Homework Assignment 3

zhanwang Chen

1.Dimensionality Reduction
(a).

A CUR decomposition is a set of three matrices C, U, R that, when multiplied together, closely approximate a given matrix. This is similar to SVD. It is used to deal with one disadvantage of SVD - density of orthogonal matrices U and V. If we have matrices with millions of rows and columns, SVD decomposition becomes almost impossible. In CUR, matrices C and R (analogous to U and V and constructed by picking up random amount of columns or rows respectively. The probability of picking a row or column is determined by the ratio of Frobenius norm of a vector to the Frobenius norm of the initial matrix) are sparse, which makes computational task simpler. The third matrix, U, which stays between C and R, is constructed by a pseudo-inverse of the intersection of the chosen rows and columns.How the 3 matrices C,U,R is calculated is given below:
Let M be a matrix of m rows and n columns. Pick a target number of “concepts” r to be used in the decomposition. A CUR-decomposition of M is a randomly chosen set of r columns of M , which form the m x r matrix C , and a randomly chosen set of r rows of M , which form the r x n matrix R . There is also an r x r matrix U that is constructed from C and R as follows:
1. Let W be the r x r matrix that is the intersection of the chosen columns of C and the chosen rows of R .
2. Compute the SVD of W ; say W = X Σ T .
3. Compute Σ + , the Moore-Penrose pseudoinverse of the diagonal matrix .
4. Let U = Y Σ (+) 2 XT .

(b).

	Matrix	Alien	Star Wars	Casablanca	Titanic
Joe	1	1	1	0	0
Jim	3	3	3	0	0
John	4	4	4	0	0
Jack	5	5	5	0	0
Jill	0	0	0	4	4
Jenny	0	0	0	5	5
Jane	0	0	0	2	2

Figure 11.12: Matrix M, repeated from Fig. 11.6

$$M = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \\ 0 & 0 & 0 & 2 & 2 \end{bmatrix}$$

sum of squares of elements : 243

a) Matrix and Alien

squared Frobenius norm : $1^2 + 3^2 + 4^2 + 5^2 = 51$

Matrix P_1 = Alien $P_2 = \frac{51}{243} = 0.21$

after division by $\sqrt{2 \times 0.21} = 0.648$

Matrix and Alien :

$$C: \begin{bmatrix} 1.54 & 1.54 \\ 4.63 & 4.63 \\ 6.17 & 6.17 \\ 7.72 & 7.72 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Row Jim : squared Frobenius norm = 27, $P_3 = 0.111$

Row John : squared Frobenius norm = 48 $P_4 = 0.198$

after the division by $\sqrt{(2\times 0.111)} = 0.471$ on Jim

$\sqrt{2\times 0.198} = 0.629$ on John,

matrix $\underline{R}$ is $\begin{pmatrix} 6.37 & 6.37 & 6.37 & 0 & 0 \\ 6.36 & 6.36 & 6.36 & 0 & 0 \end{pmatrix}$

$W = \begin{pmatrix} 3 & 3 \\ 4 & 4 \end{pmatrix}$

$\cancel{U = X}$ $U = Y(\Sigma^+)^2 X^T$

SVD Result: $W = X\Sigma Y^T$

$$W : \begin{pmatrix} -0.60 & -0.80 \\ 0.80 & 0.60 \end{pmatrix} \begin{pmatrix} 7.07 & 0.00 \\ 0.00 & 0.00 \end{pmatrix} \begin{pmatrix} -0.71 & -0.71 \\ -0.71 & -0.71 \end{pmatrix}$$

$$\Sigma^+ : \begin{pmatrix} 0.141 & 0.000 \\ 0.000 & 0.000 \end{pmatrix} \quad \Sigma^+ : \begin{pmatrix} 0.020 & 0.000 \\ 0.000 & 0.000 \end{pmatrix}$$

$$U = Y(\Sigma^+)^2 X^T = \begin{pmatrix} -0.71 & -0.71 \\ -0.71 & 0.71 \end{pmatrix} \begin{pmatrix} 0.020 & 0.000 \\ 0.000 & 0.000 \end{pmatrix} \begin{pmatrix} -0.60 & -0.80 \\ -0.80 & 0.60 \end{pmatrix}$$

