Sublinear Time Approximate Clustering

Nina Mishra [*] Dan Oblinger [†] Leonard Pitt [‡]

Abstract

Clustering is of central importance in a number of disciplines including Machine Learning, Statistics, and Data Mining. This paper has two foci: (1) It describes how existing algorithms for clustering can benefit from simple sampling techniques arising from work in statistics [Pol84]. (2) It motivates and introduces a new model of clustering that is in the spirit of the "PAC (probably approximately correct)" learning model, and gives examples of efficient PAC-clustering algorithms.

1 Introduction

The problem of clustering data into subsets that are "similar" has recently become an almost necessary tool for data mining applications that deal with very large datasets such as web pages, click streams, multimedia data, business transactions, or telecommunications phone records.

In this paper we give general techniques for transforming existing approximate clustering algorithms into ones which access much less of the input dataset, yet yield comparable approximation guarantees. Using sampling techniques from statistics and computational learning theory [Pol84, Hau92], we show that the number of records needed is at most logarithmic and in some cases independent of the cardinality of the input dataset. The sampling results also motivate a new model of clustering that is in the spirit of the "PAC (probably approximately correct)" learning model. We show how the standard k–median problem fits into this new setting, and also give an efficient algorithm for clustering k-term DNF expressions.

k–median clustering Let (X, d) be a metric space. The k–median problem is as follows. Given a finite subset $R \subseteq X$ of points, find a set $c_1, \ldots, c_k$ that minimizes $\sum_{x \in R} \min_i d(x, c_i)$. That is, find k cluster centers so that the total distance from each point in S to its nearest center is minimized. When it is required that each c_i be chosen from R, we call this the *discrete* k-median problem. If we allow also that $c_i \in X - R$, we call this the *continuous* version of the k-median problem. The problems are NP-hard, and numerous approximation algorithms have been considered (e.g., [CG99, CGTS99, JV99]) with running times that are typically $\tilde{O}(n^2)$ where $n = |R|$. (Note that for an arbitrary metric space, to query for all pairwise distances alone requires $O(n^2)$ time.)

In many of the data mining applications mentioned above, the number of data items n is so large that it tends to dominate other parameters, hence the desire for algorithms that are not only polynomial, but in fact are sublinear in n. Indyk [Ind99] gives a $\tilde{O}(nk)$ algorithm that outputs 2k centers that are a $O(1)$-approximation to the optimum k centers with constant probability. The sample sizes needed for his algorithm are $\tilde{O}(\sqrt{nk})$. Similar sampling techniques are used to approximate nearest neighbor computations. That these approaches work is often shown by arguing that sampled data is in some sense representative of the entire dataset.

Previous work in statistics on uniform convergence characterizes conditions under which a single sample S is sufficiently large so that for any function f chosen from a class F, the empirical mean of f computed on the sample is close ("within epsilon") of the true mean of f on the entire distribution. Similarly, work in computational learning theory has addressed sample bounds necessary and sufficient for computing the average error of a hypothesis f with respect to an unknown function to be learned, when f is chosen from some class F of possible hypotheses.

A key property in both of these approaches is that the sample bounds *do not depend on the domain of* f (which may be infinite), but rely instead on any of a variety of combinatorial or structural characteristics of the class F itself. (For example, the VC-dimension of F, the pseudo-dimension or metric dimension, $\ln F$ when F is finite.) These results say, in essence, that one does not need a sample S that is representative of the entire dataset R, but rather it is sufficient to obtain a sample S that adequately represents *the behavior of every function* of F on the entire dataset.

We recast clustering as the problem of choosing from a class of possible clustering functions C a clus-

[*] Hewlett-Packard Labs, Palo Alto, CA 94304. Email: nmishra@hpl.hp.com

[†] IBM TJ Watson Labs. Email: oblinger@us.ibm.com

[‡] University of Illinois at Urbana-Champaign, Urbana, IL 61801. Research supported in part by NSF Grants SBR-9720304 and IIS 99-07483. Email: pitt@uiuc.edu

tering function that has the smallest mean value over the dataset. Applying sampling results we will show that it is sufficient to simply find a clustering function that has the smallest mean value on the sample. Now if a standard k–median algorithm for finding approximately optimal k-clusterings is applied to the sample, we obtain an approximation algorithm that requires a sample whose size depends only on the desired accuracy and on other necessary parameters (which in some cases involve the diameter of the space) but is independent of the cardinality of the set R to be clustered.

More specifically, suppose that (X, d) is a metric space, with $d : X \times X \rightarrow [0, M]$. Further suppose that there exists an approximation algorithm for k–median clustering on (X, d) such that for any finite $R \subseteq X$ to be clustered, the clustering algorithm runs in time $T(n)$ and guarantees that $A(R) \leq \alpha\mathrm{Opt}(R)$, where $n = |R|$, $A(R)$ is the sum of distances from each point to its nearest center as found by the clustering algorithm, $\mathrm{Opt}(R)$ is the sum of distances from each point to its nearest center in an optimum k–median clustering, and α is some fixed constant. Then we can apply the above technique to obtain a new k–median clustering algorithm that runs in time $T(\tilde{O}((\frac{M\alpha}{\epsilon})^2(k \ln \frac{n}{\delta})))$ and that with probability at least $1 - \delta$, finds a k–median clustering such that for any subset $R \subseteq X$ $A(R) \leq 2\alpha\mathrm{Opt}(R) + \epsilon$. Various concrete results are now obtained by considering values of α and T from extant clustering approximation algorithms. Thus, our results can replace existing $O(n^2)$ approximate clustering algorithms with algorithms whose dependence on n is $\tilde{O}(\ln^2 n)$. The price paid is the ratio of the diameter M of the space to the desired accuracy ϵ.

