"Joint Sparse Principal Component Analysis"

M.Sc. Data Science
Linear Algebra and its Applications

Ashutosh Maurya (MDS202110)
Athul Prakash (MDS202111)
Avik Das (MDS202112)
Ayush Srivastava (MDS202113)

Instructor:
Dr Kavita Sutar

Lecturer, Chennai Mathematical Institute
ksutar@cmi.ac.in
https://www.cmi.ac.in/~ksutar

24th May, 2022

Contents

1 Introduction **2**
- 1.1 PCA and SPCA . . . 2
- 1.2 Motivation for JSPCA . . . 2

2 Methodology **3**
- 2.1 Notation . . . 3
- 2.2 Objective function . . . 3
- 2.3 Optimal Solution . . . 4
  - 2.3.1 Step 1 of Optimal Solution . . . 4
  - 2.3.2 Step 2 of Optimal Solution . . . 5
- 2.4 Convergence Analysis . . . 5

3 Algorithm and Implementation **7**
- 3.1 JSPCA Algorithm . . . 7
- 3.2 Implementation . . . 7
- 3.3 Results on the Breast-Cancer Wisconsin dataset . . . 7
- 3.4 PCA vs JSPCA . . . 8
- 3.5 Computational Complexity . . . 8

1 Introduction

1.1 PCA and SPCA

Large datasets are increasingly common and are often difficult to interpret. Principal component analysis (PCA) is a technique for reducing the dimensionality of such datasets, increasing interpretability but at the same time minimizing information loss. It does so by creating new uncorrelated variables, i.e., the Principal Components (PCs), that successively capture maximal variance.
Although PCA works well to manage the size of large datasets by reducing the dimension, there can be a few problems arising. The primary problem occurs while interpreting the results as the PCs are formed by some combination of all the variables of the dataset; it is often difficult to make sense of what a PC represents. Also, PCA is sensitive to outliers since its covariance matrix is derived from ℓ_2-norm which is sensitive to outliers.
Sparse Principal Component Analysis (SPCA) is an extension of the PCA model which aims to reduce the loadings of the PCs. Generating sparsely loaded PCs helps in generating more consistent results and easier interpretations of them.

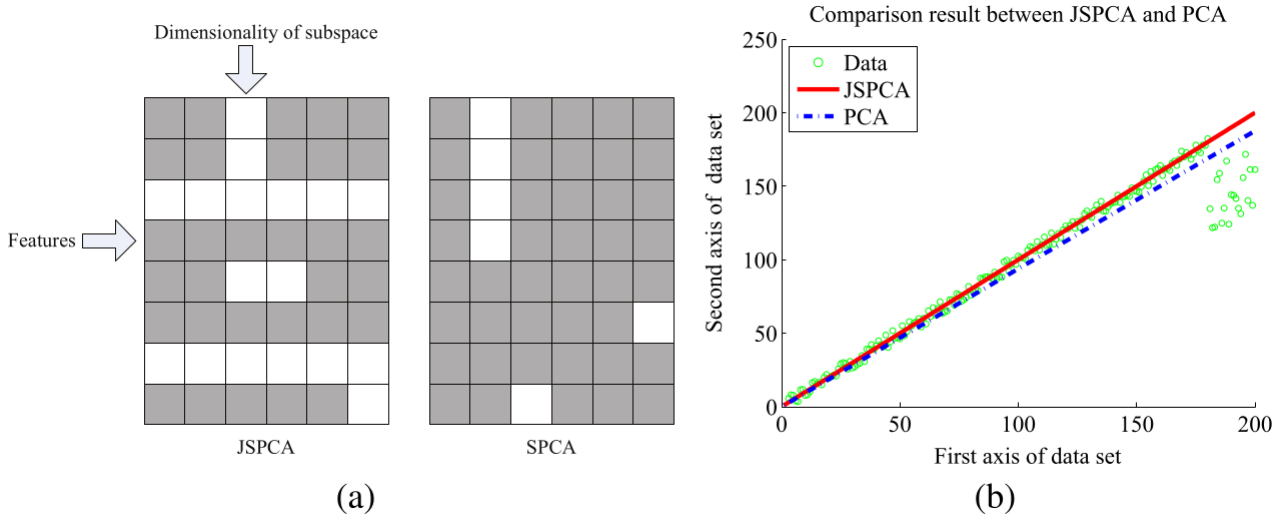

Figure 1: Motivations of JSPCA (a) JSPCA tells us useless features while SPCA cannot and (b) JSPCA is more robust to outliers