$$= \begin{pmatrix} 0.009 & 0.011 \\ 0.009 & 0.011 \end{pmatrix}$$

$$CUR = \begin{pmatrix} 0.39 & 0.39 & 0.39 & 0 & 0 \\ 1.72 & 1.72 & 1.72 & 0 & 0 \\ 1.562 & 1.36 & 1.56 & 0 & 0 \\ 1.934 & 1.95 & 1.95 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

b) Alien and star wars

squared Frobenius norm

$1^2+3^2+4^2+5^2=51$

$p_1 = p_2 = 51/243 = 0.21$

$\sqrt{rp_1} = \sqrt{rp_2} = \sqrt{2\times0.21} = 0.648$

$$C = \begin{pmatrix} 1.54 & 1.54 \\ 4.63 & 4.63 \\ 6.17 & 6.17 \\ 7.72 & 7.72 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

Jack $p_3 = 0.309$

Jill $p_4 = 0.132$

$$R = \begin{pmatrix} 6.36 & 6.36 & 6.36 & 0 & 0 \\ 0 & 0 & 0 & 7.78 & 7.78 \end{pmatrix}$$

$$W = \begin{pmatrix} 5 & 5 \\ 0 & 0 \end{pmatrix} \Rightarrow SVD \quad W = \underset{X}{\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}} \underset{\Sigma}{\begin{pmatrix} 7.071 & 0 \\ 0.00 & 0 \end{pmatrix}} \underset{Y^T}{\begin{pmatrix} 0.707 & 0.707 \\ -0.707 & 0.707 \end{pmatrix}}$$

$$U = Y(\Sigma^+)^2 X^T = \begin{pmatrix} 0.014 & 0 \\ 0.014 & 0 \end{pmatrix}$$

(b) CUR:

0.2742	0.2742	0.2742	0	0
0.8245	0.8245	0.8245	0	0
1.0988	1.0988	1.0988	0	0
1.3748	1.3748	1.3748	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

(c)
Matrix
sf Norm = 51
p1 = 51/243=0.210

Titannic:
sf Norm =45
p2=45/243=0.185

c=

1.5430	0
4.6300	0
6.1730	0
7.7160	0
0	6.5790
0	8.2240
0	3.2890

Row Joe
norn =3
p3 = 0.012

Row Jane
p4=0.033

R=

6.4520	6.4520	6.4520	0	0
0	0	0	7.7820	7.7820

U=

1.0000	0
0	0.2500

CUR =

9.9361	9.9361	9.9361	2.9961	2.9961
29.8728	29.8728	29.8728	9.0077	9.0077
39.8088	39.8088	39.8088	12.0037	12.0037
49.8094	49.8094	49.8094	15.0193	15.0193
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

2. Item Based Collaborative Filtering

User	Movie A	Movie B	Movie C	Movie D
Ryan	1	5	3	5
Stavros			2	
Brahma		4	1	1
Brodie	4	3		2
Zosimus		5	1	

(a) Compute the item similarity matrix (denoted by R) using Pearson Correlation as a measure.

$$r_{AB} = \frac{(1-2.5)(5-4)+(4-2.5)(3-4)}{\sqrt{(1-2.5)^2+(4-2.5)^2}\sqrt{(5-4)^2+(3-4)^2}} = -1$$

$$r_{AD} = \frac{(1-2.5)(5-3.5)+(4-2.5)(2-3.5)}{\sqrt{(1-2.5)^2+(4-2.5)^2}\sqrt{(5-3.5)^2+(2-3.5)^2}} = -1$$

$$r_{BC} = \frac{(5-4.67)(3-1.67)+(4-4.67)(1-1.67)+(5-4.67)(1-1.67)}{\sqrt{(5-4.67)^2+(4-4.67)^2+(5-4.67)^2}\sqrt{(3-1.67)^2+(1-1.67)^2+(1-1.67)^2}} = 0.5$$

$$r_{BD} = \frac{(5-4)(5-2.67)+(4-4)(1-2.67)+(3-4)(2-2.67)}{\sqrt{(5-4)^2+(4-4)^2+(3-4)^2}\sqrt{(5-2.67)^2+(1-2.67)^2+(2-2.67)^2}} = 0.72$$

$$r_{CD} = \frac{(3-2)(5-3)+(1-2)(1-3)}{\sqrt{(3-2)^2+(1-2)^2}\sqrt{(5-3)^2+(1-3)^2}} = 1$$

