Big Data Machine Learning: Mahout

Thanks to Mohamed Eltabakh
Data Analytics

• Include machine learning and data mining tools
  – Analyze/mine/summarize large datasets
  – Extract knowledge from past data
  – Predict trends in future data
Data Mining & Machine Learning

• Subset of Artificial Intelligence (AI)
• Lots of related fields and applications
  – Information Retrieval
  – Stats
  – Biology
  – Linear algebra
  – Marketing and Sales
Tools & Algorithms

• Collaborative Filtering
• Clustering Techniques
• Classification Algorithms
• Association Rules
• Frequent Pattern Mining
• Statistical libraries (Regression, SVM, ...)
• Others...
Common Use Cases

- Recommend friends/dates/products
- Classify content into predefined groups
- Find similar content
- Find associations/patterns in actions/behaviors
- Identify key topics/summarize text
  - Documents and Corpora
- Detect anomalies/fraud
- Ranking search results
- Others?
In Our Context...

--Efficient in analyzing/mining data
--Do not scale

--Efficient in managing big data
--Does not analyze or mine the data

How to integrate these two worlds together
On Going Research Effort

Ricardo (VLDB’10): Integrating Hadoop and R using Jaql

Halooop (SIGMOD’10): Supporting iterative processing in Hadoop
Other Projects

• **Apache Mahout**
  – Open-source package on Hadoop for data mining and machine learning

• **Revolution R (R-Hadoop)**
  – Extensions to R package to run Hadoop
Apache Mahout
Apache Mahout

• Apache Software Foundation project
• Create scalable machine learning libraries
• **Why Mahout?** Many Open Source ML libraries either:
  – Lack Community
  – Lack Documentation and Examples
  – Lack Scalability
  – Or are research-oriented
Goal 1: Machine Learning
Goal 2: Scalability

- Be as fast and efficient as the possible given the intrinsic design of the algorithm
- Most Mahout implementations are Map Reduce enabled
- Work in Progress
Mahout Package

What Can I do with Mahout Right Now?

https://cwiki.apache.org/MAHOUT/algorithms.html

3C + FPM + O = Mahout

C1: Collaborative Filtering / Recommenders
C2: Clustering
C3: Classification

FPM: Frequent Pattern Mining
Others
C1: Collaborative Filtering

- Extensive framework for collaborative filtering (recommenders)
- Recommenders
  - User based
  - Item based
- Online and Offline support
  - Offline can utilize Hadoop
- Many different Similarity measures
  - Cosine, LLR, Tanimoto, Pearson, others
C2: Clustering

• Group similar objects together

• K-Means, Fuzzy K-Means, Density-Based,…

• Different distance measures
  – Manhattan, Euclidean, …
C3: Classification

• Place new items into predefined categories:
  – Sports, politics, entertainment
  – Recommenders

• Implementations
  – Naïve Bayes (M/R)
  – Compl. Naïve Bayes (M/R)
  – Decision Forests (M/R)
  – Linear Regression (Seq. but Fast!)
FPM: Frequent Pattern Mining

- **Find the frequent itemsets**
  - \(<\text{milk, bread, cheese}>\) are sold frequently together

- **Very common in market analysis, access pattern analysis, etc...**
O: Others

• Outlier detection
• Math library
  – Vectors, matrices, etc.
• Noise reduction
Today We Focus On...

- **Clustering** ➔ K-Means
- **Classification** ➔ Naïve Bayes
- **Frequent Pattern Mining** ➔ Apriori

---

- Technique logic
- How to implement in Hadoop
K-Means Algorithm

Demonstration of the standard algorithm

1) k initial "means" (in this case k=3) are randomly selected from the data set (shown in color).
2) k clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means.
3) The centroid of each of the k clusters becomes the new means.
4) Steps 2 and 3 are repeated until convergence has been reached.

