Factor Analysis (FA)

Non-negative Matrix Factorization (NMF)

CSE 5290 - Artificial Intelligence Grad Project

Dr. Debasis Mitra

Group 6

Taher Patanwala

Zubin Kadva

Factor Analysis (FA)

1. Introduction

Factor analysis is used to determine the variability among the data. It is a technique that can be used to reduce the dimensionality of the data. The idea behind factor analysis is to identify the variance in the observed data, and determine the unobserved data that is smaller than the actual data, but represents the same thing.

It becomes easier to analyze the data, after it has been reduced, allowing us to focus more on key distinguishing factors, rather than wasting time on too many variables.

In order to perform factor analysis, we have to operate under the assumption that there exists a linear relationship between the variables of the data set. We identify factors by determining the correlations between the variables in the data set.

Factor loading is the measure of how much correlation does the variable have with the factor. Thus, a higher factor loading means that the variables are closely related to the identified factor.

2. Explanatory Factor Analysis

There are two types of Factor Analysis. Explanatory Factor Analysis(EFA) and Confirmatory Factor Analysis(CFA). CFA is a more complex approach. EFA can be termed as an advanced version of Principal Component Analysis(PCA). EFA splits the dataset in to different categories making it easier to analyze. EFA usually works better with larger sample sizes, but if the factor loading among the variables is high then smaller sample sizes can also be used. The correlation needs to be higher than 0.30, otherwise it would mean that the relationship is very weak between the variable and the factor.

Problems arise when a variable falls into more than one factor. Such a situation is called split loadings. The variable has low correlation with multiple factors, and it becomes difficult to put such a variable into a specific factor group.

3. Theoretical Background

The theory behind the working of EFA can be explained using the mathematical and geometrical approaches.

3.1. Mathematical Approach

In this approach *p* denotes the number of variables in the dataset($X_1, X_2, ..., X_p$) and *m* denotes the number of factors($F_1, F_2, ..., F_m$). Each variable in the dataset is represented mathematically as below:

$$X_j = a_{j1}F_1 + a_{j2}F_2 + \cdots + a_{jm}F_m + e_j$$

where j = 1,2,3,…,p.

In the above equation $a_{j1}, a_{j2}, \ldots, a_{jm}$ are the factor loadings. This specifies how much effect the variable has on a particular factor. The specific or unique factor is denoted by e_j. It can be said that the Factor Analysis is similar to weights. A higher factor loading represents that there is a high correlation between the factor and the variable. So, factor loading determines the strength of correlation between the variable and the factor.

We need to calculate the correlation coefficient, and in order to do that, we need to identify the common features in the variables and based on that either create a correlation matrix or a covariance matrix. The correlation coefficient is used to determine the relationship between two variables.

Let us say that we have *p* variables and *m* factors. For every two pairs of variables we have to try to extract factors such that there are no intercorrelations left between those variables, as the factor itself will behave as the intercorrelations. Factor analysis can be represented by the below equation:

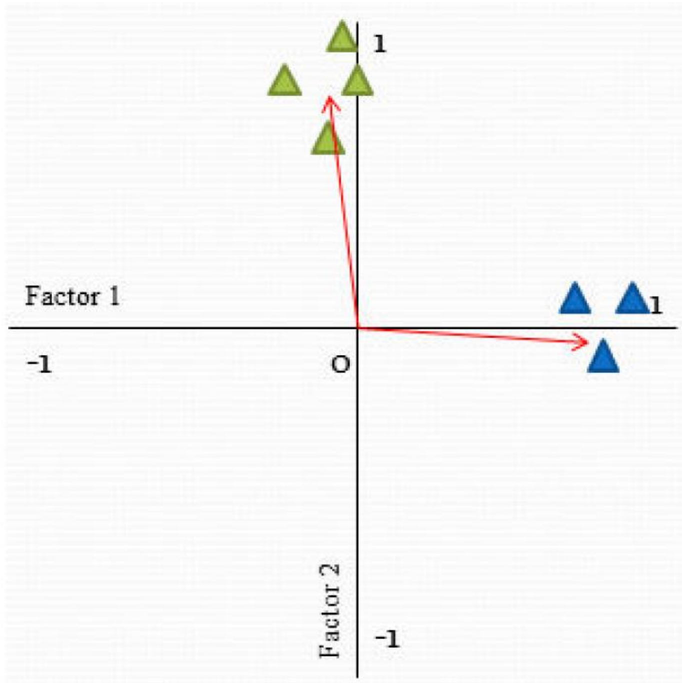
$$R = P\ C\ P' + U^2$$

where R = correlation coefficients matrix
P = Factor loading matrix
C = correlation matrix
U = diagonal matrix of unique variances of each variable

Communality can be produced in factor analysis by using variances. It is the square of the summation of factor loadings for a particular variable.
The formula is $h_j^2 = a_{j1}^2 + a_{j2}^2 + \cdots + a_{jm}^2$.

