Similarity Methods and Clustering

Kenneth Benoit & Pablo Barberá

MY 459: Quantitative Text Analysis

March 5th, 2018

Course website: lse-my459.github.io

Outline

- Documents as feature vectors
- Similarity foundations
- Similarity Measures
  - cosine similarity
  - Euclidean distance
  - Jacquard
- Clustering methods
  - k-means clustering
  - hierarchical clustering
- Preview of topic models

Documents as vectors

- The idea is that (weighted) features form a vector for each document, and that these vectors can be judged using metrics of similarity
- A document's vector for us is simply (for us) the row of the document-feature matrix

Characteristics of similarity measures

Let A and B be any two documents in a set and $d(A, B)$ be the distance between A and B.

1. $d(x, y) \geq 0$ (the distance between any two points must be non-negative)
2. $d(A, B) = 0$ iff $A = B$ (the distance between two documents must be zero if and only if the two objects are identical)
3. $d(A, B) = d(B, A)$ (distance must be symmetric: A to B is the same distance as from B to A)
4. $d(A, C) \leq d(A, B) + d(B, C)$ (the measure must satisfy the triangle inequality)

Euclidean distance

Between document A and B where j indexes their features, where y_{ij} is the value for feature j of document i

- Euclidean distance is based on the Pythagorean theorem
- Formula

$$\sqrt{\sum_{j=1}^{j}(y_{Aj} - y_{Bj})^2} \tag{1}$$

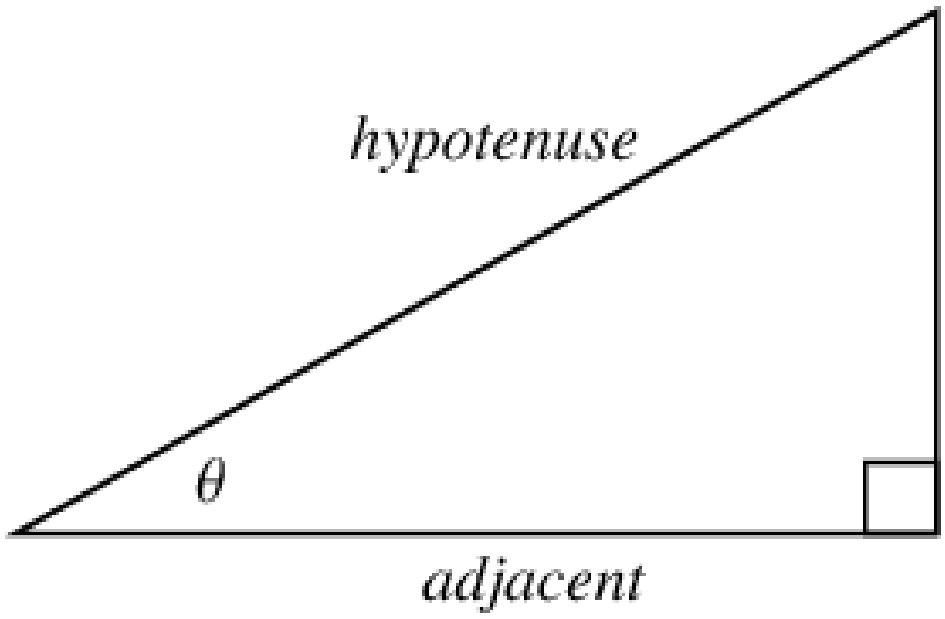
- In vector notation:

$$\|\mathbf{y}_A - \mathbf{y}_B\| \tag{2}$$

- Can be performed for any number of features J (or V as the vocabulary size is sometimes called – the number of columns in of the dfm, same as the number of feature types in the corpus)

A geometric interpretation of "distance"

In a right angled triangle, the cosine of an angle θ or $\cos(\theta)$ is the length of the adjacent side divided by the length of the hypotenuse



We can use the vectors to represent the text location in a V-dimensional vector space and compute the angles between them

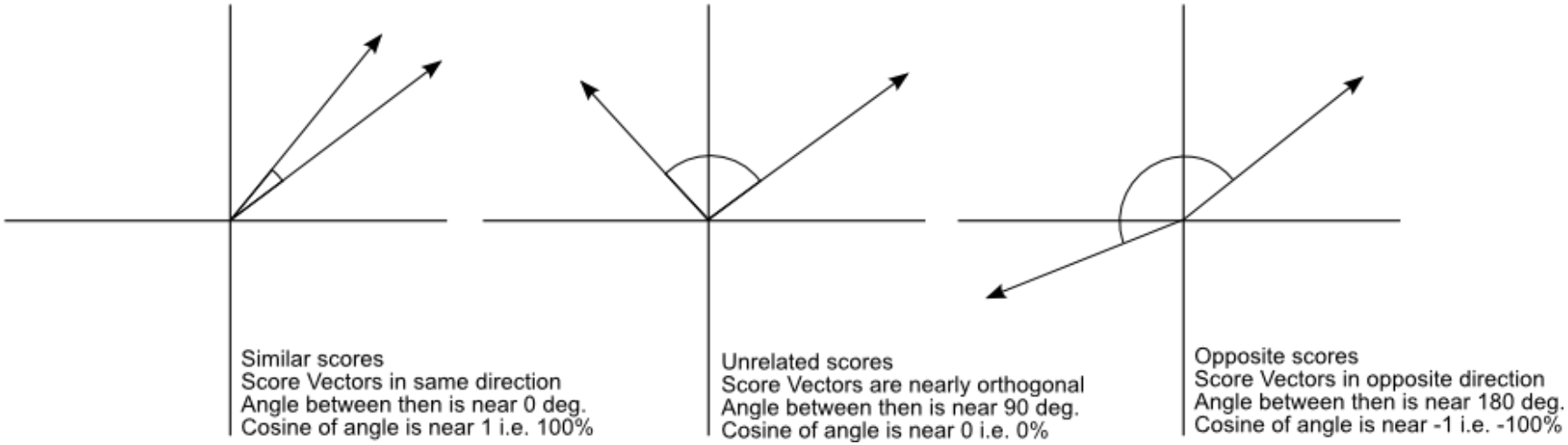
Cosine similarity

- Cosine distance is based on the size of the angle between the vectors
- Formula

$$\frac{\mathbf{y}_A \cdot \mathbf{y}_B}{\|\mathbf{y}_A\| \|\mathbf{y}_B\|} \tag{3}$$