Iterative algorithm until converges
K-Means Algorithm

• **Step 1:** Select K points at random (Centers)
• **Step 2:** For each data point, assign it to the closest center
  – Now we formed K clusters
• **Step 3:** For each cluster, re-compute the centers
  – E.g., in the case of 2D points ➔
    • X: average over all x-axis points in the cluster
    • Y: average over all y-axis points in the cluster
• **Step 4:** If the new centers are different from the old centers (previous iteration) ➔ Go to Step 2
K-Means in MapReduce

• **Input**
  – Dataset (set of points in 2D) --Large
  – Initial centroids (K points) --Small

• **Map Side**
  – Each map reads the K-centroids + one block from dataset
  – Assign each point to the closest centroid
  – Output <centroid, point>
K-Means in MapReduce (Cont’d)

• **Reduce Side**
  – Gets all points for a given centroid
  – Re-compute a new centroid for this cluster
  – Output: <new centroid>

• **Iteration Control**
  – Compare the old and new set of K-centroids
    • If similar ➔ Stop
    • Else
      – If max iterations has reached ➔ Stop
      – Else ➔ Start another Map-Reduce Iteration
K-Means Optimizations

- **Use of Combiners**
  - Similar to the reducer
  - Computes for each centroid the local sums (and counts) of the assigned points
  - Sends to the reducer <centroid, <partial sums>>

- **Use of Single Reducer**
  - Amount of data to reducers is very small
  - Single reducer can tell whether any of the centers has changed or not
  - Creates a single output file
Naïve Bayes Classifier

• Given a dataset (training data), we learn (build) a statistical model
  – This model is called “Classifier”

• Each point in the training data is in the form of:
  – <label, feature 1, feature 2, ....feature N>
  – Label ➙ is the class label
  – Features 1..N ➙ the features (dimensions of the point)

• Then, given a point without a label <??, feature 1, ....feature N>
  – Use the model to decide on its label
Naïve Bayes Classifier: Example

- Best described through an example

### Training dataset

<table>
<thead>
<tr>
<th>sex</th>
<th>height (feet)</th>
<th>weight (lbs)</th>
<th>foot size (inches)</th>
</tr>
</thead>
<tbody>
<tr>
<td>male</td>
<td>6</td>
<td>180</td>
<td>12</td>
</tr>
<tr>
<td>male</td>
<td>5.92 (5'11&quot;)</td>
<td>190</td>
<td>11</td>
</tr>
<tr>
<td>male</td>
<td>5.58 (5'7&quot;)</td>
<td>170</td>
<td>12</td>
</tr>
<tr>
<td>male</td>
<td>5.92 (5'11&quot;)</td>
<td>165</td>
<td>10</td>
</tr>
<tr>
<td>female</td>
<td>5</td>
<td>100</td>
<td>6</td>
</tr>
<tr>
<td>female</td>
<td>5.5 (5'6&quot;)</td>
<td>150</td>
<td>8</td>
</tr>
<tr>
<td>female</td>
<td>5.42 (5'5&quot;)</td>
<td>130</td>
<td>7</td>
</tr>
<tr>
<td>female</td>
<td>5.75 (5'9&quot;)</td>
<td>150</td>
<td>9</td>
</tr>
</tbody>
</table>

- **Class label (male or female)**
- **Three features**

26
Naïve Bayes Classifier (Cont’d)

• For each feature in each label
  – Compute the mean and variance

That is the model (classifier)
Naïve Bayes: Classify New Object

- For each label ➔ Compute **posterior** value
- The label with the largest posterior is the suggested label

$$
\text{posterior(male)} = \frac{P(male)p(height|male)p(weight|male)p(footsize|male)}{\text{evidence}}
$$

$$
\text{posterior(female)} = \frac{P(female)p(height|female)p(weight|female)p(footsize|female)}{\text{evidence}}
$$
Naïve Bayes: Classify New Object (Cont’d)

Male or female?

