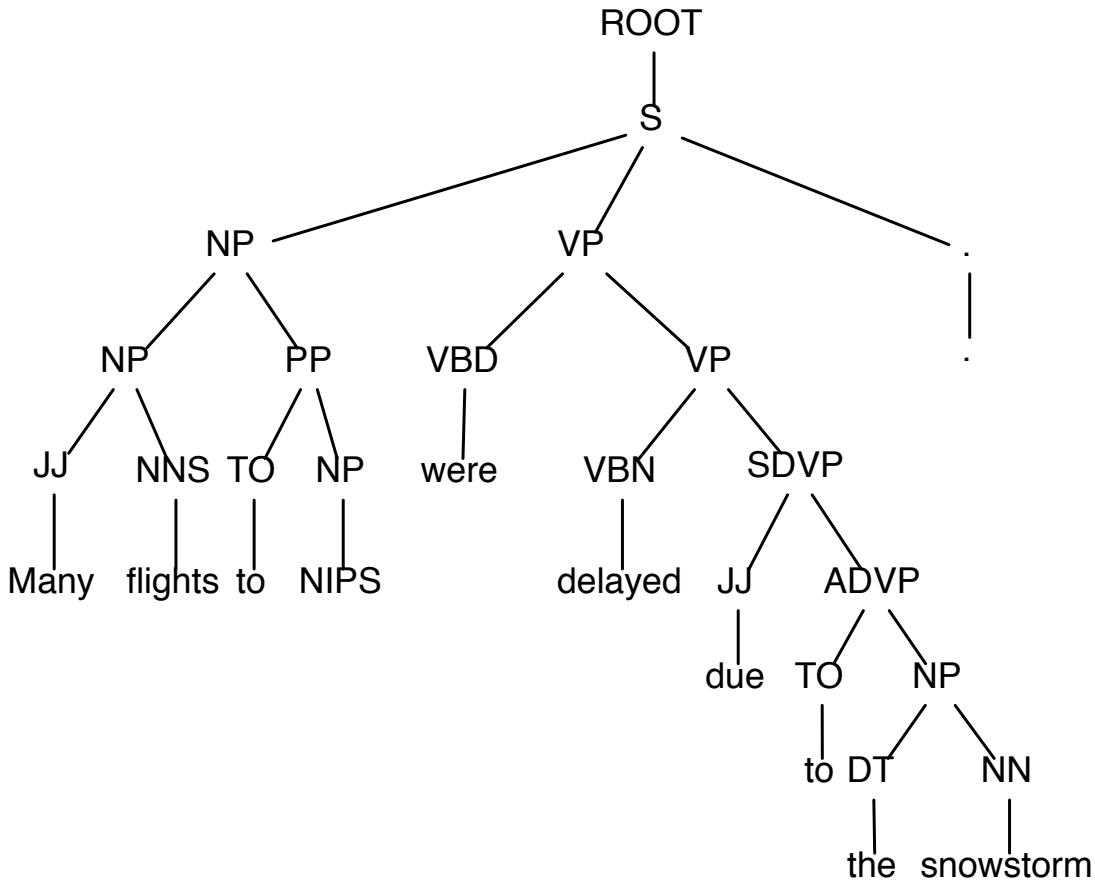
Hierarchical Clustering

Yee Whye Teh
Gatsby Computational Neuroscience Unit
UCL

January 23, 2008
Sheffield EPSRC Winter School

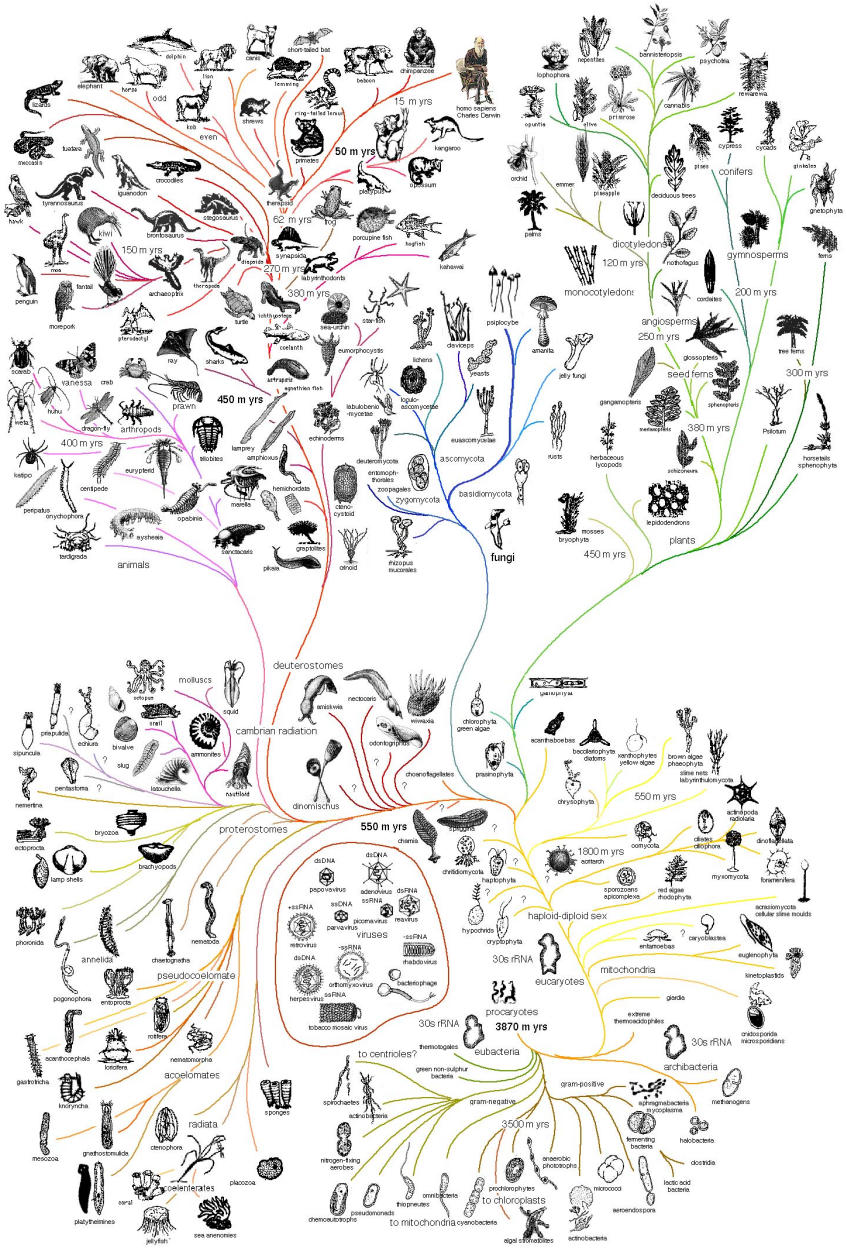
Hierarchical Representations

Many types of data have a hierarchical, tree-structured nature.

