Effective early disease risk assessment with matrix factorization on a large-scale medical database

PLOS One

Chu-Yu Chin[1], Sun-Yuan Hsieh[1], Vincent S. Tseng[2]

[1]National Cheng Kung University, [2]National Chiao Tung University

dx.doi.org/10.17504/protocols.io.rv2d68e

Chu-Yu Chin

ABSTRACT

The early assessment of disease risk is an emerging topic in medical informatics. If diseases are detected at an early stage, prognosis can be improved and medical resources can be used more efficiently. A number of recent studies have considered risk factor analysis approaches, such as association rule mining, sequential rule mining, regression, and medical expert advice.

In this study, for improving disease risk assessment, non-negative matrix factorization and support vector machine (SVM) were integrated to discover important and implicit risk factors.

To make the method easy to follow, here we provide an experimental protocal. This experimental protocal comprises three main stages: data preprocessing, risk factor optimization, and early disease risk assessment. To discover the optimized risk factors, the NMF algorithm with parameter optimization was used for constructing the NMF-based matrix. In the assessment model learning and early disease risk assessment stages, the machine learning classifier SVM was used for disease modeling with the NMF-based matrix, yielding the final disease risk assessment, which serves as an excellent reference for physicians and patients.

EXTERNAL LINK

https://doi.org/10.1371/journal.pone.0207579

THIS PROTOCOL ACCOMPANIES THE FOLLOWING PUBLICATION

Chin C, Hsieh S, Tseng VS (2018) *eDRAM*: Effective early disease risk assessment with matrix factorization on a large-scale medical database: A case study on rheumatoid arthritis. PLoS ONE 13(11): e0207579. doi: 10.1371/journal.pone.0207579

PROTOCOL STATUS

Working

Install Prerequisite software and libraries

1 This protocol requires:
- Matlab 2016a
- LibSVM version 3.22

Each instruction can be verified on property websites.

The following steps include: dataset preparation, executing NMF multiplicative update algorithm, paremeter settings for SVM and effectiveness evaluation on the SVM-based disease risk assessment.

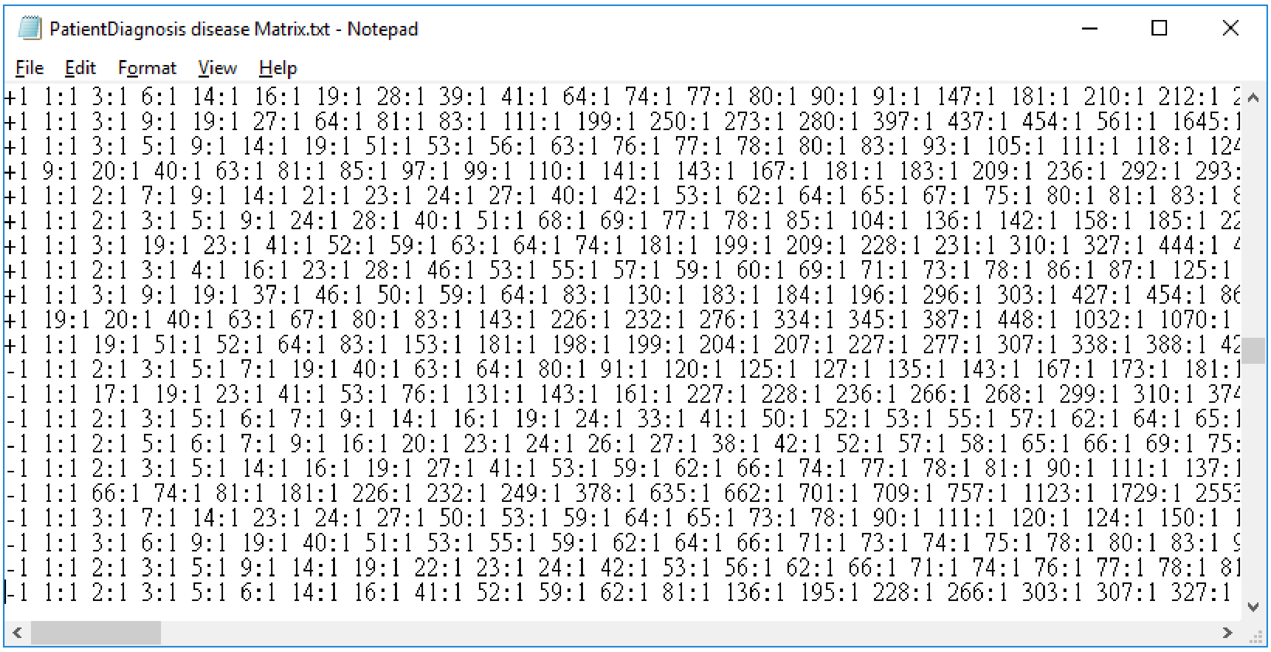
Prepare and load a Patient-Diagnosis Disease Matrix by the LibSVM format with label