For the particular case of clustering in d-dimensional Euclidean space, we obtain time and sample bounds completely independent of the input dataset. By running any α factor approximation algorithm on a sample of size $\tilde{O}((\frac{M\alpha d}{\epsilon})^2 k)$, we obtain a k-clustering with value A, and for which the same guarantee holds: For any finite set $X \subseteq \mathrm{R}^d$ to be clustered $A(X) \leq \alpha\mathrm{Opt}(X) + \epsilon$.

The dependence on M in the sample size and running time cannot be removed by isometric scaling of the problem. In particular, by normalizing a problem with large diameter M down to 1 (thus reducing the required sample size by a factor of $1/M^2$), clustering, and then rescaling back to M, the resulting clustering guarantee will blow up to $\alpha\mathrm{Opt}(X) + \epsilon M$.

Unless otherwise stated, all of our results apply to both the continuous k-median problem, and the discrete k-median problem.

PAC clustering

A common view of clustering is that of partitioning a collection of points into similar sets. A more general view takes into consideration two other aspects of the problem: First, the points may come from an external environment and represent only a sampling of points whose classification or partitioning is of interest despite their absence from the dataset. Second, we may be interested in an objective function that measures the quality of a clustering based not only on the finite sample, but rather on the entire distribution-weighted space. (For example, objective functions that incorporate notions of average distribution-weighted distance). In the case of k–median, the clustering output is automatically a partition of the entire space via the induced Voronoi diagram, so this first issue, though not the second, is addressed. Other types of clustering may address neither of these points.

To properly deal with these two aspects of clustering, we introduce "probably approximately correct (PAC)" clustering of arbitrary probability distributions on a potentially infinite space, as well as the use of "conceptual" clustering, where each cluster is chosen from some concept class of cluster descriptions. Our definitions and results are similar to investigations on PAC learning [Val84] from learning theory and machine learning. We show how k–median clustering fits into the new framework, and then finish with a new application to DNF clustering.

Related Work Besides k–median, numerous other measures exist for evaluating the cost of a clustering. One common measure, known as k–center, is the maximum radius of any cluster. Minimizing the maximum radius is NP-hard, although constant factor approximation algorithms do exist [HS86, Gon85, FG88]. Other cost measures have been proposed and studied that we do not describe here. New clustering measures are still being defined, e.g., Kannan, Vempala and Vetta [KVV00] use the notion of conductance to evaluate a clustering.

Algorithms for solving these clustering problems tend to share a common behavior: they make multiple passes through the data. Thus applying such algorithms to very large datasets may be difficult.

Techniques for coping with large datasets typically involve computing some compressed representation (usually a smaller set of points). For example, for the k–center problem, k points are shown to be a sufficient representation for a constant-factor approximation algorithm in the incremental model [CCFM97]. More recently, Alon, Dar, Parnas, and Ron [ADPR00] compute a compressed representation using sampling. Their algorithms find clusterings where all but a small fraction of the points have approximately optimum cost.

For the k–median problem, we have already noted that sampling can be used to obtain a good representation [Ind99]. The results given here also support the viability of sampling. Another technique [GMMO00], based on divide and conquer, partitions the data into pieces and uses clustering itself as a method of computing a compressed set of points. Divide and conquer techniques do not compete with sampling and in fact can be used in tandem to obtain more efficient algorithms [GMMO00].

2 Preliminaries

As discussed in the introduction, we express clusterings as functions whose mean values are to be estimated. Let (X, d) be a metric space, and let $c_1, c_2, \ldots, c_k \in X$. (These are the cluster "centers".)

The choice of $c_1, \ldots, c_k$ determine a natural clustering which we denote $f_{c_1,\ldots,c_k}$: each point of X is assigned to the closest center c_i, i.e., $f_{c_1,\ldots,c_k}(x) = \min_i\{d(x, c_i)\}$. The *cost* of a clustering with respect to X is just $\sum_{x \in X} f_{c_1,\ldots,c_k}(x)$. The k–median clustering problem is the following: Given finite X find centers $c_1, \ldots, c_k$ that minimize cost $\sum_{x \in X} f_{c_1,\ldots,c_k}(x)$. Clearly, this is equivalent to finding centers that minimize the expected value $E_X(f_{c_1,\ldots,c_k})$. The class of "clustering cost functions" that we may choose from are exactly these: $F_X = \{f_{c_1,\ldots,c_k} : c_1, c_2, \ldots, c_k \in X\}$.

Because X may be large (we'll see how to deal with infinite X in Section 4) we will draw a sample S from X and show that with high probability minimizing $E_S(f)$ is like minimizing $E_X(f)$. Moreover, approximately minimizing $E_S(f)$ is like approximately minimizing $E_X(f)$. Suppose that f_S has minimum sample cost and that f_X has minimum true cost over X. If we have a constant (α) factor approximation algorithm, we show that a sample of sufficiently large size exists such that with probability at least $1-\delta$, $E_X(f_X) \leq \alpha E_X(f_S) + \epsilon$.

