Liqi Zhu's Assignment 5

10.2

> Suppose that the data mining task is to cluster points (with (x,y) representing location)
> into three clusters, where the points are
> A1(2,10),A2(2,5),A3(8,4),B1(5,8),B2(7,5),B3(6,4),C1(1,2),C2(4,9).
> The distance function is Euclidean distance. Suppose initially we assign A1, B1, and C1
> as the center of each cluster, respectively. Use the k-means algorithm to show only
> (a) The three cluster centers after the first round of execution.
> (b) The final three clusters

(a)

First Round:

Center/Distance	A2	A3	B2	B3	C2
A1	5.00	8.49	7.07	7.21	2.24
B1	4.24	5.00	3.61	4.12	1.41
C1	3.16	7.94	6.71	5.39	7.62

Clusters: {A1},{B1,A3,B2,B3,C2},{C1,A2}.
Centers:(2,10),(6,6),(1.5,3.5)

(b)

FInal Clusters:{A1,B1,C2},{A3,B2,B3},{A2,C1}

10.4

> For the k-means algorithm, it is interesting to note that by choosing the initial cluster
> centers carefully, we may be able to not only speed up the algorithm’s convergence, but
> also guarantee the quality of the final clustering. The k-means++ algorithm is a variant of k-means, which chooses the initial centers as follows. First, it selects one center
> uniformly at random from the objects in the data set. Iteratively, for each object p other
> than the chosen center, it chooses an object as the new center. This object is chosen at
> random with probability proportional to dist(p)2, where dist(p) is the distance from p
> to the closest center that has already been chosen. The iteration continues until k centers
> are selected.
> Explain why this method will not only speed up the convergence of the k-means algorithm, but also guarantee the quality of the final clustering results.

Firstly, comapred to K-means algorithm's randomly picking, k-means++ algorithm picks its initial center with a more smart and even concept, so as to lower the SSEs and thus lower the K-means cost. From this perspective, less iteration are needed. Additionally, K-means++ can lead to a better result since random picked centroids that are not distributed over the data set can cause the result to get stuck in local optimal. K-means++ is obviously prior to K-means considering the distribution of the centers.

10.6

> Both k-means and k-medoids algorithms can perform effective clustering.
> (a) Illustrate the strength and weakness of k-means in comparison with k-medoids.
> (b) Illustrate the strength and weakness of these schemes in comparison with a hierarchical clustering scheme (e.g., AGNES)

(a)

Medoid is less sensitive by outliers or noise in the data set than a mean, the K-medoids algorithm shows more tolerance with more robustnes. While K-medoids algorithm costs more for processing.

(b)

Firstly, K-means and K-medoids are partitioning based methods, compared to AGNES, the first advantage of partitioning based method is that it's easy to undo a step by iterative relocation while it's impossible for AGNES to undo process. Hierarchical based clustering methods can not make any adjustments to the former split or merge, such weakness influences the quality of their result.
Secondly, partitioning based clustering shows better capability of finding spherical-shaped patterns for small to medium size databases. While compared to automatically determined number of clusters with AGNES or other hierarchical methods, partitioning based methods need the initialization of the number of clusters.
Hisrarchical methods have its difficulties for scaling as each decision may require examination and evaluation of large amount of data. Integration with all kinds of clustering approaches should be considered in order to improve the clustering process.

11.2

> AllElectronics carries 1000 products, P1, . . . , P1000. Consider customers Ada, Bob, and
> Cathy such that Ada and Bob purchase three products in common, P1,P2, and P3. For
> the other 997 products, Ada and Bob independently purchase seven of them randomly.
> Cathy purchases 10 products, randomly selected from the 1000 products. In Euclidean
> distance, what is the probability that dist(Ada,Bob) > dist(Ada,Cathy)? What if Jaccard
> similarity (Chapter 2) is used? What can you learn from this example?

Probability of Ada and Bob purchasing m same product:

$$P_{AB} = \frac{C_{990}^{10-m}}{C_{997}^{7}}$$

Probability of Ada and Cathy purchasing n same products:

$$P_{AC} = \frac{C_{990}^{10-n}}{C_{1000}^{10}}$$

With Euclidean distance,

$$P[dist(Ada, Bob) > dist(Ada, Cathy)] = P(m < n) = \sum_{m=3}^{10}(P_{AB} \sum_{n=m+1}^{10} P_{AC})$$

With Jaccard similarity

$$dist(Ada, Bob) = \frac{(10-m)+(10-m)}{m+(10-m)+(10-m)} = \frac{20-2m}{20-m}$$

,

$$dist(Ada, Cathy) = \frac{(10-n)+(10-n)}{n+(10-n)+(10-n)} = \frac{20-2n}{20-n}$$

.

