Clustering via Similarity Functions: Theoretical Foundations and Algorithms*

Maria-Florina Balcan† Avrim Blum‡ Santosh Vempala§

Abstract

Problems of clustering data from pairwise similarity information arise in many different fields. Yet questions of which algorithms are best to use under what conditions, and how good a similarity measure is needed to produce accurate clusters for a given task remains poorly understood.

In this work we propose a new general framework for analyzing clustering from similarity information that directly addresses this question of what properties of a similarity measure are sufficient to cluster accurately and by what kinds of algorithms. We use this framework to show that a wide variety of interesting learning-theoretic and game-theoretic properties, including properties motivated by mathematical biology, can be used to cluster well, and we design new efficient algorithms that are able to take advantage of them. We consider two natural clustering objectives: (a) *list clustering*, where the algorithm's goal is to produce a small list of clusterings such that at least one of them is approximately correct, and (b) *hierarchical clustering*, where 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 further develop a notion of *clustering complexity* for a given property, analogous to notions of *capacity* in learning theory, which we analyze for a wide range of properties, giving tight upper and lower bounds. 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. This yields very efficient algorithms, though proving correctness requires subtle analysis based on regularity-type results.

Our framework can be viewed as an analog of discriminative models for supervised classification (i.e., the Statistical Learning Theory framework and the PAC learning model), where our goal is to cluster accurately given a *property* or relation the similarity function is believed to satisfy with respect to the ground truth clustering. More specifically our framework is analogous to that of data-dependent concept classes in supervised learning, where conditions such as the large margin property have been central in the analysis of kernel methods.

Our framework also makes sense for exploratory clustering, where the property itself can define the quality that makes a clustering desirable or interesting, and the hierarchy or list that our algorithms output will then contain approximations to all such desirable clusterings.

Keywords: Clustering, Machine Learning, Similarity Functions, Sample Complexity, Efficient Algorithms, Hierarchical Clustering, List Clustering, Linkage based Algorithms, Inductive Setting.

*A preliminary version of this paper appears as "A Discriminative Framework for Clustering via Similarity Functions," Proceedings of the 40*th* ACM Symposium on Theory of Computing (STOC), 2008. This work was supported in part by the National Science Foundation under grants CCF-0514922 and CCF-0721503, by a Raytheon fellowship, and by an IBM Graduate Fellowship.

†School of Computer Science, Georgia Institute of Technology. ninamf@cc.gatech.edu

‡Department of Computer Science, Carnegie Mellon University. avrim@cs.cmu.edu

§School of Computer Science, Georgia Institute of Technology. vempala@cc.gatech.edu

1 Introduction

Clustering is a central task in the analysis and exploration of data. It has a wide range of applications from computational biology to computer vision to information retrieval. It has many variants and formulations and it has been extensively studied in many different communities. However, while many different clustering algorithms have been developed, theoretical analysis has typically involved either making strong assumptions about the uniformity of clusters or else optimizing distance-based objective functions only secondarily related to the true goals.

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 [Charikar *et al.*, 1999, Kannan *et al.*, 2004, Jain and Vazirani, 2001]. That is, the graph is viewed as "ground truth" and 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 proteins by function, ground truth is really the unknown true topic or true function 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. That is, the goal is not so much to optimize a distance-based objective but rather to produce a clustering that agrees as much as possible with the unknown true categories. Alternatively, methods developed both in the algorithms and in the machine learning literature for learning mixtures of distributions [Achlioptas and McSherry, 2005, Arora and Kannan, 2001, Kannan *et al.*, 2005, Vempala and Wang, 2004, Dasgupta, 1999, Dasgupta *et al.*, 2005] explicitly have a notion of ground-truth clusters which they aim to recover. However, such methods make strong probabilistic 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 we might only be able to 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 conditions. Even nonparametric Bayesian models such as (hierarchical) Dirichlet Processes make fairly specific probabilistic assumptions about how data is generated [Teh *et al.*, 2006].

In this work, we develop a theoretical approach to analyzing clustering that is able to talk about *accuracy* of a solution produced without resorting to a probabilistic generative model for the data. In particular, motivated by work on similarity functions in the context of Supervised Learning that asks "what natural properties of a given kernel (or similarity) function $\mathcal{K}$ are sufficient to allow one to *learn* well?" [Herbrich, 2002, Shawe-Taylor and Cristianini, 2004, Scholkopf *et al.*, 2004, Balcan and Blum, 2006, Balcan *et al.*, 2006] 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 terms of its relation to the target. This is much like the approach taken in the study of kernel-based learning; we expand on this connection further in Section 1.1.

The main difficulty that appears when phrasing the problem in this general way is that in clustering there is no labeled data. Therefore, if one defines success as outputting *a single clustering* that closely approximates the correct clustering, then one needs to assume very strong conditions in order to cluster well. For example, if the similarity function provided by our expert is so good that $\mathcal{K}(x, y) > 0$ for all pairs x and y that should be in the same cluster, and $\mathcal{K}(x, y) < 0$ for all pairs x and y that should be in different clusters, then it would be trivial to use it to recover the

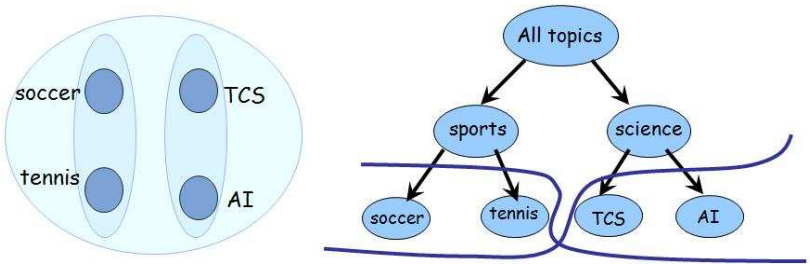

Figure 1: Data lies in four regions A, B, C, D (e.g., think of as documents on tennis, soccer, TCS, and AI). Suppose that $\mathcal{K}(x, y) = 1$ if x and y belong to the same region, $\mathcal{K}(x, y) = 1/2$ if $x \in A$ and $y \in B$ or if $x \in C$ and $y \in D$, and $\mathcal{K}(x, y) = 0$ otherwise. Even assuming that all points are more similar to all 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.

clusters. However, if we slightly weaken this condition to just 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 (but without a common cutoff value), 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 highly distinct 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 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 [Elias, 1957, Guruswami and Sudan, 1999] 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 start at the top level and 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." They are also natural for settings where one has additional side constraints that an algorithm could then use in a subsequent post-processing step, as in the database deduplication work of [Chaudhuri *et al.*, 2007]. We then show that with these relaxations, a number of interesting, natural learning-theoretic and game-theoretic properties of the similarity measure are sufficient to be able to cluster well. For some of these properties we prove guarantees for traditional clustering algorithms, while for other more general properties, we show such methods may fail and instead develop new algorithmic techniques that are able to take advantage of them. We also define a notion of the *clustering complexity* of a given property that expresses the length of the shortest list of clusterings needed to ensure that at least one of them is approximately correct.

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? Generically speaking, our analysis provides a unified framework for understanding under what conditions a similarity function can be used to find a good approximation to the ground-truth clustering.

Our framework also provides a natural way to formalize the problem of *exploratory clustering*, where the similarity function is given and the property itself can be viewed as a user-provided definition of the criterion for an "interesting clustering". In this case, our results can be thought of as asking for a compact representation (via a tree), or a short list of clusterings, that contains all clusterings of interest. The clustering complexity of a property in this view is then an upper-bound on the number of "substantially different" interesting clusterings.

1.1 Perspective

There has been significant work in machine learning and theoretical computer science on clustering or learning with mixture models [Achlioptas and McSherry, 2005, Arora and Kannan, 2001, Duda *et al.*, 2001, Devroye *et al.*, 1996, Kannan *et al.*, 2005, Vempala and Wang, 2004, Dasgupta, 1999]. 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 specific than the types of properties we will be considering. This is similarly the case with work on planted partitions in graphs [Alon and Kahale, 1997, McSherry, 2001, Dasgupta *et al.*, 2006]. Abstractly speaking, this view of clustering parallels the *generative* classification setting [Devroye *et al.*, 1996], while the framework we propose parallels the *discriminative* classification setting (i.e. the PAC model of Valiant [Valiant, 1984] and the Statistical Learning Theory framework of Vapnik [Vapnik, 1998]).

In the PAC model for learning [Valiant, 1984], the basic object of study is a *concept class*, and one asks what natural classes are efficiently learnable and by what algorithms. In our setting, the basic object of study is a *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, ideally finding the simplest efficient algorithm for each such property. In some cases, we can even characterize necessary and sufficient conditions for natural algorithms such as single linkage to be successful in our framework.

Our framework also makes sense for exploratory clustering, where rather than a single target clustering, one would like to produce all clusterings satisfying some condition. In that case, our goal can be viewed as to output either explicitly (via a list) or implicitly (via a tree) an ϵ-cover of the set of all clusterings of interest.

1.2 Our Results

We provide a general PAC-style framework for analyzing what properties of a similarity function are sufficient to allow one to cluster well under the above two relaxations (list and tree) of the clustering objective. We analyze a wide variety of natural properties in this framework, both from an algorithmic and information theoretic point of view. Specific results include:

- As a warmup, we show that the strong property discussed above (that all points are more similar to points in their own cluster than to any points in any other cluster) is sufficient to cluster well by the simple single-linkage algorithm. Moreover we show that a much less restrictive "agnostic" version is sufficient to cluster well but using a more sophisticated approach (Property 2 and Theorem 3.3). We also describe natural properties that characterize exactly when single linkage will be successful (Theorem 5.1).

- We consider a family of natural stability-based properties, showing (Theorems 5.2 and 5.4) that a natural generalization of the "stable marriage" property is sufficient to produce a hierarchical clustering via a common *average linkage* algorithm. 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 (see Property 8) and it has close connections with notions analyzed in Mathematical Biology [Bryant and Berry, 2001]. Moreover, we show that a significantly weaker notion of stability is also sufficient to produce a hierarchical clustering, and to prove this we develop a new algorithmic technique based on generating candidate clusters and then molding them using pairwise consistency tests (Theorem 5.7).

- We show that a weaker "average-attraction" property is provably not enough to produce a hierarchy but is sufficient to produce a small list of clusterings (Theorem 4.1). We then give generalizations to even weaker conditions that generalize the notion of large-margin kernel functions, using recent results in learning theory (Theorem 4.4).

- We 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 here (Theorems 6.1 and 6.2).

We define the *clustering complexity* of a given property (the minimum possible list length that an algorithm could hope to guarantee) and provide both upper and lower bounds for the properties we consider. This notion is analogous to notions of capacity in classification [Boucheron *et al.*, 2005, Devroye *et al.*, 1996, Vapnik, 1998] and it provides a formal measure of the inherent usefulness of a given property.

We also 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 [Frieze and Kannan, 1999, Alon *et al.*, 2003] (Theorem 7.3).

More generally, the proposed 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 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.

One concrete implication of this framework is that we can use it to get around certain fundamental limitations that arise in the approximation-algorithms approach to clustering. In particular, in subsequent work within this framework, [Balcan *et al.*, 2009] and [Balcan and Braverman, 2009] have shown that the implicit assumption made by approximation algorithms for the standard

k-means, k-median, and min-sum objectives (that nearly-optimal clusterings according to the objective will be close to the desired clustering in terms of accuracy) imply structure one can use to achieve performance as good as if one were able to optimize these objectives *even to a level that is known to be NP-hard*.

1.3 Related Work

We review here some of the existing theoretical approaches to clustering and how they relate to our framework.