2 The medical diagnostic record data to be analyzed is converted into a patient-diagnosis disease matrix in advance.
Given an $N \times M$ matrix, each row represents the medical history of a diagnosed patient across all diseases or symptoms (DS). Each column indicates the diagnostic record status of all patients for a single DS.

Running the command below in the Matlab to load patient-diagnosis disease matrix by the libSVM Format.

```
>> [label, inst] = libsvmread('C:\SVM_data\200.libsvm.txt');
```

An example of patient-diagnosis disease matrix with the LibSVM format.



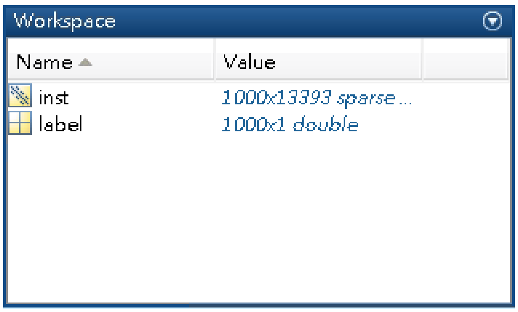
PatientDiagnosis disease Matrix.txt - Notepad

File Edit Format View Help

```
+1 1:1 3:1 6:1 14:1 16:1 19:1 28:1 39:1 41:1 64:1 74:1 77:1 80:1 90:1 91:1 147:1 181:1 210:1 212:1 2
+1 1:1 3:1 9:1 19:1 27:1 64:1 81:1 83:1 111:1 199:1 250:1 273:1 280:1 397:1 437:1 454:1 561:1 1645:1
+1 1:1 3:1 5:1 9:1 14:1 19:1 51:1 53:1 56:1 63:1 76:1 77:1 78:1 80:1 83:1 93:1 105:1 111:1 118:1 124
+1 9:1 20:1 40:1 63:1 81:1 85:1 97:1 99:1 110:1 141:1 143:1 167:1 181:1 183:1 209:1 236:1 292:1 293:
+1 1:1 2:1 7:1 9:1 14:1 21:1 23:1 24:1 27:1 40:1 42:1 53:1 62:1 64:1 65:1 67:1 75:1 80:1 81:1 83:1 8
+1 1:1 2:1 3:1 5:1 9:1 24:1 28:1 40:1 51:1 68:1 69:1 77:1 78:1 85:1 104:1 136:1 142:1 158:1 185:1 22
+1 1:1 3:1 19:1 23:1 41:1 52:1 59:1 63:1 64:1 74:1 181:1 199:1 209:1 228:1 231:1 310:1 327:1 444:1 4
+1 1:1 2:1 3:1 4:1 16:1 23:1 28:1 46:1 53:1 55:1 57:1 59:1 60:1 69:1 71:1 73:1 78:1 86:1 87:1 125:1
+1 1:1 3:1 9:1 19:1 37:1 46:1 50:1 59:1 64:1 83:1 130:1 183:1 184:1 196:1 296:1 303:1 427:1 454:1 86
+1 19:1 20:1 40:1 63:1 67:1 80:1 83:1 143:1 226:1 232:1 276:1 334:1 345:1 387:1 448:1 1032:1 1070:1
+1 1:1 19:1 51:1 52:1 64:1 83:1 153:1 181:1 198:1 199:1 204:1 207:1 227:1 277:1 307:1 338:1 388:1 42
-1 1:1 2:1 3:1 5:1 7:1 19:1 40:1 63:1 64:1 80:1 91:1 120:1 125:1 127:1 135:1 143:1 167:1 173:1 181:1
-1 1:1 17:1 19:1 23:1 41:1 53:1 76:1 131:1 143:1 161:1 227:1 228:1 236:1 266:1 268:1 299:1 310:1 374
-1 1:1 2:1 3:1 5:1 6:1 7:1 9:1 14:1 16:1 19:1 24:1 33:1 41:1 50:1 52:1 53:1 55:1 57:1 62:1 64:1 65:1
-1 1:1 2:1 5:1 6:1 7:1 9:1 16:1 20:1 23:1 24:1 26:1 27:1 38:1 42:1 52:1 57:1 58:1 65:1 66:1 69:1 75:
-1 1:1 2:1 3:1 5:1 14:1 16:1 19:1 27:1 41:1 53:1 59:1 62:1 66:1 74:1 77:1 78:1 81:1 90:1 111:1 137:1
-1 1:1 66:1 74:1 81:1 181:1 226:1 232:1 249:1 378:1 635:1 662:1 701:1 709:1 757:1 1123:1 1729:1 2553
-1 1:1 3:1 7:1 14:1 23:1 24:1 27:1 50:1 53:1 59:1 64:1 65:1 73:1 78:1 90:1 111:1 120:1 124:1 150:1 1
-1 1:1 3:1 6:1 9:1 19:1 40:1 51:1 53:1 55:1 59:1 62:1 64:1 66:1 71:1 73:1 74:1 75:1 78:1 80:1 83:1 9
-1 1:1 2:1 3:1 5:1 9:1 14:1 19:1 22:1 23:1 24:1 42:1 53:1 56:1 62:1 66:1 71:1 74:1 76:1 77:1 78:1 81
-1 1:1 2:1 3:1 5:1 6:1 14:1 16:1 41:1 52:1 59:1 62:1 81:1 136:1 195:1 228:1 266:1 303:1 307:1 327:1
```

