Data Mining & Machine Learning

Yong Zheng

Illinois Institute of Technology
Chicago, IL, 60616, USA

ILLINOIS TECH
College of Computing

Schedule

- Outlier Detection
- Associate Rule Mining
  - Application: Web Usage Mining

Schedule

- Outlier Detection
- Associate Rule Mining
  - Application: Web Usage Mining

Anomaly/Outlier Detection

- What are anomalies/outliers?
  - The set of data points that are considerably different than the remainder or the majority of the data
- Variants of Anomaly/Outlier Detection Problems
  - Given a database D, find all the data points **x** $\in$ D with anomaly scores greater than some threshold t
  - Given a database D, find all the data points **x** $\in$ D having the top-n largest anomaly scores f(**x**)
  - Given a database D, containing mostly normal (but unlabeled) data points, and a test point **x**, compute the anomaly score of **x** with respect to D
- Applications:
  - Credit card fraud detection, telecommunication fraud detection, network intrusion detection, fault detection

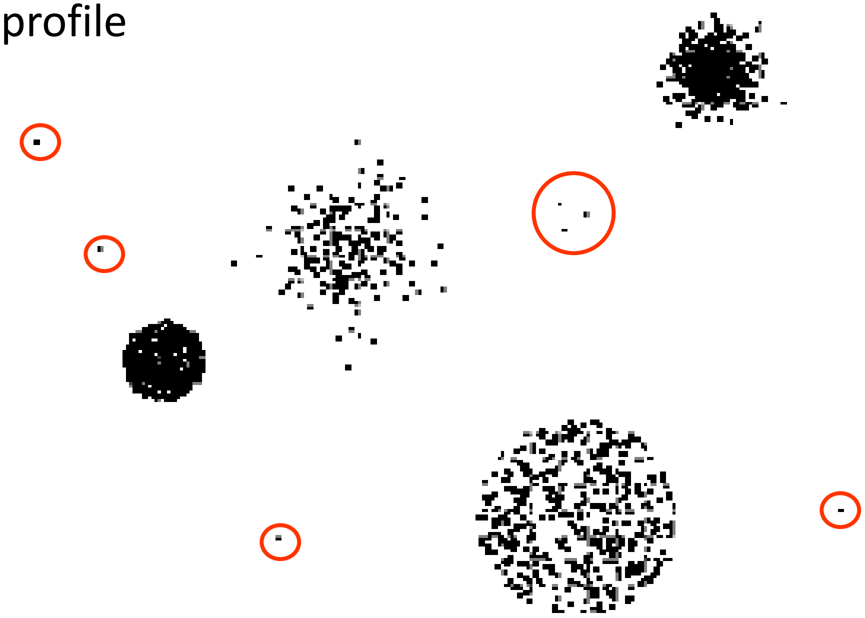
Anomaly/Outlier Detection

- Notes
  - Outliers are just anomaly data points
  - Outliers are not necessary to be "influential points"
  - Influential points are the data points which leave negative impacts on models
  - Influential points are usually outliers
- We can identify outliers from different detection techniques. But they are not necessary to have negative impact on models

Anomaly Detection Schemes

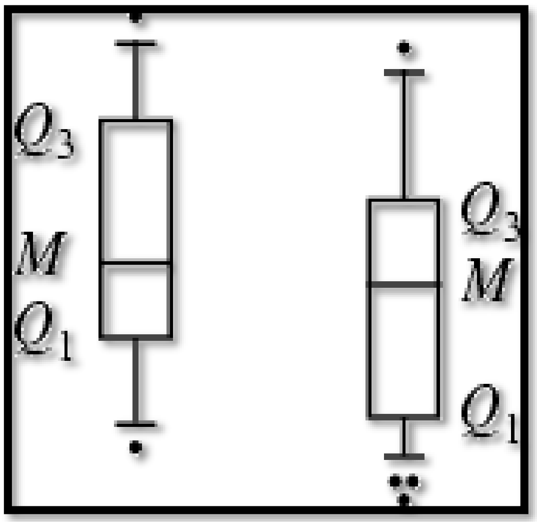
- General Steps
  - Build a profile of the "normal" behavior
    - Profile can be patterns or summary statistics for the overall population
  - Use the "normal" profile to detect anomalies
    - Anomalies are observations whose characteristics differ significantly from the normal profile

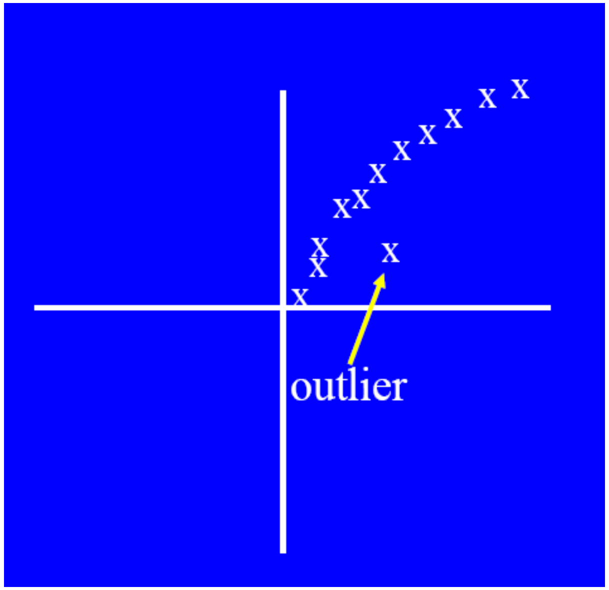
- Types of anomaly detection schemes
  - Graphical
  - Model-based
  - Distance-based
  - Clustering-based

