User-Friendly Clustering for Atmospheric Data Analysis

Benjamin J. Anderson
David R. Musicant
Anna M. Ritz
Dept. of Math and Computer Science
Carleton College
Northfield, MN 55057
{andersbe, dmusican, ritza}@carleton.edu

Andrew Ault
Deborah Gross
Melanie Yuen
Dept. of Chemistry
Carleton College
Northfield, MN 55057
{aulta, dgross, yuenm}@carleton.edu

Markus Gälli
TSI, Inc.
500 Cardigan Road
Shoreview, MN 55126
markus.gaelli@tsi.com

ABSTRACT
Atmospheric data analysis is an important area of scientific endeavor, with both government and industrial applications. Our work focuses on clustering particle data acquired via an Aerosol Time-of-Flight Mass Spectrometer (ATOFMS), which is sold and marketed by TSI, Inc. Most papers and software tools developed by the single-particle mass spectrometry community use the ART-2a clustering algorithm. We present in this paper a comparison of the well-known K-means algorithm with ART-2a in this application area. Specifically, we show that despite the entrenched position of the ART-2a algorithm in this domain, K-means is faster, more scalable, and considerably easier for practitioners to use while obtaining results of similar accuracy. For data mining practitioners in general and for those who develop software in particular, our work shows that in an important application area K-means is much easier for users to use than ART-2a without sacrificing accuracy. For researchers in the single-particle mass spectrometry community, our experiments demonstrate that ART-2a presents some issues that may be of concern. We propose that K-means offers an attractive alternative.

Categories and Subject Descriptors
I.5.3 [**Pattern Recognition**]: Clustering

General Terms
Algorithms, Measurement, Performance

Keywords
K-means, ART-2a, Atmospheric data analysis

1. INTRODUCTION
Atmospheric data analysis is an important area of scientific endeavor. Examples such as the study of atmospheric pollution and the detection of contaminants in the air indicate that there is a strong need for systems that can provide accurate and detailed information. Our work focuses on the analysis of single-particle mass spectrometry (SPMS) data. We use the Aerosol Time-of-Flight Mass Spectrometer (ATOFMS), which is sold and marketed by TSI, Inc. (Shoreview, MN). These instruments analyze the chemical makeup of aerosol particles suspended in air, one particle at a time. ATOFMS analysis has been used for a wide variety of research, industry and government applications. This technology is in active development for detection of airborne toxins and disease-related contaminants [5], which makes its importance for government applications all the more clear.

The NSF-funded EDAM project (Exploratory Data Analysis and Monitoring) is a joint collaboration between computer scientists, chemists, and atmospheric scientists. One of the major goals of this project is to provide software tools to atmospheric scientists to facilitate and automate the data mining of ATOFMS and other related forms of data. In this paper, we focus on the problem of clustering the data. A typical SPMS dataset consists of a series of mass spectra, each one of which is associated with an aerosol particle. The goal is to use these spectra to cluster these particles into similar groups. Most papers and software tools developed by the SPMS community use the ART-2a clustering algorithm. We present in this paper a comparison of the well-known K-means algorithm with ART-2a in this application area, and show that K-means as described in this paper is faster, more scalable, and considerably easier for practitioners to use while obtaining results of similar accuracy to ART-2a. These results are important for two reasons. For data mining practitioners in general and for those who develop software in particular, our work shows that in an important application area K-means is much easier for users to use than ART-2a is without sacrificing accuracy. For researchers in the SPMS community, our experiments demonstrate that ART-2a presents some issues that may be of concern.

K-means has been used in a variety of atmospheric contexts [4, 7]. However, the use of K-means on aerosol atmospheric data seems to be limited. In fact, any clustering analysis on such data is fairly recent. As single-particle chemical analysis has become more common, both with home-built and commercial instruments, research started focusing more on data analysis and results from field campaigns and lab experiments rather than on instrument development. In the mid-1990's, the first publications about specific methods for analyzing the data from SPMS instruments appeared, and their numbers have increased ever since, from 2 in 1996 to 10 and 8, respectively, in 2003 and 2004. In recent years, 20-40% of the publications dealing with SPMS focused on specific data analysis methods. The two methods discussed most in the literature are ART-2a [18, 19] and fuzzy-clustering algorithms[12, 13], with ART-2a being the method used in about

58% of all data analysis publications. These results are shown in Figure 1.

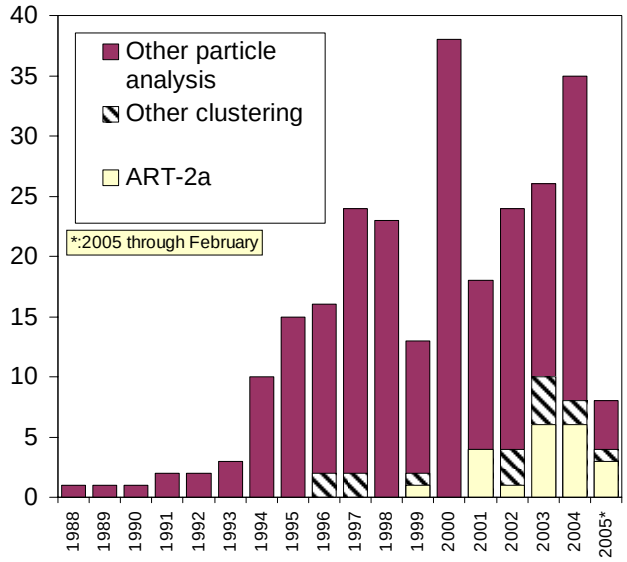

Figure 1: Distribution of single particle analysis research. ART-2a is a common choice.

Our contributions in this paper are as follows:

- We show that ART-2a, one of the dominant clustering algorithms used for ATOFMS data, is not as well suited to the task as the well-known K-means algorithm. K-means does not seem to have been used on this kind of data prior to this paper.
- We examine the usability characteristics of K-means when compared with ART-2a, and argue that K-means is the more user-friendly algorithm for a variety of reasons. Practitioners concerned with making clustering "easy to do," regardless of application area, should consider K-means over ART-2a.
- We demonstrate experimentally that on ATOFMS data, K-means performs faster and more automatically than ART-2a with comparable accuracy.