3 Sublinear Time Approximate k–median Clustering

Given a constant (α) factor approximation algorithm that runs in time $T(n)$, we show the following (1) In an arbitrary metric space (X, d) we can compute in time $T(\tilde{O}((\frac{M\alpha}{\epsilon})^2 k \log n))$ a k clustering such that for all X, $(A(X) - 2\alpha OPT(X)) \leq \epsilon$ with high probability. (2) Assuming a distance metric on R^d, we can compute in time $T(\tilde{O}((\frac{M\alpha d}{\epsilon})^2 k))$ a k clustering such that for all X $(A(X) - \alpha OPT(X)) \leq \epsilon$ with high probability. The section concludes with a discussion of how to obtain M if it is unknown.

3.1 Clustering in Metric Space

Let (X, d) be a metric space and let S be a sample drawn independently and identically from X. Assume we compute a k–median clustering with approximately minimum *sample cost*, i.e., approximately minimum average distance from a point in S to its closest center. We'll show that this clustering also has approximately minimum *true cost*, i.e., approximately minimum average distance from a point in X to its closest center. To make this more formal, consider the family of k–median cost functions $F_D = \{f_{c_1,\ldots,c_k} : c_i \in D, f_{c_1,\ldots,c_k}(x) = \min_i d(x, c_i)\}$. We show that the discrete k–median clustering d_S with minimum sample cost, i.e., $d_S = \text{argmin}_{f \in F_S} E_S(f)$, is an approximately good clustering of X, in other words, $E_X(d_S)$ is close to $\min_{f \in F_X} E_X(f)$.

To prove our result, we show that the sample cost converges to the true cost uniformly for each k-median cost function quickly. Then combining known relationships between optimum discrete and continuous clusterings with uniform convergence we obtain our result. We begin with a uniform convergence lemma due to Haussler [Hau92]/Pollard [Pol84].

LEMMA 3.1. (HAUSSLER/POLLARD) *Let F be a finite set of functions on X with $0 \leq f(x) \leq M$ for all $f \in F$ and $x \in X$. Let $S = x_1, \ldots, x_m$ be a sequence of m examples drawn independently and identically from X and let $\epsilon > 0$. $Pr(\exists f \in F : |E_X(f) - E_S(f)| \geq \epsilon) \leq \delta$ when $m \geq \frac{M^2}{2\epsilon^2}(\ln|F| + \ln\frac{2}{\delta})$.*

The above lemma implies fast uniform convergence of the k–median family of cost functions. Note that $|F| = O(n^k)$ since there are $\binom{n}{k}$ ways to select k centers from n points. Thus the probability that there exists a k–median clustering f whose sample cost deviates from its true cost by more than ϵ is at most δ when the sample size is $\tilde{O}((\frac{M\alpha}{\epsilon})^2 k \log n)$.

The following folklore lemma describes the relationship between the optimum discrete and continuous clusterings (see for example [GMMO00]).

LEMMA 3.2. *The sample cost of d_S is no more than twice the sample cost of c_S, where c_S is the optimum continuous clustering of S, i.e., $c_s = \text{argmin}_{f \in F_X} E_S(f)$.*

The previous lemmas can now be combined to obtain our metric space result. Let $\hat{d}_S$ be a constant (α) factor approximation to d_S and d_X be the optimum k–median clustering of X.

THEOREM 3.1. *In an arbitrary metric space (X, d) assuming a constant α-factor k–median approximation algorithm that runs in time $T(n)$, we can draw a sample*

	$\hat{d}_S$		d_S		c_S		d_X
E_S	(2)	$\leq \alpha\cdot$	(3)	$\leq 2\cdot$	(4)	$\leq$	(5)
	$\approx$						$\approx$
E_X	(1)						(6)

Table 1: Proof of Theorem 3.1.

S of size at least $8(\frac{\alpha M}{\epsilon})^2(k \ln n + \ln \frac{4}{\delta})$ and obtain a k–median clustering $\hat{d}_S$ in time $T(|S|)$ such that with probability at least $1-\delta$, $E_X(\hat{d}_S) - 2\alpha E_X(d_X) \leq \epsilon$.

Proof. The main idea is to prove that the sample and true costs of $\hat{d}_S$ and d_X are all roughly the same. The first and last steps of the proof utilize uniform convergence (Lemma 3.1). The middle steps of the proof use properties of the type of clustering computed. The sequence of steps is shown in Table 1. The rows of the table correspond to the sample and true cost and the columns correspond to the different clusterings.

By uniform convergence and by the sample size given in the statement of this theorem, the sample and true costs for each $f \in F_X$ are within $\frac{\epsilon}{4\alpha}$ with high probability, i.e., $|E_S(f) - E_X(f)| \leq \frac{\epsilon}{4\alpha}$ with probability at least $1-\frac{\delta}{2}$. We apply uniform convergence (Lemma 3.1) to the two clusterings $\hat{d}_s$ and d_X to obtain that the values (1) and (2) as well as the values (5) and (6) in Table 1 are close.

Observe that the sample cost of $\hat{d}_S$ is within a factor of α of d_s since we ran an α-approximation algorithm on the sample S, hence the inequality between (2) and (3) in Table 1. The relationship between between (3) and (4) follows since the sample cost of d_S is within a factor of 2 of the sample cost of c_S (Lemma 3.2). By the optimality of c_S, the sample cost of c_S is less than the sample cost of d_X, hence the inequality between (4) and (5).

The theorem follows by combining the above steps appropriately. □

3.2 Clustering in Euclidean Space

In this subsection we assume a distance metric on R^d. There are two problems we can consider. The first is given a collection X of n points in R^d, how large of a sample must we draw from X before clustering the sample well implies that we clustered X well. This problem was solved in the previous section since it is a restricted version of the metric space clustering problem. The second problem we can define is suppose X is really R^d, or perhaps some subspace of R^d, how large of a sample must we draw from X before clustering the sample well implies that we approximately minimized the expected true clustering cost[1]. We now consider this second problem (although the results will also still apply to the first problem).

