Modeling and Understanding the Evolution of Innovation in Academia

Mengjie Cheng, Ziyue Wang, Lantao Mei
Stanford University

Motivation

Innovation is one of the core values of development. So far, many works focus on modeling innovation qualitatively with big data. We hope to use machine learning methods to understand and model the evolution of scientific terms in academia.

Data and Feature Engineering

This project uses JSON format data from Aminer which covers 166,192,182 papers in total.
Feature vector is designed to be the frequency distributions from 1967 to 2016 of each chosen key word:

$$X_{list}(keyword) = \frac{N_{list}(keyword, y)}{N(y)}$$

Where $N_{list}(keyword, y)$ is the number of papers at year y that contain the keyword in their word lists. N(y) is the total number of keywords at year y.
Gaussian kernel smoothing is applied to smooth the distributions obtained above. Kernel k is defined as:

$$\ker(x_i, x_j) = e^{\frac{-(x_i - x_j)^2}{2\sigma^2}}$$

Where σ is chosen to be 1 year. x=$(x_1, \dots, x_{50})^T$.
Therefore, each element of the smoothed data is given by:

$$x_{si} = \frac{\sum_{j=1}^{m} \ker(x_i, x_j) x_j}{\sum_{j=1}^{m} \ker(x_i, x_j)}$$

Keyword Extraction

Keywords mainly come from two sources. 1. Using a phrase mining framework named **Autophrase** to extract from abstract. 2. Using the "keyword lists" from the raw data.

Features for Decision Tree

Recognition	• avg # of citations for supporting papers with a terms. • avg/total # of citations for relevant authors • avg/total # of citations for relevant venues
Past success	• **PaperCount**: # of publications where a term is mentioned • changes of probability occuring in publications
Competition	• term's localing clustering coefficient
Closeness	• term's avg similarity to different clusters

Learning Algorithms

K-means

Repeat two steps until convergence:

(i) Assign each training example x(i) to the closest cluster centroid.
(ii) Move each center to the mean of the points assigned to it.

K-Spectral Centroid (KSC) Clustering

Define distance d(x,y) between time series x,y as follows:

$$d(x, y) = \min_{\alpha, q} \frac{\|x - \alpha y_q\|}{\|x\|}$$

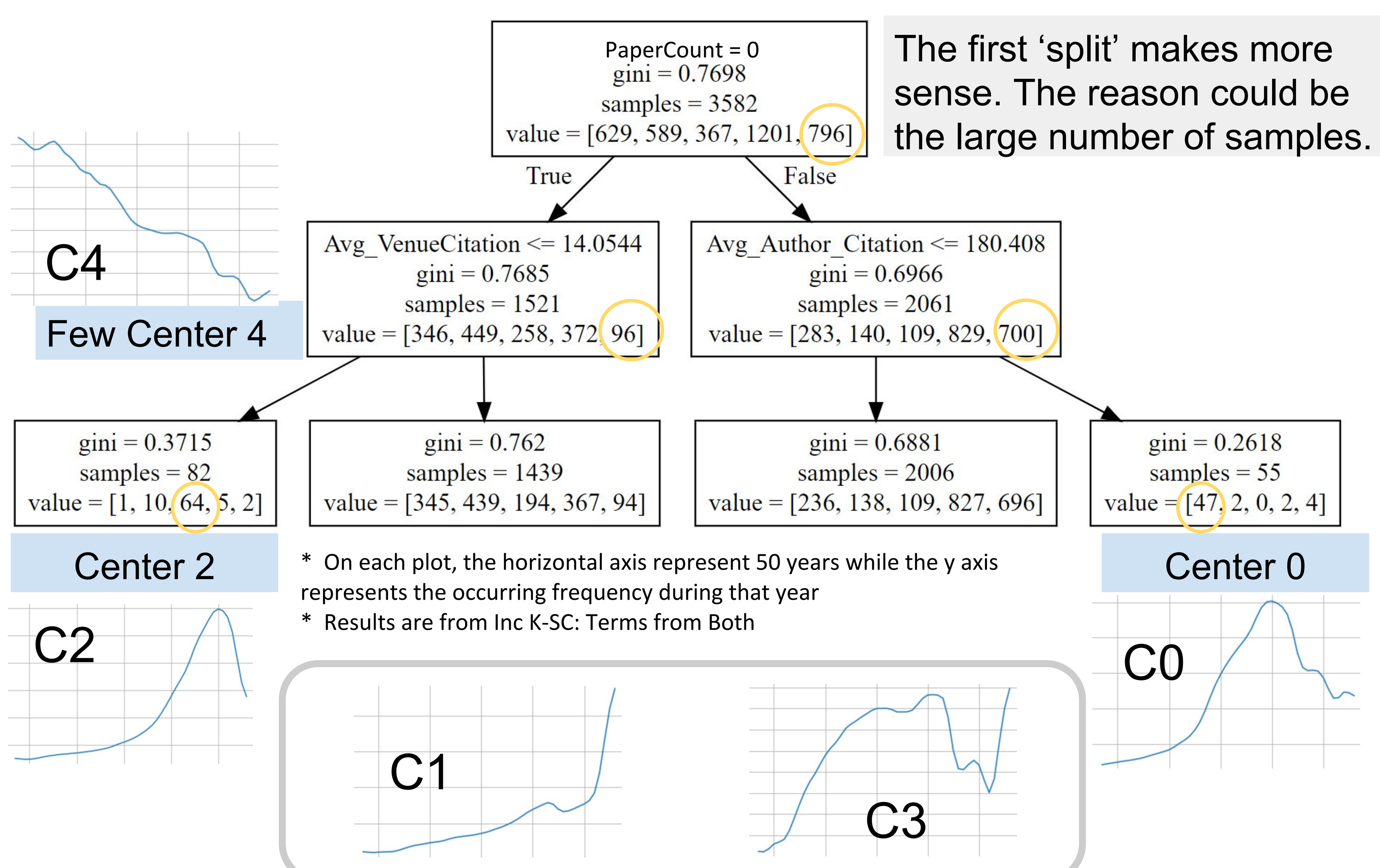
Repeat two steps until convergence:

(i) Assign each training example x(i) to the closest cluster centroid based on distance d. (ii) Update the new cluster center be the minimum of the sum of d$(\mu_i, \mu_j)^2$.

