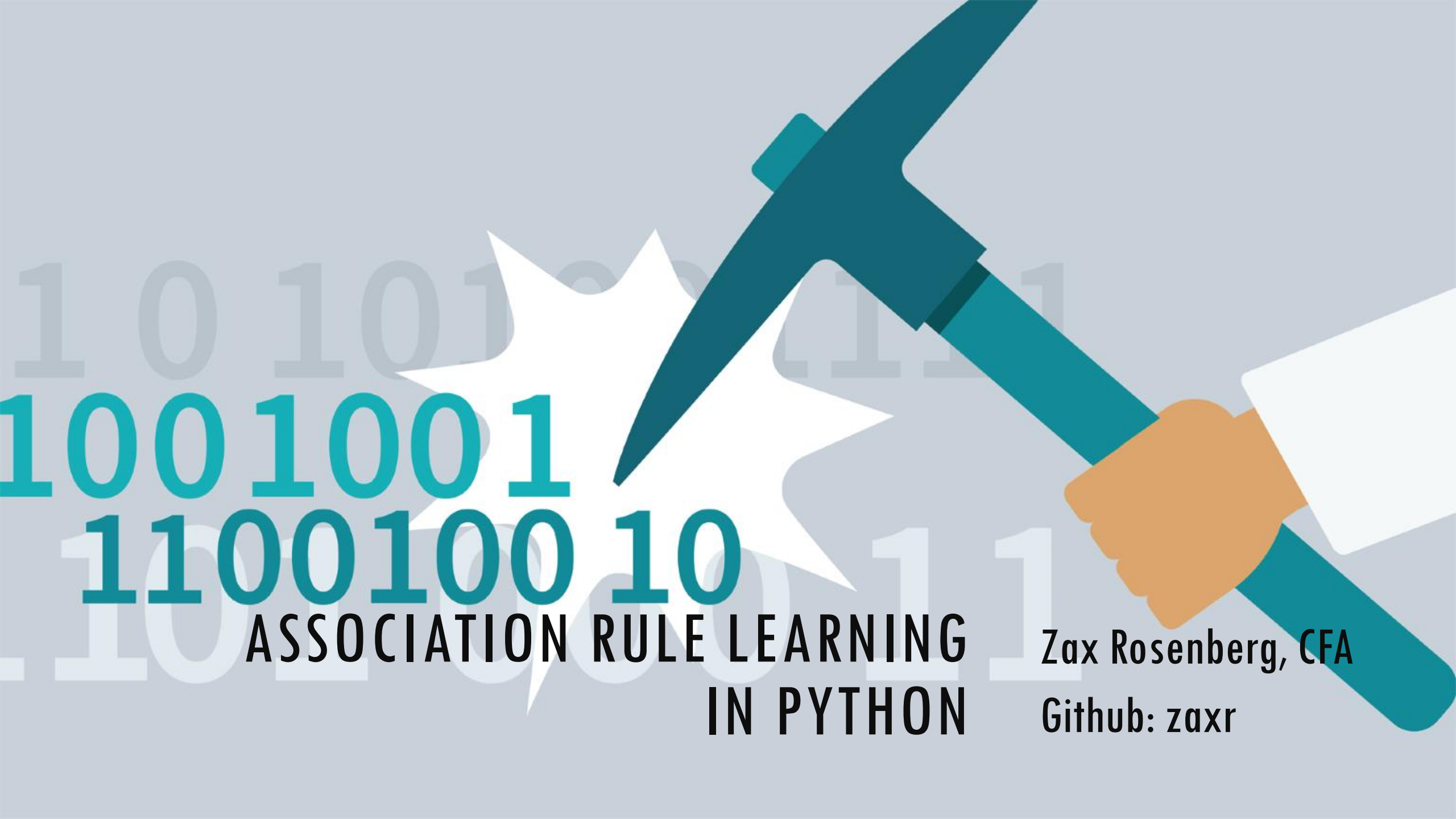


ASSOCIATION RULE LEARNING IN PYTHON

Zax Rosenberg, CFA

Github: zaxr

WHAT WE'LL BE TALKING ABOUT

1. **What is association rule learning?**
2. **What can we do with it?**
3. **How do we use it…**
   - Walk through a traditional market basket example
4. **…efficiently…**
   - Introductions to Apriori and FP-growth algorithms
5. **…and in some advanced ways?**
6. **A real world example**
   - Identifying IBD-safe foods

ASSOCIATION RULE LEARNING - DEFINED

A rule-based machine learning (data mining) method for discovering interesting patterns between variables in large databases, in a human-understandable way. Two steps:

1. **Frequent Itemset Mining (FIM).** Find all frequent subsets of items (itemsets), generally as measured by a Support threshold.
2. [**Association**] **Rule Generation.** Generate "interesting" rules, commonly as measured by Confidence and Lift.

EXAMPLE RULE

It can help to think of them like IF … THEN statements (though that's not technically correct). They help identify (not necessarily predict) the occurrence of an item[set] based on the occurrences of other items in the transaction.

Market-Basket Transactions

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

Example of Association Rules

{Diaper} → {Beer}
{Milk, Bread} → {Eggs, Coke}
{Beer, Bread} → {Milk}

Important note: This relationship implies co-occurrence, not causality!

WHICH RULES ARE IMPORTANT?

Generally,

- Ones that occur frequently.
- Ones that are “interesting” based on a predetermined measure.

Is frequency really necessary?

- Otherwise it might happen by chance.
- Probably uninteresting from a business perspective.

But what about…

- analyzing rare combinations?
- identifying early trends?

- We’ll touch on these questions when we discuss advanced topics.

WHAT IS GOOD FOR? ABSOLUTELY…LOTS.

Market-basket analysis,

Web mining,

Document analysis,

Telecommunication alarm diagnosis,

Network intrusion detection,

Bioinformatics

Example #1: In 2004, Walmart mined their retail transactions to see what people in Florida buy prior to the expected arrival of a Hurricane.

Example #2: NASA intern identified predictive patterns for geomagnetic events on earth from characteristics of solar storms on the sun.

Anything where you want to find frequent relationships in data.

WALKING THROUGH AN EXAMPLE

1. **We'll walk through a market-basket problem,** while learning terms.
2. **Walk through the steps.** Internalize the distinct steps now – it will help when we discuss different algorithms and efficiency.
   - Data preparation
   - Frequent Itemset Mining (FIM)
   - Rule Generation
3. **Build our way to efficiency**

BASIC TERMINOLOGY

Transaction. A complete record, made up of underlying items (e.g. a customer's purchase in a single visit).

Itemset. A subset of items within a transaction (e.g. {Eggs, Coke} from {Milk, Bread, Eggs, Coke}).

Antecedent. The left-hand side of a rule (e.g. {Milk, Bread} in the rule {Milk, Bread} $\rightarrow$ {Eggs,Coke}).

Consequent. The right-hand side of a rule (e.g. {Milk, Bread} in the rule {Milk, Bread} $\rightarrow$ {Eggs,Coke}).

DATA PREPARATION

Binarize the data. Is an item in our "basket" or not?

- Often a list of lists in Python, if data fits in memory
- Alternatively, preprocessing.binarize in scikit-learn

Generally **horizontal layout,** but can be vertical or compressed

Horizontal Data Layout

TID	Items
1	A,B,E
2	B,C,D
3	C,E
4	A,C,D
5	A,B,C,D
6	A,E
7	A,B
8	A,B,C
9	A,C,D
10	B

Vertical Data Layout

A	B	C	D	E
1	1	2	2	1
4	2	3	4	3
5	5	4	5	6
6	7	8	9	
7	8	9		
8	10			
9				

FREQUENT ITEMSET MINING - SUPPORT

What does it mean for an itemset to be "frequent?": having a support that meets a minimum threshold parameter ("minsup").

Support: How often an itemset appears as subset of any transaction. It can be given as an absolute amount or (more often) as a percentage.

- **Absolute Support** of {Milk, Beer, Diaper} = 2
- **Relative Support** of {Milk, Beer, Diaper} = 2/5

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

RULE GENERATION

From frequent itemsets, we want to generate representative rules.

Split each frequent itemset into all combinations of antedecent → consequent…

- {Milk, Beer, Diaper}:
  - {Milk} → {Beer}
  - {Milk} → {Diaper}
  - {Milk} → {Beer, Diaper}
  - {Beer} → {Milk}

    …

…that meet a threshold for the chosen measure of interestingness.

INTERESTINGNESS MEASURES

Measure the strength of association rules

- The most common are Confidence and Lift, but there are a ton of alternatives:

Added Value	Descriptive Confirmed Confidence	J-Measure	Mutual Information
All-confidence	Difference of Confidence	Kappa	Odds Ratio
Casual Confidence	Example and Counter-Example Rate	Klosgen	Phi Correlation Coefficient
Casual Support	Fisher's Exact Test	Kulczynski	Ralambrodrainy Measure
Certainty Factor	Gini Index	Goodman-Kruskal Lambda	Relative Linkage Disequilibrium
Chi-Squared	Hyper-Confidence	Laplace Corrected Confidence	Rule Power Factor
Cross-Support Ratio	Hyper-Lift	Least Contradiction	Sebag-Schoenauer Measure
Collective Strength	Imbalance Ratio	Lerman Similarity	Varying Rates Liaison
Conviction	Improvement	Leverage	Yule's Q
Cosine	Jaccard Coefficient	MaxConf	Yule's Y
Coverage			

INTERESTINGNESS MEASURES - CONFIDENCE

Confidence: How frequently items in the consequent appear in transactions that contain the antecedent. Only the denominator changes versus Support.

