

Data Mining (Mineria de Dades)

Lluís A. Belanche
belanche@lsi.upc.edu

Soft Computing Research Group

Dept. de Llenguatges i Sistemes Informàtics (Software department)

Universitat Politècnica de Catalunya

2011-2012

Data Mining (Part II)

Contents

1. Association Rules
2. Bayesian decision theory
3. Bayesian classifiers: LDA/QDA
4. Bayesian classifiers: Naïve Bayes
5. Bayesian classifiers: Nearest Neighbours
6. Neural Networks: Multilayer Perceptrons
7. Support Vector Machines
8. Ensemble Learning

Data Mining

Association rules

The MBA problem assumes that we have a big set of *items* (like products: milk, bread, ...) with which we can fill *transactions* (market baskets).

The goal pursued is twofold:

1. know which items appear often in the same transactions (which products are often bought together)
2. use this information to increase some kind of benefit (e.g. alter prices, offer pack discounts, redeploy products in corridors, stock previsions)

Data Mining

Association rules

The applications are countless:

- transactions = documents; items = words. “Which words appear often together in documents”. May indicate the existence of complex concepts
- transactions = documents; items = sentences. “Which sentences appear often together in documents”. May indicate the existence of plagiarism
- transactions = web pages; items = html tags. May indicate web mirroring (needs research for the web 2.0, much more interesting to analyze)
- transactions = courses; items = enrolments. “Which courses are taken often together by students”. We do that at the FIB, useful for making enrolment forecasting

Data Mining

Association rules

We are particularly interested in:

1. Frequent itemsets (FIs): those itemsets that appear together quite often (just occasionally we will be interested in products that are seldom sold)

A FI is a set of items that appear together above a given level of frequency

2. Association rules (ARs): $LHS \Rightarrow RHS$, LHS is an itemset, RHS is a singleton. Useful for making predictions on the probability of the item in the RHS (given the itemset in the LHS), which we call the confidence of the rule

Data Mining

Association rules: Causality

The kind of implication in ARs is not the same as causality. Examples:

- $\{milk, butter\} \Rightarrow bread$. This rule may have a high confidence just because many people buy bread
- $\{pasta\ sauce\} \Rightarrow pasta$. This rule could well imply causality. How could we check it?

Raise the price of the pasta and lower the price of the pasta sauce!

If there really is a causality, people who buy pasta will continue buying the sauce (and it is cheaper now!), because the only reason to buy the sauce is having bought the pasta first ...

What if we make a “special pack” (pasta + pasta sauce)? Should we price it above, equal or below the separate price?

Data Mining

Association rules: Formalization

We have a binary data base (transactions $\times$ items):

	items
transactions	01000...11000
	00100...10010
	$\vdots$
	01100...10010

Let R be the total set of items and T the set of transactions. We call *itemset* to any $X \subseteq R$. Let $X = \{X_1, \dots, X_n\}$ an itemset. Define the *support* (or frequency) of X as:

$$s(X) = \frac{|t \in T / X_1 = X_2 = \dots = X_n = 1|}{|T|}$$

Let $\alpha_f \in (0, 1)$ fixed. We say an itemset X is frequent if $s(X) \geq \alpha_f$

Our first goal is thus to find all the frequent itemsets in the data base

Consider again $X = \{X_1, \dots, X_n\}$ an itemset, $Y \in R$ an item, where $Y \notin X$. A rule is of the form $\{X_1, \dots, X_n\} \Rightarrow Y$. Define the *frequency* of the rule as:

$$freq(X \Rightarrow Y) = s(X \cup \{Y\})$$

and the *confidence* of the rule (aka rule cover) as:

$$conf(X \Rightarrow Y) = \frac{s(X \cup \{Y\})}{s(X)}$$

The confidence can be seen as a frequentist estimation of $Prob(Y|X)$, given the database. Notice $freq(X \Rightarrow Y) = freq(Y \Rightarrow X)$ but $conf(X \Rightarrow Y) \neq conf(Y \Rightarrow X)$.

Let $\alpha_f, \alpha_c \in (0, 1)$ fixed. We say a rule $X \Rightarrow Y$ is a basic rule if:

1. $freq(X \Rightarrow Y) \geq \alpha_f$

2. $conf(X \Rightarrow Y) \geq \alpha_c$

Our second (and main) goal is thus an algorithm to find all the basic rules in the data base (for fixed α_f, α_c) in an efficient way.

