Factorization Techniques in Nonnegative Matrix Factorization

## Zack Tillotson

## Oct 2010

# Abstract

# Background

Nonnegative matrix factorization (NMF) is a technique which has gained prominence in the last decade as a powerful machine learning tool. NMF works by extracting the prominent features of a dataset and describing each data point as to how well they can be attributed each of the features. This approach is not only powerful in its representational ability, but also in how the underlying structure of the feature based approach can be intuitively understood and applied. This contrasts with other classification techniques such as Support Vector Machines, Neural Networks, and Principle Component Analysis which are largely black boxes in which the model works, but no deeper understanding of the data is gained by understanding the model itself.

NMF has been used in a wide variety of tasks, such as classifying images[1], grouping text documents for text mining[1][3], spectral analysis [3], and collaborative filtering in reviewing movies [4]. This wide variety of applications, with many different environments and constraints, shows that NMF is a powerful tool

For an example of how NMF works consider a text mining application in which many text documents have been broken down into a list of the non-trivial words in that document and the number of times each word appears. This can be converted into a matrix where each row is a document and each column is one of the non-trivial words. The value of an element in the matrix is the number of times that word appears in that document. For example:

[ text mining input matrix pic]

In NMF an input dataset is factorized into two new matrices - the features matrix and the weights matrix. When these two new matrices are multiplied back together they recreate the original input matrix. The factorization takes the form:

[ nmf decomposition example ]

The features matrix describes the features of the data set, in our example our features might be document themes such as "Political" or "Sports", but the features could be any other descriptions of the documents as well, such as "Long" or "Biographical". The values in this matrix describe how much that word is important to describing feature. The values in the "Sports" feature might have high values for the words "Ball", "Play", and "Teebow" and low scores for the words "Congress" and "Madonna".

The weights matrix describes each document by scoring each it with regard to each of these features. A lengthy article about politics should then get a high score for the "Political" and "Long" features and a small score for "Sports". These scores are all constrained to be non-negative, this additive representation enforces that each element is described as the sum of a set of parts since no part can be "cancelled out" by another part.

In order to create the perfect factorization it would be easiest to assign each document one feature which describes it exactly, but by having a small number of features that each document must be assigned to, similar documents are categorized together and themes can be discovered. This understanding via generalization is a consistent theme in machine learning.

In an actual NMF implementation the features are not given, instead of describing a feature and then scoring each item on it, the input matrix is blindly factorized into the two submatrices such that the error when they are unfactorized is minimized. The features and weights matrices are then interpretted as describing important features of the data set, but obviously don't have explicit descriptions. One common way humans interpret a feature is to look at the highest weighted objects relating to it. In the text mining example, the highest scoring words in the features matrix would show which words compose that feature and the highest scoring articles in the weights matrix would show which articles are best described by the feature.

Simply being able to understand the deconstructed model is not the only way in which NMF is a powerful tool, though. NMF can be used to approximate missing members of an incomplete input matrix. For example, consider a movie suggestion engine in which users rate movies they have seen and then wish to be provided with probable scores for other movies they have not yet seen. The form of this input would be a sparse matrix with a row for each user and a column for each movie. Given a large number of movies in a typical movie database, most users will not have seen even a small fraction of the movies, so only very few of the elements will have scores, like this:

[ movie review input example ]

If NMF is used on this input, the input matrix can be factored into the features and weights matrix using just the few scores that do exist. Doing the unfactoring will then will provide a full matrix with values in every element. These elements can be viewed as guesses for how the person will score that movie.

How the input matrix is factored into the submatrices is still an open question in many regards. Several algorithms are popular, including Multiplicative Update (MU) [2], Singular Value Decomposition (SVD), and Alternating Least Squares (ALS) [5]. Each of these will be described and analyzed in the context of NMF.

# Algorithms

## Multiplicative Update

## Singular Value Decomposition

## Alternating Least Squares

# Conclusion

# References

1. **Learning the parts of objects by non-negative matrix factorization**, Daniel D. Lee & H. Sebastian Seung, 1999 - <http://www.seas.upenn.edu/~ddlee/Papers/nmf.pdf>
2. **Algorithms for Non-negative Matrix Factorization**, Daniel D. Lee & H. Sebastian Seung, 2001 - <http://hebb.mit.edu/people/seung/papers/nmfconverge.pdf>
3. **Algorithms and Applications for Approximate Nonnegative Matrix Factorization**, Michael W. Berry, et al., 2006 - http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.140.1963&rep=rep1&type=pdf
4. Y. Koren, ["The BellKor Solution to the Netflix Grand Prize"](http://www.netflixprize.com/assets/GrandPrize2009_BPC_BellKor.pdf), (2009). - http://www.netflixprize.com/assets/GrandPrize2009\_BPC\_BellKor.pdf
5. Non-Negative Matrix Factorization Based on Alternating Non-Negativity Constrained Least Squares and Active Set Method, [Kim, Hyunsoo](http://smartech.gatech.edu/browse?value=Kim,%20Hyunsoo&type=author) & [Park, Haesun](http://smartech.gatech.edu/browse?value=Park,%20Haesun&type=author), 2007 - <http://www.cc.gatech.edu/~hpark/papers/simax-nmf.pdf>
6. Programming Collective Intelligence, Toby Segaran 2007
7. On the Convergence of Multiplicative Update Algorithms for Non-negative Matrix Factorization, Chih-Jen Lin, 2007? - http://www.csie.ntu.edu.tw/~cjlin/papers/multconv.pdf
8. Bag of Words dataset, UC Irvine machine learning dataset repository - <http://archive.ics.uci.edu/ml/datasets/Bag+of+Words>