COMP 307 Assignment 4

Yan Zichu 300476924

Part 1 : Classical Planing

Question 1:

Initial state:
At(Monkey,A) ∧ At(Box,B) ∧ At(Bananas,C) ∧ Height(Monkey,Low) ∧ Height(Box,Lox) ∧ Height(Banana,High)

Goal state:
Hold(Monkey,Banana)

Question 2: Lowercase = variable || Capital letters = value

Go(X,Y)
    Precondition: At(Monkey,X) ∧ ¬Height(Monkey,High) ∧ (X≠ Y)
    Effect: At(Monkey,Y) ∧ ¬At(Monkey,X)

Push(Box,X,Y)
    Precondition: At(Box,X) ∧ At(Monkey,X) ∧ Height(Monkey,Low) ∧ (X≠ Y)
    Effect: At(Box,Y) ∧ At(Monkey,Y)

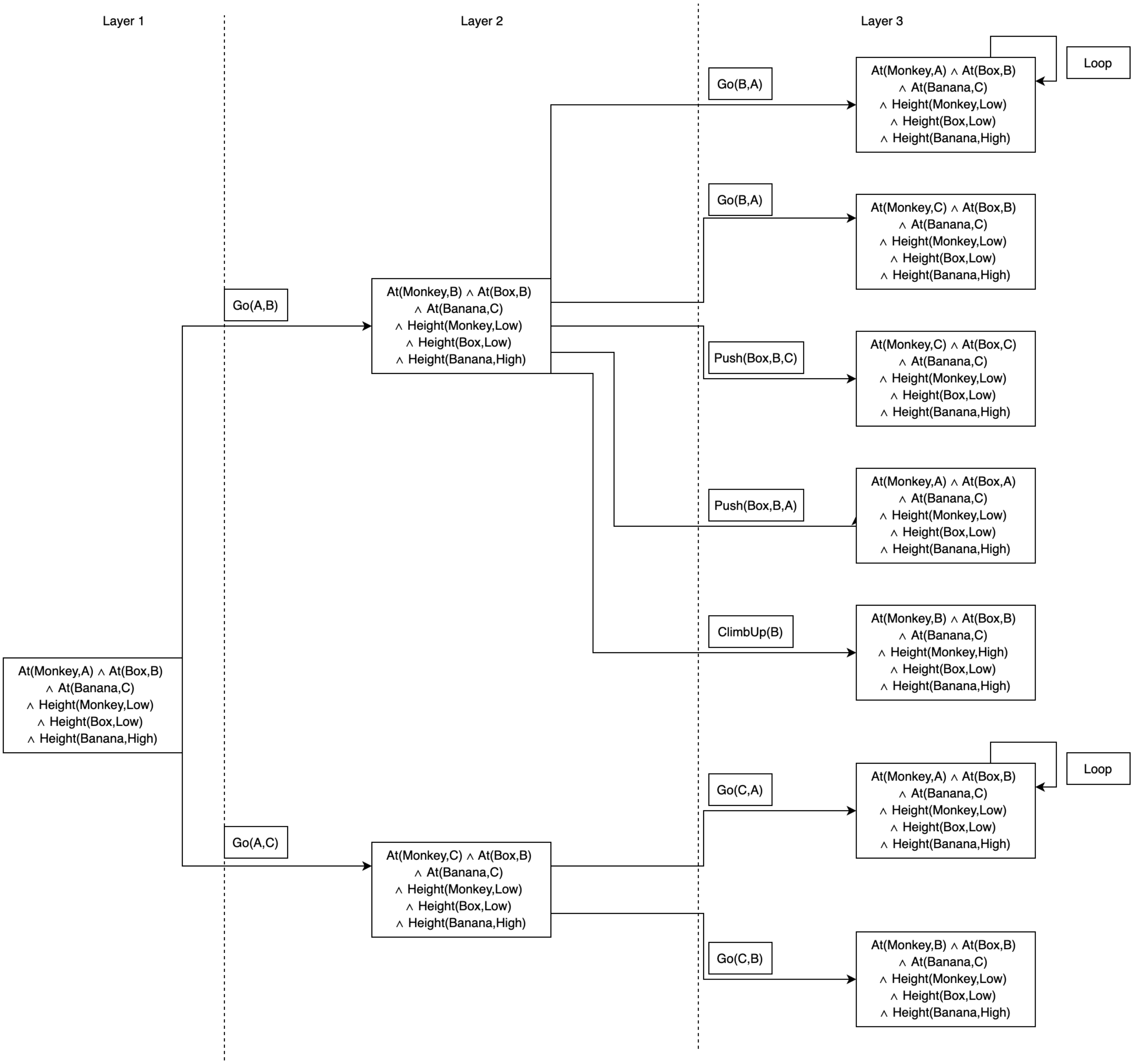
ClimbUp
    Precondition: At(Monkey,X) ∧ At(Box,X) ∧ Height(Monkey,Low)
    Effect: At(Monkey,X) ∧ At(Box,X) ∧ Height(Monkey,High) ∧ ¬Height(Monkey,Low)

