Sparse Principal Component Analysis with Model Order Reduction

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Prashanth Bharadwaj Sivaraman

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

Mihailo R. Jovanović

June, 2016

Acknowledgements

There are many people that have earned my gratitude for their contribution to my time in graduate school. I would like to thank my family, adviser, professors, colleagues and friends who made this thesis possible.

I thank my adviser Prof. Mihailo Jovanović for providing a conducive environment and an air of positivity which helped me work efficiently. I have benefited immensely from the classes taught by him. Professor Mihailo Jovanovićs classes on Linear Systems and Optimal Control and Nonlinear Systems have laid the foundation of the skills I have learned in my graduate studies. Special thanks to Neil Dhingra for his patience and support during the course of my thesis. I thank Prof. Peter Seiler, Jarvis Haupt for being on the Committee. My deepest gratitude goes to my father Sivaraman Jagadeesan who has always been the leading light and inspiration to succeed in life and imbibe in me the importance of education. I dedicate this thesis to my family for their continued support through the years without whom I would not be able to be where I am today.

Abstract

Principal Component Analysis (PCA) has become a standard tool for identification of the maximal variance in data. The directions of maximum variance provide very insightful information about the data in a lot of applications. By augmenting the PCA problem with a penalty term that promotes sparsity, we are able to obtain sparse vectors describing the direction of maximum variance in the data. A sparse vector becomes very useful in many applications like finance, where it has a direct impact on cost. An algorithm which computes principal component vector in in a reduced space by using model order reduction techniques and enforces sparsity in the full space is described in this work. We achieve computational savings by enforcing sparsity in different coordinates than those in which the principal components are computed. This is illustrated by applying the algorithm to synthetic data. The algorithm is also applied to the linearized Navier-Stokes equations for a plane channel flow.

Contents

List of Figures

Chapter 1

Introduction

Principal Component Analysis (PCA) is a powerful tool with widespread applications in data analysis, data compression, and data visualization. It identifies principal components - vectors that describe the direction of maximal variance in available data. For different applications the structural features highlighted by these principal components may have different physical interpretations.

Sparse Principal Component Analysis (SPCA) is a variation of the PCA problem enforcing a l_1 regularization to promote zero terms in the principal component vector. The number of non zero terms in the principal component vector has a direct influence in many practical applications. For example, in financial applications, the non zero terms directly influence the cost. Hence, a sparsity promoting version of PCA is a very interesting problem with a wide range of applications. One major factor of Sparse PCA problem is the non smoothness of the objective function. The sub-gradient methods available have a complexity of $\mathcal{O}(1/\epsilon^2)$ to compute a ϵ-approximate solution. D'aspremont *et al.* in [2] developed a semi definite programming formulation of the sparse PCA problem based on the method described in [3]. In this work, a variation of the method developed in [2] using model order reduction to is presented. The method explores how a model order reduction of the covariance matrix of the data can be used to compute a sparse principal component vector. The thesis also explains how this method can be applied to study input-output properties of Linearized Navier Stokes equation.

Chapter 2

Background

The standard principal component analysis (PCA) problem,

$$\begin{array}{ll} \text{maximize} & x^*Qx \\ \text{subject to} & x^*x = 1 \end{array} \tag{2.1}$$

where $x \in \mathbb{C}^n$ and Q is a Hermitian positive semidefinite matrix. It is well known that the solution to 2.1 is given by the principal eigenvector of Q, i.e., the eigenvector corresponding to the largest eigenvalue. We consider a variant of PCA in which we want x to be a sparse vector. The challenge in achieving this objective comes from identifying the sparsity structure of x; finding the optimal vector for a given sparsity structure is straightforward. One approach is to augment the PCA problem with a sparsity-promoting penalty function $g(x)$,

$$\begin{array}{ll} \text{maximize} & x^*Qx - \gamma g(x) \\ \text{subject to} & x^*x = 1 \end{array} \tag{2.2}$$

where γ specifies the emphasis on sparsity. It is useful to restate 2.2 as a more tractable optimization problem. By introducing a new optimization variable, $X = xx^*$, the problem 2.1 can be reformulated as

$$\begin{array}{ll} \text{maximize} & \mathbf{Tr}(QX) - \gamma g(X) \\ \text{subject to} & \mathbf{Tr}(X) = 1 \\ & rank(X) = 1 \\ & X \geq 0 \end{array} \tag{2.3}$$

For convex $g(X)$, the only source of non-convexity in 2.3 is the rank constraint. Thus, dropping the rank constraint yields a convex relaxation of 2.3 which can be used to obtain a lower bound on the original optimization problem. Furthermore, when $g(X)$ is the l_1 or weighted-l_1 norm, dropping the rank constraint yields a Semidenite Program (SDP).

