Integrating Classification and Association Rule Mining

Bing Liu **Wynne Hsu** **Yiming Ma**

School of Computing
National University of Singapore

{liub, whsu, mayiming}@iscs.nus.edu.sg

http://www.comp.nus.edu/~dm-ii

Introduction

Classification: mine a small set of rules to form a classifier.

Target: only one -- the class attribute

$$X = 5,\ 5 < Y \leq 6 \quad \rightarrow \quad Class = Yes$$

Association: find all rules existing in the database

Target: any attribute (or item)

$$Cheese,\ Milk \quad \rightarrow \quad Bread\ \ (sup = 5\%,\ confid = 70\%)$$

OR $Cheese = T,\ Milk = T \rightarrow Bread = T\ (sup = 5\%,\ confid = 70\%)$

- This paper proposes to integrate the two mining techniques.

Introduction (cont …)

Integration: focusing on a subset of association rules, whose right-hand-sides are **<u>restricted</u>** to the classification class attribute.

- Call this subset of rules *class association rules* (CARs).

Proposed algorithm: ***CBA*** (***C****lassification-****B****ased on* ***A****ssociations*).

- This single algorithm can perform three tasks:

  1. mine all classification rules (CARs)
  2. produce an accurate classifier from CARs (in general, more accurate than C4.5)
  3. mine normal association rules

  These multiple capabilities give the user great convenience.

- CBA helps to solve a number of problems in classification systems

Some problems with classification rule mining

Key feature of classification systems: use biases to generate **a small subset of rules** that exist in data to form a classifier.

- In applications, user wants **classifiers**, as well as **interesting rules.**

Problems: Since only **a small subset of rules** is discovered, and biases may not be in agreement with the knowledge of the user,

1. The discovered rules may not be understandable or interesting.
2. Understandable and/or interesting rules may not be discovered.

Another Problem:

3. Standard classification systems need to load the entire database into the main memory.

CBA helps to solve the problems

Since CBA generate all rules, then

- Finding understandable rules and interesting (novel or unexpected) rules becomes a post-processing task.
- Post-analysis techniques (Liu, Hsu AAAI-96; Liu, Hsu and Chen KDD-97) can be used to identify understandable rules and interesting rules.
- The proposed system also **allows the data to reside on disk**.

Problem statement

- Assume a relational table D with n attributes. An attribute can be discrete or continuous. (CBA also works with transactional data)
- There is a discrete class attribute (for classification).
- For a continuous attribute, it is discretized into intervals.
- **Item**: (*attribute*, *value*)
- Let I be the set of all items in D, and Y be the set of class labels.
- A class association rule (CAR) is an implication of the form

  $X \rightarrow y$, where $X \subseteq I$, and $y \in Y$.
- A rule $X \rightarrow y$ holds in D with *confidence* and *support* (as in normal association rule mining).

Generating the complete set of CARs

Our objectives are:

1 **To generate the complete set of CARs** that satisfy the minimum support (called *minsup*) and minimum confidence (called *minconf*) constraints, and

2 **To build a classifier** from the CARs.

We propose algorithm CBA. It consists of two parts,

- A *rule generator* (called CBA-RG), based on algorithm **Apriori** for finding association rules in (Agrawal and Srikant 1994),
- A *classifier builder* (called CBA-CB).

Three (3) Steps of the CBA-RG algorithm

1. **Find all *frequent ruleitems***: by making multiple passes over the data similar to the Apriori algorithm. A *ruleitem* is:

   <*condset*, *y*>, where *condset* is a set of items, $y \in Y$ is a class.

   A ruleitem is *frequent* if its support is above *minsup*.

2. **Generate *Class Association Rules*** (CARs): from the frequent ruleitems. For all *ruleitems* with the same *condset*, the *ruleitem* with the highest confidence is chosen as the *possible rule* (PR). E.g.,

   1. <{(A, 1), (B, 1)}, (class: 1)> [confidence = 2/3]
   2. <{(A, 1), (B, 1)}, (class: 2)> [confidence = 1/3]

   With these two *ruleitems*, one PR is produced (assume $|D| = 10$):

   (A, 1), (B, 1) $\rightarrow$ (class, 1) [*supt* = 20%, *confd*= 66.7%]

Three (3) Steps of the CBA-RG algorithm (cont …)

- If the confidence is greater than *minconf*, we say the rule is *accurate*.

