01 - Association Analysis

Frequent Itemsets, Market Basket Analysis, and Association Rule Mining

SYS 4582/6018 | Spring 2019

01-association.pdf

Contents

1 Reading

- MMDS 6.1 and 6.2
- ITDM 5.1-5.3; 5.7-5.8
- ESL 14.2
- R package `arules`
- R package `arulesViz`

2 Association Analysis Motivation

2.1 Market-Basket Analysis

- A grocery store records the items purchased in a set of transactions. For example,

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}

- If they can discover items **frequently** purchased together, they may be able to increase business.
  - If {Diapers, Beer} are frequently purchased together, the store could put diapers on sale and increase the price of beer.
  - If {Bread, Peanut Butter} are frequently purchased together, they may put some Peanut Butter on the shelf next to bread.
  - In on-line retail, you will often see "People who buy {X,Y} also tend to buy {Z}".
- The form of association can also be important. Would it be better to put Peanut Butter on the shelf next to Bread or Bread in the Peanut Butter aisle?
- **Association Rules** are rules that describe potential relationships *of interest* between itemsets.
  - Given a set of transactions, find rules that will predict the occurrence of an item based on the occurrences of other items in the transaction
  - For example, the rule {Diapers} $\rightarrow$ {Beer} suggests that customers who buy Diapers tend to purchase Beer more than *expected*.
  - Of course, we need to discuss metrics that will permit us to quantitatively evaluation the discovered rules.
  - **Co-occurrence doesn't necessitate causality!**

2.2 Other Applications

- Market Basket
  - Items: Products for sale
  - Baskets: Sets of products purchased in single trip
  - Needs rule to occur frequently to make worthwhile
- Text Mining
  - Items: Documents
  - Baskets: Sentences
  - Frequent itemsets could indicate plagiarism
- Health Mining
  - Items: Drugs and Side-Effects
  - Baskets: Patients

- Discover combinations of drugs that result in particular side-effects
  - {Drug 1, Drug 4, Drug 6} $\rightarrow$ {Psychosis}
- Note that absence of drug/side-effect could also be important
  - {Drug 1, Drug 4, Not Drug 7} $\rightarrow$ {Psychosis}

Patient	Items
1	{Drug 1, Drug 3, Rash, Weight-Loss}
2	{Drug 2, Drug 3, Headaches}

- Transportation
  - Items: Features collected after vehicle crash
  - Baskets: The set of features present during vehicle crash
  - Discover dangerous/safe combinations

CrashID	Items
1	{Wet, Night, Rural}
2	{Alcohol, Night, Lights-off}
3	{Day, Flat tire, Elderly}

- Survey Data
  - Items: Potential Responses to questionnaire
  - Baskets: Set of responses by respondent
  - {Married, Own house} $\rightarrow$ {income $\in$ [60K, 100K]}

ID	Items
1	{Sex = *Male*, Age = [20,25], Type of home = *House*, ... }
2	{Sex = *Female*, Age = [30,35], Type of home = *Apt*, ... }
3	{Sex = *Male*, Age = [65+], Type of home = *House*, ... }

3 Market Basket Model

3.1 Data

There are two equivalent data formats:

- **Transaction Lists**

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}

- **Binary Incident Table/Matrix**

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

3.2 Definitions

- K possible items
- N total transactions
- Let I be an itemset
  - E.g., $I = \{\text{Cola, Bread, Fruit}\}$, $I = \{\text{Bread}\}$
  - There are $2^K - 1$ possible itemsets
- The **support** of I is the fractions of transactions that contain I
  - $S(I) = (\text{\# of transactions containing } I)/N$
- The support is an estimate of the probability of getting I in a future basket.
  - $S(I) \hat{=} \Pr(I \subseteq T)$, where T is a future basket
- A **Frequent Itemset** is one that has support greater than or equal to the **support threshold (minsup)**, s.
- **Association Rule** is an if-then expression about the content of the baskets
  - The expression $I \to J$ means that if a basket contains I, then it is *more likely* to also contain J.
  - I and J should be *disjoint*, e.g., $I \cap J = \emptyset$
- The **support** of an association rule is the frequency in which both itemsets are in the same basket
  - $S(I \to J) = (\text{\# of transactions containing } I \text{ and } J)/N$
  - $S(I \to J) = S(I, J) \hat{=} \Pr(I \subseteq T, J \subseteq T)$, where T is a future basket.

