Active Learning for Clustering Bundled Data*

Eric Friedman†

Abstract

Bundling is an important tool in marketing and economics. In this paper we consider the relationship between bundling and clustering. In particular, we consider the problem of clustering customers based on valuations of products where their valuations may only be known for bundles of goods and not item by item. This can arise from sales data of bundled goods. It can also arise in survey data when the number of goods is so large that eliciting item by item valuations is impractical, which commonly arises in e-commerce.

We consider a modification of the well known k-means clustering algorithm to bundled data and examine its efficiency. We show that this new algorithm, the bundled k-means (BKM) algorithm is relatively fast and robust.

Lastly, we consider the design of bundles that facilitate clustering. We show that one can significantly increase the quality of the clustering obtained using an easy to implement algorithm. This provides a new area of application of active learning to unsupervised learning whereas most previous work in active learning has applied to classification.

Keywords: bundling, active learning, clustering, k-means algorithm.

1 Introduction

Bundling is an important tool in economics and marketing [Sch84, AY76]. The basic idea is that one can increase profits by selling bundles of goods. The classic example of this comes from selling 2 goods (1 and 2), with 0 marginal cost, to 2 classes of customers (A and B). Customer of class A value good 1 at $5 and good 2 at $2, while customers of class B have the opposite valuations, they value good 1 at $2 and good 2 at $5. The optimal single product pricing is to charge $5 (or $4.99) for each good in which case the profit will be $5 per customer, since each class will only buy one of the 2 products. However, by bundling the two goods and selling them as a single item one can increase profits significantly. Set the bundle price to $7 and the profits will be $7 per customer, an increase of 40%.

Bundling is extremely common and some examples include software (such as office suites) and cable television. In fact, one can view many "information goods" as bundles. For example, a magazine is a collection of articles which could be sold individually, but are often only sold as a single bundle, while access to online financial information sites can also be viewed as bundles of tools and data, of which several popular combinations are usually offered. The ubiquity of bundling has grown dramatically through electronic markets as bundling is easy and cheap to implement [SBB00, Bak01, Kau01]. In addition, many of these goods have very low marginal cost, which means that while they may be expensive to produce initially, they can be sold over and over again at minimal additional cost. Bakos and Brynjolfsson's pioneering work [BB99] explained the effectiveness of bundling in many e-commerce applications, using arguments based on the central limit theorem. The key idea is that while customers' valuations of individual goods may vary significantly, if their valuations of different goods are sufficiently independent, and there are sufficiently many goods, then the customers' values for large bundles should clump around the mean values, eliminating the heterogeneity that reduces profits.

Our goal in this paper is to initiate the study of the connections between bundling and clustering. Consider the case where n customers have valuations of m goods. If goods were sold individually, then one could use sales data to cluster the customers. However, if the goods are sold as bundles then the problem of clustering becomes more challenging as different customers may have been offered and purchased different bundles.

Alternatively one could use modern survey techniques to elicit the valuations from customers. However, if m is large it might be impractical to elicit the value of all goods from a participant. It is far more effective to elicit the values on bundles of goods, but then we have the same problem – clustering with bundled data.

In addition, this example raises an important question. What bundles should be used? This is a question similar to those raised in active learning, where one tries to elicit the most useful data.

In this paper we provide preliminary answers to these problems and questions. We consider the workhorse of clustering, the k-means algorithm [Mac67] and show how to modify it to provide a simple yet robust method for clustering with bundled data. We test this bundled k-means (BKM) algorithm on models used

[*] This work has been supported in part by NSF under grants ITR-0325453 and CDI-0835706.

[†] School of ORIE, Cornell University.

to model customer valuations and then discuss the active learning issues. We show that the BKM algorithm appears to be robust and efficient and also construct an easily implemental bundling procedures that significantly increase its effectiveness.

While we do not know of previous work on clustering with bundled data, there is a large literature modeling consumer choice behavior and the attempt to elicit customer valuations, see e.g., [McF74, RA03, KAR02, BD01, SW02].