We can find that from Jaccrd method we have the same result as $P[dist(Ada, Bob) > dist(Ada, Cathy)] = P(j < k)$, leads to the same probaility using Eculidean distance. As a conclusion, no matter what measurement method is applied to two binary attribute sets, the result would not be affected.

9.1

> Using Weca , solve 9.1 with MLNN, SVM, and another classifier of your choice

Following results are generated with the settings of 10-fold cross-validation with all available evalution metrics.

MLNN

=== Run information ===

Scheme: weka.classifiers.functions.MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

Relation: 9.1

Instances: 165

Attributes: 4

department

status

age

salary

Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

Sigmoid Node 0

Inputs Weights

Threshold 5.077849919304931

Node 2 -0.798870007989803

Node 3 -3.2553307300492724

Node 4 -1.6849103640762992

Node 5 -1.9722357346589903

Node 6 -0.6077028566801436

Node 7 0.6877457905152902

Node 8 -0.49825100408460005

Node 9 -2.614381465228299

Node 10 -2.1185097501440233

Sigmoid Node 1

Inputs Weights

Threshold -5.077776341999067

Node 2 0.8299328096639218

Node 3 3.2668069308453087

Node 4 1.6310381053303158

Node 5 1.9751693228544362
Node 6 0.5808522121372128
Node 7 -0.6837530207251162
Node 8 0.4846290073259015
Node 9 2.6150624262231497
Node 10 2.1484680707196433
Sigmoid Node 2
Inputs Weights
Threshold -0.08620349986676684
Attrib department=sales 0.012600993238492612
Attrib department=systems 0.3191193091224657
Attrib department=marketing 0.01847411578253666
Attrib department=secretary -0.19171693697429368
Attrib age=31¡35 -0.3122482391062593
Attrib age=26¡30 0.5835492730772195
Attrib age=21¡25 0.7542810828120736
Attrib age=41¡45 -0.25163364381064085
Attrib age=36¡40 -0.38342085512708035
Attrib age=46¡50 -0.20928257942558
Attrib salary=46K¡50K -0.571945387696409
Attrib salary=26K¡30K 0.3499069945630241
Attrib salary=31K¡35K 0.8599736495383075
Attrib salary=66K¡70K -0.6757140926272822
Attrib salary=41K¡45K 0.47382264656480233
Attrib salary=36K¡40K -0.23159494351598084
Sigmoid Node 3
Inputs Weights
Threshold 0.025346272016822595
Attrib department=sales -0.32864194581758754
Attrib department=systems 0.5920642195263259
Attrib department=marketing 0.20156508031761183
Attrib department=secretary -0.6007585551542152
Attrib age=31¡35 -0.6062600733319657

Attrib age=26¡30 1.2852738771258667
Attrib age=21¡25 1.5699313688961294
Attrib age=41¡45 -0.7509412765773987
Attrib age=36¡40 -1.0309086447642049
Attrib age=46¡50 -0.7302655318150391
Attrib salary=46K¡50K -1.2139777194307757
Attrib salary=26K¡30K 0.48350101620000774
Attrib salary=31K¡35K 1.8622117941871033
Attrib salary=66K¡70K -1.8354040575268293
Attrib salary=41K¡45K 1.170525103478124
Attrib salary=36K¡40K -0.7178887222234804
Sigmoid Node 4
Inputs Weights
Threshold 0.0465277861983857
Attrib department=sales -0.18370043339263006
Attrib department=systems 0.38317269323577663
Attrib department=marketing 0.06872941128984131
Attrib department=secretary -0.3696472894376704
Attrib age=31¡35 -0.4057412390102155
Attrib age=26¡30 0.9276241611072675
Attrib age=21¡25 1.0970233668154243
Attrib age=41¡45 -0.44715060590159267
Attrib age=36¡40 -0.6263550280798215
Attrib age=46¡50 -0.42646235986262754
Attrib salary=46K¡50K -0.8282809963284237
Attrib salary=26K¡30K 0.3958174335925043
Attrib salary=31K¡35K 1.332524172687338
Attrib salary=66K¡70K -1.1400832034287853
Attrib salary=41K¡45K 0.6956100621626622
Attrib salary=36K¡40K -0.511068938201242
Sigmoid Node 5
Inputs Weights
Threshold -0.022426765433450343

