Efficient Active Algorithms for Hierarchical Clustering

Akshay Krishnamurthy AKSHAYKR@CS.CMU.EDU
Sivaraman Balakrishnan SBALAKRI@CS.CMU.EDU
Min Xu MINX@CS.CMU.EDU
Aarti Singh AARTI@CS.CMU.EDU
Carnegie Mellon University 5000 Forbes Avenue, Pittsburgh, PA 15213

Abstract

Advances in sensing technologies and the growth of the internet have resulted in an explosion in the size of modern datasets, while storage and processing power continue to lag behind. This motivates the need for algorithms that are efficient, both in terms of the number of measurements needed and running time. To combat the challenges associated with large datasets, we propose a general framework for *active* hierarchical clustering that repeatedly runs an off-the-shelf clustering algorithm on small subsets of the data and comes with guarantees on performance, measurement complexity and runtime complexity. We instantiate this framework with a simple spectral clustering algorithm and provide concrete results on its performance, showing that, under some assumptions, this algorithm recovers all clusters of size $\Omega(\log n)$ using $O(n \log^2 n)$ similarities and runs in $O(n \log^3 n)$ time for a dataset of n objects. Through extensive experimentation we also demonstrate that this framework is practically alluring.

1. Introduction

Clustering is a ubiquitous task in exploratory data analysis, data mining, and several application domains. In clustering, we assign each object to one or more groups so that objects in the same group are very similar while objects in different groups are dissimilar. In a hierarchical clustering, the groups have multiple resolutions, so that a large cluster may be recursively divided into smaller sub-clusters. There exist many effective algorithms for clustering, but as modern data sets get larger, the fact that these algorithms require *every* pairwise similarity between objects poses a serious measurement and/or computational burden and limits the practicality of these algorithms. It is therefore practically appealing to develop clustering algorithms that are effective on large scale problems from both a measurement and a computational perspective.

To achieve both measurement and computational improvements, we focus on reducing the number of similarity measurements required for clustering. This approach results in immediate reduction in measurement overhead in applications where similarities are observed directly, but it can also provide dramatic computational gains in applications where similarities between objects are computed via some kernel evaluated on observed object features. The case of internet topology inference is an example of the former, where covariance in the packet delays observed at nodes reflects the similarity between them. Obtaining these similarities requires injecting probe packets into the network and places a significant burden on network infrastructure. Phylogenetic inference and other biological sequence analyses are examples of the latter, where computationally intensive edit distances are often used. In both cases our algorithms are dramatically faster than many popular algorithms.

In this paper, we propose a novel framework for speeding up hierarchical clustering algorithms through *activization*–creating active versions of the algorithms where only a small number of informative similarities are measured. Our framework allows the user to specify various levels of activeness and we provide theoretical analysis that quantifies the resulting trade-off between measurement overhead and computation time on one hand, and statistical accuracy on the other.

As a detailed example, we apply our framework to spectral clustering. Spectral clustering is a very popular clustering technique that relies on the structure

Appearing in *Proceedings of the 28th International Conference on Machine Learning*, Bellevue, WA, USA, 2011.

of the eigenvectors of the Laplacian of the similarity matrix. These algorithms have received considerable attention in recent years because of their empirical success, but they suffer from the fact that they require all $n(n-1)/2$ similarities between the n objects to be clustered and must compute a spectral decomposition, which on large datasets can be computationally prohibitive. Our active algorithm avoids this limitation by subsampling few objects in each round and only computing eigenvectors of very small sub-matrices. By appealing to previous statistical guarantees (Balakrishnan et al., 2011), we can show that this algorithm has desirable theoretical properties, both in terms of statistical and computational performance.

2. Related Work

There is a large body of work on hierarchical and partitional clustering algorithms, many coming with various theoretical guarantees, but only few algorithms attempt to minimize the number of pairwise similarities used (Eriksson et al., 2011; Balcan & Gupta, 2010; Shamir & Tishby, 2011). Along this line, the work of Eriksson et. al. (2011) and Shamir and Tishby (2011) is closest in flavor to ours.

Eriksson et. al. (2011) develop an active algorithm for hierarchical clustering and analyze the correctness and measurement complexity of this algorithm under noise model where a small fraction of the similarities are inconsistent with the hierarchy. They show that for a constant fraction of inconsistent similarities, their algorithm can recover hierarchical clusters up to size $\Omega(\log n)$ using $O(n \log^2 n)$ similarities. Our analysis for ActiveSpectral yields similar results in terms of noise tolerance, measurement complexity, and resolution, but in the context of i.i.d. subgaussian noise rather than inconsistencies. Our algorithm is also computationally more efficient.

Another approach to minimizing the number of similarities used is via perturbation theory, which suggests that randomly sampling the entries of a similarity matrix preserves many of its important properties, such as its spectral norm (Achlioptas & McSherry, 2001). With this result, the Davis-Kahan theorem suggests that spectral clustering algorithms, which look at the eigenvectors of the Laplacian associated with the similarity matrix, can succeed in recovering the clusters. This intuition is formalized by Shamir and Tishby (2011) who analyze a binary spectral algorithm that randomly samples b entries from the similarity matrix. Their results imply that as long as $b = \Omega(n \log^{3/2} n)$ their algorithm will find flat k-way clusters of size $\Omega(n)$ with high probability. Our work, translated to the flat clustering setting improves this guarantee; Theorem 2 implies that $O(n \log n)$ similarities are needed to recover the clustering. Furthermore, we can give guarantees on the size of smallest cluster $\Omega(\log n)$ that can be recovered in a hierarchy by *selectively* sampling similarities at each level.

Recently (Voevodski et al., 2012) proposed an active algorithm for flat k-way clustering that selects $O(k)$ landmarks and partitions the objects using distances to these landmarks. Theoretically, the authors guarantee approximate-recovery of clusters of size $\Omega(n)$ using $O(nk)$ pairwise distances. This idea of selecting landmarks bears strong resemblence to the first phase of our active clustering algorithm and also has connections to the Landmark MDS algorithm of de Silva and Tenenbaum (2002). These approaches are tied to specific algorithms, while our framework is much more general. Moreover, we guarantee exact cluster recovery (under mild assumptions) rather than approximate recovery, which translates into guarantees on hierarchical clustering.

A related direction is the body of work on efficient streaming and online algorithms for approximating the k-means and k-medians objectives (See for example (Charikar et al., 2003; Shindler et al., 2011)). As with (Voevodski et al., 2012), the guarantees for these algorithms do not immediately translate into an exact recovery guarantee, making it challenging to transform these approaches into hierarchical clustering algorithms. Moreover, the success of spectral clustering in practice suggests that an efficient spectral algorithm would also be very appealing. While there have been advances in this direction, the majority of these require the entire similarity matrix or graph to be known *a priori* (Frieze et al., 2004). Apart from (Shamir & Tishby, 2011), we know of no other spectral algorithm that optimizes the number of similarities needed.

3. Main Results

Before proceeding with our main results, we first clarify some notation and introduce a hierarchical clustering model that we will analyze. We refer to $\mathcal{A}$ as any flat clustering algorithm, which takes as parameters a dataset and a natural number k, indicating the number of clusters to produce. Throughout the paper, k will denote the number of clusters at any split, and we will assume that k is known and fixed across the hierarchy. We let n be the number of objects in a datasets and define s to be a parameter to our algorithms, influencing the number of measurements used by our algorithm, where smaller s implies fewer measurements. The parameter s reflects a tradeoff between the measurement overhead and the statistical accuracy of our algorithms; increasing s increases the

Algorithm 1 ACTIVECLUSTER$(\mathcal{A}, s, \{x_i\}_{i=1}^n, k)$

if $n \leq s$ **then return** $\{x_i\}_{i=1}^n$
Draw $S \subseteq \{x_i\}_{i=1}^n$ of size s uniformly at random.
$C'_1, \dots C'_k \leftarrow \mathcal{A}(S, k)$.
Set $C_1 \leftarrow C'_1, \dots C_k \leftarrow C'_k$.
for $x_i \in \{x_i\}_{i=1}^n \setminus S$ **do**
  $\forall j \in [k],\ \alpha_j \leftarrow \frac{1}{|C'_j|} \sum_{x_l \in C'_j} K(x_i, x_l)$.
  $C_{\text{argmax}_{j\in[k]}\alpha_j} \leftarrow C_{\text{argmax}_{j\in[k]}\alpha_j} \cup \{x_i\}$.
end for
return $\{C_j,$ ACTIVECLUSTER$(\mathcal{A}, s, C_j, k)\}_{j=1}^k$

robustness of our method, albeit at the cost of requiring more measurements. Finally, our algorithms employ an abstract, possibly noisy similarity function K, which can model both cases where similarities are measured directly and where they are computed via some kernel function based on observed object features.