ClimbDown
    Precondition: At(Monkey,X) ∧ At(Box,X) ∧ Height(Monkey,High)
    Effect: At(Monkey,X) ∧ At(Box,X)∧ Height(Monkey,Low) ∧ ¬Height(Monkey,High)

Grasp(X,h)
    Precondition: At(Monkey,X) ∧ At(Banana,X) ∧ Height(Monkey,h) ∧ Height(Bananas)
    Effect: Hold(Monkey,Bananas)

Ungrasp(Bananas)
    Precondition: Hold(Monkey,Bananas)
    Effect: ¬Hold(Monkey,Bananas)

Question 3:


Layer 1
Layer 2
Layer 3
At(Monkey,A) ∧ At(Box,B) ∧ At(Banana,C) ∧ Height(Monkey,Low) ∧ Height(Box,Low) ∧ Height(Banana,High)
Go(A,B)
Go(A,C)
At(Monkey,B) ∧ At(Box,B) ∧ At(Banana,C) ∧ Height(Monkey,Low) ∧ Height(Box,Low) ∧ Height(Banana,High)
At(Monkey,C) ∧ At(Box,B) ∧ At(Banana,C) ∧ Height(Monkey,Low) ∧ Height(Box,Low) ∧ Height(Banana,High)
Go(B,A)
At(Monkey,A) ∧ At(Box,B) ∧ At(Banana,C) ∧ Height(Monkey,Low) ∧ Height(Box,Low) ∧ Height(Banana,High)
Loop
Go(B,A)
At(Monkey,C) ∧ At(Box,B) ∧ At(Banana,C) ∧ Height(Monkey,Low) ∧ Height(Box,Low) ∧ Height(Banana,High)
Push(Box,B,C)
At(Monkey,C) ∧ At(Box,C) ∧ At(Banana,C) ∧ Height(Monkey,Low) ∧ Height(Box,Low) ∧ Height(Banana,High)
Push(Box,B,A)
At(Monkey,A) ∧ At(Box,A) ∧ At(Banana,C) ∧ Height(Monkey,Low) ∧ Height(Box,Low) ∧ Height(Banana,High)
ClimbUp(B)
At(Monkey,B) ∧ At(Box,B) ∧ At(Banana,C) ∧ Height(Monkey,High) ∧ Height(Box,Low) ∧ Height(Banana,High)
Go(C,A)
At(Monkey,A) ∧ At(Box,B) ∧ At(Banana,C) ∧ Height(Monkey,Low) ∧ Height(Box,Low) ∧ Height(Banana,High)
Loop
Go(C,B)
At(Monkey,B) ∧ At(Box,B) ∧ At(Banana,C) ∧ Height(Monkey,Low) ∧ Height(Box,Low) ∧ Height(Banana,High)

Question 4:

Initial state: At(Monkey,A) ∧ At(Box,B) ∧ At(Bananas,C) ∧ Height(Monkey,Low) ∧ Height(Box,Lox) ∧ Height(Banana,High) ∧ ¬Hold(Monkey,Banana)

Action1: Go(A,B)

State1: At(Monkey,B) ∧ At(Box,B) ∧ At(Bananas,C) ∧ Height(Monkey,Low) ∧ Height(Box,Lox) ∧ Height(Banana,High)

Action2: Push(Box,B,C)