Example:

- Support of {Milk, Beer, Diapers} is 2
- For the rule {Milk} → {Beer, Diapers} the support of the antecedent is 4
- The rule's Confidence 2/4 = 0.50

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

Why use Confidence?

- Measures the reliability of the inference made by a rule
- Also provides a useful/easy to interpret estimate of conditional probability. Given that a basket has milk, we're 50% sure there's also beer and diapers.

Problem: Can misrepresent importance, since it doesn't consider consequent frequency. I.e. is the confidence greater than that in the general population?

INTERESTINGNESS MEASURES - LIFT

Lift: Measures how much more often X and Y occur together than expected if they were statistically independent. A value of 1 indicates independence.

Example

- Support of {Milk, Beer, Diapers} is 2
- For the rule {Milk} → {Beer, Diapers} the support of the antecedent is 4
- The rule's Confidence 2/4 = 0.50
- The support of the consequent is 3/5 = 0.60
- The rule's Lift is 0.50 / 0.60 = 0.833

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

{Beer, Diapers} already appears in 60% of our transactions, but our rule says we're only 50% sure, meaning we're actually less confident than we'd expect.

THE NAIVE WAY - SET-UP

1. **List all possible association rules**
   - Good luck with that – the number of possible rules is $3^d - 2^{d+1} + 1$ where d is the number of items in the dataset.

2. **Compute the support and confidence for each rule**

3. **Prune rules that fail the *minsup* and *minconf* thresholds**

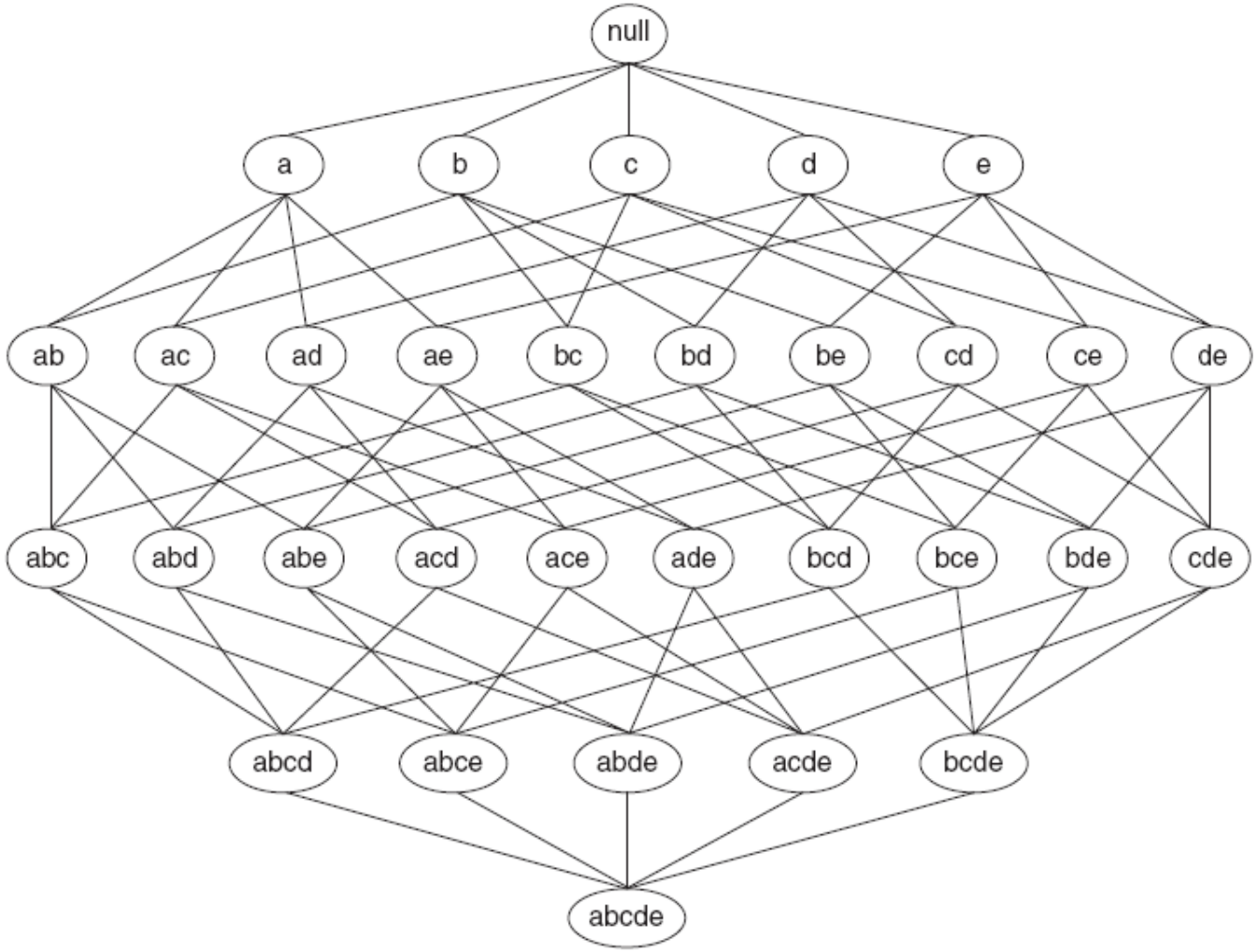
THE NAIVE WAY - FIM

Even every possible itemset combination is a LOT…

…in fact, this **brute force method is still exponential** (2^n-1).

With just 40 items, there are 1.1 **trillion** possible itemsets!

There has to be a better way…



ENTER THE APRIORI RULE

Seminal paper from 1993 by Agrawal, R.; Imielinski, T.; and Swami, A. Mining Association Rules between Sets of Items in Large Databases. Improved from 1994-1998.

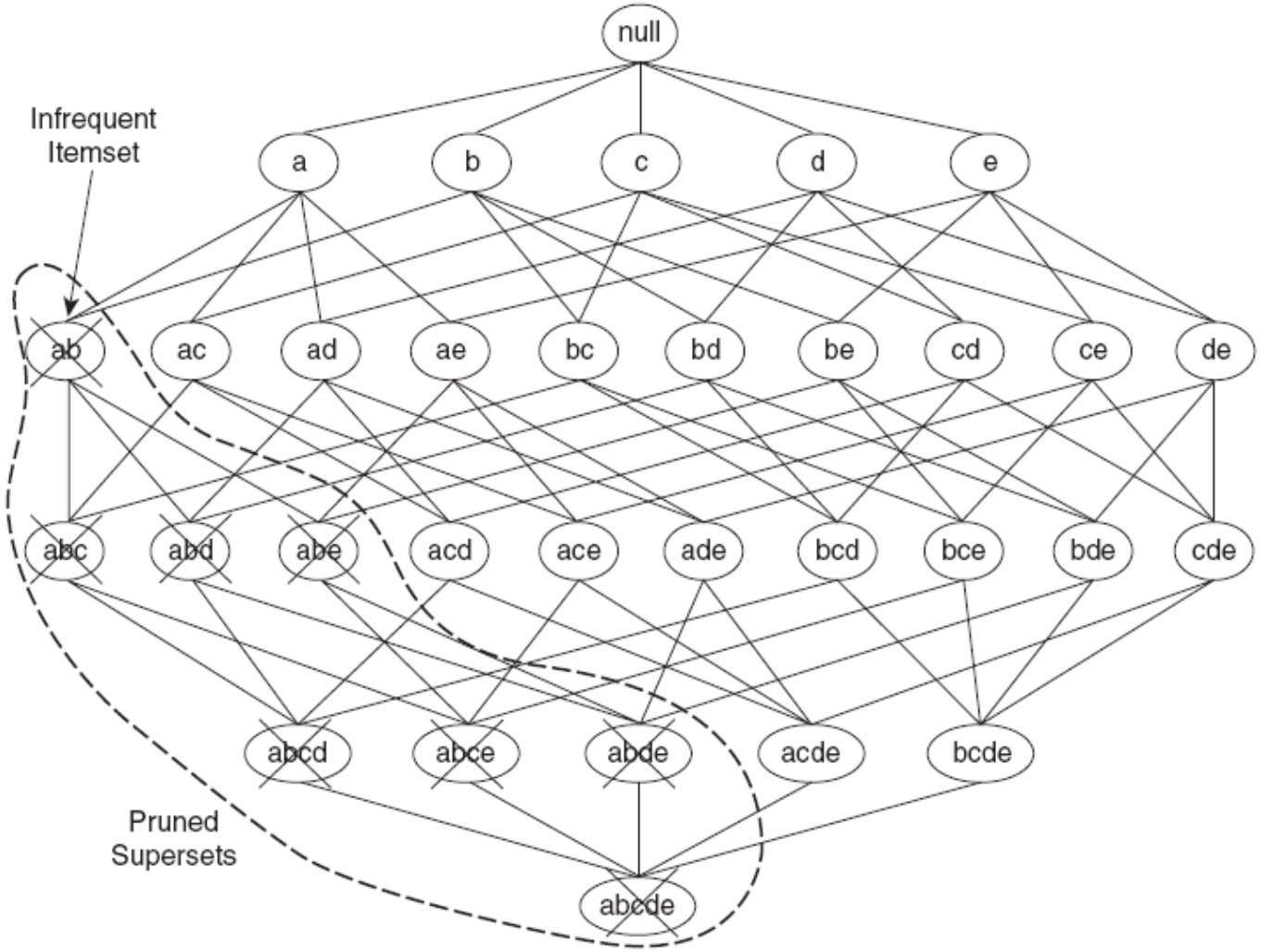
Lesser known fact: the idea dates back to the mid-1960s with Petr Hàjek's GUHA method (General Unary Hypothesis Automaton).