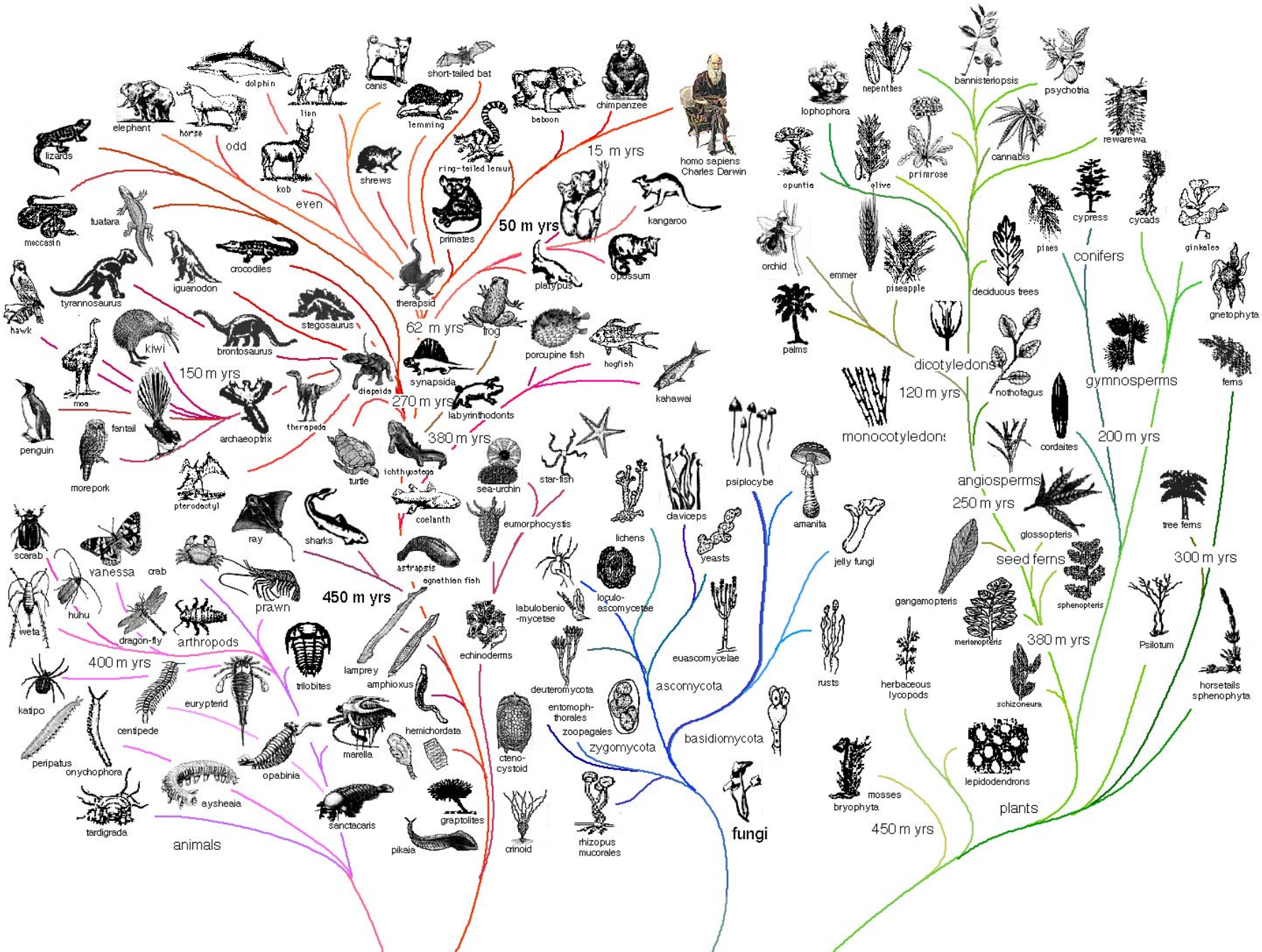


Hierarchical Representations

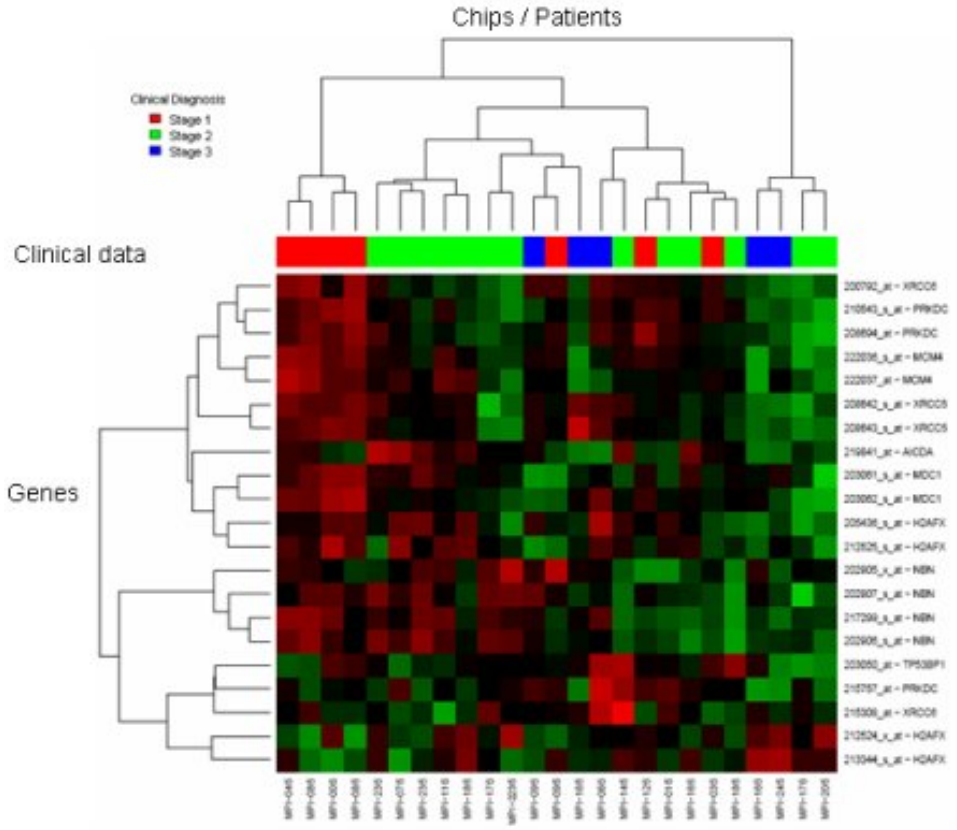
Hierarchical Representations



Hierarchical Representations

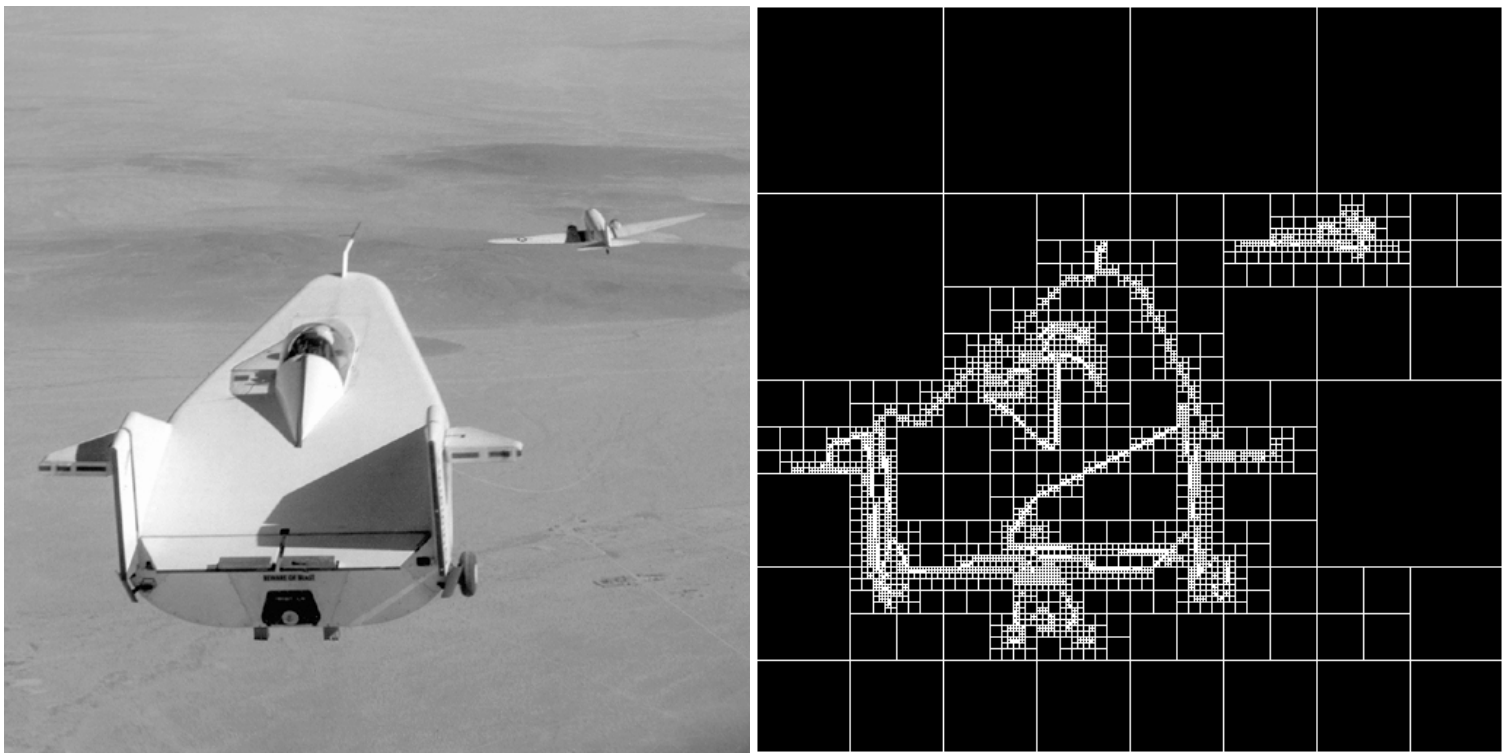


Hierarchical Representations

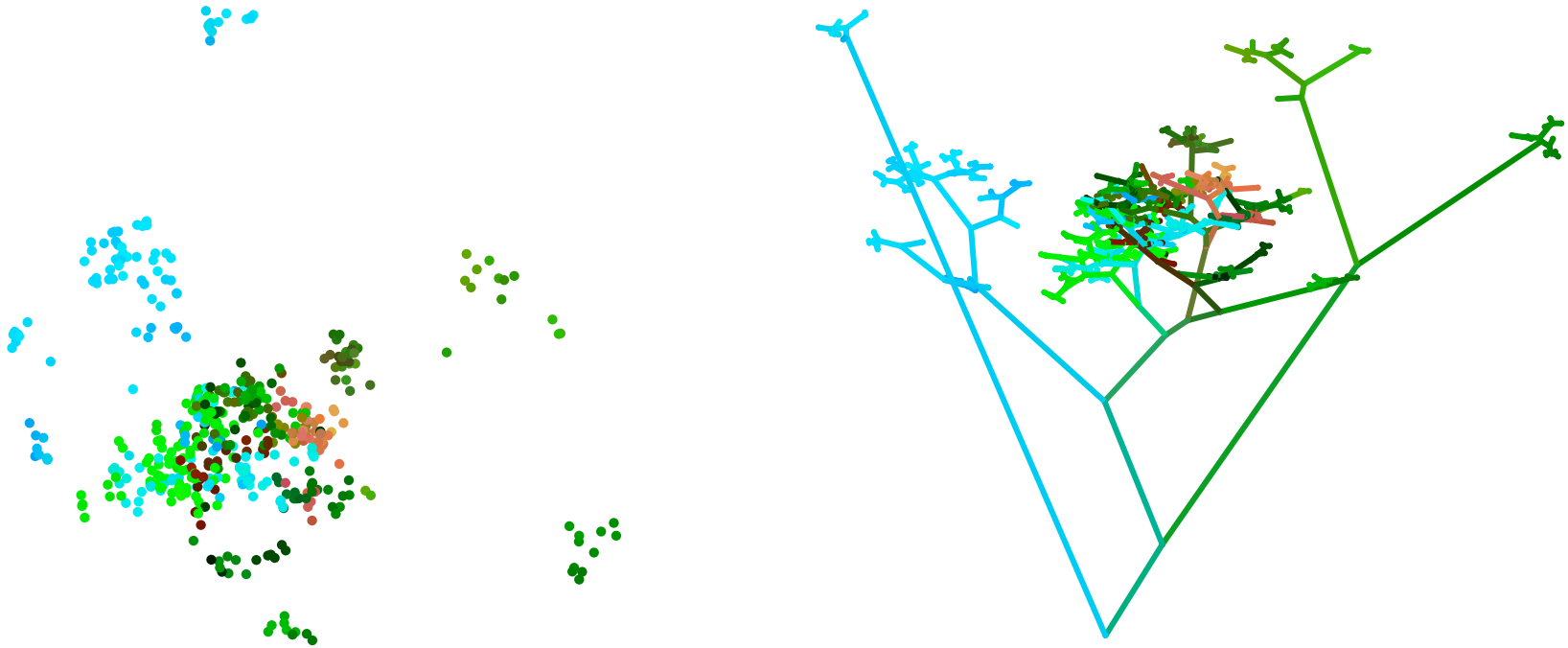


Hierarchical Representations

Multi-scale representations in signal and image processing, e.g. quadtrees, wavelet decompositions.



Hierarchical Clustering

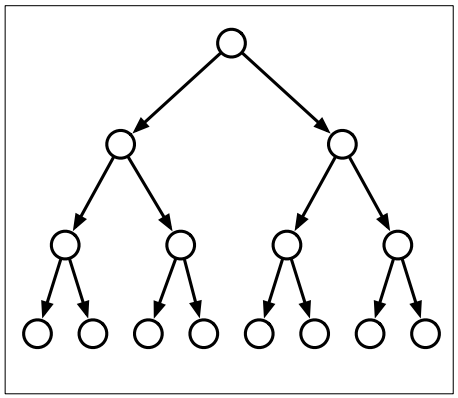


- Clusters.
- Substructure in clusters $\Rightarrow$ subclusters $\Rightarrow$ hierarchical clustering.

Hierarchical Clustering

Uses of hierarchical clustering:

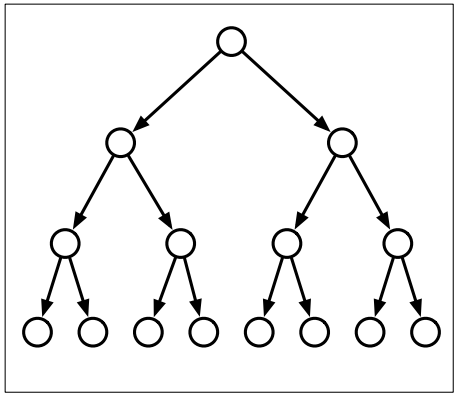
- Visualize data.
  - Understand relationships among data items.
- Summarize data.
