Multi-Dimensional Association
Classification by Association

Cse634

DATA MINING

Professor Anita Wasilewska

Computer Science Department

Stony Brook University

Mining Multi-Dimensional Association

- Single-dimensional rules:

buys(X, "milk") $\Rightarrow$ buys(X, "bread")

- Multi-dimensional rules: $\geq$ 2 dimensions or predicates

Inter-dimension assoc. rules (*no repeated predicates*)

age(X,"19-25") $\wedge$ occupation(X,"student") $\Rightarrow$ buys(X, "coke")

Hybrid-dimension assoc. rules (*repeated predicates*)

age(X,"19-25") $\wedge$ buys(X, "popcorn") $\Rightarrow$ buys(X, "coke")

Mining Multi-Dimensional Association

- Categorical Attributes:
- finite number of possible values, no ordering among values

- Quantitative Attributes:
- Numeric, implicit ordering among values
- Discretization, clustering:
- Numeric values are replaced by ranges or names

- In relational database
- finding all frequent k-predicate sets will require *k* or *k*+1 table scans

Example: Relational Data

Goal:

create multidimensional association rules

Student	Grade	Income	**Buys**
CS	High	Low	Milk
CS	High	High	Bread
Math	Low	Low	Bread
CS	Medium	High	Milk
Math	Low	Low	Bread

STEP 1: Data Conversion to Transaction and its count

Converted Data

Student = CS (I1)	Student =math (I2)	Grade = high (I3)	Grade =medium (I4)	Grade =low (I5)	Income =high (I6)	Income =low (I7)	Buys =milk (I8)	Buys =bread (I9)
+	-	+	-	-	-	+	+	-
+	-	+	-	-	+	-	-	+
-	+	-	-	+	-	+	-	+
+	-	-	+	-	+	-	+	-
-	+	-	-	+	-	+	-	+
3	2	2	1	2	2	3	2	3

Step 2: Apriori Algorithm
Generating 1-itemset Frequent Pattern

Scan D for support count of each ~~candidate~~

Item Set	Support Count
{I1}	3
{I2}	2
{I3}	2
{I4}	1
{I5}	2
{I6}	2
{I7}	3
{I8}	2
{I9}	3

C1

choose (no need for **pruning** here)candidates with support count >=minimum support count

Item Set	Support Count
{I1}	3
{I2}	2
{I3}	2
{I5}	2
{I6,}	2
{I7}	3
{I8}	2
{I9}	3

L1

Let, the **minimum support count be 2**

Since we have 5 records => **minimum Support** = 2/5 = **40%**

Let, **minimum confidence** required is **70%**

Generating 2-itemset Frequent Pattern

Generate C2 **candidates** from L1

Item Set
{I1,I2}
{I1,I3}
{I1,I4}
{I1,I5}
{I1,I6}
{I1,I7}
{I1,I8}
{I1,I9}
{I2,I3}
{I2,I4}
{I2,I5}
{I2,I6}
{I2,I7}
{I2,I8}
{I2,I9}
{I3,I4}
{I3,I5}
{I3,I6}
{I3,I7}
{I3,I8}
{I3,I9}
{I4,I5}
{I4,I6}
{I4,I7}
{I4,I8}
{I4,I9}
{I5,I6}
{I5,I7}
{I5,I8}
{I5,I9}
{I6,I7}
{I6,I8}
{I6,I9}
{I7,I8}
{I7,I9}
{I8,I9}

C2

No need of **pruning** here-Scan D for count of each **candidate**

Item Set	Support Count
{I1,I2}	0
{I1,I3}	2
{I1,I4}	1
{I1,I5}	0
{I1,I6}	2
{I1,I7}	1
{I1,I8}	2
{I1,I9}	1
{I2,I3}	0
{I2,I4}	0
{I2,I5}	2
{I2,I6}	0
{I2,I7}	2
{I2,I8}	0
{I2,I9}	2
{I3,I4}	0
{I3,I5}	0
{I3,I6}	1
{I3,I7}	1
{I3,I8}	1
{I3,I9}	1
{I4,I5}	0
{I4,I6}	1
{I4,I7}	0
{I4,I8}	1
{I4,I9}	0
{I5,I6}	0
{I5,I7}	2
{I5,I8}	0
{I5,I9}	2
{I6,I7}	0
{I6,I8}	1
{I6,I9}	0
{I7,I8}	1
{I7,I9}	2
{I8,I9}	0

C2

choose **candidates** with support count >= **minimum support count**

Item Set	Support Count
{I1,I3}	2
{I1,I6}	2
{I1,I8}	2
{I2,I5}	2
{I2,I7}	2
{I2,I9}	2
{I5,I7}	2
{I5,I9}	2
{I7,I9}	2

L2

Generating Candidates: $\mathbf{C_k}$

- Join Step: $\mathbf{C_k}$ is generated by **joining** $\mathbf{L_{k-1}}$ with itself

- Prune Step: Any (k-1)-item set that is not frequent **cannot** be a subset of a frequent k-item set

Example: Joining and Pruning

1. The join step: To find **C_k**, a set of candidate k-itemsets is generated by **joining** **L_{k-1}** with **itself**.

