COMP6237 Data Mining

"Market Basket" Analysis & Association Rule Mining

Jonathon Hare
jsh2@ecs.soton.ac.uk

Content based on material from slides from *Evgueni Smirnov* at the University of Maastricht, slides from *João Mendes Moreira* & *José Luís Borges* at the University of Porto, and notes from *Nitin Patel* at MIT

Introduction

- Association Rule Problem
- Applications
- The Apriori Algorithm
- Discovering Association Rules
- Measures for Association Rules

Association Rule

if X then Y

$$\mathbf{X \Rightarrow Y}$$

Association Rule Problem

- Given a database of transactions:

Transaction	Items
t_1	Bread,Jelly,PeanutButter
t_2	Bread,PeanutButter
t_3	Bread,Milk,PeanutButter
t_4	Beer,Bread
t_5	Beer,Milk

Find all the association rules:

$X \Rightarrow Y$	s	α
Bread $\Rightarrow$ PeanutButter	60%	75%
PeanutButter $\Rightarrow$ Bread	60%	100%
Beer $\Rightarrow$ Bread	20%	50%
PeanutButter $\Rightarrow$ Jelly	20%	33.3%
Jelly $\Rightarrow$ PeanutButter	20%	100%
Jelly $\Rightarrow$ Milk	0%	0%

Applications 1

- **Market Basket Analysis:**
  - given a database of customer transactions, where each transaction is a set of items the goal is to find groups of items which are frequently purchased together.
  - One *basket* tells you about what one customer purchased at one time.

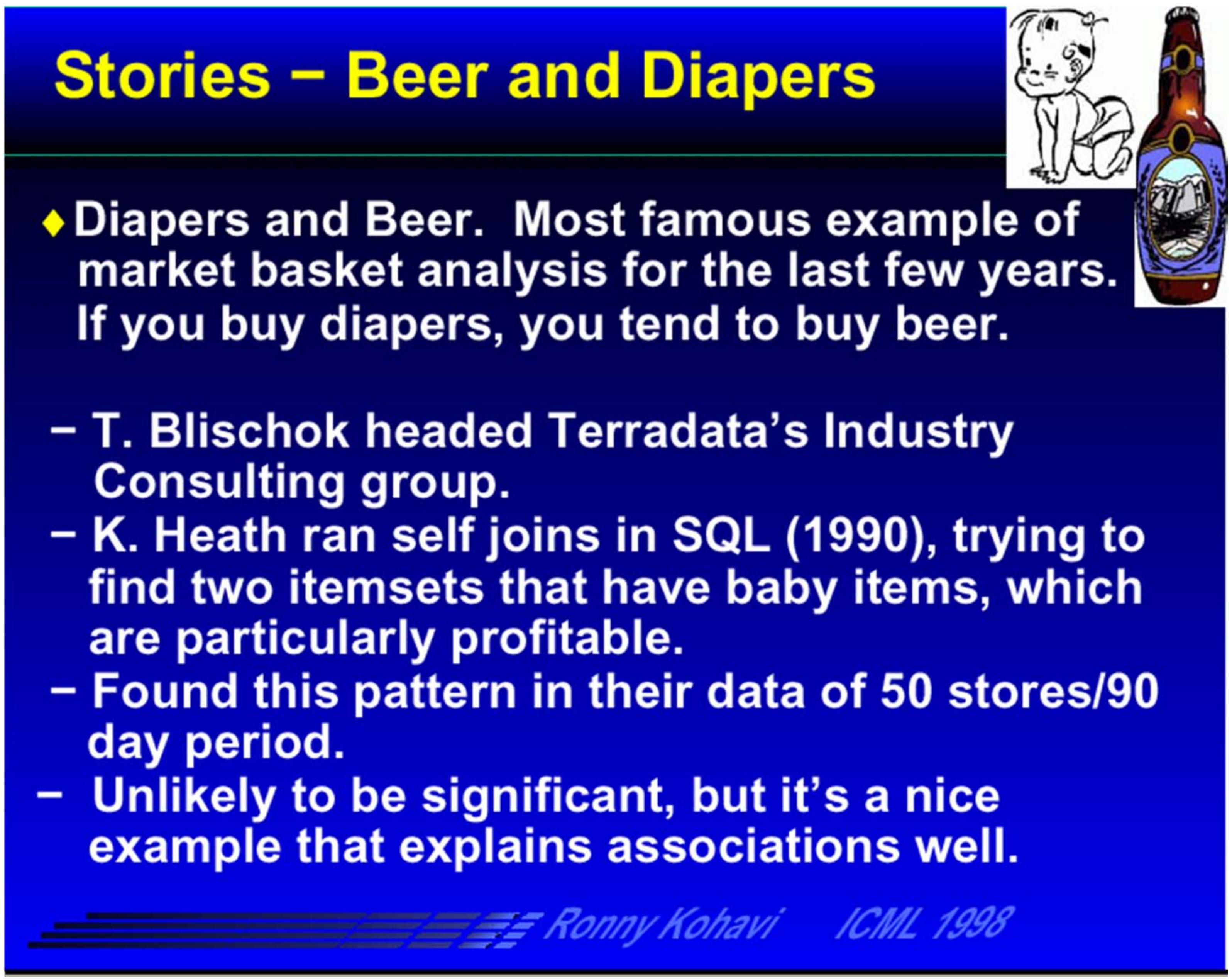
Why analyse market baskets?