$$\begin{array}{ll} \text{maximize} & \mathbf{Tr}(QX) - \gamma g(X) \\ \text{subject to} & \mathbf{Tr}(X) = 1 \\ & X \geq 0 \end{array} \tag{2.4}$$

Although this formulation makes the problem more tractable, it increases the size of the optimization variable from n to n^2, which can greatly limit the efficiency of SDP solvers for problems of large dimension. A more computationally efficient algorithm to address the size of the problem based on Nesterov's method [2][3] and model reduction is discussed in subsequent chapters in this thesis.

Chapter 3

Model Order Reduction

3.1 Motivation

A major factor which affects the efficiency of the semi definite program described in 2.3 is the size of the problem. In this thesis, this problem is tackled using Model order reduction technique to form an approximation of the covariance matrix Q. Model order reduction aid in lowering complexity of algorithms by reducing the dimension or degrees of freedom of the problem. This approximation can be used in the algorithm resulting in a lower complexity with a slight loss in accuracy.

3.2 Model Order Reduction using a tall orthonormal matrix

In place of using the full sized covariance matrix Q to perform the semi definite program discussed in 2.3, Q can be transformed to a matrix Q_r of arbitrary lower dimension using an orthonormal matrix P.

$$Q_r = P^T Q P \tag{3.1}$$

where, $Q_r \in \mathbb{R}^{r \times r}$, $Q \in \mathbb{R}^{n \times n}$, $P \in \mathbb{R}^{n \times r}$, $r \leq n$. The matrix Q_r will have r number of

eigenvalues of the n eigenvalues from matrix Q based on the orthonormal matrix P. The semi definite program described in 2.3 can be reformulated as,

$$\begin{aligned} \text{maximise} \quad & \mathbf{Tr} Q_r X_r - \gamma ||X||_1 \\ \text{subject to} \quad & P X_r P^T - X = 0 \\ & \mathbf{Tr} X_r = 1 \\ & X_r \geq 0 \end{aligned} \tag{3.2}$$

Although Q_r is used to compute a sparse principal component vector in the lower dimension, the penalty for sparsity is imposed in the original full space of the required principal component.

Chapter 4

Algorithm

4.1 Optimal First order minimization of non-smooth function using a smoothing technique

The algorithm presented in this thesis is based on the algorithm explained in [2]. The algorithm in [2] has been modified using model order reduction as explained in chapter 3.

4.1.1 Algorithm

The semidefinite program 2.3 can be solved efficiently for small problems. For large scale problems there are numerical problems which have to be addressed. The numerical difficulties arising in large scale problems stem form two distinct origins: The issue of *memory* and *smoothness*. The issue of smoothness arises from the non-smooth constraint $X \geq 0$. This complexity can be handled usnig additional structural information of the problem and there is an identified structural information of the problem which can be used to handle this complexity.
An efficient first-order scheme for convex minimization has been proposed in [3] based on a smoothing argument. The structural assumption on the function to minimize is the that it has a *saddle-function* format:

$$f(x) = \hat{f}(x) + \max_{u}\{< Tx, u > -\hat{\phi}(x) : u \in Q_2\} \tag{4.1}$$

where f is defined over a compact convex set $Q_1 \in \mathbb{R}^n$, $\hat{f}(x)$ is convex and differentiable and has a Lipschitz continuous gradient with constant $M > 0$, T is an element of $\mathbb{R}^{n\times n}$ and $\hat{\phi}(u)$ is a continuous convex function over some closed compact set $Q_2 \in \mathbb{R}^n$. This assumes that the function $\hat{\phi}(u)$ and the set Q_2 are simple enough so that the optimization subproblem can be solved efficiently. When the function f can be expressed in this particular format, [3] presents a method using a *smoothing technique* which reduces the complexity of solving. This method entails two steps,

Regularization. By adding a strongly convex penalty to the saddle function representation of f in 4.1, the algorithm first computes a *smooth* ϵ-approximation of f with Lipschitz continuous gradient.

Optimal first-order minimization. As a second step, the algorithm applies the optimal first order scheme for functions with Lipschitz continuous gradient detailed in [4] to the regularized function.

4.2 Model order reduction based algorithm for smooth minimization of non smooth functions

The algorithm has been described when the size of the orthogonal matrix P is fixed, $P \in \mathbb{C}^{n\times r}$. This algorithm is repeated for different sizes of P. The size of matrix P is increased until the objective value of the maximized function computed for subsequent sizes of matrix P are approximately equal. Hence, by using this algorithm we will be able to identify an optimal size of the reduced order model of matrix Q which can be used to compute an equally well performing sparce principal component vector. This algorithm will ensure lower complexity of the algorithm because the order of complexity of the algorithm depends on the size of the matrix Q [3].

