MAT 167: Applied Linear Algebra
Lecture 23: Text Mining II

Naoki Saito

Department of Mathematics
University of California, Davis

November 19 & 21, 2025

Outline

Outline

Using Cluster Centroids for Text Mining

- Instead of using the left singular vectors as a basis to approximate a term-document matrix, let's examine the cluster centers (centroids) obtained by k-means algorithm as a basis.
- Let $C_k = [\boldsymbol{c}_1 \dots \boldsymbol{c}_k] \in \mathbb{R}^{m \times k}$ be the k cluster centroids obtained by the k-means algorithm.
- $\boldsymbol{c}_j$'s are non-orthogonal; hence it is more convenient to obtain a set of orthonormal vectors that spans range(C_k).
- To do so, we can use the *reduced QR factorization*: $C_k = \widehat{Q}_k \widehat{R}_k$ where $\widehat{Q}_k \in \mathbb{R}^{m \times k}$, and $\widehat{R}_k \in \mathbb{R}^{k \times k}$.
- Now, let's approximate A using $\widehat{Q}_k$ in the sense of the least squares as:

$$\min_{G_k \in \mathbb{R}^{k \times n}} \|A - \widehat{Q}_k G_k\|_F.$$

- Let $G_k = [\boldsymbol{g}_1 \dots \boldsymbol{g}_n] \in \mathbb{R}^{k \times n}$. Then the above is equivalent to the following set of the LS problems:

$$\min_{\boldsymbol{g}_j \in \mathbb{R}^k} \|\boldsymbol{a}_j - \widehat{Q}_k \boldsymbol{g}_j\|_2, \quad j = 1 : n.$$

Using Cluster Centroids for Text Mining

- Instead of using the left singular vectors as a basis to approximate a term-document matrix, let's examine the cluster centers (centroids) obtained by k-means algorithm as a basis.
- Let $C_k = [\boldsymbol{c}_1 \dots \boldsymbol{c}_k] \in \mathbb{R}^{m \times k}$ be the k cluster centroids obtained by the k-means algorithm.
- $\boldsymbol{c}_j$'s are non-orthogonal; hence it is more convenient to obtain a set of orthonormal vectors that spans range(C_k).
- To do so, we can use the *reduced QR factorization*: $C_k = \widehat{Q}_k \widehat{R}_k$ where $\widehat{Q}_k \in \mathbb{R}^{m \times k}$, and $\widehat{R}_k \in \mathbb{R}^{k \times k}$.
- Now, let's approximate A using $\widehat{Q}_k$ in the sense of the least squares as:

$$\min_{G_k \in \mathbb{R}^{k \times n}} \|A - \widehat{Q}_k G_k\|_F.$$

- Let $G_k = [\boldsymbol{g}_1 \dots \boldsymbol{g}_n] \in \mathbb{R}^{k \times n}$. Then the above is equivalent to the following set of the LS problems:

$$\min_{\boldsymbol{g}_j \in \mathbb{R}^k} \|\boldsymbol{a}_j - \widehat{Q}_k \boldsymbol{g}_j\|_2, \quad j = 1 : n.$$

Using Cluster Centroids for Text Mining

- Instead of using the left singular vectors as a basis to approximate a term-document matrix, let's examine the cluster centers (centroids) obtained by k-means algorithm as a basis.
- Let $C_k = [\boldsymbol{c}_1 \dots \boldsymbol{c}_k] \in \mathbb{R}^{m \times k}$ be the k cluster centroids obtained by the k-means algorithm.
- $\boldsymbol{c}_j$'s are non-orthogonal; hence it is more convenient to obtain a set of orthonormal vectors that spans range(C_k).
- To do so, we can use the *reduced QR factorization*: $C_k = \widehat{Q}_k \widehat{R}_k$ where $\widehat{Q}_k \in \mathbb{R}^{m \times k}$, and $\widehat{R}_k \in \mathbb{R}^{k \times k}$.
- Now, let's approximate A using $\widehat{Q}_k$ in the sense of the least squares as:

$$\min_{G_k \in \mathbb{R}^{k \times n}} \|A - \widehat{Q}_k G_k\|_F.$$

- Let $G_k = [\boldsymbol{g}_1 \dots \boldsymbol{g}_n] \in \mathbb{R}^{k \times n}$. Then the above is equivalent to the following set of the LS problems:

$$\min_{\boldsymbol{g}_j \in \mathbb{R}^k} \|\boldsymbol{a}_j - \widehat{Q}_k \boldsymbol{g}_j\|_2, \quad j = 1 : n.$$

Using Cluster Centroids for Text Mining

- Instead of using the left singular vectors as a basis to approximate a term-document matrix, let's examine the cluster centers (centroids) obtained by k-means algorithm as a basis.
- Let $C_k = [\boldsymbol{c}_1 \dots \boldsymbol{c}_k] \in \mathbb{R}^{m \times k}$ be the k cluster centroids obtained by the k-means algorithm.
- $\boldsymbol{c}_j$'s are non-orthogonal; hence it is more convenient to obtain a set of orthonormal vectors that spans range(C_k).
- To do so, we can use the *reduced QR factorization*: $C_k = \widehat{Q}_k \widehat{R}_k$ where $\widehat{Q}_k \in \mathbb{R}^{m \times k}$, and $\widehat{R}_k \in \mathbb{R}^{k \times k}$.
- Now, let's approximate A using $\widehat{Q}_k$ in the sense of the least squares as:

$$\min_{G_k \in \mathbb{R}^{k \times n}} \|A - \widehat{Q}_k G_k\|_F.$$

- Let $G_k = [\boldsymbol{g}_1 \dots \boldsymbol{g}_n] \in \mathbb{R}^{k \times n}$. Then the above is equivalent to the following set of the LS problems:

$$\min_{\boldsymbol{g}_j \in \mathbb{R}^k} \|\boldsymbol{a}_j - \widehat{Q}_k \boldsymbol{g}_j\|_2, \quad j = 1 : n.$$

