Introduction to Non-negative Matrix Factorization

George Luta[1], Fajwel Fogel[2], Douglas A. Marsteller[3], Joe Maisog[4], S. Stanley Young[5]

[1]Georgetown University, Washington, DC 20057
[2]Ecole Polytechnique ParisTech, Paris, France
[3]PepsiCo, New Haven, CT 06511
[4]Glotech, Inc., Rockville, MD 20850
[5]National Institute of Statistical Sciences, Research Triangle Park, NC 27709

Abstract

In statistical practice, rectangular tables of numeric data are commonplace, and are often analyzed using dimension reduction methods like the singular value decomposition (SVD) and its close cousin, principal component analysis (PCA). This analysis produces score and loading matrices representing the rows and the columns of the original table and these matrices may be used for both prediction purposes and to gain structural understanding of the data. In some situations, the data entries are necessarily non-negative and so the matrix factors meant to represent them should arguably also contain only non-negative elements. This thinking, and the desire for parsimony, underlies such techniques as rotating factors in a search for "simple structure." The recent development of non-negative matrix factorization, or NMF, is an attractive alternative. Rather than attempt to transform a loading or score matrix of mixed signs into one with only non-negative elements, it directly seeks matrix factors containing only non-negative elements. The resulting factorization often leads to substantial improvements in interpretability of the factors. We illustrate this potential by using synthetic examples.

KEY WORDS: Principal component analysis, PCA, Singular value decomposition, SVD, Non-negative matrix factorization, NMF, latent dimensions.

1. INTRODUCTION

Rectangular tables of numeric data are wide-spread in statistical practice – for example in psychometrics where n subjects are scored on p items in a test; in microarrays where n tissues are tested with p probes; in the geosciences where p constituents are measured in n strata. Each of these settings gives rise to an $n \times p$ data matrix $\mathbf{X}$. Whereas in the past, p was typically small, many emerging areas give rise to data matrices where n and/or p may be in the thousands or tens of thousands, challenging traditional multivariate analysis approaches.

A lower-rank matrix approximation data matrix $\mathbf{X}$ is

$$\mathbf{X} = \mathbf{L}\mathbf{R}^{\mathbf{T}} + \mathbf{E}$$

where the left matrix $\mathbf{L}$ has n rows and k columns, the right matrix $\mathbf{R}$ has p rows and k columns, and $\mathbf{E}$ is a matrix of "errors". Superscript T indicates the transpose of a matrix. Each row of $\mathbf{L}$ represents one "case"; each row of $\mathbf{R}$ represents one "variable" and the k columns of both $\mathbf{L}$ and $\mathbf{R}$ represent k underlying latent variables or dimensions that relate

the rows and the columns of **X.** The hope of this approximation is that a *k* value much smaller than *n* and *p* will nevertheless be enough to give a small **E** and so to capture nearly all the structure in **X**. Then the data matrix **X** itself can be discarded, and interpretation focused on **L** to explicate relationships between the cases, and on **R** to explicate those between the variables.

The most familiar way of getting a lower-rank approximation **(Greenacre and Underhill 1982)** is through the Singular Value Decomposition, or SVD, and its close cousin Principal Component Analysis, or PCA. Starting with the exact spectral decomposition of **X**

$$\mathbf{X} = \mathbf{A}\,\mathbf{\Lambda}\,\mathbf{B}^{\mathrm{T}}$$

where **A** are the row singular vectors, **B** the column singular vectors, and $\mathbf{\Lambda}$ a diagonal matrix of the singular values, defining

$$\mathbf{L} = \mathbf{A}\mathbf{\Lambda}^{r}; \quad \mathbf{R} = \mathbf{B}\mathbf{\Lambda}^{1-r}$$

for any selected *r* gives an exact representation $\mathbf{X} = \mathbf{L}\mathbf{R}^{\mathrm{T}}$.

The eigenvalues of principal component analysis are the squares of the singular values in $\mathbf{\Lambda}$ and PCA's eigenvectors are the column singular vectors. Retaining just the first *k* columns of **L** and **R** of the exact spectral decomposition then gives a lower rank approximation to **X**. This approximation is optimal in a least squares sense– there is no other approximation using *k* latent variables that more accurately represents **X** as measured by sum of squared deviations**.**

Factor analysis (FA), a related method, also relies on the equation $\mathbf{X} = \mathbf{L}\mathbf{R}^{\mathbf{T}} + \mathbf{E}$. However, unlike the setting with the SVD, the **E** matrix of FA is modeled as having independent normal elements, conceptually making FA a quite different methodology from the SVD, despite their superficial similarity.

