Liqi Zhu Assignment 4

7.5

> Section 7.2.4 presented various ways of defining negatively correlated patterns. Consider Definition 7.3: "Suppose that itemsets X and Y are both frequent, that is, sup(X)>=min sup and sup(Y)>= min sup, where min sup is the minimum support threshold. If (P(X|Y)+P(Y|X))/2 <ϵ, where ϵ is a negative pattern threshold, then pattern X$\cup$Y is a negatively correlated pattern." Design an efficient pattern growth algorithm for mining the set of negatively correlated patterns.

Algorithm: Mining the set of negatively correlated patterns
Input: frequent itemsets X and Y and threshold ϵ
Ouput: the relation of paterns.

```
D: Dataset ;
X,Y: Frequent itemsets;
f: Threshold of negastive corrlation;
CX=0;
CY=0;
CXY=0;
for each trasaction T in dataset
   if X in T: CX++
   if Y in T: CY++
   if (X|Y) in T: CXY++
   if  (CXY/CX+CXY/CY) <2f:
   return 'X,Y is a set of negatively correlated paterns'
else
   return 'X,Y is not a set of negatively correlated paterns'
```

7.9

> Section 7.5.1 defined a pattern distance measure between closed patterns P1 and P2 as
> $$Pat.Dist(P1,P2) = 1 - \frac{|T(P1) \cap T(P2)|}{|T(P1) \cup T(P2)|}$$
> where T(P1) and T(P2) are the supporting transaction sets of P1 and P2, respectively. Is this a valid distance metric? Show the derivation to support your answer.

(1)
$D(P_1, P_2) > 0, \forall P_1 \neq P_2$
$T(P_1) \cap T(P_2) \subset T(P_1) \cup T(P_2)$ for any $P_1 \neq P_2$,
$\frac{|T(P_1) \cap T(P_2)|}{|T(P_1) \cup T(P_2)|} < 1 \Rightarrow D(P_1, P_2) > 0$.

(2)
$D(P_1, P_2) = 0, \forall P_1 = P_2$
For each $P_1 = P_2$, $P_1 \cup P_2 = P_1 \cap P_2 = P_1 = P_2$,
so $\frac{|T(P_1) \cap T(P_2)|}{|T(P_1) \cup T(P_2)|} = 1 \Rightarrow D(P_1, P_2) = 0$.
(3)
$P_1 \cup P_2 = P_2 \cup P_1, P_1 \cap P_2 = P_2 \cap P_1$,
$D(P_1, P_2) = D(P_2, P_1)$.
(4)
$D(P_1, P_2) + D(P_2, P_3) \geq D(P_1, P_3), \forall P_1, P_2, P_3$
$|T(P_1)| = a, |T(P_2)| = b, |T(P_3)| = c$
$|T(P_1) \cap T(P_2)| = b_1, |T(P_2) - T(P_1) \cap T(P_2)| = b_2$
$|T(P_1) \cap T(P_3)| = c_1, |T(P_3) - T(P_1) \cap T(P_3)| = c_2$
$|T(P_1) \cap T(P_2) \cap T(P_3)| = d_1, |T(P_2) \cap T(P_3) - T(P_1) \cap T(P_2) \cap T(P_3)| = d_2$
Since $(T(P1) \cap T(P2)) \cup (T(P1) \cap T(P3)) \subseteq T(P1)$, we have
$|T(P1) \cap T(P2)| + |T(P1) \cap T(P3)| - |T(P1) \cap T(P2) \cap T(P3)| \leq |T(P1)| \Rightarrow b_1 + c_1 - d_1 \leq a$
To prove:$D(P1, P2) + D(P2, P3) \geq D(P1, P3) \Rightarrow \frac{b_1}{a+b_2} + \frac{c_1}{a+c_2} \leq 1 + \frac{d_1+d_2}{b+c-d_1-d_2}$
$1 + \frac{d_1+d_2}{b_1+b_2+c_1+c_2-d_1-d_2} \geq 1 + \frac{d_1}{b_1+b_2+c_1+c_2-d_1} \, (d_2 \geq 0)$
$\geq 1 + \frac{d_1}{a+b_2+c_2} = \frac{a+b_2+c_2+d_1}{a+b_2+c_2}$
$\geq \frac{b_1+c_1+b_2+c_2}{a+b_2+c_2} = \frac{b_1+c_2}{a+b_2+c_2} + \frac{b_2+c_1}{a+b_2+c_2}$
$\geq \frac{b_1}{a+b_2} + \frac{c_1}{a+c_2} \, (a + b_2 \geq b_1, c_2 \geq 0, a + c_2 \geq c_1, b_2 \geq 0)$
Thus, $D(P_1, P_2) + D(P_2, P_3) \geq D(P_1, P_3), \forall P_1, P_2, P_3$ is true.