State2: At(Monkey,C) ∧ At(Box,C) ∧ At(Bananas,C) ∧ Height(Monkey,Low) ∧ Height(Box,Lox) ∧ Height(Banana,High) ∧ ¬Hold(Monkey,Banana)

Action3: ClimbUp(C)

State3: At(Monkey,C) ∧ At(Box,C) ∧ At(Bananas,C) ∧ Height(Monkey,High) ∧ Height(Box,Lox) ∧ Height(Banana,High) ∧ ¬Hold(Monkey,Banana)

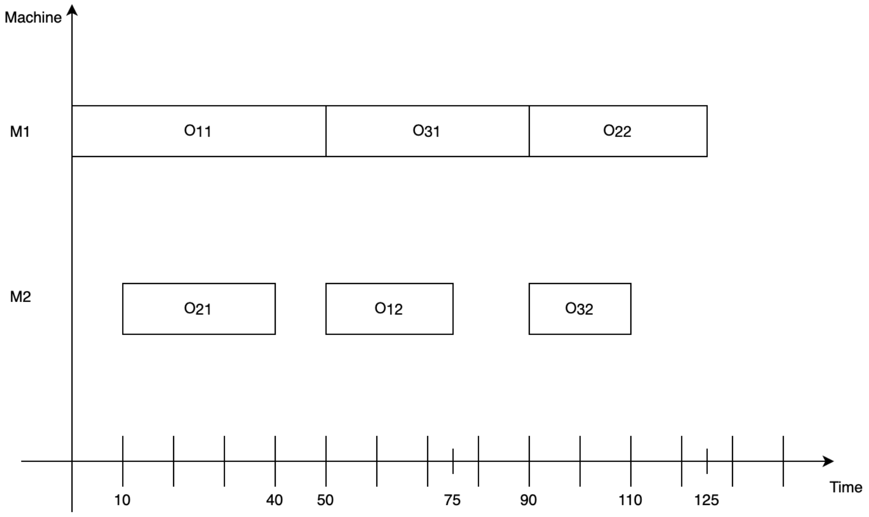
Action4: Grasp(C,High)

Goal state: At(Monkey,C) ∧ At(Box,C) ∧ At(Bananas,C) ∧ Height(Monkey,High) ∧ Height(Box,Lox) ∧ Height(Banana,High) ∧ Hold(Monkey,Banana)

Part 2 : Job Shop Scheduling

Question 1:

Process(O_{11}, M_1, t_1) → Process(O_{21}, M_2, t_2) → Process(O_{31},M_1,t_3) → Process(O_{12},M_2,t_4) → Process(O_{22},M_1,t_5) → Process(O_{32},M_2,t_6).



1.

Job ready time:

J_1: $O_{11} = 0$ $O_{12} = +\infty$

J_2: $O_{21} = +\infty$ $O_{12} = +\infty$

J_3: $O_{31} = +\infty$ $O_{32} = +\infty$

Machine Idle time:

$M_1 = 0$

$M_2 = 0$

Operation(O_{11}, M_1, ProcessTime = 0) = -50

$t_1 = 0$

2.

Job ready time:

J_1: ~~$O_{11} = 0$~~ $O_{12} = 50$

J_2: $O_{21} = 10$ $O_{12} = +\infty$

J_3: $O_{31} = 20$ $O_{32} = +\infty$

Machine Idle time:

$M_1 = 50$

$M_2 = 0$

Operation(O_{21}, M_2, t_2) = Operation(O_{21}, M_2, ProcessTime = 10) = -30

$t_2 = 10$

3.

Job ready time:

J_1: ~~$O_{11} = 0$~~ $O_{12} = 50$

J_2: ~~$O_{21} = 10$~~ $O_{12} = 40$

J_3: $O_{31} = 20$ $O_{32} = +\infty$

Machine Idle time:

$M_1 = 50$

$M_2 = 40$

Operation(O_{31}, M_2, ProcessTime) = -40

$t_3 = 50$

4.

Job ready time:

J_1: ~~$O_{11} = 0$~~ $O_{12} = 50$

J_2: ~~$O_{21} = 10$~~ $O_{12} = 40$

J_3: ~~$O_{31} = 20$~~ $O_{32} = 90$

Machine Idle time:

$M_1 = 90$

$M_2 = 40$

Operation(O_{12}, M_2, ProcessTime) = 25

$t_4 = 50$

5.

