A Faster Sampling Algorithm for Spherical $k$-means

Rameshwar Pratap, Anup Deshmukh, Pratheeksha Nair and Tarun Dutt
1 Clustering

2 Cosine Similarity

3 Problem Statement

4 Baseline Algorithms

5 Our algorithm -SPKM-MCMC
What is Clustering

- Partitioning of unlabeled data objects (examples) into disjoint clusters such that
  - data objects within a cluster are very similar
  - data objects in different clusters are very different
- Discover new categories in an unsupervised manner (unlabeled data)
Applications of text clustering

- Pre-processing for fast search
- Postprocessing of search results
- Summarizing news articles along with headlines
- Collaborative filtering algorithms in Recommender Systems
1 Clustering

2 Cosine Similarity

3 Problem Statement

4 Baseline Algorithms

5 Our algorithm -SPKM-MCMC
## Representation – The Problem of Sparsity

<table>
<thead>
<tr>
<th>Data mining items</th>
<th>Linear Algebra Items</th>
<th>Neutral Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>Linear</td>
<td>Analysis</td>
</tr>
<tr>
<td>Mining</td>
<td>Algebra</td>
<td>Application</td>
</tr>
<tr>
<td>Clustering</td>
<td>Matrix</td>
<td></td>
</tr>
<tr>
<td>Classification</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Term(s) 1</th>
<th>Research title 1</th>
<th>Research title 2</th>
<th>Research title 3</th>
<th>Research title 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Term(s) 2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Term(s) 3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>..</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>..</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>..</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>..</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>..</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>..</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>..</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Term(s) 10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>
How to measure similarity between two documents? – Cosine Similarity

- Cosine similarity is appropriate for determining similarity between documents

\[ \cos(D_i, D_j) = \frac{\langle D_i, D_j \rangle}{||D_i||.||D_j||} \]

- \( D_1 = (2, 3, 5), D_2 = (3, 7, 1), Q = (0, 0, 2) \)

\[ \cos(D_1, Q) = \frac{2 \times 0 + 3 \times 0 + 5 \times 2}{\sqrt{4 + 9 + 25\sqrt{0 + 0 + 4}}} = 0.81 \]

\[ \cos(D_2, Q) = \frac{3 \times 0 + 7 \times 0 + 1 \times 2}{\sqrt{9 + 49 + 25\sqrt{0 + 0 + 4}}} = 0.13 \]

- \( D_1 \) is 6 times closer to \( Q \) than \( D_2 \)

**Figure:** Cosine Similarity
Illustration
1. Clustering

2. Cosine Similarity

3. Problem Statement

4. Baseline Algorithms

5. Our algorithm - SPKM-MCMC
Spherical $k$-means Clustering

Input
- A set $X = \{x_1, x_2, \ldots, x_n\}$ of $n$ documents (represented as vectors)
- Number of clusters $k$

For a set $C = \{c_1, c_2, \ldots, c_k\}$ of cluster centers define:

$$\phi_X(C) = 1.5|X| - \sum_{x \in X} \cos(x, C')$$

where $\cos(x, C)$ is cosine similarity between $x$ and its closest center in $C$

Goal
To find a set $C$ of cluster centers that minimizes the objective function $\phi_X(C')$
1. Clustering

2. Cosine Similarity

3. Problem Statement

4. Baseline Algorithms

5. Our algorithm - SPKM-MCMC
Spherical $k$-means algorithm – SPKM

**SPKM Algorithm**

- Start with $k$ arbitrary centers $\{c_1, c_2, \ldots, c_k\}$ (sampled uniformly at random from data points)
- Performs an EM-type local search till convergence
Spherical $k$-means algorithm – SPKM

**SPKM Algorithm**

- Start with $k$ arbitrary centers \( \{c_1, c_2, \ldots, c_k\} \) (sampled uniformly at random from data points)
- Performs an EM-type local search till convergence

**Main advantage:** Simplicity
Limitations of SPKM

- May take many iterations to converge
- Very sensitive to initialization
  - Random initialization likely to get two centers in the same cluster, when clusters are highly skewed
  - SPKM gets stuck in a local optimum
SPKM++

Preprocessing
Normalise every data point to unit norm

SPKM++ Algorithm

- **Main idea:** Spreads out the centers
- Choose first center, $c_1$, uniformly at random from the data set: $C \leftarrow c_1$
- Repeat for $2 \leq i \leq k$:
  - Choose $c_i$ to be equal to a data point $x'$ sampled from the following distribution:
    \[
    \frac{1.5(1 - \cos(x', C))}{\phi_X(C)} \propto (1.5 - \cos(x', C))
    
    \]
  - $C \leftarrow C \cup \{c_i\}$

**Theorem:** $O(\log k)$-approximation to optimum, right after initialization
Experimental comparison between SPKM and SPKM++