	Movie A	Movie B	Movie C	Movie D
Movie A	-	-1	-	-1
Movie B	-1	-	0.5	0.72
Movie C	-	0.5	-	1
Movie D	-1	0.72	1	-

(b) Compute the item similarity matrix (denoted by S) using Jaccard Coefficient as a measure.

$$J_{AB} = \frac{|A \cap B|}{A \cup B} = \frac{2}{4} = 0.5$$

$$J_{AC} = \frac{1}{5} = 0.2$$

$$\mathrm{J}_{AD} = \frac{2}{3} = 0.67$$

$$J_{BC} = \frac{3}{5} = 0.6$$

$$J_{BD} = \frac{3}{4} = 0.75$$

$$J_{CD} = \frac{2}{5} = 0.4$$

	Movie A	Movie B	Movie C	Movie D
Movie A	-	0.5	0.2	0.67
Movie B	0.5	-	0.6	0.75
Movie C	0.2	0.6	-	0.4
Movie D	0.67	0.75	0.4	-

(c) Use matrices R and S to compute the corresponding ratings that Zosimus would give to Movies A & D.

Pearson matrices R
Movies A

$$\frac{r_{AB} rating_{Zosimus,B} + r_{AC} rating_{Zosimus,C}}{|r_{AB}| + |r_{AC}|} = \frac{-1 \cdot 5}{1} = -5$$

Movies D

$$= \frac{0.72 \cdot 5 + 1 \cdot 1}{0.72 + 1} = 2.67$$

Jaccard
matrices S
Movies A

$$\frac{J_{AB} rating_{Zosimus,B} + J_{AC} rating_{Zosimus,C}}{|J_{AB}| + |J_{AC}|} = \frac{0.5 \cdot 5 + 0.2 \cdot 1}{0.5 + 0.2} = 3.86$$

Movies D

$$= \frac{0.75 \cdot 5 + 0.4 \cdot 1}{0.75 + 0.4} = 3.6$$

(d) Use matrices R and S to compute the corresponding ratings that Stavros would give to Movies A & B.

Pearson matrices R
Movies A =0 or unavailable.

Movies B

$$= \frac{0.5 \cdot 2}{0.5} = 2$$

Jaccard matrices S
Movies A

$$= \frac{0.2 \cdot 2}{0.2} = 2$$

Movies B

$$= \frac{0.6 \cdot 2}{0.6} = 2$$

3. Network Analysis

(a) $V' = MV$

$$M = \begin{bmatrix} \frac{1}{3} & \frac{1}{2} & 0 \\ \frac{1}{3} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

initial page rank vector V_0 is $\begin{bmatrix} \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{bmatrix}$

$$V_1 = MV_0 = \begin{bmatrix} \frac{1}{9}+\frac{1}{6} \\ \frac{1}{9}+\frac{1}{6} \\ \frac{1}{9}+\frac{1}{9}+\frac{1}{6} \end{bmatrix} = \begin{bmatrix} \frac{5}{18} \\ \frac{5}{18} \\ \frac{5}{18} \end{bmatrix} = \begin{bmatrix} 0.2777 \\ 0.2777 \\ 0.4444 \end{bmatrix}$$

$$V_2 = MV_1 = \begin{bmatrix} 0.2714 \\ 0.3148 \\ 0.4537 \end{bmatrix}$$

$$V_3 = MV_2 = \begin{bmatrix} 0.2345 \\ 0.3040 \\ 0.4614 \end{bmatrix}$$

$V_4 \approx V_3$

Converged at $\begin{bmatrix} 0.23 \\ 0.30 \\ 0.46 \end{bmatrix}$

(b) $V' = \beta M v + (1-\beta)\, e/n \qquad \beta = 0.8$

$$M = \begin{bmatrix} \frac{1}{3} & \frac{1}{2} & 0 \\ \frac{1}{3} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \qquad e/n = \begin{bmatrix} \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{bmatrix}$$