2. ATOFMS DATA

An *aerosol* particle is a particle of solid or liquid material which is small enough to be suspended in air. An ATOFMS desorbs and ionizes individual particles, sampled directly from the ambient atmosphere, into their constituent ions. For each particle the positive ions and the negative ions are separately represented by a *mass spectrum*. See Figure 2 for an example. A mass spectrum is a plot of signal intensity (often normalized to the largest peak in the spectrum) versus the mass-to-charge (*m*/*z*) ratio of the ions produced by fragmenting the components of the aerosol particle. Thus, the presence of a peak indicates the presence of one or more ions containing the *m*/*z* value indicated within the ion cloud generated upon the interaction between the particle and the desorption/ionization laser beam. The mass spectra obtained by ATOFMS have 30,000 data points in each spectrum, each of which corresponds to a different mass-to-charge ratio. The goal is to cluster these spectra to determine what common patterns of particle chemical compositions appear in the data.

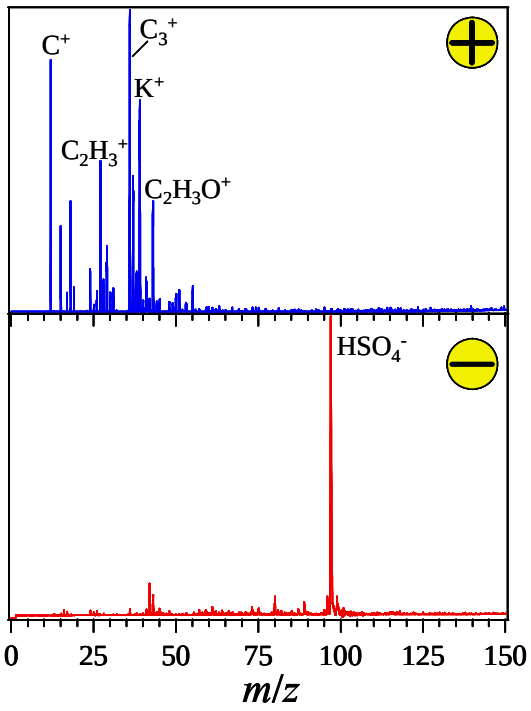

Figure 2: Calibrated single-particle time-of-flight mass spectra, showing positive (top) and negative (bottom) ions generated from the interaction of a single laser pulse with a single atmospheric aerosol particle. Major peaks are labeled with the ion composition that corresponds to the *m*/*z* value of the peak.

In order to remove some noise and to render the data more tractable, the data is preprocessed and cleaned in a number of ways. A peak detection algorithm is used to zero out data which is believed to be noise. The horizontal axis for the spectrum is calibrated and aggregated to 2000 integer *m*/*z* values in order to reduce the dimensionality of the data. Since the negative and positive spectra both represent data for a single particle, they are combined to make a virtual single spectrum for each particle that contains 4000 bins. The data for the particle can now more conveniently be thought of as a 4000 dimensional vector, each component of which corresponds to the quantity of a particular *m*/*z* value. Each spectrum is then normalized to a magnitude of 1. This is to ensure that clustering takes into account the relative magnitude of the peaks, and not the absolute. The ATOFMS might measure two different particles of different size with precisely the same chemical makeup. It is this relative makeup that we are typically interested in studying, and thus we normalize each particle. It should be noted that our dataset of high-dimensional normalized vectors now resembles those commonly found in other areas of data mining, particularly text data sets [1]. In a similar fashion to such text datasets, we measure the distance between any two spectra by the square of the Euclidean distance between them. We note that since these spectra are normalized to have a magnitude of 1, this distance is functionally equivalent for clustering purposes to using the cosine of the angle between them (which is often easily calculated as the dot product of the two vectors.)

3. ART-2A CLUSTERING

The ART-2a clustering algorithm [3] is a neural-network based approach to clustering, and is popular among its users for its flexibility. ART-2a appears to be one of the most popular clustering algorithms used by the single-particle mass spectrometry (SPMS) community for clustering data. Our particular implementation of ART-2a is based on descriptions provided in ATOFMS papers [19], which we reproduce here.

ART-2a does not require the user to specify in advance how many clusters are desired, which is different from other more well-known clustering algorithms such as K-means. Instead, the user specifies a parameter referred to as the *vigilance*. The vigilance represents the maximum distance that a point can be from its cluster center while still remaining a member of that cluster. ART-2a also requires the user to specify a *learning rate* which controls the rate at which the algorithm converges to a solution. A small learning rate results in slow convergence. A large learning rate results in wild swings as the algorithm proceeds.

The ART-2a algorithm conducts multiple iterations through the dataset. Cluster assignments for each point are not retained through the course of the algorithm; the only information retained is the centroids themselves. The first point in the dataset automatically becomes the first cluster centroid. For each successive point in the dataset, its distance to all existing centroids is determined. If the distance to all existing centroids is at least as large as the vigilance parameter, then this point is added to the set of centroids. If the distance between this point and its closest centroid is less than or equal to the vigilance parameter, then the closest centroid is shifted towards this point. The distance that the centroid shifts is determined by the learning rate: a learning rate of 1 indicates that the centroid becomes the new point itself, whereas a learning rate of 0 indicates that the centroid does not move at all. This process continues for every point in the dataset, and then the process is repeated. The number of iterations through the dataset is subject to the control of the user (and thus is technically a third parameter). At the end, each point is assigned to its closest centroid so long as the distance to that centroid is less than or equal to the vigilance value. Any points at the end with no centroids within a distance less than or equal to the vigilance value are classified as outliers. This is made precise in Algorithm 1. ART-2a is quite easy to implement, and this is one of its strengths. It has been used quite successfully in a number of ATOFMS studies [6, 19, 20].

Before addressing our concerns with the ART-2a algorithm, we will first present the K-means algorithm so that we can adequately compare the two.

Algorithm 1: ART-2a

Let S = {centroids} = $\varnothing$.
Let v = vigilance, $v > 0$.
Let α = learning rate, $0 \le \alpha \le 1$.
Let m = number of points.
Let n = number of iterations.
Let $A[j]$ = array of points, $1 \le j \le m$.
Let O = {outliers} = $\varnothing$.
For i = 1 to n
  For j = 1 to m
    if $S = \varnothing$, then $S = S \cup A[j]$.
    else
      let $s \in S$ such that dist(s,$A[j]$) $\le$ dist(t,$A[j]$) $\forall t \in S$
      let $s' = s + \alpha * (A[j]-s)$
      Replace s in S with s'.
  End
End
For j = 1 to m
  let $s \in S$ such that dist(s,$A[j]$) $\le$ dist(t,$A[j]$) $\forall t \in S$
  if dist(s,$A[j]$) > v, then $O = O \cup A[j]$.
End

4. K-MEANS CLUSTERING

K-means is likely the most well-known clustering algorithm [9-11]. It has been used in a wide variety of applications, and is the baseline to which many other clustering algorithms are compared. It has also been highly studied, and a wide variety of extensions and variations on it are known.

