資料探勘 Mid term

tags: 資料探勘

Association Rule

FP Growth

> Frequent Pattern Growth 規則
>
> Let a be a frequent itemset in DB, B be a's conditional pattern base, and b be an itemset in B. Then a 聯集 b is a frequent itemset in DB iff b is frequent in B.

why FP growth the winner?

1. Divide and Conquer (根據目前已知freq itemsets細分後找出所有子集freq)
2. 不用找candidate?
3. Compressed database?
4. 不用重複掃描整個database
5. 找出local freq items，建立sub fp tree，沒有pattern search and matching(第二次掃DB時已經將完整tree建立完成)

以下重點(why?)

> 1. 為甚麼要sort 1-itemset (by support)?
> 2. descent order方式建立fp growth?
>   - 當items的count相同，如何排序?

Multi-level Association Rule

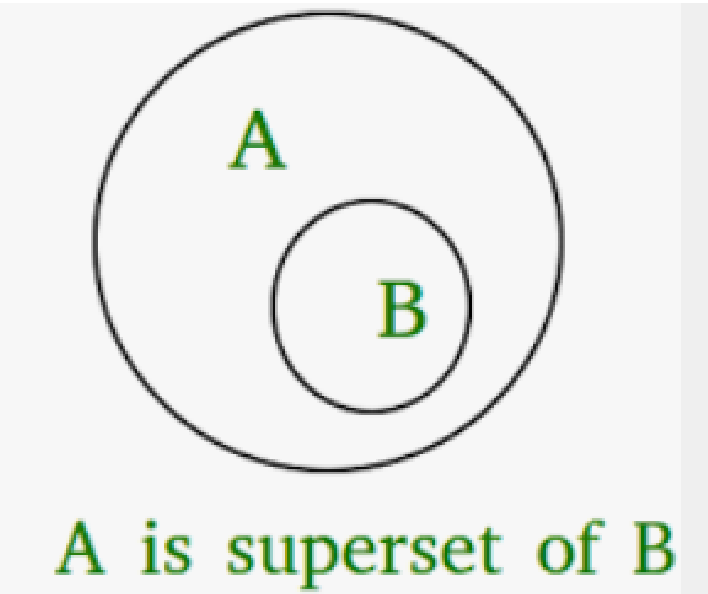
high level 的問題
相同的support值會產生很多的frequent itemsets(產生很多沒有很重要的關聯)
愈高level的item愈容易滿足min support?

uniform support 會遇到兩個問題

1. 設太高 -> 只有high level會留下
2. 設太低 -> 太多freq itemsets

Reduced Support : 4 strategies

```
A set A is a superset of another set B if all elements of the set B are elements of the s
```



Max-patterns
freq patterns without frequent super pattern。
如BCDE is max-pattern，but BCD not(even frequent as well)

Closed frequent itemsets
An itemset is closed in a data set if there exists no superset that has the same support count as this original itemset.(較寬鬆，即便superset有超過min support但不及original set，就是closed)

max patterns 和 closed frequent itemset差在哪?

`Frequent item set X∈F is maximal if it does not have any frequent supersets.`
`Frequent item set X ∈ F is closed if it has no superset with the same frequency`

A(3) → AB(3) , AC(3), AD(2)

A(count) is not greater than its immediate superset.

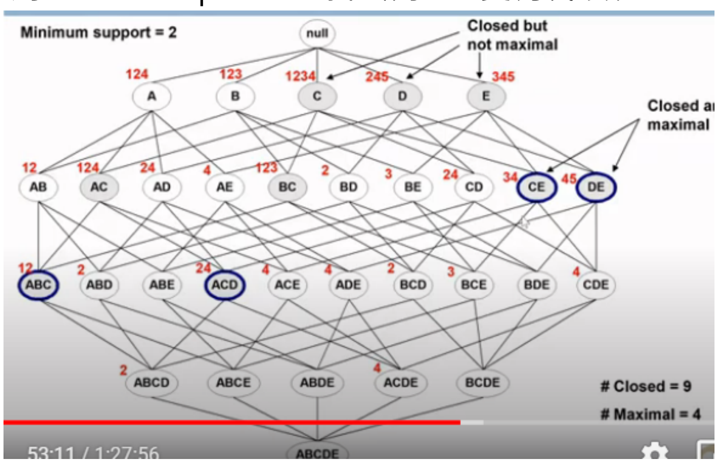
A is not closed.

