Heidelberg University
Winter semester 2022/2023

Group Parallel and Distributed Systems (PVS)
Artur Andrzejak

Problem Set 2 for lecture Mining Massive Datasets

Due November 7, 2022, 23:59 CET

Exercise 1 (2 points)

Three computers, A, B, and C, have the numerical features listed below:

Feature	A	B	C
Processor Speed	3.06	2.68	2.92
Disk Size	500	320	640
Main-Memory Size	6	4	6

We may imagine these values as defining a vector for each computer; for instance, A's vector is $[3.06, 500, 6]$. We can compute the cosine distance between any two of the vectors, but if we do not scale the components, then the disk size will dominate and make differences in the other components essentially invisible. Let us use 1 as the scale factor for processor speed, α for the disk size, and β for the main memory size.

(a) What are the angles between the vectors if $\alpha = \beta = 1$?

(b) What are the angles between the vectors if $\alpha = 0.01$ and $\beta = 0.5$?

(c) One fair way of selecting scale factors is to make each inversely proportional to the average value in its component. What would be the values of α and β, and what would be the angles between the vectors?

Exercise 2 (2 points)

Consider a web shop that sells furniture and uses a recommendation system. When a new user creates an account and *likes* one product, he will be presented with similar products on his next visit.

How can a competitor - in principle - try to steal the valuable data for recommendation from this website? Does this work better when the web shop implemented a content-based or a collaborative filtering system? What data would the competitor be able to infer? Would this technique have an impact on the recommendation system, i.e., would this attack create a bias on the data? Why is this attack probably not viable in any case?

Exercise 3 (2 points)

The following table shows a utility matrix, that represents the ratings, on a 1–5 star scale, of eight items, a through h, by three users A, B, and C.

	a	b	c	d	e	f	g	h
A	4	5		5	1		3	2
B		3	4	3	1	2	1	
C	2		1	3		4	5	3

Compute the following from the data of this matrix.

(a) Treating the utility matrix as Boolean, compute the Jaccard distance between each pair of users (see additional slides of lecture 3 for the definition of the Jaccard measures).

(b) Repeat Part (a), but use the cosine distance.

(c) Treat ratings of $3, 4$, and 5 as 1 and $1, 2$, and blank as 0. Compute the Jaccard distance between each pair of users.

(d) Repeat Part (c), but use the cosine distance.

(e) Normalize the matrix by subtracting from each nonblank entry the average value for its user.

(f) Using the normalized matrix from Part (e), compute the cosine distance between each pair of users.

Exercise 4 (2 points)

What data can you use to populate a utility matrix in the following cases? Think about what are users and what constitutes the items. What relations other than *like* or a numerical rating can be exploited to express a positive or negative connection?

a) A website where students can rate their professors. Users are students that can share their experiences of lectures, seminars and exams. If a rating is given for a so far unknown professor, there will be created a fresh profile for this professor.

b) An online community for sharing artwork. Each user can upload pictures of his or her artwork, and also rate those of others.

c) A dating platform. Every user has an online profile and can visit and *like* those of others. Users can describe a hidden profile of their "dream partner". Users can send messages to users and block users. What is special about this scenario?

Exercise 5 (8 points)

Download historical data of the audio platform Audioscrobbler[1] that has been merged with Last.fm in 2005. The file `user_artist_data_small.txt` is a file containing the tab-separated relation "user X has listened to artist Y for Z many times", represented as "<userid> <artistid> <playcount>". Write a Spark program that does the following (submit your code as a part of the solution).

a) Populate a utility matrix. Be sure to first replace bad artist ids that are due to known misspellings using the assignments in `artist_alias_small.txt`. Think about how to store the matrix in a reasonable way.

b) Implement a routine that calculates the similarity between users using Pearson correlation coefficient as similarity metric.

c) Write a method that returns the top k most similar users to a given user based on the routine from b) (i.e., the k-neighborhood of a user).

d) Create a new artificial user U with a new unique ID of your choice. Fix a set S of your five favorite artists in `artist_data_small.txt`. Add records to the dataset which simulates that U has listened 20 times to artists from S (and only those; the split of the 20 acts of listening on the five artists is your choice). Then use the

[1] Available on heibox, see the same folder as this pdf, file Datasets/dataset-problemset2-ex5.zip

above routines (and possibly more code) to recommend new artists to user U *and* to user 1029563. Submit as part of your solution your set S, the additional records resulting from this, and the recommendation result for user 1029563.