Different methods can be used to grow the orthogonal matrix P. In this thesis, a proximal gradient based method, i.e. soft thresholding of the sparse principal component vector computed is used to grow the orthogonal matrix P. The method is explained in finer detail in 4.2.1.

As previously explained in chapter 3, a major factor affecting the efficiency of the

semi definite program 2.3 is the size of the problem. In this work, we propose a method based on model order reduction and Nesterov's method [2],[3] to overcome this problem. Chapter 3 provides an explanation of how model order reduction is used for the sparse PCA problem.

A convex relaxation of 2.3 by dropping the rank constraint yields a Semidefinite Program.

$$\begin{aligned} \text{maximize} \quad & \mathbf{Tr}(QX) - \gamma 1^T |X| 1 \\ \text{subject to} \quad & \mathbf{Tr}(X) = 1 \\ & W \geq 0 \end{aligned} \tag{4.2}$$

A computationally more efficient method to solve this problem is to use model order reduction and transform the co-variance matrix Q to a matrix of arbitrarily lower dimension Q_r.

$$Q_r = P^T Q P \tag{4.3}$$

where, $Q_r \in \mathbb{C}^{r \times r}$, $Q \in \mathbb{C}^{n \times n}$, $P \in \mathbb{C}^{n \times r}$, $r \leq n$. The SDP 4.2 can be written as:

$$\begin{aligned} \text{maximize} \quad & \mathbf{Tr}(Q_r X_r) - \gamma 1^T |X| 1 \\ \text{subject to} \quad & P X_r P^T - X = 0 \\ & \mathbf{Tr}(X_r) = 1 \\ & X_r \geq 0 \end{aligned} \tag{4.4}$$

We have added an extra constraint $PX_rP^T - X = 0$ to ensure the equality of the variables X_r and X in the reduced and full dimensions when transformed.
The objective function of the problem can be expressed in a saddle function format. Expressing the objective function in this format helps in the smooth approximation of the non smooth function. A optimal first order minimization method explained 4.1 can be used to find the optimal solution problem. The dual of the problem helps in representing the objective function in a saddle function format.

$$\min_{U \in Q_1} f(U) \tag{4.5}$$

where
$Q_1 = \{U \in \mathbb{S}^n : |Uij| \leq 1, i, j = 1, ..., n\}$, $Q_2 = \{X_r \in \mathbb{S}^r : \mathbf{Tr}(X_r) = 1, X_r \geq 0\}$
$f(U) = max_{X \in Q_2} < TU, X_r > -\hat{\phi}(X_r)$, with $T = I_{n^2}$, $\hat{\phi}(X_r) = -\mathbf{Tr}(Q_r X_r)$.
As explained in [3], we associate norms and so-called prox-functions to Q_1 and Q_2. We associate the Frobenius norm in $\mathbb{S}^n$ to Q_1, and a prox-function defined for $U \in Q_1$ by :

$$d_1(U) = \frac{1}{2} U^T U$$

With this choice, the center U_0 of Q_1, defined as:

$$U_0 = \arg \min_{U \in Q_1} d_1(U)$$

is U_0, and satisfies $d_1(U_0) = 0$. Moreover, we have:

$$D_1 = \max_{U \in Q_1} d_1(U) = \frac{n^2}{2}$$

Furthermore, the function d_1 is strongly convex on its domain, with convexity parameter of $\sigma_1 = 1$ with respect to the Frobenius norm. Next, for Q_2 we use the dual of the standard matrix norm (denoted $||.||_2^*$), and a prox-function

$$d_2(X_r) = \mathbf{Tr}(X_r \log X_r) + \log r,$$

where $\log X_r$ refers to the *matrix* (and not componentwise) logarithm, obtained by replacing the eigenvalues of X_r by their logarithm. The center of the set Q_2 is $X_0 = r^{-1} I_r$, where $d_2(X_0) = 0$. We have

$$\max_{X \in Q_2} d_2(X) \leq \log r = D_2$$

The convexity parameter of d_2 with respect to $||.||_2^*$, is bounded below by $\sigma_2 = 1$. This non trivial result is proved in [5].
The (1,2) norm of the operator T introduced above is computed as follows:

$$||T||_{1,2} = \max_{X_r,U} < TX_r, U >: ||U||_F = 1, ||X_r||_2^* = 1$$
$$||T||_{1,2} = \max_{X_r} ||X_r||_2 : ||X_r||_F \leq 1$$
$$||T||_{1,2} = 1$$

To summarize, the parameters defined above are set as follows: $D_1 = \frac{n^2}{2}, \sigma_1 = 1, D_2 = \log(r), \sigma_2 = 1, ||T||1, 2 = 1$.
The following sections explain how the regularization and smooth minimization techniques can be applied to the Sparse PCA problem 4.4.