- The $\cdot$ operator is the dot product, or $\sum_j y_{Aj} y_{Bj}$
- The $\|\mathbf{y}_A\|$ is the vector norm of the (vector of) features vector $\mathbf{y}$ for document A, such that $\|\mathbf{y}_A\| = \sqrt{\sum_j y_{Aj}^2}$
- Nice property: independent of document length, because it deals only with the angle of the vectors
- Ranges from -1.0 to 1.0 for term frequencies, or 0 to 1.0 for normalized term frequencies (or tf-idf)

Cosine similarity illustrated



Example text

Hurricane Gilbert swept toward the Dominican Republic Sunday , and the Civil Defense alerted its heavily populated south coast to prepare for high **winds**, heavy **rains** and high seas.

The **storm** was approaching from the southeast with sustained **winds** of 75 mph gusting to 92 mph .

"There is no need for alarm," Civil Defense Director Eugenio Cabral said in a television alert shortly before midnight Saturday .

Cabral said residents of the province of Barahona should closely follow **Gilbert** 's movement .

An estimated 100,000 people live in the province, including 70,000 in the city of Barahona , about 125 miles west of Santo Domingo .

Tropical **Storm Gilbert** formed in the eastern Caribbean and strengthened into a **hurricane** Saturday night

The National **Hurricane** Center in Miami reported its position at 2a.m. Sunday at latitude 16.1 north , longitude 67.5 west, about 140 miles south of Ponce, Puerto Rico, and 200 miles southeast of Santo Domingo.

The National Weather Service in San Juan , Puerto Rico , said **Gilbert** was moving westward at 15 mph with a "broad area of cloudiness and heavy weather" rotating around the center of the **storm**.

The weather service issued a flash flood watch for Puerto Rico and the Virgin Islands until at least 6p.m. Sunday.

Strong **winds** associated with the **Gilbert** brought coastal flooding , strong southeast **winds** and up to 12 feet to Puerto Rico 's south coast.

Example text: selected terms

- Document 1
  Gilbert: 3, hurricane: 2, rains: 1, storm: 2, winds: 2
- Document 2
  Gilbert: 2, hurricane: 1, rains: 0, storm: 1, winds: 2

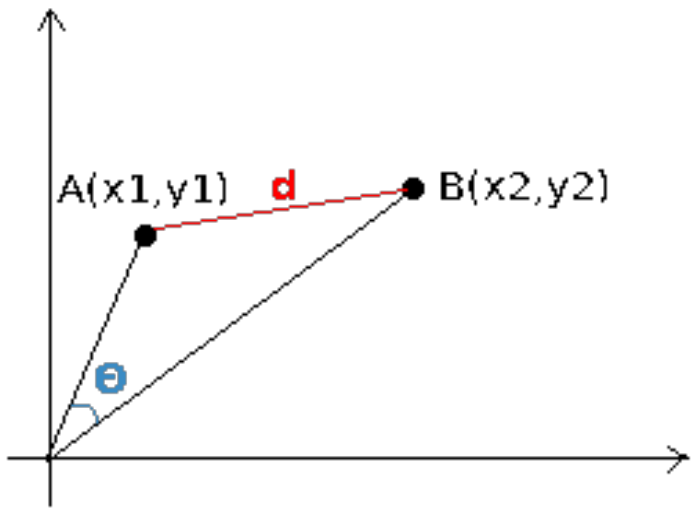
Example text: cosine similarity in R

```
toyDfm <- as.dfm(matrix(c(3,2,1,2,2, 2,1,0,1,2),
                        nrow = 2, byrow = TRUE))
colnames(toyDfm) <- c("Gilbert", "hurricane", "rain", "storm", "winds")
toyDfm
## Document-feature matrix of: 2 documents, 5 features (10% sparse).
## 2 x 5 sparse Matrix of class "dfm"
##        features
## docs    Gilbert hurricane rain storm winds
##   text1       3         2    1     2     2
##   text2       2         1    0     1     2

textstat_simil(toyDfm, method = "cosine")
##           text1
## text2 0.9438798
```

Relationship to Euclidean distance

- Cosine similarity measures the similarity of vectors with respect to the origin
- Euclidean distance measures the distance between particular points of interest along the vector



Relationship to Euclidean distance

- Euclidean distance is $\|\mathbf{y}_A - \mathbf{y}_B\|$
- $\cos(A, B) = \frac{\mathbf{y}_A \cdot \mathbf{y}_B}{\|\mathbf{y}_A\| \|\mathbf{y}_B\|}$

If A and B are normalized to unit length (term proportions instead of frequencies), such that $\|A\|^2 = \|B\|^2 = 1$, then

$$\begin{aligned} \|\mathbf{y}_A - \mathbf{y}_B\|^2 &= (A - B)'(A - B) \\ &= \|A\|^2 + \|B\|^2 - 2\ A'B \\ &= 2(1 - cos(A, B)) \end{aligned}$$

where $(1 - cos(A, B))$ is the complement of the cosine similarity, also known as *cosine distance*

so the Euclidean distance is twice the cosine distance for normalized term vectors