The Apriori Rule:

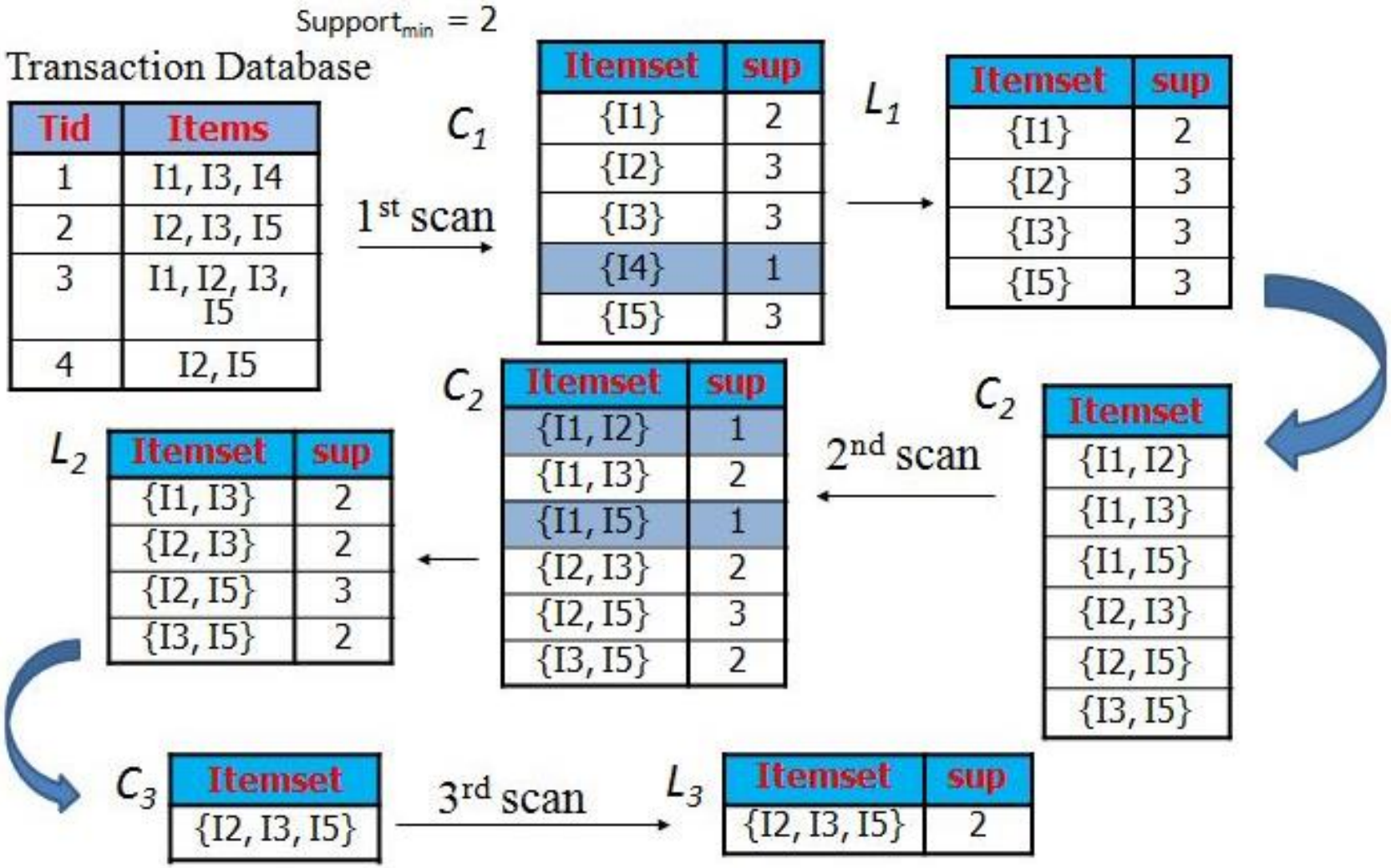
- If an itemset is frequent, then all of its subsets must also be frequent.
- Conversely, and more important: if an itemset is infrequent, then all of its supersets must be infrequent. This is known as the downward-closure property, anti-monotonicity property, or the Apriori-property.

APRIORI "SUPPORT PRUNING"

Systematically control the exponential growth of candidate itemsets by pruning those we don't need to consider.