The proof techniques are similar to the metric space result in that we rely on both uniform convergence and the sample clustering's approximate optimality. Note that since $X \subseteq R^d$, $|X|$ is now uncountably infinite and thus the number of different k–median clusterings (i.e., $|F_X|$ is also uncountably infinite). Following Pollard [Pol84], in the sample bounds we derive, the number of different k–median clusterings which was previously $O(n^k)$ is replaced with an ϵ-net of size $O((\frac{M}{\epsilon}d)^{dk})$. Thus the sample size we obtain is independent of n, namely $\tilde{O}((\frac{M\alpha d}{\epsilon})^2 k)$.

An *ϵ-net*, F_ϵ, for F is a family of functions such that for each $f \in F$ and for any set X of points there exists an $f_\epsilon \in F_\epsilon$ such that $|E_X(f - f_\epsilon)| \leq \epsilon$. Intuitively an ϵ-net [Pol84] is a rich, representative set of functions that ensures any element of F is close to some element of the ϵ-net. To obtain an ϵ–net for F, we consider the subset of k–median cost functions that correspond to centers at evenly spaced "gridpoints". We say that $\langle x_1, \ldots, x_d \rangle$ is a *d-dimensional γ-gridpoint* if for each x_i, x_i is a multiple of γ. In the following lemma we'll show that if these gridpoints are spaced close enough together then we have an ϵ-net for F. (The proof applies to any distance metric d on R^d, although for purposes of presentation, we assume the L_2 distance metric.)

Lemma 3.3. (k–median has a small ϵ-net) *Let P be a set of n points in R^d of bounded diameter M. Let $F = \{f_{c_1,\ldots,c_k} : f_{c_1,\ldots,c_k}(x) = \min_i d(x, c_i)\}$. Let $F_\epsilon = \{f_{c_1,\ldots,c_k} : f_{c_1,\ldots,c_k}(x) = \min_i d(x, c_i)$ where c_i is a d-dimensional $\frac{\epsilon}{2d}$-gridpoint in the bounded diameter space$\}$. For each f in F, there exists $f_\epsilon \in F_\epsilon$ such that $|E_P(f - f_\epsilon)| \leq \epsilon$.*

Proof. For $f = f_{c_1,\ldots,c_k} \in F$ define $f_\epsilon = f_{\hat{c_1}}, \ldots, f_{\hat{c_k}}$ to be the function in F_ϵ with the property that $\hat{c_i}$ is the nearest "grid point" to c_i, so that $d(c_i, \hat{c_i}) \leq \epsilon$. Such a $\hat{c_i}$ exists since by assumption F_ϵ is of sufficient granularity for the Euclidean distance measure: $d(c_i, \hat{c_i}) \leq \sqrt{d(\frac{\epsilon}{d})^2} = \sqrt{\frac{\epsilon^2}{d}} \leq \epsilon$.

We now show that for all $x \in X$, $|f(x) - f_\epsilon(x)| \leq \epsilon$. That $E_P|f - f_\epsilon|$ will immediately follow (since if for each x the difference is at most ϵ the expected difference can't be greater). Let x be a point that is closest to center c_i. Since we want to bound the difference between $f(x)$ and $f_\epsilon(x)$, it is sufficient to show that $d(x, c_i)$ is close to $d(x, \hat{c_i})$ since $f_\epsilon(x) \leq d(x, \hat{c_i})$. By the triangle inequality, we have $d(x, \hat{c_i}) \leq d(x, c_i) + d(c_i, \hat{c_i}) \leq d(x, c_i) + \epsilon$. A

[1] In the next section we will consider the more general problem of assuming a probability distribution on R^d.

similar argument can be used to show that $d(x, c_i) \leq d(x, \hat{c}_i) + \epsilon$. □

Note that the number of gridpoint functions $|F_\epsilon|$ is of finite size, namely $O(\frac{dM}{\epsilon})^{dk}$. Thus the sample cost for each function in F_ϵ approaches the true cost provided the sample is of size $\tilde{O}((\frac{M\alpha d}{\epsilon})^2 k)$ by Lemma 3.1.

We use the ϵ-net for the purposes of analysis only. The algorithm "run an approximation algorithm on a sample" remains the same. The ϵ-net only affects the sample size. Following Pollard [Pol84], we use this ϵ-net to prove that clustering a sample well implies that we clustered X well. Let $\hat{c}_S$ be a constant (α) factor approximation to the optimum clustering c_S of S (i.e., c_S =argmin$_{f \in F_X} E_S(f)$) and let c_X be the optimum clustering of X, (i.e., c_X =argmin$_{f \in F_X} E_X(f)$).

Theorem 3.2. *For $X \subseteq R^d$, assuming a constant α-factor k–median approximation algorithm that runs in time $T(n)$, we can draw a sample S of size at least $18(\frac{M}{\epsilon})^2(dk \ln \frac{12dM}{\epsilon} + \ln \frac{4}{\delta})$ and obtain a k–median clustering $\hat{c}_S$ in time $T(|S|)$ such that with probability at least $1 - \delta$, $E_X(\hat{c}_S) - \alpha E_X(c_X) \leq \epsilon$.*

Proof. We show that the difference between the true cost of c_S and the true cost of c_X is no more than ϵ with high probability, i.e., $E_X(c_S) - E_X(c_X) \leq \epsilon$ with probability at least $1 - \delta$. The theorem can be easily extended to the case where we compute a constant factor approximation $\hat{c}_S$ of c_S.