  - Use hierarchical clustering as a way to partition data into different (unrelated) groups.
  - Note: we don't believe the data is hierarchical at all.
- Recover underlying structure.
  - We believe that our data has an underlying tree structure, and want to recover it.



Hierarchical Clustering

Different approaches:

- Top-down decimative approach.
  - Start with one big cluster.
  - Recursively split each cluster (if advantageous).
- Bottom-up agglomerative approach.
  - Start with one cluster per data point.
  - Iteratively find two clusters to merge (if advantageous).
  - Clusters found by finding pairs with maximum similarity.
- The dominant approach is bottom-up: better search landscape, more flexible algorithms.



Hierarchical Clustering

Another dimension to different approaches:

- Linkage algorithms
  - Single, average, complete etc linkage.
- Probabilistic models
  - PCluster, Bayesian Hierarchical Clustering.
- More Bayesian approaches
  - Dirichlet diffusion trees, Coalescents.

Linkage Algorithms

- Input: data $x_1, \ldots, x_n$.
- Input: distance measure $d(x, y)$.
- Input: distance combination:

$$d(C, D) = f(d(x, y) : x \in C, y \in D)$$

- Initialize each data point in separate cluster:

$$C_i = \{x_i\} \text{ for } i = 1, \ldots, n$$

- For $t = 1, \ldots, n-1$:
  - Find cluster pair:

$$C, D \leftarrow \operatorname*{argmin}_{C \neq D} d(C, D)$$

  - Merge C and D: Remove C and D, add $C \cup D$.

[Duda & Hart 1973]

Linkage Algorithms: Similarity Choices

- Single (or minimum) linkage:

$$d(C, D) = \min_{x \in C, y \in D} d(x, y)$$

- Complete (or maximum) linkage:

$$d(C, D) = \max_{x \in C, y \in D} d(x, y)$$

- Average linkage:

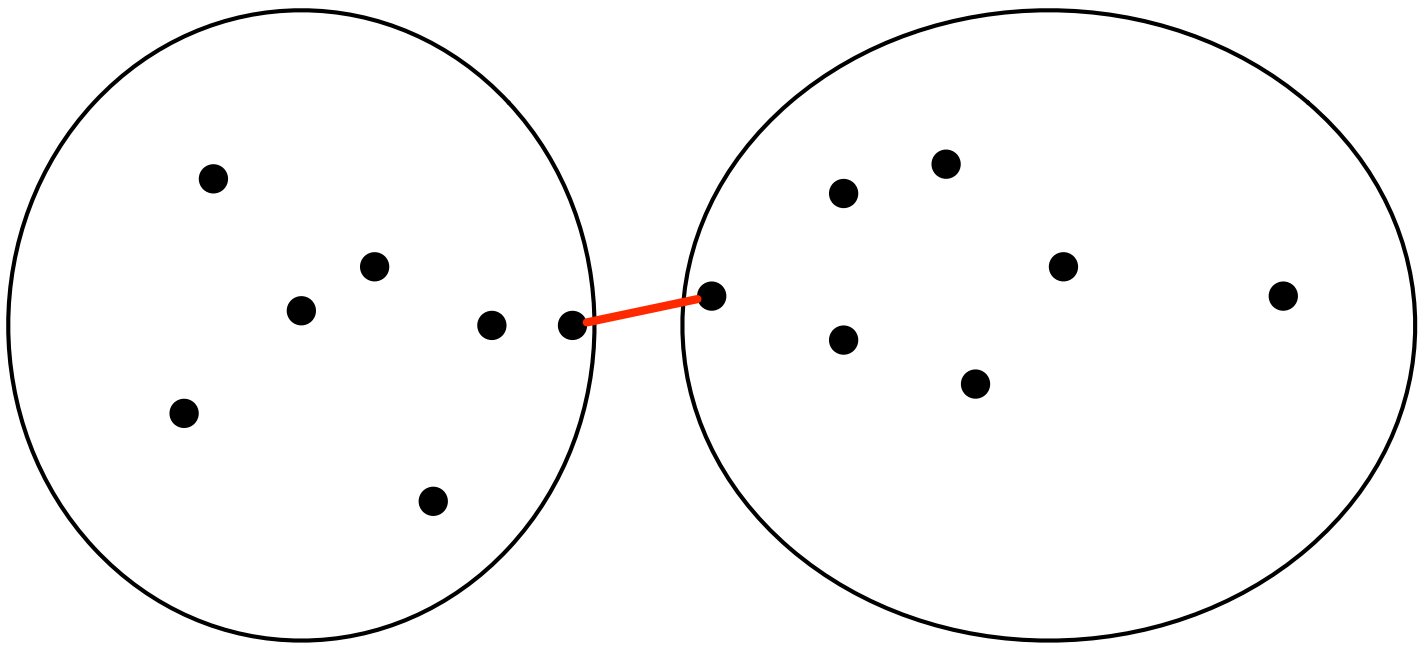
$$d(C, D) = \frac{1}{|C||D|} \sum_{x \in C, y \in D} d(x, y)$$

- Others: mean, centroid, ward, weighted versions...

Linkage Algorithms: Similarity Choices

- Single (or minimum) linkage:

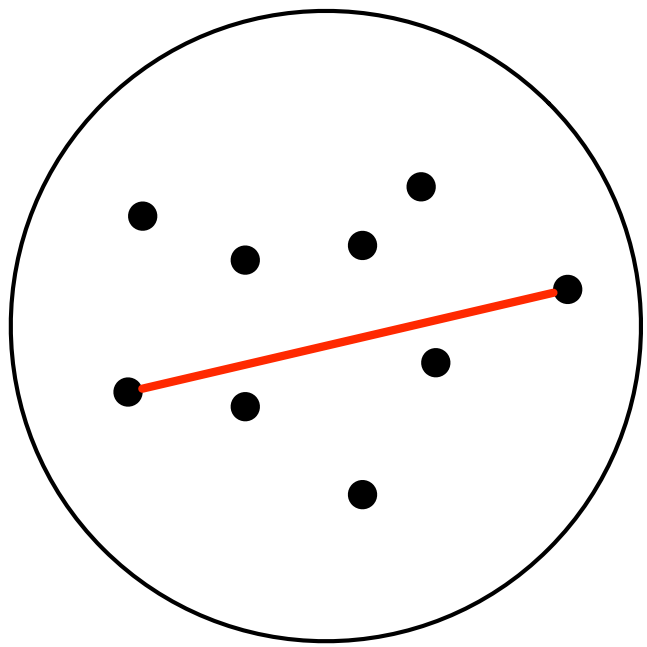
$$d(C, D) = \min_{x \in C, y \in D} d(x, y)$$



Linkage Algorithms: Similarity Choices

- Complete (or maximum) linkage:

$$d(C, D) = \max_{x \in C, y \in D} d(x, y)$$



Linkage Algorithms: Pros and Cons

+ Easy and fast.
+ Well-known and well-accepted.
– Distance metric sometimes unclear.
– Cannot handle partially observed data.
– No clear semantics for the optimality of the constructed tree.
– No uncertainty about the tree structure.

Probabilistic Hierarchical Clustering

Same framework as normal linkage algorithms.

Use probabilistic models to define cluster distance:

$$d(C, D) = -\log \frac{p(C \cup D)}{p(C)p(D)}$$

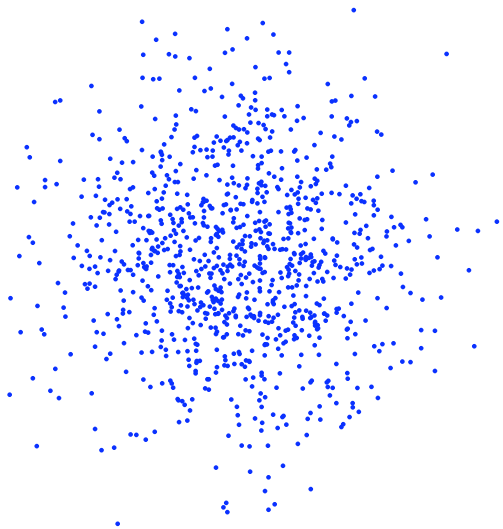
[Friedman 2003, Heller & Ghahramani 2005]

Probabilistic Hierarchical Clustering

A common probabilistic model: Gaussian

$$p(C) = \prod_{x \in C} |2\pi\sigma^2|^{-\frac{D}{2}} e^{-\frac{|x-\mu|^2}{2\sigma^2}}$$

where μ, σ are fit to data in C.
(If you are Bayesian, you integrate them out).