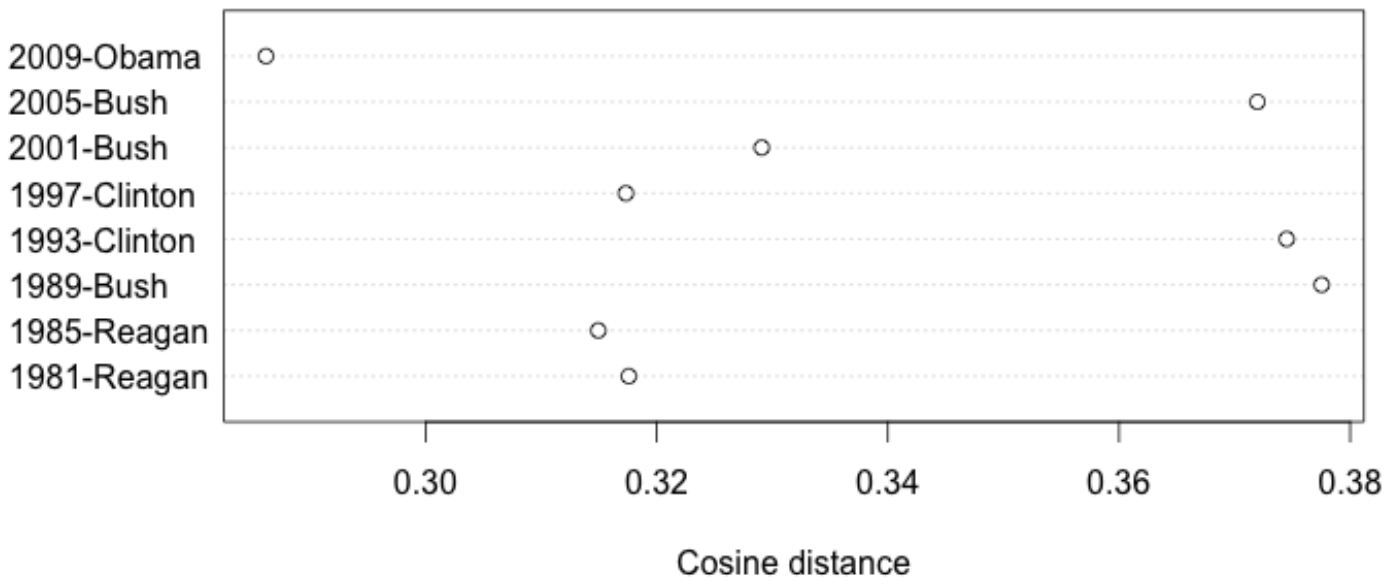
Jacquard coefficient

- Similar to the Cosine similarity
- Formula

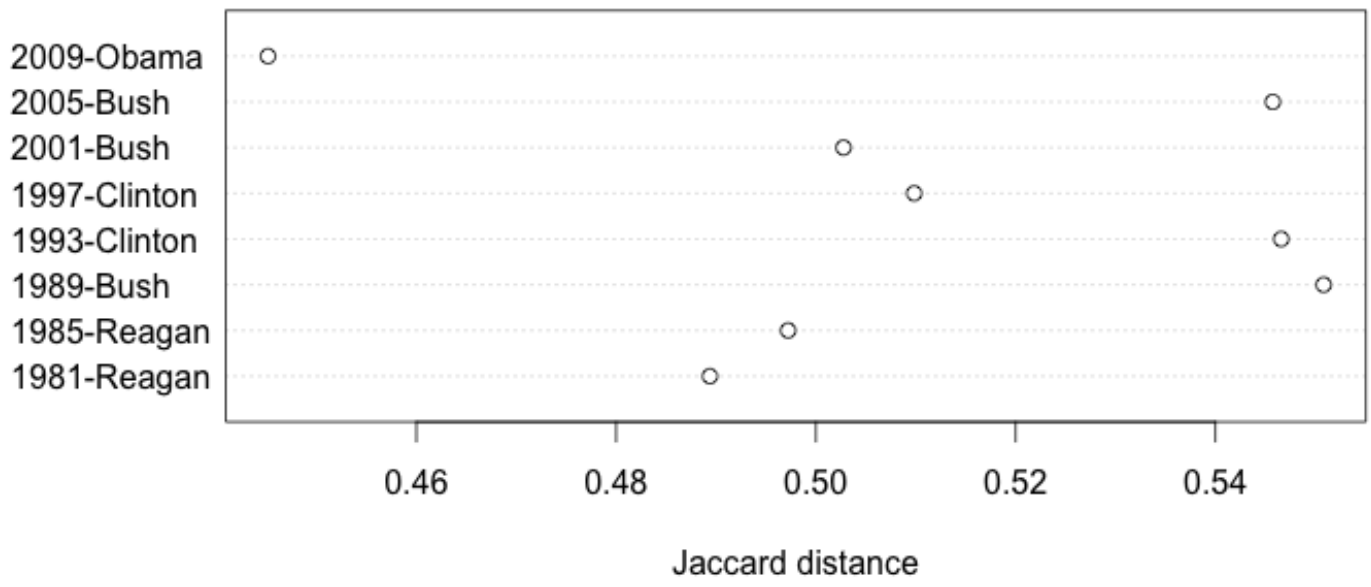
$$\frac{\mathbf{y}_A \cdot \mathbf{y}_B}{\|\mathbf{y}_A\| + \|\mathbf{y}_B\| - \mathbf{y}_A \cdot \mathbf{y}_B} \tag{4}$$