Note that there are two important measures of cluster quality which may both be of interest in this setting. The first is the correctness of the clustering: are customers put into the correct cluster? However, in many cases the items of interest are the bundle centroids themselves because these provide a guide for the optimal prices of goods. Our analysis will consider both measures.

Lastly, we note that bundling can be viewed as a simple case of more general agglomerations of data in which only multivariate functions of many variables are available. Our work can be viewed as a simple model of the more general issues that may arise. In particular, one unjustified simplification in our analysis is the assumption that value of a bundle of goods is simply the sum of the valuations of the individual goods. In future work we hope to extend our analysis to more realistic models including the analysis of complements, substitute and size effects.

The paper is organized as follows. In Section 2 we describe our basic model and underlying probabilistic model for testing our algorithms. Then Section 3 considers the construction and design of the BKM algorithm. Section 4 provides some numerical analysis of the algorithm, while Section 5 considers the optimal choice of bundles, or Active Bundling. We conclude in Section 6 with comments and open questions.

2 K-Means Clustering and Random Utility Models

Let v^i_j be the valuation of customer $i \in I$ for good $j \in J$ where $n = |I|$ and $m = |J|$. We assume that each customer has a bundle partition of the goods, given by $B^i = (B^i_1, B^i_2, \dots, B^i_{b_i})$ with $B^i_r \bigcap B^i_{r'} = \emptyset$ for $r \neq r'$ and $\bigcup_{r=1}^{b_i} B^i_r = J$ where b_i are the number of bundles for customer i. Although each customer has a valuation for every good, we assume that the clustering algorithm only gets to see bundle values, where a subscript with a bundle will represent a sum over that bundle, i.e., $v^i_{B^i_r} = \sum_{j \in B^i_r} v^i_j$.

For example, consider the case of two customers $I = \{A, B\}$ and three goods $J = \{1, 2, 3\}$. Let $B^A_1 = \{1\}$ and $B^A_2 = \{2, 3\}$ while $B^B_1 = \{1, 2, 3\}$. This could arise in a survey in which we ask A about her valuation for the first good and also her valuation for the combination of goods 2 and 3, while we only ask B about his valuation for the combination of all 3 goods. The outcome of this survey would be values such as $v^A_{\{2,3\}}$ which is A's valuation of the bundle $\{2, 3\}$.

We assume that the baseline clustering is what the k-means algorithm would find on the full, unbundled, data.[1] We recall that this is a set of clusters of customers $S^1, S^2, \dots, S^k$ which together with the cluster centroids $\mu^1, \mu^2, \dots \mu^k$ for $\mu^s \subset I$ minimize the penalty function

$$\phi(S, \mu) = \sum_{i \in I} \sum_{j \in J} \phi^i_j(\mu^{s(i)}),$$

where $s(i)$ is the cluster to which i is assigned and

$$\phi^i_j(\mu^s) = (v^i_j - \mu^s_j)^2/2.$$

For later reference we recall that the standard k-means algorithm begins with an initial (often random) choice of centroids and then alternatively applies cluster creation and centering. The cluster creation part simply puts each customer into the cluster with the lowest penalty, which happens to be the one that is closest in the Euclidean norm. The centering part of the algorithm takes as input the current set of clusters and then finds a new centroid that minimizes the penalty function. We note that the optimal centroid is the average of all the points in the cluster.

Clearly if the algorithm converges (which is not guaranteed [Mac65]) it converges to a local minima. To find the global minimum the algorithm is typically rerun repeatedly, with randomized starts, to find the "optimal solution." Empirically, the k-means algorithm tends to converge quickly and find an optimal or near optimal clustering. More sophisticated algorithms have also been used to minimize the cost function, but we will not discuss them here and refer the interested reader to [CGTS02] for an example and the references therein.

In the following, we use a simple model of customer valuations which is a simple random utility model [Man77, BD01]. These have been widely applied [AR98, KAR02, RA03].