Haar Incremental Algorithm for K-SC

Used to speed up KSC method since Haar transformations are used to compress data into lower dimensions.

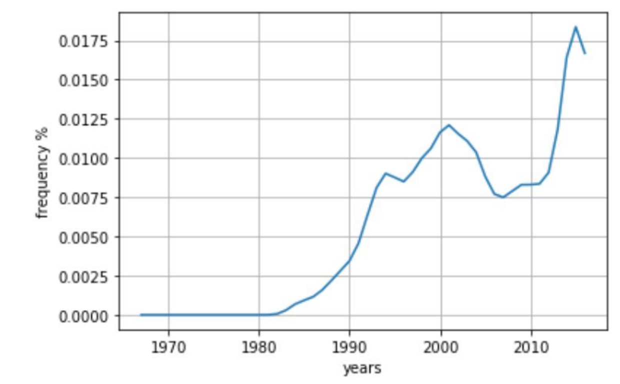
Centers & Decision Tree



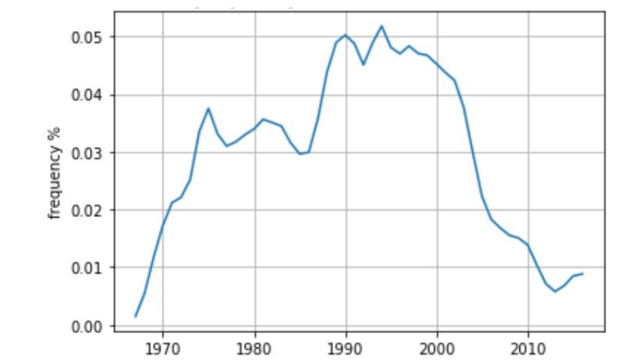
* On each plot, the horizontal axis represent 50 years while the y axis represents the occurring frequency during that year
* Results are from Inc K-SC: Terms from Both

Term Examples

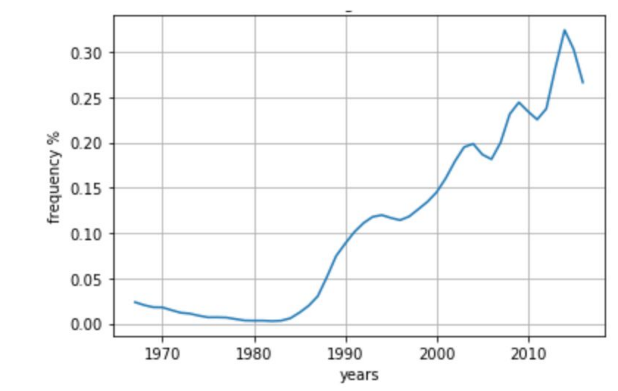
Social Phobia (C1)



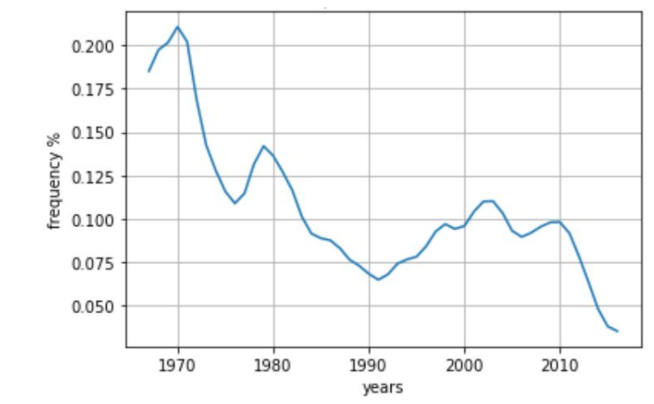
Josephson Junction (C3)



Machine Learning (C2)



Water Vapor (C4)



Cluster Quality Evaluation

Method	$\Sigma d(\mu_i, \mu_j)^2$ (higher is better)	F (lower is better)
K-means: Terms from Abstracts	1.87	471.05
Inc K-SC: Terms from Abstracts	4.54	208.24
K-means: Terms from Keyword Lists	0.14	442.09
Inc K-SC: Terms from Keyword Lists	3.59	199.48
Kmeans: Terms from Both	0.13	949.24
Inc K-SC: Terms from Both	4.18	471.23

* The performance of Incremental KSC is approximately the same as regular KSC.
* The metric $\Sigma d(\mu_i, \mu_j)^2$ is basically independent on the length of the term list.
* The metric F is strongly influenced by the length of the term list.

F is computed as:

$$F = \sum_{k=1}^{K} \sum_{x_i \in C_k} d(x_i, \mu_k)^2$$

where K is the number of clusters, C is cluster assignment, μ is the centroid.

Conclusions

- K-SC performs better than K-means for time-series data in our project.
- We did not observe significant effect on short feature vectors (~50) for Haar incremental algorithm for K-SC.
- The evolution pattern is strongly correlated with PaperCount, which indicates past success; then it comes with the author citations and venue citations, which indicates recognition.