Using Cluster Centroids for Text Mining

- Instead of using the left singular vectors as a basis to approximate a term-document matrix, let's examine the cluster centers (centroids) obtained by k-means algorithm as a basis.
- Let $C_k = [\boldsymbol{c}_1 \dots \boldsymbol{c}_k] \in \mathbb{R}^{m \times k}$ be the k cluster centroids obtained by the k-means algorithm.
- $\boldsymbol{c}_j$'s are non-orthogonal; hence it is more convenient to obtain a set of orthonormal vectors that spans range(C_k).
- To do so, we can use the *reduced QR factorization*: $C_k = \widehat{Q}_k \widehat{R}_k$ where $\widehat{Q}_k \in \mathbb{R}^{m \times k}$, and $\widehat{R}_k \in \mathbb{R}^{k \times k}$.
- Now, let's approximate A using $\widehat{Q}_k$ in the sense of the least squares as:

$$\min_{G_k \in \mathbb{R}^{k \times n}} \|A - \widehat{Q}_k G_k\|_F.$$

- Let $G_k = [\boldsymbol{g}_1 \dots \boldsymbol{g}_n] \in \mathbb{R}^{k \times n}$. Then the above is equivalent to the following set of the LS problems:

$$\min_{\boldsymbol{g}_j \in \mathbb{R}^k} \|\boldsymbol{a}_j - \widehat{Q}_k \boldsymbol{g}_j\|_2, \quad j = 1 : n.$$

Using Cluster Centroids for Text Mining

- Instead of using the left singular vectors as a basis to approximate a term-document matrix, let's examine the cluster centers (centroids) obtained by k-means algorithm as a basis.
- Let $C_k = [\boldsymbol{c}_1 \dots \boldsymbol{c}_k] \in \mathbb{R}^{m \times k}$ be the k cluster centroids obtained by the k-means algorithm.
- $\boldsymbol{c}_j$'s are non-orthogonal; hence it is more convenient to obtain a set of orthonormal vectors that spans range(C_k).
- To do so, we can use the *reduced QR factorization*: $C_k = \widehat{Q}_k \widehat{R}_k$ where $\widehat{Q}_k \in \mathbb{R}^{m \times k}$, and $\widehat{R}_k \in \mathbb{R}^{k \times k}$.
- Now, let's approximate A using $\widehat{Q}_k$ in the sense of the least squares as:

$$\min_{G_k \in \mathbb{R}^{k \times n}} \|A - \widehat{Q}_k G_k\|_F.$$

- Let $G_k = [\boldsymbol{g}_1 \dots \boldsymbol{g}_n] \in \mathbb{R}^{k \times n}$. Then the above is equivalent to the following set of the LS problems:

$$\min_{\boldsymbol{g}_j \in \mathbb{R}^k} \|\boldsymbol{a}_j - \widehat{Q}_k \boldsymbol{g}_j\|_2, \quad j = 1 : n.$$

Using Cluster Centroids for Text Mining...

- Since the columns of $\widehat{Q}_k$ are orthonormal, we can get the following LS solution: $\boldsymbol{g}_j = \widehat{Q}_k^\mathsf{T} \boldsymbol{a}_j$, $j = 1 : n$. Hence $G_k = \widehat{Q}_k^\mathsf{T} A$.
- The inner product between the query vector $\boldsymbol{q}$ and the document vector $\boldsymbol{a}_j$ can be approximated as:

$$\boldsymbol{q}^\mathsf{T} \boldsymbol{a}_j \approx \boldsymbol{q}^\mathsf{T} \widehat{Q}_k \boldsymbol{g}_j = (\widehat{Q}_k^\mathsf{T} \boldsymbol{q})^\mathsf{T} \boldsymbol{g}_j = \boldsymbol{q}_k^\mathsf{T} \boldsymbol{g}_j, \; \boldsymbol{q}_k := \widehat{Q}_k^\mathsf{T} \boldsymbol{q}.$$

- Hence, the cosine similarity can be approximated as:

$$\frac{\boldsymbol{q}^\mathsf{T} \boldsymbol{a}_j}{\|\boldsymbol{q}\|_2 \|\boldsymbol{a}_j\|_2} \approx \frac{\boldsymbol{q}_k^\mathsf{T} \boldsymbol{g}_j}{\|\boldsymbol{q}\|_2 \|\boldsymbol{g}_j\|_2}.$$

Using Cluster Centroids for Text Mining...

- Since the columns of $\widehat{Q}_k$ are orthonormal, we can get the following LS solution: $\boldsymbol{g}_j = \widehat{Q}_k^{\mathsf{T}} \boldsymbol{a}_j$, $j = 1:n$. Hence $G_k = \widehat{Q}_k^{\mathsf{T}} A$.
- The inner product between the query vector $\boldsymbol{q}$ and the document vector $\boldsymbol{a}_j$ can be approximated as:

$$\boldsymbol{q}^{\mathsf{T}} \boldsymbol{a}_j \approx \boldsymbol{q}^{\mathsf{T}} \widehat{Q}_k \boldsymbol{g}_j = (\widehat{Q}_k^{\mathsf{T}} \boldsymbol{q})^{\mathsf{T}} \boldsymbol{g}_j = \boldsymbol{q}_k^{\mathsf{T}} \boldsymbol{g}_j, \; \boldsymbol{q}_k := \widehat{Q}_k^{\mathsf{T}} \boldsymbol{q}.$$

- Hence, the cosine similarity can be approximated as:

$$\frac{\boldsymbol{q}^{\mathsf{T}} \boldsymbol{a}_j}{\|\boldsymbol{q}\|_2 \|\boldsymbol{a}_j\|_2} \approx \frac{\boldsymbol{q}_k^{\mathsf{T}} \boldsymbol{g}_j}{\|\boldsymbol{q}\|_2 \|\boldsymbol{g}_j\|_2}.$$

Using Cluster Centroids for Text Mining...