<table>
<thead>
<tr>
<th>sex</th>
<th>height (feet)</th>
<th>weight (lbs)</th>
<th>foot size (inches)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sample</td>
<td>6</td>
<td>130</td>
<td>8</td>
</tr>
</tbody>
</table>

$$\text{posterior(male)} = \frac{\text{P(male)} \text{p(height|male)} \text{p(weight|male)} \text{p(footsize|male)}}{\text{evidence}}$$

$$\text{posterior(female)} = \frac{\text{P(female)} \text{p(height|female)} \text{p(weight|female)} \text{p(footsize|female)}}{\text{evidence}}$$

>> **evidence**: Can be ignored since it is the same constant for all labels

>> **P(label)**: % of training points with this label

>> **Assumption**: continuous values associated with each class are distributed according to a Gaussian distribution

>> $$\text{p(feature|label)=p(f|label)=} = \frac{1}{\sqrt{2\pi \sigma^2}} \exp \left( -\frac{(f - \mu)^2}{2\sigma^2} \right)$$, f is feature value in sample
Naïve Bayes: Classify New Object (Cont’d)

Male or female?

\[
posterior(male) = \frac{P(male) p(\text{height}|male) p(\text{weight}|male) p(\text{foot size}|male)}{\text{evidence}}
\]

\[
P(male) = 0.5
\]
\[
p(\text{height}|male) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left( \frac{-(6 - \mu)^2}{2\sigma^2} \right) \approx 1.5789,
\]
\[
p(\text{weight}|male) = 5.9881e - 06
\]
\[
p(\text{foot size}|male) = 1.3112e - 3
\]
\[
\text{posterior numerator (male)} = \text{their product} = 6.1984e - 09
\]
Naïve Bayes: Classify New Object (Cont’d)

Male or female?

\[\text{posterior} (\text{male}) = \frac{P(\text{male}) p(\text{height}|\text{male}) p(\text{weight}|\text{male}) p(\text{foot size}|\text{male})}{\text{evidence}}\]

\[P(\text{female}) = 0.5\]
\[p(\text{height}|\text{female}) = 2.2346e - 1\]
\[p(\text{weight}|\text{female}) = 1.6789e - 2\]
\[p(\text{foot size}|\text{female}) = 2.8669e - 1\]

posterior numerator (female) = their product = \(5.3778e - 04\)

*The sample is predicted to be female*
Naïve Bayes in Hadoop

• How to implement Naïve Bayes as a map-reduce job?
• Input: Lots of data points to classify
• A training set

Class Discussion
Frequent Pattern Mining

• Very common problem in Market-Basket applications

• Given a set of items \( I = \{\text{milk, bread, jelly, ...}\} \)

• Given a set of transactions where each transaction contains subset of items
  – \( t_1 = \{\text{milk, bread, water}\} \)
  – \( t_2 = \{\text{milk, nuts, butter, rice}\} \)
Frequent Pattern Mining

- Given a set of items $I = \{ \text{milk, bread, jelly, …} \}$
- Given a set of transactions where each transaction contains a subset of items
  - $t_1 = \{ \text{milk, bread, water} \}$
  - $t_2 = \{ \text{milk, nuts, butter, rice} \}$

What are the itemsets frequently sold together??

% of transactions in which the itemset appears $\geq \alpha$
Example

Assume $\alpha = 60\%$, what are the frequent itemsets

- $\{\text{Bread}\} \rightarrow 80\%$
- $\{\text{PeanutButter}\} \rightarrow 60\%$
- $\{\text{Bread, PeanutButter}\} \rightarrow 60\%$

called "Support"
How to find frequent itemsets

• **Naïve Approach**
  – Enumerate all possible itemsets and then count each one

![Diagram of all possible itemsets of size 1, 2, 3, and 4](image)
Can we optimize??

