A Discriminative Framework for Clustering via Similarity Functions

Maria-Florina Balcan[*]
Dept. of Computer Science
Carnegie Mellon University
Pittsburgh, PA
ninamf@cs.cmu.edu

Avrim Blum[*]
Dept. of Computer Science
Carnegie Mellon University
Pittsburgh, PA
avrim@cs.cmu.edu

Santosh Vempala[†]
College of Computing
Georgia Inst. of Technology
Atlanta, GA
vempala@cc.gatech.edu

ABSTRACT

Problems of clustering data from pairwise similarity information are ubiquitous in Computer Science. Theoretical treatments typically view the similarity information as ground-truth and then design algorithms to (approximately) optimize various graph-based objective functions. However, in most applications, this similarity information is merely based on some heuristic; the ground truth is really the unknown correct clustering of the data points and the real goal is to achieve low error on the data. In this work, we develop a theoretical approach to clustering from this perspective. In particular, motivated by recent work in learning theory that asks "what natural properties of a similarity (or kernel) function are sufficient to be able to learn well?" we ask "what natural properties of a similarity function are sufficient to be able to *cluster* well?"

To study this question we develop a theoretical framework that can be viewed as an analog of the PAC learning model for clustering, where the object of study, rather than being a concept class, is a class of (concept, similarity function) pairs, or equivalently, a *property* the similarity function should satisfy with respect to the ground truth clustering. We then analyze both algorithmic and information theoretic issues in our model. While quite strong properties are needed if the goal is to produce a single approximately-correct clustering, we find that a number of reasonable properties are sufficient under two natural relaxations: (a) list clustering: analogous to the notion of list-decoding, the algorithm can produce a small list of clusterings (which a user can select from) and (b) hierarchical clustering: the algorithm's goal is to produce a hierarchy such that desired clustering is some pruning of this tree (which a user could navigate). We develop a notion of the *clustering complexity* of a given property (analogous to notions of *capacity* in learning theory), that characterizes its information-theoretic usefulness for clustering. We analyze this quantity for several natural game-theoretic and learning-theoretic properties, as well as design new efficient algorithms that are able to take advantage of them. Our algorithms for hierarchical clustering combine recent learning-theoretic approaches with linkage-style methods. We also show how our algorithms can be extended to the inductive case, i.e., by using just a constant-sized sample, as in property testing. The analysis here uses regularity-type results of [20] and [3].

Categories and Subject Descriptors: F.2.0 [Analysis of Algorithms and Problem Complexity]: General

General Terms: Algorithms, Theory

Keywords: Clustering, Similarity Functions, Learning.

1. INTRODUCTION

Clustering is an important problem in the analysis and exploration of data. It has a wide range of applications in data mining, computer vision and graphics, and gene analysis. It has many variants and formulations and it has been extensively studied in many different communities.

In the Algorithms literature, clustering is typically studied by posing some objective function, such as k-median, min-sum or k-means, and then developing algorithms for approximately optimizing this objective given a data set represented as a weighted graph [12, 24, 22]. That is, the graph is viewed as "ground truth" and then the goal is to design algorithms to optimize various objectives over this graph. However, for most clustering problems such as clustering documents by topic or clustering web-search results by category, ground truth is really the unknown true topic or true category of each object. The construction of the weighted graph is just done using some heuristic: e.g., cosine-similarity for clustering documents or a Smith-Waterman score in computational biology. In all these settings, the goal is really to produce a clustering that gets the data correct. Alternatively, methods developed both in the algorithms and in the machine learning literature for learning mixtures of distributions [1, 5, 23, 36, 17, 15] explicitly have a notion of ground-truth clusters which they aim to recover. However, such methods are based on very strong assumptions: they require an embedding of the objects into R^n such that the clusters can be viewed as distributions with very specific properties (e.g., Gaussian or log-concave). In many real-world situations (e.g., clustering web-search results by topic, where different users might have different notions of what a "topic" is) we can only expect a domain expert to provide a notion of similarity between objects that is related in some reasonable ways to the desired clustering goal, and not necessarily an embedding with such strong properties.

In this work, we develop a theoretical study of the clustering problem from this perspective. In particular, motivated by work on learning with kernel functions that asks "what natural properties

[*] Supported in part by the National Science Foundation under grant CCF-0514922, and by a Google Research Grant.

[†] Supported in part by the National Science Foundation under grant CCF-0721503, and by a Raytheon fellowship.

STOC'08, May 17–20, 2008, Victoria, British Columbia, Canada.
Copyright 2008 ACM 978-1-60558-047-0/08/05 ...$5.00.

of a given kernel (or similarity) function $\mathcal{K}$ are sufficient to allow one to *learn* well?" [6, 7, 31, 29, 21] we ask the question "what natural properties of a pairwise similarity function are sufficient to allow one to *cluster* well?" To study this question we develop a theoretical framework which can be thought of as a discriminative (PAC style) model for clustering, though the basic object of study, rather than a concept class, is a *property* of the similarity function $\mathcal{K}$ in relation to the target concept, or equivalently a set of (concept, similarity function) pairs.

The main difficulty that appears when phrasing the problem in this general way is that if one defines success as outputting *a single clustering* that closely approximates the correct clustering, then one needs to assume very strong conditions on the similarity function. For example, if the function provided by our expert is extremely good, say $\mathcal{K}(x, y) > 1/2$ for all pairs x and y that should be in the same cluster, and $\mathcal{K}(x, y) < 1/2$ for all pairs x and y that should be in different clusters, then we could just use it to recover the clusters in a trivial way.[1] However, if we just slightly weaken this condition to simply require that all points x are more similar to all points y from their own cluster than to any points y from any other clusters, then this is no longer sufficient to uniquely identify even a good approximation to the correct answer. For instance, in the example in Figure 1, there are multiple clusterings consistent with this property. Even if one is told the correct clustering has 3 clusters, there is no way for an algorithm to tell which of the two (very different) possible solutions is correct. In fact, results of Kleinberg [25] can be viewed as effectively ruling out a broad class of scale-invariant properties such as this one as being sufficient for producing the correct answer.

In our work we overcome this problem by considering two relaxations of the clustering objective that are natural for many clustering applications. The first is as in list-decoding to allow the algorithm to produce a small *list* of clusterings such that at least one of them has low error. The second is instead to allow the clustering algorithm to produce a *tree* (a hierarchical clustering) such that the correct answer is approximately some pruning of this tree. For instance, the example in Figure 1 has a natural hierarchical decomposition of this form. Both relaxed objectives make sense for settings in which we imagine the output being fed to a user who will then decide what she likes best. For example, with the tree relaxation, we allow the clustering algorithm to effectively say: "I wasn't sure how specific you wanted to be, so if any of these clusters are too broad, just click and I will split it for you." We then show that with these relaxations, a number of interesting, natural learning-theoretic and game-theoretic properties can be defined that each are sufficient to allow an algorithm to cluster well.

At the high level, our framework has two goals. The first is to provide advice about what type of *algorithms* to use given certain beliefs about the relation of the similarity function to the clustering task. That is, if a domain expert handed us a similarity function that they believed satisfied a certain natural property with respect to the true clustering, what algorithm would be most appropriate to use? The second goal is providing advice to the *designer* of a similarity function for a given clustering task (such as clustering web-pages by topic). That is, if a domain expert is trying up to come up with a similarity measure, what properties should they aim for?

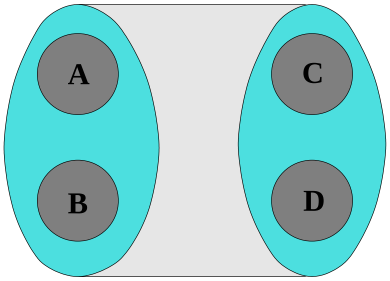

Figure 1: Data lies in four regions A, B, C, D (e.g., think of as documents on baseball, football, TCS, and AI). Suppose that $\mathcal{K}(x, y) = 1$ if x and y belong to the same region, $\mathcal{K}(x, y) = 1/2$ if $x \in A$ and $y \in B$ or if $x \in C$ and $y \in D$, and $\mathcal{K}(x, y) = 0$ otherwise. Even assuming that all points are more similar to other points in their own cluster than to any point in any other cluster, there are still multiple consistent clusterings, including two consistent 3-clusterings (($A \cup B$, C, D) or (A, B, $C \cup D$)). However, there is a single hierarchical decomposition such that any consistent clustering is a pruning of this tree.

1.1 Perspective

The standard approach in theoretical computer science to clustering is to choose some objective function (e.g., k-median) and then to develop algorithms that approximately optimize that objective [12, 24, 22, 18]. If the true goal is to achieve low error with respect to an underlying correct clustering (e.g., a user's desired clustering of search results by topic), however, then one can view this as implicitly making the strong assumption that not only does the correct clustering have a good objective value, but also that all clusterings that approximately optimize the objective must be close to the correct clustering as well. In this work, we instead explicitly consider the goal of producing a clustering of low error and then ask what natural properties of the similarity function in relation to the target clustering are sufficient to allow an algorithm to do well.

