ABSTRACT
In this paper, we define a new notion for a clustering to be useful, called actionable clustering. This notion is motivated by applications across various domains such as in business, education, public policy and healthcare. We formalize this notion by adding a novel constraint to traditional unsupervised clustering. We argue that this notion is different from semi-supervised clustering, supervised clustering, weighted clustering and cannot be effectively modeled in these frameworks. We elucidate the cases when there is a feasible solution to our problem and propose a new algorithm called actionable kmeans, which is a modification of the standard kmeans algorithm to respect this constraint. We then analyze our algorithm by proving that it converges along with analysis of its time complexity. We empirically portray on synthetic and real data the superior performance of our algorithm relative to this new notion in terms of the value of the unsupervised clustering objective obtained compared with supervised and semi-supervised clustering methods adapted to this setting.

1. INTRODUCTION
Clustering is one of the most widely studied topics in machine learning and data mining [3, 14]. This is rightly so, since finding homogeneous groups in a dataset that can lead to potentially actionable insight has applications across multiple domains ranging from biology [5] to marketing [12] to manufacturing [25]. In traditional unsupervised clustering, we have a dataset that we want to partition into $k$ parts. There are multiple algorithms that optimize various objectives and output a hard [21, 18] or soft [11, 8] partitioning. It is often difficult for a practitioner to choose the appropriate algorithm that will give him a useful clustering. There has been recent work [1], where the properties of different clustering algorithms were studied with the intent of providing a guideline for the same. Although this work was insightful about the behavior of different algorithms, it is still a challenge for a practitioner to translate these desirable mathematical properties into choosing an algorithm that will provide him with a clustering that is actionable and thus useful.

Given that there is only so much one can do to ensure a useful clustering in the traditional setting, other forms have been introduced, which have access to more information. In weighted clustering, we have access to instance weights [2] in addition to the standard information, which guides the clustering in terms of which instances should be given more importance. This again may not directly lead to a useful clustering. The more popular setting is the semi-supervised clustering setting, where certain pairs of instances may have must-link (ML) or cannot-link (CL) constraints [23, 7] or the
dataset may be partially labeled [17, 6]. The ML constraint is transitive and specifies pairs of instances that must lie in the same cluster. The CL constraint is not transitive and specifies pairs of instances that must never lie in the same cluster. If such constraints are feasible [10], the final clustering is likely to have semantic value that is of use to the practitioner. If the dataset is partially labeled, then we want to obtain a clustering that minimizes the clustering objective but at the same time respects the labeling. An extreme case of this is supervised clustering [13, 15, 4], where the entire dataset is labeled. Here again, if the labeling is respected in the obtained clustering, then it is likely to be actionable. Note that the partially/fully labeled settings could be modeled as pairwise constraints and the machinery used for constraint based clustering could be used in this case too, though the number of constraints could potentially be quadratic in number of labeled examples.

New Clustering Framework: In this paper, we define a new notion for a clustering to be useful, called actionable clustering. This notion is motivated by applications across multiple industries and domains. For example, many big businesses want to cluster their spend data [9] to find out areas of high/low spend and high/low non-compliance to be able to take the appropriate action. The appropriate action may be remedial in nature or they may want to reward certain entities. Given this, it is impractical for a business to take the appropriate action at the transaction level, rather they may be able to put in place policies and processes at higher levels that respect their organizational structure such as the category (viz. marketing, human resources, IT, etc.) level. Given this they would hope that the clustering will point towards one or more categories that they ought to target. This would require a high percentage of transactions corresponding to at least one category to lie in a single cluster. If the clustering is conducted in unsupervised fashion the transactions corresponding to the different categories may be spread across the various clusters rendering the clustering useless. However, it might have been the case that for a not much worse clustering, most of the transactions in marketing would have landed in the same cluster, which would have made the clustering actionable and thus useful. Note that the actionable entity may not just be based on a single attribute such as category but could be a combination of attributes such as category and business unit. A similar need can be seen in education, where governing states might want to identify a school/schools under their jurisdiction that have been performing subpar/above expectation based on their students test scores, academic honors, athletic achievements, etc. in order to decide their funding levels. In this case too, unsupervised clustering might not provide actionable clusters, where most of the students in a particular school belong to a specific cluster, thus precluding the possibility of finding a consistent pattern at the school level and making the clustering unusable. Many such examples are seen in other domains viz. healthcare and public policy, where decisions can be made only at a certain higher level of granularity and we want our clustering to respect this fact, at the expense of obtaining a slightly worse clustering from the mathematical standpoint. From the above examples we can see that our definition for a clustering to be actionable is that there should exist at least one cluster which has a significant fraction $t$ of instances belonging to one of the groups we are interested in. In the above examples, a group may correspond to a particular category or a specific school. A visual illustration of this is seen in figure 1. If $t = 0.75$, i.e., we want some cluster to have 75% or more instances from one of the 3 groups depicted by the colors, then the right hand side clustering would be actionable since it has 3 out of the 4 blue instances belonging to a cluster. The left hand side clustering is what would be obtained for its unsupervised counterpart. We can see here that the actionable clustering may be obtained for a slight penalty. Note that our constraint does not eliminate the possibility of having multiple groups being well represented in the clustering, rather it ensures that at least one such group is represented well enough.