The representation of the approximation given by the SVD is not unique. If we retain the first *k* columns of **L** and **R** and let **C** be any non-singular $k \times k$ matrix, replacing **L** by $\mathbf{L}^{*} = \mathbf{L}\mathbf{C}$ and **R** by $\mathbf{R}^{*} = \mathbf{R}(\mathbf{C}^{\mathrm{T}})^{-1}$ or more compactly $\mathbf{R}^{*} = \mathbf{R}\mathbf{C}^{-\mathrm{T}}$ gives an approximation

$$\mathbf{X} \approx \mathbf{L}^{*}\mathbf{R}^{*\mathbf{T}} = (\mathbf{L}\mathbf{C})\,(\mathbf{R}\mathbf{C}^{-\mathrm{T}}) = \mathbf{L}\mathbf{R}^{\mathbf{T}}$$

This factorization gives identical approximations to all elements of **X**, but uses left and right factors $\mathbf{L}^{*}$ and $\mathbf{R}^{*}$ that may look very different from **L** and **R**. This fact underlies the variety of methods used for example in factor analysis to rotate a hard-to-interpret loading matrix to one whose elements are easier to interpret, as summarized in the mantra of "simple structure."

When performing a PCA or a SVD, it is common to center the columns of **X** by subtracting a column (or sometimes a global) mean from each element of **X**, however this centering is not central to the methods. Frequently, leaving the data uncentered and adding one latent dimension to the fit to accommodate the mean leads to essentially the same structure as is obtained from the centered data, as will be illustrated later in this paper.

In some settings **X** consists of non-negative elements, apart, perhaps, from a few negative elements resulting from measurement error. For example intensities in microarrays, chemical compositions in geology and biological chemistry, and test scores in psychometrics are usually non-negative. When this is the case, it is generally desirable to use matrix approximations

$$\mathbf{X} \approx \mathbf{L}\mathbf{R}^{\mathbf{T}}$$

whose **L** and **R** factors are also comprised of non-negative numbers. This will greatly simplify interpretation, since negative values in **L** and **R** are hard to make sense of. In psychometric testing, for example, a negative loading in the R matrix would imply that item was negatively associated with the latent aptitude being tested which, unless the item was designed with a reversed scale, would make no sense. But the SVD approximation does not give non-negative factors **L** and **R** – the orthogonality of the successive singular vectors makes this impossible except in the degenerate case that the singular vectors are some permutation of the identity matrix. This challenge leads to the rotation methods that take a loading matrix with mixed signs and attempt to find a non-singular transformation that will give an equally explanatory matrix consisting of more interpretable numbers – for example with few or no negative elements. But this post-processing of a SVD approximation to something even approximating non-negativity is a daunting task.

Note that if **X** is non-negative, mean-centering will create a matrix of mixed sign, and so mean centering is not performed when one is interested in non-negative factorization. An alternative method that will remove column effects but not destroy the non-negativity is to subtract the column minimum from all elements of each column. However this approach is not widely used.

Focusing on the issue of identifying the underlying structure, suppose that a matrix **X** that is non-negative, except perhaps for some random noise, is in fact generated by two non-negative *k*-column matrices:

$$\mathbf{X} = \mathbf{L}\mathbf{R}^{\mathrm{T}} + \mathbf{E}$$

To uncover the generating mechanism, we would like to recover **L** and **R**, the generating vector pairs. The fact that the SVD is a minimum-variance approximation of any given rank is a mixed blessing, as it can lead to multiple mechanisms being conflated in a single latent variable. Suppose that we are looking at a data matrix of subjects and their gene expression data and we have a diagnosis for each person: diseased or not. Now, we are given one named disease, but there may be multiple etiologies that lead to the same symptoms and hence the one named disease/condition; metabolic syndrome is a good current example. It is quite possible that any of several etiologies might lead to the same clinical picture, for instance, overweight especially in the upper body, insulin resistance, metabolic abnormalities, clinical failure, or other clinical indications. SVD will tend to conflate the different etiologies, putting all of them into the first component with positive coefficients, and then differentiating them in subsequent components by giving one etiology positive coefficients and another one negative coefficients. While the true nature of the different etiologies can in principle be recovered by rotating the different components to separate them out, this can be an uncertain and tedious operation.