Both ART-2a and K-means require the user to supply certain information in advance. The information that K-means requires, however, is considerably easier for the user to handle. We will first review our implementation of the K-means algorithm, and then we will address the distinctions between it and ART-2a from a user's perspective.

K-means requires the user to specify in advance the number of clusters that should be found. This parameter thus serves a similar role as the vigilance in ART-2a. While it is true that the user typically does not know in advance how many clusters should be present in the data, a number of well-known techniques exist for determining what an appropriate number of clusters is after trying a series of possibilities [8, 14, 21].

K-means in its pure sense requires that a set of initial centroids be provided to serve as seeds for the algorithm. K-means is a local optimization algorithm, and its results are highly dependent on the choice of initial centroids. In our implementation, we use the refined initial starting points algorithm of Bradley and Fayyad [2]. This technique draws a series of samples of points from the original dataset and clusters each sample separately. The resulting centroids from clustering each of these samples are then clustered themselves to result in the starting centroids for clustering on the entire dataset. This "centroid clustering" is done repeatedly, one time for each sample, where the centroids found for each individual sample are used as starting centroids. For each of these centroid clustering attempts, a new set of centroids is produced. The set of centroids produced from this process that has the least error when compared to all other sampled centroids is the one that serves as the starting set of centroids for clustering on the entire dataset. We choose 50 samples from the dataset to use in determining refined centroids. We point out that we leave this number of samples fixed within our program, and do not change it from dataset to dataset. Within each sample, though, we still need to choose starting centroids. We do so via the heuristic technique of choosing the first point in the sample as the first centroid, then choosing each successive centroid to be the point in the dataset whose distance to its closest centroid is greatest. This heuristic is computationally slow, but runs quite fast since we use it on small samples in these examples.

Once the initial centroids have been chosen, K-means makes repeated passes through the dataset. During each pass, each point

is assigned to the cluster whose centroid is closest to that point. At the end of each pass, a new centroid for each cluster is found by averaging all of the points assigned to it. K-means is typically run until the results stabilize, i.e. until two successive iterations produce identical cluster assignments. This is made precise below.

Algorithm 2: K-means

Let k = number of clusters, $k > 0$.
Let $C[i]$ = initial centroids as described above, $1 \le i \le k$.
Let m = number of points.
Let $A[j]$ = array of points, $1 \le j \le m$.
Let $B[j]$ = cluster assignments for each point,
$1 \le j \le m$, $1 \le B[j] \le k$.
While clusters continue to change
For $j = 1$ to m
Let $B[j] = i \in \{1,...,k\}$ such that
$\text{dist}(A[j],C[i]) \le \text{dist}(A[j],C[i']) \; \forall i' \in \{1,...,k\}$
End
For i = 1 to k

$$\text{Let } C[i] = \frac{1}{\text{number of } j \text{ where } B[j] = i} \sum_{B[j]=i} A[j]$$

End
End

5. ART-2A COMPARED WITH K-MEANS

K-means plus refined starting centroids is considerably easier for practitioners to use, though ART-2a is easier to implement. There are a number of significant issues with the ART-2a algorithm that should be addressed.

ART-2a, as it has been used, requires the user to specify three parameters: vigilance, learning rate, and number of iterations. We will examine each of these in turn.

The vigilance parameter determines "how similar" points must be to pre-existing centroids in order to be assigned to a pre-existing cluster. If a point is not similar enough to pre-existing centroids, it becomes a centroid itself. The difficulty, of course, is that it is not clear for a given dataset what a good value for the vigilance parameter is. The user must experiment with a variety of vigilance values, look at the results, and intuit what the right value should be. This is quite difficult and time consuming for the practitioner who is not an expert in ART-2a as applied to SPMS datasets. In much of the SPMS work, it appears as though practitioners settle on a particular vigilance value and stick with it for a particular application. In fact, sometimes the authors of one paper [20] will start with the vigilance value provided in another paper [19]. We will provide evidence in the experimental section of this paper that the results from ART-2a are extremely sensitive to the choice of vigilance parameter, and that this practice of fixing the vigilance parameter may be dangerous. K-means has a similar parameter which controls the number of clusters that will be formed. However, many techniques exist in the data mining literature for estimating after the fact the right number of clusters [8, 14, 21]. Such techniques do not seem to be well-known for ART-2a in helping to choose vigilance. Moreover, the number of clusters is an integer parameter, and thus there are a limited number of discrete possibilities which are reasonable. Vigilance, on the other hand, is a distance value. This means that it is a positive real number, and thus it is unclear how many different vigilance parameter values must be tried, or how many significant digits are required.

The learning rate controls how dramatically ART-2a moves its centroids. A high learning rate means that the centroids move quickly, and are influenced highly by the most recent points. A low learning rate means that the centroids are quite inertial, and change very slowly over time. A low learning rate will likely cause ART-2a to proceed in a reasonably stable but slow manner towards a local optimal solution. A high learning rate may cause ART-2a to proceed more rapidly towards a solution, but it can also swing and exhibit wild behavior. It is not clear how one should choose an appropriate learning rate, and the papers that use ART-2a in SPMS contexts seem to choose the learning rate either arbitrarily or via experimentation. K-means requires no such parameter.

The number of iterations controls how long ART-2a runs in trying to stabilize the cluster centroids. This value is typically set via experimentation in the ART-2a literature [19]. Specifically, the practitioner runs as many iterations as necessary to ensure that the cluster centroids do not dramatically change. This therefore requires additional labor from the practitioner, and the correct number of iterations to run can easily change for different datasets, vigilance settings, and learning rates.

