# Compression using Singular Value Decomposition

JajimoggaRaghavendar
Assoc Professor
NallaMalla Reddy Engineering College
Telangana India

Abstract:

The Singular Value Decomposition expresses image data in terms of number of eigen vectors depending upon the dimension of an image. The psycho visual redundancies in an image are used for compression. Thus an image can be compressed without affecting the image quality. This paper presents one such image compression technique called as SVD. Basic mathematics of SVD is dealt with in detail and results of applying SVD on an image are also discussed. The MSE and compression ratio are used as thresholding, parameters for reconstruction. SVD is applied on variety of images for experimentation. The work is concentrated to reduce the number of eigen values required to reconstruct an image.

## INTRODUCTION

One common way to represent datasets is as vectors in a feature space. For example, if we let each dimension be a movie, then we can represent users as points. Though we cannot visualize this in more than three dimensions, the idea works for any number of dimensions.

The multilayer perceptron, when working in auto-association mode, is sometimes considered as an interesting candidate to perform data compression or dimensionality reduction of the feature space in information processing applications. The present paper shows that, for auto-association, the nonlinearities of the hidden units are useless and that the optimal parameter values can be derived directly by purely linear techniques relying on singular value decomposition and low rank matrix approximation, similar in spirit to the well-known Karhunen-Loève transform. This approach appears thus as an efficient alternative to the general error back-propagation algorithm commonly used for training multilayer perceptrons. Moreover, it also gives a clear interpretation of the rôle of the different parameters.

One natural question to ask in this setting is whether or not it is possible to reduce the number of dimensions we need to represent the data. For example, if every user who likes *The Matrix* also likes *Star Wars*, then we can group them together to form an agglomerative movie or *feature*. We can then compare two users by looking at their ratings for different features rather than for individual movies.

There are several reasons we might want to do this. The first is scalability. If we have a dataset with 17,000 movies, than each user is a vector of 17,000 coordinates, and this makes storing and comparing users relatively slow and memory-intensive. It turns out, however, that using a smaller number of dimensions can actually improve prediction accuracy. For example, suppose we have two users who both like science fiction movies. If one user has rated *Star Wars* highly and the other has rated *Empire Strikes Back* highly, then it makes sense to say the users are similar. If we compare the users based on individual movies, however, only those movies that both users have rated will affect their similarity. This is an extreme example, but one can certainly imagine that there are various classes of movies that should be compared.

Singular Value Decomposition (SVD) has recently emerged as a new paradigm for processing different types of images. SVD is an attractive algebraic transform for image processing applications. The paper proposes an experimental survey for the SVD as an efficient transform in image processing applications. Despite the well-known fact that SVD offers attractive properties in imaging, the exploring of using its properties in various image applications is currently at its infancy. Since the SVD has many attractive properties have not been utilized, this paper contributes in using these generous properties in newly image applications and gives a highly recommendation for more research challenges. In this paper, the SVD properties for images are experimentally presented to be utilized in developing new SVD-based image processing applications. The paper offers survey on the developed SVD based image applications. The paper also proposes some new contributions that were originated from SVD properties analysis in different image processing. The aim of this paper is to provide a better understanding of the SVD in image processing and identify important various applications and open research directions in this increasingly important area; SVD based image processing in the future research.

