COMS 4721: Machine Learning for Data Science
Lecture 23, 4/20/2017

Prof. John Paisley

Department of Electrical Engineering
& Data Science Institute

Columbia University

ASSOCIATION ANALYSIS

Finding high probable sub sets of data.

Setup

Many businesses have massive amounts of customer purchasing data.

- Amazon has your order history
- A grocery store knows objects purchased in each transaction
- Other retailers have data on purchases in their stores

Using this data, we may want to find sub-groups of products that tend to co-occur in purchasing or viewing behavior.

- Retailers can use this to cross-promote products through "deals"
- Grocery stores can use this to strategically place items
- Online retailers can use this to recommend content
- This is more general than finding purchasing patterns

MARKET BASKET ANALYSIS

Association analysis is the task of understanding these patterns.

For example consider the following "market baskets" of five customers.

TID	Items
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}

Items in one basket

which objects tend to appear in baskets at what percentage / what rate?

Using such data, we want to analyze patterns of co-occurance within it. We can use these patterns to define *association rules*. For example,

Association rule.

$$\{\text{diapers}\} \;\Rightarrow\; \{\text{beer}\}$$

For ex: If milk is being purchased, it's not going to distinguish between 1 or 2 bottles of milk. It's going to give an index that milk is being purchased.

Imagine we have:

- p different objects indexed by $\{1, \ldots, p\}$
- A collection of subsets of these objects $X_n \subset \{1, \ldots, p\}$. Think of X_n as the index of things purchased by n^{th} customer $n = 1, \ldots, N$.

Objects of interest that we want to discover: (which items are being purchased.)

1 **Association analysis**: Find subsets of objects that often appear together. For example, if $\mathcal{K} \subset \{1, \ldots, p\}$ indexes objects that frequently co-occur, then

subset of index values between 1 and p.

Goal: to find subsets where $P(\mathcal{K})$ is a large no.

$$P(\mathcal{K}) = \frac{\#\{n \text{ such that } \mathcal{K} \subseteq X_n\}}{N} \quad \text{is large relatively speaking}$$

fraction of market baskets

Example: $\mathcal{K} = \{$`peanut_butter, jelly, bread`$\}$

If a certain subset of objects appear in X_n, makes it more likely that another object is going to appear in that set as well.

2 **Association rules**: Learn correlations. Let A and B be disjoint sets. Then $A \Rightarrow B$ means purchasing A increases likelihood of also purchasing B.

Example: $\{$`peanut_butter, jelly`$\} \Rightarrow \{$`bread`$\}$

TID	Items
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}

Figure: An example of 5 baskets.

Indicator of an item being present in a particular person's basket.

TID	Bread	Milk	Diapers	Beer	Eggs	Cola
1	1	1	0	0	0	0
2	1	0	1	1	1	0
3	0	1	1	1	0	1
4	1	1	1	1	0	0
5	1	1	1	0	0	1

Figure: A binary representation of these 5 baskets for analysis.

PROCESSING THE BASKET

TID	Bread	Milk	Diapers	Beer	Eggs	Cola
1	1	1	0	0	0	0
2	1	0	1	1	1	0
3	0	1	1	1	0	1
4	1	1	1	1	0	0
5	1	1	1	0	0	1

we set

Want to find subsets that occur with probability above some threshold.

For example, does {`bread, milk`} occur relatively frequently?

- Go to each of the 5 baskets and count the number that contain both.
- Divide this number by 5 to get the frequency. 3/5 or 60%
- Aside: Notice that the basket might have more items in it.

Trivial problem when data is small, we simply have to brute-force count every subset.

When $N = 5$ and $p = 6$ as in this case, we can easily check every possible combination. However, real problems might have $N \approx 10^8$ and $p \approx 10^4$.

SOME COMBINATORICS

Some combinatorial analysis will show that brute-force search isn't possible.

Q: How many different subsets $\mathcal{K} \subseteq \{1, \ldots, p\}$ are there?

1 indicates objects in that subset and 0 indicates object is not.

A: Each subset can be represented by a binary indicator vector of length p. The total number of possible vectors is 2^p. [Very large no. as soon as p even gets moderately big.]

Q: Nobody will have a basket with every item in it, so we shouldn't check every combination. How about if we only check up to k items?

Only care about checking a subset of k objects picked from a superset of P total objects.

A: The number of sets of size k picked from p items is $\binom{p}{k} = \frac{p!}{k!(p-k)!}$. For example, if $p = 10^4$ and $k = 5$, then $\binom{p}{k} \approx 10^{18}$.