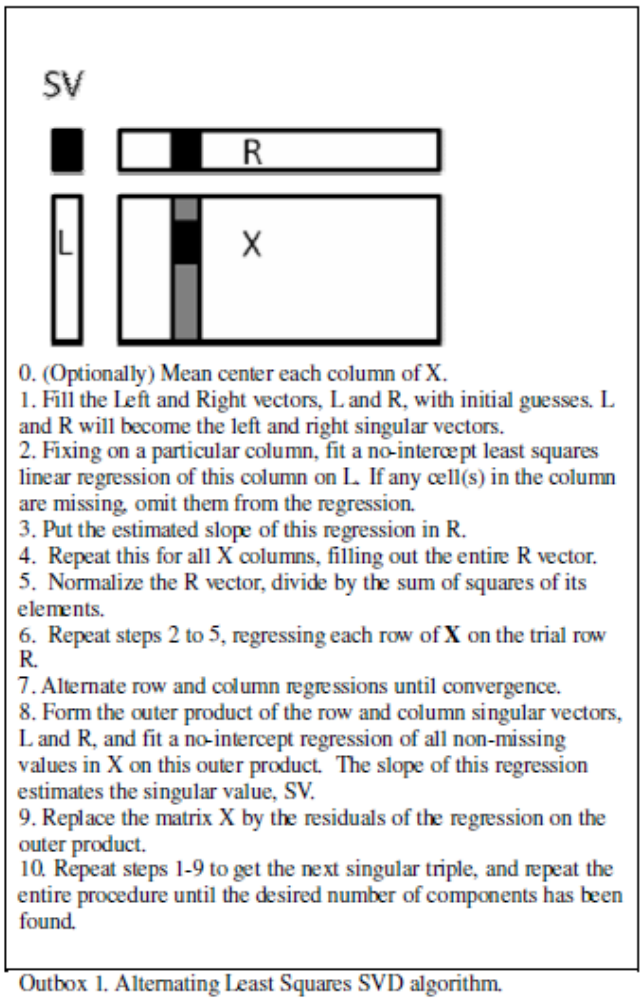
Non-negative matrix factorization, (**Lee and Seung 1999)**, where the elements of the factoring matrices are also non-negative, addresses this problem head-on. Unlike the SVD, NMF is not able to conflate the mechanisms in different components with mixed signs, and will instead tend to identify each of the syndromes with its own component, making interpretation transparent.

Not all data matrices are conceptually generated by the product of non-negative matrices plus random noise. We have mentioned many examples where this is the case, but there are many others where the signs in **X** are genuinely mixed; when this is the case, NMF would not be a good choice for factorization and could lead to seriously misleading results.

Over the years, there has been accumulating evidence from many different fields that NMF is capable of finding parts - see the NMF review paper of **Devarajan (2008).** All this evidence raises a question of why and when NMF is much better at finding parts than, say, principal components analysis.

The paper by **Donoho and Stodden (2003)** (D&S) considers the "why" question, producing a set of rules describing “Separable Factorial Articulation Families”, and shows that if these rules are satisfied there is a unique exact non-negative factorization. These are sufficient, but not necessary conditions.

An additional issue is that D&S’s sufficient conditions relate to the underlying true generating model, whereas all an analyst typically has is a data matrix, whose underlying generating model is unknown, and for whom the sufficient conditions can not be checked. This uncertainty may be tolerable for the data analyst who can fit the NMF model and decide whether it makes sense in subject-matter terms, but it is an issue one would like to see resolved.



Outbox 1. Alternating Least Squares SVD algorithm.

In this paper, we provide some clean synthetic examples and counter-examples that contrast SVD and NMF. Finally we make some summary points in a discussion and point out some intriguing open questions about NMF.

2. METHODS

2.1 Singular Value Decomposition via Alternating Least Squares (ALS)

Good (1969) points out that the SVD of a matrix is central to the computing of many statistical methods. He gives the alternating least squares algorithm for computing a SVD in a few sentences. See **Outbox 1**. Modifications to ALS by **Gabriel and Zamir (1979)** and by **Liu et al. (2003)** can be used to compute an analog of SVD that is both robust to outliers and accommodates missing information. Common practice in fitting the SVD is to center the columns by subtracting their means. This is not an essential feature of the SVD, and so to better highlight the similarities and differences of the SVD and NMF, we will not center the data in what follows.

The ALS algorithm is a key to understanding both how to obtain a good approximation to a given data matrix and to why, in the case of SVD, that the approximation can lead to confusion. In addition to mixing generating vectors, which restricts interpretability, SVD can pull in noise as if it were a feature (**Faber *et al.*, 1995**). This is in addition to the interpretability problems of having a possibly user-unfriendly basis in a *k*-dimension space that can be transformed in many ways.

The mean centering is described as optional. If omitted, the first singular triplet (left and right singular vector and singular value) tends to accommodate the data location and then the more interesting structure emerges from the second singular triplet on, so operationally, centering tends to remove one component which captures the general mean. Due to the orthogonality issue, SVD produces factors that include both positive and negative elements.

For a given rank k, we start by initializing the elements of the nonnegative matrices $\mathbf{L}$ and $\mathbf{R}$: $\mathbf{L}_{ia} > 0$, $\mathbf{R}_{aj} > 0$, $\forall\, i, j, a$, where i, j and a are the row, column and component indexes.

