**Customer Segmentation Framework**

**Author: Kecheng Xu, Yang Xing**

In this paper, we will try following approaches to perform customer segmentation and compare them on the testing data set. All the approaches are unsupervised learning methods, some of approaches are Soft segmentation, some are hard segmentation. In the end, we will use a bagging ensemble method to combine all approaches and validate if the ensemble method has a better performance on the testing set. The final model will use hard segmentation which will only give the predicted segmentation class for each customer but not the prob distribution. The suggesting approaches are as following:

1. SVD + K-means
2. PCA + SVD + k-means
3. Funk-SVD + K-means
4. PCA + Funk-SVD + k-means
5. PCA + NMF
6. LDA

We will also try hierarchical methods like HLDA and perform above methods under each pre-determined parent segment. This part will in the last section of this paper. The details about each approach are listed below for your reference.

Notice: 1. SVD is only typical matrix decomposition, it can not guarantee all decomposed matrix to be non-negative. Thus, after applying SVD we need to apply K-means on it to get the hard segmentation.

2. PCA is converting covariance matrix to diagonal, SVD is converting coefficient matrix or design matrix to diagonal, so we can use them together. Diss later.

3. Funk-SVD assumes the word that is not appeared in the document as a missing data and use an optimization method to fit this missing data, but SVD takes that word as zero value.

4. NMF is also an optimization method and is able to get non-negative matrix, so it is more popular in topic model.

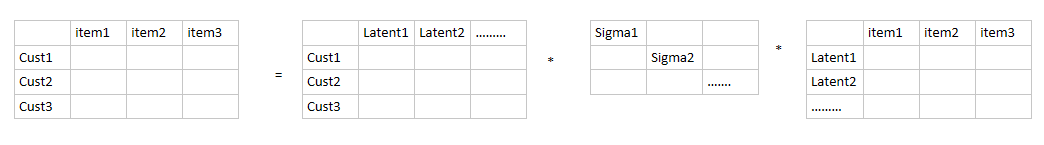
5. we want to convince that matrix decomposition methods is better than LDA, but it is up to the final result.

1. **SVD:**

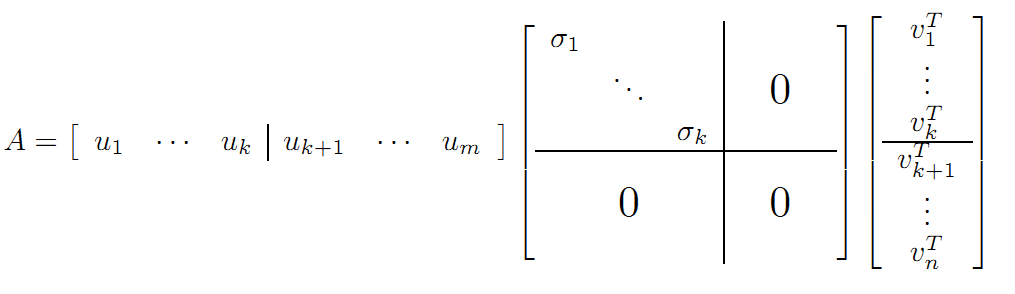
For any matrix , it turns out that can be decomposed by two orthogonal matrix U and V and a diagonal matrix as following:

SVD is very popular in image compression. In NLP, the latent semantic analysis (LSA) use SVD to mining the latent semantic class or hidden topics of different documents. Moreover, in NLP many similar model such as Latent semantic indexing (LSI), latent factor model (LFM) are all very close to LSA whose core algorithm is just SVD. In the recommendation system, people use collaborative filtering model (CF) to recommend top-N items for users, whose one of core algorithm is also SVD and its variants.

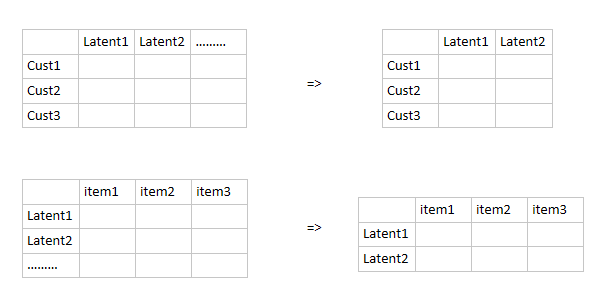
In terms of the collaborative filtering model (CF):



