Discussion Section 2 (CS145)

2015-10-09

Week 02

Outline

- Frequent Itemset mining with constraints
  - Pattern Space constraints
  - Data Space constraints
- Sequential pattern mining
  - Prefix-Scan
- Clustering algorithms
  - K-means
  - PAM
- Homework questions

Constraint-Based Frequent Pattern Mining

- Pattern Space
  - Anti-monotonic
    - If an itemset S violates the constraint, so does any of its superset
    - *Ex: sum(S.price) < v*
  - Monotonic
    - If an itemset S satisfies the constraint, so does any of its superset
    - *Ex: min(S.price) < v*
  - Succinct
    - Without looking at the transaction database, whether an itemset S satisfies the constraint can be determined based on the selection of items
    - *Ex: min(S.price) < v*
  - Convertible
    - A constraint *c* is neither monotonic nor antimonotonic, but can be converted into one
    - *Ex: avg(S.price) < v*

Constraint-Based Frequent Pattern Mining

Data Space: anti-monotone

- A constraint c is *data anti-monotone* if, for a pattern p, it cannot be satisfied by a transaction t in p-projected database, it cannot be satisfied by t's projection on p's superset either
- The key for data anti-monotone is *recursive data reduction*
- Ex. 1. $sum(S.Price) \geq v$ is data anti-monotone
- Ex. 2. $min(S.Price) \leq v$ is data anti-monotone
- Ex. 3. C: $range(S.profit) \geq 25$ is data anti-monotone
  - Itemset {b}'s projected DB:
    - T10': {c, d, f, h}, T20': {c, d, f, g, h}, T30': {c, d, f, g}
  - C cannot be satisfied by T10', T10' can be pruned

TDB (min_sup=2)

TID	Transaction
10	a, b, c, d, f, h
20	b, c, d, f, g, h
30	b, c, d, f, g
40	c, e, f, g

Item	Profit
a	40
b	0
c	-20
d	-15
e	-30
f	-10
g	20
h	-5

Prefix Scan

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

Min_sup = 2

Step1: Find length-1 sequential patterns;

<a>:4, <b>:4, <c>:4, <d>:3, <e>:3, <f>:3

support

pattern

Step2: Divide search space;

six subsets according to the six prefixes;

Step3: Find subsets of sequential patterns;

By constructing corresponding projected databases and mine each recursively.

Prefix Scan

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

Prefix	Projected(suffix) databases	Sequential Patterns
<a>	<(abc)(ac)d(cf)>, <(_d)c(bc)(ae)>, <(_b)(df)cb>, <(_f)cbc>	<a>,<aa>,<ab><a(bc)>,<a(bc)a>, <aba>,<abc>,<(ab)>,<(ab)c>,<(ab)d>,<(ab)f>,<(ab)dc>,<ac>,<aca> ,<acb>,<acc>,<ad>,<adc>,<af>

Prefix Scan

Find sequential patterns having prefix <a>:

1. Scan sequence database S once. Sequences in S containing <a> are projected w.r.t <a> to form the <a>-projected database.