$$V' = \begin{bmatrix} \frac{4}{15} & \frac{2}{5} & 0 \\ \frac{4}{15} & 0 & \frac{2}{5} \\ \frac{4}{15} & \frac{2}{5} & \frac{2}{5} \end{bmatrix} V + \begin{bmatrix} \frac{1}{15} \\ \frac{1}{15} \\ \frac{1}{15} \end{bmatrix}$$

initial page rank vector

$$V: \begin{bmatrix} \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{bmatrix}$$

$$V_1 = \begin{bmatrix} \frac{4}{15} & \frac{2}{5} & 0 \\ \frac{4}{15} & 0 & \frac{2}{5} \\ \frac{4}{15} & \frac{2}{5} & \frac{2}{5} \end{bmatrix} \begin{bmatrix} \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{bmatrix} + \begin{bmatrix} \frac{1}{15} \\ \frac{1}{15} \\ \frac{1}{15} \end{bmatrix} = \begin{bmatrix} \frac{4}{15} \cdot \frac{1}{3} + \frac{2}{5} \cdot \frac{1}{3} + \frac{1}{15} \\ \frac{4}{15} \cdot \frac{1}{3} + \frac{2}{5} \cdot \frac{1}{3} + \frac{1}{15} \\ \frac{4}{15} \cdot \frac{1}{3} + \frac{2}{5} \cdot \frac{1}{3} + \frac{2}{5} \cdot \frac{1}{3} + \frac{1}{15} \end{bmatrix} = \begin{bmatrix} \frac{13}{45} \\ \frac{13}{45} \\ \frac{19}{45} \end{bmatrix}$$

$$V_1 = \begin{bmatrix} 0.29 \\ 0.29 \\ 0.42 \end{bmatrix}$$

$= \cancel{0.29}$ $\cancel{0.29}$

$$V_2 = \beta M V_1 + (1-\beta)\, e/n = \begin{bmatrix} 0.26 \\ 0.31 \\ 0.43 \end{bmatrix} \qquad V_3 = \begin{bmatrix} 0.26 \\ 0.30 \\ 0.43 \end{bmatrix} \approx V_4$$

coverged at

$$V_3 = \begin{bmatrix} 0.26 \\ 0.30 \\ 0.43 \end{bmatrix}$$

4. Support Vector Machines

a. SVMs is a class of Machine Learning algorithms, which basic concept is finding a hyperplane in a high dimensional hyperspace of features (those features were mapped to that hyperspace by a special map function φ) that divides data points into two classes based on a large margin criterion. SVMs are usually used for classification tasks.

b. SVMs indeed can be used for multiclass classifications. In fact, the procedure stays the same as for the two-class classification SVM. For example, in one-versus-all approach we assume that our available classes have been united into two: one class of interest and another consisting of all other classes. We have now the same classification task as in 2a. We need to build N such classifiers – one for each class in our dataset, and then combine all the decision boundaries. In one-vs-one approach, we build for each class N-1 classifiers (one for each other class). Thus we will have in total N(N-1)/2 classifiers. The latter approach is useful when dealing with imbalanced classes, but is more computationally expensive.

One-versus-all approach: construct set of binary classifiers
f 1 , f 2 , . . . f m , each trained to separate one class from rest. This strategy involves training a single classifier per class, with the samples of that class as positive samples and all other samples as negatives.
one-against-one approach: In the one-vs.-one (OvO) reduction, one trains K (K − 1) / 2 binary classifiers for a K-way multiclass problem; each receives the samples of a pair of classes from the original training set, and must learn to distinguish these two classes. At prediction time, a voting scheme is applied: all K (K − 1) / 2 classifiers are applied to an unseen sample and the class that got the highest number of "+1" predictions gets predicted by the combined classifier.

c. Running time of the one-against-rest approach is $O\left(kn^r\right)$ and of the one-against-one is

$$O\left(\frac{k(k-1)}{2}n^r\right)$$

5. Classification: Building a Spam Filter with Apache Spark

Test set accuracy = 0.8662477558348295

Term frequency-inverse document frequency (TF-IDF) is a feature vectorization method to reflect the importance of a term to a document in the corpus.