In this respect we are closer to work done in the area of clustering or learning with mixture models [1, 5, 23, 36, 17]. That work, like ours, has an explicit notion of a correct ground-truth clustering of the data points and to some extent can be viewed as addressing the question of what properties of an *embedding of data into* R^n would be sufficient for an algorithm to cluster well. However, unlike our focus, the types of assumptions made are distributional and in that sense are much more stringent than the types of properties we will be considering. This is similarly the case with work on planted partitions in graphs [4, 28, 16]. Abstractly speaking, this view of clustering parallels the *generative* classification setting [19], while the framework we propose parallels the *discriminative* classification setting (i.e. the PAC model of Valiant [34] and the Statistical Learning Theory framework of Vapnik [35]).

In the PAC model for learning [34], the basic object of study is the *concept class*, and one asks what natural classes are efficiently learnable and by what algorithms. In our setting, the basic object of study is *property*, which can be viewed as a set of (concept, similarity function) pairs, i.e., the pairs for which the target concept and similarity function satisfy the desired relation. As with the PAC model for learning, we then ask what natural properties are sufficient to efficiently cluster well (in either the tree or list models) and by what algorithms.

1.2 Our Results

We provide a PAC-style framework for analyzing what properties of a similarity function are sufficient to allow one to cluster

[1] Correlation Clustering can be viewed as a relaxation that allows *some* pairs to fail to satisfy this condition, and the algorithms of [10, 13, 33, 2] show this is sufficient to cluster well if the number of pairs that fail is small. *Planted partition* models [4, 28, 16] allow for many failures so long as they occur at *random*. We will be interested in much more drastic relaxations, however.

well under the above two relaxations (list and tree) of the clustering objective. We analyze both algorithmic and information theoretic questions in our model and provide results for several natural game-theoretic and learning-theoretic properties. Specifically:

- We consider a family of stability-based properties, showing that a natural generalization of the "stable marriage" property is sufficient to produce a hierarchical clustering. (The property is that no two subsets $A \subset C$, $A' \subset C'$ of clusters $C \neq C'$ in the correct clustering are both more similar on average to each other than to the rest of their own clusters.) Moreover, a significantly weaker notion of stability is also sufficient to produce a hierarchical clustering, but requires a more involved algorithm.

- We show that a weaker "average-attraction" property (which is provably not enough to produce a single correct hierarchical clustering) is sufficient to produce a small list of clusterings, and give generalizations to even weaker conditions that generalize the notion of large-margin kernel functions.

- We define the *clustering complexity* of a given property (the minimum possible list length that can be guaranteed by any algorithm) and provide both upper and lower bounds for the properties we consider. This notion is analogous to notions of capacity in classification [11, 19, 35] and it provides a formal measure of the inherent usefulness of a given property.

- We also show that properties implicitly assumed by approximation algorithms for standard graph-based objective functions can be viewed as special cases of some of the properties considered above.

- We show how our methods can be extended to the *inductive* case, i.e., by using just a *constant-sized sample*, as in property testing. While most of our algorithms extend in a natural way, for certain properties their analysis requires more involved arguments using regularity-type results of [20, 3].

More generally, our framework provides a formal way to analyze what properties of a similarity function would be sufficient to produce low-error clusterings, as well as what algorithms are suited for a given property. For some of our properties we are able to show that known algorithms succeed (e.g. variations of bottom-up hierarchical linkage based algorithms), but for the most general ones we need new algorithms that are able to take advantage of them.

1.3 Connections to other Related Work

Some of the questions we address can be viewed as a generalization of questions studied in machine learning of what properties of similarity functions (especially kernel functions) are sufficient to allow one to *learn* well [6, 7, 21, 31, 29]. E.g., the usual statement is that if a kernel function satisfies the property that the target function is separable by a large margin in the implicit kernel space, then learning can be done from few labeled examples. The clustering problem is more difficult because there is no labeled data, and even in the relaxations we consider, the forms of feedback allowed are much weaker.

We note that as in learning, given an embedding of data into some metric space, the similarity function $\mathcal{K}(x, x')$ need *not* be a direct translation of distance like $e^{-d(x,x')}$, but rather may be a derived function based on the entire dataset. For example, in the *diffusion kernel* of [27], the similarity $\mathcal{K}(x, x')$ is related to the effective resistance between x and x' in a weighted graph defined from distances in the original metric. This would be a natural similarity function to use, for instance, if data lies in two well-separated pancakes.

In the inductive setting, where we imagine our given data is only a small random sample of the entire data set, our framework is close in spirit to recent work done on sample-based clustering (e.g., [9]) in the context of clustering algorithms designed to optimize a certain objective. Based on such a sample, these algorithms have to output a clustering of the full domain set, that is evaluated with respect to the underlying distribution.

2. DEFINITIONS AND PRELIMINARIES

We consider a clustering problem (S, ℓ) specified as follows. Assume we have a data set S of n objects, where each object is an element of an abstract instance space X. Each $x \in S$ has some (unknown) "ground-truth" label $\ell(x)$ in $Y = \{1, \dots, k\}$, where we will think of k as much smaller than n. The goal is to produce a hypothesis $h : X \to Y$ of low error up to isomorphism of label names. Formally, we define the error of h to be $err(h) = \min_{\sigma \in \mathcal{S}_k} [\Pr_{x \in S} [\sigma(h(x)) \neq \ell(x)]]$. We will assume that a target error rate ϵ, as well as k, are given as input to the algorithm.

We will be considering clustering algorithms whose only access to their data is via a pairwise similarity function $\mathcal{K}(x, x')$ that given two examples outputs a number in the range $[-1, 1]$.[2] We will say that $\mathcal{K}$ is a symmetric similarity function if $\mathcal{K}(x, x') = \mathcal{K}(x', x)$ for all x, x'.

Our focus is to analyze natural properties that sufficient for a similarity function $\mathcal{K}$ to be *good* for a clustering problem (S, ℓ) which (ideally) are intuitive, broad, and imply that such a similarity function results in the ability to *cluster well*. As mentioned in the introduction, however, requiring an algorithm to output a single low-error clustering rules out even quite strong properties. Instead we will consider two objectives that are natural if one assumes the ability to get some limited additional feedback from a user. Specifically, we consider the following two models:

1. **List model:** In this model, the goal of the algorithm is to propose a small number of clusterings such that at least one has error at most ϵ. As in work on property testing, the list length should depend on ϵ and k only, and be independent of n. This list would then go to a domain expert or some hypothesis-testing portion of the system which would then pick out the best clustering.

2. **Tree model:** In this model, the goal of the algorithm is to produce a hierarchical clustering: that is, a tree on subsets such that the root is the set S, and the children of any node S' in the tree form a partition of S'. The requirement is that there must exist a *pruning* h of the tree (not necessarily using nodes all at the same level) that has error at most ϵ. In many applications (e.g. document clustering) this is a significantly more user-friendly output than the list model. Note that any given tree has at most 2^{2k} prunings of size k [26], so this model is at least as strict as the list model.

Transductive vs Inductive. Clustering is typically posed as a "transductive" [35] problem in that we are asked to cluster a *given* set of points S. We can also consider an *inductive* model in which S

[2] That is, the input to the clustering algorithm is just a weighted graph. However, we still want to conceptually view $\mathcal{K}$ as a *function* over abstract objects.

is merely a small random subset of points from a much larger abstract instance space X, and our goal is to produce a hypothesis $h : X \to Y$ of low error on X. For a given property of our similarity function (with respect to X) we can then ask how large a set S we need to see in order for our list or tree produced with respect to S to induce a good solution with respect to X. For clarity of exposition, for most of this paper we will focus on the transductive setting. In Section 6 we show how our algorithms can be adapted to the inductive setting.

Notation. We will denote the underlying ground-truth clusters as $C_1, \ldots, C_k$ (some of which may be empty). For $x \in X$, we use $C(x)$ to denote the cluster $C_{\ell(x)}$ to which point x belongs. For $A \subseteq X, B \subseteq X$, let $\mathcal{K}(A, B) = \mathbf{E}_{x \in A, x' \in B}[\mathcal{K}(x, x')]$. We call this the *average attraction* of A to B. Let $\mathcal{K}_{max}(A, B) = \max_{x \in A, x' \in B} \mathcal{K}(x, x')$; we call this *maximum attraction* of A to B. Given two clusterings g and h we define the distance $d(g, h) = \min_{\sigma \in \mathcal{S}_k} [\Pr_{x \in S}[\sigma(h(x)) \neq g(x)]]$, i.e., the fraction of points in the symmetric difference under the optimal renumbering of the clusters.

