目录

一、聚类（organized by TW） 3

1.1 DBSCAN 3

1.2 EM 3

1.3 FarestFirst 4

1.4 HierarchicalClusterer 4

1.5 KMeans 5

二、分类（organized by TW） 5

2.1 J48 5

2.2 JRip 6

2.3 LinearRegression 6

2.4 M5P 6

2.5 NaiveBayes 7

2.6 RandomTree 7

2.7 RandomForest 8

三、关联规则（organized by TW） 8

3.1 Apriori 8

3.2 PredictiveApriori 9

四、时序预测（organized by ZX） 9

4.1 ARIMA 9

4.2 ES 10

4.3 BPNN 10

4.4 SVR 10

五、组合建模（organized by GC） 11

六、集成学习（organized by GC） 11

七、个性化推荐（organized by WJK） 11

7.1 SlopeOne 11

7.2 GlobalAverage 11

7.3 ItemAverage 12

7.4 UserAverage 13

7.5 UserItemBaseline 13

7.6 ItemKNNCosine 14
7.7 ItemKNNPearson 14
7.8 UserKNNCosine 15
7.9 UserKNNPearson 16
7.10 BPNN 16
7.11 LogisticRegressionMatrixFactorization 17
7.12 CoClustering 18
7.13 UserKNN 18
7.14 ItemKNN 19
7.15 BPRMF 20
7.16 WRMF 20
7.17 MostPopular 21
7.18 Random 22
八、异常检测（organized by LZ） 22
8.1 基于统计模型的离群点检测方法 StatisOutlierDetect 22
8.2 基于 KNN 的离群点检测方法 KNNOutlierDetect 23
8.3 基于密度的局部离群点检测方法 LofOutlierDetect 24
8.4 基于 k-means 的离群点检测方法 KMeansOutlierDetect 24
九、文本挖掘（organized by QY） 25
十、中文分词（organized by QY） 25
十一、词干抽取（organized by QY） 25

一、聚类（organized by TW）

1.1 DBSCAN

1.1.1 工具：WEKA

1.1.2 语言：Java

1.1.3 参数

参数	默认值	含义	备注
e	0.3	领域半径	
minpoints	4	领域最小点数	
input	D:// data//weather.numeric.arff"	输入文件位置	
output	D:// output//output.txt	输出文件位置	

1.1.4 输入数据格式

@数据名称@属性{属性值}@数据值

样例：iris.2D.arff

1.1.5 运行方法

java –classpath J.jar edu.hfut.cluster.mian.WK_DBSCAN -e 0.3 –minpoints 4

-input D:// data//weather.numeric.arff" –output D:// output//output.txt

输出：output.txt

1.2 EM

1.2.1 工具：EM

1.2.2 语言：WEKA

1.2.3 参数

参数	默认值	含义	备注
n	2	生成类数	
input	D:// data//weather.numeric.arff"	输入文件位置	
output	D:// output//output.txt	输出文件位置	

1.2.4 输入数据格式

@数据名称@属性{属性值}@数据值

样例：weather.norminal.arff

1.2.5 运行方法

java –classpath J.jar edu.hfut.cluster.mian.WK_EM：-n 2

-input D:// data//weather.numeric.arff" –output D:// output//output.txt

输出：output.txt

1.3 FarestFirst

1.3.1 工具：WEKA
1.3.2 语言：Java
1.3.3 参数

参数	默认值	含义	备注
n	2	生成类数	
input	D:// data//weather.numeric.arff"	输入文件位置	
output	D:// output//output.txt	输出文件位置	

1.3.4 输入数据格式
@数据名称@属性{属性值}@数据值
样例：weather.numeric.arff
1.3.5 运行方法
java –classpath J.jar edu.hfut.cluster.mian.WK_FarestFirst：-n 2
-input D:// data//weather.numeric.arff" –output D:// output//output.txt
输出：output.txt

1.4 HierarchicalClusterer

1.4.1 工具：WEKA
1.4.2 语言：Java
1.4.3 参数

参数	默认值	含义	备注
n	2	生成类数	
input	D:// data//weather.numeric.arff"	输入文件位置	
output	D:// output//output.txt	输出文件位置	

1.4.4 输入数据格式
@数据名称@属性{属性值}@数据值
样例：iris.2D.arff
1.4.5 运行方法
java –classpath J.jar edu.hfut.cluster.mian.WK_HierarchicalClusterer -n 2
-input D:// data//weather.numeric.arff" –output D:// output//output.txt
输出：output.txt

