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

Naoki Saito

Department of Mathematics
University of California, Davis

May 30/June 1, 2012

Outline

1. Clustering
2. Nonnegative Matrix Factorization

Outline

1. Clustering

2. Nonnegative Matrix Factorization

- 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 $C_k = [\mathbf{c}_1 \ \ldots \ \mathbf{c}_k] \in \mathbb{R}^{m \times k}$ be the k cluster centroids obtained by the k-means algorithm.
- $\mathbf{c}_j$'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: $C_k = Q_k R_k$ where $Q_k \in \mathbb{R}^{m \times k}$, and $R_k \in \mathbb{R}^{k \times k}$.
- Now, let's approximate A using Q_k in the sense of the least squares as:

$$\min_{G_k \in \mathbb{R}^{k \times k}} \|A - Q_k G_k\|_F.$$

- Let $G_k = [\mathbf{g}_1 \ \ldots \ \mathbf{g}_k]$. Then the above is equivalent to the following set of the LS problems:

$$\min_{\mathbf{g}_j \in \mathbb{R}^k} \|\mathbf{a}_j - Q_k \mathbf{g}_j\|_2, \ j = 1 : k.$$

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 $C_k = [\mathbf{c}_1 \; \ldots \; \mathbf{c}_k] \in \mathbb{R}^{m \times k}$ be the k cluster centroids obtained by the k-means algorithm.
- $\mathbf{c}_j$'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: $C_k = Q_k R_k$ where $Q_k \in \mathbb{R}^{m \times k}$, and $R_k \in \mathbb{R}^{k \times k}$.
- Now, let's approximate A using Q_k in the sense of the least squares as:

$$\min_{G_k \in \mathbb{R}^{k \times k}} \|A - Q_k G_k\|_F.$$

- Let $G_k = [\mathbf{g}_1 \; \ldots \; \mathbf{g}_k]$. Then the above is equivalent to the following set of the LS problems:

$$\min_{\mathbf{g}_j \in \mathbb{R}^k} \|\mathbf{a}_j - Q_k \mathbf{g}_j\|_2, \; j = 1 : k.$$

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 $C_k = [\mathbf{c}_1 \; \ldots \; \mathbf{c}_k] \in \mathbb{R}^{m \times k}$ be the k cluster centroids obtained by the k-means algorithm.
- $\mathbf{c}_j$'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: $C_k = Q_k R_k$ where $Q_k \in \mathbb{R}^{m \times k}$, and $R_k \in \mathbb{R}^{k \times k}$.
- Now, let's approximate A using Q_k in the sense of the least squares as:

$$\min_{G_k \in \mathbb{R}^{k \times k}} \|A - Q_k G_k\|_F.$$

- Let $G_k = [\mathbf{g}_1 \; \ldots \; \mathbf{g}_k]$. Then the above is equivalent to the following set of the LS problems:

$$\min_{\mathbf{g}_j \in \mathbb{R}^k} \|\mathbf{a}_j - Q_k \mathbf{g}_j\|_2, \; j = 1 : k.$$

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 $C_k = [\mathbf{c}_1 \ \ldots \ \mathbf{c}_k] \in \mathbb{R}^{m \times k}$ be the k cluster centroids obtained by the k-means algorithm.
- $\mathbf{c}_j$'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: $C_k = Q_k R_k$ where $Q_k \in \mathbb{R}^{m \times k}$, and $R_k \in \mathbb{R}^{k \times k}$.
- Now, let's approximate A using Q_k in the sense of the least squares as:

$$\min_{G_k \in \mathbb{R}^{k \times k}} \|A - Q_k G_k\|_F.$$

- Let $G_k = [\mathbf{g}_1 \ \ldots \ \mathbf{g}_k]$. Then the above is equivalent to the following set of the LS problems:

$$\min_{\mathbf{g}_j \in \mathbb{R}^k} \|\mathbf{a}_j - Q_k \mathbf{g}_j\|_2, \ j = 1 : k.$$