Regularization

The method in [3] defines a regularization parameter

$$\mu = \frac{\epsilon}{2D_2}$$

This method produces an ϵ-suboptimal optimal value and a corresponding suboptimal solution in $\frac{4||T||_{1,2}}{\epsilon}\sqrt{\frac{D_1 D_2}{\sigma_1 \sigma_2}}$ number of steps.
The non-smooth objective $f(X_r)$ of the original problem is replaced with

$$\min_{U \in Q_1} f_\mu(U),$$

where f_μ is the penalized function involving the prox-function d_2:

$$f_\mu(U) = \max_{X_r \in Q_2} < TU, X_r > -\hat{\phi}(X_r) - \mu d_2(X_r).$$

The function f_μ is a smooth approximation to f everywhere on Q_2, with maximal error $\mu D_2 = \frac{\epsilon}{2}$. The function f_μ has a Lipschitz continuous gradient, with Lipschitz constant, $L = \frac{D_2||T||_{1,2}^2}{\epsilon 2\sigma_2}$ and is a uniform approximation of the function f. The function f_μ is computed explicitly as:

$$f_\mu(U) = \mu \log(\mathbf{Tr}\exp((Q_r + P^T U P)/\mu)) - \mu \log r,$$

which is a smooth approximation to the function $f(U) = \lambda_{max}(Q_r + P^T U P)$.

First-order minimization

An optimal gradient algorithm for minimizing convex functions with Lipschitz continuous gradients as explained in [4] is applied to the function f_μ.

For the Sparse PCA problem, when is the regularization parameter μ is fixed, the algorithm is as follows.

Repeat

- Compute $f_\mu(U_k)$ and $\nabla f_\mu(U_k)$
- Find $Y_k = \arg\min_{Y \in Q_1} < \nabla f_\mu(U_k), Y > + \frac{1}{2} L ||U_k - Y||_F^2$
- Find $W_k = \arg\min_{W \in Q_1} \{ \frac{L d_1(W)}{\sigma_1 +} \sum_{i=0}^{k} \frac{i+1}{2} (f_\mu(U_i) + < \nabla f_\mu(U_i), W - U_i >\}$
- $U_{k+1} = \frac{2}{k+3} W_k + \frac{k+1}{k+3} Y_k$

Until gap $\leq \epsilon$

Step one above computes the function value and gradient. The second step computes the *gradient mapping*, which matches the gradient step for unconstrained problems [5]. Step three and four update an estimate sequence [5] of f_μ whose minimum can be explicitly computed and givs and increasingly tighter upper bound on the minimum of f_μ.

Step 1. The first step is the computation f_μ and its gradient. This is the most expensive step in the algorithm. By setting $Z = Q_r + P^T U P$, the problem boils down to computing

$$u^*(z) = \arg\max_{X_r \in Q_2} < Z, X_r > -\mu d_2(X_r) \tag{4.6}$$

and the associated optimal value of the function $f_\mu(U)$. This problem has a very simple solution requiring only an eigenvalue decomposition for $Z = Q_r + P^T U P$. The gradient of the objective function with respect to Z is is set to the maximizer of $u^*(Z)$, so the gradient with respect to U is $\nabla f_\mu(U) = U^*(Q_r + P^T U P)$. We form an eigenvalue decomposition $Z = V D V^T$, with $D = diag(d)$ the matrix with diagonal d, to compute $U^*(Z)$. Set

$$h_i = \frac{exp(\frac{d_i - d_{max}}{\mu})}{\sum_{j=1}^{n} exp(\frac{d_j - d_{max}}{\mu})}, \text{ i = 1,...,r,}$$

where $d_{max} = max_{j=1,...,n} d_j$ is used to mitigate large numbers. Let $u^*(z) = VHV^T$, with $H = Pdiag(h)P^T$. The corresponding function values is given by,

$$f_\mu(U) = \mu \log(\mathbf{Tr} \exp((Q_r + P^T U P)/\mu)) - \mu \log(r) \tag{4.7}$$

which is computed as:

$$f_\mu(U) = d_{max} + \mu \log(\sum_{i=1}^{r} \exp(\frac{d_i - d_{max}}{\mu})) - \mu \log(r). \tag{4.8}$$