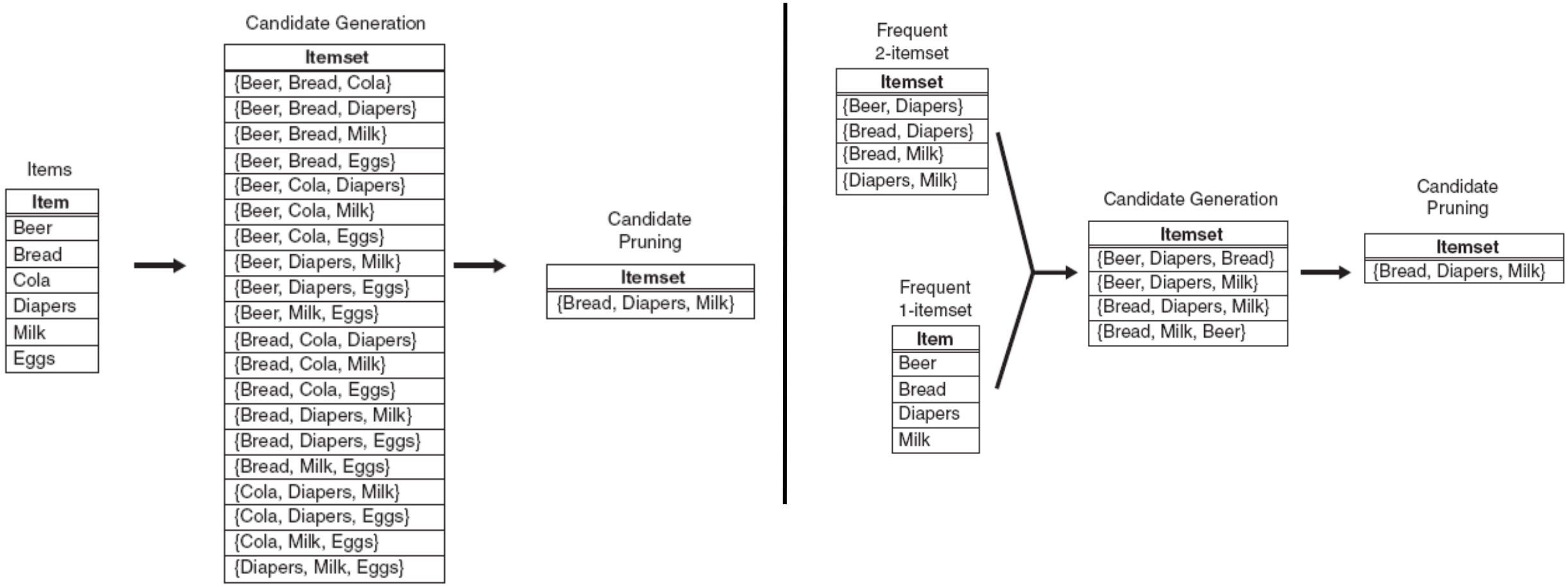


LET'S SEE IT IN ACTION

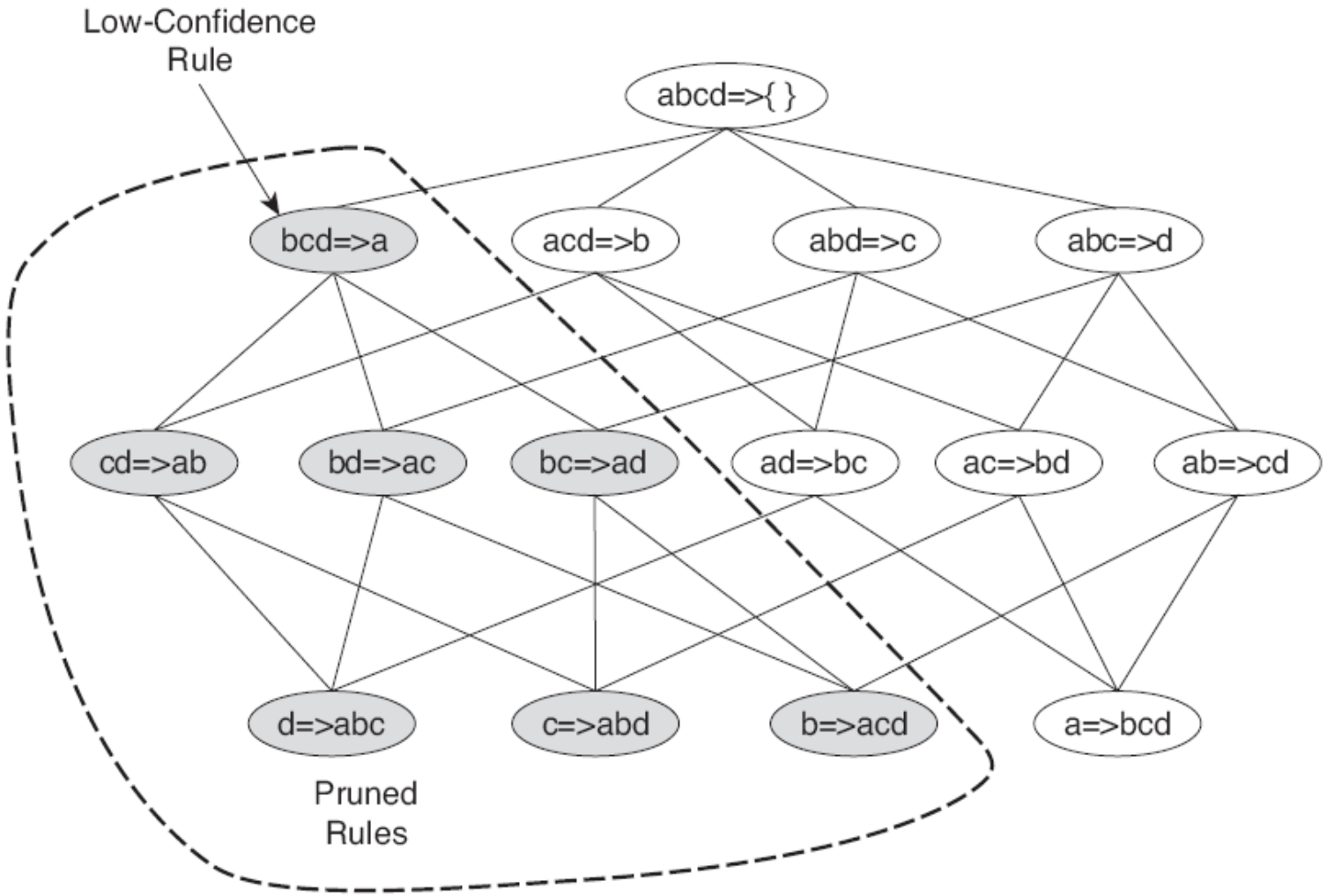


BRUTE FORCE VS. APRIORI

Generating 3-candidate itemsets



APRIORI "CONFIDENCE PRUNING"



PYTHON IMPLEMENTATION NOTES

Defaultdict(int) –or – Counter(). We'll be doing lots of counting

Frozenset()s are immutable and hashable, so can be used as the key in a dictionary. They also have the standard set operations (e.g. difference, intersection, issubset, union).

itertools chain() and combinations() allow for a more streamlined way to find all candidate subsets of different lengths.

Generators allow us to analyze datasets that don't fit in memory

Pre-made options: Apyori, Mlextend, Orange3-Associate, PyFim, Spark

PROBLEMS WITH APRIORI

1. Generating unnecessary candidates is still slow.

2. The transaction database still needs to be scanned n+1 times.

3. The support threshold, dimensionality, number of transactions, and average transaction width all have a large impact on complexity.

ENTER FP-GROWTH

Introduced in 2000 by Han, Pei, and Yin in Mining Frequent Patterns without Candidate Generation.

As opposed to a **Generate and Test Approach** for FIM (generate candidate itemsets and test for those that are frequent) like Apriori, FP-Growth uses a **Pattern Growth Approach**, counting occurrences in transactions for frequency of itemsets, then extracting frequent itemsets.

Advantages:

- Avoids excess candidate generation
- Uses only two database scans
- Uses a compact data structure

HOW FP-GROWTH WORKS

Part I: [Iteratively] build compact data structure called an FP-Tree**,** using two passes over the database.

- **Pass I: Collect the Support** counts for all items (using a dictionary), so we can eliminate transactions including infrequent items (thanks to the Apriori rule).
- **Pass II: Generate FP-Tree** (a compact data structure).

Part II: [Recursively] extract frequent itemsets (FIM) from the FP-Tree using a divide and conquer approach

FP-GROWTH- PART I, PASS I

Minsup = 3

TID	Transaction
1	r, z, h, j, p
2	z, y, x, w, v, u, t, s
3	z
4	r, x, n, o, s
5	y, r, x, z, q, t, p
6	y, z, x, e, q, s, t, m

Support
r:3
z:5
h:1
j:1
p:2
y:3
x:4
w:1
v:1
u:1
t:3
s:3
n:1
o:1
q:2
e:1
m:1

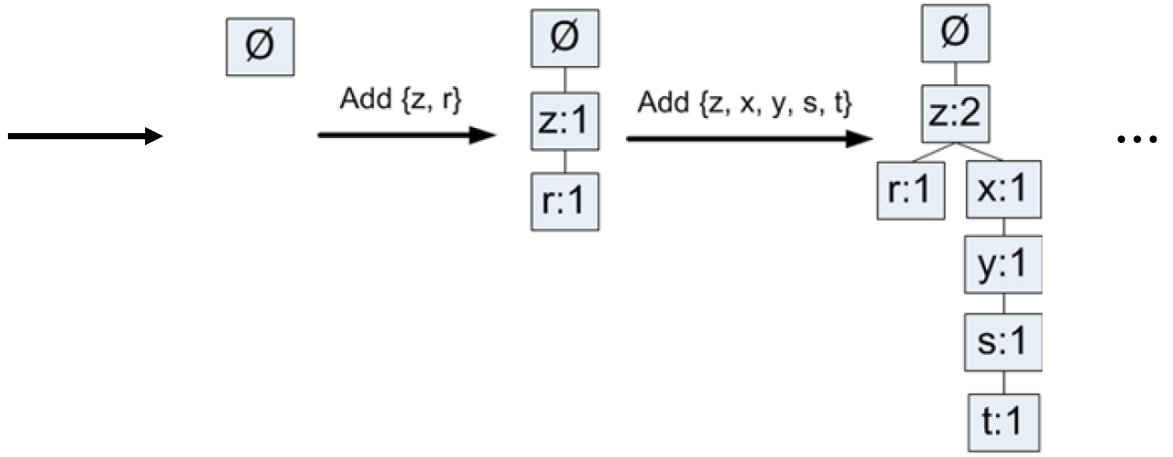
Header Table
z:5
r:3
x:4
y:3
s:3
t:3

TID	Original Transaction	Filtered and Sorted Transactions
1	r, z, ~~h, j, p~~	z, r
2	z, y, x, ~~w, v, u,~~ t, s	z, x, y, s, t
3	z	z
4	r, x, ~~n, o,~~ s	x, s, r
5	y, r, x, z, ~~q,~~ t~~, p~~	z, x, y, r, t
6	y, z, x, ~~e, q,~~ s, t~~, m~~	z, x, y, s, t

Transactions are sorted by header counts, descending. A common heuristic to allow common prefixes to be shared.

FP-GROWTH- PART I, PASS II (FP-TREE)

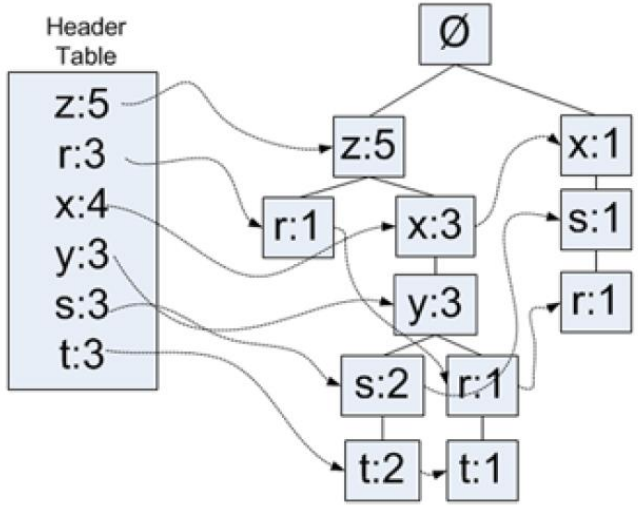
TID	Filtered and Sorted Transactions
1	z, r
2	z, x, y, s, t
3	z
4	x, s, r
5	z, x, y, r, t
6	z, x, y, s, t



If the first item in the transaction doesn't exist as a node, create one…

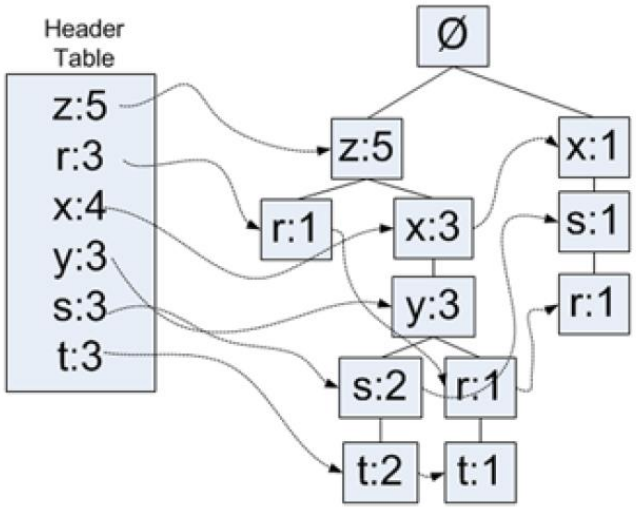
… while also creating a pointer from the header (or previous node for the same item, if one exists).

Pointers are maintained between nodes containing the same item, creating singly linked lists (squiggly arrows).



