Resolving the Sign Ambiguity in the Singular Value Decomposition

R. Bro, E. Acar, T. Kolda

Sandia National Laboratories

Resolving the Sign Ambiguity in the Singular Value Decomposition

Rasmus Bro
Dept. Food Science, Faculty of Life Sciences
University of Copenhagen
DK-1958 Frederiksberg C
rb@life.ku.d

Evrim Acar
Computer Science Department
Rensselaer Polytechnic Institute
Troy, New York USA

Tamara Kolda
Sandia National Laboratories
Livermore, California 94551-0969

Abstract

Many modern data analysis methods involve computing a matrix singular
value decomposition (SVD) or eigenvalue decomposition (EVD). Principal
components analysis is the time-honored example, but more recent
applications include latent semantic indexing, hypertext induced topic
selection (HITS), clustering, classification, etc. Though the SVD and EVD
are well-established and can be computed via state-of-the-art algorithms, it
is not commonly mentioned that there is an intrinsic sign indeterminacy
that can significantly impact the conclusions and interpretations drawn
from their results. Here we provide a solution to the sign ambiguity
problem and show how it leads to more sensible solutions.

This page intentionally left blank.

Table Of Contents

Figures

TABLES

Equations

This page intentionally left blank.

INTRODUCTION

The singular value decomposition (SVD) is of fundamental importance in a huge number of applications in various fields of data analysis including principal components analysis (PCA) [Jackson 1980, Jackson 1981], latent semantic indexing (LSI) [Dumais et al. 1988], hypertext induced topic selection (HITS) [Kleinberg 1999], clustering, classification, etc.. For a real-valued matrix, $\mathbf{X} \in \mathsf{R}^{I \times J}$ the singular value decomposition theorem states that there exists orthogonal matrices

$$\mathbf{U} = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_I\} \in \mathsf{R}^{I \times I} \text{ and } \mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_J\} \in \mathsf{R}^{J \times J}$$

Such that

Equation 1.

$$\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^{\mathrm{T}}$$

where $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_P)$ with $P = \min\{I, J\}$ and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_P \geq 0$. The diagonal entries of Σ are the *singular values* while the columns of $\mathbf{U}$ and $\mathbf{V}$ are, respectively, the *left and right singular vectors*. If these singular values are distinct, the decomposition is said to be unique [Kahaner et al. 1989, Schott 1997]. In many data analysis situations, it is most practical to work with the truncated form of the SVD where only the first $K < P$ singular values and vectors are used so that

$$\mathbf{X} \approx \mathbf{U}_K\Sigma_K\mathbf{V}_K^{\mathrm{T}} = \sum_{k=1}^{K} \sigma_k\mathbf{u}_k\mathbf{v}_k^{\mathrm{T}}$$

where

$$\mathbf{U}_K = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_K\} \in \mathsf{R}^{I \times K}, \mathbf{V}_K = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_K\} \in \mathsf{R}^{J \times K}, \text{ and } \Sigma_K = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_K)$$

This is no longer an exact decomposition of the matrix $\mathbf{X}$, but it is the best rank-K approximation in a least squares sense and is still unique if the singular values are distinct.

However, the decomposition is only unique up to a reflection of each set of singular vectors, because for any set of singular vectors, k, it holds that

Equation 2.

$$\sigma_k\mathbf{u}_k\mathbf{v}_k^{\mathrm{T}} = \sigma_k(-\mathbf{u}_k)(-\mathbf{v}_k)^{\mathrm{T}}.$$

Thus, the SVD itself provides no means for assessing the sign of each singular vector. In actual algorithmic implementations of SVD, this indeterminacy is inherited so that the individual singular vectors have an 'arbitrary' sign. The actual sign is determined as a by-product of the computations that are used to ensure numerical stability. This determination of sign is essentially the same as assigning the sign randomly and hence the sign has no meaningful interpretation in terms of the data that the decomposition represents.

Even though it makes no difference mathematically, the current arbitrariness in the sign convention has important and significant ramifications in a number of applications.

- For example, statistical evaluation of uncertainty through bootstrapping where many models are calculated on slightly altered data. Comparison across these different models can be problematic when the sign can switch.