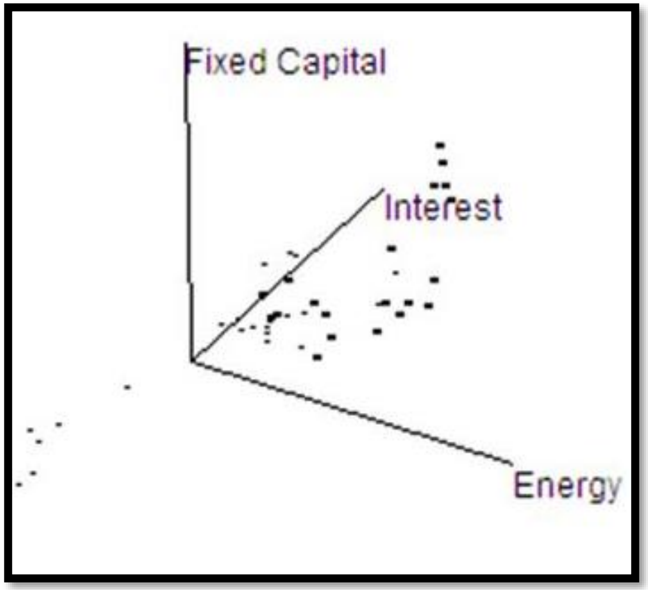


Graphical Approaches

- Boxplot (1-D)
- Scatter plot (2-D)
- Spin/3D plot (3-D)







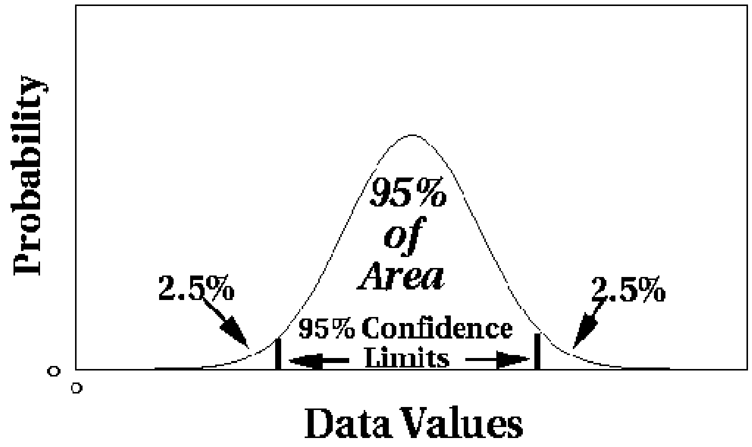
Graphical Approaches

- Limitations
  - Time consuming
  - Subjective
- Notes
  - A data set usually has several dimensions
  - Boxplot can only identify outliers from 1 dimension
  - Scatter plot can only identify outliers from 2 dims
  - Outliers on 1 or 2 dimensions are not necessary to be outliers on multi-dimensions

Statistical Approaches---Model-based

- Assume a parametric model describing the distribution of the data (e.g., normal distribution)

- Apply a statistical test that depends on
  - Data distribution
  - Parameter of distribution (e.g., mean, variance)
  - Number of expected outliers (confidence limit)



Distance-based Approaches

- Data is represented as a vector of features

- Three major approaches
  - Nearest-neighbor based
  - Density based
  - Clustering based

Nearest-Neighbor Based Approach

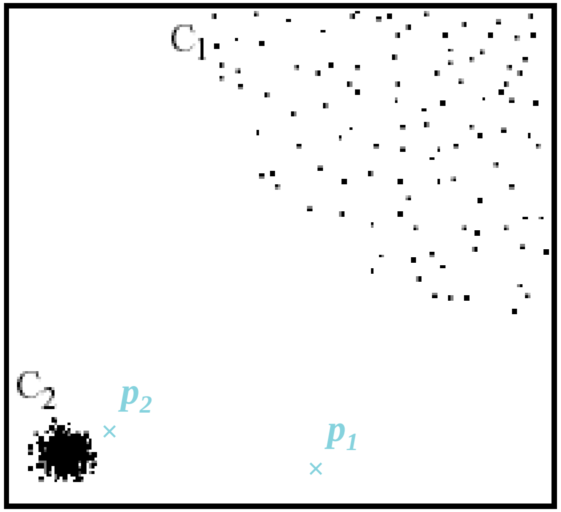
- Approach:
  - Compute the distance between every pair of data points
  - There are various ways to define outliers:
    - Data points for which there are fewer than *p* neighboring points within a distance *D*
    - The top n data points whose distance to the kth nearest neighbor is largest
    - The top n data points whose average distance to the k nearest neighbors is largest

Clustering-Based

- Idea: Use a clustering algorithm that has some notion of outliers!
- The data which are far away from the centroid could be outliers
- The set of data in a small cluster may be outliers
- Clustering-based methods can work together with 3D visualizations

Density-based: LOF approach

- For each point, compute the density of its local neighborhood; e.g. use DBSCAN's approach
- Compute local outlier factor (LOF) of a sample *p* as the average of the ratios of the density of sample *p* and the density of its nearest neighbors
- Outliers are points with largest LOF value



In the NN approach, p_2 is not considered as outlier, while LOF approach find both p_1 and p_2 as outliers

It depends on the threshold in LOF approach
We will learn Python coding for LOF later

Schedule

- Outlier Detection
- Associate Rule Mining
  - Application: Web Usage Mining

Market Basket Analysis

- Goal of MBA is to find associations (affinities) among groups of items occurring in a transactional database
  - has roots in analysis of point-of-sale data, as in supermarkets
  - but, has found applications in many other areas
- Association Rule Discovery
  - most common type of MBA technique
  - Find all rules that associate the presence of one set of items with that of another set of items.
  - Example: ***98% of people who purchase tires and auto accessories also get automotive services done***
  - We are interested in rules that are
    - non-trivial (and possibly unexpected)
    - actionable
    - easily explainable

Example: Diaper and Beer

What Is Association Mining?

