# Constraint Pattern Mining (13 points)

## Purpose

* + Understand the basic concepts of constraint pattern mining.

## Requirements

a.(3, L1) For a set of values *S* and value *v*, Characterize the constraint *min*(*s*) *≥ v* (label if it satisfies anti-monotonic, monotonic, or succinct constraint category). Provide a simple example to explain your answer.

### 1.a

Min(S)>= v function is an **anti-monotonicity** and **succinct** constraint.

Set:

S :

|  |  |  |
| --- | --- | --- |
| Item | Profit | Price |
| A | -30 | 30 |
| B | 0 | 5 |
| C | 50 | 25 |
| D | 20 | 20 |
| E | -10 | 50 |
|  |  |  |

v = 25

For min(price)>=25, B and D fail, we can exclude all subsets of B or D.

A, C, E meet the requirement, all sets in S meet the requirement must be a subset belong to them.

b.(3, L1) For a set of values *S* and value *v*, Characterize the constraint *max*(*s*) *≤ v* (label if it satisfies anti-monotonic, monotonic, or succinct constraint category). Provide a simple example to explain your answer.

### 1.b

Max(S)<= v function is an **anti-monotonicity** and **succinct** constraint.

Set:

S =

|  |  |  |
| --- | --- | --- |
| Item | Profit | Price |
| A | -30 | 30 |
| B | 0 | 5 |
| C | 50 | 25 |
| D | 20 | 20 |
| E | -10 | 50 |
|  |  |  |

v = 25

For max(price)<=25, A and E fail, we can exclude all subsets of A or E.

B, C, D meet the requirement, all sets in S meet the requirement must be a subset belong to them.

c.(3, L1) [T, F] Convertible constraints can potentially have the same pruning power as anti-montonic and monotonic constraints.

### 1.c

T

d.(4, L3) Between anti-monotonic or monotonic constraints, which constraint can prune more effectively? Explain why.

### 1.d

Anti-monotonic will be more effective if few sets cannot meet the requirement (fail). Since anti-monotonic is fail once fail all, when there are few sets to be pruned, it would be a better solution.

Monotonic will be more effective if few sets can meet the requirement (succeed). Since monotonic is succeed once succeed forever, we can easily find the sets contain element meet requirement and get the right answer.

# Sequential Pattern Mining (9 points)

Suppose a sequence database D contains three sequences as follows. Note (*bc*) means that items b and c are purchased at the same time (i.e., in the same transaction). Let the minimum support be 3. You are going to use PrefixSpan to mine the frequent sequential patterns.

|  |  |
| --- | --- |
| **Customer ID** | **Shopping sequence** |
| 1 | *a*(*bc*)(*ade*)*f* |
| 2 | (*bd*)*c*(*fad*) |
| 3 | *a*(*bc*)*f* (*bc*)(*ef* ) |

Table 1: Transaction database to mine sequential patterns

## Purpose

a.(3, L1)Use Prefix Span. Show b’s projected database.

Since the minsup for d and e <3, the b’s projected database should be:

(\_c)(\_)f

(\_)c(f\_)

(\_c)f (bc)(f )

b.(3, L1) Use Prefix Span. What frequent patterns will you get from b’s projected database? (min support is 3)

bb 1

bc 3

(bc) 2

bf 3

The frequent pattern should be <bc> and <bf>.

c.(3, L3) What is the major benefit of Prefix Span over GSP?

No candidate sequence needs to be generated

Projected (parititioned) databases keep shrinking

# Decision Trees (18 points)

ID3 is a simple algorithm for decision tree construction using information gain. The steps of the ID3 algorithm are similar to those introduced in the lecture. In particular, ID3 uses information gain to select decision attributes, and each attribute is used at most once in any root-to-leaf path in the decision tree. You will use ID3 to build a decision tree that predicts whether it is possible to play a game of tennis given a set of weather conditions.

## Purpose

* + Understand and practice basic decision tree construction, calculation of information gain measures, and classifier evaluation.

