Assignment 2 Instructions: Content-Based Recommenders

Overview

In this assignment, you will hand-create and use some content-based profiles. You'll go through a set of variations to see how certain features of the computation can introduce (or reduce) biases.

Instructions

The Data Set

First, download the Assignment 2 dataset (linked here as a spreadsheet). It contains a table of content attributes for 20 documents across 10 attributes. It also lists two users' evaluations of five documents each. For purposes of this assignment, we're treating content attributes as boolean (either an article is about a topic or it isn't) and we're treating evaluations as positive (liked it), negative (disliked it), or unknown (never saw it).

Part 1. Build and use a very basic profile

First, you will build a very simple profile of user preferences for attributes. In this profile, you'll count the total the number of positive and negative evaluations associated with each attribute, and create a profile with the total score for each attribute for each user. For example, user 1's score for "Family" will get a +1 from doc1 (positive evaluation) and a -1 from doc 19 (negative evaluation) for a total profile value of 0 (neutral). In contrast, user 2's score for Europe will be +3 (+1 each for doc2, doc4, and doc17).

You can compute the profiles and place them in the "User Profiles" section of the spreadsheet.

Now compute the predicted score for each user for each document (a simple dot-product). Type in the answers to the following questions (answers may include already-rated articles) when you go to Assignment 2:

- Which document does the simple profile predict user 1 will like best?
- What score does that prediction get?
- How many documents does the model predict U2 will dislike (prediction score that is negative)?

Notice that this model is consistent with the users' ratings -- it predicts liking for all the positive documents and disliking for all the negative ones.

Part 2. Next, let's treat all articles as having unit weight ...

You may have noticed that in our computation an article that had many attributes checked could have more influence on the overall profile than one that had only a few. doc 1 and doc 19 each have five attributes, while doc6, doc7, and doc18 only have 2 attributes each.

We want to explore whether our simple model may be counting these attribute-heavy documents too much. For example, we might conclude that liking doc6 says more about liking baseball (since it is one of only two attributes for the article along with Europe) than liking doc1 says (since doc1 is also about politics, Asia, soccer, and family).

To try this out, make a copy of the attributes matrix on another sheet. Then we're going to have you normalize each row to be a unit length vector. We can do this in two steps:

1. Count the total number of items in the row (you can do this we a SUM or COUNT function).
2. Divide each value by the square root of that number of items. If you do this right, doc1's values will all change from 1 to 0.447214 (approx). Documents with 4 attributes will change to 0.5 (since 4 * .5^2 = 1), and so forth. Remember, don't have the SUM or COUNT depend on the copy of the cells you're changing or you'll get a circular dependency. Have your new sheet depend on values on your old sheet.

Once you have the new values, compute your second set of user profiles and new predictions. If you did this right, you'll see a prediction of 1.0090 (approx) for user1/doc1. Don't worry about the scale of the numbers (they'll all be smaller, in absolute value terms), but look at the order of them.

This time we'll start with user2. With our simple profile, doc7 and doc19 both had similar "like" predictions (+2 each). Now they don't. Enter the predictions for doc7 and doc19 for user2 using these new models:

doc7: num input: 0.7444 (plus or minus 0.01)

doc 19: num input: 0.4834 (plus or minus 0.01)

The difference here can be seen by looking at the profile attribute values. Doc7 is 50% about one of user2's favorite topics (security) which is now more heavily emphasized).

Now let's look at user 1. In our simple model, the second/third place recommendation was a tie between doc1 and doc 12. Neither of those is in second place with this new model.

Type in the answers to the following questions when you go to Assignment 2:

- Which document is now in second with this new model?
- What prediction score does it have?

Part 3. Finally, let's consider how common different terms are among our documents …

We're going to do one more model -- one that accounts for the fact the the content attributes have vastly different frequencies.

We're going to include an IDF (inverse document frequency) term into our equation. Start with your spreadsheet from part 2. Add a row that shows 1/DF where DF is the number of documents in which each content attribute occurs. For example, baseball occurs in 4 documents, so baseball's entry will be 0.25. Politics occurs in 10 documents, so it will get an IDF score of 0.1 (1 / 10).

Note that this is far more dramatic a computation than is usually used with large datasets (more common is 1 / log(DF)), but we need a dramatic value to see differences with a small dataset.

Next, update your prediction formula to do a three-way dot product: document vector * profile * IDF (fortunately, SUMPRODUCT can handle a third array). If you did this right, you'll see a prediction of about 0.2476 for user1/doc1.

Ok, now let's look at the results.

Type in the answers to the following questions when you go to Assignment 2:

- Compare doc1 and doc9 for user1. What's user1's prediction for doc9 in the new IDF weighted model? See how there's a dramatic difference from the prior model?
- Now let's look at user 2. Look at doc6. It was moderately positive before and now is slightly negative. Why did that change?