- Association rule mining searches for relationships between items in a data set:
  - Finding association, correlation, or causal structures among sets of items or objects in transaction databases, relational databases, etc.
- Rule form:
  - Body ==> Head [support, confidence]
  - Body and Head can be represented as sets of items or as predicates
- Examples:
  - {diaper, milk, Thursday} ==> {beer} [0.5%, 78%]
  - buys(x, "bread") ==> buys(x, "milk") [0.6%, 65%]
  - major(x, "CS") /\ takes(x, "DB") ==> grade(x, "A") [1%, 75%]
  - age(X,30-45) /\ income(X, 50K-75K) ==> buys(X, SUVcar)
  - age="30-45", income="50K-75K" ==> car="SUV"

It can be considered as an unsupervised learning process.
Because we have no idea about what kind of patterns we can find

Different Kinds of Association Rules

- Boolean vs. Quantitative
  - associations on discrete and categorical data vs. continuous data
- Single Vs. Multiple Dimensions
  - one predicate = single dimension; multiple predicates = multiple dimensions
  - buys(x, “milk”) ==> buys(x, “butter”)
  - age(X,30-45) /\ income(X, 50K-75K) ==> buys(X, SUVcar)
- Single level vs. multiple-level analysis
  - Based on the level of abstractions involved
  - buys(x, “bread”) ==> buys(x, “milk”)
  - buys(x, “<u>wheat bread</u>”) ==> buys(x, “<u>2% milk</u>”)
- Simple vs. constraint-based
  - Constraints can be added on the rules to be discovered

Basic Concepts

- We start with a set I of items and a set D of transactions
  - $I = \{i_1, i_2, \ldots, i_m\}$
  - ***D*** is all of the transactions relevant to the mining task
- A transaction *T* is a set of items (a subset of I): $\boxed{T \subseteq I}$
- An Association Rule is an implication on *itemsets* *X* and *Y* , denoted by *X* ==> *Y*, where

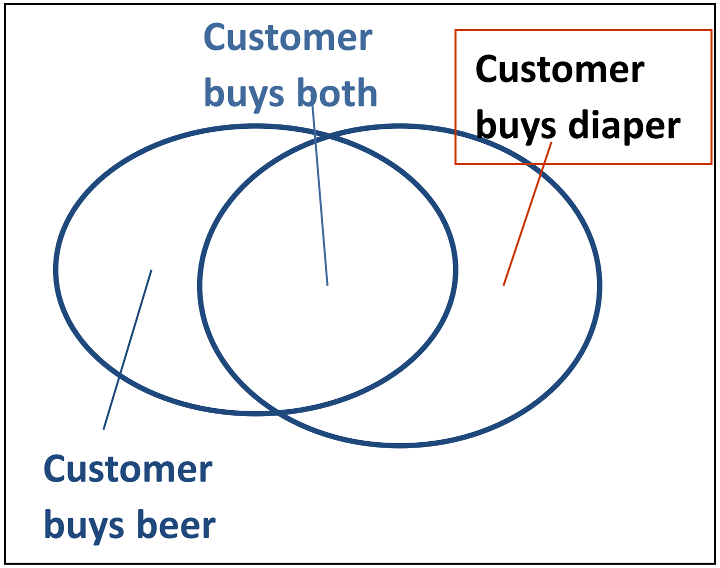
$$\boxed{X \subseteq I, Y \subseteq I, \quad X \cap Y = \emptyset}$$

- The rule meets a minimum confidence of *c*, meaning that *c*% of transactions in D which contain X also contain Y

$$\boxed{c \geq |X \cup Y| / |X|}$$

- In addition a minimum support of s is satisfied $\boxed{s \geq |X \cup Y| / |D|}$

Support and Confidence



Find all the rules $X \Rightarrow Y$ with minimum confidence and support

- Support = probability that a transaction contains {X,Y}
  - i.e., ratio of transactions in which X, Y occur together to all transactions
- Confidence = conditional probability that a transaction having X also contains *Y*
  - i.e., ratio of transactions in which X, Y occur together to those in which X occurs.

In general confidence of a rule LHS => RHS can be computed as the support of the whole itemset divided by the support of LHS:

Confidence (LHS => RHS) = Support(LHS ∪ RHS) / Support(LHS)

Support and Confidence - Example

Transaction ID	Items Bought
1001	A, B, C
1002	A, C
1003	A, D
1004	B, E, F
1005	A, D, F

Itemset {A, C} has a support of 2/5 = 40%

Rule {A} ==> {C} has confidence of 50%

Rule {C} ==> {A} has confidence of 100%

Support for {A, C, E} ?
Support for {A, D, F} ?

Confidence for {A, D} ==> {F} ?
Confidence for {A} ==> {D, F} ?

Improvement (Lift)

- High confidence rules are not necessarily useful
  - what if confidence of {A, B} ==> {C} is less than Pr(C)?
  - improvement gives the predictive power of a rule compared to just random chance:

$$improvement = \frac{\Pr(result \mid condition)}{\Pr(result)} = \frac{confidence(rule)}{support(result)}$$

Transaction ID	Items Bought
1001	A, B, C
1002	A, C
1003	A, D
1004	B, E, F
1005	A, D, F

Itemset {A} has a support of 4/5
Rule {C} ==> {A} has confidence of 2/2

Improvement = 5/4 = 1.25

Itemset {A} has a support of 4/5
Rule {B} ==> {A} has confidence of 1/2

Improvement = 5/8 = 0.625

Steps in Association Rule Discovery

- Find the *frequent* itemsets
  - Frequent item sets are the sets of items that have minimum support
  - a subset of a frequent itemset must also be a frequent itemset
    - if {AB} is a frequent itemset, both {A} and {B} are frequent itemsets
    - this also means that if an itemset that doesn't satisfy minimum support, none of its supersets will either (this is essential for pruning search space)
- Use the frequent itemsets to generate association rules

