The smaller minsup is, the larger the number of frequent itemsetsSupport monotonicity property:if J ⊆ I, sup(J) ≥ sup(I)
(if I is contained in a transacXon, J is contained in

Downward closure property: every subset of a frequent itemset

$$\text{conf}(I \rightarrow j) = \frac{\text{support}(I \cup j)}{\text{support}(I)}$$

We'll revisit this in a bit:Downward closure property: every subset of a frequent itemset is also frequent

The lift of the rule X ⇒ Y is:

$$\text{lift}(X \Rightarrow Y) = \frac{\sup(X \cup Y)}{\sup(X)\sup(Y)}$$

Lift(X,Y) = 1

X and Y are independent

Lift(X,Y) > 1

X and Y are positively correlated

Lift(X,Y) < 1

X and Y are negatively correlated

Confidence Monotonicity

In general, confidence does not have a monotonicity property

conf(ABC →D) can be larger or smaller than conf(AB →D)

But confidence of rules generated from the same itemset has a monotonicity property!

Example: Generating the Rules

Suppose {A,B,C,D} is a frequent itemset, then the candidate rules are:

A →BCD, B →ACD, C →ABD, D →ABC,

AB →CD, AC → BD, AD → BC, BC →AD, BD →AC, CD →AB,

ABC →D, ABD →C, ACD →B, BCD →A,

If |itemset| = k, then there are $2^k - 2$ candidate association rules

$$\text{conf}(X_L \Rightarrow I - X_L) \geq \text{conf}(X_S \Rightarrow I - X_S)$$

Suppose {A,B,C,D} is a frequent itemset, then:

c(ABC → D) ≥ c(AB → CD)

and

c(AB → CD) ≥ c(A → BCD)

Toivonen's Algorithm

Pass 1:

Start as in the SON algorithm, but lower the threshold slightly for the subset

Add to the itemsets that are frequent in the sample the negative border of these items sets

Pass 2:

Count all candidate frequent itemsets from the first pass, and also count sets in their negative border

If no itemset from the negative border turns out to be frequent, then we found all the frequent itemsets.

What if we find that something in the negative border is frequent?

We must start over again with another sample!

Try to choose the support threshold so the probability of failure is low, while the number of itemsets checked on the second pass fits in main memory

Betweenness centrality

$$g(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

$$C_c'(A) = \left[\frac{\sum_{j=1}^{N} d(A,j)}{N-1}\right]^{-1} = \left[\frac{1+2+3+4}{4}\right]^{-1} = \left[\frac{10}{4}\right]^{-1} = 0.4$$

Negative border: Example

{A,B,C,D} is in the negative border if and only if:

1. It is not frequent in the sample, but
2. All of {A,B,C}, {B,C,D}, {A,C,D}, and {A,B,D} are.

Negative border = an itemset is in the negative border if it is not frequent in the sample but all its immediate subsets are

Immediate subset = "delete exactly one element"

Objective direct measures:

support, confidence, correlation, ...

issues?

Subjective direct measures:

User-based — let users decide if a rule is unexpected, fresh, timely, etc.?

issues?

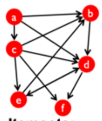
Objective indirect measure?

Put rule into practice (like A/B testing)

e.g., put beer next to diapers and measure sales

issues?

Example



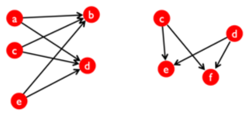
Support threshold s=2

{b,d}: support 3

{e,f}: support 2

And we just found 2 bipartite subgraphs:

Itemsets:
a = {b,c,d}
b = {d}
c = {b,d,e,f}
d = {e,f}
e = {b,d}
f = {}



Interesting Association Rules

Not all high-confidence rules are interesting

The rule X → milk may have high confidence for many item sets X, because milk is just purchased very often (independent of X) and the confidence will be high

One idea: Lift

What makes a good cluster?

Maximize the number of within-cluster connections

Minimize the number of between-cluster connections

$$\phi(A) = \frac{|\{(i,j) \in E; i \in A, j \notin A\}|}{\min(vol(A),\ 2m - vol(A))}$$

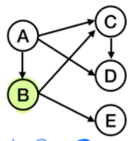
Normalization

divide degree by max (N-1)

Seed = {B}

T =

PPR

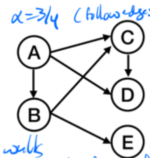


$\frac{3}{4}$ [0 1/3 1/3 1/3 0; 0 0 1/2 0 1/2; 0 0 0 1 0; ...] + 1/4 [0 1 0 0 0; 0 1 0 0 0; 0 1 0 0 0; 0 1 0 0 0; 0 1 0 0 0]

SAME

= [0 1/2 1/4 1/4 0; ...]