3.2 Geometrical Approach

We can represent factor analysis using geometry to for better understanding. In this representation each axis represents a factor and the vectors or lines on the graph represents the variable. So if a variable is highly correlated to a factor than, it will be very close to that axis.
The axis will range from -1 to 1 which represents the factor loading.



The above figure is an example of factor analysis in geometrical representation. In this figure the two axis are the two factors. Factor 1 and Factor 2 and the triangles are the variables. The Blue triangles have higher correlation with Factor 1 and the Green triangles have a higher correlation with Factor 2.

4. Factor Extraction

There are several techniques available that can be used to determine the factors from the variables. We can select a technique based on the requirements and the research that we are trying to perform. Maximum Likelihood, Principal Axis Factor, are some of these techniques. Principal Component Analysis(PCA) can also be used, as it a used for data reduction. An issue has been raised that weather PCA is actually a factor analysis technique on its own.

After factors are identified, rotation is performed on the factors. The objective of performing rotation on the factors is to reduce ambiguity. This allows us to fit more variables into less number of factors.

Next we need to determine the strength of the factors by examining the factor loadings. We need to examine the factors with high factor loading, and factors with low factor loading, and ensure that they are consistent with the data. Like there should not be cases where a factor that should have low correlation with a variable has high factor loading. Also there should be very low split loadings, meaning variables that load into more than one factor should be less.

The next step is to determine the number of factors to retain. This is very important, because if we keep too many factors than it may result in a high error variance, on the other end keeping very few factors may result in loss of important data. We can use eigenvalues and scree test to determine the number of factors to retain. It is recommended to use both the techniques together, as using only eigenvalues may result in overestimation.

5. Data Set Description

We will use the data from AEIS(Academic Excellence Indicator System) which is provided by Texas Education Agency. This dataset has records of thousands of schools in Texas. We will use factor analysis to reduce the dimensionality of the data, making it easier to analyze this huge dataset.

6. Difference between FA and NMF

FA	NMF
Can work with negative data	Cannot work with negative data
It reduces data dimensionality by identifying factors and can be used to represent the variables	It reduces the data, by splitting the data into smaller subsets
It identifies the factors between the variables with high correlation, and then the we can use to factors to analyze our data set	It splits the data, such that the distance(Euclidean or Frobenius) between the original matrix and the subset matrices is minimum

7. References

[1] An Gie Yong and Sean Pearce. “A Beginner’s Guide to Factor Analysis: Focusing on Exploratory Factor Analysis” published in “Tutorials in Quantitative Methods for Psychology”, Vol. 9(2), p. 79-94, 2013.

[2] Academic Excellence Indicator System. (n.d.). Retrieved October 11, 2016, from “https://rptsvr1.tea.texas.gov/perfreport/aeis/”

Non-negative matrix factorization (NMF)

1. Introduction

NMF is a group of algorithms where a matrix V can be decomposed into two matrices W and H, each of which are easier to work with and when multiplied together, yield the original matrix.

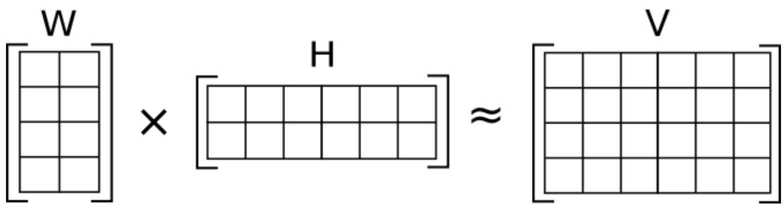


Fig. V (4 X 6) is approximated as W (4 X 2) multiplied by H (2 X 6)

(Source: https://en.wikipedia.org/wiki/Non-negative_matrix_factorization)

Given, a matrix V of dimension $m \times n$ and $v_{ij} >= 0$, NMF decomposes it into 2 matrices W and H of dimension $m \times r$ and $r \times n$ where

$$W_{ij} >= 0$$

$$H_{ij} >= 0$$

$$r < min\ (m, n)$$

Thus, V is decomposed into a tall, skinny matrix W and a short, wide matrix H. The user can specify r as the inner dimension of W and H as long as $r < min\ (m, n)$.

