Fast optimization of common basis for matrix set through Common Singular Value Decomposition

Jarek Duda
Jagiellonian University, Golebia 24, 31-007 Krakow, Poland, Email: dudajar@gmail.com

Abstract—SVD (singular value decomposition) is one of the basic tools of machine learning, allowing to optimize basis for a given matrix. However, sometimes we have a set of matrices \( \{A_k\}_k \) instead, and would like to optimize a single common basis for them: find orthogonal matrices \( U, V \), such that \( \{U^T A_k V\} \) set of matrices is somehow simpler. For example DCT-II is orthonormal basis of functions commonly used in image/video compression - as discussed here, this kind of basis can be quickly automatically optimized for a given dataset. While also discussed gradient descent optimization might be computationally costly, there is proposed CSVD (common SVD): fast general approach based on SVD. Specifically, we choose \( U \) as built of eigenvectors of \( \sum_k (w_k)^2 (A_k^T A_k)^p \) and \( V \) of \( \sum_k (w_k)^2 (A_k^T A_k)^q \), where \( w_k \) are their weights, \( p, q > 0 \) are some chosen powers e.g. 1/2, optionally with normalization e.g. \( A \rightarrow A - re^T \) where \( r_i = \sum_j A_{ij}, \ c_j = \sum_i A_{ij} \).

Keywords: machine learning, statistics, feature extraction, matrix set, function basis, SVD, PCA, DCT, data compression, hierarchical correlation analysis

I. INTRODUCTION

SVD (singular value decomposition) \cite{1}, PCA (principal component analysis) \cite{2}, Karhunen-Loève transform \cite{3}, CCA (canonical correlation analysis) \cite{4}, MCA (multiple correspondence analysis) are related basic tools of statistics and machine learning, e.g. allowing to optimize a basis. They are focused on optimization of basis a single matrix, bringing a natural question of extensions to optimization of single basis simultaneously for multiple matrices.

PCA can be seen as applying SVD to covariance matrix \( C_{XX} = E[(X - E[X])(X - E[X])^T] \), in CCA cross-covariance matrix \( E[XY^T] \) for whitened variables \( X = C^{-1/2}_{XX}(X - E[X]) \). If splitting the sample into subsamples in \( (w_k) \) proportion, the final e.g. covariance matrix is weighted average of matrices for these subsamples with \( (w_k) \) weights.

Hence optimization of common basis of set of \( \{A_k\}_k \) with \( (w_k) \) weights, can be performed with SVD of \( \sum_k w_k A_k \). The question is what if they are a different type of matrices?

Practical application example is optimization of basis \( (f_i(x))_i \) for which we want to use product basis \( (f_i(x)f_j(y))_{ij} \). It is convenient to work with orthonormal basis, but there is large freedom of choosing a specific one. For example in lossy image/video compression like JPEG \cite{6} there was finally chosen DCT-II \cite{7}: \( \cos(\pi(n + 1/2)k/N) \). But there are 4 types of DCT (discrete cosine transform),
also bases using sines instead, orthonormal polynomials, etc. - bringing question why to choose DCT-II? Maybe we could choose a better basis optimized for a given dataset? Split data using bases optimized for each cluster? To optimize it based on dataset, we can e.g. take $8 \times 8$ pixel blocks, perform PCA getting 64 eigenvectors giving $8 \times 8$ matrices, and eigenvalues suggesting their importance: weights, then optimize common basis for these 64 matrices - done in Fig. 2, confirming that DCT-II is a very good choice, and allowing further e.g. local optimizations.

The direct author’s motivation was very similar, presented in Fig. 1 - also optimization of basis (starting with orthonormal polynomials), which product basis we want to optimize based on PCA from dataset - this time representing joint probability distribution in autocorrelation analysis. This application allows to optimize more universal features - instead of dataset dependent PCA features, optimize basis to further use simple universal canonical features.