Apriori Algorithm: Find Frequent Itemset

C_k : Candidate itemset of size k

L_k : Frequent itemset of size k

```
L_1 = {frequent items};
for (k = 1; L_k !=∅; k++) do begin
     C_{k+1} = candidates generated from L_k;
    for each transaction t in database do
            increment the count of all candidates in
       C_{k+1}   that are contained in t
    L_{k+1}  = candidates in C_{k+1} with min_support
    end
return ∪_k L_k;
```

Join Step: C_k is generated by joining $L_{k-1}$with itself

Prune Step: Any (k-1)-itemset that is not frequent cannot be a subset of a frequent k-itemset

Example of Generating Candidates

- L_3={*abc, abd, acd, ace, bcd*}
- Self-joining: L_3*L_3
  - *abcd* from *abc* and *abd*
  - *acde* from *acd* and *ace*
- Pruning:
  - *acde* is removed because *ade* is not in L_3
- C_4 = {*abcd*}

Apriori Algorithm - An Example

Assume minimum support = 2

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

Note: {1,2,3}{1,2,5} and {1,3,5} not in C_3

Apriori Algorithm - An Example

L_2

item set	sup
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

L_3

itemset	sup
{2 3 5}	2

The final "frequent" item sets are those remaining in L2 and L3.
However, {2,3}, {2,5}, and {3,5} are all contained in the larger item set {2, 3, 5}. Thus, the final group of item sets reported by Apriori are {1,3} and {2,3,5}. These are the only item sets from which we will generate association rules.

Generating Association Rules from Frequent Itemsets

- Only strong association rules are generated
- Frequent itemsets satisfy minimum support threshold
- Strong rules are those that satisfy minimum confidence threshold
- $confidence(A \Longrightarrow B) = \Pr(B \mid A) = \boxed{\dfrac{support(A \cup B)}{support(A)}}$

```
For each frequent itemset, f, generate all non-empty subsets of f
For every non-empty subset s of f do
    if support(f)/support(s) ≥ min_confidence then
        output rule s ==> (f-s)
end
```

Generating Association Rules
(Example Continued)

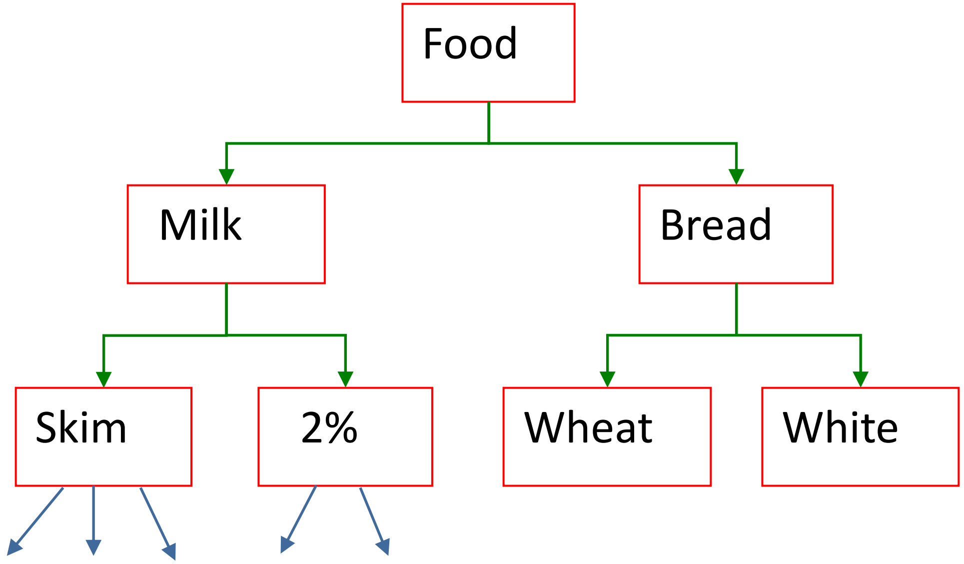
- Item sets: {1,3} and {2,3,5}
- Recall that confidence of a rule LHS → RHS is Support of itemset (i.e. LHS È RHS) divided by support of LHS.

Candidate rules for {1,3}		Candidate rules for {2,3,5}			
Rule	**Conf.**	**Rule**	**Conf.**	**Rule**	**Conf.**
{1}→{3}	2/2 = 1.0	{2,3}→{5}	2/2 = 1.00	{2}→{5}	3/3 = 1.00
{3}→{1}	2/3 = 0.67	{2,5}→{3}	2/3 = 0.67	{2}→{3}	2/3 = 0.67
		{3,5}→{2}	2/2 = 1.00	{3}→{2}	2/3 = 0.67
		{2}→{3,5}	2/3 = 0.67	{3}→{5}	2/3 = 0.67
		{3}→{2,5}	2/3 = 0.67	{5}→{2}	3/3 = 1.00
		{5}→{2,3}	2/3 = 0.67	{5}→{3}	2/3 = 0.67

Assuming a min. confidence of 75%, the final set of rules reported by Apriori are: {1}→{3}, {3,5}→{2}, {5}→{2} and {2}→{5}

Extension: Multiple-Level Rules

- Items often form a hierarchy
  - Items at the lower level are expected to have lower support
  - Rules regarding itemsets at appropriate levels could be quite useful
  - Transaction database can be encoded based on dimensions and levels



Pros: find finer-grained rules
Cons: support may be low

Extension: Quantitative Rules

RecordID	Age	Married	NumCars
100	23	No	1
200	25	Yes	1
300	29	No	0
400	34	Yes	2
500	38	Yes	2

Sample Rules	Support	Confidence
<age:30..39> and <married: yes> ==> <numCars:2>	40%	100%
<NumCars: 0..1> ==> <Married: No>	40%	66.70%

Handling quantitative rules may require mapping of the continuous variables into Boolean or categorical ones

Schedule

- Outlier Detection
- Associate Rule Mining
  - Application: Web Usage Mining

What is Web Mining

- From its very beginning, the potential of extracting valuable knowledge from the Web has been quite evident
  – Web mining is the collection of technologies to fulfill this potential.

