



*Association Rule Mining With Tweets: **Thinking Outside the Basket***

Dr. Ami Gates,
Director Analytics
Georgetown



*GEORGETOWN
UNIVERSITY*

Pre - Thank You's

Thank you to Marck Vaisman!

Marck and I teach together at Georgetown and he introduced me to DRC.

Thank you to Jared Lander!

It's a pleasure to be a part of the DCR Conference.



About Me



- 1) **Job:** Director and Associate Professor of Teaching at Georgetown University.
- 2) **Background:** PhD Computer Engineering (focus ML, DA), MS Computer Science, MS Education, BA Math
- 3) **Teaching:** Yes – this is my 29th year in front of a captive audience.
- 4) **Love:** My husband, My Coffee, and My Mountains



What is Association Rule Mining

- 1) **Unsupervised Learning**
- 2) Evaluates “**transactions**” (collections of sets) for correlations/associations.
- 3) Most common example: **Market Basket**
- 4) **Can also apply to :**
 - Image identification
 - Text: like **Twitter data**
 - Any collection of words
 - Click streams
 - Bio data – binding sites, AA’s in proteins, etc.

Quick Review: Examples of association rules

<i>TID</i>	<i>Items</i>
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Introduction to Data Mining, 2nd Edition

$\{\text{Diapers}\} \rightarrow \{\text{Beer}\}$

$\{\text{Milk, Bread}\} \rightarrow \{\text{Coke}\}$

$\{\text{Milk, Bread}\} \rightarrow \{\text{Coke, Diaper}\}$

$\{\text{Diapers}\} \rightarrow \{\text{Beer, Bread}\}$

** Association (like correlation) is a measure of **co-occurrence** NOT causality.

Measures of Set Correlation

Let X and Y be sets and assume rule $X \rightarrow Y$

1) Support:

$$\text{Sup}(X, Y) = P(X, Y)$$

(Count of X and Y together) / (Total # Trans)

2) Confidence:

$$\text{Conf}(X, Y) = P(Y|X) = P(X, Y) / P(X)$$

(Count of X and Y together) / (Count of X)

Lift

Measure of dependent or correlated events: Lift

$$\text{Lift } (A \Rightarrow B) = \text{support}(\{A, B\}) / (\text{support}(A) \times \text{support}(B))$$

$$lift(A \rightarrow B) = \frac{P(A \cap B)}{P(A)P(B)}$$

Association rules should have >1 lift to be meaningful.

Quick Measure Examples

TID	Items
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Given: {Beer} → {Diaper}

Introduction to Data Mining, 2nd Edition

$$\text{Sup}(\{\text{Beer}\}, \{\text{Diaper}\}) = 2/5 = .40 = 40\%$$

$\text{Conf}(\{\text{Beer}\}, \{\text{Diaper}\})$

$$= P(\{\text{Beer}\}, \{\text{Diaper}\}) / P(\{\text{Beer}\})$$

$$= (2/5) / (3/5) = 66.7\%$$

$$\text{Lift } (\{\text{Beer}\}, \{\text{Diaper}\}) = \text{Sup}(\{\text{Beer}\}, \{\text{Diaper}\}) / \\ \text{Sup}(\{\text{Beer}\}) * \text{Sup}(\{\text{Diaper}\}) = (2/5) / (3/5) * (3/5) = 1.11$$

Conceptually

$X \rightarrow Y$

Sup: $P(X \text{ and } Y)$ –

Measure of joint occurrence.

The more X and Y occur together, the higher the Support. Range: 0 to 1.

Conf: $P(X \text{ and } Y) / P(X) = P(Y|X)$ –

Measure of joint occurrence assuming X

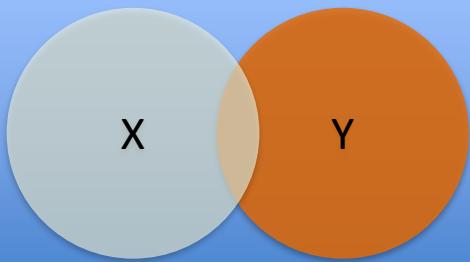
As $P(X)$ increases, conf $X \rightarrow Y$ decreases. Range: 0 to 1.

Lift: $P(X \text{ and } Y) / P(X)P(Y)$ –

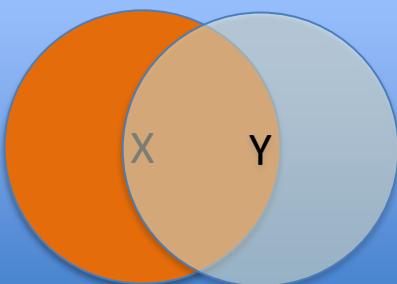
= 1 when X and Y are independent.

< 1 when X and Y have little or no intersection.