This model is very simple but will provide a basic test of our methods. Generate μ^s_j iid $N(0, \sigma)$, then for each customer randomly choose a cluster s and

[1] We choose the k-means algorithm as our baseline because it is so commonly used in these settings. We do not mean to imply that it is the best clustering procedure for this setting as clustering is a large and complex subject. We postpone the study of extensions to other clustering procedures, including important statistical and econometric approaches to future work.

generate v_j^i as $N(\mu_j^s, 1)$ where, for simplicity, σ is a single parameter. Note that that this is equivalent to generating v_j^i as follows:

$$v_j^i = \mu_j^{s(i)} + z_j^i$$

where $s(i)$ is chosen with uniform probability from S, $\mu_j^s \sim N(0, \sigma)$ and $z_j^i \sim N(0, 1)$ where the first term is known as the factor effect and the second as the idiosyncratic effect.

In our analysis we will measure the efficiency of k-means and BKM clustering with $k = 2$ on this model under a variety of parameters. We considered two complementary error measures.

The first is the clustering error, this is the probability that a randomly chosen customer will be put in the correct cluster. Formally, let $s(i) \in \{1, 2, \dots, k\}$ be the cluster label found for customer i and $s^*(i) \in \{1, 2, \dots, k\}$ their true cluster label. Since labels are arbitrary define the cluster error as

$$ErrC = \min_{f \in F} \sum_{i \in I} \delta(f(s(i)), s^*(i))$$

where F is the set of permutations on $\{1, 2, \dots, k\}$ and $\delta(\cdot, \cdot)$ is 1 if both arguments are the same and 0 otherwise.

The second is the cluster centroid error defined by

$$Err\mu = \min_{f \in F} ||\mu^{f(s)} - \hat{\mu}^s||_2$$

where F is the set of permutations on $\{1, 2, \dots, k\}$ and $\hat{\mu}^s$ is the true centroid, which will be defined by our generative model.

3 Penalty Functions for Bundled Data

Recall that the standard k-means algorithm uses a heuristic two stage approach to minimize the penalty function

$$\phi(S, \mu) = \sum_{i \in I} \sum_{j \in J} \phi_j^i(\mu^{s(i)}),$$

where $s(i)$ is the cluster to which i is assigned and

$$\phi_j^i(\mu^s) = (v_j^i - \mu_j^s)^2/2.$$

While we would like to minimize this function, with bundled data this is not possible since we can't compute ϕ exactly. Our approach will be to attempt to minimize a function which is close to the true penalty function using only bundled data.

To do this we note that when the data is bundled, instead of looking at penalties of the form $(v_j^i - \mu_j^s)^2$ it seems natural to estimate the values of such terms as implied by the bundles,

$$\phi_{B_r^i}^i = (v_{B_r^i}^i - \mu_{B_r^i}^s)^2/2.$$

Then we can construct the penalty function by combining these according to a weighting scheme

$$\phi^B(S, \mu) = \sum_{i \in I} \sum_{r=1}^{b_i} \alpha(|B_r^i|) \phi_{B_r^i}^i(\mu^{s(i)})$$

where $|B_r^i|$ is the number of goods in customer i's r'th bundle and $\alpha(\cdot)$ is the weighting function.

We will consider two useful weighting functions. The first is simply uniform weighting, $\alpha(k) = 1$ which is an obvious choice. A second natural choice is proportional weighting, where $\alpha(k) = 1/k$ where bundles are discounted according to their size. In addition to the intuition that larger bundles are less precise so should be down weighted, the proportional weighting scheme has one very important property.

Recall that for the unbundled case, the minimizers of the penalty function satisfy the condition that the cluster centroids are simply the means of the points in the cluster, i.e., $\mu_j^s = \sum_{i \in S^s} v_j^i / |S^s|$. The proportional weighting method guarantees that the sum of cluster centroids is correct. To see this, we compute the first order conditions, which will be useful later.

Note that

$$\frac{\partial}{\partial \mu_j^s} \phi_{B_r^i}^i = \alpha(|B_r^i|)(v_{B_r^i}^i - \mu_{B_r^i}^s)$$

if $i \in S^s$ and $j \in B_r^i$. Otherwise

$$\frac{\partial}{\partial \mu_j^s} \phi_{B_r^i}^i = 0.$$

Combining these we see that

$$\frac{\partial}{\partial \mu_j^s} \phi^i(\mu) = \alpha(|B_r^i|)(v_{B_r^i}^i - \mu_{B_r^i}^s)$$

for $i \in S^s$ and $j \in B_r^i$.

