Big Data Computing

Homework 2

Pietro Spadaccino

Assignment 1

The code is provided into the file `common_friends.py`. It is a small variation of the one we did on the lab: instead of producing tuples `<(node1, node2), 1>` to be aggregated via a sum by key, now the code produces tuples `<(node1, node2), node3>` and aggregates them via a group by key, obtaining as result tuples having with key two nodes and as values the list of their common friends. After calling `saveAsTextFile` method on this last RDD, the function `merge_results` takes all `part-*` files created by Spark and merges into a single one.

I also wrote a small utility function `extract_sample` that extracts a sub-sample from the provided one, in case my computer hadn't enough power to use it all, but at the end I didn't make use of it.

Assignment 2

The code is provided into the file `document_clustering.py` and was run both on a subset of 6 topics and on all 20 ones. To retrieve the top words associated with a topic, we used the top components of:

- k-means centers on original data
- PCA decomposition
- SVD decomposition
- k-means centers on projected space via PCA
- k-means centers on projected space via SVD

Both decomposition methods yielded similar results, with PCA requiring more time ($103.2s$ vs. $0.9s$) and a large amount of RAM than SVD, since the data has to be transformed into a dense matrix to be normalized. Overall, using the decompositions and k-means returned better results than clustering only, and in less time also ($283.3s$ for k-means on the original data).

The results are reported inside the file `document_clustering_result.txt` for both 6 and 20 categories, since they are too big to be reported inside two pages of Latex.

Assignment 3

1. We show that B is the inverse of A by showing $BA = I$:

$$BA = \left(\sum_i^n \frac{1}{\sigma_i}\mathbf{v}_i\mathbf{u}_i^T\right)\left(\sum_i^n \sigma_i\mathbf{u}_i\mathbf{v}_i^T\right) = \sum_i^n\sum_j^n \frac{1}{\sigma_i}\mathbf{v}_i\mathbf{u}_i^T\sigma_j\mathbf{u}_j\mathbf{v}_j^T =$$

$$= \sum_i^n \frac{1}{\sigma_i}\mathbf{v}_i\mathbf{u}_i^T\sigma_i\mathbf{u}_i\mathbf{v}_i^T + \sum_i^n\sum_{j\neq i}^n \frac{1}{\sigma_i}\mathbf{v}_i\mathbf{u}_i^T\sigma_j\mathbf{u}_j\mathbf{v}_j^T =$$

$$= \sum_i^n \mathbf{v}_i\mathbf{u}_i^T\mathbf{u}_i\mathbf{v}_i^T = \sum_i^n \mathbf{v}_i\mathbf{v}_i^T = VV^T = I$$

Where we used the orthonormal property with $\mathbf{u}_i^T\mathbf{u}_{j\neq i} = 0$ and $\mathbf{v}_i^T\mathbf{v}_i = 1$, and $\sum_i^n \mathbf{v}_i\mathbf{v}_i^T$ is the same thing as writing VV^T.

2. Following the same reasoning as before we can show:

$$BA = \left(\sum_i^r \frac{1}{\sigma_i}\mathbf{v}_i\mathbf{u}_i^T\right)\left(\sum_i^r \sigma_i\mathbf{u}_i\mathbf{v}_i^T\right) = \sum_i^r \mathbf{v}_i\mathbf{v}_i^T$$

Since A may be not invertible, the rank may be not maximum $r < n$, and so the last sum is not always equal to $VV^T = I$ because we are no longer summing over all $\mathbf{v}_{1,\dots,n}$. Let us left and right multiply by $\mathbf{x}$:

$$BA\mathbf{x} = \left(\sum_i^r \mathbf{v}_i\mathbf{v}_i^T\right)\left(\sum_i^r \alpha_i\mathbf{v}_i\right) = \sum_i^r \mathbf{v}_i\mathbf{v}_i^T\alpha_i\mathbf{v}_i + \sum_i^r\sum_{j\neq i}^r \mathbf{v}_i\mathbf{v}_i^T\alpha_j\mathbf{v}_j =$$

$$= \sum_i^r \alpha_i\mathbf{v}_i\mathbf{v}_i^T\mathbf{v}_i = \sum_i^r \alpha_i\mathbf{v}_i = \mathbf{x}$$

Since $\mathbf{v}_i^T\mathbf{v}_i = 1$ and $\mathbf{v}_i^T\mathbf{v}_{j\neq i} = 0$.