<table>
<thead>
<tr>
<th>Data Set</th>
<th>No. of documents</th>
<th>No. of words in the vocab (dimension)</th>
<th>Max no. of words in a document (sparsity)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NIPS full papers</td>
<td>1500</td>
<td>12419</td>
<td>914</td>
</tr>
<tr>
<td>KOS blog entries</td>
<td>3430</td>
<td>6906</td>
<td>457</td>
</tr>
<tr>
<td>BBC</td>
<td>9635</td>
<td>2225</td>
<td>128</td>
</tr>
<tr>
<td>20News</td>
<td>1700</td>
<td>56916</td>
<td>734</td>
</tr>
</tbody>
</table>

**Figure:** Dataset description

**Figure:** Comparison in terms of clustering cost and time
Limitations of SPKM++

- Needs $k$ passes over the data for choosing $k$ initial cluster center
- In large data applications, not only the data is massive, but also $k$ is typically large (e.g., easily 1000) and hence is not scalable
1. Clustering
2. Cosine Similarity
3. Problem Statement
4. Baseline Algorithms
5. Our algorithm -SPKM-MCMC
Key Idea
Approximate SPKM++ sampling \textit{via} Markov chain sampling

Crux of the Analysis
- We show that for every iterations \((1 \leq i \leq k)\) Markov chain distribution is close to the underlying SPKM++ distribution
- As a consequence of above, we show that expected clustering cost is also preserved
Theorem: $O(\log k) + \epsilon AV(\mathcal{X})$-approximation to optimum.
Where, $d(x, C) = \min_{c \in C} (1.5 - \langle x, c \rangle)$; $m = 1 + O(\frac{1}{\epsilon} \log \frac{k}{\epsilon})$;
$AV(\mathcal{X}) = |\mathcal{X}| - \sum_{x \in \mathcal{X}} \langle x, FM(\mathcal{X}) \rangle$; $FM(\mathcal{X}) = \frac{\sum_{x \in \mathcal{X}} x}{\| \sum_{x \in \mathcal{X}} x \|}$
# Experimental comparison between SPKM++ and SPKM-MCMC

<table>
<thead>
<tr>
<th>$k = 10$</th>
<th>KOS</th>
<th>BBC</th>
<th>NIPS</th>
<th>20News</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPKM++</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
</tr>
<tr>
<td>SPKM-MCMC (m=5)</td>
<td>-0.03%</td>
<td>0.07%</td>
<td>0.08%</td>
<td>0.48%</td>
</tr>
<tr>
<td>SPKM-MCMC (m=30)</td>
<td>-0.07%</td>
<td>-0.03%</td>
<td>0.08%</td>
<td>0.03%</td>
</tr>
<tr>
<td>SPKM-MCMC (m=100)</td>
<td>-0.06%</td>
<td>-0.03%</td>
<td>0.09%</td>
<td>-0.14%</td>
</tr>
<tr>
<td>SPKM-MCMC (m=500)</td>
<td>-0.43%</td>
<td>0.06%</td>
<td>-0.13%</td>
<td>-0.08%</td>
</tr>
</tbody>
</table>

**Figure:** Comparison of clustering cost between two algorithms for different values of $m$

<table>
<thead>
<tr>
<th>$k = 10$</th>
<th>KOS</th>
<th>BBC</th>
<th>NIPS</th>
<th>20News</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPKM++</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>SPKM-MCMC (m=5)</td>
<td>×8.0</td>
<td>×7.5</td>
<td>×5.4</td>
<td>×4.8</td>
</tr>
<tr>
<td>SPKM-MCMC (m=30)</td>
<td>×7.6</td>
<td>×7.0</td>
<td>×5.0</td>
<td>×3.3</td>
</tr>
<tr>
<td>SPKM-MCMC (m=100)</td>
<td>×6.6</td>
<td>×5.7</td>
<td>×4.2</td>
<td>×1.8</td>
</tr>
<tr>
<td>SPKM-MCMC (m=500)</td>
<td>×4.0</td>
<td>×2.7</td>
<td>×2.2</td>
<td>×0.5</td>
</tr>
</tbody>
</table>

**Figure:** Comparison of seeding time between two algorithms for different values of $m$
Comparing SPKM++ and SPKM-MCMC

Cost=1357.22  100 iterations

Cost=998.13  100 iterations
Comparing SPKM++ and SPKM-MCMC

Cost=913.82  500 iterations

Cost=781.56  500 iterations
Comparing SPKM++ and SPKM-MCMC

Cost=751.8 872 iterations

Cost=729.1 520 iterations
Practicality of SPKM++ algorithm

We conducted thorough experimental evaluations of SPKM++ (Endo and Miyamoto, 2015) on publicly available datasets to demonstrate its applicability, which was not addressed in their paper. We obtain improved clustering quality in addition to better running time with respect to vanilla SPKM (Dhillon and Modha, 2001).

SPKM-MCMC

We propose a Markov chain based algorithm (SPKM-MCMC) for initial seeding of $k$ points. The theoretical guarantee on the clustering cost of our algorithm is close to SPKM++ while achieving a significant speed-up in the seeding time.

Implementation

Link to our Github code: https://github.com/nair-p/SPKM
Questions/Comments?
Thank You!