We are interested in natural *properties* that we might ask a similarity function to satisfy with respect to the ground truth clustering. For example, one (strong) property would be that all points x are more similar to all points $x' \in C(x)$ than to any $x' \notin C(x)$ – we call this the *strict separation* property. A weaker property would be to just require that points x are *on average* more similar to their own cluster than to any other cluster, that is, $\mathcal{K}(x, C(x) - \{x\}) > \mathcal{K}(x, C_i)$ for all $C_i \neq C(x)$. We will also consider intermediate "stability" conditions. For properties such as these we will be interested in the size of the smallest list any algorithm could hope to output that would guarantee that at least one clustering in the list has error at most ϵ. Specifically, we define the *clustering complexity* of a property as:

DEFINITION 1. *Given a property $\mathcal{P}$ and similarity function $\mathcal{K}$, define the (ϵ, k)-***clustering complexity** *of the pair $(\mathcal{P}, \mathcal{K})$ to be the length of the shortest list of clusterings $h_1, \ldots, h_t$ such that any consistent k-clustering is ϵ-close to some clustering in the list.*[3] *That is, at least one h_i must have error at most ϵ. The (ϵ, k)-***clustering complexity of** *$\mathcal{P}$ is the maximum of this quantity over all similarity functions $\mathcal{K}$.*

The clustering complexity notion is analogous to notions of capacity in classification [11, 19, 35] and it provides a formal measure of the inherent usefulness of a given property.

In the following sections we analyze the clustering complexity of several natural properties and provide efficient algorithms to take advantage of such functions. We start by analyzing the strict separation property as well as a natural relaxation in Section 3. We also give formal relationships between these properties and those considered implicitly by approximation algorithms for standard clustering objectives. We then analyze a much weaker average-attraction property in Section 4 that has close connections to large margin properties studied in Learning Theory [6, 7, 21, 31, 29]. This property is not sufficient to produce a hierarchical clustering, however, so we then turn to the question of how weak a property can be and still be sufficient for hierarchical clustering, which leads us to analyze properties motivated by game-theoretic notions of stability in Section 5.

Our framework allows one to study computational hardness results as well. While our focus is on getting positive algorithmic results, we discuss a simple few hardness examples in the full version of the paper [8].

3. SIMPLE PROPERTIES

We begin with the simple strict separation property mentioned above.

PROPERTY 1. *The similarity function $\mathcal{K}$ satisfies the* **strict separation** *property for the clustering problem (S, ℓ) if all x are strictly more similar to any point $x' \in C(x)$ than to every $x' \notin C(x)$.*

Given a similarity function satisfying the strict separation property, we can efficiently construct a tree such that the ground-truth clustering is a pruning of this tree (Theorem 2). As mentioned above, a consequence of this fact is a $2^{O(k)}$ upper bound on the clustering complexity of this property. We begin by showing a matching $2^{\Omega(k)}$ lower bound.

THEOREM 1. *For $\epsilon < \frac{1}{2k}$, the strict separation property has (ϵ, k)-clustering complexity at least $2^{k/2}$.*

PROOF. The similarity function is a generalization of the picture in Figure 1. Specifically, partition the n points into k subsets $\{R_1, \ldots, R_k\}$ of n/k points each. Group the subsets into pairs $\{(R_1, R_2), (R_3, R_4), \ldots\}$, and let $\mathcal{K}(x, x') = 1$ if x and x' belong to the same R_i, $\mathcal{K}(x, x') = 1/2$ if x and x' belong to two subsets in the same pair, and $\mathcal{K}(x, x') = 0$ otherwise. Notice that in this setting there are $2^{\frac{k}{2}}$ clusterings (corresponding to whether or not to split each pair $R_i \cup R_{i+1}$) that are consistent with Property 1 and differ from each other on at least n/k points. Since $\epsilon < \frac{1}{2k}$, any given hypothesis clustering can be ϵ-close to at most one of these and so the clustering complexity is at least $2^{k/2}$. □

We now present the upper bound.

THEOREM 2. *Let $\mathcal{K}$ be a similarity function satisfying the strict separation property. Then we can efficiently construct a tree such that the ground-truth clustering is a pruning of this tree.*

PROOF. If $\mathcal{K}$ is symmetric, then to produce a tree we can simply use bottom up "single linkage" (i.e., Kruskal's algorithm). That is, we begin with n clusters of size 1 and at each step we merge the two clusters C, C' maximizing $\mathcal{K}_{max}(C, C')$. This maintains the invariant that at each step the current clustering is laminar with respect to the ground-truth: if the algorithm merges two clusters C and C', and C is strictly contained in some cluster C_r of the ground truth, then by the strict separation property we must have $C' \subset C_r$ as well. If $\mathcal{K}$ is not symmetric, then single linkage may fail.[4] However, in this case, the following "Boruvka-inspired" algorithm can be used. Starting with n clusters of size 1, draw a directed edge from each cluster C to the cluster C' maximizing $\mathcal{K}_{max}(C, C')$. Then pick some cycle produced (there must be at least one cycle) and collapse it into a single cluster, and repeat. Note that if a cluster C in the cycle is strictly contained in some ground-truth cluster C_r, then by the strict separation property its out-neighbor must be as well, and so on around the cycle. So this collapsing maintains laminarity as desired. □

One can also consider the property that $\mathcal{K}$ satisfies strict separation for *most* of the data.

PROPERTY 2. *The similarity function $\mathcal{K}$ satisfies ν-***strict separation** *for the clustering problem (S, ℓ) if for some $S' \subseteq S$ of size $(1 - \nu)n$, $\mathcal{K}$ satisfies strict separation for (S', ℓ).*

[3] A clustering $\mathcal{C}$ is consistent if $\mathcal{K}$ has property $\mathcal{P}$ with respect to $\mathcal{C}$.

[4] Consider 3 points x, y, z whose correct clustering is $(\{x\}, \{y, z\})$. If $\mathcal{K}(x, y) = 1$, $\mathcal{K}(y, z) = \mathcal{K}(z, y) = 1/2$, and $\mathcal{K}(y, x) = \mathcal{K}(z, x) = 0$, then this is consistent with strict separation and yet the algorithm will incorrectly merge x and y in its first step.

THEOREM 3. *If $\mathcal{K}$ satisfies ν-**strict separation**, then so long as the smallest correct cluster has size greater than $5\nu n$, we can produce a tree such that the ground-truth clustering is ν-close to a pruning of this tree.*

PROOF. See appendix. □

Approximation Assumptions: When developing a c-approximation algorithm for some clustering objective function F, if the goal is to actually get the points correct, then one is implicitly making the assumption (or hope) that any c-approximation to F must be ϵ-close in symmetric difference to the target clustering. We show here we show how assumptions of this kind can be viewed as special cases of the ν-strict separation property.

PROPERTY 3. *Given objective function F, we say that a metric d over point set S satisfies the (c, ϵ)-F property with respect to target $\mathcal{C}$ if all clusterings $\mathcal{C}'$ that are within a factor c of optimal in terms of objective F are ϵ-close to $\mathcal{C}$.*

We now consider in particular the k-median and k-center objective functions.

THEOREM 4. *If metric d satisfies the $(2, \epsilon)$-k-median property for dataset S, then the similarity function $-d$ satisfies the ν-strict separation property for $\nu = 3\epsilon$.*

PROOF. Suppose the data does not satisfy strict separation. Then there must exist points a_1, b_1, c_1 with a_1 and b_1 in one cluster and c_1 in another such that $d(a_1, b_1) > d(a_1, c_1)$. Remove these three points and repeat with a_2, b_2, c_2. Suppose for contradiction, this process continues past $a_{\epsilon n}, b_{\epsilon n}, c_{\epsilon n}$. Then, moving all a_i into the clusters of the corresponding c_i will increase the k-median objective by at most $\sum_i [d(a_i, c_i) + \mathrm{cost}(c_i)]$, where $\mathrm{cost}(x)$ is the contribution of x to the k-median objective function. By definition of the a_i and by triangle inequality, this is at most $\sum_i [\mathrm{cost}(a_i) + \mathrm{cost}(b_i) + \mathrm{cost}(c_i)]$, which in turn is at most $\sum_{x \in S} \mathrm{cost}(x)$. Thus, the k-median objective at most doubles, contradicting our initial assumption. □

THEOREM 5. *If metric d satisfies the $(3, \epsilon)$-k-center property, then the similarity function $-d$ satisfies the ν-strict separation property for $\nu = 3\epsilon$.*

(proof omitted). In fact, the $(2, \epsilon)$-k-median property is quite a bit more restrictive than ν-strict separation. It implies, for instance, that except for an $O(\epsilon)$ fraction of "bad" points, there exists d such that all points in the same cluster have distance much less than d and all points in different clusters have distance much greater than d. In contrast, ν-strict separation would allow for different distance scales at different parts of the graph.