- The **confidence** of a rule is an estimate of the conditional probability of J being in a basket given the basket already contains I
  - $C(I \to J) = \frac{S(I,J)}{S(I)} \triangleq \frac{\Pr(I \subseteq T, J \subseteq T)}{\Pr(I)} = P(J \subseteq T | I \subseteq T)$
- A **High Confidence** rule is one that has confidence greater than or equal to the **confidence threshold (minconf)**, c.
- The classic **Association Rule Discovery** task is to find all rules that have *support* $\geq s$ and *confidence* $\geq c$
  1. Find all *frequent* itemsets
  2. For every subset $A \subset I$, generate the rule $A \to A \setminus I$
  3. It is easy to find the *largest* confidence set derived from I. Do you see how?

3.3 Practice

Your Turn #1

TID	Items
0001	{a, d, e}
0024	{a, b, c, e}
0012	{a, b, d, e}
0031	{a, c, d, e}
0015	{b, c, e}
0022	{b, d, e}
0029	{c, d}
0040	{a, b, c}
0033	{a, d, e}
0038	{a, b, e}

1. What are the value of K and N?
2. What is the *support* of {e}, {b, d}, and {b, d, e}?
3. What is the *confidence* of {b, d} $\to$ {e} and {e} $\to$ {b, d}? Is confidence symmetric?
4. Think of an approach/algorithm to find *all* **Association Rules** with $s = 4$ and $c = .70$.

3.4 Measures of Interestingness

The utility of a particular association rule is a based on the needs of the business/research (e.g., sell more products, increase profit). When in a purely exploratory mode, the **interestingness** of a rule determines its value. So far, we have only examined *support* and *confidence*, but there are many other measures of interestingness.

3.4.1 Think About It

Your Turn #2

For each of the following questions, provide an example of an association rule from the market basket domain that satisfies the following conditions. Also, describe whether such rules are subjectively interesting.

1. A rule that has high support and high confidence.
2. A rule that has reasonably high support but low confidence.
3. A rule that has low support and low confidence.
4. A rule that has low support and high confidence.

3.4.2 Lift

- Why would the rule Vodka -> Caviar be interesting? Even if it has low support?
- In general, an interesting rule is one that occurs more (or less) than *expected*.
- Support and Confidence are based on an expected **uniform** distribution.
- **Lift**, on the other hand is based on statistical independence.
  - Notice that lift is **not** directional

$$L(I \to J) = \frac{C(I \to J)}{S(J)}$$

$$= L(J \to I)$$

- If I and J are independent, then $\Pr(I, J) = \Pr(I)\Pr(J)$.

- Lift > 1 indicates positive association
- Lift < 1 indicates negative association (I and J *inhibit* each other)

- Because its a ratio, *lift* can be large even when the *support* is low and small when the *confidence* is large.

Your Turn #3

Consider the following beverage preferences from 1000 people:

	Coffee	No Coffee
Tea	150	50
No Tea	650	150

1. What is the *support* of $\{\text{Tea}\} \to \{\text{Coffee}\}$?
2. What is the *confidence* of $\{\text{Tea}\} \to \{\text{Coffee}\}$?
3. What is the *lift* of $\{\text{Tea}\} \to \{\text{Coffee}\}$?
4. Interpret.