EXPECTED RESULT

The instance data matrix and label will be show in the Matlab workspace.



Executing the NMF multiplicative update algorithm

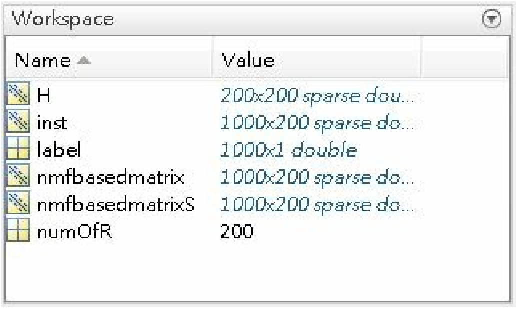
3 The step comprises three phases. First, a number of dimension should be set. The appropriate values can be selected based on the training data by performing step 3-5. Second, to generate the NMF-Based matrix, the NMF multiplicative update algorithm is performed. Third, the NMF-Based matrix is stored in the LibSVM file format.

```
>> numOfR=200 >> rng(1) >> [nmfbasedmatrix,H] = nnmf(inst, numOfR, 'algorithm','mult');  >> %Create sparse matrix >>
```

```
nmfbasedmatrixS = sparse(nmfbasedmatrix)    >> %write the matrix to the file by the libSVM format. >>
libsvmwrite('C:\NMF_Based_matrix\200.libsvm',label, nmfbasedmatrixS);
```

EXPECTED RESULT

A new matrix called the NMF-based matrix has been generated and displayed in the matlab workspace.



The NMF-based matrix is stored in the SVM file format.

An example of the NMF-based matrix with the LibSVM format.

Obtaining the SVM parameters by using the grid search method with stratified ten-fold cross-validation.

4 In order to obtain more suitable SVM parameters, based on stratified ten-fold cross-validation, the grid search method that derived by the LibSVM is utilized.

```
c:\>python C:\libsvm-3.22\tools\grid.py C:\NMF_Based_matrix\200.libsvm -v 10
```

