Supervised and Unsupervised Learning (II)

Yong Zheng
Center for Web Intelligence
DePaul University, Chicago

IPD 346 - Data Science for Business Program
DePaul University, Chicago, USA
• **Intro: Supervised and Unsupervised Learning**
  - Example: Categorization
  - What is Supervised/Unsupervised Learning
  - More Examples/Applications
• **Part I. Supervised Learning**
  - Classification Tasks and Standard Workflow
  - Popular Classification Algorithms
  - K-Nearest Neighbor (KNN) Classifier
  - Naïve Bayes
  - Decision Trees
• **Part II. Unsupervised Learning**
  - Clustering
  - Association Rules
  - Other Tasks and Approaches
• **Data Mining Toolkit: Weka**
Part II
Unsupervised Learning
Supervised/Unsupervised Learning

- **Supervised Learning**: infer a (predictive) function from data associated with pre-defined targets/classes/labels
  
  **Example**: group objects by color: red, blue, green
  
  **Goal**: Learn a model from labelled data for future predictions
  
  **Evaluation**: error/accuracy, and other more metrics

- **Unsupervised Learning**: discover or describe underlying structure/correlations from unlabelled data
  
  **Example**: group objects by shape + color + weight + cost
  
  **Goal**: Learn the structure from unlabelled data
  
  **Evaluation**: No clear performance, but there are some metrics
More Examples/Applications

- **Unsupervised Learning:** discover or describe underlying structure/correlations from unlabelled data
  
  **Example:** group objects by shape + color + weight + cost
  
  **Goal:** Learn the structure from unlabelled data
  
  **Evaluation:** No clear performance, but there are some metrics

**Examples/Applications**

- Topic detection from a list of articles/reviews/tags, etc
- Build a category tree: categories and subcategories
- Association analysis: correlations among products in purchases

**Applications/Approaches Covered in this talk**

- Clustering
- Association Rules
- Intro: other applications
Clustering
Clustering

- **Clustering**: a unsupervised way to group objects

- **Goal**: Finding groups of objects in data such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups

Inter-cluster distances are maximized

Intra-cluster distances are minimized

Inter-cluster distances are maximized
Solutions for Clustering

- **Goal:** Finding groups of objects in data such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups.
- We can reuse the distance measures to calculate instance similarities.
Distance or Similarity Measures

- **Common Distance Measures:**

  - Manhattan distance:
    \[
    dist(X,Y) = |x_1 - y_1| + |x_2 - y_2| + \cdots + |x_n - y_n|
    \]

  - Euclidean distance:
    \[
    dist(X,Y) = \sqrt{(x_1 - y_1)^2 + \cdots + (x_n - y_n)^2}
    \]

  - Cosine similarity:
    \[
    dist(X,Y) = 1 - sim(X,Y)
    \]
    \[
    sim(X,Y) = \frac{\sum_i (x_i \times y_i)}{\sqrt{\sum_i x_i^2} \times \sqrt{\sum_i y_i^2}}
    \]
Clustering Tasks/Approaches

• **Normal Clustering**: just group objects to minimize intra-cluster distances and maximize inter-cluster distances
  Example: Document Categorization without known labels

• **Hierarchical Clustering**: a clustering process in order to discover the hierarchical structure
  Example: categories and subcategories; taxonomies
**K-Means Clustering Algorithm**

- Assume we have many examples/instances, each example can be represented by a vector of features, where the features must be numerical ones, e.g., weight, size, price, profits, etc.

- So that, we can use the distance measures to calculate the similarity or the dissimilarity (i.e., distance) between each two examples.

- With such setting, we are able to apply a K-Means clustering algorithms to perform the normal clustering task.
K-Means Clustering Algorithm

**Init:** Users specifies k – number of clusters used to group the data

**Step 1.** Pick k points from the dataset at random -> initial cluster centroids

**Step 2.** Assign the remaining data points to the cluster with the nearest cluster center (based on some similarity or distance function)

