# Information Retrieval

Peadar Coyle

August 8, 2012

## 1 Textual features

I'd like to introduce the concepts of how **features** are extracted, and we shall consider these proxies of actual meanings. One classic representation we can use is **bag-of-words** (BoW). Let us define this representation, this means we list all of the distinct words in the document together with how often each one appears. This is easy to calculate from the text. **Vectors** One way we could try to code up the bag-of-words is to use vectors. Let each component of the vector correspond to a different word in the total **lexicon** of our document collection, in a fixed, standardised order. The value of the component would be the number of times the word appears, possibly including zero. We use this vector bag-of-words representation of documents for two big reasons:

- There is a huge pre-existing technology for vectors: people have worked out, in excruciating detail, how to compare them, compose them, simplify them, etc. Why not exploit that, rather than coming up with stuff from scratch?
- In practice, it's proved to work pretty well.

We can store data from a corpus in the form of a matrix. Each row corresponds to a distinct **case** (or instance **instance**, **unit, subject**,...) - here, a document - and each column to a distinct feature. Conventionally, the number of cases is n and the number of features is p. It is no coincidence that this is the same format as the data matrix $\mathbf{X}$ in linear regression.

## 2 Measuring Similarity

Right now, we are interested in saying which documents are similar to each other because we want to do a search by content. But measuring **similarity** - or equivalently measuring **dissimilarity** or **distance** - is fundamental to data mining. Most of what we will do will rely on having a sensible way of saying how similar to each other different objects are, or how close they are in some geometric setting. Getting the right measure of closeness will have a huge impact on our results.

This is where representing the data as vectors comes in so handy. We already know a nice way of saying how far apart two vectors are, the ordinary or

**Euclidean distance**, which we can calculate with the Pythagorean formula:

$$\|\bar{x} - \bar{y}\| = \sqrt{\sum_{i=1}^{p}(x_i - y_i)^2}$$

where $x_i$, $y_i$ are the $i^{th}$ components of $\bar{x}$ and $\bar{y}$. Remember that for bag-of-words vectors each distinct word - each entry in the lexicon - is a component or a fecture. We can also use our Linear Algebra skills to calculate the **Euclidean norm** or **Euclidean distance** . Of any vector this is $\|\bar{x}\| = \sqrt{\sum_{i=1}^{p} x_i^2}$ so the distance between two vectors is the norm of their distance $\bar{x} - \bar{y}$. Equivalently, the norm of a vector is the distance from it to the origin, $\bar{0}$

Obviously, one can just look up a topology textbook and remind oneself of other metrics such as the **taxicab** metric.

## 2.1 Normalisation

Just looking at the Euclidean distances between document vectors doesn't work, at least if the documents are at all different in size. Instead, we need to **normalise** by document size, so that we can fairly compare short texts with long ones. There are (at least) two ways of doing this.
**Document length normalisation** Divide the word counts by the total number of words in the document. In symbols,

$$\bar{x} \Rightarrow \frac{\bar{x}}{\sum_{i=1}^{p} x_i} \tag{1}$$

Notice that all the entries in the normalised vector are non-negative fractions, which sum to 1. The i-th component is thus the probability that if we pick a word out of the bag at random, it's the i-th entry in the lexicon.
**Cosine 'distance'** is actually a similarity measure, not a distance‘:

$$d_{cos}\bar{x}, \bar{y} = \frac{\sum_i x_i y_i}{\|\bar{x}\|\|\bar{y}\|} \tag{2}$$

It's the cosine of the angle between the vectors $\bar{x}$ and $\bar{y}$.