Bayesian Hierarchical Clustering

Katherine Heller

Zoubin Ghahramani

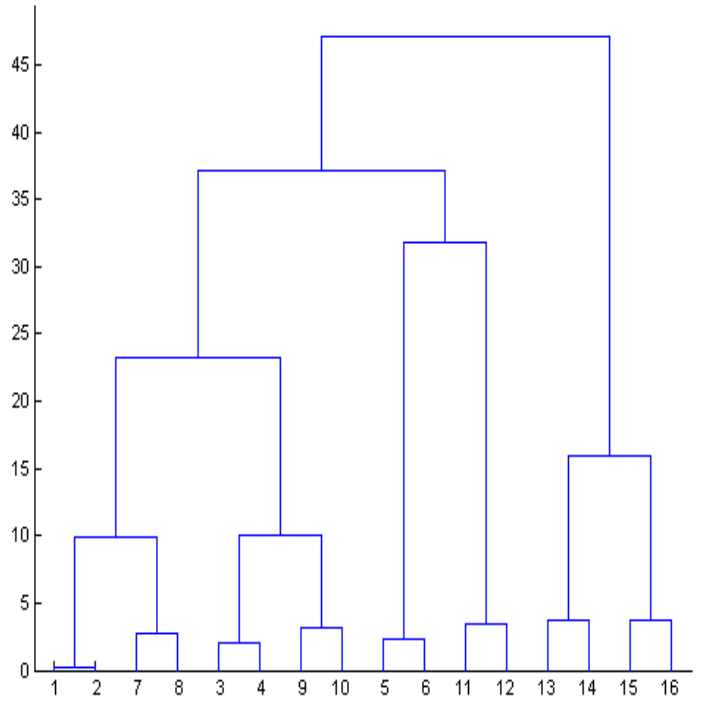
Presented by

Soumya Ghosh

Slides courtesy:
Katherine Heller

Hierarchical Clustering

- Classic algorithm
- Agglomerative, bottom up clustering
- Initialize with each data instance as its own cluster.
- Progressively merge the most similar pairs creating a binary tree



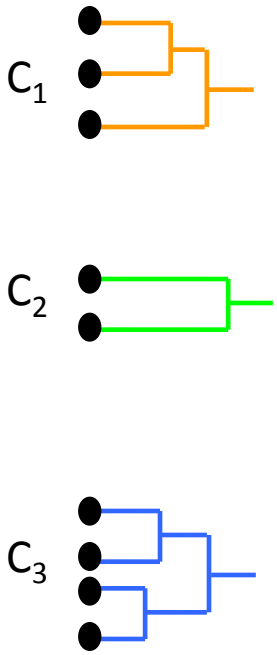
Problems

- No probabilistic model of the data :
  - Difficult to deal with new data instances
  - Can’t be compared to or combined with other probabilistic models
  - No notion of how good a particular clustering of the data is

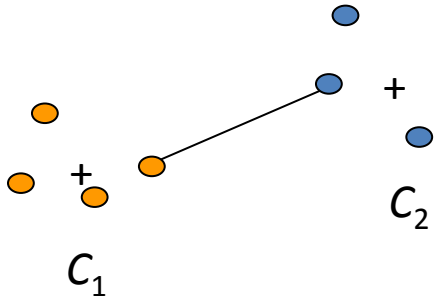
- Correct distance metric?

- More importantly, need to specify distance between groups

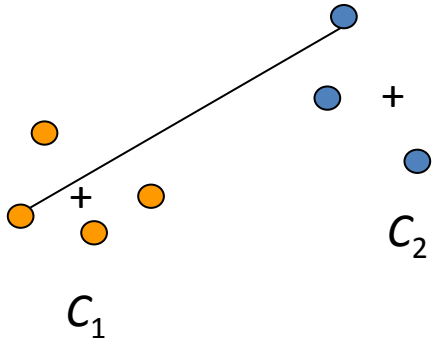
Problems



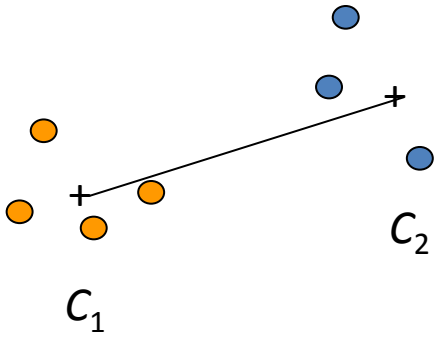
Merge which pair of clusters?