Looking to find subsets that tend to occur together very frequently. The problem is we can't do a

Takeaway: Though the problem only requires counting, we need an algorithm that can tell us which $\mathcal{K}$ we should count and which we can ignore.

QUANTITIES OF INTEREST (What is it that we want to count?)

Before we find an efficient counting algorithm, what do we want to count?
Let A and B be 2 subset of the superset {1... P} and then we define A and B to be a partition of K into 2 disjoint sets.

- Again, let $\mathcal{K} \subset \{1, \ldots, p\}$ and $A, B \subset \mathcal{K}$, where $A \cup B = \mathcal{K}, A \cap B = \emptyset$.

We're interested in the following empirically-calculated probabilities:

joint probability of A and B.

1. $P(\mathcal{K}) = P(A, B)$: The *prevalence* (or support) of items in set $\mathcal{K}$. We want to find which combinations co-occur often.

conditional probability of the set B given the set A.

2. $P(B|A) = \frac{P(\mathcal{K})}{P(A)}$ ← P(A,B): The *confidence* that B appears in the basket given A is in the basket. We use this to define a *rule* $A \Rightarrow B$. *
3. $L(A, B) = \frac{P(A,B)}{P(A)P(B)}$ ** $= \frac{P(B|A)}{P(B)}$: The *lift* of the rule $A \Rightarrow B$. This is a measure of how much *more* confident we are in B given that we see A.

* So if you tell me that A is my basket, what is the probability that items in set B are also in my basket.

** Joint probability of A and B divided by the marginal probability of A times the marginal probability of B.

EXAMPLE

For example, let

A & B don't share any item and the partition

$$\mathcal{K} = \{\texttt{peanut\_butter, jelly, bread}\},$$

$$A = \{\texttt{peanut\_butter, jelly}\}, B = \{\texttt{bread}\}$$

- A *prevalence* of 0.03 means that `peanut_butter, jelly` and `bread` appeared together in 3% of baskets. (purchases)
- A *confidence* of 0.82 means that when both `peanut_butter` and `jelly` were purchased, 82% of the time `bread` was also purchased. $P(B|A)$

Means that in 82% of baskets I am restricting to, bread was also purchased.

- A *lift* of 1.95 means that it's 1.95 more probable that `bread` will be purchased given that `peanut_butter` and `jelly` were purchased. $\frac{P(B|A)}{P(B)}$

compared to the probability that bread was purchased at all.

The fraction says what's the lift or the bump in the probability that I get. If you tell me how much more confident I was in B without any knowledge

Apriori algorithm

without having to search all 2^p subsets.

⇛ The goal of the **Apriori algorithm** is to quickly find all of the subsets $\mathcal{K} \subset \{1, \ldots, p\}$ that have probability greater than a predefined threshold t.

- Such a $\mathcal{K}$ will contain items that appear in at least $N \cdot t$ of the N baskets.
- A small fraction of such $\mathcal{K}$ should exist out of the 2^p possibilities.

Apriori uses properties about $P(\mathcal{K})$ to reduce the number of subsets that need to be checked to a small fraction of all 2^p sets. (using simple rules of logic.)

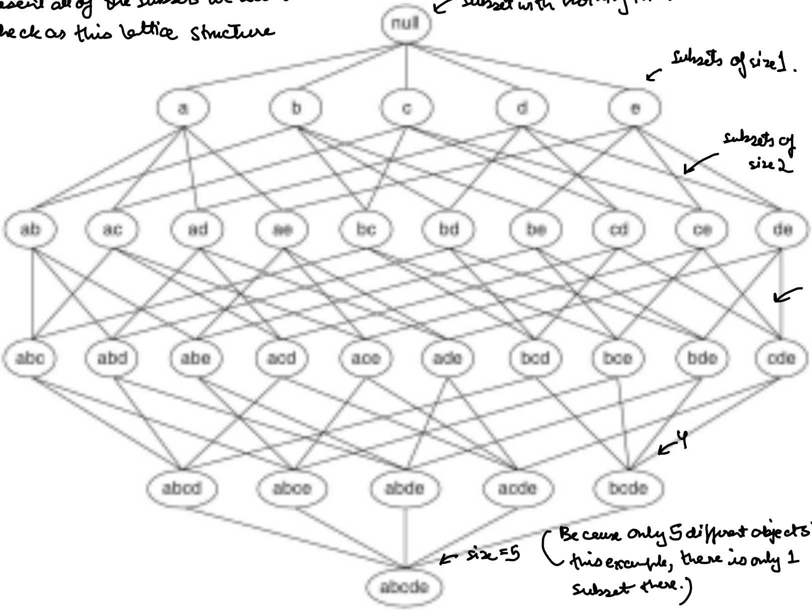
Start by constructing all subsets of size 1. Some surviving & some not surviving. Based on that, it's going to move subsets of size 2.