In A's immediate superset, itemset are present with min.

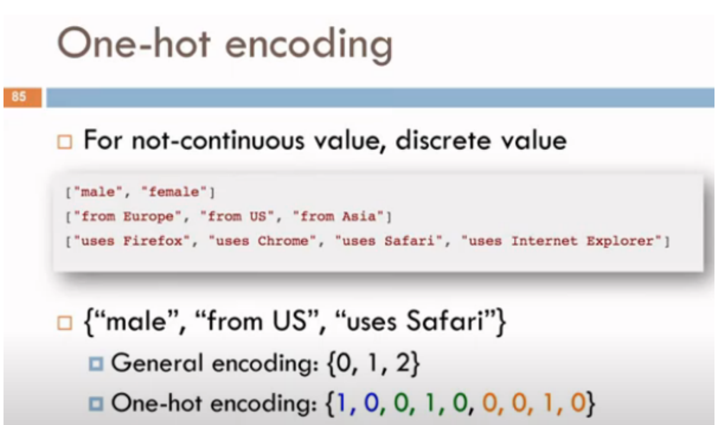
support count i.e. 3.

A is not maximal

用closed freq itemset找出的rule更有代表性。



Quantitative 關聯法則



問題:

當attribute被切的很多，資料本身各item的 support value很低，confidence很容易就很高 (attribute的 support value低)

Text Analysis

Inverted index:
給定文字，輸出output為文章id及在文章內位置

Lexical processing

1. tokenization
2. stemming (複數 字根 去除等。)
3. removing stop words 降低size reduction

TF-IDF
IDFj = log(total documents in the set / docus which contain the term W)

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

$tf_{i,j}$ = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

BM25

```
TF-IDF 的複雜版。算兩個向量的SCORE
```

LSA & LSI example

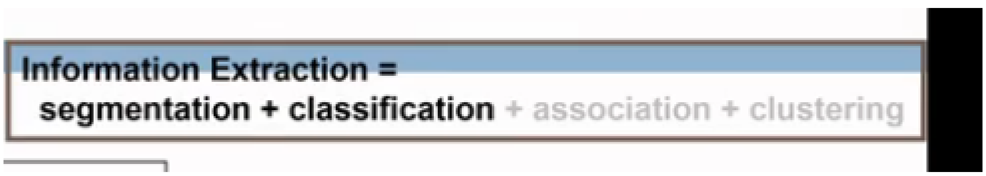
```
svd -> 無法運算大量文本
```

word embedding

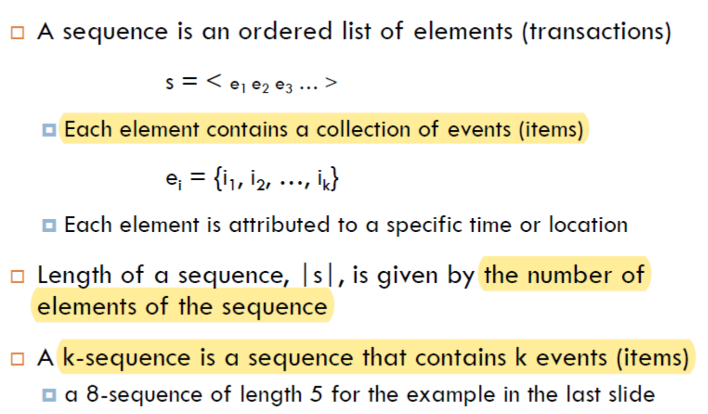
問題:
遇到沒看過的字詞(out of bag)，沒有分辨及預測力

information extraction

workflow
1. 斷字和辭意分系(lexical analysis)
2. paper name idenfication
3. shallow parsing? (syntactic analysis)
4. building relations
5. inferencing?