**Step 3.** Compute the average point for each cluster -> new cluster center

**Step 4.** Repeat 2,3 until no points move between clusters
K-Means Clustering Algorithm
Example: K-Means

Example: Clustering Documents

Initial (arbitrary) assignment:
C1 = \{D1,D2\},
C2 = \{D3,D4\},
C3 = \{D5,D6\}

Cluster Centroids

<table>
<thead>
<tr>
<th></th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
<th>T5</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>D2</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>D3</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>D4</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>D5</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>D6</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>D7</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>D8</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>C1</td>
<td>4/2</td>
<td>4/2</td>
<td>3/2</td>
<td>1/2</td>
<td>4/2</td>
</tr>
<tr>
<td>C2</td>
<td>0/2</td>
<td>7/2</td>
<td>0/2</td>
<td>3/2</td>
<td>5/2</td>
</tr>
<tr>
<td>C3</td>
<td>2/2</td>
<td>3/2</td>
<td>3/2</td>
<td>0/2</td>
<td>5/2</td>
</tr>
</tbody>
</table>
Now compute the similarity (or distance) of each item with each cluster, resulting a cluster-document similarity matrix (here we use dot product as the similarity measure).

For each document, reallocate the document to the cluster to which it has the highest similarity (shown in red in the above table). After the reallocation we have the following new clusters. Note that the previously unassigned D7 and D8 have been assigned, and that D1 and D6 have been reallocated from their original assignment.

\[
\begin{align*}
C1 &= \{D2,D7,D8\}, & C2 &= \{D1,D3,D4,D6\}, & C3 &= \{D5\}
\end{align*}
\]

This is the end of first iteration (i.e., the first reallocation). Next, we repeat the process for another reallocation…
Now compute new cluster centroids using the original document-term matrix.

This will lead to a new cluster-doc similarity matrix similar to previous slide. Again, the items are reallocated to clusters with highest similarity.

New assignment

<table>
<thead>
<tr>
<th></th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
<th>T5</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>D2</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>D3</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>D4</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>D5</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>D6</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>D7</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>D8</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

C1 = \{D2,D7,D8\},  \ C2 = \{D1,D3,D4,D6\},  \ C3 = \{D5\}

Note: This process is now repeated with new clusters. However, the next iteration in this example will show no change to the clusters, thus terminating the algorithm.
K-Means Clustering Algorithm

• **Strength of the *k*-means:**
  - Relatively efficient: $O(tkn)$, where $n$ is # of objects, $k$ is # of clusters, and $t$ is # of iterations. Normally, $k, t << n$
  - Often terminates at a *local optimum*

• **Weakness of the *k*-means:**
  - What about categorical data?
  - Performance is sensitive to initializations, e.g., K and centroids
  - Need to specify $k$, the *number* of clusters, in advance
  - Unable to handle noisy data and *outliers*

• **Variations of K-Means usually differ in:**
  - Selection of the initial $k$ means
  - Dissimilarity calculations
  - Strategies to calculate cluster means
Hierarchical Clustering

• Similarly, we can use distance measures to do hierarchical clustering
Hierarchical Agglomerative Clustering

Dendrogram for a hierarchy of clusters
Association Rules
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
What Is Association Mining?

- **Association rule mining searches for relationships between items:**
  - Finding association, correlation, or causal structures among sets of items or objects in transaction databases, relational databases, etc.

- **It can be viewed as unsupervised learning, since**
  - We do not know what kinds of rules can be found
  - We want to discover useful rules as a process of pattern discovery

- **Examples: [support, confidence]**
  - \{diaper, milk, Thursday\} $\Rightarrow$ \{beer\} [0.5%, 78%]
  - buys(x, "bread") $\Rightarrow$ buys(x, "milk") [0.6%, 65%]
  - major(x, "CS") $\land$ takes(x, "DB") $\Rightarrow$ grade(x, "A") [1%, 75%]
  - age(X,30-45) $\land$ income(X, 50K-75K) $\Rightarrow$ buys(X, SUVcar)
  - age="30-45", income="50K-75K" $\Rightarrow$ car="SUV"
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 $I$): $T \subseteq I$

- An Association Rule is an implication on itemsets $X$ and $Y$, denoted by $X \rightarrow Y$, where
  - $X \subseteq I$, $Y \subseteq I$, $X \cap Y = \emptyset$

- The rule meets a minimum confidence of $c$, meaning that $c\%$ of transactions in $D$ which contain $X$ also contain $Y$
  - $c \geq \frac{|X \cup Y|}{|X|}$

- In addition a minimum support of $s$ is satisfied
  - $s \geq \frac{|X \cup Y|}{|D|}$
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 in database.
  - **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 $LHS \Rightarrow RHS$ can be computed as the support of the whole itemset divided by the support of LHS:

$$\text{Confidence (LHS } \Rightarrow \text{ RHS)} = \frac{\text{Support}(LHS \cup RHS)}{\text{Support}(LHS)}$$
Improvement (Lift)