- Ranges from 0 to 1.0

Example: Inaugural speeches



Example: Inaugural speeches



Can be made very general for binary features

Example: In the Choi et al paper, they compare vectors of features for (binary) absence or presence – called ("operational taxonomic units")

Table 1 OTUs Expression of Binary Instances i and j

j \ i	1 (Presence)	0 (Absence)	Sum
1 (Presence)	$a = i \bullet j$	$b = \bar{i} \bullet j$	$a+b$
0 (Absence)	$c = i \bullet \bar{j}$	$d = \bar{i} \bullet \bar{j}$	$c+d$
Sum	$a+c$	$b+d$	$n=a+b+c+d$

- Cosine similarity:

$$s_{\text{cosine}} = \frac{a}{\sqrt{(a+b)(a+c)}} \tag{5}$$

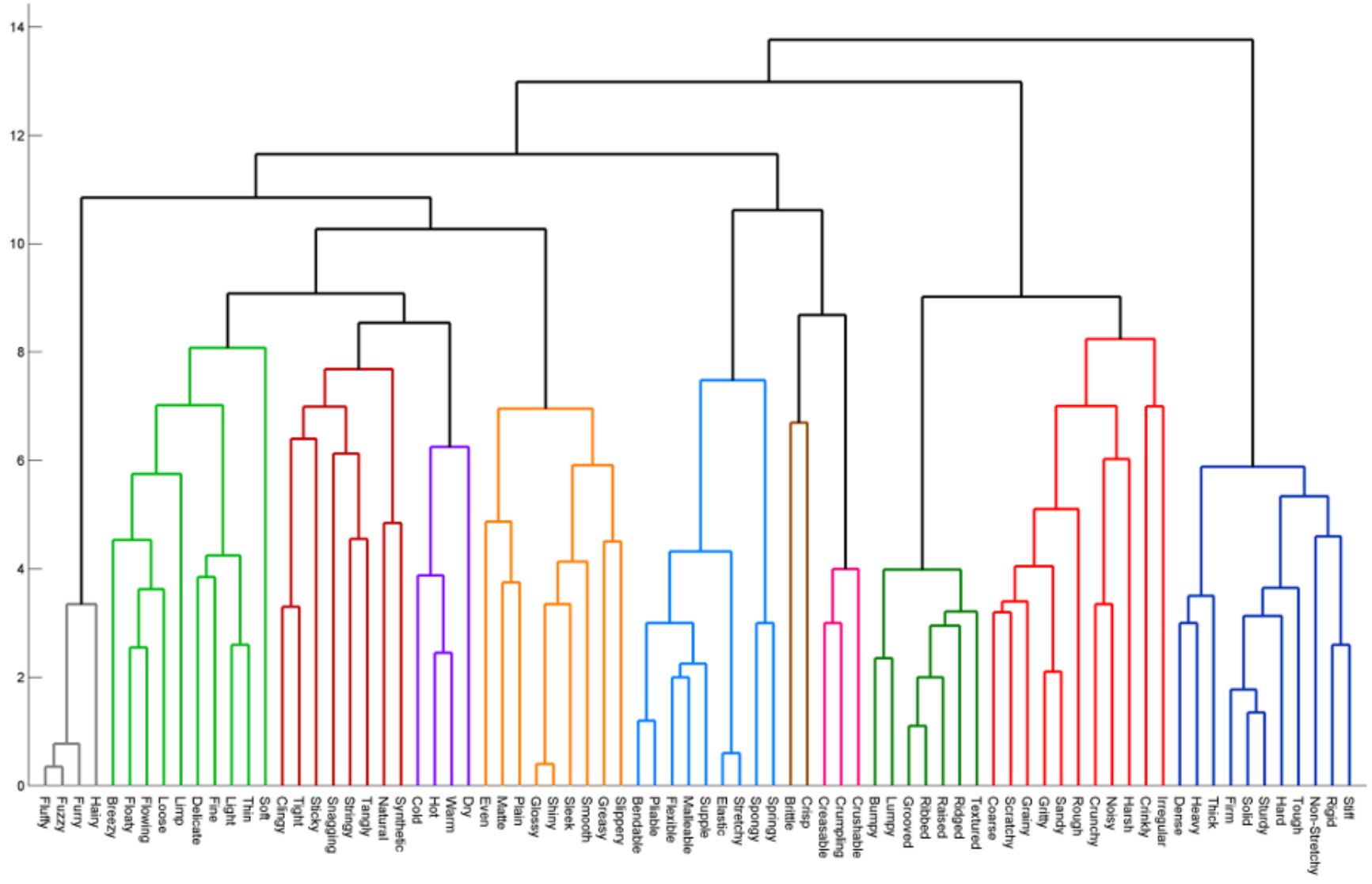
- Jaccard similarity:

$$s_{\text{Jaccard}} = \frac{a}{\sqrt{(a+b+c)}} \tag{6}$$

Typical features

- Normalized term frequency (almost certainly)
- Very common to use tf-idf – if not, similarity is boosted by common words (stop words)
- Not as common to use binary features

Uses for similarity measures: Clustering



Other uses, extensions

- Used extensively in information retrieval
- Summmary measures of how far apart two texts are – but be careful exactly how you define "features"
- Some but not many applications in social sciences to measure substantive similarity — scaling models are generally preferred
- Can be used to generalize or represent features in machine learning, by combining features using kernel methods to compute similarities between textual (sub)sequences without extracting the features explicitly (as we have done here)

The idea of "clusters"