L_k – Itemsets **C_k – Candidates**

For example in our case:

Considering {I2,I5} , {I7,I9} from **L2** to arrive at **C3** we **Join** **L2*L2**

and we obtain for example {I2,I5,I7}, {I2,I5,I9} as resultant **candidates** in **C3** generated from **L2**

Considering {I1,I3} , {I1,I6} from **L2** we generate a **candidate** {I1,I3,I6} in **C3**

Example: Joining and Pruning

2. The prune step:

C_k is a superset of L_k, that is, its members may or may not be frequent

Ck however, **can be huge** and we **prune it** applying Apriori Principle
"if A is a frequent item set, then each of its subsets is a frequent item set"
It is expressed by formulation of the

Prune Step: Any (k-1)-item set that is **not frequent cannot** be a subset of a frequent k-item set

Thus, {I2,I5,I7}, {I2,I5,I9} **from join step** are considered since all their subsets are frequent

but {I1,I3,I6} is **discarded** since it subset {I3,I6} is **not frequent**, i.e. was not in **L2**

Generating 3-itemset Frequent Pattern

Generate and **prune** C3 **candidates** from **L2**

Item Set
{I2,I5,I7}
{I2,I5,I9}
{I2,I7,I9}
{I5,I7,I9}

C3

Scan D for **count** of candidates

Item Set	Support Count
{I2,I5,I7}	2
{I2,I5,I9}	2
{I2,I7,I9}	2
{I5,I7,I9}	2

C3

choose candidates with support count >= **min support count**

Item Set	Support Count
{I2,I5,I7}	2
{I2,I5,I9}	2
{I2,I7,I9}	2
{I5,I7,I9}	2

L3

Generating 4-itemset Frequent Pattern

Generate and **prune** C4 **candidates** from **L3**

Item Set
{I2,I5,I7,I9}

C4

Scan D for count of each candidate

Item Set	Support Count
{I2,I5,I7,I9}	2

C4

choose candidates with support count >= **min support count**

Item Set	Support Count
{I2,I5,I7,I9}	2

L4

Generating Multidimentional Association Rules

Let **minimum confidence** required be **70%**

- **For example**, let's consider 4-item frequent set
- **I={I2,I5,I7,I9}**
- Its **nonempty subsets** needed to create **rules**
- (we write {2} instead of {I2} .. etc) are:
- **{2}, {5}, {7}, {9},**
- **{2,5}, {2,7}, {2,9}, {5,7}, {5,9}, {7,9},**
- **{2,5,7}, {2,5,9}, {2,7,9}, {5,7,9}**

We create **for example** some **association rules** as follows

R1 : $2 \wedge 5 \wedge 7 \rightarrow 9$ **R2 :** $2 \wedge 5 \wedge 9 \rightarrow 7$ **R3 :** $5 \wedge 7 \rightarrow 2 \wedge 9$

Multidimentional Association Rules

- R1 : **2 ^5 ^ 7** → 9

$student(x, math) \wedge grade(X, low) \wedge income(x, low)$

$\Rightarrow buys(X, bread)$

- R2 : **2 ^5 ^ 9** → **7**

$student(x, math) \wedge grade(X, low) \wedge buys(X, bread)$

$\Rightarrow income(x, low)$

- R3 : **5 ^ 7** → **2^9**

$grade(X, low) \wedge income(x, low) \Rightarrow student(x, math) \wedge buys(X, bread)$

Example: Classification Data

Student	Grade	Income	Buys
CS	High	Low	Milk
CS	High	High	Bread
Math	Low	Low	Bread
CS	Medium	High	Milk
Math	Low	Low	Bread

Converted Data

Student = CS (I1)	Student =math (I2)	Grade = high (I3)	Grade =medium (I4)	Grade =low (I5)	Income =high (I6)	Income =low (I7)	Buys =milk (I8)	Buys =bread (I9)
+	-	+	-	-	-	+	+	-
+	-	+	-	-	+	-	-	+
-	+	-	-	+	-	+	-	+
+	-	-	+	-	+	-	+	-
-	+	-	-	+	-	+	-	+
3	2	2	1	2	2	3	2	3