Single Linkage



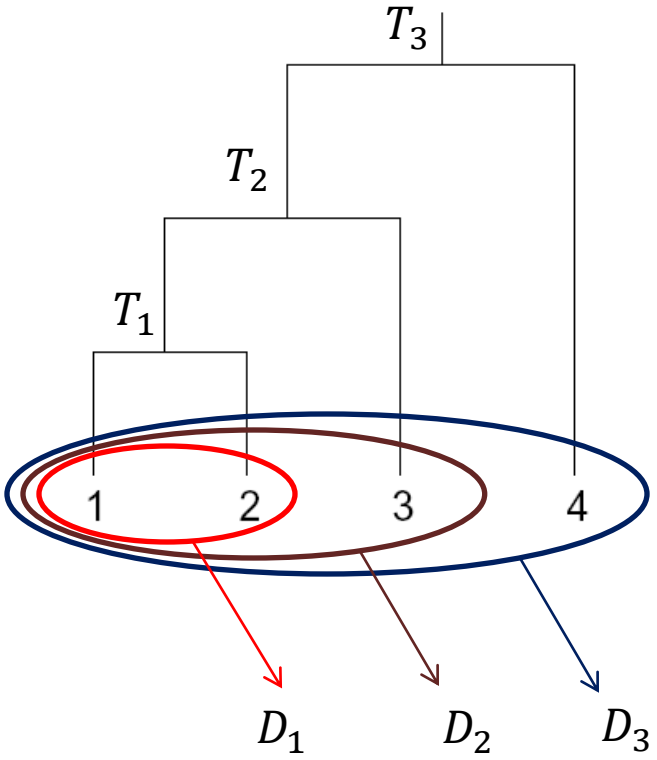
Complete Linkage



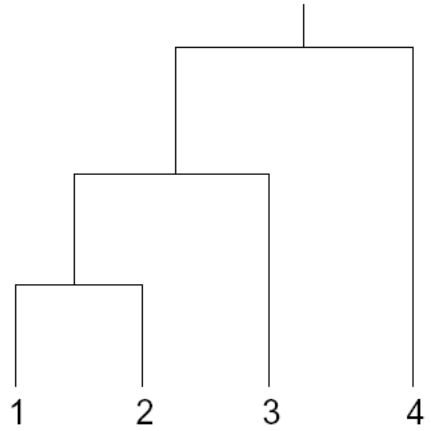
Centroid Linkage

Bayesian Hierarchical Clustering

- Notation
  - $D = \{x^1, \ldots, x^n\}$
  - $D_i \subset D$, data at leaves of tree T_i



Tree-Consistent Partitions



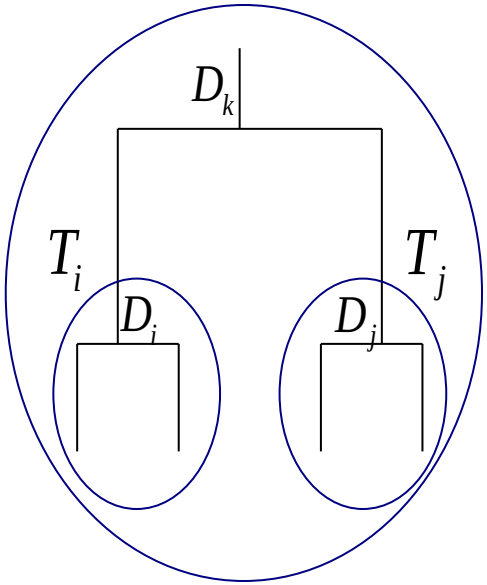
- Consider the above tree and all 15 possible partitions of {1,2,3,4}:

  (1)(2)(3)(4), (1 2)(3)(4), (1 3)(2)(4), (1 4)(2)(3), (2 3)(1)(4),
  (2 4)(1)(3), (3 4)(1)(2), (1 2)(3 4), (1 3)(2 4), (1 4)(2 3),
  (1 2 3)(4), (1 2 4)(3), (1 3 4)(2), (2 3 4)(1), (1 2 3 4)

- (1 2) (3) (4) and (1 2 3) (4) are tree-consistent partitions
- (1)(2 3)(4) and (1 3)(2 4) are not tree-consistent partitions

Bayesian Hierarchical Clustering

- Data generated from a Dirichlet Process Mixture.
- Similarity is now measured through a statistical test.
- For each candidate merge compare two hypotheses:
  - H_1 : all data in D_k generated from the same component
  - H_2 : data in D_k came from some other clustering consistent with the sub trees $T_i \; and \; T_j$.



Computing the Marginal Likelihood for H_1

- Given that our model is a DPM we can compute
  - $P\left(D_k \middle| H_1^k\right)$ - *data at tree* T_k *was generated from the same cluster.*
  - $P\left(D_k \middle| H_1^k\right) = \int p(D_k|\theta)p(\theta|\beta)d\theta$
  - Easy to compute if the model has conjugacy.

Marginal Likelihood for the alternative hypothesis

- $P\left(D_k \middle| H_2^k\right)$ - *D_k was generated from two or more components defining partitions consistent with trees T_i and T_j*
  - $P\left(D_k \middle| H_2^k\right) = P(D_i|T_i)P\left(D_j \middle| T_j\right)$
  - $P(D_k|T_k) = \pi_k p\left(D_k \middle| H_k^1\right) + (1 - \pi_k)P\left(D_k \middle| H_2^k\right)$
    - $\pi_k = p(H_k^1)$

Algorithm Details

input: data $\mathcal{D} = \{\mathbf{x}^{(1)} \ldots \mathbf{x}^{(n)}\}$, model $p(\mathbf{x}|\theta)$, prior $p(\theta|\beta)$

initialize: number of clusters $c = n$, and $\mathcal{D}_i = \{\mathbf{x}^{(i)}\}$ for $i = 1 \ldots n$

while $c > 1$ **do**

Find the pair $\mathcal{D}_i$ and $\mathcal{D}_j$ with the highest probability of the merged hypothesis:

$$r_k = \frac{\pi_k p(\mathcal{D}_k|\mathcal{H}_1^k)}{p(\mathcal{D}_k|T_k)}$$

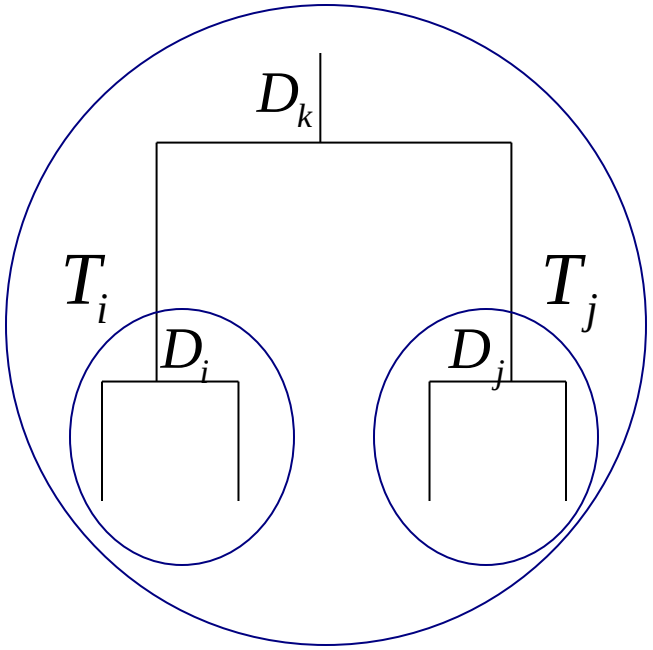
Merge $\mathcal{D}_k \leftarrow \mathcal{D}_i \cup \mathcal{D}_j, \quad T_k \leftarrow (T_i, T_j)$

Delete D_i and D_j, $c \leftarrow c - 1$

end while

output: Bayesian mixture model where each tree node is a mixture component

The tree can be cut at points where $r_k < 0.5$



Computing the Prior for H_1^k

- π_k is the relative mass of the partition where all points are in one cluster vs all other partitions consistent with the subtrees, in a Dirichlet process mixture model
- Can be computed bottom up

Initialise each leaf i to have $d_i = \alpha$, $\pi_i = 1$
for each internal node k **do**
$d_k = \alpha\Gamma(n_k) + d_{\text{left}_k}\ d_{\text{right}_k}$
$\pi_k = \frac{\alpha\Gamma(n_k)}{d_k}$
end for

Marginal Likelihood of a Dirchlet Process Mixture

- Marginal Likelihood :

$$p(\mathcal{D}|\alpha,\beta) = \sum_{v\in\mathcal{V}} p(v|\alpha)p(\mathcal{D}|v,\beta)$$

- $\nu = \{\nu_1, \ldots \nu_N\}$
- From the CRP (distribution over partitions) we have

$$p(\nu_N = l|\nu_1,\nu_2,\ldots,\nu_{N-1}) = \begin{cases} \frac{n_l}{N-1+\alpha} & \text{if } l \leq m \\ \frac{\alpha}{N-1+\alpha} & \text{otherwise} \end{cases}$$

Marginal Likelihood of a Dirchlet Process Mixture

$$p(\nu|\alpha) = p(\nu_1)p(\nu_2|\nu_1)p(\nu_3|\nu_2,\nu_1)\ldots$$

$$= \frac{\alpha^m \prod_l \Gamma(n_l)}{\frac{\Gamma(N+\alpha)}{\Gamma(\alpha)}}$$

$$p(D|\nu,\beta) = \prod_l P(D_l|\beta)$$

Lemma 1:

$$p(\mathcal{D}_k) = \sum_{v\in\mathcal{V}} \frac{\alpha^{m_v} \prod_{\ell=1}^{m_v} \Gamma(n_\ell^v)}{\left[\frac{\Gamma(n_k+\alpha)}{\Gamma(\alpha)}\right]} \prod_{\ell=1}^{m_v} p(\mathcal{D}_\ell^v)$$

Marginal Likelihood of Tree Consistent Partitions

$$p(\mathcal{D}_k|T_k) = \sum_{v \in \mathcal{V}_T} \frac{\alpha^{m_v} \prod_{\ell=1}^{m_v} \Gamma(n_\ell^v)}{d_k} \prod_{\ell=1}^{m_v} p(\mathcal{D}_\ell^v)$$

$$\frac{d_k \Gamma(\alpha)}{\Gamma(n_k + \alpha)} p(\mathcal{D}_k|T_k) \leq p(\mathcal{D}_k)$$

- Lower bounds the true DPM marginal likelihood

Combinatorial Lower Bounds

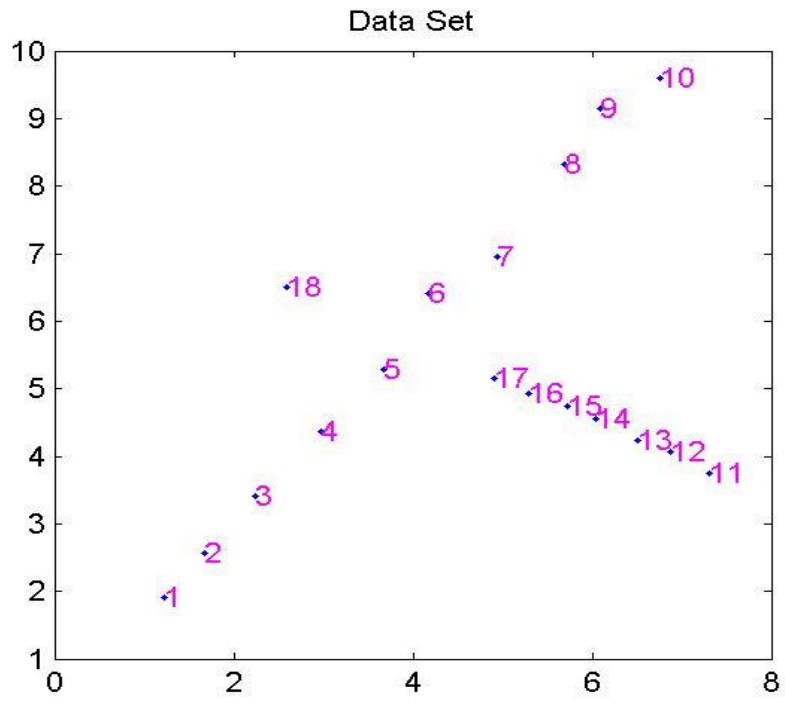
- BHC forms a lower bound for the marginal likelihood of an infinite mixture model by efficiently summing over an exponentially large **subset** of all partitions.