THEOREM 3.1. *Let μ^s be the minimizer of the proportionally weighted penalty function. Then*

$$\sum_{j \in J} \mu_j^s = \sum_{i \in S^s} \sum_{j \in J} v_j^i.$$

Proof: Fix s and $i \in S^s$ and sum all the first order conditions over $j \in J$. This yields the desired relation. QED

Next we note that the minimizers of the penalty functions need not be unique. We say that the set of bundles is complete if the minimizer is unique. For each bundle B^i define its centering matrix C^i to be the $m \times m$ symmetric matrix of partial derivatives $\frac{\partial}{\partial \mu_j^s} \phi^i(\mu)$ and note that we can write the penalty function as

$$\phi^i(\mu^s, v^i) = (v^i - \mu^s)^t C^i (v^i - \mu^s) \quad (*)$$

and its first order conditions as

$$C^s \mu^s = \sum_{i \in S^s} C^i v^i \quad (*)$$

where $C^s = \sum_{i \in S^s} C^i$ and μ and v^i are viewed as column vectors. From this, one can see that the optimizer is unique if C^s has full rank for each s. The existence of the optimizer follows from the convexity of the penalty function, which clearly has a finite minimum. It is interesting to note that for proportional weighting the matrix C^i is doubly stochastic and thus has the largest eigenvalue equal to 1. This is not true for the unweighted penalty function.

In the following, we will numerically solve this equation to find the optimal centroids. However, when the solution is not unique we simply choose the solution generated by the pseudoinverse of C which finds the solution with the smallest Euclidean norm.

First we note that one can use this expression to see that the optimizers will be correct in expectation.

Theorem 3.2. *Suppose that for each $i \in S^s$, v^i is drawn i.i.d. from some distribution with finite $E[v^i]$ and that the set of bundles is complete. Then $E[\mu^s] = E[v^i]$.*

Proof: Taking expectations of both sides of (*) and then summing the right hand side yields $E[C^s \mu^s] = E[C^s v^i]$ which by linearity implies that $C^s E[\mu^s] = C^s E[v^i]$ which implies the theorem by the assumed invertibility of C^s. QED

Note that under additional assumptions, if we also assume that the bundles are generated i.i.d. then we can extend the analysis to show that as the number of customers grows large, the computed centroids will approach the mean with probability 1.

It is not obvious which method will be better in practice. A first measure is whether C is singular, or more generally the condition number of C. To test this we constructed sets of random bundles and varied the number of goods, customers, and bundles. The results clearly showed that the proportional weighting method had smaller condition numbers, typically the ratio was between 0.6 and 0.8.

A second question is whether similarly sized bundles are more efficient than bundles of varying size, assuming the number of bundles is fixed. In this case we see that uniformly sized bundles have lower condition numbers where the ratio was typically between 0.8 and 1.0.

In absolute terms, for large numbers of customers, goods and bundles, the condition number seems to converge to 2.5 for uniformly sized bundles and proportional weighting while for varying sized bundles the number is about 2.9. The respective numbers for unweighted matrices are 3.2 and 4.0. Some representative values for the proportionally weighted case with uniform bundle sizes are given in Table 1.

Condition number	Average Error	m	n	b
Inf	0.98	4	4	2
4.37	0.14	4	64	2
3.65	0.03	4	1024	2
1.89E+18	0.86	64	4	8
20.96	0.37	64	64	8
9.29E+33	0.56	256	4	8
7.75E+18	0.90	256	4	32
1.94E+19	0.55	256	64	2
3.27	0.15	256	64	128
91.32	0.40	256	256	8
2.71	0.07	256	256	128
289.67	0.38	128	1024	2
4.66	0.05	128	1024	32

Table 1: Some sample condition numbers and solution accuracy for randomly generated bundles, using Proportional weighting and uniformly sized bundles. All tables in this paper were generated from 256 repetitions and m, n ranging from 4 to 1024, b ranging from 2 to $m/2$.