1.2 Motivation for JSPCA

To facilitate interpretation, SPCA was proposed. However, SPCA has no ability to jointly select the useful features because the ℓ_1-norm is imposed on each transformation vector and ℓ_1-norm cannot select the consistent features. Moreover, SPCA still suffers from the effect of outliers because the ℓ_2-normnorm is imposed on loss.

Joint sparse principal component analysis (JSPCA) integrates feature selection into subspace learning to exclude the redundant features. Specifically, JSPCA imposes joint $\ell_{2,1}$-norm on both loss term and regularization term. In this way, our method can discard the useless features on one hand and reduce the effect of outliers on the other hand. The main contributions of the paper are described as follows:

1. JSPCA relaxes the orthogonal constraint of transformation matrix and introduces another transformation matrix to together recover the original data from the subspace spanned by the selected

features, which makes JSPCA have more freedom to jointly select useful features for low-dimensional representation.

2. Unlike PCA and its existing extensions, JSPCA uses joint sparse constraints on the objective function, i.e., $\ell_{2,1}$-norm is imposed on the loss term and the transformation matrix, to do feature selection and learn the optimal transformation matrix simultaneously.

3. A simple yet effective optimal solution of JSPCA is provided. Furthermore, a series of theoretical analyses including convergence analysis, essence of JSPCA, and computational complexity are provided to validate the feasibility and effectiveness of JSPCA.

2 Methodology

2.1 Notation

- Given data matrix is $X = [x_1, x_2, \ldots, x_n] \in R^{m \times n}$, where m denotes the features and n denotes the number of training samples.
- The matrix $l_{2,1}$ norm: $||M||_{2,1} = \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{m} M_{ij}^2}$

2.2 Objective function

The objective function is given by

$$\arg\min_{Q,P} J(Q,P) = \arg\min_{Q,P} ||X - PQ^T X||_{2,1} + \lambda ||Q||_{2,1} \tag{1}$$

where transformation matrix $Q \in R^{m \times d}$ is first used to project the data matrix X onto a low-dimensional subspace and another transformation matrix $P \in R^{m \times d}$ is then used to recover the data matrix X. $\lambda \geq 0$ is the regularization parameter.

Directly solving the equation is tough, so the objective function is transformed in terms of Frobenius norm. This is done by first simplifying the $l_{2,1}$ norm into trace of matrices, and then introducing two diagonal matrices for further simplification of the objective function.
Hence, the objective function becomes

$$\arg\min_{Q,P} J(Q,P) = \arg\min_{Q,P} \; ||\sqrt{D_1}^T (X - PQ^T X)||_F^2 + 2\lambda ||\sqrt{D_2} Q)||_F^2 \tag{2}$$

where

$$D_1 = \begin{bmatrix} \frac{1}{\sqrt{\sum_{i=1}^{m} (X - PQ^T X)_{1i}^2}} & & \\ & \ddots & \\ & & \frac{1}{\sqrt{\sum_{i=1}^{m} (X - PQ^T X)_{mi}^2}} \end{bmatrix} = \begin{bmatrix} \frac{1}{2||(X - PQ^T X)^1||_2} & & \\ & \ddots & \\ & & \frac{1}{2||(X - PQ^T X)^m||_2} \end{bmatrix} \tag{3}$$

and

$$D_2 = \begin{bmatrix} \frac{1}{\sqrt{\sum_{i=1}^{m} Q_{1i}^2}} & & \\ & \ddots & \\ & & \frac{1}{\sqrt{\sum_{i=1}^{m} Q_{mi}^2}} \end{bmatrix} = \begin{bmatrix} \frac{1}{2||Q^1||_2} & & \\ & \ddots & \\ & & \frac{1}{2||Q^m||_2} \end{bmatrix} \tag{4}$$