Mixture and Planted Partition Models: In mixture models, one assumes that data is generated by a mixture of simple probability distributions (e.g., Gaussians), one per cluster, and aims to recover these component distributions. As mentioned in Section 1.1, there has been significant work in machine learning and theoretical computer science on clustering or learning with mixture models [Achlioptas and McSherry, 2005, Arora and Kannan, 2001, Duda *et al.*, 2001, Devroye *et al.*, 1996, Kannan *et al.*, 2005, Vempala and Wang, 2004, Dasgupta, 1999]. That work is similar to our framework in that there is an explicit notion of a correct ground-truth clustering of the data points. However, unlike our framework, mixture models make very specific probabilistic assumptions about the data that generally imply a large degree of intra-cluster uniformity. For instance, the example of Figure 1 would not fit a typical mixture model well if the desired clustering was {sports, TCS, AI}. In planted partition models [Alon and Kahale, 1997, McSherry, 2001, Dasgupta *et al.*, 2006], one begins with a set of disconnected cliques and then adds random noise. These models similarly make very specific probabilistic assumptions, implying substantial intra-cluster as well as inter-cluster uniformity.

Approximation Algorithms: Work on approximation algorithms, like ours, makes no probabilistic assumptions about the data. Instead, one chooses some objective function (e.g., k-median, k-means, min-sum, or correlation clustering), and aims to develop algorithms that approximately optimize that objective [Ailon *et al.*, 2005, Bartal *et al.*, 2001, Charikar *et al.*, 1999, Kannan *et al.*, 2004, Jain and Vazirani, 2001, de la Vega *et al.*, 2003]. For example the best known approximation algorithm for the k-median problem is a $(3+\epsilon)$-approximation [Arya *et al.*, 2004], and the best approximation for the min-sum problem in general metric spaces is a $O(\log^{1+\delta} n)$-approximation [Bartal *et al.*, 2001]. However, while often motivated by problems such as clustering search results by topic, the approximation algorithms approach does not explicitly consider how close the solution produced is to an underlying desired clustering, and without any assumptions the clusterings produced might be quite far away. If the true goal is indeed to achieve low error with respect to a target clustering, then one is *implicitly* making the 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. We can make this *explicit* by saying that a data set satisfies the (c, ϵ) property for some objective function Φ if all c-approximations to Φ on this data are ϵ-close to the target clustering. In Section 6 we show that for some of these objectives, this assumption is in fact a special case of properties considered and analyzed here. Subsequent to this work, Balcan et al. [2009] and Balcan and Braverman [2009] have further analyzed these assumptions, giving algorithms that can find accurate clusterings under the (c, ϵ) property for a number of common objectives Φ (including k-median, k-means, and min-sum) even for values c such that finding a c-approximation to the objective is NP-hard. This shows that for the goal of achieving low error

on the data, one can bypass approximation hardness results by making these implicit assumptions explicit, and using the structure they imply. We discuss these results further in Section 8.

Bayesian and Hierarchical Bayesian Clustering: Bayesian methods postulate a prior over probabilistic models (ground truths), which in turn generate the observed data. Given the observed data, there is then a well-defined highest-probability model that one can then hope to compute. For example, Bayesian mixture models place a prior over the parameters of the mixture; nonparametric models such as the Dirichlet / Chinese Restaurant Process allow for the number components to be a random variable as well, which one can then infer from the data [Teh *et al.*, 2006]. Hierarchical Bayesian methods model the ground truth itself as a hierarchy, allowing for sharing of model components across multiple clusters [Teh *et al.*, 2006, Heller, 2008]. Our framework is similar to these in that our goal is also to approximate a target clustering. However, unlike these approaches, our framework makes no probabilistic assumptions about the data or target clustering. Instead, we assume only that it is consistent with the given similarity measure according to the property at hand, and our use of a hierarchy is as a relaxation on the output rather than an assumption about the target.

Identifying special clusters: Bryant and Berry [2001] consider and analyze various notions of "stable clusters" and design efficient algorithms to produce them. While their perspective is different from ours, some of the definitions they consider are related to our simplest notions of strict separation and stability and further motivate the notions we consider. Bandelt and Dress [1989] also consider the problem of identifying clusters satisfying certain consistency conditions, motivated by concerns in computational biology. For more discussion see Sections 3, 5, and Appendix A.

Axiomatic Approaches and Other Work on Clustering: There have recently been a number of results on axiomatizing clustering in the sense of describing natural properties of *algorithms*, such as scale-invariance and others, and analyzing which collections of such properties are or are not achievable [Kleinberg, 2002, Ackerman and Ben-David., 2008]. In this approach there is no notion of a ground-truth clustering, however, and so the question is whether an algorithm will satisfy certain conditions rather than whether it produces an accurate output. Related theoretical directions includes work on comparing clusterings [Meila, 2003, Meila, 2005], and on efficiently testing if a given data set has a clustering satisfying certain properties [Alon *et al.*, 2000]. There is also other interesting work addressing stability of various clustering algorithms with connections to model selection [Ben-David *et al.*, 2006, Ben-David *et al.*, 2007].

Relation to learning with Kernels: Some of the questions we address can be viewed as a generalization of questions studied in supervised learning that ask what properties of similarity functions (especially kernel functions) are sufficient to allow one to *learn* well [Balcan and Blum, 2006, Balcan *et al.*, 2006, Herbrich, 2002, Shawe-Taylor and Cristianini, 2004, Scholkopf *et al.*, 2004]. For example, it is well-known 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 such as $e^{-d(x,x')}$, but rather may be a derived function based on the entire dataset. For example, in the *diffusion kernel* of [Kondor and Lafferty, 2002], 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.

Inductive Setting: 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., [Mishra *et al.*, 2001, Ben-David, 2007, Czumaj and Sohler, 2004]) 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 in terms of this objective value with respect to the underlying distribution. This work does not assume a target clustering.

2 Definitions and Preliminaries

We consider a clustering problem (S, ℓ) specified as follows. Assume we have a data set S of n objects. Each $x \in S$ has some (unknown) "ground-truth" label $\ell(x)$ in $Y = \{1, \ldots, k\}$, where we will think of k as much smaller than n. We let $C_i = \{x \in S : \ell(x) = i\}$ denote the set of points of label i (which could be empty), and denote the target clustering as $\mathcal{C} = \{C_1, \ldots, C_k\}$. The goal is to produce a hypothesis $h : S \to Y$ of low error up to permutation of label names. Formally, we define the error of h to be

$$err(h) = \min_{\sigma \in \mathcal{S}_k} \left[\Pr_{x \in S} [\sigma(h(x)) \neq \ell(x)] \right],$$

where $\mathcal{S}_k$ is the set of all permutations on $\{1, \ldots, k\}$. Equivalently, the error of a clustering $\mathcal{C}' = \{C'_1, \ldots, C'_k\}$ is $\min_{\sigma \in \mathcal{S}_k} \frac{1}{n} \sum_i |C_i - C'_{\sigma(i)}|$. It will be convenient to extend this definition to clusterings $\mathcal{C}'$ of $k' > k$ clusters: in this case we simply view the target as having $k' - k$ additional empty clusters $C'_{k+1}, \ldots, C'_{k'}$ and apply the definition as above with "k'" as "k". We will assume that a target error rate ϵ, as well as the number of target clusters 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 $[-1, 1]$.[1] 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 on analyzing natural properties of a similarity function $\mathcal{K}$ that are sufficient for an algorithm to produce accurate clusterings with respect to the ground-truth clustering $\mathcal{C}$. Formally, a property $\mathcal{P}$ is a relation $\{(\mathcal{C}, \mathcal{K})\}$ between the target clustering and the similarity function and we say that $\mathcal{K}$ has property $\mathcal{P}$ with respect to $\mathcal{C}$ if $(\mathcal{C}, \mathcal{K}) \in \mathcal{P}$. For example, one (strong) property would be that all points x are more similar to all points x' in their own cluster than to any x'' in any other cluster– 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. We will also consider intermediate "stability" conditions.

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 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

[1] 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.

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 [Knuth, 1997], so this model is at least as strict as the list model.

Transductive vs Inductive. Clustering is typically posed as a "transductive" problem [Vapnik, 1998] in that we are asked to cluster a *given* set of points S. We can also consider an *inductive* model in which S 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 7 we show how our algorithms can be adapted to the inductive setting.

Realizable vs Agnostic. For most of the properties we consider here, our assumptions are analogous to the *realizable* case in supervised learning and our goal is to get ϵ-close to the target (in a tree or list) for any desired $\epsilon > 0$. For other properties, our assumptions are more like the *agnostic* case in that we will assume only that $1-\nu$ fraction of the data satisfies a certain condition. In these cases our goal is to get $\nu + \epsilon$-close to the target.

Notation. 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

$$\mathcal{K}(A, B) = \mathbf{E}_{x \in A, x' \in B}\big[\mathcal{K}(x, x')\big].$$

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

$$d(g, h) = \min_{\sigma \in \mathcal{S}_k} \left[\Pr_{x \in S} \left[\sigma(h(x)) \neq g(x)\right] \right],$$

i.e., the fraction of points in the symmetric difference under the optimal renumbering of the clusters.

As mentioned above, we are interested in analyzing natural *properties* that we might ask a similarity function to satisfy with respect to the ground truth clustering. For a given property, one key quantity we will be interested in is 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*

k-clustering $\mathcal{C}'$ consistent with the property (i.e., satisfying $(\mathcal{C}', \mathcal{K}) \in \mathcal{P}$) must be ϵ-close to some clustering in the list. 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 [Boucheron *et al.*, 2005, Devroye *et al.*, 1996, Vapnik, 1998] and it provides a formal measure of the inherent usefulness of a given property.

Computational Complexity. In the transductive case, our goal will be to produce a list or a tree in time polynomial in n and ideally polynomial in ϵ and k as well. We will indicate when our running times involve a non-polynomial dependence on these parameters. In the inductive case, we want the running time to depend only on k and ϵ and to be independent of the size of the overall instance space X, under the assumption that we have an oracle that in constant time can sample a random point from X.

2.1 Structure of this paper

In the following sections we analyze both the clustering complexity and the computational complexity of several natural properties and provide efficient algorithms to take advantage of similarity functions satisfying them. We start by analyzing the strict separation property as well as a natural relaxation in Section 3. We then analyze a much weaker average-attraction property in Section 4 which has close connections to large margin properties studied in Learning Theory [Balcan and Blum, 2006, Balcan *et al.*, 2006, Herbrich, 2002, Shawe-Taylor and Cristianini, 2004, Scholkopf *et al.*, 2004].) 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. In Section 6 we give formal relationships between these properties and those considered implicitly by approximation algorithms for standard clustering objectives. Then in Section 7 we consider clustering in the inductive setting.

Our framework allows one to study computational hardness results as well. While our focus is on getting positive algorithmic results, we discuss a few simple hardness results in Section B.1.

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 $x \in S$ 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 3.2). 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 3.1 *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 that used 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. For the case that $\mathcal{K}$ is symmetric, it is known that single-linkage will produce a tree of the desired form (see, e.g., [Bryant and Berry, 2001]). However, when $\mathcal{K}$ is asymmetric, single-linkage may fail and instead we use a more "Boruvka-inspired" algorithm.

Theorem 3.2 *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 we can use the single linkage algorithm (i.e., Kruskal's algorithm) to produce the desired tree. 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 procedure maintains the invariant that at each step the current clustering is laminar with respect to the ground-truth (every cluster is either contained in, equal to, or a union of target clusters). In particular, 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. Since at each step the clustering is laminar with respect to the target, the target clustering must be a pruning of the final tree. Unfortunately, if $\mathcal{K}$ is not symmetric, then single linkage may fail.[2] 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 by the directed edges (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. ■

We can also consider an agnostic version of the strict separation property, where we relax the condition to require only that $\mathcal{K}$ satisfies strict separation with respect to *most* of the data. We distinguish here two forms of this relaxation: an "easy version" for which simple bottom-up algorithms are still successful and a harder, more general version which requires a more involved approach.

In the easy version, we suppose that there exists a set S' containing most of S such that all $x \in S'$ are more similar to all $x' \in C(x) \cap S'$ than to any $x'' \in S - (C(x) \cap S')$. That is, the points not in S' act as distant outliers. We can address this version by noticing that this property implies that $\mathcal{K}$ satisfies strict separation with respect to a modified version $\tilde{\mathcal{C}}$ of the target clustering in which each point in $S - S'$ is assigned its own cluster. Since $\tilde{\mathcal{C}}$ has low error, Theorem 3.2 implies that single-linkage will still produce a low-error tree.

[2]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.

