Similarity and Clustering

Raf Alvarado
UVA DS 5001

Similarity and distance, distance measures, normalization, clustering, hierarchical, dendrograns, wine review examples

Review: Vector Space Models

Vector space models are the **foundation** of much of text analytics

Vector Space Models allow us to treat **texts as points** in a coordinate space

- This allows us to **compare texts** using various **distance measures**
- See Salton, Wong, and Yang (1975) for an **early description** of the method — IR, not AI!

Vector spaces have a general **structure** that applies to many things

- Features = **items**
- Events = **containers** = baskets = contexts = embeddings

More of a **form of representation** than a **model**

Similarity and Difference

	Document 1	Document 2	Document 3	Document 4	Document 5	Document 6	Document 7	Document 8
Term(s) 1	10	0	1	0	0	0	0	2
Term(s) 2	0	2	0	0	0	18	0	2
Term(s) 3	0	0	0	0	0	0	0	2
Term(s) 4	6	0	0	4	6	0	0	0
Term(s) 5	0	0	0	0	0	0	0	2
Term(s) 6	0	0	1	0	0	1	0	0
Term(s) 7	0	1	8	0	0	0	0	0
Term(s) 8	0	0	0	0	0	3	0	0

Word Vector
(Passage Vector)

Document Vector

Recall that a **bag-of-words** representation of the **TOKEN** table in our data model can be converted into a **matrix**, which can be viewed as a vector space, **either of word or document vectors**

This matrix consists of a collection of vector **pairs**

$$t_2 = (0, 2, 0, 0, 0, 18, 0, 0)$$
$$t_8 = (0, 0, 0, 0, 0, 3, 0, 0)$$

$$d_1 = (10, 0, 0, 6, 0, 0, 0, 0)$$
$$d_4 = (0, 0, 0, 4, 0, 0, 0, 0)$$

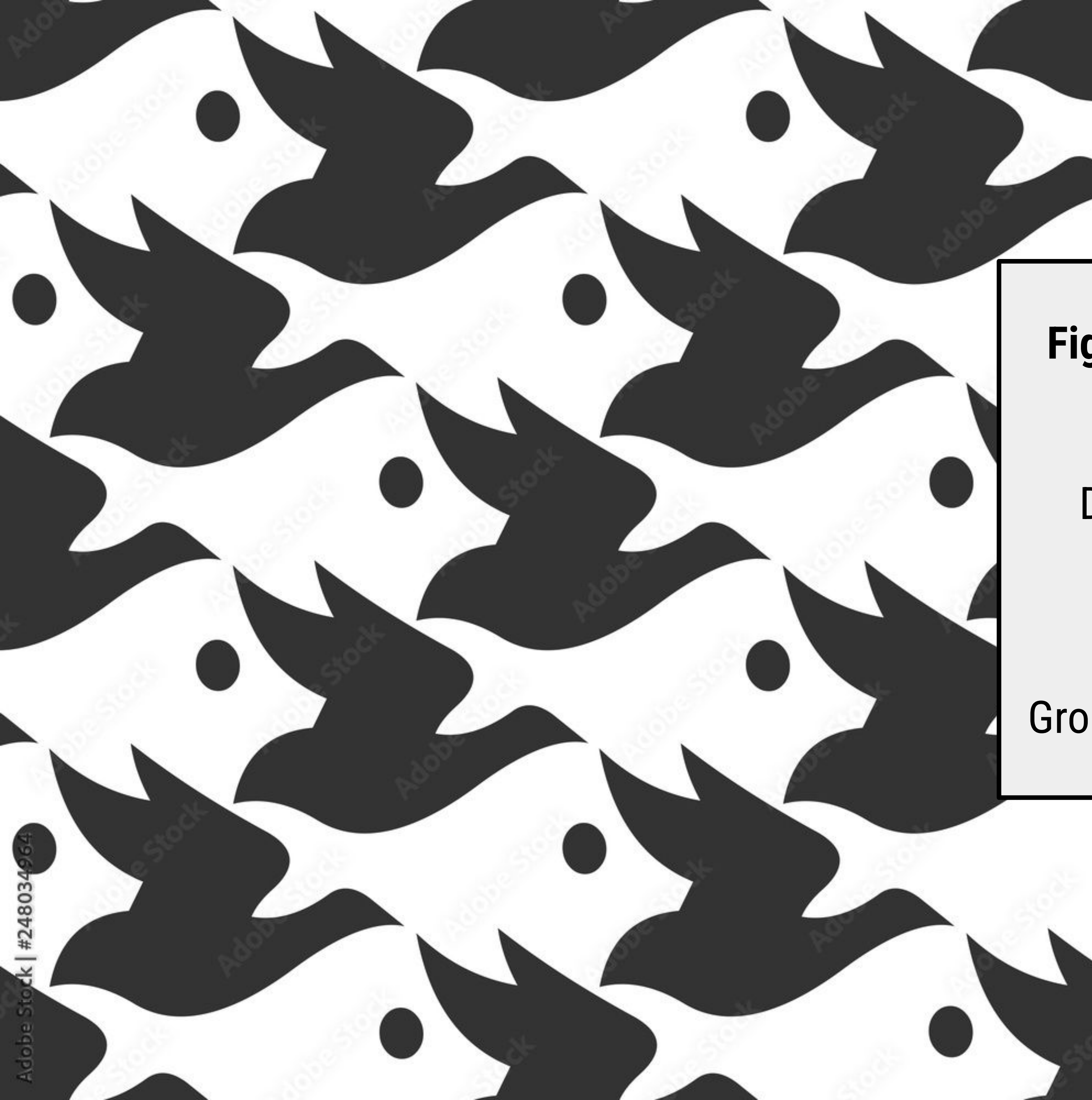


Figure and **Ground** Reversal

Docs : Words ::
Birds : Fish

Figure = item
Ground = embedding

Uses of Document and Word Vectors

The **vector space model** allows us to to **find similar documents *or* words**

With **document vectors** we can

- **find** documents that match **queries**
- **group** similar documents together (**clustering**)

With **word vectors** we can

- find **synonyms** or generate word **networks**

In **combination** we can

- Use word networks to **connect** documents
  - where documents are **nodes** and words are **edges**

Term-Term Matrices and Vector Semantics

Word-Context matrices are often **converted** into **term-term** matrices to explore **word similarity**

- Both axes contain the **vocabulary**
- Each cell contains the number of times the row and column words **co-occur** (in an OHCO container)
- Word **similarity** is computed by comparing word vector pairs

We will explore these relations (**vector semantics**) in more detail when we look at **PCA** and **word embedding**

	aardvark	...	computer	data	pinch	result	sugar	...
apricot	0	...	0	0	1	0	1	
pineapple	0	...	0	0	1	0	1	
digital	0	...	2	1	0	1	0	
information	0	...	1	6	0	4	0	

Figure 6.5 Co-occurrence vectors for four words, computed from the Brown corpus, showing only six of the dimensions (hand-picked for pedagogical purposes). The vector for the word *digital* is outlined in red. Note that a real vector would have vastly more dimensions and thus be much sparser.

Role of TF-IDF

We often **use TF-IDF weighted document vectors** to compute similarities among documents

- Documents that **share the same significant words** are considered **similar**

We also can **cull the most significant terms** to create shorter vectors of significant words

- Shorter vectors mean **faster** compute times
- Useful when comparing **all pairs of vectors**

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

n = number of vectors (docs)
k = 2 (for pairs)

= $1{,}249{,}975{,}000.0$
for $n = 50{,}000$

Comparing Documents in Word Space

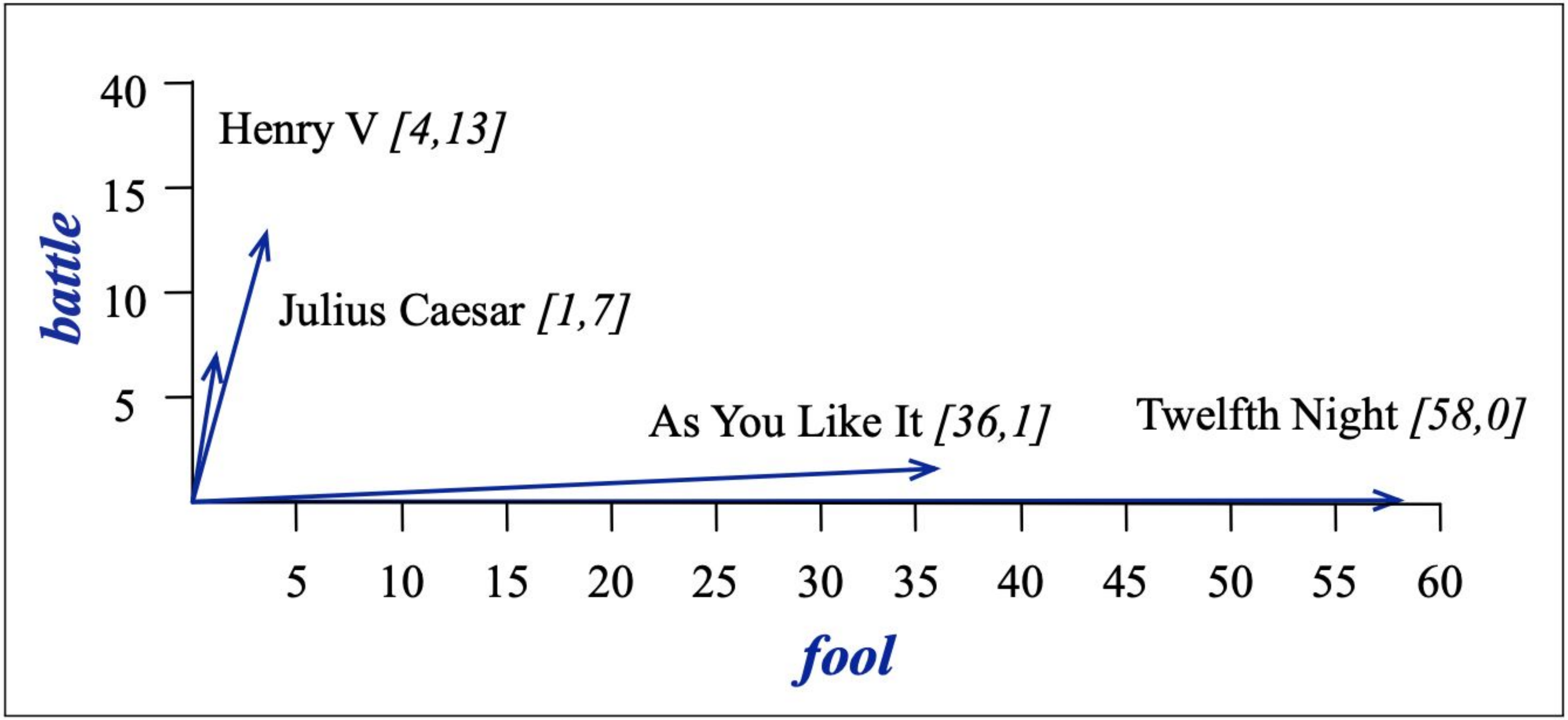
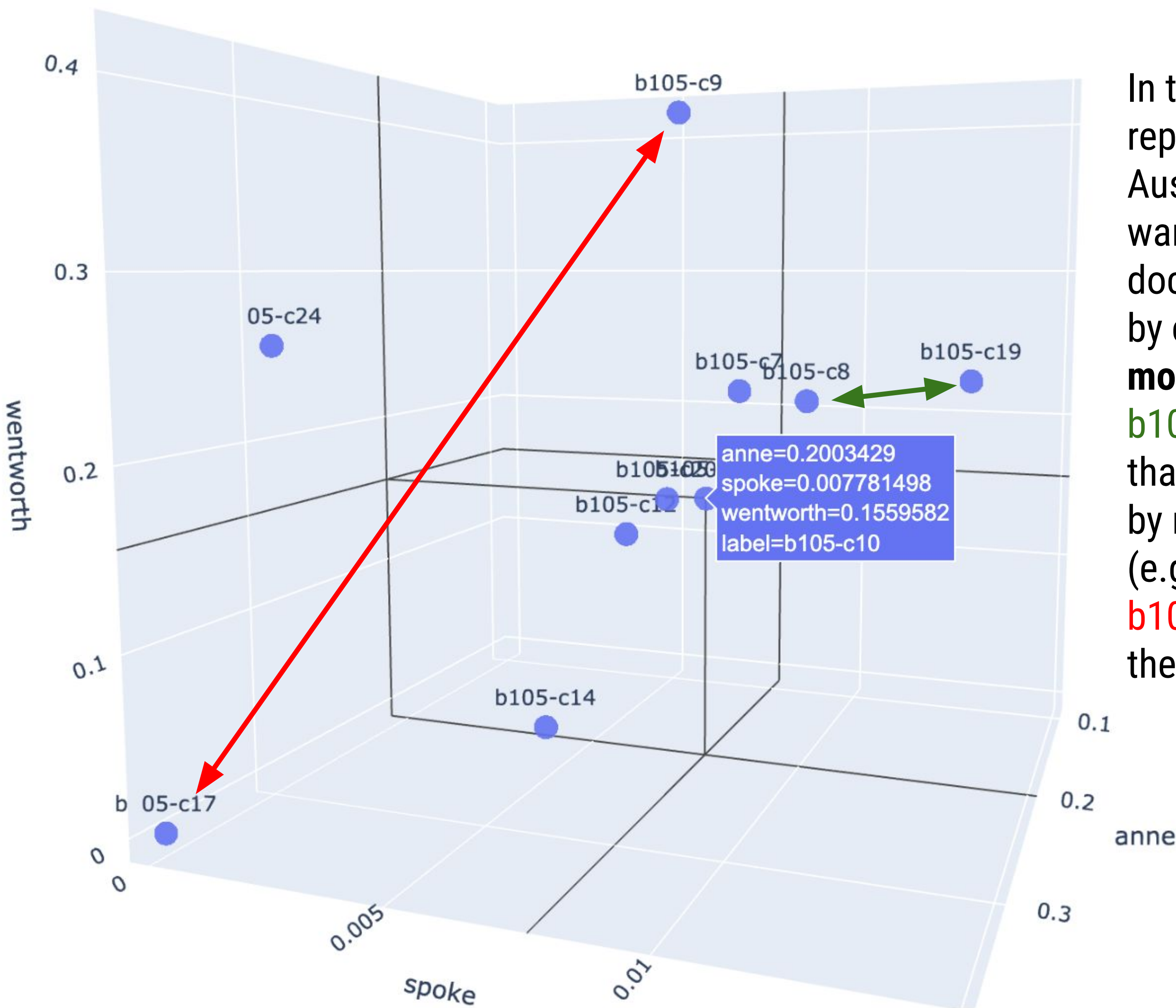


Figure 6.4 A spatial visualization of the document vectors for the four Shakespeare play documents, showing just two of the dimensions, corresponding to the words *battle* and *fool*. The comedies have high values for the *fool* dimension and low values for the *battle* dimension.



In this vector space representation of Jane Austen's *Persuasion*, we want to say that documents represented by **close coordinates are more similar** (e.g. b105-c**8** and b105-c**19** than those represented by more distant ones (e.g. b105-c**17** and b105-c**9**) relative to these dimensions

Similarity and Distance Measures

A variant of the Statistical Semantics Hypothesis

Perhaps: *Geometrical* Semantics Hypothesis?