It is evident that the smaller D_2^{ii} is, the more important the i-th feature is. Moreover, if $||(X - PQ^TX)^i||_2$ and $||Q^i||_2$ are small, then D_1 and D_2 are large. So, the minimization of $(X - PQ^T)^T D_1(X - PQ^TX)) + 2\lambda \text{tr}(Q^T D_2 Q)$ forces $||(X - PQ^TX)^i||_2$ and $||Q^i||_2$ to have very small values.
Thus, we obtain a joint sparse Q and a small reconstruction matrix P.

Next, let $\sqrt{D_1}P = \overline{P}$ and $\sqrt{D_1}^{-1}Q = \overline{Q}$. Then eq. (2) becomes

$$\arg\min_{Q,P} J(Q,P) = \arg\min_{Q,P} ||\sqrt{D_1}^T X - \overline{PQ}^T \sqrt{D_1}X)||_F^2 + 2\lambda||\sqrt{D_1}\sqrt{D_2}\overline{Q})||_F^2 \tag{5}$$

In order to reduce the feature redundancy, the orthogonality constraint $\overline{P}^T\overline{P} = I^{d\times d}$ is imposed.
Therefore, we have our final objective function

$$\begin{aligned}\arg\min_{Q,P} J(Q,P) = \arg\min_{Q,P} &||\sqrt{D_1}^T X - \overline{PQ}^T \sqrt{D_1}X)||_F^2 \\ &+ 2\lambda||\sqrt{D_1}\sqrt{D_2}\overline{Q})||_F^2 \\ \text{s.t. } &\overline{P}^T\overline{P} = I^{d\times d}\end{aligned} \tag{6}$$

where $\overline{Q} \in R^{m\times d}$ is first used to project the weighted data matrix $\sqrt{D_1}X$ and $\overline{P} \in R^{m\times d}$ is then used to recover it.

2.3 Optimal Solution

The solution of eq. (6) is divided into two steps.

2.3.1 Step 1 of Optimal Solution

In the Step 1, we are fixing the parameter $\overline{P}$ and want to update the parameter $\overline{Q}$.

Now, for a given $\overline{P}$, there exists an optimal matrix $\overline{P}_\perp$ such that $[\overline{P}, \overline{P}_\perp]$ is $m \times m$ column orthogonal matrix.

Then, optimization problem becomes

$$\arg\min_{Q} ||\sqrt{D_1}^T X - \overline{PQ}^T \sqrt{D_1}X)||_F^2 + \lambda||\sqrt{D_1}\sqrt{D_2}\overline{Q})||_F^2 \tag{7}$$

Now, we can reduce the first term of eq. (7) as

$$||X^T\sqrt{D_1}\overline{P} - X^T\sqrt{D_1}\overline{Q}||_F^2 + ||X^T\sqrt{D_1}\overline{P}_\perp||_F^2 \tag{8}$$

Since $\overline{P}$ is fixed and $||X^T\sqrt{D_1}\overline{P}_\perp||_F^2$ is constant, the optimization problem becomes

$$\arg\min_{Q} ||X^T\sqrt{D_1}\overline{P} - X^T\sqrt{D_1}\overline{Q}||_F^2 + \lambda||\sqrt{D_1}\sqrt{D_2}\overline{Q})||_F^2 \tag{9}$$

By the derivatives of eq. (9) w.r.t $\overline{Q}$ to be 0, we get,

$$\overline{Q} = (\lambda\sqrt{D_1}D_2\sqrt{D_1} + \sqrt{D_1}XX^T\sqrt{D_1})^{-1}\sqrt{D_1}XX^T\sqrt{D_1}\overline{P} \tag{10}$$

Therefore,

$$Q = (\lambda D_2 + XX^T)^{-1}XX^T\sqrt{D_1}\overline{P} \tag{11}$$

2.3.2 Step 2 of Optimal Solution

In the Step 2, we are fixing the parameter $\overline{Q}$ and want to update the parameter $\overline{P}$.

