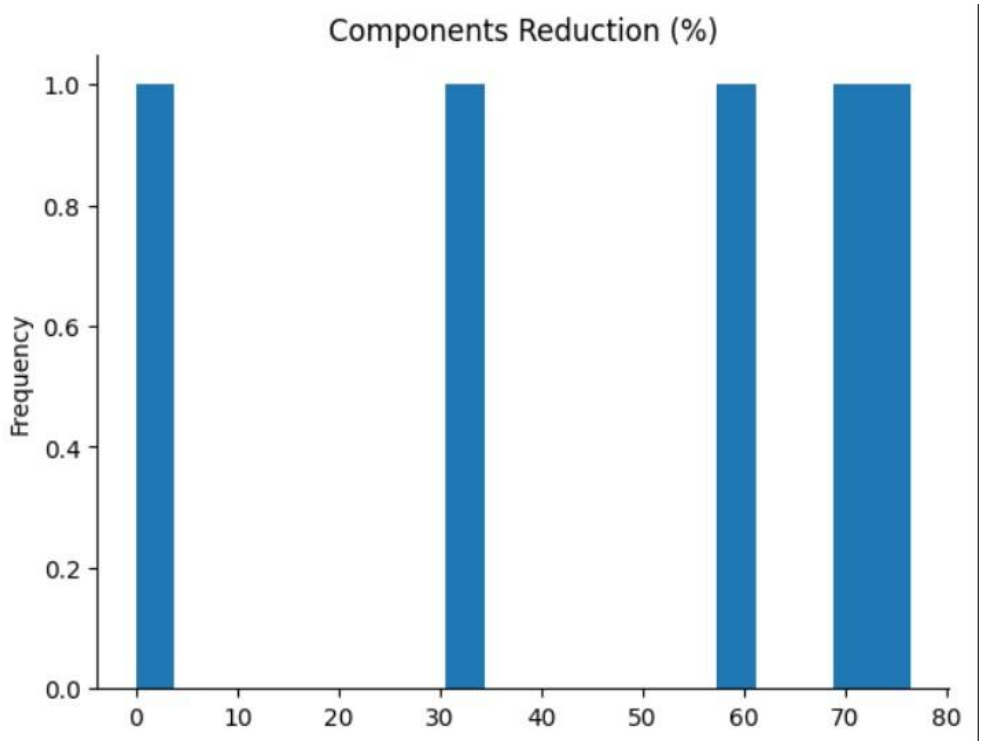


Fig 7:dimension reduction

Chapter-5

Conclusion

The effectiveness of pettiest components in mode detection and dimensionality reduction, contrasting them with traditional principal components analysis (PCA). Through experimental analysis using both synthetic and real-world datasets, several key findings emerged.Firstly, our experiments demonstrated that pettiest components, obtained through rotation and reduction of the dataset, outperformed principal components in mode detection tasks. The application of PRIM and fastPRIM algorithms to both principal and pettiest components revealed significant differences in box volumes and nesting patterns. Notably, fastPRIM exhibited superior performance with pettiest components, achieving substantially smaller final boxes compared to PRIM.the application of pettiest components to the MNIST dataset for handwritten digit recognition showcased their effectiveness in detecting modal patterns. The analysis of active information values revealed that pettiest components yielded higher information gain compared to principal components, particularly when applied with fastPRIM. highlights the importance of considering pettiest components as a viable alternative to principal components in mode detection tasks. The superior performance of pettiest components, particularly when coupled with efficient algorithms like fastPRIM, suggests their potential for enhancing pattern recognition and dimensionality reduction tasks in various domains.the findings of this study underscore the significance of exploring alternative approaches to traditional PCA, such as pettiest component analysis, in order to achieve better performance in mode detection and dimensionality reduction tasks. Future research could further investigate the applications of pettiest components in other domains and explore additional optimization techniques to enhance their effectiveness.

REFERENCES

1. K. V. Mardia, J. T. Kent, and J. M. Bibby, Multivariate Analysis. Cambridge, MA, USA: AcademicPress, 1979.
2. D. R. Cox, “Notes on some aspects of regression analysis,” J. Roy. Statist. Soc. A, vol. 131, pp. 265–279, 1968. [Online]. Available: https://doi.org/10.2307/2343523
3. D. R. Cox, “Notes on some aspects of regression analysis,” J. Roy. Statist. Soc. A, vol. 131, pp. 265–279, 1968. [Online]. Available: https://doi.org/10.2307/2343523
4. V. Vapnik, Statistical Learning Theory. Hoboken, NJ, USA: Wiley, 1998. 5. T. Tao, Topics in Random Matrix Theory. Providence, RI, USA: Amer. Math. Soc., 2012.
6. W.A.DembskiandR.J.Marks,II,“Bernoulli’s principle of insuffi cient reason and conservation of information in computer search,” in Proc. IEEEInt.Conf.Syst., Man,Cybern., 2009,pp.2647–2652.
7. Y. LeCun, C. Cortes, and C. J. C. Burges, “The MNIST database of handwritten digits,” 2010. [Online]. Available: http://yann.lecun. com/exdb/mnist
8. J. H. Friedman and N. I. Fisher, “Bump hunting in high-dimen sional data,” Stat. Comput., vol. 9, pp. 123–143, 1999. [Online]. Available: https://doi.org/10.1023/A:1008894516817
9. D. A. Dıaz-Pachon, J. P. Saenz, and J. S. Rao, “Hypothesis testing with active information,” Statist. Probability Lett., vol. 161, 2020, Art. no. 108742.
10. .-E. Dazard and J. S. Rao, “Local sparse bump hunting,” J Comput. Graph. Stat., vol. 19, no. 4, pp. 900–929, 2010. [Online]. Available: https://doi.org/10.1198/jcgs.2010.09029
11. E. Parzen, “On estimation of a probability density function and mode,” Ann. Math. Stat., vol. 33, no. 3, pp. 1065–1076, 1962
12. D. R. Cox, “Notes on some aspects of regression analysis,” J. Roy. Statist. Soc. A, vol. 131, pp. 265–279, 1968. [Online]. Available: https://doi.org/10.2307/2343523
13. J. VanderPlas, Python Data Science Handbook: Essential Tools for

