# Unsupervised Learning

Previously, our methods have focused on predicting a particular variable with a number of other numerical features (word counts, usually). This is what is called **supervised learning** because the learning is supervised by the variable we are trying to predict.

Now, we move to a new set of techniques for **unsupervised learning**. These can be applied to a set of numeric features without reference to a supervising variable.

# Distances

The methods we will study in the third section of the course are all focused on the distances between documents. If we have two documents described by a series of word counts, we can describe their distance using the usual Euclidean distance: the sum of squared differences of the features.

Notice that these distances have nothing directly to do with a predictive modelling task.

# Two Unsupervised Learning Tasks

There are two unsupervised learning tasks that we will preform using distances between documents.

Today, we will see the task of **dimensionality reduction**. This is the task of approximating the distances between documents in a lower dimensional space.

Next class, we will see **clustering analysis**. This is the task of grouping documents that are close to one another together.

# Dimensionality Reduction

In dimensionality reduction, we will try to map each document into a smaller dimensional space. For example, into two dimensions we would try to map each document as:

**doc00001 => x1, y1**
**doc00002 => x2, y2**
**doc00003 => x3, y3**
**doc00004 => x4, y4**

And so forth, using the raw word counts to produce the new x's and y's. The idea is that the distances between documents in the x-y space should approximate the distances in the larger 10k dimensional space of word counts.

# Dimensionality Reduction

**Applications**

- Can plot the data in a low-dimensional space to visualise the data.
- Can use as a pre-processing step before a modeling algorithm.

**Cautions**

- Dimensionality reduction is only an appoximation.
- The new features cannot (usually) be easily interpretted.

# Method I: PCA

PCA, or principal component analysis, is the analogous dimensionality reduction approach to linear regression. It finds the best approximation of Euclidean distances in a lower dimensional subspace, restricted to linear combinations of the input features.

**Defining Features**

- Can be computed very quickly, even for a relatively large number of components.
- The k-dimensional PCA is a strict subset of the (k+1)-dimensional PCA.
- Can easily apply the same transformation to newly observered data.
- Closely related to the singular values of the data matrix.

# Method II: UMAP

The other method we will use is called UMAP. It works by preserving the closest neighbors of each observation, without directly trying to preserve distances overall.

**Defining Features**

- Mostly used for visualisation; most applicable for 2 or 3 dimensions.
- Excellent for transforming data that is in a large, sparse space.
- Transformations are non-linear; produces some better relationships at the cost of often having some outliers.
- Cannot (easily) be applied to new data.