- Essentially: groups of items such that inside a cluster they are very similar to each other, but very different from those outside the cluster
- "unsupervised classification": cluster is not to relate features to classes or latent traits, but rather to estimate membership of distinct groups
- groups are given labels through post-estimation interpretation of their elements
- typically used when we do not and never will know the "true" class labels
- issues: how to weight distance is arbitrary
  - which dimensionality? (determined by which features are selected)
  - how to weight distance is arbitrary
  - different metrics for distance

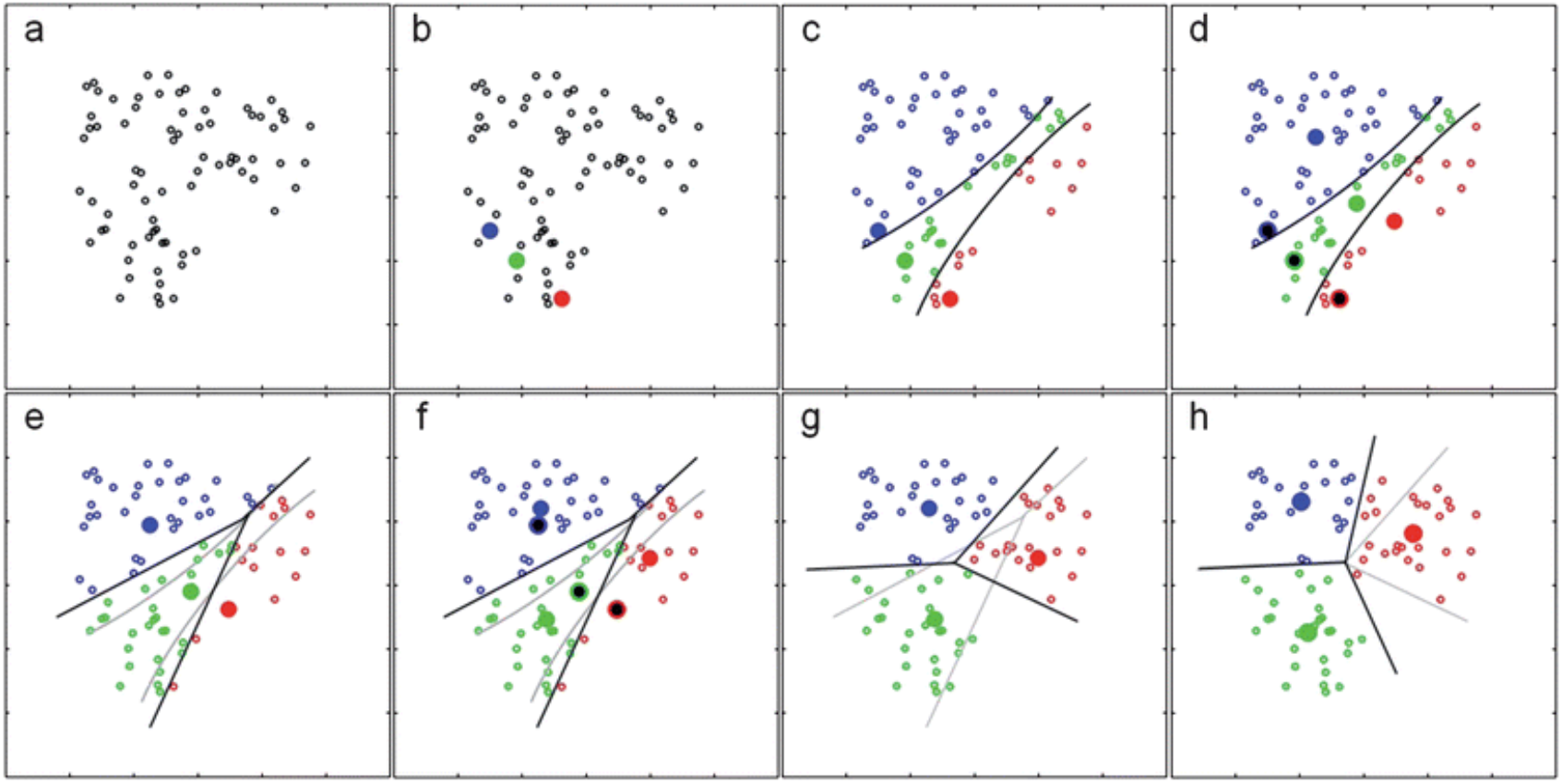
k-means clustering

- Essence: assign each item to one of k clusters, where the goal is to minimised within-cluster difference and maximize between-cluster differences
- Uses random starting positions and iterates until stable
- as with *kNN*, k-means clustering treats feature values as coordinates in a multi-dimensional space
- Advantages
  - simplicity
  - highly flexible
  - efficient
- Disadvantages
  - no fixed rules for determining k
  - uses an element of randomness for starting values

algorithm details

1. Choose starting values
   - assign random positions to k starting values that will serve as the "cluster centres", known as "centroids" ; or,
   - assign each feature randomly to one of k classes
2. assign each item to the class of the centroid that is "closest"
   - Euclidean distance is most common
   - any others may also be used (Manhattan, Minkowski, Mahalanobis, etc.)
   - (assumes feature vectors have been normalised within item)
3. update: recompute the cluster centroids as the mean value of the points assigned to that cluster
4. repeat reassignment of points and updating centroids
5. repeat 2–4 until some stopping condition is satisfied
   - e.g. when no items are reclassified following update of centroids