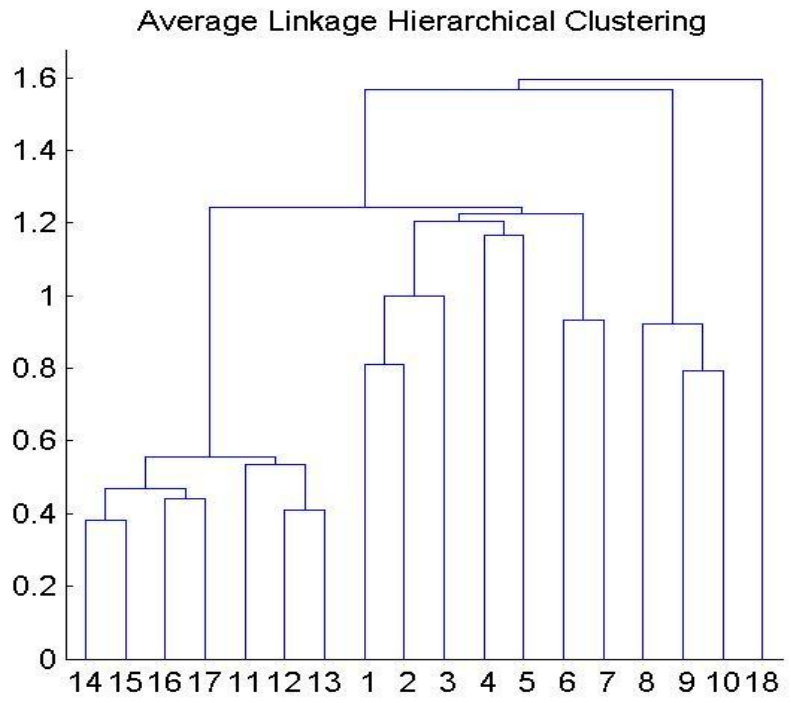
- Idea is to deterministically sum over partitions with high probability, thereby accounting for most of the mass.

Experimental Results

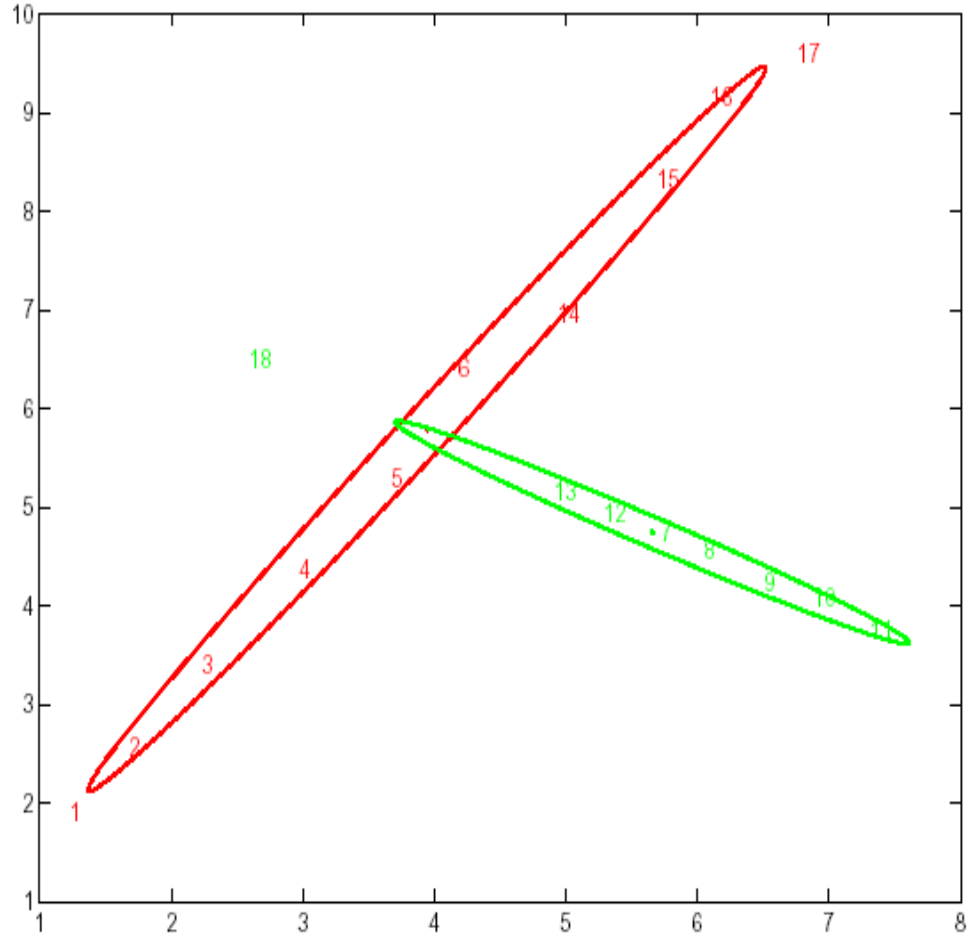
- Toy Example
- UCI Datasets
- Newsgroup Clustering