2. Scan <a>-projected database once, get six length-2 sequential patterns having prefix <a> :

   <a>:2 , <b>:4, <(_b)>:2, <c>:4, <d>:2, <f>:2

   <aa>:2 , <ab>:4, <(ab)>:2, <ac>:4, <ad>:2, <af>:2

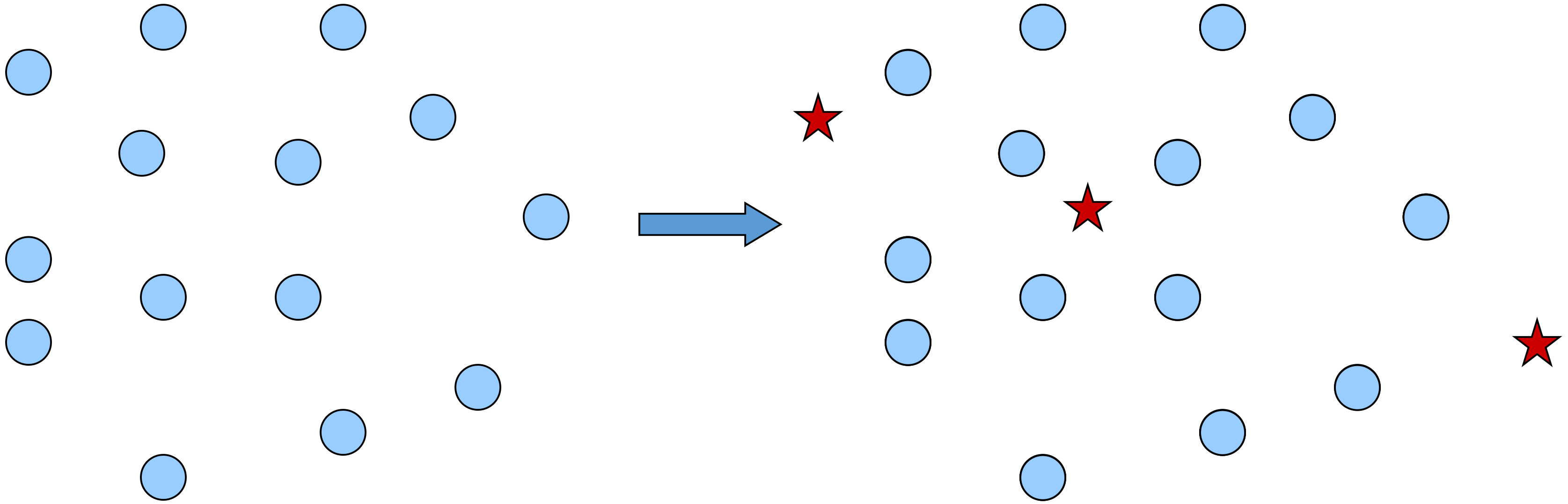
3. Recursively, all sequential patterns having prefix <a> can be further partitioned into 6 subsets. Construct respective projected databases and mine each.

   e.g. <aa>-projected database has two sequences :

   <(_bc)(ac)d(cf)> and <(_e)>.

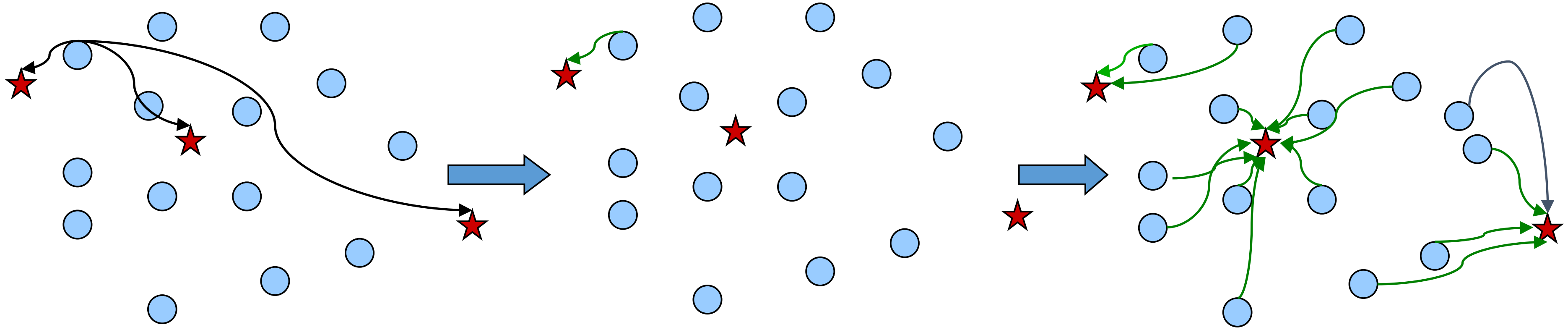
Cluster Analysis

- K-Means Clustering
  - Step 1: Randomly select K centers
  - Step 2: Assign elements to these centers
  - Step 3: Recalculate the centers for each group
  - Step 4: Reassign the elements by repeating step 2-3 until stable.