*Step 2.*This step involves a problem of the form:

$$\arg \min_{Y \in Q_1} < \nabla f_\mu(U), Y > + \tfrac{1}{2} L ||U - Y||_F^2,$$

where U is given. The above problem is reduced to an Euclidean projection problem:

$$\arg \min_{||Y||_\infty} ||Y - V||_F, \tag{4.9}$$

where $V = U - L^{-1} \nabla f_\mu(U)$. The solution for this Euclidean projection problem is given by:

$$Y_{ij} = \mathbf{sgn}(V_{ij}) \min(|V_{ij}|, 1),\ i, j = 1, ..., n.$$

Step 3. The third step involves an Euclidean projection as in step 2 4.9, where V is defined by:

$$V = -\tfrac{\sigma_1}{L} \textstyle\sum_{i=0}^{k} \frac{i+1}{2} \nabla f_\mu(U_i)$$

Stopping Criteria. The algorithm can be stopped when the duality gap is smaller than ϵ.

$$gap_k = \lambda_{max}(Q_r + P^T U_k P) - \mathbf{Tr} Q_r X r_k + 1^T |X_k| 1 \leq \epsilon$$

The matrix Q is scaled by a $\frac{1}{\gamma}$ to induce for sparsity in the primal and dual variables. The duality gap is necessarily non-negative since both the primal variable (Xr_k) and the dual variable (U_k) are feasible in the corresponding problems.

4.2.1 Proximal Gradient based method to augment matrix P

The orthogonal matrix P is augmented by a soft thresholding step performed on the sparse principal component vector obtained for Q_r. The vector obtained based from this soft thresholding step is augmented to P, i.e. $P = [P\ P_s]$, where P_s is the vector computed using soft thresholding. The augmented matrix P is then orthogonalized using Gram-Schmidt process.

Soft Thresholding

Proximal gradient methods are a generalized form of projection used to solve non-differentiable convex optimization problems. The proximal mapping or proximal operator of a convex function f is

$$\mathbf{prox}_f(x) = \arg\min_u (f(u) + \frac{1}{2}||u - x||_2^2) \tag{4.10}$$

The proximal gradient method for a function $f(x)$ split in two components

$$f(x) = g(x) + h(x) \tag{4.11}$$

where g is convex, differentiable and h is closed, convex, possibly non differentiable, the proximal gradient algorithm is :

$$x^k = \mathbf{prox}_{\alpha h}(x^{k-1} - \alpha \nabla g(x^{k-1})) \tag{4.12}$$

α is step size which can be constant or determined by line search.

In our algorithm to compute the sparse principal component vector, the proximal operator is used to find a vector P_s for the function

$$f(x) = x^T Q_r x + \gamma ||x||_1 \tag{4.13}$$

for the problem considered, $g(x) = x^T Q_r x$ and $\gamma||x||_1$. The proximal gradient algorithm for the problem becomes,

$$x^k = \mathbf{prox}_{\alpha h}(x^{k-1} - \alpha Q_r x^{k-1}) \tag{4.14}$$

We fix a threshold $t = \alpha \times \gamma$. The $\mathbf{prox}_h$ function is the soft threshold operation defined as follows :

$$S_t(x^{k-1}) = \begin{cases} 0, |x^{k-1}| \leq t \\ x^{k-1} - t, x^{k-1} \geq t \\ x^{k-1} + t, x^{k-1} \leq -t \end{cases}$$

The vector $S_t(x^{k-1})$ is augmented to the matrix P, i.e $P = [P\ S_t(x^{k-1})]$. The augmented matrix P is then orthogonalized using Gram-Schmidt process.

Chapter 5

Numerical Results and Applications

In this chapter, we present the performance results of the proposed method on Artificial and real-life data sets. A practical application of the algorithm is presented in this chapter.

5.1 Application of Algorithm for Linearized Navier Stokes Equation

In many applications, it is of great interest to study the linearized system around an equilibrium point a steady-state velocity profile. For channel flows, the dynamics of these linearized equations can be simplified from the three velocity components, $u(t, x, y, z), v(t, x, y, z), \text{and} w(t, x, y, z)$ into the wall normal velocity, $v(t, x, y, z)$, and the wall-normal vorticity, $\eta(t, x, y, z) := \partial z_u(t, x, y, z)\partial_x w(t, x, y, z)$.
We consider a channel ow with an infinite streamwise and spanwise domain, i.e., $x, z \in (\infty, \infty), y \in [1, 1]$, with no-slip boundary conditions on the flow velocity elds are $u(t, x, \pm 1, z) = v(t, x, \pm 1, z) = w(t, x, \pm 1, z) = 0$. After linearizing the Navier-Stokes equations and reducing the dynamics, we obtain