ART-2a is further complicated by the fact that the error metric (average of the squares of the distances of all points to their nearest centroids) does not converge monotonically, i.e. it is not guaranteed to decrease at each iteration. In fact, it is easily seen that with a learning rate of 1 the centroids will continue to drift from point to point without ever stabilizing. Unlike many other clustering algorithms, it is not known at any given point whether the error in the next ART-2a iteration will get worse or better. K-means, on the other hand, is theoretically guaranteed to reduce its error metric at each iteration. Therefore, K-means will continue to improve at each iteration until the cluster centroids stop moving. This frees the practitioner from needing to worry about how many iterations to run, or from trying to estimate if the changes from iteration to iteration are "small enough" in order to stop. K-means has a clear stopping point.

Finally, ART-2a does not scale well to datasets that are too large to fit into core memory. Each iteration of ART-2a requires a full scan over the dataset. Modifying ART-2a to scale reasonably is, to the best of our knowledge, an open research question. K-means differs significantly here as there are a considerable number of algorithms that adapt it, or algorithms related to it, to perform well on massive data sets [15, 17, 22].

K-means is a more user-friendly algorithm than ART-2a for the reasons presented above. When ease of use is desired, K-means should clearly be preferred over ART-2a. We now demonstrate that in addition to being easier to use, K-means is also faster than ART-2a and provides results of comparable accuracy.

6. EXPERIMENTAL RESULTS

In order to measure the effectiveness of K-means vs. ART-2a on ATOFMS data, we run three different sets of experiments. The first set of experiments is on a small set of aerosol particles that we transformed into a larger set by adding synthetic noise. This dataset allows us to cluster it with full knowledge of what the results should be. The second and third datasets are real ATOFMS

data, acquired from an atmospheric and a laboratory source, respectively.

6.1 SYNTHETIC DATASET

We generated a synthetic dataset by starting with spectra from seven actual particles. These particles represent common types of particles observed in atmospheric sampling, including three particles containing organic carbon-containing compounds, one particle containing elemental carbon, two compounds containing metal ions, and one particle that contains a mixture. Two of the particles were laboratory generated. The remaining five were sampled from ambient air: one particle was sampled from Atlanta, GA, two particles were sampled from St. Louis, MO, and the remaining two particles were sampled from Mt. Horeb, WI.

Based on these seven particles, we created 2000 artificial particles by adding noise to the spectra of the seven particles. All random noise that we generated to add to the spectra was drawn from a Gaussian distribution with mean zero and standard deviation σ, where σ is a parameter that we chose in advance to represent the "magnitude" of the noise. We added noise to each spectrum in three different ways to model different particle characteristics:

- Different particles sampled from the same source can have varying quantities of the same chemical substances. Therefore, for all peaks that were already present in each spectrum, we added to their areas random numbers drawn from the above Gaussian distribution.
- In actual experiments, peaks sometimes arise in completely unexpected locations due to measurement noise or other effects. We thus choose 20 random locations (drawn uniformly) in each spectrum that did not already have peaks. At each of these 20 locations, we added Gaussian noise as described above.
- Particles sampled from ambient air may occasionally have other substances within them that represent other background effects. Particles sampled from the same source could be contaminated by such substances. To reflect this, for each particle we randomly chose 8 locations from a common set of 24 predetermined locations. For each of these 8 locations, we added Gaussian noise.

For all three of the above cases, if the peak area would have become negative due to large negative amounts of noise, we set the area to zero.

We wanted to add enough noise to the dataset to make it challenging to cluster, but not so much so that the dataset ended up being completely dominated by random noise. Therefore, we generated a variety of versions of this dataset with different noise levels, and clustered it using K-means and ART-2a with some fixed parameter values just to get a rough sense of how "clusterable" the data was. We then plotted the clustering error (average distance of each point to its closest centroid), which is shown in Figure 3. The chart shows the error to increase as the noise increases, which makes sense: for a fixed number of centroids, the distance of each point from its nearest centroid will increase as noise is added to the data. Based on this chart, we chose two standard deviations from which to generate synthetic datasets: 1000 and 4000. The first dataset has only a moderate amount of noise, and is used as an example of an easily clustered dataset. A standard deviation of 4000 adds considerable noise to the dataset, while still retaining some of the pattern within.

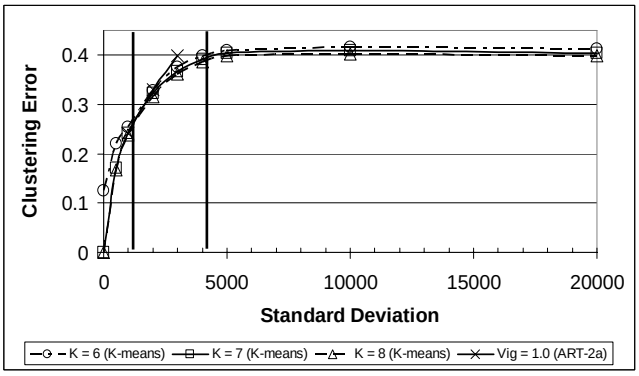

Figure 3: Clustering error vs. Gaussian noise standard deviation

We compare the output of ART-2a with K-means by measuring error defined as the average of the squares of all distances from each point to its closest centroid. For ART-2a this error metric is not guaranteed to decrease monotonically. In fact, when it reaches "stability" it often bounces back and forth between various similar solutions. Rather than specify a number of iterations in advance, we automate termination by tracking the minimum error that ART-2a manages to achieve (even if that error gets worse in later iterations), and halting if after 10 successive iterations it fails to find a new minimum error.

We set the learning rate in ART-2a according to the following heuristic:

$$\alpha = \frac{1}{\text{expected \# points per cluster}} = \frac{1}{m / |S|}$$

where m is the number of points, and $|S|$ is the expected number of clusters. In our synthetic dataset, $|S| = 7$. In general, if the user does not have a good intuitive sense of how many clusters to expect, an "arbitrary but small" learning rate (such as 0.01) can be used to run a dummy pass to get an estimate of $|S|$. This may not be particularly accurate, but might lead to a reasonable estimate for the learning rate. This heuristic is an attempt to allow each point to contribute to its centroid in roughly the same manner as it would in calculating the mean of all points in the centroid. This heuristic gets smaller as the number of expected points per cluster gets larger, which makes sense. We acknowledge that in general there is no clear way to pick a learning rate *a priori*, which is one of the significant difficulties with ART-2a.