Sequence Pattern




```
element是時間t的大單位，一個element細分為多個items
```

Subsequence

Formal Definition of a **Subsequence**

- A sequence $<a_1\, a_2 \ldots a_n>$ is contained in another sequence $<b_1\, b_2 \ldots b_m>$ $(m \geq n)$ if there exist integers $i_1 < i_2 < \ldots < i_n$ such that $a_1 \subseteq b_{i1}$, $a_2 \subseteq b_{i1}$, …, $a_n \subseteq b_{in}$

Data sequence	Subsequence	Contain?
< {2,4} {3,5,6} {8} >	< {2} {3,5} >	Yes
< {1,2} {3,4} >	< {1} {2} >	No
< {2,4} {2,4} {2,5} >	< {2} {4} >	Yes

- The support of a subsequence w is defined as the fraction of data sequences that contain w
- A *sequential pattern* is a frequent subsequence (i.e., a subsequence whose support is ≥ *minsup*)

i1=1 i2=2 i3=4
b = {Milk,Bread}{Apples}{Sausages}{Beer,Bread}
a = {Milk}{Apples}{Bread}

Sequential pattern mining 目標為?

`給定一組序列，找出所有其frequent subsequences`

- Given a set of sequences, find the complete set of frequent subsequences

A *sequence database*

SID	sequence
10	<a(abc)(ac)d(cf)>
20	<(ad)c(bc)(ae)>
30	<(ef)(ab)(df)cb>
40	<eg(af)cbc>

A *sequence* : < (ef) (ab) (df) c b >

An element may contain a set of items. Items within an element are unordered and we list them alphabetically.

<a(bc)dc> is a *subsequence* of <a(abc)(ac)d(cf)>

Given *support threshold* *min_sup* =2, <(ab)c> is a *sequential pattern*

Challenge

`1. 計算量大 2. many scan of databases 3. 長序列準度問題`

- Given a sequence: <{a b} {c d e} {f} {g h i}>
  - Examples of subsequences: <{a} {c d} {f} {g} >, < {c d e} >, < {b} {g} >, etc.
- How many k-subsequences can be extracted from a given n-sequence?

<{a b} {c d e} {f} {g h i}> n = 9

k=4:

<{a} {d, e} {f}>

Answer :

$$\binom{n}{k} = \binom{9}{4} = 126$$

algorithm

特殊情況 將items 做mapping時將同個element中大於兩個freq item
皆做組合

Customer Id	Original Customer Sequence	Transformed Customer Sequence	After Mapping
1	< (30) (90) >	< {(30)} {(90)} >	< {1} {5} >
2	< (10 20) (30) (40 60 70) >	< {(30)} {(40) (70) (40 70)} >	< {1} {2, 3, 4} >
3	< (30 50 70) >	< {(30), (70)} >	< {1, 3} >
4	< (30) (40 70) (90) >	< {(30)} {(40) (70) (40 70)} {(90)} >	< {1} {2, 3, 4} {5} >
5	< (90) >	< {(90)} >	< {5} >

注意:

(3)(5)是兩個不同時間的pattern，不是(3 5)的子集

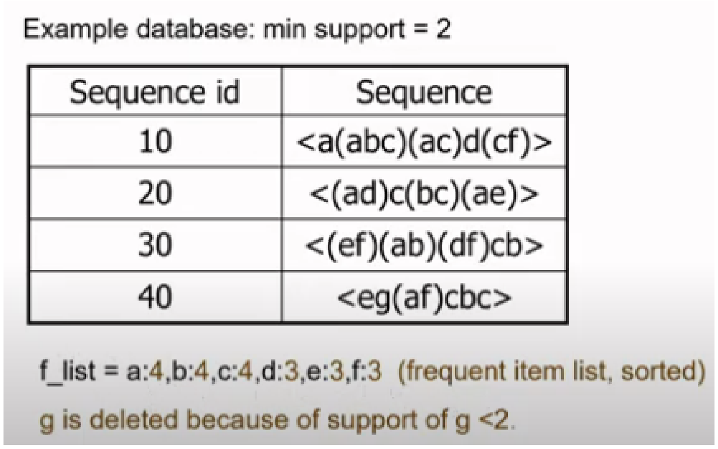
Maximal Sequence

- <(3) (4 5) (8)> is contained by <(7) (3 8) (9) (4 5 6) (8)>
- <(3) (5)> is not contained in <(35)>, and vice versa
- In a set of sequences, a sequence s is maximal if s is not contained in any other sequences in the set