1.5 KMeans

1.5.1 工具：WEKA
1.5.2 语言：Java
1.5.3 参数

参数	默认值	含义	备注
k	2	初始点数	
input	D:// data//weather.numeric.arff"	输入文件位置	
output	D:// output//output.txt	输出文件位置	

1.5.4 输入数据格式
@数据名称@属性{属性值}@数据值
样例：：weather.numeric.arff
1.5.5 运行方法
java –classpath J.jar edu.hfut.cluster.mian.WK_KMeans -k 2
-input D:// data//weather.numeric.arff" –output D:// output//output.txt
输出：output.txt

二、分类（organized by TW）

2.1 J48

2.1.1 工具：WEKA
2.1.2 语言：Java
2.1.3 参数

参数	默认值	含义	备注
n	2	生成类数	
input	D:// data//weather.nominal.arff"	输入文件位置	
output	D:// output//output.txt	输出文件位置	

2.1.4 输入数据格式
@数据名称@属性{属性值}@数据值
样例：：weather.norminal.arff
2.1.5 运行方法
Weka - src – Test – WK_J48 #打开包所在目录
java –classpath J.jar edu.hfut.classify.mian.WK_J48 -n 2
-input D:// data//weather.numeric.arff" –output D:// output//output.txt

输出：output.txt

2.2 JRip

2.2.1 工具：WEKA
2.2.2 语言：Java
2.2.3 参数

参数	默认值	含义	备注
minno	2	样本最小权重值和	
input	D:// data//weather.nominal.arff"	输入文件位置	
output	D:// output//output.txt	输出文件位置	

2.2.4 输入数据格式
@数据名称@属性{属性值}@数据值
样例：weather.norminal.arff
2.2.5 运行方法
java –classpath J.jar edu.hfut.classify.mian.WK_JRip -minno 2
-input D:// data//weather.numeric.arff" –output D:// output//output.txt
输出：output.txt

2.3 LinearRegression

2.3.1 工具：WEKA
2.3.2 语言：Java
2.3.3 参数：
2.3.4 输入数据格式
@数据名称@属性{属性值}@数据值
样例：weather.norminal.arff
2.3.5 运行方法
java –classpath J.jar edu.hfut.classify.mian.WK_LinearRegression
-input D:// data//weather.numeric.arff" –output D:// output//output.txt
输出：output.txt

2.4 M5P

2.4.1 工具：WEKA
2.4.2 语言：Java
2.4.3 参数

参数	默认值	含义	备注
minIns	4	叶节点的最小样	

		本数	
input	D:// data//weather.nominal.arff"	输入文件位置	
output	D:// output//output.txt	输出文件位置	

2.4.4 输入数据格式
@数据名称@属性{属性值}@数据值
样例：cpu.arff
2.4.5 运行方法
java –classpath J.jar edu.hfut.classify.mian.WK_M5P -minIns 4
-input D:// data//weather.numeric.arff" –output D:// output//output.txt
输出：output.txt

2.5 NaiveBayes

2.5.1 工具：WEKA
2.5.2 语言：Java
2.5.3 参数：
2.5.4 输入数据格式
@数据名称@属性{属性值}@数据值
样例：weather.norminal.arff
2.5.5 运行方法
java –classpath J.jar edu.hfut.classify.mian.WK_NaiveBayes
-input D:// data//weather.numeric.arff" –output D:// output//output.txt
输出：output.txt

2.6 RandomTree

2.6.1 工具：WEKA
2.6.2 语言：Java
2.6.3 参数

参数	默认值	含义	备注
minnum	1.0	叶节点权重	
input	D:// data//weather.nominal.arff"	输入文件位置	
output	D:// output//output.txt	输出文件位置	

2.6.4 输入数据格式
@数据名称@属性{属性值}@数据值
样例：weather.norminal.arff
2.6.5 运行方法
java –classpath J.jar edu.hfut.classify.mian.WK_RandomTree -minnum 1.0

-input D:// data//weather.numeric.arff" –output D:// output//output.txt
输出：output.txt

2.7 RandomForest

2.7.1 工具：WEKA
2.7.2 语言：Java
2.7.3 参数

参数	默认值	含义	备注
numtree	5	生成树数	
numfeature	1	选择的属性数	
input	D:// data//weather.nominal.arff"	输入文件位置	
output	D:// output//output.txt	输出文件位置	