The entry in each row represents the ratings that a customer give to a specific item. We can decompose the matrix as three-matrix multiplication. The left matrix is the ratings or interests a customer has for different latent fields, such as sports, games, movies, food, travel and so on. The number of such latent fields is arbitrary, but it turns out that some of the sigma will contain most of the information of that matrix. Therefore, we can only keep some sigma on the diagonal. For instance, we only keep k number of sigma, then the matrix A can be decomposed as following:

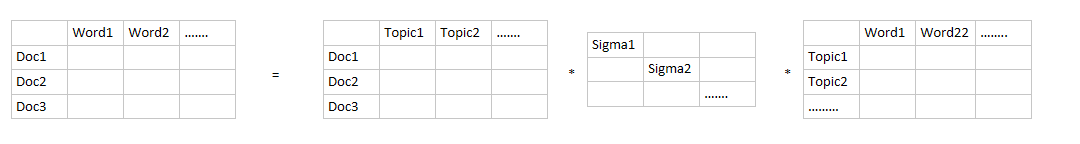


Based on the knowledge of block matrix, because the upper left corner submatrix is the only non-zero matrix in the block matrix, the left matrix U and right matrix V can be embedded into lower dimension. Therefore, we have following conversion:

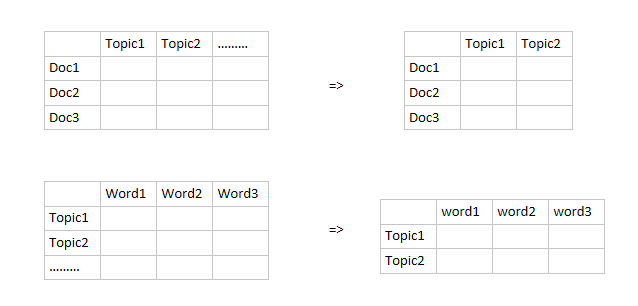


The left matrix represents the interest rating a customer to a specific latent fields, the right matrix represents the correlation coefficients or weights between the latent fields and different items. The middle sigma diagonal matrix tells us which latent fields should be remained. However, in the practical, the original matrix A is very sparse, because a customer can only rate or purchase tiny part of all items (products). Therefore, many entry of the matrix A will be missing value. Such matrix can’t be decomposed by SVD. Usually, mean or median value will be plug into the empty entry in order to solve this issue. Some optimization solution developed to solve this issue such as funk-SVD. It turns out that funk-SVD is very efficient and popular. Afterwards, the filled matrix can be decomposed and we only keep some of sigma with big values. Then we multiply three matrix to reconstruct the previous matrix A and use such reconstructed matrix to substitute previous one. Eventually, the top-N recommendation task is very easy, based on the reconstructed matrix, for each customer (for each row), we rank the items rating and choose the top-N-items with the biggest rating value.

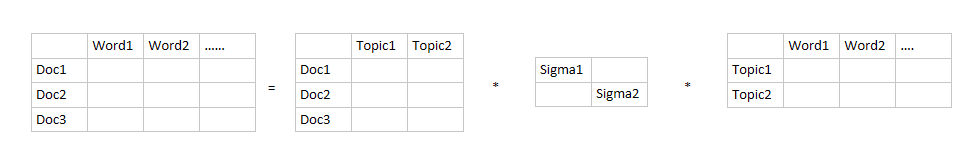
In terms of LSA/topic model, we have tfidf-matrix. Very similar to above CF model:



Similarly, if we only keep two latent topics here, we will have following conversion:



The reconstructed matrix is as following:



The topic model can help us cluster documents based on the latent topic distribution. The first matrix on the right hand side of the equal sign represents the correlation weights between each document and each latent topic. The third matrix is represents the correlation weights between each latent topic and each word. The middle diagonal matrix decide which latent topics should be remained in the model.

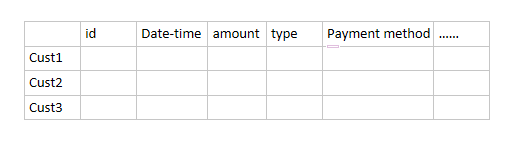
The customer segmentation issue is the most fundamental and important issue in the financial industry especially in the bank industry. The common methods and procedures of customer segmentation is from the business perspective. Specifically, the customer acceptance and on-boarding process applies to all new customers, CIP and CDD have to apply to all customers to make sure bank has completed the basic identification and investigation on each customer.