FP-GROWTH- PART II (FIM)

1. **For each item in our header (starting with the last item), we follow the linked list to create a sub-tree** (known as a Conditional FP-tree). This tree includes all frequent paths up to, but excluding the suffix we're considering. The list of all possible paths (and their associated counts) that end with that suffix is collectively known as the Conditional-Pattern base (and made up of Prefix Paths).
2. **Each tree is processed recursively to extract frequent itemsets.**
3. **Merge solutions.**



FP-GROWTH- PART II, EXAMPLE I

{t}'s sub-tree (conditional pattern bases)

- Two prefix paths: {z, x, y, s}, which appears twice and {z, x, y, r}, which appears once – making {t} frequent (we already knew that)

{t, s}'s sub-tree

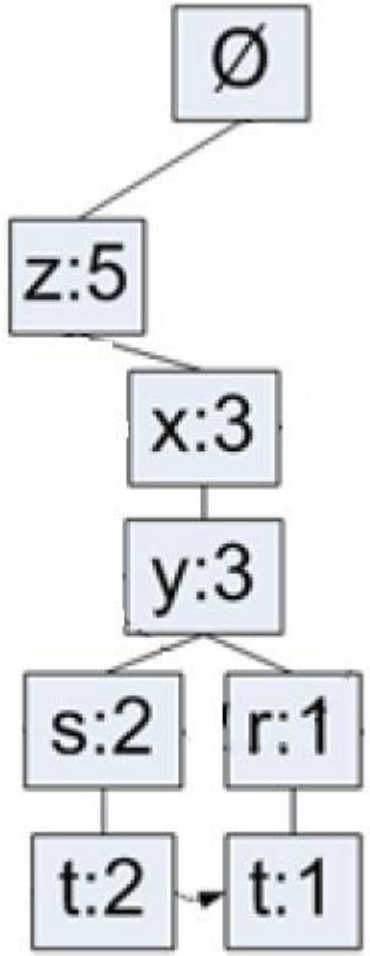
- One prefix path: {z, x, y}, which appears twice – meaning {t, s} isn't frequent.

{t, r} isn't frequent either.

{t, z}, {t, x}, {t, y} all are.

{t, y, x}, {t, x, z}, {t, y, z} all are.

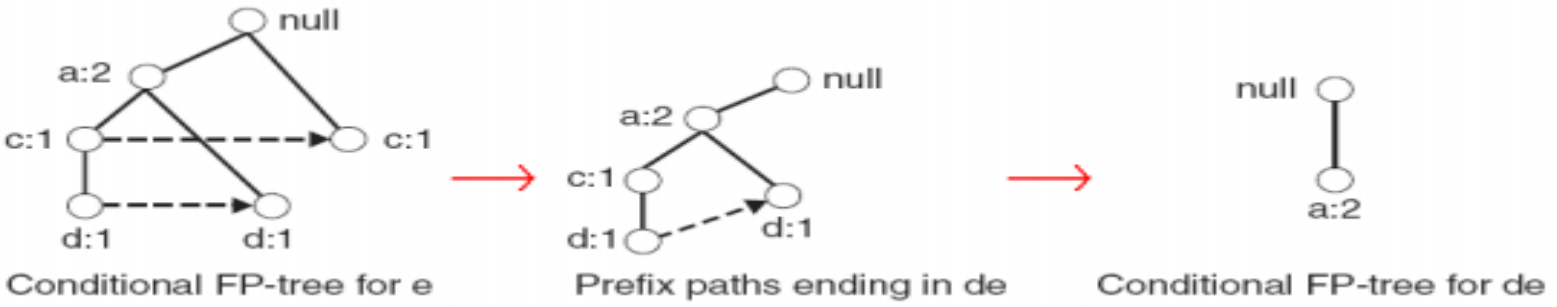
{t, x, y, z} is.



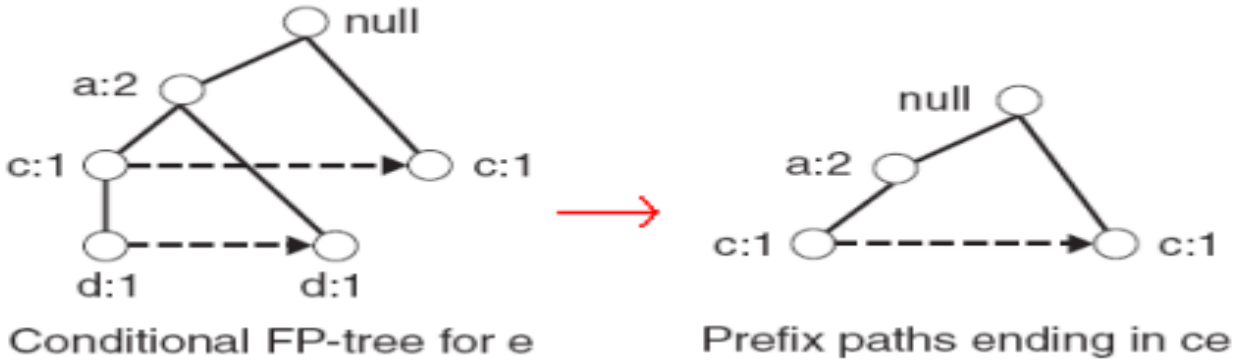
FP-GROWTH- PART II, EXAMPLE II

minsup = 2 for this example

For {e}'s conditional FP-tree, {d, e} is frequent. For {d, e}'s tree, {a,d,e} is frequent.



For {e}'s tree, {c, e} is frequent



PYTHON IMPLEMENTATION NOTES

Nothing beyond the notes for Apriori

Pre-made options: Orange3-Associate, PyFim, Spark

IS FP-GROWTH ALWAYS BETTER THAN APRIORI?

No! Which is better? The technically correct answer I always hate hearing: "it depends on your use case."

When to use each algorithm:

- **Apriori:** When you want to look at low-support items.
- **FP-growth:** When there are lots of duplicates and/or high dimensionality (candidate itemsets would be large for Apriori).

ARE THERE OTHER OPTIONS?

Algorithm	Type of search	Database representation
Apriori [3]	breadth-first (candidate generation)	Horizontal
Apriori − *TID* [3]	breadth-first (candidate generation)	Vertical (TID-lists)
Eclat [28]	depth-first (candidate generation)	Vertical (TID-lists, diffsets)
FP − *Growth* [27]	depth-first (pattern-growth)	Horizontal (prefix-tree)
H − *Mine* [29]	depth-first (pattern-growth)	Horizontal (hyperlink structure)
LCM [30]	depth-first (pattern-growth)	Horizontal (with transaction merging)

ADVANCED TOPICS

1. Tweaking data/parameters
2. More on efficiency
3. Choosing a measure of interestingness / rule selection
4. Infrequent rules
5. Specialized Algorithms
6. More advanced types of rules

TWEAKING DATA/PARAMETERS

Support threshold

- Unfortunately, trial and error. Based on the dataset.
- General tip: Try 0.20 (20%) and work your way down.

Number of items (dimensionality)

- Many dimensionality reduction techniques
- Tips: eliminations based on domain expertise –or- naively grouping like items

Number of transactions

- Relevant subsetting (e.g. time windows, single store in chain); sampling

Average transaction width

- Similar options to dimensionality reduction

MORE ON EFFICIENCY

We've already talked about reducing computational cost by reducing passes over the database (e.g. FP-Growth's advantage), reducing data/data presentation, and adding constraints.

Other considerations:

1. **Reduce accuracy**
   - Keep an approx. set of frequent itemsets rather than an exact set.
2. **Parallelization**
3. **Non-batch updating**
   - Incremental: Update frequent itemsets on database update; can be accomplished by keeping a buffer of almost-frequent itemsets in memory.
   - Stream: Method described under "reduce accuracy"; Popular algorithm: estDec+
   - Interactive: Mine only needed itemsets on-the-fly, as needed; Itemset-Tree algorithm

SELECTING INTERESTINGNESS MEASURES

What are you trying to measure?

Properties and appropriate interesting measures can be found in Selecting the right objective measure for association analysis:

- **Property I: Symmetry under variable permutation**
  - Asymmetric measures are used for implication rules, where distinguishing between the strength of A -> B and B -> A matters.
  - Asymmetric measures: confidence, conviction, Laplace, J-measure.
- **Property II: Row/Column Scaling Invariance**
- **Property III: Anti-symmetry Under Row/Column Permutation**
- **Property IV: Inversion Invariance**
- **Property V: Null Invariance**

INFREQUENT ITEMSETS/RULES

Low support, but high confidence rules:

- **Use case**: Rare association of symptoms indicating a rare disease.
- [**One**] **Solution**: The aptly-named **Apriori-Inverse** (2005) works like Aprior, but ignores itemsets above maxsup (instead of below minsup).
- **Other solutions**: AprioriRare, CORI, etc.

SPECIALIZED ALGORITHMS

- What set of items have recently been bought together? (Weighted Itemset Mining)
- People who bought these X items also bought… (Itemset-Tree)
- Which frequent baskets are generating the most profit? (HUIM - EFIM)
- How well are our discounting strategies working? (HUIM)
- What are the differences between men's and women's shopping behavior? (Emerging Pattern Mining)
- What item sets are purchased periodically? What patterns predict those purchases? (Periodic Pattern Mining, Sequential Pattern Mining)

RULE GENERATION ALTERNATIVES

Variants of association rules:

- Context (e.g. market basket analysis based on day of the week)
- Hierarchical
- Categorical
- Clustering
- Quantitative / Importance-weighted

Related topics:

- **Sequential Rule Mining**. Same as association rule, but with an order restriction. Useful for web page prefetching, anti-pattern detection, alarm sequence analysis, and restaurant recommendations.
- **Episode Mining**. Kind of like sequential, but in a single sequence rather than in a set of sequences. Useful for sensor readings, sequences of events on an assembly line, and network traffic data.
- **Sub-graph Mining**.