2.7.4 输入数据格式
@数据名称@属性{属性值}@数据值
样例：weather.norminal.arff
2.7.5 运行方法
java –classpath J.jar edu.hfut.classify.mian.WK_RandomForest -numtree 5 –numfeature 1 -input D:// data//weather.numeric.arff" –output D:// output//output.txt
输出：output.txt

三、关联规则（organized by TW）

3.1 Apriori

3.1.1 工具：WEKA
3.1.2 语言：Java
3.1.3 参数

参数	默认值	含义	备注
delta	**0.05**	支持度减少值	
lbminsupport	**0.1**	最高支持度	
ubminsupport	**1.0**	最低支持度	
nRules	**10**	规则数	
minMet	**0.5**	最小置信度	

input	D:// data//weather.nominal.arff"	输入文件位置	
output	D:// output//output.txt	输出文件位置	

3.1.4 输入数据格式

@数据名称@属性{属性值}@数据值

样例：weather.norminal.arff

3.1.5 运行方法

java –classpath J.jar edu.hfut.associationrule.mian.WK_Apriori –delta 0.05 –lbminsupport 0.1 –ubminsupport 1.0 –nRules 10 –minMet 0.5 -input D:// data//weather.numeric.arff" output D:// output//output.txt

输出：output.txt

3.2 PredictiveApriori

3.2.1 工具：WEKA

3.2.2 语言：Java

3.2.3 参数

参数	默认值	含义	备注
nRules	10	规则数	
Input	D:// data//weather.nominal.arff"	输入文件位置	
Output	D:// output//output.txt	输出文件位置	

3.2.4 输入数据格式

@数据名称@属性{属性值}@数据值

样例：weather.norminal.arff

3.2.5 运行方法

java –classpath J.jar edu.hfut.associationrule.mian.WK_PredictiveApriori –nRules 10 -input D:// data//weather.numeric.arff" –output D:// output//output.txt

输出：output.txt

四、时序预测（organized by ZX）

4.1 ARIMA

4.1.1 工具：ARIMA

4.1.2 语言：Java

4.1.3 参数：无

4.1.4 输入数据格式

历史数据
样例：testdata.txt
4.1.5 运行方法

4.2 ES

4.2.1 工具：ES
4.2.2 语言：Java
4.2.3 参数：无
4.2.4 输入数据格式
历史数据
样例：testdata.txt
4.2.5 运行方法

4.3 BPNN

4.3.1 工具：BPNN
4.3.2 语言：Java
4.3.3 参数：层数，神经元个数，梯度下降速率（学习速率）
4.3.4 输入数据格式
训练集；测试集
样例：traindata.txt；testdata.txt
4.3.5 运行方法

4.4 SVR

4.4.1 工具：SVR
4.4.2 语言：Java
4.4.3 参数：核函数、迭代次数、激活速率
4.4.4 输入数据格式
训练集；测试集
样例：traindata.txt；testdata.txt
4.4.5 运行方法

五、组合建模（organized by GC）

六、集成学习（organized by GC）

七、个性化推荐（organized by WJK）

7.1 SlopeOne

7.1.1 工具：MyMediaLite

7.1.2 语言：Java

7.1.3 参数

参数	默认值	含义	备注
--training-file	不能为空	训练数据	
--test-file	无	测试数据	
--test-ratio	无	测试数据为空时，训练数据分割比率	若 test-file 不为空，则 test-ratio 为空
--data-dir	无	文件根目录	
--recommender	SlopeOne	评分预测方法	SlopeOne 算法消耗内存较大，有时会导致内存不足

7.1.4 输入数据格式

第一列：userid，表示用户索引

第二列：itemid，表示产品索引

第三列：rating，表示用户对产品的评分

样例：sample-data-rating.txt

7.1.5 运行方法

```
cd /usr/local/ml            #打开所在包的目录
java -classpath mymedialite.jar edu.hfut.recommender.main.RatingPrediction –training file=ratings.txt --test-ratio=0.2 --data-dir=D:\data\movielens --recommender=SlopeOne
#运行
```

7.2 GlobalAverage

7.2.1 工具：MyMediaLite

7.2.2 语言：Java

7.2.3 参数

参数	默认值	含义	备注

--training-file	不能为空	训练数据	
--test-file	无	测试数据	
--test-ratio	无	测试数据为空时，训练数据分割比率	若 test-file 不为空，则 test-ratio 为空
--data-dir	无	文件根目录	
--recommender	GlobalAverage	评分预测方法	