Then we apply the following multiplicative update rules until the difference between two iterations is small:

$$\mathbf{L}_{ia} \leftarrow \mathbf{L}_{ia} \, (\mathbf{XR})_{ia} \,/\, (\mathbf{LR}^{\mathrm{T}}\mathbf{R})_{ia} \,, \; \forall\, i, a$$

$$\mathbf{R}_{ja} \leftarrow \mathbf{R}_{ja} \, (\mathbf{L}^{\mathrm{T}}\mathbf{X})_{aj} \,/\, (\mathbf{L}^{\mathrm{T}}\mathbf{LR}^{\mathrm{T}})_{aj} \,, \; \forall\, j, a$$

At each iteration we update the current elements of $\mathbf{L}$ and $\mathbf{R}$ using specific multiplicative factors that relate to the current quality of the intended approximation, see Lin (2005) for further details regarding the properties of this algorithm and extensions using projected gradient methods.

Outbox 2. Multiplicative Update Rules, NMF algorithm.

2.2 Non-negative matrix factorization

The ALS algorithm for the SVD described above finds the latent dimensions sequentially, one at a time. **Lee and Seung (1999, 2001)** describe an algorithm for NMF based on multiplicative update rules, but fitting all *k* terms of the factorization at the same time. The algorithm does not require that all elements of the data matrix **X** be non-negative, and so can accommodate the situation that the original matrix **X** contains a few negative elements. As long as these negatives are few and moderate, the factor matrices will contain only non-negative elements. See **Outbox** 2. Suggestions for the necessary initial values are given later.

3. SYNTHETIC EXAMPLES

In all of the synthetic examples to follow, the data matrices were constructed by matrix multiplication of left and right generating vectors, $\mathbf{X=LR^T}$. The left vectors are termed weighting vectors, and the right vectors are called spectral vectors. The elements of both generating matrices **L** and **R** are either zero or uniformly distributed, U(1,2). A small, normally-distributed noise with mean 0 and standard deviation 0.1 was added to each cell. The generating vectors and the data matrices are presented visually as heatmaps. Our goal will be to retrieve the generating vectors of the synthesized matrices. Thus, for all synthetic examples we will compute SVD and NMF on the original, uncentered data matrices with their known rank. The choice of the factorization rank, which in real situations is unknown, will be discussed briefly in Section 4.

3.1 Orthogonal generating vectors

There are four generating left and right vectors. The vectors are orthogonal (**Figure 1**). This is as simple a situation as imaginable, so matrix factorization should return the generating vectors.

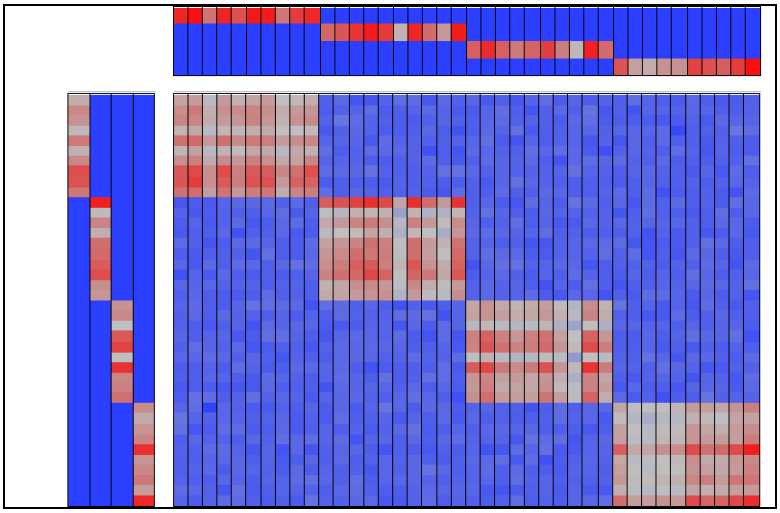


Figure 1: Simple synthetic example with four generating left and right vectors. The vectors are orthogonal.

We computed the SVD and NMF on this synthetic dataset. SVD recovered four vector pairs with singular values of 25.2, 24.7, 23.3 and 23.1. We also computed the NMF with four vector pairs. For this fully orthogonal example, both SVD and NMF (data not shown) recover the correct left and right generating vectors.

3.2 Realistic synthetic mixture

We now turn to another synthetic but more realistic example. There are two right and left generating vectors, but there are four kinds of individuals (**Figure 2**). Group 1 individuals are normal controls. Groups 2, 3 and 4 are diseased, but although there is but one named disease, say diabetes, there are two different etiologies, E1 and E2. Groups 2 and 3 suffer from one of these two “pure” etiologies. The unfortunate people in Group 4 suffer from both etiologies. Finally, there are a number of genes that do not participate in either etiology. These non-participatory columns add a “real” non-participatory noise component. There are two right generating vectors, each having ten active genes, one set for etiology 1 and the other for etiology 2. There are 20 inactive genes. A small normal noise with mean 0 and standard deviation 0.1 was added to each cell in the table.