- Another example is cross-validation. One of the methods for cross-validation is the classical Eastment & Krzanowski [1982] approach in which two PCA models are calculated on slightly different data and then combined. Even in the original approach, the sign indeterminacy was realized to be a problem, but only an ad hoc solution (the so called parity check) is used to circumvent the problem.

- In exploratory analysis through principal component analysis, the signs of the scores and loadings (corresponding to scaled singular vectors) often flip, e.g., upon removal of outliers. While this is trivial mathematically, it has consequences for less-experienced data analysts that are not aware of the arbitrariness of the sign of the solution.

Common to the above examples is the lack of a fundamental technique for determining the sign. Therefore ad hoc approaches are used such as setting the maximum element in a singular vector to be positive. Unfortunately, such approaches do not solve the problem from a data analytical or interpretational point of view.

Consider the following simple example:

$$\mathbf{X} = \begin{bmatrix} 4 & 22 & 3 & 5 \\ 1 & 5 & 1 & 1 \\ 11 & 69 & 10 & 14 \\ 11 & 69 & 10 & 14 \end{bmatrix}$$

MATLAB has two standard methods for computing the SVD. The 'svd' method uses the LAPACK DGESVD command which is based on the QR method (see [Anderson et al. 1999]). The 'svds' command, on the other hand, uses ARPACK which is based on using Lanczos iterations (see [Lehoucq et al. 1998]). These are simply two different methods but should ideally always produce the same results.

Using 'svd' to calculate the SVD of X, we get the following left singular vectors

$$\mathbf{U} = \begin{bmatrix} -.22 & -.97 & .07 & .00 \\ -.05 & -.06 & -1.00 & -.00 \\ -.69 & .16 & .03 & -.71 \\ -.69 & .16 & .03 & .71 \end{bmatrix}.$$

However, 'svds' flips the signs of the 1ˢᵗ three pairs of the singular vectors. Below the left singular vectors are shown (the right singular vectors have a corresponding sign switch).

$$\mathbf{U} = \begin{bmatrix} .22 & .97 & -.07 & .00 \\ .05 & .06 & 1.00 & -.00 \\ .69 & -.16 & -.03 & -.71 \\ .69 & -.16 & -.03 & .71 \end{bmatrix}.$$

Moreover, because the 'svds' Lanczos-based algorithm has a random component, the results may differ on repeated calculations for the exact same matrix.

Obviously, this ambiguity can pose problems especially for an unaware user. As a practical example consider the following simple dataset published in Time Magazine, January 1996 showing average consumption of liquor, wine and beer (L/yr) as well as life expectancy in years and heart disease rate per 100.000/yr (Table 1).

Table 1. Data from Time Magazine.

	Liquor L/year	Wine L/year	Beer L/yr	Life Exp year	HeartD 100.000/yr
France	2.5	63.5	40.1	78	61.1
Italy	0.9	58.0	25.1	78	94.1
Switzerland	1.7	46.0	65.0	78	106.4
Australia	1.2	15.7	102.1	78	173.0
Great Britain	1.5	12.2	100.0	77	199.7
United States	2.0	8.9	87.8	76	176.0
Russia	3.8	2.7	17.1	69	373.6
Czech Rep	1.0	1.7	140.0	73	283.7
Japan	2.1	1.0	55.0	79	34.7
Mexico	0.8	0.2	50.4	73	36.4

When performing an SVD on these clearly positive data, the first right singular vector is

$$\mathbf{u} = \begin{bmatrix} -.008 \\ -.068 \\ -.321 \\ -.320 \\ -.889 \end{bmatrix}.$$

As can be seen the singular vector is all-negative even though the data points are clearly pointing in the opposite direction. In fact, the current LAPACK-based implementation of SVD in MATLAB is such that the first singular vectors from an all-positive matrix will always have all-negative elements!

In the following, a convention will be developed that leads to a completely identified solution for SVD and a solution which is also meaningful in terms of the data being decomposed.