- Since the columns of $\widehat{Q}_k$ are orthonormal, we can get the following LS solution: $\boldsymbol{g}_j = \widehat{Q}_k^{\mathsf{T}} \boldsymbol{a}_j$, $j = 1:n$. Hence $G_k = \widehat{Q}_k^{\mathsf{T}} A$.
- The inner product between the query vector $\boldsymbol{q}$ and the document vector $\boldsymbol{a}_j$ can be approximated as:

$$\boldsymbol{q}^{\mathsf{T}} \boldsymbol{a}_j \approx \boldsymbol{q}^{\mathsf{T}} \widehat{Q}_k \boldsymbol{g}_j = (\widehat{Q}_k^{\mathsf{T}} \boldsymbol{q})^{\mathsf{T}} \boldsymbol{g}_j = \boldsymbol{q}_k^{\mathsf{T}} \boldsymbol{g}_j, \; \boldsymbol{q}_k := \widehat{Q}_k^{\mathsf{T}} \boldsymbol{q}.$$

- Hence, the cosine similarity can be approximated as:

$$\frac{\boldsymbol{q}^{\mathsf{T}} \boldsymbol{a}_j}{\|\boldsymbol{q}\|_2 \|\boldsymbol{a}_j\|_2} \approx \frac{\boldsymbol{q}_k^{\mathsf{T}} \boldsymbol{g}_j}{\|\boldsymbol{q}\|_2 \|\boldsymbol{g}_j\|_2}.$$

An Example Trial with the NIPS Data

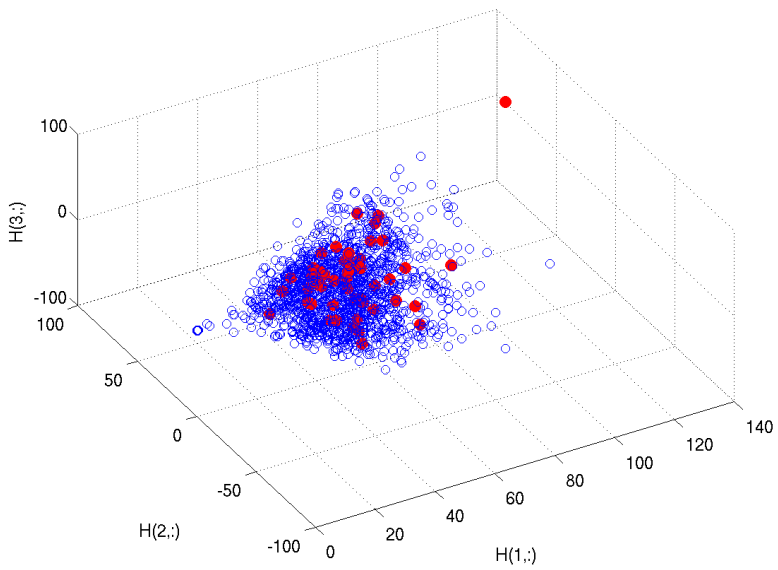
- $k = 50$; the same query vector ('entropy', 'minimum', 'maximum').
- The approximation error between $\widehat{Q}_k G_k$ and A was $\|A - \widehat{Q}_k G_k\|_F / \|A\|_F \approx 0.7227$, which was worse than that using the top 100 SVD basis.

An Example Trial with the NIPS Data

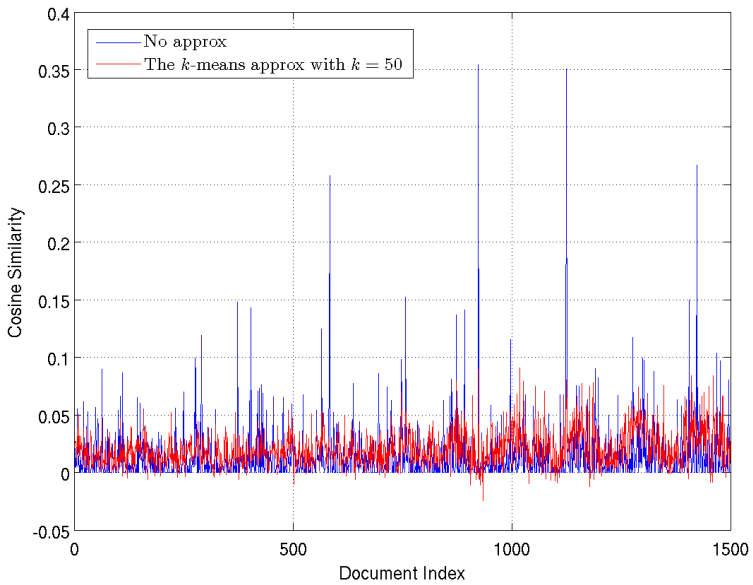
- $k = 50$; the same query vector ('entropy', 'minimum', 'maximum').
- The approximation error between $\widehat{Q}_k G_k$ and A was $\|A - \widehat{Q}_k G_k\|_F / \|A\|_F \approx 0.7227$, which was worse than that using the top 100 SVD basis.

An Example Trial with the NIPS Data

- $k = 50$; the same query vector ('entropy', 'minimum', 'maximum').
- The approximation error between $\widehat{Q}_k G_k$ and A was $\|A - \widehat{Q}_k G_k\|_F / \|A\|_F \approx 0.7227$, which was worse than that using the top 100 SVD basis.



Documents in $U_{100}[:, 1:3]$



Cosine Similarity

With the 50-means based approximation, tol=0.2, 0.1, 0.05 correspond to 0, 0, 81 returned documents; Compare these with the no approximation case: 4, 15, 89; and with the best 100 approximation using SVD: 0, 4, 72.