4. WEAKER PROPERTIES

A much weaker property to ask of a similarity function is just that most points are noticeably more similar *on average* to points in their own cluster than to points in any other cluster. Specifically, we define:

PROPERTY 4. *A similarity function $\mathcal{K}$ satisfies the (ν, γ)-**average attraction** property for the clustering problem (S, ℓ) if a $1 - \nu$ fraction of examples x satisfy:*

$$\mathcal{K}(x, C(x)) \geq \mathcal{K}(x, C_i) + \gamma \text{ for all } i \in Y, i \neq \ell(x).$$

This is a fairly natural property to ask of a similarity function: if a point x is more similar on average to points in a different cluster than to those in its own, it is hard to expect an algorithm to label it correctly. The following is a simple clustering algorithm that given a similarity function $\mathcal{K}$ satisfying the average attraction property produces a list of clusterings of size that depends only on ϵ, k, and γ. Specifically,

Algorithm 1 Sampling Based Algorithm, List Model

```
Input: Data set $S$, similarity function $\mathcal{K}$, parameters $\gamma, \epsilon > 0$, $k \in Z^+$; $N(\epsilon, \gamma, k)$, $s(\epsilon, \gamma, k)$.

  • Set $\mathcal{L} = \emptyset$.
  • Repeat $N(\epsilon, \gamma, k)$ times
      For $k' = 1, \ldots, k$ do:
        - Pick a set $R_S^{k'}$ of $s(\epsilon, \gamma, k)$ random points from $S$.
        - Let $h$ be the average-nearest neighbor hypothesis induced by the sets $R_S^i$, $1 \leq i \leq k'$. That is, for any point $x \in S$, define $h(x) = \mathrm{argmax}_{i \in \{1, \ldots k'\}} [\mathcal{K}(x, R_S^i)]$.
          Add $h$ to $\mathcal{L}$.
  • Output the list $\mathcal{L}$.
```

THEOREM 6. *Let $\mathcal{K}$ be a similarity function satisfying the (ν, γ)-average attraction property for the clustering problem (S, ℓ). Using Algorithm 1 with the parameters $s(\epsilon, \gamma, k) = \frac{4}{\gamma^2} \ln\left(\frac{8k}{\epsilon\delta}\right)$ and $N(\epsilon, \gamma, k) = \left(\frac{2k}{\epsilon}\right)^{\frac{4k}{\gamma^2} \ln\left(\frac{8k}{\epsilon\delta}\right)} \ln(\frac{1}{\delta})$ we can produce a list of at most $k^{O\left(\frac{k}{\gamma^2} \ln\left(\frac{1}{\epsilon}\right) \ln\left(\frac{k}{\epsilon\delta}\right)\right)}$ clusterings such that with probability $1 - \delta$ at least one of them is $(\nu + \epsilon)$-close to the ground-truth.*

PROOF. We say that a ground-truth cluster is big if it has probability mass at least $\frac{\epsilon}{2k}$; otherwise, we say that the cluster is small. Let k' be the number of "big" ground-truth clusters. Clearly the probability mass in all the small clusters is at most $\epsilon/2$.

Let us arbitrarily number the big clusters $C_1, \ldots, C_{k'}$. Notice that in each round there is at least a $\left(\frac{\epsilon}{2k}\right)^{s(\epsilon,\gamma,k)}$ probability that $R_S{}^i \subseteq C_i$, and so at least a $\left(\frac{\epsilon}{2k}\right)^{ks(\epsilon,\gamma,k)}$ probability that $R_S{}^i \subseteq C_i$ for all $i \leq k'$. Thus the number of rounds $\left(\frac{2k}{\epsilon}\right)^{\frac{4k}{\gamma^2} \ln\left(\frac{8k}{\epsilon\delta}\right)} \ln(\frac{1}{\delta})$ is large enough so that with probability at least $1 - \delta/2$, in at least one of the $N(\epsilon, \gamma, k)$ rounds we have $R_S{}^i \subseteq C_i$ for all $i \leq k'$. Let us fix now one such good round. We argue next that the clustering induced by the sets picked in this round has error at most $\nu + \epsilon$ with probability at least $1 - \delta$.

Let Good be the set of x in the big clusters satisfying

$$\mathcal{K}(x, C(x)) \geq \mathcal{K}(x, C_j) + \gamma \text{ for all } j \in Y, j \neq \ell(x).$$

By assumption and from the previous observations, $\Pr_{x \sim S}[x \in \mathsf{Good}] \geq 1 - \nu - \epsilon/2$. Now, fix $x \in \mathsf{Good}$. Since $\mathcal{K}(x, x') \in [-1, 1]$, by Hoeffding bounds we have that over the random draw of $R_S{}^j$, conditioned on $R_S{}^j \subseteq C_j$,

$$\Pr_{R_S{}^j}\left(\left|\mathbf{E}_{x' \sim R_S{}^j}[\mathcal{K}(x, x')] - \mathcal{K}(x, C_j)\right| \geq \gamma/2\right) \leq 2e^{-2|R_S{}^j|\gamma^2/4},$$

for all $j \in \{1, \ldots, k'\}$. By our choice of $R_S{}^j$, each of these probabilities is at most $\epsilon\delta/4k$. So, for any given $x \in \mathsf{Good}$, there is at most a $\epsilon\delta/4$ probability of error over the draw of the sets $R_S{}^j$. Since this is true for any $x \in \mathsf{Good}$, it implies that the *expected* error of this procedure, over $x \in \mathsf{Good}$, is at most $\epsilon\delta/4$, which by Markov's inequality implies that there is at most a $\delta/2$ probability that the error rate over Good is more than $\epsilon/2$. Adding in the

$\nu + \epsilon/2$ probability mass of points not in Good yields the theorem. □

Note that Theorem 6 immediately implies a corresponding upper bound on the (ϵ, k)-clustering complexity of the $(\epsilon/2, \gamma)$-average attraction property. We can also give a lower bound showing that the exponential dependence on γ is necessary, and furthermore this property is not sufficient to cluster in the tree model:

THEOREM 7. *For $\epsilon < \gamma/2$, the (ϵ, k)-clustering complexity of the $(0, \gamma)$-average attraction property is at least $\max_{k' \leq k} k'^{\frac{1}{\gamma}}/k'!$, and moreover this property is not sufficient to cluster in the tree model.*

PROOF. Omitted. See the full version of the paper [8]. □

Note: In fact, the clustering complexity bound immediately implies one cannot cluster in the tree model since for $k = 2$ the bound is greater than 1.

One can even weaken the above property to ask only that there *exists* an (unknown) weighting function over data points (thought of as a "reasonableness score"), such that most points are on average more similar to the *reasonable* points of their own cluster than to the *reasonable* points of any other cluster. This is a generalization of the notion of $\mathcal{K}$ being a kernel function with the large margin property [6, 32, 35, 30]. For details see the full version of the paper [8].

5. STABILITY-BASED PROPERTIES

The properties in Section 4 are fairly general and allow construction of a list whose length depends only on on ϵ and k (for constant γ), but are not sufficient to produce a single tree. In this section, we show that several natural stability-based properties that lie between those considered in Sections 3 and 4 are in fact sufficient for *hierarchical* clustering.

For simplicity, we focus on symmetric similarity functions. We consider the following relaxations of Property 1 which ask that the ground truth be "stable" in the stable-marriage sense:

PROPERTY 5. *A similarity function $\mathcal{K}$ satisfies the* **strong stability** *property for the clustering problem (S, ℓ) if for all clusters C_r, $C_{r'}$, $r \neq r'$ in the ground-truth, for all $A \subset C_r$, $A' \subseteq C_{r'}$ we have*

$$\mathcal{K}(A, C_r \setminus A) > \mathcal{K}(A, A').$$

PROPERTY 6. *A similarity function $\mathcal{K}$ satisfies the* **weak stability** *property for the clustering problem (S, ℓ) if for all C_r, $C_{r'}$, $r \neq r'$, for all $A \subset C_r$, $A' \subseteq C_{r'}$, we have:*

- *If $A' \subset C_{r'}$ then either $\mathcal{K}(A, C_r \setminus A) > \mathcal{K}(A, A')$ or $\mathcal{K}(A', C_{r'} \setminus A') > \mathcal{K}(A', A)$.*
- *If $A' = C_{r'}$ then $\mathcal{K}(A, C_r \setminus A) > \mathcal{K}(A, A')$.*

We can interpret weak stability as saying that for any two clusters in the ground truth, there does not exist a subset A of one and subset A' of the other that are more attracted to each other than to the remainder of their true clusters (with technical conditions at the boundary cases) much as in the classic notion of stable-marriage. Strong stability asks that *both* be more attracted to their true clusters. To further motivate these properties, note that if we take the example from Figure 1 and set a small random fraction of the edges inside each dark-shaded region to 0, then with high probability this would still satisfy strong stability with respect to all the natural clusters even though it no longer satisfies strict separation (or even ν-strict separation for any $\nu < 1$ if we included at least one edge incident to each vertex). Nonetheless, we can show that these stability notions are sufficient to produce a hierarchical clustering. We prove this for strong stability here; the proof for weak stability appear in the full version of the paper [8].