In the harder, general version, we allow points in $S - S'$ to behave arbitrarily, without any requirement for consistency with respect to similarities in S'. This is analogous to the setting of learning with malicious noise or agnostic learning. Formally, we define:

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). That is, for all $x, x', x'' \in S'$ with $x' \in C(x)$ and $x'' \notin C(x)$ we have $\mathcal{K}(x, x') > \mathcal{K}(x, x'')$.*

Note that now even a single point in $S - S'$ (i.e., $\nu = 1/n$) is enough to cause the single-linkage algorithm to fail. For instance, given set S' satisfying strict separation such that $\mathcal{K}(x, y) < 1$ for all $x, y \in S'$, add a new point u such that $\mathcal{K}(u, x) = 1$ for all $x \in S'$. Single linkage will now just connect every point to u. Nonetheless, using a different non-bottom-up style algorithm we can show the following.

Theorem 3.3 *If $\mathcal{K}$ satisfies* **ν-strict separation**, *then so long as the smallest target 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.*

We defer the algorithm and proof to Section 6, where we also show that properties implicitly assumed by approximation algorithms for standard graph-based objective functions can be viewed as special cases of the ν-strict separation property.

Strict separation and spectral partitioning: We end this section by pointing out that even though the strict separation property is quite strong, a similarity function satisfying this property can still fool a top-down spectral clustering approach.

In particular, Figure 2 shows that it is possible for a similarity function to satisfy the strict separation property for which Theorem 3.2 gives a good algorithm, but nonetheless to fool a straightforward spectral (top down) clustering approach.

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 3 *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:*

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

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 cluster it correctly. Note, however, that unlike properties considered in the previous section, average attraction is not sufficient to cluster in the tree model. Consider, for instance, three regions R_1, R_2, R_3 with $n/3$ points each of similarity 1 within each region and similarity 0 between regions; any grouping $(R_i, R_j \cup R_k)$ of these three regions into two clusters satisfies the $(0, 1/2)$-average attraction property and yet these 2-clusterings are not laminar with respect to each other. On the

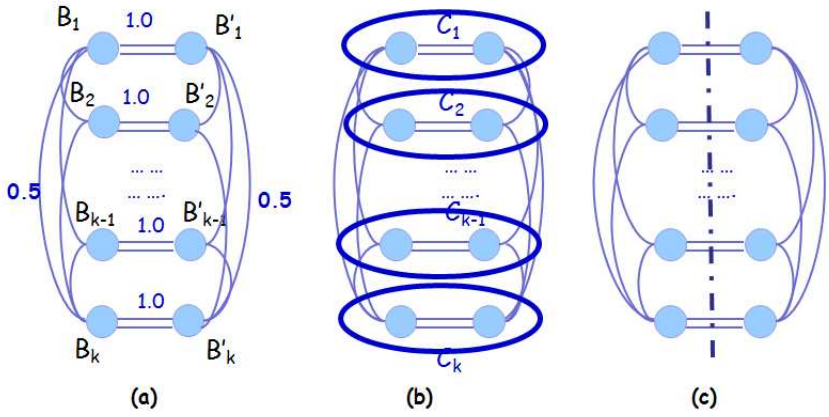

Figure 2: Consider $2k$ sets $B_1, B_2, \ldots, B_k, B'_1, B'_2, \ldots, B'_k$ of equal probability mass. Points inside the same set have similarity 1. Assume that $\mathcal{K}(x, x') = 1$ if $x \in B_i$ and $x' \in B'_i$. Assume also $\mathcal{K}(x, x') = 0.5$ if $x \in B_i$ and $x' \in B_j$ or $x \in B'_i$ and $x' \in B'_j$, for $i \neq j$; let $\mathcal{K}(x, x') = 0$ otherwise. Let $C_i = B_i \cup B'_i$, for all $i \in \{1, \ldots, k\}$. It is easy to verify that the clustering $C_1, \ldots, C_k$ is consistent with Property 1 (part (b)). However, for k large enough the cut of min-conductance is the cut that splits the graph into parts $\{B_1, B_2, \ldots, B_k\}$ and $\{B'_1, B'_2, \ldots, B'_k\}$ (part (c)).

other hand, we can cluster in the list model and give nearly tight upper and lower bounds on the clustering complexity of this property. Specifically, 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.

Algorithm 1 Sampling Based Algorithm, List Model

Input: Data set S, similarity function $\mathcal{K}$, parameters $k, N, s \in Z^+$.

- Set $\mathcal{L} = \emptyset$.
- Repeat N times

  For $k' = 1, \ldots, k$ do:

  - Pick a set $R_S^{k'}$ of s 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 $h(x) = \text{argmax}_{i \in \{1, \ldots k'\}}[\mathcal{K}(x, R_S^i)]$. Add h to $\mathcal{L}$.

- Output the list $\mathcal{L}$.

Theorem 4.1 *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 $s = \frac{4}{\gamma^2} \ln\left(\frac{8k}{\epsilon\delta}\right)$ and $N = \left(\frac{2k}{\epsilon}\right)^{\frac{4k}{\gamma^2} \ln\left(\frac{8k}{\epsilon\delta}\right)} \ln\left(\frac{1}{\delta}\right)$ 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 $\left(\frac{\epsilon}{2k}\right)^s$ probability that ${R_S}^i \subseteq C_i$, and so at least a $\left(\frac{\epsilon}{2k}\right)^{ks}$ 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 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

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

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. ■

Theorem 4.1 implies a corresponding upper bound on the (ϵ,k)-clustering complexity of the $(\epsilon/2,\gamma)$-average attraction property by the following Lemma.

Lemma 4.2 *Suppose there exists a randomized algorithm for a given similarity function $\mathcal{K}$ and property $\mathcal{P}$ that produces a list of at most L clusterings such that for any k-clustering $\mathcal{C}'$ consistent with $\mathcal{P}$ (i.e., $(\mathcal{C}',\mathcal{K}) \in \mathcal{P}$), with probability $\geq 1/2$ at least one of the clusterings in the list is $\epsilon/2$-close to $\mathcal{C}'$. Then the (ϵ,k)-clustering complexity of $(\mathcal{K},\mathcal{P})$ is at most $2L$.*

Proof: Fix $\mathcal{K}$ and let $h_1,\ldots,h_t$ be a maximal ϵ-net of k-clusterings consistent with $\mathcal{P}$; that is, $d(h_i,h_j) > \epsilon$ for all $i \neq j$, and for any h consistent with $\mathcal{P}$, $d(h,h_i) \leq \epsilon$ for some i.

By the triangle inequality, any given clustering h can be $\epsilon/2$-close to at most one h_i. This in turn implies that $t \leq 2L$. In particular, for *any* list of L k-clusterings, if $i \in \{1,\ldots,t\}$ at *random*, then the probability that some clustering in the list is $\epsilon/2$-close to h_i is at most L/t. Therefore, for any randomized procedure for producing such a list there must *exist* h_i such that the probability is at most L/t. By our given assumption, this must be at least $1/2$.

Finally, since $h_1,\ldots,h_t$ satisfy the condition that for any h consistent with $\mathcal{P}$, $d(h,h_i) \leq \epsilon$ for some i, the (ϵ,k)-clustering complexity of $(\mathcal{K},\mathcal{P})$ is at most $t \leq 2L$. ■

Note that the bound of Theorem 4.1 combined with Lemma 4.2, however, is not polynomial in k and $1/\gamma$. We can also give a lower bound showing that the exponential dependence on k and $1/\gamma$ is necessary.

Theorem 4.3 *For $\epsilon \leq 1/4$, the (ϵ, k)-clustering complexity of the $(0, \gamma)$-average attraction property is $\Omega(k^{\frac{k}{8\gamma}})$ for $k > (2e)^4$ and $\gamma \leq \frac{1}{3 \ln 2k}$.*

Proof: Consider $N = \frac{k}{\gamma}$ regions $\{R_1, \ldots, R_{k/\gamma}\}$ each with $\gamma n/k$ points. Assume $\mathcal{K}(x, x') = 1$ if x and x' belong to the same region R_i and $\mathcal{K}(x, x') = 0$, otherwise. We now show that we can have at least $k^{\frac{k}{8\gamma}}$ clusterings that are at distance at least $1/2$ from each other and yet satisfy the $(0, \gamma)$-average attraction property. We do this using a probabilistic construction. Specifically imagine that we construct the clustering by putting each region R_i uniformly at random into a cluster C_r, $r \in \{1, ..., k\}$, with equal probability $1/k$. Given a permutation π on $\{1, \ldots, k\}$, let X_π be a random variable which specifies, for two clusterings $\mathcal{C}, \mathcal{C}'$ chosen in this way, the number of regions R_i that agree on their clusters in $\mathcal{C}$ and $\mathcal{C}'$ with respect to permutation π (i.e., $R_i \in C_j$ and $R_i \in C'_{\pi(j)}$ for some j). We have $\mathbf{E}[X_\pi] = N/k$ and from the Chernoff bound we know that $\Pr[X_\pi \geq t\mathbf{E}[X_\pi]] \leq \left(\frac{e^{t-1}}{t^t}\right)^{\mathbf{E}[X_\pi]}$. So, considering $t = k/2$ we obtain that

$$\Pr[X_\pi \geq N/2] \leq \left(\frac{2e}{k}\right)^{(k/2)(N/k)} = \left(\frac{2e}{k}\right)^{k/(2\gamma)}.$$

So, we get that the probability that in a list of size m there exist a permutation π and two clusterings that agree on more than $N/2$ regions under π is at most $m^2 k! \left(\frac{2e}{k}\right)^{k/(2\gamma)}$. For $m = k^{\frac{k}{8\gamma}}$ this is at most $k^{k+\frac{k}{4\gamma}-\frac{k}{2\gamma}}(2e)^{\frac{k}{2\gamma}} \leq k^{-\frac{k}{8\gamma}}(2e)^{\frac{k}{2\gamma}} = o(1)$, where the second-to-last step uses $\gamma < 1/8$ and the last step uses $k > (2e)^4$.

We now show that there is at least a $1/2$ probability that none of the clusters have more than $2/\gamma$ regions and so the clustering satisfies the $\gamma/2$ average attraction property. Specifically, for each cluster, the chance that it has more than $2/\gamma$ regions is at most $e^{-1/3\gamma}$, which is at most $\frac{1}{2k}$ for $\gamma \leq \frac{1}{3 \ln 2k}$.

So, discarding all clusterings that do not satisfy the property, we have with high probability constructed a list of $\Omega(k^{\frac{k}{8\gamma}})$ clusterings satisfying $\gamma/2$ average attraction all at distance at least $1/2$ from each other. Thus, such a clustering must exist, and therefore the clustering complexity for $\epsilon < 1/4$ is $\Omega(k^{\frac{k}{8\gamma}})$. ■

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 [Vapnik, 1998, Shawe-Taylor *et al.*, 1998] as shown in [Balcan and Blum, 2006, Srebro, 2007, Balcan *et al.*, 2008].

Property 4 *A similarity function $\mathcal{K}$ satisfies the (ν, γ, τ)-**generalized large margin** property for the clustering problem (S, ℓ) if there exist a (possibly probabilistic) indicator function R (viewed as indicating a set of "reasonable" points) such that:*

1. *At least $1 - \nu$ fraction of examples x satisfy:*

$$\mathbf{E}_{x'}[\mathcal{K}(x, x')|R(x'), \ell(x) = \ell(x')] \geq \mathcal{K}_{x' \in C_r}[\mathcal{K}(x, x')|R(x'), \ell(x') = r] + \gamma,$$

for all clusters $r \in Y, r \neq \ell(x)$.