Web Mining Definition

application of data mining and machine learning techniques to extract useful knowledge from the content, structure, and usage of Web resources.

Types of Web Mining

Web Mining

Web Content Mining

Applications:
- document clustering or categorization
- topic identification / tracking
- concept discovery
- focused crawling
- content-based personalization
- intelligent search tools

Web Usage Mining

Applications:
- user and customer behavior modeling
- Web site optimization
- e-customer relationship management
- Web marketing
- targeted advertising
- recommender systems

Web Structure Mining

Applications:
- document retrieval and ranking (e.g., Google)
- discovery of “hubs” and “authorities”
- discovery of Web communities
- social network analysis

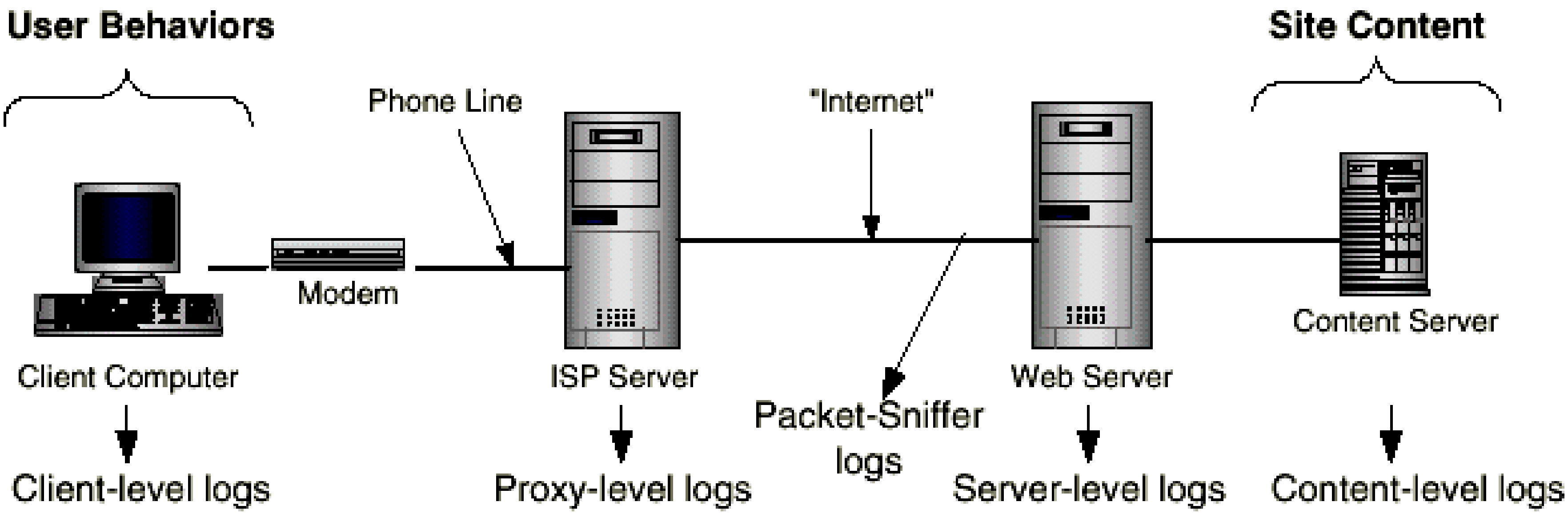
Web Usage Mining

- Web Logs
- Usage Data Preprocessing
  - Data Cleaning
  - User/Session Identification
  - Page View Identification
  - Path Completion
- Web Mining by Association Rules
  - Web Association Mining
  - Web Sequential Mining

Web Usage Mining

- Web Logs
- Usage Data Preprocessing
  - Data Cleaning
  - User/Session Identification
  - Page View Identification
  - Path Completion
- Web Mining by Association Rules
  - Web Association Mining
  - Web Sequential Mining

Simplified Web Access Layout



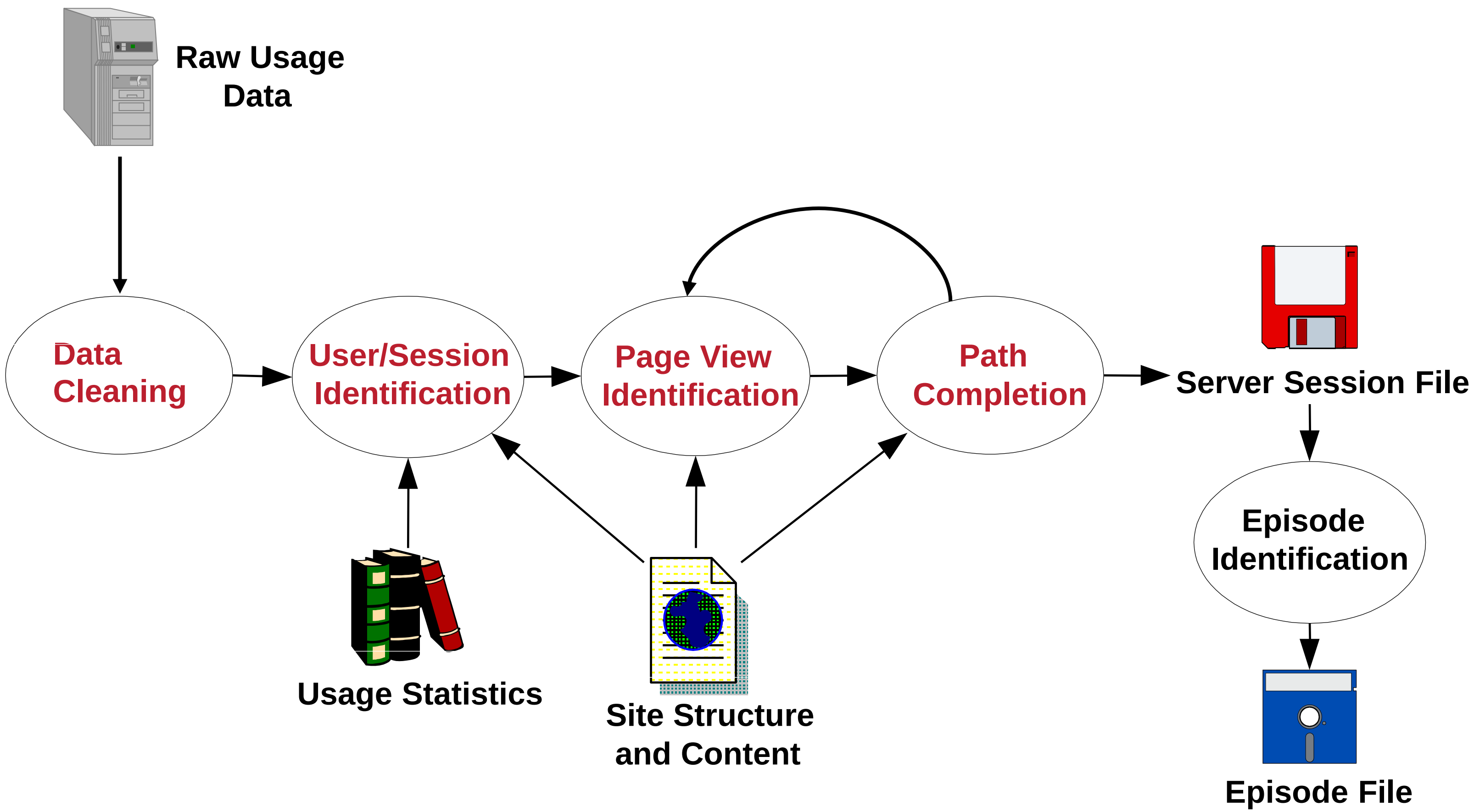
What’s in a Typical Server Log?