$$\begin{bmatrix} \dot{v} \\ \dot{\eta} \end{bmatrix} = \begin{bmatrix} \mathcal{A}_{\mathcal{OS}} & 0 \\ \mathcal{A}_{\mathcal{C}} & \mathcal{A}_{\mathcal{S}} \end{bmatrix} \begin{bmatrix} v \\ \eta \end{bmatrix}$$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathcal{C} \begin{bmatrix} v \\ \eta \end{bmatrix}$$

where $\mathcal{A}_{\mathcal{OS}}$ is the Orr-Sommerfeld operator, $\mathcal{A}_{\mathcal{S}}$ is the Squire operator, $\mathcal{A}_{\mathcal{C}}$ couples v and η. The reader is referred to [8] for details and discussions on the specific form of the equations, their derivation, and analysis of their dynamics. The algorithm was implemented on the linearized NS equations for streamwise constant disturbances. In the algorithm presented, the principal component vector is computed for a reduce order model of matrix Q while sparsity is enforced for the full size matrix Q. This setup is synonymous with the problem formulation to study the input-output relationship of the Linearized Navier Stokes equation because we would like to compute a principal component vector in the space spanned by $[v\ \eta]'$ while enforcing sparsity in the space spanned by $[u\ v\ w]'$.

5.2 Artificial data

To check the efficiency of the algorithm, we consider the simulation example proposed in [7]. In this example, three hidden factors are created:
$V_1 \sim \mathcal{N}(0, 290)$, $V_2 \sim \mathcal{N}(0, 300)$, $V_3 = -0.3V_1 + 0.925V_2$, $\epsilon \sim \mathcal{N}(0, 300)$
with V_1, V_2 and ϵ independent. Afterward, 10 observed variables are generated as follows :

$$X_i = V_j + \epsilon_i^j,\ \epsilon_i^j \sim \mathcal{N}(0, 1)$$

with $j = 1$ for $i = 1, ..., 4$, $j = 2$ for $i = 5, ..., 8$ and $j = 3$ for $i = 9, 10$ and ϵ_i^j independent for $j = 1, 2, 3$, $i = 1, ..., 10$.

The algorithm was tested on multiple matrices generated according to the method described. In all cases, the algorithm was tested for a lower order model Q_r of Q for r =3,...,10. A consistent observation from all the examples tested is, the objective value of the SDP problem, i.e., the maximum variance of Q_r tends to plateau to an approximately same value to that of matrix Q, when $r < 10$. This is illustrated in figure 5.1.