- C*lass association rules* (CARs): consists of all the PRs that are both *frequent* and *accurate*.

3. **Prune the rules** (CARs): using the pessimistic error rate based pruning in C4.5.

   This pruning cuts down the number of rules generated substantially without reducing the prediction accuracy.

Generating normal Association Rules using CBA

Since CBA-RG first finds all *ruleitems*:

$<condset, y>$, where *condset* is a set of items, $y \in Y$ is a class.

- If we take y as a ***dummy***, then we get *itemset* (= *condset*) of the traditional association rule mining.

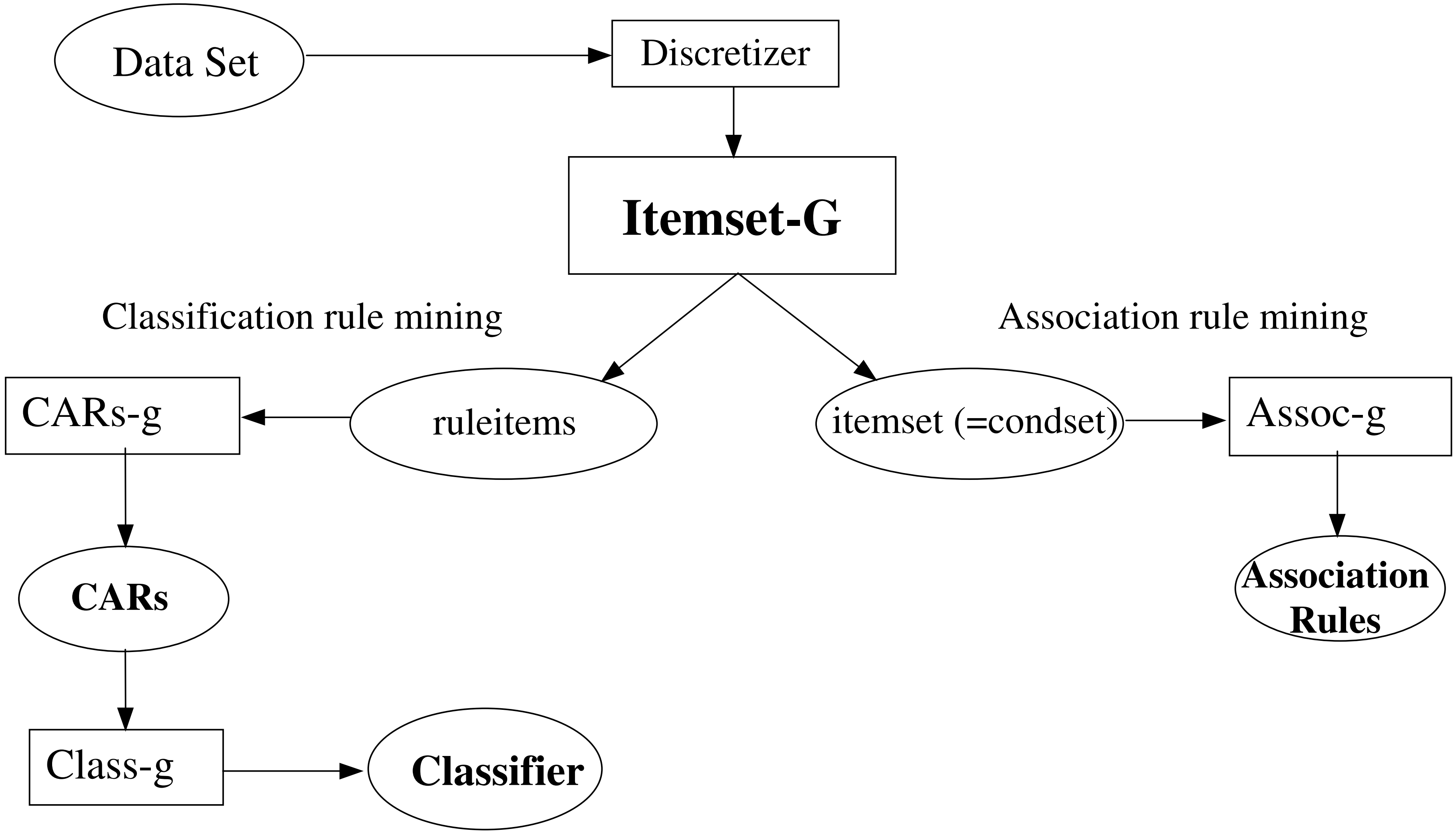
Why not use an association rule miner to mine CARs?