1	2006-02-01 00:08:43 1.2.3.4 - GET /classes/cs589/papers.html - 200 9221 HTTP/1.1 maya.cs.depaul.edu Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1;+SV1;+.NET+CLR+2.0.50727) http://dataminingresources.blogspot.com/
2	2006-02-01 00:08:46 1.2.3.4 - GET /classes/cs589/papers/cms-tai.pdf - 200 4096 HTTP/1.1 maya.cs.depaul.edu Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1;+SV1;+.NET+CLR+2.0.50727) http://maya.cs.depaul.edu/~classes/cs589/papers.html
3	2006-02-01 08:01:28 2.3.4.5 - GET /classes/ds575/papers/hyperlink.pdf - 200 318814 HTTP/1.1 maya.cs.depaul.edu Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1) http://www.google.com/search?hl=en&lr=&q=hyperlink+analysis+for+the+web+survey
4	2006-02-02 19:34:45 3.4.5.6 - GET /classes/cs480/announce.html - 200 3794 HTTP/1.1 maya.cs.depaul.edu Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1;+SV1) http://maya.cs.depaul.edu/~classes/cs480/
5	2006-02-02 19:34:45 3.4.5.6 - GET /classes/cs480/styles2.css - 200 1636 HTTP/1.1 maya.cs.depaul.edu Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1;+SV1) http://maya.cs.depaul.edu/~classes/cs480/announce.html
6	2006-02-02 19:34:45 3.4.5.6 - GET /classes/cs480/header.gif - 200 6027 HTTP/1.1 maya.cs.depaul.edu Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1;+SV1) http://maya.cs.depaul.edu/~classes/cs480/announce.html

Web Usage Mining

- Web Logs
- Usage Data Preprocessing
  - Data Cleaning
  - User/Session Identification
  - Page View Identification
  - Path Completion
- Web Mining by Association Rules
  - Web Association Mining
  - Web Sequential Mining

Usage Data Preprocessing



Example

	IP	Time	URL	Referrer	Agent
1	www.aol.com	08:30:00	A	#	Mozilla/5.0; Win NT
2	www.aol.com	08:30:01	B	E	Mozilla/5.0; Win NT
3	www.aol.com	08:30:01	C	B	Mozilla/5.0; Win NT
4	www.aol.com	08:30:02	B	#	Mozilla/5.0; Win 95
5	www.aol.com	08:30:03	C	B	Mozilla/5.0; Win 95
6	www.aol.com	08:30:04	F	#	Mozilla/5.0; Win 95
7	www.aol.com	08:30:04	B	A	Mozilla/5.0; Win NT
8	www.aol.com	08:30:05	G	B	Mozilla/5.0; Win NT

Two major challenges in PreProcessing

- Identification of Users
  - Log data have mixed info of users and transactions
  - Some times, a user may not login the system
- Identification of Sessions
  - A user may visit a same site for several times
  - A user may leave the computer for a while
  - User may have different intents in different sessions

Mechanisms for User Identification

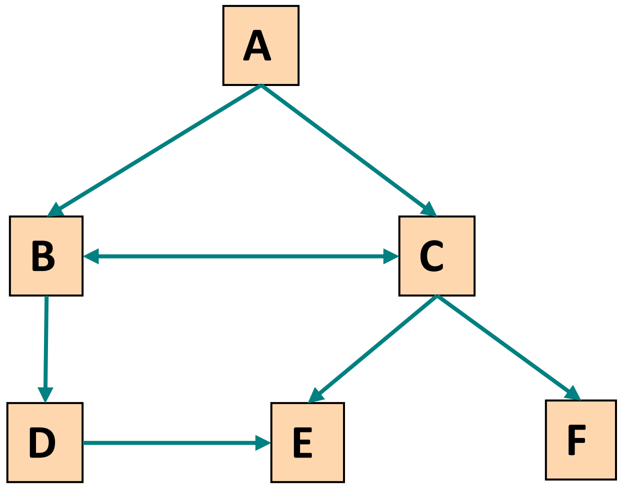
Method	Description	Privacy Concern	Advantages	Disadvantages
IP Address & Agent	Assume each unique IP address/Agent pair is a unique user.	Low	Always available. No additional technology required.	Not guaranteed to be unique. Defeated by random or rotating IP.
Embedded Session ID	Use dynamically generated pages to insert ID into every link.	Low/ Medium	Always available. Independent of IP address.	No concept of a repeat visit. Requires fully dynamic site.
Registration	Users explicitly sign-in to site.	Medium	Can track single individuals, not just browsers.	Not all users may be willing to register.
Cookie	Save an identifier on the client machine	Medium/ High	Can track repeat visits.	Can be disabled. Negative public image.
Software Agent	Program loaded into browser that sends back usage data.	High	Accurate usage data for a single Web site.	Likely to be refused. Negative public image.
Modified Browser	Browser records usage data.	Very High	Accurate usage data across entire Web	Users must explicitly ask for software.