My Reaction

- Running the k-means algorithm with large m and n is slow in general.
- If your document set really consists of k different topics (or categories), then this k-means-based approach should work well. Example: *The Science News Dataset* consisting of articles in the area of *Anthropology, Astronomy, Behavioral Sciences, Earth Sciences, Life Sciences, Math & CS, Medicine, Physics*. Which value of k should be used is still a question though.
- However, in the case of the NIPS data where there is not much clustering structure, it may not worth trying this approach considering the computational cost.

My Reaction

- Running the k-means algorithm with large m and n is slow in general.
- If your document set really consists of k different topics (or categories), then this k-means-based approach should work well. Example: *The Science News Dataset* consisting of articles in the area of *Anthropology, Astronomy, Behavioral Sciences, Earth Sciences, Life Sciences, Math & CS, Medicine, Physics*. Which value of k should be used is still a question though.
- However, in the case of the NIPS data where there is not much clustering structure, it may not worth trying this approach considering the computational cost.

My Reaction

- Running the k-means algorithm with large m and n is slow in general.
- If your document set really consists of k different topics (or categories), then this k-means-based approach should work well. Example: *The Science News Dataset* consisting of articles in the area of *Anthropology, Astronomy, Behavioral Sciences, Earth Sciences, Life Sciences, Math & CS, Medicine, Physics*. Which value of k should be used is still a question though.
- However, in the case of the NIPS data where there is not much clustering structure, it may not worth trying this approach considering the computational cost.

Outline

Using NNMF for Text Mining

- Consider the NNMF of a term-document matrix $A \approx WH$ where $W \in \mathbb{R}^{m \times k}$, $H \in \mathbb{R}^{k \times n}$, $1 < k \leq \min(m, n)$.
- We want to represent (or approximate) both query vectors and the term-document matrix using the basis vectors $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k\}$ and do query task in that basis (or coordinates).
- $\boldsymbol{a}_j$ is already approximated using $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k\}$ with the coordinate vector $\boldsymbol{h}_j$, $j = 1 : n$, i.e., $\boldsymbol{a}_j \approx W\boldsymbol{h}_j$.
- We need to approximate $\boldsymbol{q}$ in the basis of W. To do so, we seek the LS approximation of $\boldsymbol{q}$ in range(W), i.e., $\min_{\widehat{\boldsymbol{q}} \in \mathbb{R}^k} \|\boldsymbol{q} - W\widehat{\boldsymbol{q}}\|_2$.
- Hence we need to solve the normal equation: $W^{\mathsf{T}} W \widehat{\boldsymbol{q}} = W^{\mathsf{T}} \boldsymbol{q}$.
- To do so, we use the reduced QR factorization of $W = \widehat{Q}\widehat{R}$.
- Then, using the argument of Lecture 10, the normal equation above is equivalent to $\widehat{R}\widehat{\boldsymbol{q}} = \widehat{Q}^{\mathsf{T}} \boldsymbol{q}$, i.e., $\widehat{\boldsymbol{q}} = \widehat{R}^{-1} \widehat{Q}^{\mathsf{T}} \boldsymbol{q}$.
- The cosine similarity in the basis of $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k\}$ can be written as:

$$\frac{\widehat{\boldsymbol{q}}^{\mathsf{T}} \boldsymbol{h}_j}{\|\widehat{\boldsymbol{q}}\|_2 \|\boldsymbol{h}_j\|_2}, \quad j = 1 : n.$$

Using NNMF for Text Mining

- Consider the NNMF of a term-document matrix $A \approx WH$ where $W \in \mathbb{R}^{m \times k}$, $H \in \mathbb{R}^{k \times n}$, $1 < k \leq \min(m, n)$.
- We want to represent (or approximate) both query vectors and the term-document matrix using the basis vectors $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k\}$ and do query task in that basis (or coordinates).
- $\boldsymbol{a}_j$ is already approximated using $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k\}$ with the coordinate vector $\boldsymbol{h}_j$, $j = 1 : n$, i.e., $\boldsymbol{a}_j \approx W\boldsymbol{h}_j$.
- We need to approximate $\boldsymbol{q}$ in the basis of W. To do so, we seek the LS approximation of $\boldsymbol{q}$ in range(W), i.e., $\min_{\widehat{\boldsymbol{q}} \in \mathbb{R}^k} \|\boldsymbol{q} - W\widehat{\boldsymbol{q}}\|_2$.
- Hence we need to solve the normal equation: $W^{\mathsf{T}} W \widehat{\boldsymbol{q}} = W^{\mathsf{T}} \boldsymbol{q}$.
- To do so, we use the reduced QR factorization of $W = \widehat{Q}\widehat{R}$.
- Then, using the argument of Lecture 10, the normal equation above is equivalent to $\widehat{R}\widehat{\boldsymbol{q}} = \widehat{Q}^{\mathsf{T}} \boldsymbol{q}$, i.e., $\widehat{\boldsymbol{q}} = \widehat{R}^{-1} \widehat{Q}^{\mathsf{T}} \boldsymbol{q}$.
- The cosine similarity in the basis of $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k\}$ can be written as:

$$\frac{\widehat{\boldsymbol{q}}^{\mathsf{T}} \boldsymbol{h}_j}{\|\widehat{\boldsymbol{q}}\|_2 \|\boldsymbol{h}_j\|_2}, \quad j = 1 : n.$$

