Mux-Kmeans: Multiplex Kmeans for Clustering Large-scale Data Set

Chen Li, Yanfeng Zhang, Minghai Jiao, Ge Yu
Northeastern University, China
BIG DATA IN A SINGLE DAY ONLINE

ENOUGH INFORMATION IS CONSUMED TO FILL
168 MILLION DVDS
294bn E-MAILS ARE SENT
MINUTES SPENT ON FACEBOOK 4.7M
2 MILLION BLOG POSTS ARE WRITTEN
VIDEO.UPLOADED TO YOUTUBE 864,000 HRS
MORE IPHONES ARE SOLD THAN BABIES BORN

Clustering

Kmeans
Kmeans

• Kmeans: accept $K$ center patterns and a data set, divide the set into $K$ clusters

• Goal:
  ✓ 1. similar – data patterns in same cluster;
  ✓ 2. dissimilar – data patterns in different clusters.

**Kmeans on MapReduce**

**procedure** MAPPHASE(\(<i, x_i>\))

for \(c_j, j \in \{1, \ldots, k\}\) do

\(j \leftarrow \text{find the closest centroid } c_j \text{ to } x_i, \quad j \in \{1, \ldots, k\}\)

end for

output \(<j, x_i>\)

end procedure

**procedure** REDUCEPHASE(\(<j, [x_i]>\))

\(c_j \leftarrow \text{average } [x_i]\)

output \(<j, c_j>\)

end procedure
Shortcoming of Kmeans

- The result of Kmeans clustering is affected by the value of $K$ and the selection of initial centroids.

Current solution: multiple attempts (in series)
- Start from multiple groups of initial centroids
- Execute multiple kmeans processes, obtain multiple local optimal cluster results
- Pick the one with the highest cluster quality

Efficiency Problem
Mux-Kmeans

• Idea:
  – Execute multiple Kmeans attempts in parallel
  – Share states across different Kmeans processes
  – Terminate “hopeless” attempts in early stage
  – Expand searching scale and try more attempts

• Goal:
  – Guarantee the clustering quality
  – Decrease the runtime
Naive Kmeans VS. Mux-Kmeans
Naive Kmeans VS. Mux-Kmeans
Inside Mux-Kmeans Job

1. **Kmeans clustering**
2. **Evaluate** intermediate result's quality
3. **Prune** some Kmeans attempts with low quality
4. **Incubate** new groups of centroids for next iteration
Kmeans, Evaluate & Prune

- Use Kmeans algorithm to do clustering
  - Get multiple updated centroid groups
- Use Total Within-Cluster Variation (TWCV) to evaluate different centroid groups’ quality.
- Prune x% centroid groups with relatively low quality
Permute

- A preprocessing step before the incubate step
- Aim to find the related centroids between different groups

\[
\text{sim}(c_i, c_j) = \frac{1}{1 + \text{distance}(c_i, c_j)}, \quad \text{sim}(c_i, c_j) \in [0, 1]
\]
Incubate
Random Search within a Definite Scope

1. Star = compute the middle of two related centroids
2. Random search around the star, radius = 2*distance(centroid, start)
Incubate 2: ADGP

Average of Dissimilar Group Pairs
Implementation

Job: Permute
(4)

Permuted Groups

Job: Kmeans & Evaluate
(1)

Points Set

Centroid Groups

New and Survived Groups

Job: Incubate
(3)

Survived Groups

Updated Groups

Job: Prune
(2)
Experiment Setup

• Experiment Environment
  – Amazon EC2: 16 nodes, Ubuntu 12.04, Hadoop 1.4.1
  – Each node: 2 ECUs and 1 CPU; 3.7 GB memory; 410 GB storage; moderate network performance

• Experiment Dataset

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Points</th>
<th>features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bio_train</td>
<td>145751</td>
<td>74</td>
</tr>
<tr>
<td>Netflix</td>
<td>17770</td>
<td>1000</td>
</tr>
<tr>
<td>Lastfm</td>
<td>359330</td>
<td>40</td>
</tr>
</tbody>
</table>
Clustering Quality

Different data set

Lastfm  
Bio_train  
Netflix

$K$ value: 6  
centroid group amount: 8
Clustering Quality

Same data set: Bio_train

※ S: centroid groups amount
Elapsed Time

Lastfm

Bio_train

Netflix
Summary & Future Work

• The Mux-Kmeans algorithm.
  – Idea: execute multiple attempts in parallel, share states across different kmeans processes, terminate “hopeless” attempts in early stage, expand search scale and try more attempts
  – Implementation: deployed on MapReduce
  – Result: better clustering quality and shorter runtime when processing multiple centroid groups

• Future work
  – Different k in different centroid groups
  – Many possible Mux-XXX algorithms (Mux-EM, Mux-FCM, etc.)
Thank you!

QUESTIONS?