Definition 1 *A **hierarchical clustering** $\mathcal{C}$ on objects $\{x_i\}_{i=1}^n$ is a collection of clusters such that $C_0 \triangleq \{x_i\}_{i=1}^n \in \mathcal{C}$ and for each $C_i, C_j \in \mathcal{C}$ either $C_i \subset C_j, C_j \subset C_i$ or $C_i \cap C_j = \emptyset$. For any cluster C, if $\exists C'$ with $C' \subset C$, then there exists a set $\{C_i\}_{i=1}^k$ of disjoint clusters such that $\bigcup_{i=1}^k C_i = C$.*

Every hierarchical clustering $\mathcal{C}$ has a parameter η that quantifies how balanced the clusters are at any split. Formally, $\eta \geq \max_{\text{splits}\{C_1,\dots,C_k\}} \frac{\max_i |C_i|}{\min_i |C_i|}$, where each split is a non-terminal cluster, partitioned into $\{C_i\}_{i=1}^k$. η upper bounds the ratio between the largest and smallest clusters sizes across all splits in $\mathcal{C}$. This type of balancedness parameter has been used in previous analyses of clustering algorithms (Eriksson et al., 2011; Balakrishnan et al., 2011), and it is common to assume that the clustering is not too unbalanced. For clarity of presentation, we will state our results assuming $\eta = O(1)$, although our proofs contain a precise dependence between the level of activeness s and η.

3.1. An Active Clustering Framework

Our primary contribution is the introduction of a novel framework for hierarchical clustering that is efficient both in terms of the number of similarities used and the algorithmic running time. To recover any single split of the hierarchy, we run a flat clustering algorithm $\mathcal{A}$ on a small subset of the data to compute a seed clustering of the dataset. Using this initial clustering, we place each remaining object into the seed cluster for which it is most similar on average. This results in a flat clustering of the entire dataset, using only similarities to the objects in the small subset.

By recursively applying this procedure to each cluster, we obtain a hierarchical clustering, using a small fraction of the similarities. In this recursive phase, we do not observe any measurements between clusters at the previous split, i.e. to partition C_j, we only observe similarities between objects in C_j. This results in an *active* algorithm that focuses its measurements to resolve the higher-resolution cluster structure.

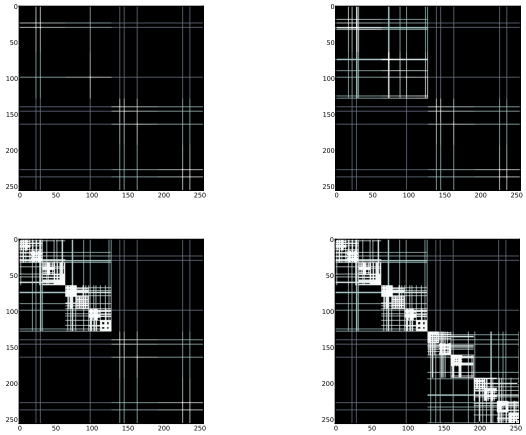


Figure 1. Sampling pattern of ACTIVECLUSTER.

Pseudocode for ACTIVECLUSTER is shown in Algorithm 1. As a demonstration, in Figure 1, we show the sampling pattern of ACTIVECLUSTER on the first, and second splits of a hierarchy (top row), in addition to the patterns at the end of the computation (bottom right). Only the similarities shown in white are needed. As is readily noticeable, the algorithm uses very few similarities yet can recover this hierarchy.

We are now ready to state our main theoretical contribution which characterizes ACTIVECLUSTER in terms of probability of success in recovering the true hierarchy (denoted $\mathcal{C}^\star$), measurement and runtime complexity. In order to make these guarantees we will need to place some mild restrictions on the similarity function K, which ensure that the similarities agree with the hierarchy (up to some random noise):

K1 For each $x_i \in C_j \in \mathcal{C}^\star$ and $j' \neq j$:

$$\min_{x_k \in C_j} \mathbb{E}[K(x_i, x_k)] - \max_{x_k \in C_{j'}} \mathbb{E}[K(x_i, x_k)] \geq \gamma > 0$$

where expectations are taken with respect to the possible noise on K.

K2 For each object $x_i \in C_j$, a set of M_j objects of size m_j drawn uniformly from cluster C_j satisfies:

$$\mathbb{P}\left(\min_{x_k \in C_j} E[K(x_i, x_k)] - \sum_{x_k \in M_j} \frac{K(x_i, x_k)}{m_j} > \epsilon \right) \leq 2\exp\{\frac{-2m_j\epsilon^2}{\sigma^2}\}$$

Where $\sigma^2 \geq 0$ parameterizes the noise on the similarity function K. Similarly, a set $M_{j'}$ of size $m_{j'}$ drawn uniformly from cluster $C_{j'}$ with $j' \neq j$ satisfies:

$$\mathbb{P}\left(\sum_{x_k \in M_{j'}} \frac{K(x_i, x_k)}{m_{j'}} - \max_{x_k \in C_{j'}} E[K(x_i, x_k)] > \epsilon\right) \leq 2\exp\{\frac{-2m_{j'}\epsilon^2}{\sigma^2}\}$$

K1 states that the similarity from an object x_i to its cluster should, in expectation, be larger than the the similarity from that object to any other cluster. This is related to the Tight-Clustering condition used in (Eriksson et al., 2011) and less stringent than earlier results which assume that within- and between-cluster similarities are constant and bounded in expectation (Rohe et al., 2010). Moreover, an assumption of this form seems necessary to ensure that even in expectation one could identify the clustering. Lastly, K2 enforces that within- and between-cluster similarities concentrate away from each other. This condition is satisfied, for example, if similarities are constant in expectation, perturbed with any subgaussian noise. We emphasize that K2 subsumes many of the assumptions of previous clustering analyses (for example (Balakrishnan et al., 2011; Rohe et al., 2010)).

Theorem 1 *Let $\{x_i\}_{i=1}^n$ be a dataset with true hierarchical clustering $\mathcal{C}^\star$, let K be a similarity function satisfing assumptions K1 and K2 and consider any flat clustering algorithm $\mathcal{A}$ with the following property:*

A1 *For any dataset $\{y_i\}_{i=1}^m$ with clustering $\mathcal{C}'^\star$ where K satisfies K1 and K2, $\mathcal{A}(\{y_i\}_{i=1}^m, k)$ returns the first split of $\mathcal{C}'^\star$ into k clusters with probability $\geq 1 - o(\frac{k}{c_1 e^m})$ for some constant $c_1 > 0$.*

Then ActiveCluster$(\mathcal{A}, s, \{x_i\}_{i=1}^n, k)$*:*

R1 *recovers all clusters of size at least s with probability $1 - o(n^2 e^{-cs})$, for some constant $c = c(\eta, \gamma)$. This probability of success is $1 - o(1)$ as long as:*

$$s \geq \max\left\{\begin{array}{l} \frac{1}{c_1}\log n \\ 4(1+\eta)^2 \log n \\ 24\frac{1+\eta}{\gamma^2}\log(4C_\eta kn) \end{array}\right\} = \Omega(\log(nk)) \quad (1)$$

R2 *uses $O(ns\log n)$ similarity measurements.*

R3 *runs in time $O(nA_s + ns\log n)$ where $\mathcal{A}$ on a datasets of size s runs in time $O(A_s)$.*

At a high level, the theorem says that the clustering guarantee for a flat, non-active algorithm, $\mathcal{A}$, can be translated into a hierarchical clustering guarantee for an active version of $\mathcal{A}$, and that this active algorithm enjoys significantly reduced measurement and runtime complexity. The only property needed by $\mathcal{A}$ is that it recovers a flat clustering with very high probability. While the probability of success seems strangely high, we will show that for a fairly intuitive model, a simple spectral clustering algorithm enjoys this kind of guarantee. Verifying that the model satisfies the conditions K1 and K2, immediately results in a guarantee for the active version of this spectral algorithm.