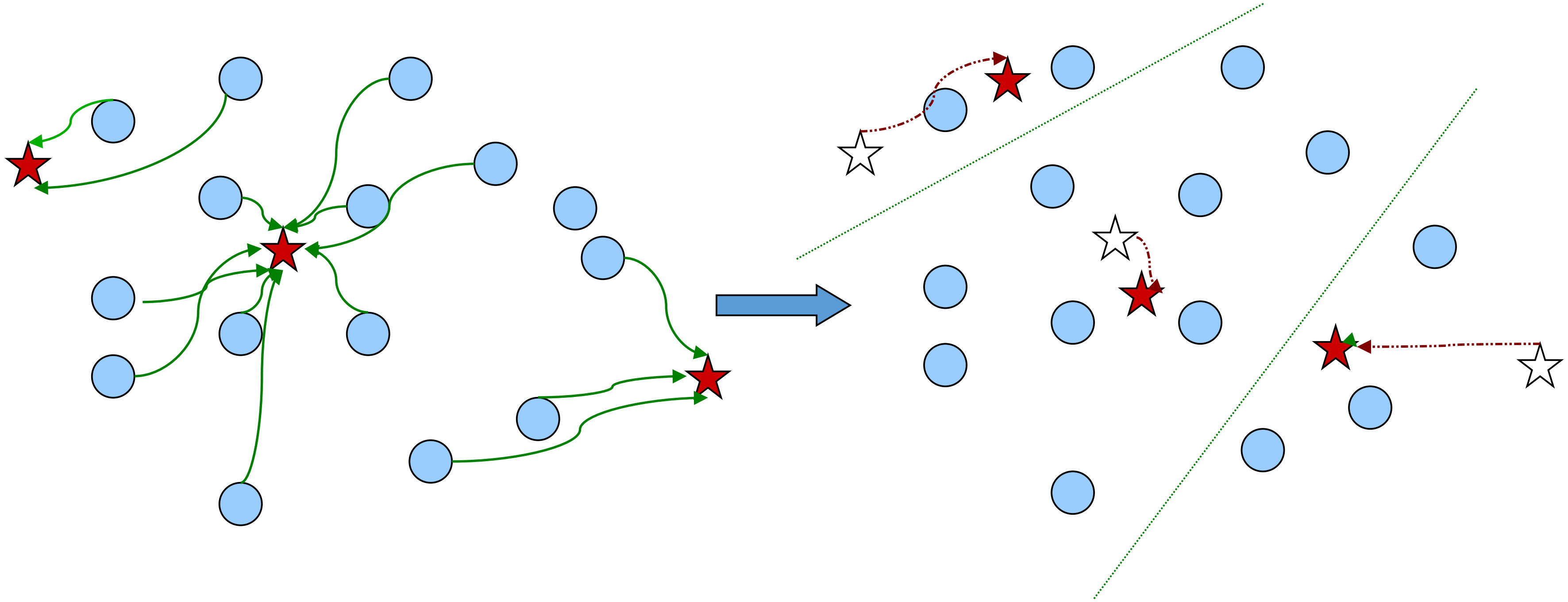
Cluster Analysis

- K-Means Clustering
  - Step 1: Randomly select K centers
  - Step 2: Assign elements to these centers
  - Step 3: Recalculate the centers for each group
  - Step 4: Reassign the elements by repeating step 2-3 until stable.



