Week 5 Video 4

Relationship Mining

Sequential Pattern Mining

Association Rule Mining

- Try to automatically find if-then rules within the data set

Sequential Pattern Mining

- Try to automatically find *temporal* patterns within the data set

ARM Example

- If person X buys diapers,
- Person X buys beer

- Purchases occur *at the same time*

SPM Example

- If person X takes Intro Stats now,
- Person X takes Advanced Data Mining in a later semester

- Conclusion: recommend Advanced Data Mining to students who have previously taken Intro Stats

- Doesn’t matter if they take other courses in between

SPM Example

- Learners in virtual environments have different sequences of behavior depending on their degree of self-regulated learning

- High self-regulated learning: Tend to gather information and then immediately record it carefully

- Low self-regulated learning: Tend to gather more information without pausing to record it

(Sabourin, Mott, & Lester, 2011)

Different Constraints than ARM

- If-then elements do not need to occur in the same data point
- Instead
  - If-then elements should involve the same student (or other organizing variable, like teacher or school)
  - If elements can be within a certain time window of each other
  - Then element time should be within a certain window after if times

Sequential Pattern Mining

- Find all subsequences in data with high support

- Support calculated as number of sequences that contain subsequence, divided by total number of sequences

GSP (Generalized Sequential Pattern)

- □ Classic Algorithm for SPM
- □ (Srikant & Agrawal, 1996)

Data pre-processing

- Data transformed from individual actions to sequences by user

- Bob: {GAMING and BORED, OFF-TASK and BORED, ON-TASK and BORED, GAMING and BORED, GAMING and FRUSTRATED, ON-TASK and BORED}

Data pre-processing

- In some cases, time also included

- Bob: {GAMING and BORED 5:05:20, OFF-TASK and BORED 5:05:40, ON-TASK and BORED 5:06:00, GAMING and BORED 5:06:20, GAMING and FRUSTRATED 5:06:40, ON-TASK and BORED 5:07:00}

Algorithm

- Take the whole set of sequences of length 1
  - May include "ANDed" combinations at same time
- Find which sequences of length 1 have support over pre-chosen threshold
- Compose potential sequences out of pairs of sequences of length 1 with acceptable support
- Find which sequences of length 2 have support over pre-chosen threshold
- Compose potential sequences out of triplets of sequences of length 1 and 2 with acceptable support
- Continue until no new sequences found

Let’s execute GPS algorithm

- With min support = 20%

- Chuck: a, abc, ac, de, cef
- Darlene: af, ab, acd, dabc, ef
- Egoberto: aef, ab, aceh, d, ae
- Francine: a, bc, acf, d, abeg

Let's execute GPS algorithm

- With min support = 20%

- Chuck: a, abc, ac, de, cef
- Darlene: af, ab, acd, dabc, ef
- Egoberto: aef, ab, aceh, d, ae
- Francine: a, bc, acf, d, abeg

a, b, c, d, e, f

Let's execute GPS algorithm

- With min support = 20%

- Chuck: **a**, ab**c**, ac, de, cef
- Darlene: af, ab, acd, dabc, ef
- Egoberto: aef, ab, aceh, d, ae
- Francine: a, bc, acf, d, abeg

a, b, c, d, e, f, ***ac***

Let's execute GPS algorithm

- With min support = 20%

- Chuck: **a**, abc, a**c**, de, cef
- Darlene: af, ab, acd, dabc, ef
- Egoberto: aef, ab, aceh, d, ae
- Francine: a, bc, acf, d, abeg

a, b, c, d, e, f, ***ac***

Let's execute GPS algorithm

- With min support = 20%

- Chuck: **a**, abc, ac, de, **c**ef
- Darlene: af, ab, acd, dabc, ef
- Egoberto: aef, ab, aceh, d, ae
- Francine: a, bc, acf, d, abeg

a, b, c, d, e, f, ***ac***

Let’s execute GPS algorithm

- With min support = 20%

- Chuck: a, **a**bc, a**c**, de, cef
- Darlene: af, ab, acd, dabc, ef
- Egoberto: aef, ab, aceh, d, ae
- Francine: a, bc, acf, d, abeg