TF and IDF are implemented in HashingTF and IDF in spark.

```
val tf = new HashingTF(numFeatures = 100)
val spamFeatures = spam.map(email => tf.transform(email.split(" ")))
val hamFeatures = ham.map(email => tf.transform(email.split(" ")))
val spamFeaturest = spamtest.map(email => tf.transform(email.split(" ")))
val hamFeaturest = hamtest.map(email => tf.transform(email.split(" ")))

val idf  = new IDF().fit(spamFeatures++hamFeatures++spamFeaturest++hamFeaturest)
val spamFeaturesidf =idf.transform(hamFeatures)
val hamFeaturesidf =idf.transform(hamFeatures)
val spamFeaturestidf =idf.transform(spamFeaturest)
val hamFeaturestidf =idf.transform(hamFeaturest)
```

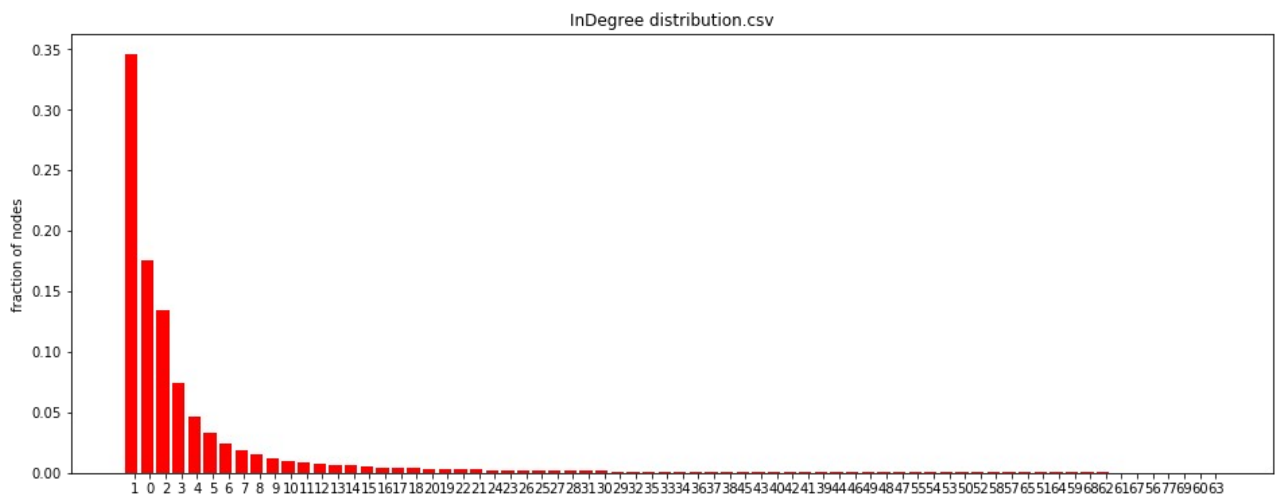
Naive Bayes is a simple multiclass classification algorithm with the assumption of independence between every pair of features. Naive Bayes can be trained very efficiently. Within a single pass to the training data, it computes the conditional probability distribution of each feature given label, and then it applies Bayes' theorem to compute the conditional probability distribution of label given an observation and use it for prediction.