Another common model: Bernoulli

$$p(C) = \prod_{x \in C} \prod_{k} \pi_k^{\delta(x_k=1)} (1-\pi_k)^{\delta(x_k=0)}$$

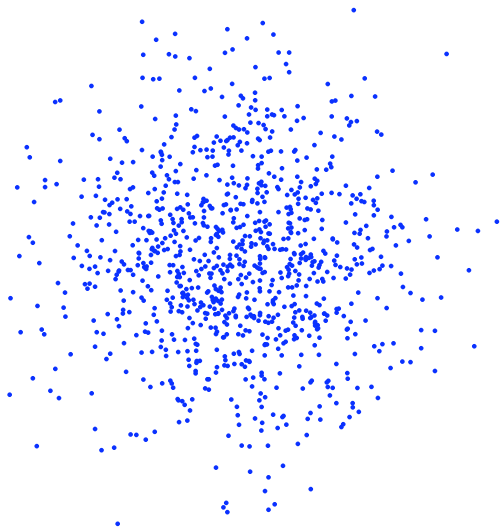
[Friedman 2003]

Probabilistic Hierarchical Clustering

- The Gaussian imposes a strong constraint on how it thinks clusters should shape like.
- The cluster distance measures how spherical (or Gaussian) the cluster is

$$d(C, D) = -\log \frac{p(C \cup D)}{p(C)p(D)}$$

- Clusters are merged if the merger produces a more Gaussian looking cluster.



[Friedman 2003]

Probabilistic Hierarchical Clustering

Different interpretation: mixture model.

- Model data set with a (standard) mixture model.
- Start with each data item x_i in its own cluster $C_i = \{x_i\}$.
- For $t = 1, \ldots, n-1$:
  - Find pair of clusters such that the likelihood of the data is maximum after merger. Equivalent to finding

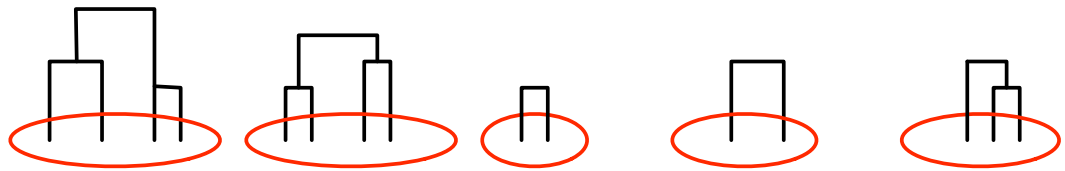
$$C, D \leftarrow \operatorname*{argmax}_{C \neq D} \log \frac{p(C \cup D)}{p(C)p(D)}$$

  - If $\log \frac{p(C \cup D)}{p(C)p(D)} > 0$ merge C and D, else stop.

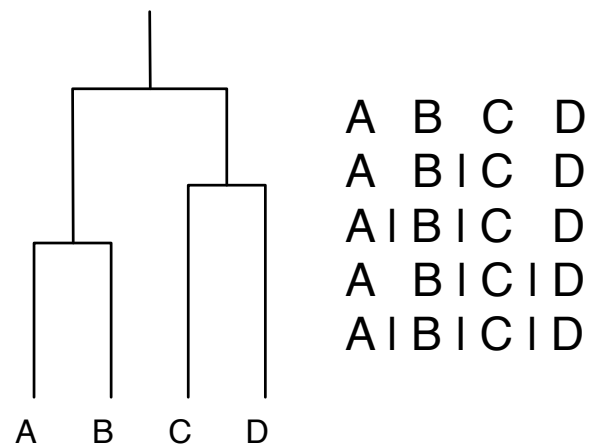
[Friedman 2003]

Probabilistic Hierarchical Clustering

[Friedman 2003] assumes that a partially constructed tree corresponds to a mixture model with each subtree being a mixture component.



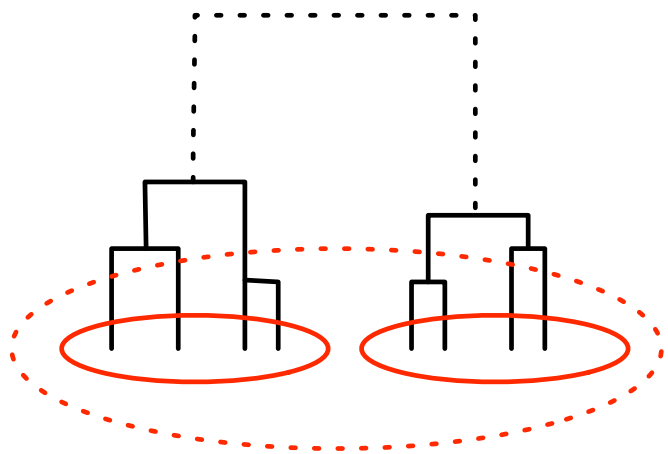
[Heller & Ghahramani 2005] assumes that each subtree itself corresponds to a mixture model.



Probabilistic Hierarchical Clustering

Probability of data under a subtree can be computed recursively:

$$p(\langle S, T \rangle) = \pi p_0(\text{Data}(S) \cup \text{Data}(T)) + (1 - \pi) p(S) p(T)$$



The approach can be used to obtain a lower bound on the probability of data under a *Dirichlet process mixture model*.

[Heller & Ghahramani 2005]

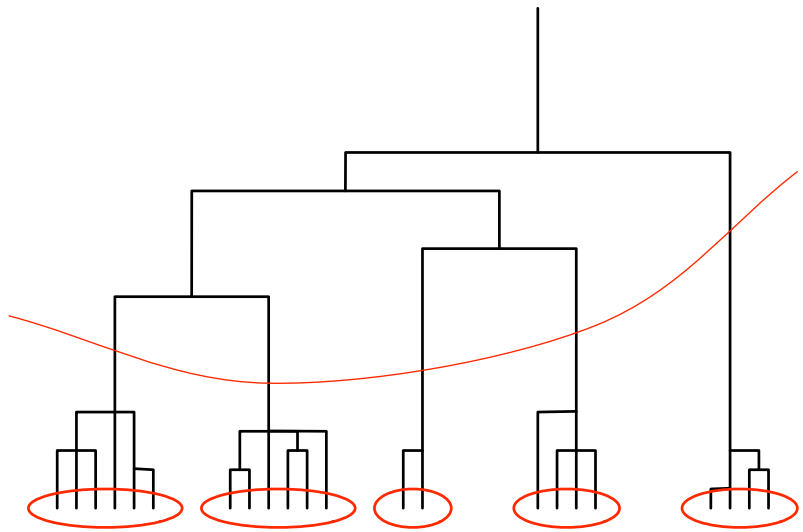
Probabilistic Hierarchical Clustering: Pros and Cons

+ Easy and efficient as well.
+ Same framework as normal linkage algorithms.
+− Probabilistic models more interpretable, but less flexible than distance metrics.
+ Deals nicely with partially observed data.
+ There is a coherent measure of goodness-of-fit for resulting model.
− No notion of uncertainty in the tree structure.