Before delving into the proof of the theorem, some remarks are in order. First, by plugging in the lower bound for s into the upper bound on the measurement complexity, we see that ActiveCluster needs $O(n\log(nk)\log n)$ similarities, which is considerably less than the $O(n^2)$ similarities required by a non-active algorithm. Second, at the lower bound for s, we see that unless $\mathcal{A}$ runs in exponential time, ActiveCluster runs in $\tilde{O}(n)$, which is significantly faster than *any* clustering algorithm that observes all of the similarities and must take $\Omega(n^2)$ time.

We now turn to the proof of R1. Due to space limitations, we defer many details and technical lemmas to the appendix. The proofs for R2 and R3 are straightforward, involving counting arguments on trees, and are also available in the appendix.

Proof for R1: We study the sampling, clustering and averaging phases of ActiveCluster in turn. In the sampling phase, we demonstrate that choosing s objects at random does not result in a highly unbalanced subset. Using bernoulli concentration inequalities and a union bound we show that the balance factor across all splits is at most $2\eta + 1$ with probability $\geq 1 - o(ne^{-c_{\eta,\gamma}s})$, which goes to 1 under Equation 1.

For the clustering phase, Lemma 3 (in the appendix) shows that the total number of calls to $\mathcal{A}$ is at most $\frac{n}{k-1}$ and each time we call $\mathcal{A}$ we have a probability of success $\geq 1 - o(\frac{k}{e^{c_1 s}})$ by assumption A1. It is now easy to see that the probability of $\mathcal{A}$ failing at any split is $o(n\exp\{-s\})$, which is $o(1)$ under Equation 1.

In the averaging phase, we need to show that for each split of the hierarchy and object x_i, the sample average within cluster similarity is larger than the sample average between cluster similarity. Under assumption K1 and K2, we know that these quantities concentrate away from each other. Via a union bound across all objects and all levels of the hierarchy, we can conclude that the probability of making a mistake in any averaging procedure is $O(nk\log n\exp\{\frac{-\gamma^2 s}{4(1+\eta)}\})$ which again goes to zero as long as s satisfies Equation 1.

Algorithm 2 SPECTRALCLUSTER(W)

Compute Laplacian $L = D - W$, $D_{ii} = \sum_{j=1}^{n} W_{ij}$
$v_2 \leftarrow$ smallest non-constant eigenvector of L.
$C_1 \leftarrow \{i : v_2(i) \geq 0\}$, $C_2 \leftarrow \{j : v_2(j) < 0\}$
output $\{C_1, C_2\}$.

3.2. Active Spectral Clustering

To make the guarantees in Theorem 1 more concrete, we show how to translate this into real guarantees for a specific subroutine algorithm $\mathcal{A}$. In this section, we turn a simple spectral algorithm (See pseudocode in Algorithm 2) into an active clustering algorithm, using the analysis from (Balakrishnan et al., 2011). The algorithm operates on hierarchically structured similarity matrices refered to as the **noisy Hierarchical Block Matrices** (again from (Balakrishnan et al., 2011)). These are defined as follows:

Definition 2 *A similarity matrix W is a* **noisy hierarchical block matrix** *(noisy HBM) if $W \triangleq A + R$ where A is ideal and R is a perturbation matrix:*

- *An* **ideal similarity matrix** *is characterized by ranges of off-block diagonal similarity values $[\alpha_\xi, \beta_\xi]$ for each cluster C_ξ such that if $x \in C_{\xi \circ L}$ and $y \in C_{\xi \circ R}$, where $C_{\xi \circ L}$ and $C_{\xi \circ R}$ are two sub-clusters of C_ξ at the next level in a binary hierarchy, then $\alpha_\xi \leq A_{x,y} \leq \beta_\xi$. Additionally, $\min\{\alpha_{\xi \circ R}, \alpha_{\xi \circ L}\} \geq \beta_\xi$.*
- *A symmetric $(n \times n)$ matrix R is a* **perturbation matrix** *with parameter σ if (a) $\mathbb{E}(R_{ij}) = 0$, (b) the entries of R are subgaussian, that is $\mathbb{E}(\exp(tR_{ij})) \leq \exp\left(\frac{\sigma^2 t^2}{2}\right)$ and (c) for each row i, $R_{i1}, \ldots R_{in}$ are independent.*

To apply Theorem 1, we need to verify that the assumption K1 and K2 are met and SPECTRALCLUSTER succeeds with exponentially high probability. Checking that these conditions hold as long as $\sigma = O(1)$ results in the following guarantees for ACTIVESPECTRAL, the active version of SPECTRALCLUSTER. Proof of this theorem is deferred to the appendix.

Theorem 2 *Let W be a noisy HBM with $\sigma = O(1)$ and $\eta = O(1)$. Then,* ACTIVESPECTRAL *succeeds in recovering all clusters of size s with probability $1 - o(1)$ as long as Equation 1 holds. Moreover,* ACTIVESPECTRAL *uses $O(ns \log n)$ measurements and runs in $O(ns^2 \log s + ns \log n)$ time.*

The results of this theorem quantify one extreme of the tradeoff between statistical robustness and measurement complexity for hierarchical spectral algorithms. In particular, it states that ACTIVESPECTRAL can tolerate a constant amount of noise while using only $O(n \log^2 n)$ measurements. At the other end of this spectrum is the result of Balakrishnan et. al. (2011), showing that using $O(n^2)$ measurements, one can tolerate noise that grows fairly rapidly with n. Varying s allows for interpolation between these two extremes.

3.3. Active k-means clustering

It is also possible to activize the popular k-means algorithm in our framework, but we cannot prove statistical performance guarantees since it is unknown whether k-means satisfies assumption A1. Activizing k-means helps illuminate the differences between observing similarities directly and computing similarities from directly observed object features. Conventionally, k-means fits into the latter framework. Here, the active version does not enjoy a reduced measurement complexity, because all of the objects must be observed, but it can reduce the number of similarity computations from nkT to $skT + (n-s)kT$, since the iterative subroutine runs on only s objects for T iterations. In cases where the similarity function is expensive to compute, such as edit distance, this can lead to gains in running time.

A less traditional way to use k-means is to represent each object as a n-dimensional vector of its similarity to each other object. Here, we can apply k-means to a $n \times n$ similarity matrix, much like we can apply SPECTRALCLUSTER and this algorithm can be activized using our framework. While we cannot develop theoretical guarantees for this algorithm, which we call ACTIVEKMEANS, our experiments demonstrate that it performs very well in practice.

3.4. Some Practical Considerations

Our algorithm as stated has some shortcomings that enable theoretical analysis but that are undesirable for practical applications. Specifically, the fact that k is known and constant across splits in the hierarchy, and the balancedness condition are both assumptions that are likely to be violated in any real-world setting. We therefore develop a variant of ACTIVESPECTRAL, called HEURSPEC, with several heuristics.

First, we employ the popular eigengap heuristic, in which the number of clusters k is chosen so that the gap in eigenvalues $\lambda_{k+1} - \lambda_k$ of the Laplacian is large. Secondly, we propose discarding all subsampled objects with low degree (when restricted to the sample) in the hopes of removing underrepresented clusters from the sample. In the averaging phase, if an object is not highly similar to any cluster represented in the sample, we create a new cluster for this object. We expect that in tandem, these two heuristics will help us recover small clusters. By comparing the performance of HEURSPEC to that of ACTIVESPECTRAL, we indirectly evaluate these heuristics.

