Fast Automatic Determination of Cluster Numbers for High Dimensional Big data

Zohreh Safari
Ph.D Proposal

Advised by Dr. Yu Zhuang
Computer Science Department

Zohreh.safari@ttu.edu

April 10, 2021
“Fast Automatic Determination of Cluster Numbers for High Dimensional Big Data

Authors: Zohreh Safari, Khalid T. Mursi, Yu Zhuang

https://dl.acm.org/doi/abs/10.1145/3388142.3388164
INTRODUCTION

Clustering

- Unsupervised learning process to seek insights, and
- Find hidden relationships
The basic idea behind K-Means clustering method is:
To have clusters such that the sum of squared distances of points \(x_i\) to their cluster centroids \(z_k\) is minimized.

\[
SSD = \sum_{k=1}^{K} \sum_{i=1}^{n_k} (||x_i - z_k||)^2
\]
The more compact clusters, the less SSD measures are, which leads to improved clustering quality.

K-Means Algorithm:

Input: data set: $x_1, x_2, ..., x_n$, $K$, and $N$.

1. Initialize $K$ centroids.
2. For $i = 1$ to $N$ do
3. compute distance from $x_i$ to centroid $z_k$, $k = 1, 2, ..., K$,
4. Assign $x_i$ to the closest centroid.
5. End of for loop.
6. Re-calculate centroid for each new cluster.
7. Repeat step 2 till no change of clusters.

____________________________________
PROBLEM

One Challenge:

- **Predefine number of clusters** for any data set.
PROBLEM

- How automatically determine the best K?
RELATED WORK

- Silhottes Gap-test
- IKMEANS
- X-Means
- ISODATA
- Ray and Turi
• The previous methods were built upon K-Means, but
  • They consider only quality, not efficiency,
  • Some of them worked only for data sets with small number of features.
    • Ex: R&T method.

❖ Our goal: Considering both Quality and Efficiency
How Our Method Works?
OUR PROPOSED METHOD

Our method trying to address following issues:

- Predefined number of clusters.
- Randomly chosen of initial centers.
- More number of K-means Executions.

Executing K-Means many times highly affected the rise of time computation, and it also can be sharply increased when initial centers are chosen randomly.

- We used BK-Means in our modified method to have better initial centers.
BK-Means Algorithm:

Input: data set: $x_1, x_2, ..., x_n$, K, and N.
1. Partition whole data set to 2 clusters $C_1$ and $C_2$, num_clusters=2
2. Assume we have clusters $C_1, C_2, ..., C_k$, where $k=\text{num\_clusters}$.
3. Choose one cluster with biggest SSD among all existing clusters to split.
4. num_clusters +=1
5. Go to step 2 until the desired number of clusters is reached K.

BK-Means can be viewed as selecting the initial K centers.
BK-Means

- Partition whole data set to 2 clusters
BK-Means

After Splitting:

C1

C2
BK-Means

-A cluster with biggest SSD chosen to split.
BK-Means

After Splitting:

C1

C2

C3
BK-Means

A cluster with biggest SSD chosen to split.
BK-Means

After Splitting:

C1
C2
C3
C4
- 4 clusters are created.
Proposed Method Algorithm:

Input: data set, $K_{\text{min}}$, $K_{\text{max}}$, and milestone

1. Use BK-Means to partition the data set into $K_{\text{min}}$ clusters.
2. Set $k = K_{\text{min}}$, and compute validity of the current partition
3. While (validity does not meet the criteria && $k < K_{\text{max}}$)
   a. Choose the “biggest” cluster
   b. Partition it into 2 clusters.
   c. $k += 1$
   d. If ($k$ mod milestone == 0), then K-Means($k$)
   e. Compute validity of the current partition.
4. End of while loop.
5. Return clusters=$K$, validates
✓ **Milestone**, in our method, is defined to avoid too many K-Means running.

✓ The difference here, in our method, is that:
  ✓ running K-Means and updating centers occurs *in each iteration of milestone* instead of each step of cluster creation.
VALIDITY MEASURE

Metrics:

- Clustering quality metrics
  - **Compactness** (*Minimal intra cluster distance*), and
  - **Separation of clusters** (*Maximal inter cluster distance*).