Sessionization Heuristics

- Time-Oriented Heuristics:
  - ***h1***: Total session duration may not exceed a threshold θ. Given t_0, the timestamp for the first request in a constructed session *S*, the request with timestamp *t* is assigned to *S*, iff $t - t_0 \leq \theta$.
  - ***h2***: Total time spent on a page may not exceed a threshold δ. Given t_1, the timestamp for request assigned to constructed session *S*, the next request with timestamp t_2 is assigned to *S*, iff $t_2 - t_1 \leq \delta$.
- Referrer-Based Heuristic:
  - ***href***: Given two consecutive requests *p* and *q*, with *p* belonging to constructed session *S*. Then *q* is assigned to *S*, if the referrer for *q* was previously invoked in *S*.

Note: in practice, it is often useful to use a combination of time- and navigation-oriented heuristics in session identification.

Sessionization Example



Time	IP	URL	Ref	Agent
0:01	1.2.3.4	A	-	IE5;Win2k
0:09	1.2.3.4	B	A	IE5;Win2k
0:10	2.3.4.5	C	-	IE4;Win98
0:12	2.3.4.5	B	C	IE4;Win98
0:15	2.3.4.5	E	C	IE4;Win98
0:19	1.2.3.4	C	A	IE5;Win2k
0:22	2.3.4.5	D	B	IE4;Win98
0:22	1.2.3.4	A	-	IE4;Win98
0:25	1.2.3.4	E	C	IE5;Win2k
0:25	1.2.3.4	C	A	IE4;Win98
0:33	1.2.3.4	B	C	IE4;Win98
0:58	1.2.3.4	D	B	IE4;Win98
1:10	1.2.3.4	E	D	IE4;Win98
1:15	1.2.3.4	A	-	IE5;Win2k
1:16	1.2.3.4	C	A	IE5;Win2k
1:17	1.2.3.4	F	C	IE4;Win98
1:25	1.2.3.4	F	C	IE5;Win2k
1:30	1.2.3.4	B	A	IE5;Win2k
1:36	1.2.3.4	D	B	IE5;Win2k

Sessionization Example

1. Sort users (based on IP+Agent)

Time	IP	URL	Ref	Agent
0:01	1.2.3.4	A	-	IE5;Win2k
0:09	1.2.3.4	B	A	IE5;Win2k
0:10	2.3.4.5	C	-	IE4;Win98
0:12	2.3.4.5	B	C	IE4;Win98
0:15	2.3.4.5	E	C	IE4;Win98
0:19	1.2.3.4	C	A	IE5;Win2k
0:22	2.3.4.5	D	B	IE4;Win98
0:22	1.2.3.4	A	-	IE4;Win98
0:25	1.2.3.4	E	C	IE5;Win2k
0:25	1.2.3.4	C	A	IE4;Win98
0:33	1.2.3.4	B	C	IE4;Win98
0:58	1.2.3.4	D	B	IE4;Win98
1:10	1.2.3.4	E	D	IE4;Win98
1:15	1.2.3.4	A	-	IE5;Win2k
1:16	1.2.3.4	C	A	IE5;Win2k
1:17	1.2.3.4	F	C	IE4;Win98
1:26	1.2.3.4	F	C	IE5;Win2k
1:30	1.2.3.4	B	A	IE5;Win2k
1:36	1.2.3.4	D	B	IE5;Win2k

0:01	1.2.3.4	A	-	IE5;Win2k
0:09	1.2.3.4	B	A	IE5;Win2k
0:19	1.2.3.4	C	A	IE5;Win2k
0:25	1.2.3.4	E	C	IE5;Win2k
1:15	1.2.3.4	A	-	IE5;Win2k
1:26	1.2.3.4	F	C	IE5;Win2k
1:30	1.2.3.4	B	A	IE5;Win2k
1:36	1.2.3.4	D	B	IE5;Win2k

0:10	2.3.4.5	C	-	IE4;Win98
0:12	2.3.4.5	B	C	IE4;Win98
0:15	2.3.4.5	E	C	IE4;Win98
0:22	2.3.4.5	D	B	IE4;Win98

0:22	1.2.3.4	A	-	IE4;Win98
0:25	1.2.3.4	C	A	IE4;Win98
0:33	1.2.3.4	B	C	IE4;Win98
0:58	1.2.3.4	D	B	IE4;Win98
1:10	1.2.3.4	E	D	IE4;Win98
1:17	1.2.3.4	F	C	IE4;Win98

Sessionization Example

2. Sessionize using heuristics (h1: total duration)

0:01	1.2.3.4	A	-	IE5;Win2k
0:09	1.2.3.4	B	A	IE5;Win2k
0:19	1.2.3.4	C	A	IE5;Win2k
0:25	1.2.3.4	E	C	IE5;Win2k
1:15	1.2.3.4	A	-	IE5;Win2k
1:26	1.2.3.4	F	C	IE5;Win2k
1:30	1.2.3.4	B	A	IE5;Win2k
1:36	1.2.3.4	D	B	IE5;Win2k