2. We have $\Pr_x[R(x)|\ell(x) = r] \geq \tau$, *for all* r.

If we have $\mathcal{K}$ a similarity function satisfying the (ν, γ, τ)-generalized large margin property for the clustering problem (S, ℓ), then we can again cluster well in the list model. Specifically:

Theorem 4.4 *Let $\mathcal{K}$ be a similarity function satisfying the (ν, γ, τ)-generalized large margin property for the clustering problem (S, ℓ). Using Algorithm 1 with the parameters $s = \frac{4}{\tau\gamma^2}\ln\left(\frac{8k}{\epsilon\delta}\right)$ and $N = \left(\frac{2k}{\tau\epsilon}\right)^{\frac{4k}{\tau\gamma^2}\ln\left(\frac{8k}{\epsilon\delta}\right)}\ln(\frac{1}{\delta})$ we can produce a list of at most $k^{O\left(\frac{k}{\gamma^2}\ln\left(\frac{1}{\tau\epsilon}\right)\ln\left(\frac{k}{\epsilon\delta}\right)\right)}$ clusterings such that with probability $1-\delta$ at least one of them is $(\nu+\epsilon)$-close to the ground-truth.*

Proof: The proof proceeds as in theorem 4.1. 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 $\left(\frac{\epsilon\tau}{2k}\right)^s$ probability that ${R_S}^i \subseteq C_i$, and so at least a $\left(\frac{\epsilon\tau}{2k}\right)^{ks}$ probability that ${R_S}^i \subseteq C_i$ for all $i \leq k'$. Thus the number of rounds $\left(\frac{2k}{\epsilon\tau}\right)^{\frac{4k}{\gamma^2}\ln\left(\frac{8k}{\epsilon\delta}\right)}\ln(\frac{1}{\delta})$ is large enough so that with probability at least $1-\delta/2$, in at least one of the N rounds we have ${R_S}^i \subseteq C_i$ for all $i \leq k'$. Let us fix one such good round. The remainder of the argument now continues exactly as in the proof of Theorem 4.1 and we have that the clustering induced by the sets picked in this round has error at most $\nu+\epsilon$ with probability at least $1-\delta$. ■

A too-weak property: One could imagine further relaxing the average attraction property to simply require that the average similarity *within* any ground-truth cluster C_i is larger by γ than the average similarity *between* C_i and any other ground-truth cluster C_j. However, even for $k = 2$ and $\gamma = 1/4$, this is *not sufficient* to produce clustering complexity independent of (or even polynomial in) n. In particular, let us define:

Property 5 *A similarity function $\mathcal{K}$ satisfies the* **γ-weak average attraction** *property if for all $i \neq j$ we have $\mathcal{K}(C_i, C_i) \geq \mathcal{K}(C_i, C_j) + \gamma$.*

Then we have:

Theorem 4.5 *The γ-weak average attraction property has clustering complexity exponential in n even for $k = 2$, $\gamma = 1/4$, and $\epsilon = 1/8$.*

Proof: Partition S into two sets A, B of $n/2$ points each, and let $\mathcal{K}(x, x') = 1$ for x, x' in the same set (A or B) and $\mathcal{K}(x, x') = 0$ for x, x' in different sets (one in A and one in B). Consider any 2-clustering $\{C_1, C_2\}$ such that C_1 contains 75% of A and 25% of B (and so C_2 contains 25% of A and 75% of B). For such a 2-clustering we have $\mathcal{K}(C_1, C_1) = \mathcal{K}(C_2, C_2) = (3/4)^2 + (1/4)^2 = 5/8$ and $\mathcal{K}(C_1, C_2) = 2(1/4)(3/4) = 3/8$. Thus, the γ-weak average attraction property is satisfied for $\gamma = 1/4$. However, not only are there exponentially many such 2-clusterings, but two randomly-chosen such 2-clusterings have expected distance 3/8 from each other, and the probability their distance is at most 1/4 is exponentially small in n. Thus, any list of clusterings such that all such 2-clusterings have distance at most $\epsilon = 1/8$ to some clustering in the list must have length exponential in n. ■

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.

5.1 Max Stability

We begin with a stability property that relaxes strict separation and asks that the ground truth clustering be "stable" in a certain sense. Interestingly, we show this property *characterizes* the desiderata for single-linkage in that it is both necessary and sufficient for single-linkage to produce a tree such that the target clustering is a pruning of the tree.

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

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

Theorem 5.1 *For a symmetric similarity function $\mathcal{K}$, Property 6 is a necessary and sufficient condition for single-linkage to produce a tree such that the ground-truth clustering is a pruning of this tree.*

Proof: We first show that if $\mathcal{K}$ satisfies Property 6, then the single linkage algorithm will produce a correct tree. The proof proceeds exactly as in Theorem 3.2: by induction we maintain the invariant that at each step the current clustering is laminar with respect to the ground-truth. In particular, if some current cluster $A \subset C_r$ for some target cluster C_r is merged with some other cluster B, Property 6 implies that B must also be contained within C_r.

In the other direction, if the property is not satisfied, then there exist A, A' such that $\mathcal{K}_{max}(A, C_r \setminus A) \leq \mathcal{K}_{max}(A, A')$. Let y be the point not in C_r maximizing $\mathcal{K}(A, y)$. Let us now watch the algorithm until it makes the first merge between a cluster C contained within A and a cluster C' disjoint from A. By assumption it must be the case that $y \in C'$, so the algorithm will fail. ■

5.2 Average Stability

The above property states that no piece A of some target cluster C_r would prefer to join another piece A' of some $C_{r'}$ if we define "prefer" according to maximum similarity between pairs. A perhaps more natural notion of stability is to define "prefer" with respect to the average. The result is a notion much like stability in the "stable marriage" sense, but for clusterings. In particular we define the following.

Property 7 *A similarity function $\mathcal{K}$ satisfies the* **strong stability** *property for the clustering problem (S, ℓ) if for all target clusters C_r, $C_{r'}$, $r \neq r'$, for all $A \subset C_r$, $A' \subseteq C_{r'}$ we have*

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

Property 8 *A similarity function $\mathcal{K}$ satisfies the* **weak stability** *property for the clustering problem (S, ℓ) if for all target clusters 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. Bryant and Berry [2001] define a quite similar condition to strong stability (though technically a bit stronger) motivated by concerns in computational biology. We discuss formal relations between their definition and ours in Appendix A. 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 of the regions A, B, C, D 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 using the *average*-linkage algorithm (when $\mathcal{K}$ is symmetric) or a cycle-collapsing version when $\mathcal{K}$ is not symmetric. We prove these results below, after which in Section 5.3 we analyze an even more general stability notion.

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 5.2 *Let $\mathcal{K}$ be a symmetric similarity function satisfying strong stability. Then the average single-linkage algorithm constructs a binary tree such that the ground-truth clustering is a pruning of this tree.*

Proof: 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 7, $\mathcal{K}(C', C_r - C') > \mathcal{K}(C', x)$ 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. ■

If the similarity function is asymmetric then even if strong stability is satisfied the average linkage algorithm may fail. However, as in the case of strict separation, for the asymmetric case we can use a cycle-collapsing version instead, given here as Algorithm 3.

Algorithm 3 Cycle-collapsing Average Linkage

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

1. Begin with n singleton clusters and repeat until only one cluster remains:
   (a) For each cluster C, draw a directed edge to the cluster C' maximizing $\mathcal{K}(C, C')$.
   (b) Find a directed cycle in this graph and collapse all clusters in the cycle into a single cluster.
2. Output the tree with single elements as leaves and internal nodes corresponding to all the merges performed.

Theorem 5.3 *Let $\mathcal{K}$ be an asymmetric similarity function satisfying strong stability. Then Algorithm 3 constructs a binary tree such that the ground-truth clustering is a pruning of this tree.*

Proof: Assume by induction that the current clustering is laminar with respect to the target, and consider a cycle produced in Step 1b of the algorithm. If all clusters in the cycle are target clusters or unions of target clusters, then laminarity is clearly maintained. Otherwise, let C' be some cluster in the cycle that is a strict subset of some target cluster C_r. By the strong stability property, it must be the case that the cluster C'' maximizing $\mathcal{K}(C', C'')$ is also a subset of C_r (because at least one must have similarity score at least as high as $\mathcal{K}(C', C_r \setminus C')$. This holds likewise for C'' and throughout the cycle. Thus, all clusters in the cycle are contained within C_r and laminarity is maintained. ■

Theorem 5.4 *Let $\mathcal{K}$ be a symmetric similarity function satisfying the weak stability property. Then the average single linkage algorithm constructs a binary tree such that the ground-truth clustering is a pruning of this tree.*

Proof: 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 $C_{r'}$ and yet C is disjoint from $C_{r'}$.

We distinguish a few cases. First, assume that C is a cluster C_r of the ground-truth. Then by definition, $\mathcal{K}(C', C_{r'} - C') > \mathcal{K}(C', C)$. 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

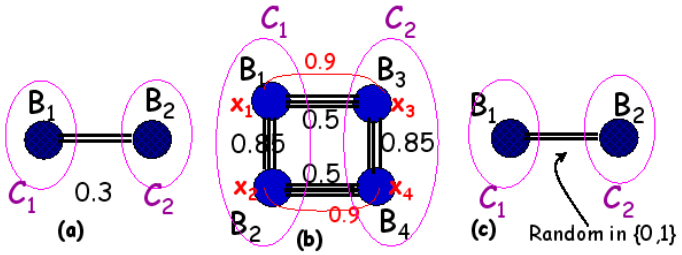

Figure 3: Part (a): Consider two sets B_1, B_2 with m points each. Assume that $\mathcal{K}(x,x') = 0.3$ if $x \in B_1$ and $x' \in B_2$, $\mathcal{K}(x,x')$ is random in $\{0,1\}$ if $x, x' \in B_i$ for all i. Clustering C_1, C_2 does not satisfy strict separation, but for large enough m, w.h.p. will satisfy strong stability. Part (b): Consider four sets B_1, B_2, B_3, B_4 of m points each. Assume $\mathcal{K}(x,x') = 1$ if $x, x' \in B_i$, for all i, $\mathcal{K}(x,x') = 0.85$ if $x \in B_1$ and $x' \in B_2$, $\mathcal{K}(x,x') = 0.85$ if $x \in B_3$ and $x' \in B_4$, $\mathcal{K}(x,x') = 0$ if $x \in B_1$ and $x' \in B_4$, $\mathcal{K}(x,x') = 0$ if $x \in B_2$ and $x' \in B_3$. Now $\mathcal{K}(x,x') = 0.5$ for all points $x \in B_1$ and $x' \in B_3$, except for two special points $x_1 \in B_1$ and $x_3 \in B_3$ for which $\mathcal{K}(x_1,x_3) = 0.9$. Similarly $\mathcal{K}(x,x') = 0.5$ for all points $x \in B_2$ and $x' \in B_4$, except for two special points $x_2 \in B_2$ and $x_4 \in B_4$ for which $\mathcal{K}(x_2,x_4) = 0.9$. For large enough m, clustering C_1, C_2 satisfies strong stability. Part (c): Consider two sets B_1, B_2 of m points each, with similarities within a set all equal to 0.7, and similarities between sets chosen uniformly at random from $\{0,1\}$.

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.

Second, assume that C is strictly contained in one of the ground-truth clusters C_r. Then, by the weak stability property, either $\mathcal{K}(C, C_r - C) > \mathcal{K}(C, C')$ or $\mathcal{K}(C', C_{r'} - C') > \mathcal{K}(C, C')$. This again contradicts the specification of the algorithm as in the previous case.

Finally assume that C is a union of clusters in the ground-truth $C_1, \ldots C_{k'}$. Then by definition, $\mathcal{K}(C', C_{r'} - C') > \mathcal{K}(C', C_i)$, for $i = 1, \ldots k'$, and so $\mathcal{K}(C', C_{r'} - C') > \mathcal{K}(C', C)$. This again leads to a contradiction as argued above. ■

Linkage based algorithms and strong stability: We end this section with a few examples relating strict separation, strong stability, and linkage-based algorithms. Figure 3 (a) gives an example of a similarity function that does not satisfy the strict separation property, but for large enough m, w.h.p. will satisfy the strong stability property. (This is because there are at most m^k subsets A of size k, and each one has failure probability only $e^{-O(mk)}$.) However, single-linkage using $\mathcal{K}_{max}(C, C')$ would still work well here. Figure 3 (b) extends this to an example where single-linkage using $\mathcal{K}_{max}(C, C')$ fails. Figure 3 (c) gives an example where strong stability is not satisfied and average linkage would fail too. However notice that the average attraction property is satisfied and Algorithm 1 will succeed. This example motivates our relaxed definition in Section 5.3 below.