Let $c_{S,\epsilon}$ and $c_{X,\epsilon}$ refer to the closest gridpoint functions of c_S and c_X, respectively. We now explain the chain of inequalities as shown in Table 2 needed for the proof. Note that by the sample size given in the statement of the theorem we have uniform convergence for each $f \in F_{\frac{\epsilon}{6}}$. Thus the sample and true costs for each gridpoint or ϵ-net clustering are close. This implies that the values (2) and (3) as well as the values (5) and (6) in Table 2 are close.

Further, the (sample or true) cost of any clustering and its nearest gridpoint clustering is no more than $\frac{\epsilon}{6}$, hence the values (1) and (2) as well as the values (3) and (4), as well as the values (6) and (7) are close. Finally, since c_S is the optimum clustering of S, it's sample cost is better than the sample cost of $c_{X,\epsilon}$. Hence the inequality between (4) and (5). The theorem can be obtained by appropriately chaining these inequalities together. □

	c_S		$c_{S,\epsilon}$		c_S		$c_{X,\epsilon}$		c_X
E_S			(3)	$\approx$	(4)	$\leq$	(5)		
			$\approx$				$\approx$		
E_X	(1)	$\approx$	(2)				(6)	$\approx$	(7)

Table 2: Proof of Theorem 3.2.

3.3 Clustering Datasets of Bounded but Unknown Range

While any finite X fall in a bounded range, we may not know apriori what that bound is. Further, determining the bound may be expensive if, for example, it requires scanning through a very large dataset. In the case that X is infinite, determining the bound by scanning X is clearly impossible.

We give a way to estimate M with M' assuming $X \subseteq R^d$ and show that an approximately good clustering can still be obtained with M', although not as good as if we knew M.

The algorithm is as follows. First we draw a sample of size given in Lemma 3.4 and compute M', the largest distance between any pair of points in the sample. Then we draw a sample of size proportional to $\tilde{O}((\frac{M'\alpha d}{\epsilon})^2 k)$ as in Theorem 3.2 and cluster that sample.

For purposes of presentation we assume the points fall in $[0, B]^d$ and we wish to estimate B.

Lemma 3.4. *Let $H = [0, B]^d$ and $G = [x, B - y]^d$ with the property that the fraction of points on any strip between G and H is at most $\frac{\epsilon}{2d}$. The probability that no point is drawn from any one of these strips is at most δ when a sample of size $m \geq \frac{2d}{\epsilon} \log \frac{2d}{\delta}$ is drawn.*

Proof. The probability we fail to draw a point in a particular strip between G and H in m trials is at most $(1 - \frac{\epsilon}{2d})^m$. This probability is at most $\frac{\delta}{2d}$ when $m \geq \frac{2d}{\epsilon} \log \frac{2d}{\delta}$. The probability we fail to draw a point in all $2d$ strips between G and H in m trials is thus at most δ by the sample size given in the statement of the lemma. □

The lemma implies that the cost for the points in the cube G is at most ϵ and the cost for the points between G and H is at most ϵM. Thus we can claim the following theorem.

Theorem 3.3. *If a bound M on the space is unknown then estimating M with M' on a sample of size given in Lemma 3.4 and running a constant (α) factor approximation algorithm on a sample of size $\tilde{O}((\frac{M'\alpha d}{\epsilon})^2 k)$ yields a clustering f such that $E_X(f) - \alpha E_X(f_X) \leq \epsilon(1 + M)$*

4 PAC clustering concepts

In this section we extend our results in two directions: incorporating a notion of "probably approximately correct (PAC)" clustering of an infinite data set, and addressing the issue of "conceptual" clustering, involving clusters that are more than merely a collection of data points. Our definitions and results are motivated by similar investigations on PAC learning [Val84] from learning theory and machine learning. In what follows, we show how k-median clustering fits into the new framework, and then finish with a new application to DNF clustering.

4.1 Conceptual clustering

In many applications, algorithms that output conclusions such as "this listing of 43Mb of data points are in one cluster" are of little utility. In order to be useful within the context of an intelligent system or in data analysis and decision-support, clusters should have meaning that transcends their composition. Within machine learning, this motivated investigation into "conceptual clustering" (e.g., [PR88]). Similarly, in learning theory, one speaks of the learnability of a class of *representations* of classifiers, e.g., DNF expressions, polytopes, finite automata, etc. In these cases, the goal is to learn a classifier by finding a *representation* from the specified class that has small classification error on unseen examples. Not only does this conceptual view offer a meaningful description of data, it also serves as a predictor of future data. Unlike k-median algorithms, cluster membership for unseen data is not necessarily a byproduct of sample data clustering.

Define a *concept class* as a pair (X, C) where X is the *example space* and C, the *concept space*, is a collection of representations of subsets of X. Typically X and C are families $\{X_d\}$, $\{C_s\}$ paramaterized by some complexity measure (e.g., $X_d = \{0,1\}^d$, and $C_s =$ DNFs expressible in s bits). Define an (X, C) conceptual k-clustering to be any choice $\langle c_1, c_2, \ldots, c_k\rangle \in C^k$ of k concepts from C. **Example:** For the k–median problem, we can take X as d-dimensional Euclidean space, C as the set of k-tuples of points representing the induced Voronoi partition.