The article further discusses direct optimization of chosen evaluation function - through gradient descend in space of orthogonal matrices, however, it is extremely costly and often non-unique. There is also proposed CSVD approach family (common SVD): which is SVD for weighted sum of $(A_k A_k^T)^p$, with freedom of choosing the power $p$ (or a different function), adding normalization, etc.

II. DIRECT OPTIMIZATION BY GRADIENT DESCEND

The choice of $U, V$ orthogonal matrices, transforms the given $\{A_k\}$ matrices into $\{U^T A_k V\}$ with coefficients:

$$c_{ijk} = (U^T A_k V)_{ij} = (U_{\bullet i})^T A_k V_{\bullet j}$$

(1)

for $U_{\bullet i} = (U_{ji})_i, V_{\bullet j} = (V_{ij})_i$ denoting their columns as vectors of the optimized orthonormal bases.

We need to define the evaluation function on $(c_{ijk})$. Its simple form is coordinate-wise:

$$\min_{U,V:UU^T=I_n,VV^T=I_m} g \quad \text{for} \quad g = \sum_{ijk} w_k f(c_{ijk})$$

(2)

Generally it might be also worth to consider matrix-wise evaluation functions: $\sum_k w_k f((c_{ijk})_i)$ using more complex matrix norms/functions, or even completely general $f((w_k)c_{ijk}, (e_{ijk})_{i,j,k})$ e.g. optimizing maximal norm over the matrix set.

Optimization over orthogonal matrices, after a step $U \rightarrow U(1 + \epsilon G)$ needs to remain orthogonal: $I = UU^T \rightarrow U(I + \epsilon G)(I + \epsilon G^T)U^T$, zeroing of $\epsilon$ term says generator $G$ has to be anti-symmetric: $G^T = -G$.

Hence the $G, H$ generators have correspondingly $n(n-1)/2, m(m-1)/2$ coefficients, which can be referred with $G_{ab} = -G_{ba}, H_{ab} = -H_{ba}$ indexes for $a < b$. 

---

Figure 2. Comparison of DCT-II basis popular in image/video compression (left, orange plots), with basis optimized by proposed CSVD (right, blue plots). For dataset of 48 grayscale 8bit images $512 \times 512$ (source: http://decsai.ugr.es/cvg/CIBase.htm), there were extracted $48 \times 64 \times 64$ blocks of $8 \times 8$ pixels, for which there was performed PCA - getting 64 eigenvectors, presented in DCT-II basis (top left). For these 64 matrices $8 \times 8$ there was also applied discussed CSVD optimization (with eigenvalues as weights), getting basis plotted in blue, in which these 64 matrices are presented top right. We can see this optimization has automatically lead to nearly identical basis as DCT-II, confirming this choice and e.g. allowing dedicated local optimizations for some specific datasets. Presented 64 matrices ideally should have only a single nonzero values - such basis is used in practice, but this is not exactly true here (allowing for optimizations), the optimized ones on the right are visually slightly smoother.
This way e.g. \( G_{ab} \) adds \( U_{ab} \rightarrow U_{ab} + G_{ab}U_{ab} \) and also subtracts \( U_{ab} \rightarrow U_{ab} - G_{ab}U_{ab} \), leading to derivatives for \( U \rightarrow U(1 + \epsilon G) \), \( V \rightarrow V(1 + \epsilon H) \):

\[
\frac{\partial g}{\partial G_{ab}} = \sum_{jk} w_k (f'(c_{ajk})c_{akj} - f'(c_{ajk})c_{bjk})
\]

\[
\frac{\partial g}{\partial H_{ab}} = \sum_{ik} w_k (f'(c_{ibk})c_{iak} - f'(c_{ibk})c_{ibk})
\]

For square matrix and symmetric \( U^T = V \) case, we have only a single generator \( G = H \) and derivative is sum of the two above. Having the gradient we can perform gradient descent step \( U \rightarrow U(1 - \epsilon G), V \rightarrow V(1 - \epsilon H) \), which needs to interleaved with some orthonormalization steps e.g. Gram-Schmidt. The choice of \( \epsilon > 0 \) step is difficult, there can be tested a few and chosen the one with best evaluation.