3.4.3 Other Measures

There are many other measures. Here are a few:

- **Piatetsky-Shapiro (PS)/Leverage**: difference of observed and expected.

$$\text{PS}(I, J) = S(I, J) - S(I)S(J)$$

- **Correlation** (ϕ): normalized version of **PS**.

$$\phi(I \to J) = \frac{\text{PS}(I, J)}{\sqrt{S(I)(1 - S(I)S(J)(1 - S(J)))}}$$

- **Added Value (AV)**: difference between conditional and unconditional.

$$\begin{aligned} \text{AV}(I \to J) &= C(I \to J) - S(J) \\ &= \text{PS}(I, J)/S(I) \\ &\widehat{=} \Pr(J|I) - \Pr(J) \end{aligned}$$

- **Conditional Entropy H(J|I)**: amount of info needed to describe distribution of J given I.

$$H(J|I) = H(I, J) - H(I)$$

- **Mutual Information (MI)**: type of Kullback-Leibler divergence where independence is used for alternative distribution.

$$\mathrm{MI(I, J) = H(J) - H(J|I)}$$

Your Turn #4

Consider the following results from 1000 people:

	Honey	No Honey
Tea	100	100
No Tea	20	780

1. What is the *support* of {Tea} → {Honey}?
2. What is the *confidence* of {Tea} → {Honey}?
3. What is the *lift* of {Tea} → {Honey}?
4. What is the *Added Value (AV)* of {Tea} → {Honey}?
5. What is the *Piatetsky-Shapiro (PS)* score for {Tea} → {Honey}?
6. Interpret.

4 Apriori Algorithm

4.1 Introduction

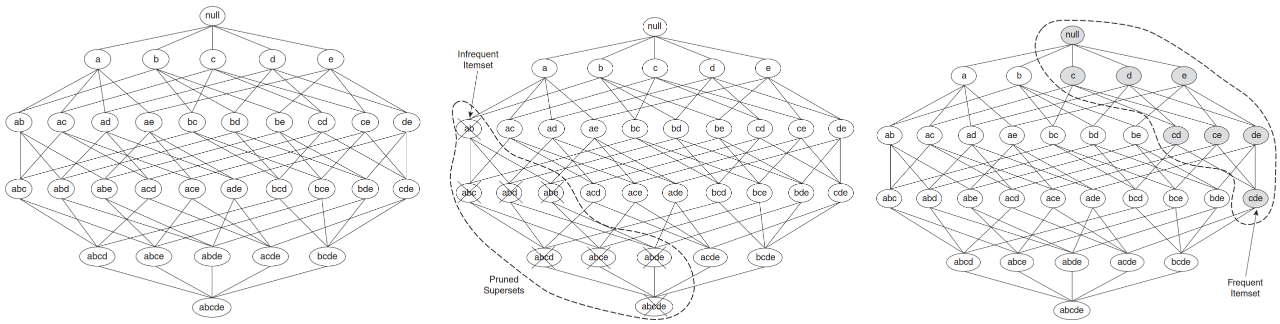
- **Goal:** Discover associations between itemsets.
  - Interested in discovering *frequent itemsets* - those that appear in $\geq s$ baskets.
  - Interested in discovering *high confidence rules* - those with confidence $\geq c$ scores.
- **Problem:** There are $2^K - 1$ possible itemsets and $3^K - 2^{K+1} + 1$ possible rules.
  - Wal-mart has 100K items
  - 10B webpages
  - Memory and computing time issues
- **Solution:** The Apriori principle
  - *If an itemset is frequent, than all of its subsets must also be frequent*.
  - Equivalently, *If an itemset is not frequent, then none of its unions will be frequent*.

- Thus if item i is does not appear in at least s baskets (i.e., not frequent), then no itemsets that include item i will appear in at least s baskets.

4.2 Apriori Approach

- Because of the size of association data, traditional approaches have focused on finding the rules that have support $\geq s$ and confidence $\geq c$.
- This can be efficiently approached in a two step process
  1. Find all frequent itemsets
  2. Find rules, using the frequent itemsets, with high confidence

4.2.1 Finding Frequent Itemsets



4.2.2 Finding High Confidence Rules

Your Turn #5

A frequent itemset consists of $k = 4$ items {a,b,c,d}.

