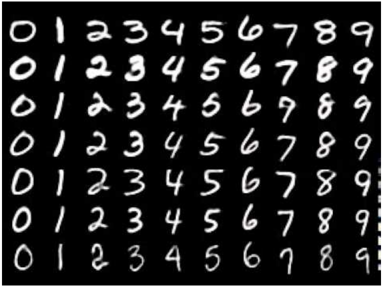
Dimensionality reduction

What does 784-D space look like?

Example: MNIST http://yann.lecun.com/exdb/mnist/

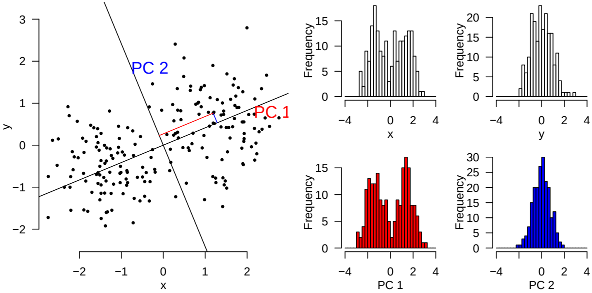


https://github.com/cazala/mnist

- The “hello world” of deep networks.
- MNIST ZIP Code handwritten images.
- Input 28x28 pixel images [0,255].
- Output: label as integer classes [0-9]
- Training set: 60,000 samples
- Test set: 10,000 samples

Dimensionality reduction / PCA

What: Maximise variance of encoding.
How: Eigenvectors of covariance matrix of inputs. Fractions of variance given by eigenvalues.



Multidimensional scaling (MDS)

- For each point X_i in some high-dim space, we have an equivalent point $Y_i = (y_{i1}, \ldots y_{id})$ in some low (d= 2 or 3) dim space.

$$o_{ij} = dist(X_i, X_j) \qquad \text{(fixed)}$$

$$d_{ij} = dist(Y_i, Y_j) = (\sum_{k=1}^{d} (y_{ik} - y_{jk})^2)^{0.5}$$

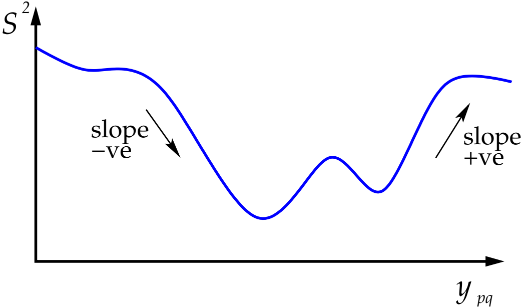
- Project down into lower dimension so that $o_{ij} \approx d_{ij}$.
- o_{ij} are fixed, so we vary points y_{ij} to minimise stress term:

$$S^2 = \frac{\sum_i \sum_j (d_{ij} - o_{ij})^2}{\sum_i \sum_j o_{ij}^2}$$

MDS: Minimisation of Stress term

- Starting from some guess for initial points o_{ij}, calculate S^2 and then evaluate gradient for each parameter y_{pq}.

$$\Delta y_{pq} = -\alpha \frac{\partial S^2}{\partial y_{pq}}$$



- Iteratively update y_{pq} until local minimum (gradient is zero).

Relationship between MDS and PCA

- MDS and ‘PCA are equivalent when using Euclidean distance measure with stress measure: $S^2 = \sum_{i,j}(d_{ij} - o_{ij})^2$.
- Other stress measures can be used (e.g. Sammon, next slide) to emphasise certain aspects of data.
- Non-metric versions (NMDS) do not consider absolute distances, but preserve only ranking of distances.

Sammon mapping

- The Sammon mapping uses the error measure:

$$E = \sum_{i<j} \frac{(o_{ij} - d_{ij})^2}{o_{ij}}$$

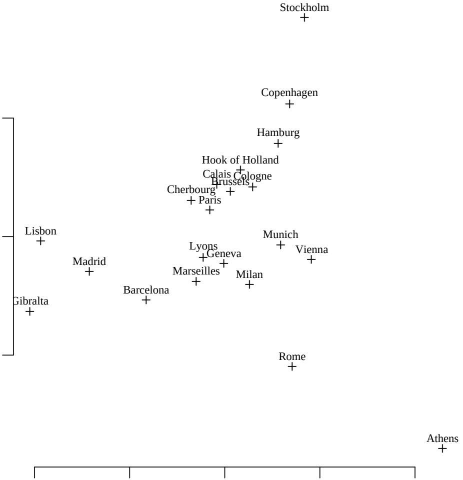
See separate video on how we apply gradient descent so that we take the derivative of E with respect to each element of each y. As we move each y, the d_{ij} are recalculated such that E decreases.

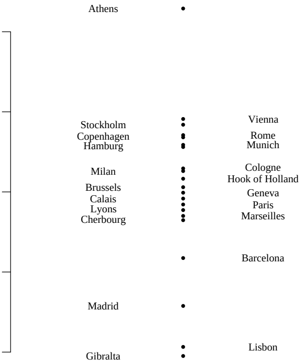
eurodist: the distances between European cities

The eurodist dataset (a lower triangular matrix) tells you the road distance in km between 21 cities:

```
  > eurodist
                Athens Barcelona Brussels Calais Cherbourg Colo
Barcelona         3313
Brussels          2963      1318
Calais            3175      1326      204
Cherbourg         3339      1294      583    460
Cologne           2762      1498      206    409       785
Copenhagen        3276      2218      966   1136      1545     7
```