After all, "class association rules (CARs)" $\subseteq$ "all association rules"

However,

- Transactional data used in traditional association rule mining do not have a huge number of associations.
- Classification data sets tend to contain a huge number of associations, which lead to **combinatorial explosion**.
- Using CBA to focus on class association rules (CARs) reduces the number of rules generated (and also time taken).
- CBA can generate normal association rules. A typical association rule miner cannot mine only CARs without substantial modification.

Data mining with CBA



Building a Classifier

- The CBA-CB algorithm builds a classifier using CARs (or prCARs).

Definition: Given two rules, r_i and r_j, $r_i \succ r_j$ (also called r_i precedes r_j or r_i has a higher precedence than r_j) if

1. the confidence of r_i is greater than that of r_j, or
2. their confidences are equal, but the support of r_i is greater than that of r_j, or
3. both the confidences and supports of r_i and r_j are the same, but r_i is generated earlier than r_j;

- Let R be the set of generated rules. The basic idea of the algorithm is to choose a set C of high precedence rules in R to cover D.
- These rules in C form the classifier.

The classifier and the naive algorithm (M1)

Our classifier:

$<r_1, r_2, \ldots, r_n, default\_class>$, where $r_i \in R$, $r_a \succ r_b$ if $b > a$.

Classify a new case: The first rule, that satisfies the case, classifies it. If no rule applies, it takes default class.

A naive algorithm (called M1) for building such a classifier is:

Step 1: Sort the set of generated rules *R* according to the relation "$\succ$".

Step 2: Using the rules in *R* to cover the training data (in sorted sequence). After each rule, the covered cases by the rule are removed.

A set of rules *C* is selected from *R* that covers all training data.

Step 3: Discard those rules in *C* that do not improve the accuracy.

The improved algorithm (M2)

M1 is simple, but very inefficient because it needs to make many passes over the data.

- An improved algorithm (called M2) only makes **slightly more than one pass** over the database.

M2: Instead of making one pass over the remaining data for each rule (in M1), M2 finds the best rule in *R* to cover each case.

M2 is rather involved, refer to the paper.

Empirical evaluation

- Compare the classifiers produced by CBA with those produced by C4.5 (tree and rules) (Release 8).
- 26 datasets from UCI ML Repository are used.
- In our experiments, *minconf* is set to 50%. *minsup* has a strong effect on the quality of the classifier.
- We observe that once *minsup* is lowered to 1-2%, the classifier built is already quite accurate compared to that built by C4.5.
- In the experiments reported, we set *minsup* to 1%.
- We also set a limit of 80,000 on the total number of rules in memory.
- Discretization of continuous attributes uses the Entropy method in (Fayyad and Irani 1993). The code is taken from MLC++ machine learning library (Kohavi et al 1994)