In bottom of Fig. there are shown effects of such direct coordinate-wise optimization for two coordinate-wise:

- \( f(x) = |x| \): L1-like evaluation function hoping to get sparse representation like in so called lasso, and for
- \( f(x) = -x^4 \): sums of 2nd powers are rotation invariant, maximizing 4th powers wants to get high contrasts.

It is quite costly numerically, for 10 \( \times 10 \) matrices already requiring thousands of gradient descent steps. It might be worth to expand to 2nd order methods.

Also the results are not that satisfactory - visually better looking and much less expensive computationally is the discussed next SVD-based optimization, which could be also used as initial point for further direct optimization.

### III. CSVD - COMMON SVD OPTIMIZATION

Returning to PCA analogy: \( E[(X - E[X])(X - E[X])^T] \) covariance matrixes for subsets of dataset in \( \{w_k\}_k \) proportions, covariance matrix for the entire dataset is their weighted average with \( \{w_k\}_k \) weights - we can optimize the common basis through SVD of this weighted average.

Having general \( \{A_k\}_k \) set of matrices of the same size \( n \times m \) with \( \{w_k\}_k \) weights, in some cases it might be worth to directly consider their weighted average \( M = \sum_k w_k A_k \) and then perform SVD. what in practice can be done by calculating eigenvectors of \( MM^T \) and of \( M^T M \) (both symmetric, positive defined) and building \( U, V \) orthogonal matrices from these eigenvectors as columns.

However, experimentally much better behavior is obtained by performing weighted average of \( n \times n \) matrices \( A_k A_k^T \) and \( m \times m \) matrices \( A_k^T A_k \). In covariance matrix analogy, this way imagining the matrices as \( m \) size \( n \) vector dataset, or the opposite. We can also include some functions before weighting to optimize for specific applications, e.g. powers:

**Common SVD (CSVD):** for \( \{A_k\}_k \) real \( n \times m \) matrices with \( \{w_k\}_k \) weights, and some chosen \( p, q > 0 \) powers: choose orthogonal \( U \) as built of eigenvectors of \( \sum_k (w_k)^q (A_k A_k^T)^p \), and \( V \) of \( \sum_k (w_k)^q (A_k^T A_k)^p \) (optionally \( A_k \) normalized).

### A. Basic analysis

To optimize a common basis, we should not fix the order of its vectors between the matrices. In contrast, SVD finds vectors sorted by singular values. To formulate SVD with permutation freedom, observe that while \( \text{Tr}(MM^T) = \sum_{ij} (M_{ij})^2 \) is unchanged under \( M \rightarrow U^T MV \) transformation, SVD finds \( U, V \) zeroing all non-diagonal coefficients of \( U^T MV \) (leaving only singular values in diagonal), what allows to formulate it through optimization:
Permutation-free SVD of $M$ matrix - maximize:

$$\max_{U:U^TU = I, V:V^TV = I} \sum_i ((U^TMV)_{ii})^2 = \sum_{ij} (M_{ij})^2 = \text{Tr}(MM^T)$$

(3)

For symmetric $M = M^T$ it can be simplified:

$$\max_{U:U^TU = I} \sum_i ((U^TMU)_{ii})^2 = \text{Tr}(M^2) = \sum_i (\lambda_i)^2$$

(4)

Where $\lambda_i$ are the eigenvalues. In discussed CSVD, (4) is applied to linear combination $\sum_k \bar{w}B_k$ for

$$\bar{w}_k = (w_k)^q \quad B_k = (A_kA_k^T)^p$$

or some more complex function e.g. containing normalization. Using (4) we find $UU^T = I$ maximizing:

$$\sum_i \left( (U^TMU_{ii}) \sum_k \bar{w}_kB_k \right)^2 =$$

$$= \sum_i \left( \sum_k \bar{w}_kB_k^2 \right)^2 =$$

$$\sum_{k,i} \bar{w}_k^2U_{i*}(B_k)^2U_{i*} + \sum_{k\neq k',i} \bar{w}_k\bar{w}_{k'}U_{i*}B_{k'}B_kU_{i*}$$

(6)

The left hand side seems naturally weighted (4), especially if choosing $p = q = 1/2$. However, the right hand side part mixes between the matrices, what might be unwanted. Its contribution could be removed by using such found basis as initial step of further direct optimization.