For all of our ART-2a experiments, we left the learning rate set at 0.0035, which is the value from this heuristic for 2000 points distributed over 7 clusters. We observed that the progress that ART-2a made with this learning rate is quite slow, and thus this is likely a "conservative" learning rate that would earn us better accuracy at the expense of more iterations. We acknowledge that it is entirely possible that a different learning rate might give better results, though we point out that it took considerably more work to generate the ART-2a results than the K-means results due to choosing the vigilance parameter as described below. Even if it is theoretically possible that a better learning rate would have yielded slightly better results, it is difficult to know in advance what that learning rate should be. A practitioner using clustering

software would not wish to spend massive amounts of time experimenting to find the right learning rate.

The vigilance parameter of ART-2a was the one that we made the most effort to set carefully so that we could perform a fair comparison with K-means. Note that ART-2a looks at each point and assigns it to its closest centroid *if the distance to that closest centroid is less than or equal to the vigilance parameter*. The vigilance parameter thus strongly affects the number of clusters: a high vigilance parameter results in few clusters, whereas a low vigilance parameter results in many clusters. To compare K-means with ART-2a, then, we used the number of clusters as the common factor between them. Setting the number of clusters for K-means is trivial, since this is the parameter that K-means expects. To set ART-2a to give us a specified number of clusters, we conducted a binary search to find a vigilance setting that would give us the desired number of clusters.

We should point out that in comparing the results of our ART-2a experiments with those that may be found in other places, it should be taken into account that we are using the square of the Euclidean distance as our distance metric. Some of the ART-2a papers in this context use the cosine distance. For points of unit magnitude (which all of ours are), these two metrics result in the same clustering results. Specifically, given two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ of unit magnitude, the cosine distance between them is given as

$$x \bullet y = \sum x_i y_i$$

and the squared Euclidean distance between two such vectors is given as

$$\begin{aligned}\sum (x_i - y_i)^2 &= \sum (x_i^2 - 2x_i y_i + y_i^2) \\ &= 2 - \sum 2x_i y_i = 2(1 - x \bullet y)\end{aligned}$$

To compare a vigilance parameter from a paper that uses cosine distance with ours, one should subtract it from 1, then double it. The results turn out the same because clustering approaches that use cosine measure seek to maximize this cosine value, whereas we seek to minimize error.

Table 1 and Figure 4 show the results of these experiments. The number of nonzero decimal places in the vigilance parameter indicates how carefully we had to search to find a value that gave us the desired number of clusters. In other words, the number of digits in the vigilance parameter indicates how sensitive that region is to changes in the number of clusters compared with changes in vigilance. The results show that ART-2a and K-means provide comparable results from an accuracy perspective. From a usability perspective, however, the difference is dramatic. Setting the vigilance parameter based on heuristics or prior experiments is clearly dangerous, as the number of clusters that results can be extremely sensitive to the value chosen. In order to appropriately determine the right clustering, one should try a series of vigilance parameters and compare results. However, the varying sensitivity to vigilance makes it difficult to see how one would systematically vary the vigilance in a reasonable way except via something similar to our approach here. In that case, however, it would seem that setting the number of clusters directly (such as is done in K-means) is much simpler, faster, and more direct.

In order to characterize how much time each algorithm takes, we report three different kinds of passes. "Data passes" are the number of passes that the algorithm makes over the entire dataset, and is thus the most important measurement on which to focus. This count indicates how many times the algorithm needed to scan the entire dataset and compare each point within to all known centroids. If we were to run these same algorithms on datasets of millions of particles, the amount of time that these data passes take would completely dominate all other measurements. We show that K-means needs dramatically fewer passes over the entire dataset, which is a major advantage.

The other two measurements for K-means, i.e. "sampling passes" and "centroid passes," refer to passes made over *much smaller datasets* during the process of choosing refined starting centroids. In using the refined centroid process of Bradley and Fayyad [2], we break the dataset up into 50 samples. The number of sampling passes refers to the first stage of the refined centroid technique, where we cluster *small subsets* of the data (1/50 the size of the original). The number of centroid passes refers to the second stage of choosing refined centroids, where we cluster all of the centroids that result from the first pass. We do this clustering 50 times, each time using for the starting points the centroids that were found at each first pass iteration. Thus, the passes counted here are not for subsets of the dataset at all, but instead for a small dataset consisting of potential centroids. If we were to scale our technique to work on a much larger dataset that did not fit in core memory, we could still sample the dataset such that the refined centroids algorithm could be done in memory. We thus re-emphasize that it is the "data passes" measurement that truly indicates which algorithm would perform better on a large scale. Furthermore, it should be taken into account that there were many further ART-2a runs which were necessary that are not shown here. In order to obtain a particular number of clusters, we had to try a number of different vigilance values. These extra runs are quite time consuming, and add to the time it takes to run ART-2a if one is searching for optimal clustering.

σ = 1000	K-means				ART-2a			
# clusters	average error	# sampling passes	# centroid passes	# data passes	average error	vigilance	# data passes	# outliers
1	0.6818	150	150	4	0.6822	1.60	18	0
2	0.6171	223	233	4	0.4531	1.58	15	0
3	0.3913	200	196	4	0.3915	1.50	32	0
4	0.3397	195	230	4	0.3399	1.40	41	0
5	0.2751	198	176	4	0.2752	1.38	40	0
6	0.2612	177	316	6	0.2613	1.20	48	0
7	0.2424	168	440	6	0.2425	1.00	49	0
8	0.2373	163	506	13	0.2379	0.95	84	0
9	0.2336	156	590	14	0.2341	0.93	107	0
10	0.2311	153	545	14	0.2305	0.91	75	0
11	0.2284	151	627	16	0.2289	0.90	112	0
12	0.2250	151	721	20	*			
13	0.2227	150	772	12	0.2224	0.76	122	1

* A vigilance value with fewer than 5 decimal places could not be found.

σ = 4000	K-means				ART-2a			
# clusters	average error	# sampling passes	# centroid passes	# data passes	average error	vigilance	# data passes	# outliers
1	0.5040	150	150	4	0.5044	1.200	14	0
2	0.4437	269	259	6	0.4440	1.150	23	0
3	0.4262	269	309	6	0.4265	1.100	31	0
4	0.4153	259	403	23	0.4135	0.950	49	0
5	0.4060	232	457	19	0.4052	0.930	88	0
6	0.3974	225	560	20	0.3994	0.920	75	0
7	0.3917	205	644	27	0.3937	0.910	65	0
8	0.3863	195	665	14	0.3887	0.900	93	0
9	0.3821	192	778	14	0.3824	0.890	97	0
10	0.3776	182	856	23	0.3784	0.870	93	0
11	0.3748	173	818	19	0.3773	0.850	99	0
12	0.3714	168	872	22	0.3715	0.843	109	0
13	0.3686	161	906	27	0.3677	0.840	99	0

Table 1: K-means vs. Art-2a results for synthetic data. Average error is nearly identical for both techniques, and number of data passes for K-means is significantly fewer.