k-means clustering illustrated

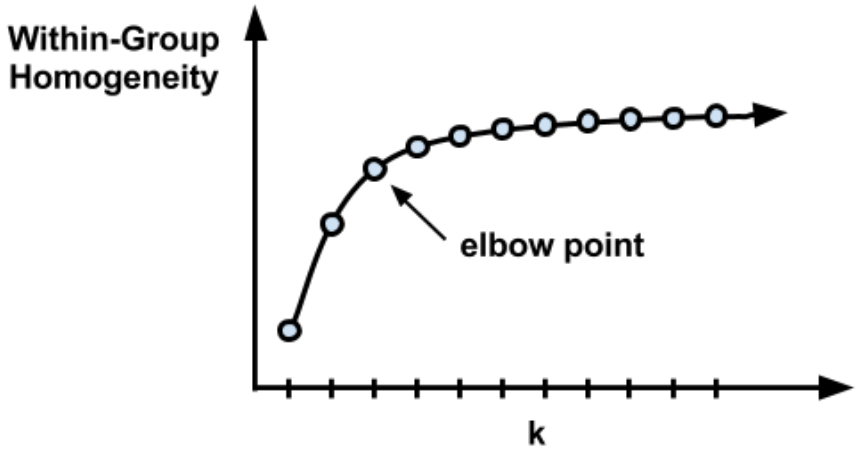


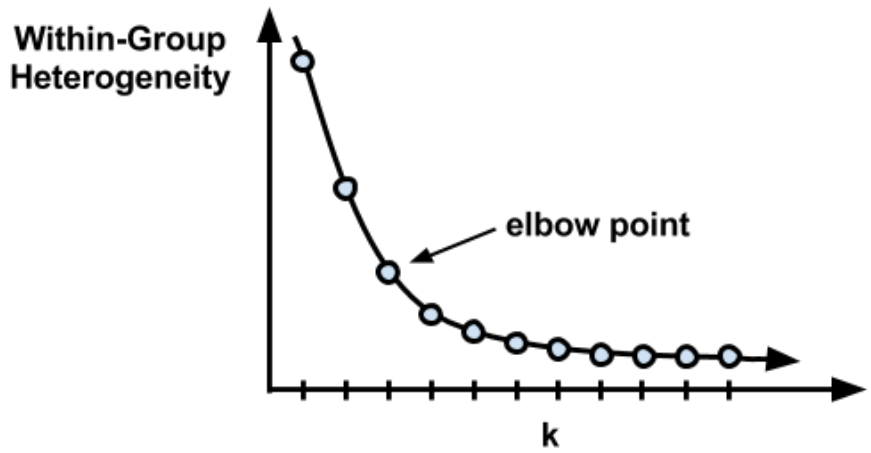
choosing the appropriate number of clusters

- very often based on prior information about the number of categories sought
  - for example, you need to cluster people in a class into a fixed number of (like-minded) tutorial groups
- a (rough!) guideline: set $k = \sqrt{N/2}$ where N is the number of items to be classified
  - usually too big: setting k to large values will improve within-cluster similarity, but risks *overfitting*

choosing the appropriate number of clusters

- "elbow plots": fit multiple clusters with different k values, and choose k beyond which are diminishing gains





choosing the appropriate number of clusters

- "fit" statistics to measure homogeneity within clusters and heterogeneity in between
-
- numerous examples exist
- "iterative heuristic fitting"* (IHF) (trying different values and looking at what seems most plausible)

* Warning: This is my (slightly facetious) term only!

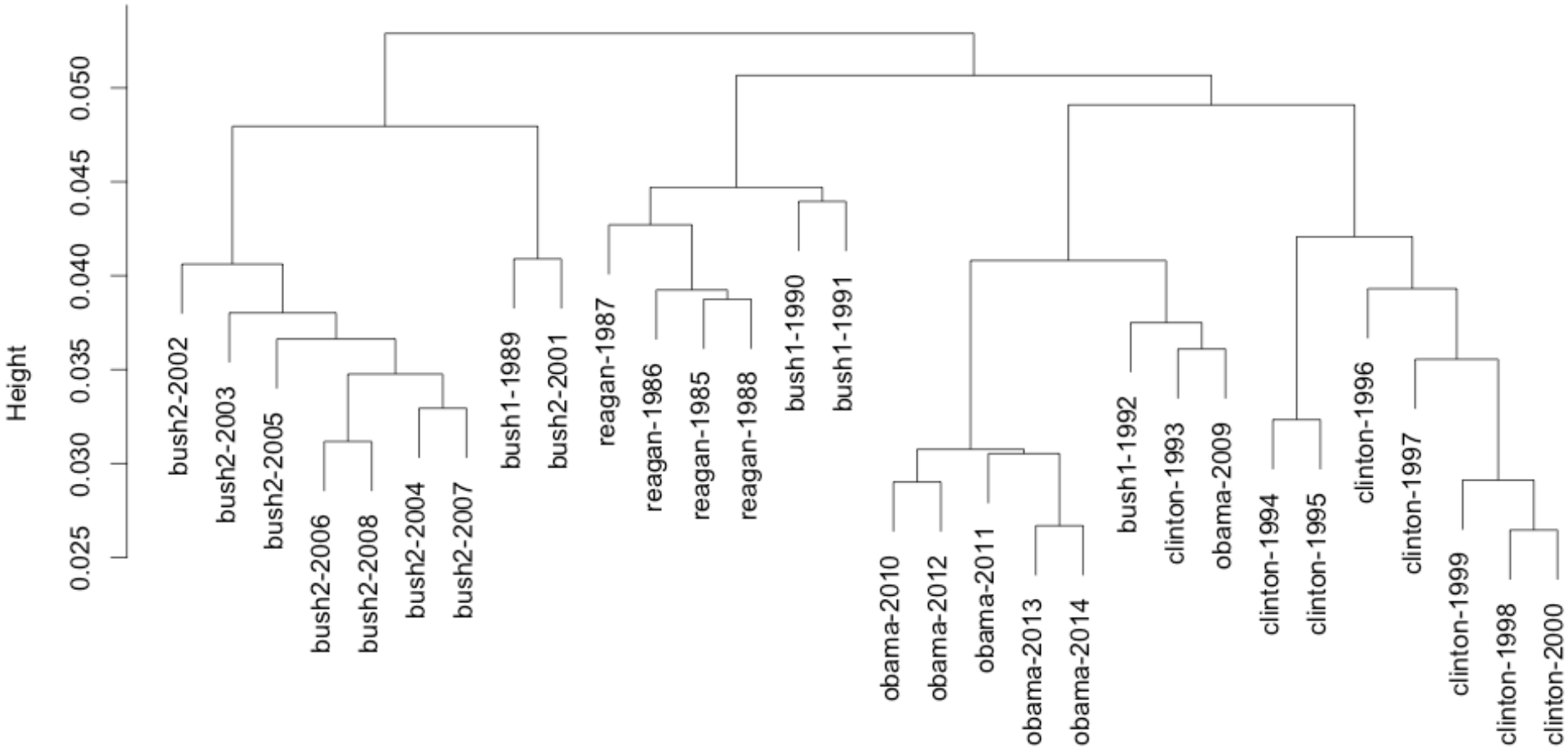
Other clustering methods: hierarchical clustering