Why Cluster all the Data? Variants such as performing clustering independently on each group and then picking the group with the tightest (least divergence) $t$ fraction cluster, would still require applying our constraint to each group independently to make it actionable with the problem remaining NP-hard. Moreover, this would be more restrictive than our approach since, one would not be able to decipher if multiple groups can be well represented in the same cluster. This is important information as the corresponding organization can then better design and put into place the same action plan across these groups based on the knowledge that they are similar, in contrast to designing multiple action plans which can be significantly more expensive and laborious to implement. Thus, clustering all the data together subject to our constraint is preferable as it naturally leads to the insight of multiple groups being similar. Note that one can always use only the relevant data to perform actionable clustering by removing groups that are known to be unimportant operationally or possibly because they have very few transactions.

Contrast with Other Frameworks: The above exposition does not imply that the clusters should be homogeneous with respect to (w.r.t.) the groups as in supervised clustering, where the groups could be considered as proxies for class labels, but rather a large fraction of instances belonging to

Figure 2: Above we see the contrast between supervised clustering and actionable clustering.
some group should be present in some cluster. That cluster may very well have instances belonging to other groups. Moreover, a clustering which is excellent from the supervised perspective may not be feasible relative to our constraint, as each cluster may be homogeneous and contain only a single group but no cluster may contain at least \( t \) fraction of the instances from any specific group. An example of this is seen in figure 2. Given 4 clusters with \( t = 0.75 \) as before, the clustering on the left has no impurity and is an excellent clustering from the supervised clustering perspective. However, it is not actionable based on our definition, since a consistent strategy is hard to put into place at the group level, given that the instances are spread across the different clusters. In particular, the spread implies a lack of cohesion within/amongst any of the groups for a practitioner, making it difficult for him/her to qualitatively interpret the groups based on the clustering. The clustering on the right is what would be reasonable in our setting. In the cases that supervised clustering does satisfy our constraint, we might see that it is an overkill as it unnecessarily deviates the unsupervised clustering objective giving us a much worse clustering since, it strives to enforce homogeneity across all clusters. We will see evidence of this in the experimental section.

Our definition of usefulness cannot be effectively captured in the constraint based or label based semi-supervised clustering frameworks either. The reason being that we do not know which \( t \) fraction of the instances belonging to a group should be assigned to some cluster, so as to obtain an excellent clustering. We can of course randomly choose these instances and then perform semi-supervised clustering. However, we might have missed a different set of instances which if we had chosen as the \( t \) fraction, would have resulted in a much better clustering. Thus, if we knew this better set a priori, then we could model it with ML constraints or assign its instances the same label. Unfortunately, we do not and hence, the clustering algorithm ideally has to find this out. Our algorithm in section 3, performs this task and can be shown to converge. Our framework therefore requires the dynamic identification of instances from a group that will lie in the same cluster, which makes modeling in the semi-supervised framework challenging. It is also not easy to model our notion of usefulness in the weighted clustering setting. A weight is associated with a particular instance, which does not enforce any condition on groups of instances to be assigned to the same cluster.

With this new notion we do not in any way imply that it covers all possibilities for a clustering to be useful but rather that it can lead to actionable clusterings in many applications. Moreover, none of the current frameworks or algorithms can effectively model our notion, which fosters the need for the design of new techniques.

2. FRAMEWORK

We now introduce some notation so as to formally define our notion. Let \( D \) denote the dataset of size \( N \) we want to partition into \( k \) clusters \( \mathcal{C} = \{C_1,\ldots,C_k\} \). Let \( \mathcal{F}(\mathcal{C}) \) denote the unsupervised clustering objective function we want to minimize over \( D \) for a fixed \( k \). Let \( G = \{g_1,\ldots,g_m\} \) denote the partition of \( D \) into \( m \) groups. Let \(|\cdot|\) denote cardinality and \( \lceil \cdot \rceil \) denote ceiling. Given this and with \( t \in [0,1] \) we want to solve the following problem:

\[
\text{Actionable Clustering} \\
\text{argmin}_{\mathcal{C}} \mathcal{F}(\mathcal{C}) \\
\text{subject to:} \\
\exists g \in G \exists C \in \mathcal{C} \quad \text{such that} \quad \frac{|C \cap g|}{|g|} \geq t
\]

It is easy to see that as \( t \) approaches 1, the constraint potentially becomes harder to satisfy with the optimal value for the objective potentially becoming worse i.e. higher. Consequently, we might want to know under what conditions a feasible clustering exists. Let without loss of generality (w.l.o.g.) \( g_s \in G \) denote the smallest size group in \( G \), i.e., \( g_s = \arg\min_{g \in G} |g| \). With this the following lemma states that for any \( t \) and any \( k \in \{1,\ldots,N-\lceil t|g_s| \rceil + 1\} \) there always exists a feasible clustering.