FreeSpan

運用概念 pattern projected

1. 將各序列依照item分別映射(project)到更小的projected database
2. 根據projected database繼續往下長subsequence
3. divide and conquer作法
4. 可以將完整的序列資料分成各種subset。

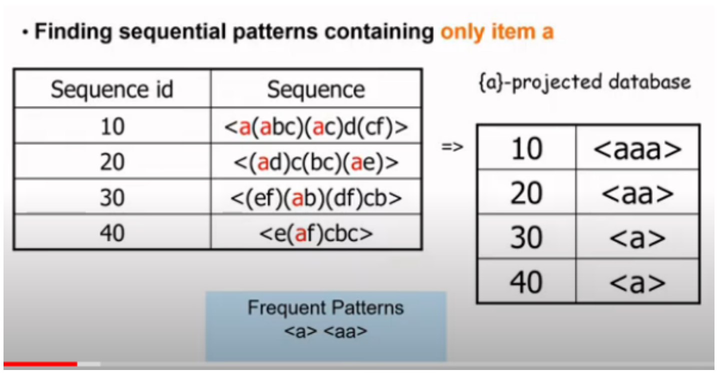


Example database: min support = 2

Sequence id	Sequence
10	<a(abc)(ac)d(cf)>
20	<(ad)c(bc)(ae)>
30	<(ef)(ab)(df)cb>
40	<eg(af)cbc>

f_list = a:4,b:4,c:4,d:3,e:3,f:3 (frequent item list, sorted)

g is deleted because of support of g <2.



• Finding sequential patterns containing only item a

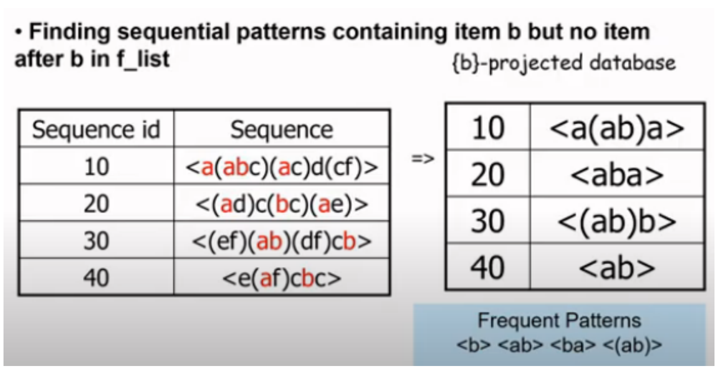
Sequence id	Sequence
10	<a(abc)(ac)d(cf)>
20	<(ad)c(bc)(ae)>
30	<(ef)(ab)(df)cb>
40	<e(af)cbc>

=>

{a}-projected database

10	<aaa>
20	<aa>
30	<a>
40	<a>

Frequent Patterns
<a> <aa>



• Finding sequential patterns containing item b but no item after b in f_list

Sequence id	Sequence
10	<a(abc)(ac)d(cf)>
20	<(ad)c(bc)(ae)>
30	<(ef)(ab)(df)cb>
40	<e(af)cbc>