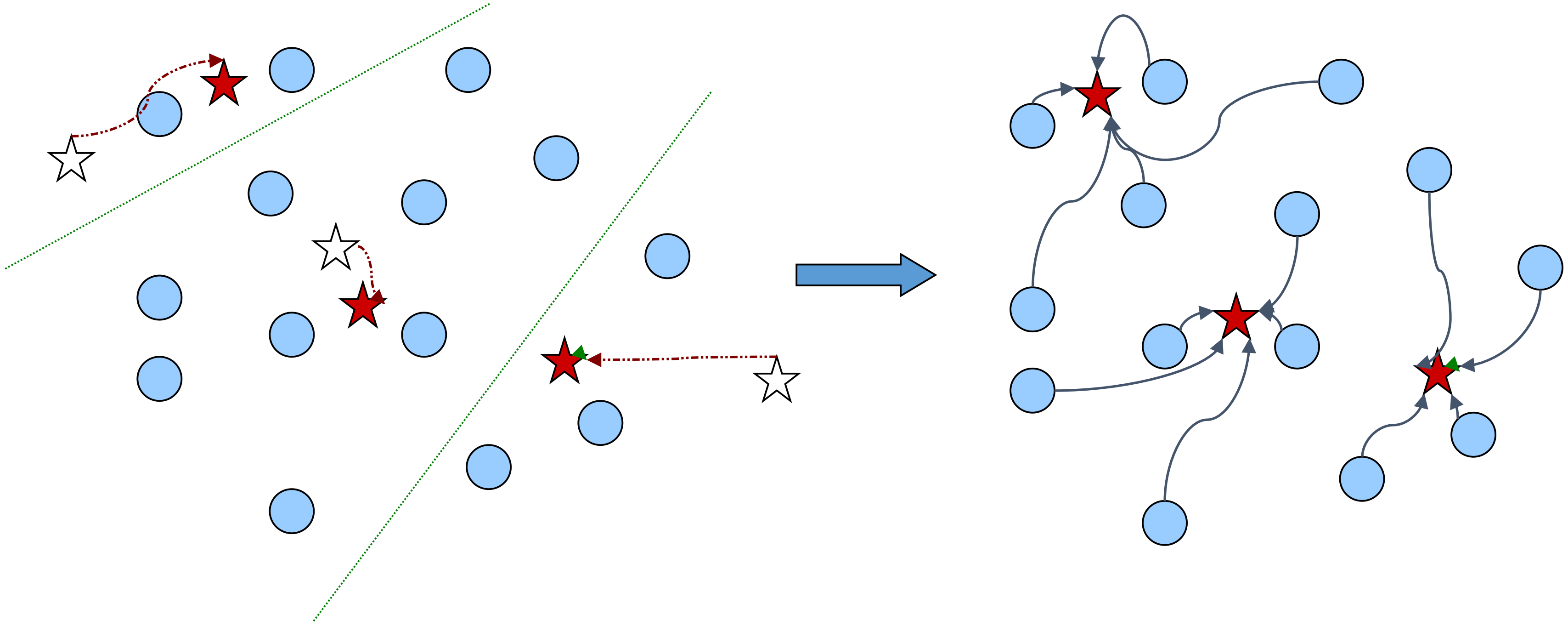
Cluster Analysis

- K-Means Clustering
  - Step 1: Randomly select K centers
  - Step 2: Assign elements to these centers
  - Step 3: Recalculate the centers for each group
  - Step 4: Reassign the elements by repeating step 2-3 until stable.



Cluster Analysis

- K-Means Clustering
  - Step 1: Randomly select K centers
  - Step 2: Assign elements to these centers
  - Step 3: Recalculate the centers for each group
  - Step 4: Reassign the elements by repeating step 2-3 until stable.



Differences between K-means and K-medoid?

- K-means: it uses the mean (“virtual object”) as the center.
- K-medoid: it uses the median (“real object”) as the center.

PAM

- Framework:
- (1) Arbitrarily choose K objects as the initial centers.
- (2) Until no change, do
  - Reassign each object to the nearest cluster.
  - Randomly select a non-medoid object o’, compute the total cost, S.
  - If S<0 then swap o with o’

PAM

- Randomly select a non-medoid object o', compute the total cost, S.

S=Eo'-Eo

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} d(p, o_i)^2$$

- (1) Candidate space:
  - all the other objects excluding those medoids.
- (2) Do we have to assign the objects again?
  - Yes

Homework questions