**Lemma 1.** Given a dataset \( D \) of size \( N \) partitioned into \( m \) groups \( G \), then \( \forall t \in [0,1] \) and \( \forall k \in \{1,\ldots,N-\lceil t|g_s| \rceil + 1\} \), \( \exists \) a clustering \( \mathcal{C} = \{C_1,\ldots,C_k\} \) such that \( \frac{|C \cap g|}{|g|} \geq t \) for some \( C \in \mathcal{C} \) and \( g \in G \).

**Proof.** When \( k = 1 \) our constraint is automatically satisfied \( \forall t \), since the ratio \( \frac{|C \cap g|}{|g|} = 1 \forall g \in G \).

For \( k \in \{2,\ldots,N-\lceil t|g_s| \rceil + 1\} \), we can always choose \( \lceil t|g_s| \rceil \) instances from \( g_s \) and assign them to cluster \( C_1 \). Following this we can perform \( k-1 \) unsupervised clustering on the remaining instances. By construction, here again our constraint is satisfied \( \forall t \).

For \( k > N-\lceil t|g_s| \rceil + 1 \), it is impossible to create a \( k \) partition and at the same time satisfy our constraint. In practice though, we usually desire only a few clusters for a concise interpretation of our data and thus feasibility is unlikely to be an issue.

3. METHOD

In this section we first describe our method and then provide some analysis. Given that kmeans is arguable one of the most commonly used clustering techniques we in this section provide an extension of it to the actionable setting and we refer to it as actionable kmeans (Akmeans). The objective that we are minimizing is thus the sum of squares error (SSE) given by, \( \mathcal{F}(\mathcal{C}) = \sum_{i=1}^{k} \sum_{x \in C_i} ||x - \mu_i||^2 \), where \( \mu_i \) is the mean (vector) of \( C_i \) and \( ||\cdot|| \) denotes the \( l_2 \) norm.

3.1 Description

Our method takes as input the groups \( G \) and the fraction \( t \), besides the standard inputs to kmeans, \( \tau \) and \( \epsilon \) are used to specify termination conditions, where \( \tau \) is the maximum number of allowed iterations, while \( \epsilon \) is the maximum difference between the objective function at successive iterations or at below which we claim convergence. We may choose both of these conditions to indicate termination as in algorithm 1 or either of them.

The crux of the algorithm and where it differs from standard kmeans is that it has to choose \( t \) fraction of the instances belonging to a group that must lie in a cluster at each iteration. This implies that every intermediate clustering based on our algorithm in the path to convergence is also feasible. Hence, due to computational limitations or time constraints if one terminates our algorithm before
**Algorithm 1 Akmeans algorithm.**

**Input:** $D = \{x_1, ..., x_N\}$, $G$, $k$, $t$, $\tau$ and $\epsilon$ {To determine convergence we take as input the maximum number of iterations $\tau$ and the minimum reduction in objective value $\epsilon$ between successive runs.}

**Output:** $\mathcal{C}$ {Output is the feasible clustering at convergence if $k \leq N - \lceil t g_N \rceil + 1$.}

Let $\forall g \in G$, $\forall y \in \{1, ..., \lceil t g \rceil\}$

Let $N_x = \min_{g \in G} N_y$ {Finding the size of the smallest group}

if $k > N - \lceil t N_x \rceil + 1$ then

Clustering infeasible.

Return $\phi$

end if

Let $\mu_1^{(0)}, ..., \mu_k^{(0)}$ be the initial cluster centers {These could be randomly chosen $k$ different instances in $D$.}