Hierarchical Clustering

Two distinct beliefs about the underlying structure of data:

- We believe data comes in unrelated groups or clusters.
  - Mixture model.
  - Use hierarchical clustering as an efficient search procedure.
- We believe data has an underlying tree structure:
  - Use hierarchical clustering to find the tree.

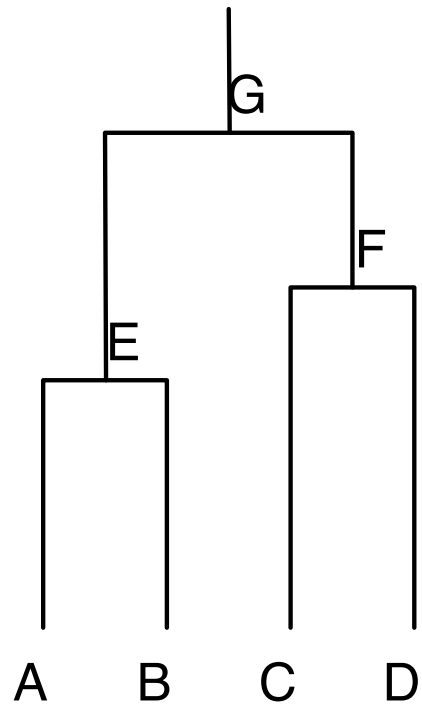


Tree-structured Probabilistic Models

Model data under each subtree using a tree-structured model:

$$
\begin{aligned}
& p(A, B, C, D|T) \\
= & \sum_{E,F,G} p(G) \\
& \cdot p(E|G)p(F|G) \\
& \cdot p(A|E)p(B|E)p(C|F)p(D|F)
\end{aligned}
$$

[Williams 2000, Neal 2001, Teh et al 2007]



Tree-Structured Probabilistic Models

- Model data set with a tree-structured model.
- Start with each data item x_i in its own subtree $T_i = \langle x_i \rangle$.
- For $t = 1, \ldots, n-1$:
  - Find pair of subtrees such that the likelihood of the data is maximum after merger. Equivalent to finding

$$S, T \leftarrow \operatorname*{argmax}_{S \neq T} \log \frac{p(\text{Data}(S) \cup \text{Data}(T) | \langle S, T \rangle)}{p(\text{Data}(S)|S)p(\text{Data}(T)|T)}$$

  - If $\log \frac{p(\text{Data}(S) \cup \text{Data}(T) | \langle S, T \rangle)}{p(\text{Data}(S)|S)p(\text{Data}(T)|T)} > 0$ merge S and T, else stop.

$p(\text{Data}(S) \cup \text{Data}(T) | \langle S, T \rangle)$ can be computed efficiently in a recursive manner using belief propagation.

[Teh et al 2007]

Bayesian Tree-Structured Models

- To model uncertainty over trees, use a distribution over trees:

$$p(T|\text{Data}) = \frac{p(T)p(\text{Data}|T)}{p(\text{Data})}$$

- Model for data $p(\text{Data}|T)$ is tree-structured.
- Posterior is often intractable, approaches include greedy agglomerative construction (previous slide), Markov chain Monte Carlo, and sequential Monte Carlo.
- Monte Carlo algorithms more intricate and expensive.
- Interesting nonparametric Bayesian priors over trees.

[Williams 2000, Neal 2001, Teh et al 2007]