=>

{b}-projected database

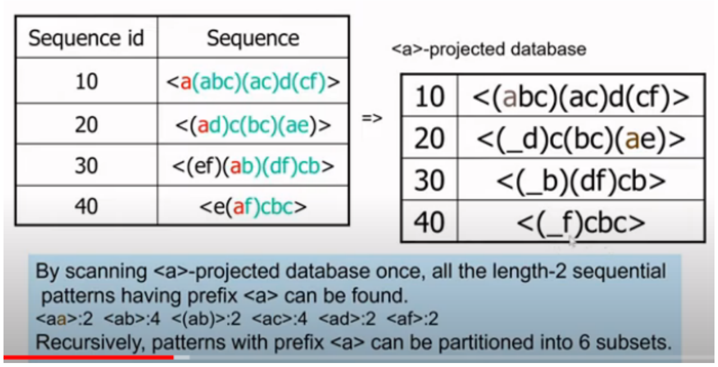
10	<a(ab)a>
20	<aba>
30	<(ab)b>
40	<ab>

Frequent Patterns
<b> <ab> <ba> <(ab)>

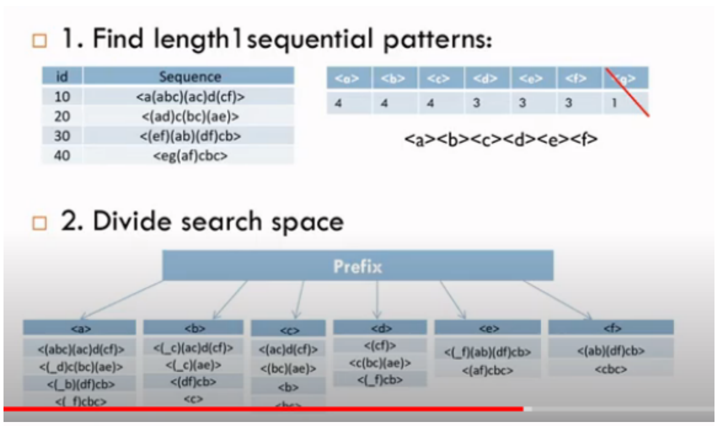
Prefix Span

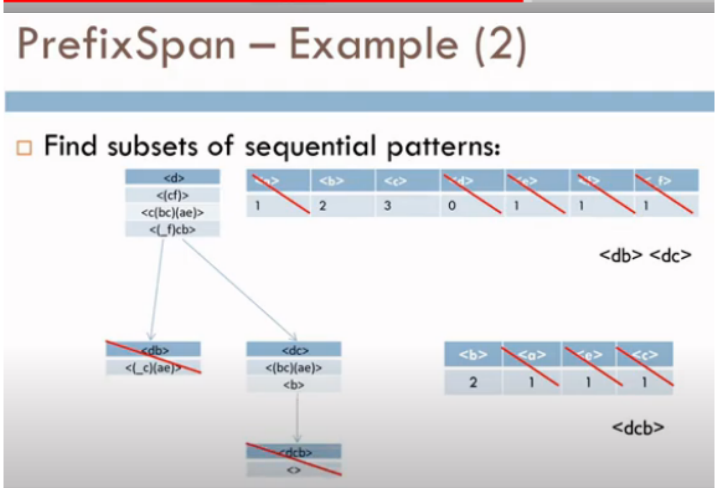
優勢:

1. no candidate subsets to be generated
2. projected DBs keep shrinking



每次針對item建立projected DB 時可以找到subset





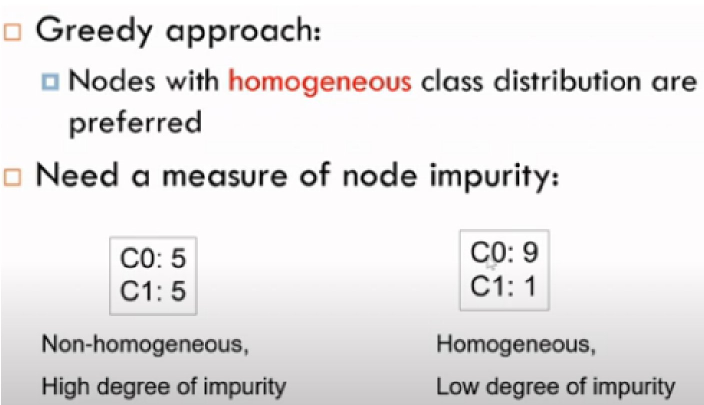
prefix span 精神:
先用prefix分別找projection db -> divide and conquer
從db找Sequential pattern -> 和prefix 組合也是sp

