Robust Principal Component Analysis on Graphs

Sofiane Ezzehi
École Normale Supérieure Paris-Saclay
École des Ponts ParisTech
sofiane.ezzehi@eleves.enpc.fr

Alexandre Lutt
École Normale Supérieure Paris-Saclay
École des Ponts ParisTech
alexandre.lutt@eleves.enpc.fr

Figure 1: Low-rank reconstruction of corrupted images using the proposed method of [1].

1 ABSTRACT

Principal Component Analysis (PCA) is a very popular method for dimensionality reduction, and is used by thousands accross the world to provide 2D or 3D visualisations and insights about high-dimension data.

However, its main drawback is that it is very sensitive to outliers, and thus cannot be used in many real-world applications. This issue has been solved by the introduction of a robust variants of PCA, RPCA [3]. However, this algorithm is very slow on large datasets, which makes it impractical for many applications. This motivated the introduction of another PCA variant called GLPCA [2] which uses graph Laplacian regularization to improve the robustness of the algorithm while keeping the computational cost low. Finally, a third variant of PCA similar to the previous ones has been proposed by the authors of [1] to improve the robustness of the algorithm while keeping the computational cost low.

This project aims to provide a simple and efficient implementation of those main variants of the PCA algorithm, as well as a benchmark of those methods on different tasks (clustering and low-rank recovery for corrupted data on real-life and artificial datsets).

2 INTRODUCTION

3 ALGORITHMS

In this section, we are going to present the algorithms we implemented and compared for the benchmark.

- **Classsical PCA**: We used Scikit-Learn implementation of PCA.
- **RPCA**: We implemented (see algorithm 1) the algorithm described in [3] using the ADMM (Alternating Direction Method of Multipliers) and following as closely as possible the pseudo-code provided in the paper.
- **GLPCA**: We also implemented (see algorithm 2) the algorithm described in [2] using the closed-form solution proved in the original paper.
- **RGLPCA**: Moreover, we also implemented (see algorithm 3) the robust version of the previous algorithm mentionned in [2].
- **Proposed PCA**: Finally, we implemented (see algorithm 4) the algorithm described in [1] which claims to be more robust than the previous ones while keeping the computational cost low.

In order to make notations more concise, we will define the following operators (defined on the space of square matrices):

Shrinkage operator:

$$\mathcal{S}_\tau(\mathbf{Z}) = \text{sign}(\mathbf{Z}) \times \max\left(\mathbf{0}, |\mathbf{Z}| - \tau\mathbf{I}\right)$$

Singular value tresholding operator:

$$\mathcal{D}_\tau(\mathbf{Z}) = \mathbf{U}\mathcal{S}_\tau(\Sigma)\mathbf{V}^T \text{ if } \mathbf{Z} = \mathbf{U}\Sigma\mathbf{V}^T$$

Algorithm 1 RPCA algorithm

Require: $\mathbf{X} \in \mathbb{R}^{m \times n}$, ε, n_{iter}
  $\lambda \leftarrow \frac{1}{\sqrt{\max(m,n)}}$
  $\mu \leftarrow \frac{\max(m,n)}{4 \times ||\mathbf{X}||_1}$
  $\mathbf{S} \leftarrow \mathbf{0}$
  $\mathbf{Y} \leftarrow \mathbf{0}$
  for $i = 1$ **to** n_{iter} **do**
    $\mathbf{L} \leftarrow \mathcal{D}_{\mu}(\mathbf{X} - \mathbf{S} + \frac{1}{\mu}\mathbf{Y})$
    $\mathbf{S} \leftarrow \mathcal{S}_{\frac{\lambda}{\mu}}(\mathbf{X} - \mathbf{L} + \frac{1}{\mu}\mathbf{Y})$
    $\mathbf{Y} \leftarrow \mathbf{Y} + \mu(\mathbf{X} - \mathbf{L} - \mathbf{S})$
    if $||\mathbf{X} - \mathbf{L} - \mathbf{S}||_F \leqslant \varepsilon$ **then**
      break
    end if
  end for
  return L, S

Algorithm 2 GLPCA algorithm

Require: $\mathbf{X} \in \mathbb{R}^{m \times n}$, $\mathcal{G}$, β
  $\mathbf{W} \leftarrow adj(\mathcal{G})$
  $\mathbf{D} \leftarrow diag(\{d_1, d_2, ..., d_{n_{\text{nodes}}}\})$
  $\mathbf{L} \leftarrow \mathbf{D} - \mathbf{W}$
  $\lambda_n \leftarrow \Re(\max(eigenval(\mathbf{X}^T\mathbf{X}))$
  $\xi_n \leftarrow \Re(\max(eigenval(\mathbf{L}))$
  $\mathbf{G}_\beta \leftarrow (1 - \beta)(\mathbf{I} - \frac{1}{\lambda_n}\mathbf{X}^T\mathbf{X}) + \frac{\beta}{\xi_n}\mathbf{L}$
  $\mathbf{Q} \leftarrow \Re(eigenvect(\mathbf{G}_\beta))$
  $\mathbf{U} \leftarrow \mathbf{X}\mathbf{Q}$
  return Q, U