- **Identify** who customers are (not by name)
- **Understand** why they make certain purchases
- **Gain insight** about products:
  - Fast and slow movers
  - Products which are purchased together
  - Products which might benefit from promotion
- **Take action**:
  - Store layouts
  - Which products to put on specials, promote, coupons...

More than just the contents of a shopping basket…

- **What customers do not purchase, and why:**
  - If customers purchase baking powder, but no flour, what are they baking?
  - If customers purchase a mobile phone, but no case, are you missing an opportunity?
- **Key drivers of purchases:**
  - e.g. gourmet mustard that seems to lie on a shelf collecting dust until a customer buys that particular brand of special gourmet mustard in a shopping excursion that includes hundreds of pounds worth of other products.
    - Would eliminating the mustard (to replace it with a better-selling item) threaten the entire customer relationship?

A story about *MBA*



Probably mom was calling dad at work to buy diapers on way home and he decided to buy a six-pack as well.

The retailer could move diapers and beers to separate places and position high-profit items of interest to young fathers along the path.

Applications 2

- **Telecommunication** (each customer is a transaction containing the set of phone calls)
- **Credit Cards/ Banking Services** (each card/account is a transaction containing the set of customer's payments)
- **Medical Treatments** (each patient is represented as a transaction containing the ordered set of diseases)
- **Basketball-Game Analysis** (each game is represented as a transaction containing the ordered set of ball passes)
- *… more coming later…*

Mining Association Rules

Association Rule Definitions

- $I=\{i_1, i_2, \ldots, i_n\}$: a set of all the items
- Transaction T: a set of items such that $T \subseteq I$
- Transaction Database D: a set of transactions
- A transaction $T \subseteq I$ contains a set $X \subseteq I$ of some items, if $X \subseteq T$
- **An Association Rule**: is an implication of the form $X \Longrightarrow Y$, where $X, Y \subseteq I$

Association Rule Definitions

- A set of items is referred as an **itemset**.
  - An itemset that contains k items is a k-itemset.
- The **support** s of an itemset X is the percentage of transactions in the transaction database D that contain X.
- The **support** of the rule $X \Longrightarrow Y$ in the transaction database D is the support of the items set $X \cup Y$ in D.
- The **confidence** of the rule $X \Longrightarrow Y$ in the transaction database D is the ratio of the number of transactions in D that contain $X \cup Y$ to the number of transactions that contain X in D.