1. How many association rules can be formed from this itemset?
2. Suppose the *confidence* for rule {b,c,d} $\rightarrow$ {a} is low (i.e., $C(\{b,c,d\} \rightarrow \{a\}) < c$).
   a. Do we need to calculate $C(\{d\} \rightarrow \{a,b,c\})$?
   b. Do we need to calculate $C(\{a,b\} \rightarrow \{c,d\})$?

4.3 Targeted Items

- Suppose we are especially interested in one item (e.g., Honey).
- If we want to find an association itemset that *increases the probability* the item of interest is in a basket (i.e., item of interest on the rhs), then it may help to convert the problem into a *supervised learning* classification problem.

- Same "association is not causation" caveat applies
- If we want to see how the item of interest associates with other items (i.e., item of interest on lhs), then we don't need to run a full search, but only build up the rhs items sequentially.

5 Example: Instacart Data

5.1 Instacart Contest Data

Instatcart is an on-line grocery delivery company. The created a Kaggle contest and released some of their data. Here are the details:

> Curious about the food Americans eat? Look no further.
>
> Instacart is excited to announce our first public dataset release, "The Instacart Online Grocery Shopping Dataset 2017". This anonymized dataset contains a sample of over 3 million grocery orders from more than 200,000 Instacart users.
>
> For each user, we provide between 4 and 100 of their orders, with the sequence of products purchased in each order. We also provide the week and hour of day the order was placed, and a relative measure of time between orders.

- The Press release is here: https://tech.instacart.com/3-million-instacart-orders-open-sourced-d40d29ead6f2
- The original contest goal was to predict which items will be re-purchased, but we will run association analysis and see what we will discover.
  - Notice that customers that bought {Hass Avocado, Small} frequently bought {Red Vine Tomato} and {Yellow Onions, Loose}
- Data Available (~200MB) https://www.instacart.com/datasets/grocery-shopping-2017

order_id	product_id	product_name
1	49302	Bulgarian Yogurt
1	11109	Organic 4% Milk Fat Whole Milk Cottage Cheese
1	10246	Organic Celery Hearts
1	49683	Cucumber Kirby
1	43633	Lightly Smoked Sardines in Olive Oil
1	13176	Bag of Organic Bananas
1	47209	Organic Hass Avocado
1	22035	Organic Whole String Cheese
36	39612	Grated Pecorino Romano Cheese
36	19660	Spring Water
36	49235	Organic Half & Half
36	43086	Super Greens Salad
36	46620	Cage Free Extra Large Grade AA Eggs
36	34497	Prosciutto, Americano
36	48679	Organic Garnet Sweet Potato (Yam)

6 R Code

The R Code *instacart.R* (see course webpage) contains a detailed example.

6.1 Load Data

1. Download data from https://www.instacart.com/datasets/grocery-shopping-2017

- It is compressed in a tar.gz format
- Use a program to unzip (e.g., 7-zip) and extract the files
- We will be using the files: *order_products__train.csv* and *products.csv*
- Note: its possible to do all of this from within R

2. Load into R.
   - Here using `readr::read_csv()`

```
library(readr)
data.dir = <"path_to_data">    # set path to the uncompressed data
orders = read_csv(file.path(data.dir, "order_products__train.csv"))
products = read_csv(file.path(data.dir, "products.csv"))
```

3. Join orders and products to get data frame of *order_id*, *product_id*, and *product_name*.
   - Here using `dplyr` functions.

```
library(dplyr)
Y = left_join(orders %>% select(order_id, product_id),
              products %>% select(product_id, product_name),
              by = "product_id") %>%
  arrange(order_id)
```

6.2 Clean Data

- The data should have a one-to-one mapping between *product_name* and *product_id*
- Transactions should not contain duplicate items
- May need to clean item names or ids