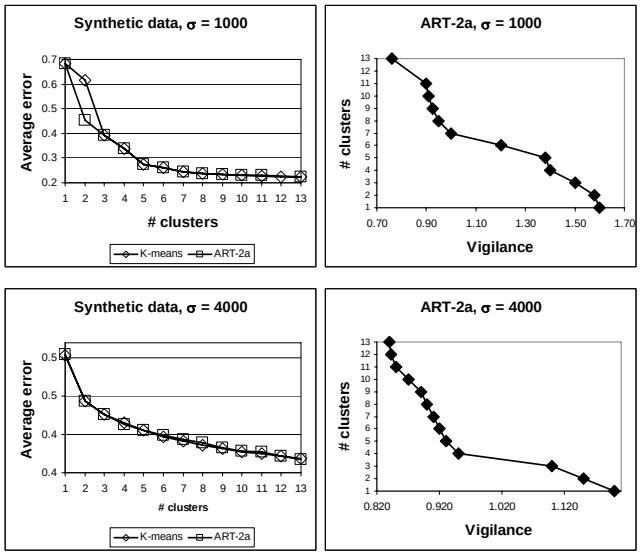

Figure 4: Synthetic data with Gaussian noise. The left plots show that the error for K-means and ART-2a are nearly identical. The right plots show that the results from ART-2a can be highly sensitive to small changes in vigilance.

In order to provide another look at the comparative accuracy of these two algorithms, we consider the *homogeneity* of the clusters. Since the "true" clusters of each particle in these synthetic datasets are known, we can measure how homogeneous each cluster is with respect to the original seven true particles. To do this, we look at which particles are assigned to each cluster centroid, at differing total numbers of clusters. At low noise levels (1000), the results from ART-2a and K-means are almost identical up to the highest number of clusters we investigated. For 9 clusters, only 4.6 % of the particles are assigned differently, and all but three clusters contain only one particle type; those that are mixed have at most 2 of 72 (2.7 %) cross assignments. There are two clusters assigned by ART-2a which contain this mixing, while there is only one assigned by K-means. As expected, at the higher noise level (4000), the results show considerably more variability. Figure 5 illustrates this graphically, showing the assignments of each of the seven particle types to each of the 9 clusters, using both algorithms. In these results, we can see that K-means and ART-2a are in remarkably good agreement in their assignments, even with noisy data forced into more clusters than inherently exist in the data. With such a high level of noise, neither algorithm could correctly cluster the data completely. Nonetheless, certain particle types (eg. OC-2, Inorganic-2, and Mixture) stay separate from the other particle types even under these conditions.

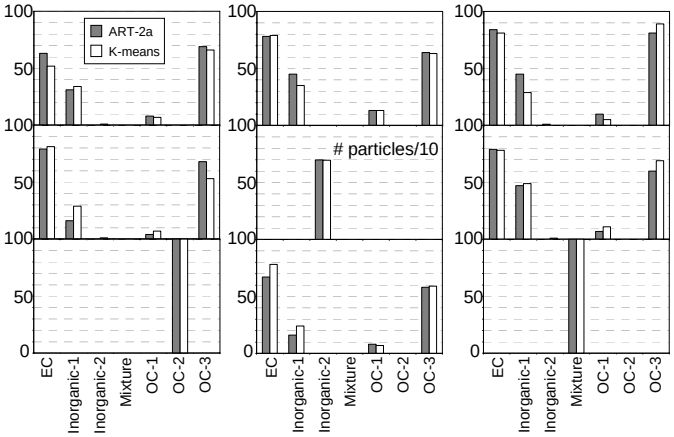

Figure 5: Cluster populations for clusters 1 through 9 when comparing ART-2a (gray bars) with K-means (white bars) for 9 clusters with noise level of 4000. The x-axis labels indicate the particle types. The middle cluster has 700 particles and has been divided by 10 to fit the same scale.

6.2 ACTUAL DATASET

In order to compare the differences in these two clustering algorithms on a real dataset, we used a dataset consisting of 2966 particles obtained in St. Louis, Missouri at the EPA SuperSite location. The data was collected during February 2004, using a TSI Model 3800 ATOFMS instrument. We will focus here on the results obtained when the data is sorted into 12 clusters. We chose 12 by looking at the clustering error vs. number of clusters (see Figure 7), and looking for the "knee" of the curve, i.e. the point at which adding further clusters produces marginal improvements in clustering error. (We acknowledge that there are significantly better quantitative methods for making this decision [8, 14, 21] that we hope to integrate in future work.) The cluster centroids that result from these two algorithms are again remarkably similar. This is illustrated by the results shown in Figure 6, which shows two of the 12 cluster centroids, graphed as overlaid centroid mass spectra, from each algorithm. The similarity is striking, with both centroids having peaks of very similar peak area at the same *m*/*z* values.

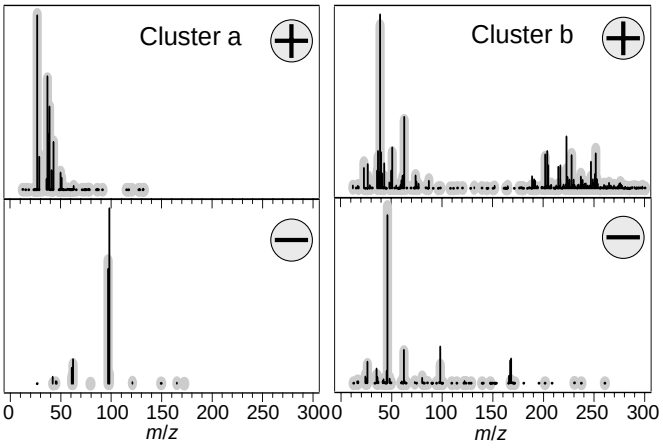

Figure 6: Cluster centers, in the form of peak area versus *m/z* for two of the 12 clusters obtained from the St. Louis data. ART-2a results are shown as thick gray bars, and K-means results are shown as thin black bars. The positive and negative ions together comprise the cluster center.