- *agglomerative*: works from the bottom up to create clusters
- like *k*-means, usually involves *projection*: reducing the features through either selection or projection to a lower-dimensional representation
  1. local projection: reducing features within document
  2. global projection: reducing features across all documents (Schütze and Silverstein, 1997)
  3. SVD methods, such PCA on a normalised feature matrix
  4. usually simple threshold-based truncation is used (keep all but 100 highest frequency or tf-idf terms)
- frequently/always involves weighting (normalising term frequency, tf-idf)

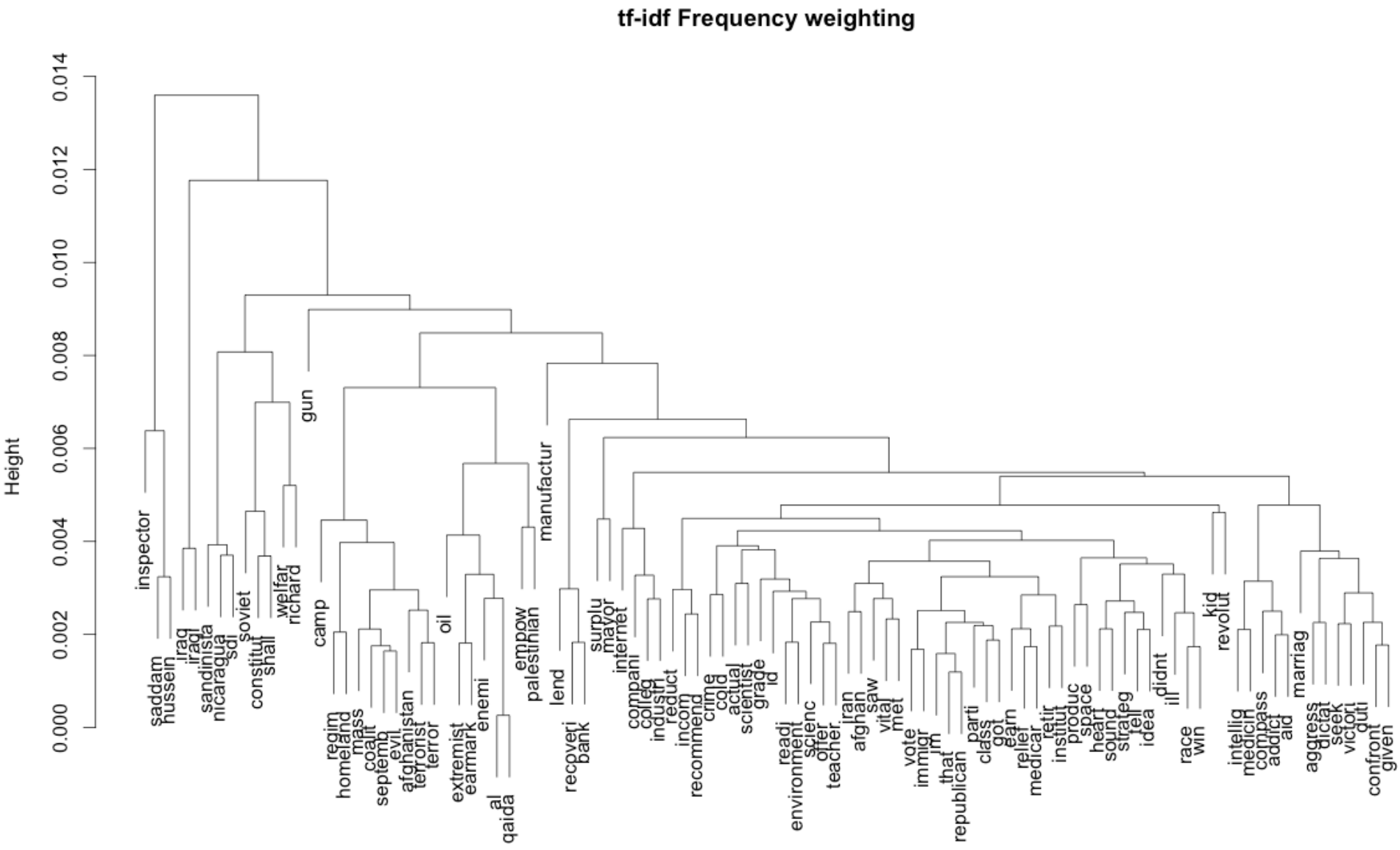
hierarchical clustering algorithm

1. start by considering each item as its own cluster, for n clusters
2. calculate the $N(N-1)/2$ pairwise distances between each of the n clusters, store in a matrix D_0
3. find smallest (off-diagonal) distance in D_0, and merge the items corresponding to the i, j indexes in D_0 into a new "cluster"
4. recalculate distance matrix D_1 with new cluster(s). options for determining the location of a cluster include:
   - centroids (mean)
   - most dissimilar objects
   - Ward's measure(s) based on minimising variance
5. repeat 3–4 until a stopping condition is reached
   - e.g. all items have been merged into a single cluster
6. to plot the *dendrograms*, need decisions on ordering, since there are $2^{(N-1)}$ possible orderings