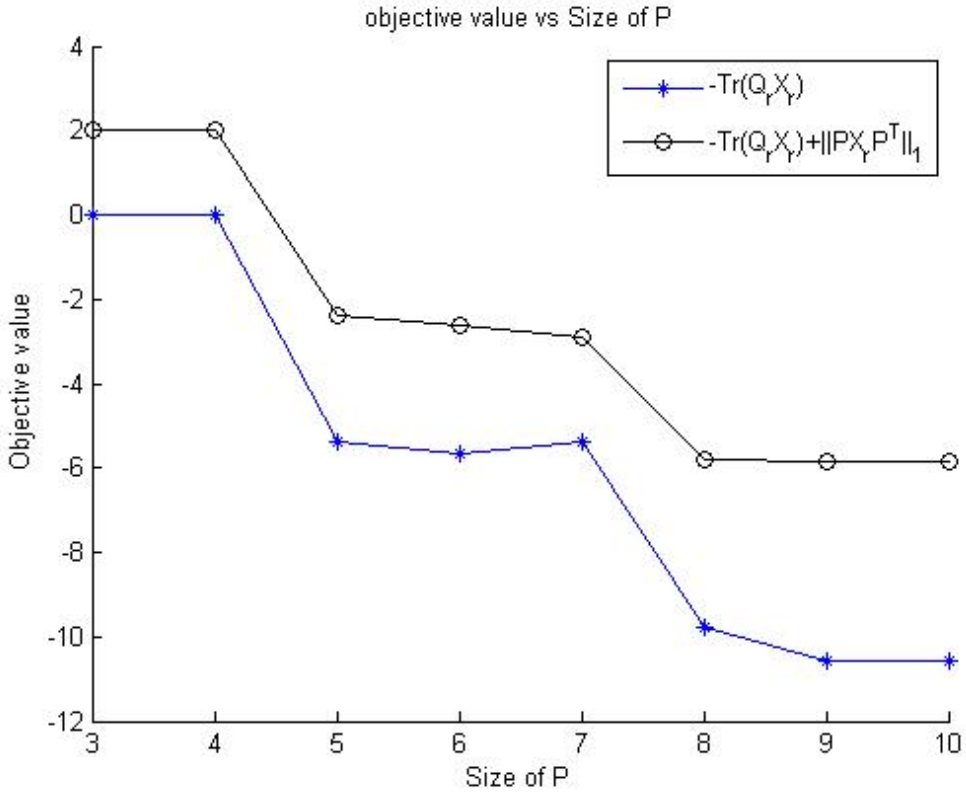

Figure 5.1: Plot of Objective value of maximized function for different sizes of P

From figure 5.1 it is observed that the objective value of the maximized function when size of Q_r is 8×8 and the objective value of the maximized function when size of Q_r is 10×10 are approximately equal. This means that a lower order model Q_r of matrix Q can be used to compute a sparse principal component vector for the matrix Q using the proposed algorithm. Thus, using a lower order model to compute the principal component vector reduces the complexity of the problem.

5.3 Pit props data

The pit props data is a benchmark example used to test sparse PCA codes. *J.Jeffers* [6] introduced the data and it consists of 180 observations and 13 measured variables.

The algorithm was tested on matrices Q_r for r = 3,...,13. As can be seen from the plots 5.2 and 5.3, the objective value, i.e., maximum variance of Q_r of the SDP problems for different sizes of Q_r are approximately equal when r = 8,...,13. This means that a lower order model of matrix Q of size 8×8 can be used to compute a sparse principal component vector for the matrix Q using the proposed algorithm. Thus, using a lower order model to compute the principal component vector reduces the complexity of the problem.

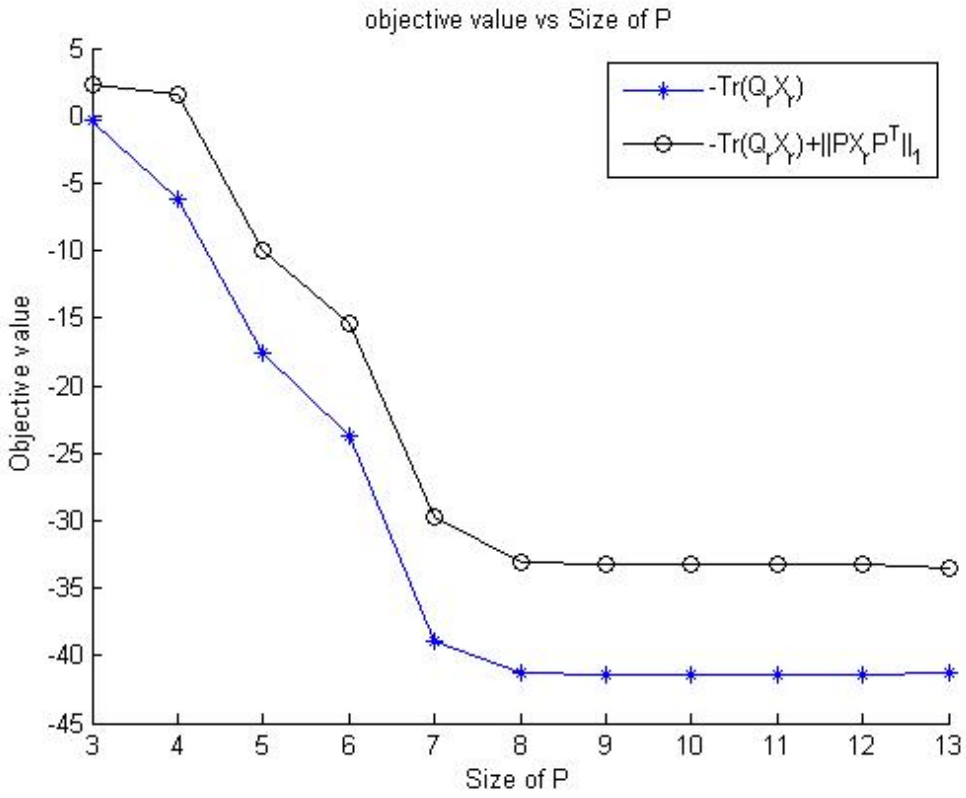

Figure 5.2: Plot of Objective value of maximized function for different sizes of P