Association Rule Problem

- **Given:**
  - a set *I* of all the items;
  - a database *D* of transactions;
  - minimum support *s*;
  - minimum confidence *c*;
- **Find:**
  - all association rules $X \Rightarrow Y$ with a minimum support *s* and confidence *c*.

Example

- Given a database of transactions:

Transaction	Items
t_1	Bread,Jelly,PeanutButter
t_2	Bread,PeanutButter
t_3	Bread,Milk,PeanutButter
t_4	Beer,Bread
t_5	Beer,Milk

Find all the association rules:

$X \Rightarrow Y$	s	α
Bread $\Rightarrow$ PeanutButter	60%	75%
PeanutButter $\Rightarrow$ Bread	60%	100%
Beer $\Rightarrow$ Bread	20%	50%
PeanutButter $\Rightarrow$ Jelly	20%	33.3%
Jelly $\Rightarrow$ PeanutButter	20%	100%
Jelly $\Rightarrow$ Milk	0%	0%

Problem Decomposition

1. Find all sets of items that have minimum support (frequent itemsets)

2. Use the frequent itemsets to generate the desired rules

Problem Decomposition

Transaction ID	Items Bought
1	Shoes, Shirt, Jacket
2	Shoes,Jacket
3	Shoes, Jeans
4	Shirt, Sweatshirt

If the **minimum support** is 50%, then {Shoes,Jacket} is the only 2- itemset that satisfies the minimum support.

Frequent Itemset	Support
{Shoes}	75%
{Shirt}	50%
{Jacket}	50%
{Shoes, Jacket}	50%

If the **minimum confidence** is 50%, then the only two rules generated from this 2-itemset, that have confidence greater than 50%, are:

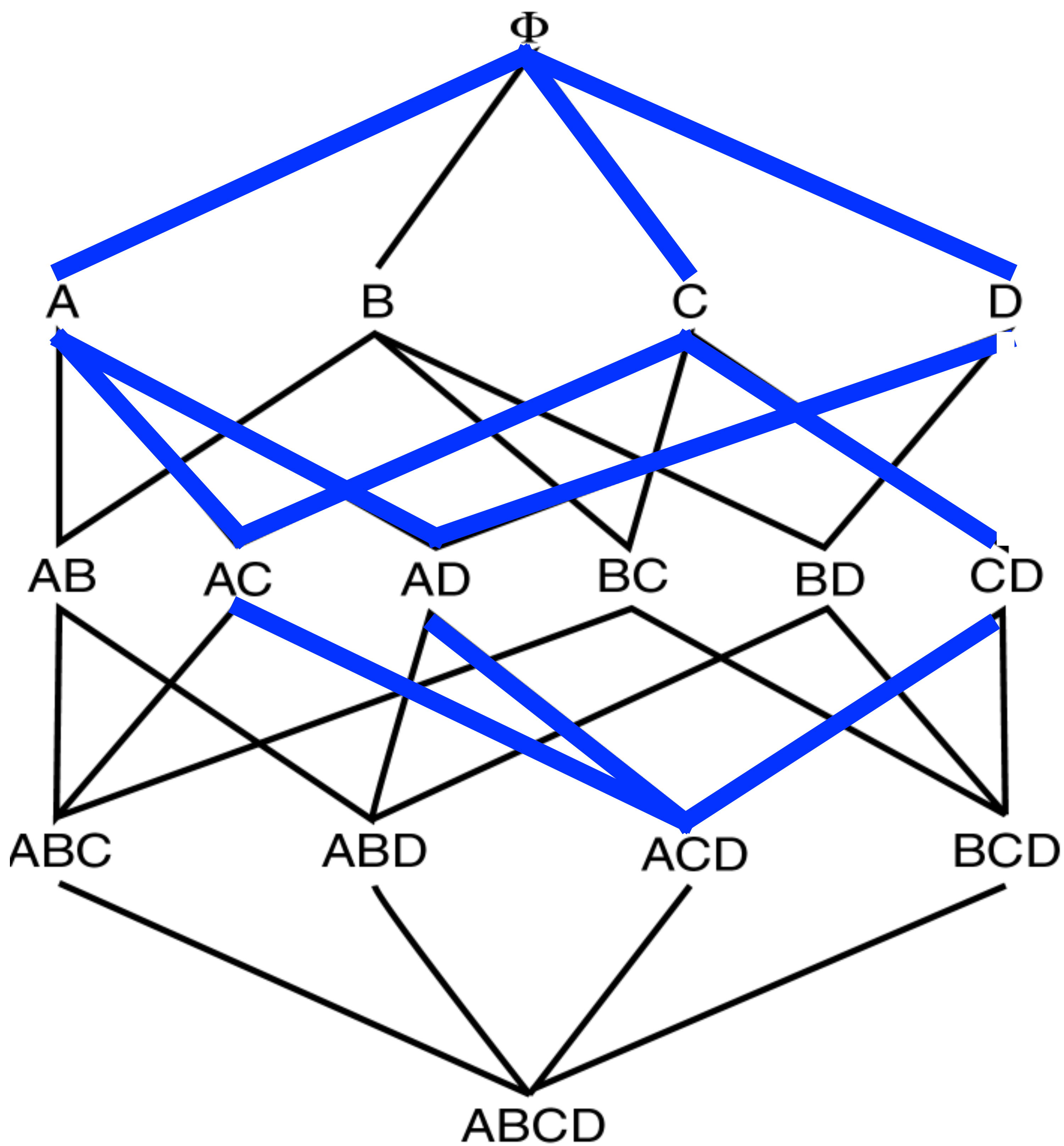
Shoes $\Rightarrow$ Jacket Support=50%, Confidence=66%