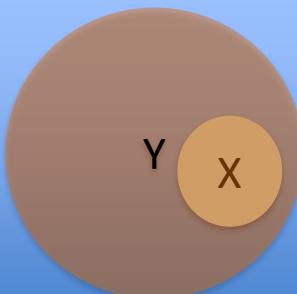
> 1 when X and Y have an intersection larger than their probability product.



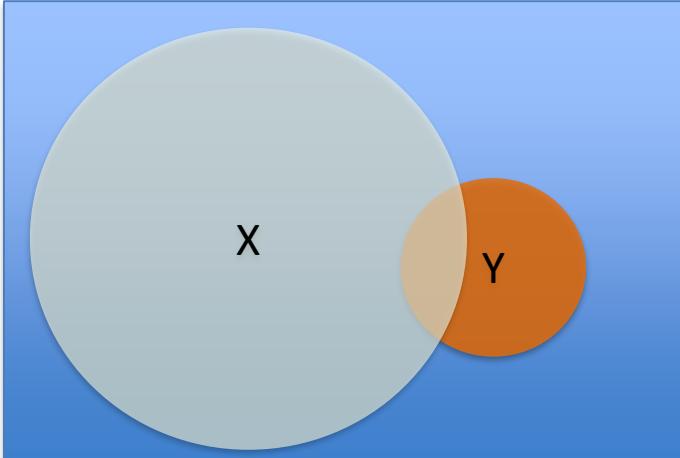
Support: $P(X \text{ and } Y) = \text{low}$
 Conf: $P(X \text{ and } Y)/P(X) = \text{higher than Sup, but still low}$
 Lift: $P(X \text{ and } Y)/ P(X)*P(Y) < 1$



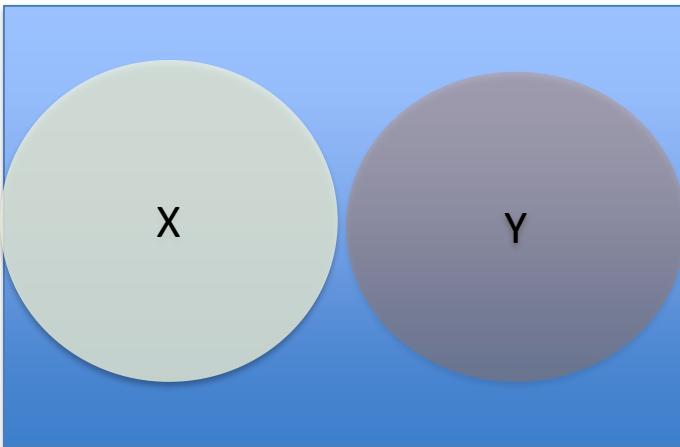
Support: $P(X \text{ and } Y) = \text{high}$
 Conf: $P(X \text{ and } Y)/P(X) = \text{high and higher than Sup}$
 Lift: $P(X \text{ and } Y)/ P(X)*P(Y) > 1$ Interesting



Support: $P(X \text{ and } Y) = P(X) = \text{small} - \text{based on X}$
 Conf: $P(X \text{ and } Y)/P(X) = 1$ (highest possible)
 Lift: $P(X \text{ and } Y)/ P(X)*P(Y) = 1/P(Y) > 1$
 Interesting because X only occurs if Y



Support: $P(X \text{ and } Y) = \text{low}$
Conf: $P(X \text{ and } Y)/P(X) = \text{higher than Sup, but still low}$
Lift: $P(X \text{ and } Y)/ P(X)*P(Y) < 1$