Then for a given $\overline{Q}$, we have to compute $\overline{P}$,and the optimization problem becomes

$$\arg\min_{\overline{P}} ||\sqrt{D_1}X - \overline{P}\overline{Q}^T\sqrt{D_1}X)||_F^2 \text{ s.t. } \overline{P}^T\overline{P} = I^{d\times d} \tag{12}$$

The update of $\overline{P}$ of minimizing of eq. (11) with the given constraint means that $\overline{P}$ is orthogonal in the columns. In order to compute $\overline{P}$, we introduce a lemma.

Lemma 1: Let $Z^{n\times m}$ and $V^{n\times d}$ be two matrices. Consider the constrained minimization problem,

$$\arg\min_{\overline{P}} ||Z - VP^T||_F^2 \text{ s.t. } P^TP = I^{d\times d} \tag{13}$$

Suppose the SVD of Z^TV is EDU^T, then the optimal solution is $P = EU^T$.

The proof involves simplification of the terms, and observing that

$$\text{tr}(AB) \leq \text{tr}(A) \tag{14}$$

where A is a diagonal matrix with non-negative entries and B is unitary.

Here, we have

$$Z^TV = \sqrt{D_1}XX^T\sqrt{D_1}\overline{Q} \tag{15}$$

And if the SVD of $\sqrt{D_1}XX^T\sqrt{D_1}\overline{Q} = EDU^T$, then according to the Lemma 1, we have,

$$\overline{P} = EU^T \tag{16}$$

Therefore,

$$P = \sqrt{D_1}^{-1}EU^T \tag{17}$$

2.4 Convergence Analysis

Before giving the proof of convergence of the proposed optimal algorithm, we need to give the following lemma.

Lemma 2: For any non-zero vectors $p, q \in R^c$, the following result holds:

$$||p||_2 - \frac{||p||_2^2}{2||q||_2} \leq ||q||_2 - \frac{||q||_2^2}{2||q||_2} \tag{18}$$

Theorem 1: Given all the variables in eq. (1) except P, Q, the optimization problem in eq. (1) will monotonically decrease the objective function value in each iteration and converge to the local optimal solution.

Proof:
We denote the objective function as $J(Q, P) = J(Q, P, D_1, D_2)$. Suppose for the $(t-1)$-th iteration, we

obtain $P^{(t-1)}, Q^{(t-1)}, D_1^{(t-1)}$ and $D_2^{(t-1)}$. Since we are updating Q by taking derivative w.r.t. Q, so from eq (11), we can say

$$J(Q^{(t)}, P^{(t-1)}, D_1^{(t-1)}, D_2^{(t-1)}) \leq J(Q^{(t-1)}, P^{(t-1)}, D_1^{(t-1)}, D_2^{(t-1)}) \tag{19}$$

Since the SVD gives the optimal $P^{(t)}$ that further decreases the objective value, we have

$$J(Q^{(t)}, P^{(t)}, D_1^{(t-1)}, D_2^{(t-1)}) \leq J(Q^{(t-1)}, P^{(t-1)}, D_1^{(t-1)}, D_2^{(t-1)}) \tag{20}$$

Once the optimal $P^{(t)}$ and $Q^{(t)}$ are obtained, we have

$$\begin{aligned}&\mathrm{tr}((X - P^{(t)}Q^{(t)}X)^T D_1^{(t-1)}(X - P^{(t)}Q^{(t)}X)) + \lambda(\mathrm{Q^{(t)T}})\mathrm{D_2^{(t-1)}}\mathrm{Q^{(t)}}\\ &\leq \mathrm{tr}((X - P^{(t-1)}Q^{(t-1)}X)^T D_1^{(t-1)}(X - P^{(t-1)}Q^{(t-1)}X)) + \lambda(\mathrm{Q^{(t-1)T}})\mathrm{D_2^{(t-1)}}\mathrm{Q^{(t-1)}}\end{aligned}$$