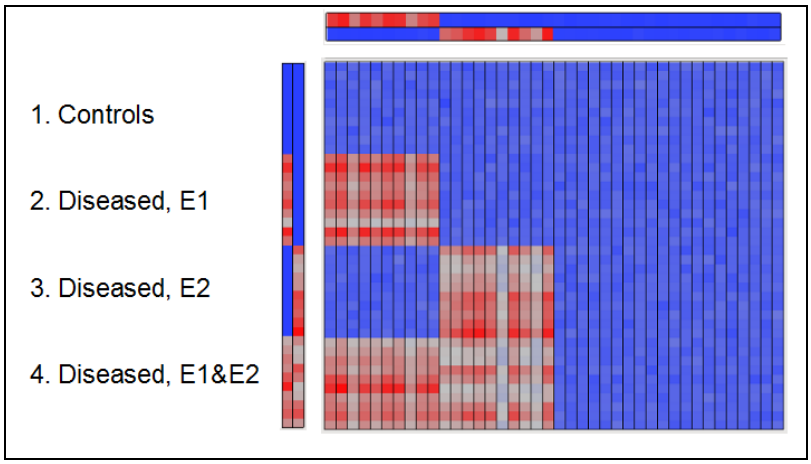


Figure 2: More realistic synthetic example. Two generating vectors lead to four kinds of individuals - normal controls and three disease groups with two etiologies.

Although synthetic, this model is based on a real situation. The Pima Indians in Arizona have a very high incidence of diabetes (**Baier and Hanson 2004).** The genetics is complicated and it is still not well understood, but it is believed to involve multiple mechanisms. Furthermore, it is believed that some people have more than one mechanism; therefore, that situation would correspond to this synthetic example.

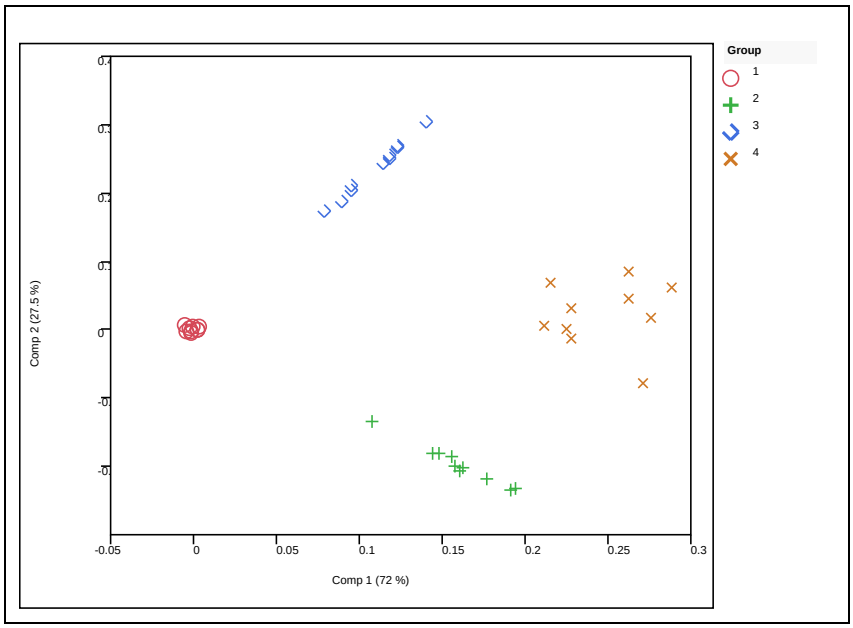


Figure 3: Two dimensions capture nearly all variance. Score plot shows four groups, but is oriented unhelpfully. “0” are disease free; “+” are Etiology 1; “◊” are Etiology 2; “x” have both diseases.

3.2.1 SVD analysis

How well does SVD treat this data set? Two vector pairs capture nearly all of the variance in the matrix, 99.5%, with singular values of 45 and 27.8.

The score plot clearly shows the four groups, but it has oriented the groups along a non-diseased/diseased principal axis. Component 2 contrasts the two etiologies, but does not provide useful information about how they differ (**Figure 3**).

It is a simple fact that the two generating vector pairs are equally important, yet SVD has very unequal singular values. SVD's first component (**Figure 4)** simply contrasts all affected and non-affected samples with no clues to the deeper structure of the data. Its second component does provide the additional insight on the structure of the data, but only by rotating the two components is there a clear picture of the two mechanisms.

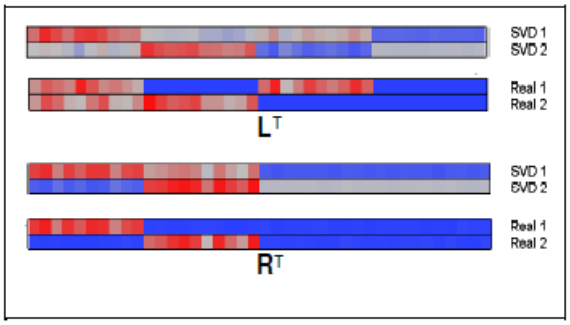


Figure 4: Heatmaps of SVD left and right singular vectors. Generating vectors are not recovered.