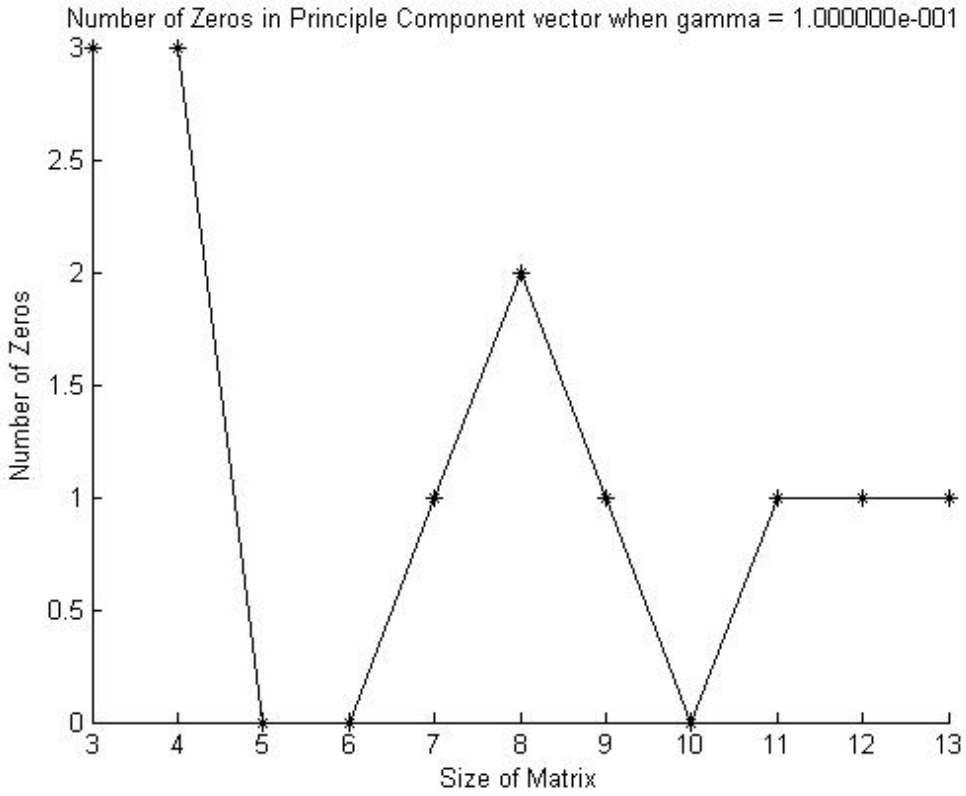

Figure 5.3: Plot of number of zero elements in Principle component vector for different sizes of P

Chapter 6

Conclusion

In this thesis work, an algorithm for Sparse principle component analysis incorporating techniques from model order reduction has been presented. The incorporation of model order reduction technique to the algorithm for sparse principal component analysis [2] increases the scope of the algorithm to wider range of problems. The algorithm described in thesis work has been applied to the Linearized Navier Stokes equation. The advantage of using this algorithm for Linearized Navier Stokes equation is that we can compute a principal component vector in the space spanned by $[v\ \eta]'$ while enforcing sparsity in the space spanned by $[u\ v\ w]'$. It is also showed how using model order reduction in for the sparse principal component analysis can improve the complexity to compute a sparse principal component vector. The improvement in complexity is achieved from the fact that the complexity of the algorithm depends on the size of the covariance matrix and it has been shown that a reduced order model of the covariance matrix can be used to compute a sparse principal component vector.

References

[1] Neil K. Dhingra, Mihailo R. Jovanovic, and Peter J. Schmid, *Identification of spatially-localized flow structures via sparse proper orthogonal decomposition*, Bulletin of the American Physical Society, November 2013.

[2] Alexandre D'Aspremont, Laurent EL Ghaoui, Michael I, Jordan and Gert R.G. Lanckriet, *A direct formulation for sparse PCA using semidefinite programming.*

[3] Yu. Nesterov, *Smooth minimization of non-smooth functions*

[4] Yu. Nesterov, *A method of solving a convex programming problem with convergence rate* $\mathcal{O}(1/k^2)$, Soviet Mathematics Doklady, 27 (1983), pp. 372-276.

[5] Yu. Nesterov, *Smoothing techniques and its application in semidefinite optimization*, CORE Discussion Paper No. 2004/73, (2004).

[6] J.N.R. Jeffers, *Two Case Studies in the Application of Principal Component Analysis.*

[7] H. Zou, T. Hastie, and R. Tibshirani, *Sparse principal component analysis*, J. Comput. Graphical Statist., 15 (2006), pp. 265286.

[8] M. R. Jovanovic and B. Bamieh, *Componentwise energy amplification in channel flows*, J. Fluid Mech., vol. 534, pp. 145183, July 2005.