Let $q = 1$, $r = \infty$, $D = N \times k$ matrix, $P = N \times k$ matrix {The distance and penalty matrices respectively.}

while $q \leq \tau$ and $r > \epsilon$ do

for $i = 1$ to $N$ do

$d = \infty$

for $j = 1$ to $k$ do

Compute $D_{ij} = ||x_i - \mu_j^{(q-1)}||^2$ {Compute distances.}

if $d < D_{ij}$ then

$d = D_{ij}$ {Compute distance to closest cluster.}

end if

end for

for $j = 1$ to $k$ do

Compute $P_{ij} = D_{ij} - d$ {Compute penalty of assigning instance $x_i$ to cluster $C_j$.}

end for

end for

Let $s = \infty$, $w = 0$, $u = \phi$, $c = 0$

for $j = 1$ to $k$ do

for all $g \in G$ do

$w =$ sum of the lowest $N_g$ penalties in $P_{ij}|x_i \in g$

if $s < w$ then

$s = w$

$u =$ the corresponding lowest penalty $N_g$

instances

$c = j$

end if

end for

end for

Assign instances $u$ to cluster $C_u^{(q)}$ corresponding to the mean $\mu_u^{(q-1)}$

Assign the remaining instances $D \setminus u$ to the closest cluster center based on the already computed distances in $D$

Compute the new cluster means $\mu_1^{(q)}, ..., \mu_k^{(q)}$ based on the above assignment, which corresponds to the clustering $C^{(q)} = \{C_1^{(q)}, ..., C_k^{(q)}\}$.

Compute $F(C^{(q)})$ {This is the sum of squares objective function value at iteration $q$.}

if $q > 1$ then

$r = F(C^{(q-1)}) - F(C^{(q)})$

end if

$q = q + 1$

end while

Return $C^{(q-1)}$

Convergence setting $\tau$ to a low value, we still will obtain a feasible clustering but of lower quality.

As in standard kmeans, in this case too we compute a $N \times k$ distance matrix $D$, which stores the squared Euclidean distances between instances and the current cluster centers. However, for Akmeans we compute another $N \times k$ matrix called the penalty matrix $P$, which computes the penalty of assigning an instance to a specific cluster. The penalty of assigning an instance $x_i$ to cluster $C_j$ is given by,

$$P_{ij} = D_{ij} - \min_{s \in \{1, ..., k\}} D_{is} \quad (2)$$

Consequently, if $\mu_j$ the cluster center of $C_j$ is the closest cluster center to $x_i$, then $P_{ij} = 0$. It is greater than zero for farther away clusters. Thus, $P_{ij}$, $\forall j \in \{1, ..., k\}$ is the excess amount that would be added to the clustering objective if $x_i$ is assigned to $C_j$ rather than to its closest cluster during the current assignment step. In our algorithm, we try to choose $t$ fraction of the instances belonging to a particular group along with their assignment to a specific cluster such that the sum of their penalties is minimum. Hence, for each group and cluster we select $t$ fraction of the instances belonging to that group with the lowest penalties and compute their sum. The set of instances with the lowest sum is assigned to the corresponding cluster. The remaining instances are assigned as in standard kmeans to the closest cluster. We then compute the means of these new clusters and iterate through the above two steps until one of the termination conditions is reached.

At each iteration, it is better to choose the $t$ fraction of the instances based on the penalty matrix than the distance matrix, since we are deviating the least from the unconstrained version relative to the attained objective value. If we were to choose the instances based on the minimum sum of the distances of $t$ fraction of the instances belonging to a group, then we may not achieve reduction in objective value to the extent possible during that iteration. A simple illustration of this is seen in figure 3. If we are to assign one of the two instances $x_1$ or $x_2$ to $\mu_2$, then based on squared distances we would assign $x_1$ to $\mu_2$ since $x_1$ is closer to $\mu_2$ than $x_2$ is to $\mu_2$. With this assignment the objective value has increased by $15 - 2 = 13$ over the objective based on unsupervised
3.2 Analysis

In this subsection we first show that our method converges. We then analyze the time complexity of each iteration.

3.2.1 Convergence

To prove convergence we show that the objective value monotonically decreases with each iteration. This is sufficient to prove convergence since there exist only a finite number of partitions with the objective being lower bounded by zero.

**Lemma 2.** The sequence $F(C^{(1)}), F(C^{(2)}), ...$ in algorithm 1 is monotonically decreasing.

**Proof.** At any iteration $q + 1$, $F(C^{(q)})$ is the objective value at the end of the previous iteration. $\mu^{(q)}_1, ..., \mu^{(q)}_k$ and $C^{(q)} = \{C_1^{(q)}, ..., C_k^{(q)}\}$ are the corresponding cluster centers and the clusters respectively.