Next we compare the two methods in terms of solutions. We assume there is a single cluster and generate values according to our simple random utility model, discussed earlier. When we measure the distance between the true means and the ones computed from bundled data we see essentially the same results as those for the condition numbers with similar ratios. Some of this data is displayed Table 1.

Thus, we see that our method is reasonably accurate and that the proportional weighting appears to dominate the unweighted procedure. Therefore, in the following we will focus on the weighted method. Also uniform bundles seem to dominate uneven bundles, a useful observation.

Lastly, we note that condition numbers provide a simple way to approximate the accuracy of a method. An insight we will exploit in Section 5.

3.1 The BKM Algorithm

Now that we have settled on a penalty function the construction of the BKM algorithm is straightforward.

The BKM Algorithm

```
1. Choose $\mu$ at random.
2. Repeat until $S$ does not change.
    (a) For all $i \in I$ assign $i$ to the set $S^s$
        which minimizes $\phi^i(\mu^s, v^i)$.
```

```
(b) For each $s = 1 \ldots k$ compute $\mu^s$ from
    equation (*).
```

4 Numerical Evaluation

In this section we perform an empirical study of the BKM algorithm (using synthetic data). In the first part we consider the computational complexity of the algorithm and compare it to the k-means algorithm without bundled data to estimate the complexity that bundled data adds. In the second part we consider the accuracy of the BKM algorithm with respect to the standard k-means algorithm to delineate the loss of accuracy that bundles cause.

4.1 Complexity of the BKM Algorithm It is obvious that the BKM algorithm is more computationally expensive than the standard k-means algorithm. In particular, each iteration of the BKM algorithm requires the solution of an $m \times m$ linear equality. In this section, we will focus on the number of iterations in the main loop of the BKM algorithm, compared to that for the regular k-means algorithm. Note that this is only meant as a simple estimate of the extra complexity induced by the bundle structure as we do not consider more sophisticated algorithmic approaches, such as kd-trees [KMN+02] or sampling methods which estimate the clusters on a subset of the data before applying it to the full data set [BF98].

First we consider the simple case where the algorithm has a good estimate of the true cluster centroids. We will use the simple random utility model discussed in Section 2 with the centroids chosen to be the true statistical center of the model. This can be considered a proxy for a statistical sampling method as mentioned above.

Our results were encouraging. Over a wide range of parameter values (including bundle diversity) we see no significant change in the number of iterations between the BKM and k-means algorithms. The ratio between the two is typically around 1 and was between 1/2 and 2.

Next we consider the number of iterations required when centroids are chosen randomly. Again in this case the differences are quite small. Once again the ratios were typically about 1 ranging from 1/2 to 2.

Thus, at least for randomly generated bundles, the BKM appears to require a similar number of iterations as the standard k-means. Thus, it appears that there should be few computational roadblocks to efficiently implementing the BKM algorithm. The main increase of complexity comes from solving k linear relations at each iteration. However, since the matrices involved are $O(m)$ this does not pose any significant problems even for problems with thousands of goods since the number of iterations for most k-means applications are typically quite reasonable.

4.2 Accuracy of the BKM Algorithm In this section we provide a quick qualitative description of our numerical experiments.

To reduce computational overhead, since we are only interested in the accuracy of the methods, in the numerical computations we chose our initial μ using information about the distribution of the data, thus allowing fast convergence of the algorithms.

Our first observation is that the BKM algorithm is very inaccurate in estimating centroids when there is not enough information to do so because of too coarse a bundle structure. A simple measure of this lack of information is the total number of bundles divided by the number of goods, which we denote by $\gamma = nb/m$. Note that $\gamma = n$ for the standard k-means algorithm where all items are in their own bundle. When this number is small, $\gamma < 2$, the BKM does a poor job of estimating the centroids (an average of more than one standard deviation per good) while in some cases the k-means is accurate. When $\gamma \geq 8$ we see that both the BKM and k-means algorithms have comparable accuracy on estimating the centroids. Although k-means appears to be superior most of the time, there are many cases where the BKM outperforms it. The effect of bundle size distributions on these results are quite small – it appears that the total number of bundles is more important than their distribution.