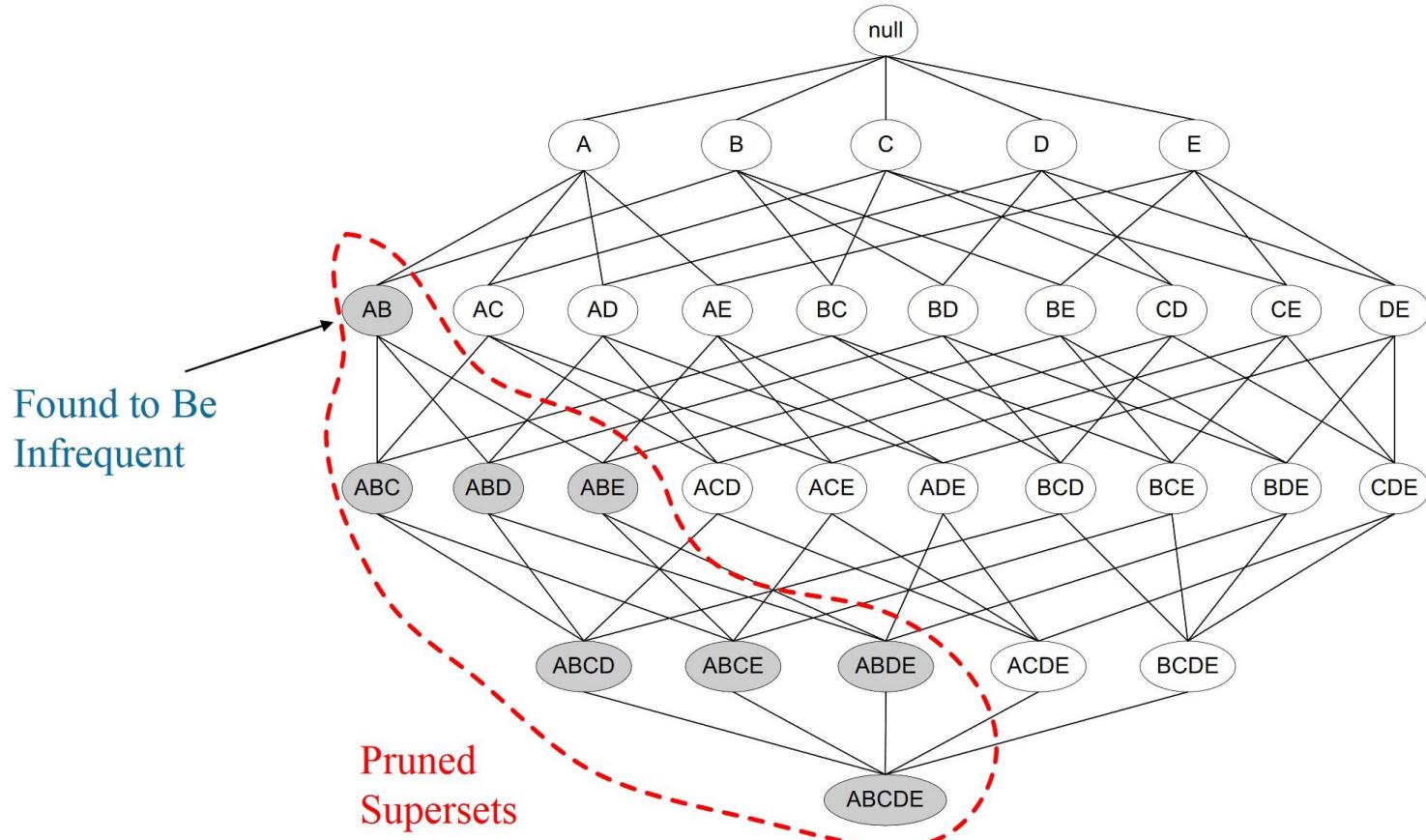


Support: $P(X \text{ and } Y) = 0$
Conf: $P(X \text{ and } Y)/P(X) = 0$
Lift: $P(X \text{ and } Y)/ P(X)*P(Y) = 0$

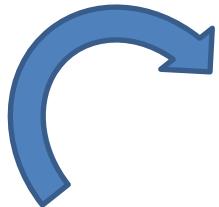


Support: $P(X \text{ and } Y) = \text{low}$
Conf: $P(X \text{ and } Y)/P(X) = \text{high}$
Lift: $P(X \text{ and } Y)/ P(X)*P(Y) = \text{very high}$

Quick Reminder: The apriori algorithm



Other Ways to Represent Transaction Data



TID	Items
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Introduction to Data Mining, 2nd Edition

1	Bread
1	Coke
1	Milk
2	Beer
2	Bread
3	Beer
3	Coke
3	Diaper
3	Milk
4	Beer
4	Bread
4	Diaper
4	Milk
5	Coke
5	Diaper
5	Milk



TID	Bread	Coke	Milk	Beer	Diaper
1	1		1	1	0
2	1		0	0	1
3	0		1	1	1
4	1		0	1	1
5	0		1	1	0

quinoa	soymilk	coffee	chocloate
quinoa	soymilk	kale	tea
quinoa	kale		
quinoa	soymilk	coffee	chocloate
quinoa	soymilk	carrot	tea
quinoa	kale		
quinoa	soymilk	coffee	chocloate
quinoa	soymilk	kale	tea
quinoa	carrot		
quinoa	soymilk	coffee	chocloate
quinoa	soymilk	kale	tea
quinoa	carrot		
quinoa	soymilk	coffee	chocloate
quinoa	soymilk		carrot
quinoa	soymilk		tea
quinoa	kale		
quinoa	soymilk	coffee	chocloate
quinoa	soymilk	carrot	
quinoa	carrot		
quinoa	soymilk	coffee	chocloate
quinoa	soymilk		