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

\[
\text{improvement} = \frac{\Pr(\text{result} \mid \text{condition})}{\Pr(\text{result})} = \frac{\text{confidence}(\text{rule})}{\text{support}(\text{result})}
\]

<table>
<thead>
<tr>
<th>Transaction ID</th>
<th>Items Bought</th>
</tr>
</thead>
<tbody>
<tr>
<td>1001</td>
<td>A, B, C</td>
</tr>
<tr>
<td>1002</td>
<td>A, C</td>
</tr>
<tr>
<td>1003</td>
<td>A, D</td>
</tr>
<tr>
<td>1004</td>
<td>B, E, F</td>
</tr>
<tr>
<td>1005</td>
<td>A, D, F</td>
</tr>
</tbody>
</table>

Itemset \{A\} has a support of 4/5
Rule \{C\} \implies \{A\} has confidence of 2/2
Improvement = 5/4 = 1.25

Itemset \{A\} has a support of 4/5
Rule \{B\} \implies \{A\} has confidence of 1/2
Improvement = 5/8 = 0.625
Mining Association Rules - An Example

Frequent itemset is defined as a set of items with minimum support

<table>
<thead>
<tr>
<th>Transaction ID</th>
<th>Items Bought</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>A,B,C</td>
</tr>
<tr>
<td>1000</td>
<td>A,C</td>
</tr>
<tr>
<td>4000</td>
<td>A,D</td>
</tr>
<tr>
<td>5000</td>
<td>B,E,F</td>
</tr>
</tbody>
</table>

Min. support 50%
Min. confidence 50%

Only need to keep these since {A} and {C} are subsets of {A,C}

Frequent Itemset | Support |
-----------------|---------|
{A}              | 75%     |
{B}              | 50%     |
{C}              | 50%     |
{A,C}            | 50%     |

For rule $A \Rightarrow C$:

support = support($\{A, C\}$) = 50%
confidence = support($\{A, C\}$)/support($\{A\}$) = 66.6%
Apriori Algorithm

$C_k$: Candidate itemset of size $k$
$L_k$: Frequent itemset of size $k$

$L_1 = \{\text{frequent items}\}$

for $(k = 1; L_k \neq \emptyset; k++)$ 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 $t$
    end
    $L_{k+1}$ = candidates in $C_{k+1}$ with min_support
end
return $\bigcup_k L_k$

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
Apriori Algorithm - An Example

Assume minimum support = 2

Database D

<table>
<thead>
<tr>
<th>TID</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>1 3 4</td>
</tr>
<tr>
<td>200</td>
<td>2 3 5</td>
</tr>
<tr>
<td>300</td>
<td>1 2 3 5</td>
</tr>
<tr>
<td>400</td>
<td>2 5</td>
</tr>
</tbody>
</table>

C₁

<table>
<thead>
<tr>
<th>item set</th>
<th>sup.</th>
</tr>
</thead>
<tbody>
<tr>
<td>{1}</td>
<td>2</td>
</tr>
<tr>
<td>{2}</td>
<td>3</td>
</tr>
<tr>
<td>{3}</td>
<td>3</td>
</tr>
<tr>
<td>{4}</td>
<td>1</td>
</tr>
<tr>
<td>{5}</td>
<td>3</td>
</tr>
</tbody>
</table>

Scan D

L₁

<table>
<thead>
<tr>
<th>item set</th>
<th>sup.</th>
</tr>
</thead>
<tbody>
<tr>
<td>{1}</td>
<td>2</td>
</tr>
<tr>
<td>{2}</td>
<td>3</td>
</tr>
<tr>
<td>{3}</td>
<td>3</td>
</tr>
<tr>
<td>{5}</td>
<td>3</td>
</tr>
</tbody>
</table>

C₂

<table>
<thead>
<tr>
<th>item set</th>
<th>sup.</th>
</tr>
</thead>
<tbody>
<tr>
<td>{1 2}</td>
<td>1</td>
</tr>
<tr>
<td>{1 3}</td>
<td>2</td>
</tr>
<tr>
<td>{1 5}</td>
<td>1</td>
</tr>
<tr>
<td>{2 3}</td>
<td>2</td>
</tr>
<tr>
<td>{2 5}</td>
<td>3</td>
</tr>
<tr>
<td>{3 5}</td>
<td>2</td>
</tr>
</tbody>
</table>