4.2 Clustering data from distributions

How "good" is a conceptual k-clustering? In practical applications, the set S of data to be clustered is typically a subcollection of a much larger, possibly infinite set, sampled from an unknown probability distribution. We describe a model of clustering from a probability distribution that we believe to be new and of independent interest. The model is similar in spirit to the PAC model of learning, in that error is distribution-weighted. We will require that a clustering algorithm find a clustering (approximately) optimizing some objective function while covering most of the distribution.

Let D be an arbitrary probability distribution on X. Most generally, the quality of a clustering depends simultaneously on all clusters in the clustering, and on the distribution. Thus, the goal will be to minimize (or maximize) some objective function $Q(\langle c_1, c_2, \ldots, c_k\rangle, D)$ over all choices of k-tuples c_i.

Example continued: For k–median, if $C_1, \ldots, C_k$ are the Voronoi cells with centers $c_1, \ldots, c_k$, in the finite case the tightness of a clustering is the sum of intracluster distances. For infinite data, this sum may well be infinite depending on the distribution. We naturally define the tightness of a cluster C_i as the expected distance *over the entire cell* from a point to its center c_i: $T(C_i) = \sum_{x \in C_i} dist(x, c_i) D(x|C_i)$. These can then be accumulated into a single objective function by summing[2] over the individual clusters, weighted by their likelihood: $Q(\langle C_1, C_2, \ldots, C_k\rangle, D) = \sum_i T(C_i)D(C_i)$. Often the most natural objective function is obtained in this way – taking a distribution-weighted average of individual cluster tightnesses.

Definition 4.1. *A concept class (X, C) is (α, β) PAC-clusterable with objective function Q iff there is an algorithm A such that for all probability distributions D on X, if A is given as input an integer $k > 0$, numbers γ, δ, and $\epsilon < 1$, and access to examples drawn iid from D, then with probability at least $1 - \delta$, A outputs a collection of βk clusters $\langle c_1, \ldots, c_{\beta k}\rangle$ such that $D(\cup_i c_i) \geq 1 - \gamma$, and $|Q(\langle c_1, \ldots, c_{\beta k}\rangle, D) - \alpha Q(\langle c_1^*, \ldots, c_k^*\rangle, D) < \epsilon$, where $\langle c_1^*, \ldots, c_k^*\rangle$ is the k-clustering that covers the entire space X and minimizes (or maximizes) Q on D.*

The allowance for neglecting some fraction of data points as "outliers" has been addressed in other clustering algorithms [Sch00, ADPR00]. In our context it appears necessary in cases when finding clusters that cover the entire space may prove combinatorially difficult (is a given DNF a tautology?).

When the objective function is a weighted average of individual cluster tightness, as we defined for k–median, the function can often be rewritten as the expected value of a function. As a consequence, uniform convergence results as described in the last section can be applied to obtain sampling bounds for clustering that are independent of the data set.

Lemma 4.1. *Let $c_1, \ldots, c_k$ be centers corresponding to clusters $C_1, \ldots, C_k \subseteq X$. Let c be a function that given x returns the distance from x to its closest center $c_1, \ldots, c_k$. $Q(\langle C_1, C_2, \ldots, C_k\rangle, D) = E_D(c)$.*

Proof.

$$Q(\langle C_1, \ldots, C_k\rangle, D) \quad = \quad \sum_i T(C_i)D(C_i)$$

[2] In this abstract we consider only discrete distributions to avoid dealing with measurability issues.

$$\begin{aligned} &= \sum_i \sum_{x \in C_i} d(x, c_i) D(x|C_i)) D(C_i) \\ &= \sum_i \sum_{x \in C_i} d(x, c_i) D(x) \\ &= \sum_i \sum_{x \in C_i} c(x) D(x) \\ &= E_D(c) \end{aligned}$$

□

In section 3 we showed that when c is the k-median cost function described above, $E_S(c)$ for a finite sample S converges uniformly to $E_D(c)$, hence the approximate clustering method of "sample and apply a known approximation algorithm" provides a PAC-clustering of comparable quality.

THEOREM 4.1. *If the k–median problem has an (α, β) bicriterion clustering approximation, then k–median is (α, β)-PAC clusterable.*

4.3 PAC-clustering (disjoint) k-term-DNF

We give a $d^{O(k^2)}$ algorithm for (optimally) PAC-clustering disjoint conjunctions over d Boolean variables. A k-clustering forms a disjoint k-term-DNF expression. The algorithm has applications to unsupervised learning of decision trees, as the leaves of a k-leaf decision-tree are describable by k mutually disjoint conjunctions. Since even supervised learning of small decision trees have been found to be useful in practice [AHM95], the extension here is of particular interest.

Let $X = \{0,1\}^d$ and let concepts be *terms* (conjunctions of literals), where each literal is one of the d Boolean variables $\{x_1, x_2, \ldots x_d\}$ or their negations. A k-clustering is a set of k disjoint terms, $\{t_1, \ldots, t_k\}$, (where no two t_i's are satisfied by any one assignment). Define the quality function $Q(\langle t_1, \ldots, t_k \rangle, D)$ $= \sum_{i=1}^{k} |t_i| D(t_i)$, where $D(t_i)$ is the fraction of the distribution, D, satisfied by t_i. Thus, the objective is to *maximize the length* of the cluster descriptions (longer, more specific terms, are more "tight"), weighted by the probabilities of the clusters. (See [PR88] for a related view of clustering via long conjunctions.) It is easy to see that an optimum k-clustering is always at least as good as an optimum $k-1$ clustering, since any cluster can be split into two by constraining some variable, obtaining two tighter clusters with the same cumulative distributional weight.