7.2.4 输入数据格式

第一列：userid，表示用户索引

第二列：itemid，表示产品索引

第三列：rating，表示用户对产品的评分

样例：sample-data-ratings.txt

7.2.5 运行方法

cd /usr/local/ml　　#打开所在包的目录

java -classpath mymedialite.jar edu.hfut.recommender.main.RatingPrediction –training file=ratings.txt --test-ratio=0.2 --data-dir=D:\data\movielens --recommender=GlobalAverage #运行

7.3 ItemAverage

7.3.1 工具：MyMediaLite

7.3.2 语言：Java

7.3.3 参数

参数	默认值	含义	备注
--training-file	不能为空	训练数据	
--test-file	无	测试数据	
--test-ratio	无	测试数据为空时，训练数据分割比率	若 test-file 不为空，则 test-ratio 为空
--data-dir	无	文件根目录	
--recommender	ItemAverage	评分预测方法	

7.3.4 输入数据格式

第一列：userid，表示用户索引

第二列：itemid，表示产品索引

第三列：rating，表示用户对产品的评分

样例：sample-data-ratings.txt

7.3.5 运行方法

cd /usr/local/ml　　#打开所在包的目录

java -classpath mymedialite.jar edu.hfut.recommender.main.RatingPrediction --training-file=ratings.txt --test-ratio=0.2 --data-dir=D:\data\movielens --recommender=ItemAverage　#运行

7.4 UserAverage

7.4.1 工具：MyMediaLite

7.4.2 语言：Java

7.4.3 参数

参数	默认值	含义	备注
--training-file	不能为空	训练数据	
--test-file	无	测试数据	
--test-ratio	无	测试数据为空时，训练数据分割比率	若 test-file 不为空，则 test-ratio 为空
--data-dir	无	文件根目录	
--recommender	UserAverage	评分预测方法	

7.4.4 输入数据格式

第一列：userid，表示用户索引

第二列：itemid，表示产品索引

第三列：rating，表示用户对产品的评分

样例：sample-data-ratings.txt

7.4.5 运行方法

cd /usr/local/ml　　#打开所在包的目录

java -classpath mymedialite.jar edu.hfut.recommender.main.RatingPrediction --training-file=ratings.txt --test-ratio=0.2 --data-dir=D:\data\movielens --recommender=UserAverage　#运行

7.5 UserItemBaseline

7.5.1 工具：MyMediaLite

7.5.2 语言：Java

7.5.3 参数

参数	默认值	含义	备注
--training-file	不能为空	训练数据	
--test-file	无	测试数据	
--test-ratio	无	测试数据为空时，训练数据分割比率	若 test-file 不为空，则 test-ratio 为空
--data-dir	无	文件根目录	
--recommender	UserItemBaseline	评分预测方法	
regU	15	用户的正则化参数	
regI	10	项目的正则化参数	
numIter	10	迭代的次数	

7.5.4 输入数据格式

第一列：userid，表示用户索引

第二列：itemid，表示产品索引

第三列：rating，表示用户对产品的评分

样例：sample-data-ratings.txt

7.5.5 运行方法

cd /usr/local/ml　　#打开所在包的目录

java -classpath mymedialite.jar edu.hfut.recommender.main.RatingPrediction --training-file=ratings.txt --test-ratio=0.2 --data-dir=D:\data\movielens --recommender=UserItemBaseline --regU=20 --regI=11 --numIter=5　#运行

7.6 ItemKNNCosine

7.6.1 工具：MyMediaLite

7.6.2 语言：Java

7.6.3 参数

参数	默认值	含义	备注
--training-file	不能为空	训练数据	
--test-file	无	测试数据	
--test-ratio	无	测试数据为空时，训练数据分割比率	若 test-file 不为空，则 test-ratio 为空
--data-dir	无	文件根目录	
--recommender	ItemKNNCosine	评分预测方法	
k	20	预测所考虑的邻居数	

7.6.4 输入数据格式

第一列：userid，表示用户索引

第二列：itemid，表示产品索引

第三列：rating，表示用户对产品的评分

样例：sample-data-ratings.txt

7.6.5 运行方法

cd /usr/local/ml　　#打开所在包的目录

java -classpath mymedialite.jar edu.hfut.recommender.main.RatingPrediction --training-file=ratings.txt --test-ratio=0.2 --data-dir=D:\data\movielens --recommender=ItemKNNCosine –Knn=12　#运行