先把答案整理好，一個個往下做，和其他條獨立，很快收斂，速度快。

Machine Learning

決策樹

1. hunt's algo : 隨機選擇feature去分類 —> overfitting
2. Greedy Strategy
   split the records based on an attribute test that optimizes certain criterion
   就是找到一個最佳的attribute可以使得目標被最大滿足(min | max)
   (在這時間點最好的解)
   nodes with homogeneous class distribution are preferred
   利用node impurity計算不純度



常用計算node impurity算法

Measures of Node Impurity

- Gini Index $$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$
- Entropy $$Entropy(t) = -\sum_j p(j \mid t) \log p(j \mid t)$$
- Misclassification error $$Error(t) = 1 - \max_j p(j \mid t)$$

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1
Gini = 1 – $P(C1)^2$ – $P(C2)^2$ = 1 – 0 – 1 = 0

C1	**1**
C2	**5**

P(C1) = 1/6 P(C2) = 5/6
Gini = 1 – $(1/6)^2$ – $(5/6)^2$ = 0.278

C1	**2**
C2	**4**

P(C1) = 2/6 P(C2) = 4/6
Gini = 1 – $(2/6)^2$ – $(4/6)^2$ = 0.444

```
決策樹訓練的目標函數為
information gain = parent node entropy - weighted sum entropy(選擇能將info gain最大化的feat
```

gini 和 entropy計算方式皆prefer splits that result in large num of partitions, each being small but pure。

leaf node (stop) criterion

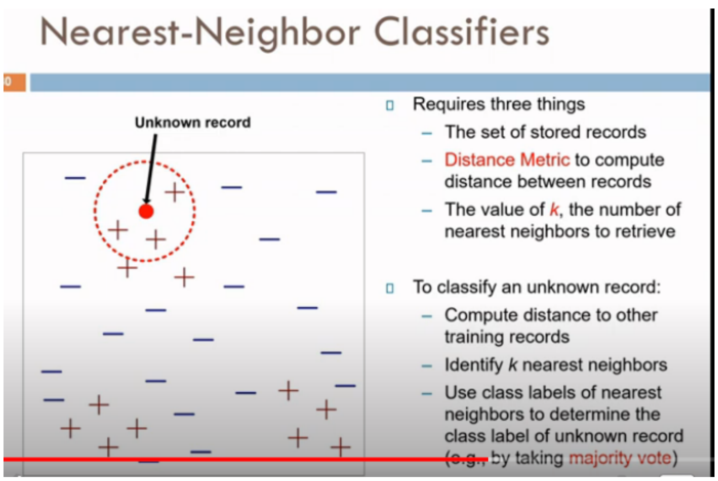
1. 當劃分後每筆資料都是同個類別
2. 當劃分後每筆資料都有相同的features
3. early stopping -> reduce overfitting

```
優點：
1. 計算快速
2. 可以很簡單的解釋data
3. 表現和其他分類模型不會差很多
4. 對於symbolic feature表現特好。

問題:
1. 有缺值對tree的訓練影響很大。
2. nodes次數越多，愈容易overfitting。
3. 如果feature交互作用才對結果有影響，決策樹沒辦法分類。
4. 代表DT僅能找出單一feature對結果的影響。
5. 對noise 很sensitive

解決overfitting
1. pre-pruning
    用更嚴謹的方式設定停損點
2. ...
```

KNN



K值選取tricks

1. 如果k 太小，則很有可能會因為鄰近為noise data產生錯誤分類
2. k太大也可能因為選到距離太遠的feature(與自己太不像了還要選)

貝氏

直接假設各feature之間條件獨立。

- Assume independence among attributes A_i when class is given:
  - $P(A_1, A_2, \ldots, A_n | C) = P(A_1| C_j) P(A_2| C_j)\ldots P(A_n| C_j)$
  - Can estimate $P(A_i| C_j)$ for all A_i and C_j.
  - New point is classified to C_j if $P(C_j) \prod P(A_i| C_j)$ is maximal.