METHODOLOGY

Mathematically, there is no way to avoid the sign ambiguity of a multiplicative term such as the pair of singular vectors. Hence mathematics cannot guide the choice of the sign. However, data analysis is more than algebra. In order to identify the sign of a singular vector, it is suggested that it be similar to the sign of the majority of vectors it is representing. Geometrically, it should point in the same, not the opposite, direction as the points it is representing. In Figure 1 some examples are given of two-dimensional data and the corresponding first right singular vector as well as the sign-corrected vector.
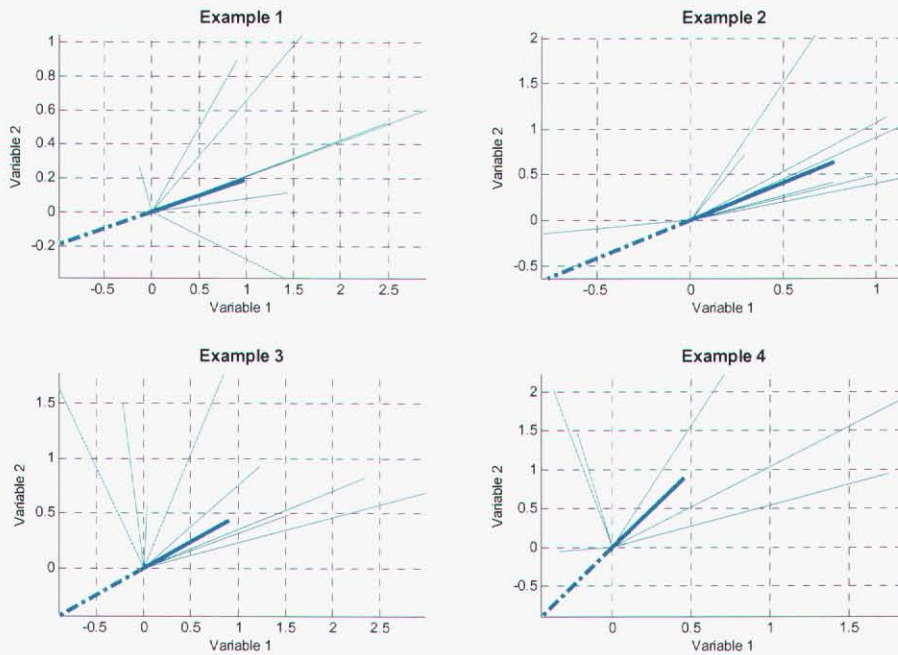
Figure 1. Four examples of random 10×2 matrices. The elements are drawn from a uniform distribution with a positive bias to make the direction obvious. Each row is shown as a thin line and the first right singular vector is shown in dashed thick. The sign-corrected right singular vector is shown as a thick solid line.

Another example is given in Figure 2 showing a 201-dimensional dataset of fluorescence spectra. The spectra are seen to have a common positive shape while the first singular vector has the opposite direction. A sign-corrected singular vector will point in the positive direction and hence reflect the common direction of the individual data points.
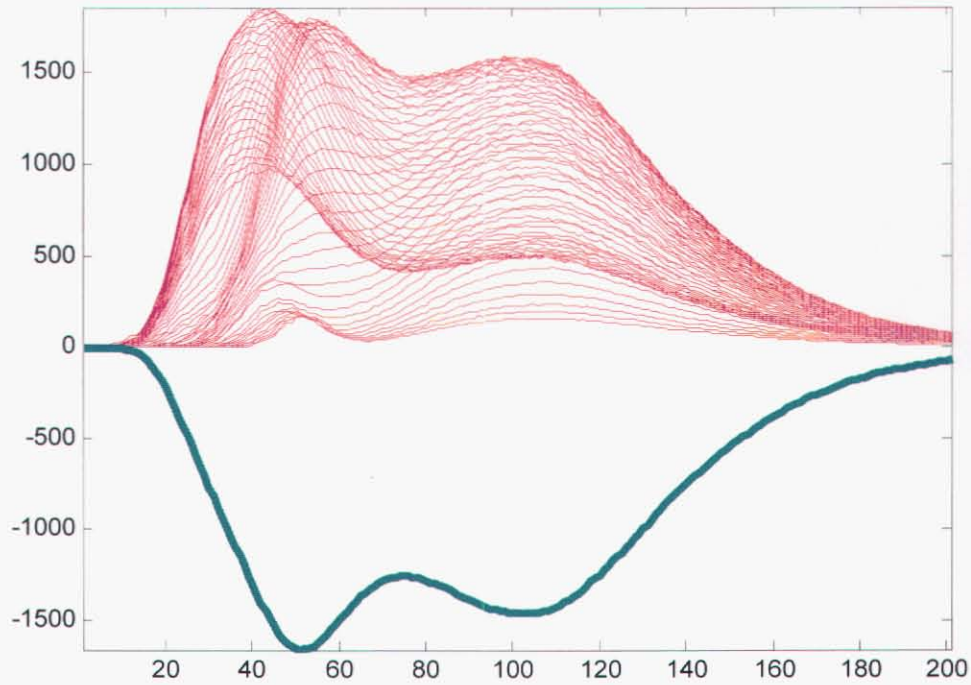
Figure 2. Example of a data matrix. Each thin line represents one row in a 61×201 matrix and the thick line the associated first right singular vector. In this case, the singular vector is said to have the wrong sign because it points in the opposite direction of the vectors it represents.