- It starts with $\mathcal{K}$ containing 1 item. It then moves to 2 items, etc.
- Sets of size $k-1$ that "survive" this process are going to factor in when constructing subsets of size k to check. help determine sets of size k to check.
- Important: Apriori finds *every* set $\mathcal{K}$ such that $P(\mathcal{K}) > t$, but it's going to use certain properties about probabilities to eliminate subsets

Next slide: The structure of the problem can be organized in a lattice.

LATTICE REPRESENTATION

Represent all of the subsets we want to check as this lattice structure



subset with nothing in it

subsets of size 1.

subsets of size 2

3

4

size = 5

(Because only 5 different objects in this example, there is only 1 subset there.)

* So we take k union with some other set a is also not gonna be big enou
So in other words if p(k) is less than t,
then we know that the p(k) prime has to also be less than t.

We can use two properties to develop an algorithm for efficiently counting.

Meaning that we have a certain set k that set k does not appear in fraction t of baskets we're looking at. $[P(K)<t]$

1. If the set $\mathcal{K}$ is not big enough, then $\mathcal{K}' = \mathcal{K} \cup A$ with $A \subset \{1, \ldots, p\}$ is * not big enough. In other words: $P(\mathcal{K}) < t$ implies $P(\mathcal{K}') < t$

   we can avoid having to construct any larger subset of k then we automatically know that p(a) > p(k), has to greater than t.

   e.g., Let $\mathcal{K} = \{a, b\}$. If these items appear together in x baskets, then the set of items $\mathcal{K}' = \{a, b, c\}$ appears in $\leq x$ baskets since $\underline{\underline{\mathcal{K} \subset \mathcal{K}'}}$.

   Mathematically: $P(\mathcal{K}') = P(\mathcal{K}, A) = P(A|\mathcal{K})P(\mathcal{K}) \leq \underbrace{P(\mathcal{K})}_{\text{by assumption}} < t$

   because probability has to be less than 1. we know the probability of these 2 no.s must be less then p(k).

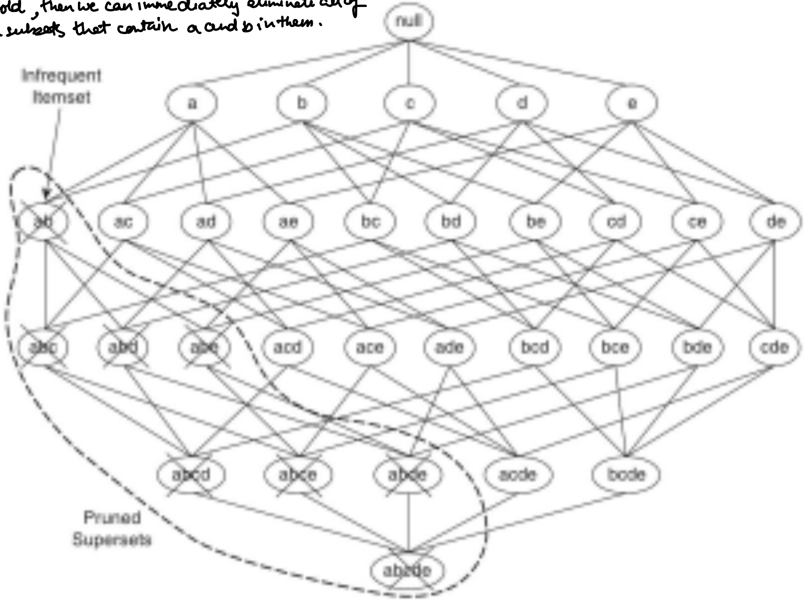
2. By the converse, if $P(\mathcal{K}) > t$ and $A \subset \mathcal{K}$, then $P(A) > P(\mathcal{K}) > t$.

   So if k satisfies our threshold, all subsets of k also have to satisfy our threshold.

   We take p(k) & multiply it by a no. less then 1. reduce the size of P(k). If 1, it would be equal.