MDS example: European distances



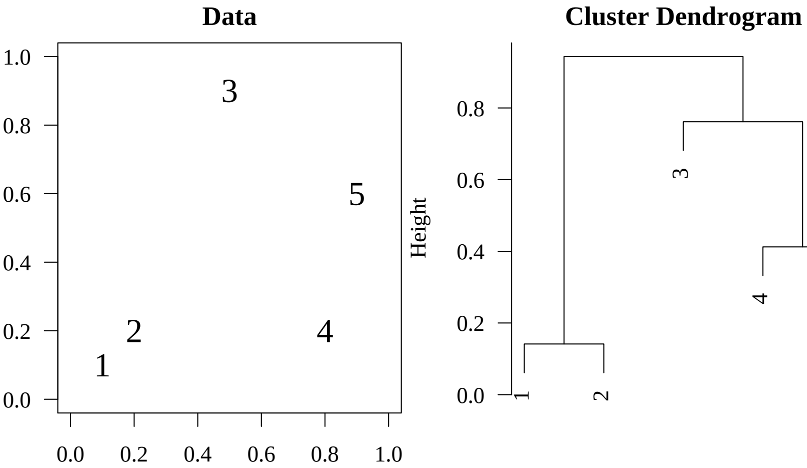


(R notes: eurodist data from MASS package, with cmdscale() and sammon())

t-SNE: t-distributed stochastic neighbour embedding

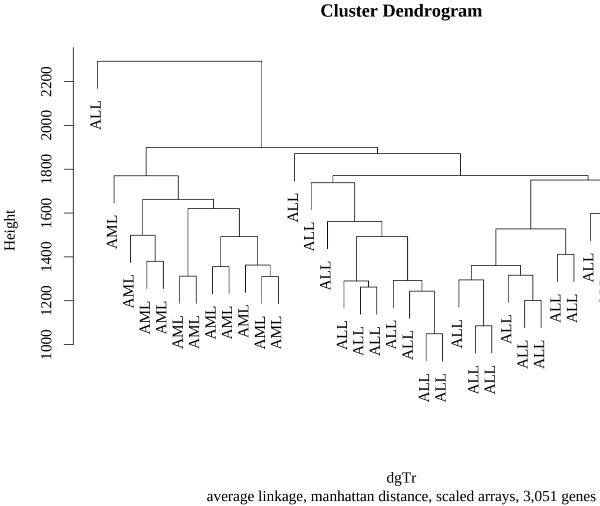
1. Another gradient descent approach (van der Maarten and Hinton, 2008).
2. Probabilistic approach, with distances scaled by local density of data (locality controlled by parameter “perplexity”). Measure $p(i,j)$ as probability of being neighbours in high-D space, and fix. $q(i,j)$ is adjusted in low-D space by gradient descent on KL divergence.
3. Computationally expensive (UMAP has mathematical and computational advantages). Used in genomics a lot, but beware of limitations about intepretation. essential reading: Wattenberg et al (2016) *How to use t-SNE effectively* `https://distill.pub/2016/misread-tsne/`). `https://pair-code.github.io/understanding-umap/`

Hierarchical clustering

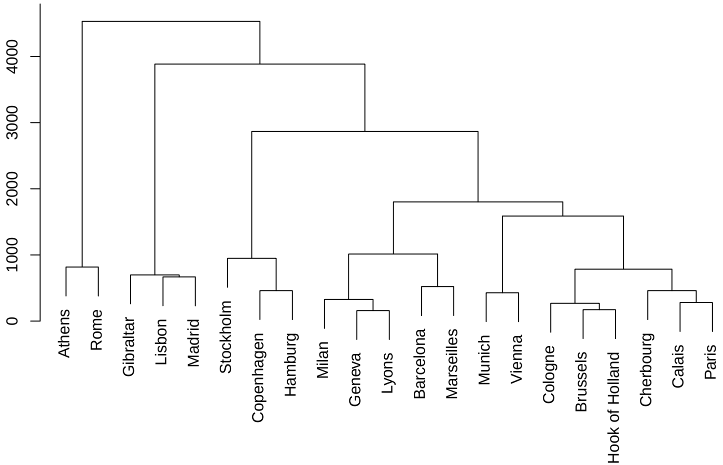


AML / ALL example (Golub et al. 1999)

Here we have information (type of cancer) associated with each input vector (gene-expression data).



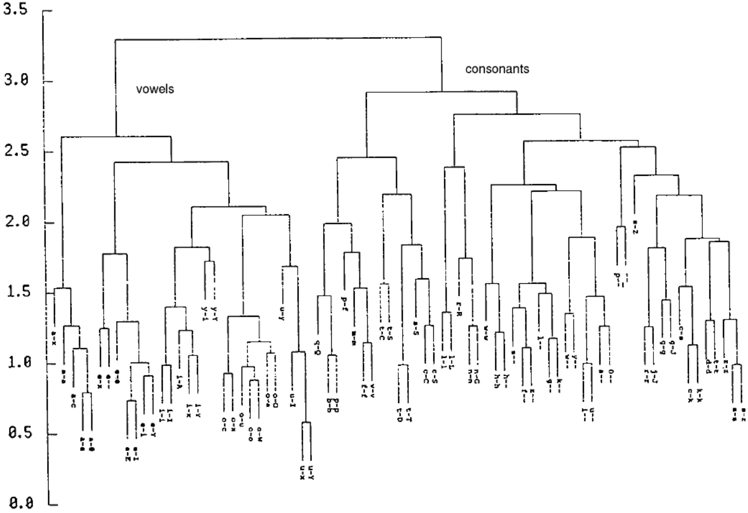
European cities



```
plot(hclust(eurodist), main='')
```

NetTalk: Hidden unit analysis

What features are the network extracting? Compute 80-d vector of average activity for given input–output pair (of which there are 79).



So, what does 784D space look like?

Excellent online resource for visualisation:
`http://colah.github.io/posts/2014-10-Visualizing-MNIST/`