Geometry of/as Meaning → Structuralism

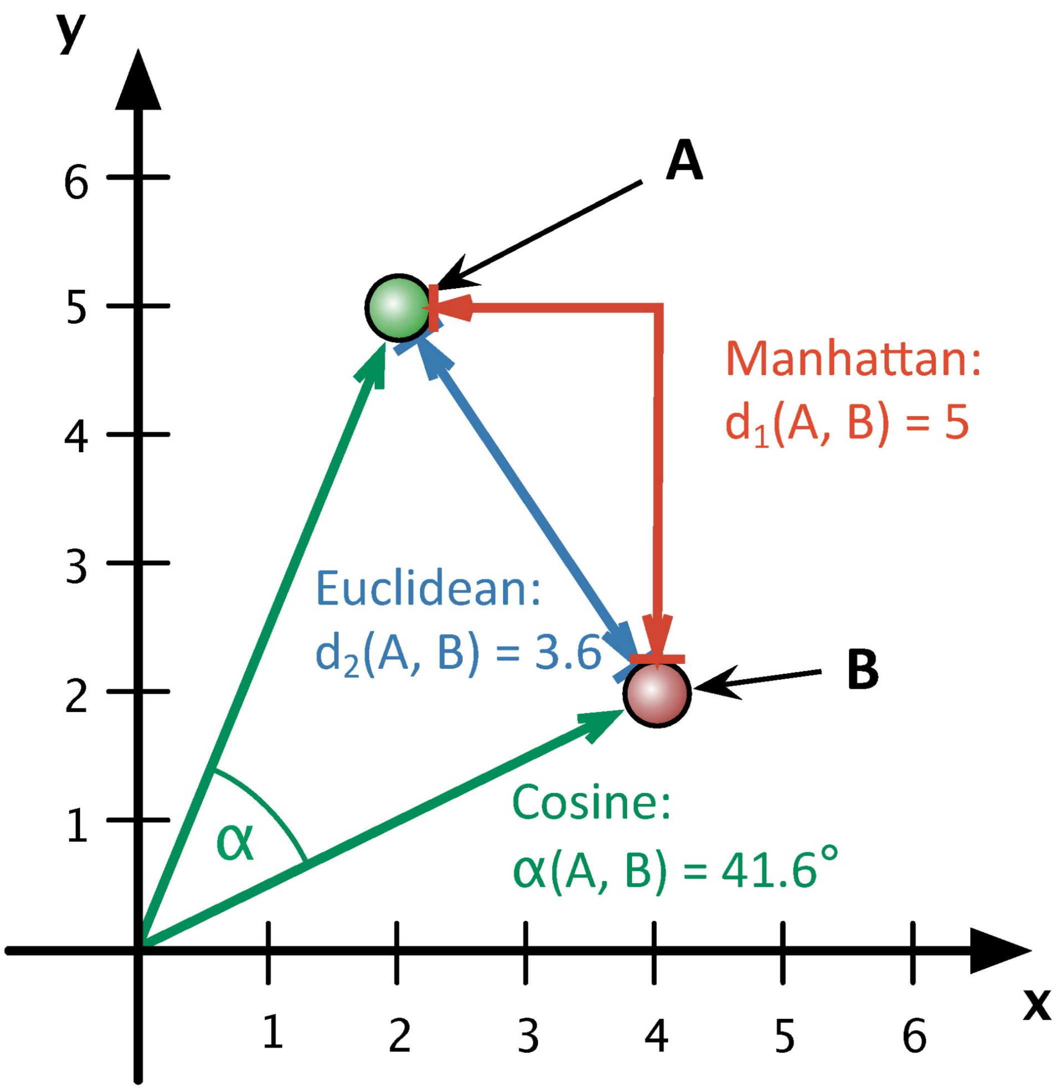
Qualitative concept operationalized in terms of **space**

similarity → **proximity** in vector space
difference → **distance** in vector space

Expressed as a **functions** of vector pairs

similarity: $\mathrm{sim}(a, b)$ greater is **closer** is **more** similar
distance: $d(a, b)$ greater is **farther** is **less** similar

Many functions have been developed for each measure



Here are **three common measures** in Cartesian space

A, B = Documents
x, y = Words (terms)

Euclidean and Manhattan are **distance** measures

Cosine is a **similarity** measure

We will look at each of these and others

Similarity and Distance Measures

Quantitatively, **similarity** is often computed as a **reverse function of distance**

Method depends on range of distance function

By inversion

$$\text{sim}(a, b) = 1 / d(a, b)$$

$\text{sim}(a, b) = 1 / (d(a, b) + 1)$ to avoid division by zero where $a = b$

By subtraction

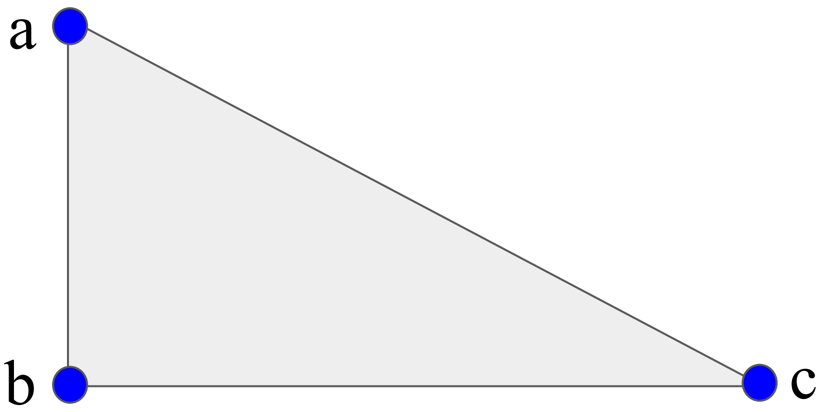
$$\text{sim}(a, b) = 1 - d(a, b)$$

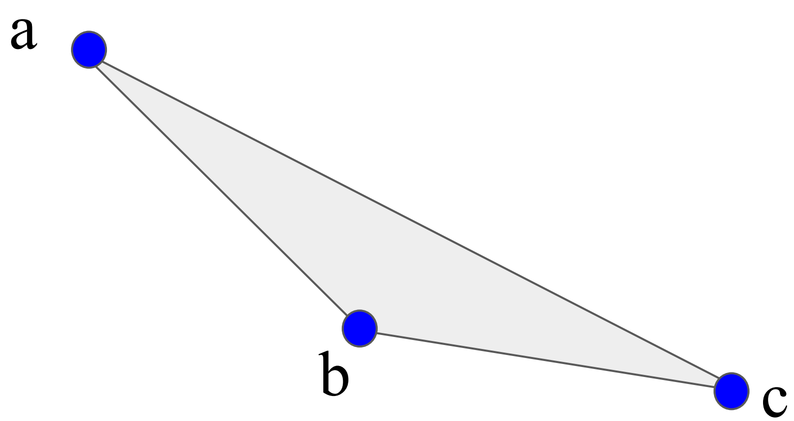
$$d(a, b) = 1 - \text{sim}(a, b)$$

Some Properties of Distance Measures

Distance, as a true **metric**, is subject to the **triangle inequality law**

$$d(a, c) \leq d(a, b) + d(b, c)$$





Distance is also a **symmetric** property

$$d(a, b) = d(b, a)$$

Divergence measures (such as Kullback-Leibler) **are not** subject to these laws (!)

Why is Divergence Asymmetric?

Definition 5. ***Conditional Entropy of X given Y***

$$H(X \mid Y) \triangleq \mathbb{E}\left[\log \frac{1}{\boxed{p(X \mid Y)}}\right] \tag{30}$$

$$= \sum_{x,y} p(x,y) \log \frac{1}{p(x \mid y)} \tag{31}$$

$$= \sum_{y} p(y) \left[\sum_{x} p(x \mid y) \log \frac{1}{p(x \mid y)}\right] \tag{32}$$

$$= \sum_{y} p(y) H(X \mid Y = y). \tag{33}$$

Source (Weissman 2018)

Entropy measures are transformations of probability measures

Relative entropy is a transformation of **conditional probability**

Varieties of Measures

There are **many** measures of distance, similarity, and divergence

These may be **grouped according to the kind of count values** in the vector space:

For **binary** counts, use measures based on **set theory**

For **numeric** counts, use measures based on **geometry**

For **probabilities** (vectors normed to values that sum to one), use measures based on **information theory**

For **strings** (discrete symbol sequences), use edit distance measures (also based on info theory)

	Binary counts Set theory	**Numeric counts** Geometry	**Probabilities** Info. Theory
SIMILARITY	Matching coefficient Dice **Jaccard** Overlap Cosine	Dot product Cosine Harmonic Mean Pearson's Corr.	
DISTANCE		**Manhattan** **Euclidean** Minkowski	Manhattan
DIVERGENCE			**Kullback-Leibler** Jensen-Shannon Info Radius Mutual Information

Count Distance Measures

Manhattan

$$d(a, b) = \sum_{i=1}^{n} | a_i - b_i|$$

This is also called `cityblock` and `taxicab` distance.

Euclidean

$$d(a, b) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2}$$

This is just the Pythagorean theorem.

Minkowski

$$d(a, b) = (\sum_{i=1}^{n} | a_i - b_i |^p)^{\frac{1}{p}}$$

Note that Minkowski distance is just the general rule.

- Manhattan distance is just where $p = 1$
- Euclidean distance is just where $p = 2$.

a, b : a pair of vectors of equal length

n : number of elements in each vector

i : index of element in a vector

Count Similarity Measures

Simple dot product

$$sim(a, b) = \sum_{n=i}^{n} a_i b_i = a \cdot b$$

This returns an unbounded value, and favors long documents.

Remember that the **dot product** of two vectors is just the **sum** of their elements' products

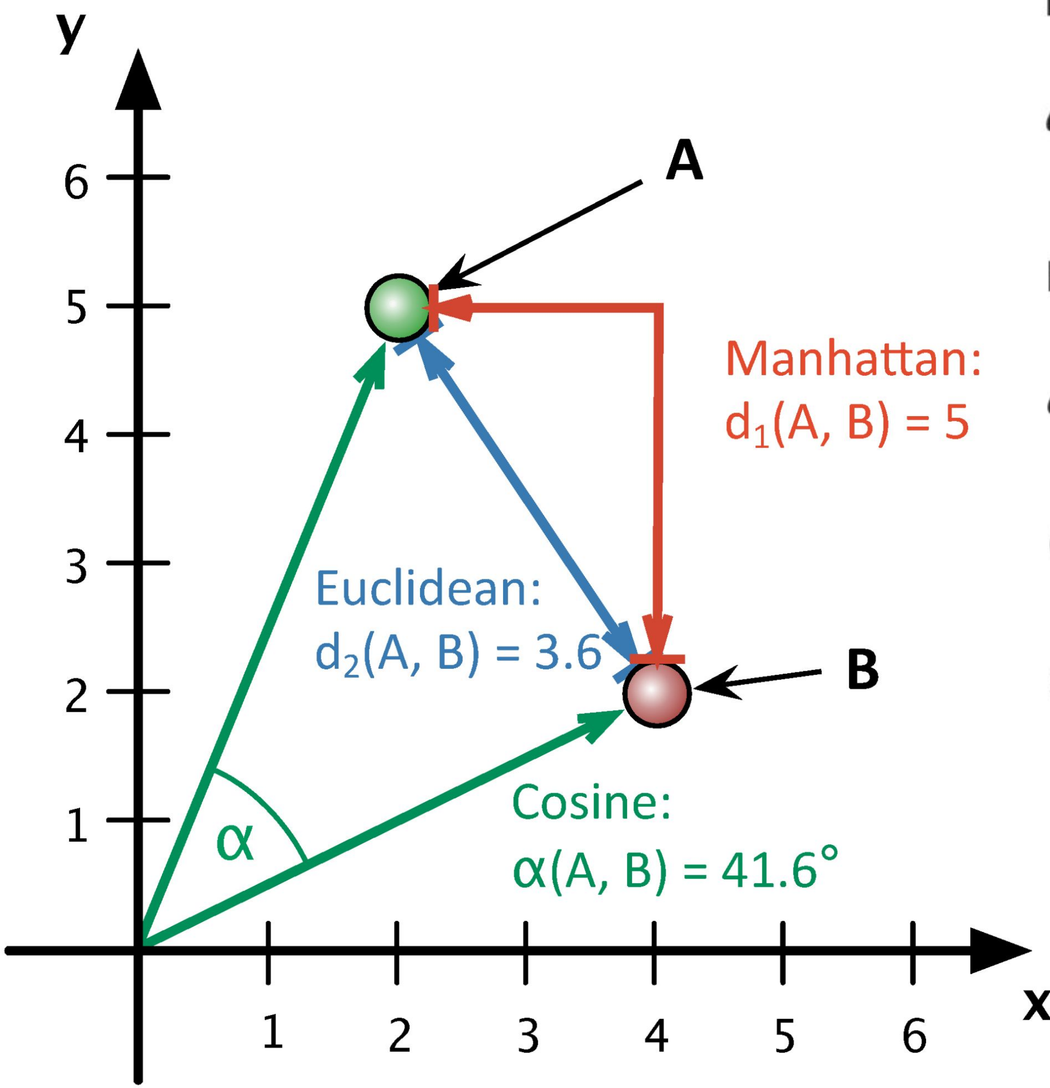
Cosine

$$sim(a, b) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} a_i^2} \sqrt{\sum_{i=1}^{n} b_i^2}} = \frac{a \cdot b}{\|a\| \|b\|}$$

This is better — it returns a values between -1 and 1,
or 0 and 1 if all values are non-negative.

By far the most common metric used in text analytics. Assumes Euclidean space.

Also written as $CosSim(a, b)$.



Manhattan

$$d(a, b) = \sum_{i=1}^{n} | a_i - b_i |$$

Euclidean

$$d(a, b) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2}$$

Cosine

$$sim(a, b) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} a_i^2} \sqrt{\sum_{i=1}^{n} b_i^2}} = \frac{a \cdot b}{\|a\| \|b\|}$$

Normalization

To normalize a vector is to **divide each element by the length** of its norm $\mathrm{L_p}$

Useful when we want to **discount length** (e.g. length of docs)

Also for computational advantages

Length measures **vary** by p

$\mathrm{L_1}$ = sum of absolute value of each element (Probability)

$\mathrm{L_2}$ = square root of sum of each element squared

$\mathrm{L_P}$ = $\left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}$

Sum of $\mathrm{L_p}$ normed values raised to p should $= 1$

$\mathrm{L_p}$ is Minkowski distance from origin, the general formula

Normalization as Distance to Origin

$$d(a, b) = \left(\sum_{i=1}^{n} \mid a_i - b_i \mid^p\right)^{\frac{1}{p}}$$

Minkowski Distance

$$L_P = \left(\sum_{i=1}^{n} |a_i|^p\right)^{1/p}$$

L_p Norm

A norm is just the distance from the origin, i.e. where $b_i = 0$

Cosine Similarity is just the
L_2 **normalized dot product** of two vectors

It is identical to taking the dot product
of two L_2 normalized vectors

L_2 = Euclidean

Intuitive Understanding of Vector Difference

	Likes Cats	Likes Dogs	Likes Meat	Likes Movies	Likes Python	SUM
Bob	0	1	1	1	0	
Sue	1	1	0	1	1	
DIFF	1	0	$\|-1\| = 1$	0	1	3

Here **distance** is **difference** (subtraction)

Same as logical **XOR** for truth tables

The higher the value, the more different

What would the SUM be if they are identical?