## Requirements

* + Show the calculations for selecting the decision tree attributes and the labels for each leaf.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **id** | **Outlook** | **Temp** | **Humidity** | **Wind** | **Play Tennis** |
| 1 | Sunny | Hot | High | Weak | No |
| 2 | Sunny | Hot | High | Strong | No |
| 3 | Overcast | Hot | High | Weak | Yes |
| 4 | Rain | Mild | High | Weak | Yes |
| 5 | Rain | Cool | Normal | Weak | Yes |
| 6 | Rain | Cool | Normal | Strong | No |
| 7 | Overcast | Cool | Normal | Strong | Yes |
| 8 | Sunny | Mild | High | Weak | No |
| 9 | Sunny | Cool | Normal | Weak | Yes |
| 10 | rain | Mild | Normal | Weak | Yes |
| 11 | Sunny | Mild | Normal | Strong | Yes |
| 12 | Overcast | Mild | High | Strong | Yes |
| 13 | Overcast | Hot | Normal | Weak | Yes |
| 14 | Rain | Mild | High | Strong | No |

Table 2: Training data for the decision tree problem

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **id** | **Outlook** | **Temp** | **Humidity** | **Wind** | **PlayTennis** |
| 1 | Overcast  Sunny  Rain  Overcast | Hot  Hot  Mild  Cool | High  Normal  Normal  High | Strong  Weak  Strong  Strong | No |
| 2 | Yes |
| 3 | No |
| 4 | Yes |

Table 3: Test data for the decision tree problem

a.(6, L2) Use the ID3 algorithm to construct a decision tree using the training data in Table2[.](#_bookmark0) When multiple attributes have same information gain, select the one whose name appears earliest in the alphabetical order. Show the final decision tree and the calculations to derive the tree.

b.(4, L2) Evaluate your constructed decision tree using the test data in Table3[in](#_bookmark1) terms

of precision and recall for the classifier. Show your calculations with the help of a confusion matrix.

c.(4, L2) ID3 is often biased towards multivalued attributes. Would your decision about the attribute selected for the very first split in part (a) change if you used Gain Ratio? Show the calculations to support your answer.

d.(4, L3) Each root-to-leaf path in any decision tree can be converted into a rule, such

=*T rue* =*False*

as a path *A*1 *− −→ A*2 *− − → class* = +1 can be converted to the rule “If attribute

*Ai* is true and attribute *A*2 is false, then the instance has class +1”. 1.Generate the rules for each leaf of your constructed decision tree.

2.Is it possible to construct a decision tree from a set of rules? Explain your answer.

# Bayes Classifier (18 points)

Using the same training data as in Table2[,](#_bookmark0) your will train a classifier using the Naive Bayes method.

## Purpose

* + Understand and practice the principles of Naive Bayes classifier and its training algorithm; compare the trained classification models to see their pros/cons.

## Requirementsi

* + Show the steps and calculations to derive the classifier.
  + Show the formulas you used to calculate the results.

a.(1, L2) What is the prior probability of PlayTennis being yes/no from the given data?

b.(4, L2) What is the conditional probability of the attribute Outlook taking each of the values in *{*Overcast, Sunny, Rain*}* given PlayTennis=yes? Also calculate the conditional probabilities for the attributes {Temp, Humidity, Wind}taking each of its possible values given that PlayTennis=yes.

c.(4, L2) Calculate similiar conditional probabilities asked in (b) with the conditions replaced by PlayTennis=no.

d.(4, L2) Based the results you got from (a)-(c), what are your classification results for the test data given in Table3[?](#_bookmark1)

e.(2, L2) Determine the precision and recall of your classifier.

e.(3, L3) Discuss the pros and cons of the classification models trained using decision trees and Naive Bayes (Name one pro and one con for each model).

# AdaBoost (12 points)

You will be guided through the steps of building an ensemble classifier using AdaBoost. The data points to be classified are given in Table4[.](#_bookmark2) Each classifier in the ensemble will have the form *x < v* or *x > v*. You will select all the data points for each round without replacement.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **x** | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| **y** | -1 | -1 | -1 | 1 | 1 | 1 | 1 | 1 | -1 | -1 |

Table 4: Data points for the AdaBoost problem

## Purpose

* + Understand and practice AdaBoost algorithm by walking through the steps.

## Requirements

* + Show all the steps and calculations needed to derive each classifier.

a.(2, L2) Assume that data weight distribution *D*1 in Round 1 is uniform. Find classifier

*h*1 that has minimum weighted error with data weight distribution *D*1.

b.(2, L2) What is the weighted error rate of classifier *h*1 with data weights *D*1?

c.(2, L2) After re-weighting the data according to the results from Round 1, what is the updated data weight distribution *D*2 for Round 2? Normalize the weights so that they sum to 1.

d.(2, L2) Find classifier *h*2 for Round 2 that have the minimum weighted error rate for the data weight distribution *D*2.

f.(4, L3) What is the ensemble classifier *h* that combines *h*1 and