Interestingly even though the BKM estimates of centroids are poor for small γ, in many of these instances it still finds nearly perfect clusters so this ratio is of less importance for finding cluster membership than estimating centroids. However, when centroid estimates are good it does an excellent job of clustering. In general when k-means clusters effectively so does the BKM; however there are a small fraction of instances where it clusters much more poorly than the k-means algorithm. We have not discovered a general rule for when these cases arise and leave it to future work for a more in depth analysis.

Nonetheless, we believe that the results are sufficiently promising to warrant continued study.

5 Active Clustering: Bundle Construction

In many situations the sets of bundles are not fixed and can be altered by the analyst. For example in an online survey one could modify the bundles during the course of the survey to try to optimize the accuracy of the results. This is a kind of active learning procedure.

A simple approach to this problem is based on the matrix equation for computing centroids. As we discussed in Section 3 an important parameter that affects the accuracy of the BKM algorithm is the condition number of the matrices C^s. Thus, it seems reasonable to choose bundles to try to minimize the condition number. To begin with we consider the simple case of a single cluster. For convenience we drop the s superscript in our notation and number the customers $1 \dots n$ in order of arrival.

Recall that after customer n we must solve $\overline{C}^n \mu = \sum_{i=1}^n C^i v^i / n$ where $\overline{C}^n = \sum_{i=1}^n C^i / n$ and note that $\overline{C}^n$ is a stochastic matrix. Our goal is to choose B^{n+1} such that $\overline{C}^{n+1}$ has a small condition number.

To begin the analysis, let λ^n be the smallest eigenvalue of $\overline{C}^n$ and x^n the respective eigenvector. Recall that the largest eigenvalue is 1 and thus the condition number of $\overline{C}^n$ is $1/|\lambda|$ so our goal is choose B^n to maximize $|\lambda^{n+1}|$. Now, we would like to apply a search algorithm to find the partition that maximizes $|\lambda^{n+1}|$ but repeatedly computing this value exactly is too computationally expensive. However, we can approximate this value using the following procedure where we compute the gradient of the eigenvalue.

Let $\lambda^{n+1} = \lambda^n + \omega/(n+1)$ and $x^{n+1} = x^n + y/(n+1)$ then

$$[\frac{n}{n+1}\overline{C}^n + \frac{1}{n+1}C^{n+1}]\frac{x^n + y}{n+1} = (\lambda^n + \frac{\omega}{n+1}(x^n + \frac{y}{n+1}))$$

which we can approximate as

$$-\overline{C}^n x^n + \overline{C}^n y + C^{n+1} x^n = \lambda^n y + \omega x^n$$

after multiplying by $n+1$ and dropping terms of $o(1)$.[2] Now we rewrite this as

$$-\omega x^n + (C^{n+1} - \overline{C}^n) x^n = (\lambda^n - \overline{C}^n) y$$

and note that the right hand side $(\lambda^n - \overline{C}^n)y$ must be orthogonal to x and if the lowest eigenvalue is unique can attain any value in the subspace orthogonal to x. Thus this constraint is equivalent to

$$(x^n)^t[-\omega x^n + (C^{n+1} - \overline{C}^n) x^n] = 0$$

which yields

$$\lambda^{n+1} = \frac{n}{n+1}\lambda^n + \frac{1}{n+1}\frac{(x^n)^t (C^{n+1}) x^n}{||x^n||^2} + o(\frac{1}{n+1}).$$

Thus, if we assume that x^n is chosen such that $||x^n|| = 1$ then we only need to compare $(x^n)^t C^{n+1} x^n$ for different choices of C^{n+1} to find the (approximate) optimal set of bundles.

For example, we could start with some initial bundle matrix C^{n+1} and then perform a local search by considering moves of items or swaps of item pairs between different bundles. A simple approach would be to apply random search using this formula to speed up the algorithm; however, we can improve on this as follows.