6.3 arules package

- The R package `arules` has functions for performing association analysis
  – More details in the vignette https://cran.r-project.org/web/packages/arules/vignettes/arules.pdf
- The main function is `apriori()`
  – the arguments *parameter*, *appearance*, and *control* are used to run certain analyses
- The data needs to be converted to *transaction* class (essentially a sparse binary matrix)
  – This is done by using the function `as(tList, "transactions")`, where *tList* is a **transaction list**
  – This means we need to convert our transaction data frame into a transaction list

6.4 Find Frequent Itemsets

- Frequent itemsets can be found by running the function

```
apriori(trans, parameter = list(support = .01, target="frequent"))
```

- *trans* is the transactions object
- support threshold is set to .01
- target="frequent" indicates we only want frequency, not rules

- The `parameter=` argument can be used focus on certain classes of itemsets
  – E.g., support, minlen, maxlen
- Add *lift* with the `interestMeasure()` function

6.5 Find Association Rules

- Association rules can be found by running the function

```
apriori(trans, parameter = list(support = .001, confidence=.50,
```

  - *trans* is the transactions object
  - support threshold is set to $s = .01$
  - confidence threshold is set to $c = .50$
  - target="rules" indicates we only want rules

- Add interest measures with the `interestMeasure()` function

6.6 Plot Association Rules

- the `arulesViz` R package provides some plotting functionality
  - More details in the vignette https://cran.r-project.org/web/packages/arulesViz/vignettes/arulesViz.pdf

6.7 Target specific itemsets

- Specific itemsets can be targeted by setting the `appearance=` parameter

7 Statistical Issues

7.1 Repeatable Patterns

- **Co-occurrence does not mean causality**

- The patterns/associations of interest are those that are *repeatable*, i.e., the patterns will appear in future transactions.

- If future transactions will come from a different distribution than the observed data, proceed with caution!

- Rule metrics are constructed from *counts* e.g., $N(I), N(J), N(I, J)$.
  - If the transactions are *independent*, then $N(A) \sim \mathrm{Bino(p_A, N)}$.
    * What is a good estimate of p_A?
    * What are the confidence intervals for p_A?
  - What if the the transactions are **not** independent?

Your Turn #6

Suppose we have $N = 10\,000$ transactions and found two itemsets I and J with $N(I, J) = 500$, $N(I) = 1000$, $N(J) = 4000$.

1. What is a good point estimate of $\Pr(I, J)$?
2. What are the confidence intervals for $\Pr(I, J)$ (pick the confidence level)?
3. What is a good point estimate of $P(J|I)$?
4. Confidence interval for $P(J|I)$?
5. What should we do if the transactions are *not* independent? E.g., the N transactions are made by $M = 100$ customers.

7.2 Probability Estimation and Statistical Significance

- **Sampling:** If metrics are based on counts/probability estimates, then sampling can help alleviate the computational burden while still discovering the associations
  - Much in the same way as polling is used to estimate the percentage of people who will vote for a particular candidate.
  - Note: Should you sample items or transactions, or both?
- **Multiple Testing:** Statistical significance and confidence intervals are approaches to help us avoid *spurious associations*. One challenge with association analysis is that we are searching through a large data set and are likely to find (empirically) strong associations that are due to chance alone.
  - We need to be especially careful about rules/associations that are based on small counts as they will have the most variation (especially metrics based on ratios).
- **Simpson's Paradox:** See ITDM 5.7.3