Using NNMF for Text Mining

- Consider the NNMF of a term-document matrix $A \approx WH$ where $W \in \mathbb{R}^{m \times k}$, $H \in \mathbb{R}^{k \times n}$, $1 < k \leq \min(m, n)$.
- We want to represent (or approximate) both query vectors and the term-document matrix using the basis vectors $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k\}$ and do query task in that basis (or coordinates).
- $\boldsymbol{a}_j$ is already approximated using $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k\}$ with the coordinate vector $\boldsymbol{h}_j$, $j = 1 : n$, i.e., $\boldsymbol{a}_j \approx W\boldsymbol{h}_j$.
- We need to approximate $\boldsymbol{q}$ in the basis of W. To do so, we seek the LS approximation of $\boldsymbol{q}$ in range(W), i.e., $\min_{\widehat{\boldsymbol{q}} \in \mathbb{R}^k} \|\boldsymbol{q} - W\widehat{\boldsymbol{q}}\|_2$.
- Hence we need to solve the normal equation: $W^{\mathsf{T}} W \widehat{\boldsymbol{q}} = W^{\mathsf{T}} \boldsymbol{q}$.
- To do so, we use the reduced QR factorization of $W = \widehat{Q}\widehat{R}$.
- Then, using the argument of Lecture 10, the normal equation above is equivalent to $\widehat{R}\widehat{\boldsymbol{q}} = \widehat{Q}^{\mathsf{T}} \boldsymbol{q}$, i.e., $\widehat{\boldsymbol{q}} = \widehat{R}^{-1} \widehat{Q}^{\mathsf{T}} \boldsymbol{q}$.
- The cosine similarity in the basis of $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k\}$ can be written as:

$$\frac{\widehat{\boldsymbol{q}}^{\mathsf{T}} \boldsymbol{h}_j}{\|\widehat{\boldsymbol{q}}\|_2 \|\boldsymbol{h}_j\|_2}, \quad j = 1 : n.$$

Using NNMF for Text Mining

- Consider the NNMF of a term-document matrix $A \approx WH$ where $W \in \mathbb{R}^{m \times k}$, $H \in \mathbb{R}^{k \times n}$, $1 < k \leq \min(m, n)$.
- We want to represent (or approximate) both query vectors and the term-document matrix using the basis vectors $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k\}$ and do query task in that basis (or coordinates).
- $\boldsymbol{a}_j$ is already approximated using $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k\}$ with the coordinate vector $\boldsymbol{h}_j$, $j = 1 : n$, i.e., $\boldsymbol{a}_j \approx W\boldsymbol{h}_j$.
- We need to approximate $\boldsymbol{q}$ in the basis of W. To do so, we seek the LS approximation of $\boldsymbol{q}$ in range(W), i.e., $\min_{\widehat{\boldsymbol{q}} \in \mathbb{R}^k} \|\boldsymbol{q} - W\widehat{\boldsymbol{q}}\|_2$.
- Hence we need to solve the normal equation: $W^{\mathsf{T}} W \widehat{\boldsymbol{q}} = W^{\mathsf{T}} \boldsymbol{q}$.
- To do so, we use the reduced QR factorization of $W = \widehat{Q}\widehat{R}$.
- Then, using the argument of Lecture 10, the normal equation above is equivalent to $\widehat{R}\widehat{\boldsymbol{q}} = \widehat{Q}^{\mathsf{T}} \boldsymbol{q}$, i.e., $\widehat{\boldsymbol{q}} = \widehat{R}^{-1} \widehat{Q}^{\mathsf{T}} \boldsymbol{q}$.
- The cosine similarity in the basis of $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k\}$ can be written as:

$$\frac{\widehat{\boldsymbol{q}}^{\mathsf{T}} \boldsymbol{h}_j}{\|\widehat{\boldsymbol{q}}\|_2 \|\boldsymbol{h}_j\|_2}, \quad j = 1 : n.$$

Using NNMF for Text Mining

- Consider the NNMF of a term-document matrix $A \approx WH$ where $W \in \mathbb{R}^{m \times k}$, $H \in \mathbb{R}^{k \times n}$, $1 < k \leq \min(m, n)$.
- We want to represent (or approximate) both query vectors and the term-document matrix using the basis vectors $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k\}$ and do query task in that basis (or coordinates).
- $\boldsymbol{a}_j$ is already approximated using $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k\}$ with the coordinate vector $\boldsymbol{h}_j$, $j = 1 : n$, i.e., $\boldsymbol{a}_j \approx W\boldsymbol{h}_j$.
- We need to approximate $\boldsymbol{q}$ in the basis of W. To do so, we seek the LS approximation of $\boldsymbol{q}$ in range(W), i.e., $\min_{\widehat{\boldsymbol{q}} \in \mathbb{R}^k} \|\boldsymbol{q} - W\widehat{\boldsymbol{q}}\|_2$.
- Hence we need to solve the normal equation: $W^{\mathsf{T}} W \widehat{\boldsymbol{q}} = W^{\mathsf{T}} \boldsymbol{q}$.
- To do so, we use the reduced QR factorization of $W = \widehat{Q}\widehat{R}$.
- Then, using the argument of Lecture 10, the normal equation above is equivalent to $\widehat{R}\widehat{\boldsymbol{q}} = \widehat{Q}^{\mathsf{T}} \boldsymbol{q}$, i.e., $\widehat{\boldsymbol{q}} = \widehat{R}^{-1} \widehat{Q}^{\mathsf{T}} \boldsymbol{q}$.
- The cosine similarity in the basis of $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k\}$ can be written as:

$$\frac{\widehat{\boldsymbol{q}}^{\mathsf{T}} \boldsymbol{h}_j}{\|\widehat{\boldsymbol{q}}\|_2 \|\boldsymbol{h}_j\|_2}, \quad j = 1 : n.$$