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 $C_k = [\mathbf{c}_1 \ \ldots \ \mathbf{c}_k] \in \mathbb{R}^{m \times k}$ be the k cluster centroids obtained by the k-means algorithm.
- $\mathbf{c}_j$'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: $C_k = Q_k R_k$ where $Q_k \in \mathbb{R}^{m \times k}$, and $R_k \in \mathbb{R}^{k \times k}$.
- Now, let's approximate A using Q_k in the sense of the least squares as:

$$\min_{G_k \in \mathbb{R}^{k \times k}} \|A - Q_k G_k\|_F.$$

- Let $G_k = [\mathbf{g}_1 \ \ldots \ \mathbf{g}_k]$. Then the above is equivalent to the following set of the LS problems:

$$\min_{\mathbf{g}_j \in \mathbb{R}^k} \|\mathbf{a}_j - Q_k \mathbf{g}_j\|_2, \ j = 1 : k.$$

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 $C_k = [\mathbf{c}_1 \; \ldots \; \mathbf{c}_k] \in \mathbb{R}^{m \times k}$ be the k cluster centroids obtained by the k-means algorithm.
- $\mathbf{c}_j$'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: $C_k = Q_k R_k$ where $Q_k \in \mathbb{R}^{m \times k}$, and $R_k \in \mathbb{R}^{k \times k}$.
- Now, let's approximate A using Q_k in the sense of the least squares as:

$$\min_{G_k \in \mathbb{R}^{k \times k}} \|A - Q_k G_k\|_F.$$

- Let $G_k = [\mathbf{g}_1 \; \ldots \; \mathbf{g}_k]$. Then the above is equivalent to the following set of the LS problems:

$$\min_{\mathbf{g}_j \in \mathbb{R}^k} \|\mathbf{a}_j - Q_k \mathbf{g}_j\|_2, \; j = 1 : k.$$

Using Cluster Centroids for Text Mining...

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

$$\mathbf{q}^{\mathsf{T}} \mathbf{a}_j \approx \mathbf{q}^{\mathsf{T}} Q_k \mathbf{g}_j = (Q_k^{\mathsf{T}} \mathbf{q})^{\mathsf{T}} \mathbf{g}_j = \mathbf{q}_k^{\mathsf{T}} \mathbf{g}_j, \ \mathbf{q}_k := Q_k^{\mathsf{T}} \mathbf{q}.$$

- Hence, the cosine similarity can be approximated as:

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

Using Cluster Centroids for Text Mining...

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

$$\mathbf{q}^{\mathsf{T}} \mathbf{a}_j \approx \mathbf{q}^{\mathsf{T}} Q_k \mathbf{g}_j = (Q_k^{\mathsf{T}} \mathbf{q})^{\mathsf{T}} \mathbf{g}_j = \mathbf{q}_k^{\mathsf{T}} \mathbf{g}_j, \ \mathbf{q}_k := Q_k^{\mathsf{T}} \mathbf{q}.$$

- Hence, the cosine similarity can be approximated as:

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

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

$$\mathbf{q}^{\mathsf{T}} \mathbf{a}_j \approx \mathbf{q}^{\mathsf{T}} Q_k \mathbf{g}_j = (Q_k^{\mathsf{T}} \mathbf{q})^{\mathsf{T}} \mathbf{g}_j = \mathbf{q}_k^{\mathsf{T}} \mathbf{g}_j, \ \mathbf{q}_k := Q_k^{\mathsf{T}} \mathbf{q}.$$

- Hence, the cosine similarity can be approximated as:

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

An Example Trial with the NIPS Data

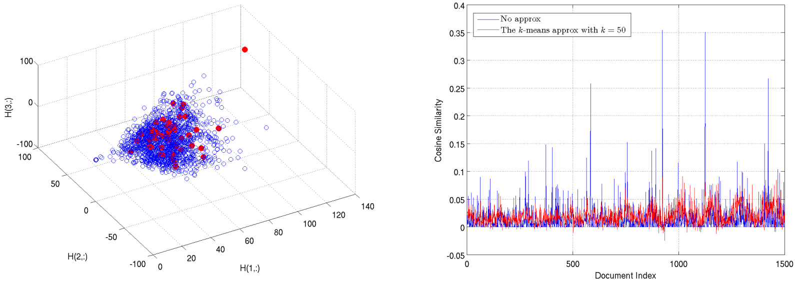
- $k = 50$; the same query vector ('entropy', 'minimum', 'maximum').
- The approximation error between $Q_k G_k$ and A was $\|A - Q_k G_k\|_F / \|A\|_F \approx 0.7227$, which was worse than that using the top 100 SVD basis.

An Example Trial with the NIPS Data

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



(a) Documents in $U_{100}(:, 1:3)$ (b) Cosine Similarity

Figure: 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 Reactiona

- 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

1. Clustering
2. Nonnegative Matrix Factorization

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 $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ and do query task in that basis (or coordinates).
- $\mathbf{a}_j$ is already approximated using $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ with the coordinate vector $\mathbf{h}_j$, $j = 1 : n$, i.e., $\mathbf{a}_j \approx W\mathbf{h}_j$.
- We need to approximate $\mathbf{q}$ in the basis of W. To do so, we seek the LS approximation of $\mathbf{q}$ in range(W), i.e., $\min_{\hat{\mathbf{q}} \in \mathbb{R}^k} \|\mathbf{q} - W\hat{\mathbf{q}}\|_2$.
- Hence we need to solve the normal equation: $W^{\mathsf{T}} W \hat{\mathbf{q}} = W^{\mathsf{T}} \mathbf{q}$.
- 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 $\widehat{R}\hat{\mathbf{q}} = \widehat{Q}^{\mathsf{T}}\mathbf{q}$, i.e., $\hat{\mathbf{q}} = \widehat{R}^{-1}\widehat{Q}^{\mathsf{T}}\mathbf{q}$.
- The cosine similarity in the basis of $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ can be written as:

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

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 $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ and do query task in that basis (or coordinates).
- $\mathbf{a}_j$ is already approximated using $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ with the coordinate vector $\mathbf{h}_j$, $j = 1 : n$, i.e., $\mathbf{a}_j \approx W\mathbf{h}_j$.
- We need to approximate $\mathbf{q}$ in the basis of W. To do so, we seek the LS approximation of $\mathbf{q}$ in range(W), i.e., $\min_{\hat{\mathbf{q}} \in \mathbb{R}^k} \|\mathbf{q} - W\hat{\mathbf{q}}\|_2$.
- Hence we need to solve the normal equation: $W^{\mathsf{T}} W \hat{\mathbf{q}} = W^{\mathsf{T}} \mathbf{q}$.
- 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 $\widehat{R}\hat{\mathbf{q}} = \widehat{Q}^{\mathsf{T}}\mathbf{q}$, i.e., $\hat{\mathbf{q}} = \widehat{R}^{-1}\widehat{Q}^{\mathsf{T}}\mathbf{q}$.
- The cosine similarity in the basis of $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ can be written as:

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

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 $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ and do query task in that basis (or coordinates).
- $\mathbf{a}_j$ is already approximated using $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ with the coordinate vector $\mathbf{h}_j$, $j = 1 : n$, i.e., $\mathbf{a}_j \approx W\mathbf{h}_j$.
- We need to approximate $\mathbf{q}$ in the basis of W. To do so, we seek the LS approximation of $\mathbf{q}$ in range(W), i.e., $\min_{\hat{\mathbf{q}} \in \mathbb{R}^k} \|\mathbf{q} - W\hat{\mathbf{q}}\|_2$.
- Hence we need to solve the normal equation: $W^\mathsf{T} W \hat{\mathbf{q}} = W^\mathsf{T}\mathbf{q}$.
- 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 $\widehat{R}\hat{\mathbf{q}} = \widehat{Q}^\mathsf{T}\mathbf{q}$, i.e., $\hat{\mathbf{q}} = \widehat{R}^{-1}\widehat{Q}^\mathsf{T}\mathbf{q}$.
- The cosine similarity in the basis of $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ can be written as:

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

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 $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ and do query task in that basis (or coordinates).
- $\mathbf{a}_j$ is already approximated using $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ with the coordinate vector $\mathbf{h}_j$, $j = 1 : n$, i.e., $\mathbf{a}_j \approx W\mathbf{h}_j$.
- We need to approximate $\mathbf{q}$ in the basis of W. To do so, we seek the LS approximation of $\mathbf{q}$ in range(W), i.e., $\min_{\hat{\mathbf{q}} \in \mathbb{R}^k} \|\mathbf{q} - W\hat{\mathbf{q}}\|_2$.
- Hence we need to solve the normal equation: $W^{\mathsf{T}} W \hat{\mathbf{q}} = W^{\mathsf{T}} \mathbf{q}$.
- 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 $\widehat{R}\hat{\mathbf{q}} = \widehat{Q}^{\mathsf{T}}\mathbf{q}$, i.e., $\hat{\mathbf{q}} = \widehat{R}^{-1}\widehat{Q}^{\mathsf{T}}\mathbf{q}$.
- The cosine similarity in the basis of $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ can be written as:

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

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 $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ and do query task in that basis (or coordinates).
- $\mathbf{a}_j$ is already approximated using $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ with the coordinate vector $\mathbf{h}_j$, $j = 1 : n$, i.e., $\mathbf{a}_j \approx W\mathbf{h}_j$.
- We need to approximate $\mathbf{q}$ in the basis of W. To do so, we seek the LS approximation of $\mathbf{q}$ in range(W), i.e., $\min_{\hat{\mathbf{q}} \in \mathbb{R}^k} \|\mathbf{q} - W\hat{\mathbf{q}}\|_2$.
- Hence we need to solve the normal equation: $W^\mathsf{T} W\hat{\mathbf{q}} = W^\mathsf{T}\mathbf{q}$.
- 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 $\widehat{R}\hat{\mathbf{q}} = \widehat{Q}^\mathsf{T}\mathbf{q}$, i.e., $\hat{\mathbf{q}} = \widehat{R}^{-1}\widehat{Q}^\mathsf{T}\mathbf{q}$.
- The cosine similarity in the basis of $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ can be written as:

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

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 $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ and do query task in that basis (or coordinates).
- $\mathbf{a}_j$ is already approximated using $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ with the coordinate vector $\mathbf{h}_j$, $j = 1 : n$, i.e., $\mathbf{a}_j \approx W\mathbf{h}_j$.
- We need to approximate $\mathbf{q}$ in the basis of W. To do so, we seek the LS approximation of $\mathbf{q}$ in range(W), i.e., $\min_{\hat{\mathbf{q}} \in \mathbb{R}^k} \|\mathbf{q} - W\hat{\mathbf{q}}\|_2$.
- Hence we need to solve the normal equation: $W^{\mathsf{T}} W \hat{\mathbf{q}} = W^{\mathsf{T}} \mathbf{q}$.
- 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 $\widehat{R}\hat{\mathbf{q}} = \widehat{Q}^{\mathsf{T}}\mathbf{q}$, i.e., $\hat{\mathbf{q}} = \widehat{R}^{-1}\widehat{Q}^{\mathsf{T}}\mathbf{q}$.
- The cosine similarity in the basis of $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ can be written as:

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

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 $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ and do query task in that basis (or coordinates).
- $\mathbf{a}_j$ is already approximated using $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ with the coordinate vector $\mathbf{h}_j$, $j = 1 : n$, i.e., $\mathbf{a}_j \approx W\mathbf{h}_j$.
- We need to approximate $\mathbf{q}$ in the basis of W. To do so, we seek the LS approximation of $\mathbf{q}$ in range(W), i.e., $\min_{\hat{\mathbf{q}} \in \mathbb{R}^k} \|\mathbf{q} - W\hat{\mathbf{q}}\|_2$.
- Hence we need to solve the normal equation: $W^{\mathsf{T}} W\hat{\mathbf{q}} = W^{\mathsf{T}}\mathbf{q}$.
- 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 $\widehat{R}\hat{\mathbf{q}} = \widehat{Q}^{\mathsf{T}}\mathbf{q}$, i.e., $\hat{\mathbf{q}} = \widehat{R}^{-1}\widehat{Q}^{\mathsf{T}}\mathbf{q}$.
- The cosine similarity in the basis of $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ can be written as:

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

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 $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ and do query task in that basis (or coordinates).
- $\mathbf{a}_j$ is already approximated using $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ with the coordinate vector $\mathbf{h}_j$, $j = 1 : n$, i.e., $\mathbf{a}_j \approx W\mathbf{h}_j$.
- We need to approximate $\mathbf{q}$ in the basis of W. To do so, we seek the LS approximation of $\mathbf{q}$ in range(W), i.e., $\min_{\hat{\mathbf{q}} \in \mathbb{R}^k} \|\mathbf{q} - W\hat{\mathbf{q}}\|_2$.
- Hence we need to solve the normal equation: $W^{\mathsf{T}} W \hat{\mathbf{q}} = W^{\mathsf{T}} \mathbf{q}$.
- 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 $\widehat{R}\hat{\mathbf{q}} = \widehat{Q}^{\mathsf{T}}\mathbf{q}$, i.e., $\hat{\mathbf{q}} = \widehat{R}^{-1}\widehat{Q}^{\mathsf{T}}\mathbf{q}$.
- The cosine similarity in the basis of $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ can be written as:

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

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 $\mathbf{w}_j$ concentrates on one term, and is close to the canonical vector $\mathbf{e}_i \in \mathbb{R}^m$ for some i.
- The peaks of $\mathbf{w}_j$, $j = 1 : 10$, correspond to: 'network', 'model', 'learning', 'function', 'unit', 'algorithm', 'input', 'data', 'neuron', 'cell', which are quite similar to the $\mathbf{u}_1$ vector or the 10 most frequently used terms.
- On the other hand, because $\mathbf{w}_j$'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 $\mathbf{w}_j$ concentrates on one term, and is close to the canonical vector $\mathbf{e}_i \in \mathbb{R}^m$ for some i.
- The peaks of $\mathbf{w}_j$, $j = 1 : 10$, correspond to: 'network', 'model', 'learning', 'function', 'unit', 'algorithm', 'input', 'data', 'neuron', 'cell', which are quite similar to the $\mathbf{u}_1$ vector or the 10 most frequently used terms.
- On the other hand, because $\mathbf{w}_j$'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 $\mathbf{w}_j$ concentrates on one term, and is close to the canonical vector $\mathbf{e}_i \in \mathbb{R}^m$ for some i.
- The peaks of $\mathbf{w}_j$, $j = 1 : 10$, correspond to: 'network', 'model', 'learning', 'function', 'unit', 'algorithm', 'input', 'data', 'neuron', 'cell', which are quite similar to the $\mathbf{u}_1$ vector or the 10 most frequently used terms.
- On the other hand, because $\mathbf{w}_j$'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 $\mathbf{w}_j$ concentrates on one term, and is close to the canonical vector $\mathbf{e}_i \in \mathbb{R}^m$ for some i.
- The peaks of $\mathbf{w}_j$, $j = 1 : 10$, correspond to: 'network', 'model', 'learning', 'function', 'unit', 'algorithm', 'input', 'data', 'neuron', 'cell', which are quite similar to the $\mathbf{u}_1$ vector or the 10 most frequently used terms.
- On the other hand, because $\mathbf{w}_j$'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 $\mathbf{w}_j$ concentrates on one term, and is close to the canonical vector $\mathbf{e}_i \in \mathbb{R}^m$ for some i.
- The peaks of $\mathbf{w}_j$, $j = 1:10$, correspond to: 'network', 'model', 'learning', 'function', 'unit', 'algorithm', 'input', 'data', 'neuron', 'cell', which are quite similar to the $\mathbf{u}_1$ vector or the 10 most frequently used terms.
- On the other hand, because $\mathbf{w}_j$'s are localized, the interpretation of the row vectors of H matrix becomes easy.