The sign suggested here can be determined from the sign of the inner product of the singular vector and the individual data vectors. The data vectors may have different orientation but it then makes intuitive as well as practical sense to choose the direction in which the majority of the vectors point. This can be found by assessing the sign of the sum of the signed inner products. Suppose, for instance, that we wish to determine the sign for the kth left singular vector, $\mathbf{u}_k$ Ideally, we will have

$$(\mathbf{u}_k)^{T}(\mathbf{x}_j) > 0 \text{ for } j = 1,\dots,J$$

if the singular vector is aligned correctly sign-wise with the column vectors, $\mathbf{x}_j$, of the data matrix $\mathbf{X} \in \square^{I \times J}$. Specifically, we choose the sign to maximize

$$s = \sum_{j=1}^{J} \text{sign}(\mathbf{u}_k^{T}\mathbf{x}_j)(\mathbf{u}_k^{T}\mathbf{x}_j)^2$$

where $\mathbf{x}_j$ is the jth column of the matrix $\mathbf{X}$. The sign of the corresponding right singular vector is determined similarly using the *rows* of the matrix $\mathbf{X}$ (or, think of it as the columns of $\mathbf{X}^T$). If the two optimal signs of the left and right singular vectors disagree when assessed this way, the overall sign can be determined by choosing the sign based on which of the left and right singular vector has the highest absolute summed value. This will help overcome ambiguity caused by one mode having close to arbitrary sign (e.g., because of centered data).

```
SignFlip Function