Jacket $\Rightarrow$ Shoes Support=50%, Confidence=100%

The Apriori Algorithm

- Frequent itemset property:
  - *Any subset of a frequent itemset is frequent.*
- Contrapositive:
  - *If an itemset is not frequent, none of its supersets are frequent.*

Frequent Itemset Property



The Apriori Algorithm

- L_k: Set of frequent itemsets of size k (with min support)
- C_k: Set of candidate itemset of size k (potentially frequent itemsets)

```
L1 = {frequent items};
for (k= 1; Lk !=∅; k++) do
    Ck+1 = candidates generated from Lk;
    for each transaction t in database do:
        increment the count of all candidates
         in Ck+1 that are contained in t;
    Lk+1 = candidates in Ck+1 with min_support;
return ∪kLk;
```

The Apriori Algorithm — Example

Min support =50%

Database D

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D

C_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

L_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3

C_2

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

Scan D

C_2

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

L_2

itemset	sup
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

C_3

itemset
{2 3 5}

Scan D

L_3

itemset	sup
{2 3 5}	2

How to Generate Candidates

```
Input: L_{i-1} : set of frequent itemsets of size i-1
Output: C_i : set of candidate itemsets of size i
C_i = empty set;
for each itemset J in L_{i-1} do
  for each itemset K in L_{i-1} s.t. K≠J do
    if i-2 of the elements in J and K are equal then
      if all subsets of {K ∪ J} are in L_{i-1} then
        C_i = C_i ∪ {K ∪ J}
return C_i;
```

Example of Generating Candidates

- $L_3 = \{abc, abd, acd, ace, bcd\}$
- Generating C_4 from L_3
  - *abcd* from *abc* and *abd* (or *abc* and *acd*, …)
  - *acde* from *acd* and *ace*
- Pruning:
  - *acde* is removed because *ade* is not in L_3
- $C_4 = \{abcd\}$

How can we generate rules?