That is,

$$\sum_{i=1}^{m} \frac{||X - P_i^{(t)}Q_i^{(t)^T}X||_2^2}{||X - P_i^{(t-1)}Q_i^{(t-1)T}X||_2} + \lambda \sum_{i=1}^{m} \frac{||Q_i^{(t)}||_2^2}{||Q^{(t-1)}||_2} \leq \sum_{i=1}^{m} \frac{||X - P_i^{(t-1)}Q_i^{(t-1)T}X||_2^2}{||X - P_i^{(t-1)}Q_i^{(t-1)T}X||_2} + \lambda \sum_{i=1}^{m} \frac{||Q_i^{(t)}||_2^2}{||Q^{(t-1)}||_2} \tag{21}$$

From Lemma 2, we have,

$$||X - P_i^{(t)}Q_i^{(t)^T}X||_2 - \frac{||X - P_i^{(t)}Q_i^{(t)^T}X||_2^2}{||X - P_i^{(t-1)}Q_i^{(t-1)T}X||_2} \leq ||X - P_i^{(t-1)}Q_i^{(t-1)T}X||_2 - \frac{||X - P_i^{(t-1)}Q_i^{(t-1)T}X||_2^2}{||X - P_i^{(t-1)}Q_i^{(t-1)T}X||_2} \tag{22}$$

Using the matrix calculus on eq. (22), we have,

$$\begin{aligned}&\sum_{i=1}^{m} \left(||X - P_i^{(t)}Q_i^{(t)^T}X||_2 - \frac{||X - P_i^{(t)}Q_i^{(t)^T}X||_2^2}{||X - P_i^{(t-1)}Q_i^{(t-1)T}X||_2} \right)\\ &\leq \sum_{i=1}^{m} \left(||X - P_i^{(t-1)}Q_i^{(t-1)T}X||_2 - \frac{||X - P_i^{(t-1)}Q_i^{(t-1)T}X||_2^2}{||X - P_i^{(t-1)}Q_i^{(t-1)T}X||_2} \right)\end{aligned} \tag{23}$$

Again, from Lemma 2, we have,

$$||Q_i^{(t)}||_2 - \frac{||Q_i^{(t)}||_2^2}{||Q_i^{(t-1)}||_2} \leq ||Q_i^{(t-1)}||_2 - \frac{||Q_i^{(t-1)}||_2^2}{||Q_i^{(t-1)}||_2} \tag{24}$$

Again, using matrix calculation, we have,

$$\sum_{i=1}^{m} \left(||Q_i^{(t)}||_2 - \frac{||Q_i^{(t)}||_2^2}{||Q_i^{(t-1)}||_2} \right) \leq \sum_{i=1}^{m} \left(|||Q_i^{(t-1)}||_2 - \frac{||Q_i^{(t-1)}||_2^2}{||Q_i^{(t-1)}||_2} \right) \tag{25}$$

Combining eq. (21), (23) and (25), we have

$$\begin{aligned}&J(Q^{(t)}, P^{(t)}, D_1^{(t-1)}, D_2^{(t-1)}) = ||XP^{(t)}Q^{(t)T}X||_{2,1} + \lambda||Q^{(t)}||_{2,1}\\ &\leq ||XP^{(t-1)}Q^{(t-1)T}X||_{2,1} + \lambda||Q^{(t-1)}||_{2,1}\\ &= J(Q^{(t-1)}, P^{(t-1)}, D_1^{(t-1)}, D_2^{(t-1)})\end{aligned} \tag{26}$$

That is,

$$J(Q^{(t)}, P^{(t)}) = J(Q^{(t)}, P^{(t)}, D_1^{(t)}, D_2^{(t)}) \leq J(Q^{(t-1)}, P^{(t-1)}, D_1^{(t-1)}, D_2^{(t-1)}) = J(Q^{(t-1)}, P^{(t-1)})$$

Hence, proved. ■

3 Algorithm and Implementation

3.1 JSPCA Algorithm

Input: Training sample set X, parameter λ and dimensionality d.