Intuitive Understanding of Vector Similarity

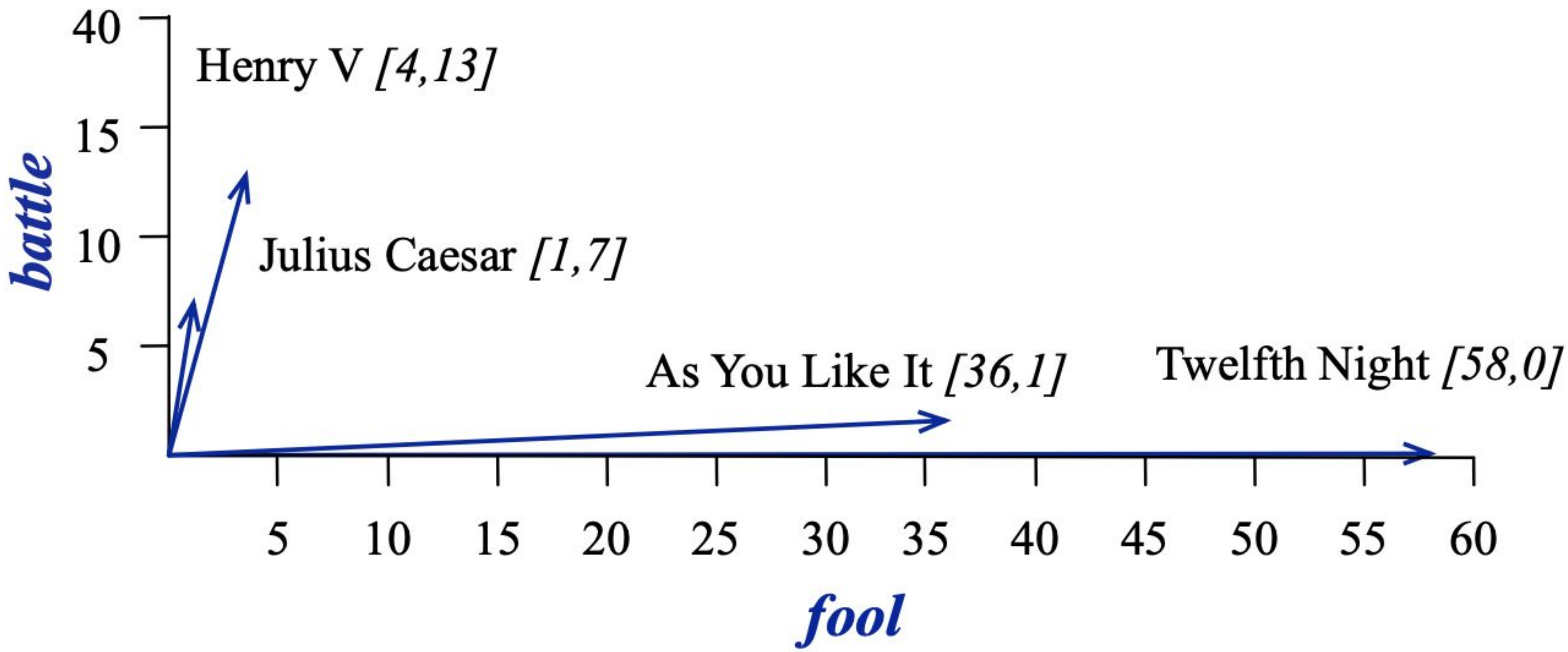
	Likes Cats	Likes Dogs	Likes Meat	Likes Movies	Likes Python	SUM
Bob	0	1	1	1	0	
Sue	1	1	0	1	1	
MATCH	0	1	0	1	0	2

Here **similarity** is a computed by **multiplication**

Same as logical **AND** for truth tables

The higher the value, the more similar

If each MATCH value is 0, how are the vectors oriented?



Two vectors that have nothing in common are **orthogonal**

Their angle is **90°** (and the cosine of 90° = 0)

You can see how **Henry V** and **Twelfth Night** are roughly orthogonal in this space

Negative Values for Cosine Similarity

- Negative values are sometimes considered **uninterpretable**
  - Sample sizes need to be **large** to ensure meaningful values
  - Often **converted** to 0
- In ETA, we **may consider negatives**
  - May signify **oppositions** — a key concept in **structuralist poetics**
  - Human symbolic structures are built out **dyads**
    - left / right, male / female, sun / moon, etc.
  - Statistical significance not as important to establish non-randomness
    - Writer's intent and/or reader's response may account for "significance"

Binary Similarity Measures (Set Theory)

Matching Coefficient

$$sim(a,b) = |a \cap b|$$

This is just the sum of the intersection of ones in both vectors.
Recall that as a set operation, intersection counts **unique** terms shared by both vectors.
The value is unbounded, so privileges vector length.

Dice

$$sim(a,b) = \frac{2|a \cap b|}{|a| + |b|}$$

This normalizes the matching coefficient, and returns of a value of 0 or 1.

Jaccard

$$sim(a,b) = \frac{|a \cap b|}{|a \cup b|} = \frac{|ab|}{|a \cup b|}$$

This penalizes vectors that have small overlap.

Overlap

$$sim(a,b) = \frac{|a \cap b|}{min(|a|,|b|)}$$

Cosine

$$sim(a,b) = \frac{|a \cap b|}{\sqrt{|a| \times |b|}}$$

Probability Divergence Measures

Kullback-Leibler (KL)

$$D_{KL}(a||b) = \sum_{i=1}^{n} a_i log(\frac{a_i}{b_i})$$

This is asymmetric; $D(a||b) \neq D(b||a)$.

Jensen-Shannon (JSD)

$$D_{JSD}(a||b) = \frac{D_{KL}(a||b)+D_{KL}(b||a)}{2}$$

Makes KL symmetric.

Review: Relative Entropy

Shannon, 1948: 24

The ratio of the entropy of a source to the maximum value it could have while still restricted to the same symbols will be called its *relative entropy*. This is the maximum compression possible when we encode into the same alphabet. One minus the relative entropy is the *redundancy*.

$$H / H_{max}$$

Wikipedia

In mathematical statistics, the **Kullback–Leibler (KL) divergence** (also called **relative entropy** and **I-divergence**[1]), denoted $D_{\mathrm{KL}}(P \parallel Q)$, is a type of statistical distance: a measure of how one probability distribution ***P*** is different from a second, reference probability distribution ***Q***.[2][3]

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log\left(\frac{P(x)}{Q(x)}\right)$$

$$P : Q :: H : H_{max}$$

Lee (1999) proposed that, for finding word similarities, measures that focused more on overlapping coordinates and less on the importance of negative features (i.e., coordinates where one word has a nonzero value and the other has a zero value) appear to perform better. In Lee's experiments, the Jaccard, Jensen-Shannon, and L1 measures seemed to perform best. Weeds et al. (2004) studied the linguistic and statistical properties of the similar words returned by various similarity measures and found that the measures can be grouped into three classes:

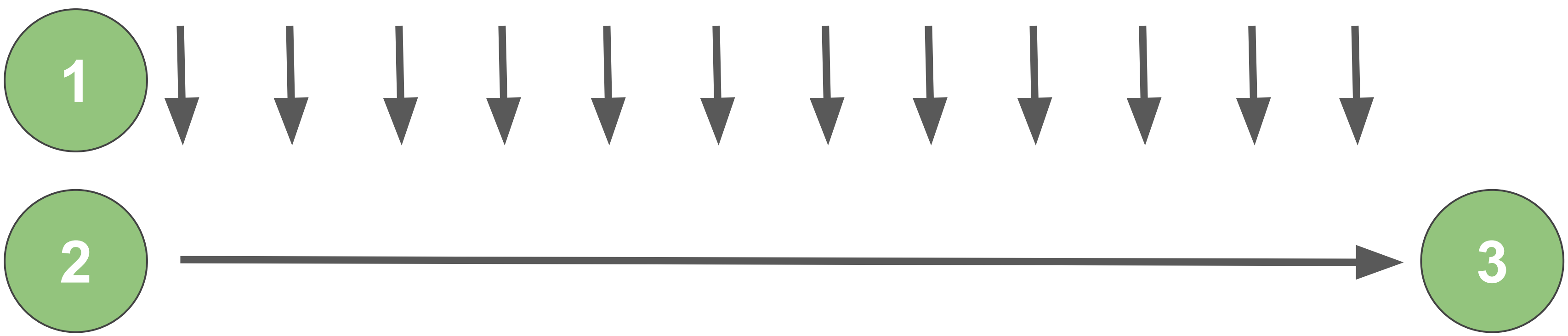
1. high-frequency sensitive measures (cosine, Jensen-Shannon, α-skew, recall),
2. low-frequency sensitive measures (precision), and
3. similar-frequency sensitive methods (Jaccard, Jaccard+MI, Lin, harmonic mean).

Given a word w_0, if we use a high-frequency sensitive measure to score other words w_i according to their similarity with w_0, higher frequency words will tend to get higher scores than lower frequency words. If we use a low-frequency sensitive measure, there will be a bias towards lower frequency words. Similar-frequency sensitive methods prefer a word w_i that has approximately the same frequency as w_0. In one experiment on determining the compositionality of collocations, high-frequency sensitive measures outperformed the other classes (Weeds et al., 2004). We believe that determining the most appropriate similarity measure is inherently dependent on the similarity task, the sparsity of the statistics, the frequency distribution of the elements being compared, and the smoothing method applied to the matrix.

From Turney and Pantel, p. 162.

Common Structure of Distance Measures

	0	1	2	3	4	5	6	7	8	9	10	11
a	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}
b	b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}	b_{11}



1: Pair-wise operations e.g. add, subtract, multiply, square, etc.

2. Result-wise operation, e.g. sum

3. Normalize result from 2

A Caution — The Curse of Dimensionality

As the number of dimensions increase, the less significant distance becomes

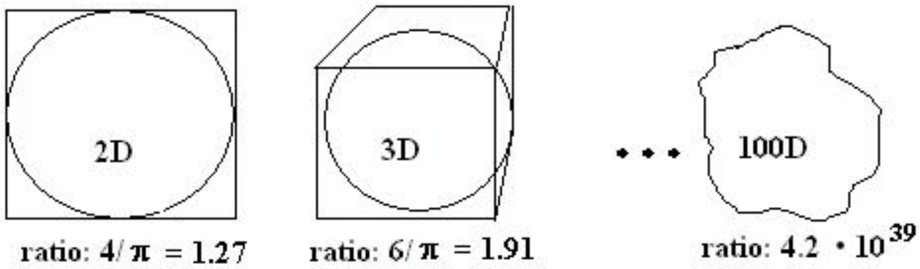
- Distances between pairs have less difference among them
- Therefore distance functions lose their meaning

Related to the **exponential growth in volume** ...

- **Ratio** of inscribed hypersphere to hypercube becomes **small**
- The hypersphere is the **unit sphere**, which is related to normalization