To obtain a more quantitative measurement of the similarity between the cluster centers obtained with ART-2a and K-means, we measure the distance between the two cluster centers, by calculating the sum of the squares of the difference between the peak areas of the cluster centers at each *m/z* value. The average value is $3.1x10^{-2} \pm 4.4x10^{-2}$, indicating a large spread but very small magnitude in the errors. These errors are all significantly smaller than the clustering error.

The most important aspect of these results, from the data analysis perspective, is that they make chemical sense. The experienced ATOFMS user can look at the cluster-centers and recognize the particle types represented.

We carried out similar experiments on another real data set, comprising 1812 particles sampled from a laboratory-based diesel engine [16]. This time, we focused on 8 clusters (chosen via the same heuristic mechanism as above). We obtained similar results for this analysis, with the difference between K-means and ART-2a characterized by a similar average error as we found for the St. Louis dataset ($2.6x10^{-2} \pm 4.5x10^{-2}$), showing that both algorithms perform similarly, and very well. As with the data from St. Louis, the cluster centers compare well with known particle types observed in the data.

Finally, Figure 7 and Table 2 show detailed results from these experiments. We see in Figure 7 very similar results to those seen in the synthetic datasets. One difference here is that ART-2a produces a number of outliers. Recall that when ART-2a has finished, any points where the square of the distance to the closest centroid is greater than the vigilance parameter are declared to be outliers. Since ART-2a does not consider these points to be part of any cluster at all, it is not customary to report any error for these points. However, this is in some sense unfair when comparing to K-means, since K-means assigns all points to their nearest centroid regardless of how far they are located from the centroid. ART-2a could conceivably classify *all* points as outliers and report zero error. We thus report three error metrics here for ART-2a. The first, "average error," is the average distance of each point from its closest centroid with the outliers ignored. The second metric, "best case," assumes optimistically that each outlier is just outside the range of its closest centroid. Therefore, the "best case" estimate is obtained by adding an error equal to the vigilance for each outlier. Finally, the "worst case" assumes pessimistically that each outlier is as far away as possible from its nearest centroid; hence an error of 2 is added for each outlier before averaging. (2 is an upper bound for the squared Euclidean distance between two positive normalized vectors). We see that as in the case for a synthetic set of particles, K-means has comparable error with ART-2a with many fewer data passes.

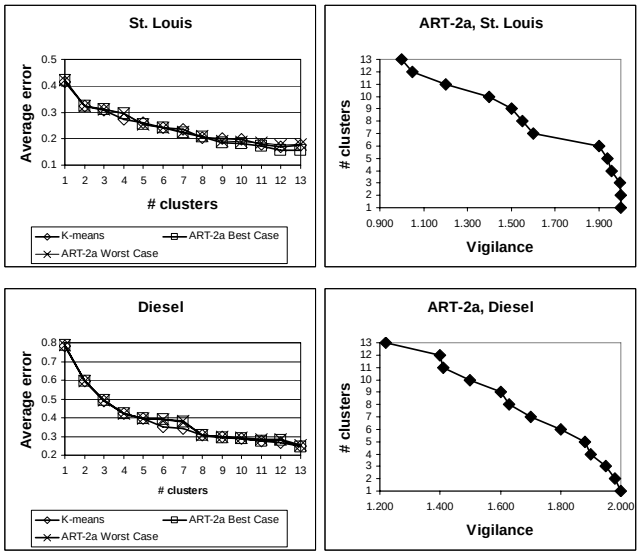

Figure 7: St. Louis and Diesel Engine data. We again see that K-means has similar error to ART-2a, and that ART-2a exhibits hypersensitivity to the vigilance parameter.

St. Louis	K-means				ART-2a					
# clusters	average error	# sampling passes	# centroid passes	# data passes	average error	vigilance	# data passes	# outliers	error with outliers best case	error with outliers worst case
1	0.4148	150	150	4	0.4207	2.000	14	0	0.4207	0.4207
2	0.3227	198	230	7	0.3223	1.999	35	1	0.3229	0.3229
3	0.3069	222	300	6	0.3095	1.995	111	1	0.3101	0.3101
4	0.2734	221	405	9	0.2943	1.960	105	4	0.2965	0.2966
5	0.2614	256	475	8	0.2501	1.940	107	5	0.2529	0.2530
6	0.2413	243	422	16	0.2370	1.900	121	7	0.2409	0.2411
7	0.2347	274	496	14	0.2076	1.600	127	30	0.2217	0.2258
8	0.2051	278	527	22	0.1915	1.550	79	31	0.2057	0.2104
9	0.1993	286	592	25	0.1738	1.500	113	28	0.1863	0.1910
10	0.1970	309	584	21	0.1684	1.400	35	31	0.1812	0.1875
11	0.1795	308	658	14	0.1575	1.200	99	44	0.1729	0.1848
12	0.1682	298	747	24	0.1419	1.050	116	52	0.1578	0.1745
13	0.1754	291	683	16	0.1417	1.000	137	57	0.1582	0.1775

Diesel	K-means				ART-2a					
# clusters	average error	# sampling passes	# centroid passes	# data passes	average error	vigilance	# data passes	# outliers	error with outliers best case	error with outliers worst case
1	0.7843	150	150	4	0.7864	2.000	15	0	0.7864	0.7864
2	0.5961	223	256	16	0.5971	1.980	27	0	0.5971	0.5971
3	0.4906	220	418	26	0.4914	1.950	68	0	0.4914	0.4914
4	0.4214	223	392	21	0.4211	1.900	85	0	0.4211	0.4211
5	0.3970	215	487	33	0.3974	1.880	244	0	0.3974	0.3974
6	0.3532	214	445	29	0.3901	1.800	126	1	0.3909	0.3910
7	0.3426	201	555	28	0.3761	1.700	160	5	0.3798	0.3806
8	0.3084	202	590	34	0.2998	1.630	140	8	0.3057	0.3073
9	0.2965	208	583	12	0.2890	1.600	190	9	0.2955	0.2975
10	0.2908	210	711	9	0.2825	1.500	118	11	0.2899	0.2929
11	0.2740	203	734	13	0.2591	1.410	206	25	0.2750	0.2831
12	0.2676	195	812	14	0.2653	1.400	43	20	0.2778	0.2844
13	0.2498	198	789	13	0.2278	1.220	107	28	0.2431	0.2552

Table 2: K-means vs. Art-2a results for St. Louis and diesel engine data. Average error is nearly identical for both techniques, and number of data passes for K-means is significantly lower. Since outliers do not contribute to ART-2a error, "best case" and "worst case" provide more accurate comparisons to K-means results.