EXPECTED RESULT

In this example, the suitable parameters values of C and g are 2 and 0.03125, respectively.

```
[local] 1 -15 46.4 (best c=2.0, g=0.03125, rate=76.5)
[local] 1 3 49.4 (best c=2.0, g=0.03125, rate=76.5)
[local] 1 -9 72.5 (best c=2.0, g=0.03125, rate=76.5)
[local] 5 -3 71.3 (best c=2.0, g=0.03125, rate=76.5)
[local] -1 -3 73.4 (best c=2.0, g=0.03125, rate=76.5)
[local] 11 -3 66.0 (best c=2.0, g=0.03125, rate=76.5)
[local] -3 -3 66.0 (best c=2.0, g=0.03125, rate=76.5)
[local] 9 -3 66.8 (best c=2.0, g=0.03125, rate=76.5)
[local] 3 -3 72.1 (best c=2.0, g=0.03125, rate=76.5)
[local] 15 -3 63.5 (best c=2.0, g=0.03125, rate=76.5)
[local] -5 -3 60.0 (best c=2.0, g=0.03125, rate=76.5)
[local] 7 -3 70.0 (best c=2.0, g=0.03125, rate=76.5)
[local] 1 -3 73.9 (best c=2.0, g=0.03125, rate=76.5)
[local] 13 -7 67.6 (best c=2.0, g=0.03125, rate=76.5)
[local] 13 -1 61.0 (best c=2.0, g=0.03125, rate=76.5)
[local] 13 -13 73.7 (best c=2.0, g=0.03125, rate=76.5)
[local] 13 1 51.9 (best c=2.0, g=0.03125, rate=76.5)
[local] 13 -11 73.6 (best c=2.0, g=0.03125, rate=76.5)
[local] 13 -5 65.7 (best c=2.0, g=0.03125, rate=76.5)
[local] 13 -15 72.5 (best c=2.0, g=0.03125, rate=76.5)
[local] 13 3 49.8 (best c=2.0, g=0.03125, rate=76.5)
[local] 13 -9 71.2 (best c=2.0, g=0.03125, rate=76.5)
[local] 13 -3 63.5 (best c=2.0, g=0.03125, rate=76.5)
2.0 0.03125 76.5
```