7.10

> Association rule mining often generates a large number of rules, many of which may be similar, thus not containing much novel information. Design an efficient algorithm that compresses a large set of patterns into a small compact set. Discuss whether your mining method is robust under different pattern similarity definitions.

For two frequent patterns fp_i, fp_j with support of s_i, s_j , if $s_i = s_j$ and $fp_i \subset fp_j$, we could remove the pattern fp_i.

Algorithm: Compress frequent patterns

```
FP: A set of frequent paterns and supports;
for i in range(length(FP)):
    for j in range(i+1,length(FP):
        if s_i=s_j and fp_i in fp_j:
            remove fp_i from FP
return FP
```

8.3

> Given a decision tree, you have the option of (a) converting the decision tree to rules and then pruning the resulting rules, or (b) pruning the decision tree and then converting the pruned tree to rules.What advantage does (a) have over (b)?

Compare to method (a), we could clear the subtree completely with method (b) while can't remove any precondition of it, which can be accomplished with method (a). Method (a) is less restrictive.

8.5

> Given a 5-GB data set with 50 attributes (each containing 100 distinct values) and 512 MB of main memory in your laptop, outline an efficient method that constructs decision trees in such large data sets. Justify your answer by rough calculation of your main memory usage.

Apply concept of BOAT: Seperate the data into 10 subsets, each subset will construct a tree and these trees will produce a new tree. Then the memory cost each time cost can be low.
100x50x(the number of class labels), assume 10kb each lable and the cost each time is about 50mb.

8.7

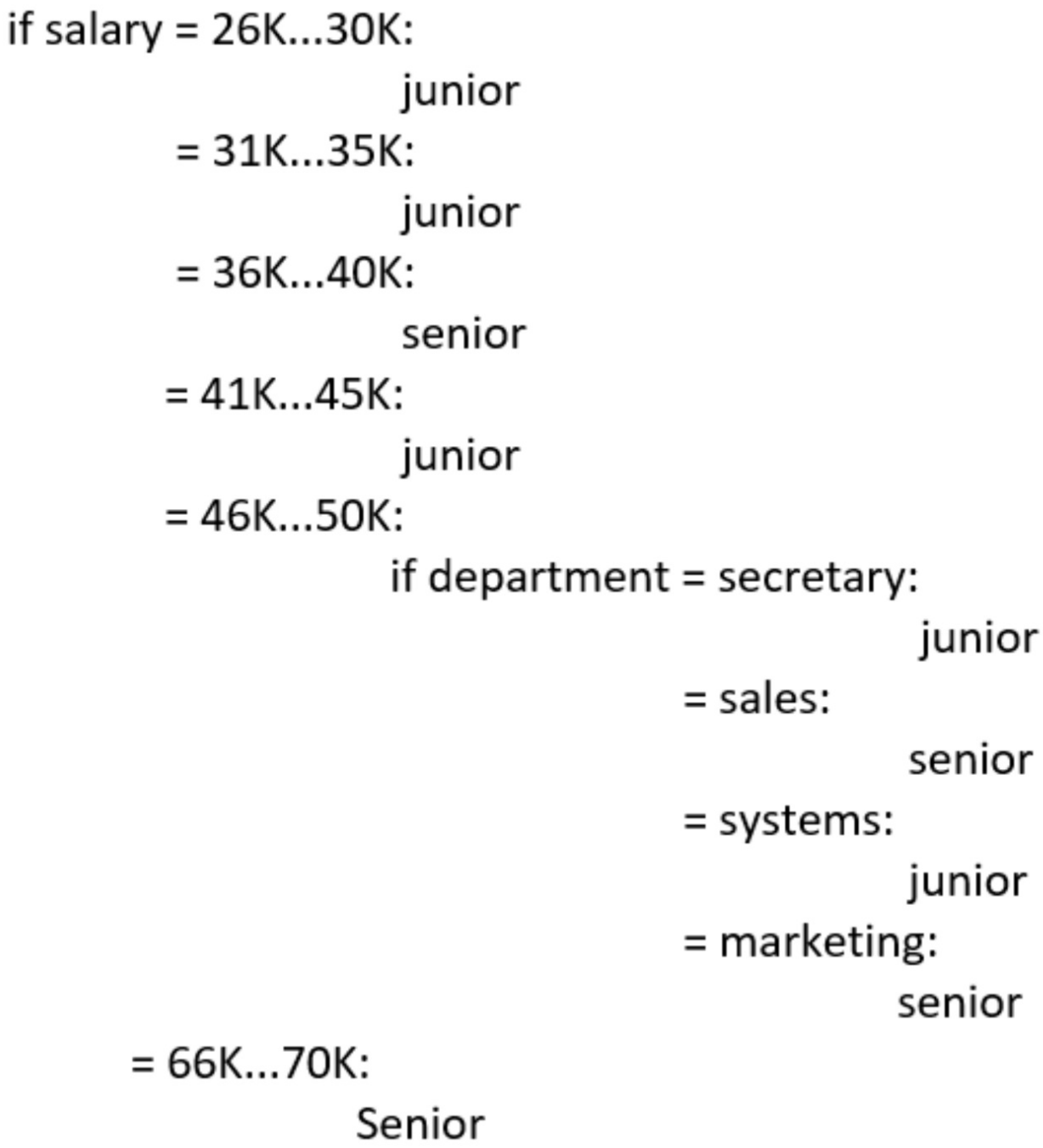
> The following table consists of training data from an employee database. The data have been generalized. For example, “31 : : : 35” for age represents the age range of 31 to 35. For a given row entry, count represents the number of data tuples having the values for department, status, age, and salary given in that row.