Algorithm 2 Average Linkage, Tree Model

```
Input: Data set $S$, similarity function $\mathcal{K}$. Output: A tree on subsets.

• Begin with $n$ singleton clusters.

• Repeat till only one cluster remains: Find clusters $C, C'$ in
  the current list which maximize $K(C, C')$ and merge them
  into a single cluster.

• Output the tree with single elements as leaves and internal
  nodes corresponding to all the merges performed.
```

THEOREM 8. *Let $\mathcal{K}$ be a symmetric similarity function satisfying Property 5. Then we can efficiently construct a binary tree such that the ground-truth clustering is a pruning of this tree.*

Proof Sketch: We will show that Algorithm 2 (Average Linkage) will produce the desired result. Note that the algorithm uses $\mathcal{K}(C, C')$ rather than $\mathcal{K}_{max}(C, C')$ as in single linkage.

We prove correctness by induction. In particular, assume that our current clustering is laminar with respect to the ground truth clustering (which is true at the start). That is, for each cluster C in our current clustering and each C_r in the ground truth, we have either $C \subseteq C_r$, or $\mathcal{C}_r \subseteq C$ or $C \cap C_r = \emptyset$. Now, consider a merge of two clusters C and C'. The only way that laminarity could fail to be satisfied after the merge is if one of the two clusters, say, C', is strictly contained inside some ground-truth cluster C_r (so, $C_r - C' \neq \emptyset$) and yet C is disjoint from C_r. Now, note that by Property 5, $\mathcal{K}(C', C_r - C') > \mathcal{K}(C', x)$ for all $x \notin C_r$, and so in particular we have $\mathcal{K}(C', C_r - C') > \mathcal{K}(C', C)$. Furthermore, $\mathcal{K}(C', C_r - C')$ is a weighted average of the $\mathcal{K}(C', C'')$ over the sets $C'' \subseteq C_r - C'$ in our current clustering and so at least one such C'' must satisfy $\mathcal{K}(C', C'') > \mathcal{K}(C', C)$. However, this contradicts the specification of the algorithm, since by definition it merges the pair C, C' such that $\mathcal{K}(C', C)$ is greatest. □

While natural, Properties 5 and 6 are still somewhat brittle: in the example of Figure 1, for instance, if one adds a small number of edges with similarity 1 *between* the natural clusters, then the properties are no longer satisfied for them (because pairs of elements connected by these edges will want to defect). We can make the properties more robust by requiring that stability hold only for *large* sets. This will break the average-linkage algorithm used above, but we can show that a more involved algorithm building on the approach used in Section 4 will nonetheless find an approximately correct tree. For simplicity, we focus on broadening the strong stability property, as follows (one should view s as small compared to ϵ/k in this definition):

PROPERTY 7. *The similarity function $\mathcal{K}$ satisfies the (s, γ)-***strong stability of large subsets** *property for the clustering problem (S, ℓ) if for all clusters C_r, $C_{r'}$, $r \neq r'$ in the ground-truth, for all $A \subset C_r$, $A' \subseteq C_{r'}$ with $|A| + |A'| \geq sn$ we have*

$$\mathcal{K}(A, C_r \setminus A) > \mathcal{K}(A, A') + \gamma.$$

The idea of how we can use this property is we will first run an algorithm for the list model much like Algorithm 1, viewing its output as simply a long list of candidate clusters (rather than cluster*ings*). In particular, we will get a list $\mathcal{L}$ of $k^{O\left(\frac{k}{\gamma^2}\log\frac{1}{\epsilon}\log\frac{k}{\delta f}\right)}$ clusters such that with probability at least $1-\delta$ any cluster in the ground-truth of size at least $\frac{\epsilon}{4k}$ is close to one of the clusters in the list. We then run a second "tester" algorithm that is able to throw away candidates that are sufficiently non-laminar with respect to the correct clustering and assembles the ones that remain into a tree. We present and analyze the tester algorithm, Algorithm 3, below.

Algorithm 3 Testing Based Algorithm, Tree Model.

Input: Data set S, similarity function $\mathcal{K}$, parameters $\gamma > 0$, $k \in Z^+$, $f, g, s, \alpha > 0$. A list of clusters $\mathcal{L}$ with the property that any cluster C in the ground-truth is at least f-close to one of them.
Output: A tree on subsets.

1. Throw out all clusters of size at most αn. For every pair of clusters C, C' in our list $\mathcal{L}$ of clusters that are sufficiently "non-laminar" with respect to each other in that $|C \setminus C'| \geq gn$, $|C' \setminus C| \geq gn$ and $|C \cap C'| \geq gn$, compute $\mathcal{K}(C \cap C', C \setminus C')$ and $\mathcal{K}(C \cap C', C' \setminus C)$. Throw out whichever one does worse: i.e., throw out C if the first similarity is smaller, else throw out C'. Let $\mathcal{L}'$ be the remaining list of clusters at the end of the process.

2. Greedily sparsify the list $\mathcal{L}'$ so that no two clusters are approximately equal (that is, choose a cluster, throw out all that are approximately equal to it, and repeat). We say two clusters C, C' are approximately equal if $|C \setminus C'| \leq gn$, $|C' \setminus C| \leq gn$ and $|C' \cap C| \geq gn$. Let $\mathcal{L}''$ be the list remaining.

3. Construct a forest on the remaining list $\mathcal{L}''$. C becomes a child of C' in this forest if C' approximately contains C, i.e. $|C \setminus C'| \leq gn$, $|C' \setminus C| \geq gn$ and $|C' \cap C| \geq gn$.

4. Complete the forest arbitrarily into a tree.

THEOREM 9. *Let $\mathcal{K}$ be a similarity function satisfying (s, γ)-strong stability of large subsets for the clustering problem (S, ℓ). Let $\mathcal{L}$ be a list of clusters such that any cluster in the ground-truth of size at least αn is f-close to one of the clusters in the list. Then Algorithm 3 with parameters satisfying $s + f \leq g$, $f \leq g\gamma/10$ and $\alpha > 6kg$ yields a tree such that the ground-truth clustering is $2\alpha k$-close to a pruning of this tree.*

Proof Sketch: Let k' be the number of "big" ground-truth clusters: the clusters of size at least αn; without loss of generality assume that C_1, ..., $C_{k'}$ are the big clusters.

Let C'_1, ...,$C'_{k'}$ be clusters in $\mathcal{L}$ such that $d(C_i, C'_i)$ is at most f for all i. By Property 7 and Lemma 10 (stated below), we know that after Step 1 (the "testing of clusters" step) all the clusters C'_1, ...,$C'_{k'}$ survive; furthermore, we have three types of relations between the remaining clusters. Specifically, either:

(a) C and C' are approximately equal; that means $|C \setminus C'| \leq gn$, $|C' \setminus C| \leq gn$ and $|C' \cap C| \geq gn$.

(b) C and C' are approximately disjoint; that means $|C \setminus C'| \geq gn$, $|C' \setminus C| \geq gn$ and $|C' \cap C| \leq gn$.

(c) or C' approximately contains C; that means $|C \setminus C'| \leq gn$, $|C' \setminus C| \geq gn$ and $|C' \cap C| \geq gn$.

Let $\mathcal{L}''$ be the remaining list of clusters after sparsification. It's easy to show that there exists $C''_1, ..., C''_{k'}$ in $\mathcal{L}''$ such that $d(C_i, C''_i)$ is at most $(f + 2g)$, for all i. Moreover, all the elements in $\mathcal{L}''$ are either in the relation "subset" or "disjoint". Also, since all the clusters C_1, ..., $C_{k'}$ have size at least αn, we also have that C''_i, C''_j are in the relation "disjoint", for all i, j, $i \neq j$. That is, in the forest we construct C''_i are not descendants of one another.

