We can predict which language an article is written in by running pagerank on the different concept networks in Wikidata resolved concept ID's for each link. Calculate personalized pagerank with a 1 for the seed 2-hop depth first search from seed article 'Arab-Israeli Conflict'. The combined adjacency matrix for Arabic and Israeli graphs (1) links that each article does or does not contain.

Data Set And Experiment Setup
- Wikipedia links from Wikipedia.org
- Wikidata resolved concept ID’s for each link

(1) Clustering Results
- 2-hop depth first search from seed article ‘Arab-Israeli Conflict’ in each language
- Calculate personalized pagerank with a 1 for the seed article

(2) Logistic Regression and SVD setup
- Find the principal components of the data (the axes that explain the most variance across the data) using the formula [3]:

\[ X = UDV^T \]

Confusion Matrices

(1) Clustering
- Hierarchical agglomerative clustering
- Using a distance metric, assign each element (or group of elements) to closest cluster until there is only one cluster
- i.e. Euclidean metric (I used Euclidean and cosine) [1]:

\[ d(i,j) = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2} \]

(2) Logistic Regression
- Model how features interact to predict label as coefficients on features, \( \beta \)
- Assign label according to [2]:

\[ \hat{y}(x) = \frac{1}{1 + e^{-\beta^T x}} \]

- Split the data into train and test sets randomly.
- Learn \( \beta \) in order to maximize likelihood of training data.
- Predict on test data.

(2) SVD
- Find the principal components of the data (the axes that explain the most variance across the data) using the formula [3]:

\[ X = UDV^T \]

Confusion Matrices

(1) Clustering
- Hierarchical agglomerative clustering
- Using a distance metric, assign each element (or group of elements) to closest cluster until there is only one cluster
- i.e. Euclidean metric (I used Euclidean and cosine) [1]:

\[ d(i,j) = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2} \]

(2) Logistic Regression
- Model how features interact to predict label as coefficients on features, \( \beta \)
- Assign label according to [2]:

\[ \hat{y}(x) = \frac{1}{1 + e^{-\beta^T x}} \]

- Split the data into train and test sets randomly.
- Learn \( \beta \) in order to maximize likelihood of training data.
- Predict on test data.

(2) SVD
- Find the principal components of the data (the axes that explain the most variance across the data) using the formula [3]:

\[ X = UDV^T \]

Confusion Matrices

(1) Clustering
- Hierarchical agglomerative clustering
- Using a distance metric, assign each element (or group of elements) to closest cluster until there is only one cluster
- i.e. Euclidean metric (I used Euclidean and cosine) [1]:

\[ d(i,j) = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2} \]

(2) Logistic Regression
- Model how features interact to predict label as coefficients on features, \( \beta \)
- Assign label according to [2]:

\[ \hat{y}(x) = \frac{1}{1 + e^{-\beta^T x}} \]

- Split the data into train and test sets randomly.
- Learn \( \beta \) in order to maximize likelihood of training data.
- Predict on test data.

(2) SVD
- Find the principal components of the data (the axes that explain the most variance across the data) using the formula [3]:

\[ X = UDV^T \]

Confusion Matrices

(1) Clustering
- Hierarchical agglomerative clustering
- Using a distance metric, assign each element (or group of elements) to closest cluster until there is only one cluster
- i.e. Euclidean metric (I used Euclidean and cosine) [1]:

\[ d(i,j) = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2} \]

(2) Logistic Regression
- Model how features interact to predict label as coefficients on features, \( \beta \)
- Assign label according to [2]:

\[ \hat{y}(x) = \frac{1}{1 + e^{-\beta^T x}} \]

- Split the data into train and test sets randomly.
- Learn \( \beta \) in order to maximize likelihood of training data.
- Predict on test data.

(2) SVD
- Find the principal components of the data (the axes that explain the most variance across the data) using the formula [3]:

\[ X = UDV^T \]

Confusion Matrices

(1) Clustering
- Hierarchical agglomerative clustering
- Using a distance metric, assign each element (or group of elements) to closest cluster until there is only one cluster
- i.e. Euclidean metric (I used Euclidean and cosine) [1]:

\[ d(i,j) = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2} \]

(2) Logistic Regression
- Model how features interact to predict label as coefficients on features, \( \beta \)
- Assign label according to [2]:

\[ \hat{y}(x) = \frac{1}{1 + e^{-\beta^T x}} \]

- Split the data into train and test sets randomly.
- Learn \( \beta \) in order to maximize likelihood of training data.
- Predict on test data.

(2) SVD
- Find the principal components of the data (the axes that explain the most variance across the data) using the formula [3]:

\[ X = UDV^T \]

Confusion Matrices

(1) Clustering
- Hierarchical agglomerative clustering
- Using a distance metric, assign each element (or group of elements) to closest cluster until there is only one cluster
- i.e. Euclidean metric (I used Euclidean and cosine) [1]:

\[ d(i,j) = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2} \]

(2) Logistic Regression
- Model how features interact to predict label as coefficients on features, \( \beta \)
- Assign label according to [2]:

\[ \hat{y}(x) = \frac{1}{1 + e^{-\beta^T x}} \]

- Split the data into train and test sets randomly.
- Learn \( \beta \) in order to maximize likelihood of training data.
- Predict on test data.

(2) SVD
- Find the principal components of the data (the axes that explain the most variance across the data) using the formula [3]:

\[ X = UDV^T \]

Confusion Matrices

(1) Clustering
- Hierarchical agglomerative clustering
- Using a distance metric, assign each element (or group of elements) to closest cluster until there is only one cluster
- i.e. Euclidean metric (I used Euclidean and cosine) [1]:

\[ d(i,j) = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2} \]

(2) Logistic Regression
- Model how features interact to predict label as coefficients on features, \( \beta \)
- Assign label according to [2]:

\[ \hat{y}(x) = \frac{1}{1 + e^{-\beta^T x}} \]

- Split the data into train and test sets randomly.
- Learn \( \beta \) in order to maximize likelihood of training data.
- Predict on test data.

(2) SVD
- Find the principal components of the data (the axes that explain the most variance across the data) using the formula [3]:

\[ X = UDV^T \]

Confusion Matrices

(1) Clustering
- Hierarchical agglomerative clustering
- Using a distance metric, assign each element (or group of elements) to closest cluster until there is only one cluster
- i.e. Euclidean metric (I used Euclidean and cosine) [1]:

\[ d(i,j) = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2} \]

(2) Logistic Regression
- Model how features interact to predict label as coefficients on features, \( \beta \)
- Assign label according to [2]:

\[ \hat{y}(x) = \frac{1}{1 + e^{-\beta^T x}} \]

- Split the data into train and test sets randomly.
- Learn \( \beta \) in order to maximize likelihood of training data.
- Predict on test data.

(2) SVD
- Find the principal components of the data (the axes that explain the most variance across the data) using the formula [3]:

\[ X = UDV^T \]