Now we know that $C^{(q)}$ is a feasible clustering based on the discussion in the previous subsection. Let w.l.o.g. $C_1^{(q)}$ be the cluster which has at least $t$ fraction of the instances belonging to some group $g$. Now let us keep these $t$ fraction of the instances in cluster $C_1^{(q)}$ and reassign the remaining instances to the closest cluster center based on $\mu^{(q)}_1, ..., \mu^{(q)}_k$. Let us denote this new clustering by $C^{(q+1)}_1$ and the objective which computes the sum of squared errors based on the means in iteration $q$ by $F_q(.)^2$. In this new clustering since we have reassigned the remaining instances to the closest center,

$$F_q(C^{(q+1)}_1) \leq F(C^{(q)}) \quad (3)$$

Now our algorithm allows more freedom than what we did to obtain $C^{(q+1)}_1$. We choose the $t$ fraction from any group not restricted to the same cluster with the lowest sum of penalties relative to a cluster center. We then assign these instances to the corresponding cluster. The remaining instances are assigned to the closest cluster and thus have penalty equal to zero. With this, if we denote the unrestricted assignment of instances based on supervised (or standard) kmeans during the current iteration by $c^{(q+1)}_u$ and the assignment based on our algorithm by $C^{(q+1)}_u$, then

$$F_q(C^{(q+1)}_u) - F_q(c^{(q+1)}_u) \leq F_q(c^{(q+1)}_u) - F_q(c^{(q+1)}_u) \quad (4)$$

From equations 3 and 4 we have,

$$F_q(C^{(q+1)}_u) \leq F_q(c^{(q+1)}_u)$$

$$F_q(C^{(q+1)}_u) \leq F_q(c^{(q+1)}_u)$$

After reassignment, we compute the cluster means, which minimize the sum of squared errors and thus,

$$F_q(C^{(q+1)}_u) \leq F(C^{(q)}) \quad (5)$$

From equations 5 and 6 we have,

$$F(C^{(q+1)}_u) \leq F_q(c^{(q+1)}_u) \quad (6)$$

$$F(C^{(q+1)}_u) \leq F_q(C^{(q)}) \quad (7)$$

3.2.2 Time Complexity

Now that we have shown that our algorithm converges, let us analyze its time complexity per iteration. Standard kmeans requires computation of distances between every instance and every cluster center at each iteration. Consequently, if the dimensionality of the space is $d$, the time complexity per iteration is $O(Nkd)$.

Our algorithm also requires computation of the distance matrix, but the additional cost comes in determining the $t$ fraction and the cluster it should be assigned to. For every group and cluster we have to find the sum of penalties of $t$ fraction of the instances belonging to the group with lowest penalties. To find this lowest penalty set we can sort the penalties in ascending order. If $\alpha_G = \max_{x \in C} [g]$, and since there are $mk$ group-cluster combinations the time complexity of our algorithm (per iteration) would be $O(mkn_{G\text{log}\alpha^G})$. Thus, we have an extra $\log\alpha^G$ factor because of the sorting. If one has access to multiple processors one could also parallelize the sorting of penalties for each group and cluster, which should make the procedure more efficient.

4. EXPERIMENTS

In this section we empirically compare Akmeans with 5 other state-of-the-art methods on real and synthetic data. We showcase the quality of the clustering for each of the methods by reporting the SSE, since most of them are kmeans extensions. We observed similar qualitative results with other measures such as Silhouette and Davies-Bouldin index.

For the other methods, since we cannot implicitly enforce satisfaction of our constraint for each initialization, we report results averaged over only the runs that resulted in feasible clusterings. Thus, the SSE considered for each of the methods is only over clusterings that satisfied our constraint and consequently we would prefer algorithms that provide tight clusters with low SSE. We also depict the performance of unsupervised kmeans to show that our method gives the same high quality clusterings as its unsupervised counterpart when the constraint is trivially satisfied by it. The 5 methods we compare against are a mix of supervised and semi-supervised clustering methods. Amongst the supervised methods we compare against supervised kmeans (Skmeans) [4] and SVM based supervised iterative kmeans (SSIkmeans) [16]. For Skmeans we set a high weight for the group number attribute so that it satisfies our constraint even for high values of $t$. We implement SSIkmeans installing the python interfaces [15] to SVM-light [19]. We use the iterative variant rather than the spectral variant, since it is underconstrained.
for supervised clustering and hence should yield better quality clusterings when our constraint is satisfied.

For the semi-supervised methods we consider the following 3 approaches namely, constrained kmeans (COP-kmeans) [23], constrained spectral clustering (CSC) [20, 24], and semi-supervised learning based on Gaussian fields and harmonic functions (GFHF) [26]. The first two are constraint based semi-supervised clustering approaches, while the last is a label based semi-supervised approach. For the constraint based approaches we randomly select $t$ fraction of the instances belonging to some group, which is also randomly chosen, and then assign ML constraints to them. COP-kmeans incorporates these ML constraints into the kmeans objective. CSC on the other hand, modifies the graph affinity matrix based on these ML constraints and then performs spectral clustering on the modified graph. For the label based approach we again randomly choose $t$ fraction of the instances belonging to some group $g$ but here we assign the same label, which may be the group number, to the corresponding instances rather than adding constraints. We also randomly choose a small fraction ($\approx 5\%$) of the instances from $k - 1$ other groups, i.e. excluding $g$, where each small fraction belongs to a different group and hence has a different label$^3$. Such an initialization will result in GFHF outputting a $k$ partition. For all the 3 approaches, we average the results over multiple (100) such randomizations.