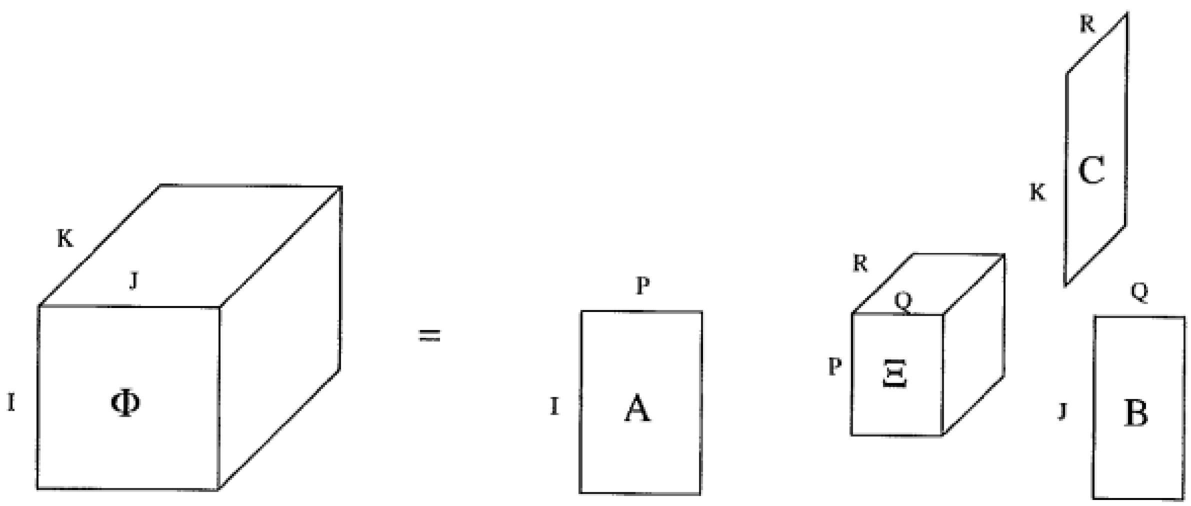


Figure 1: Third-order Singular Value Decomposition

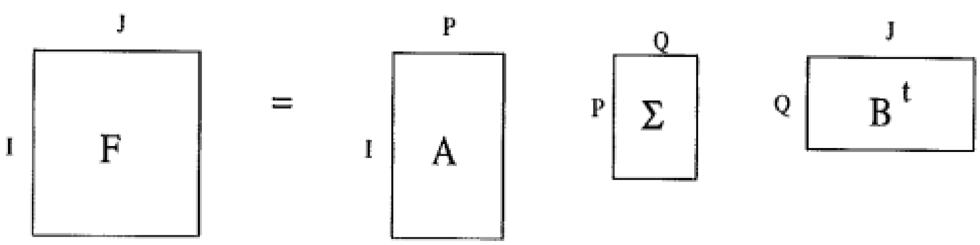


Fig:1 Second order singular value decomposition

THE SINGULAR VALUE DECOMPOSITION

In an important paper, Deerwester et al. examined the dimensionality reduction problem in the context of information retrieval [2]. They were trying to compare documents using the words they contained, and they proposed the idea of creating features representing multiple words and then comparing those. To accomplish this, they made use of a mathematical technique known as Singular Value Decomposition. More recently, Sarwar et al. made use of this technique for recommender systems [3].

The Singular Value Decomposition (SVD) is a well known matrix factorization technique that factors an $m$ by $n$ matrix **X** into three matrices as follows:

$$\underset{m \times n}{\overset{X}{\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & \\ \vdots & \vdots & \ddots & \\ x_{m1} & & & x_{mn} \end{pmatrix}}} = \underset{m \times r}{\overset{U}{\begin{pmatrix} u_{11} & \cdots & u_{1r} \\ \vdots & \ddots & \\ u_{m1} & & u_{mr} \end{pmatrix}}} \underset{r \times r}{\overset{S}{\begin{pmatrix} s_{11} & 0 & \cdots \\ 0 & \ddots & \\ \vdots & & s_{rr} \end{pmatrix}}} \underset{r \times n}{\overset{V^{\mathsf{T}}}{\begin{pmatrix} v_{11} & \cdots & v_{1n} \\ \vdots & \ddots & \\ v_{r1} & & v_{rn} \end{pmatrix}}}$$

The matrix **S** is a diagonal matrix containing the *singular values* of the matrix **X**. There are exactly *r* singular values, where *r* is the *rank* of **X**. The rank of a matrix is the number of linearly independent rows or columns in the matrix. Recall that two vectors are linearly independent if they can not be written as the sum or scalar multiple of any other vectors in the space. Observe that linear independence somehow captures the notion of a feature or agglomerative item that we are trying to get at. To return to our previous example, if every user who liked *Star Wars* also liked *The Matrix*, the two movie vectors would be linearly dependent and would only contribute one to the rank.

We can do more, however. We would really like to compare movies if most users who like one also like the other. To accomplish we can simply keep the first *k* singular values in **S**, where *k*<*r*. This will give us the best rank-*k* approximation to **X**, and thus has effectively reduced the dimensionality of our original space. Thus we have

Given that the SVD somehow reduces the dimensionality of our dataset and captures the "features" that we can use to compare users, how do we actually predict ratings? The first step is to represent the data set as a matrix where the users are rows, movies are columns, and the individual entries are specific ratings. In order to provide a baseline, we fill in all of the empty cells with the average rating for that movie and then compute the svd. Once we reduce the SVD to get X_hat, we can predict a rating by simply looking up the entry for the appropriate user/movie pair in the matrix X_hat. Further details can be found in [2,3]

UPDATING THE SVD

One of the challenges of using an SVD-based algorithm for recommender systems is the high cost of finding the singular value decomposition. Though it can be computed offline, finding the svd can still be computationally intractable for very large databases. To address this problem, a number of researchers have examined incremental techniques to update an existing svd without recomputing it from scratch [1,4]. We looked at a method proposed by Matthew Brand for adding and modifying users in an existing SVD [1]. Brand focuses on so-called *rank 1* updates, where a single column is modified or added to the orignalmatrix. Formally, given the singular value decomposition of a matrix **X**, we want to find the singular value decomposition of the matrix $\mathbf{X+ab}^T$, where **a** and **b** are column vectors.