Based on the CIP and CDD information, bank is able to perform customer segmentation. For instance, bank usually segment customers into individual and non-individual, furthermore, individual can be split as natural individual and legal individual, non-individual can be split as bank, non-bank division. On the other hand, from the customer social and attributes perspective, customer can be classified as individual, government, entity.

However, the above customer segmentation do not consider the transaction behaviors and financial behavior of a customer. The issue is many models in bank industry require not only the static customer segmentation but also require the customer segmentation based on the historical transaction and the previous transaction behavior of the customer. Such segmentation method is data driven and dynamic and is able to find the hidden pattern behind the customer’s financial behavior.

This article try to apply the LSA model in NLP to the customer segmentation issue. Firstly, we need to classify the rationale behind this application.

The transaction table for each customer is as following:



The transaction has many features, we can concatenate some of features as a string.

Suppose we have three features for a transaction:

Txn: F1, F2, F3

Such transaction can be concatenated to following words:

Txn: F1, F2, F3🡪’F1’, ’F2’, ’F3’, ’F1\_F2’, ’F2\_F3’, ’F1\_F3’, ’F1\_F2\_F3’

FICO convert each transaction to a highest ordered interaction item

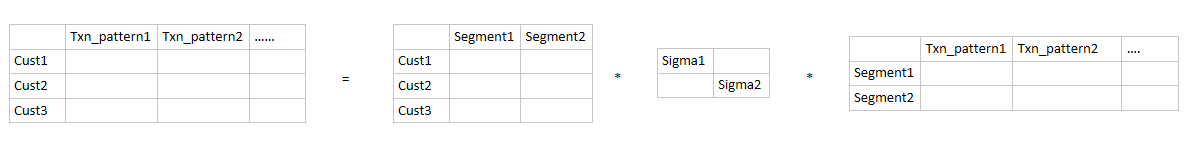
Txn: F1, F2, F3🡪’F1\_F2\_F3’

We should consider not only the highest order interaction item but also low order interaction item. In our case, one txn will be converted to many words as above. Such a word we call it one txn pattern.

Each customer has a bunch of transactions, each transaction has many words (patterns). If we group/collect all words for a customer, then such group of words consist a document. Therefore, a customer is just a document in LSA, a txn pattern here is just a word in LSA.

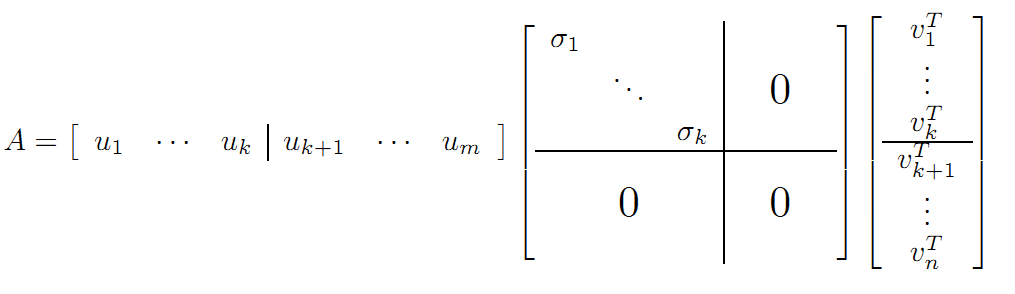
We don’t need to consider the sequence information because LSA is a bag of word model.

Therefore, we can convert the customer segmentation issue to a normal LSA issue as following:



The first matrix on the right side of equal sign represent the distribution of correlation weights between customer and the latent segmentation. The segment with biggest value for a customer in the left matrix is the belonging segment for this customer.

Before we dive into the above application, let’s take a look at following formula again and reconsider the matrix decomposition from the perspective of subspace and transformation.



The first left matrix on the right side of equal sign is matrix U which is consisted of orthogonal basis of m-dimension vector subspace. Similarly, the third decomposed matrix V is consisted of orthogonal basis of n-dimension vector subspace. However, is there any way that we can make sure these two subspaces to be the same subspace so that all column vectors in U and row vectors in V represent the same orthogonal basis? Here you can imagine this matrix A is just the tfidf-matrix or the customer-transaction pattern matrix.