Using NNMF for Text Mining

- Consider the NNMF of a term-document matrix $A \approx WH$ where $W \in \mathbb{R}^{m \times k}$, $H \in \mathbb{R}^{k \times n}$, $1 < k \leq \min(m, n)$.
- We want to represent (or approximate) both query vectors and the term-document matrix using the basis vectors $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k\}$ and do query task in that basis (or coordinates).
- $\boldsymbol{a}_j$ is already approximated using $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k\}$ with the coordinate vector $\boldsymbol{h}_j$, $j = 1 : n$, i.e., $\boldsymbol{a}_j \approx W\boldsymbol{h}_j$.
- We need to approximate $\boldsymbol{q}$ in the basis of W. To do so, we seek the LS approximation of $\boldsymbol{q}$ in range(W), i.e., $\min_{\widehat{\boldsymbol{q}} \in \mathbb{R}^k} \|\boldsymbol{q} - W\widehat{\boldsymbol{q}}\|_2$.
- Hence we need to solve the normal equation: $W^{\mathsf{T}} W \widehat{\boldsymbol{q}} = W^{\mathsf{T}} \boldsymbol{q}$.
- To do so, we use the reduced QR factorization of $W = \widehat{Q}\widehat{R}$.
- Then, using the argument of Lecture 10, the normal equation above is equivalent to $\widehat{R}\widehat{\boldsymbol{q}} = \widehat{Q}^{\mathsf{T}} \boldsymbol{q}$, i.e., $\widehat{\boldsymbol{q}} = \widehat{R}^{-1} \widehat{Q}^{\mathsf{T}} \boldsymbol{q}$.
- The cosine similarity in the basis of $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k\}$ can be written as:

$$\frac{\widehat{\boldsymbol{q}}^{\mathsf{T}} \boldsymbol{h}_j}{\|\widehat{\boldsymbol{q}}\|_2 \|\boldsymbol{h}_j\|_2}, \quad j = 1 : n.$$

Using NNMF for Text Mining

- Consider the NNMF of a term-document matrix $A \approx WH$ where $W \in \mathbb{R}^{m \times k}$, $H \in \mathbb{R}^{k \times n}$, $1 < k \leq \min(m, n)$.
- We want to represent (or approximate) both query vectors and the term-document matrix using the basis vectors $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k\}$ and do query task in that basis (or coordinates).
- $\boldsymbol{a}_j$ is already approximated using $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k\}$ with the coordinate vector $\boldsymbol{h}_j$, $j = 1 : n$, i.e., $\boldsymbol{a}_j \approx W\boldsymbol{h}_j$.
- We need to approximate $\boldsymbol{q}$ in the basis of W. To do so, we seek the LS approximation of $\boldsymbol{q}$ in range(W), i.e., $\min_{\widehat{\boldsymbol{q}} \in \mathbb{R}^k} \|\boldsymbol{q} - W\widehat{\boldsymbol{q}}\|_2$.
- Hence we need to solve the normal equation: $W^{\mathsf{T}} W \widehat{\boldsymbol{q}} = W^{\mathsf{T}} \boldsymbol{q}$.
- To do so, we use the reduced QR factorization of $W = \widehat{Q}\widehat{R}$.
- Then, using the argument of Lecture 10, the normal equation above is equivalent to $\widehat{R}\widehat{\boldsymbol{q}} = \widehat{Q}^{\mathsf{T}} \boldsymbol{q}$, i.e., $\widehat{\boldsymbol{q}} = \widehat{R}^{-1} \widehat{Q}^{\mathsf{T}} \boldsymbol{q}$.
- The cosine similarity in the basis of $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k\}$ can be written as:

$$\frac{\widehat{\boldsymbol{q}}^{\mathsf{T}} \boldsymbol{h}_j}{\|\widehat{\boldsymbol{q}}\|_2 \|\boldsymbol{h}_j\|_2}, \quad j = 1 : n.$$

Using NNMF for Text Mining

- Consider the NNMF of a term-document matrix $A \approx WH$ where $W \in \mathbb{R}^{m\times k}$, $H \in \mathbb{R}^{k\times n}$, $1 < k \leq \min(m,n)$.
- We want to represent (or approximate) both query vectors and the term-document matrix using the basis vectors $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k\}$ and do query task in that basis (or coordinates).
- $\boldsymbol{a}_j$ is already approximated using $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k\}$ with the coordinate vector $\boldsymbol{h}_j$, $j = 1:n$, i.e., $\boldsymbol{a}_j \approx W\boldsymbol{h}_j$.
- We need to approximate $\boldsymbol{q}$ in the basis of W. To do so, we seek the LS approximation of $\boldsymbol{q}$ in range(W), i.e., $\min_{\widehat{\boldsymbol{q}}\in\mathbb{R}^k} \|\boldsymbol{q} - W\widehat{\boldsymbol{q}}\|_2$.
- Hence we need to solve the normal equation: $W^{\mathsf{T}}W\widehat{\boldsymbol{q}} = W^{\mathsf{T}}\boldsymbol{q}$.
- To do so, we use the reduced QR factorization of $W = \widehat{Q}\widehat{R}$.
- Then, using the argument of Lecture 10, the normal equation above is equivalent to $\widehat{R}\widehat{\boldsymbol{q}} = \widehat{Q}^{\mathsf{T}}\boldsymbol{q}$, i.e., $\widehat{\boldsymbol{q}} = \widehat{R}^{-1}\widehat{Q}^{\mathsf{T}}\boldsymbol{q}$.
- The cosine similarity in the basis of $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k\}$ can be written as:

$$\frac{\widehat{\boldsymbol{q}}^{\mathsf{T}}\boldsymbol{h}_j}{\|\widehat{\boldsymbol{q}}\|_2\|\boldsymbol{h}_j\|_2}, \quad j = 1:n.$$

An Example Trial with the NIPS Data

- $k = 100$ was used.
- $\|A - WH\|_F / \|A\|_F \approx 0.6302$, which was *slightly* worse than that using the top 100 SVD basis (0.6074).
- Each $\boldsymbol{w}_j$ concentrates on one term, and is close to the canonical vector $\boldsymbol{e}_i \in \mathbb{R}^m$ for some i (recall: NNMF applied to the face database in Lecture 20).
- The peaks of $\boldsymbol{w}_j$, $j = 1:10$, correspond to: 'network', 'model', 'learning', 'function', 'unit', 'algorithm', 'input', 'data', 'neuron', 'cell', which are quite similar to the $\boldsymbol{u}_1$ vector or the 10 most frequently used terms.
- On the other hand, because $\boldsymbol{w}_j$'s are localized, the interpretation of the row vectors of H matrix becomes easy.