This algorithm can then be used with different values of α_f, α_c to create sets of interesting rules (novel, non-trivial and useful).

Data Mining

Association rules: Example

id	A	B	C	D	E	F	G	H	I	J	K
t_1	1	1	1	1	0	0	1	0	0	0	0
t_2	1	1	0	0	1	1	0	0	0	0	0
t_3	0	1	0	0	0	0	0	0	1	0	1
t_4	1	1	0	0	0	0	0	1	0	0	0
t_5	0	0	0	0	1	0	1	0	0	1	0

Note the sparseness of the matrix, something to be expected.

Compute the basic rules if $\alpha_f = 0,3$ and $\alpha_c = 0,9$.

Solution: $freq(\{A\} \Rightarrow B) = \frac{3}{5}$, $conf(\{A\} \Rightarrow B) = 1, \dots$. We arrive at the set of frequent itemsets: $\mathcal{F} = \{\{A\}, \{B\}, \{E\}, \{G\}, \{A, B\}\}$. The only possible rules are therefore $\{A\} \Rightarrow B$ and $\{B\} \Rightarrow A$. Since $freq(\{B\} \Rightarrow A) = \frac{3}{5}$, $conf(\{B\} \Rightarrow A) = \frac{3}{4}$, the answer is the sole rule $\{A\} \Rightarrow B$.

Data Mining

Association rules: Method (I)

The previous example suggests this way of proceeding:

1. Find all the frequent itemsets $\mathcal{F}$
2. Generate all the rules of the form $F \setminus \{Z\} \Rightarrow Z$, where $Z \in F \in \mathcal{F}$.
3. Check if these rules are basic

And do it in an efficient way ... what is the computational cost of “generate & test” all the rules by brute force? $O(|R|^2 \cdot 2^{|R|-1} \cdot |T|)$

Data Mining

Association rules: Method (II)

What ??????? How on Earth can we make this efficient?

Observation:

F is frequent $\implies$ all the subsets of F are frequent

(the opposite is not true: the best we can say is that we have no evidence to discard F as frequent)

Proof. Hint: $X \subseteq Y \Rightarrow s(X) \geq s(Y)$. If X is not frequent, then neither is Y . Since this is valid for all $Y \supseteq X$, if an $X \subseteq Y$ is not frequent, none of its supersets will be. So, if Y is frequent, all of its subsets must be.

We have the inspiring idea of starting with all the frequent itemsets of size 1, use them to discard itemsets of size 2, check if the rest are, use these to discard itemsets of size 3, and so forth ...

Data Mining

Association rules: an Algorithm

Algorithm 1: A Priori

input : T set of transactions, R set of items, α_f

output: $\mathcal{F}$ set of all the frequent itemsets

```
1   $i \leftarrow 1; C_i \leftarrow \{\{A\} \mid A \in R\};$ 
2  while  $C_i \neq \emptyset$  do
3       $\mathcal{L}_i \leftarrow \emptyset$ 
4      for  $c \in C_i$  do
5          if  $s(c) \geq \alpha_f$  then  $\mathcal{L}_i \leftarrow \mathcal{L}_i \cup \{c\}$ 
6      end
7       $C_{i+1} \leftarrow \{\text{sets of size } i + 1 \text{ having all of their subsets of size } i \text{ within } \mathcal{L}_i\};$ 
8       $i \leftarrow i + 1$ 
9  end
10  $\mathcal{F} \leftarrow \cup_i \mathcal{L}_i$ 
```

Data Mining

Questions, questions ...

1. How do we form the set C_{i+1} from the set $\mathcal{L}_i$?
2. How efficiently can we compute $s(c) \geq \alpha_f$ for all $c \in C_i$?
3. How do we generate the rules once we have $\mathcal{F}$? Where do we get their support and confidences?

Data Mining

Answers, answers ...

How do we form the set C_{i+1} from the set $\mathcal{L}_i$?

join : find all the set pairs $\{A, B\}$, where $A, B \in \mathcal{L}_i$, such that $|A \cup B| = i+1$
(“potential candidates”)

prune : check whether each potential candidate is really a candidate: all
of its subsets of size i must belong to $\mathcal{L}_i$

Data Mining

Answers, answers ...

How efficiently can we compute $s(c) \geq \alpha_f$ for all $c \in C_i$?