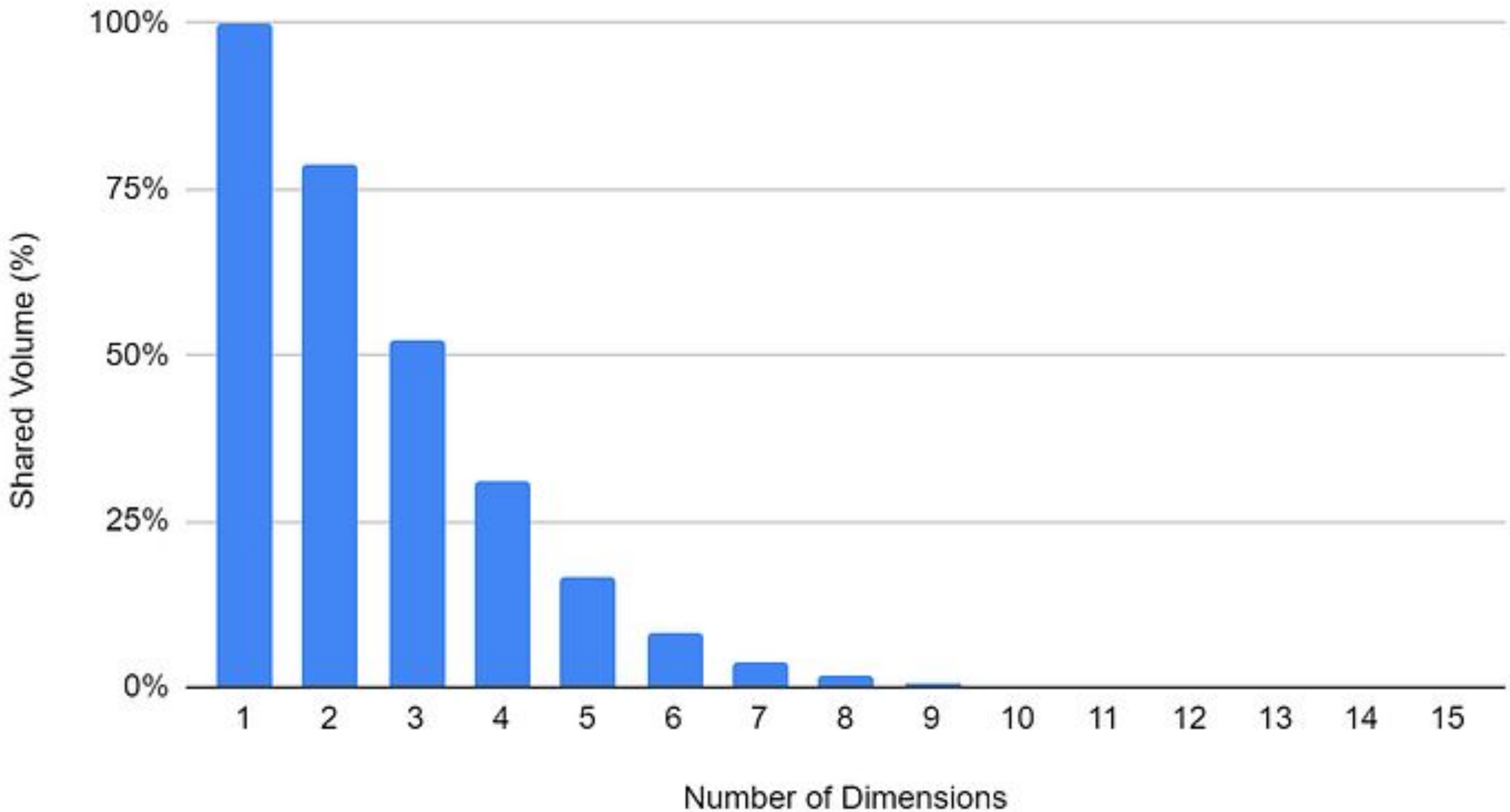
This is **complex** and **weird**

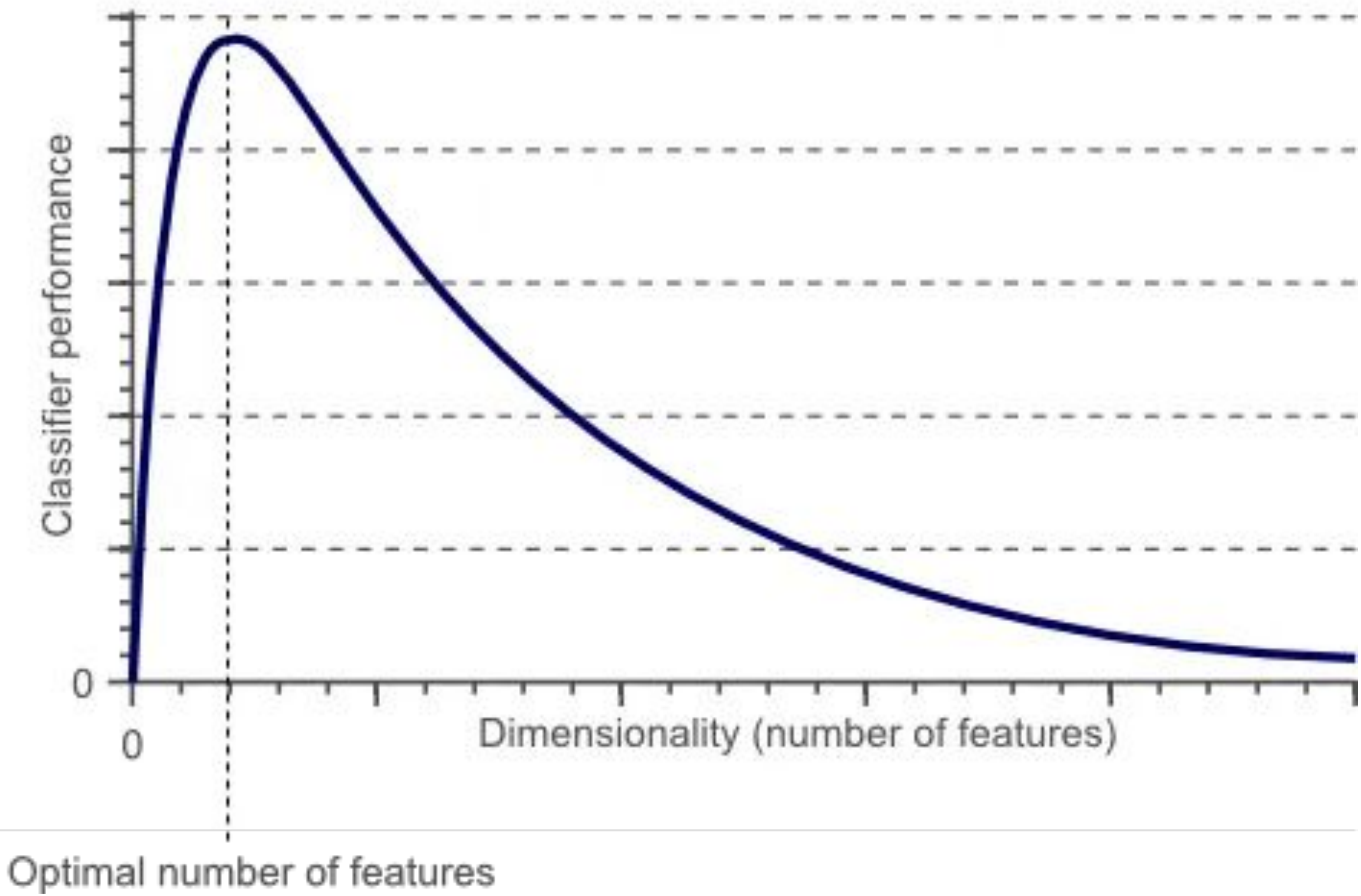
- Just know that distance measures have a sweet spot for dimensionality



$$\frac{\textit{volume hypersphere}}{\textit{volume hypercube}} = \frac{\frac{\pi^{n/2} r^n}{\Gamma(n/2+1)}}{(2r)^n}$$

Shared Volume between Hypercube with Inscribed Hypersphere





[Source](#) (Raj 2019)

Summary

The most **common** measurements in text analytics are:

- Manhattan, Euclidean, Cosine, Jaccard, and Jensen-Shannon

Among these, **cosine similarity is often used** because it is already **normalized for length**

- Euclidean distance on non-normalized vectors is sensitive to length
    - Two documents will rank as dissimilar if they discuss the same content but have different lengths
- Equivalent to the dot product of two normalized vectors

Jensen-Shannon has the value of being **based in information theory**

Tools

Python offers a number of libraries to compute distances, etc.

NumPy

`norm()` to normalize vectors (and matrices)

SciPy

`scipy.spatial.distance` → `pdist()`, `squareform()`
To create pair matrices of vectors by distance metric

SciKit Learn

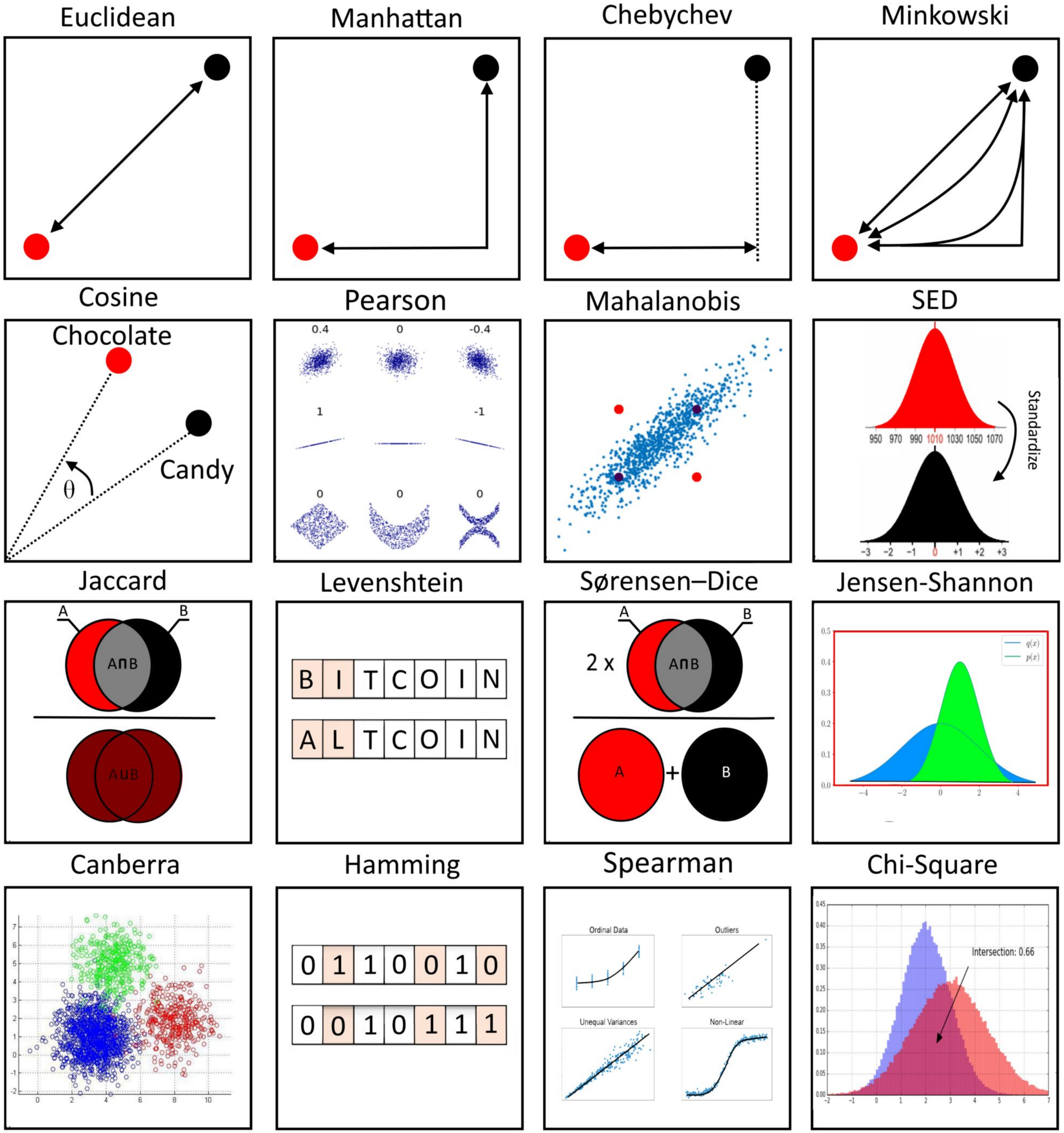
`preprocessing.normalize()` to normalize vectors and matrices

Pandas

`df.corr()`, `df.cov()` ⟶ but these are slow

Pydist2

`pdist1()` and `pdist2()`
Methods for calculating distances between observations



https://towardsdatascience.com/17-types-of-similarity-and-dissimilarity-measures-used-in-data-science-3eb914d2

Clustering

Clustering

Often used as a synonym for **unsupervised learning**

Actually, one of a few methods (include graph-based methods)

Clustering

Clustering is just the **grouping of vectors** (coordinates) based on their distances to each other

- Generates groups of documents based on their **pairwise distances**

Two main uses in NLP (and ETA):

- **Exploration (**EDA): Hence prominence of clustering for exploratory text analytics
- **Generalization**: Creating groups that may be used to draw inferences
  - e.g. our data has the prepositions for some **days of the week**, but not all. If days of the week form a cluster, we can treat them as a class and induce that all days of week take the seme preposition.

Clustering Algorithms

Many clustering algorithms, but in general there are **two types**:

Flat

Start with a set number of unrelated clusters (k)

Iteratively assign vectors to each cluster

K-means is classic example

→ Fast

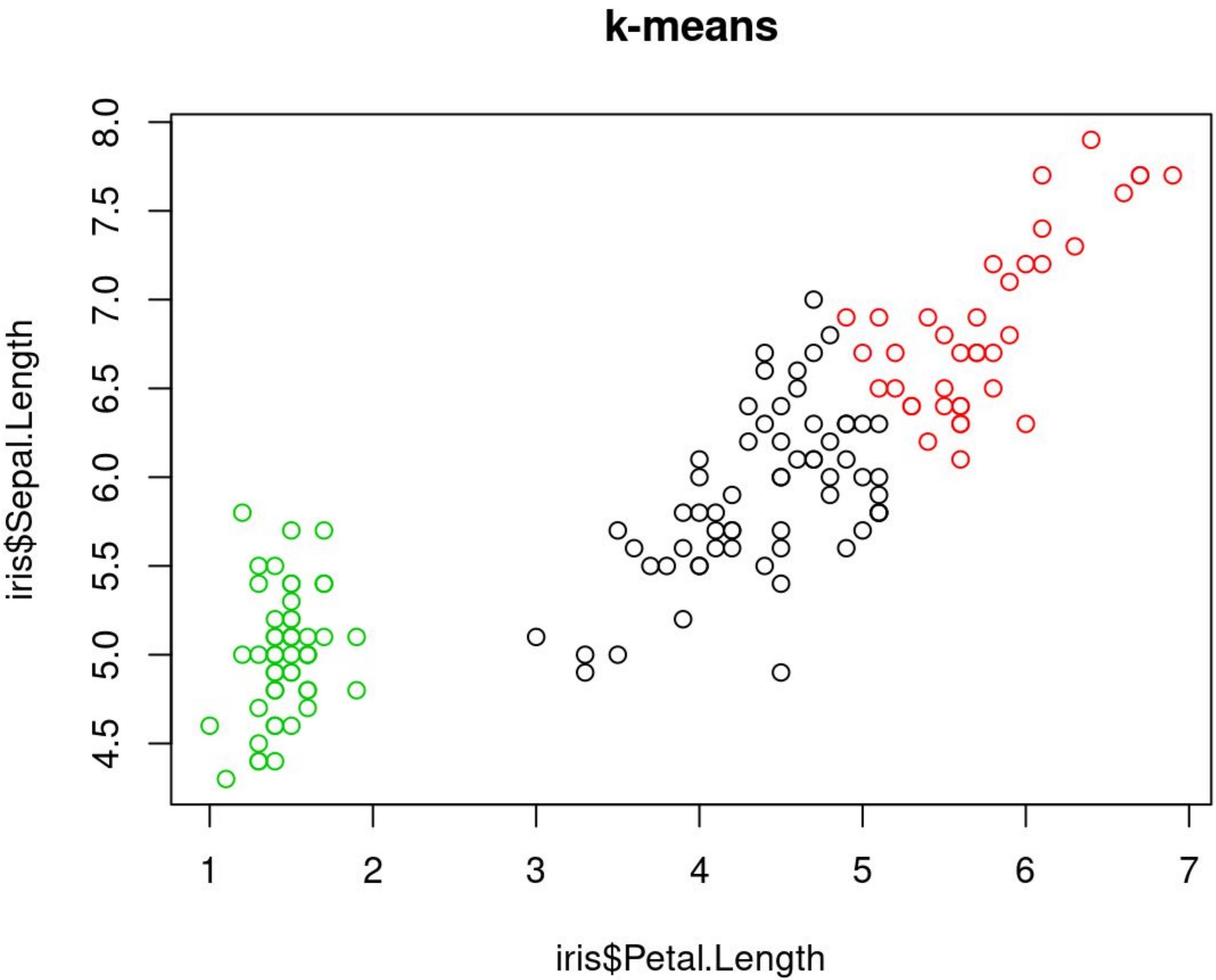
Hierarchical

Clusters are related in a parent-child graph (classes+subclasses)

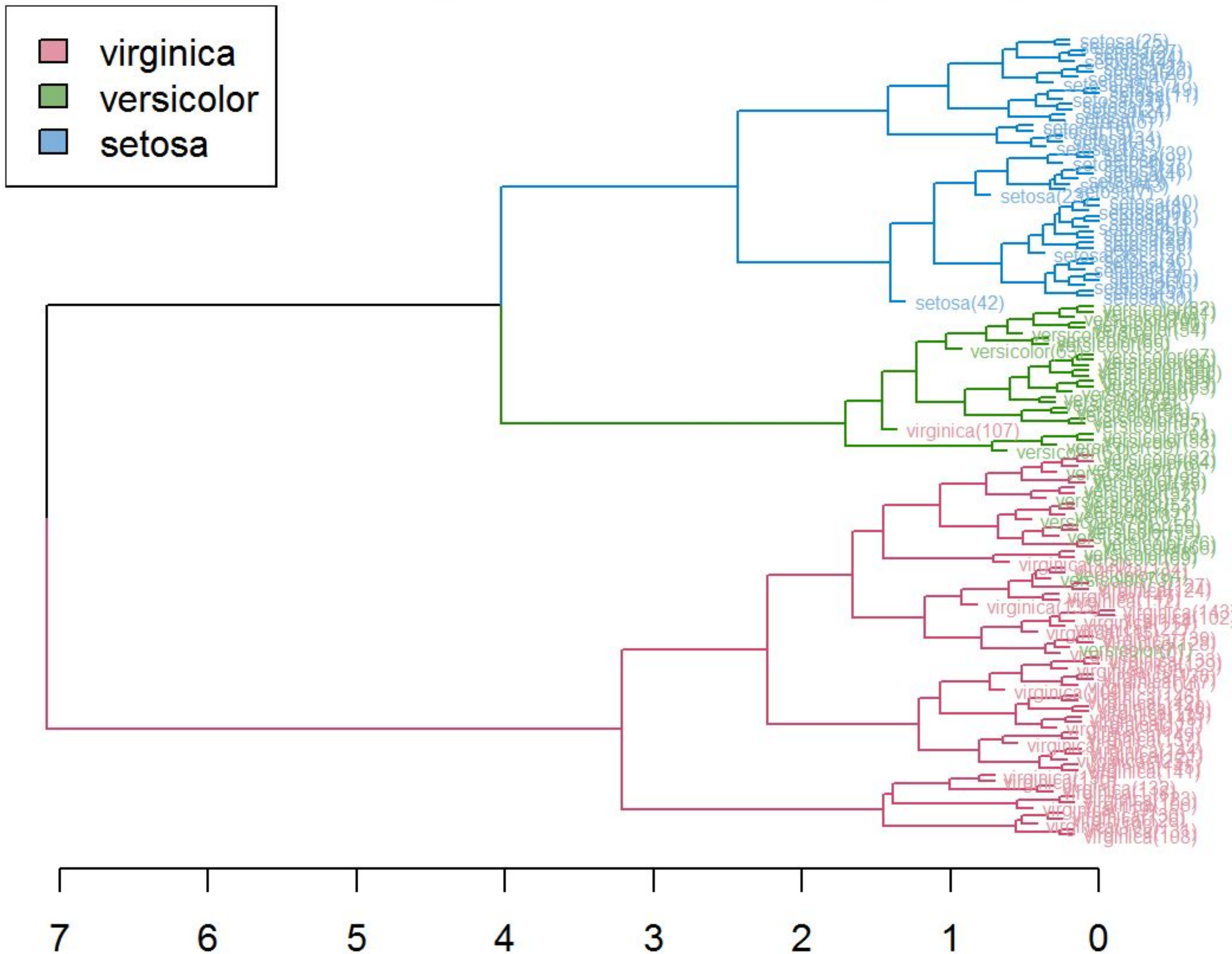
Terminal nodes (leaves) stand for clustered objects