- Let use consider the 3-itemset {I1, I2, I5}:
  - $I1 \wedge I2 \Longrightarrow I5$
  - $I1 \wedge I5 \Longrightarrow I2$
  - $I2 \wedge I5 \Longrightarrow I1$
  - $I1 \wedge I2 \Longrightarrow I5$
  - $I2 \wedge I1 \Longrightarrow I5$
  - $I5 \wedge I1 \Longrightarrow I2$

Consider all permutations of rules from the items

Discovering Rules

```
for each frequent itemset I do
  for each subset C of I do
     if (support(I) / support(I - C) >= minconf) then
         output the rule (I - C) ⟹ C,
         with confidence = support(I) / support (I - C)
             and support = support(I)
```

Considerations of the Apriori algorithm

- Advantages:
  - Uses large itemset property
  - Easily parallelised
  - Easy to implement
- Disadvantages:
  - Assumes transaction database is memory resident
  - Requires many database scans

Better measures for rules

Problems with confidence

- The **confidence** of $X \Longrightarrow Y$ in database D is the ratio of the number of transactions containing $X \cup Y$ to the number of transactions that contain X:

$$conf(X \to Y) = \frac{\dfrac{numTrans(X \cup Y)}{|D|}}{\dfrac{numTrans(X)}{|D|}} = \frac{p(X \wedge Y)}{p(X)} = p(Y \mid X)$$

- But, when Y is independent of X: $p(Y) = p(Y \mid X)$.
  - If $p(Y)$ is high we'll have a rule with high confidence that associates independent itemsets!
    - For example, if p("milk") = 80% and "milk" is independent from "salmon", then the rule "salmon" $\Longrightarrow$ "milk" will have confidence 80%!

Alternative Measures for Association Rules

- The **lift** measure indicates the departure from independence of *X* and *Y*.

  - The lift of X ⟹ Y is:

$$lift(X \rightarrow Y) = \frac{conf(X \rightarrow Y)}{p(Y)} = \frac{\frac{p(X \wedge Y)}{p(X)}}{p(Y)} = \frac{p(X \wedge Y)}{p(X)\,p(Y)}$$

- But, the lift measure is symmetric; i.e., it does not take into account the direction of implications!

Alternative Measures for Association Rules

- The conviction measure indicates the departure from independence of X and Y taking into account the implication direction.

  - The conviction of $X \Longrightarrow Y$ is:

$$conv(X \rightarrow Y) = \frac{p(X)\,p(\neg Y)}{p(X \wedge \neg Y)}$$

Further applications (not shopping related)

Finding Linked Concepts

- “Baskets” = **documents**
- “items” = **words** in those documents
  - Lets us find words that appear together unusually frequently, i.e., **linked concepts**.

	Word1	Word2	Word3	Word4
Doc1	1	0	1	1
Doc2	0	0	1	1
Doc3	1	1	1	0

- Word4 $\Longrightarrow$ Word3
  - When Word4 occurs in a document there a big probability of Word3 occurring

Detecting Plagiarism

- “Baskets” = **sentences**
- “items” = **documents** containing those sentences
  - Items that appear together too often could represent **plagiarism.**

	Doc1	Doc2	Doc3	Doc4
Sent1	1	0	1	1
Sent2	0	0	1	1
Sent3	1	1	1	0

  - Doc4 $\Longrightarrow$ Doc3
    - When a sentence occurs in document 4 there is a big probability of occurring in document 3

Working with webpages

- "Baskets" = **Web pages**
- "items" = **linked pages**
  - Pairs of pages with many common references may be about the same topic.
- "Baskets" = **Web pages, p_i**
- "items" = **pages that link to p_i**
  - Pages with many of the same links may be mirrors or about the same topic.

Summary

- Association Rules form a very applied data mining approach
  - lots of potential uses
- Association Rules are derived from frequent itemsets
- The Apriori algorithm is an efficient algorithm for finding all frequent itemsets
  - implements level-wise search using the frequent item property
- There are many measures for association rules.