Second-Order PCA-Style Dimensionality Reduction Algorithm by Semidefinite Programming

Rong Jin, Wei Tong
Department of Computer Science and Engineering
Michigan State University, East Lansing, MI 48824
{rongjin, tongwei}@cse.msu.edu

1 Introduction and related work

Principal Component Analysis(PCA) is a widely used dimensionality reduction method in computer vision, pattern recognition and machine learning. It is however computationally expensive when the dimension of data is very high. Recently, several second-order PCA-style (SOPCA) dimensionality reduction algorithms have been proposed in order to overcome the computational difficulty in applying PCA to matrices that are conventionally viewed as vectors of high dimensionality. The main idea behind these algorithms is to avoid converting a matrix to a high dimensional vector but use the matrix representation directly.

One of the most representative work of SOPCA is Generalized Low Rank Approximations of Matrices(GLRAM) [4]. It reduces the dimensionality of a matrix by multiplying it with a left and a right projection matrices. More specifically, we denote $\mathcal{M} = (M_1, M_2, \dots, M_n)$ as the collection of matrices for dimensionality reduction, where each $M_i \in \mathbb{R}^{p\times q}$ is of size $p \times q$. The goal is to identify two matrices $L \in \mathbb{R}^{p\times k_1}$ and $R \in \mathbb{R}^{q\times k_2}$, such that each matrix M_i can be well approximated by $LA_iR^\top$ where $A_i \in \mathbb{R}^{k_1\times k_2}$. Often in practice, we set $k_1 = k_2 = k$, where k is the target dimensionality to be reduced to. It is casted into the following optimization problem:

$$\begin{aligned} \min_{L,R,A_i} \quad & \sum_{i=1}^{n} \|M_i - LA_iR^\top\|_2^2 \\ \text{s. t.} \quad & L^\top L = R^\top R = I_k \end{aligned} \tag{1}$$

It is well known that (1) is a non-convex optimization problem, and usually an iterative procedure is employed to obtain the local optimal solution. Other related works like 2-D SVD and Tensor-PCA encounter the similar non-convex optimization problems and only local optimal solutions can be achieved.

2 A convex formulation for SOPCA

The key difficulty in identifying the left and right projection matrices L and R in GLRAM arises from their dependency, namely, the solution of L depends on the solution of R and vice versa. To address this difficulty, we consider first rewriting each matrix M_i into the product of two matrices U_i and V_i, and then compute the optimal projection matrices L and R based on the eigenvectors of $\sum_{i=1}^n U_iU_i^\top$ and $\sum_{i=1}^n V_iV_i^\top$. Therefore, the key question in this framework is to decide the appropriate factorization for M_i.

In order to motivate the right formulation for matrix dimensionality reduction, we first consider the case with a single matrix M. It is well known that the optimal projection matrices are from the singular value decomposition of M, which is related to the definition of the trace norm or Ky Fan norm [1]:

Definition 1. *The trace norm $\|M\|_{tr}$ of a matrix is given by any of the three quantities:*

1. $\min_{U,V}\left\{\|U\|_F\|V\|_F : M = UV^\top\right\}$

2. $\min_{U,V}\left\{\frac{1}{2}(\|U\|_F^2 + \|V\|_F^2) : M = UV^\top\right\}$

3. The sum of the singular values of M

Furthermore, If $M = U\Sigma V^\top$ is the singular value decomposition of M, then the matrices $U\sqrt{\Sigma}$ and $V\sqrt{\Sigma}$ minimize the first two quantities.

Minimizing the trace norm of M can be represented as a Semidefinite Programming (SDP) [2] problem:

$$\begin{aligned} \min_{P\in\mathbb{S}_+^{p\times p}, Q\in\mathbb{S}_+^{q\times q}} \quad & \mathrm{tr}(P) + \mathrm{tr}(Q) \\ \text{s. t.} \quad & \begin{pmatrix} P & M \\ M^\top & Q \end{pmatrix} \succeq 0 \end{aligned} \tag{2}$$

where $\mathbf{S}_+^{n\times n}$ represents positive semidefinite matrices, P and Q represent a particular factorization of matrix M. We then extend (2) to the problem of extracting the largest k singular values of matrix M, known as the Ky Fan k-norm [3], and further apply the idea to multiple matrices. We finally get the problem as:

$$\begin{aligned} \min_{P_i\in\mathbf{S}_+^{p\times p}, Q_i\in\mathbf{S}_+^{q\times q}} \quad & S_k\left(\sum_{i=1}^n P_i\right) + S_k\left(\sum_{i=1}^n Q_i\right) \\ \text{s. t.} \quad & \begin{pmatrix} P_i & M_i \\ M_i^\top & Q_i \end{pmatrix} \succeq 0,\ i = 1, 2, \ldots, n \end{aligned} \tag{3}$$

where $S_k(M)$ denotes the Ky Fan k-norm of matrix M. It can be shown that the optimal value of (3), denoted by f_1, and the optimal value of (1), denoted by f_2, have the relationship as $f_2 \le \frac{1}{2}f_1^2$, which indicates that f_1 provides an upper bound for f_2. This, to some degree, justifies the usage of (3) for finding the optimal projection for matrices.

Solving the optimization problem in (3) is difficult due to function $S_k(M)$, however, it can be converted to an equivalent SDP formulation:

$$\begin{aligned} \max_{T_p\in\mathbf{S}_+^{p\times p} T_Q\in\mathbf{S}_+^{q\times q} Z_i\in\mathbb{R}^{p\times q}} \quad & \sum_{i=1}^n \mathrm{tr}(Z_i^\top M_i) \\ \text{s. t.} \quad & \begin{pmatrix} T_P & Z_i \\ Z_i^\top & T_Q \end{pmatrix} \succeq 0,\ i = 1, 2, \ldots, n \\ & T_P \preceq I,\ T_Q \preceq I,\ \mathrm{tr}(T_P) \le k,\ \mathrm{tr}(T_Q) \le k \end{aligned} \tag{4}$$

The problem (4) is convex, thus the global optimal solution can be achieved. Directly solving (4) for large-size data sets, however, could be computational expensive. We have constructed an alternative algorithm which obtains an approximate solution to (4). Our preliminary experiments on the task of face image classification show encouraging results of the proposed algorithm in comparison to the state-of-the-art algorithms for SOPCA.

Topic:learning algorithms, **Preference: oral/poster: Rong Jin**

References

[1] N. Srebro. *Learning with Matrix Factorizations.* PhD thesis, MIT, Aug. 2004.

[2] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996.

[3] G. A. Watson. On matrix approximation problems with Ky Fan k-norms. *Numerical Algorithms*, (5):5–263, 1993.

[4] J. Ye. Generalized low rank approximations of matrices. *Machine Learning*, 61:167–191, 2005.