E.g. **Hierarchical Agglomerative Clustering**

→ Intuitive


k-means
iris$Sepal.Length
8.0
7.5
7.0
6.5
6.0
5.5
5.0
4.5
1
2
3
4
5
6
7
iris$Petal.Length

Hierarchical



Comparison

Hierarchical

Better for **detailed data analysis** -- gives more info -- than flat

No single best algo (see diagram below)

Less efficient than flat

No objective way to know how many clusters result

Flat

Have to guess at number of clusters (k)

Better for **efficiency**

Conceptually simplest, so use first on new data

Meant for continuous data, so no good for nominal data

See K-Modes

Hierarchical Clustering Algorithms

Bottom-up vs Top-Down

Both iterative

Bottom-up

Begins with **one cluster for each doc**
Uses **similarity** to determine which clusters get merged each step
Also called **agglomerative —** HAC

Top-down

Begins with **one cluster for all docs** (one big cluster)
Uses **coherency** (max within-group similarity) to split clusters at each step
Also called **divisive**

Think of clusters as provisional **labels** (d, c)

Agglomerative Clustering Algorithms

Greedy – starts with a **separate cluster for each object**

In each step, the two most similar clusters are determined and then merged into a new cluster

Terminates when one large cluster containing all objects has been formed

Agglomerative Clustering Algorithms

1. Compute the **distance** between each document pair (as matrix)
2. Consider each individual document as its own **cluster**
3. Do:
    1. Find and **merge** two **closest**[1] clusters in the matrix
    2. **Update** the distance table
4. Repeat 3 **until** one single cluster remains, $|C| = 1$

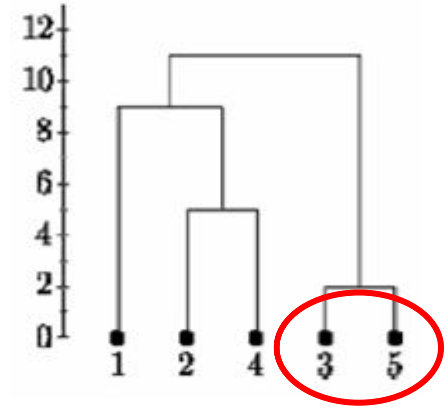
[1] Re 3.1 -- There are **many ways to measure the distance**, aka **linkage**, between two clusters

Different from the distance measure in 1

Linkage defines **what** is measured, not **how**

	1	2	3	4	5
1	0				
2	9	0			
3	3	7	0		
4	6	5	9	0	
5	11	10	2	8	0

	35	1	2	4
35	0			
1	11	0		
2	10	9	0	
4	9	6	5	0

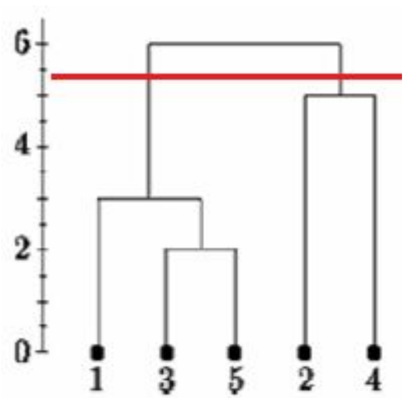


The first pair ...

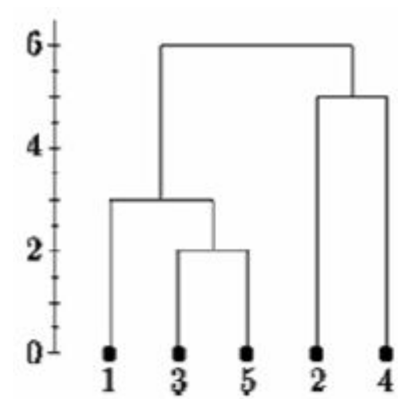
1 Start with **closest** pair, e.g. $d(3, 5) = 2$

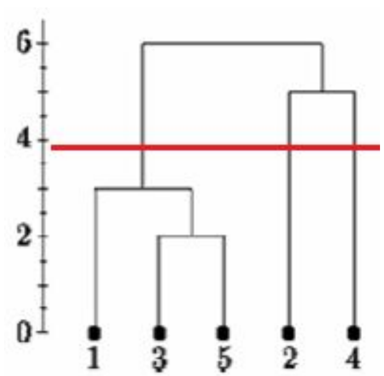
2 **Combine** pair into one (35), and pick **maximum** value from two in comparison to the rest, i.e. Complete Link

3 After 6 steps, everything is clustered



4 The same data closest by taking the **minimum** value of the group pair, i.e. Single Link





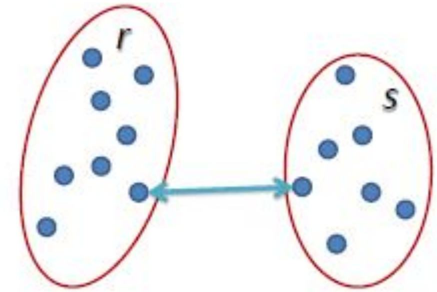
In either case, **no objective way to define groups**, i.e. pick a cut-off line

5

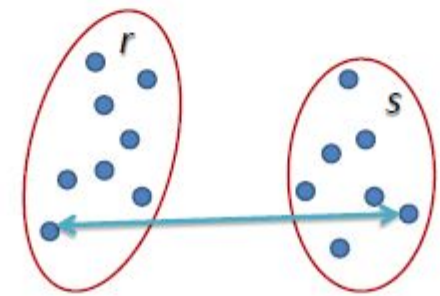
From https://online.stat.psu.edu/stat555/node/86/

Linkage Measures

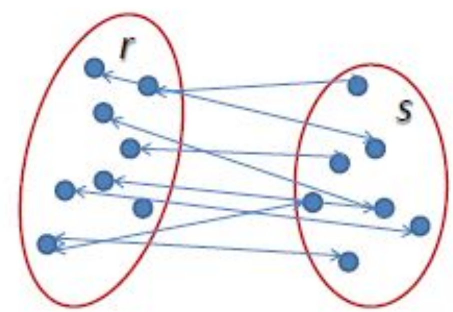
Single link distance: minimum distance between two points in each cluster (two most similar members)



Complete link distance: maximum distance between two points in each cluster (two least similar members)



Average link distance: average distance between each point in one cluster to every point in the other cluster



Centroid distance: distance between centroid two clusters

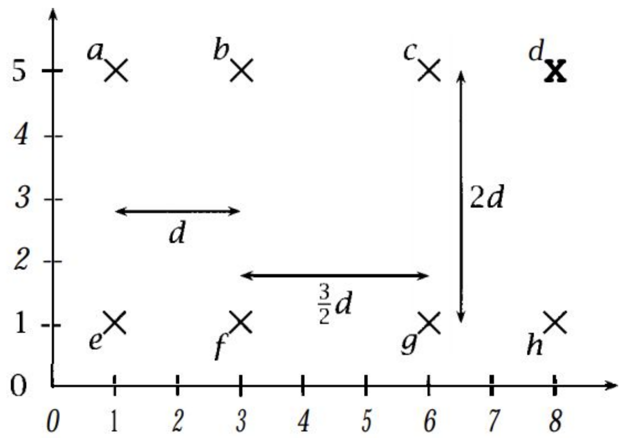


Figure 14.4 A cloud of points in a plane.

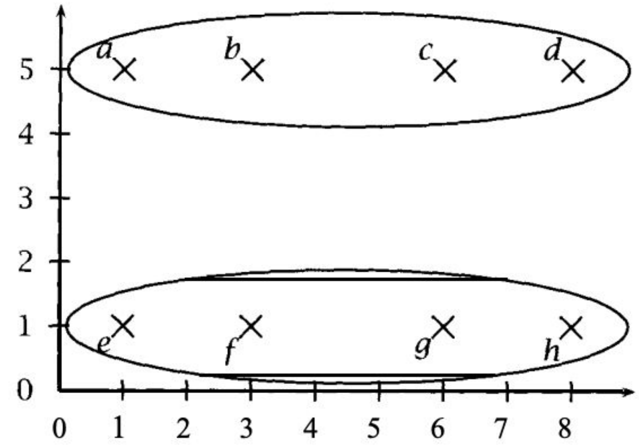


Figure 14.6 Single-link clustering of the points in figure 14.4.

Single

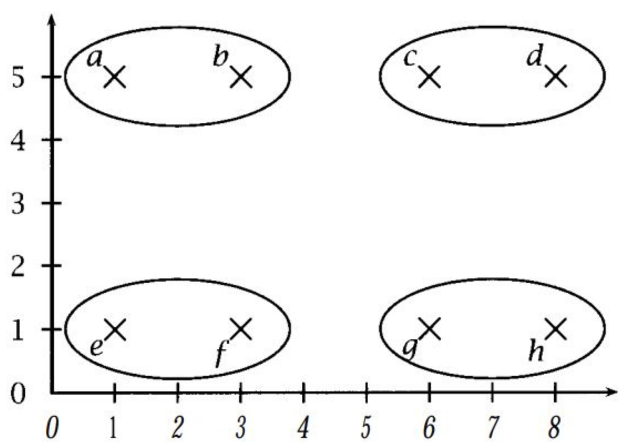


Figure 14.5 Intermediate clustering of the points in figure 14.4.

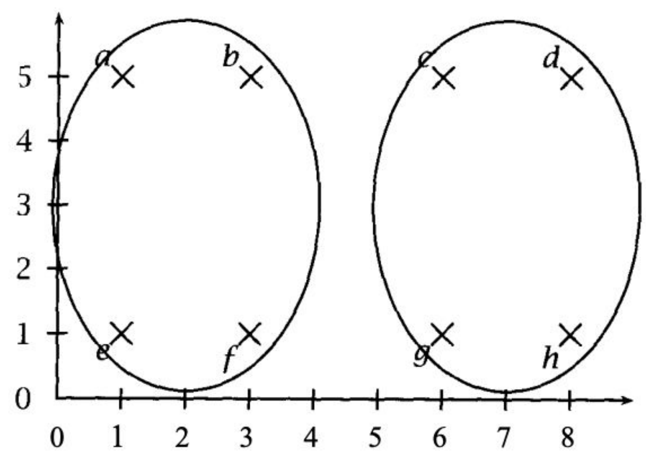


Figure 14.7 Complete-link clustering of the points in figure 14.4.

Complete

Single vs Complete link results

Linkage Measures

Single Link	Can handle non-elliptical shapes Good **local** coherence, but bad **global** quality i.e. produces long, elongated clusters (chaining). **Sensitive to outliers and noise.**
Complete Link	Focuses on global qualities of clusters Produces more balanced clusters (with equal diameter). **Less susceptible to noise. Often breaks very large clusters.** Small clusters are merged with large ones.
Group Average Link	Compromise. Less susceptible to noise and outliers.**Biased** towards globular clusters.

The most appropriate measure
depends on the underlying process

E.g. **volcano** chains modeled by single link

For **language** modeling,
complete link clustering is preferable

i.e. linkage methods
that are more **spherical**

E.g. **Ward** clustering

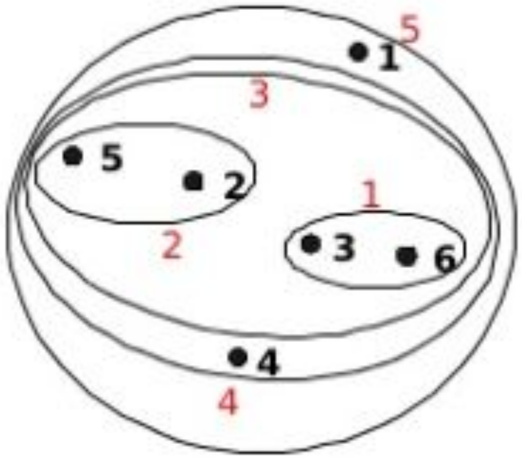
Ward Distance

The difference between the **total within-cluster sum of squares** (WCSS) for the two clusters **separately**, and the WCSS resulting from **merging** the two clusters

$$D_W(C_i, C_j) = \sum_{x \in C_i} (x - r_i)^2 + \sum_{x \in C_j} (x - r_j)^2 - \sum_{x \in C_{ij}} (x - r_{ij})^2$$

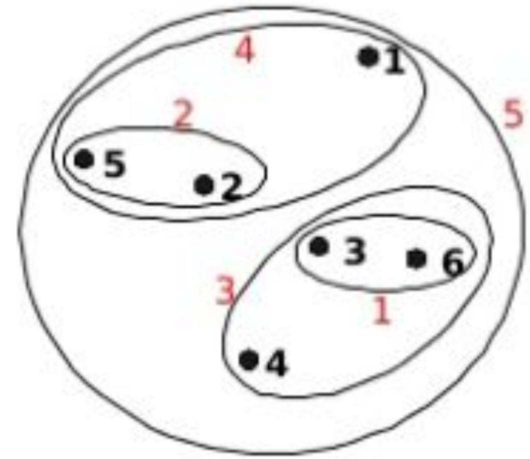
where r is the centroid of cluster C

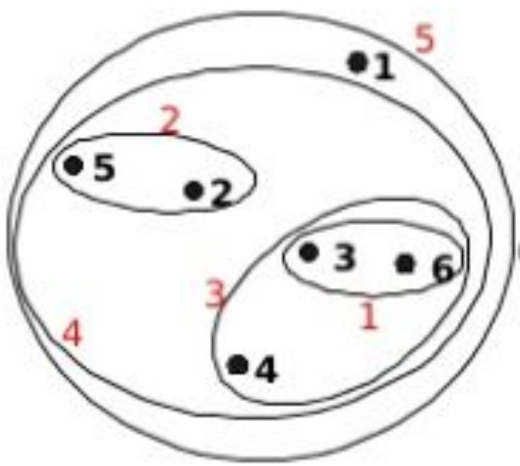
Favors **minimal increase of sum of squares**