[local] 5 -7 74.2 (best c=32.0, g=0.0078125, rate=74.2)
[local] -1 -7 71.9 (best c=32.0, g=0.0078125, rate=74.2)
[local] 5 -1 62.9 (best c=32.0, g=0.0078125, rate=74.2)
[local] -1 -1 57.5 (best c=32.0, g=0.0078125, rate=74.2)
[local] 11 -7 69.2 (best c=32.0, g=0.0078125, rate=74.2)
[local] 11 -1 61.0 (best c=32.0, g=0.0078125, rate=74.2)
[local] 5 -13 72.4 (best c=32.0, g=0.0078125, rate=74.2)
[local] -1 -13 46.4 (best c=32.0, g=0.0078125, rate=74.2)
[local] 11 -13 72.7 (best c=32.0, g=0.0078125, rate=74.2)
[local] -3 -7 57.2 (best c=32.0, g=0.0078125, rate=74.2)
[local] -3 -1 54.9 (best c=32.0, g=0.0078125, rate=74.2)
[local] -3 -13 46.4 (best c=32.0, g=0.0078125, rate=74.2)
[local] 5 1 54.3 (best c=32.0, g=0.0078125, rate=74.2)
[local] -1 1 50.3 (best c=32.0, g=0.0078125, rate=74.2)
[local] 11 1 51.9 (best c=32.0, g=0.0078125, rate=74.2)
[local] -3 1 46.9 (best c=32.0, g=0.0078125, rate=74.2)
[local] 9 -7 71.6 (best c=32.0, g=0.0078125, rate=74.2)
[local] 9 -1 60.9 (best c=32.0, g=0.0078125, rate=74.2)
[local] 9 -13 75.7 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 9 1 51.9 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 5 -11 75.3 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] -1 -11 46.4 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 11 -11 73.9 (best c=512.0, g=0.0001220703125, rate=75.7)

```
[local] -3 -11 46.4 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 9 -11 73.3 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 3 -7 75.1 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 3 -1 63.7 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 3 -13 60.0 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 3 1 55.2 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 3 -11 72.7 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 5 -5 72.9 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] -1 -5 75.2 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 11 -5 68.2 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] -3 -5 70.8 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 9 -5 68.7 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 3 -5 74.2 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 15 -7 66.1 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 15 -1 61.0 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 15 -13 72.9 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 15 1 51.9 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 15 -11 72.2 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 15 -5 64.1 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 5 -15 60.0 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] -1 -15 46.4 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 11 -15 75.6 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] -3 -15 46.4 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 9 -15 75.3 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 3 -15 46.4 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 15 -15 73.7 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] -5 -7 46.4 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] -5 -1 46.4 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] -5 -13 46.4 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] -5 1 46.4 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] -5 -11 46.4 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] -5 -5 47.9 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] -5 -15 46.4 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 5 3 49.6 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] -1 3 48.0 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 11 3 49.8 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] -3 3 46.4 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 9 3 49.8 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 3 3 49.4 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 15 3 49.8 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] -5 3 46.4 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 7 -7 73.5 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 7 -1 61.4 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 7 -13 75.2 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 7 1 52.1 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 7 -11 75.7 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 7 -5 71.4 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 7 -15 72.3 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 7 3 49.8 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 5 -9 75.7 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] -1 -9 59.5 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 11 -9 73.0 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] -3 -9 46.4 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 9 -9 73.6 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 3 -9 75.3 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 15 -9 68.6 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] -5 -9 46.4 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 7 -9 73.7 (best c=512.0, g=0.0001220703125, rate=75.7)
```

```
[local] 1 -7 75.3 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 1 -1 63.8 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 1 -13 46.4 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 1 1 55.2 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 1 -11 60.0 (best c=512.0, g=0.0001220703125, rate=75.7)
[local] 1 -5 76.5 (best c=2.0, g=0.03125, rate=76.5)
[local] 1 -15 46.4 (best c=2.0, g=0.03125, rate=76.5)
[local] 1 3 49.4 (best c=2.0, g=0.03125, rate=76.5)
[local] 1 -9 72.5 (best c=2.0, g=0.03125, rate=76.5)
[local] 5 -3 71.3 (best c=2.0, g=0.03125, rate=76.5)
[local] -1 -3 73.4 (best c=2.0, g=0.03125, rate=76.5)
[local] 11 -3 66.0 (best c=2.0, g=0.03125, rate=76.5)
[local] -3 -3 66.0 (best c=2.0, g=0.03125, rate=76.5)
[local] 9 -3 66.8 (best c=2.0, g=0.03125, rate=76.5)
[local] 3 -3 72.1 (best c=2.0, g=0.03125, rate=76.5)
[local] 15 -3 63.5 (best c=2.0, g=0.03125, rate=76.5)
[local] -5 -3 60.0 (best c=2.0, g=0.03125, rate=76.5)
[local] 7 -3 70.0 (best c=2.0, g=0.03125, rate=76.5)
[local] 1 -3 73.9 (best c=2.0, g=0.03125, rate=76.5)
[local] 13 -7 67.6 (best c=2.0, g=0.03125, rate=76.5)
[local] 13 -1 61.0 (best c=2.0, g=0.03125, rate=76.5)
[local] 13 -13 73.7 (best c=2.0, g=0.03125, rate=76.5)
[local] 13 1 51.9 (best c=2.0, g=0.03125, rate=76.5)
[local] 13 -11 73.6 (best c=2.0, g=0.03125, rate=76.5)
[local] 13 -5 65.7 (best c=2.0, g=0.03125, rate=76.5)
[local] 13 -15 72.5 (best c=2.0, g=0.03125, rate=76.5)
[local] 13 3 49.8 (best c=2.0, g=0.03125, rate=76.5)
[local] 13 -9 71.2 (best c=2.0, g=0.03125, rate=76.5)
[local] 13 -3 63.5 (best c=2.0, g=0.03125, rate=76.5)
2.0 0.03125 76.5
```

Effectiveness Evaluation on the SVM-based disease risk assessment model

5 For Effectiveness Evaluation on the SVM-based disease risk assessment model, the stratified ten-fold cross-validation is conducted by the LibSVM.

```
C:\NMF_Based_matrix>c:\libsvm-3.22\windows\svm-train.exe -v 10 -c 2 -g 0.03125 testing-1.libsvm
```

EXPECTED RESULT

The execution result will be shown finally.

```
AUC = 0.82188
Accuracy = 74.3% (743/1000)
AP = 0.834817
BAC = 0.743
Sensitivity = 0.74 (370/(370+130))
Specificity = 0.746 (373/(373+127))
Cross Validation = 74.3%

C:\NMF_Based_matrix>_
```