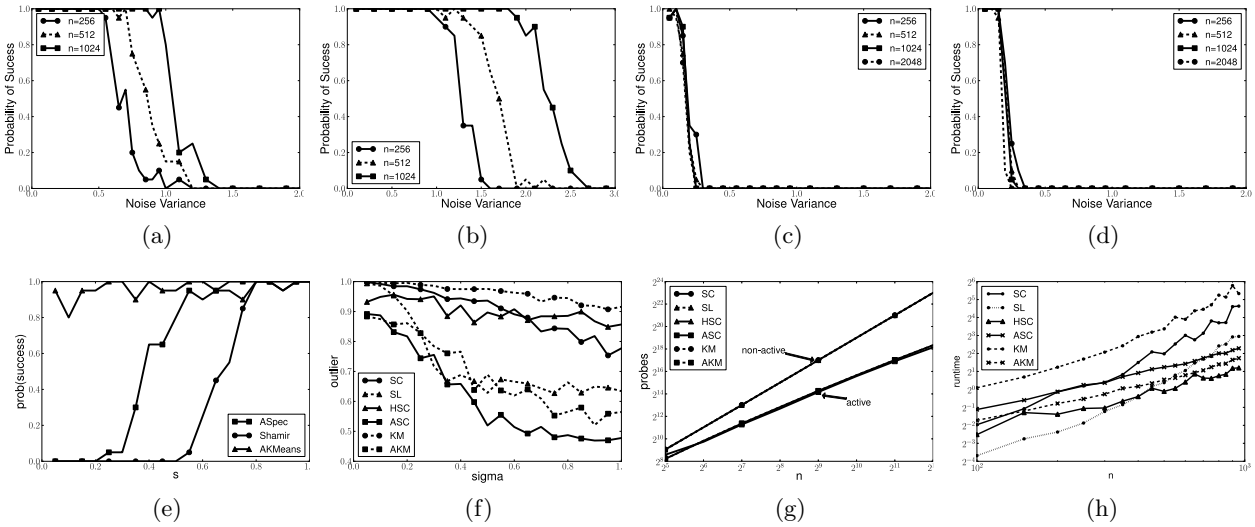


Figure 2. Simulation experiments. Top row: Noise thresholds for SpectralCluster, K-Means, ActiveSpectral, and ActiveKMeans with $s = \log(n)$ for active algorithms. Bottom row from left to right: probability of success as a function of s for $n = 256, \sigma = 0.75$, outlier fractions on noisy HBM, probing complexity, and runtime complexity.

4. Simulations

In this section we present experiments that verify our theoretical results. By Theorem 2, we expect ActiveSpectral to be robust to a constant amount of noise σ, meaning that it will recover all sufficiently large splits with high probability. In comparison, Balakrishnan et. al. (2011), show that SpectralCluster can tolerate noise growing with n. We constrast these guarantees by plotting the probability of successful recovery of the first split in a noisy HBM as a function of σ for different n in Figure 2. 2(a) demonstrates that indeed the noise tolerance of SpectralCluster grows with n while 2(c) demonstrates that ActiveSpectral enjoys constant noise tolerance. Figures 2(b) and 2(d) suggest that similar guarantees may hold for k-means and ActiveKMeans.

Our theory also predicts that increasing the activeness parameter improves the statistical performance of ActiveSpectral. To demonstrate this, we plot the probability of successful recovery of the first split of a noisy HBM of size $n = 256$ as a function of s for fixed noise variance. We compare three algorithms, ActiveSpectral, ActiveKMeans, and Algorithm 1 from (Shamir & Tishby, 2011), which subsamples entries of the similarity matrix. In theory, ActiveSpectral requires $\Omega(n \log n)$ total measurements to recover a single split, whereas (Shamir & Tishby, 2011) show that their algorithm requires $\Omega(n \log^{3/2} n)$. Figure 2(e) demonstrates that this improvement is also noticeable in practice. ActiveKMeans seems to enjoy an even more favorable dependence on s.

The simulations in Figures 2(a)-(e) only examine the ability of our algorithms to recover the first split of a hierarchy, while our theory predicts that all sufficiently large clusters can be reliably recovered. One way to measure this is the **outlier fraction** metric between the clustering returned by an algorithm and the true hierarchy (Eriksson et al., 2011). For any triplet of objects x_i, x_j, x_k we say that the two clusterings **agree** on this triplet if they both group the same pair of objects deeper in the hierarchy relative to the third object and disagree otherwise. The outlier fraction is simply the fraction of triplets for which the two clusterings agree.

In Figure 2(f), we plot the outlier fraction for six algorithms as a function of σ on the noisy HBM. The algorithms are: Hierarchical Spectral (SC), Single Linkage (SL), HeurSpec (HSC), ActiveSpectral (ASC), Hierarchical k-Means (KM), and ActiveKMeans (AKM). These experiments demonstrate that the non-active algorithms (except single linkage) are much more robust to noise than the corresponding active ones, as predicted by our theory, but also that the heuristics described in Section 3.4 have dramatic impact on performance.

Lastly, we verify the measurement and run time complexity guarantees for our active algorithms in comparison to the non-active versions. In Figure 2(g) and 2(h), we plot the number of measurements and running time as a function of n on a log-log plot for each algorithm. The three non-active algorithms have steeper slopes than the active ones, suggesting that they are polynomially more expensive in both cases.

5. Real World Experiments

To demonstrate the practical performance of the ActiveCluster framework, we apply our algorithms to

Algorithm	HKM	HRC	Probes	Time (s)
	SNP			
HEURSPEC	0.022	475	0.38	1350
ACTIVESPEC	0.019	19.1	0.13	450
ACTIVEKMEANS	0.018	12.5	0.12	420
k-means	0.0028	18.7	1	160
Spectral	0.0075	130	1	5660
	Phylo			
HEURSPEC	0.020	371	0.29	2500
ACTIVESPEC	0.012	22.9	0.071	600
ACTIVEKMEANS	0.012	25	0.071	555
k-means	0.0017	22.9	1	967
Spectral	0.0022	23.5	1	997
	NIPS			
HEURSPEC	0.0088	65.7	0.19	140
ACTIVESPEC	0.010	1.5	0.094	79.4
ACTIVEKMEANS	0.011	1.37	0.12	29
k-means	0.0017	1.66	1	723
Spectral	0.0033	6.30	1	26200
	RTW			
HEURSPEC	0.0079	18.1	0.41	419
ACTIVESPEC	0.0084	0.64	0.13	151
ACTIVEKMEANS	0.0073	0.485	0.22	70.9

Table 1. Real World Experiments

Algorithm	SNP	Phylo
HEURSPEC	0.596	0.878
ACTIVESPEC	0.374	0.971
ACTIVEKMEANS	0.383	0.94

Table 2. Outlier Fractions on Real Datasets

three real-world datasets and one additional synthetic dataset. The datasets are: The set of articles from NIPS volumes 0 through 12 from (Roweis, 2002), a subset of NPIC500 co-occurence data from the Read-the-Web project (Mitchell, 2009) which we call RTW, a SNP dataset from the HGDP (Pemberton et al., 2008), and a synthetic phylogeny dataset produced using `phyclust` (Chen, 2010). We refer the reader to the appendix for additional details on these datasets.

In the phylogeny and SNP datasets, we have access to a reference tree that can be used in our evaluation. In these cases we can report the outlier fraction, as we did in simulation. However, the other datasets lack such ground truth and without it, evaluating the performance of each algorithm is non-trivial. Indeed, there is no well-established metric for this sort of evaluation.