An Example Trial with the NIPS Data

- $k = 100$ was used.
- $\|A - WH\|_F / \|A\|_F \approx 0.6302$, which was *slightly* worse than that using the top 100 SVD basis (0.6074).
- Each $\boldsymbol{w}_j$ concentrates on one term, and is close to the canonical vector $\boldsymbol{e}_i \in \mathbb{R}^m$ for some i (recall: NNMF applied to the face database in Lecture 20).
- The peaks of $\boldsymbol{w}_j$, $j = 1:10$, correspond to: 'network', 'model', 'learning', 'function', 'unit', 'algorithm', 'input', 'data', 'neuron', 'cell', which are quite similar to the $\boldsymbol{u}_1$ vector or the 10 most frequently used terms.
- On the other hand, because $\boldsymbol{w}_j$'s are localized, the interpretation of the row vectors of H matrix becomes easy.

An Example Trial with the NIPS Data

- $k = 100$ was used.
- $\|A - WH\|_F / \|A\|_F \approx 0.6302$, which was *slightly* worse than that using the top 100 SVD basis (0.6074).
- Each $\boldsymbol{w}_j$ concentrates on one term, and is close to the canonical vector $\boldsymbol{e}_i \in \mathbb{R}^m$ for some i (recall: NNMF applied to the face database in Lecture 20).
- The peaks of $\boldsymbol{w}_j$, $j = 1:10$, correspond to: 'network', 'model', 'learning', 'function', 'unit', 'algorithm', 'input', 'data', 'neuron', 'cell', which are quite similar to the $\boldsymbol{u}_1$ vector or the 10 most frequently used terms.
- On the other hand, because $\boldsymbol{w}_j$'s are localized, the interpretation of the row vectors of H matrix becomes easy.

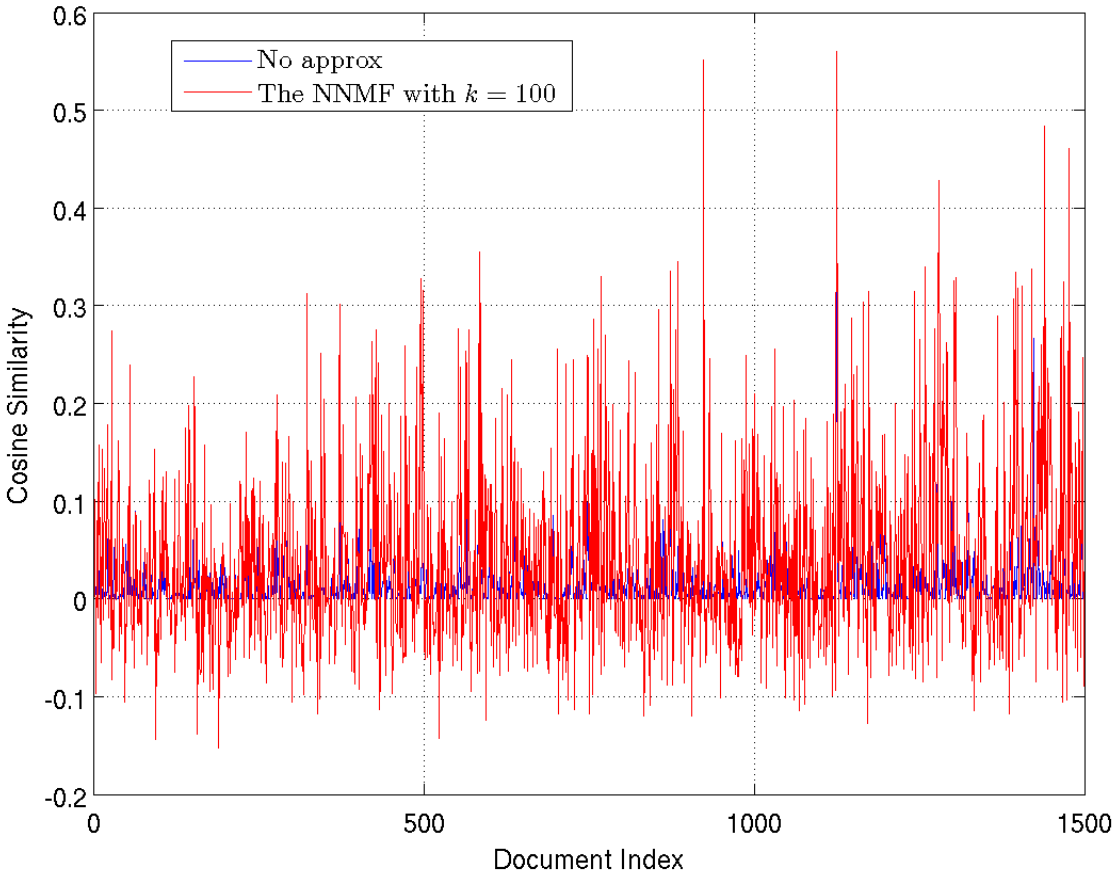
An Example Trial with the NIPS Data

- $k = 100$ was used.
- $\|A - WH\|_F / \|A\|_F \approx 0.6302$, which was *slightly* worse than that using the top 100 SVD basis (0.6074).
- Each $\boldsymbol{w}_j$ concentrates on one term, and is close to the canonical vector $\boldsymbol{e}_i \in \mathbb{R}^m$ for some i (recall: NNMF applied to the face database in Lecture 20).
- The peaks of $\boldsymbol{w}_j$, $j = 1:10$, correspond to: 'network', 'model', 'learning', 'function', 'unit', 'algorithm', 'input', 'data', 'neuron', 'cell', which are quite similar to the $\boldsymbol{u}_1$ vector or the 10 most frequently used terms.
- On the other hand, because $\boldsymbol{w}_j$'s are localized, the interpretation of the row vectors of H matrix becomes easy.