Plots of the scores, elements of the left factoring vectors, correctly show four distinct groups, however, assessing the nature of the data set is complicated. Singular vector pair 1 contrasts diseased versus non-diseased and singular vector pair 2 seems to contrast one etiology with the other, Groups 2 and 3, but the contrast is, at best, not clear.

3.2.2 NMF analysis

NMF analysis has no such difficulties of interpretation. The left vectors clearly show that there are four types of people. The right vectors show that there are two etiologies (**Figure 5**). This exactly matches the generating mechanism. Regarding variance explained, SVD is a least squares method and gives the minimum residual variance possible from any *k* component approximation. NMF, as we are using it here, is also a least-squares method, but with non-negativity constraints, and so its residual variance is necessarily at least as large as that of a SVD with the same number of components. In this data set, NMF recovers 99.5% of the variance, almost as high as the SVD and providing an indication that the non-negativity constrained NMF solution is an adequate fit to the data.

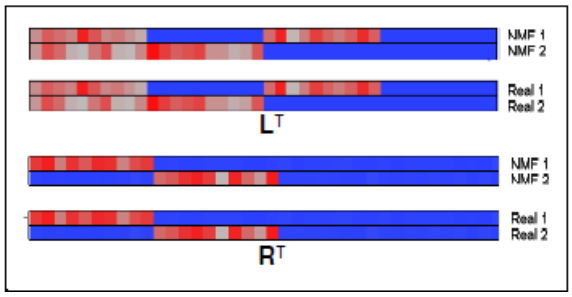


Figure 5: Heatmaps of the left and right vectors by NMF, showing perfect agreement with the real generating vectors.

The NMF score plot **(Figure 6)** gives a very satisfying 2x2 factorial layout. Controls are at the origin, the single etiologies are on each axis and the double etiologies are just where they should be.

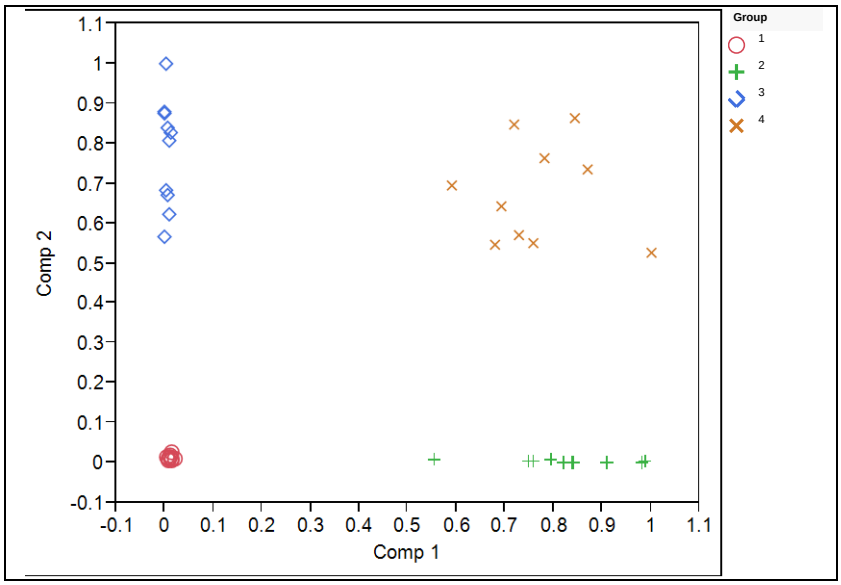


Figure 6: NMF score plot, showing 2x2 factorial layout with controls at the origin, single etiologies along each axis, and double etiologies loading on both axes.

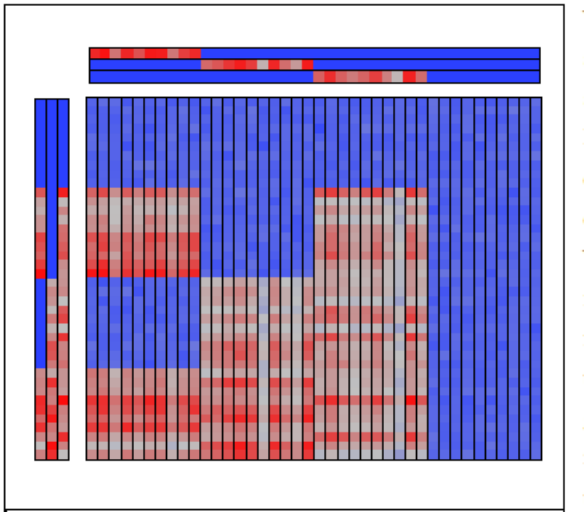


Figure 7: Even more realistic synthetic example. Two generating vectors lead to four kinds of individuals - normal controls and three disease groups with two etiologies. A third generating vector leads to a group of genes which are activated in multiple disease groups, e.g. inflammation genes, including both etiologies of the disease at issue.