We go transaction after transaction, keeping a counter γ_c (initialized to 0) for every $c \in C_i$.

The counter γ_c is incremented by one when a transaction *covers* c

(a transaction **covers** a set if all the items in the set have value 1 in the transaction)

So, all the **for** loop (lines 4-6) can be done in just one pass through the database

Data Mining

Answers, answers ...

How do we generate the rules once we have $\mathcal{F}$? Where do we get their support and confidences?

First we simplify the set, by removing every $F \in \mathcal{F}$ for which $\exists G \in \mathcal{F}$ such that $G \supset F$. Now letting $F \in \mathcal{F}$ such that $|F| \geq 2$; do for each $Z \in F$:

1. generate the rule $F \setminus \{Z\} \Rightarrow Z$
2. the frequency of the rule is $freq(F \setminus \{Z\} \Rightarrow Z) = s((F \setminus \{Z\}) \cup \{Z\})$.
Since $Z \in F$, this is equal to $s(F)$
3. the confidence of the rule is $conf(F \setminus \{Z\} \Rightarrow Z) = \frac{s(F)}{s(F \setminus \{Z\})}$

The fact is that we already have these quantities, since both F and $F \setminus \{Z\}$ are frequent sets, and thus the algorithm computed their frequencies

Data Mining

Example

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}	t_{13}	t_{14}	t_{15}	t_{16}	t_{17}	t_{18}	t_{19}	t_{20}
A	1	1	0	1	0	1	1	1	1	1	1	1	1	0	1	1	0	0	1	1
B	1	0	1	1	1	0	1	1	0	1	1	1	1	1	1	1	1	1	0	1
C	0	1	1	1	0	1	0	0	1	1	0	0	1	1	0	0	0	1	0	0
D	0	0	1	0	0	1	0	0	1	0	0	1	1	0	1	0	0	0	0	0
E	1	1	0	0	1	0	1	0	0	1	1	1	1	0	0	1	1	1	1	1
F	0	1	1	0	1	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0

Execute Apriori to obtain $\mathcal{F}$ and use this set to obtain all the basic rules
if $\alpha_f = \frac{1}{4}, \alpha_c = \frac{3}{4}$.

Data Mining

Solution to the example

$$s(\{A\}) = \frac{3}{4}, s(\{B\}) = \frac{4}{5}, s(\{C\}) = \frac{9}{20}, s(\{D\}) = \frac{3}{10}, s(\{E\}) = \frac{13}{20}, s(\{F\}) = \frac{3}{10}, C_1 = \{\{A\}, \{B\}, \{C\}, \{D\}, \{E\}, \{F\}\}, \mathcal{L}_1 = \{\{A\}, \{B\}, \{C\}, \{D\}, \{E\}, \{F\}\}$$

$$C_2 = \{\{A, B\}, \{A, C\}, \dots, \{E, F\}\}$$
$$s(\{A, B\}) = \frac{11}{20}, s(\{A, C\}) = \frac{3}{10}, \dots, s(\{E, F\}) = \frac{1}{5}$$

$$\mathcal{L}_2 = \{\{A, B\}, \{A, C\}, \{A, D\}, \{A, E\}, \{B, C\}, \{B, E\}, \{C, F\}\}$$

$$C_3 = \{\{A, B, C\}, \{A, B, E\}\} \quad // \text{ why } \{A, C, F\} \notin C_3? \text{ because } \{A, F\} \notin \mathcal{L}_2$$
$$s(\{A, B, C\}) = \frac{3}{20}, s(\{A, B, E\}) = \frac{8}{20}$$

$$\mathcal{L}_3 = \{\{A, B, E\}\}$$

$$C_4 = \emptyset$$

Data Mining

Solution to the example

$$\mathcal{F} = \{\{A\}, \{B\}, \{C\}, \{D\}, \{E\}, \{F\}, \{A, B\}, \{A, C\}, \{A, D\}, \{A, E\}, \{B, C\}, \{B, E\}, \{C, F\}, \{A, B, E\}\}$$

which (for rule generation) simplifies to $\mathcal{F} = \{\{A, C\}, \{A, D\}, \{B, C\}, \{C, F\}, \{A, B, E\}\}$

If we now generate the rules and compute their confidences:

- $conf(\{E\} \Rightarrow A) = \frac{10}{13}$
- $conf(\{E\} \Rightarrow B) = \frac{11}{13}$
- $conf(\{A, E\} \Rightarrow B) = \frac{8}{10}$
- $conf(\{D\} \Rightarrow A) = \frac{5}{6}$

are the only basic rules. Note that $\{E\} \Rightarrow B$ is a more general rule than $\{A, E\} \Rightarrow B$ (and the latter has lower confidence). Therefore we could eliminate $\{A, E\} \Rightarrow B$.