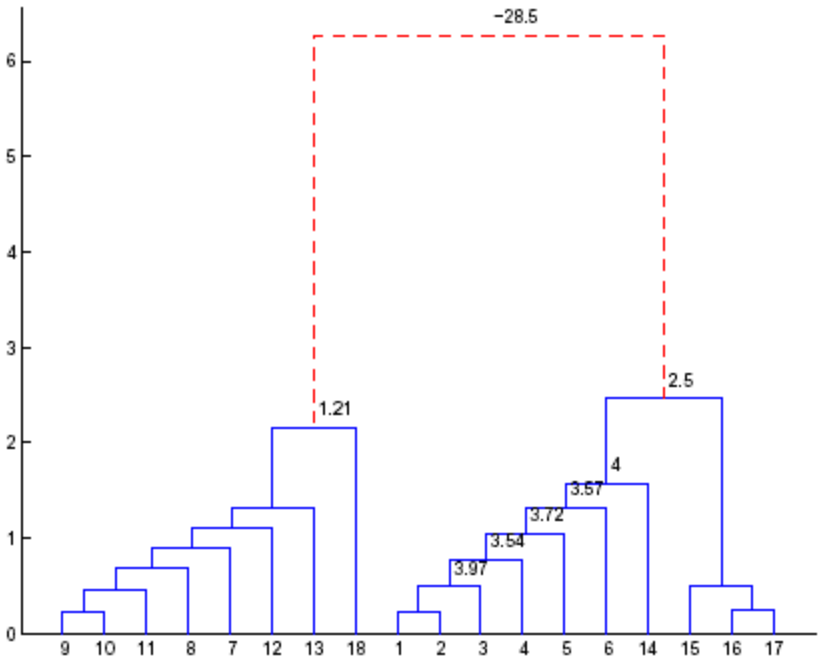
Results: a Toy Example



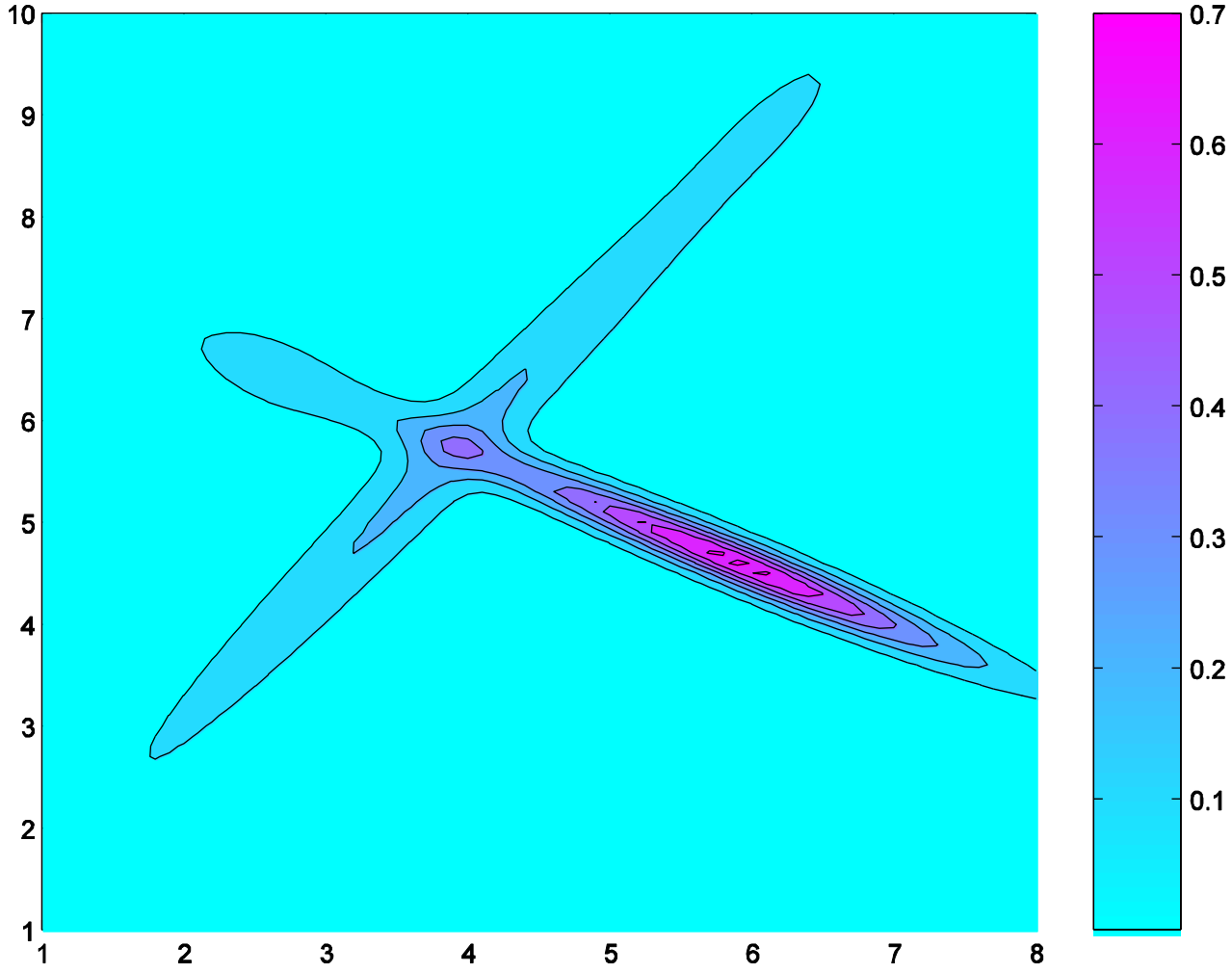


Results: a Toy Example





Predicting New Data Points



Results: Purity Scores

Data Set	Single Linkage	Complete Linkage	Average Linkage	BHC
Synthetic	0.599 ± 0.033	0.634 ± 0.024	0.668 ± 0.040	$\mathbf{0.828 \pm 0.025}$
Newsgroups	0.275 ± 0.001	0.315 ± 0.008	0.282 ± 0.002	$\mathbf{0.465 \pm 0.016}$
Spambase	0.598 ± 0.017	0.699 ± 0.017	0.668 ± 0.019	$\mathbf{0.728 \pm 0.029}$
3Digits	0.545 ± 0.015	0.654 ± 0.013	0.742 ± 0.018	$\mathbf{0.807 \pm 0.022}$
10Digits	0.224 ± 0.004	0.299 ± 0.006	0.342 ± 0.005	$\mathbf{0.393 \pm 0.015}$
Glass	0.478 ± 0.009	0.476 ± 0.009	$\mathbf{0.491 \pm 0.009}$	0.467 ± 0.011