Dendrogram: Presidential State of the Union addresses



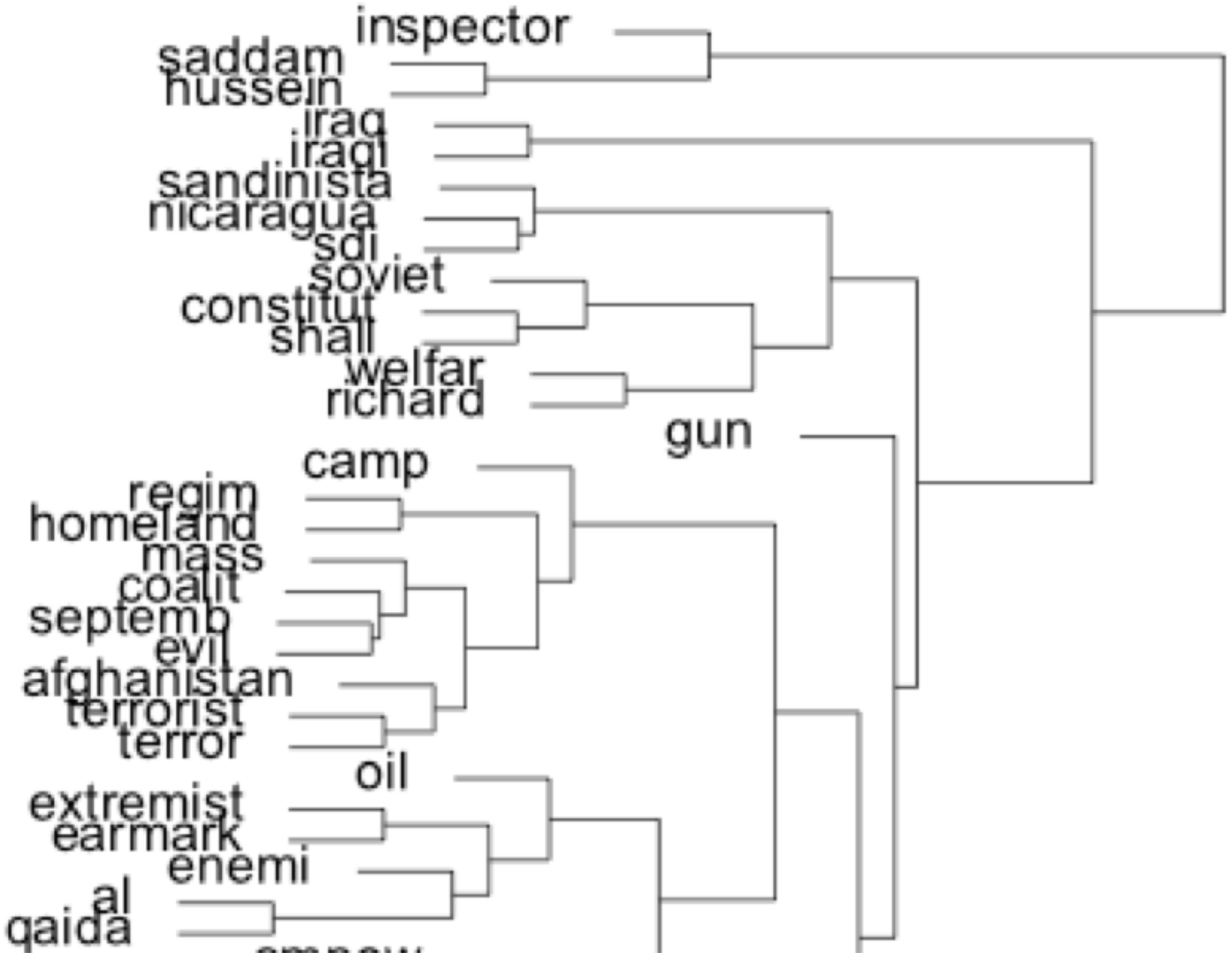
Dendrogram: Presidential State of the Union addresses



pros and cons of hierarchical clustering

- advantages
  - deterministic, unlike k-means
  - no need to decide on k in advance (although can specify as a stopping condition)
  - allows hierarchical relations to be examined (usually through *dendrograms*)
- disadvantages
  - more complex to compute: quadratic in complexity: $O(n^2)$ – whereas k-means has complexity that is $O(n)$
  - the decision about where to create branches and in what order can be somewhat arbitrary, determined by method of declaring the "distance" to already formed clusters
  - for words, tends to identify collocations as base-level clusters (e.g. "saddam" and "hussein")

Dendrogram: Presidential State of the Union addresses



Topic Models

- Topic models are algorithms for discovering the main "themes" in an unstructured corpus
- Requires no prior information, training set, or special annotation of the texts
  – only a decision on K (number of topics)
- A probabilistic, generative advance on several earlier methods, "Latent Semantic Analysis" (LSA) and "probabilistic latent semantic indexing" (pLSI)