Locality Sensitive Hashes for Approximate Spectral Clustering Applied to Text Clustering in Social Media

**Background**

Opinion Space is a social media tool that uses concepts from dimensionality reduction, collaborative filtering, and deliberative polling to collect ideas about a topic.

We will use SPARK to perform an approximate spectral clustering algorithm to visualize and cluster related words from participant textual comments.

**Spectral Clustering**

- Family of clustering and dimensionality reduction algorithms related to Spectral Graph theory and eigenvector analysis of Graph Laplacian matrices.
- Non-linear techniques with high accuracy for non-convex cluster decision regions.
- Accuracy comes with $O(n^3)$ time complexity, and with a dictionary of 10,000 words, this can mean $O(10^{12})$ operations.

**Text and Clustering Model**

- Construct a sparse weighted graph of correlated words weighted by their Pearson correlation.
- Use a two-step embedding and clustering technique similar to Ng et al. (2001).
  - Find a low-dimensional embedding that preserves a word's distance (weight) relative to its neighbors in the graph.
  - Assigns embedded nodes to clusters with k-means.

**Approximate Spectral Clustering**

- For $N$ words, the formation of a graph and associated $N \times N$ Laplacian matrix is expensive.
- Prior work in approximate spectral clustering addresses this problem.
  - Yan et al. (2009), Chen et al. (2011), Hafeeda et al. (2012)
- Our solution is a Spark-powered parallel workflow based on Locality Sensitive Hashes (LSH) similar to Hafeeda et al.

**Locality-Sensitive Hash**

- A Random Projection LSH uses random hyperplane cutting to probabilistically bucket data.
- These buckets are used as a heuristic for finding neighbors in the graphical model of words.

**Experimental Results**

- 2 Opinion Space Projects
  - Foreign Policy (11,497 words and 2149 comments)
  - Online Learning (1,141 words and 153 comments)
- Results
  - Clustered words into 17 groups for the Foreign Policy dataset and 7 groups for the Online Learning dataset.
  - Visualized clusters and an associated 2D embedding in Figure 1, 2

**Scalability and Spark**

- Use SPARK to form the Laplacian matrix in parallel.
- Spark in-memory computation (RDD's) for efficient analysis of the calculated Laplacian matrix.
- Broadcast variables for efficient eigenvector power-iteration.

**Future Work**

- Compare method to other approximations such as KASP, RASP.
- Evaluate results of partitioning data and running independent clustering.
- Apply same workflow to other domains such as robot learning.
- Replace k-means with another clustering technique in the embedded space.

**References**


We would like to acknowledge the UC Berkeley AMPLab and SPARK team that made this project possible. Special thanks to Reynold Xin and Siamak Faridani for help on this project.

sanjay@eecs.berkeley.edu