RESOURCES

Python Packages

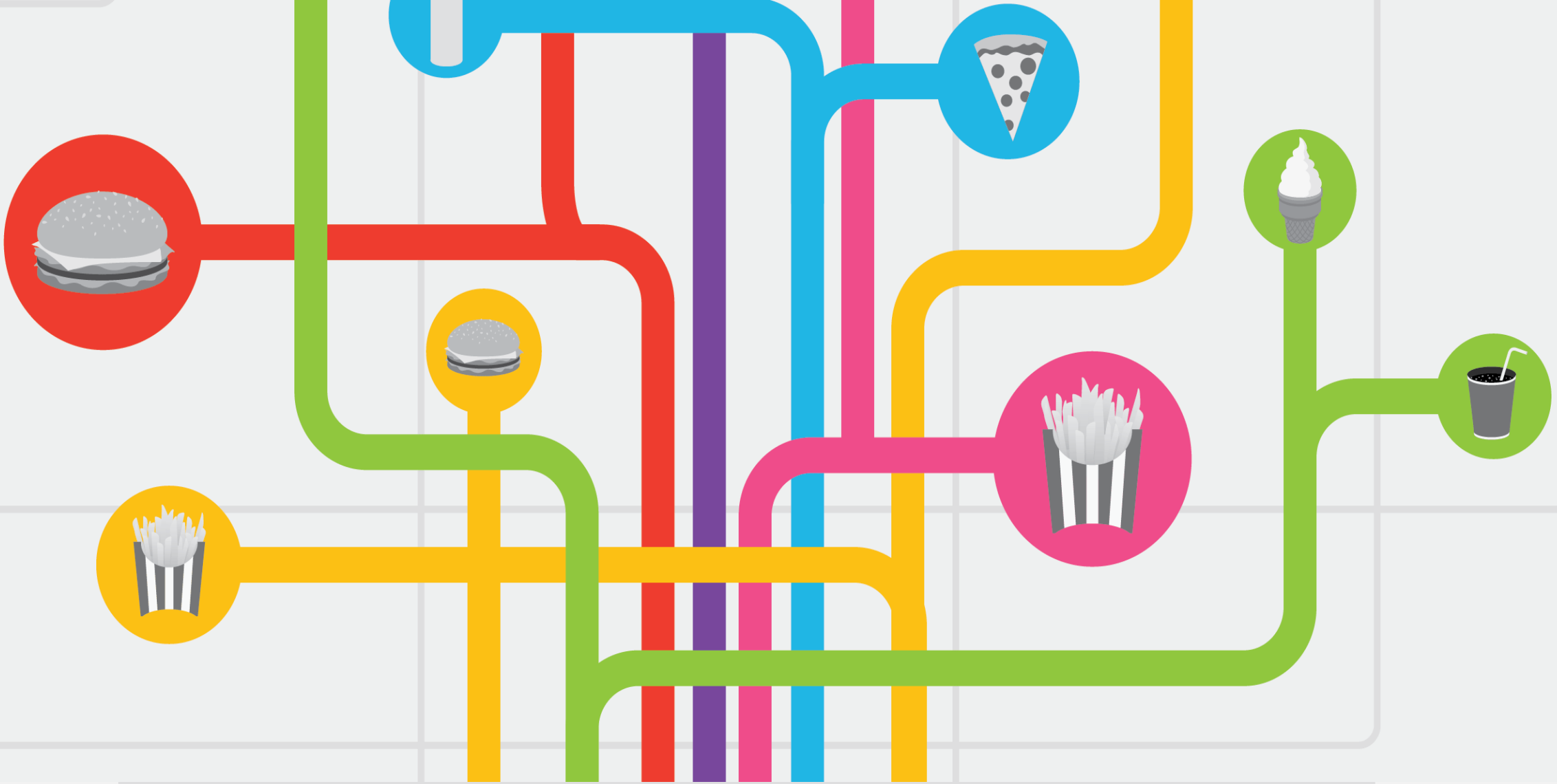
- **Apriori**: Apyori, Mlextend, Orange3-Associate, PyFim, Spark
- **FP-Growth**: Orange3-Associate, PyFim, Spark

Theory

- Introduction to Data Mining [Book] by Tan, Steinbach, and Kumar. Chapter 6 is free online.
- Mining of Massive Datasets [Book] by Leskovec, Rajaraman, and Ullman. Free Stanford class online.
- www.philippe-fournier-viger.com (+ Java implementations at /spmf)
- http://michael.hahsler.net/research/association_rules/measures.html (+ R implementations)

Implementation

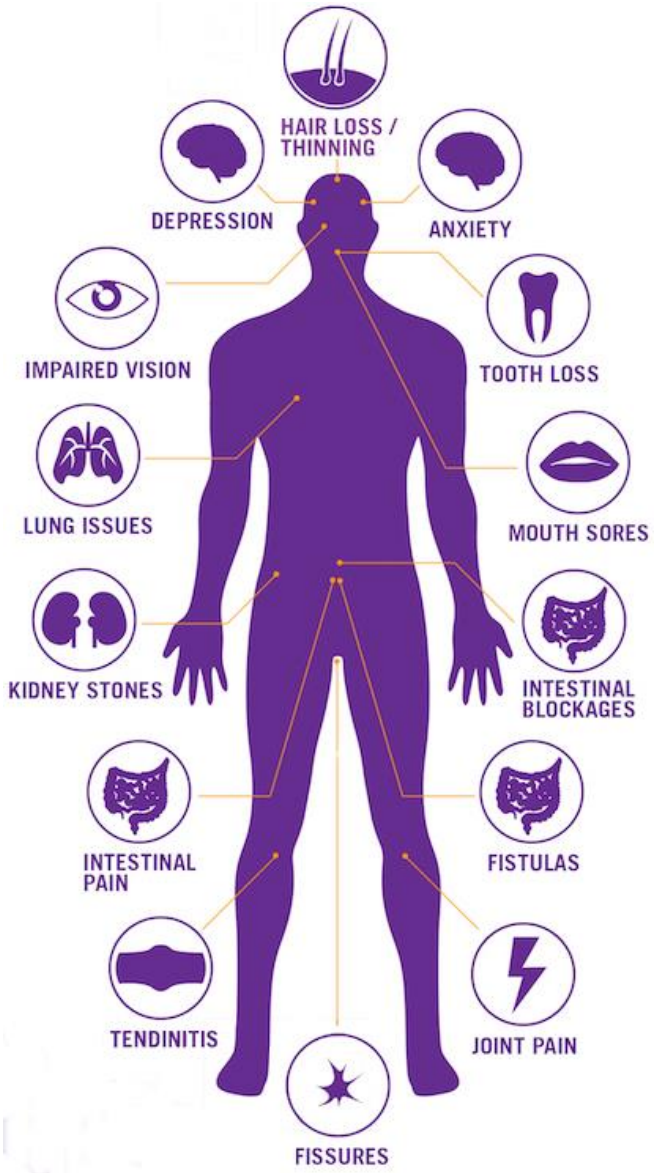
- Machine Learning in Action by Peter Harrington (Chapter 11: Apriori, Chapter 12: FP-Growth)



DETERMINING IBD TRIGGER FOODS USING MACHINE LEARNING AND PYTHON

WHAT'S IBD?

- Inflammatory bowel disease (IBD) describes a group of conditions, including Ulcerative Colitis (UC) and Crohn's disease (CD), impacting 1.6 million people in the US alone.
- Characterized by "gut" inflammation.
- Symptoms range from mild annoyances to life-threatening issues (blockages, cancer).
- Autoimmune, caused by a combination of genetic and environment factors.



WHAT'S FOOD GOT TO DO WITH IT?

- While foods' relationship with IBD remains understudied and controversial…
- …57% of IBD sufferers think diet can trigger symptom flare…
- …leading to food avoidance/malnourishment.
- Safe foods are thought to be person specific, in contrast to diseases like Celiac or lactose intolerance, where food issues are known.

WHY IT'S PERSONAL TO ME?

- In February 2016 I was diagnosed with Crohn's disease… and 10 ulcers.
- Medication has me ulcer free, but not symptom free.
- Certain foods can trigger flares lasting weeks.
- Trial and error to find safe foods is painful and takes a long time.

Real ulcers are gross, so here's some clipart:

You're welcome.

GOAL – WHAT CAN IBD SUFFERERS EAT?

1) Sub-clusters of diet?

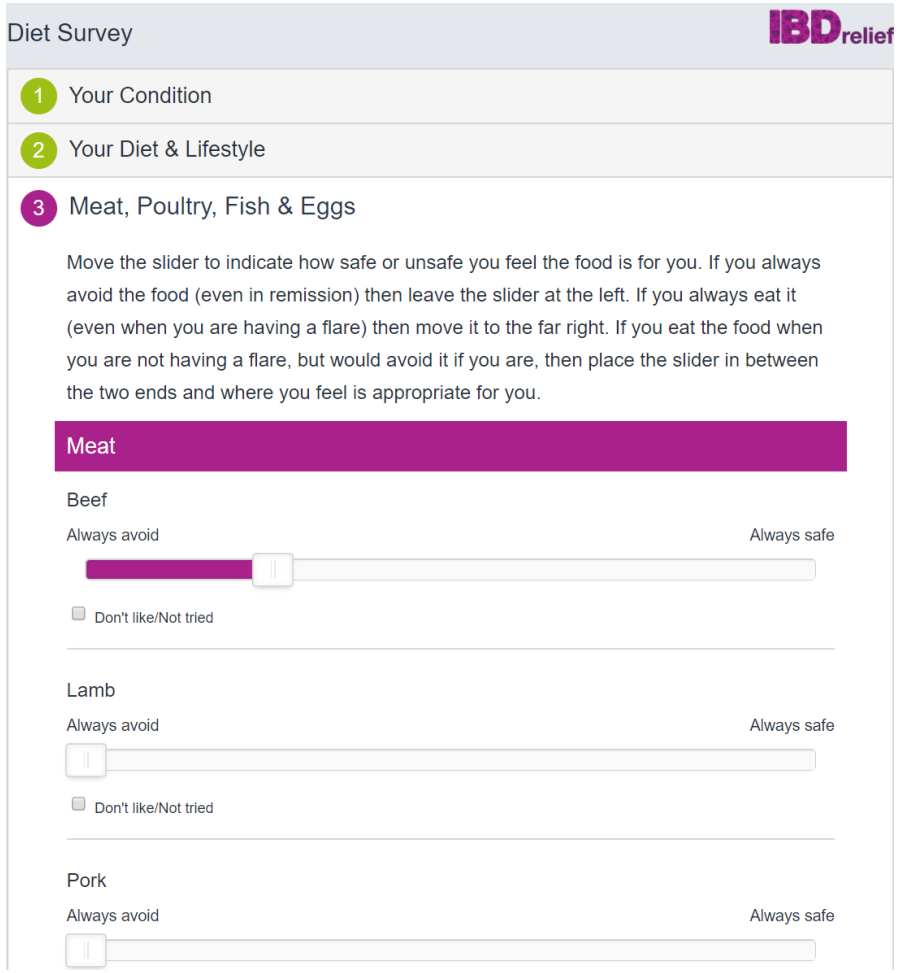
2) Relationships between individual foods or groups of foods?