Algorithm 3 RGLPCA algorithm

Require: $\mathbf{X} \in \mathbb{R}^{m \times n}$, $\mathcal{G}$, β, k, ρ, n_{iter}
  $\mathbf{X}_0 \leftarrow \mathbf{X}$
  $\beta_0 \leftarrow \beta$
  $\mathbf{E} \leftarrow \mathbf{1}$
  $\mathbf{C} \leftarrow \mathbf{1}$
  $\mu \leftarrow 1$
  $\mathbf{W} \leftarrow adj(\mathcal{G})$
  $\mathbf{D} \leftarrow diag(\{d_1, d_2, ..., d_{n_{\text{nodes}}}\})$
  $\mathbf{L} \leftarrow \mathbf{D} - \mathbf{W}$
  $\xi \leftarrow \Re(max(eigenval(\mathbf{X}_0^T\mathbf{X}_0)))$
  for $i = 1$ **to** n_{iter} **do**
    $\mathbf{X}_i \leftarrow \mathbf{X} - \mathbf{E} - \frac{1}{\mu}\mathbf{C}$
    $\lambda \leftarrow \Re(\max(eigenval(\mathbf{X}_i^T\mathbf{X}_i)))$
    $\alpha \leftarrow \frac{2\beta_0\lambda}{\mu(1-\beta_0)\xi}$
    $\beta \leftarrow \frac{\alpha\xi}{\lambda+\alpha\xi}$
    $\mathbf{Q}, \mathbf{U} \leftarrow GLPCA(\mathbf{X}, \mathcal{G}, \beta, k)$
    $\mathbf{A} \leftarrow \mathbf{X}_k - \mathbf{U}\mathbf{Q}^T - \frac{1}{\mu}\mathbf{C}$
    $\mathbf{a} \leftarrow (||\mathbf{A}_1||_2, ||\mathbf{A}_2||_2, ..., ||\mathbf{A}_m||_2)$
    $\mathbf{E} \leftarrow \mathcal{S}_{\frac{1}{\mu}}(\mathbf{a})$
    $\mathbf{C} \leftarrow \mathbf{C} + \mu(\mathbf{E} - \mathbf{X} - \mathbf{U}\mathbf{Q}^T)$
    $\mu \leftarrow \rho\mu$
  end for
  return Q, U, E

Algorithm 4 Proposed PCA algorithm

Require: $\mathbf{X} \in \mathbb{R}^{p \times n}$, $\mathcal{G}$, n_{iter}, γ, ε
  $\mathbf{A} \leftarrow adj(\mathcal{G})$
  $\mathbf{D} \leftarrow diag(\{d_1, d_2, ..., d_{n_{\text{nodes}}}\})$
  $\boldsymbol{\phi} \leftarrow D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$
  $\lambda \leftarrow \frac{1}{\sqrt{\max(p,n)}}$
  $\mathbf{L} \leftarrow random(p, n)$
  $\mathbf{W} \leftarrow random(p, n)$
  $\mathbf{S} \leftarrow random(p, n)$
  $r_1 \leftarrow 1$
  $r_2 \leftarrow 1$
  $\mathbf{Z}_1 \leftarrow \mathbf{X} - \mathbf{L} - \mathbf{S}$
  $\mathbf{Z}_2 \leftarrow \mathbf{W} - \mathbf{L}$
  for $i = 1$ **to** n_{iter} **do**
    $\mathbf{H}_1 \leftarrow \mathbf{X} - \mathbf{S} + \frac{1}{r_1}\mathbf{Z}_1$
    $\mathbf{H}_2 \leftarrow \mathbf{W} + \frac{1}{r_2}\mathbf{Z}_2$
    $\mathbf{H} \leftarrow \frac{1}{r_1+r_2}(r_1\mathbf{H}_1 + r_2\mathbf{H}_2)$
    $\mathbf{L} \leftarrow \mathcal{D}_{\frac{2}{r_1+r_2}}(\mathbf{H})$
    $\mathbf{S} \leftarrow \mathcal{S}_{\frac{\lambda}{r_1}}(\mathbf{X} - \mathbf{L} + \frac{1}{r_1}\mathbf{Z}_1)$
    $\mathbf{W} \leftarrow r_2(\gamma\boldsymbol{\phi} + r_2\mathbf{I})^{-1}(L - \frac{1}{r_2}Z_2)$
    $\mathbf{Z}_1 \leftarrow \mathbf{Z}_1 + r_1(\mathbf{X} - \mathbf{L} - \mathbf{S})$
    $\mathbf{Z}_2 \leftarrow \mathbf{Z}_2 + r_2(\mathbf{W} - \mathbf{L})$
    if $||\mathbf{X} - \mathbf{L} - \mathbf{S}||_F \leqslant \varepsilon$ **then**
      break
    end if
  end for
  return L, S

4 EXPERIMENTAL SETUP

5 RESULTS

6 LIMITATIONS

7 CONCLUSION

REFERENCES

[1] Patricia S. Abril and Robert Plant. 2015. Robust Principal Component Analysis on Graphs. (April 2015). http://arxiv.org/abs/1504.06151
[2] Bin Luo Bo Jiang, Chris Ding and Jin Tang. 2013. Graph-Laplacian PCA: Closed-form Solution and Robustness. (January 2013).
[3] Yi Ma Emmanuel J. Candès, Xiaodong Li and John Wright. 2009. Robust Principal Component Analysis? (December 2009). http://arxiv.org/abs/0912.3599