For this reason, we employ two distinct metrics to evaluate the quality of hierarchical clusterings. They are a hierarchical K-means objective (HKM) (Kauchak & Dasgupta, 2003) and an analogous hierarchical ratio-cut (HRC) objective, both of which are natural generalizations of the k-means and ratio cut objectives respectively, averaging across clusters, and removing small clusters as they bias the objectives. Formally,

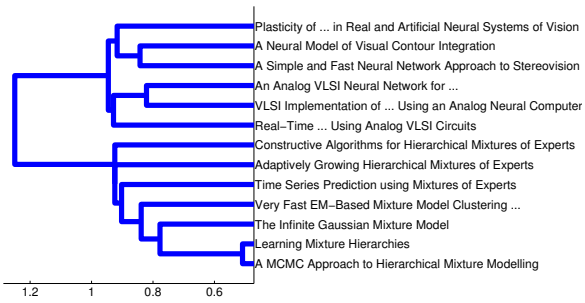


Figure 3. The ACTIVEKMEANS hierarchy restricted to a subset of NIPS articles.

let $\mathcal{C}$ be the hierarchical clustering and let $\bar{\mathcal{C}}$ be all of the clusters in $\mathcal{C}$ that are larger than $\log n$. For each $C \in \bar{\mathcal{C}}$ let x_C be the cluster center. Then:

$$\text{HKM}(\mathcal{C}) = \frac{1}{|\bar{\mathcal{C}}|} \sum_{C \in \bar{\mathcal{C}}} \frac{1}{|C|} \sum_{x_j \in C} \frac{x_j^T x_C}{||x_j|| ||x_C||} \quad (2)$$

$$\text{HRC}(\mathcal{C}) = \frac{1}{|\bar{\mathcal{C}}|} \sum_{C \in \bar{\mathcal{C}}} \sum_{C_k \subseteq C} \frac{K(C_k, C \setminus C_k)}{2|C_k|} \quad (3)$$

In Table 1 and 2, we record experimental results across the datasets for our algorithms. On the read-the-web dataset, we were unable to run the non-active algorithms. On the SNP and phylogeny datasets, we include computing similarities via edit distance in the running time of each algorithm, noting that computing all pairs takes 6500 and 15000 seconds respectively. The immediate observation is that these algorithms are extremly fast; on the SNP and phylogeny datasets where computing similarities is the bottleneck, activization leads to significant performance improvements. Moreover, the algorithms perform well by our metrics; they find clusterings that score well according to HKM and HRC, or that have reasonable agreement with the reference clustering[1].

We are also interested in more qualitatively understanding the performance of these algorithms. For the NIPS data, we manually collected a small subset of articles and visualized the hierarchy produced by ACTIVEKMEANS restricted to these objects. The hierarchy in Figure 3 is what one would expect on the subset, attesting to the performance ACTIVEKMEANS. On the other hand, this same kind of evaluation on the RTW data demonstrates that active algorithms do not perform well on this dataset, while the non-active algorithms do. We suspect this is caused by the RTW dataset consisting of many small clusters that do not get sampled by the ACTIVECLUSTER framework.

[1] The SNP dataset is a k-way hierarchy and our algorithms (apart from HEURSPEC) recover binary hierarchies that cannot have high agreement with the reference.

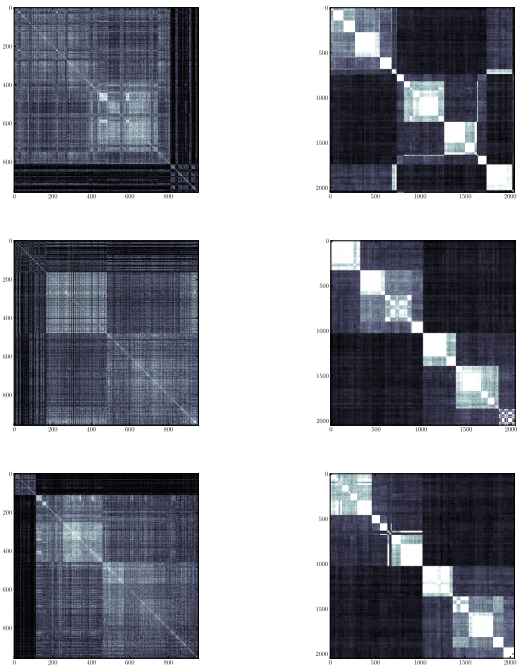


Figure 4. Heatmaps of permuted matrices for SNP (left) and Phylo (right). Algorithms are HeurSpec, ActiveSpectral, and ActiveKMeans from top to bottom.

For the SNP and phylogeny datasets, the permuted heatmaps are clear enough to be used in qualitative evaluations. These heatmaps are shown in Figure 4, and they suggest that all three active algorithms perform very well on these datasets. Heatmaps for the remaining datasets are less clear, but for completeness we include them in the appendix.

6. Discussion

Our results in this paper, showing that a family of active hierarchical clustering algorithms have strong performance guarantees, raise several interesting questions. We showed that ActiveSpectral enjoys reasonable statistical performance, but can other algorithms be activized while retaining statistical properties? Second, are there principled ways to circumvent a balancedness condition, even when objects are subsampled? Finally, is there a theoretically justified approach for estimating the number of clusters, k?

Another direction relates not toward clustering, but toward the recently popular matrix completion problem. On hierarchically structured matrices, our results imply that an active algorithm can recover high-rank (rank $n/\log n$) matrices using $O(n \log^2 n)$ similarities, an improvement over non-active approaches. Active algorithms may therefore yield impressive guarantees for matrix completion and related problems, and we hope to explore this direction in the future.

Acknowledgements

This research is supported in part by AFOSR under grant FA9550-10-1-0382 and NSF under grant IIS-1116458. AK is supported in part by a NSF Graduate Research Fellowship.

References

Achlioptas, D. and McSherry, F. Fast computation of low rank matrix approximations. In *STOC*, 2001.

Balakrishnan, S., Xu, M., Krishnamurthy, A., and Singh, A. Noise thresholds for spectral clustering. In *NIPS*, 2011.

Balcan, M. F. and Gupta, P. Robust hierarchical clustering. In *COLT*, 2010.

Charikar, M., O'Callaghan, L., and Panigrahy, Rina. Better streaming algorithms for clustering problems. In *STOC*, 2003.

Chen, W. C. *Phylogenetic Clustering with R package phyclust*, 2010. URL http://thirteen-01.stat.iastate.edu/snoweye/phyclust/.

de Silva, V. and Tenenbaum, J. B. Global versus local methods in nonlinear dimensionality reduction. In *NIPS*, 2002.

Eriksson, B., Dasarathy, G., Singh, A., and Nowak, R. Active Clustering: Robust and Efficient Hierarchical Clustering using Adaptively Selected Similarities. *CoRR*, 2011.

Frieze, A. M., Kannan, R., and Vempala, S. Fast monte-carlo algorithms for finding low-rank approximations. *J. ACM*, 2004.

Kauchak, D. and Dasgupta, S. An iterative improvement procedure for hierarchical clustering. In *NIPS*, 2003.

Mitchell, T. Noun phrases in context 500 dataset, 2009. URL http://www.cs.cmu.edu/~tom/10709_fall09/RTWdata.html.

Pemberton, T. J., Jakobsson, M., Conrad, D. F., Coop, G., Wall, J. D., Pritchard, J. K., Patel, P. I., and Rosenberg, N. A. Using population mixtures to optimize the utility of genomic databases: linkage disequilibrium and association study design in India. *Ann. Hum. Genet.*, 2008.

Rohe, K., Chatterjee, S., and Yu, B. Spectral Clustering and the High-Dimensional Stochastic Block Model. *Technical Report 791, Statistics Department, UC Berkeley*, 2010.

Roweis, S. NIPS Articles 1987-1999, 2002. URL http://cs.nyu.edu/~roweis/data.html.

Shamir, O. and Tishby, N. Spectral clustering on a budget. *AISTATS*, 2011.

Shindler, M., Meyerson, A., and Wong, A. Fast and accurate k-means for large datasets. In *NIPS*, 2011.

Voevodski, K., Balcan, M. F., Röglin, H., Teng, S., and Xia, Y. Active Clustering of Biological Sequences. *Journal of Machine Learning Research*, 2012.

A. Proof of Theorem 1

Before beginning the proof of the three claims in Theorem 1, we first state and prove two simple Lemmas bounding the number of splits and levels in a balanced hierarchy.