department	*status*	*age*	*salary*	*count*
sales	senior	31...35	46K...50K	30
sales	junior	26...30	26K...30K	40
sales	junior	31...35	31K...35K	40
systems	junior	21...25	46K...50K	20
systems	senior	31...35	66K...70K	5
systems	junior	26...30	46K...50K	3
systems	senior	41...45	66K...70K	3
marketing	senior	36...40	46K...50K	10
marketing	junior	31...35	41K...45K	4
secretary	senior	46...50	36K...40K	4
secretary	junior	26...30	26K...30K	6

Let status be the class label attribute.
(a) How would you modify the basic decision tree algorithm to take into consideration the count of each generalized data tuple (i.e., of each row entry)?
(b) Use your algorithm to construct a decision tree from the given data.
(c) Given a data tuple having the values "systems," "26 . . . 30," and "46–50K" for the attributes department, age, and salary, respectively, what would a na¨ıve Bayesian classification of the status for the tuple be?

(a)
Discrete valued attributes may be encoded such that there is one input unit per domain value. We have 4 inputs for "Department", 6 inputs for "Age", and 6 inputs for "Salary". There will be 16 input units in total.
In order to simplify the decision tree we can choose the salary for the first layer and then the subtree can be build with the input of "Department". "Age" is removed because there shows multible response of same age.

(b)



(c)

$$P(X|senior) = 0; P(X|junior) = 0.018$$

Naive Bayesian classification predicts "junior"