Comparisons

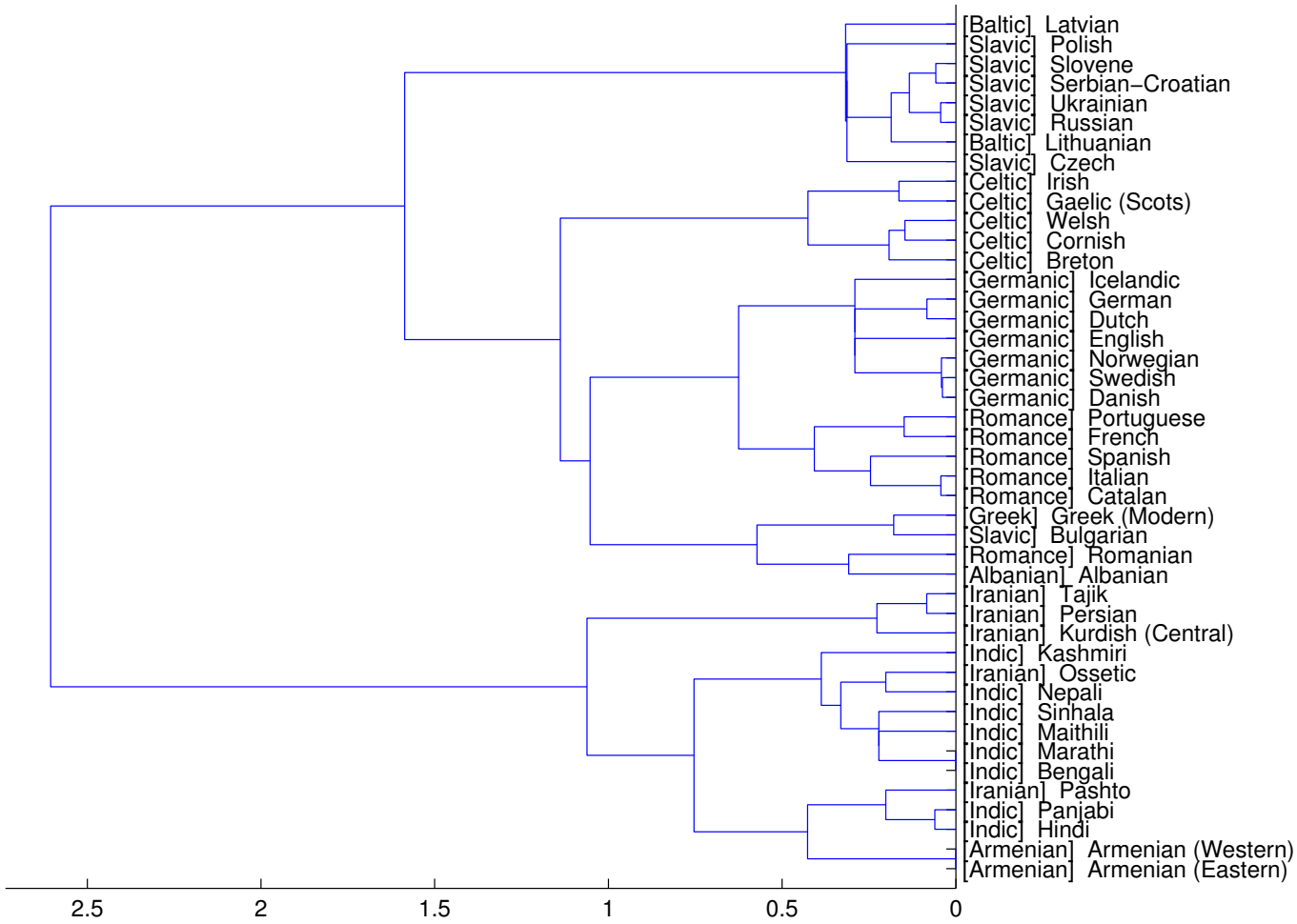
	MNIST		
	Avg-link	**HG**	**TDR**
Purity	.363±.004	.392±.006	**.412±.006**
Subtree	.581±.005	.579±.005	**.610±.005**
LOO-acc	.755±.005	.763±.005	**.773±.005**
	SPAMBASE		
	Avg-link	**HG**	**TDR**
Purity	.616±.007	**.711±.010**	.689±.008
Subtree	.607±.011	.549±.015	**.661±.012**
LOO-Acc	.846±.010	.832±.010	**.861±.008**

Comparisons

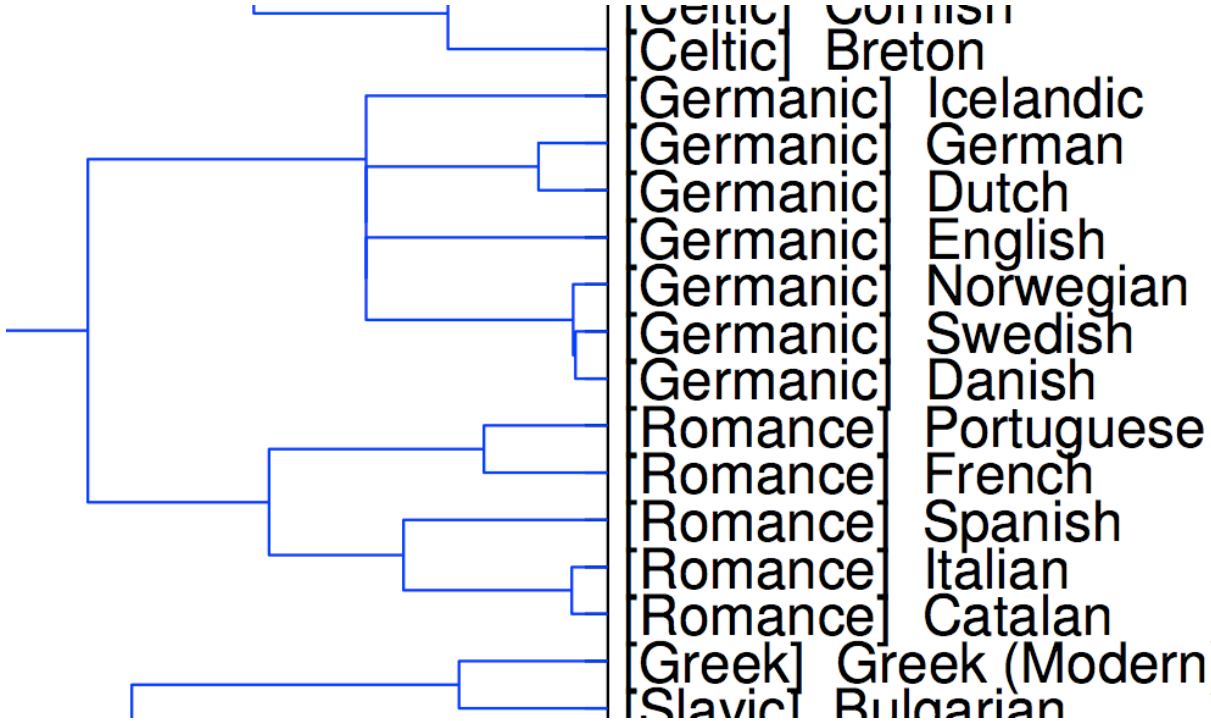
	WALS (Indo-European)		
	Avg-link	**HG**	**TDR**
Purity	.510	.491	.**813**
Subtree	.414	.414	.**690**
LOO-acc	.538	.590	.**769**

	WALS (Whole World)		
	Avg-link	**HG**	**TDR**
Purity	.162	.160	.**269**
Subtree	.**227**	.099	.177
LOO-acc	.080	.248	.**369**

Phylolinguistics



Phylolinguistics



Bayesian Tree-Structured Models: Pros and Cons

- − Can be expensive if want full posterior.
- − Less common, less well-understood.
- + Fully generative probabilistic models.
- + Deals nicely with partially observed data.
- + There is a coherent measure of goodness-of-fit for resulting model.
- + Notion of uncertainty in the tree structure.

Discussion

- A quick overview of some popular and promising approaches to hierarchical clustering.
- Dimension 1: top-down vs bottom-up vs Monte Carlo search.
- Dimension 2: flat clustering (mixture model) vs tree-structured model.
- Dimension 3: algorithmic vs probabilistic vs Bayesian.