5.3 Stability of large subsets

While natural, the weak and strong stability properties 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 9 *The similarity function $\mathcal{K}$ satisfies the* (s,γ)-**strong stability of large subsets** *property for the clustering problem (S,ℓ) if for all target clusters C_r, $C_{r'}$, $r \neq r'$, 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 $k^{O\left(\frac{k}{\gamma^2}\log\frac{1}{\epsilon}\log\frac{k}{\delta\gamma\epsilon}\right)}$ 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, so that the clusters remaining form an approximate hierarchy and contain good approximations to all clusters in the target. We finally run a procedure that fixes the clusters so they are perfectly laminar and assembles them into a tree. We present and analyze the tester algorithm, Algorithm 4, below.

Algorithm 4 Testing Based Algorithm, Tree Model.

Input: Data set S, similarity function $\mathcal{K}$, parameters $f, g, s, \alpha > 0$. A list of clusters $\mathcal{L}$ with the property that any cluster C in the ground-truth is at least f-close to some cluster in $\mathcal{L}$.

Output: A tree on subsets.

1. Remove all clusters of size at most αn from $\mathcal{L}$. Next, for every pair of clusters C, C' in $\mathcal{L}$ 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 $\mathcal{K}(C \cap C', C' \setminus C)$. Remove C if the first quantity is smaller, else remove 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 (choose a cluster, remove all that are approximately equal to it, and repeat), where 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. Add the cluster containing all of S to $\mathcal{L}''$ if it is not in $\mathcal{L}''$ already, and construct a tree T on $\mathcal{L}''$ ordered by approximate inclusion. Specifically, C becomes a child of C' in tree T if $|C \setminus C'| < gn$, $|C' \setminus C| \geq gn$ and $|C' \cap C| \geq gn$.

4. Feed T to Algorithm 5 which cleans up the clusters in T so that the resulting tree is a legal hierarchy (all clusters are completely laminar and each cluster is the union of its children). Output this tree as the result of the algorithm.

We now analyze Algorithm 4 and its subroutine Algorithm 5, showing that the target clustering is approximated by some pruning of the resulting tree.

Algorithm 5 Tree-fixing subroutine for Algorithm 4.

Input: Tree T on clusters in S, each of size at least αn, ordered by approximate inclusion. Parameters $g, \alpha > 0$.

Output: A tree with the same root as T that forms a legal hierarchy.

1. Let C_R be the root of T. Replace each cluster C in T, with $C \cap C_R$. So, all clusters of T are now contained within C_R.
2. While there exists a child C of C_R such that $|C| \geq |C_R| - \alpha n/2$, remove C from the tree, connecting its children directly to C_R.
3. Greedily make the children of C_R disjoint: choose the largest child C, replace each other child C' with $C' \setminus C$, then repeat with the next smaller child until all are disjoint.
4. Delete any child C of C_R of size less than $\alpha n/2$ along with its descendants. If the children of C_R do not cover all of C_R, and C_R is not a leaf, create a new child with all remaining points in C_R so that the children of C_R now form a legal partition of C_R.
5. Recursively run this procedure on each non-leaf child of C_R.

Theorem 5.5 *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 4 with parameters satisfying $s + f \leq g$, $f \leq g\gamma/10$ and $\alpha > 4\sqrt{g}$ yields a tree such that the ground-truth clustering is $2\alpha k$-close to a pruning of this tree.*

Proof: 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 9 and Lemma 5.6 (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 $|C' \cap C| < gn$.

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

Let $\mathcal{L}''$ be the remaining list of clusters after sparsification. It is immediate from the greedy sparsification procedure 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 tree T given to Algorithm 5, the C''_i are not descendants of one another.

We now analyze Algorithm 5. It is clear by design of the procedure that the tree produced is a legal hierarchy: all points in S are covered and the children of each node in the tree form a partition of the cluster associated with that node. Moreover, except possibly for leaves added in Step 4, all clusters in the tree have size at least $\alpha n/2$, and by Step 2, all are smaller than their parent clusters by at least $\alpha n/2$. Therefore, the total number of nodes in the tree, not including the "filler" clusters of size less than $\alpha n/2$ added in Step 4, is at most $4/\alpha$.

We now must argue that all of the big ground-truth clusters $C_1, \ldots, C_{k'}$ still have close approximations in the tree produced by Algorithm 5. First, let us consider the total amount by which a cluster C_i'' can possibly be trimmed in Steps 1 or 3. Since there are at most $4/\alpha$ non-filler clusters in the tree and all clusters are initially either approximately disjoint or one is an approximate subset of the other, any given C_i'' can be trimmed by at most $(4/\alpha)gn$ points. This in turn is at most $\alpha n/4$ since $\alpha^2 \geq 16g$. Note that since initially C_i'' has size at least αn, this means it will not be deleted in Step 4. However, it could be that cluster C_i'' is deleted in Step 2: in this case, reassign C_i'' to the parent cluster C_R. Thus, the overall distance between C_i'' and C_i can increase due to both trimming and reassigning by at most $3\alpha/4$. Using the fact that initially we had $d(C_i, C_i'') \leq f + 2g \leq \alpha/4$, this means each big ground-truth C_i has some representative in the final tree with error at most αn. Thus the total error on big clusters is at most $\alpha k n$, and adding in the at most k small clusters we have an overall error at most $2\alpha k n$. ■

Lemma 5.6 *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*

$$|C \cap C'| \geq gn \quad \text{and} \quad |C \setminus C'| \geq gn \quad \text{and} \quad |C' \setminus C| \geq gn.$$

Let C^* be a cluster in the underlying ground-truth such that

$$|C^* \setminus C| \leq fn \quad \text{and} \quad |C \setminus C^*| \leq fn.$$

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: Let $I^* = I \cap C^*$. So, $I^* = C \cap C' \cap C^*$. We prove first that

$$\mathcal{K}(I, C \setminus I) > \mathcal{K}(I^*, C^* \setminus I^*) - \gamma/2. \tag{1}$$

Since $\mathcal{K}(x, x') \geq -1$, we have

$$\mathcal{K}(I, C \setminus I) \geq (1 - p_1)\mathcal{K}(I \cap C^*, (C \setminus I) \cap C^*) - p_1,$$

where $1 - p_1 = \frac{|I^*|}{|I|} \cdot \frac{|(C \setminus I) \cap C^*|}{|C \setminus I|}$. By assumption we have both

$$|I| \geq gn \quad \text{and} \quad |I \setminus I^*| \leq fn,$$

which imply:

$$\frac{|I^*|}{|I|} = \frac{|I| - |I \setminus I^*|}{|I|} \geq \frac{g - f}{g}.$$

Similarly, we have both

$$|C \setminus I| \geq gn \quad \text{and} \quad \left|(C \setminus I) \cap \bar{C}^*\right| \leq |C \setminus C^*| \leq fn,$$

which imply:

$$\frac{|(C \setminus I) \cap C^*|}{|C \setminus I|} = \frac{|C \setminus I| - \left|(C \setminus I) \cap \bar{C^*}\right|}{|C \setminus I|} \geq \frac{g-f}{g}.$$

Let us denote by $1-p$ the quantity $\left(\frac{g-f}{g}\right)^2$. We have:

$$\mathcal{K}(I, C \setminus I) \geq (1-p)\mathcal{K}(I^*, (C \setminus I) \cap C^*) - p. \tag{2}$$

Let $A = (C^* \setminus I^*) \cap C$ and $B = (C^* \setminus I^*) \cap \bar{C}$. We have

$$\mathcal{K}(I^*, C^* \setminus I^*) = (1-\alpha)\mathcal{K}(I^*, A) - \alpha\mathcal{K}(I^*, B), \tag{3}$$

where $1-\alpha = \frac{|A|}{|C^* \setminus I^*|}$. Note that $A = (C \setminus I) \cap C^*$ since we have both

$$A = (C^* \setminus I^*) \cap C = (C^* \cap C) \setminus (I^* \cap C) = (C^* \cap C) \setminus I^*$$

and

$$(C \setminus I) \cap C^* = (C \cap C^*) \setminus (I \cap C^*) = (C^* \cap C) \setminus I^*.$$

Furthermore

$$|A| = |(C \setminus I) \cap C^*| \geq |C \setminus C'| - |C \setminus C^*| \geq gn - fn.$$

We also have $|B| = |(C^* \setminus I^*) \cap \bar{C}| \geq |C^* \setminus C| \leq fn$. These imply both

$$1-\alpha = \frac{|A|}{|A|+|B|} = \frac{1}{1+|B|/|A|} \geq \frac{g-f}{g},$$

and

$$\frac{\alpha}{1-\alpha} \leq \frac{f}{g-f}.$$

Inequality (3) implies

$$\mathcal{K}(I^*, A) = \frac{1}{1-\alpha}\mathcal{K}(I^*, C^* \setminus I^*) - \frac{\alpha}{1-\alpha}\mathcal{K}(I^*, B)$$

and since $\mathcal{K}(x, x') \leq 1$, we obtain:

$$\mathcal{K}(I^*, A) \geq \mathcal{K}(I^*, C^* \setminus I^*) - f/(g-f). \tag{4}$$

Overall, combining (2) and (4) we obtain:

$$\mathcal{K}(I, C \setminus I) \geq (1-p)\left[\mathcal{K}(I^*, C^* \setminus I^*) - f/(g-f)\right] - p,$$

so

$$\mathcal{K}(I, C \setminus I) \geq \mathcal{K}(I^*, C^* \setminus I^*) - 2p - (1-p) \cdot f/(g-f).$$

Since $1-p = \left(\frac{g-f}{g}\right)^2$, we have $p = \frac{2gf-f^2}{g^2}$. Using this together with the assumption that $f \leq g\gamma/10$. it is easy to verify that

$$2p + (1-p) \cdot f/(g-f) \leq \gamma/2,$$

which finally implies inequality (1).

Our assumption that $\mathcal{K}$ is a similarity function satisfying the strong stability property with a threshold sn and a γ-gap for our clustering problem (S, ℓ), together with the assumption $s + f \leq g$ implies

$$\mathcal{K}(I^*, C^* \setminus I^*) \geq \mathcal{K}(I^*, C' \setminus (I^* \cup C^*)) + \gamma. \tag{5}$$

We finally prove that

$$\mathcal{K}(I^*, C' \setminus (I^* \cup C^*)) \geq \mathcal{K}(I, C' \setminus I) - \gamma/2. \tag{6}$$

The proof is similar to the proof of statement (1). First note that

$$\mathcal{K}(I, C' \setminus I) \leq (1 - p_2)\mathcal{K}(I^*, (C' \setminus I) \cap \bar{C^*}) + p_2,$$

where

$$1 - p_2 = \frac{|I^*|}{|I|} \cdot \frac{\left|(C' \setminus I) \cap \bar{C^*}\right|}{|C' \setminus I|}.$$

We know from above that $\frac{|I^*|}{|I|} \geq \frac{g-f}{g}$, and we can also show $\frac{\left|(C' \setminus I) \cap \bar{C^*}\right|}{|C' \setminus I|} \geq \frac{g-f}{g}$. So $1 - p_2 \geq \left(\frac{g-f}{g}\right)^2$, and so $p_2 \leq 2\frac{g}{f} \leq \gamma/2$, as desired.