The full derivation of Brand's method is beyond the scope of this document, but we will provide a brief discussion of the algorithm. Given the SVD $\mathbf{X=USV}^T$, let $\mathbf{m=U}^T\mathbf{a}$, $\mathbf{p=a\text{-}Um}$, $p$=sqrt($\mathbf{p}^T$p) and $\mathbf{P}=\mathbf{p}/p$. Similarly let $\mathbf{n=V}^T\mathbf{b}$, $\mathbf{p=b\text{-}Vn}$, $q$=sqrt($\mathbf{q}^T$q) and $\mathbf{Q}=\mathbf{q}/q$. Then we first find the singular value decomposition $\mathbf{U'S'V'}^T$ of the matrixThen the svd of our new matrix is given bySince we can use the low rank approximations of **U**, **S**, and **V**, this algorithm is quite fast, and Brand shows that the entire SVD can be built in this manner in $O(mnk)$ time, where $m$ and $n$ are the dimensions of the matrix and $k$ is the reduced rank of the approximation. It is observed that if the value of k chosen is equal to the rank of the image, the reconstructed image is closer to the original image. As the value of k decreases from the rank image quality degrades. Second observation is that as the compression ratio is high image quality is poorer and if compression ratio is low, image with superior quality can be reconstructed but with less compression.

Therefore compression ratio and image quality is required to select appropriately. Section I presents literature survey and brief outline of the paper. Section II presents the SVD computations on images. Section III presents the experimentation and the proposed procedure to estimate the percentage of eigenvectors required to reconstruct the original image using the SVD and PSNR. Section IV presents the results and the discussions. Finally Section V presents the conclusion

**PROPERTIES OF SVD**

Following are some of the important properties of SVD 1. The rank of matrix G is equal to the number of non-zero singular values. 2. GGT = LD2 LT and hence L daigonalizesthe matrix G and column vectors of L are the eigenvectors of GGT . 3. Similarly, GT G = RD2 RT , R diagonalizes G and hence eigenvectors of GT G are column vectors of R. 4. The singular values of matrix G i.e. σ1, σ2, σ3……….,σN are unique, however the matrices L and R are not unique.

Image compression involves reducing the redundant or irrelevant information in an image. Redundancies in an image may be in the form of 1. Psycho visual redundancy, which is due to the limitations of the human visual system to interpret very fine details in an image.(i.e. visually nonessential information) 2. Inter pixelredundancy, due to similarities in the neighboring pixels. 3. Coding redundancy, in which more number of bits than required are used to encode the image data for transmission.(i.e. less than optimal code words are used) Image compression techniques reduce the number of bits required to represent an image by taking advantage of these redundancies. Removing the redundancies is equivalent to reducing the number of bits required to represent an image without much compromise in the image quality. Different image compression techniques apply different methods or more appropriately coding algorithms to achieve this. Now let us see how this is achieved using SVD. By applying SVD on an image, the image matrix G is decomposed into 3 different matrices L, D and R. However, simply applying SVD on an image does not compress it. To compress an image, after applying SVD, only a few singular values are retained while other singular values are discarded. This follows from the fact that singular values are arranged in descending order on the diagonal of D and that first singular value contains the greatest amount of information and subsequent singular values contain decreasing amounts of image information. Thus, the lower singular values containing negligible or less important information can be discarded without significant image distortion. Furthermore, property 1 of SVD (section 3) says that "the number of non-zero singular values is equal to the rank of G." But even if the lower order singular values after the rank of the matrix are not zero, they have negligible values and are treated as noise

This implies that value of k should be smaller than m*n/(m+n+1) in order to compress any image thus putting an upper limit on the value of k. In short, value of k is chosen such that good amount of compression is achieved while image quality is maintained above the minimum acceptable limit. To compare the results of different compression techniques and also to measure the degree to which an image is compressed, many performance measures are available such as

Compression Ratio:

Compression Ratio is the ratio of the storage space required to store original image to that required to store a compressed image and is given by

Compression Ratio = m*n / (k*(m+n+1))

It measures the degree to which an image is compressed.

2. Mean Square Error(MSE): MSE is the measure of deterioration of image quality as compared to the original image when an image is compressed. It is defined as square of the difference between pixel value of original image and the corresponding pixel value of the compressed image averaged over the entire image. Mathematically

CONCLUSION

The idea of reducing the dimensionality of a dataset is not limited to the singular value decomposition. While SVDs provide one of the most theoretically grounded techniques for finding features, there are a number of approximation algorithms that can be used on very large datasets. We did not explore this area in great depth, but we did use a method proposed on the Netflix forums by simonfunk and implemented in C by timelydevelopment. We ported this code to Java and tried it with both the Netflix and Movielens datasets. The fact that this algorithm performed the best among all of those we tried suggests that dimensionality reduction is a powerful idea that would be worth exploring in the future.

REFERENCES

[1] M. Brand. Fast online svd revisions for lightweight recommender systems. In Proceedings of the 3rd SIAM International Conference on Data Mining, 2003.

[2] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman.Indexing by latent semantic analysis.Journal of the American Society for Information Science, 41(6), 1990.

[3] B.M. Sarwar, G. Karypis, J.A. Konstan, and J.Reidl.Application of dimensionality reduction in recommender system - a case study.In ACM WebKDD 2000 Web Mining for E-Commerce Workshop, 2000.

[4] B.M. Sarwar, G.Karypis, J.A. Konstan, and J. Reidl.Incremental singular value deocmposition algorithms for highly scalable recommender systems.In Proceedings of the Fifth International Conference on Computer and Information Technology (ICCIT), 2002.