Transaction Data

Notice: It is not necessary to have a numbered transaction ID

Basic ARM R Code

```
library(arules)

Foods <- read.transactions("HealthyBasketData.csv",
                           rm.duplicates = FALSE,
                           format = "basket",
                           sep=",",
                           cols=NULL)
inspect(Foods)

rules <- arules::apriori(Foods, parameter = list(support=.2,
                                                   confidence=.2, minlen=2))
inspect(rules)

SortedRules <- sort(rules, by="confidence", decreasing=TRUE)
inspect(SortedRules[1:10])

SortedRulesL <- sort(rules, by="lift", decreasing=TRUE)
inspect(SortedRulesL[1:10])
```

```

> SortedRules <- sort(rules, by="confidence", decreasing=TRUE)
> inspect(SortedRules[1:10])
      lhs          rhs    support confidence   lift    count
[1] {kale}      => {quinoa} 0.30       1 1.000000 6
[2] {tea}        => {soymilk} 0.25       1 1.428571 5
[3] {tea}        => {quinoa} 0.25       1 1.000000 5
[4] {carrot}     => {quinoa} 0.35       1 1.000000 7
[5] {coffee}     => {chocloate} 0.35       1 2.857143 7
[6] {chocloate}  => {coffee}   0.35       1 2.857143 7
[7] {coffee}     => {soymilk}  0.35       1 1.428571 7
[8] {coffee}     => {quinoa}   0.35       1 1.000000 7
[9] {chocloate}  => {soymilk}  0.35       1 1.428571 7
[10] {chocloate} => {quinoa}   0.35       1 1.000000 7
>
> SortedRulesL <- sort(rules, by="lift", decreasing=TRUE)
> inspect(SortedRulesL[1:10])
      lhs          rhs    support confidence   lift    count
[1] {coffee}     => {chocloate} 0.35 1.0000000 2.857143 7
[2] {chocloate}  => {coffee}   0.35 1.0000000 2.857143 7
[3] {coffee, soymilk} => {chocloate} 0.35 1.0000000 2.857143 7
[4] {chocloate, soymilk} => {coffee}   0.35 1.0000000 2.857143 7
[5] {coffee, quinoa}  => {chocloate} 0.35 1.0000000 2.857143 7
[6] {chocloate, quinoa} => {coffee}   0.35 1.0000000 2.857143 7
[7] {coffee, quinoa, soymilk} => {chocloate} 0.35 1.0000000 2.857143 7
[8] {chocloate, quinoa, soymilk} => {coffee}   0.35 1.0000000 2.857143 7
[9] {tea}         => {soymilk}  0.25 1.0000000 1.428571 5
[10] {soymilk}    => {tea}      0.25 0.3571429 1.428571 5

```

Read Two Common Formats

```
Foods <- read.transactions("KumarGroceriesTransData.csv",
  rm.duplicates = FALSE,
  format = "single", ##or basket
  sep=",",
  skip=0,
  cols=c(1,2) ## for single, 1 ID col , 2 is item
  ## default is NULL for basket. Null means no IDs
)
arules::inspect(Foods)
```

```
Foods2 <- read.transactions("KumarGroceriesTransData_ASTRANS.csv",
  rm.duplicates = FALSE,
  format = "basket",
  sep=",",
  cols=1 ##ID in col 1 if no ID then cols=NULL
)
arules::inspect(Foods2)
```



Thinking Outside the Basket

Twitter Data

- 1) **Do not want a “bag of words” or a table of word frequencies.**
- 2) Will need to create a **“set of transactions”** – one for each Tweet.
- 3) Items in the transactions will be words.

R Association Rules and Twitter: libraries

```
library(arules)
library(rtweet)
library(twitteR)
library(ROAuth)
library(jsonlite)
#library(streamR)
library(rjson)
library(tokenizers)
library(tidyverse)
library(plyr)
library(dplyr)
library(ggplot2)
#install.packages("syuzhet")
## sentiment analysis
library(syuzhet)
library(stringr)
library(arulesViz) ## load last
```

Trouble with arulesViz?

```
## FIRST - you MUST register and log into github
## install_github("mhahsler/arulesViz")
## RE: https://github.com/mhahsler/arulesViz
```

Set Up Twitter Dev Account First

<https://developer.twitter.com/>

The screenshot shows the Twitter Developer website interface. At the top, there is a purple navigation bar with links for 'Developer', 'Use cases', 'Products', 'Docs', 'More', 'Dashboard', and a user profile for 'DrGates309'. A 'Create an app' button is located in the top right corner. Below the navigation bar, the page title 'Apps' is displayed. A card for an application named 'GatesTwitterMining' is shown, featuring a blue Twitter logo icon, the app name, and its App ID '135'. There are 'Details' and more options buttons at the bottom right of the card.



Developer

Use cases

Products

Docs

More

Apps / GatesTwitterMining

[App details](#)[Keys and tokens](#)[Permissions](#)

App details

Details and URLs



App icon

App icon is default, click

App Name

GatesTwitterMining

Description

Twitter Data Mining for Education

Apps / GatesTwitterMining

[App details](#)[Keys and tokens](#)[Permissions](#)

Keys and tokens

Keys, secret keys and access tokens management.