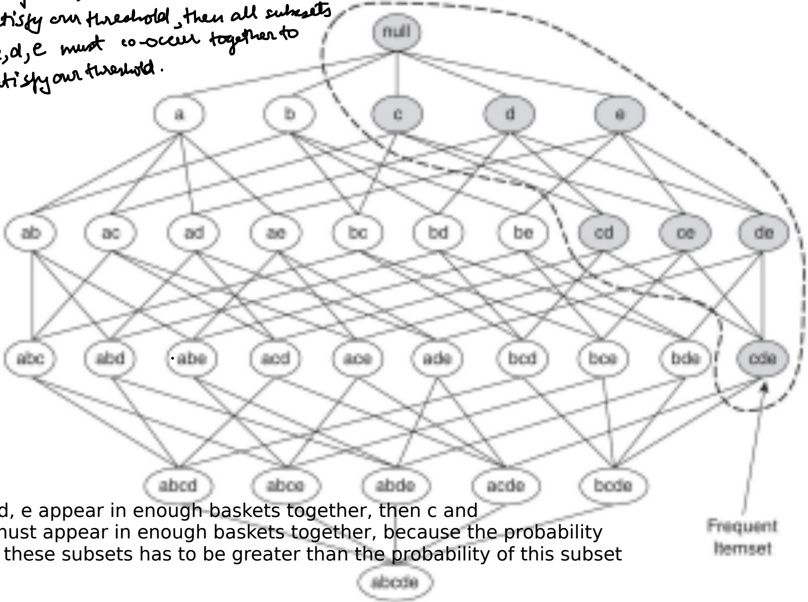
FREQUENCY DEPENDENCE: PROPERTY 1

If a & b don't co-occur enough times to pass our threshold, then we can immediately eliminate all of other subsets that contain a and b in them.



FREQUENCY DEPENDENCE: PROPERTY 2

So: if object c, d, e occur together enough times to satisfy our threshold, then all subsets of c, d, e must co-occur together to satisfy our threshold.



So if c, d, e appear in enough baskets together, then c and d also must appear in enough baskets together, because the probability of all of these subsets has to be greater than the probability of this subset

Here is a basic version of the algorithm. It can be improved in clever ways.

that we don't expect there to be many subsets that co-occur together more than Nt in our dataset.

Apriori algorithm

Set a threshold $N \cdot t$, where $0 < t < 1$ (but relatively small).

check subsets of size 1 that occur frequently enough.

1. $|\mathcal{K}| = 1$: Check each object and keep those that appear in $\geq N \cdot t$ baskets.

2. $|\mathcal{K}| = 2$: Check all pairs of objects that survived Step 1 and keep the sets that appear in $\geq N \cdot t$ baskets.

⋮

k. $|\mathcal{K}| = k$: Using all sets of size $k - 1$ that appear in $\geq N \cdot t$ baskets,
   - Increment each set with an object surviving Step 1 not already in the set.
   - Keep all sets that appear in $\geq N \cdot t$ baskets

So it's still a brute force search, but it's reducing that amount of search we need to do by not fruitlessly searching

It should be clear that as k increases, we can hope that the number of sets that survive decrease. At a certain $k < p$, no sets will survive and we're done.

More considerations

So we're going to assume if we're presented with a subset, we'll able to verify whether it's something we should keep or not. So really the only question is will we check every subset that passes this threshold.

by brute force counting.

1. We can show that this algorithm returns *every* set $\mathcal{K}$ for which $P(\mathcal{K}) > t$.

- Imagine we know every set of size $k-1$ for which $P(\mathcal{K}) > t$. Then every potential set of size k that could have $P(\mathcal{K}) > t$ will be checked. Then we need to show that every subset of size $k-1$, we're going to check every potential subset of size k that could possibly have empirical probability greater then t.

  For ex: All subsets of this set of 3 objects has to also appear in more then Nt baskets.

  e.g. Let $k = 3$: The set $\{a, b, c\}$ appears in $> N \cdot t$ baskets. Will we check it?

  Known: $\{a, b\}$ and $\{c\}$ must appear in $> N \cdot t$ baskets.

  Assumption: We've found $\mathcal{K} = \{a, b\}$ as a set satisfying $P(\mathcal{K}) > t$. union of a & b with c

  * **Apriori algorithm**: We know $P(\{c\}) > t$ and so will check $\{a, b\} \cup \{c\}$.

  ** **Induction**: We have all $|\mathcal{K}| = 1$ by brute-force search (start induction).

By Apriori algorithm, check these 2 sets as if they were different sets when in fact they're the same set.

2. As written, this can lead to duplicate sets for checking, e.g., $\{a, b\} \cup \{c\}$ and $\{a, c\} \cup \{b\}$. Indexing methods can ensure we create $\{a, b, c\}$ once.

3. For each proposed $\mathcal{K}$, should we iterate through each basket for checking? There are tricks to make this faster that takes structure into account.

So we don't have to actually iterate through all n baskets to check whether a proposed k appears in that basket or not.

* So we know that we will check a, b and c at the III^{rd} step of this algorithm.