Scan D

L₂

<table>
<thead>
<tr>
<th>item set</th>
</tr>
</thead>
<tbody>
<tr>
<td>{1 2}</td>
</tr>
<tr>
<td>{1 3}</td>
</tr>
<tr>
<td>{1 5}</td>
</tr>
<tr>
<td>{2 3}</td>
</tr>
<tr>
<td>{2 5}</td>
</tr>
<tr>
<td>{3 5}</td>
</tr>
</tbody>
</table>

C₃

<table>
<thead>
<tr>
<th>item set</th>
</tr>
</thead>
<tbody>
<tr>
<td>{2 3 5}</td>
</tr>
</tbody>
</table>

Scan D

L₃

<table>
<thead>
<tr>
<th>item set</th>
<th>sup.</th>
</tr>
</thead>
<tbody>
<tr>
<td>{2 3 5}</td>
<td>2</td>
</tr>
</tbody>
</table>

Note: {1,2,3} {1,2,5} and {1,3,5} not in C₃
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

• \( \text{confidence}(A \implies B) = \Pr(B \mid A) = \frac{\text{support}(A \cup B)}{\text{support}(A)} \)

For each frequent itemset, \( f \), generate all non-empty subsets of \( f \)
For every non-empty subset \( s \) of \( f \) do
  if \( \text{support}(f)/\text{support}(s) \geq \text{min} \_ \text{confidence} \) then
    output rule \( s \implies (f-s) \)
end
Generating Association Rules
(Example Continued)

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

<table>
<thead>
<tr>
<th>Candidate rules for ({1,3})</th>
<th>Candidate rules for ({2,3,5})</th>
</tr>
</thead>
<tbody>
<tr>
<td>({1} \rightarrow {3})</td>
<td>(\frac{2}{2} = 1.0)</td>
</tr>
<tr>
<td>({3} \rightarrow {1})</td>
<td>(\frac{2}{3} = 0.67)</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Assuming a min. confidence of 75%, the final set of rules reported by Apriori are: \(\{1\} \rightarrow \{3\}\), \(\{3,5\} \rightarrow \{2\}\), \(\{5\} \rightarrow \{2\}\) and \(\{2\} \rightarrow \{5\}\)
Other Applications/Approaches
Supervised/Unsupervised Learning

• **Supervised Learning:** infer a (predictive) function from data associated with pre-defined targets/classes/labels
  Example: group objects by color: red, blue, green
  Goal: Learn a model from labelled data for future predictions
  Evaluation: error/accuracy, and other more metrics

• **Unsupervised Learning:** discover or describe underlying structure/correlations from unlabelled data
  Example: group objects by shape + color + weight + cost
  Goal: Learn the structure from unlabelled data
  Evaluation: No clear performance, but there are some metrics
Supervised/Unsupervised Learning

• **Application: grouping objects**
  Supervised Learning: classification
  Unsupervised Learning: clustering

• **Application: feature selection/reduction**
  Supervised Learning: Linear discriminant analysis (LDA)
  Unsupervised Learning: Principal component analysis (PCA)
Feature Selection/Reduction

• Application: feature selection/reduction
  Assume there is a large data for the classification task
  We have 1000 features and only 10 labels
  Not all the 1000 features are relevant or influential
  We need to find the best set of relevant features to perform task

• Solutions
  Supervised Learning: Linear discriminant analysis (LDA)
  Unsupervised Learning: Principal component analysis (PCA)

  Same goal: try to find a subset of features which can help capture
  the largest variance of the original data
  Difference: LDA tries to find features to distinguish examples from
  different labels; PCA does not take labels into consideration
Data Mining Toolkit
Weka
Weka

- Weka is a collection of algorithms for data mining tasks.
- It is a Java-based open source toolkit
- You can obtain it from http://www.cs.waikato.ac.nz/ml/weka/
- Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization.
Short notes on using Weka

• More details: Lab_An Introduction to WEKA_By Yizhou Sun.ppt
• But, here are a list of notes for your lab practice:
  □ The input file to Weka can be either csv or arff, normally;
  □ The KNN classifier is named as IBK in Weka;
    You can use J48 as decision tree classifier in Weka;
  □ Some classifiers require feature normalization, you may perform preprocessing before using it;
  □ Other algorithms may require data transformation, such as numerical data to categorical data, etc, you should be careful;
  □ In terms of evaluations, we suggest to use 10-folds cross evaluation, unless the data assigns you the training and testing sets.
Supervised and Unsupervised Learning

Yong Zheng
DePaul University,
Chicago, USA