7.7 ItemKNNPearson

7.7.1 工具：MyMediaLite

7.7.2 语言：Java

7.7.3 参数

参数	默认值	含义	备注
--training-file	不能为空	训练数据	
--test-file	无	测试数据	

--test-ratio	无	测试数据为空时，训练数据分割比率	若 test-file 不为空，则 test-ratio 为空
--data-dir	无	文件根目录	
--recommender	ItemKNNPearson	评分预测方法	
k	20	预测所考虑的邻居数	

7.7.4 输入数据格式

第一列：userid，表示用户索引

第二列：itemid，表示产品索引

第三列：rating，表示用户对产品的评分

样例：sample-data-ratings.txt

7.7.5 运行方法

cd /usr/local/ml　　#打开所在包的目录

java -classpath mymedialite.jar edu.hfut.recommender.main.RatingPrediction --training-file=ratings.txt --test-ratio=0.2 --data-dir=D:\data\movielens --recommender= ItemKNNPearson –Knn=2　#运行

7.8 UserKNNCosine

7.8.1 工具：MyMediaLite

7.8.2 语言：Java

7.8.3 参数

参数	默认值	含义	备注
--training-file	不能为空	训练数据	
--test-file	无	测试数据	
--test-ratio	无	测试数据为空时，训练数据分割比率	若 test-file 不为空，则 test-ratio 为空
--data-dir	无	文件根目录	
--recommender	UserKNNCosine	评分预测方法	
k	20	预测所考虑的邻居数	

7.8.4 输入数据格式

第一列：userid，表示用户索引

第二列：itemid，表示产品索引

第三列：rating，表示用户对产品的评分

样例：sample-data-ratings.txt

7.8.5 运行方法

cd /usr/local/ml　　#打开所在包的目录

java -classpath mymedialite.jar edu.hfut.recommender.main.RatingPrediction --training-file=ratings.txt --test-ratio=0.2 --data-dir=D:\data\movielens --recommender=

UserKNNCosine –Knn=30　#运行

7.9 UserKNNPearson

7.9.1 工具：MyMediaLite

7.9.2 语言：Java

7.9.3 参数

参数	默认值	含义	备注
--training-file	不能为空	训练数据	
--test-file	无	测试数据	
--test-ratio	无	测试数据为空时，训练数据分割比率	若 test-file 不为空，则 test-ratio 为空
--data-dir	无	文件根目录	
--recommender	UserKNNPearson	评分预测方法	
k	20	预测所考虑的邻居数	

7.9.4 输入数据格式

第一列：userid，表示用户索引

第二列：itemid，表示产品索引

第三列：rating，表示用户对产品的评分

样例：sample-data-ratings.txt

7.9.5 运行方法

cd /usr/local/ml　　#打开所在包的目录

java -classpath mymedialite.jar edu.hfut.recommender.main.RatingPrediction --training-file=ratings.txt --test-ratio=0.2 --data-dir=D:\data\movielens --recommender=UserKNNPearson --Knn=20　#运行

7.10 BPNN

7.10.1 工具：MyMediaLite

7.10.2 语言：Java

7.10.3 参数

参数	默认值	含义	备注
--training-file	不能为空	训练数据	
--test-file	无	测试数据	
--test-ratio	无	测试数据为空时，训练数据分割比率	若 test-file 不为空，则 test-ratio 为空
--data-dir	无	文件根目录	
--recommender	BiasedMatrixFactorization	评分预测方法	

numFactors	10	潜在因素的数量	
biasReg	0.0001	偏差项的正则化常数	
regularization	0.015	正则化参数	
learnRate	0.01	学习速率	
numIter	10	训练集的迭代次数	

7.10.4 输入数据格式

第一列：userid，表示用户索引

第二列：itemid，表示产品索引

第三列：rating，表示用户对产品的评分

样例：sample-data-ratings.txt

7.10.5 运行方法

cd /usr/local/ml #打开所在包的目录

java -classpath mymedialite.jar edu.hfut.recommender.main.RatingPrediction --training-file=ratings.txt --test-ratio=0.2 --data-dir=D:\data\movielens --recommender= BiasedMatrixFactorization--numFactors=12 --biasReg=0.001 --regularization=0.02 --learnRate=0.02 --numIter=20 #运行

7.11 LogisticRegressionMatrixFactorization

7.11.1 工具：MyMediaLite

7.11.2 语言：Java

7.11.3 参数

参数	默认值	含义	备注
--training-file	不能为空	训练数据	
--test-file	无	测试数据	
--test-ratio	无	测试数据为空时，训练数据分割比率	若 test-file 不为空，则 test-ratio 为空
--data-dir	无	文件根目录	
--recommender	BiasedMatrixFactorization	评分预测方法	
numFactors	10	潜在因素的数量	
biasReg	0.0001	偏差项的正则化常数	
regU_MF	0.01	用户的正则化参数	
regI_MF	0.01	产品的正则化参数	
learnRate	0.01	学习速率	
numIter	10	训练集迭代次数	