Data Mining

Measuring how good the rules are

Once we have the (polished) basic rules we are interested in “sorting’em out”, from “best” to “worst”. Unfortunately, confidence is not enough in many situations.

To begin with, confidence does not obey augmentation:

$$\text{conf}(\{A\} \Rightarrow C) \not\leq \text{conf}(\{A, B\} \Rightarrow C)$$

Why confidence does not tell the big picture? See an example:

Consider the rule $R : \{milk, butter\} \Rightarrow bread$ with $\text{conf}(R) = 0,9$ (very high, indeed). May be simply a lot of people buy bread ...

Define $\text{lift}(X \Rightarrow Y) = \frac{\text{conf}(X \Rightarrow Y)}{s(Y)} \in [0, \infty)$ and suppose $s(\{milk, butter\}) = 0,7$, $s(\{bread\}) = 0,9$ and $\text{freq}(\{milk, butter, bread\}) = 0,63$. Then $\text{conf}(R) = \frac{0,63}{0,7} = 0,9$, $\text{lift}(R) = \frac{0,9}{0,9} = 1$.

Data Mining

Measuring how good the rules are

$$\text{lift}(X \Rightarrow Y) \begin{cases} < 1 & \text{the rule is worse than } P(Y) \text{ in predicting } Y \\ = 1 & \text{the rule is equal than } P(Y) \text{ in predicting } Y \\ > 1 & \text{the rule is better than } P(Y) \text{ in predicting } Y \end{cases}$$

In our example, the conclusion is that the buying of milk and butter does not condition the buying of bread, because the proportions are the same

Hint: the lift measures “deviation from independence”, i.e. from $P(X, Y) = P(X)P(Y)$ (in the example, $X = \{milk, butter\}$ and $Y = \{bread\}$).

Data Mining

A worked example

Problem: Adult Data Set. Prediction task is to determine whether a person makes over 50K a year, based on census data (age, workclass, education, marital status, ...)

Number of transactions: 48,842 (the variables need to be binarized)

We find the rule: $\{marital_status = Married_civ_spouse\} \Rightarrow native_country = United_States$ (obviously there is no causality!)

The reason is that the data come from the US Census (though there are some nationals from Mexico, the Philippines and many other countries).

$$s(\{native_country = United_States\}) = 0,8974$$

$$s(\{marital_status = Married_civ_spouse\}) = 0,4582$$

If being married (civil wedding) and having US nationality are independent, we can expect that:

$$s(\{native_country = United_States, marital_status = Married_civ_spouse\}) \approx 0,41$$

Therefore $freq(\{marital_status = Married_civ_spouse\} \Rightarrow native_country = United_States) \approx 0,41$ and $conf(\{marital_status = Married_civ_spouse\} \Rightarrow native_country = United_States) \approx \frac{0,41}{0,4582} \approx 0,895$ (quite high!)

... but $lift(\{marital_status = Married_civ_spouse\} \Rightarrow native_country = United_States) \approx \frac{0,895}{0,8974} \approx 0,997$.

Data Mining

Practical matters

- A particularly important operation is that of quickly checking whether a transaction covers an itemset (e.g, via short-cut low-level boolean operators)
- Another one is fetching the frequency of an itemset (e.g., via a hash table)
- It is convenient to store the elements of $\mathcal{L}_i$ in lexicographical order,
e.g. $\mathcal{L}_3 = \{\{A, B, C\}, \{A, B, D\}, \{A, C, D\}, \{A, C, E\}, \{B, C, D\}\}$
(we store $\{A, B, C\}$ and not $\{B, A, C\}$)
now it is a matter of merging only those pairs of subsets with the same prefix of size $i - 1$ (these sets are consecutive), e.g. $\{A, B, C\} \cup \{A, B, D\} = \{A, B, C, D\}$