Purity is a measure of how well the hierarchical tree structure is correlated with the labels of the known classes.

[2] Let T be a tree with leaves $1, \ldots, n$ and $c_1, \ldots, c_n$ be the known discrete class labels for the data points at the leaves. Pick a leaf ℓ uniformly at random; pick another leaf j uniformly in the same class, i.e. $c_\ell = c_j$. Find the smallest subtree containing ℓ and j. Measure the fraction of leaves in that subtree which are in the same class (c_ℓ). The expected value of this fraction is the dendrogram purity, and can be computed exactly in a bottom up recursion on the dendrogram. The purity is 1 iff all leaves in each class are contained in some pure subtree.

4 Newsgroups Results

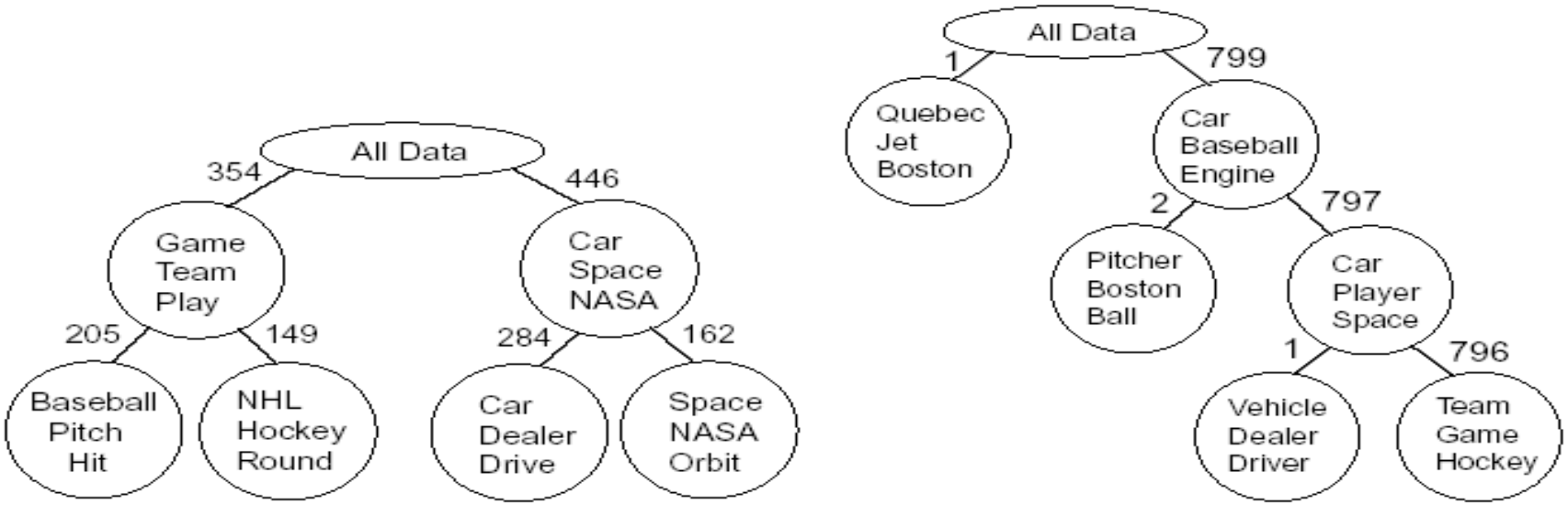
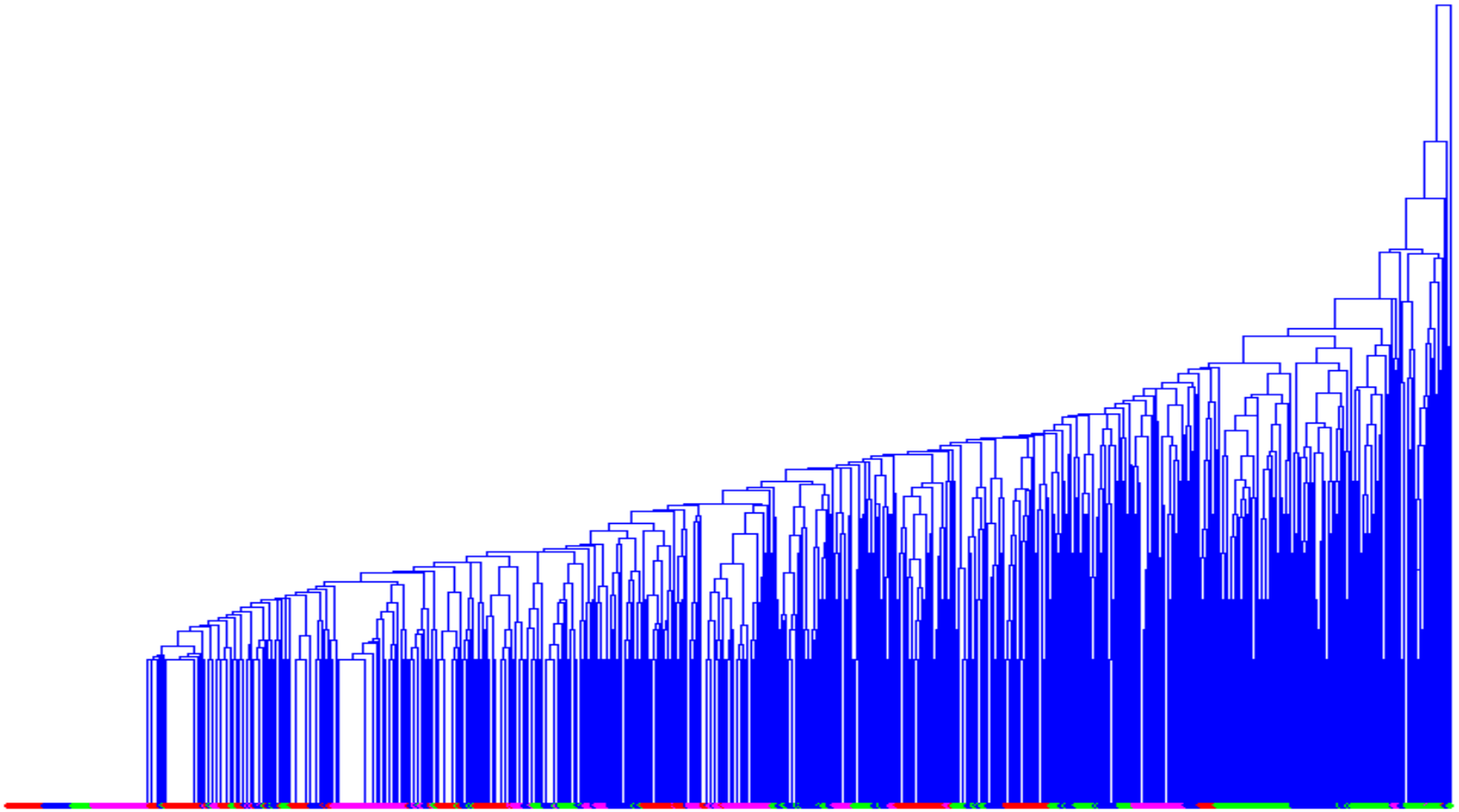