An Example Trial with the NIPS Data . . .

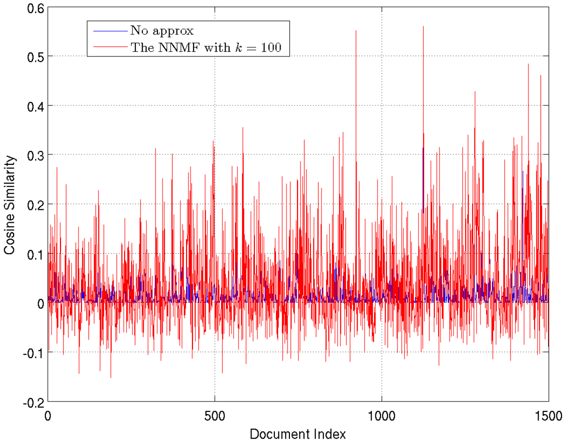


Figure: With the NNMF-based approach using $k = 100$, tol=0.2, 0.1, 0.05 correspond to 101, 312, 535 returned documents; Compare with the no approximation case: 4, 15, 89. Changing the tol=0.4, 0.3, 0.2 with NNMF returns 5, 26, 101 documents.

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:

$$\frac{\mathbf{q}^{\mathsf{T}}\mathbf{a}_j}{\|\mathbf{q}\|_2\|\mathbf{a}_j\|_2} \approx \frac{\mathbf{q}^{\mathsf{T}}W\mathbf{h}_j}{\|\mathbf{q}\|_2\|W\mathbf{h}_j\|_2}.$$

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:

$$\frac{\mathbf{q}^\mathsf{T}\mathbf{a}_j}{\|\mathbf{q}\|_2\|\mathbf{a}_j\|_2} \approx \frac{\mathbf{q}^\mathsf{T} W\mathbf{h}_j}{\|\mathbf{q}\|_2\|W\mathbf{h}_j\|_2}.$$

My Reaction . . .

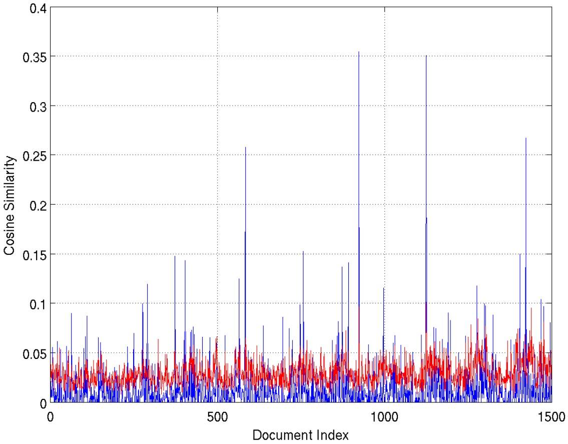


Figure: With the NNMF-based approach using $k = 100$ using the above cosine similarity approximation, tol=0.2, 0.1, 0.05 correspond to 0, 1, 97 returned documents; Compare with the no approximation case: 4, 15, 89. Without using the LS query, some of the relevant documents do not stick out clearly.