Consumer API keys

mnDC09[REDACTED] (API key)

qzwDO9[REDACTED] (API secret key)

[Regenerate](#)

Access token & access token secret

838558602[REDACTED] (Access token)

hswxbxErm[REDACTED] (Access token secret)

Read and write (Access level)

R Twitter Options

```
##### Using twitteR #####
setup_twitter_oauth(consumerKey, consumerSecret, access_Token, access_Secret)

Search<-twitteR::searchTwitter("#ILoveChocolate", n=100, since="2018-09-09")
(Search_DF <- twListToDF(Search))
TransactionTweetsFile = "Choc.csv"
```

	text					
1	The other day I woke up craving chocolate cupcakes. Today I'm craving @HersheyCompany chocolate bars. think the u... https://t.co/NtGH4eaSRc					
2	WHO SAID "CHOCOLATE"?\\n________________________________\\n#feed #feedsmartfood #honey #weovechocolate... https://t.co/DzzmvJlKEh					
3	@ClaireValy @LowngSnake @firebox #ILOVECHOCOATE\\nI love Chocolate very very much.					
4	#HealthTips #momlife #sahmlife #toddlers #ilovechocolate #homeschoolmom #bethechange #oingitformygirls #fitmom #feeltheburn					
5	RT @Kelly_Hawrylysh: #Fairtrade sourcing needed more than ever to avoid chocapocalypse!!! https://t.o/dbxw3eQfTc #SDG12 @FairtradeAfrica...					
6	RT @Kelly_Hawrylysh: #Fairtrade sourcing needed more than ever to avoid chocapocalypse!!! https://t.o/dbxw3eQfTc #SDG12 @FairtradeAfrica					
	favorited	favoriteCount	replyToSN	created	truncated	replyToSID
1	FALSE	0	<NA>	2018-09-27 12:12:52	TRUE	<NA>
2	FALSE	0	<NA>	2018-09-27 10:51:42	TRUE	<NA>
3	FALSE	0	ClaireValy	2018-09-27 00:45:43	FALSE	1044897146326208513
4	FALSE	0	templin_katie	2018-09-26 19:49:55	FALSE	1045037612388536321
5	FALSE	0	<NA>	2018-09-26 16:24:22	FALSE	<NA>
6	FALSE	0	<NA>	2018-09-26 16:23:42	FALSE	<NA>
	id	replyToUID				
1	1045285140505735169	<NA>				
2	1045264712118734848	<NA>				
3	1045112213915226113	2878148959				
4	1045037771050618881	1035584652722036736				
5	1044986045975220224	<NA>				
6	1044985877456392194	<NA>				
	statusSource					
1	Twitter for Android					
2	Instagram					
3	Twitter Web Client					
4	Twitter for iPhone					
5	Twitter for Android					
6	Twitter for Android					
	screenName	retweetCount	isRetweet	retweeted	longitude	latitude
1	RachelTBue	0	FALSE	FALSE	<NA>	<NA>
2	Niklaus_R	0	FALSE	FALSE	4.35008	50.845
3	saminaseem16	0	FALSE	FALSE	<NA>	<NA>

Build the Transaction File: Step 1

- 1) Each tweet should be one transaction.
- 2) Each word (token) in the tweet should be in its own column.

```
> (Search_DF$text[1])
[1] "The other day I woke up craving chocolate cupcakes. Today I'm craving @HersheyCompany chocolate bars. I
think the u... https://t.co/NtGH4eaSRC"
```

Build The Transaction File: Step 2

```
## Start the file
Trans <- file(TransactionTweetsFile)
## Tokenize to words
Tokens<-tokenizers::tokenize_words(Search_DF$text[1],stopwords = stopwords::stopwords("en"),
    lowercase = TRUE, strip_punct = TRUE, strip_numeric = TRUE,simplify = TRUE)
## Write squished tokens
cat(unlist(str_squish(Tokens)), "\n", file=Trans, sep=",")
close(Trans)

## Append remaining lists of tokens into file
## Recall - a list of tokens is the set of words from a Tweet
Trans <- file(TransactionTweetsFile, open = "a")
for(i in 2:nrow(Search_DF)){
  Tokens<-tokenize_words(Search_DF$text[i],stopwords = stopwords::stopwords("en"),
    lowercase = TRUE, strip_punct = TRUE, simplify = TRUE)
  cat(unlist(str_squish(Tokens)), "\n", file=Trans, sep=",")
}
close(Trans)
```