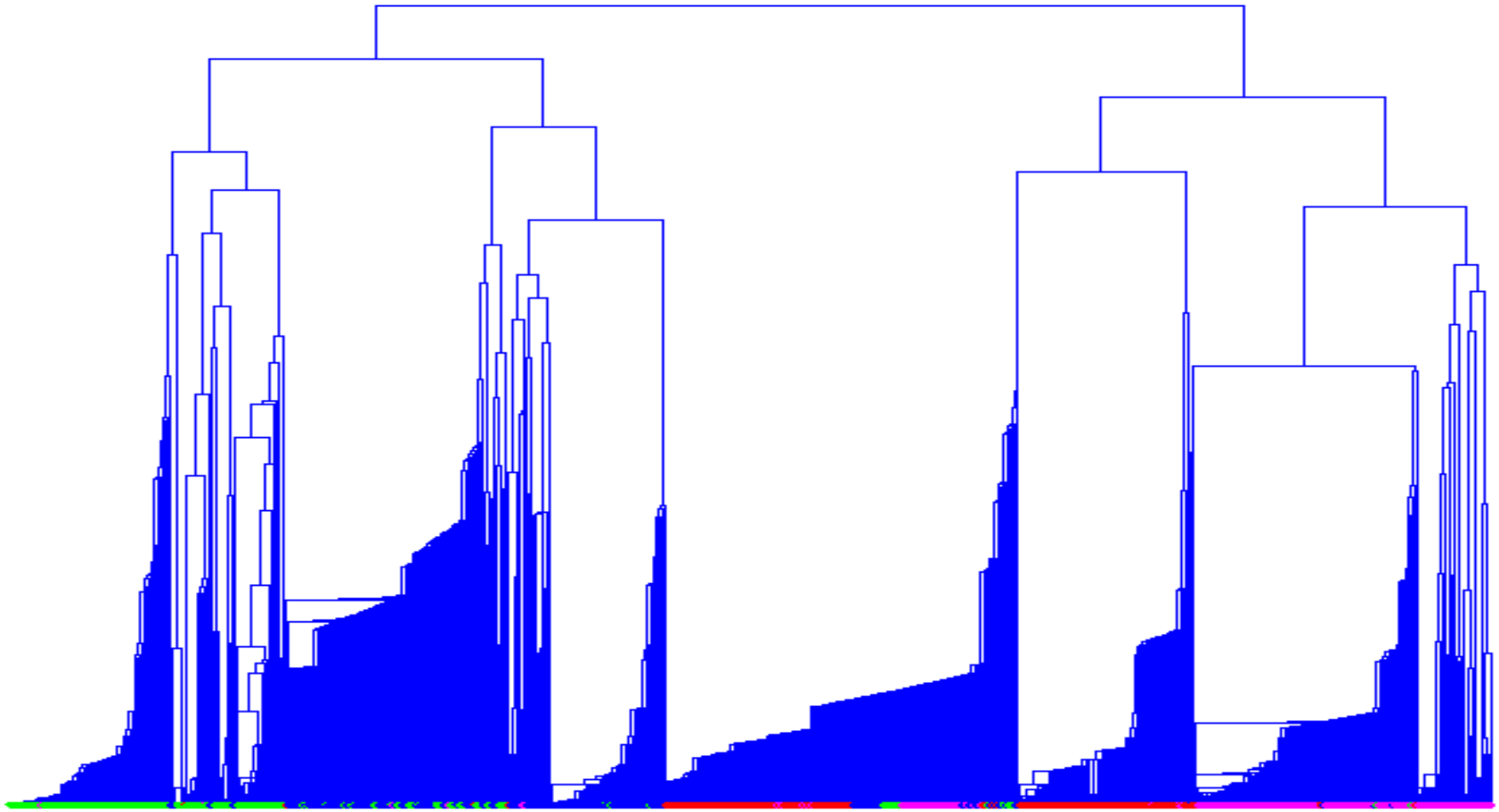
Figure 5. Top level structure, of BHC (left) vs. Average Linkage HC, for the newsgroup dataset. The 3 words shown at each node have the highest mutual information between the cluster of documents at that node versus its sibling, and occur with higher frequency in that cluster. The number of documents at each cluster is also given.