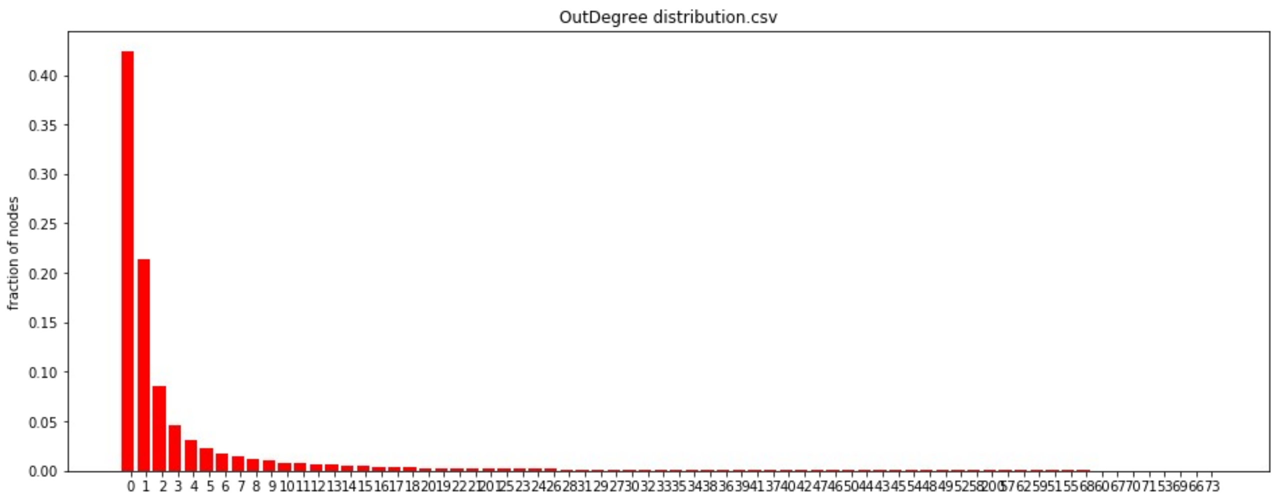
```
val model = NaiveBayes.train(trainingData, lambda = 1.0, modelType = "multinomial")

val predictionAndLabel = testData.map(p => (model.predict(p.features), p.label))
val accuracy = 1.0 * predictionAndLabel.filter(x => x._1 == x._2).count() / testData.count()
println("Test set accuracy = " + accuracy)
```

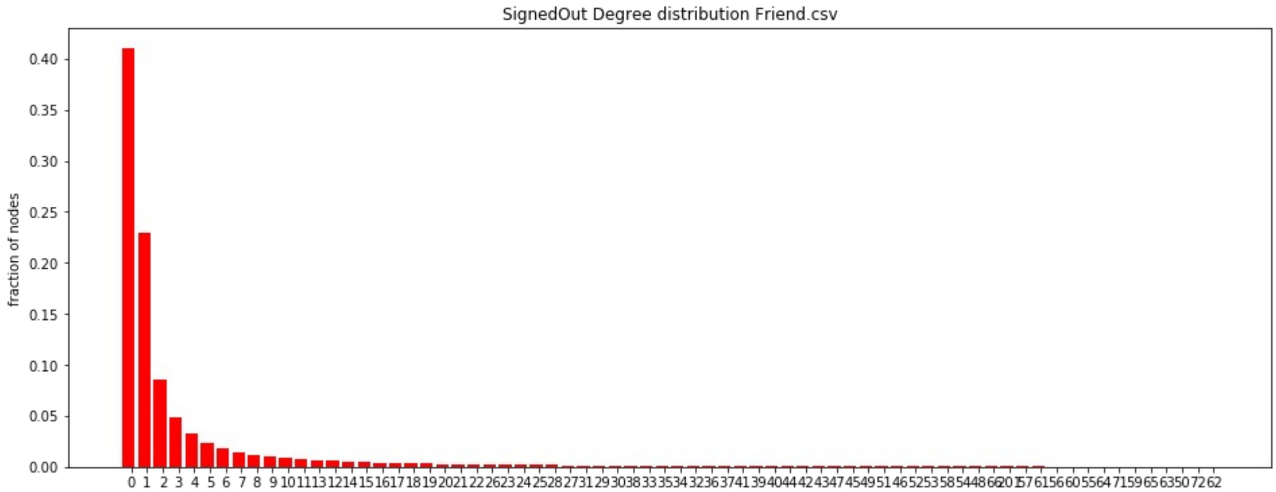
6. Network Statistics in Apache Flink

AverageFriendFoeRatio=2.85836





(c) Nearly half of the users don’t rate anyone as Friend , about 23% of the users have 1 Friend. The degree distribution is long tail.



(d) More than half of the users don’t rate anyone as foe, about 20% of the users have 1 foe. The degree distribution is long tail.