Evaluating Information Retrieval

STA 325, Supplemental Material

Information Retrieval

One of the fundamental problems with having a lot of data is finding what you're looking for.

This is called **information retrieval**!

Recap of the last lecture

- The simplest way we can represent a document is using a bag of words (BoW) representation
- A BoW is a vector that gives the number of times each word occurred in the document.
- A BoW abstracts away the grammatical structure and context, leaving us with a matrix whose rows are feature vectors (a data frame).
- In order to find document that are similar to a given document (query) Y, we calculate the distance between Y and all the other documents (X).
- There are many types of distance we can use such as the Euclidean, Manhattan, cosine, Jaccard.

Evaluating similarity searches

We someone uses a search engine, they have some idea of what they are looking for.

For example, if I search for "Chinese linking rings" using Google, I typically know keywords that I'm looking for which include:

- "rings, magic, illusion, metal, link, unlink, ancient"

They would not include keywords:

- "engagement, wedding bands, marriage"

Evaluating similarity searches

There are actually two aspects to finding relevant documents, both of which are important:

1. Most of the results should be relevant, meaning the precision of the search should be high.
2. Most of the relevant items should be returned as results, meaning the recall should be high.

What are the precision and recall?

Precision and recall

- Suppose we have a corpus of documents containing N items.
- Suppose there are R relevant items in the entire corpus of N items.
- Suppose a query returns k items, of which r are releveant.

Then the precision is the ratio

$$\frac{r}{k}$$

and the recall is

$$\frac{r}{R}.$$

Note: this is for one query but we can average over queries if there is more than one.

Note: $r \leq k$ so there are limits on how high the recall can be when k is small.

Precision and recall

- As we vary k for a given query, we get different values for the precision and recall.

We expect that increasing k will

- increase the recall (more relevant things come in)
- but lower the precision (more irrelevant things can come in too)!

A search method is thought to be good where the trade-off between the precision and recall is not very sharp.

This means we gain a lot of recall without losing too much precision.

Visualizing the precision and recall

- We can visualize the precision and recall by plotting the precision (vertical axis) against the recall (horizontal axis) for multiple values of k.

If the method is working well then:

1. when k is small, the precision should be high but the recall will be limited by k.
2. As k grows, the recall should increase, moving us to the right. The precision will fall, moving us down.
3. So the precision-recall plot should go someone near $(1, 0)$ to somewhere near $(0, 1)$.

This total area under the curve is often used a measure of how good a search method is.

Search, Hypothesis Testing, Signal Detection, ROC

There are many connections to the precision-recall plot to other frequently used methods in statistics and computer science.

The difference between precision and recall is very like the difference between type I and type II errors in hypothesis testing.

High precision is like having a low type I error rate (most of the "hits" are real)

High recall is like having a low type II error rate (most things which should be "hits" really are hits).

Search, Hypothesis Testing, Signal Detection, ROC

The same idea applies to signal detection.

- A type I error is called a "false alarm" (you thought there was signal when there was just noise)

A type II error is called a "miss" (you mistook signal for noise).

The precision-recall curves actually come from signal detection theory, where they are called receiver operating characteristic curves, or ROC curves.

Visualization

One common question that often arises is how might we visualize a BoW representation?

Recall that the BoW vectors representing our documents typically have more than three dimensions, and this is hard to visualize.

But, we can compute the distance between any two vectors, so we know how far apart these vectors are.

Visualization: Multidimensional Scaling

Multidimensional scaling (MDS) is the general name for a family of algorithms which take high-dimensional vectors and map them down to two- or three-dimensional vectors, trying to preserve all the relevant distances.

Visualization: Multidimensional Scaling

Abstractly, we start with vectors $v_1, \ldots, v_n$ that live in a p-dimensional space, where p is large.

We want to find new vectors $x_1, \ldots, x_n$ such that

$$\sum_{i=1}^{n} \sum_{j \neq i} (\delta(v_1, v_2) - d(x_1, x_2)),$$

where δ is the distance in the original space and d is the Euclidean distance in the new space.

Note that the new or image points x_i are representations of the v_i, i.e., representations of representations.

Visualization: Multidimensional Scaling

These plots can be very useful because they are nice visualization tools.

They are often used to visualize data reduction methods because sometimes we want to preserve properties besides distances.

Visualization: Multidimensional Scaling

1. BoW representation gives each of our original coordinates/features some meaning because it says something very definite about the document being represented.
2. This is not the case with the coordinates we get after doing the MDS. Why?

- We could rotate all of the image points arbitrarily.
- This would make it very hard to assign any interpretation to where the images fall on the axes.
- While this is not a proof, hopefully it gives you some intuition.
- This is true of many other dimensionality-reduction methods as well.

Multidimensional scaling in R

To make a multi-dimensional scaling plot in R, you will want to use the `cmdscale` function().

Exercise: use the help function to figure out how to use this function on your own.