Attrib department=sales -0.19306856622245014
Attrib department=systems 0.4779361865593974
Attrib department=marketing 0.16898916784784349
Attrib department=secretary -0.4262141188821358
Attrib age=31¡35 -0.46731068585513974
Attrib age=26¡30 0.9470242446874014
Attrib age=21¡25 1.1610232481211777
Attrib age=41¡45 -0.56156635519177
Attrib age=36¡40 -0.7644067381327756
Attrib age=46¡50 -0.5476332961238081
Attrib salary=46K¡50K -0.9307277394440308
Attrib salary=26K¡30K 0.4105143519212113
Attrib salary=31K¡35K 1.4298906672631688
Attrib salary=66K¡70K -1.3146083724285471
Attrib salary=41K¡45K 0.870902790922854
Attrib salary=36K¡40K -0.5146412094931446
Sigmoid Node 6
Inputs Weights
Threshold -0.11246408558877943
Attrib department=sales 0.030943733941151127
Attrib department=systems 0.2707166034357742
Attrib department=marketing 0.003653113313281002
Attrib department=secretary -0.11375860909842499
Attrib age=31¡35 -0.22835596759234983
Attrib age=26¡30 0.5702668345109629
Attrib age=21¡25 0.6152697926035567
Attrib age=41¡45 -0.18140268271770174
Attrib age=36¡40 -0.22993512543217498
Attrib age=46¡50 -0.17777278513441389
Attrib salary=46K¡50K -0.5029538515802372
Attrib salary=26K¡30K 0.3585084705191638
Attrib salary=31K¡35K 0.7343350310882023
Attrib salary=66K¡70K -0.5033851892889031

Attrib salary=41K¡45K 0.41705013418198206
Attrib salary=36K¡40K -0.15310408987359056
Sigmoid Node 7
Inputs Weights
Threshold 0.14359968748690133
Attrib department=sales 0.0308844930321132
Attrib department=systems -0.038245001369986015
Attrib department=marketing -0.1722482354589961
Attrib department=secretary -0.1157825138492811
Attrib age=31¡35 -0.0457880723340485
Attrib age=26¡30 -0.0939867531181069
Attrib age=21¡25 -0.10098280549550315
Attrib age=41¡45 -0.07971900177279614
Attrib age=36¡40 -0.1273617352323285
Attrib age=46¡50 -0.09413156744851643
Attrib salary=46K¡50K -0.16651353883117026
Attrib salary=26K¡30K -0.035995972716067914
Attrib salary=31K¡35K -0.0981135045576773
Attrib salary=66K¡70K -0.07394911962561224
Attrib salary=41K¡45K -0.21370187168690238
Attrib salary=36K¡40K -0.06538005656135208
Sigmoid Node 8
Inputs Weights
Threshold -0.07037686547883158
Attrib department=sales 0.061932218451711145
Attrib department=systems 0.18996592605127247
Attrib department=marketing 0.04851228462146839
Attrib department=secretary -0.07753869898783228
Attrib age=31¡35 -0.22139508044791156
Attrib age=26¡30 0.4862724224154825
Attrib age=21¡25 0.6094957485005332
Attrib age=41¡45 -0.10289434094536438
Attrib age=36¡40 -0.22298678451458484

Attrib age=46¡50 -0.14035104663170997
Attrib salary=46K¡50K -0.4836930970715099
Attrib salary=26K¡30K 0.34113831771448133
Attrib salary=31K¡35K 0.7354872646431403
Attrib salary=66K¡70K -0.3849517106785908
Attrib salary=41K¡45K 0.37576312175178983
Attrib salary=36K¡40K -0.09983684200290624
Sigmoid Node 9
Inputs Weights
Threshold 0.016058930899995354
Attrib department=sales -0.28192817133154097
Attrib department=systems 0.5355771588258466
Attrib department=marketing 0.1766652263286512
Attrib department=secretary -0.493289269014781
Attrib age=31¡35 -0.5385435118436851
Attrib age=26¡30 1.1396325250733903
Attrib age=21¡25 1.3671122817007393
Attrib age=41¡45 -0.6635552565499636
Attrib age=36¡40 -0.9116240401660546
Attrib age=46¡50 -0.65155542422012
Attrib salary=46K¡50K -1.0697808162408653
Attrib salary=26K¡30K 0.4478045557254808
Attrib salary=31K¡35K 1.6680581330567885
Attrib salary=66K¡70K -1.5936574851889078
Attrib salary=41K¡45K 1.0555492088761849
Attrib salary=36K¡40K -0.6457927285160276
Sigmoid Node 10
Inputs Weights
Threshold 0.055144735855013755
Attrib department=sales -0.1922017825107403
Attrib department=systems 0.507332887135344
Attrib department=marketing 0.13934348432502755
Attrib department=secretary -0.45433208104598666

Attrib age=31¡35 -0.49774153597825016
Attrib age=26¡30 1.0117806818794415
Attrib age=21¡25 1.2301685688075434
Attrib age=41¡45 -0.5648255767286017
Attrib age=36¡40 -0.7788934770237211
Attrib age=46¡50 -0.5608808550266506
Attrib salary=46K¡50K -0.9605860742985314
Attrib salary=26K¡30K 0.39450429042784996
Attrib salary=31K¡35K 1.4922512282746332
Attrib salary=66K¡70K -1.3967736509967557
Attrib salary=41K¡45K 0.9466130237580793
Attrib salary=36K¡40K -0.5329714314981593
Class senior
Input
Node 0
Class junior
Input
Node 1