7.11.4 输入数据格式

第一列：userid，表示用户索引

第二列：itemid，表示产品索引

第三列：rating，表示用户对产品的评分

样例：sample-data-ratings.txt

7.11.5 运行方法

cd /usr/local/ml　　#打开所在包的目录

java -classpath mymedialite.jar edu.hfut.recommender.main.RatingPrediction --training-file=ratings.txt --test-ratio=0.2 --data-dir=D:\data\movielens --recommender=LogisticRegressionMatrixFactorization --numFactors=12 --biasReg=0.001 --regU_MF=0.02 --regI_MF=0.02 --learnRate=0.02 --numIter=20　#运行

7.12 CoClustering

7.12.1 工具：MyMediaLite

7.12.2 语言：Java

7.12.3 参数

参数	默认值	含义	备注
--training-file	不能为空	训练数据	
--test-file	无	测试数据	
--test-ratio	无	测试数据为空时，训练数据分割比率	若 test-file 不为空，则 test-ratio 为空
--data-dir	无	文件根目录	
--recommender	CoClustering	评分预测方法	
numUserClusters	3	用户类簇数量	
numItemClusters	3	产品类簇数量	
numIter	10	迭代的最大次数	如果该算法收敛到一个稳定的解决方案，它会提前终止。

7.12.4 输入数据格式

第一列：userid，表示用户索引

第二列：itemid，表示产品索引

第三列：rating，表示用户对产品的评分

样例：sample-data-ratings.txt

7.12.5 运行方法

cd /usr/local/ml　　#打开所在包的目录

java -classpath mymedialite.jar edu.hfut.recommender.main.RatingPrediction --training-file=ratings.txt --test-ratio=0.2 --data-dir=D:\data\movielens --recommender=CoClustering --numUserClusters=4 --numItemClusters=5 --numIter=12　#运行

7.13 UserKNN

7.13.1 工具：MyMediaLite

7.13.2 语言：Java

7.13.3 参数

参数	默认值	含义	备注
--training-file	不能为空	训练数据	
--test-file	无	测试数据	
--test-ratio	无	测试数据为空时，训练数据分割比率	若 test-file 不为空，则 test-ratio 为空
--data-dir	无	文件根目录	
--recommender	UserKNN	项目推荐方法	
k	80	预测时考虑的邻居数	

7.13.4 输入数据格式

第一列：userid，表示用户索引

第二列：itemid，表示产品索引

样例：sample-data-train.txt

7.13.5 运行方法

cd /usr/local/ml　　#打开所在包的目录

java -classpath mymedialite.jar edu.hfut.recommender.main. ItemRecommendation --training-file=train.txt --test-file=test.txt --test-ratio=0.2 --data-dir=D:\data\lastfm --recommender=UserKNN --knn=10　#运行

7.14 ItemKNN

7.14.1 工具：MyMediaLite

7.14.2 语言：Java

7.14.3 参数

参数	默认值	含义	备注
--training-file	不能为空	训练数据	
--test-file	无	测试数据	
--test-ratio	无	测试数据为空时，训练数据分割比率	若 test-file 不为空，则 test-ratio 为空
--data-dir	无	文件根目录	
--recommender	UserKNN	项目推荐方法	
k	80	预测时考虑的邻居数	

7.14.4 输入数据格式

第一列：userid，表示用户索引

第二列：itemid，表示产品索引

样例：sample-data-train.txt

7.14.5 运行方法

cd /usr/local/ml　　#打开所在包的目录

java -classpath mymedialite.jar edu.hfut.recommender.main. ItemRecommendation --training-file=train.txt --test-file=test.txt --test-ratio=0.2 --data-dir=D:\data\lastfm

--recommender=ItemKNN --knn=10　#运行

7.15 BPRMF

7.15.1 工具：MyMediaLite

7.15.2 语言：Java

7.15.3 参数

参数	默认值	含义	备注
--training-file	不能为空	训练数据	
--test-file	无	测试数据	
--test-ratio	无	测试数据为空时，训练数据分割比率	若 test-file 不为空，则 test-ratio 为空
--data-dir	无	文件根目录	
--recommender	BPRMF	项目推荐方法	
numFactors	10	每个用户/产品潜在因子的数量	
biasReg	0.01	偏差项的正则化参数	
regU	0.0025	用户因子的正则化参数	
regI	0.0025	positive item factors 的正则化参数	
regJ	0.00025	negative item factors 的正则化参数	
numIter	30	训练数据的迭代次数	
learnRate	0.05	学习速率 α	