Lemma 3 *A k-way hierarchical clustering on n objects has at most $\frac{n}{k-1}$ splits.*

Proof A hierarchical clustering can be represented as a rooted tree $\mathcal{T}$, where each leaf is a singleton cluster and each internal node corresponds to a cluster containing all objects below this node. Every k-way hierarchy can be represented by a k-ary tree and compute the number of internal nodes in the k-ary tree exactly corresponds to the number of splits in the k-way hierarchy. Let $f(x)$ be the number of internal nodes in a k-ary tree with x leaves. It is easy to see that the recurrence $f(x) = f(x-k+1)+1$ holds for all $x \geq k$ and $f(x) = 1$ for all $0 < x < k$. Solving this recurrence, we see that $f(n) \leq \frac{n}{k-1}$ proving the Lemma.

Lemma 4 *Let η be the balance factor of the hierarchy and let l be the total number of levels in the hierarchy. Then:*

$$l \leq \frac{1}{\log\left(\frac{1+\eta}{\eta}\right)} \log n \leq C_\eta \log n \tag{4}$$

Proof Note that for any split, the larger of the two clusters has $\frac{\eta}{1+\eta}$ fraction of the nodes. After l levels, we want the largest cluster to have size at most 1:

$$\left(\frac{\eta}{1+\eta}\right)^l n \leq 1$$

Solving for l in this equation yields the result.

We now turn to proving the theorem. In the proof, we we'll define several failure events and first show that the algorithm succeeds if none of the failure events occur. We will then proceed to bound the probability of each of the failure events.

We will have to define some notation before proceeding. In the true hierarchy, we will denote each partition problem (or split) by $\mathcal{S}_1, \ldots, \mathcal{S}_{\frac{n}{k-1}}$ (recall that by Lemma 3, there are at most $\frac{n}{k-1}$ of these). Moreover, each split except for the split at the root of the heirarchy has a parent split, which is the clustering problem directly above it in the hierarchy. For a split $\mathcal{S}_i$, denote it's parent split by $\mathcal{S}_{\pi(i)}$ so that $\pi(i)$ is the index of i's parent in the hierarchy.

For each split $\mathcal{S}_i$, we have three types of error events: a subsampling error event, a error event on the correctness of the algorithms $\mathcal{A}$ and an error event on the averaging phase. In the subsamping phase, we will report an error, if the subsampled balance factor for the clustering problem at split i, $\hat{\eta}$ is larger than $2\eta + 1$ (we will precisely define $\hat{\eta}$ subsequently). If $\hat{\eta} \leq 2\eta+1$, then since $\eta = O(1)$, we will obtain that $\hat{\eta} = O(1)$ so that $\mathcal{A}$ has some hope of successfully clustering the subsample. Formally, these error events are defined as follows:

$$S_i = \{\text{at split } \mathcal{S}_i,\ \hat{\eta} \geq 2\eta + 1\} \tag{5}$$

$$A_i = \{\text{Algorithm } \mathcal{A} \text{ fails at split } \mathcal{S}_i\} \tag{6}$$

$$V_i = \{\text{Averaging fails at level } \mathcal{S}_i\} \tag{7}$$

With these definition, it is easy to see that:

$$\mathbb{P}[\text{failure}] \leq \mathbb{P}[\bigcup_{i=1}^{n} S_i \cup A_i \cup V_i] \tag{8}$$

Or the algorithm only fails if one of these error events occurs. Note that this is an upper bound because the algorithm may still succeed, even if the event S_i occurs for example. At this point, one could use a union bound to decompose this further into a sum of failure probabilities, but it is challenging to bound each failure probability independently of the other events. Instead, we will appeal to the following lemma to upper bound the right hand side via a more suitable decomposition.

Lemma 5 *Let $B_0, B_1, \ldots, B_t$ be events in some measureable space. Then:*

$$\mathbb{P}[\bigcup_{i=0}^{t} B_i] \leq \mathbb{P}[B_0] + \sum_{i=1}^{t} \mathbb{P}[B_i | \neg B_0, \ldots, \neg B_{i-1}] \tag{9}$$

Proof First, the following identity is fairly obvious:

$$\bigcup_{i=0}^{t} B_i = \bigcup_{i=0}^{t} \left(B_i \cap \bigcap_{j=0}^{i} \neg B_j \right)$$

So that the probability mass of each set is also equivalent. Now, using a union bound and the chain rule:

$$\mathbb{P}\left[\bigcup_{i=0}^{t} B_i\right] \leq \sum_{i=0}^{t} \mathbb{P}\left[B_i \bigcap_{j=0}^{i} \neg B_j\right]$$

$$= \sum_{i=0}^{t} \mathbb{P}\left[\bigcap_{j=0}^{i} \neg B_j\right] \mathbb{P}\left[B_i | \bigcap_{j=0}^{i} \neg B_j\right]$$

$$\leq \sum_{i=0}^{t} \mathbb{P}[B_i | \neg B_0, \ldots, \neg B_{i-1}]$$

Where in the last step we used that probabilities must be upper bounded by 1. This proves the lemma.

Using Lemma 5, we can decompose the right hand side of Equation 8 as:

$$\mathbb{P}[S_1] + \mathbb{P}[A_1|\neg S_1] + \mathbb{P}[V_1|\neg S_1, \neg A_1] + \sum_{i=2}^{\frac{n}{k-1}} \mathbb{P}[S_i|\neg S_0, \neg A_0, \neg V_0, \ldots, \neg S_{i-1}, \neg A_{i-1}, \neg V_{i-1}] + \mathbb{P}[A_i|\neg S_0, \neg A_0, \neg V_0, \ldots, \neg S_{i-1}, \neg A_{i-1}, \neg V_{i-1}, \neg S_i] + \mathbb{P}[V_i|\neg S_0, \neg A_0, \neg V_0, \ldots, \neg S_{i-1}, \neg A_{i-1}, \neg V_{i-1}, \neg S_i, \neg A_i]$$

Next we exploit independence of events to simplify each of the expressions. In particular we have the following independence assertions: each subsampling phase is independent of all previous error events, conditioned on the successful recovery of the corresponding parent clustering, each execution of the algorithm succeeds (or fails) independent of every previous failure event, conditioned on the success of subsampling at that split, and each averaging phase succeeds (or fails) independent of every previous failure event, conditioned on the success of sampling and the black-box algorithm at that split. With this assertions we can reduce the above expression to:

$$\begin{aligned} & \mathbb{P}[S_1] + \\ \textstyle\sum_{i=2}^{\frac{n}{k-1}} \quad & \mathbb{P}[S_i|\neg A_{\pi(i)}, \neg V_{\pi(i)}] + \\ \textstyle\sum_{i=1}^{\frac{n}{k-1}} \quad & \mathbb{P}[A_i|\neg S_i] + \\ \textstyle\sum_{i=1}^{\frac{n}{k-1}} \quad & \mathbb{P}[V_i|\neg S_i, \neg A_i] \end{aligned}$$

In the subsequent sections, we will bound each of these conditional probabilities. By showing that the sum of these conditionals is small, we will arrive an an upper bound on the failure probability of our algorithm.