Both of these real data sets are in fact small subsets of the data acquired in each experiment. We anticipate we will learn about

new particle types by running the clustering algorithms on the entire data sets.

7. CONCLUSIONS AND FUTURE WORK

Despite its common use for SPMS data analysis, the ART-2a algorithm is quite difficult and time consuming for the practitioner to use. It requires the use of a number of parameters that the user does not have clear guidelines on how to set, and it is difficult to determine when the algorithm has terminated. K-means, on the other hand, is considerably easier for users to manage while providing accuracies comparable with ART-2a.

We are in the process of developing an open-source software environment for atmospheric data analysis, and we thus intend to make K-means one of the primary clustering tools available within. We will be providing an ART-2a implementation as well so that users can make comparisons. Now that we have established that K-means is a stronger algorithm than ART-2a for this purpose, we will begin integrating scalable clustering algorithms into our system so that we can cluster massive datasets. We also plan to look at outlier-resistant algorithms such as K-medians to see if we can further improve clustering quality.

8. ACKNOWLEDGMENTS

This research is supported by NSF ITR grant IIS-0326328 and by Carleton College.

9. REFERENCES

[1] M. W. Berry, *Survey of Text Mining: Clustering, Classification, and Retrieval*: Springer-Verlag, 2003.

[2] P. S. Bradley and U. M. Fayyad, "Refining Initial Points for K-Means Clustering," presented at Proc. 15th International Conf. on Machine Learning, Madison, WI, 1998.

[3] G. Carpenter, S. Grossberg, and D. Rosen, "ART 2-A: An adaptive resonance algorithm for rapid category learning and recognition," *Neural Networks*, vol. 4, pp. 493-504, 1991.

[4] J. G. Dy and C. E. Brodley, "Feature Selection for Unsupervised Learning " *J. Mach. Learn. Res.* , vol. 5 pp. 845-889 2004

[5] D. P. Fergenson, M. E. Pitesky, H. J. Tobias, P. T. Steele, G. A. Czerwieniec, D. H. Russell, C. B. Lebrilla, J. M. Horn, K. R. Coffee, A. Srivastava, S. P. Pillai, M.-T. P. Shih, H. L. Hall, A. J. Ramponi, J. T. Chang, R. G. Langlois, P. L. Estacio, R. T. Hadley, M. Frank, and E. E. Gard, "Reagentless Detection and Classification of Individual Bioaerosol Particles in Seconds," *Analytical Chemistry*, vol. 76, pp. 373-378, 2004.

[6] D. P. Fergenson, X.-H. Song, Z. Ramadan, J. O. Allen, L. S. Hughes, G. R. Cass, P. K. Hopke, and K. A. Prather, "Quantification of ATOFMS Data by Multivariate Methods," *Analytical Chemistry*, vol. 73, pp. 3535-3541, 2001.

[7] S. Gaffney and P. Smyth, "Trajectory clustering with mixtures of regression models," in *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Diego, California, United States: ACM Press, 1999, pp. 63-72.

[8] A. Gordon, *Classification*. London: Chapman and Hall/CRC Press, 1999.

[9] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*: Morgan Kaufmann Publishers, 2000.

[10] D. J. Hand, H. Mannila, and P. Smyth, *Principles of Data Mining*: The MIT Press, 2001.

[11] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning*: Springer, 2001.

[12] K.-P. Hinz, M. Greweling, F. Drews, and B. Spengler, "Data Processing in On-Line Laser Mass Spectrometry of Inorganic, Organic, or Biological Airborne Particles," *Journal of the American Society for Mass Spectrometry*, vol. 10, pp. 648-660, 1999.

[13] K.-P. Hinz, R. Kaufmann, and B. Spengler, "On-line measurement and characterization of single particles using laser mass spectrometry and multivariate data analysis," *Journal of Aerosol Science*, vol. 27, pp. S171, 1996.

[14] G. W. Milligan and M. C. Cooper, "An examination of procedures for determining the number of clusters in a data set," *Psychometrika*, vol. 50, pp. 159-179, 1985.

[15] R. T. Ng and J. Han, "CLARANS: A Method for Clustering Objects for Spatial Data Mining " *IEEE Transactions on Knowledge and Data Engineering* vol. 14 pp. 1003-1016 2002

[16] S. Okada, C.-B. Kweon, J. C. Stetter, D. E. Foster, M. M. Shafer, C. G. Christensen, J. J. Schauer, A. M. Schmitt, A. M. Silverberg, and D. S. Gross, "Measurement of Trace Metal Composition in Diesel Engine Particulate and Its Potential for Determining Oil Consumption: ICPMS (Inductively Coupled Plasma Mass Spectrometer) and ATOFMS (Aerosol Time of Flight Mass Spectrometer) Measurements," presented at 2003 SAE World Congress, Detroit, MI, 2003.

[17] C. Ordonez, "Clustering binary data streams with K-means," in *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*. San Diego, California: ACM Press, 2003, pp. 12-19.

[18] D. J. Phares, K. P. Rhoads, A. S. Wexler, D. B. Kane, and M. V. Johnston, "Application of the ART-2a Algorithm to Laser Ablation Aerosol Mass Spectrometry of Particle Standards," *Analytical Chemistry*, vol. 73, pp. 2338-2344, 2001.

[19] X.-H. Song, P. K. Hopke, D. P. Fergenson, and K. A. Prather, "Classification of Single Particles Analyzed by ATOFMS Using an Artificial Neural Network, ART-2A," *Analytical Chemistry*, vol. 71, pp. 860-865, 1999.

[20] P. V. Tan, O. Malpica, G. J. Evans, S. Owega, and M. S. Fila, "Chemically-Assigned Classification of Aerosol Mass Spectra," *J Am Soc Mass Spectrom*, vol. 13, pp. 826–838, 2002.

[21] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a dataset via the Gap statistic," *Journal of the Royal Statistical Society: Series B*, vol. 63, pp. 411-423, 2001.

[22] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: an efficient data clustering method for very large databases," in *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*. Montreal, Quebec, Canada: ACM Press, 1996, pp. 103-114.