Input:  $\mathbf{X} \in \square^{I \times J}$ and its possibly truncated singular value decomposition $(\mathbf{U}, \mathbf{V}, \mathbf{S})$

Output:  $\mathbf{U}'$ and $\mathbf{V}'$ (left and right singular vectors with appropriate signs)

    (Step 1) for each left singular vector, $k=1, 2 \ldots K$ and for $\mathbf{y}_j$ being the $j$th column of $\mathbf{Y}$

            $$\mathbf{Y} = \mathbf{X} - \sum_{m=1, \, m \neq k}^{K} \sigma_m \mathbf{u}_m \mathbf{v}_m^T$$

            $$\text{Let } s_k^{left} = \sum_{j=1}^{J} \text{sign}(\mathbf{u}_k^T \mathbf{y}_j)(\mathbf{u}_k^T \mathbf{y}_j)^2$$

        endfor

    (Step 2) for each right singular vector, $k=1, 2 \ldots K$ and
            for $\mathbf{y}_i$ being the $i$th transposed row of $\mathbf{Y}$

            $$\mathbf{Y} = \mathbf{X} - \sum_{m=1, \, m \neq k}^{K} \sigma_m \mathbf{u}_m \mathbf{v}_m^T$$

            $$\text{Let } s_k^{right} = \sum_{i=1}^{I} \text{sign}(\mathbf{v}_k^T \mathbf{y}_i)(\mathbf{v}_k^T \mathbf{y}_i)^2$$

        endfor

    (Step 3) for each singular vector, $k=1, 2 \ldots K$
            if $(s_k^{left})(s_k^{right}) < 0$ then
                if $s_k^{left} < s_k^{right}$ then
                    $s_k^{left} = -s_k^{left}$
                else
                    $s_k^{right} = -s_k^{right}$
                endif
            endif
            $\mathbf{u}'_k = \text{sign}(s_k^{left})\mathbf{u}_k$
            $\mathbf{v}'_k = \text{sign}(s_k^{right})\mathbf{v}_k$
        endfor
```
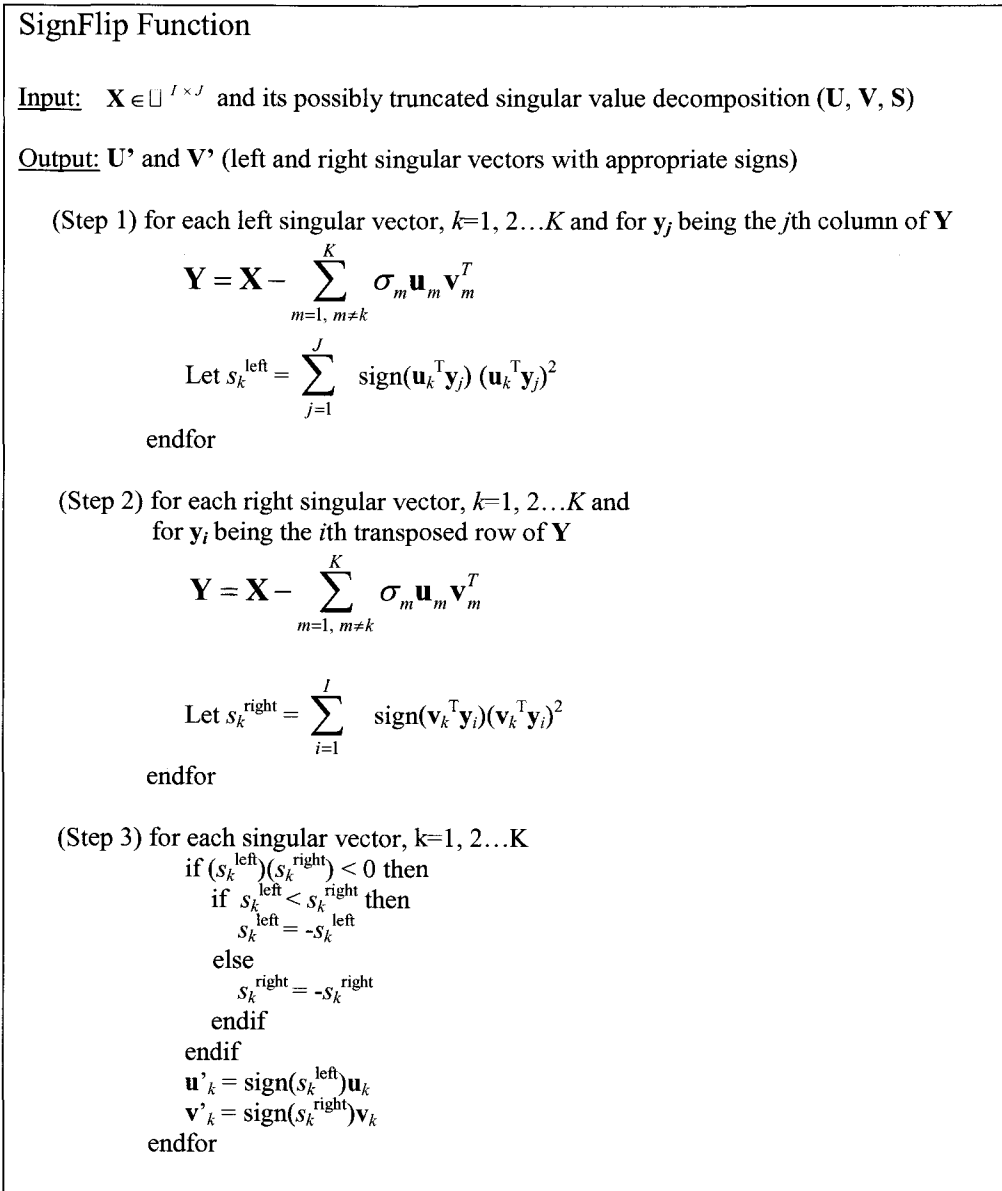
Figure 3.   Algorithm for determining the signs of singular vectors.