Single

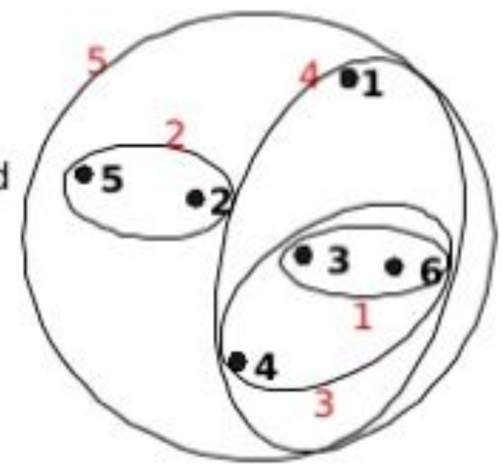
Complete

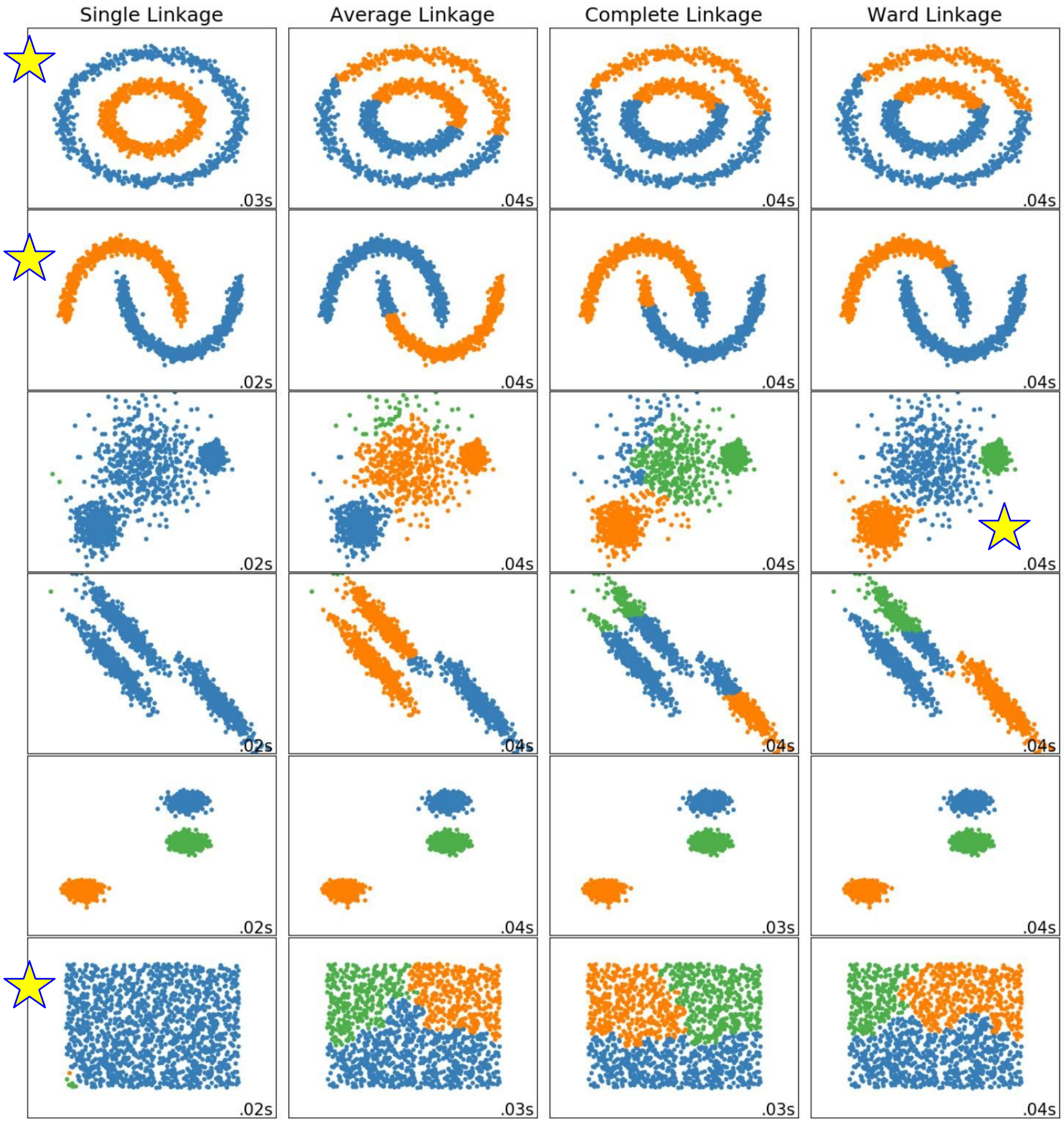




Group Average

Ward's Method





No single best algo

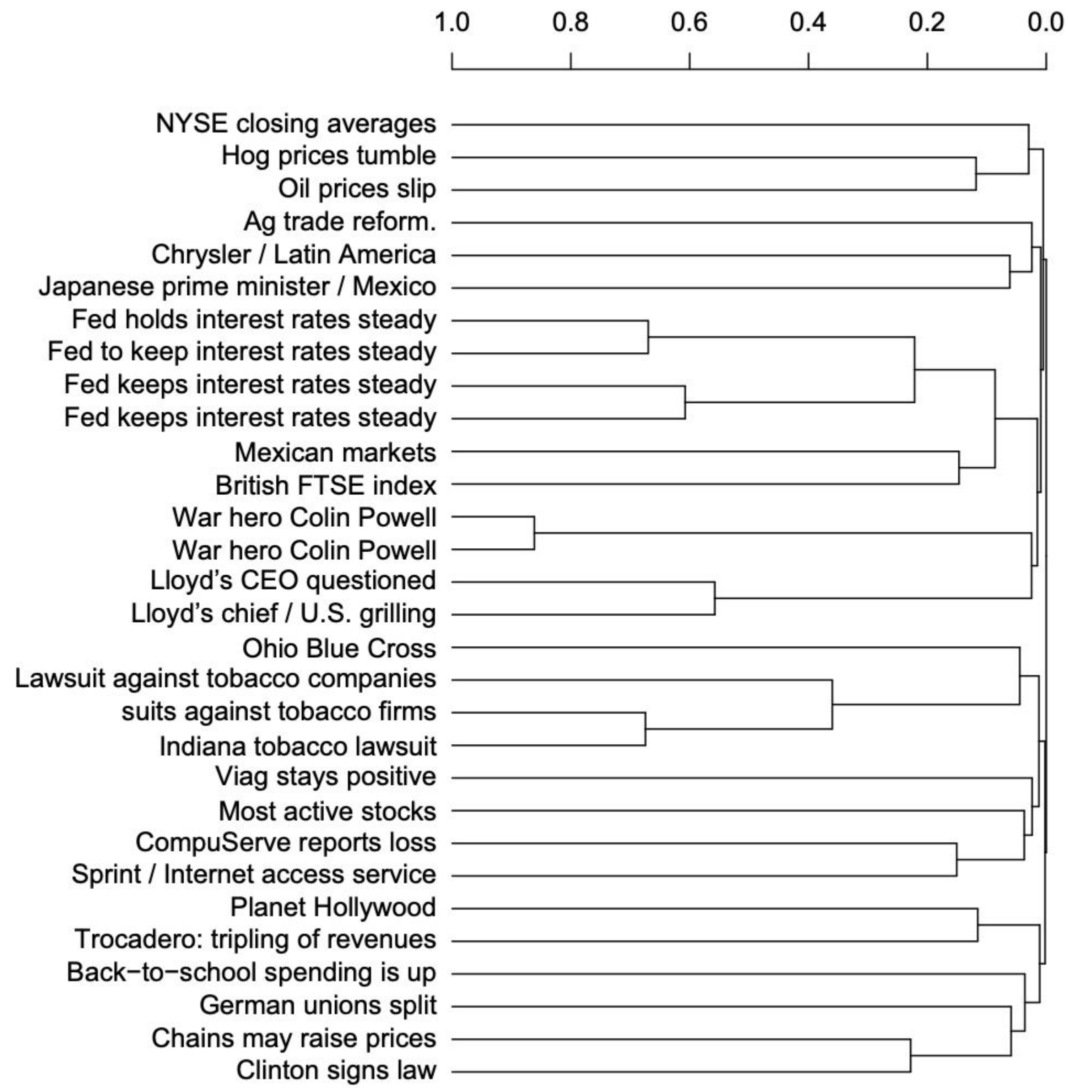
Depends on underlying process

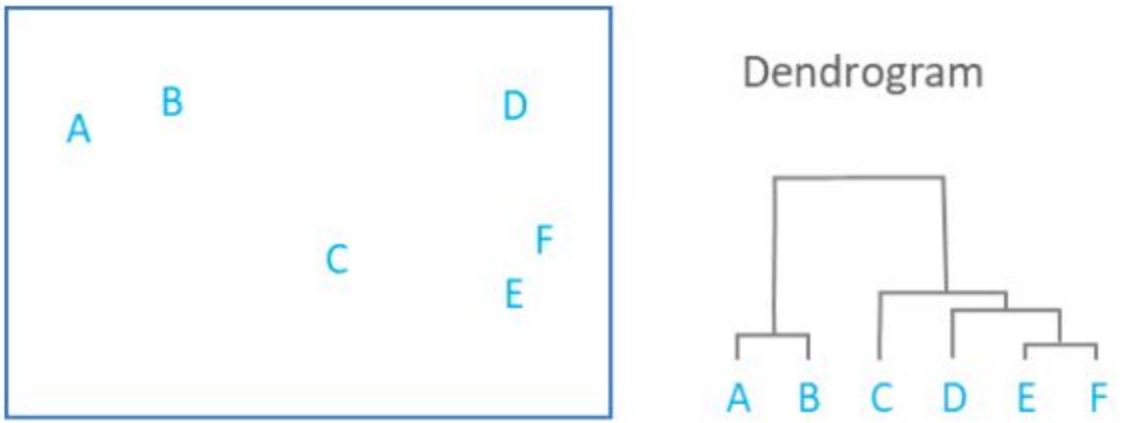
Dendrograms

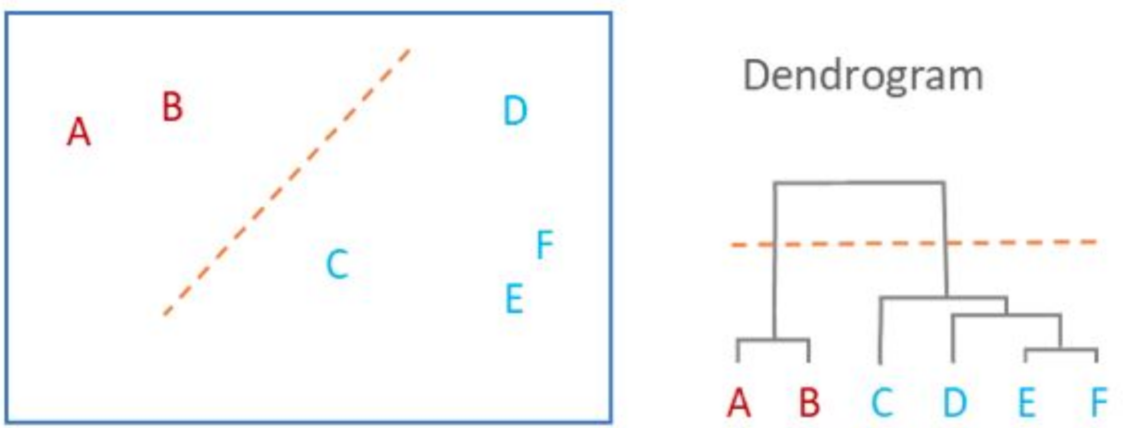
Hierarchical clusters are often represented in **dendrograms** like the one on the right

The diagram shows which documents are most similar by a given metric

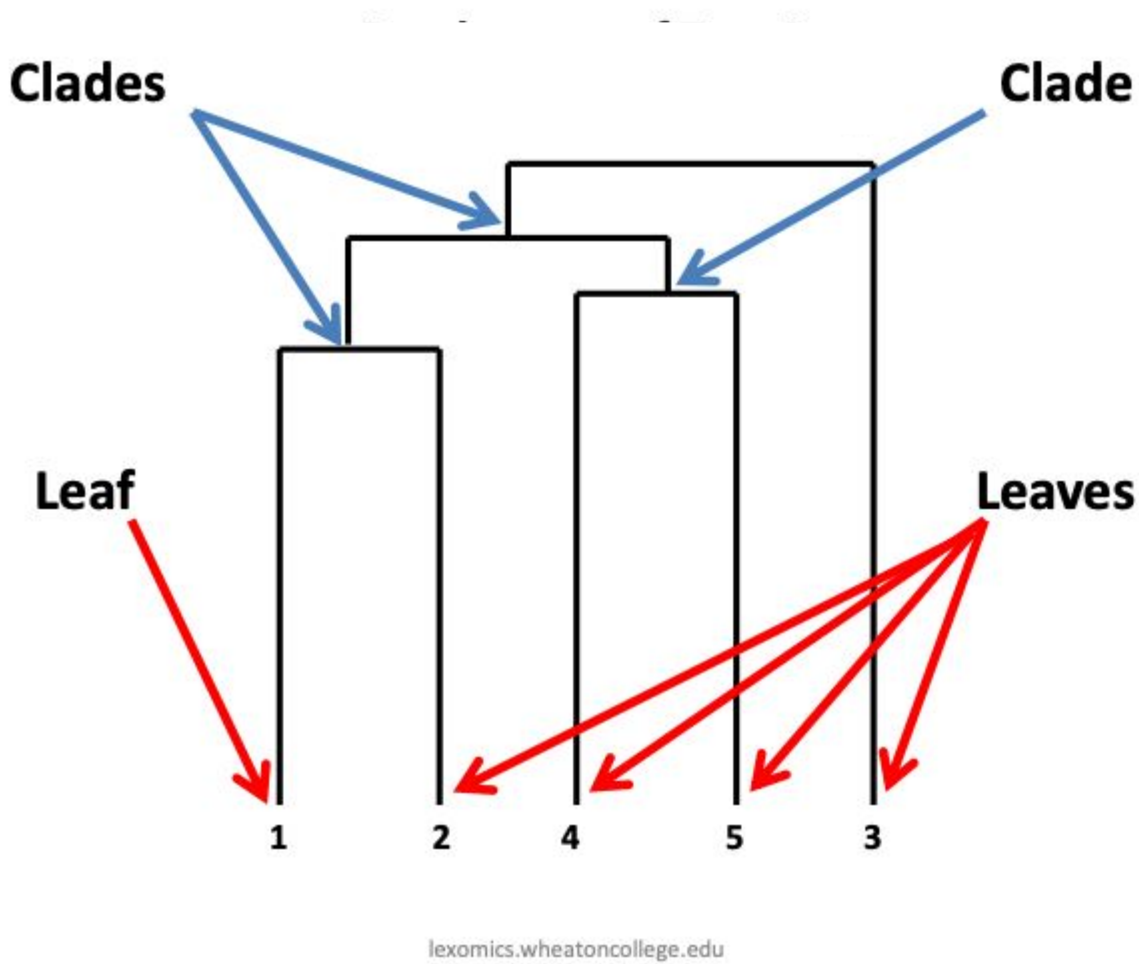
The **length** of the grouped branches denotes the **distance** between the grouped items







Here we see the relationship between point space and cluster diagrams



Here are some terms used to describe dendrograms

Clustering and ETA

Hierarchical **Agglomerative** Clustering (HAC) works well for many ETA tasks

E.g. Showing distances among novels from an author or set of authors

The **Ward** methods also works well for linkage

Requires Euclidean distance measures (incl. Cosine)

We also reduce the **dimensionality** of the document-term matrix

e.g. DF-IDF to yeild ~4,000 significant terms

In addition, for display purposes, we want a relatively **small number of observations**

We often take an **aggregate** by some label or container

E.g. group documents by author, year, book, etc.

Tools for Computing Clusters

Python offers at least two libraries to compute clusters

SciPy

`scipy.cluster`

https://docs.scipy.org/doc/scipy-1.2.1/reference/cluster.html

`scipy.cluster.hierarchy.dendrogram`
To generate dendrogram images; used by SciKit Learn

SciKit Learn

`sklearn.cluster`

https://scikit-learn.org/stable/modules/classes.html#module-sklearn.cluster

Clustering Wine Reviews

The Corpus

129,971 wine reviews from *Wine Enthusiast*

- Pre-scraped and downloaded from **Kaggle**

Each review is **very short** – one or two sentences, e.g.

> "A year in wood and 30 months in bottle before release have allowed this attractive wine to fully develop its solid yet smooth texture. It has concentration, layers of bright black currant fruit and vibrant acidity. Ready to drink."
>
> > Review of *Adega Cooperativa de Borba*, 2012, Montes Claros Garrafeira Red (Alentejo) by Roger Voss.