3) Nutrients that impact food tolerance?

4) Can food tolerance/intolerance be predicted with a reasonable degree of accuracy for an IBD sufferer with only a few “known” safe/unsafe foods?

MATERIALS

- **Small data set: 670, 250-food survey responses** from IBD sufferers about food tolerances. 570 usable.
- **Nutrient information for each surveyed food** from the USDA's nutrient database API.
- **Python** 3.6.1 and **Jupyter Notebook**
  - Analysis: apyori, numpy, pandas, PyFIM, scikit-learn, scipy, sqlite
  - Visualization: graphviz, matplotlib, seaborn



The [online] survey utilizes a sliding scale to accept answer inputs, which are stored as integer values in a range from 0 through 10. A checkbox for each question gives the option to not answer questions individually.

Check out Introduction to Data Mining by Tan, Steinbach, and Kumar, Chapter 6 for an introduction to the basic concepts (free online).

ANALYSIS – ASSOCIATION RULE LEARNING

- **Rules for can eat AND can't eat.**
  - E.g. Apple and "Not Apple"

```
def df_to_association_input(df):
    #Convert df to acceptable input format
    transactions = df.copy()
    for col in transactions.columns:
        transactions[col].replace([1, 0.5, 0], [col, '', 'not '+col], inplace=True)

    transactions = transactions.values.tolist()

    for i, entry in enumerate(transactions):
        transactions[i][:] = filter(bool, entry)

    return transactions
```

- **High dimensionality.** Want to keep as many foods as possible

FP-GROWTH FOR THE EFFICIENCY WIN

- **FP-Growth** took 8 seconds versus Apriori's 4 minutes and 30 seconds.

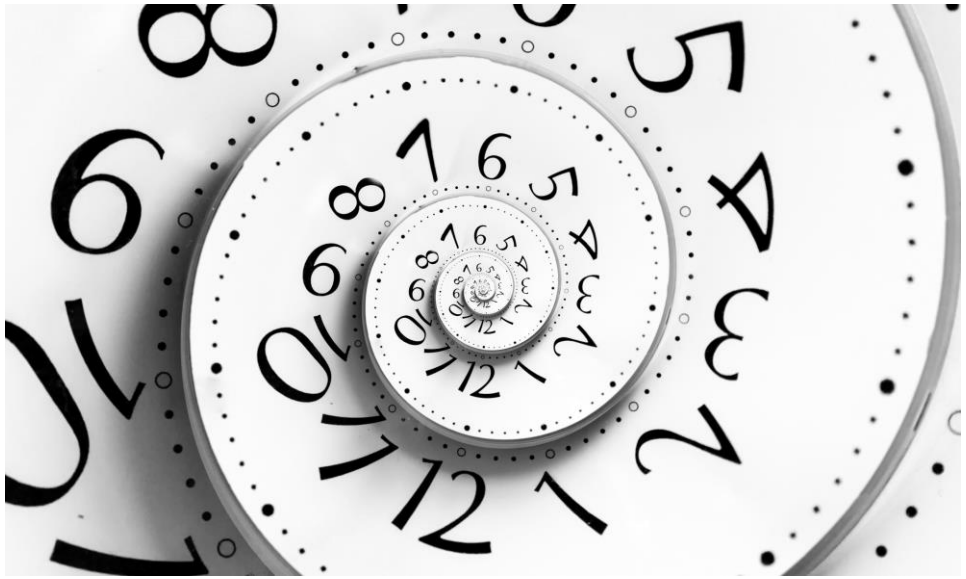
- **PyFim**: No many-to-many rules.

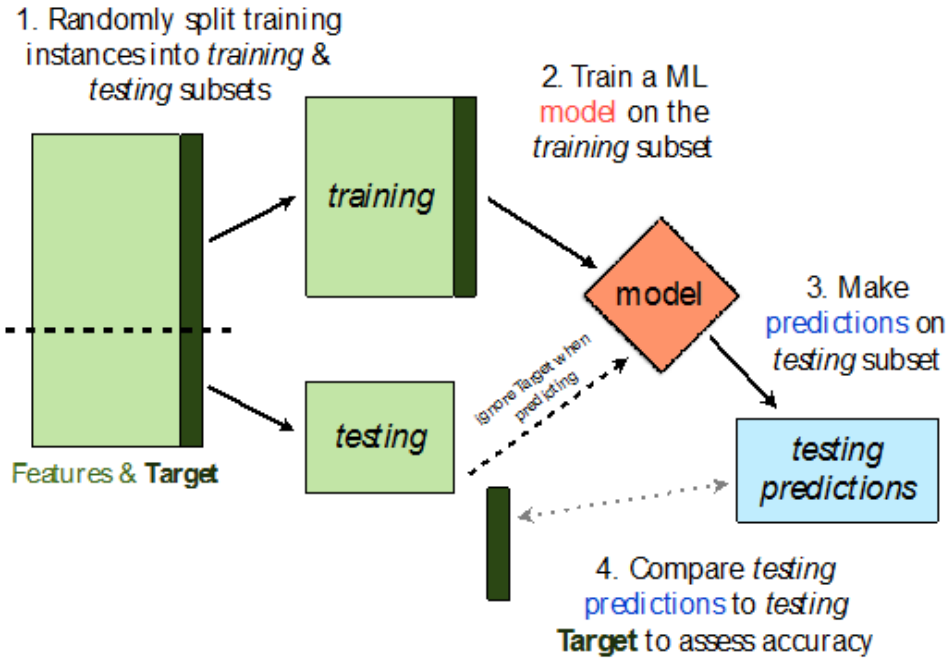
```
import pandas as pd
from fim import fpgrowth

rules_df = pd.DataFrame(fpgrowth(data, target='r', supp=10, conf=80, report='bCl',
                                 eval='l', agg='n', thresh=1, appear={'Filtered Water': '-', 'Tap Water': '-'}),
                        columns=['Consequent', 'Antecedents', 'Support (Count)', 'Confidence (%)','Lift (x)'],
                        index=None, copy=False)
```

  - **Note: Use `help(fpgrowth)` for argument options**

- **How good is the model?**

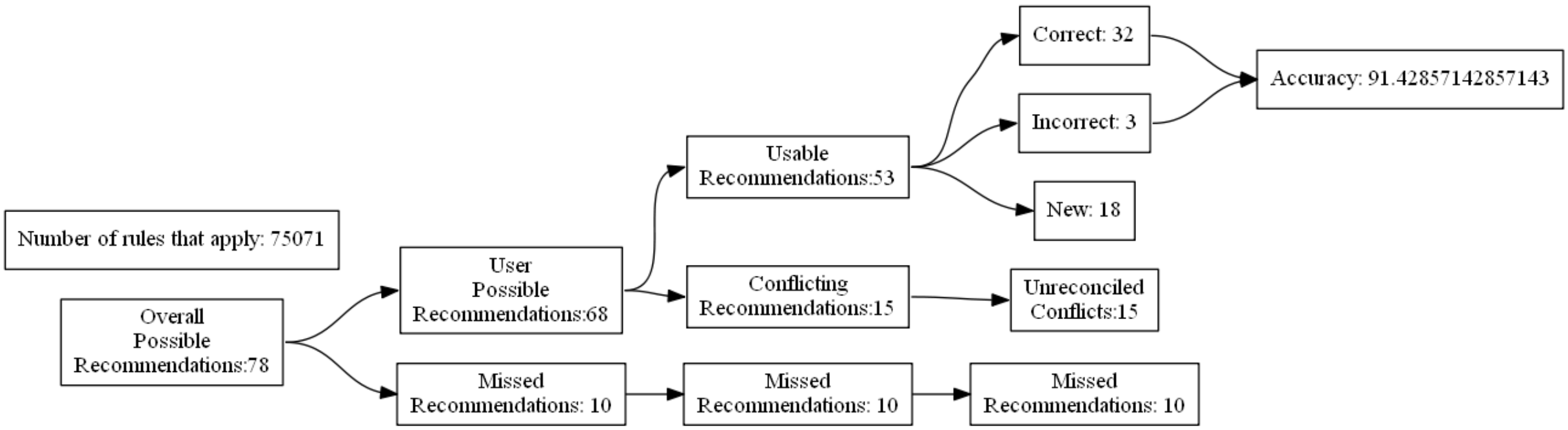




[SEMI-]NOVEL APPROACHES

1. **Logically ternary data** instead of binary
   - Adds information, but creates conflicts
   - New method of conflict resolution needed
2. **Monte Carlo cross-validation**
   - Association Rule Learning is inherently self validating, but need model comparability
   - Evaluation method (accuracy) determined by applicable subsets of rules, per tested transactions