We show $C''_1, ..., C''_{k'}$ are part of a pruning of small error rate of the final tree. We do so by exhibiting a small extension to a list of clusters $\mathcal{L}'''$ that are all approximately disjoint and nothing else in $\mathcal{L}''$ is approximately disjoint from any of the clusters in $\mathcal{L}'''$ (thus $\mathcal{L}'''$ will be the desired pruning). Specifically greedily pick a cluster $\tilde{C}_1$ in $\mathcal{L}''$ that is approximately disjoint from $C''_1, ..., C''_{k'}$, and in general in step $i > 1$ greedily pick a cluster $\tilde{C}_1$ in $\mathcal{L}''$ that is approximately disjoint from $C''_1, ..., C''_{k'}, \tilde{C}_1, \ldots, \tilde{C}_{i-1}$. Let $C''_1, ..., C''_{k'}, \tilde{C}_1, \ldots, \tilde{C}_{\tilde{k}}$ be the list $\mathcal{L}'''$. By design, $\mathcal{L}'''$ will be a pruning of the final tree and we now claim its total error is at most $2\alpha kn$. In particular, note that the total number of points missing from $C''_1, ..., C''_{k'}$ is at most $k(f + 2g)n + k\alpha n \leq \frac{3}{2}k\alpha n$. Also, by construction, each $\tilde{C}_i$ must contain at least $\alpha n - (k + i)gn$ new points, which together with the above implies that $\tilde{k} \leq 2k$. Thus, the total error of $\mathcal{L}'''$ overall is at most $\frac{3}{2}\alpha kn + 2kk'gn \leq 2\alpha kn$. □

LEMMA 10. *Let $\mathcal{K}$ be a similarity function satisfying the (s, γ)-strong stability of large subsets property for the clustering problem (S, ℓ). Let C, C' be such that $|C \cap C'| \geq gn$, $|C \setminus C'| \geq gn$ and $|C' \setminus C| \geq gn$. Let C^* be a cluster in the underlying ground-truth such that $|C^* \setminus C| \leq fn$ and $|C \setminus C^*| \leq fn$. Let $I = C \cap C'$. If $s + f \leq g$ and $f \leq g\gamma/10$, then $\mathcal{K}(I, C \setminus I) > \mathcal{K}(I, C' \setminus I)$.*

PROOF. Omitted. See the full version of the paper [8]. □

THEOREM 11. *Let $\mathcal{K}$ be a similarity function satisfying the (s, γ)-strong stability of large subsets property for the clustering problem (S, ℓ). Assume that $s = O(\epsilon^2\gamma/k^2)$. Then using Algorithm 3 with parameters $\alpha = O(\epsilon/k)$, $g = O(\epsilon^2/k^2)$, $f = O(\epsilon^2\gamma/k^2)$, together with Algorithm 1 we can with probability $1 - \delta$ produce a tree with the property that the ground-truth is ϵ-close to a pruning of this tree. Moreover, the size of this tree is $O(k/\epsilon)$.*

Proof Sketch: First, we run Algorithm 1 get a list $\mathcal{L}$ of clusters such that with probability at least $1 - \delta$ any cluster in the ground-truth of size at least $\frac{\epsilon}{4k}$ is f-close to one of the clusters in the list. We can ensure that our list $\mathcal{L}$ has size at most $k^{O\left(\frac{k}{\gamma^2}\log\frac{1}{\epsilon}\log\frac{k}{\delta f}\right)}$. We then run Procedure 3 with parameters $\alpha = O(\epsilon/k)$, $g = O(\epsilon^2/k^2)$, $f = O(\epsilon^2\gamma/k^2)$. We thus obtain a tree with the guarantee that the ground-truth is ϵ-close to a pruning of this tree (see Theorem 9). To complete the proof we only need to show that this tree has $O(k/\epsilon)$ leaves. This follows from the fact that all leaves of our tree have at least αn points and the overlap between any two of them is at most gn. □

6. INDUCTIVE SETTING

In this section we consider an *inductive* model in which S is merely a small random subset of points from a much larger abstract instance space X, and clustering is represented *implicitly* through a hypothesis $h : X \rightarrow Y$. In the list model our goal is to produce a list of hypotheses, $\{h_1, \ldots, h_t\}$ such that at least one of them has error at most ϵ. In the tree model we assume that each node in the tree induces a cluster which is implicitly represented as a function $f : X \to \{0, 1\}$. For a fixed tree T and a point x, we define $T(x)$ as the subset of nodes in T that contain x (the subset of nodes $f \in T$ with $f(x) = 1$). We say that a tree T has error at most ϵ if $T(X)$ has a pruning $f_1, ..., f_{k'}$ of error at most ϵ.

We analyze in the following, for each of our properties, how large a set S we need to see in order for our list or tree produced with respect to S to induce a good solution with respect to X.

The average attraction property. Our algorithms for the average attraction property (Property 4) and the average weighted attraction property are already inherently inductive.

The strict separation property. We can adapt the algorithm in Theorem 2 to the inductive setting as follows. We first draw a set S of $n = O\big(\frac{k}{\epsilon} \ln\big(\frac{k}{\delta}\big)\big)$ unlabeled examples. We run the algorithm described in Theorem 2 on this set and obtain a tree T on the subsets of S. Let Q be the set of leaves of this tree. We associate each node u in T a boolean function f_u specified as follows. Consider $x \in X$, and let $q(x) \in Q$ be the leaf given by $\mathrm{argmax}_{q \in Q} \mathcal{K}(x, q)$; if u appears on the path from $q(x)$ to the root, then set $f_u(x) = 1$, otherwise set $f_u(x) = 0$.

Note that n is large enough to ensure that with probability at least $1 - \delta$, S includes at least a point in each cluster of size at least $\frac{\epsilon}{k}$. Remember that $\mathcal{C} = \{C_1, \ldots, C_k\}$ is the correct clustering of the entire domain. Let $\mathcal{C}_S$ be the (induced) correct clustering on our sample S of size n. Since our property is hereditary, Theorem 2 implies that $\mathcal{C}_S$ is a pruning of T. It then follows from the specification of our algorithm and from the definition of the strict separation property that with probability at least $1 - \delta$ the partition induced over the whole space by this pruning is ϵ-close to $\mathcal{C}$.

The strong stability of large subsets property. We can also naturally extend the algorithm for Property 7 to the inductive setting. The main difference in the inductive setting is that we have to *estimate* (rather than *compute*) the $|C_r \setminus C_{r'}|$, $|C_{r'} \setminus C_r|$, $|C_r \cap C_{r'}|$, $\mathcal{K}(C_r \cap C_{r'}, C_r \setminus C_{r'})$ and $\mathcal{K}(C_r \cap C_{r'}, C_{r'} \setminus C_r)$ for any two clusters C_r, $C_{r'}$ in the list $\mathcal{L}$. We can easily do that with only $\mathrm{poly}(k, 1/\epsilon, 1/\gamma, 1/\delta) \log(|\mathcal{L}|))$ additional points, where $\mathcal{L}$ is the input list in Algorithm 3 (whose size depends on $1/\epsilon$, $1/\gamma$ and k only). Specifically, using a modification of the proof in Theorem 11 and standard concentration inequalities (e.g. the McDiarmid inequality [19]) we can show that:

THEOREM 12. *Assume that $\mathcal{K}$ is a similarity function satisfying the (s, γ)-strong stability of large subsets property for (X, ℓ). Assume that $s = O(\epsilon^2\gamma/k^2)$. Then using Algorithm 3 with parameters $\alpha = O(\epsilon/k)$, $g = O(\epsilon^2/k^2)$, $f = O(\epsilon^2\gamma/k^2)$, together with Algorithm 1 we can produce a tree with the property that the ground-truth is ϵ-close to a pruning of this tree. Moreover, the size of this tree is $O(k/\epsilon)$. We use $O\big(\frac{k}{\gamma^2} \ln\big(\frac{k}{\epsilon\delta}\big) \cdot \big(\frac{k}{\epsilon}\big)^{\frac{4k}{\gamma^2} \ln\big(\frac{k}{\epsilon\delta}\big)} \ln(\frac{1}{\delta})\big)$ points in the first phase and $O\big(\frac{1}{\gamma^2} \frac{1}{g^2} \frac{k}{\gamma^2} \log \frac{1}{\epsilon} \log \frac{k}{\delta f} \log k\big)$ points in the second phase.*

Note that each cluster is represented as a nearest neighbor hypothesis over at most k sets.

The strong stability property. We first note that we need to consider a variant of our property that has a γ-gap. To see why this is necessary consider the following example. Suppose all $\mathcal{K}(x, x')$ values are equal to $1/2$, except for a special single center point x_i in each cluster C_i with $\mathcal{K}(x_i, x) = 1$ for all x in C_i. This satisfies strong-stability since for every $A \subset C_i$ we have $\mathcal{K}(A, C_i \setminus A)$ is strictly larger than $1/2$. Yet it is impossible to cluster in the inductive model because our sample is unlikely to contain the center points. The variant of our property that is suited to the inductive setting is the following:

PROPERTY 8. *The similarity function $\mathcal{K}$ satisfies the γ-**strong stability** property for the clustering problem (X, ℓ) if for all clusters C_r, $C_{r'}$, $r \neq r'$ in the ground-truth, for all $A \subset C_r$, for all $A' \subseteq C_{r'}$ we have*

$$\mathcal{K}(A, C_r \setminus A) > \mathcal{K}(A, A') + \gamma.$$

For this property, we could always run the algorithm for Theorem 12, though running time would be exponential in k and $1/\gamma$. We show here how we can get polynomial dependence on these parameters by adapting Algorithm 2 to the inductive setting as in the case of the strict order property. Specifically, we first draw a set S of n unlabeled examples. We run the average linkage algorithm on this set and obtain a tree T on the subsets of S. We then attach each new point x to its most similar leaf in this tree as well as to the set of nodes on the path from that leaf to the root. For a formal description see Algorithm 4. While this algorithm looks natural, proving its correctness requires more involved arguments.