Job ready time:

J_1: ~~$O_{11} = 0$~~ ~~$O_{12} = 50$~~

J_2: ~~$O_{21} = 10$~~ $O_{12} = 40$

J_3: ~~$O_{31} = 20$~~ $O_{32} = 90$

Machine Idle time:

$M_1 = 90$

$M_2 = 75$

Operation(O_{22}, M_1, ProcessTime=90) = -35

$t_5 = 90$

6.

Job ready time:

J_3: $O_{32} = 90$

Machine Idle time:

$M_1 = 125$

$M_2 = 75$

Operation(O_{32}, M_2, ProcessTime=90) = -20

$t_6 = 90$

$t_1 = 0$, $t_2 = 10$, $t_3 = 50$, $t_4 = 50$, $t_5 = 90$, $t_6 = 90$

So $t_1 < t_2 < t_3 = t_4 < t_5 = t_6$

Question 2:

Finishing time of $J_1 = t_4 + \text{processTime}(O_{12}) = 75$

Finishing time of $J_2 = t_5 + \text{processTime}(O_{22}) = 125$

Finishing time of $J_3 = t_6 + \text{processTime}(O_{32}) = 110$

The makespan of this solution is the time of the job finished latest which is 125.

Question 3:

Step1:

Partial solution : Process(O_{11}, M_1, ProcessTime=0)

earliest Idle time(M_1)=50, earliest Idle time(M_2)=0

earliest ready time (O_{12})=50,

earliest ready time (O_{21})=10, earliest ready time (O_{22})= +∞

earliest ready time (O_{31})=20, earliest ready time (O_{32})= +∞

Step2:

Partial solution : Process(O_{11}, M_1, ProcessTime=0) ➡ Process(O_{21}, M_2, ProcessTime=10)

earliest Idle time(M_1)=50, earliest Idle time(M_2)=40

earliest ready time (O_{12})=50,

earliest ready time (O_{22})=40,

earliest ready time (O_{31})=20, earliest ready time (O_{32})= +∞

Step3:

Partial solution : Process(O_{11}, M_1, ProcessTime=0) ➡ Process(O_{21}, M_2, ProcessTime=10) ➡ Process(O_{12}, M_2, ProcessTime=50)

earliest Idle time(M_1)=50, earliest Idle time(M_2)=75

earliest ready time (O_{22})=40,

earliest ready time (O_{31})=20, earliest ready time (O_{32})= +∞

Step4:

Partial solution : Process(O_{11}, M_1, ProcessTime=0) ➡ Process(O_{21}, M_2, ProcessTime=10) ➡ Process(O_{12}, M_2, ProcessTime=50) ➡ Process(O_{22}, M_1, ProcessTime=50)

…….

Final solution:
Process(O_{11}, M_1, ProcessTime=0) ➡ Process(O_{21}, M_2, ProcessTime=10) ➡ Process(O_{12}, M_2, ProcessTime=50) ➡ Process(O_{22}, M_1, ProcessTime=50) ➡ Process(O_{31}, M_1, ProcessTime=85) ➡ Process(O_{32}, M_2, ProcessTime=125)

Question 4:
The completion time of J_1 = 75
The completion time of J_2 = 85
The completion time of J_3 = 145

The makespan of SPT is MakeSpan(J_1, J_2, J_3) = 145.
Comparing the makespans of FCFS with SPT, 125<145, in this case, FCDS is better than SPT in makespans.

Question 5:

No, it doesn’t. One solution is better than the other does not mean that the rule that generates the better solution is better than the other rule. It just means one rule is more appropriate than the other rule in this case.

Part 3 : Search Techniques and Machine Learning Basics: Questions

Question1:
1.
(i) Depth-first search. Since the cost from the intermediate state to the target state is unknown, depth-first search can help search from the intermediate state to the target state.
(ii) Breadth first search. Because the cost of going from the initial state to the intermediate state is unknown, breadth first search is suitable for searching the intermediate state.
(iii) Heuristic search. Heuristic functions estimate the cost of going from the current state to the target state.
2.
For workshop issues, local optimal issues is a problem, sometimes we need to get worse local result first and then get best final result in the end. Heuristic search —genetic beam search can help us jump out of local optima and get a better result.
3.
(i) Breadth first search : 1,2,3,4,5,6,7,8,9,10,11,12,13
(ii) Iterative deepening search :
Limit = 0
1
Limit = 1
1,2,3
Limit = 2

1,2,4,5,3,6,7
Limit = 3
1,2,4,8,9,5,10,11,3,6,12,13,7

Question 2:

1.