a, b, c, d, e, f, ***ac***

Let’s execute GPS algorithm

- With min support = 20%

- Chuck: a, **a**bc, ac, de, **c**ef
- Darlene: af, ab, acd, dabc, ef
- Egoberto: aef, ab, aceh, d, ae
- Francine: a, bc, acf, d, abeg

a, b, c, d, e, f, ***ac***

Let's execute GPS algorithm

- With min support = 20%

- Chuck: a, abc, **a**c, de, **c**ef
- Darlene: af, ab, acd, dabc, ef
- Egoberto: aef, ab, aceh, d, ae
- Francine: a, bc, acf, d, abeg

a, b, c, d, e, f, ***ac***

Let’s execute GPS algorithm

- With min support = 20%

- Chuck: a, abc, ac, de, cef
- Darlene: **a**f, ab, a**c**d, dabc, ef
- Egoberto: aef, ab, aceh, d, ae
- Francine: a, bc, acf, d, abeg

a, b, c, d, e, f, ***ac***

Let's execute GPS algorithm

- With min support = 20%

- Chuck: a, abc, ac, de, cef
- Darlene: **a**f, ab, acd, dab**c**, ef
- Egoberto: aef, ab, aceh, d, ae
- Francine: a, bc, acf, d, abeg

a, b, c, d, e, f, ***ac***

Let's execute GPS algorithm

- With min support = 20%

- Chuck: a, abc, ac, de, cef
- Darlene: af, **a**b, a**c**d, dabc, ef
- Egoberto: aef, ab, aceh, d, ae
- Francine: a, bc, acf, d, abeg

a, b, c, d, e, f, ***ac***

Let’s execute GPS algorithm

- With min support = 20%

- Chuck: a, abc, ac, de, cef
- Darlene: af, **a**b, acd, dab**c**, ef
- Egoberto: aef, ab, aceh, d, ae
- Francine: a, bc, acf, d, abeg

a, b, c, d, e, f, ***ac***

Let's execute GPS algorithm

- With min support = 20%

- Chuck: a, abc, ac, de, cef
- Darlene: af, ab, **a**cd, dab**c**, ef
- Egoberto: aef, ab, aceh, d, ae
- Francine: a, bc, acf, d, abeg

a, b, c, d, e, f, ***ac***

Let's execute GPS algorithm

- With min support = 20%

- Chuck: a, abc, ac, de, cef
- Darlene: af, ab, acd, dabc, ef
- Egoberto: **a**ef, ab, a**c**eh, d, ae
- Francine: a, bc, acf, d, abeg

a, b, c, d, e, f, ***ac***

Let's execute GPS algorithm

- With min support = 20%

- Chuck: a, abc, ac, de, cef
- Darlene: af, ab, acd, dabc, ef
- Egoberto: aef, **a**b, a**c**eh, d, ae
- Francine: a, bc, acf, d, abeg

a, b, c, d, e, f, ***ac***

Let's execute GPS algorithm

- With min support = 20%

- Chuck: a, abc, ac, de, cef
- Darlene: af, ab, acd, dabc, ef
- Egoberto: aef, ab, aceh, d, ae
- Francine: **a**, b**c**, acf, d, abeg

a, b, c, d, e, f, ***ac***

Let’s execute GPS algorithm

- With min support = 20%

- Chuck: a, abc, ac, de, cef
- Darlene: af, ab, acd, dabc, ef
- Egoberto: aef, ab, aceh, d, ae
- Francine: **a**, bc, a**c**f, d, abeg

a, b, c, d, e, f, ***ac***

Let’s execute GPS algorithm

- With min support = 20%

- Chuck: a, abc, ac, de, cef
- Darlene: af, ab, acd, dabc, ef
- Egoberto: aef, ab, aceh, d, ae
- Francine: a, bc, acf, d, abeg

a, b, c, d, e, f, **ac**(14/40=35%)