Algorithm 4 Inductive Average Linkage, Tree Model

```
Input: Similarity function $\mathcal{K}$, parameters $\gamma, \epsilon > 0$, $k \in Z^+$;
$n = n(\epsilon, \gamma, k, \delta)$;

• Pick a set $S = \{x_1, \ldots, x_n\}$ of $n$ random examples from $X$

• Run the average linkage algorithm (Algorithm 2) on the set
  $S$ and obtain a tree $T$ on the subsets of $S$. Let $Q$ be the set
  of leaves of this tree.

• Associate each node $u$ in $T$ a function $f_u$ (which induces a
  cluster) specified as follows.
  Consider $x \in X$, and let $q(x) \in Q$ be the leaf given by
  $\mathrm{argmax}_{q \in Q} \mathcal{K}(x, q)$; if $u$ appears on the path from $q(x)$ to
  the root, then set $f_u(x) = 1$, otherwise set $f_u(x) = 0$.

• Output the tree $T$.
```

We show in the following that for $n = \mathrm{poly}(k, 1/\epsilon, 1/\gamma, 1/\delta)$ we obtain a tree T which has a pruning $f_1, ..., f_{k'}$ of error at most ϵ. Specifically:

THEOREM 13. *Let $\mathcal{K}$ be a similarity function satisfying the strong stability property for the clustering problem (X, ℓ). Then using Algorithm 4 with parameters $n = \mathrm{poly}(k, 1/\epsilon, 1/\gamma, 1/\delta)$, we can produce a tree with the property that the ground-truth is ϵ-close to a pruning of this tree.*

Proof Sketch: Remember that $\mathcal{C} = \{C_1, \ldots, C_k\}$ is the ground-truth clustering of the entire domain. Let $\mathcal{C}_S = \{C'_1, \ldots, C'_k\}$ be the (induced) correct clustering on our sample S of size n. As in the previous arguments we assume that a cluster is big if it has probability mass at least $\frac{\epsilon}{2k}$.

First, Theorem 14 below implies that with high probability the clusters C'_i corresponding to the large ground-truth clusters satisfy our property with a gap $\gamma/2$. (Just perform a union bound over $x \in S \setminus C'_i$.) It may be that C'_i corresponding to the small ground-truth clusters do not satisfy the property. However, a careful analysis of the argument in Theorem 8 shows that that with high probability $\mathcal{C}_S$ is a pruning of the tree T. Furthermore since n is large enough we also have that with high probability $\mathcal{K}(x, C(x))$ is within $\gamma/2$

of $\mathcal{K}(x, C'(x))$ for a $1-\epsilon$ fraction of points x. This ensures that with high probability, for any such good x the leaf $q(x)$ belongs to $C(x)$. This finally implies that the partition induced over the whole space by the pruning $\mathcal{C}_S$ of the tree T is ϵ-close to $\mathcal{C}$. □

Note that each cluster u is implicitly represented by the function f_u defined in the description of Algorithm 4.

We prove in the following that for a sufficiently large value of n sampling preserves stability. Specifically:

THEOREM 14. *Let $C_1, C_2, \ldots, C_k$ be a partition of a set X such that for any $A \subseteq C_i$ and any $x \notin C_i$,*

$$K(A, C_i \setminus A) \geq K(A, x) + \gamma.$$

Let $x \notin C_i$ and let C_i' be a random subset of n' elements of C_i. Then, $n' = poly(1/\gamma, \log(1/\delta))$ is sufficient so that with probability $1-\delta$, for any $A \subset C_i'$,

$$K(A, C_i' \setminus A) \geq K(A, x) + \frac{\gamma}{2}.$$

Proof Sketch: First of all, the claim holds for singleton subsets A with high probability using a Chernoff bound. This implies the condition is also satisfied for every subset A of size at most $\gamma n'/2$. Thus, it remains to prove the claim for large subsets. We do this using the cut-decomposition of [20] and the random sampling analysis of [3].

Let $N = |C_i|$. By [20], we can decompose the similarity matrix for C_i into a sum of cut-matrices $B_1 + B_2 + \ldots + B_s$ plus a low cut-norm matrix W with the following properties. First, each B_j is a cut-matrix, meaning that for some subset S_{j1} of the rows and subset S_{j2} of the columns and some value d_j, we have: $B_j[xy] = d_j$ for $x \in S_{j1}, y \in S_{j2}$ and all $B_j[xy] = 0$ otherwise. Second, each $d_j = O(1)$. Finally, $s = 1/\epsilon^2$ cut-matrices are sufficient so that matrix W has cut-norm at most $\epsilon^2 N$: that is, for any partition of the vertices A, A', we have $|\sum_{x \in A, y \in A'} W[xy]| \leq \epsilon N^2$; moreover, $||W||_\infty \leq 1/\varepsilon$ and $||W||_F \leq N$.

We now closely follow arguments in [3]. First, let us imagine that we have exact equality $C_i = B_1 + \ldots + B_s$, and we will add in the matrix W later. We are given that for all A, $K(A, C_i \setminus A) \geq K(A, x) + \gamma$. In particular, this trivially means that for each "profile" of sizes $\{t_{jr}\}$, there is no set A satisfying

$$\begin{aligned} |A \cap S_{jr}| &\in [t_{jr} - \alpha, t_{jr} + \alpha]N \\ |A| &\geq (\gamma/4)N \end{aligned}$$

that violates our given condition. The reason for considering cut-matrices is that the values $|A \cap S_{jr}|$ completely determine the quantity $K(A, C_i \setminus A)$. We now set α so that the above constraints determine $K(A, C_i \setminus A)$ up to $\pm\gamma/4$. In particular, choosing $\alpha = o(\gamma^2/s)$ suffices. This means that fixing a profile of values $\{t_{jr}\}$, we can replace "violates our given condition" with $K(A, x) \geq c_0$ for some value c_0 depending on the profile, losing only an amount $\gamma/4$. We now apply Theorem 9 (random subprograms of LPs) of [3]. This theorem states that with probability $1-\delta$, in the subgraph C_i', there is no set A' satisfying the above inequalities where the right-hand-sides and objective c_0 are reduced by $O(\sqrt{\log(1/\delta)}/\sqrt{n})$. Choosing $n \gg \log(1/\delta)/\alpha^2$ we get that with high probability the induced cut-matrices B_i' have the property that there is no A' satisfying

$$\begin{aligned} |A' \cap S_{jr}'| &\in [t_{jr} - \alpha/2, t_{jr} + \alpha/2]N \\ |A'| &\geq (\gamma/2)n' \end{aligned}$$

with the objective value c_0 reduced by at most $\gamma/4$. We now simply do a union-bound over all possible profiles $\{t_{jr}\}$ consisting of multiples of α to complete the argument.

Finally, we incorporate the additional matrix W using the following result from [3].

LEMMA 15. *[3][Random submatrix] For $\varepsilon, \delta > 0$, and any W an $N \times N$ real matrix with cut-norm $||W||_C \leq \varepsilon N^2$, $||W||_\infty \leq 1/\varepsilon$ and $||W||_F \leq N$, let S' be a random subset of the rows of W with $n' = |S'|$ and let W' be the $n' \times n'$ submatrix of W corresponding to W. For $n' > (c_1/\varepsilon^4\delta^5)\log(2/\varepsilon)$, with probability at least $1-\delta$,*

$$||W'||_C \leq c_2 \frac{\varepsilon}{\sqrt{\delta}} n'^2$$

where c_1, c_2 are absolute constants.

We want the addition of W' to influence the values $K(A, C_i' - A)$ by $o(\gamma)$. We now use the fact that we only care about the case that $|A| \geq \gamma n'/2$ and $|C_i' - A| \geq \gamma n'/2$, so that it suffices to affect the sum $\sum_{x \in A, y \in C_i' - A} K(x, y)$ by $o(\gamma^2 n'^2)$. In particular, this means it suffices to have $\epsilon = \tilde{o}(\gamma^2)$, or equivalently $s = \tilde{O}(1/\gamma^4)$. This in turn implies that it suffices to have $\alpha = \tilde{o}(\gamma^6)$, which implies that $n' = \tilde{O}(1/\gamma^{12})$ suffices for the theorem. □

7. CONCLUSIONS AND OPEN QUESTIONS

In this paper we provide a generic framework for analyzing what properties of a similarity function are sufficient to allow it to be useful for clustering, under two natural relaxations of the clustering objective. We propose a measure of the *clustering complexity* of a given property that characterizes its information-theoretic usefulness for clustering, and analyze this complexity for a broad class of properties, as well as develop efficient algorithms that are able to take advantage of them.

Our work can be viewed both in terms of providing formal advice to the *designer* of a similarity function for a given clustering task (such as clustering query search results) and in terms of advice about what *algorithms* to use given certain beliefs about the relation of the similarity function to the clustering task. Our model also provides a better understanding of when (in terms of the relation between the similarity measure and the ground-truth clustering) different hierarchical linkage-based algorithms will fare better than others. Abstractly speaking, our notion of a *property* parallels that of a *data-dependent concept class* [35] (such as large-margin separators) in the context of classification.

