

資料探勘 Mid-term --2

tags: 資料探勘

□ Recall:

▣ The fraction of the **relevant documents (R)** which has been retrieved

□ Precision:

▣ The fraction of the **retrieved documents (A)** which is relevant

$$\text{precision} = \frac{\text{true positives}}{\text{no. of predicted positive}}$$

$$\rightarrow \text{recall} = \frac{\text{true positives}}{\text{no. of actual positive}}$$

Precision - recall trade-off plot

precision & recall 會因為不同情況對threshold設定大小的關係有不同的變動，例如僅有在非常高機率的預測下才會predict 1，則precision 會很高，但是recall 反而會下降(因為很多真實為1的都沒有predict為1，門檻太高。)

Trading off precision and recall

→ Logistic regression: $0 \leq h_{\theta}(x) \leq 1$

Predict 1 if $h_{\theta}(x) \geq 0.5$ *0.7 0.9*

Predict 0 if $h_{\theta}(x) < 0.5$ *0.4 0.9*

Suppose we want to predict $y = 1$ (cancer) only if very confident.

→ Higher precision, lower recall.

Suppose we want to avoid missing too many cases of cancer (avoid false negatives).

precision & recall curve 資訊檢索 (<https://ithelp.ithome.com.tw/articles/10192869>)

Average Precision

在取出relevant的情況下，平均的precision。(每次的檢索結果會依照各個docu累進計算當下的precision & recall)

MAP

不同QUERY下，AP的平均 -> 考量每一個狀況的全盤指標

Average Precision 問題

1. 不能偵測單一不正常分類的部分
2. 預知道特定query的表現

Precision at k

針對搜尋引擎的問題，更在乎在總檢所文章總數為k時，precision為多少。
(特定的前幾筆文章，用precision at k衡量較能符合使用者感受)

但是一旦relevant文章總數高，自然precision at k也會高。

R-Precision

the precision at the R-th position in the ranking

將K設為relevant文章總數，則precision = recall(break even pnt)

marcoaveraging: 重視種類

所有類別的每一個統計指標值的算數平均值

microaveraging: 重視量

針對data所有instance不分類別得做confusion matrix

sensitivity : 所有ground truth為Positive的data中，總共有多少比例被正確分類為positive。

Specificity : 所有ground truth為negative總共正確抓出多少比例的true negative。

sample class不balance時用accu不能完整表達模型的預測能力

ROC curve

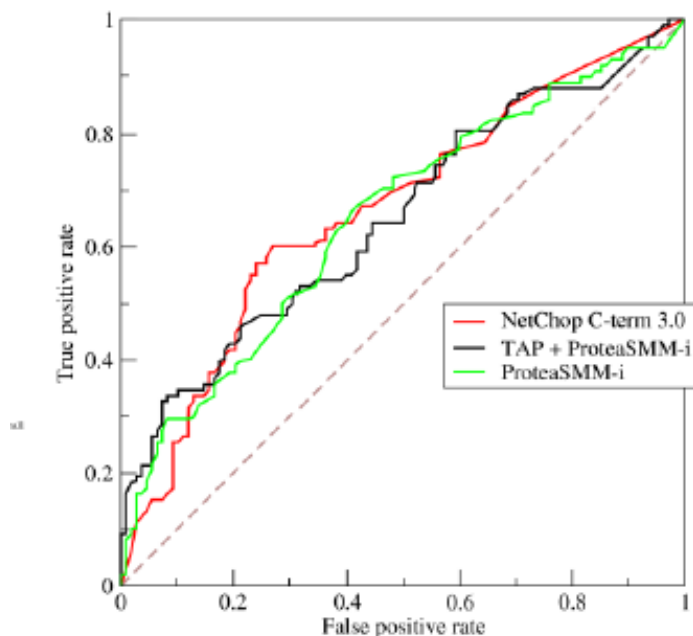
一個sensitivity vs (1-specificity) 曲線，曲線變動來自於對於分類threshold的設定大小變動。

(true positive vs false positive rate)

y軸為答案是1，也正確猜1的比例，x是答案是0，但錯誤地猜成1的比例。

如果threshold設很低(很容易就猜1)，則sensitivity很高，但(1-specificity)也很高。

把threshold設更高，兩者可能皆會降低，要找一個threshold，使地有最大true posi/false posi 比例。



對於imbalanced data，利用Precision or recall有更好地解釋能力。

auc 幫助判斷哪個分類器表現更好。(與閥值設定無關)