- To determine the optimal number of clusters

  ✓ Calinski and Harabasz Index (CH)

  ✓ Ray&Turi Index (R&T)
VALIDITY MEASURE

Metrics:

✓ **Calinski and Harabasz Index (CH)**

\[ CH = \frac{\text{Trace } B}{\text{Trace } W} \times \frac{N - K}{K - 1} \]

\[ \text{Trace } B = \sum_{k=1}^{n} n_k (||z_k - z||)^2 \]

\[ \text{Trace } W = \sum_{k=1}^{K} \sum_{i=1}^{n_k} n_k (||x_i - z_k||)^2 \]
VALIDITY MEASURE

Metrics:

- Ray&Turi Index

\[ R&T \text{ Index} = \frac{\text{Intra}}{\text{Inter}} \]

\[
\text{Intra} = \frac{1}{N} \sum_{k=1}^{K} \sum_{i=1}^{n_k} n_k (||x_i - z_k||)^2
\]

\[
\text{Inter} = \min ||z_i - z_j||^2,
\]

where \( i = 1,2, \ldots K-1, \quad j = i + 1, \ldots K \).
Experimental Results
&
Comparison with Ray and Turi Method
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Type</th>
<th>No. of Points</th>
<th>No. of Dims</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIRCH1</td>
<td>Synthetic</td>
<td>100,000</td>
<td>2</td>
</tr>
<tr>
<td>DIM15</td>
<td>Hardware Responses</td>
<td>10,126</td>
<td>15</td>
</tr>
<tr>
<td>DIM64</td>
<td>Hardware Responses</td>
<td>1,024</td>
<td>64</td>
</tr>
<tr>
<td>8 XOR PUF(0s)</td>
<td>Synthetic</td>
<td>11,917</td>
<td>64</td>
</tr>
<tr>
<td>8 XOR PUF(1s)</td>
<td>Synthetic</td>
<td>12,083</td>
<td>64</td>
</tr>
</tbody>
</table>
### Table 3. Comparison of R&T procedure and Our Method on the CH index

<table>
<thead>
<tr>
<th>Dataset</th>
<th>(K_{\text{max}}) ((K_{\text{min}} = 3))</th>
<th>Time</th>
<th>Best K</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIM15</td>
<td>50</td>
<td>8.31</td>
<td>2.43</td>
</tr>
<tr>
<td>DIM64</td>
<td>300</td>
<td>14.3</td>
<td>3.84</td>
</tr>
<tr>
<td>BIRCH1</td>
<td>110</td>
<td>94.2</td>
<td>25.2</td>
</tr>
<tr>
<td>8 XOR PUF(0s)</td>
<td>200</td>
<td>355</td>
<td>131</td>
</tr>
<tr>
<td>8 XOR PUF(1s)</td>
<td>200</td>
<td>369</td>
<td>119</td>
</tr>
</tbody>
</table>
Comparison of run time for both methods using CH index

- Brich1
- 8 XOR PUF Ones
- 8 XOR PUF Zeros
- DIM64
- DIM15

R&T Method | Our Method
Table 4. Comparison of **R&T procedure** and **Our Method** on the **R&T index**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$K_{\text{max}}$ ($K_{\text{min}} = 3$)</th>
<th><strong>Time</strong></th>
<th><strong>Best K</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>R&amp;T</td>
<td>Our Method</td>
</tr>
<tr>
<td>DIM15</td>
<td>50</td>
<td>8.27</td>
<td>2.41</td>
</tr>
<tr>
<td>DIM64</td>
<td>300</td>
<td>14.9</td>
<td>4.49</td>
</tr>
<tr>
<td>BIRCH1</td>
<td>110</td>
<td>95.2</td>
<td>25.1</td>
</tr>
</tbody>
</table>
Compersion of run time for both methods using R&T index
Conclusion

- Our method indicated that:
  - Maintain the quality of clusters.
  - Reduce time computation.
  - Flexibility of choosing different validity measure.
Thank You For Your Attention

Zohreh.safari@ttu.edu

https://dl.acm.org/doi/abs/10.1145/3388142.3388164