To solve the problem, we define the *signature* of a disjoint DNF expression, which is similar to the "discriminant" used by Angluin [Ang87], and often used in learning restricted forms of DNF with membership and equivalence queries (see e.g., [BKK+94]). Define a k-signature to be a sequence $\langle \ell_{ij} \rangle_{1 \le i < j \le k}$ where each ℓ_{ij} is a literal in $\{x_1, \ldots, x_d, \bar{x_1}, \ldots, \bar{x_d}\}$. Associated with each k-signature s is a "skeleton" k-term disjoint DNF $s_1 + \cdots + s_k$, where term s_i contains exactly those literals ℓ_{ij} for $i < j$, and the complements of literals ℓ_{ki} for $k < i$. A k-term DNF $t_1 + \cdots + t_k$ is a *specialization* of a k-skeleton $s_1 + \cdots + s_k$ iff for each i, the set of literals in s_i is contained in the set of literals in t_i. Clearly, if s is a k-signature, then the skeleton induced by s is a k-term disjoint DNF, as is any k-term DNF that is a specialization of that skeleton. Furthermore, every k-term disjoint DNF expression is a specialization of some skeleton induced by a k-signature.

For a sample S and a term t, let $S(t) = |\{x \in S : t(x) = 1\}|/|S|$, the observed frequency in S of points that satisfy t.

```
Algorithm cluster k-term-disjoint-DNF ($(k, \gamma, \delta, \epsilon)$)

1. Draw a sample $S$ of cardinality $m$ from $D$ as
   described in Theorem 4.2
2. For each choice of signature $s$ and associated skele-
   ton terms $s_1, s_2, \ldots, s_k$,
   (a) Partition $S$ into buckets with $x$ in bucket $B_i$
       iff $x$ satisfies skeleton term $s_i$. If some $x \in S$
       satisfies no term, then start over at step 2.
       with the next signature.
   (b) For each bucket $B_i$
       i. Let $t_i$ be the most specific term satisfied
          by all examples in $B_i$.
       ii. Let $C_s$, the clustering induced by signa-
           ture $s$, be the collection of all such terms
           $t_i$.
       iii. Compute the empirical frequency $S(t_i)$ as
            defined above.
   (c) Define $Q(C_s, S) = \sum_i |t_i| S(t_i)$ be the esti-
       mated value of $Q(C_s, D)$.
3. Output the clustering $C_s$ associated with the sig-
   nature $s$ for which the computed estimate $Q(C_s, S)$
   is maximized.
```

THEOREM 4.2. *(Algorithm correctness)*
Let D be an arbitrary distribution on $\{0,1\}^n$, and let C^D be an optimal disjoint k-term DNF clustering for D. With probability at least $1 - \delta$, the algorithm above outputs a disjoint k-term DNF clustering C such that $D(C) \ge 1 - \gamma$ and $Q(C^D, D) - Q(C, D) < \epsilon$ provided that the sample S drawn in Step 1 of algorithm is of size at least $\min\{\frac{1}{\gamma}(dk \ln 3 + \ln \frac{2}{\delta}), \frac{2d^2k^2}{\epsilon^2}(d \ln 3 + \ln \frac{2}{\delta})\}$. Hence, disjoint k-term DNF expressions are (1,1)-PAC clusterable for constant k.

Let C be the clustering output by the algorithm. To prove theorem 4.2 we rely on the following three lemmas.

Lemma 4.2. *(i) C has at most k mutually disjoint terms, and (ii) C maximizes $Q(C,S)$ over all such disjoint k-term DNF expressions.*

Lemma 4.3. *$Pr[D(C) < 1-\gamma] < \frac{\delta}{2}$*

Lemma 4.4. *$Pr[(\exists$ a disjoint k-term DNF clustering $T)$ $|Q(T,S) - Q(T,D)| > \frac{\epsilon}{2}] < \frac{\delta}{2}$.*

We prove the lemmas below. Here we prove Theorem 4.2 assuming that the lemmas are true. Let C^D be a clustering that maximizes quality Q over the distributiom D. We will show that $\Pr[Q(C^D,D) - Q(C,D) > \epsilon] < \frac{\delta}{2}$. From this and Lemmas 4.3 and 4.2 (i), the theorem follows immediately.

By Lemma 4.2 (ii), C is the best clustering on the sample (C maximizes $Q(C,S)$), hence by Lemma 4.4, with high probability the quality of every clustering on the sample is close to its quality on the distribution. Chaining inequalities in a manner similar to Theorems 3.1 and 3.2, the result is obtained. □

The running time of the algorithm is dominated by the enumeration of $d^{O(k^2)}$ signatures.

Proof of Lemma 4.2: Part (i): By construction, for some k-signature s, C is a specialization of the skeleton DNF associated with s. It follows by comments above that C contains exactly k mutually disjoint terms.