Working With Data. Sebastopol, CA, USA: O'Reilly Media, 2016.

14. Y. LeCun, C. Cortes, and C. J. C. Burges, "The MNIST database of handwritten digits," 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist/
15. D. A. Dıaz-Pachon, J. P. Saenz, J. S. Rao, and J.-E. Dazard, "Mode hunting through active information," Appl. Stochastic Models Bus. Ind., vol. 35, no. 2, pp. 376–393, 2019
16. D. A. Dıaz-Pachon and R. J. Marks, II, "Generalized active infor mation: Extensions to unbounded domains," BIO-Complexity, vol. 2020, no. 3, pp. 1–6, 2020.
17. W. Polonik and Z. Wang, "PRIM analysis," J. Multivar. Anal., vol. 101, no. 3, pp. 525–540, 2010. [Online]. Available: https://doi.org/10.1016/j.jmva.2009.08.010
18. S. Kotz, T. J. Kozubowski, and K. Podgorski, The Laplace Distribu tion and Generalizations. Basel, Switzerland: Birkhauser, 2001.
19. J.-E. Dazard and J. S. Rao, "Local sparse bump hunting," J Comput. Graph. Stat., vol. 19, no. 4, pp. 900–929, 2010. [Online]. Available: https://doi.org/10.1198/jcgs.2010.09029
20. Y. LeCun, C. Cortes, and C. J. C. Burges, "The MNIST database of handwritten digits," 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist/
21. W. A. Dembski and R. J. Marks, II, "Conservation of information in search: Measuring the cost of success," IEEE Trans. Syst., Man, Cybern.- A: Syst. Hum., vol. 5, no. 5, pp. 1051–1061, Sep. 2009.
22. T. Tao, Topics in Random Matrix Theory. Providence, RI, USA: Amer. Math. Soc., 2012.
23. J. E. Chacon, "The modal age of statistics," Int. Stat. Rev., vol. 88, pp. 122–141, 2020. [Online]. Available: https://doi.org/10.1111/ insr.12340
24. V. Vapnik, Statistical Learning Theory. Hoboken, NJ, USA: Wiley, 1998.
25. G.James, D. Witten, T. Hastie, and R. Tibshirani, An Introduction to Statist. Learn.. Berlin, Germany: Springer, 2013.

26. P. S. Ruzankin and A. V. Logashov, “A fast mode estimator in multidimensional space,” Statist. Probab. Lett., vol. 158, 2020, Art. no. 108670. [Online]. Available: https://doi.org/10.1016/j. spl.2019.108670
27. D. A. Dıaz-Pachon, J. P. Saenz, and J. S. Rao, “Hypothesis testing with active information,” Statist. Probability Lett., vol. 161, 2020, Art. no. 108742.
28. D. A. Dıaz-Pachon, J. P. Saenz, J. S. Rao, and J.-E. Dazard, “Mode hunting through active information,” Appl. Stochastic Models Bus. Ind., vol. 35, no. 2, pp. 376–393, 2019.
29. D. A. Dıaz-Pachon and R. J. Marks, II, “Generalized active infor mation: Extensions to unbounded domains,” BIO-Complexity, vol. 2020, no. 3, pp. 1–6, 2020.
30. J. H. Friedman and N. I. Fisher, “Bump hunting in high-dimen sional data,” Stat. Comput., vol. 9, pp. 123–143, 1999. [Online]. Available: https://doi.org/10.1023/A:1008894516817
31. W. Polonik and Z. Wang, “PRIM analysis,” J. Multivar. Anal., vol. 101, no. 3, pp. 525–540, 2010. [Online]. Available: https://doi. org/10.1016/j.jmva.2009.08.010
32. J.-E. Dazard and J. S. Rao, “Local sparse bump hunting,” J Comput. Graph. Stat., vol. 19, no. 4, pp. 900–929, 2010. [Online]. Available: https://doi.org/10.1198/jcgs.2010.09029