800 examples, 50 attributes: rec.sport.baseball, rec.sports.hockey, rec.autos, sci.space

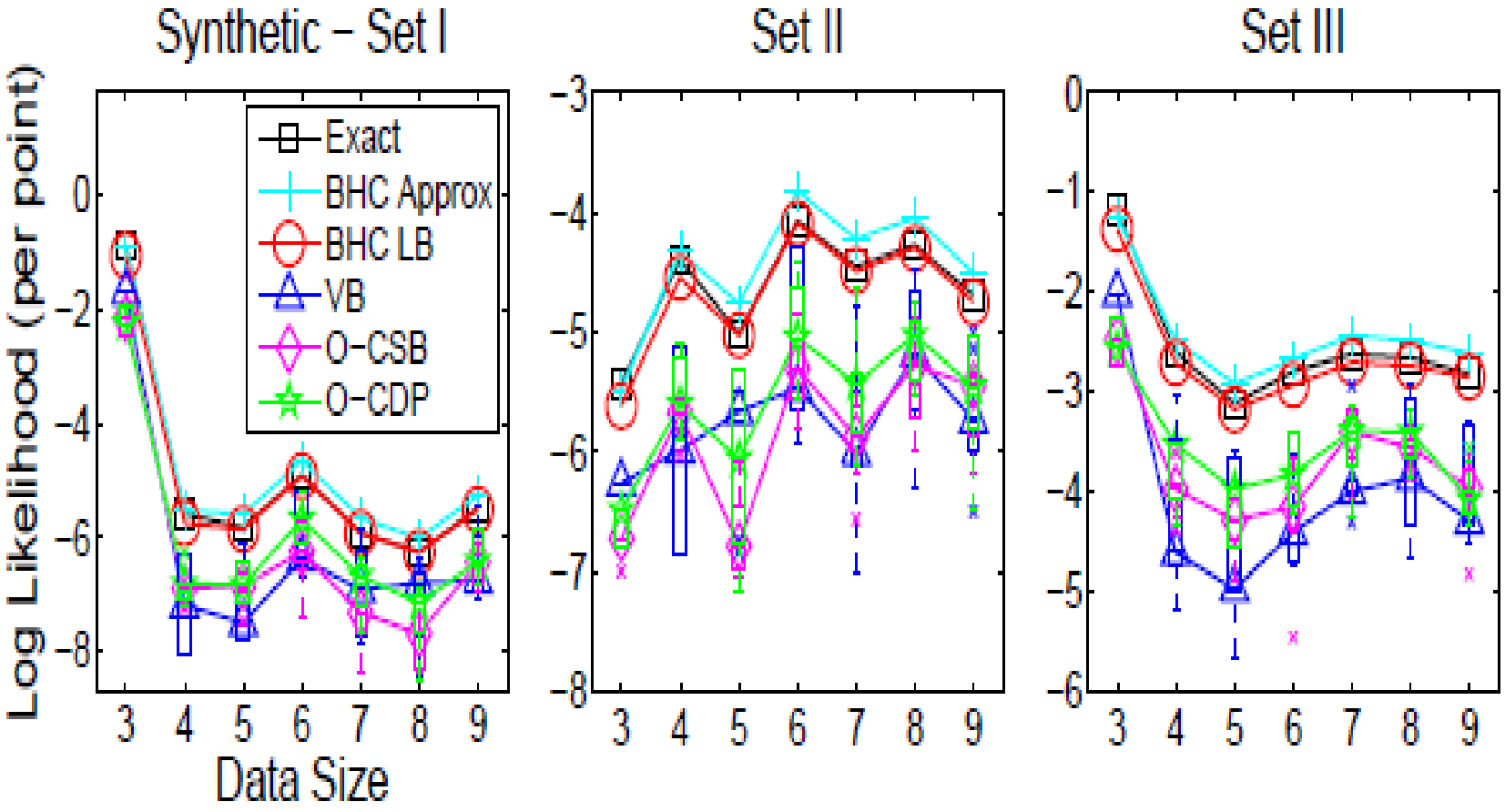
Newsgroups: Average Linkage HC



Newsgroups: Bayesian HC



Comparison with Mean Field Lower Bound



Issues and Opportunities

- Greedy algorithm:
  - The algorithm may not find the globally optimal tree
- No tree uncertainty:
  - The algorithm finds a single tree, rather than a distribution over plausible trees
- $O(n^2)$ complexity for building tree
- Extend inference algorithm to more sophisticated models.