The detailed algorithm is given in Figure 3 which also includes a subtraction of additional components before determining the sign of a given component. This is not necessary in standard SVD but it is useful if the components are correlated which can be the case in alternative bilinear models such as multivariate curve resolution or partial least squares regression. The algorithm can be expected to work when the magnitude of the inner products are not close to zero. When the magnitudes come close to zero, then the sign will become arbitrary, essentially because the vectors point equally much in all directions. This is partially remedied in the algorithm by considering the combined magnitude of both the left and the right singular vectors, but of course, in the extreme, the sign will be arbitrary.

13

EXAMPLES

Several examples on the use of the sign convention are provided in the following to illustrate its usefulness.

The Effect of Sign Ambiguity on Eigenfaces

An example for the illustration of sign ambiguity in the SVD is a well-known technique called Eigenfaces [Turk & Pentland 1991], often used in face recognition. The underlying idea behind Eigenfaces is to represent a set of face images arranged as a matrix using the significant eigenvectors of the pixelwise covariance matrix of the image dataset. Let $X \in \square^{I \times J}$ represent an image dataset, where J is the number of images and I is the number of pixels per image. We can reduce the dimensionality of the data and represent the image dataset by using the significant eigenvectors of XX^T; in other words, the significant left singular vectors of the original matrix X. Consequently, also in the applications of Eigenfaces, singular vectors may flip sign due to the intrinsic sign ambiguity of SVD.

In order to illustrate the effect of sign flip on Eigenfaces, we compute the Eigenfaces of an image dataset containing 265 images of 10 subjects in different poses from the UMIST Face Database (currently The Sheffield Face Database) [Graham & Allinson 1998]. Figure 4 shows the Eigenfaces corresponding to the first three left singular vectors of the image dataset obtained from two different runs of the 'svd' method in MATLAB when 200 images out of 265 images are randomly selected at each run. We observe that as a result of the sign ambiguity in SVD, we obtain the photographic negative for the second and occasionally the third eigenface on different runs. On the other hand, when our sign flip approach is used, we consistently obtain the Eigenfaces given in Figure 5 which are seen to be positive rather than photographic negative.
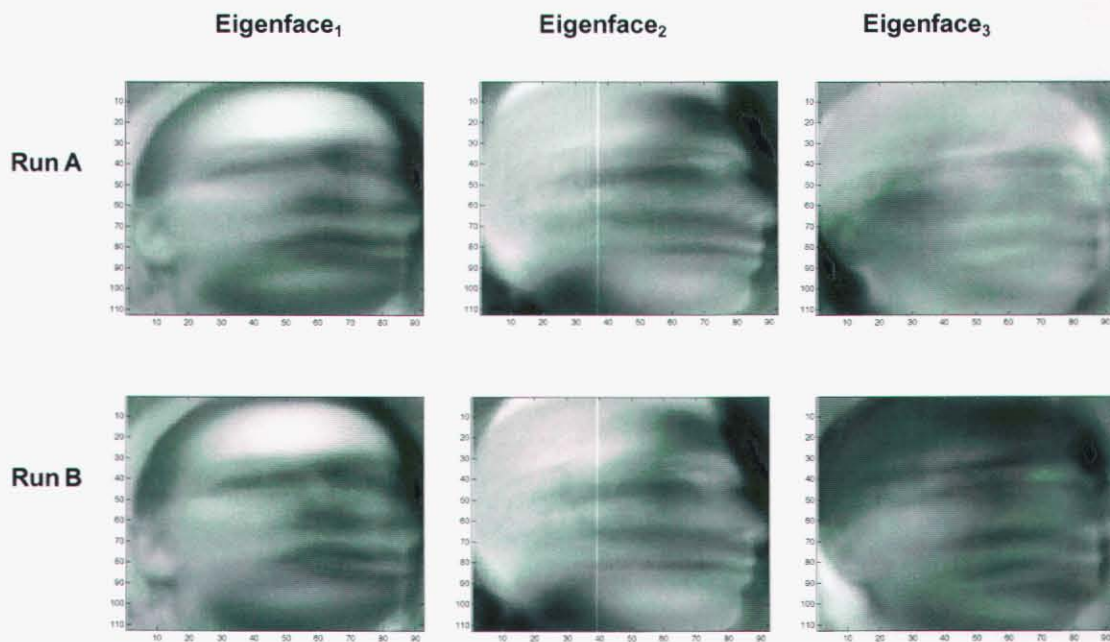

