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: 

\[ X_{list}(keyword) = \frac{N(y)}{N(Y)} \]

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:

\[ k(x_i, x_j) = e^{-\frac{(x_i - x_j)^2}{\sigma}} \]

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

\[ x_{kj} = \frac{\sum_{i=1}^{m} k(x_i, x_j)}{\sum_{j=1}^{n} \sum_{i=1}^{m} k(x_i, x_j)} \]

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 term.
  - 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 occurring in publications
- Competition
  - term’s localizing 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:

\[ d(x, y) = \min_{a, q} \frac{1}{x} \frac{1}{2} \left| x - aq \right| \]

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.

Term Examples
- Social Phobia (C1)
- Josephson Junction (C3)
- Machine Learning (C2)
- Water Vapor (C4)

Cluster Quality Evaluation

<table>
<thead>
<tr>
<th>Method</th>
<th>[ E(k, \mu)^2 ] (higher is better)</th>
<th>[ F ] (lower is better)</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-means: Terms from Abstracts</td>
<td>1.87</td>
<td>471.05</td>
</tr>
<tr>
<td>Inc K-SC: Terms from Abstracts</td>
<td>4.54</td>
<td>208.24</td>
</tr>
<tr>
<td>K-means: Terms from Keyword Lists</td>
<td>0.14</td>
<td>442.09</td>
</tr>
<tr>
<td>Inc K-SC: Terms from Keyword Lists</td>
<td>3.59</td>
<td>199.48</td>
</tr>
<tr>
<td>K-means: Terms from Both</td>
<td>0.13</td>
<td>949.24</td>
</tr>
<tr>
<td>Inc K-SC: Terms from Both</td>
<td>4.18</td>
<td>471.23</td>
</tr>
</tbody>
</table>

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.