For the graph-based approaches (viz. CSC and GFHF), we constructed the graphs using radial basis kernel after standardizing the data. For kmeans and its variants, which require initial cluster centers, we randomly choose them such that they all belong to different groups. Thus, when $k = m$ we have exactly one instance randomly chosen from each group to be a cluster center. When $k < m$ we have cluster centers randomly chosen from $k$ different groups. We again report the results averaged over multiple (100) such initializations. For each of these methods we set $\epsilon = 10^{-7}$ to detect convergence.

### 4.1 Real Data Experiments

In this subsection, we evaluate the performance of the various methods on a proprietary Spend dataset obtained from a large corporation and 6 UCI datasets. In the case of the Spend dataset we also describe the insight gained and the experts feedback on the actions that are likely to be taken based on our analysis.

#### 4.1.1 Spend dataset

The Spend dataset contains a couple of years worth of (spend) transactions spread across various categories belonging to a large corporation. There are 145963 transactions which are indicative of the companies expenditure in this time frame. The dataset has 13 attributes namely: requestor name, cost center name, description code, category, vendor name, business unit name, region, purchase order type, addressable, spend type, compliant, invoice spend amount. Given this the goal is to identify certain spending and/or non-compliant tendencies amongst one or more of the 25 categories. With this information the company would be able to put in place appropriate policies and practices for the identified categories that could lead to potentially huge savings in the future.

We thus have 25 groups in our dataset. We with the help of experts decided that at least 80% of the transactions belonging to a category should exhibit the tendency or pattern for them to consider taking any action. Consequently, we set $t = 0.8$. The results from the clustering of the different methods for multiple values of $k$ are seen in figure 4. Here again, the confidence intervals based on the randomizations described above are provided in table 1. For SSIkmeans we randomly choose $k$ groups and train the model on this data. We then apply the model to the whole dataset to obtain

**Table 1:** The above table shows half the width of the 95% confidence interval (based on the randomizations) for the different methods and for different values of $k$ around the corresponding means w.r.t. the Spend dataset.

<table>
<thead>
<tr>
<th>$k$</th>
<th>kmeans</th>
<th>Akmeans</th>
<th>Skmeans</th>
<th>SSIkmeans</th>
<th>COPkmeans</th>
<th>CSC</th>
<th>GFHF</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>513.5</td>
<td>513.5</td>
<td>482.3</td>
<td>6492.5</td>
<td>22199.8</td>
<td>16595.7</td>
<td>37903.4</td>
</tr>
<tr>
<td>3</td>
<td>999.2</td>
<td>1571.5</td>
<td>5973.5</td>
<td>19225.3</td>
<td>45340.9</td>
<td>16112.2</td>
<td>45340.9</td>
</tr>
<tr>
<td>4</td>
<td>1178.7</td>
<td>3867.8</td>
<td>6173.8</td>
<td>23943.1</td>
<td>17581.5</td>
<td>17581.5</td>
<td>48660.2</td>
</tr>
<tr>
<td>5</td>
<td>1002</td>
<td>2148.9</td>
<td>5292.6</td>
<td>23943.1</td>
<td>26589</td>
<td>26589</td>
<td>50149.1</td>
</tr>
<tr>
<td>6</td>
<td>520.5</td>
<td>1574.9</td>
<td>5382.3</td>
<td>25700</td>
<td>25362.8</td>
<td>25362.8</td>
<td>50216.9</td>
</tr>
<tr>
<td>7</td>
<td>420.7</td>
<td>2177.1</td>
<td>3756.7</td>
<td>25234.7</td>
<td>11883.6</td>
<td>11883.6</td>
<td>50216.9</td>
</tr>
</tbody>
</table>

[3]In all our experiments $k \leq m$, which allows us to appropriately initialize this method. Moreover, usually in practice, we could have many groups but we want only a few clusters so that the results are interpretable and hence, potentially insightful.

![Figure 4](image-url)
a $k$ clustering. We average the results over 100 such randomizations. We need to perform the above procedure for SSIKmeans, since it implicitly assumes that the number of clusters is equal to the number of different groups that it is trained on.

We see from the figure that the unsupervised objective, which does not satisfy our constraint for $k > 2$, flattens out more or less at $k = 5$. This implies that there are probably 5 clusters in the dataset. We observe that Akmeans, which satisfies our constraint, is the closest in performance to unsupervised clustering at this and other values. It is in fact significantly better than its (adapted) competitors.