Each column of V, v_i can be calculated as:

$$v_i = W * h_i$$

Thus, each column of W is weighted by its corresponding row in h_i, which are then added together to form columns of V.

2. Purpose

Suppose V is a large dataset where each column is an observation and each row is a feature. For example, in a database of images, a column might represent some image and a row can represent a pixel. In machine learning, it is necessary to reduce the feature space for easy computation. In the above example, it is difficult to consider each pixel value every time an image is handled, so it is worthwhile to break it down into fewer components. Thus, NMF is used as a new way of reducing the dimensionality of data.

Since NMF has a non-negative constraint, it is used to represent data with positive features. This advantage can be used in image processing since each image has a positive pixel value.

NMF is similar to PCA where each base is assigned a weight. But in NMF, the weights are constrained to be positive.

2. Applications

2.1. Computer vision

NMF is beginning to be used in many fields. It is used in computer vision to reduce the feature space in images. This can be useful in identifying and classifying images.

2.2. Text mining

NMF is also used in text mining. For example, you might organize a series of documents into a matrix where each column may represent the frequency a particular word and a row might represent the document. Then you would extract sematic features about the data.

2.3. Speech denoising

NMF is used to break audio recordings of speech into speech parts and noise parts so that the speech parts alone can be isolated.

3. The problem

A fundamental model in NMF utilizes the least squares cost function to measure the closeness of matrices, resulting in the following standard NMF problem:

$$\text{Minimize } \|V - WH\|^2 \; subject\ to\ W, H \ \geq 0$$

where $\|\cdot\|$ is Frobenius norm, and the inequalities are component-wise [1].

4. Algorithm

The method used for solving the above problem seems to be the alternating least squares (ALS) algorithm utilized by Paatero and Tapper in 1994 [2]. It minimizes the least squares cost function with respect to either W or H, one at a time, while fixing the other and disregarding non-negativity, and then sets any negative entries to zero after each least squares step.

The conventional approach to find W and H is by minimizing the difference between V and W H:

$$f(W,H) \equiv \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{m}\left(v_{ij} - (WH)_{ij}\right)^2$$

$$where\ W_{ia} \geq 0, H_{bj} \geq 0, \quad \forall\ i, a, b, j$$

Lee and Seung (2001) have shown that the function value is non-increasing after every update [3]:

$$f(W^{k+1}, H^k) \leq f(W^k, H^k) \text{ and}$$
$$f(W^{k+1}, H^{k+1}) \leq f(W^{k+1}, H^k)$$

From the non-increasing property, the multiplicative update algorithm is a special case of a general framework, which alternatively fixes one matrix and improves the other:

$$\text{Find } W^{k+1} \text{ such that } f(W^{k+1}, H^k) \leq f(W^k, H^k) \text{ and}$$
$$\text{Find } H^{k+1} \text{ such that } f(W^{k+1}, H^{k+1}) \leq f(W^{k+1}, H^k)$$

4.1. Alternating non-negative least squares [4]:

1. Initialize $W_{ia}^1 \geq 0, H_{bj}^1 \geq 0, \quad \forall\, i, a, b, j.$

2. For k = 1, 2, . . .

$$W^{k+1} = \arg \min_{W \geq 0} f(W, H^k)$$

$$H^{k+1} = \arg \min_{H \geq 0} f(W^{k+1}, H)$$

This approach is the “block coordinate descent” method in bound-constrained optimization (Bertsekas, 1999) [5], where sequentially one block of variables is minimized under corresponding constraints and the remaining blocks are fixed. For NMF, we have the simplest case of only two block variables W and H.

5. Data

We consider an image problem:

CBCL face image database [6]

http://cbcl.mit.edu/cbcl/software-datasets/FaceData2.html

This data set consists of 472 faces, 23,573 non-faces. The idea is to reduce the dimensionality of the above data so that it can be used for further analysis and computation.

6. References

[1] Yin Zhang, "*An Alternating Direction Algorithm for Nonnegative Matrix Factorization*", Rice University, January 2010

[2] Paatero, P. and Tapper, U., "*Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values*", Environmetrics, 1994

[3] Lee, D. and Seung, H., "*Algorithms for Non-Negative Matrix Factorization*". Advances in Neural Information Processing Systems, 2001

[4] Chih-Jen Lin, "*Projected Gradient Methods for Non-Negative Matrix Factorization*", Neural Computation, MIT Press, 2007

[5] Dimitri P. Bertsekas. "*Nonlinear Programming*", Athena Scientific, 1999

[6] CBCL Face Database #1, MIT Center for Biological and Computation Learning, *http://www.ai.mit.edu/projects/cbcl*