<table>
<thead>
<tr>
<th>Transaction</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>Bread, Jelly, PeanutButter</td>
</tr>
<tr>
<td>$t_2$</td>
<td>Bread, PeanutButter</td>
</tr>
<tr>
<td>$t_3$</td>
<td>Bread, Milk, PeanutButter</td>
</tr>
<tr>
<td>$t_4$</td>
<td>Beer, Bread</td>
</tr>
<tr>
<td>$t_5$</td>
<td>Beer, Milk</td>
</tr>
</tbody>
</table>

Assume $\alpha = 60\%$, what are the frequent itemsets

- $\{\text{Bread}\} \to 80\%$
- $\{\text{PeanutButter}\} \to 60\%$
- $\{\text{Bread, PeanutButter}\} \to 60\%$

called “Support”

**Property**

For itemset $S=\{X, Y, Z, \ldots\}$ of size $n$ to be frequent, all its subsets of size $n-1$ must be frequent as well
Apriori Algorithm

• **Executes in scans (iterations), each scan has two phases**
  – Given a list of candidate itemsets of size n, count their appearance and find frequent ones
  – From the frequent ones generate candidates of size n+1 (*previous property must hold*)
    • All subsets of size n must be frequent to be a candidate
  – Start the algorithm where n =1, then repeat

Use the property reduce the number of itemsets to check
# Apriori Example

<table>
<thead>
<tr>
<th>Transaction</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>Blouse</td>
</tr>
<tr>
<td>$t_2$</td>
<td>Shoes, Skirt, TShirt</td>
</tr>
<tr>
<td>$t_3$</td>
<td>Jeans, TShirt</td>
</tr>
<tr>
<td>$t_4$</td>
<td>Jeans, Shoes, TShirt</td>
</tr>
<tr>
<td>$t_5$</td>
<td>Jeans, Shorts</td>
</tr>
<tr>
<td>$t_6$</td>
<td>Shoes, TShirt</td>
</tr>
<tr>
<td>$t_7$</td>
<td>Jeans, Skirt</td>
</tr>
<tr>
<td>$t_8$</td>
<td>Jeans, Shoes, Shorts, TShirt</td>
</tr>
<tr>
<td>$t_9$</td>
<td>Jeans</td>
</tr>
<tr>
<td>$t_{10}$</td>
<td>Jeans, Shoes, TShirt</td>
</tr>
<tr>
<td>$t_{11}$</td>
<td>TShirt</td>
</tr>
<tr>
<td>$t_{12}$</td>
<td>Blouse, Jeans, Shoes, Skirt, TShirt</td>
</tr>
<tr>
<td>$t_{13}$</td>
<td>Jeans, Shoes, Shorts, TShirt</td>
</tr>
<tr>
<td>$t_{14}$</td>
<td>Shoes, Skirt, TShirt</td>
</tr>
<tr>
<td>$t_{15}$</td>
<td>Jeans, TShirt</td>
</tr>
<tr>
<td>$t_{16}$</td>
<td>Skirt, TShirt</td>
</tr>
<tr>
<td>$t_{17}$</td>
<td>Blouse, Jeans, Skirt</td>
</tr>
<tr>
<td>$t_{18}$</td>
<td>Jeans, Shoes, Shorts, TShirt</td>
</tr>
<tr>
<td>$t_{19}$</td>
<td>Jeans</td>
</tr>
<tr>
<td>$t_{20}$</td>
<td>Jeans, Shoes, Shorts, TShirt</td>
</tr>
</tbody>
</table>
### Apriori Example (Cont’d)

<table>
<thead>
<tr>
<th>Scan</th>
<th>Candidates</th>
<th>Large Itemsets</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{Blouse}, {Jeans}, {Shoes}, {Shorts}, {Skirt}, {TShirt}</td>
<td>{Jeans}, {Shoes}, {Shorts}, {Skirt}, {TShirt}</td>
</tr>
<tr>
<td>4</td>
<td>{Jeans,Shoes,Shorts,TShirt}</td>
<td>{Jeans,Shoes,Shorts,TShirt}</td>
</tr>
<tr>
<td>5</td>
<td>$\emptyset$</td>
<td>$\emptyset$</td>
</tr>
</tbody>
</table>
FPM in Hadoop

• How to implement FMP as map-reduce jobs?
Apache Mahout