Figure 4. Eigenfaces corresponding to the first three left singular vectors obtained at different runs of the 'svd' method in MATLAB when 200 out of 265 images are randomly sampled at each run.
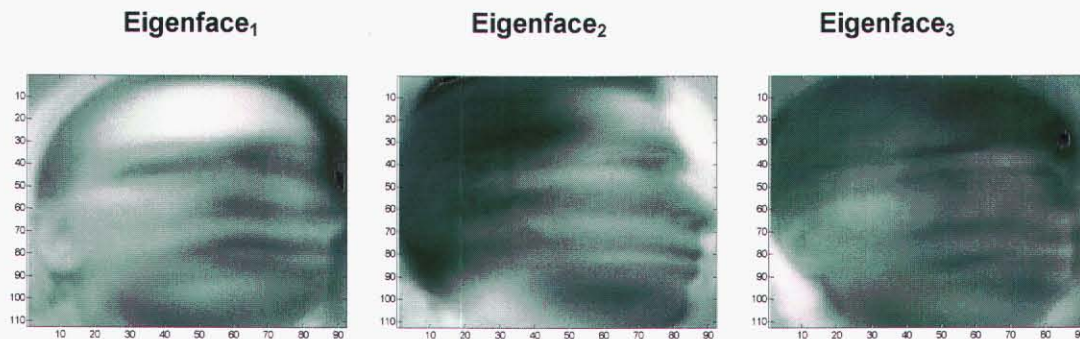
14

| Eigenface₁ | Eigenface₂ | Eigenface₃ |

Figure 5. Eigenfaces corresponding to the first three left singular vectors obtained consistently at different runs with SignFlip function when 200 out of 265 images are randomly sampled at each run.

The Effect of Sign Ambiguity on Spectral Data

A set of fluorescence emission spectra each of dimension 201 is given for 61 different excitation wavelengths and held in 61×201 matrix Figure 6. These spectra represent three underlying spectral components and hence the three largest singular components should represent the systematic variation in the data which is indeed found to be the case.
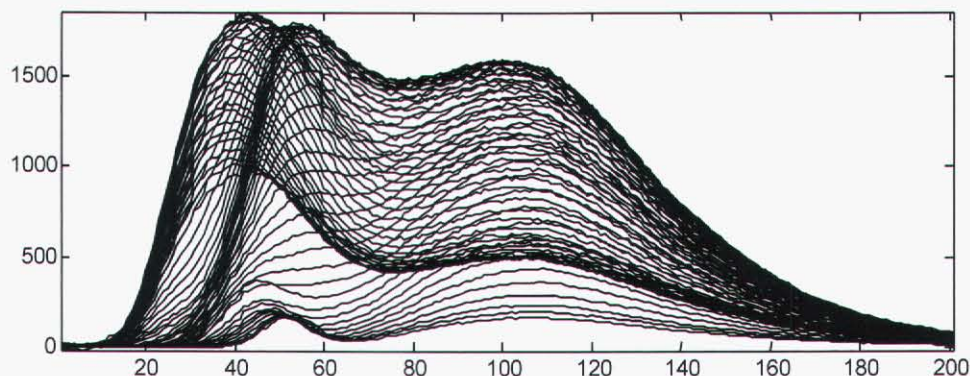

Figure 6. Sixty-one 201-dimensional fluorescence emission spectra.

In an experiment, these three components are bootstrapped 100 times in order to be able to evaluate the uncertainty of the estimated components. The bootstrapping is done by sampling 61 rows with replacement 100 times, and the results are shown in the upper half of Figure 7.