Datasets	c4.5rules w/o discr.	c4.5rules discr.	CBA (CARs + infreq) w/o pru.		CBA (CARs) w/o pru		No. of CARs w/o pru.		Run time (sec) (CBA-RG) w/o pru		Run time (sec) (CBA-CB) M1		No. Rules in C
			w/o pru.	pru.	w/o pru	pru.	w/o pru.	pru.	w/o pru	pru.	M1	M2	
anneal*	5.2	6.5	1.9	1.9	3.2	3.6	65081	611	14.33	14.36	0.08	0.06	34
austra*	15.3	13.5	13.5	13.4	13.2	13.4	46564	4064	5.00	5.05	0.20	0.22	148
auto*	19.9	29.2	21.0	23.1	24.0	27.2	50236	3969	3.30	3.55	0.12	0.06	54
breast-w	5.0	3.9	3.9	3.9	4.2	4.2	2831	399	0.30	0.33	0.02	0.03	49
cleve*	21.8	18.2	18.1	19.1	16.7	16.7	48854	1634	4.00	4.30	0.04	0.06	78
crx*	15.1	15.9	14.3	14.3	14.1	14.1	42877	4717	4.90	5.06	0.43	0.30	142
diabetes	25.8	27.6	24.8	25.5	24.7	25.3	3315	162	0.25	0.28	0.03	0.01	57
german*	27.7	29.5	27.2	26.5	25.2	26.5	69277	4561	5.60	6.00	1.04	0.28	172
glass	31.3	27.5	27.4	27.4	27.4	27.4	4234	291	0.20	0.22	0.02	0.00	27
heart	19.2	18.9	19.6	19.6	18.5	18.5	52309	624	4.70	4.60	0.04	0.03	52
hepatitis*	19.4	22.6	15.1	15.1	15.1	15.1	63134	2275	2.80	2.79	0.09	0.05	23
horse*	17.4	16.3	18.2	17.9	18.7	18.7	62745	7846	3.2	3.33	0.35	0.19	97
hypo*	0.8	1.2	1.6	1.6	1.9	1.7	37631	493	45.60	45.30	1.02	0.40	35
ionos*	10.0	8.0	7.9	7.9	8.2	8.2	55701	10055	3.75	4.00	0.56	0.41	45
iris	4.7	5.3	7.1	7.1	7.1	7.1	72	23	0.00	0.00	0.00	0.00	5
labor	20.7	21.0	17.0	17.0	17.0	17.0	5565	313	0.17	0.20	0.00	0.00	12
led7	26.5	26.5	27.8	27.8	27.8	27.8	464	336	0.40	0.45	0.11	0.10	71
lymph*	26.5	21.0	20.3	18.9	20.3	19.6	40401	2965	2.70	2.70	0.07	0.05	36
pima	24.5	27.5	26.9	27.0	27.4	27.6	2977	125	0.23	0.25	0.04	0.02	45
sick*	1.5	2.1	2.8	2.8	2.7	2.7	71828	627	32.60	33.40	0.62	0.40	46
sonar*	29.8	27.8	24.3	21.7	24.3	21.7	57061	1693	5.34	5.22	0.30	0.12	37
tic-tac-toe	0.6	0.6	0.0	0.0	0.0	0.0	7063	1378	0.62	0.71	0.12	0.08	8
vehicle*	27.4	33.6	31.3	31.2	31.5	31.3	23446	5704	6.33	6.33	1.40	0.40	125
wavef*	21.9	24.6	20.2	20.2	20.4	20.6	9699	3396	13.65	13.55	2.72	1.12	386
wine	7.3	7.9	8.4	8.4	8.4	8.4	38070	1494	2.34	2.65	0.11	0.04	10
zoo*	7.8	7.8	5.4	5.4	5.4	5.4	52198	2049	2.73	2.70	0.61	0.32	7
Average	16.7	17.1	15.6	15.6	15.7	15.8	35140	2377	6.35	6.44	0.39	0.18	69

Two other important results

- Although we cannot find all rules in 16 of the 26 datasets using the 80,000 limit, the classifiers built are already quite accurate.

  In fact, when the limit reaches 60,000 in the 26 datasets, the accuracy of the resulting classifiers starts to stabilize.

- **Data on disk:** we ran CBA with the datasets on disk, and increased the number of training cases of all datasets by up to 32 times (e.g., 160,000 cases).

  Experimental results show that both CBA-RG and CBA-CB (M2) have linear scaleup.

Conclusion

Contributions:

- Propose **CBA** to integrate classification and association mining
  1. mine all classification rules
  2. produce an accurate classifier (compared with C4.5)
  3. mine normal association rules
- Present a new way to construct classifiers
- Help to solve a number of problems in classification systems.
- Build a classifier with data on disk rather than in memory.

Future work:

- build more accurate classifiers using the framework.
- mine class association rules (CARs) without pre-discretization.