Transaction File: Each Row is a Tweet

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	day	woke	craving	chocolate	cupcakes	today	craving	hersheyco	chocolate	bars	think	u	https	t.co	ntgh4easrc
2	said	chocolate	feed	feedsmart	honey	welovechc	https	t.co	dzzmvjlkeh						
3	clairevaly	lowngsnak	firebox	ilovechocc	love	chocolate	much								
4	healthtips	momlife	sahmlife	toddlers	ilovechocc	homescho	bethechar	doingitfor	fitmom	feeltheburn					
5	rt	kelly_haw	fairtrade	sourcing	needed	ever	avoid	chocapoca	https	t.co	dbxw3eqftsdg12				fairtradeafrica
6	rt	kelly_haw	fairtrade	sourcing	needed	ever	avoid	chocapoca	https	t.co	dbxw3eqftsdg12				fairtradeafrica
7	cada	día	estamos	mas	listos	para	navidad	taza		3 pack	de	venta	en	cityclub	navidad
8	fairtrade	sourcing	needed	ever	avoid	chocapoca	https	t.co	dbxw3eqftsdg12	https	t.co				rgmtaombom
9	ilovechocc	chocolate	adictaalch	https	t.co	kpzofu8ix2									
10	see	big	chocolate	show	saturday	night	ilovechocolate								
11	else	can	say	thehousec	braziliantr	truffles	brigadeiro	desserts	https	t.co	pzayia63ir				
12	touch	cocoa	please	ilovechocc	bless	https	t.co	vx7v7csfr5							
13	bako_nw	weekendb	choc	dome	hiding	double	chocolate	cheesecak	ilovechocc	https	t.co	f2ginuvtfq			
14	ilovechocolate														
15	los	lunes	lucen	tan	malos	si	los	ves	con	la	actitud	correcta	chocolate	iniciodes	felizlunes
16	enough	words	express	thankful	amazing	coworkers	thank	ccriheathe	onl	https	t.co	2gljgtudhh			
17	casa	ino	nostra	przedstaw	hotel	hotelwgór	taty	podhale	zakopane	nowytarg	deser	slodycz	suflet	https	t.co
18	rt	ccfchocola	crunchy	biscuit	dipped	chocolate	foodporn	yummy	sweets	love	instafood	food	delicious	choco	dessert
19	crunchy	biscuit	dipped	chocolate	foodporn	yummy	sweets	love	instafood	food	delicious	choco	dessert	https	t.co
20	bbcmiami:	light	ilovechocolate												

Read and Inspect the Transactions

```
##### Read in the tweet transactions
TweetTrans <- read.transactions(TransactionTweetsFile,
                                rm.duplicates = FALSE,
                                format = "basket",
                                sep=","
                                ## cols =
                                )
inspect(TweetTrans)
## See the words that occur the most
sample_Trans <- sample(TweetTrans, 50)
summary(Sample_Trans)
```

most frequent items:

https	t.co	chocolate	ilovechocolate	rt
35	35	25	23	9

```
[59] {1,  
    along,  
    box,  
    chocolates,  
    days,  
    domme,  
    findom,  
    finsub,  
    godiva,  
    ilovechocolate,  
    pay,  
    send}
```

```
[60] {chocolate,  
    delicious,  
    food,  
    foodporn,  
    https,  
    instafood,  
    introducing,  
    love,  
    mango,  
    marzipan,  
    sweets,  
    t.co,  
    truffles,  
    u17wpqhhxh,  
    yummy}
```

Transaction Sets and Summary

Clean Up

```
## Read the transactions data into a dataframe  
TweetDF <- read.csv(TransactionTweetsFile, header = FALSE, sep = ",")  
head(TweetDF)
```

```
> TweetDF <- read.csv(TransactionTweetsFile, header = FALSE, sep = ",")  
> head(TweetDF)  
      v1          v2          v3          v4          v5  
1    day        woke  craving  chocolate cupcakes  
2   said  chocolate _____ feed  feedsmartfood  
3 clairevaly  lowngsnake  firebox ilovechocolate love  
4 healhtips     momlife sahmlife toddlers ilovechocolate  
5           rt kelly_hawrylysh fairtrade  sourcing needed  
6           rt kelly_hawrylysh fairtrade  sourcing needed  
      v6          v7          v8          v9          v10         v11         v12          v13  
1   today  craving  hersheycompany chocolate      bars   think       u      https  
2   honey welovechocolate          https      t.co dzzmvjlkeh  
3  chocolate      much          v10         v11         v12          v13  
4 homeschoolmom bethechange doingitformygirls fitmom feeltheburn  
5   ever      avoid  chocapocalypse          https      t.co dbxw3eqftc sdg12 fairtradeafrica  
6   ever      avoid  chocapocalypse          https      t.co dbxw3eqftc sdg12 fairtradeafrica
```

most frequent items:

https	t.co	chocolate	ilovechocolate	rt
35	35	25	23	9

Specifically Remove Words

```
## Convert all columns to char
TweetDF<-TweetDF %>%
  mutate_all(as.character)
(str(TweetDF))
# We can now remove certain words
TweetDF[TweetDF == "t.co"] <- ""
TweetDF[TweetDF == "rt"] <- ""
TweetDF[TweetDF == "http"] <- ""
TweetDF[TweetDF == "https"] <- ""

## Clean with grep1 - every row in each column
MyDF<-NULL
for (i in 1:ncol(TweetDF)){
  MyList=c() # each list is a column of logicals ...
  MyList=c(MyList,grep1("[[:digit:]]", TweetDF[[i]]))
  MyDF<-cbind(MyDF,MyList) ## create a logical DF
  ## TRUE is when a cell has a word that contains digits
}
## For all TRUE, replace with blank
TweetDF[MyDF] <- ""
(TweetDF)
```