在非常不balanced的data用roc做比較都有較stable的解釋力

Q1 : f1 score 和 break even pnt關係？

break even point -> precision = recall = f1 score

$$F1 = \frac{2 * precision * recall}{precision + recall}$$

- Q: Prove that the F1 is equal to the Dice coefficient of the retrieved and relevant document sets.
 - ▣ $\text{Dice}(X, Y) = 2|X \cap Y| / (|X| + |Y|)$
- A:
 - ▣ $F1 = 2PR / (P + R)$, $P = tp / (tp + fp)$, $R = tp / (tp + fn) \rightarrow F1 = 2tp / (2tp + fp + fn)$
 - ▣ $|x| = tp + fp$, $|y| = tp + fn \rightarrow \text{Dice}(x, y) = tp / (2tp + fp + fn)$

Methods of Estimation

- Holdout
 - ▣ Reserve 2/3 for training and 1/3 for testing
- Random subsampling
 - ▣ Repeated holdout
- Cross validation
 - ▣ Partition data into k disjoint subsets
 - ▣ k-fold: train on k-1 partitions, test on the remaining one
 - ▣ Leave-one-out (LOOCV): $k = n$
- Stratified sampling
 - ▣ oversampling vs undersampling
- Bootstrap
 - ▣ Sampling with replacement

Ranked list

1. NDCG

度量一個query中各個docu的gain，根據docu在預測中排序的位置

DCG example

- D1, D2, D3, D4, D5 with relevance score **2, 1, 0, 2, 0** (2: highly relevance, 1: relevance, 0: non-relevance)
- DCG_5 of this list = $2 + (1/1 + 0/\log_2 3 + 2/\log_2 4 + 0/\log_2 5)$
= $2 + 1 + 1 = 4$
- **Ideal order** (2,2,1,0,0 perfect) $IDCG_5 = 2 + 2 + 1/\log_2 3 = 4.63$
- **NDCG=Normalized** $DCG_5 = DCG_5 / IDCG_5 = 4/4.63 = 0.86$
- What are NDCGs of lists (1, 2, 2, 0, 0) and (**2, 1**, 0, 2, 0) ?

2. Kendall-tau

度量兩組具順序的list之間的關聯性。

Kendall-tau

- measure the association between two measured quantities
- $(\# \text{concordant} - \# \text{discordant}) / (n(n+1)/2)$
- E.g.,
 - Ground truth : 1 2 3 4 5, Result list: 2 1 5 3 4
 - $\# \text{concordant} = 7, \# \text{discordant} = 3, \text{Kendall-tau} = (7-3)/10 = 0.4$
 - Try another list 2 1 3 4 5
- Sensitive to few bad ranked results
- Compare: Rand Index

$$R = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{n}{2}}$$



共有 C_n^2 種組合。(兩組中一致的組合順序相同，視為一組 concordant)

問題：如果有一些 bad ranked data，則 kendall 數值下降很快。
(sensitive)

Cohen's Kappa

== 度量兩個 raters 之間的同意一致性。其中一個 rater 為分類器，另一個為 ground truth。

假設兩個 raters 的決定互相獨立，可以算期望的 agreement。

□ Agreement $\Pr(a) = (10+15)/30=0.83$

□ $\Pr(e)$

□ $P(A=Y)=10/30=0.33$

□ $P(B=Y)=15/30=0.5$

□ $P(A=Y, B=Y) = 0.33*0.5 = 0.17$

□ $P(A=N, B=N) = 0.66*0.5 = 0.33$

□ $\rightarrow \Pr(e) = 0.17 + 0.33 = 0.5$

□ $K = (0.83-0.5) / (1-0.5) = 0.66$

		B	
		Y	N
A	Y	10	0
	N	5	15

Poor agreement = Less than 0.20
 Fair agreement = 0.20 to 0.40
 Moderate agreement = 0.40 to 0.60
 Good agreement = 0.60 to 0.80
 Very good agreement = 0.80 to 1.00

關聯法則

定義

support : fraction of transactions that contain an itemset

Mining Association Rules

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Example of Rules:

$\{Milk, Diaper\} \rightarrow \{Beer\}$ ($s=0.4, c=0.67$)
 $\{Milk, Beer\} \rightarrow \{Diaper\}$ ($s=0.4, c=1.0$)
 $\{Diaper, Beer\} \rightarrow \{Milk\}$ ($s=0.4, c=0.67$)
 $\{Beer\} \rightarrow \{Milk, Diaper\}$ ($s=0.4, c=0.67$)
 $\{Diaper\} \rightarrow \{Milk, Beer\}$ ($s=0.4, c=0.5$)
 $\{Milk\} \rightarrow \{Diaper, Beer\}$ ($s=0.4, c=0.5$)

Observations:

- All the above rules are binary partitions of the same itemset: $\{Milk, Diaper, Beer\}$
- Rules originating from the same itemset have **identical support** but **can have different confidence**

Apriori algo

apriori property(anti monotone)

核心概念: 一個 frequent itemset 的所有 subset 必定也是 frequent
 一組 itemset 的 support 不會大於其任一 subset 的 support