Non-personalized PR



α=3/4 (follow edges)

T = transition matrix
tells us P_{ij}
(Markov chain)
↳ encodes the random walk

T = 3/4 [A B C D E; A: 0 1/3 1/3 1/3 0; B: 0 0 1/2 0 1/2; C: 0 0 0 1 0; D: 1/5 1/5 ...; E: 1/5 1/5 ...] + 1/4 [A B C D E; A: 1/5 1/5 1/5 1/5 1/5; B: 1/5 1/5 1/5 ...]

Hashing Columns (signatures)

- **Key idea:** "hash" each column ***C*** to a small ***signature h(C)***, such that:
  - **(1) *h(C)*** is small enough that the signature fits in RAM
  - **(2) *sim(C_1, C_2)*** is the same as the "similarity" of signatures ***h(C_1)*** and ***h(C_2)***

We don't want to compare columns c1, c2, since they might be too large, slowing down the computation

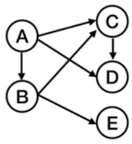
Instead, we compute signatures h(c1), h(c2) that are smaller in size than c1 and c2

The hope:

If c1=c2, then prob(h(c1)=h(c2)) is large

If c1≠c2, then prob(h(c1)=h(c2)) is small

PPR



$\vec{X}_{0,B} = [1\ 0\ 0\ 0\ 0]$

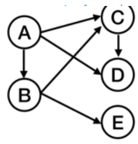
$\vec{X}_{1,B} = \vec{X}_{0,B} \cdot T$

$= [0\ 1/2\ 1/4\ 1/4\ 0]$

$\vec{X}_{2,B} = \vec{X}_{1,B} \cdot T = \ldots$

$\vec{X}_{n,B} = \vec{X}_{n-1,B} = \text{PPR}(B) = [\ldots]$

Non-personalized PR



T = [A B C D E; A: 1/20 6/20 6/20 6/20 1/20; ...: 1/20 1/20 13/40 1/20 17/40; ...]

Solu:

$\vec{X}_0 = [1\ 0\ 0\ 0\ 0]$ ← where is the random walker?

$\vec{X}_1 = \vec{X}_0 \cdot T = [1/20\ 6/20\ 6/20\ 6/20\ 1/20]$

$\vec{X}_2 = \vec{X}_1 \cdot T = \ldots$

$\vec{X}_n = \vec{X}_{n-1}$ → convergence → PR

Finding Similar Columns

- **Next Goal: Find similar columns, Small signatures**
- **Naïve approach:**
  - **1) Signatures of columns:** small summaries of columns (is slow)
  - **2) Examine pairs of signatures** to find similar columns
    - **Essential:** Similarities of signatures and columns are related
  - **3) Optional:** Check that columns with similar signatures are really similar
- **Warnings:**
  - Comparing all pairs may take too much time: **Job for LSH**
    - These methods can produce false negatives, and even false positives (if the optional check is not made)

b bands, r rows/band

- Columns C_1 and C_2 have similarity ***t***
- Pick any band (***r*** rows)
  - Prob. that all rows in band equal =
    - t^r
  - Prob. that some row in band unequal =
    - $1 - t^r$
- Prob. that no band identical =
  - $(1 - t^r)^b$
- Prob. that at least 1 band identical =
  - $1 - (1 - t^r)^b$

A B C D E
PPR(B) = [0.2 0.6 0.1 0.05 0.05]

B, A, C, D, E

Graph Mining (so far)

Frequent itemsets —> graph mining

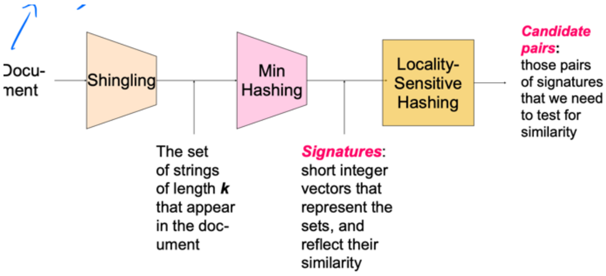
Finding Important Nodes —> centrality measures

Social networks —> locality triadic closure, weak ties, ...

Community detection

Girvan-Newman, PPR (conductance)

Later in the semester —> graph/node embeddings!



1. Shingling: Convert documents to sets

2. Min-Hashing: Convert large sets to short signatures, while preserving similarity

3. Locality-Sensitive Hashing: Focus on pairs of signatures likely to be from similar documents

Random walk provides a measure of **similarity** between two nodes

Maybe we can rank all nodes with respect to a **seed node** using random walks

Find **breaks** in the ranks to identify clusters