Data Mining

Practical matters

Algorithm 2: $\mathcal{L}_i$ to C_{i+1}

input : $i, \mathcal{L}_i$ (ordered lexicographically)

output: C_{i+1}

```
1  $C_{i+1} \leftarrow \emptyset;$ 
2 for  $X, Y \in \mathcal{L}_i$ , with  $X[1, \dots, i-1] = Y[1, \dots, i-1] \wedge X[i] < Y[i]$  do
3    $T \leftarrow X \cup \{Y[i]\}$ 
4   if  $\forall T' \subset T : |T'| = i : T' \in \mathcal{L}_i$  then  $C_{i+1} \leftarrow C_{i+1} \cup \{T\}$ 
5 end
```

Data Mining

More on confidence

1. Confidence does not obey augmentation: $\text{conf}(\{A\} \Rightarrow C) \not\leq \text{conf}(\{A, B\} \Rightarrow C)$
2. Yet, this is a general rule; exercise: can you find an example where it does?
3. Principle of monotony: $\text{conf}(X \setminus Z \Rightarrow Y \cup Z) \leq \text{conf}(X \Rightarrow Y)$,
e.g. $\text{conf}(\{A\} \Rightarrow \{B, C, D\}) \leq \text{conf}(\{A, B\} \Rightarrow \{C, D\}) \leq \text{conf}(\{A, B, C\} \Rightarrow \{D\})$

Proof. A simpler way to express it is:

for all $X, Z, Z' \subseteq R$, where $Z, Z' \subset X$, $\text{conf}(X \setminus Z \Rightarrow Z) \leq \text{conf}(X \setminus Z' \Rightarrow Z')$ if $Z' \subseteq Z$.

Now $\text{conf}(X \setminus Z \Rightarrow Z) = \frac{s(X)}{s(X \setminus Z)}$; just use that $s(X \setminus Z') \leq s(X \setminus Z)$ if $Z' \subseteq Z$.

Data Mining

Frequency lower bounding

Let $X, Y, Z \subseteq R$ itemsets; we have:

$$s(X \cup Y \cup Z) \geq s(X \cup Y) + s(X \cup Z) - s(X)$$

Proof. Let $c(X)$ be the set of transactions that cover an itemset X . We have:

$$s(X \cup Y \cup Z) = |c(X \cup Y) \cap c(Y \cup Z)|$$

$$= |c(X \cup Y) \setminus [c(X \cup Y) \setminus c(X \cup Z)]|$$

$$\geq |c(X \cup Y) \setminus [c(X) \setminus c(X \cup Z)]|$$

$$\geq |c(X \cup Y)| - |c(X) \setminus c(X \cup Z)|$$

$$= |c(X \cup Y)| - (|c(X)| - |c(X \cup Z)|)$$

$$= s(X \cup Y) + s(X \cup Z) - s(X)$$

$$A \cap B = A \setminus (A \setminus B)$$

$$X \subseteq X \cup Y \Rightarrow c(X) \supseteq c(X \cup Y)$$

$$|A \setminus B| \geq |A| - |B|$$

Data Mining

Problems with Association Rules

- Users are overwhelmed by the number of rules identified
- The mining algorithm assumes sparsity, which is not always true
- An 'optimal' value for minimum support can not be known beforehand
- What if we are interested in negative associations?

Data Mining

Closed and Maximal Itemsets

- An itemset X is **closed** if there exists no superset of X with the same support count as X .
- A closed itemset X is frequent (or not) as usual, i.e., if X is frequent
- A frequent itemset Y is **maximal** if it is not a proper subset of any other frequent itemset.
- A **maximal itemset** is closed but a closed itemset is not necessarily maximal.

Frequent maximal itemsets determine all frequent itemsets! Therefore the aim of any association rule algorithm is to find these sets.

Example. Suppose $s(\{B, D\}) = s(\{B, C, D\}) = 8$ and $s(\{C, D\}) = 9$. Then $\{C, D\}$ is closed but not maximal.

Data Mining

Other methods

- There exist other algorithms besides Apriori to find frequent itemsets (also closed and maximal) : Éclat, FP-growth, CHARM, ...
- The interested student can find the excellent work by C. Borgelt here:
<http://www.borgelt.net/fpm.html>
- It is believed that Apriori is generally better than other algorithms if the support required is high
- If the support is very low, none of the algorithms is able to handle large frequent sets smoothly
- In any other case (intermediate supports), Apriori may not be the best option