0:01	1.2.3.4	A	-	IE5;Win2k
0:09	1.2.3.4	B	A	IE5;Win2k
0:19	1.2.3.4	C	A	IE5;Win2k
0:25	1.2.3.4	E	C	IE5;Win2k

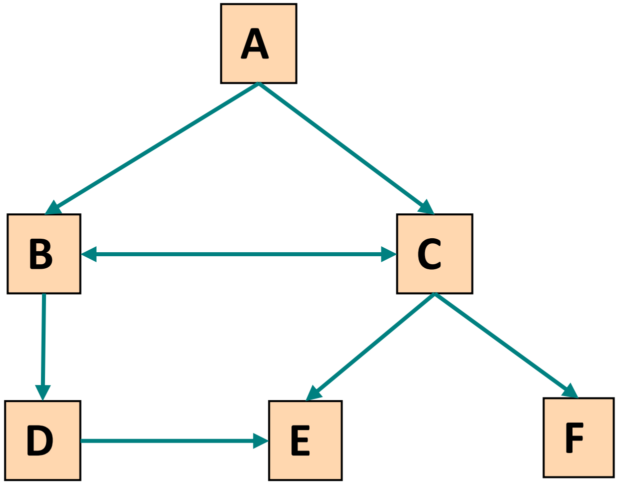
1:15	1.2.3.4	A	-	IE5;Win2k
1:26	1.2.3.4	F	C	IE5;Win2k
1:30	1.2.3.4	B	A	IE5;Win2k
1:36	1.2.3.4	D	B	IE5;Win2k

The *h*1 heuristic (with timeout variable of 30 minutes) will result in the two sessions given above.

How about the heuristic *href*?
How about heuristic *h*2 with a timeout variable of 10 minutes?

Sessionization Example

2. Sessionize using heuristics (h2: duration on each page)



0:22	1.2.3.4	A	-	IE4;Win98
0:25	1.2.3.4	C	A	IE4;Win98
0:33	1.2.3.4	B	C	IE4;Win98
0:58	1.2.3.4	D	B	IE4;Win98
1:10	1.2.3.4	E	D	IE4;Win98
1:17	1.2.3.4	F	C	IE4;Win98

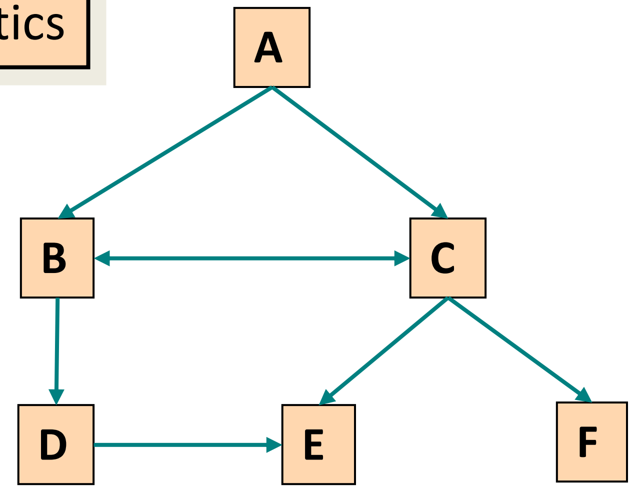
In this case, the referrer-based heuristics will result in a single session, while the *h*1 heuristic (with timeout = 30 minutes) will result in two different sessions.

How about heuristic *h*2 with timeout = 10 minutes?

Sessionization Example

3. Referrer-Based Heuristics

0:22	1.2.3.4	A	-	IE4;Win98
0:25	1.2.3.4	C	A	IE4;Win98
0:33	1.2.3.4	B	C	IE4;Win98
0:58	1.2.3.4	D	B	IE4;Win98
1:10	1.2.3.4	E	D	IE4;Win98
1:17	1.2.3.4	F	C	IE4;Win98



A=>C , C=>B , B=>D , D=>E , C=>F

Path completion

Need to look for the shortest backwards path from E to C based on the site topology. Note, however, that the elements of the path need to have occurred in the user trail previously.

E=>D, D=>B, B=>C

Therefore, there is only 1 session

Web Usage Mining

- Web Logs
- Usage Data Preprocessing
  - Data Cleaning
  - User/Session Identification
  - Page View Identification
  - Path Completion
- Web Mining by Association Rules
  - Web Association Mining
  - Web Sequential Mining

Market Analysis vs Web Mining

- Market Analysis
  - We explore associations among items in transactional databases
  - Items may show up together in different transactions, such as each receipt
- Web Mining
  - We can explore the associations among Web pages or behaviors in Web logs
  - Web pages or behaviors may show up together in different sessions

Web Usage Mining by Association Rules

- Web Association Rule Mining
  - The process is similar to association rule mining, but you need to apply the rule mining per sessions
  - Examples
    - 60% of clients who accessed **`/products/`**, also accessed **`/products/software/webminer.htm`**
    - 30% of clients who accessed **`/special-offer.html`**, placed an online order in **`/products/software/`**
- Web Sequential Mining
  - In association rule mining, the sequence does not matter. But on the Web, the sequence takes a key role. For example, {A->B->C} -> {D} may be very different from {B->A->C} -> {D}
  - The process is similar to the association rule mining, but you need to consider sequences when you calculate support and confidence values

Web Log Data

If you'd like to work on Web mining…

- NASA Web Logs, http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html
- Wikipedia Web Logs, http://opensource.indeedeng.io/imhotep/docs/sample-data/
- MSNBC.com Web Data, http://archive.ics.uci.edu/ml/datasets/MSNBC.com+Anonymous+Web+Data
- Microsoft Web Data, http://archive.ics.uci.edu/ml/datasets/Anonymous+Microsoft+Web+Data
- DePaul CTI Web Logs, http://facweb.cs.depaul.edu/mobasher/classes/ect584/lectures/cti-april2003-clean-log.zip