In the Akmeans clustering at $k = 5$, we observed that the constraint was satisfied for the category marketing. The corresponding transactions in marketing were characterized by high spend with most of them being non-compliant. This insight can be very useful for a company as it can now focus its efforts on a particular category rather than spreading itself too thin by expending effort across multiple categories. In fact, based on a review of these results with experts they acknowledged that this was indeed insightful and could lead to all or some of the following actions:

- Stricter monitoring of travel expenditure of employees in marketing.
- Tighter controls and extra approvals for marketing campaigns, advertisements that have expenditures greater than a certain amount.
- Close monitoring of spend with certain vendors.

### 4.1.2 UCI datasets

We now evaluate the methods on 6 UCI datasets used in previous clustering studies [24] namely: a) Glass, b) Heart, c) Ionosphere, d) Breast Cancer, e) Iris and f) Wine. The Glass dataset, the Heart dataset, the Ionosphere dataset, the Breast Cancer dataset, the Iris dataset and the Wine dataset are partitioned into 6 groups, 2 groups, 2 groups, 2 groups, 3 groups and 3 groups respectively, as indicated by their ground truth labels.

The performance of the various methods on these datasets is seen in figure 5. $k$ is set to the number of groups and we vary $t$ in each case. Given the space constraints, we plot the mean and the confidence intervals in the figures themselves. However, to keep the exposition clear we depict the results using bar charts for low, medium and high values of $t$. In particular, we compare the different methods for $t = (0.2, 0.5, 0.8)$.

We see from the figure that across all the datasets Akmeans matches the performance of kmeans for low and medium values of $t$ when kmeans satisfies our constraint. This again reaffirms the fact that our method can provide the same quality clustering as standard kmeans when our constraint is trivially satisfied. The other methods have consistently higher (mean) error and in some cases even higher variance than our method.

For high values of $t$, where kmeans does not satisfy our constraint such as on Heart, Ionosphere and Wine, Akmeans is only incrementally worse than kmeans, though it provides a feasible clustering. In this case too, when kmeans does provide a feasible clustering our method outputs the same quality clustering. The other methods are much worse in most cases with again higher error and in some cases higher variance.

Amongst the other methods COPkmeans seems to be performing the best overall with lower error and moderate variance in many cases. However, its higher error and in many cases higher variance relative to our method is again because of its sensitivity to the chosen $t$ fraction that it must assign to the same cluster. This gap is much lesser on the Iris dataset, since the groups are relatively well separated with not much overlap. The sensitivity issue is also prevalent in CSC for the same reasons. For GFHF besides the sensitivity to the $t$ fraction its performance is also affected by the instances we choose from the other groups to initialize it. This is the reason for its excessively high variance in multiple cases.

The supervised methods perform much worse than our method in general, since they strive to cluster all instances in a manner that is consistent with the groups they belong to. As we can see, this procedure turns out to be excessively demanding leading to lower quality (i.e. less cohesive) feasible clusterings.

### 4.2 Synthetic Experiments

In this subsection we want to see if our method gives the same quality clustering as its unsupervised counterpart when our constraint is automatically satisfied by this counterpart. In essence, we want to show that our method behaves like unsupervised kmeans when the constraint is triv-
Figure 5: Above we see the performance (mean +- 95% confidence interval) of the various methods on 6 UCI datasets namely: a) Glass, b) Heart, c) Ionosphere, d) Breast Cancer, e) Iris and f) Wine for 3 different (low, medium, high) values of $t$. The bars for which we do not see any confidence interval correspond to runs that have zero or insignificant variance.
Table 2: The above table shows half the width of the 95% confidence interval for the different methods and for different values of $t$ around the corresponding means w.r.t. the synthetic data.

