Document Clustering using Particle Swarm Optimization

Paper: Xiaohui Chui, Thomas Potok, Paul Palathingal

Presentation: Eric Reed

Document Clustering

- Partition a collection of documents into a given number of clusters
- Try to find inherent structure in the data
- Expose this structure as a set of groups
- Elements of a group should be similar
- Different clusters should be disimilar

Document Representation

- A document is a point in multi-dimensional space represented by a vector
- Each dimension corresponds to a term in the document
- The position along a dimension is based on the frequency of the term in the document compared to the frequency in the dataset
- Term i weight in document j:

$$w_{ji} = tf_{ji} \times idf_{ji} = tf_{ji} \times \log_2 (n / df_{ji})$$

Document Similarity

- Two most common metrics:
  - Euclidean distance

$$d(m_p, m_j) = \sqrt{\sum_{k=1}^{d_m} (m_{pk} - m_{jk})^2 / d_m}$$

  - Cosine correlation

$$\cos(m_p, m_j) = \frac{m_p^t m_j}{\| m_p \| \| m_j \|}$$

K-means

1. Randomly select cluster centroid vectors to set an initial dataset partition
2. Assign each document vector to the closest cluster centroids
3. Recalculate the cluster centroid vector using

$$c_j = \frac{1}{n_j} \sum_{\forall d_j \in S_j} d_j$$

4. Unless convergence is achieved GOTO 2

More K-means

- Common convergence conditions:
  - Maximum number of iterations reached
  - No change from previous iteration
- Deterministic – execution on the same dataset with the same initial centroids will give the same result
- Poor selection of initial centroids can cause K-means to get trapped in a local optima
- Very fast! Usually 10-20 iterations required

PSO Clustering Algorithm

- Each particle represents a complete clustering solution, i.e. a collection of cluster centroid vectors.
- The fitness function computes the average distance of documents to the cluster centroid (ADDC)

$$f = \frac{\sum_{i=1}^{N_c} \{ \frac{\sum_{j=1}^{p_i} d(o_i, m_{ij})}{p_i} \}}{N_c}$$

- Stop after max iterations or average change below threshold

K-means vs. PSO

- K-means very fast, but can get trapped in local optima
  - 10-20 iterations for convergence
- PSO very parallel and better results, but needs many more iterations
  - 500+ iterations for convergence
- What to do?

Hybrid PSO = K-means + PSO

- Stitch them together into an optimizing monstrosity of course!
- Do PSO for a while to get particles near global optimum (50-100 iterations)
- Then finish with K-means to converge

Actual Data!

- Datasets from Text Retrieval Conference (TREC) collections
- Stripped of very common words (a, the, in, to, I, he, she, it)
- Different forms of a word are reduced to one canonical form using Porter's algorithm (things -> thing)
- Each algorithm gets 100 iterations
  - Hybrid PSO split: 90 PSO, 10 K-means

Actual Data!

Table 1. Summary of text document datasets

Data	Number of documents	Number of terms	Number of clusters
Dataset1	414	6429	9
Dataset2	313	5804	8
Dataset3	204	5832	6
Dataset4	878	7454	10

Actual Data!

Table 2: Performance comparison of K-means, PSO, hybrid PSO algorithms

		ADDC value		
		K-means	PSO	Hybrid PSO
Dataset1	Euclidian	8.17817	8.11009	6.38039
	Cosine	8.96442	10.41271	8.14551
Dataset2	Euclidian	7.26175	6.25172	4.51753
	Cosine	8.07653	9.57786	7.21153
Dataset3	Euclidian	4.59539	4.14896	2.25961
	Cosine	4.97171	5.71146	4.00555
Dataset4	Euclidian	9.08759	8.62794	6.37872
	Cosine	10.1739	12.8927	9.5379

Actual Data!

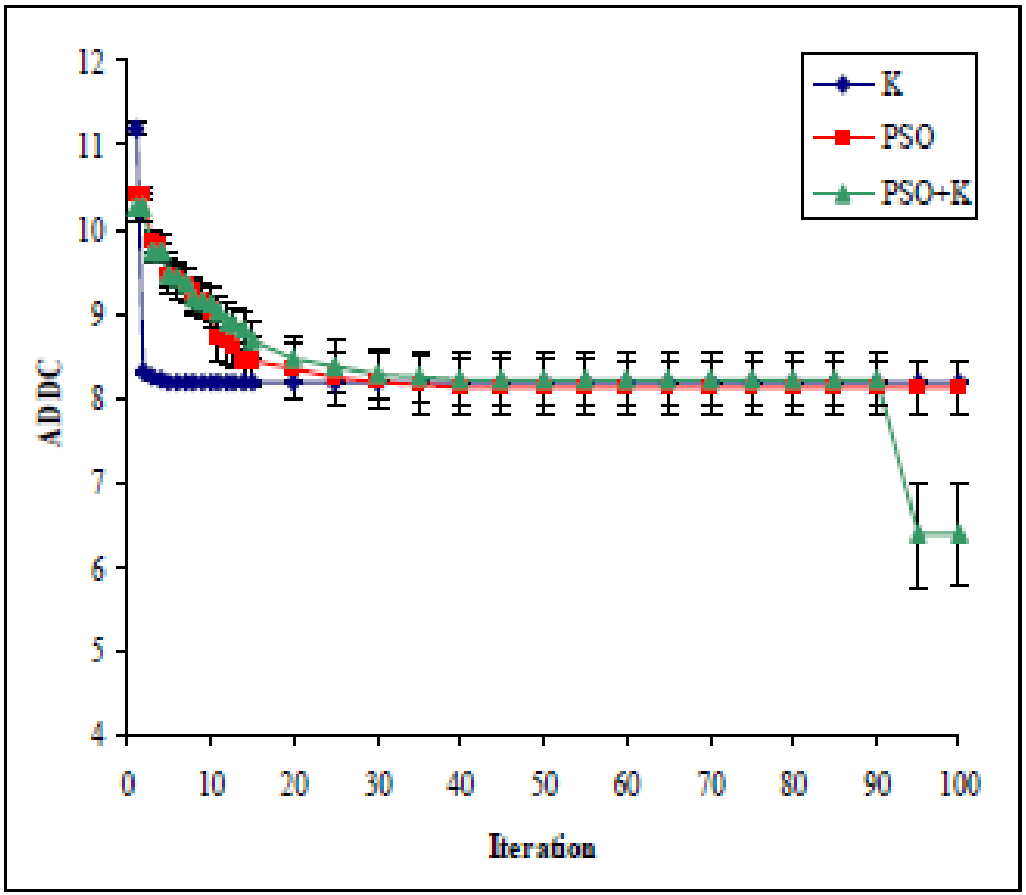

Figure 1: The convergence behaviors of different clustering algorithm (K-means, PSO and hybrid PSO algorithms)