A.1. The Subsampling Phase

Here we bound the probability of the event S_i, conditioned on the successful recovery of $\mathcal{S}_i$'s parent cluster. We need to demonstrate that the balance factor $\hat{\eta}$, restricted to the subsample, is upper bounded by $2\eta + 1$ after subsampling s objects, and moreover we have to do this across all splits of the hierarchy. Consider one split at first; we have k clusters $C_1, \ldots, C_k$, and define the random variables $X_1, \ldots, X_s \in [k]$ which indicates cluster membership of the ith draw. Define the estimators $\hat{c}_j = \sum_{i=1}^{n} \delta[X_i = j]$, so that $\mathbb{E}[\frac{\hat{c}_j}{s}] = |C_j|/n$. In both the cases of sampling with and without replacement, we can apply Hoeffding's inequality and union bound over the cluster C_i to obtain:

$$\forall j. \mathbb{P}\left(|\frac{1}{s}\hat{c}_j - \frac{|C_j|}{n}| > \epsilon \right) \leq 2k \exp\{-2s\epsilon^2\}$$

Using Lemma 3, and a union bound over the events S_i, and inverting the concentration inequality, we have that with probability $1-\delta_1$, for all splits $\mathcal{S}_i$ and cluster C_J:

$$\left| \frac{1}{s}\hat{c}_j - \frac{|C_j|}{|\mathcal{S}_i|} \right| \leq \sqrt{\frac{\log(2nk/(k-1)) + \log(1/\delta_1)}{2s}} \quad (10)$$

$$\leq \sqrt{\frac{\log 4n + \log(1/\delta_1)}{2s}} \quad (11)$$

where $|\mathcal{S}_i|$ is the total number of objects to be clustered at split $\mathcal{S}_i$. Using the fact that the hierarchy has balance factor η, which holds here since we are conditioning on successful recovery of the parent cluster at each step, we obtain:

$$\frac{1}{s}\max_j \hat{c}_j \leq \frac{\eta}{1+\eta} + \sqrt{\frac{\log 4n + \log(1/\delta_1)}{2s}} \quad (12)$$

$$\frac{1}{s}\min_j \hat{c}_j \geq \frac{1}{1+\eta} - \sqrt{\frac{\log 4n + \log(1/\delta_1)}{2s}} \quad (13)$$

And the modified balance factor $\hat{\eta}$ is the ratio of these two quantities. Setting $\delta_1 = 4n\exp\{\frac{-s}{2(1+\eta)^2}\}$ gives that $\hat{\eta} \leq 2\eta + 1 = O(1)$ so that the event S_i does not hold, across all splits $\mathcal{S}_i$. The first term in the max in Equation 1 comes from the fact that for $\delta_1 = o(1)$ we need $s \geq 4(1+\eta)^2 \log(4n)$. With this setting of s, we obtain that:

$$\mathbb{P}[S_1] + \sum_{i=2}^{\frac{n}{k-1}} \mathbb{P}[S_i|\neg A_{\pi(i)}, \neg V_{\pi(i)}] \leq \delta_1 = o(1)$$

A.2. The Clustering Phase

In the clustering phase, we simply need to add up the probabilities of failure for all executions of the algorithm $\mathcal{A}$, conditioned on the fact that the subsampling phase for this split yielded a constant balance factor. By assumption $\mathcal{A}$ fails on an input of size s with probability $o(kc_1e^{-s})$. With a union bound across all splits, the probability of any execution of $\mathcal{A}$ failing is $o(\frac{nc_1}{k-1}ke^{-s}) = o(nc_1 \exp -s)$ (where we used Lemma 3). As long as $s = \log n$, the probability of failure is $o(1)$. Thus we have that for $s \geq \log n$:

$$\sum_{i=1}^{\frac{n}{k-1}} \mathbb{P}[A_i|\neg S_i] = o(1)$$

A.3. The Averaging Phase

Here our goal is to show that as long as subsampling and the subroutine clustering algorithm succeeded, then the averaging phase will also succeed with high

probability. The guarantees for the averaging phase follow from assumption K2. In order to ensure that we place every object in its correct cluster, we require:

$$\frac{1}{\hat{c}_j}\sum_{x_k\in\hat{C}_j} K(x_i,x_j) > \frac{1}{\hat{c}_{j'}}\sum_{x_j\in\hat{C}_{j'}} K(x_i,x_j) \tag{14}$$

for all $x_i \in C_j$, for all $j' \neq j$ and across all splits. Here we say that $\hat{C}_j = \{x_j \in C_j\} \cap \{x_j \in S\}$ and $\hat{c}_j = |\hat{C}_j|$ for all j. By assumption K2 and a union bound, we have that:

$$\frac{1}{\hat{c}_j}\sum_{x_k\in\hat{C}_j} K(x_i,x_k) \geq \min_{x_k\in C_j}\mathbb{E}[K(x_i,x_k)] - \sqrt{\frac{\log(C_\eta n)+\log\log n+\log(4/\delta_3)}{2\hat{c}_j}}$$

$$\frac{1}{\hat{c}_{j'}}\sum_{x_k\in\hat{C}_{j'}} K(x_i,x_k) \leq \max_{x_k\in C_{j'}}\mathbb{E}[K(x_i,x_k)] + \sqrt{\frac{\log(C_\eta kn)+\log\log n+\log(4/\delta_3)}{2\hat{c}_{j'}}}$$

For the within cluster similarities we union bounded over each of the $C_\eta \log n$ levels, because each object belongs to only one cluster per level. For between cluster similarities, we union bounded over the $C_\eta \log n$ levels and the $k-1 \leq k$ clusters that we will compare to for each object x_i. Both equations hold with probability $1-\delta_3$, because we used $\delta_3/2$ as the individual probability of failure. Note also that we replace M_j in assumption K2 with the sets $\hat{C}_j$ and $\hat{C}_{j'}$; because those sets are chosen uniformly at random, we can make this replacement.

Replacing $\hat{c}_j$ and $\hat{c}_{j'}$ both with the lower bound on the subsampled cluster sizes, arising from the bound on $\hat{\eta}$, and observing that if the lower bound for the first expression is larger than the upper bound for the second expression, we will make no mistakes at all splits of the hierarchy, we obtain the following lower bound on γ, defined in assumption K1:

$$\gamma > 2\sqrt{2\frac{+\eta}{s}(\log(C_\eta kn)+\log\log n+\log(4/\delta_3))} \tag{15}$$

Solving this equation for δ_3 gives:

$$\delta \leq 4C_\eta nk\log n\exp\left\{\frac{-\gamma^2 s}{4(1+\eta)}\right\} \tag{16}$$

which goes to zero as long as s satisfies Equation 1. Thus we just showed that:

$$\mathbb{P}[V_1|\neg S_1,\neg A_1] + \sum_{i=2}^{\frac{n}{k-1}}\mathbb{P}[V_i|\neg S_i,\neg A_i] = o(1)$$

A.4. Wrapping Up

Adding up the results from each phase of the algorithm we have that the total failure probability of the algorithm is $o(1)$ as long as s satisfies Equation 1. This concludes the proof.

B. Proof of Theorem 2

Theorem 2 is a direct application of Theorem 1. In order to apply that theorem, we must first verify that the noisy HBM family satisfies the assumptions D1, D2, K1, K2, and that the algorithm SpectralClustersatisfies assumption A1. Assumptions D1 and D2 hold by definition of the noisy HBM and the assumption that $\eta = O(1)$. Assumption K1 holds with $\gamma \triangleq \min_{\text{clusters } C_\xi}\min\{\alpha_{\xi\circ L},\alpha_{\xi\circ R}\} - \beta_\xi$. Assumption K2 holds with σ^2 exactly corresponding to the noise variance in the subgaussian perturbation, and this follows from the fact that subgaussian random variables enjoy exponential concentration.

To check that A1 is satisfied by SpectralCluster we follow the proof in Balakrishnan et. al. Following their proof strategy exactly, except keeping track of the probabilities of failure, we see that as long as:

$$\sigma = o\left(\min\left(\kappa^{\star 5}\sqrt{\frac{m}{\log m+\log(c/\delta)}}, \kappa^{\star 4}\sqrt[4]{\frac{m}{\log m+\log c/\delta}}\right)\right)$$

Then SpectralClusterwill succeed on a $m\times m$ matrix with probability $\geq 1-\delta$. Under our assumptions, we have $\sigma = O(1)$ and with this in mind, we can solve for δ to obtain:

$$\delta \leq \max\left(sc\exp\left(-\frac{\kappa^{\star 10}s}{C^2}\right), sc\exp\left(-\frac{\kappa^{\star 16}s}{C^4}\right)\right) \tag{17}$$

C. Datasets

The NIPS dataset consists of 1740 machine learning research articles from Neural Information Processing Systems Volumes 0-12. Each article was converted into a TF-IDF vector and pairs of vectors were compared using cosine similarity.