Let’s execute GPS algorithm

- With min support = 20%

- Chuck: a, abc, ac, de, cef
- Darlene: af, ab, acd, dabc, ef
- Egoberto: aef, ab, aceh, d, ae
- Francine: a, bc, acf, d, abeg

a, b, c, d, e, f, ac, ad, ae

Let’s execute GPS algorithm

- With min support = 20%

- Chuck: **a**, **a**bc, ac, **d**e, cef
- Darlene: af, ab, acd, dabc, ef
- Egoberto: aef, ab, aceh, d, ae
- Francine: a, bc, acf, d, abeg

a, b, c, d, e, f, ac, ad, ae, ***aad***,

Let’s execute GPS algorithm

- With min support = 20%

- Chuck: **a**, abc, **a**c, **d**e, cef
- Darlene: af, ab, acd, dabc, ef
- Egoberto: aef, ab, aceh, d, ae
- Francine: a, bc, acf, d, abeg

a, b, c, d, e, f, ac, ad, ae, ***aad***

Let’s execute GPS algorithm

- With min support = 20%

- Chuck: a, abc, ac, de, cef
- Darlene: **a**f, **a**b, ac**d**, dabc, ef
- Egoberto: aef, ab, aceh, d, ae
- Francine: a, bc, acf, d, abeg

a, b, c, d, e, f, ac, ad, ae, ***aad***

Let’s execute GPS algorithm

- With min support = 20%

- Chuck: a, abc, ac, de, cef
- Darlene: **a**f, **a**b, acd, **d**abc, ef
- Egoberto: aef, ab, aceh, d, ae
- Francine: a, bc, acf, d, abeg

a, b, c, d, e, f, ac, ad, ae, ***aad***

Let's execute GPS algorithm

- With min support = 20%

- Chuck: a, abc, ac, de, cef
- Darlene: **a**f, ab, **a**cd, **d**abc, ef
- Egoberto: aef, ab, aceh, d, ae
- Francine: a, bc, acf, d, abeg

a, b, c, d, e, f, ac, ad, ae, ***aad***

Let’s execute GPS algorithm

- □ With min support = 20%

- □ Chuck: a, abc, ac, de, cef
- □ Darlene: af, ab, acd, dabc, ef
- □ Egoberto: **a**ef, **a**b, aceh, **d**, ae
- □ Francine: a, bc, acf, d, abeg

a, b, c, d, e, f, ac, ad, ae, ***aad***

Let’s execute GPS algorithm

- With min support = 20%

- Chuck: a, abc, ac, de, cef
- Darlene: af, ab, acd, dabc, ef
- Egoberto: **a**ef, ab, **a**ceh, **d**, ae
- Francine: a, bc, acf, d, abeg

a, b, c, d, e, f, ac, ad, ae, ***aad***

Let’s execute GPS algorithm

- With min support = 20%

- Chuck: a, abc, ac, de, cef
- Darlene: af, ab, acd, dabc, ef
- Egoberto: aef, ab, aceh, d, ae
- Francine: **a**, bc, **a**cf, **d**, abeg

a, b, c, d, e, f, ac, ad, ae, ***aad***

Let’s execute GPS algorithm

- □ With min support = 20%

- □ Chuck: a, abc, ac, de, cef
- □ Darlene: af, ab, acd, dabc, ef
- □ Egoberto: aef, ab, aceh, d, ae
- □ Francine: a, bc, acf, d, abeg

a, b, c, d, e, f, ac, ad, ae, aad, aae, ade

Let's execute GPS algorithm

- From
- ac, ad, ae, aad, aae, ade

- To
- a → c, a → d, a → e, a → ad, a → ae, ad → e

Other algorithms

- Free-Span
- Prefix-Span

- Select sub-sets of data to search within

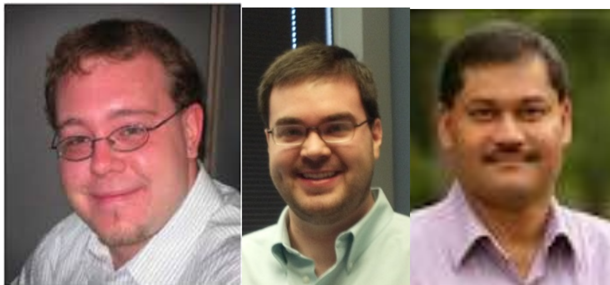
- Faster, but same basic idea as in GPS

Differential Sequence Mining (Kinnebrew et al., 2013)

- Compares the support for sequential patterns between two groups

- Such as high-performing and low-performing students

- To find the patterns that are much more common in one group than the other



MOTIF Extraction

- Another popular approach for finding sequential patterns
- Allows for minor variance between patterns – e.g., closely related patterns can be counted as the same pattern

Next lecture

- Network Analysis