The answer is YES! But an issue occurs: Since matrix A is too general, even the row amount and column amount are different. So intuitively we need to constrain A to be a square matrix. But only square matrix is not enough. Based on the Linear Algebra theorem, to guarantee square matrix A can be decomposed to a diagonal matrix, we have to constrain matrix to be symmetric. Once based on following theorem everything will be clear:

Any real symmetric matrix can be decomposed as following:

P is orthogonal matrix which is consisted of orthogonal basis in k-dimension subspace, k is the rank of A. The matrix is diagonal and can be ranked from biggest value to smallest on the diagonal. We can think of this theorem as a particular situation of SVD. Adding the constraints on A will bring us many benefits, the biggest one as we talk above is we find a set of orthogonal basis in k-subspace. BUT what is the use of this set of orthogonal basis?

As we all know from linear algebra, the matrix P is called orthogonal matrix and orthogonal matrix is consisted of orthogonal basis, we can write is a set of orthogonal basis. Recall the regression part, suppose we have a predictor coefficient matrix , each column in X is one feature or predictor vector. , assume the rank of matrix X is m which means X is full rank in column.

Therefore, any column vector of X can be linear represented by m original orthogonal basis: .

Coordinate Value based on different orthogonal basis

suppose is the coordinate vector of on the basis of . Then can be represented by basis as following:

But the question is how to represented the same column vector by other orthogonal basis? Suppose there is an orthogonal matrix , also we know all is orthogonal basis just like , Then we have:

Here if we focus on , you will find it is also a column vector:

Therefore,

Notice that is a scalar value. Thus, we know is the coordinate vector of on the basis of .

Let’s look an example, suppose there are two points on the coordinate graph (X,Y), the coordinate value of points A, B as shown on the graph. The original orthogonal basis is X=(1,0), Y=(0,1). Suppose we have an orthogonal matrix P:

Based on the above discussion, we know (

Therefore, the coordinate vector of A, B on the basis of is:

Take a look at following graph. It is obvious that the new orthogonal basis is just the green line coordinate system. Until now we can answer the first question above BUT what is the use of this set of orthogonal basis? Because when we use different basis to represent vector the coordinate vector for the same vector will be very different. Just as the above calculation result shows, the coordinate vector of A is from to and B is from to . However, does this coordinate transformation really matter? What benefits does this coordinate transformation bring to us?

An obvious benefit we have now is we convert all coordinate value of P2 to zero after we use to represent A and B points. That means when we measure the variability of points A, B, we do not need dimension anymore. Since this dimension do not contain any variability information. From this perspective, we actually reduce the dimension after using to represent A and B points. The variability here is just the covariance matrix of points A and B. When we use X, Y as basis, the covariance matrix is , but when we use as basis, the covariance matrix is . It is obvious that basis switch is able to convert covariance matrix to a diagonal matrix. It is also very easy to understand because covariance value is determined by the selection of basis. Different basis will generate different covariance matrix. Therefore, we need to find a set of orthogonal basis to make sure the corresponding covariance matrix will have the best interpretability. It is apparent that the diagonal covariance matrix should has the best interpretability, since it only contains variance which means the direction of such basis we select can explain the most of variability of the data.

Y

B

A

X

We know that the covariance matrix for any predictor coefficient matrix X has following expression if we standardize each column vector of X, we will have:

Suppose o= (1, 1, ….1) is n dimension real vector.

Then we have

We can only take a look at one item in above sum, suppose the coordinate vector of ) is :

Then we sum all of items together, then we have:

Since the mean of vector is zero, so above matrix is just the covariance matrix of X.

Because is always symmetric and furthermore it is positive definite which means all eigenvalues are positive, so we can rank all positive eigenvalues on the diagonal just as SVD.

Let’s go back to see the above theorem:

Any real symmetric matrix can be decomposed as following:

This guarantee that the covariance matrix of any matrix X (after standardized on each column) can be converted to a diagonal matrix by searching a set of orthogonal basis and such basis always exists. The whole process of searching orthogonal basis and represent coordinates vector by such basis is PCA. The benefits of PCA is:

1. By using proper basis to represent coordinates vector, the covariance matrix of predictor coefficient matrix X can be converted to a diagonal matrix. This means we find some best basis to explain the variability of data.
2. By ranking the diagonal values from biggest to smallest, we can find the sum of part top diagonal values occupies big ratio of total sum, which means only keeping that part of diagonal values can explain the most of variance of data. This will convert other diagonal values to be zero. As discussed above, the block zero sub-matrix will reduce the dimension and all predictor vectors can be linear represented by lower dimension subspace without losing much information.
3. From old basis to new basis, the corresponding orthogonal transformation of these two basis can tell us which predictors should be linear combined together as a basis. This also can be certified by the domain knowledge.
4. Notice that the positive definite matrix is just the covariance matrix of X, just mention that there is another positive definite matrix is also very important, especially in regression, which we will use later in our article. In order to use it later, it is good for us to review the regression and OLS a little bit.
5. **Regression:**

Suppose the regression model is:

Based on the OLS or MLE, we have following formula to figure out

Assume

Since , thus is a projection matrix or projection transformation.