We can also reduce this matrix mixing by normalizing shifting them to be somehow localized around zero. It is done e.g. for covariance matrix by subtracting the means $(X \to X - E[X])$, in MCA by subtracting marginal distribution using both means:

$$A \to A - rc^T \quad \text{for} \quad r_i = \sum_j A_{ij}, \quad c_j = \sum_i A_{ij}$$

(7)

individually for each matrix, maybe also applying whitening e.g. by multiplying by $C_X^{-1/2}$. Tests of various approaches suggest (7) seems the most promising.

Figure 3 contains experiments for 10 of $10 \times 10$ matrices - first randomly filled with $o = 1..16$ of 1s (the remaining positions are 0), then randomly chosen orthogonal matrices and applied to all, then used OSGD and evaluating $\sum_{ijk}(U_{i*}A_{k}V_{j})^4$ divided by of the initial matrix, depending on power $p$ in horizontal axis. We can see e.g. that (7) normalization has allowed for perfect choice of basis for $o = 1$ and improved for small $o$.

It also shows that choice of power $p$ is quite complex and important, could be optimized for various applications. Large $p$ leaves in $B_k$ nearly only projection on the highest eigenvector, tiny $p$ deforms $B_k$ into nearly identity - both can reduce the mixing term, but also deform the optimized condition.

IV. CONCLUSIONS AND FURTHER WORK

Optimization of common basis for set of matrices can allow for nearly free optimization e.g. in data compression, feature extraction, statistical analysis. While it seems a difficult task and direct optimization is quite costly, turns out inexpensive SVD-based approach gives promising results.

One direction for further work is choosing better evaluation functions e.g. for direct optimization, or maybe with inexpensive direct solutions, approximations like SVD-based.

Another related direction is finding applications and optimizing for them, e.g. in data compression - there are now considered various transformations e.g. based on cosines, sines, asymmetric - we could optimize some better ones, also locally based on dataset, maybe use some model clustering [8]: split data into clusters with automatically optimized models.

REFERENCES

[1] G. H. Golub and C. Reinsch, “Singular value decomposition and least squares solutions,” in Linear algebra. Springer, 1971, pp. 134–151.
[2] H. Hotelling, “Analysis of a complex of statistical variables into principal components.” Journal of educational psychology, vol. 24, no. 6, p. 417, 1933.
[3] K. Karhunen, *Ueber lineare Methoden in der Wahrscheinlichkeitsrechnung*. Suomalainen Tiedeakatemia, 1947.
[4] T. R. Knapp, “Canonical correlation analysis: a general parametric significance-testing system.” Psychological Bulletin, vol. 85, no. 2, p. 410, 1978.
[5] J. Duda and G. Bhatta, “Gamma-ray blazar variability: new statistical models.” Monthly Notices of the Royal Astronomical Society, vol. 508, no. 1, pp. 1446–1458, 2021.
[6] J. M. Shapiro, “Embedded image coding using zero trees of wavelet coefficients.” IEEE Transactions on signal processing, vol. 41, no. 12, pp. 3445–3462, 1993.
[7] N. Ahmed, T. Natarajan, and K. R. Rao, “Discrete cosine transform.” IEEE transactions on Computers, vol. 100, no. 1, pp. 90–93, 1974.
[8] J. Duda, “Context binning, model clustering and adaptivity for data compression of genetic data,” arXiv preprint arXiv:2201.05028, 2022.