Part I : Text Processing **[25 marks in total]**

1. **Regular Expressions:** Given the following "regular expression":

   ```
   \s*f?re{1,3}\w
   ```

   For each of the following strings, indicate (with "yes" or "no" in your script book) whether the regular expression will match. [2 marks]

   (a) `freedom` Yes
   (b) `a really good book` Yes
   (c) `sfrew` Yes
   (d) `\tf\tfr\tfre\tfree` Yes

2. **Approximate String Search:** By referring to the definitions from this subject:

   (a) Explain the problem that we wish to solve in "approximate string search", including one typical strategy and any necessary data structure(s). [2 marks]
   (b) Explain why "approximate string search" is a "Knowledge Technology" (or "knowledge task"). [2 marks]

(a) the problem is given a string, how to find a approximately match string from the dictionary, for example, to solve the problem of spelling correction and computational genomics. Strategy: n-grams. Data structures ?

(b) The solution of it is ill-defined and ambiguous since it is not a exact string match method, the result may be different, and the context is critical for example, the exact word a user want is different in different context no single solution provided and computer media between users and data

continued . . .

3

(a) it → A(1)
happens → A(1)
over → A(3) → B(1) → C(4)
and → A(2) → B(2)
again → A(1)
under → B(1)
in → B(1)
out → B(1)

(1, 1, 3, 2, 1, 0, 0.v)
(0, 0, 1, 2, 0, 1, 1, 1)
(0, 0, 4, 0, 0, 0, 0, 0)

(b) ① Fetch the inverted list for each query
Over → A(3) → B(1) → C(4)
and → A(2) → B(2)
out → B(1)
② the conjunction may be: over AND and AND out
so the intersection of these three list is {B}

(c) $W_1 = [1, 1, 3, 2, 1, 0, 0, 0]$
$W_2 = [0, 0, 1, 2, 1, 1, 1, 0]$
$W_3 = [0, 0, 4, 0, 0, 0, 0, 0]$
$W_q = [0, 0, 1, \frac{3}{2}, 3, 0, 0, 0]$

$\cos(1, q) = \frac{3+3+3}{4 \cdot \frac{7}{2}} = \frac{9}{4 \cdot \frac{7}{2}}$

$\cos(2, q) = \frac{1+3+3}{3 \cdot \frac{7}{2}} = \frac{\frac{7}{3} \cdot 4}{4 \cdot \frac{7}{2}}$

$\cos(3, q) = \frac{4}{4 \cdot \frac{7}{2}}$

∴ rank: doc2 > doc1 > doc3

4. (a) $W_{q,t} \times W_{d,t}$ is store in the Accumulator, which can provide more weight to document where the query terms appear many time and more weight for terms appear in little specific document. With accumulator, we can provide ranking more efficiently since we reduce the number of document we need to consider each step. The document with a higher result generated by its accumulator is thought to have a higher ranki

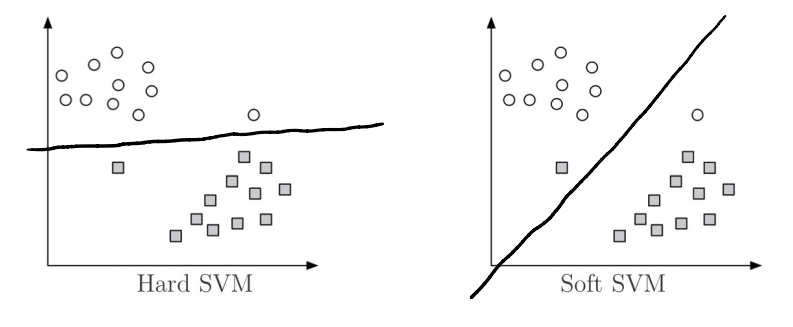
(b) cost time and space, not efficient.

(c) The "Limitting" approach

① create a empty set of Accumatator and set a limit L
② For each query term t, order by decerasing $W_{q,t}$
fetch the inverted list of t
for each $\langle d, f_{d,t} \rangle$
if $|A| < L$, generate A_d if d has no accumulator
$A \leftarrow A + W_{qt} \times W_{d,t}$
$A_d \leftarrow A_d / W_d$
③ return the r documents with the greatest A_d (order by decreasing A

5.