The RTW data is a subsampled version of the NPIC500 co-occurence dataset. It originally consisted of 88k noun-phrases and 99k contexts with NP-context co-occurrence information. We further downsampled to 2000 NPs and used TF-IDF and cosine similarity to construct a noun-phrase by noun-phrase co-occurence matrix.

The SNP dataset consists base pair information at

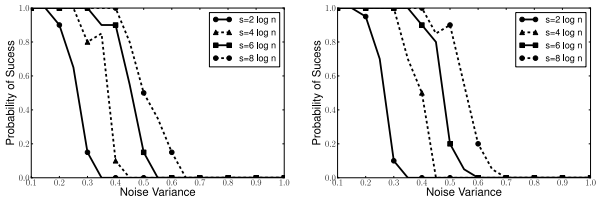


Figure 5. Probability of success curves for ActiveSpectral (left) and ActiveKMeans (right) demonstrating the tradeoff between measurement complexity and robustness.

2810 loci for 957 individuals. The dataset is hierarchically annototated into three levels, where each individual is assigned a population, country of origin, and continent. Each individual has two haplotype sequences and we arbitrarily chose the haplotype inherited from the mother. We measure similarity using edit distance. Note that in this case, computing all pairwise similarities is computational intensive; it took over 1 hour for this computation.

The phylogeny dataset is a synthetic phylogeny generated by the `phyclust` R package. It consists of 2048 genetic sequences, each consisting of 2000 base pairs. `phyclust` also generates a reference phylogeny that serves as ground truth. As with the SNP data, we measured similarity using edit distance. Computing all pairs of similarities took over 4 hours.

D. Additional Experimental Results

In simulation, we are also interested in understanding how the robustness of our algorithms scale with the sampling parameter s. To evaluate this empirically, we plot the probability of successful recovery of the first split of a noisy HBM for both ActiveSpectral and ActiveKMeans as a function of σ for different values of s. These results are in Figure 5. Here we took $n = 1024$. As expected, increasing the sampling parameter results in a better tolerance to noise, quantifying the tradeoff between measurement complexity and statistical robustness.

As in (Shamir & Tishby, 2011), we can look at the misclustering rate of our algorithms on several synthetic datasets as a function of the activeness parameter. In Table 3, we simulated 4 different datasets each with two clusters, shown in the top row and looked at the misclustering rate of three active algorithms as a function of the activeness parameter. We specifically look at ActiveSpectral, ActiveKMeans, and the algorithm from (Shamir & Tishby, 2011) that subsamples entries of the similarity matrix, rather than objects.

First, in all of the examples, both ActiveSpec-

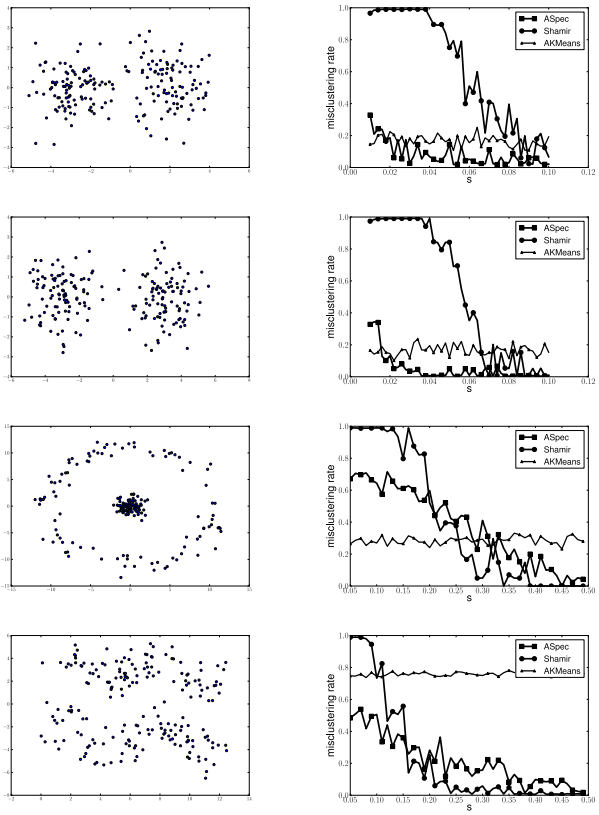
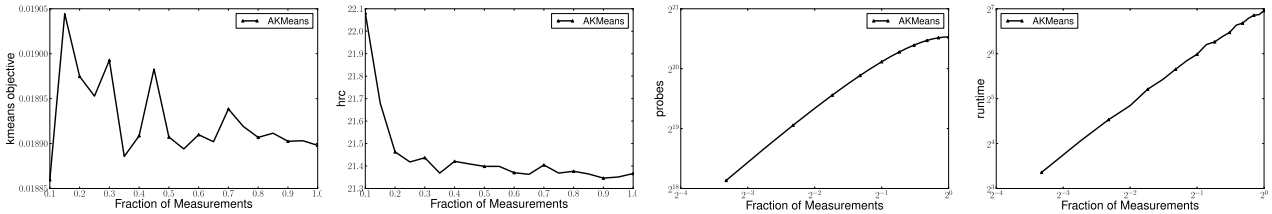


Table 3. Misclustering rate for three active algorithms as a function of s.



(a) (b) (c) (d)

Figure 6. Effect of varying s on NIPS dataset, using the ActiveKMeans algorithm.

tral and ActiveKMeans consistently outperform the algorithm from (Shamir & Tishby, 2011), which we call ST. On each dataset, both ActiveSpectral and that algorithm improve as we observe more similarities. For the algorithm from (Shamir & Tishby, 2011), the threshold for correctness seems to be much sharper than for ActiveSpectral, but we remark that ActiveSpectral consistently outperforms their algorithm for small s. We also note that ActiveKMeans does not seem to improve with n. While we make no predictions about the performance of ActiveKMeans, we suspect that it is possible to make a strong guarantee for k-means that translates into a correctness guarantee for ActiveKMeans. However, the model for which that guarantee can be made may not be satisfied by these datasets, resulting in the poor performance in Figure 3. In particular, the similarity matrices that result from these datasets (we used the Heat Kernel as a similarity function), does not satisfy assumptions K1 and K2, so there is no reason to suspect any of the algorithms would perform well.

To further demonstrate the tradeoff between statistical accuracy and measurement and runtime complexity, in Figure 6, we plot the k-means objective, ratio-cut objective, measurement overhead, and running time for the ActiveKMeans algorithm on recovering the first split of the NIPS dataset as a function of the parameter s. The four figures clearly show that as we increase s we find better clusterings under both metrics, but at the cost of using more measurements and incurring more computational overhead.

For completeness, we also include heatmaps from running our three active algorithms on the NIPS and RTW-2000 datasets. These are in Figure 7.

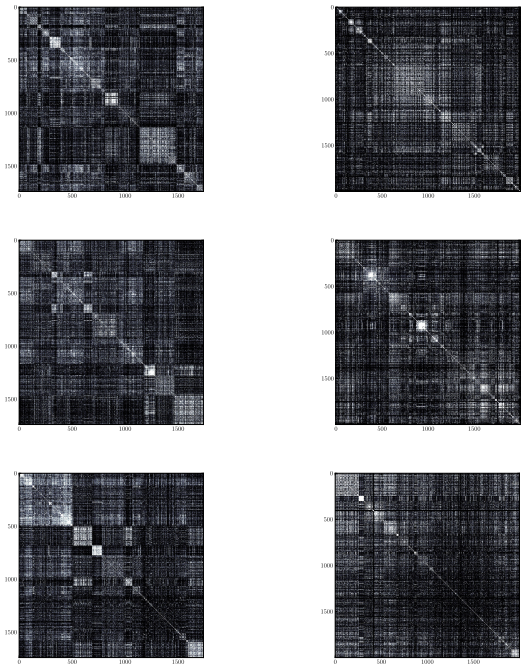


Figure 7. Heatmaps for NIPS (left) and RTW (right) datasets. Algorithms are HeurSpec, ActiveSpectral, and ActiveKMeans from top to bottom.