The Corpus – Salient Features

Title – unique identifier

Taster – 19, with Twitter names

Variety – 707, includes blends

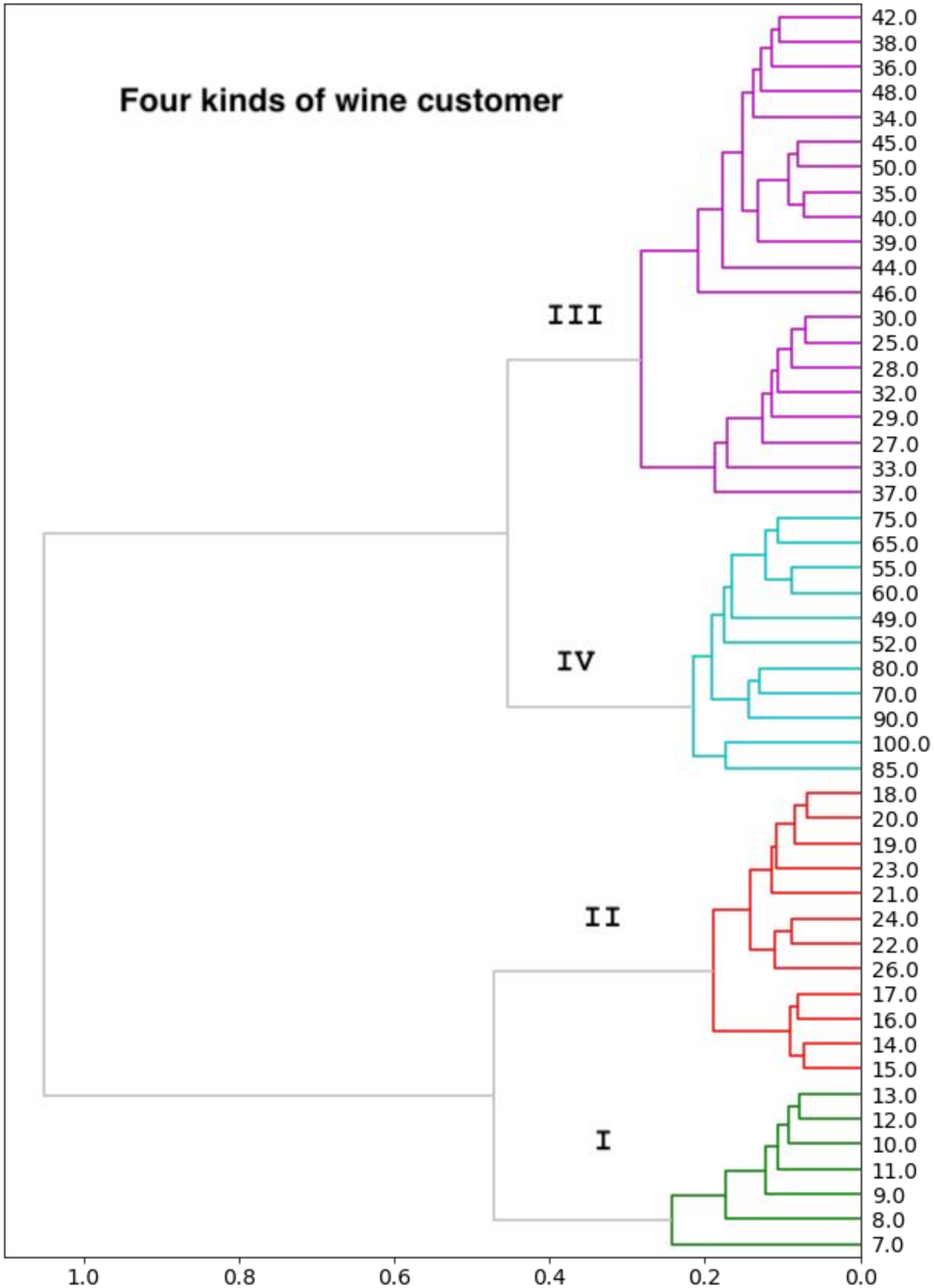
Country – 43

Province – 425

Points – a rating from 80 to 100 (integers); Mean = 88

Price – from $4 to $3,300; Median = $25

country	description	designation	points	price	province
Italy	Aromas include tropical fruit, broom, b	Vulk√† Bianco	87		Sicily & Sardi
Portugal	This is ripe and fruity, a wine that is sm	Avidagos	87	15	Douro
US	Tart and snappy, the flavors of lime flesh and rind domina		87	14	Oregon
US	Pineapple rind, lemon pith and orange	Reserve Late Harv	87	13	Michigan
US	Much like the regular bottling from 201	Vintner's Reserve \	87	65	Oregon
Spain	Blackberry and raspberry aromas sho	Ars In Vitro	87	15	Northern Spa
Italy	Here's a bright, informal red that open:	Belsito	87	16	Sicily & Sardi
France	This dry and restrained wine offers spice in profusion. Bal:		87	24	Alsace
Germany	Savory dried thyme notes accent sunn	Shine	87	12	Rheinhessen
France	This has great depth of flavor with its fr	Les Natures	87	27	Alsace
US	Soft, supple plum envelopes an oaky :	Mountain Cuv√©e	87	19	California
France	This is a dry wine, very spicy, with a tight, taut texture and		87	30	Alsace
US	Slightly reduced, this wine offers a chalky, tannic backbon		87	34	California
Italy	This is dominated by oak and oak-driv	Rosso	87		Sicily & Sardi
US	Building on 150 years and six generations of winemaking		87	12	California
Germany	Zesty orange peels and apple notes a	Devon	87	24	Mosel
Argentina	Baked plum, molasses, balsamic vine	Felix	87	30	Other
Argentina	Raw black-cherry aromas are direct ar	Winemaker Selecti	87	13	Mendoza Pro
Spain	Desiccated blackberry, leather, charre	Vendimia Seleccio	87	28	Northern Spa
US	Red fruit aromas pervade on the nose, with cigar box and		87	32	Virginia
US	Ripe aromas of dark berries mingle wi	Vin de Maison	87	23	Virginia
US	A sleek mix of tart berry, stem and herb, along with a hint c		87	20	Oregon
Italy	Delicate aromas recall white flower an	Ficiligno	87	19	Sicily & Sardi
US	This wine from the Geneseo district off	Signature Selectio	87	22	California
Italy	Aromas of prune, blackcurrant, toast a	Aynat	87	35	Sicily & Sardi
US	Oak and earth intermingle around robu	King Ridge Vineya	87	69	California
Italy	Pretty aromas of yellow flower and sto	Dalila	87	13	Sicily & Sardi
Italy	Aromas recall ripe dark berry, toast and a whiff of cake spi		87	10	Sicily & Sardi
Italy	Aromas suggest mature berry, scorche	Mascaria Barricato	87	17	Sicily & Sardi



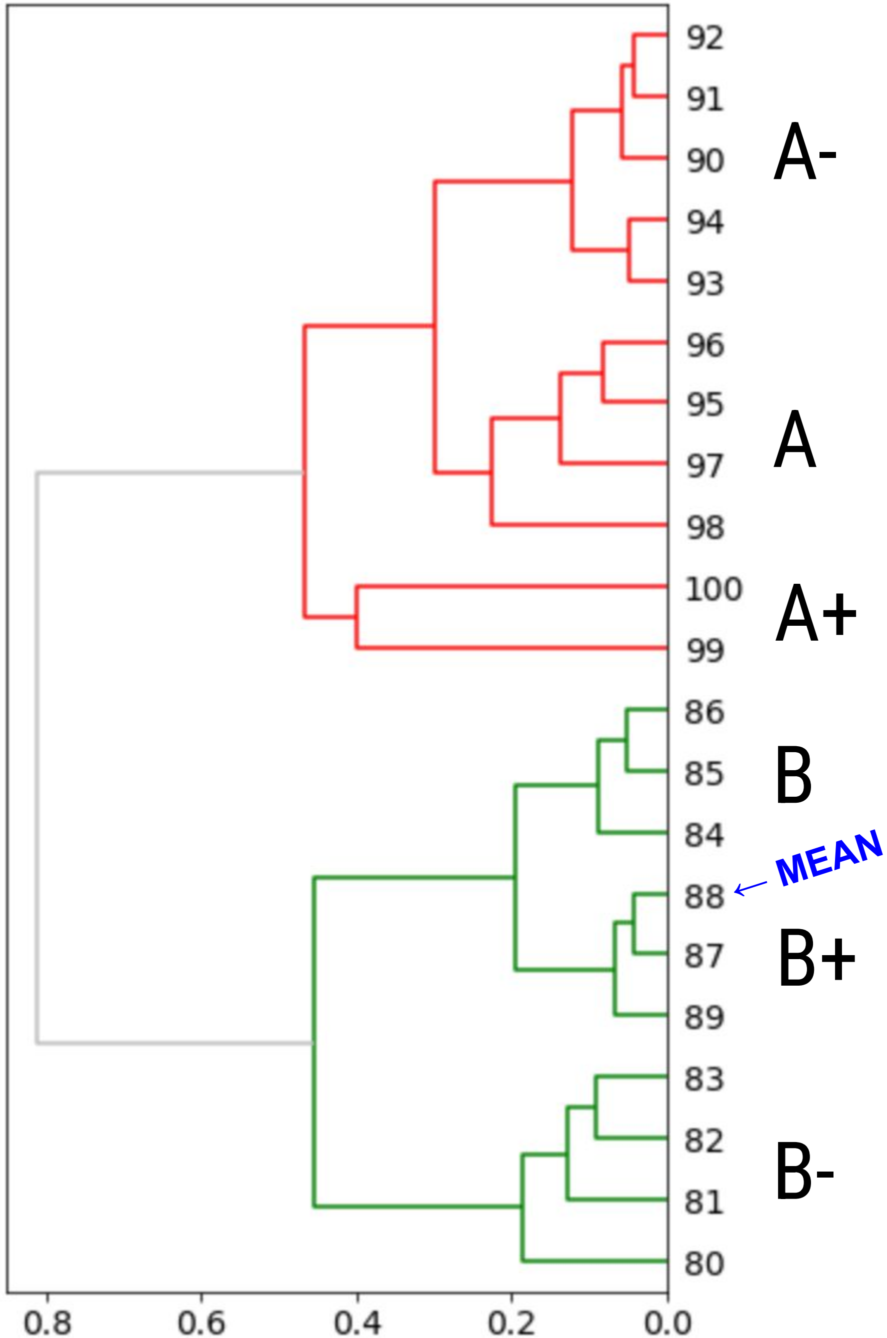
The top 50 prices ($7 to $100)

Reviews appear to chunk prices into **four main groups** – note the distance between the clusters.

I	7 — 13	(~6.25)
II	14 — 26	(~12.5)
III	27 — 50	(~25)
IV	52 — 100	(~50)

Each group doubles the range of the previous.

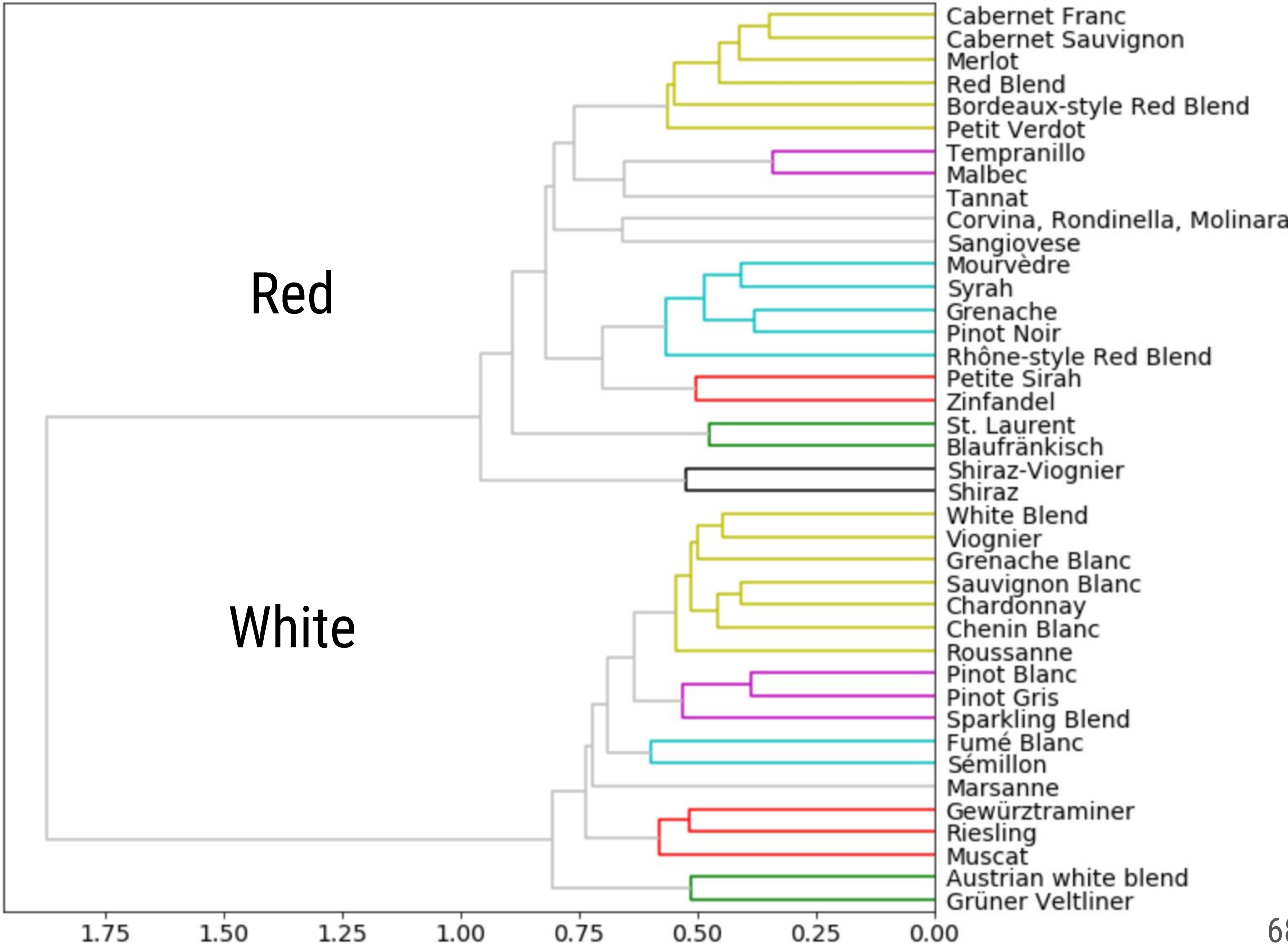
Note that some prices are out of group – e.g. 25 is in group III not II