That means Regression is a linear transformation that project the vector from the original vector space to a new vector space. The question is why do we need to perform this projection without any benefits? Actually, we have many benefits from this projection:

1. When we perform this projection, we project the vector into a sub space of and we really want to see that the dimension of this sub space is much smaller than n, even smaller than m. It turns out that the sub space dimension is just the rank of .
2. If the predictor vectors are independent, then the dimension of the above sub space is m and those predictor vectors are the basis of this sub space. If the predictor vectors are dependent, then we will face multi-collinearity issue. To solve this issue we need to collect the predictor variables as independent as possible to make sure the rank of is m.
3. In conclusion, the OLS-Regression will project observed vector into a sub space whose dimension is less than n and this sub space is spanned by which are just the columns of . Ideally, if the columns of are independent, then the dimension of this sub space is just m and are the basis of this sub space.

Each predictor vector is generated by a predictor variable. The element of predictor vector is just the observed sample value of corresponding predictor variable. In order to make all predictor vectors are as independent as possible, we need to find out the independent predictor variables. Therefore, the traditional regression feature selection process is just trying to find out a best sub space whose projection of can be closest to its self. The metric we measure the distance between and the projection is .

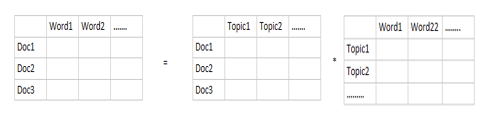
**3. Non-negative Matrix Factorization (NMF)**

NMF (Nonnegative Matrix Factorization) is one effective machine learning technique when we consider doing matrix decomposition/factorization with non-negative data. NMF has a wide range of uses, for example image processing and topic modelling. One of the biggest benefits of doing NMF is to create non-negative value on the decomposed matrices.

Suppose we are trying to factorize a matrix into two matrices and so that the product of and is equivalent to by optimizing the distance between them. This optimization method makes NMF to be a better approach in some fields compare to SVD.

In the perspective of linear algebra, we can interpret the entries in by a weighted sum of rows and columns of and .

In our case, we consider to use NMF into the Topic Modelling field, especially for term-document matrix. Each row is consider to be a Document and each column is consider to be the count of Word in that document. Further the count here not only means the raw count but also the tf-idf weighted count or some other encoding scheme.



For explanation purpose, imagine we have an article from magazine. The word ‘Basketball’ would be more likely appear in sport-related articles, and similar words like ‘NBA’ and ‘Rebound’ also. Ideally, these words will be grouped together under ‘Sport’ topic. And each article may have certain weight for of ‘Sport’ topic. And the interpretation of this example would be no sense if we have the negative values in the entries of decomposed matrices. That’s why NMF is considered to be a good approach to tackle the NLP problem.

Another important property for NMF is that it naturally produces sparse representations. This make perfect sense in the case of Topic Modelling. The similar approach in SVD respective is called Funk-SVD, a method to decompose a matrix with missing values.

As mentioned before, the NMF is a optimization method for matrix decomposition/factorization. The general approach for the optimization function is to have an objective function and use method like gradient descent to optimize the function by iteration. In the end, the objective function reaches its local minima. In NMF, the measure of distance is calculate through Frobenius norm.

In order to obtain more robust optimized result, some regularization techniques could be applied here. The existence of the regularization can prevent the weights from becoming too large. Either l1 or l2 regularization could be introduced on the objective function. NMF could regularize either one of the matrix(matrix or matrix ) or both and . To be remind, we need to introduce a weight parameter (ρ) to control the intensity. The overall regularization term would be the following:

For calculating the distance, there are another method like Kullback-Leibler (KL) divergence and Itakura-Saito (IS) divergence. Later on, we will talk more about the relationship between LDA and NMF with KL divergence.

1. Funk-SVD is also an optimization matrix decomposition methods which is close to NMF, we will complete this part soon……….