To complete the proof note that relations (1), (5) and (6) together imply the desired result, namely that $\mathcal{K}(I, C \setminus I) > \mathcal{K}(I, C' \setminus I)$. ■

Theorem 5.7 *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 4 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: 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 4 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 5.5). 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 (for a formal proof see lemma 5.8). ■

Lemma 5.8 *Let P_1, ..., P_s be a quasi-partition of S such that $|P_i| \geq n\frac{\nu}{k}$ and $|P_i \cap P_j| \leq gn$ for all $i, j \in \{1, \ldots, s\}$, $i \neq j$. If $g = \frac{\nu^2}{5k^2}$, then $s \leq 2\frac{k}{\nu}$.*

Proof: Assume for contradiction that $s > L = 2\frac{k}{\nu}$, and consider the first L parts P_1, ..., P_L. Then $\left(n\frac{\nu}{k} - 2\frac{k}{\nu}gn\right)2\frac{k}{\nu}$ is a lower bound on the number of points that belong to exactly one of the parts P_i, $i \in \{1, \ldots, L\}$. For our choice of g, $g = \frac{\nu^2}{5k^2}$, we have

$$\left(n\frac{\nu}{k} - 2\frac{k}{\nu}gn\right)2\frac{k}{\nu} = 2n - \frac{4}{5}n.$$

So $\frac{6}{5}n$ is a lower bound on the number of points that belong to exactly one of the parts P_i, $i \in \{1, \ldots, L\}$, which is impossible since $|S| = n$. So, we must have $s \leq 2\frac{k}{\nu}$. ■

We discuss other interesting stability and average attraction-style properties in Appendix B.

6 Approximation Assumptions

When developing a c-approximation algorithm for some clustering objective function Φ, if the goal is to actually cluster the points correctly, then one is implicitly making the assumption (or hope) that any c-approximation to Φ 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 10 *Given objective function Φ, we say that a metric d over point set S satisfies the (c, ϵ)-Φ property with respect to target $\mathcal{C}$ if all clusterings $\mathcal{C}'$ that are within a factor c of optimal in terms of objective Φ are ϵ-close to $\mathcal{C}$.*

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

Theorem 6.1 *If metric d satisfies the (2, ϵ)-k-median property for dataset S, then the similarity function $-d$ satisfies the ν-strict separation property for $\nu = 4\epsilon$.*

Proof: As before we denote the target clustering by $\mathcal{C} = \{C_1, C_2, ..., C_k\}$. Let OPT = $\{\text{OPT}_1, \text{OPT}_2, ..., \text{OPT}_k\}$ be the k-median optimal clustering, where

$$\sum_i |C_i \cap OPT_i| \geq (1 - \epsilon)n.$$

Mark the set of all points (at most ϵn) where C and OPT disagree.

For $j = 1, 2, \ldots$, if there exists an unmarked x_j that is more similar to some unmarked z_j in a different target cluster than to some unmarked y_j in its own cluster, we mark all three points. If this process halts after at most ϵn rounds, then we are done: the unmarked set, which has at least $(1 - 4\epsilon)n$ points, satisfies strict separation. We now claim we can get a contradiction if the process lasts longer. Specifically, begin with OPT (not $\mathcal{C}$) and move each x_j to the cluster containing point z_j. Call the result OPT$'$. Note that for all j, the pair (x_j, y_j) are in the *same* cluster in C (because we only chose x_j, y_j, z_j from points where $\mathcal{C}$ and OPT agree) but are in *different* clusters in OPT$'$; moreover, the pairs are all disjoint. So, $d(\text{OPT}', \mathcal{C}) > \epsilon n$. However, we claim that OPT$'$ has cost at most 2OPT. To see this, first define cost(x) to be the contribution of x to the k-median cost in OPT, and define cost$'(x)$ to be x's contribution in OPT$'$. Then, moving x_j into the cluster of the corresponding z_j will increase the k-median objective by at most

$$\begin{aligned} \text{cost}'(x_j) &\leq d(x_j, z_j) + \text{cost}(z_j) \leq d(x_j, y_j) + \text{cost}(z_j) \\ &\leq \text{cost}(x_j) + \text{cost}(y_j) + \text{cost}(z_j). \end{aligned}$$

Thus, since the triples (x_j, y_j, z_j) are all disjoint, the k-median objective at most doubles, contradicting our initial assumption. ■

We can similarly prove:

Theorem 6.2 *If the metric d satisfies the (3, ϵ)-k-center property, then the similarity function $(-d)$ satisfies the ν-strict separation property for $\nu = 4\epsilon$.*

So if the metric d satisfies the $(2, \epsilon)$-k-median or the $(3, \epsilon)$-k-center property for dataset S, then the similarity function $-d$ satisfies the ν-strict separation property for $\nu = 4\epsilon$. Theorem 3.3 (in Section 6.1) then implies that as long as the smallest cluster in the target has size $20\epsilon n$ we can produce a tree such that the ground-truth clustering is 4ϵ-close to a pruning of this tree.

Note: 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. This has been further exploited in subsequent work. In particular, Balcan et al. [2009] show that if we assume that any c-approximation to the k-median objective is ϵ-close to the target—then one can produce clusterings that are $O(\epsilon)$-close to the target, *even for values c for which obtaining a c-approximation is NP-hard*. We discuss this further in Section 8.

6.1 The ν-strict separation Property

We end this section by proving Theorem 3.3.

Theorem 3.3 *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 will in fact create a tree that is perfect with respect to S'; that is, if points in $S \setminus S'$ are removed, the target clustering will be exactly equal to some pruning of the tree.

We begin by generating a list $\mathcal{L}$ of n^2 clusters such that, ignoring bad points, all clusters in the ground-truth are 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 $4\nu n \leq t \leq n$. We next run a procedure that fixes or removes clusters that are non-laminar with respect to each other without hurting any of the correct clusters, until the remaining set is fully laminar. We then hook up the final set of clusters into a tree.

Specifically, after creating the set $\mathcal{L}$, while there exist two clusters C and C' that are non-laminar with respect to each other, we do the following (always choosing the first option that applies):

1. If either C or C' has size $\leq 4\nu n$, 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 its points as good points (since there are at most νn bad points total), 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. Thus, C and C' are now fully disjoint and we maintain our invariant that, with respect to S', all ground-truth clusters are in our list.

3. If C, C' are "somewhat equal" in that $|C \setminus C'| \leq 2\nu n$ and $|C' \setminus C| \leq 2\nu n$, 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. Thus, we again maintain our invariant.

4. Finally, 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 $0 < |C' \setminus C| \leq 2\nu n$. 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.

We now need to argue that the above procedure halts in a polynomial number of steps. Cases (1) and (3) each delete a cluster, and no clusters are ever added, so together they can occur at most n^2 times. Case (2) reduces the overall sum of cluster sizes, so it can occur only a polynomial number of times before an instance of case (4).

Now, case (4) could cause one of its clusters (namely, C) to grow, so we need to be a bit more careful with it. Specifically, we argue case (4) as follows. Let $\bar{C} = C \cup \{C'' : C \cap C'' \neq \phi\}$. Note that $\bar{C}$ is completely disjoint from any cluster not contained inside it, because we only apply case (4) when none of the other cases apply. Therefore, after case (4) is applied to C and all C' that it "somewhat contains", we have that all clusters in $\mathcal{L}$ are either subsets of C or disjoint from C, and this will remain true throughout the remainder of the procedure. So, the number of active clusters (those for which any of cases (1)-(4) might apply to) has decreased by 1. This can occur only a polynomial number of times.

Thus, the overall total number of steps is polynomial in n. Finally, since all clusters remaining are laminar, we can now arrange them into a forest, which we then arbitrarily complete into a tree. ■

7 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 $h : X \to Y$. 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 3) and the generalized large margin property are already inherently inductive. We simply

extend the domain of the hypotheses h produced from S to all of X.

The strict separation property. We can adapt the algorithm in Theorem 3.2 to the inductive setting as follows. We first draw a set S of $n = O\left(\frac{k}{\epsilon}\ln\left(\frac{k}{\delta}\right)\right)$ unlabeled examples. We run the algorithm described in Theorem 3.2 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 to 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 $\text{argmax}_{q\in Q}\mathcal{K}(x,q)$; 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 3.2 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 9 to the inductive setting. The main difference in the inductive setting is that we have to *estimate* (rather than *compute*) the quantities $|C_r \setminus C_{r'}|$, $|C_{r'} \setminus C_r|$, $|C_r \cap C_{r'}|$, $\mathcal{K}(C_r \cap C_{r'}, C_r \setminus C_{r'})$ 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 $\text{poly}(k, 1/\epsilon, 1/\gamma, 1/\delta)\log(|\mathcal{L}|))$ additional points, where $\mathcal{L}$ is the input list in Algorithm 4 (whose size depends on $1/\epsilon$, $1/\gamma$ and k only). Specifically, using a straightforward modification of the proof in Theorem 5.7 and standard concentration inequalities (e.g. the McDiarmid inequality [Devroye *et al.*, 1996]) we have:

Theorem 7.1 *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 4 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 $O\left(\frac{k}{\gamma^2}\ln\left(\frac{k}{\epsilon\delta}\right)\cdot\left(\frac{k}{\epsilon}\right)^{\frac{4k}{\gamma^2}\ln\left(\frac{k}{\epsilon\delta}\right)}\ln(\frac{1}{\delta})\right)$ points in the first phase and $O\left(\frac{1}{\gamma^2}\frac{1}{g^2}\frac{k}{\gamma^2}\log\frac{1}{\epsilon}\log\frac{k}{\delta f}\log k\right)$ 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. In order to use the strong stability property in the inductive setting, 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 pairwise similarities $\mathcal{K}(x,x')$ 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 (we assume here that our similarity function is symmetric):

Property 11 *The similarity function $\mathcal{K}$ satisfies the* **γ-strong stability** *property for the clustering problem (X,ℓ) if for all target clusters C_r, $C_{r'}$, $r \neq r'$, 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 7.1, 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 6. While the algorithm is simple, proving its correctness requires substantially more involved arguments. In particular, we must show that if Property 11 holds for the entire space X, then with high probability (a version of) it holds for the sample as well; i.e., that sampling preserves stability. This will require adapting regularity-style arguments of [Frieze and Kannan, 1999] and [Alon *et al.*, 2003].

Algorithm 6 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 $\text{argmax}_{q \in Q} \mathcal{K}(x, q)$; 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 $n = \text{poly}(k, 1/\epsilon, 1/\gamma, 1/\delta)$ we obtain a tree T which has a pruning $f_1, ..., f_{k'}$ of error at most ϵ. Specifically:

Theorem 7.2 *Let $\mathcal{K}$ be a similarity function satisfying the strong stability property for the clustering problem (X, ℓ). Then using Algorithm 6 with parameters $n = \text{poly}(k, 1/\epsilon, 1/\gamma, 1/\delta)$, we can produce a tree with the property that the ground-truth is ϵ-close to a pruning of this tree.*

Proof: 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 7.3 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 $x \in S \setminus C'_i$.) Therefore, by the argument in Theorem 5.2, these large C'_i will appear as nodes of the tree T, even if the C'_i corresponding to the small ground-truth clusters do not satisfy the property. Thus, with high probability, $\mathcal{C}_S$ is approximately 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 6.

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

Theorem 7.3 *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: 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 [Frieze and Kannan, 1999] and the random sampling analysis of [Alon *et al.*, 2003].

Let $N = |C_i|$. By [Frieze and Kannan, 1999], 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, the values d_j are not too large, specifically each $d_j = O(1)$. Finally, $s = 1/\tau^2$ cut-matrices are sufficient so that matrix W has cut-norm at most τN^2: that is, for any partition of the vertices A, A', we have $|\sum_{x \in A, y \in A'} W[xy]| \leq \tau N^2$; moreover, $||W||_\infty \leq 1/\tau$ and $||W||_F \leq N$.

We now closely follow arguments in [Alon *et al.*, 2003]. 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 sub-programs of LPs) of [Alon *et al.*, 2003]. 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 [Alon *et al.*, 2003].

Lemma 7.4 *[Random submatrix][Alon* et al.*, 2003] For $\tau, \delta > 0$, and any W an $N \times N$ real matrix with cut-norm $||W||_C \leq \tau N^2$, $||W||_\infty \leq 1/\tau$ 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 $n' > (c_1/\tau^4\delta^5)\log(2/\tau)$, with probability at least $1 - \delta$,*

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

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 $\tau = \tilde{o}(\gamma^2)$, 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. ■

8 Subsequent work

Following the initial publication of this work, Balcan, Blum, and Gupta [2009] have further analyzed implications of the (c, ϵ) property for k-median, k-means, and min-sum objectives. In particular, one of the main results of [Balcan *et al.*, 2009] for the k-median problem is the following:

Theorem 8.1 *For any $\alpha > 0$, if the metric d satisfies the $(1 + \alpha, \epsilon)$-k-median property for dataset S, then one can efficiently find a (single) clustering which is $O(\epsilon/\alpha)$-close to the target. Moreover, if each cluster in the target clustering has size at least $(4 + 15/\alpha)\epsilon n + 2$, then one can efficiently find a clustering that is ϵ-close to the target.*

These results also highlight a surprising conceptual difference between assuming that the *optimal* solution to the k-median objective is ϵ-close to the target, and assuming that any *approximately optimal* solution is ϵ-close to the target, even for approximation factor say $c = 1.01$. In the former case, the problem of finding a solution that is $O(\epsilon)$-close to the target remains computationally hard, and yet for the latter there is an efficient algorithm.