Our Transactions

```
> (head(TweetDF,10))
```

	v1	v2	v3	v4	v5
1	day	woke	craving	chocolate	cupcakes
2	said	chocolate		feed	feedsmartfood
3	clairevaly	lowngsnake	firebox	ilovechocolate	love
4	healthtips	momlife	sahmlife	toddlers	ilovechocolate
5		kelly_hawrylysh	fairtrade	sourcing	needed
6		kelly_hawrylysh	fairtrade	sourcing	needed
7	cada	día	estamos	mas	listos
8					
9	fairtrade	sourcing	needed	ever	avoid
10	ilovechocolate	chocolate	adictaalchocolate		
	v6	v7	v8	v9	v13
1	today	craving	hersheycompany	chocolate	bars think u
2	honey	welovechocolate			dzzmvjlkkeh
3	chocolate	much			
4	homeschoolmom	bethechange	doingitformygirls	fitmom	feeltheburn
5	ever	avoid	chocapocalypse		fairtradeafrica
6	ever	avoid	chocapocalypse		fairtradeafrica
7	para	navidad	taza	pack	de venta en
8					
9	chocapocalypse				rgmtaombom
10					
	v14	v15	v16		

```
# Now we save the dataframe using the write.table command
write.table(TweetDF, file = "UpdatedChocolate.csv", col.names = FALSE,
            row.names = FALSE, sep = ",")
TweetTrans <- read.transactions("UpdatedChocolate.csv", sep = ",",
                                format("basket"), rm.duplicates = TRUE)
inspect(TweetTrans)
```

Association Rule Mining

```
[70] {chocolate,  
delicious,  
food,  
foodporn,  
instafood,  
introducing,  
love,  
mango,  
marzipan,  
sweets,  
truffles,  
yummy}  
[71] {bali's,  
big,  
check,  
chocolatiers,  
ilovechocolate,  
six,  
theyakmag,  
theyakmagazine,  
yak}
```



Example cleaned tweets as individual transactions.

```
TweetTrans_rules = arules::apriori(TweetTrans,  
parameter = list(support=.01, confidence=.01, minlen=2))  
inspect(TweetTrans_rules[1:10])  
## sorted  
SortedRules_conf <- sort(TweetTrans_rules, by="confidence", decreasing=TRUE)  
inspect(SortedRules_conf[1:15])  
  
SortedRules_sup <- sort(TweetTrans_rules, by="support", decreasing=TRUE)  
inspect(SortedRules_sup[1:15])
```