3.3 Extended realistic mixture

We extend our realistic example by adding a set of 10 genes that are activated in either or both etiologies E1 and E2 (**Figure 7**). This situation can arise with a number of medical conditions. When a person gets sick from any of a number of diseases, multiple genes –

for example for inflammation – are turned on. These general response genes are not specific to the disease at issue and can cause great confusion as they can be taken as markers for the disease.

Three SVD vector pairs capture nearly all of the variance in the matrix, 99.7%, with singular values of 60.8, 26.2 and 13.1. NMF also recovers 99.7% of the variance. However, now that "inflammation genes" are turned on for all groups except the control group, NMF does not find the groups well and has a “ghost” in one of the NMF vectors (**Figure 8**). This clearly illustrates how deviations from D&S’s sufficient conditions can not always be surmounted, even with a SVD-based initialization scheme.

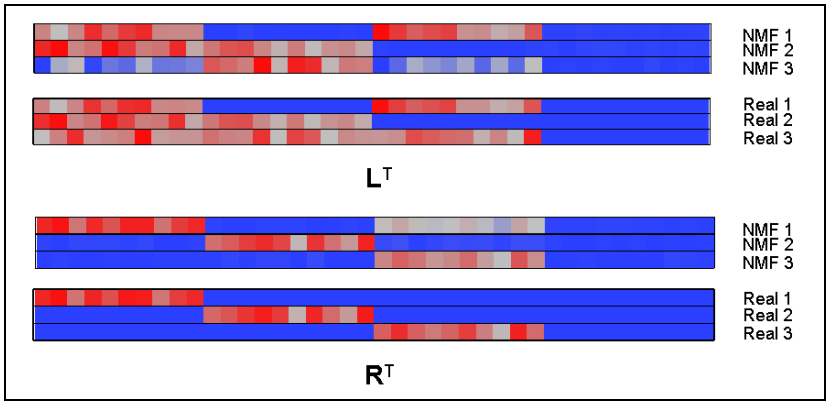


Figure 8: Heatmaps of NMF left and right singular vectors. The generating vectors are not recovered. Note the “ghost” of the third set of genes in the first right NMF vector.

4. DISCUSSION

The SVD and NMF are alternative ways to factor a data matrix. The major properties of SVD and PCA are well-known. The data matrix is decomposed into left (score) and right (loading) vector pairs. Adding more vector pairs allows better and better approximation to the elements of **X**. Conceptually the NMF is nothing more or less than an SVD in which negative elements in the left or the right singular vectors are prohibited. As such, in settings where the data matrix **X** consists of non-negative elements, the NMF factorization is likely to be more interpretable, and to more plausibly represent an actual generating mechanism. SVD can be looked at as optimizing the prediction of the elements of **X**, whereas NMF is attempting to explain the data via **L** and **R**, along the lines of **Shmueli G. (2010),** to predict or to explain**.**

As the left and right factors of the NMF are conceptual analogs of the left and right singular vectors in a SVD, they may be used for all the same purposes – for example cases may be clustered using the rows of **L**, and variables may be interpreted using the rows of **R.**

A graphic illustration of the power of the NMF method is in the dramatic picture in **Lee and Seung (1999)**, showing the decomposition of photographs of faces into ears, eyes, nose, etc. by the successive columns of the NMF. Other papers have used such terms as “meta genes” to describe groups of genes that group together **(Brunet et al. 2004).** Clearly the utility of NMF is in the direction of interpretation. The elements of the factoring vectors lead to an interpretation of how the internal components of **X** are

combined in each sample. In the case of genes, sets of genes might be up or down regulated together. In the case of metabolites, molecules in a pathway might be up or down regulated together.

Mixtures deserve serious mention. NMF appears to be able to decompose mixtures into their component parts. The earliest chemistry example of what is effectively NMF that we have found is **Lawton and Sylvestre (1971).** There, on page 628, equation (31) is the separability rule (R2) of **D&S** in the context of non-binary, spectro photometric curves (but for k=2). In psychometrics, NMF may prove a useful tool for resolving test data matrices into factor-like ability scores and matching subject loadings. In a more speculative vein for medical examples, vectors of **R** might correspond to different etiologies. For some people the disease may involve only one of the etiologies (i.e. right vectors), for other people it may involve multiple etiologies.