VALIDATION

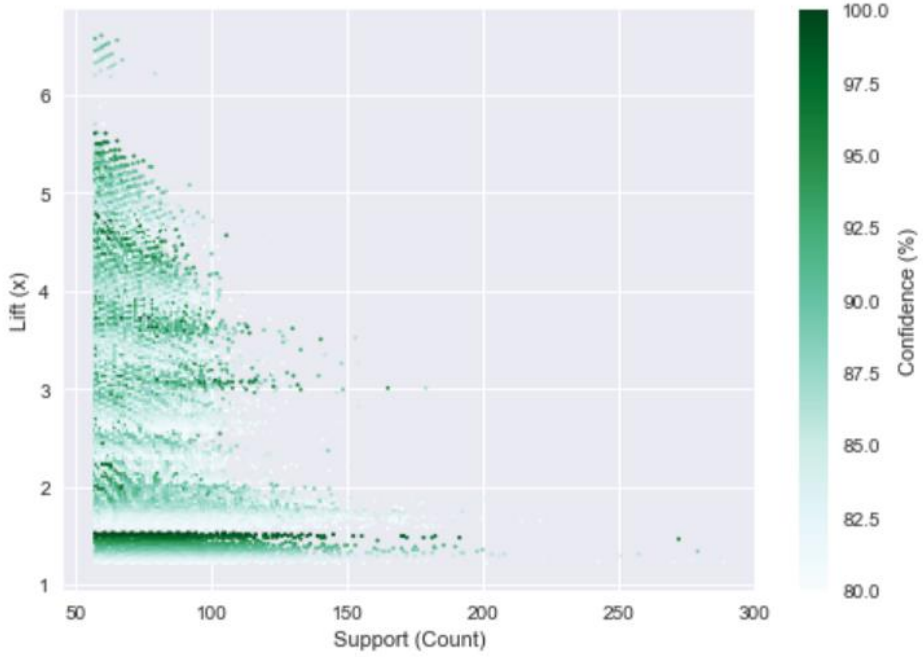

Number of rules that apply: 75071
Overall Possible Recommendations:78
User Possible Recommendations:68
Missed Recommendations: 10
Usable Recommendations:53
Conflicting Recommendations:15
Missed Recommendations: 10
Correct: 32
Incorrect: 3
New: 18
Unreconciled Conflicts:15
Missed Recommendations: 10
Accuracy: 91.42857142857143

RESULTS

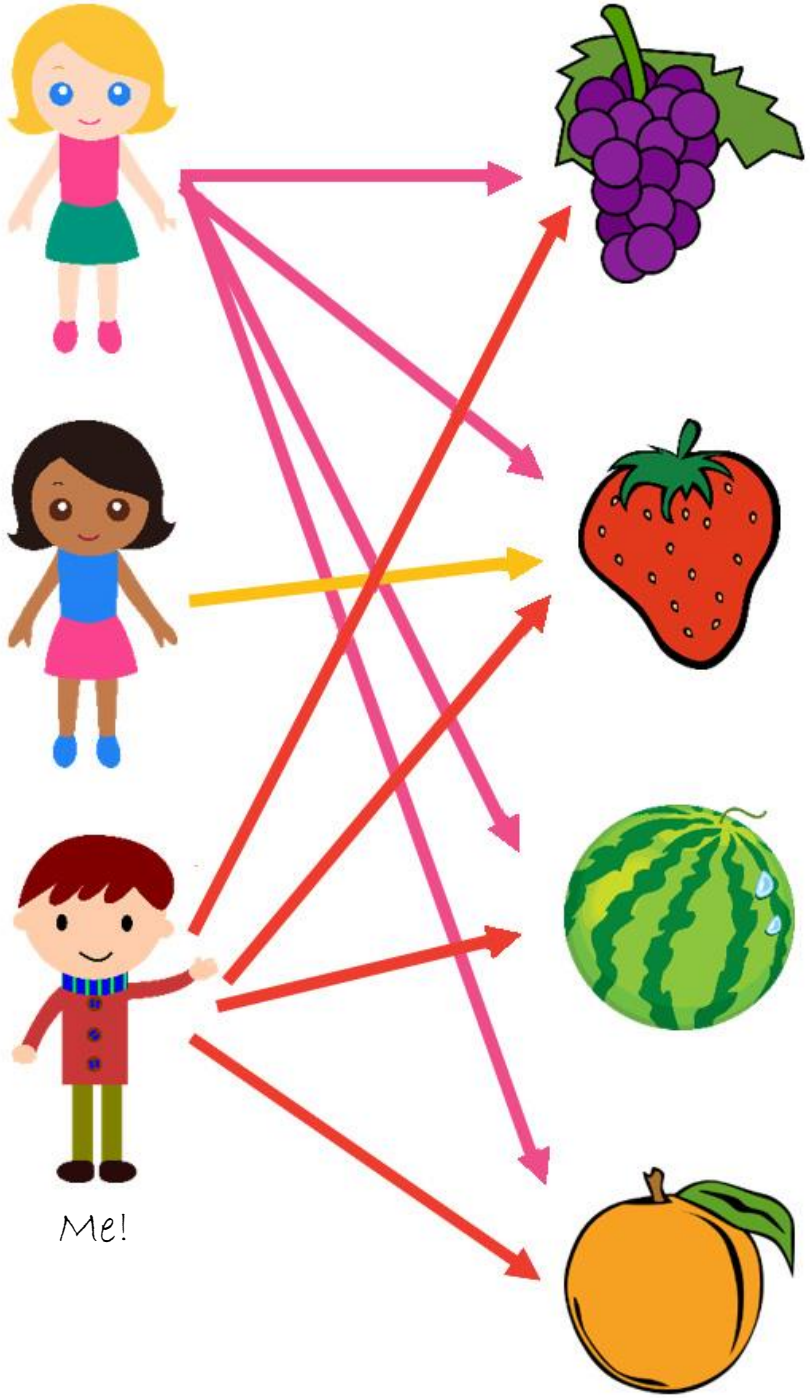
- Recommendations at least 80%+ accurate, usually 90%+
- Average 18-19 new recommendations pp.
- Commonly recommended foods: leeks, lettuce, garlic, honeydew melon, cod, cantaloupe, chicken eggs, basil, cucumber, white potatoes.
- Commonly conflicting foods: fruit, dairy, cruciferous vegetables

THE FULL MODEL

- ✓ 888,926 rules generated
- ✓ Rules for 74% of possible recommendations, with >80% confidence
  - ✓ **Can eat rules:** animals, 'staple' veges (carrots, cucumber, lettuce, tomato, potato), white rice
  - ✓ **Can't eat rules:** apple juice, coffee, cola, raisins
  - ✓ **Cut rules:** not alcohol of various types



Total Foods: 164 → Rules for: 121 → Conflicts for: 43 → Up to 78 usable



IBDALIZER

- Recommendation tool using input survey data
- Background output:

	Consequent	Antecedents	Support (Count)			Confidence (%)			Lift (x)		
		<lambda>	min	mean	max	min	mean	max	min	mean	max
0	Almond	[(Cashews, Bananas, Chicken) ...	46	49	62	80.00%	82.47%	91.30%	4.74x	4.88x	5.41x
1	Apple	[(Strawberries, Rice, White ...	46	49	98	80.00%	83.88%	97.83%	2.68x	2.81x	3.28x
2	Apple Juice	[(Cherries, Lemon, Raspbe ...	46	46	46	80.43%	80.43%	80.43%	3.60x	3.60x	3.60x
3	Bananas	[(Butter, Sweet Potato, C ...	46	50	87	80.00%	82.37%	93.88%	1.88x	1.94x	2.21x
...	...	...	...	...	...	...	...	...	...	...	...
15	not Cola	[(not Lemonade, Chicken), (not ...	57	64	120	80.00%	83.70%	90.28%	2.62x	2.74x	2.96x

```
df = df.groupby(['Consequent']).agg({'Antecedents': lambda ante: list(ante),
                                     'Support (Count)': ['min', 'mean', 'max'],
                                     'Confidence (%)': ['min', 'mean', 'max'],
                                     'Lift (x)': ['min', 'mean', 'max']
                                    }).reset_index()
```

FUTURE WORK

- **Update survey for recommendations**
- **Integrate live recommendation system into the survey** (with feedback and "learning")
- **Apply more advanced association techniques,** including hierarchical and clustering
- **Use my USDA nutrient database tool to identify relevant nutrients**

WHAT WE LEARNED

1. What association rule learning is.
2. What we can do with it
3. How to use it, via the Apriori and FP-growth algorithms
4. How to be efficient
5. Some advanced techniques

Q&A

zaxrosenberg.com

github.com/zaxr

RESOURCES

Python Packages

- **Apriori**: Apyori, Mlextend, Orange3-Associate, PyFim, Spark
- **FP-Growth**: Orange3-Associate, PyFim, Spark

Theory

- Introduction to Data Mining [Book] by Tan, Steinbach, and Kumar. Chapter 6 is free online.
- Mining of Massive Datasets [Book] by Leskovec, Rajaraman, and Ullman. Free Stanford class online.
- www.philippe-fournier-viger.com (+ Java implementations at /spmf)
- http://michael.hahsler.net/research/association_rules/measures.html (+ R implementations)

Implementation

- Machine Learning in Action by Peter Harrington (Chapter 11: Apriori, Chapter 12: FP-Growth)