1. Initialize $D_1 = I^{m \times m}$ and random $\overline{P}^{m \times d}$.
2. while not converge do
   - 2.1 Compute $\overline{Q}$ according to eq. (10)
   - 2.2 Compute Q according to eq. (11)
   - 2.3 Compute $\overline{P}$ according to eq. (16)
   - 2.4 Compute P according to eq. (17)
   - 2.5 Compute D_1 according to eq. (3)
   - 2.6 Compute D_2 according to eq. (4)
3. Normalize each column vector of Q to be identity vectors.

Output: Transformation matrix Q

3.2 Implementation

Using MATLAB, we implemented the above steps of the JSPCA iterative algorithm.
Github link for JSPCA code

One iteration of JSPCA is captured as a MATLAB function, which we can run either for a fixed number of iterations or till the cost function converges.

The code outputs the computed matrix Q as well as the decreasing cost-values across the iterations performed.

It is worth noting that since we have an iterative algorithm which relies on a regularization term to get sparse loadings, the values do not absolutely converge to 0.
Instead, they reduce to small values on the order of 10^{-2}. We must set a threshold there to trim these values to absolute 0. This step is necessary, say, when we want to quantify the sparsity of Q.

3.3 Results on the Breast-Cancer Wisconsin dataset

The dataset is used for investigating samples of breast-tissue to study the characteristics of cancerous breast cells. The features (columns) of the dataset are derived from the image data and describe the features of breast cells such as cell radius, smoothness, symmetry etc.
There are 31 columns in the dataset and the highly correlated nature of the input features highlights the need for dimensionality reduction techniques.

We run JSPCA on this dataset with the parameters:

- $d = 6$ (No. of output dimensions, i.e. No. of principal components)
- $\lambda = 3.0$ (Regularization parameter)
- $NumIterations = 50$
  We stop after 50 iterations are completed and assume convergence.

No. of total loadings	186	No. of Input Features	31	Total Variance(normalized)	31
No. of 0 (sparse) loadings	152	No. of features removed	16	Variance of PCs	8.56
Sparsity	81.7%	Joint-Sparsity	51.6%	Variance Explained	27.6%

Table 1: Results of running JSPCA

3.4 PCA vs JSPCA

We can compare the performance of PCA vs JSPCA by running both algorithms in MATLAB on the same Breast-Cancer dataset to find the first 6 PC's (principal components).

- JSPCA has eliminated 16 of the 31 input features entirely, resulting in a much smaller feature space. On the other hand, PCA results in no sparsity and cannot reduce the feature space at all.
- PCA was able to capture 88% of the variance in the first 6 PC's. JSPCA captured only 28% of the variance in the same number of PC's.

The PC's calculated by the vanilla PCA algorithm were oriented in very different directions compared to the PC's calculated by JSPCA.
We may reason that - since the algorithm and the corresponding cost function are different, along with removing the orthogonal constraint and asking for joint-sparsity to be there in the PC's, the resulting PC's found are bound to be different from the PC's found by vanilla PCA.

Hence, the components found by JSPCA are not principal components in a strict sense, but instead are components that can capture a moderate amount of the dataset's variance while simultaneously identifying redundant features through joint-sparsity.

3.5 Computational Complexity

In each iteration, there are two main steps:

1. Computing $Q = (\lambda D_2 + XX^T)^{-1} XX^T \sqrt{D_1} P$ takes $\mathcal{O}(m^3)$ time.
2. Computing SVD of $\sqrt{D_1} XX^T \sqrt{D_1} Q = EDU^T$ also takes $\mathcal{O}(m^3)$ time at most.

So, computational complexity for one iteration will be up to $\mathcal{O}(m^3)$.
For t iterations of the algorithm, it will be $\mathcal{O}(tm^3)$.

References

[1] Hui ZOU, Trevor HASTIE, and Robert TIBSHIRANI, Sparse Principal Component Analysis.

[2] Hui Zou and Trevor Hastie, Regularization and Variable Selection via the Elastic Net

[3] Sparse PCA Wikipedia Page

[4] Breast-Cancer Wisconsin dataset