Balcan et al. [2009] prove similar results for the k-means objective, and Balcan and Braverman [2009] derive similar results for the min-sum objective. In addition, Balcan and Braverman [2009] also consider the correlation clustering objective and show that for this objective, the $(1 + \alpha, \epsilon)$ property implies a $(2.5, O(\epsilon/\alpha))$ property, so one can use a state-of-the-art 2.5-approximation algorithm for minimizing disagreements [Ailon *et al.*, 2005] in order to get an accurate clustering. This contrasts with objectives such as min-sum, k-median, or k-means, where data may satisfy the (c, ϵ) property but not even the $(c', 0.49)$ property for any $c' > c$.

9 Conclusions and Discussion

In this paper we provide a general framework for analyzing what properties of a similarity function are sufficient to allow one to cluster accurately. Our framework does not rely on probabilistic generative-model assumptions, and instead parallels the notion of *data-dependent concept classes* [Vapnik, 1998] (such as large-margin separators) in the context of classification. We prove that in our framework, a number of interesting, natural properties are sufficient to cluster well under two reasonable relaxations of the clustering objective, tree clustering and list clustering. To do so, we analyze a wide variety of different types of algorithms. For some 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 properties we develop new algorithmic techniques that are able to take advantage of them, and that may prove to be more broadly useful. We also show that for certain algorithms such as single-linkage, we can describe properties that completely characterize the conditions needed for their success. We in addition define 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, providing tight upper and lower bounds.

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 existing algorithms (e.g., hierarchical linkage-based algorithms) will fare better than others.

Our framework also provides a natural way to formalize *exploratory clustering*, by allowing the property itself to be viewed as the criterion for a clustering to be "interesting". In this view, all of our formal guarantees can be interpreted as saying that the hierarchy or the list that our algorithm outputs contains (approximations to) all the desirable clusterings, and the clustering complexity of a property gives an upper-bound on the number of "substantially different" interesting clusterings.

9.1 Open questions

In terms of specific open questions, for the average attraction property (Property 3) 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 9), our algorithm produces hierarchy but has substantially larger running time than that for the simpler stability properties. Can an algorithm with running time polynomial in k and $1/\gamma$ be developed? Can one prove guarantees for stability properties based on spectral methods, e.g., the hierarchical clustering algorithm given in [Cheng *et al.*, 2006]? It would also be interesting to determine whether these stability properties can be further weakened and still admit a hierarchical clustering.

More broadly, one would like to analyze other natural properties of similarity functions, as well as to broaden the types of *output structures* produced, for applications in which the goal is not just a partition of data but more broadly a tree, DAG, or other organizational structure. Such structures arise in a number of applications including web-based knowledge acquisition and bioinformatics applications,

Finally, in this work we have focused on formalizing clustering with non-interactive feedback; that is, the tree or list produced would then be given to a user or subsequent post-processing step.

It would be interesting to formalize clustering with other more interactive forms of feedback. For example, depending on the application, different types of feedback can be most natural, such as identifying points that should or should not be in the same cluster, or clusters that should be split or merged. Some initial progress in this direction has been made in [Balcan and Blum, 2009] and [Awasthi, 2009].

References

[Achlioptas and McSherry, 2005] D. Achlioptas and F. McSherry. On spectral learning of mixtures of distributions. In *18th Annual Conference on Learning Theory*, 2005.

[Ackerman and Ben-David., 2008] M. Ackerman and S. Ben-David. Which data sets are clusterable? - a theoretical study of clusterability. In *NIPS 2008*, 2008.

[Ailon *et al.*, 2005] N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: ranking and clustering. In *Proceedings of the 37th ACM Symposium on Theory of Computing*, pages 684–693, 2005.

[Alimonti and Kann, 1997] P. Alimonti and V. Kann. Hardness of approximating problems on cubic graphs. In *Algorithms and Complexity*, 1997.

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

[Alon *et al.*, 2000] N. Alon, S. Dar, M. Parnas, and D. Ron. Testing of clustering. In *Proc. 41st Annual Symposium on Foundations of Computer Science*, 2000.

[Alon *et al.*, 2003] 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.

[Arora and Kannan, 2001] S. Arora and R. Kannan. Learning mixtures of arbitrary gaussians. In *Proceedings of the 37th ACM Symposium on Theory of Computing*, 2001.

[Arya *et al.*, 2004] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit. Local search heuristics for k-median and facility location problems. *SIAM J. Comput.*, 33(3):544–562, 2004.

[Awasthi, 2009] P. Awasthi. Interactive clustering. Manuscript, 2009.

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

[Balcan and Blum, 2009] M.-F. Balcan and A. Blum. Clustering with interactive feedback. In *Proceedings of the The 19th International Conference on Algorithmic Learning Theory*, 2009.

[Balcan and Braverman, 2009] M.-F. Balcan and M. Braverman. Finding low error clusterings. In *Proceedings of the 22nd Annual Conference on Learning Theory*, 2009.

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

[Balcan *et al.*, 2008] M.-F. Balcan, A. Blum, and N. Srebro. On a theory of learning with similarity functions. *Machine Learning Journal*, 2008.

[Balcan *et al.*, 2009] M.-F. Balcan, A. Blum, and A. Gupta. Approximate clustering without the approximation. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, 2009.

[Bandelt and Dress, 1989] H.-J. Bandelt and A.W.M. Dress. Weak hierarchies associated with similarity measures: an additive clustering technique. *Bulletin of mathematical biology*, 51(1):133–166, 1989.

[Bartal *et al.*, 2001] Y. Bartal, M. Charikar, and D. Raz. Approximating min-sum k-clustering in metric spaces. In *Proceedings on 33rd Annual ACM Symposium on Theory of Computing*, 2001.

[Ben-David *et al.*, 2006] S. Ben-David, U. von Luxburg, and D. Pal. A sober look at stability of clustering. In *Proceedings of the Annual Conference on Computational Learning Theory*, 2006.

[Ben-David *et al.*, 2007] S. Ben-David, D. Pal, and H.U. Simon. Stability of k-means clustering. In *Proceedings of the Twentieth Annual Conference on Learning Theory*, 2007.

[Ben-David, 2007] 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.

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

[Bryant and Berry, 2001] D. Bryant and V. Berry. A structured family of clustering and tree construction methods. *Advances in Applied Mathematics*, 27(4):705 – 732, 2001.

[Charikar *et al.*, 1999] M. Charikar, S. Guha, E. Tardos, and D. B. Shmoy. A constant-factor approximation algorithm for the k-median problem. In *In Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, 1999.

[Chaudhuri *et al.*, 2007] S. Chaudhuri, A. Das Sarma, V. Ganti, and R. Kaushik. Leveraging aggregate constraints for deduplication. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, June 2007.

[Cheng *et al.*, 2006] 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.

[Czumaj and Sohler, 2004] A. Czumaj and C. Sohler. Sublinear-time approximation for clustering via random samples. In *Proceedings of the 31st International Colloquium on Automata, Language and Programming*, pages 396–407, 2004.

[Dasgupta *et al.*, 2005] 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.

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

[Dasgupta, 1999] S. Dasgupta. Learning mixtures of gaussians. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, 1999.

[de la Vega *et al.*, 2003] W. Fernandez de la Vega, Marek Karpinski, Claire Kenyon, and Yuval Rabani. Approximation schemes for clustering problems. In *In Proceedings for the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, 2003.

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

[Duda *et al.*, 2001] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley, 2001.

[Elias, 1957] P. Elias. List decoding for noisy channels. Technical Report Technical Report 335, Research Laboratory of Electronics, MIT, 1957.

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

[Guruswami and Sudan, 1999] V. Guruswami and M. Sudan. Improved decoding of reed-solomon and algebraic-geometric codes. *IEEE Transactions on Information Theory*, 45:1757–1767, 1999.

[Heller, 2008] K. Heller. *Efficient Bayesian Methods for Clustering*. PhD thesis, 2008.

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

[Jain and Vazirani, 2001] 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.

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

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

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

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

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

[McSherry, 2001] F. McSherry. Spectral parititioning of random graphs. In *Proceedings of the 43rd Symposium on Foundations of Computer Science*, pages 529–537, 2001.

[Meila, 2003] M. Meila. Comparing clusterings by the variation of information. In *Proceedings of the Annual Conference on Computational Learning Theory*, 2003.

[Meila, 2005] M. Meila. Comparing clusterings – an axiomatic view. In *International Conference on Machine Learning*, 2005.

[Mishra *et al.*, 2001] Nina Mishra, Dan Oblinger, and Leonard Pitt. Sublinear time approximate clustering. In *Proceedings of Syposium on Discrete Algorithms*, pages 439–447, 2001.

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

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

[Shawe-Taylor *et al.*, 1998] 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.

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

[Teh *et al.*, 2006] Y. Teh, M. Jordan, M. Beal, and D. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101:1566–1581, 2006.

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

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

[Vempala and Wang, 2004] S. Vempala and G. Wang. A spectral algorithm for learning mixture models. *Journal of Computer and System Sciences*, 68(2):841–860, 2004.

A Relation to work on defining stable clusters

We review here the definition of stability in [Bryant and Berry, 2001] and how this relates to our notion of strong stability. We start with a few definitions from [Bryant and Berry, 2001].

First, [Bryant and Berry, 2001] focus on symmetric similarity functions. A rooted triple $ab \mid c$ denotes a grouping of a and b relative to c. The set of rooted triples of X is denoted by $\mathcal{R}(\mathcal{X})$. An isolation index is a weighting function $w : \mathcal{R}(\mathcal{X}) \to \mathcal{R}$ such that $w(ab|c) + w(ac|b) \leq 0$ for all $a, b, c \in X$. Given a similarity function $\mathcal{K}$ define:

$$\rho_{\mathcal{K}}(ab \mid c) = \mathcal{K}(a, b) - \max\{\mathcal{K}(a, c), \mathcal{K}(b, c)\}.$$

Then $\rho_{\mathcal{K}}$ is an isolation index. Given a non-empty set R of rooted triples and an isolation index w, let $\text{av}(R)$ denote the average weight of the triples in R and let $\text{avmin}(R, k)$ denote the average weight of the $k \leq |R|$ triples with minimum weight in R. Also, given disjoint non-empty sets U, V, Z define $\bar{w}(UV|Z)$ as the average weight over all triples $uv \mid z$ with $u \in U$, $v \in V$, and $z \in Z$. Denote by $U \uplus V$ the union of disjoint sets U and V.

Definition 2 *[Clustering indices] Let w be an isolation weighting and let A be a cluster of X.*

(a) The strong isolation index of A is defined

$$\iota_w(A) = \min_{uv|z} \{w(uv \mid z) : uv \mid z \in r(A)\},$$

and the strong clusters of w are $\{A : \iota_w(A) > 0\}$.

(b) The clean index of A is defined

$$\iota^c_w(A) = \text{avmin}(r(A), |A| - 1),$$

and the clean clusters of w are $\{A : \iota^c_w(A)\}$.

(c) The stability index of A is defined

$$\iota^s_w(A) = \min_{U,V,Z\neq} \{\bar{w}(UV \mid Z) : A = U \uplus V, Z \subseteq X \setminus A\},$$

and the stable clusters of w are $\{A : \iota^s_w(A) > 0\}$.