(a)



(b) Yes. because soft SVM relax the notion of linear separability, A small number of point are allowed to be in the wrong side of the line; Soft Margin SVMs are less susceptibl to noise because they introduce a slack variable that allow mis-classification

(c) ① linear separable
② Only contain two classes, A binary classificate problem

(d) main parameters in the "primal":

$\frac{b}{\|w\|}$ and w

parameter in "dual":

$$f(x) = \sum a_i y_i x_i^T x + b$$

Each non-zero a_i indicates that corresponding x_i is a support vector

6 ① consider 1 element: (support $\geq \frac{1}{3}$)

null → morning
evening
coffee
tea
pastry

$\emptyset$ → not

② consider 2 elements: (Support $\geq \frac{1}{3}$)
(morning, pastry) (coffee, pastry).

③ consider 3 elements: (support $\geq \frac{1}{3}$)
none

④ confidence $\geq \frac{3}{4}$:

(morning → pastry)

(coffee → pastry)

7.①

	P1	P2	P3	P4 P5
P1	0	1	5	10
P2	1	0	3.5	8
P3	5	3.5	0	4
P4 P5	10	8	4	0

②

	P1 P2	P3	P4 P5
P1 P2	0	5	10
P3	5	0	4
P4 P5	10	4	0

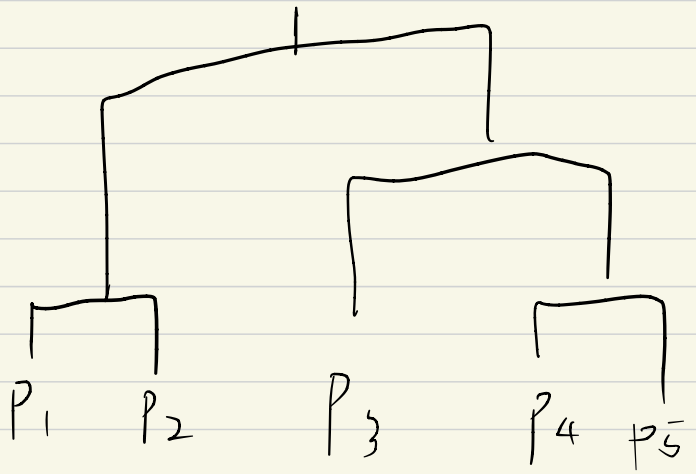
③

	P1 P2	P3 P4 P5
P1 P2	0	10
P3 P4 P5	10	0

④

	P1 P2 P3 P4 P5
P1	0
P2	
P3	
P4	
P5	

Complete link



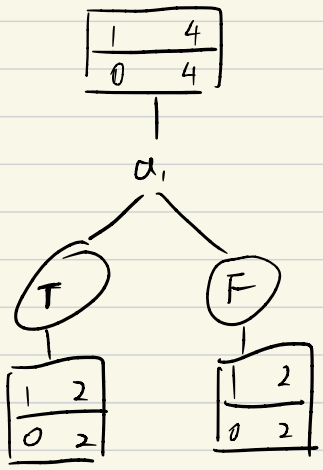
8. The features (attributes) in the data set is conditional independent to each other.

It is necessary because $P(x_1, x_2 \cdots x_n | c) = \prod P(x_i | c)$ only if $(x_i | c)$ is conditional independent to each other.

Because, it is hard to ensure features in the real world are conditional independent to each other, they can be correlative to each other.

Q. $H(LABEL) = -\left(\frac{1}{2}\log_2\frac{1}{2}\right)\times 2 = 1$

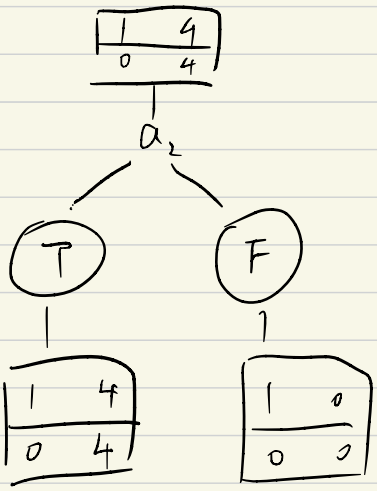
a_1:



$$-\left(\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}\right)\cdot\frac{4}{8}\cdot 2 = 1$$

$$IG(a_1 \mid R) = 1 - 1 = 0$$

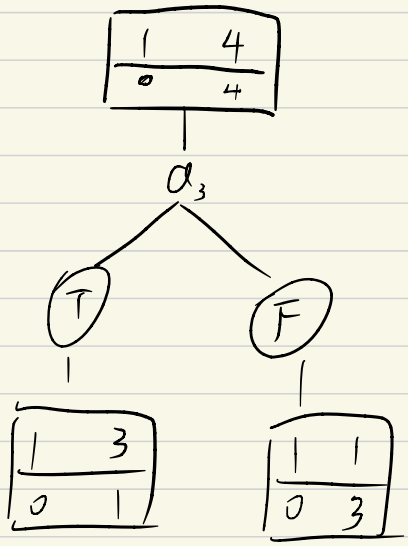
a_2:



$$-\left(\frac{1}{2}\log\frac{1}{2}\right)\times 2 = 1$$

$$IG(a_2 \mid LABEL) = 1 - 1 = 0$$

a_3



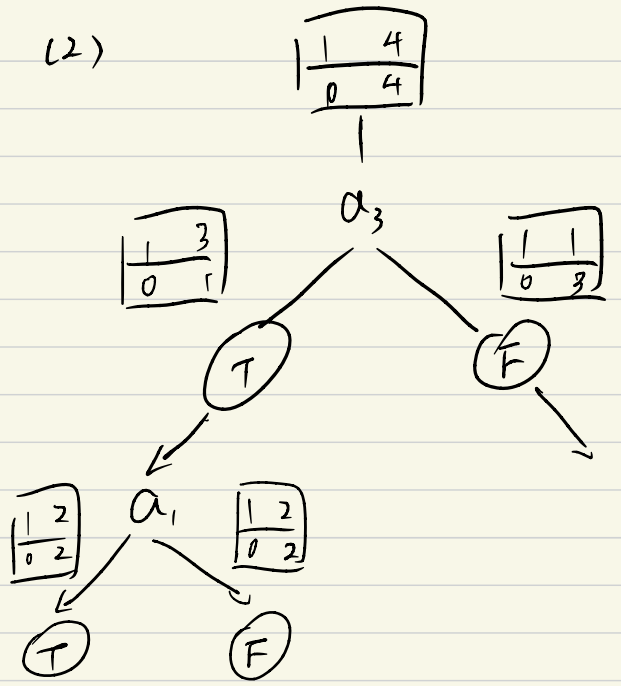
$$-\frac{1}{2}\left(\frac{1}{4}\log_2\frac{1}{4} + \frac{3}{4}\log_2\frac{3}{4}\right) + \left[-\frac{1}{2}\left(\frac{3}{4}\log_2\frac{3}{4} + \frac{1}{4}\log_2\frac{1}{4}\right)\right]$$

$$= 0.81$$

$$IG(a_3 \mid LABEL) = 1 - 0.81 = 0.19$$

So a_3 is the root.

(2)



No, a_2, a_1 can not give any information when $a_1 a_2 a_3$ = TTF, Label can be 1 or 0.

10 ① initialize a weight
② Forward pass, calculate the hidden Hi and output.
③ Calculate the error between ok and t_k, update the weighe in a specific learn rate, a small learning weight would lead to a local minimal, Large learning weight load to over shooting, until network has converged.

11. (a) SVM is appropriate, SVM can handle a huge attribute set in an affitiently. effiently to find the weight of attribut.

(b) K-Nearest Neighbour is inappropriate too many attributes which would make similarity become useless it it not right to assumes each attribute to have the same weight, calculating similarity in a high dimension is computational heavy

(c) not appropriate, two many attributes, it would need a lot of time to make the net converged

(d) not appropriate, Too many attributes, probabilities based on numeric values will be essentially random. it is over sensitive to redundant and irrelevant attrs

(b) Yes, there must be some book highly correlative to subject, by using this book to predict, for example, in DT tree, throwing those books with high IG can provide good performance

One-R we may use the keyword or tag of it to represent it, but some time user may search it based on their content.

Goal:

① Relevance

② Novelty

③ serendipitous,

④ increase recommendation diversity.