An Example Trial with the NIPS Data

- $k = 100$ was used.
- $\|A - WH\|_F / \|A\|_F \approx 0.6302$, which was *slightly* worse than that using the top 100 SVD basis (0.6074).
- Each $\boldsymbol{w}_j$ concentrates on one term, and is close to the canonical vector $\boldsymbol{e}_i \in \mathbb{R}^m$ for some i (recall: NNMF applied to the face database in Lecture 20).
- The peaks of $\boldsymbol{w}_j$, $j = 1:10$, correspond to: 'network', 'model', 'learning', 'function', 'unit', 'algorithm', 'input', 'data', 'neuron', 'cell', which are quite similar to the $\boldsymbol{u}_1$ vector or the 10 most frequently used terms.
- On the other hand, because $\boldsymbol{w}_j$'s are localized, the interpretation of the row vectors of H matrix becomes easy.

An Example Trial with the NIPS Data ...



With the NNMF-based approach using $k = 100$, tol=0.2, 0.1, 0.05 correspond to 101, 312, 535 returned documents; Compare with the no approximation case: 4, 15, 89. Changing the tol=0.4, 0.3, 0.2 with NNMF returns 5, 26, 101 documents.

My Reaction

- Using the LS solution for the query saves computational cost given the NNMF is already obtained because one can avoid the explicit computation and storage of WH.
- If we can compute and store WH, then we could use the following approximation of the original cosine similarity:

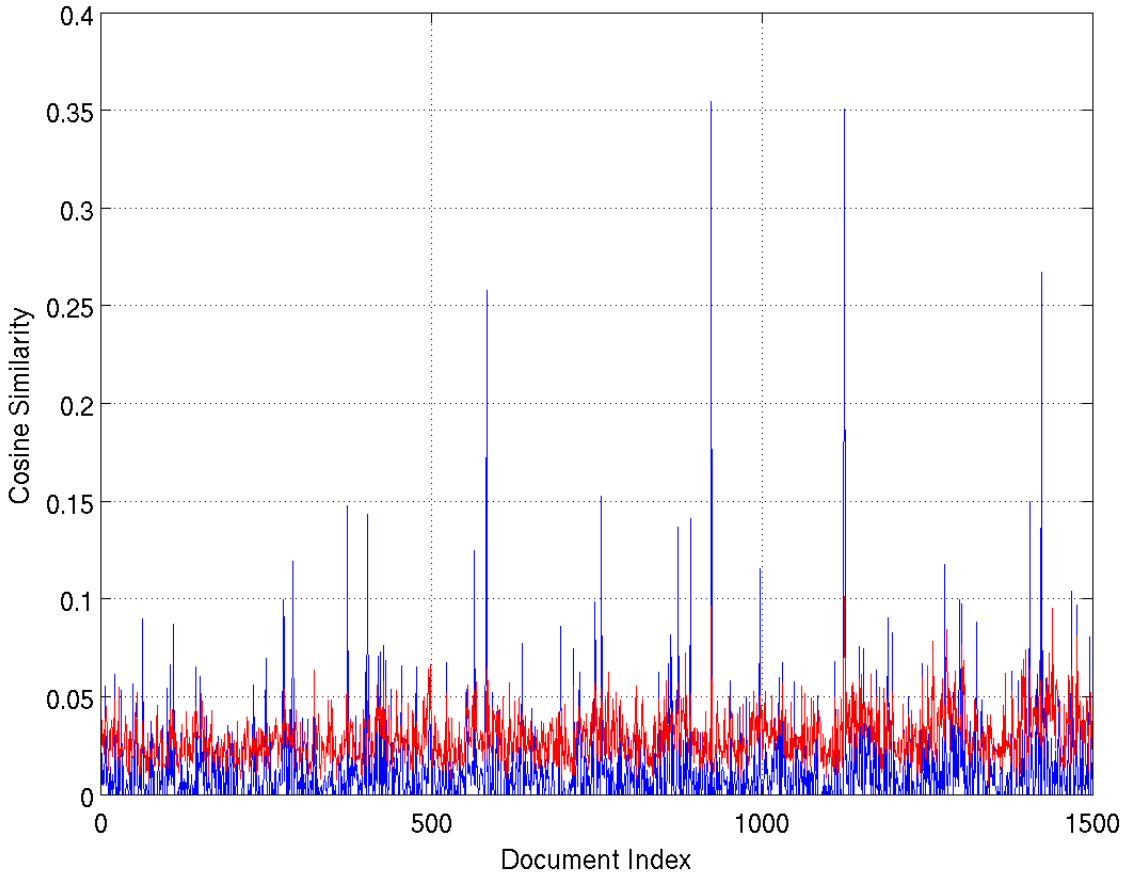
$$\frac{\boldsymbol{q}^\top \boldsymbol{a}_j}{\|\boldsymbol{q}\|_2 \|\boldsymbol{a}_j\|_2} \approx \frac{\boldsymbol{q}^\top W \boldsymbol{h}_j}{\|\boldsymbol{q}\|_2 \|W \boldsymbol{h}_j\|_2}.$$

My Reaction

- Using the LS solution for the query saves computational cost given the NNMF is already obtained because one can avoid the explicit computation and storage of WH.
- If we can compute and store WH, then we could use the following approximation of the original cosine similarity:

$$\frac{\boldsymbol{q}^\top \boldsymbol{a}_j}{\|\boldsymbol{q}\|_2 \|\boldsymbol{a}_j\|_2} \approx \frac{\boldsymbol{q}^\top W \boldsymbol{h}_j}{\|\boldsymbol{q}\|_2 \|W \boldsymbol{h}_j\|_2}.$$

My Reaction ...



With the NNMF-based approach using $k = 100$ using the above cosine similarity approximation, tol=0.2, 0.1, 0.05 correspond to 0, 1, 97 returned documents; Compare with the no approximation case: 4, 15, 89. Without using the LS query, some of the relevant documents do not stick out clearly.