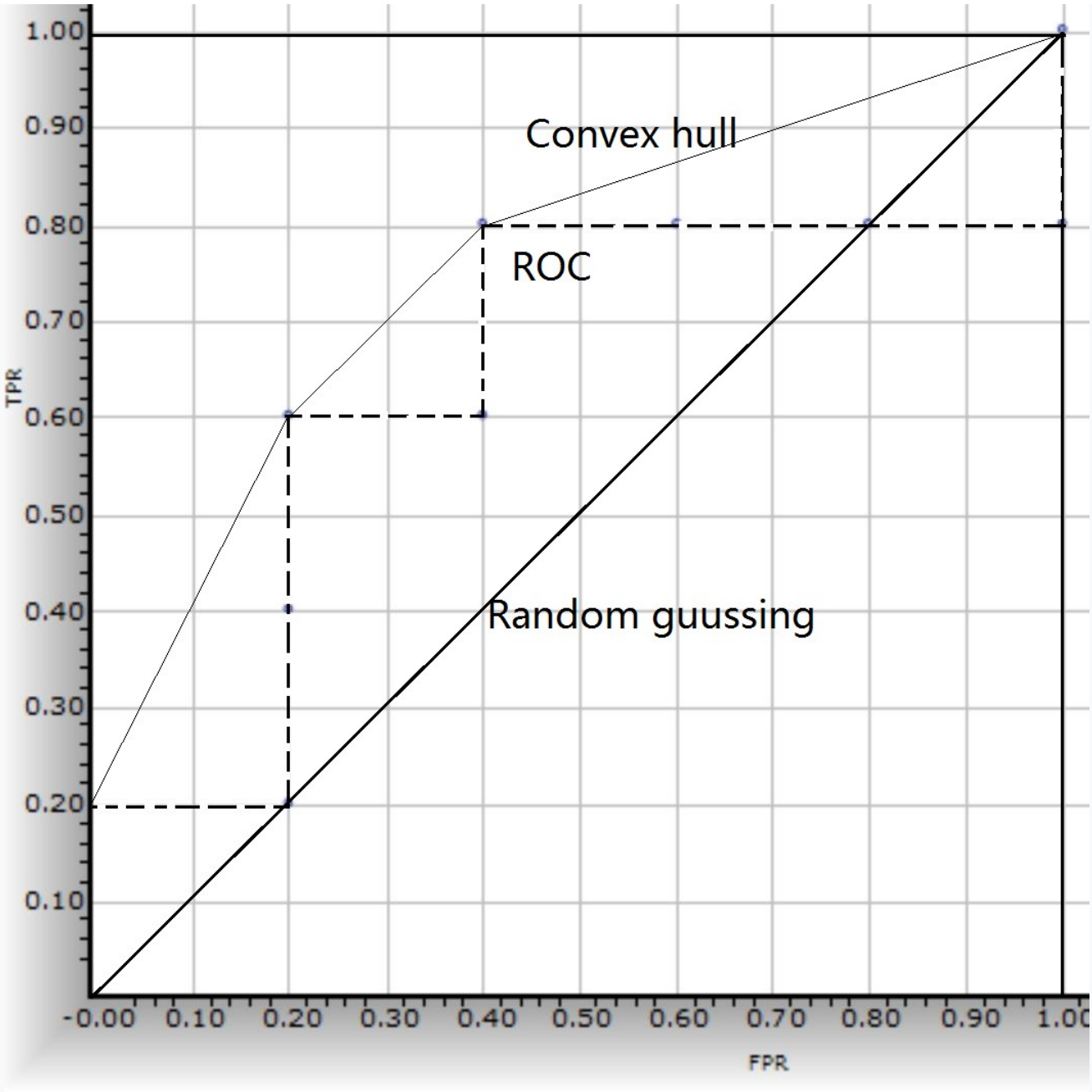
8.12

> The data tuples of Figure 8.25 are sorted by decreasing probability value, as returned by

a classifier. For each tuple, compute the values for the number of true positives .TP/, false positives .FP/, true negatives .TN/, and false negatives .FN/. Compute the true positive rate .TPR/ and false positive rate .FPR/. Plot the ROC curve for the data.

Tuple #	*Class*	*Probability*
1	*P*	0.95
2	*N*	0.85
3	*P*	0.78
4	*P*	0.66
5	*N*	0.60
6	*P*	0.55
7	*N*	0.53
8	*N*	0.52
9	*N*	0.51
10	*P*	0.40

Class	Prob.	TP	FP	TN	FN	TPR	FPR
P	0.95	1	0	5	4	0.2	0
N	0.85	1	1	4	4	0.2	0.2
P	0.78	2	1	4	3	0.4	0.2
P	0.66	3	1	4	2	0.6	0.2
N	0.60	3	2	3	2	0.6	0.4
P	0.55	4	2	3	1	0.8	0.4
N	0.53	4	3	2	1	0.8	0.6
N	0.52	4	4	1	1	0.8	0.8
N	0.51	4	5	0	1	0.8	1
P	0.40	5	5	0	0	1	1



8.14

Suppose that we want to select between two prediction models, M1 and M2. We have performed 10 rounds of 10-fold cross-validation on each model, where the same data partitioning in round i is used for both M1 and M2. The error rates obtained forM1 are 30.5, 32.2, 20.7, 20.6, 31.0, 41.0, 27.7, 26.0, 21.5, 26.0. The error rates for M2 are 22.4, 14.5, 22.4, 19.6, 20.7, 20.4, 22.1, 19.4, 16.2, 35.0. Comment on whether one model is significantly better than the other considering a significance level of 1%.

$$t = \frac{\overline{err}(M1) - \overline{err}(M2)}{\sqrt{var(M1 - M2)/k}}$$

$$var(M1 - M2) = \frac{1}{k}\sum_{i=0}^{k}[err(M1)_i - err(M2)_i - (\overline{err}(M1) - \overline{err}(M2))]$$

$\overline{err}(M1) = 27.72$
$\overline{err}(M2) = 21.27$

$$var(M1 - M2) = 68.1225$$

$$t = 2.4712$$

According to T-distribution table, when $k = 9, sig = 0.01, t = 3.250$

$$2.4712 < 3.250$$

,
So there is not a significantly better one.