Naïve Bayes Classifier

- If one of the conditional probability is zero, then the entire expression becomes zero
- Probability estimation:

$$\text{Original}: P(A_i | C) = \frac{N_{ic}}{N_c}$$

$$\text{Laplace}: P(A_i | C) = \frac{N_{ic}+1}{N_c+c}$$

$$\text{m-estimate}: P(A_i | C) = \frac{N_{ic}+mp}{N_c+m}$$

c: number of classes
p: prior probability
m: parameter

優點:

1. robust to noise
2. 能處理missing value(計算時後當作1e-6等？)

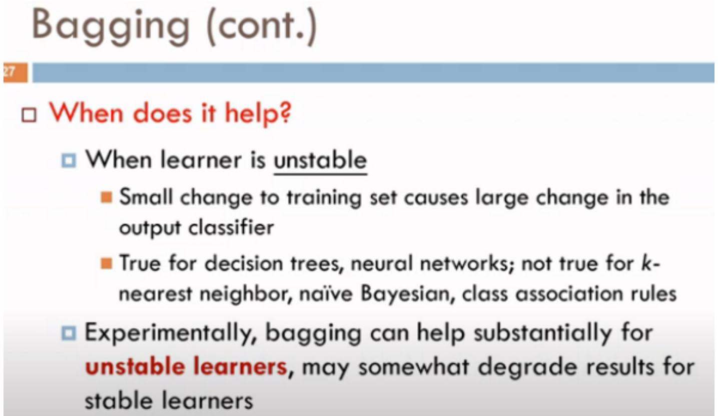
ensemble

弱分類器用majority vote方式決定label，集成。
要讓整體error rate降低， 分類器之間越獨立越好 。

Bagging

1. 隨機從data 選出不放回的方式取k個sample

when does it help?




```
用一堆unstable weak learner反而可以有好結果??
```

Boosting

> 也要求base learner 是unstable(sentitive to noise)，和bagging相同。
>
> boosting更容易受到noise影響。因為noise太多重新調整weight去訓練那些都是noise的data，模型成果可能會下降。

Adaboost

semi-supervised, unsupervised

1. small labeled data and 大量unlabeled data (LU learning)
2. 只有positive 和 一堆unlabeled data (PU learning)

解決少量label的問題 (label 生成)

1. label propagation(有點像KNN)
2. spy technique

The spy technique

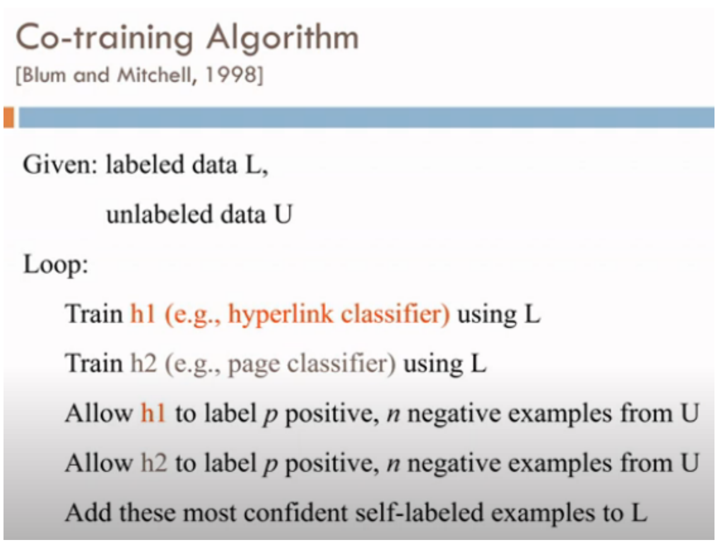
```
分類器要有rank data的能力，單純分類不適用

把一些positive混進unlabeled data，讓分類器對於unlabeled做ranking，可找出與positive差異最大的da
```

1-DNF method

```
從positive docu 找出一組字W，這些字出現頻率比unlabeled的頻率還要高。而那些完全沒包刮W的unlabelec
```

Co-training Algo



同時訓練兩個不同的分類器，將資料分別給分類器訓練
而分類器會將信心值最高的分類結果返回原training data，(augmented)，繼續訓練。