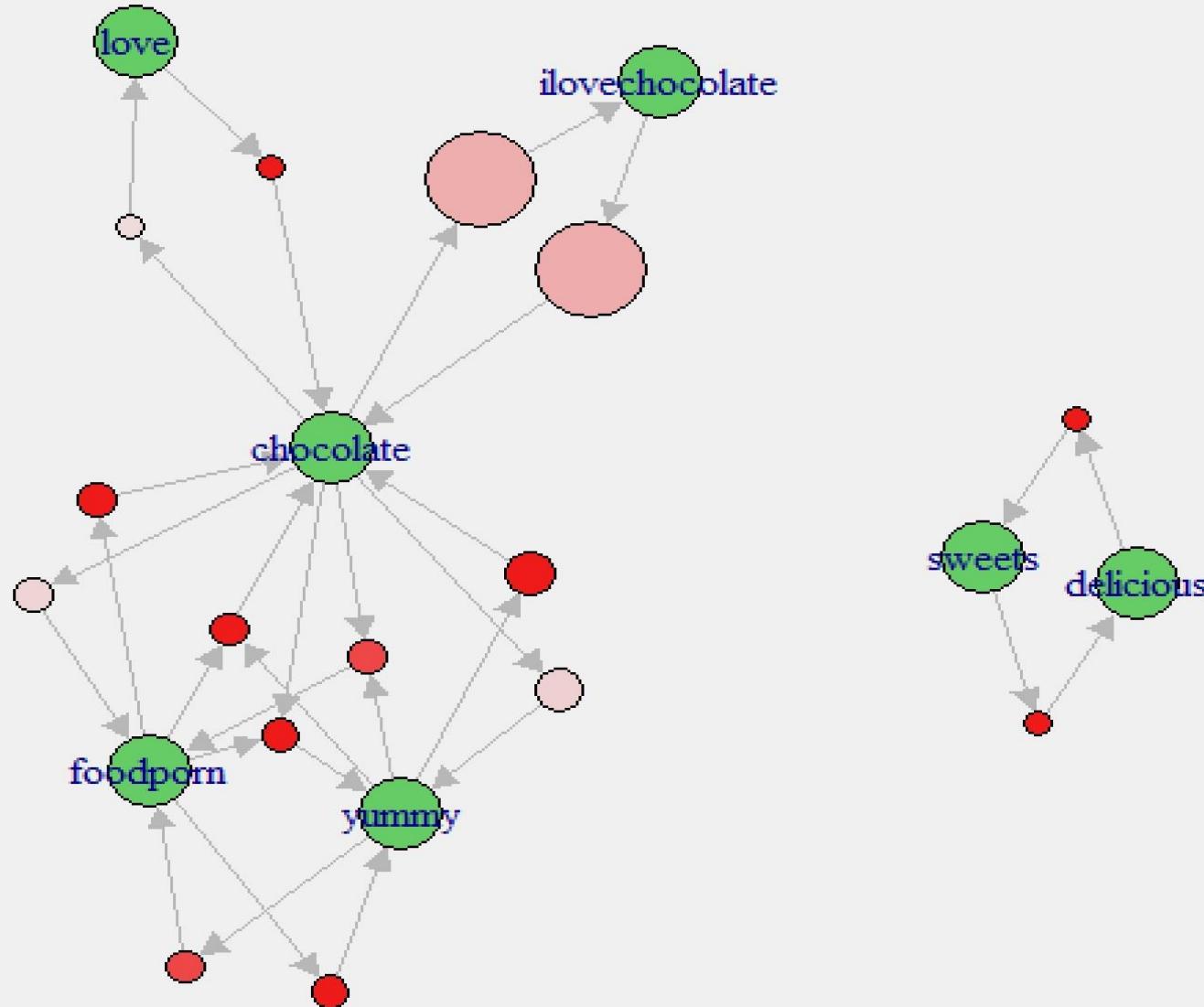
```

> SortedRules_conf <- sort(TweetTrans_rules, by="confidence", decreasing=TRUE)
> inspect(SortedRules_conf[1:15])
    lhs                      rhs          support  confidence   lift   count
[1] {light}                  => {ilovechocolate} 0.01388889 1        2.571429 1
[2] {bvlgariilcioccolato} => {ilovechocolate} 0.01388889 1        2.571429 1
[3] {bvlgariilcioccolato} => {chocolate}      0.01388889 1        2.482759 1
[4] {mourespi}               => {foramorango}   0.01388889 1       72.000000 1
[5] {foramorango}            => {mourespi}     0.01388889 1       72.000000 1
[6] {mourespi}               => {ilovechocolate} 0.01388889 1        2.571429 1
[7] {foramorango}            => {ilovechocolate} 0.01388889 1        2.571429 1
[8] {adictaalchocolate}     => {ilovechocolate} 0.01388889 1        2.571429 1
[9] {adictaalchocolate}     => {chocolate}      0.01388889 1        2.482759 1
[10] {free}                  => {sugar}        0.01388889 1       72.000000 1
[11] {sugar}                 => {free}         0.01388889 1       72.000000 1
[12] {free}                  => {days}         0.01388889 1       36.000000 1
[13] {free}                  => {ilovechocolate} 0.01388889 1        2.571429 1
[14] {free}                  => {chocolate}     0.01388889 1        2.482759 1
[15] {sugar}                 => {days}         0.01388889 1       36.000000 1
>
> SortedRules_sup <- sort(TweetTrans_rules, by="support", decreasing=TRUE)
> inspect(SortedRules_sup[1:15])
    lhs                      rhs          support  confidence   lift   count
[1] {ilovechocolate}        => {chocolate}    0.18055556 0.4642857 1.152709 13
[2] {chocolate}              => {ilovechocolate} 0.18055556 0.4482759 1.152709 13
[3] {yummy}                 => {chocolate}    0.09722222 1.0000000 2.482759  7
[4] {chocolate}              => {yummy}        0.09722222 0.2413793 2.482759  7
[5] {foodporn}               => {yummy}        0.08333333 1.0000000 10.285714  6
[6] {yummy}                  => {foodporn}    0.08333333 0.8571429 10.285714  6
[7] {foodporn}               => {chocolate}   0.08333333 1.0000000 2.482759  6
[8] {chocolate}              => {foodporn}    0.08333333 0.2068966 2.482759  6
[9] {foodporn,yummy}         => {chocolate}   0.08333333 1.0000000 2.482759  6
[10] {chocolate,foodporn}    => {yummy}        0.08333333 1.0000000 10.285714  6
[11] {chocolate,yummy}       => {foodporn}    0.08333333 0.8571429 10.285714  6
[12] {love}                  => {chocolate}   0.06944444 1.0000000 2.482759  5
[13] {chocolate}              => {love}         0.06944444 0.1724138 2.482759  5
[14] {sweets}                => {delicious}   0.06944444 1.0000000 14.400000  5
[15] {delicious}             => {sweets}       0.06944444 1.0000000 14.400000  5

```

A Quick Plot

```
plot (SortedRules_sup[1:15],method="graph",interactive=TRUE,shading="confidence")
```



Looking at More Rules

```
plot (SortedRules_sup[1:50],method="graph",interactive=TRUE,shading="confidence")
plot (SortedRules_conf[1:50],method="graph",interactive=TRUE,shading="confidence")
```