Let $w = \rho_{\mathcal{K}}$. [Bryant and Berry, 2001] show the following:

Theorem A.1 *Every stable cluster of w is a cluster in the average linkage tree of $\mathcal{K}$.*

We show in the following that our definition of strong stability is strictly more general than the definition 2 (c) in the case where the isolation index $w = \rho_{\mathcal{K}}$.

Theorem A.2 *Let $\mathcal{K}$ be a symmetric similarity function. If a clustering $\mathcal{C}$ does not satisfy the of strong stability property (Property 7) with respect to $\mathcal{K}$, then one of the target clusters does not satisfy the definition 2 (c) for the isolation index $w = \rho_{\mathcal{K}}$.*

Proof: Assume that a cluster of $\mathcal{K}$ is not stable in the sense of Property 7. That means, there exist target clusters C, C' and $A \subset C$ and $A' \subseteq C'$ such that $\mathcal{K}(A, C \setminus A) \leq \mathcal{K}(A, A')$. Let z be the point in A' such that $\mathcal{K}(A, z)$ is maximized. We have $\mathcal{K}(A, C \setminus A) \leq \mathcal{K}(A, z)$ or equivalently $\mathcal{K}(A, C \setminus A) - \mathcal{K}(A, z) \leq 0$.

Now we expand out the left-hand-side (writing the $\mathcal{K}(A, z)$ term in a different way) to get:

$$\frac{1}{|A| \cdot |C \setminus A|} \sum_{u\in A, v\in C\setminus A} \mathcal{K}(u, v) - \frac{1}{|A| \cdot |C \setminus A|} \sum_{u\in A, v\in C\setminus A} \mathcal{K}(u, z) \leq 0.$$

This implies $\frac{1}{|A|\cdot|C\setminus A|} \sum_{u\in A, v\in C\setminus A} [\mathcal{K}(u, v) - \mathcal{K}(u, z)] \leq 0$, so,

$$\frac{1}{|A| \cdot |C \setminus A|} \sum_{u\in A, v\in C\setminus A} [\mathcal{K}(u, v) - \max(\mathcal{K}(u, z), \mathcal{K}(v, z)] \leq 0.$$

This implies that Definition 2 (c) is not satisfied either, as desired. ■

On the other hand, it is possible to satisfy strong stability but not definition 2(c).

Theorem A.3 *There exists a pair clustering, similarity function $(\mathcal{C}, \mathcal{K})$ such that $\mathcal{C}$ satisfies the strong stability property (Property 7) with respect to $\mathcal{K}$, but such that one of the target clusters does not satisfy the definition 2 (c), for the isolation index $w = \rho_{\mathcal{K}}$.*

Proof: Assume that cluster target cluster A has 4 points: u_1, u_2, v_1, v_2. Assume that our similarity function satisfies $\mathcal{K}(u_1, u_2) = \mathcal{K}(v_1, v_2) = \mathcal{K}(u_2, v_2) = 1$, $\mathcal{K}(u_1, v_1) = 0$, and $\mathcal{K}(u_1, v_2) = \mathcal{K}(u_2, v_1) = 0.8$. Assume that the target cluster Z has one point z. Assume that our similarity function satisfies: $\mathcal{K}(u_1, z) = \mathcal{K}(v_1, z) = 0$, and $\mathcal{K}(u_2, z) = \mathcal{K}(v_2, z) = 0.89$.

We first note that $\mathcal{C}$ satisfies the strong stability property with respect to $\mathcal{K}$, We show this by brute-force checking over all subsets of A of size 1, 2, 3. For example $\mathcal{K}(u_2, A - u_2) = 0.933 \geq \mathcal{K}(u_2, Z) = 0.89$, $\mathcal{K}(v_2, A - v_2) = 0.933 \geq \mathcal{K}(v_2, Z) = 0.89$, $\mathcal{K}(u_2, v_2, A - u_2, v_2) = 0.9 \geq \mathcal{K}(u_2, v_2, Z) = 0.89$.

We finally show that one of the target clusters does not satisfy definition 2 (c). Consider $U = \{u_1, u_2\}$, $V = \{v_1, v_2\}$. Then the quantity $\iota_w^s(A)$ is the average of the following quantities:

$$\begin{aligned} w(u_1v_1 \mid z) &= 0 - 0 = 0, \\ w(u_1v_2 \mid z) &= 0.8 - 0.89 = -0.09, \\ w(u_2v_1 \mid z) &= 0.8 - 0.89 = -0.09, \text{ and} \\ w(u_2v_2 \mid z) &= 1.0 - 0.89 = +0.11, \end{aligned}$$

which is negative. This completes the proof. ■

B Other Aspects

B.1 Computational Hardness Results

Our framework also allows us to study computational hardness results as well. We discuss here a simple example.

Property 12 *A similarity function $\mathcal{K}$ satisfies the* **unique best cut** *property for the clustering problem (S, ℓ) if $k = 2$ and $\sum_{x \in C_1, x' \in C_2} \mathcal{K}(x, x') < \sum_{x \in A, x' \in B} \mathcal{K}(x, x')$ for all partitions $(A, B) \neq (C_1, C_2)$ of S.*

Clearly, by design the clustering complexity of Property 12 is 1. However, producing even a polynomial-length list of clusterings is NP-hard.

Theorem B.1 *List-clustering under the unique best cut property is NP-hard. That is, there exists $\epsilon > 0$ such that given a dataset S and a similarity function $\mathcal{K}$ satisfying the unique best cut property, it is NP-hard to produce a polynomial-length list of clusterings such that at least one is ϵ-close to the ground truth.*

Proof: It is known that the MAX-CUT problem on cubic graphs is APX-hard [Alimonti and Kann, 1997] (i.e. it is hard to approximate within a constant factor $\alpha < 1$).

We create a family $((S, \ell), \mathcal{K})$ of instances for our clustering property as follows. Let $G = (V, E)$ be an instance of the MAX-CUT problem on cubic graphs, $|V| = n$. For each vertex $i \in V$ in the graph we associate a point $x_i \in S$; for each edge $(i, j) \in E$ we define $\mathcal{K}(x_i, x_j) = -1$, and we define

$\mathcal{K}(x_i, x_j) = 0$ for each $(i,j) \notin E$. Let $S_{V'}$ denote the set $\{x_i : i \in V'\}$. Clearly for any given cut (V_1, V_2) in $G = (V, E)$, the value of the cut is exactly

$$F(S_{V_1}, S_{V_2}) = \sum_{x \in S_{V_1}, x' \in S_{V_2}} -\mathcal{K}(x, x').$$

Let us now add tiny perturbations to the $\mathcal{K}$ values so that there is a unique partition $(C_1, C_2) = (S_{V_1^*}, S_{V_2^*})$ minimizing the objective function Φ, and this partition corresponds to some maxcut (V_1^*, V_2^*) of G (e.g., we can do this so that this partition corresponds to the lexicographically first such cut). By design, $\mathcal{K}$ now satisfies the unique best cut property for the clustering problem S with target clustering (C_1, C_2).

Define ϵ such that any clustering which is ϵ-close to the correct clustering (C_1, C_2) must be at least α-close in terms of the max-cut objective. E.g., $\epsilon < \frac{1-\alpha}{4}$ suffices because the graph G is cubic. Now, suppose a polynomial time algorithm produced a polynomial-sized list of clusterings with the guarantee that at least one clustering in the list has error at most ϵ in terms of its accuracy with respect to (C_1, C_2). In this case, we could then just evaluate the cut value for all the clusterings in the list and pick the best one. Since at least one clustering is at least ϵ-close to (C_1, C_2) by assumption, we are guaranteed that at least one is within α of the optimum cut value. ■

Note that we can get a similar results for any clustering objective Φ that (a) is NP-hard to approximate within a constant factor, and (b) has the smoothness property that it gives approximately the same value to any two clusterings that are almost the same.

B.2 Other interesting properties

An interesting relaxation of the average attraction property is to ask only that there *exists* a target cluster so that most of the points in that cluster are noticeably more similar on average to other points in that cluster than to points in all the other clusters, and that once we remove that cluster the property becomes true recursively.[3] Formally:

Property 13 *A similarity function $\mathcal{K}$ satisfies the* **γ-weak average attraction** *property for the clustering problem (S, ℓ) if there exists cluster C_r such that all examples $x \in C_r$ satisfy:*

$$\mathcal{K}(x, C(x)) \geq \mathcal{K}(x, S \setminus C_r) + \gamma,$$

and moreover the same holds recursively on the set $S \setminus C_r$.

We can then adapt Algorithm 1 to get the following result:

Theorem B.2 *Let $\mathcal{K}$ be a similarity function satisfying γ-weak average attraction for the clustering problem (S, ℓ). Using Algorithm 1 with $s = \frac{4}{\gamma^2} \ln\left(\frac{8k}{\epsilon\delta}\right)$ and $N = \left(\frac{2k}{\epsilon}\right)^{\frac{4k}{\gamma^2} \ln\left(\frac{8k}{\epsilon\delta}\right)} \ln(\frac{1}{\delta})$ 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 ϵ-close to the ground-truth.*

[3] Thanks to Sanjoy Dasgupta for pointing out that this property is satisfied on real datasets, such as the MINST dataset.

Another interesting property that falls in between the weak stability property and the average attraction property is the following:

Property 14 *The similarity function $\mathcal{K}$ satisfies the* γ**-strong attraction** *property for the clustering problem (S, ℓ) if for all clusters C_r, $C_{r'}$, $r \neq r'$ in the ground-truth, for all $A \subset C_r$ we have*

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

We can interpret the strong attraction property as saying that for any two clusters C_r and $C_{r'}$ in the ground truth, for any subset $A \subset C_r$, the subset A is more attracted to the rest of its own cluster than to $C_{r'}$. It is easy to see that we cannot cluster in the tree model, and moreover we can show an lower bound on the sample complexity which is exponential. Specifically:

Theorem B.3 *For $\epsilon \leq \gamma/4$, the γ-strong attraction property has $(\epsilon, 2)$ clustering complexity as large as $2^{\Omega(1/\gamma)}$.*

Proof: Consider $N = \frac{1}{\gamma}$ sets of equal probability mass. Consider a special matching of these sets $\{(R_1, L_1), (R_2, L_2), \ldots, (R_{N/2}, L_{N/2})\}$ and define $\mathcal{K}(x, x') = 0$ if $x \in R_i$ and $x' \in L_i$ for some i and $\mathcal{K}(x, x') = 1$ otherwise. Then each partition of these sets into *two* pieces of equal size that fully "respects" our matching (in the sense that for all i R_i, L_i are on two different parts) satisfies Property 14 with a gap $\gamma' = 2\gamma$. The desired result then follows from the fact that the number of such partitions (which split the set of sets into two pieces of equal size and fully respect our matching) is $2^{\frac{1}{2\gamma}-1}$. ■

It would be interesting to see if one could develop algorithms especially designed for this property that provide better guarantees than Algorithm 1.

Another interesting property to analyze would be the following:

Property 15 *The similarity function $\mathcal{K}$ satisfies the* γ**-stable split** *property for the clustering problem (S, ℓ) if for all clusters C_r, $C_{r'}$, $r \neq r'$ in the ground-truth, for all $A \subset C_r$, $A' \subset C_{r'}$ we have*

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

It would be interesting to see if one could develop algorithms especially designed for this property that provides better guarantees than Algorithm 1.

B.3 Verification

A natural question is how hard is it (computationally) to determine if a proposed clustering of a given dataset S satisfies a given property or not. For example, even for $k = 2$, determining if a clustering satisfies strong stability is NP-hard (reduction from sparsest cut, see, e.g., [Bryant and Berry, 2001]). On the other hand, recall that our goal is not to produce a clustering with a given property but rather one that is accurate; the reason for the property is just to provide sufficient conditions on achieving this goal. In addition, one can also efficiently compute distances between any two given clusterings (via a weighted matching algorithm) if one has pre-clustered data for testing purposes. Note that computing the distance between the target clustering and any other clustering is the analogue of computing the empirical error rate of a given hypothesis in the PAC

setting [Valiant, 1984]; furthermore, there are many learning problems in the PAC model where the consistency problem is NP-hard (e.g. 3-Term DNF), even though the corresponding classes are learnable.