7.2.1 Simulation

Here is some simulated transaction data. All items are generated independently and all transactions are independent. Ten items (1-10) have probability of .25 of being included in any transaction, ten items (11-20) have probability of .10 of being included in any transaction, and the remaining items have probability of .01.

```
#-- Settings
K = 2000   # total number of items
N = 50000 # number of transactions
p = c(rep(.25, 10), rep(.10, 10), rep(.01, K-20)) # probabilities
```

```
#-- function to generate transactions
get_transaction <- function(p){
  K = length(p)
  (1:K)[runif(K) <= p]
}

#-- Simulate N transactions
X = replicate(N, get_transaction(p))

#-- Run apriori
library(arules)
trans = as(X, "transactions")

apriori(trans, parameter=list(support=100/N, confidence=.20, minlen=2),
        control=list(verbose=FALSE)) %>%
  sort(by="lift") %>% head(n=5) %>% inspect()
#>     lhs           rhs support confidence lift  count
#> [1] {5,10,15} => {1} 0.00208 0.3549     1.408 104
#> [2] {2,3,16}  => {8} 0.00204 0.3290     1.323 102
#> [3] {1643}    => {2} 0.00322 0.3286     1.321 161
#> [4] {3,7,16}  => {9} 0.00222 0.3265     1.308 111
#> [5] {1,10,15} => {5} 0.00208 0.3220     1.301 104
```

- There are no “real” patterns in these data, but we can find many apparent patterns
- Notice the relatively large values of *lift* (expected lift is 1)
- Notice from which groups the items come from (high, medium, low probabilities). Why do we get these patterns?
- **Statistical Testing Approaches:**
  - Due to multiple comparisons, the exact distribution of a test statistic is often elusive.
  - However, it may not be too difficult estimate the distribution of a test statistic using simulation.
  - **Independence:** randomly permute each column (in the transaction matrix) independently. However, this may change the distribution of the transaction length (number of items in a transaction). A method called *swap randomization* can help.
  - Bootstrap the transactions
  - Cross-Validation

8 Extensions

8.1 Non-Market Basket Format

- Association analysis can be performed on non-market basket data.
- It only requires a data set of *binary* features
  - Categorical Data is dummy coded
  - Numerical data is binned

8.1.1 Categorical Data

- A *categorical variable* is one that can one of k possible values
  - *nomial* is unordered (e.g., *blue, green, brown*)
  - *ordinal* is ordered (e.g., *Great, Good, Average, Below, Poor*)
- Dummy coding (*one-hot-encoding*) is the way to convert a categorical variable to a set of *binary* variables

  - A k level variable will create k binary variables
  - This is different than the usual dummy coding for regression, where $k-1$ levels are generated
  - This full k level coding is referred to as *one-hot-encoding* in the machine learning literature

```
#-- categorical vector with k=3 levels
x.cat = c('a', 'b', 'a', 'c', 'b')

#-- Convert to matrix with 3 columns
model.matrix(~x.cat-1)
#>   x.cata x.catb x.catc
#> 1      1      0      0
#> 2      0      1      0
#> 3      1      0      0
#> 4      0      0      1
#> 5      0      1      0
#> attr(,"assign")
#> [1] 1 1 1
#> attr(,"contrasts")
#> attr(,"contrasts")$x.cat
#> [1] "contr.treatment"
```

- May need to be creative in making interesting levels.
  - *Eyes* $=$ *blue* vs. *Eyes* $\neq$ *blue*

8.1.2 Numeric Data

- Numeric data must be converted into binary format
- For example, *age* $\leq 25 =$ *TRUE*
- Numeric data could also be discretized or binned. This converts it into categorical data which is then dummy encoded.
  - E.g., *age 14-17*, *age 18-24*, ..., *age 65+*