7.15.4 输入数据格式

第一列：userid，表示用户索引

第二列：itemid，表示产品索引

样例：sample-data-train.txt

7.15.5 运行方法

cd /usr/local/ml　　#打开所在包的目录

java -classpath mymedialite.jar edu.hfut.recommender.main. ItemRecommendation --training-file=train.txt --test-file=test.txt --test-ratio=0.2 --data-dir=D:\data\lastfm --recommender=BPRMF --numFactors=10 --biasReg=0.01 --regU=0.0025 --regI=0.0025 --regJ=0.00025 --numIter=30 --learnRate=0.05　#运行

7.16 WRMF

7.16.1 工具：MyMediaLite

7.16.2 语言：Java

7.16.3 参数

参数	默认值	含义	备注

--training-file	不能为空	训练数据	
--test-file	无	测试数据	
--test-ratio	无	测试数据为空时，训练数据分割比率	若 test-file 不为空，则 test-ratio 为空
--data-dir	无	文件根目录	
--recommender	WRMF	项目推荐方法	
numFactors	10	每个用户/产品潜在因子的数量	
regularization	0.015	正则化参数	
cPos	2.0	C 位置: positive observations 的权重.	The alpha value in Hu et al
numIter	30	训练集的迭代次数	

7.16.4 输入数据格式

第一列：userid，表示用户索引

第二列：itemid，表示产品索引

样例：sample-data-train.txt

7.16.5 运行方法

cd /usr/local/ml　　#打开所在包的目录

java -classpath mymedialite.jar edu.hfut.recommender.main. ItemRecommendation --training-file=train.txt --test-file=test.txt --test-ratio=0.2 --data-dir=D:\data\lastfm --recommender=WRMF --numFactors=10 --regularization=0.015 --cPos=2.0 --numIter=10 #运行

7.17 MostPopular

7.17.1 工具：MyMediaLite

7.17.2 语言：Java

7.17.3 参数

参数	默认值	含义	备注
--training-file	不能为空	训练数据	
--test-file	无	测试数据	
--test-ratio	无	测试数据为空时，训练数据分割比率	若 test-file 不为空，则 test-ratio 为空
--data-dir	无	文件根目录	
--recommender	MostPopular	项目推荐方法	

7.17.4 输入数据格式

第一列：userid，表示用户索引

第二列：itemid，表示产品索引

样例：sample-data-train.txt

7.17.5 运行方法

cd /usr/local/ml　　#打开所在包的目录

java -classpath mymedialite.jar edu.hfut.recommender.main.ItemRecommendation --training-file=train.txt --test-file=test.txt --test-ratio=0.2 --data-dir=D:\data\lastfm --recommender=MostPopular #运行

7.18 Random

7.18.1 工具：MyMediaLite

7.18.2 语言：Java

7.18.3 参数

参数	默认值	含义	备注
--training-file	不能为空	训练数据	
--test-file	无	测试数据	
--test-ratio	无	测试数据为空时，训练数据分割比率	若 test-file 不为空，则 test-ratio 为空
--data-dir	无	文件根目录	
--recommender	Random	项目推荐方法	

7.18.4 输入数据格式

第一列：userid，表示用户索引

第二列：itemid，表示产品索引

样例：sample-data-train.txt

7.18.5 运行方法

cd /usr/local/ml #打开所在包的目录

java -classpath mymedialite.jar edu.hfut.recommender.main.ItemRecommendation --training-file=train.txt --test-file=test.txt --test-ratio=0.2 --data-dir=D:\data\lastfm --recommender=Random #运行

八、异常检测（organized by LZ）

8.1 基于统计模型的离群点检测方法 StatisOutlierDetect

8.1.1 工具：

8.1.2 语言：Java

8.1.3 参数

参数	默认值	含义	备注
-t	3	检测阈值（标准差的倍数 $x_i - \bar{x} > 3\sigma$）	
-s	无	检测策略（左侧、右侧、双侧）	BOTH/LEFT/RIGHT

-in	/	输入文件路径	
-out	/	输出文件路径	

8.1.4 输入数据格式

一维数据（id+value）

1 34.2
2 30.5
3 23.5
4 26.7
5 35.6
6 2.3
7 23.9
8 30.1
.....