Part (ii): Let C_S be a k-term disjoint DNF expression that maximizes $Q(\cdot,S)$. Then by comments above, C_S is a specialization of a skeleton DNF expression induced by some k-signature s. When s is enumerated in the main loop, the skeleton DNF is formed. By construction, the DNF C_s obtained is the most specific DNF that is a specialization of the skeleton and covers S, hence is at least as specific as C_S. Since all points of S are covered by C_s, the quality $Q(C_s,S)$ cannot be increased by making terms more general so as to cover more data. The algorithm outputs a clustering C with quality at least as good as C_s on sample S, hence at least as good as C_S. □

Proof of Lemma 4.3 A standard argument (c.f. [BEHW87]) shows that the probability that some k-term DNF from class F covers less than $1-\gamma$ of the distribution D, yet covers all sample points S, is at most δ if $|S| \geq \frac{1}{\gamma}\ln\frac{|F|}{\delta}$. The number of k-term DNF expressions over d variables is at most $\binom{3^d}{k}$, since there are at most 3^d terms over d variables. Thus, if

$$|S| \geq \frac{1}{\gamma}\ln\frac{2\cdot 3^{dk}}{\delta} = \frac{1}{\gamma}(dk\ln 3 + \ln\frac{2}{\delta})$$

then the lemma follows. The sample set S is chosen at least this large when the algorithm begins. □

Proof of Lemma 4.4 For any k-term DNF $T = t_1 + \cdots + t_k$, $Q(T,D)$ depends on $|t_i|$ and $D(t_i)$ for each i. Thus, the difference between $Q(T,D)$ and $Q(T,S)$ for some sample S of D is uniquely determined by the difference of the empirical weights $S(t_i)$ from actual weights $D(t_i)$. Applying the uniform convergence results described in Lemma 3.1 with $\epsilon/2dk$ and $\delta/2$, and with F the set of terms (conjunctions) with at most d variables, the probability that any $D(t_i)$ differs from $S(t_i)$ by more than $\epsilon/2dk$ is at most $\delta/2$ provided that

$$\begin{aligned}|S| &\geq \frac{1}{2(\epsilon/2dk)^2}(\ln 3^d + \ln\frac{2}{\delta}) \\ &= \frac{2d^2k^2}{\epsilon^2}(d\ln 3 + \ln\frac{2}{\delta})\end{aligned}$$

Note that $|S(t_i) - D(t_i)| \leq \frac{\epsilon}{2dk}$ implies that $|Q(T,D) - Q(T,S)| \leq \frac{\epsilon}{2}$ since

$$\begin{aligned}|Q(T,D) - Q(T,S)| &= |t_1|[D(t_1) - S(t_1)] + \cdots \\ &\quad + |t_k|[D(t_k) - S(t_k)] \\ &\leq \frac{\epsilon}{2dk}(|t_1| + \cdots |t_k|) \\ &\leq \frac{\epsilon}{2}\end{aligned}$$

Thus since the algorithm draws a sample at least as large as needed to ensure uniform convergence of all conjunctions, the lemma follows. □

References

[ADPR00] N. Alon, S. Dar, M. Parnas, and D. Ron. Testing of clustering. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, 2000.

[AHM95] Peter Auer, Robert C. Holte, and Wolfgang Maass. Theory and applications of agnostic PAC-learning with small decision trees. In *Proc. 12th International Conference on Machine Learning*, pages 21–29. Morgan Kaufmann, 1995.

[Ang87] D. Angluin. Learning k-term DNF formulas using queries and counterexamples. Technical Report YALEU/DCS/RR-559, Department of Computer Science, Yale University, August 1987.

[BEHW87] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Occam's razor. *Information Processing Letters*, 24(6):377–380, April 1987.

[BKK$^+$94] A. Blum, R. Khardon, E. Kushilevitz, L. Pitt, and D. Roth. On learning read-k-satisfy-j DNF. In *Proc. 7th Annu. ACM Workshop on Comput. Learning Theory*, pages 110–117. ACM Press, New York, NY, 1994.

[CCFM97] Moses Charikar, Chandra Chekuri, Tomás Feder, and Rajeev Motwani. Incremental clustering and dynamic information retrieval. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, pages 626–635, El Paso, Texas, 4–6 May 1997.

[CG99] Charikar and Guha. Improved combinatorial algorithms for the facility location and k-median problems. In *FOCS: IEEE Symposium on Foundations of Computer Science (FOCS)*, 1999.

[CGTS99] Charikar, Guha, Tardos, and Shmoys. A constant-factor approximation algorithm for the k-median problem (extended abstract). In *STOC: ACM Symposium on Theory of Computing (STOC)*, 1999.

[FG88] Tomás Feder and Daniel Greene. Optimal algorithms for approximate clustering. In Richard Cole, editor, *Proceedings of the 20th Annual ACM Symposium on the Theory of Computing*, pages 434–444, Chicago, IL, May 1988. ACM Press.

[GMMO00] S. Guha, N. Mishra, R. Motwani, and L. O'Callaghan. Clustering data streams. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, 2000.

[Gon85] T. F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38(2-3):293–306, June 1985.

[Hau92] David Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100(1):78–150, September 1992.

[HS86] Dorit S. Hochbaum and David B. Shmoys. A unified approach to approximate algorithms for bottleneck problems. *Journal of the ACM*, 33(3):533–550, July 1986.

[Ind99] Indyk. Sublinear time algorithms for metric space problems. In *STOC: ACM Symposium on Theory of Computing (STOC)*, 1999.

[JV99] Jain and Vazirani. Primal-dual approximation algorithms for metric facility location and k-median problems. In *FOCS: IEEE Symposium on Foundations of Computer Science (FOCS)*, 1999.

[KVV00] Ravi Kannan, Santosh Vempala, and Adrian Vetta. On clusterings: good, bad and spectral. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, 2000.

[Pol84] D. Pollard. *Convergence of Stochastic Processes*. Springer Verlag, 1984.

[PR88] L. Pitt and R. E. Reinke. Criteria for polynomial-time (conceptual) clustering. *Machine Learning*, 2:371–396, 1988.

[Sch00] Leonard Schulman. Clustering for edge-cost minimization. In *Proc. 32nd Symposium on Theory of Computing*. ACM Press, New York, NY, 2000.

[Val84] L. G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, November 1984.