<table>
<thead>
<tr>
<th>$t$</th>
<th>kmeans</th>
<th>Akmeans</th>
<th>Skmeans</th>
<th>SSLkmeans</th>
<th>COPkmeans</th>
<th>CSC</th>
<th>GFHP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.2</td>
<td>0.1</td>
<td>877.2</td>
<td>2619.1</td>
<td>5917.9</td>
</tr>
<tr>
<td>0.2</td>
<td>0.1</td>
<td>0.1</td>
<td>0.2</td>
<td>0.1</td>
<td>683</td>
<td>3301.8</td>
<td>26937.6</td>
</tr>
<tr>
<td>0.3</td>
<td>0.1</td>
<td>0.1</td>
<td>0.2</td>
<td>0.1</td>
<td>950.5</td>
<td>4683</td>
<td>26497.9</td>
</tr>
<tr>
<td>0.4</td>
<td>0.1</td>
<td>0.1</td>
<td>0.2</td>
<td>0.1</td>
<td>2948.1</td>
<td>4372.2</td>
<td>25979.1</td>
</tr>
<tr>
<td>0.5</td>
<td>0.1</td>
<td>0.1</td>
<td>0.2</td>
<td>0.1</td>
<td>3899.1</td>
<td>4878.4</td>
<td>25875.5</td>
</tr>
<tr>
<td>0.6</td>
<td>0.1</td>
<td>0.1</td>
<td>0.2</td>
<td>0.1</td>
<td>5694.9</td>
<td>6464</td>
<td>25255.3</td>
</tr>
<tr>
<td>0.7</td>
<td>0.1</td>
<td>0.1</td>
<td>0.2</td>
<td>0.1</td>
<td>4392.4</td>
<td>7342.7</td>
<td>25809.8</td>
</tr>
<tr>
<td>0.8</td>
<td>0.1</td>
<td>0.1</td>
<td>0.2</td>
<td>0.1</td>
<td>4683.4</td>
<td>7148.5</td>
<td>25208</td>
</tr>
<tr>
<td>0.9</td>
<td>0.1</td>
<td>0.1</td>
<td>0.2</td>
<td>0.1</td>
<td>6918.4</td>
<td>9061.2</td>
<td>25224.4</td>
</tr>
</tbody>
</table>

5. DISCUSSION

In this paper, we have introduced a new notion for producing useful clusterings, called actionable clustering. We have done this by applying a novel constraint to the standard unsupervised clustering objective that ensures that a significant fraction of the instances belonging to at least one of the predefined groups lie in one cluster. Qualitatively speaking, a clustering is actionable in our setting, if it showcases homogeneity w.r.t. the attribute values of one or more of the groups rather than exhibiting homogeneity w.r.t. the clusters based on the group numbers as is the case with supervised clustering. We have argued that this notion cannot be effectively captured not just in the supervised clustering framework but also is a challenge to model in the semi-supervised and weighted clustering frameworks. Given this, we have proposed a novel algorithm called Akmeans, which outputs a feasible clustering that is consistent with our notion. We have proved convergence for this algorithm and have analyzed its time complexity. In the experimental section, we have seen strong evidence of our method outputting good quality clusterings, where it achieves the unsupervised clustering objective when the constraint is trivially satisfied, while being significantly better than the state-of-the-art methods adapted to this setting. We have seen this across a range of values for $t$ on the synthetic as well as real UCI datasets. We have also seen a similar behavior for the client Spend dataset for different values of $k$. Moreover, in all cases our method converges in less than 20 iterations.

The most natural extension of our work would be to implement soft versions of our algorithm such as actionable fuzzy $C$-means or actionable expectation maximization based clustering approaches. The more interesting extension would be to tighten the constraint to say that at least $r$ of the $m$ groups have a fraction $\geq t$ of their instances belonging to some clusters in the clustering. This may provide more actionable clusters for $r > 1$, provided the optimal clustering objective value indicative of its quality does not considerably degrade. Our Akmeans algorithm should be easily extensible to satisfy this constraint where rather than choosing the lowest sum $t$ fraction group-cluster combination, we choose $r$ of the lowest group-clusters. Since, even for finding the lowest sum $t$ fraction group-cluster we have to find the lowest sum for all group-cluster combinations, the time complexity in this more constrained case is the same. Moreover, our extended algorithm will still converge. In this case too feasibility won’t be an issue for $k \leq N - \sum_{i=1}^{m} \lceil t|g_i| \rceil + r$, where w.l.o.g. $\{g_1, \ldots, g_r\}$ are the smallest $r$ groups in $G$.
as one can always choose an arbitrary \( t \) fraction of the instances from each of these \( r \) groups and assign them to a single cluster, followed by performing \( k \) unsupervised clustering on the remaining instances. Moreover, homogeneity w.r.t. the groups is still not enforced on the clustering and so it is still different than supervised clustering.

Another extension could be to have different values of \( t \) associated with each of the groups. This could happen if the different groups have different criticality levels and thus a smaller fraction clustered together in one group may have the different groups have different criticality levels and thus so it is still different than supervised clustering.

To obtain a feasible clustering, \( k \) would have to be
\[
\leq N - \left[ t_s \cdot |g_s| \right] + 1, \quad \text{where } t_s \cdot |g_s| = \min_{g \in G} t_g. \quad \text{This is the case, since at least } t_s \cdot |g_s| \text{ instances have to lie in a single cluster to satisfy our constraints. Again for the same reasons as before, feasibility is unlikely to be an issue in practice, as the desired } k \text{ would most likely be much smaller than the right side of the above inequality.}
\]

Acknowledgement

We would like to sincerely thank Joydeep Ghosh, Kiri Wagstaff, Margareta Ackerman and Charu Aggarwal for helpful discussions and suggestions.

6. REFERENCES