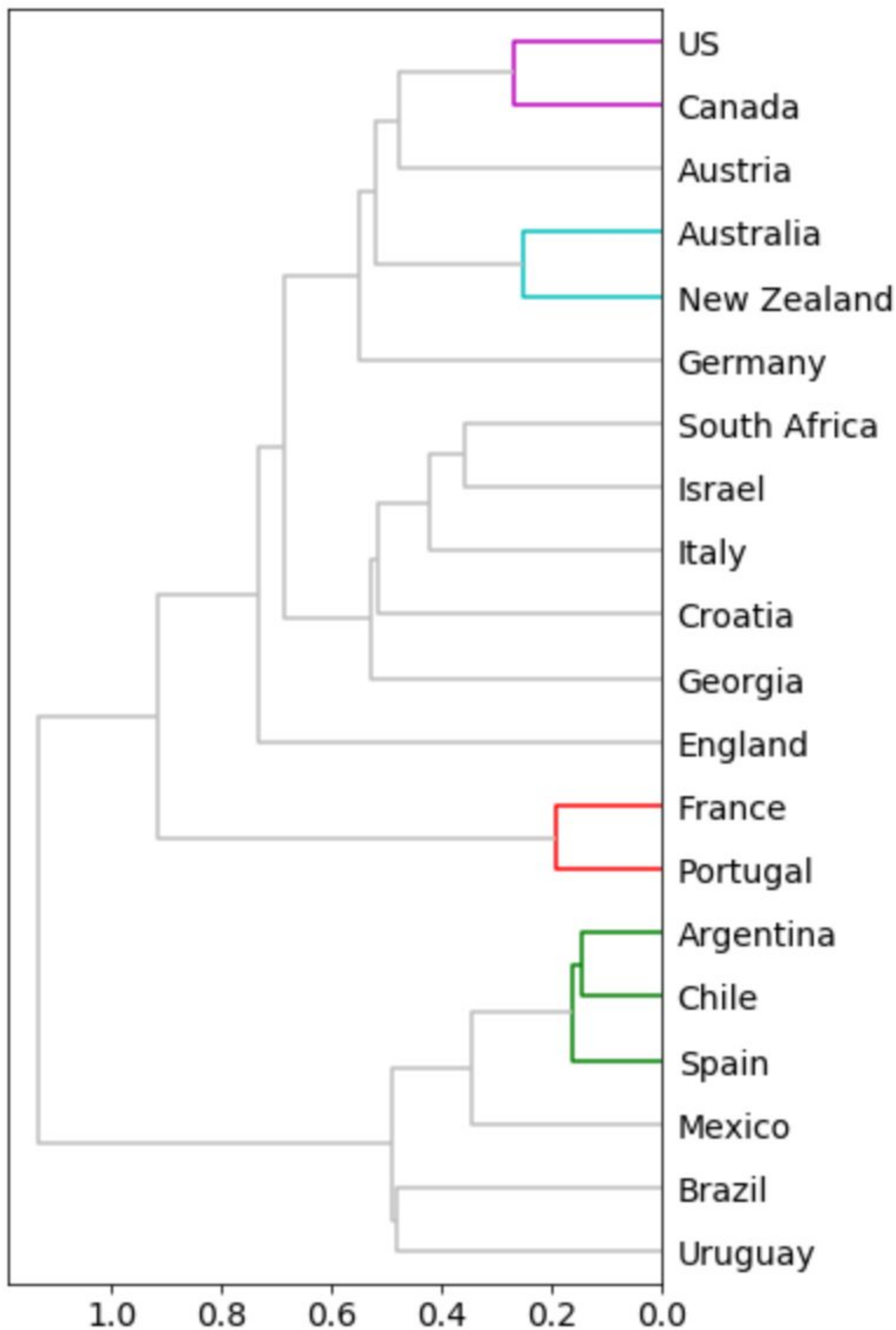


Grades appear to fall into clusters as well, matching a mental model that reflects special treatment for certain numbers and ranges.

Note that B- and A+ are on their own, and the both B and B+ and A- and A are grouped.

This suggests a bias to distinguish the very best and from very worst (or lowest).


Red
White
Cabernet Franc
Cabernet Sauvignon
Merlot
Red Blend
Bordeaux-style Red Blend
Petit Verdot
Tempranillo
Malbec
Tannat
Corvina, Rondinella, Molinara
Sangiovese
Mourvèdre
Syrah
Grenache
Pinot Noir
Rhône-style Red Blend
Petite Sirah
Zinfandel
St. Laurent
Blaufränkisch
Shiraz-Viognier
Shiraz
White Blend
Viognier
Grenache Blanc
Sauvignon Blanc
Chardonnay
Chenin Blanc
Roussanne
Pinot Blanc
Pinot Gris
Sparkling Blend
Fumé Blanc
Sémillon
Marsanne
Gewürztraminer
Riesling
Muscat
Austrian white blend
Grüner Veltliner
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00

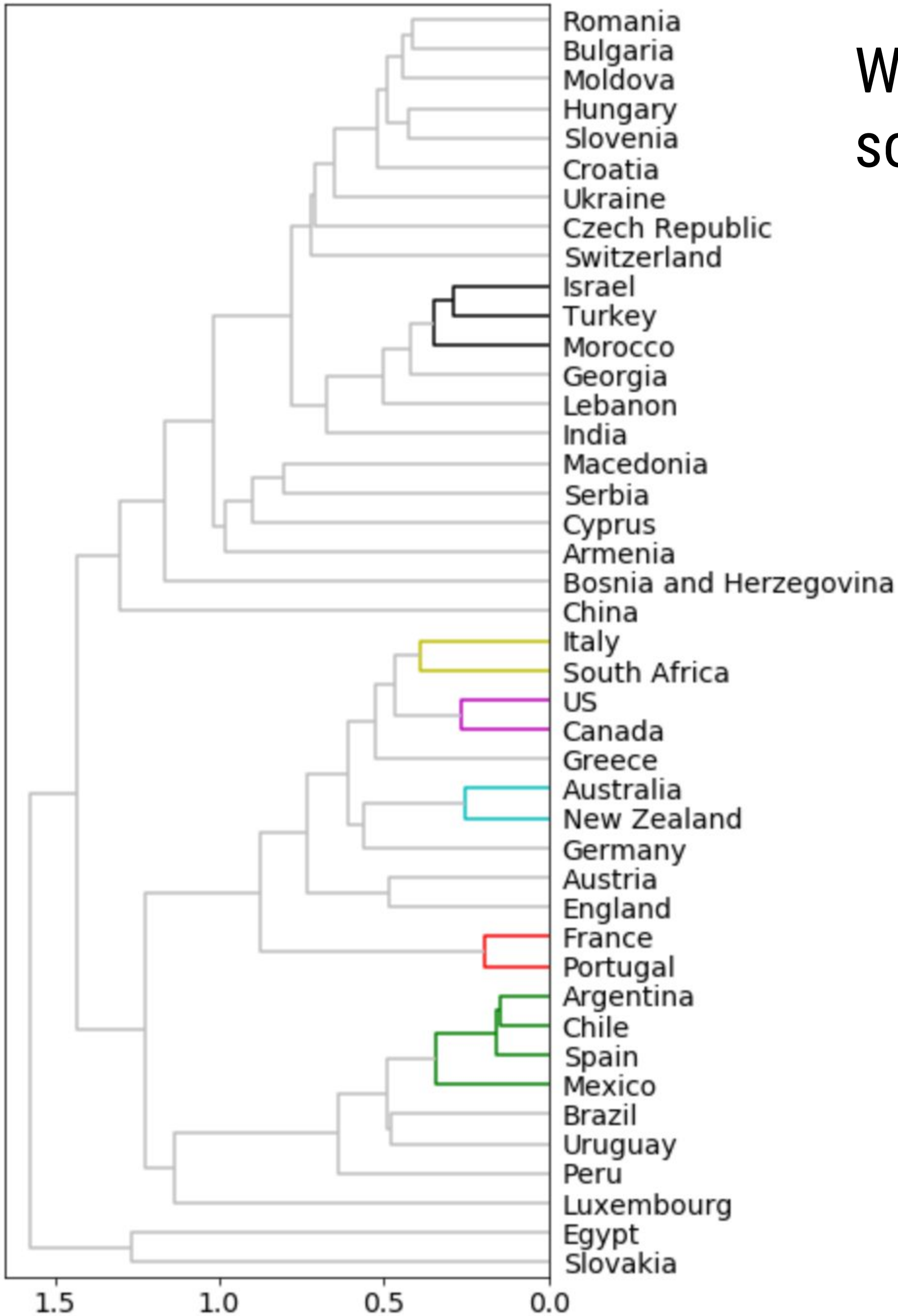


Clustering of top 20 countries by number of reviews

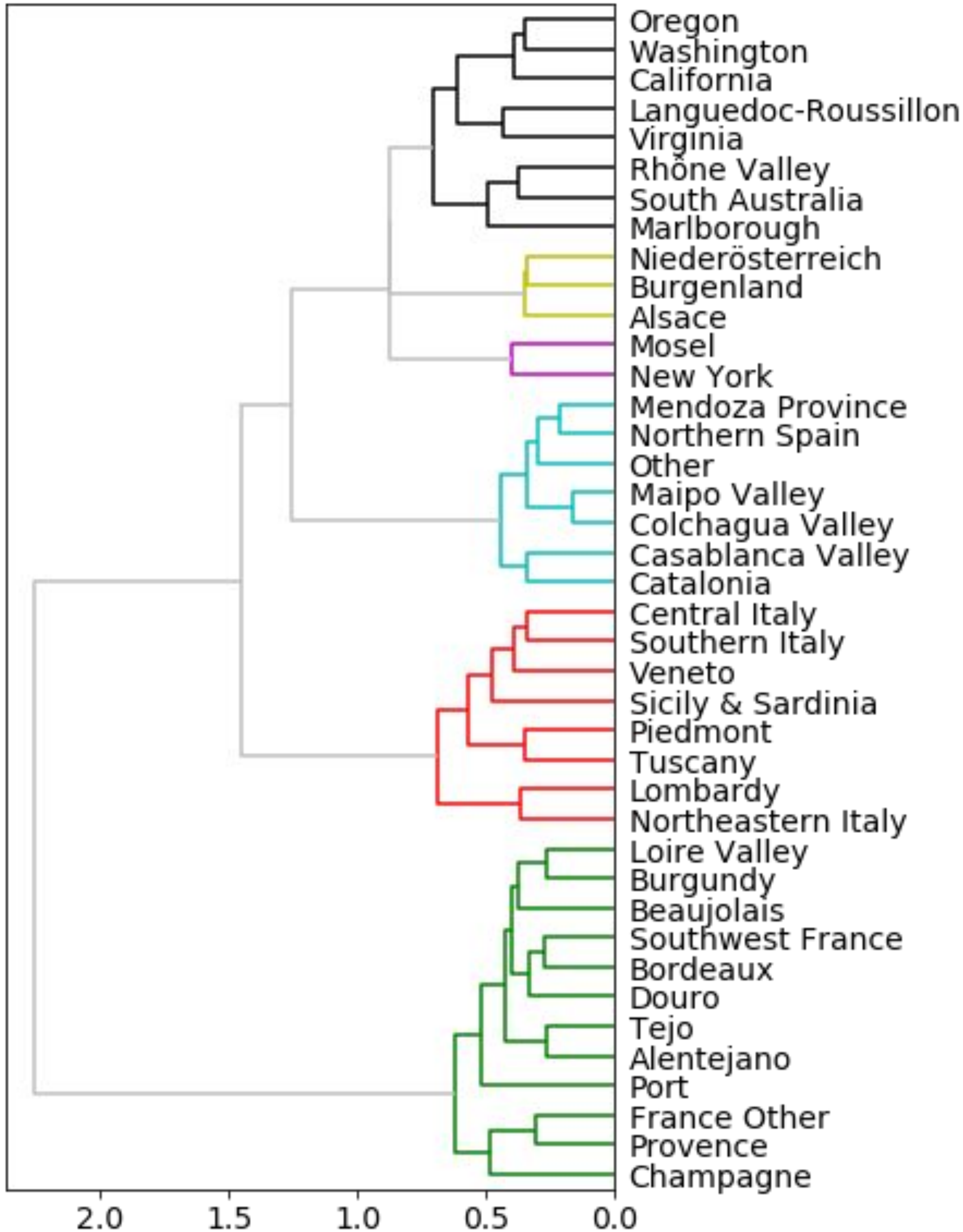
France is always grouped with **Portugal** (Why?)

Spain, Argentina, and **Chile** are always grouped

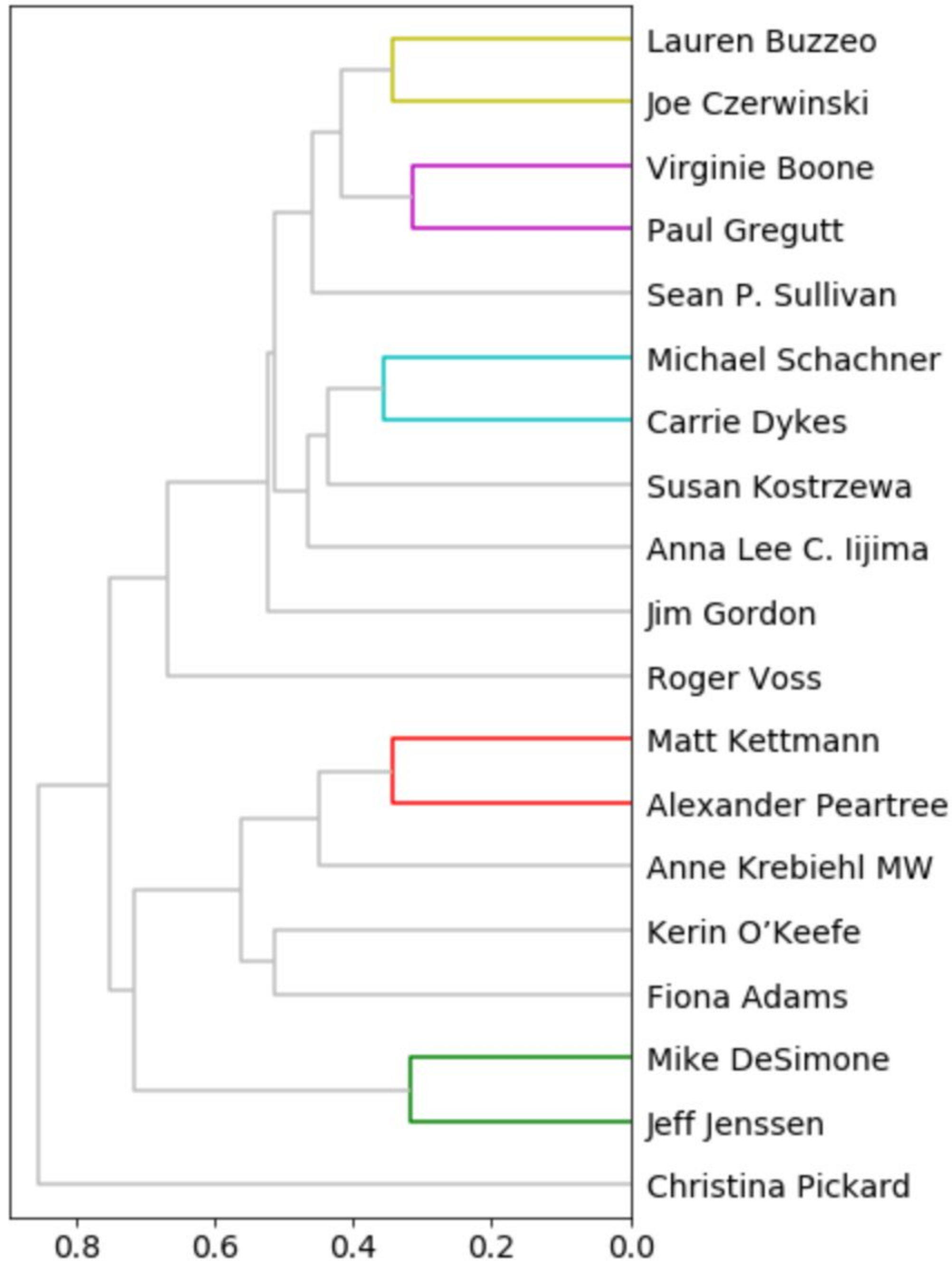
The **US** is always closer to **Italy** than to France



With more countries to the source data ...



Provinces group as expected



Not sure what to make of this

May be due to shared topic, but the first two don't overlap

Maybe stylistic differences