若 subset 非 frequent, 則其 superset 必定也非 frequent。

steps

1. 找出freq one itemset
2. 有交集用聯集產生candidate itemset (L_k self join)
3. subset check, 如果subset非freq, 則prun candidate itemset

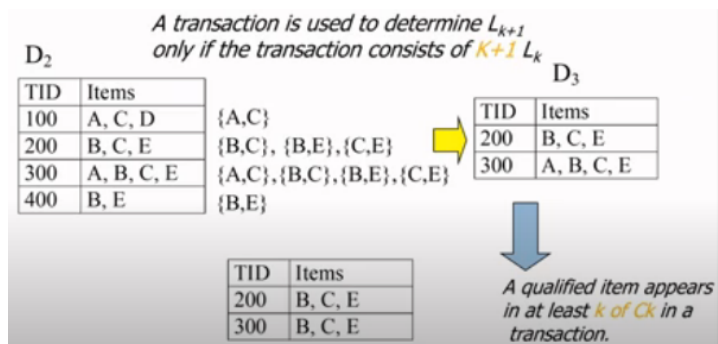
如何加速計算candidates support的方法？

1. Hash Tree
2. FP growth

Rules Generation

對於一個freq itemset m , 找出其subset p , 做出inference $p \rightarrow (m-p)$

reduction on database size



frequent pattern mining bottleneck

1. 多次掃描資料庫 costly
2. 產生太多的candidates list

FP-Growth

1. mining in main memory
2. 不做candidate generation

3. 頻率較多的items有更大的機率share item

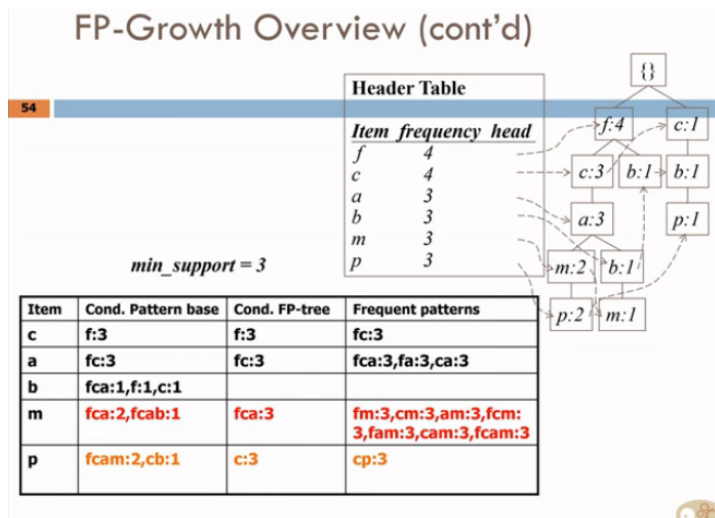
steps

1. 建立fp tree(header also)

- 掃描第一次db，建立freq one itemset
- 依照support大小排序，transactions扣除非freq後也照support排序
- 依序依照transaction插入tree建立fp tree

2. frequent pattern growth

- divide into 條件fp tree，跟一個header指向之freq item相連



For each following evaluation criteria, please briefly describe ONE prediction system in which the criterion is important.

1. NDCG

可應用於文章推薦或搜尋引擎系統，計算其檢索的相關性和排序後，衡量推薦結果的好壞，及推薦系統的預測能力。

2. Recall

對於預測罕見癌症疾病模型，若沒有將真正為癌症的病人檢驗出來會造成嚴重後果，此時recall將會是此系統判斷好壞的評估標準。

you should use recall when looking to predict whether a credit card charge is fraudulent or not. If you have a lot of false negatives, then you have a lot of fraudulent charges that are being labeled as not fraudulent and customers will have money stolen from them.

3. Top-1 precision

依照預測結果機率最大的是正確答案(positive)，precision才會是1，所以模型若要從一個query中找出最佳預測結果，注重第一名的表現，就會用top-1 precision衡量。

可以應用在圖像分類，因為最在乎一張圖片是否能成功預估某一類別(預測最高機率的類別要是ground truth)。

4. F1

f1 score對於非常imbalanced的data有較好的分辨能力，比如要分辨...的分類能力。

5. Novelty

可應用於文章推薦系統，且文章內容除了強調正確性以外，也強調每一篇文章的多樣性，要讓使用者感覺每篇文章都不同時，利用novelty可以反應該任務表現。

6. Precision

Precision is a good evaluation metric to use when the cost of a false positive is very high and the cost of a false negative is low. For example, precision is good to use if you are a restaurant owner looking to buy wine for your restaurant only if it is predicted to be good by a classifier algorithm.

總結 (<https://blog.csdn.net/shanshangyouzhiyangM/article/details/84943011>)