* *Now we follow by induction,
  - → we know that we have all subsets of size 1 by the brute-force search.
  - → And by induction, therefore we know we have all subsets of size 2, by brute force search.

    different things

  - → Again by these rules, we know we have all subsets of size 3 & so on.

    So it's an inductive proof.

FINDING ASSOCIATION RULES

Apriori algorithm has shown that we can find all sets K

We've found all $\mathcal{K}$ such that

$$P(\mathcal{K}) > t.$$

Now we want to find association rules.

$P(A \cup B) = K$ & $P(A \cap B) = \phi$

why all possible splits? Isn't there only 1 split.

These are of the form $P(A|B) > t_2$
where we split $\mathcal{K}$ into subsets A and B.

- For every single set K we want to check all possible splits of that set into sets a and b. • Such that we can find the conditional probability and if that conditional probability is greater than t_2, we keep that as an association rule. • If it's less than t_2 we throw it away.

Notice:

1. $P(A|B) = \frac{P(\mathcal{K})}{P(B)}$. ↳ joint probability of A and B.

   Find $p(K)$ & find $p(B)$ separately. After apriori algo.

2. If $P(\mathcal{K}) > t$ and A and B partition $\mathcal{K}$ (subsets), then we know $P(A) > t$ (Prob. of that subset) and $P(B) > t$.

   we don't need to go back to the data to find these conditional probability

3. Since Apriori found all $\mathcal{K}$ such that $P(\mathcal{K}) > t$, it found $P(A)$ and $P(B)$, so we can calculate $P(A|B)$ without counting again.

   separate from their union. And also found the probabilities of these 2 sets.

Message: Once we have ran the apriori algorithm and found all subsets such that their empirical probability > t. We store these subsets. And we store the empirical probabilities. And we can quickly check all conditional probabilities by taking their stored probabilities.

EXAMPLE

Feature	Demographic	# Values	Type
1	Sex	2	Categorical
2	Marital status	5	Categorical
3	Age	7	Ordinal
4	Education	6	Ordinal
5	Occupation	9	Categorical
6	Income	9	Ordinal
7	Years in Bay Area	5	Ordinal
8	Dual incomes	3	Categorical
9	Number in household	9	Ordinal
10	Number of children	9	Ordinal
11	Householder status	3	Categorical
12	Type of home	5	Categorical
13	Ethnic classification	8	Categorical
14	Language in home	3	Categorical

no ordering.

Data

$N = 6876$ questionnaires [Market baskets]

14 questions coded into $p = 50$ items

For example:

- ordinal (2 items): Pick the item based on value being $\lessgtr$ median
- categorical: item = category

  x categories $\rightarrow$ x items

- Based on the item encoding, it's clear that no "basket" can have every item.

Each basket can only have one of the categorical values for each question. It can have only one of 2 possible ordinal values.

- We see that association analysis extends to more than consumer analysis.

We want to different subsets of high probability. Already with this many responses we can't use a brute force search of all of the possible subsets

EXAMPLE

Association rules learnt

Among all of questionaries if 3 objects were true, the then income was ≥ 40,000, 80.8% of time. 19.2% their income was less.

In 6876 questionnaries, 13.4% of these had these 4 responses selected.

Association rule 1: Support 13.4%, confidence 80.8%, and lift 2.13.

Broke 4 object subset & broke it into 2 sets.

One set a of these 3 objects

One set bag containing this object

$$\begin{bmatrix} \text{language in home} & = & \textit{English} \\ \text{householder status} & = & \textit{own} \\ \text{occupation} & = & \{\textit{professional/managerial}\} \end{bmatrix}$$

$$\Downarrow$$

$$\text{income} \geq \$40{,}000$$

2 times more confident that a person makes ≥ $ 40,000 a year if you tell me these 3 things are true about that person.

satisfied all 5 of these properties.

Twice as confident that a person doesn't have a college degree, if you tell me this information about that person then I would be otherwise.

Association rule 2: Support 26.5%, confidence 82.8% and lift 2.15.

If I don't get to know anything about that person, then I am only half as confident that the person will make more than amount of dollars per year.

?

We calculate these things by counting (empirical probabilities).

Not a trivial task because we have many responses.

$$\begin{bmatrix} \text{language in home} & = & \textit{English} \\ \text{income} & < & \textit{\$40,000} \\ \text{marital status} & = & \textit{not married} \\ \text{number of children} & = & \textit{0} \end{bmatrix}$$

$$\Downarrow$$

$$\text{education} \notin \{\textit{college graduate, graduate study}\}$$

Priori algo → simply counting things we know already will not satisfy our threshold.