8.1.3 Categorical and Numeric Data in `arules` package

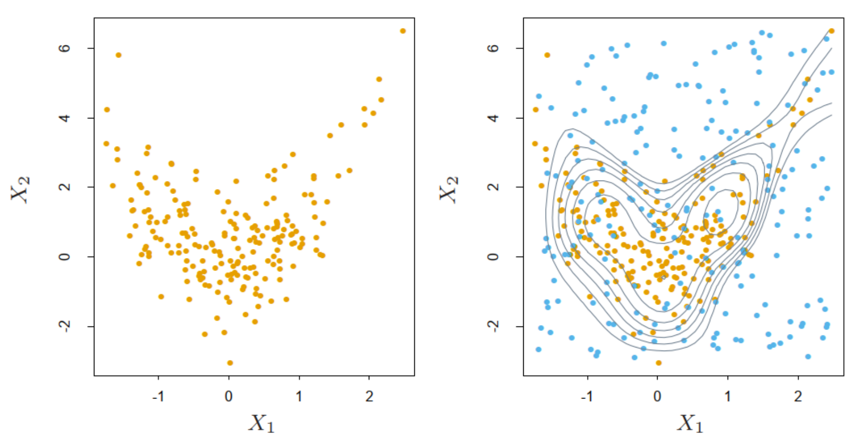
- The `arules` package has functionality to automatically handle categorical and numeric data

```
library(arules)

#-- Load data
data("IncomeESL")    # load income data from arules package
summary(IncomeESL)   # summary of data
#>       income          sex             marital status     age
#>  [0,10) :1745   male  :4075   married      :3334   14-17: 878
#>  [50,75):1308   female:4918   cohabitation: 668   18-24:2129
#>  [30,40):1110                 divorced     : 875   25-34:2249
#>  [40,50): 969                 widowed      : 302   35-44:1615
#>  75+    : 884                 single       :3654   45-54: 922
#>  [20,25): 813                 NA's         : 160   55-64: 640
#>  (Other):2164                                      65+  : 560
#>                  education                        occupation   years in bay area
#>  grade <9             : 264   professional/managerial:2820   <1   : 270
#>  grades 9-11          :1046   student                :1489   1-3 :1042
#>  high school graduate:2041   clerical/service       :1062   4-6 : 686
#>  college (1-3 years) :3066   sales                  : 770   7-10: 900
#>  college graduate    :1524   laborer                : 767   >10 :5182
```

```
#>  graduate study      : 966   (Other)                 :1949   NA's: 913
#>  NA's                :  86   NA's                    : 136
#>       dual incomes   number in household number of children
#>  not married:5438   2       :2664        0       :5724
#>  yes        :2211   3       :1670        1       :1506
#>  no         :1344   1       :1620        2       :1148
#>                     4       :1526        3       : 412
#>                     5       : 686        4       : 117
#>                     (Other): 452         5       :  46
#>                     NA's    : 375        (Other):  40
#>                householder status      type of home   ethnic classification
#>  own                       :3256     house       :5073   white    :5811
#>  rent                      :3670     condominium: 655   hispanic:1231
#>  live with parents/family:1827     apartment   :2373   black    : 910
#>  NA's                      : 240     mobile Home: 151   asian    : 477
#>                                      other       : 384   other    : 225
#>                                      NA's        : 357   (Other) : 271
#>                                                          NA's     :  68
#>  language in home
#>  english:7794
#>  spanish: 579
#>  other  : 261
#>  NA's   : 359
#>
#>
#>

#-- convert to transactions object
trans = as(IncomeESL, "transactions")
```

8.2 Supervised Learning Problem

- The ESL text (ESL 14.2.4 - 14.2.7) outlines an interesting approach for finding frequent itemsets
- The idea is to simulate additional data under an appropriate null distribution (e.g., independence, uniform), combine the simulated data with the observed data, and assign a *label* indicating if the observation comes from the observed or simulated data



- Now any supervised learning procedure can be used to find regions of deviation from the null
  - That is, regions where there are more actual observations than simulated observations
- This concept can be incorporated into other tasks, like density estimation and clustering

8.3 Research Opportunities

- How can "repeatable patterns" be confirmed? I.e., assess statistical significance.
- How to incorporate resampling (bootstrap, cross-validation)?
- How to handle structure in items?
  - e.g., *small oranges, large oranges, bag of oranges* are all oranges
- Detect temporal changes in transaction data
- Find outliers and anomalies in transaction data
- Transaction clustering
- Weighted transactions