Time taken to build model: 0.29 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances 165 100 %
Incorrectly Classified Instances 0 0 %
Kappa statistic 1
Mean absolute error 0.0044
Root mean squared error 0.0055
Relative absolute error 1.0198 %
Root relative squared error 1.1821 %
Total Number of Instances 165

=== Detailed Accuracy By Class ===

```
              TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC A
rea  PRC Area  Class
              1.000    0.000    1.000      1.000    1.000      1.000    1.000
1.000     senior
              1.000    0.000    1.000      1.000    1.000      1.000    1.000
1.000     junior
```

Weighted Avg. 1.000 0.000 1.000 1.000 1.000 1.000 1.000 1.000

=== Confusion Matrix ===

a b <-- classified as
52 0 | a = senior
0 113 | b = junior

SVM
=== Run information ===

Scheme: weka.classifiers.functions.VotedPerceptron -I 1 -E 1.0 -S 1 -M 10000
Relation: 9.1
Instances: 165
Attributes: 4
department
status
age
salary
Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

VotedPerceptron: Number of perceptrons=16

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances 163 98.7879 %
Incorrectly Classified Instances 2 1.2121 %
Kappa statistic 0.9716
Mean absolute error 0.0127
Root mean squared error 0.1095
Relative absolute error 2.9433 %
Root relative squared error 23.5649 %
Coverage of cases (0.95 level) 98.7879 %
Mean rel. region size (0.95 level) 50.303 %
Total Number of Instances 165

=== Detailed Accuracy By Class ===

```
                TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC A
rea  PRC Area  Class
                0.962    0.000    1.000      0.962    0.980      0.972    1.000
1.000     senior
                1.000    0.038    0.983      1.000    0.991      0.972    1.000
1.000     junior
```

Weighted Avg. 0.988 0.026 0.988 0.988 0.988 0.972 1.000 1.000

=== Confusion Matrix ===

a b <-- classified as
50 2 | a = senior
0 113 | b = junior

ClassificationViaRegression

=== Run information ===

Scheme: weka.classifiers.meta.ClassificationViaRegression -W
weka.classifiers.trees.M5P -- -M 4.0
Relation: 9.1
Instances: 165

Attributes: 4
department
status
age
salary
Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

Classification via Regression

Classifier for class with index 0:

M5 pruned model tree:
(using smoothed linear models)

salary=46K¡50K,66K¡70K,36K¡40K <= 0.5 : LM1 (90/0%)
salary=46K¡50K,66K¡70K,36K¡40K > 0.5 : LM2 (75/0%)

LM num: 1
status =
0.0593 * age=31¡35,41¡45,36¡40,46¡50
+ 0.0869 * salary=46K¡50K,66K¡70K,36K¡40K
- 0.029

LM num: 2
status =
0.9025 * age=31¡35,41¡45,36¡40,46¡50
+ 0.1014 * salary=46K¡50K,66K¡70K,36K¡40K
- 0.0338

Number of Rules : 2

Classifier for class with index 1:

M5 pruned model tree:

(using smoothed linear models)

salary=26K¡30K,41K¡45K,31K¡35K <= 0.5 : LM1 (75/0%)
salary=26K¡30K,41K¡45K,31K¡35K > 0.5 : LM2 (90/0%)

LM num: 1
status =
0.9025 * age=26¡30,21¡25
+ 0.1014 * salary=26K¡30K,41K¡45K,31K¡35K
+ 0.0299

LM num: 2
status =
0.0593 * age=26¡30,21¡25
+ 0.0869 * salary=26K¡30K,41K¡45K,31K¡35K
+ 0.8828

Number of Rules : 2

Time taken to build model: 0.01 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances 165 100 %
Incorrectly Classified Instances 0 0 %
Kappa statistic 1
Mean absolute error 0.0301
Root mean squared error 0.0386
Relative absolute error 6.9589 %
Root relative squared error 8.3026 %
Coverage of cases (0.95 level) 100 %
Mean rel. region size (0.95 level) 56.9697 %
Total Number of Instances 165

=== Detailed Accuracy By Class ===

```
               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC A
rea  PRC Area  Class
               1.000    0.000    1.000      1.000    1.000      1.000    1.000
1.000     senior
               1.000    0.000    1.000      1.000    1.000      1.000    1.000
1.000     junior
```

Weighted Avg. 1.000 0.000 1.000 1.000 1.000 1.000 1.000 1.000

=== Confusion Matrix ===

a b <-- classified as
52 0 | a = senior
0 113 | b = junior