<u>Possible reason 1:</u>

The choice of k has a great influence on the kNN learning model. If the value of k is too small, the prediction result will be extremely sensitive to noise sample points. In particular, when k is equal to 1, kNN degenerates into the nearest neighbor algorithm, and there is no explicit learning process. If the value of k is too large, there will be larger neighborhood training samples for prediction, which can reduce the reduction of noise sample points; but the training sample points that are farther away will contribute to the prediction result, so that the prediction result will be wrong.

<u>Solution:</u>

Let Michael try different odd k values

<u>Possible reason 2:</u>

The training set is too small, and those domains with smaller sample sizes are more prone to misclassification using this algorithm.

<u>Solution:</u>

Let Michael select more instances to join the training set while ensuring a balanced number of categories of instances.

2.

<u>Possible reason 2:</u>

I think it is probably overfitting.

<u>Solution:</u>

We can do early stop or pruning.

Question 3:

Impurity = P(popular) ∗ P(unpopular)
Feature 1: Mushroom
Yes Impurity =1/5 × 4/5 = 4/25
No Impurity =4/5 × 1/5 = 4/25
Weighted Average Impurity = 4/25 = 16%

Feature 2: Vegetarian
Yes Impurity =1/4 × 3/4 = 3/16
No Impurity =4/6 × 2/6 = 2/9
Weighted Average Impurity = 0.4 × 3/16 + 0.6 ×2/9 = 20.83%

Feature 3: Size
Small Impurity =1/3 × 2/3 = 2/9

Medium Impurity =2/3 × 1/3 = 2/9
Large Impurity =2/3 × 1/3 = 2/9
Weighted Average Impurity = 2/9 = 22.22%

Mushroom is the best feature, it should be chosen for the root of the tree.

Part 4 : Other Topics: Questions

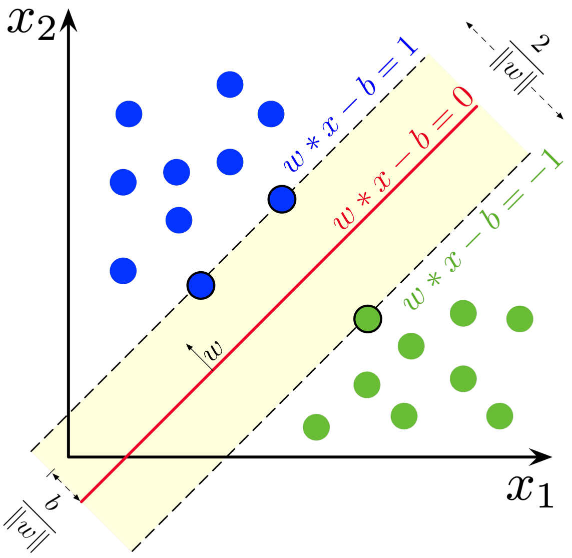
1. Deep learning is an artificial intelligence function that imitates the workings of the human brain in processing data and creating patterns for use in decision making. Deep learning is a subset of machine learning in artificial intelligence (AI) that has networks capable of learning unsupervised from data that is unstructured or unlabeled. Also known as deep neural learning or deep neural network.

   Algorithms and Examples:

   1. CNN (Convolutional neural networks) :
      CNN have been successful in identifying faces, objects, and traffic signs apart from powering vision in robots and self-driving cars.
   2. BP (Back propagation) :
      Let's say that for some reason we want to identify images with a tree. We feed the network with any kind of images and it produces an output. Since we know if the image has actually a tree or not, we can compare the output with our truth and adjust the network. As we pass more and more images, the network will make fewer and fewer mistakes.
   3. MLP (Multilayer Perceptron Neural Network) :
      Image verification and reconstruction; Speech recognition; Machine translation; Data classification.