The selection of the number of components *k* is a decision that has to be made. When you do a SVD, each new pair of singular vectors is orthogonal to all the vectors that went before. If you overfit by choosing too large a rank, you will be trying to interpret pairs of junk coefficient vectors generated by noise, a process with high potential for misleading conclusions. A common approach is to make a scree plot, a plot of the series of successive eigenvalues against the integers. Conceptually, the scree plot drops steeply while the terms are still capturing "structure" and then flattens once they are capturing "noise." Guided by this idea the analyst makes an "arts and crafts" judgment call on the value of *k*, the number of signal components. The scree plot is open to many objections: like the fact that its shape can depend a lot on whether you plot on a natural, a square or a log scale. While these objections are legitimate, the scree plot is still widely used.

Unlike the SVD, the NMF with *k* components is not a concatenation of *k* successive optimal terms, but also seeks a best rank-*k* approximation to the original data matrix within the limitations that its left and right factors have only non-negative elements. So to the extent that NMF is finding a good rank-*k* approximation, it should be close to a linear transformation of the SVD. However, NMF does not have an orthogonality constraint, and so it is able to create a new dimension by, say, copying one of the row vectors and splitting the corresponding column vector into two parts. This has much less potential for confusion than does an over fitted SVD component. So getting *k* just right should not be as critical as with SVD.

Questions remain over how to pick the dimensionality of a NMF, and of numeric diagnostics that can be applied to **X** to decide whether a NMF is likely to succeed. By fitting a NMF and SVD of the same dimension to a data set and comparing their variance explained, the user will be alerted to the situation where the non-negativity constraints are not supported, but this still leaves open the possibility that some transformation of the left and right factors would lead to better interpretation. Also of concern is that, because of non-convexity, fitting algorithms are not guaranteed to converge to the global optimum, but are to some degree at the mercy of their initialization. These are all valuable avenues for further research.

Data Sets and Software

The synthetic data sets and SAS JMP scripts used in this paper can be downloaded from www.niss.org/irMF. Orange is an open source and free statistical analysis system based on Python, http://orange.biolab.si/. There is a module, Orange-NMF, for Orange that deals with various aspects of non-negative matrix factorization, http://orange.biolab.si/addons/.

REFERENCES

Baier, L.J., and Hanson, R.L. (2004) Genetic studies of the etiology of type 2 diabetes in Pima Indians: hunting for pieces to a complicated puzzle. *Diabetes*, 53, 1181-1186.

Brunet, J.P., Tamayo, P., Golub, T.R. and Mesirov, J.P. (2004) Metagenes and molecular pattern discovery using matrix factorization. *Proceedings of the National Academy of Science*, 101, 4164–4169.

Devarajan, K., (2008) Nonnegative matrix factorization: an analytical and interpretive tool in computational biology. *PLoS Comput Biol*. 25;4(7):e1000029.

Donoho, D., and Stodden, V. (2004) When does non-negative matrix factorization give a correct decomposition into parts? *Advances in Neural Information Processing System16*, Cambridge, Mass, USA: MIT Press. pages 1141-1148.

Faber, N. M., Meinders, M. J., Geladi, P., Sjöström, M., Buydens, L. M. C., and Kateman, G., (1995) Random error bias in principal component analysis. Part I. derivation of theoretical predictions. *Analytica Chimica Acta*, 304, 257–271.

Gabriel, K.R. and Zamir, S. (1979) Lower rank approximation of matrices by least squares with any choice of weights. *Technometrics*, 21, 489–498.

Good, I.J. (1969) Some applications of the singular decomposition of a matrix. *Technometrics*, 11, 823-831.

Greenacre, M.J., and Underhill, L.G. (1982), "Scaling a data matrix in a low-dimensional Euclidean space." in DM Hawkins (ed) *Topics in Applied Multivariate Analysis*, Cambridge Universaity Press.

Lawton, W.H., and Sylvestre, E.A. (1971) Self modeling curve resolution. *Technometrics*, 13, 617-633.

Lee, D.D., and Seung, H.S. (1999) Learning the parts of objects by non-negative matrix factorization. *Nature*, 401, 788-791.

Lee, D.D., and Seung, H.S. (2001) Algorithms for non-negative matrix factorization. in Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 556-562. MIT Press.

Lin, C. J. (2005) Projected gradient methods for non-negative matrix factorization. Tech. Report Information and Support Service ISSTECH-95-013, Department of Computer Science, National Taiwan University.

Liu, L., Hawkins, D.M., Ghosh, S., and Young, S.S. (2003) Robust singular value decomposition analysis of microarray data. *Proceedings of the National. Academy of Science*, 100, 13167-13172.

Shmueli G. (2010) To Explain or to Predict? *Statistical Science*, 25, 289-310.

Wild, S, Curry, J., and Dougherty, A. (2004) Improving non-negative matrix factorizations through structured intitialization. *Pattern Recognition*, 37, 2217–2222.