Open questions: Broadly, one would like to analyze other natural properties of similarity functions, as well as to further explore and formalize other models of interactive feedback. In terms of specific open questions, for the average attraction property (Property 4) we have an algorithm that for $k = 2$ produces a list of size approximately $2^{O(1/\gamma^2 \ln 1/\epsilon)}$ and a lower bound on clustering complexity of $2^{\Omega(1/\gamma)}$. One natural open question is whether one can close that gap. A second open question is that for the strong stability of large subsets property (Property 7), our algorithm produces hierarchy but has larger running time substantially larger than that for the simpler stability properties. Can an algorithm with running time polynomial in k and $1/\gamma$ be developed? Can one prove stability properties for clustering based on spectral methods, e.g., the hierarchical clustering algorithm given in [14]? More generally, it would be interesting to determine whether these stability properties can be further weakened and still admit a hierarchical clustering. Finally, in this work we have focused on formalizing clustering with non-interactive feedback. It would be interesting to formalize clustering with other natural forms of feedback.

Acknowledgments: We thank Shai-Ben David, Sanjoy Dasgupta, and Andrew Tomkins for a number of useful discussions.

8. REFERENCES

[1] D. Achlioptas and F. McSherry. On spectral learning of mixtures of distributions. In *COLT*, 2005.

[2] N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: ranking and clustering. In *STOC*, pages 684–693, 2005.

[3] N. Alon, W. Fernandez de la Vega, R. Kannan, and M. Karpinski. Random sampling and approximation of max-csps. *Journal of Computer and Systems Sciences*, 67(2):212–243, 2003.

[4] N. Alon and N. Kahale. A spectral technique for coloring random 3-colorable graphs. *SIAM J. Computing*, 26(6):1733 – 1748, 1997.

[5] S. Arora and R. Kannan. Learning mixtures of arbitrary gaussians. In *ACM Symposium on Theory of Computing*, 2005.

[6] M.-F. Balcan and A. Blum. On a theory of learning with similarity functions. In *International Conference on Machine Learning*, 2006.

[7] M.-F. Balcan, A. Blum, and S. Vempala. On kernels, margins and low-dimensional mappings. *Machine Learning Journal*, 2006.

[8] M.-F. Balcan, A. Blum, and S. Vempala. A theory of similarity functions for clustering. Technical Report, CMU-CS-07-142, 2007.

[9] S. Ben-David. A framework for statistical clustering with constant time approximation for k-means clustering. *Machine Learning Journal*, 66(2-3):243 – 257, 2007.

[10] A. Blum, N. Bansal, and S. Chawla. Correlation clustering. *Machine Learning*, 56:89–113, 2004.

[11] S. Boucheron, O. Bousquet, and G. Lugosi. Theory of classification: a survey of recent advances. *ESAIM: Probability and Statistics*, 9:9:323–375, 2005.

[12] M. Charikar, S. Guha, E. Tardos, and D. B. Shmoy. A constant-factor approximation algorithm for the k-median problem. In *STOC*, 1999.

[13] M. Charikar, V. Guruswami, and A. Wirth. Clustering with qualitative information. In *FOCS*, pages 524–533, 2003.

[14] D. Cheng, R. Kannan, S. Vempala, and G. Wang. A divide-and-merge methodology for clustering. *ACM Trans. Database Syst.*, 31(4):1499–1525, 2006.

[15] A. Dasgupta, J. Hopcroft, J. Kleinberg, and M. Sandler. On learning mixtures of heavy-tailed distributions. In *46th IEEE Symposium on Foundations of Computer Science*, 2005.

[16] A. Dasgupta, J. E. Hopcroft, R. Kannan, and P. P. Mitra. Spectral clustering by recursive partitioning. In *ESA*, pages 256–267, 2006.

[17] S. Dasgupta. Learning mixtures of gaussians. In *Fortieth Annual IEEE Symposium on Foundations of Computer Science*, 1999.

[18] W. Fernandez de la Vega, M. Karpinski, C. Kenyon, and Y. Rabani. Approximation schemes for clustering problems. In *STOC*, 2003.

[19] L. Devroye, L. Gyorfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, 1996.

[20] A. Frieze and R. Kannan. Quick approximation to matrices and applications. *Combinatorica*, 19(2):175–220, 1999.

[21] R. Herbrich. *Learning Kernel Classifiers*. MIT Press, Cambridge, 2002.

[22] K. Jain and V. V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. *JACM*, 48(2):274 – 296, 2001.

[23] R. Kannan, H. Salmasian, and S. Vempala. The spectral method for general mixture models. In *Proc. COLT*, 2005.

[24] R. Kannan, S. Vempala, and A. Vetta. On clusterings: good, bad and spectral. *J. ACM*, 51(3):497–515, 2004.

[25] J. Kleinberg. An impossibility theorem for clustering. In *NIPS*, 2002.

[26] D. E. Knuth. *The Art of Computer Programming*. Addison-Wesley, 1997.

[27] R. I. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete structures. In *Proc. ICML*, 2002.

[28] F. McSherry. Spectral parititioning of random graphs. In *Proc. 43rd Symp. Foundations of Computer Science*, pages 529–537, 2001.

[29] B. Scholkopf, K. Tsuda, and J.-P. Vert. *Kernel Methods in Computational Biology*. MIT Press, 2004.

[30] J. Shawe-Taylor, P. L. Bartlett, R. C. Williamson, and M. Anthony. Structural risk minimization over data-dependent hierarchies. *IEEE Transactions on Information Theory*, 44(5):1926–1940, 1998.

[31] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[32] N. Srebro. How good is a kernel as a similarity function? In *Proc. 20th Annual Conference on Learning Theory*, 2007.

[33] C. Swamy. Correlation clustering: Maximizing agreements via semidefinite programming. In *SODA*, 2004.

[34] L.G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, 1984.

[35] V. N. Vapnik. *Statistical Learning Theory*. Wiley and Sons, 1998.

[36] S. Vempala and G. Wang. A spectral algorithm for learning mixture models. *J. Comp. Sys. Sci.*, 68(2):841–860, 2004.

APPENDIX

Theorem 3 *If $\mathcal{K}$ satisfies ν-***strict separation***, then so long as the smallest correct cluster has size greater than $5\nu n$, we can produce a tree such that the ground-truth clustering is ν-close to a pruning of this tree.*

PROOF. Let $S' \subseteq S$ be the set of $(1-\nu)n$ points such that $\mathcal{K}$ satisfies strict separation with respect to S'. Call the points in S' "good", and those not in S' "bad" (of course, goodness is not known to the algorithm). We first generate a list $\mathcal{L}$ of n^2 clusters such that, ignoring bad points, any cluster in the ground-truth is in the list. We can do this by for each point $x \in S$ creating a cluster of the t nearest points to it for each $4\nu n \le t \le n$.

We next run a procedure that removes points from clusters that are non-laminar with respect to each other without hurting any of the correct clusters, until the remaining set is fully laminar. Specifically, while there exist two clusters C and C' that are non-laminar with respect to each other, we do the following:

1. If either C or C' has size $\le 4\nu n$, delete it from the list. (By assumption, it cannot be one of the ground-truth clusters).

2. If C and C' are "somewhat disjoint" in that $|C \setminus C'| > 2\nu n$ and $|C' \setminus C| > 2\nu n$, each point $x \in C \cap C'$ chooses one of C or C' to belong to based on whichever of $C \setminus C'$ or $C' \setminus C$ respectively has larger *median* similarity to x. We then remove x from the cluster not chosen. Because each of $C \setminus C'$ and $C' \setminus C$ has a majority of good points, if one of C or C' is a ground-truth cluster (with respect to S'), all good points x in the intersection will make the correct choice. C and C' are now fully disjoint.

3. If C, C' are "somewhat equal" in that $|C \setminus C'| \le 2\nu n$ and $|C' \setminus C| \le 2\nu n$, we make them exactly equal based on the following related procedure. Each point x in the symmetric difference of C and C' decides *in* or *out* based on whether its similarity to the $(\nu n + 1)$st most-similar point in $C \cap C'$ is larger or smaller (respectively) than its similarity to the $(\nu n + 1)$st most similar point in $S \setminus (C \cup C')$. If x is a good point in $C \setminus C'$ and C is a ground-truth cluster (with respect to S'), then x will correctly choose *in*, whereas if C' is a ground-truth cluster then x will correctly choose *out*. Thus, we can replace C and C' with a single cluster consisting of their intersection plus all points x that chose *in*, without affecting the correct clusters.

4. If none of the other cases apply, it may still be there exist C, C' such that C "somewhat contains" C' in that $|C \setminus C'| > 2\nu n$ and $0 < |C' \setminus C| \le 2\nu n$. In this case, choose the largest such C and apply the same procedure as in Step 3, but only over the points $x \in C' \setminus C$. At the end of the procedure, we have $C \supseteq C'$ and the correct clusters have not been affected with respect to the good points.

Since all clusters remaining are laminar, we can now arrange them into a forest, which we then arbitrarily complete into a tree. □