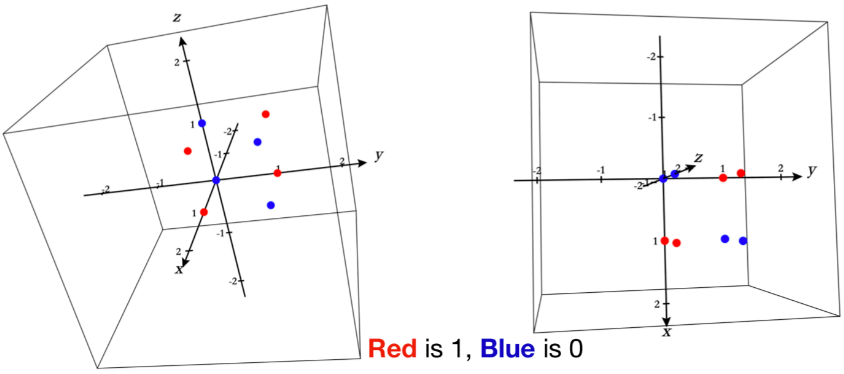
2. In machine learning, support-vector machines (SVMs, also support-vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis.

When data is linear separable, you can use a support vector machine (SVM) when your data has exactly two classes. An SVM classifies data by finding the best hyperplane that separates all data points of one class from those of the other class.
The best hyperplane (the yellow part of the graph below) for an SVM means the one with the largest margin between the two classes.



Reprinted from "Support vector machine," by Wikipedia

For those non-linear separable data I would like to mention is from assignment 2,



Red is 1, Blue is 0

For three-dimensional data like this, we cannot find a plane to distinguish the points of the two colors. *(Graph from my own assignment2)*

3. Text mining :

Example application :

Security applications: Many text mining software packages are marketed for security applications, especially monitoring and analysis of online plain text sources such as Internet news, blogs, etc. for national security purposes. It is also involved in the study of text encryption/decryption.

Algorithm :

Bag of Words (BoW): BoW is all about creating a matrix of words where the words (terms) are represented as the rows and the columns represent the document names. We can then populate the matrix with the frequency of each term within the document, ignoring the grammar and order of the terms.

Natural language processing :

Example application :

Companies like Yahoo and Google filter and classify your emails with NLP by analyzing text in emails that flow through their servers and stopping spam before they even enter your inbox.

Algorithm :

Naive Bayes :

In most cases, NBA in the Natural Language Processing sphere is used for text classification (clustering). The most known task is a spam detection filter. Most solutions in this sphere use the maximum likelihood method to estimate the parameters of Naive Bayesian models:

$$\arg\max\left[P(Q_k)\times\prod_{i=1}^{n}P(x_i \mid Q_k)\right]$$

The first multiplier defines the probability of the text class, and the second one determines the conditional probability of a word depending on the class.

4.

An expert system is software that attempts to provide an answer to a problem, or clarify uncertainties where normally one or more human experts would need to be consulted. Expert systems are most common in a specific problem domain, and is a traditional application and/or subfield of artificial intelligence.

A Decision Support System (DSS) is a class of information systems (including but not limited to computerized systems) that support business and organizational decision-making activities. A properly designed DSS is an interactive software-based system intended to help decision makers compile useful information from a combination of raw data, documents, personal knowledge, or business models to identify and solve problems and make decisions.

In the medical field, a KBS can help doctors more accurately diagnose diseases. These systems are called clinical decision-support systems in the health industry.

5.

Volume defines the huge amount of data that is produced each day by companies, for example. The generation of data is so large and complex that it can no longer be saved or analyzed using conventional data processing methods.

Variety refers to the diversity of data types and data sources. 80 percent of the data in the world today is unstructured and at first glance does not show any indication of relationships. Thanks to Big Data such algorithms, data is able to be sorted in a structured manner and examined for relationships. Data does not always comprise only conventional datasets, but also images, videos and speech recordings.

Velocity refers to the speed with which the data is generated, analyzed and reprocessed. Today this is mostly possible within a fraction of a second, known as real time.

Validity is the guarantee of the data quality or, alternatively, Veracity is the authenticity and credibility of the data. Big Data involves working with all degrees of quality, since the Volume factor usually results in a shortage of quality.

Value denotes the added value for companies. Many companies have recently established their own data platforms, filled their data pools and invested a lot of money in infrastructure. It is now a question of generating business value from their investments.