Generating **Classification Rules** by **Association**

When mining **association rules** for use in **classification** we are **only interested** in **association rules** of the form

$$i1 \ \& \ i2 \ \& \ . . . \ \& \ ik \rightarrow ic$$

where ic is an item associated with a **class label c**

- The process of finding such rules is called
- **Classification by Association**

Classification by Association

- When generating **classification by association rules**
- we are **only interested** in **association rules** of the form

- **(p1 ^ p2 ^ . . . ^ pl) → class = C**

- where the rule antecedent is a conjunction of items
- p1, p2, : : : , pl **associated** with a **class label C**

- In our **example** **class is** either **I8** or **I9**
- as we want to **predict** whether a **student with given characteristics** **buys Milk** or **buys Bread**

Generating **Classification Rules** by Association

Let **minimum confidence** required be **70%**

We run Appriori Algorithm as before and

- **For example**, let's consider 4-item frequent set
- **I={I2,I5,I7,I9}** where **I9** represents **buys-Bread**
- Its **nonempty subsets** needed to create **association rules**
- (we write {2} instead of {I2} .. etc) are:
- **{2}, {5}, {7}, {9},**
- **{2,5}, {2,7}, {2,9}, {5,7}, {5,9}, {7,9},**
- **{2,5,7}, {2,5,9}, {2,7,9}, {5,7,9}**
- To create **classification rules** we consider **only** subsets that contain the **class item 9**

Generating **Classification Rules** by Association

Consider 3- itemset Frequent Sets {2,5,9}, {2,7,9}, {5,7,9}

We create **classification** by association rules as follows

R1 : 5 ^ 7 → 9 [40%,100%]

- **Confidence** = sc{I5,I7,I9}/ sc{I5,I7} = 2/2 = 100%
- **R2** is **selected**
- **R3 : 2 ^ 7 → 9** [40%,100%]
- **Confidence** = sc{I2,I7,I9}/ sc{I2,I7} = 2/2 = 100%
- **R3** is **selected**
- **R4 : 2 ^ 5 → 9** [40%,100%]
- **Confidence** = sc{I2,I7,I9}/ sc{I2,I7} = 2/2 = 100%
- **R4** is **selected**

Generating Classification by Association Rules

Consider 2- itemset Frequent Sets {2,9}, {5,7}, {5,9}, {7,9}, and **{1,8}** from **L2**

We create **classification by association rules** as follows

R5 : **5 → 9** [40%,100%]

- **Confidence** = sc{I5,I9}/ sc{I9} = 2/2 = 100%
- **R5** is **Selected**

R6 : 2 → 9 [40%,100%]

- **Confidence** = sc{I2,I9}/ sc{I9} = 2/2 = 100%
- **R6** is **Selected**

R7 : 7 → 9 [40%,100%]

- **Confidence** = sc{I7,I9}/ sc{I9} = 2/2 = 100%
- **R7** is **Selected**

R8 : 1 → 8 [40%, 66%]

- **Confidence** = sc{I1,I8}/ sc{I1} = 2/3 = 66.66%
- **R8** is **Rejected**

List of Selected Classification by Association Rules

- **2 ^ 5 ^ 7 → 9** [40%,100%]
- **2 ^ 5 → 9** [40%,100%]
- **2 ^ 7 → 9** [40%,100%]
- **5 ^ 7 → 9** [40%,100%]
- **5 → 9** [40%,100%]
- **7 → 9** [40%,100%]
- **2 → 9** [40%,100%]

- We reduce the **confidence** to **66%** to include **I8**
- **1 → 8** [40%,66%]

Test Data

Student	Grade	Income	Buys
Math	Low	Low	Bread
CS	Low	Low	Milk
Math	Low	Low	Milk
Math	Low	Low	Bread
CS	Medium	High	Bread

- **First Tuple**

is **correctly classified by the rule**

I2 & I5 & I7 → I9

Student=math & grade=low & income=low → buys=bread **[Success]**

- **Second Tuple:**

There is no rule for class I8: buys=bredI8 [Error]

- **Third Tuple:**

There is no rule for class I8: buys=bredI8 [Error]

Test Data

Student	Grade	Income	Buys
Math	Low	Low	Bread
CS	Low	Low	Milk
Math	Low	Low	Milk
Math	High	Low	Bread
CS	Medium	High	Bread

- **FourthTuple**

is correctly classify by the rule I2 ^ I7 → I9 [Success]
•Student=Math & Income=Low → Buys=Bread

- **Fifth Tuple**

is correctly classify by the rule I1 → I9 [Success]
Student=CS → Buys=Bread

Hence we have **80% predictive accuracy**
And **20% Error rate**