While the sign-flipping may be due to the bootstrapping it is also likely to be due to the semi-random nature of the sign of the singular vectors. In the lower half of Figure 7, the result of applying the proposed sign convention is shown and as can been seen, all singular vectors can now be immediately compared because their signs do not change as long as they represent similar aspects of the data.
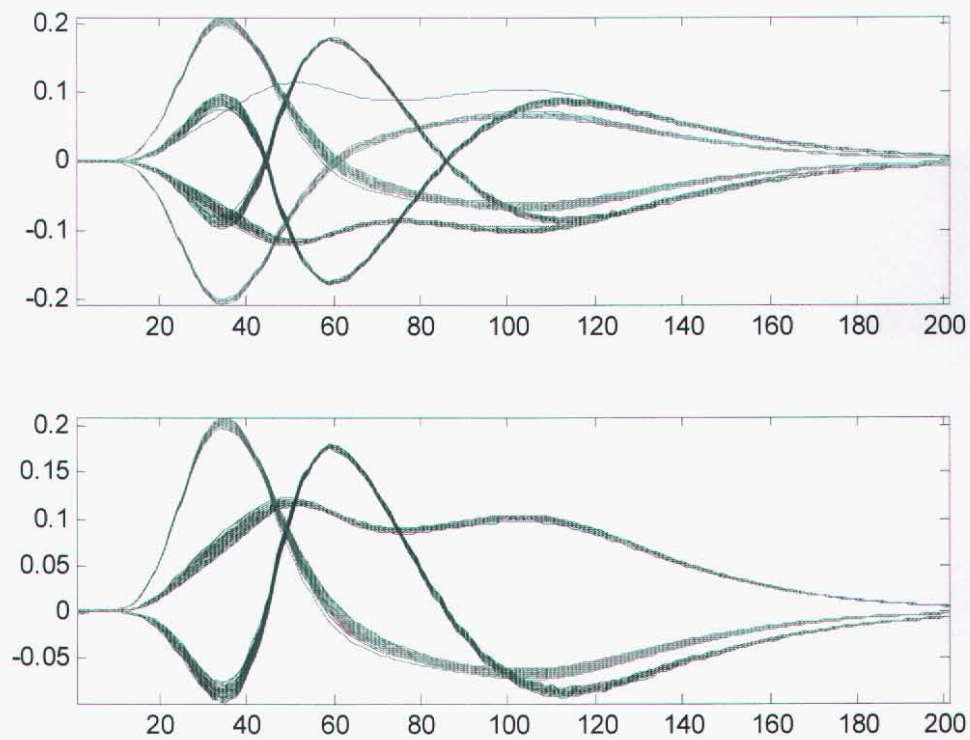
15

Figure 7. Bootstrapped three first right singular vectors from Figure 6 before (top) and after (bottom) sign correction.

Conclusion

A rule has been developed for applying meaningful signs to singular vectors. This rule makes it possible, for the first time to obtain unique parameters from the SVD that are meaningful from a data representation and interpretation point of view. This has implications for all situations where SVD is used as the computational engine for data analysis.

REFERENCE

1. Anderson E, Bai Z, Bischof C, Blackford S, Demmel J, Dongarra J, Du Croz J, Greenbaum A, Hammarling S, McKenney A, Sorensen D, LAPACK User's Guide.1999, **Third**, SIAM.

2. Dumais ST, Furnas GW, Landauer TK, Deerwester S, Harshman RA, Using latent semantic analysis to improve access to textual information, *CHI '88*, ACM Press, 1988, 281-285.

3. Eastment HT, Krzanowski WJ, Cross-validatory choice of the number of components from a principal component analysis, *Technometrics,* 1982, **24**, 73-77.

4. Graham DB, Allinson NM, Characterizing Virtual Eigensignatures for General Purpose Face Recognition,.1998, 446-456.

5. Jackson JE, Principal components and factor analysis: part I - principal components, *Journal of Quality Technology,* 1980, **12**, 201-213.

6. Jackson JE, Principal components and factor analysis: part II - additional topics related to principal components, *Journal of Quality Technology,* 1981, **13**, 46-58.

7. Kahaner D, Moler C, Nash S, *Numerical Methods and Software.* Prentice Hall, Upper Saddle River, NJ, USA, 1989,

8. Kleinberg JM, Authoritative sources in a hyperlinked environment, *Journal of the Acm,* 1999, **46**, 604-632.

9. Lehoucq RB, Sorensen DC, Yang C, ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods.1998, SIAM publications.

10. Schott JR, *Matrix analysis for statistics.* Wiley and Sons, New York, 1997,

11. Turk M, Pentland A, Eigenfaces for recognition, *Journal of Cognitive Neuroscience,* 1991, **3**, 72-86.

DISTRIBUTION

1	MS 9159	Heidi Ammerlahn	8962
5	MS 9159	Tamara G. Kolda	8962
2	MS9018	Central Technical Files	8944
2	MS0899	Technical Library	4536