样例：Statsample.txt

8.1.5 运行方法

-t 3 -s BOTH -in /Statsample.txt -out /Statout.txt

8.2 基于 KNN 的离群点检测方法 KNNOutlierDetect

8.2.1 工具：

8.2.2 语言：Java

8.2.3 参数

参数	默认值	含义	备注
-t	无	距离阈值	
-p	无	分数阈值（单位%）	
-d	Euclidean	距离计算策略	Euclidean/Manhattan/Cos
-in	/	输入文件路径	
-out	/	输出文件路径	

8.2.4 输入数据格式

多维数据（id+value1+value2+value3.....）

1 2 5 3
2 5 3 6
3 5 5 6
4 5 5 1
5 6 3 2
6 22 5 2
7 2 3 6
8 0 0 2
.....

样例：KNNsample.txt

8.2.5 运行方法

-t 10 -p 5 -d Euclidean -in /KNNsample.txt -out /KNNout.txt

8.3 基于密度的局部离群点检测方法 LofOutlierDetect

8.3.1 工具：

8.3.2 语言：Java

8.3.3 参数

参数	默认值	含义	备注
-k	无	用户指定参数	用于确定对象密度的最小邻域
-t	无	局部离群因子阈值	簇中较中心的点值接近于1
-d	Euclidean	距离计算策略	Euclidean/Manhattan/Cos
-in	/	输入文件路径	
-out	/	输出文件路径	

8.3.4 输入数据格式

多维数据（id+value1+value2+value3……）

1 2 5 3
2 5 3 6
3 5 5 6
4 5 5 1
5 6 3 2
6 22 5 2
7 2 3 6
8 0 0 2
……

样例：Lofsample.txt

8.3.5 运行方法

-k 3 -t 4 -d Euclidean -in /Lofsample.txt -out /Lofout.txt

8.4 基于 k-means 的离群点检测方法 KMeansOutlierDetect

8.4.1 工具：

8.4.2 语言：Java

8.4.3 参数

参数	默认值	含义	备注
-k	无	聚簇值	
-t	无	阈值	对象与最近簇中心距离/最近簇间的平均距离
-d	Euclidean	距离计算策略	Euclidean/Manhattan/Cos
-iter	500	最大迭代次数	
-in	/	输入文件路径	
-out	/	输出文件路径	

8.4.4 输入数据格式

多维数据（id+value1+value2+value3.....）

1 2 5 3
2 5 3 6
3 5 5 6
4 5 5 1
5 6 3 2
6 22 5 2
7 2 3 6
8 0 0 2
.....

样例：KMeanssample.txt

8.4.5 运行方法

-k 3 -t 2 -d Euclidean -iter 500 -in /KMeanssample.txt -out /KMeansout.txt

九、文本挖掘（organized by QY）

十、中文分词（organized by QY）

十一、词干抽取（organized by QY）

任务	开源工具	语言	算法	算法名称	负责人员
聚类	Weka	Java	DBSCAN	Java	童伟
			FarestFirst		
			HierarchicalClusterer		
			K-Means		
			EM		
			C4.5(J48)		
			JRip		
分类	Weka	Java	LinearRegression	Java	童伟
			M5P		
			NaiveBayes		
			RandomTree		
			RandomForest		
关联规则	Weka	Java	Apriori		童伟
			PredictiveApriori	Java	
时序预测			ARIMA	Java	张雪
			ES		
			BPNN	R	
			SVR	R	
组合建模					高畅

集成学习	sklearn	Python	Bagging		Python	高畅
			Boosting			
个性化推荐	MyMediaLite	Java	Rating Prediction	SlopeOne	Java	王锦坤
				GlobalAverage		
				ItemAverage		
				UserAverage		
				UserItemBaseline		
				ItemKNNCosine		
				ItemKNNPearson		
				UserKNNCosine		
				UserKNNPearson		
				BiasedMatrixFactorization		
				CoClustering		
	MyMediaLite	Java	Item Recommendation	UserKNN		
				ItemKNN		
				BPRMF		
				WRMF		
				MostPopular		
				Random		
异常检测			KNN			李哲
			LOF			
			CBLOF			
			SVM			
文本挖掘	Mallet		LDA		Java	钱洋
中文分词	NLPIR		/		Java	钱洋
词干抽取	Stanford Core NLP		/		Java	钱洋