For simplicity, assume that $\lambda^n > 0$ and that bundle sizes are all the same and fixed, i.e., the number goods must equal a multiple of the bundle sizes. Then we could begin with the simple matrix C^{n+1} which arises from the bundles constructed in the natural order of elements. Other bundle matrices which come from the same number of fixed sized bundles are simply permutations (both row and column) of this matrix. Let P be the permutation matrix for the rows of C^{n+1} and then the new matrix after the permutation will be $P^t C^{n+1} P$. In order to compute the effect of this on the eigenvalue we need to compute $x^t P^t C^{n+1} P x$; however this is the same effect as swapping the elements of x to get $\hat{x} = Px$ and computing $\hat{x}^t C^{n+1} \hat{x}$. Now, we maximize this value.

This is accomplished in the following simple manner. Sort the goods in the order of x_j and then put the first n/b items in the first bundle, the next n/b items in the next bundle and continue this process. We call this procedure Active Bundle Selection.

Active Bundle Selection

```
1. Assume that the number of goods, $|J|$ is
   divisible by $b$.

2. Given $\overline{C}^n$ compute the eigenvector $x$ which
   has the smallest magnitude eigenvalue
   $|\lambda^n|$.

3. Find a permutation $P$ such that $Px$ is
   sorted in numerical value.

4. Let $\hat{B}^{n+1}$ be set of $b$ equal sized bundles
   where $\hat{B}_r^{n+1} = \{rb+1, rb+2, \dots, rb+b\}$.

5. Let $B^{n+1}$ be obtained from $\hat{B}^{n+1}$ after
   permuting the goods by $P$.
```

We first test the accuracy of this method by generating n bundles at random, then applying Active Bundle Selection iteratively to generate the next k bundles. Then we compute the condition number of the new matrix C^{n+k} and compare it to the condition number when the k bundles are constructed at random. In all cases we see that Active Bundle Selection is superior to random bundling. For example when $m = 8$, $n = 30$, $b = 2$ and $k = 10$, the average condition number is 15.3 for random bundles but under active bundling the condition

[2] We note that our analysis here is not rigorous but could be made so.

number is 9.3, a reduction of almost 40%. Changing to $n = 100$ the reduction is still about 20%. Similarly, for $m = 64$m $n = 100$, $b = 2$ and $k = 10$ the reduction is about 15%. Lastly, for $m = 8$, $n = 100$, $b = 2$ and $k = 100$ the reduction is about 20%. Thus, we see that active bundling has the potential to significantly improve statistical estimates.

Next we consider the effect of Active Bundle Selection on BKM clustering. Table 2 lists some representative results from the random utility model. We consider the case where an initial set of customers $(1 - \rho)n$ were given random bundles and then the remaining ρn customers were actively bundled.

$Err\mu$	$ErrC$	m	n	b	σ	ρ
0.32	0.01	4	64	2	10	0
0.32	0.01	4	64	2	10	0.25
0.31	0.01	4	64	2	10	0.5
0.29	0.01	4	64	2	10	0.75
1.22	0.00	16	64	2	10	0
0.98	0.01	16	64	2	10	0.75
0.30	0.00	16	64	8	2	0
0.27	0.00	16	64	8	2	0.75
0.65	0.22	16	256	2	1.0	0
0.63	0.23	16	256	2	1.0	0.75

Table 2: Numerical results for Active Bundle Selection. Clustering errors for 2 clusters.

As we see Active Bundle Selection can significantly increase the quality of the bundle centroids with gains of $10 - 20\%$ in many cases. This occurs uniformly when parameters are such that the clustering is reasonably accurate, but less consistently when clustering errors are large $> 25\%$.

Note that the clustering error may actually increase even though the centering error decreases. This somewhat counter-intuitive result arises because high condition numbers do not imply poor clustering. For example if all customers have the same bundles then one can still cluster effectively even though it is impossible to estimate the item by item centroids, as one can only determine the bundle centroids. However, in most cases, especially when bundling is effective, the active bundling procedure decreases both errors and often the reduction is significant.

Lastly, we comment that one might expect active bundling to perform better in practice that what we see in our data. This is because random bundles are most likely better than the nonrandom ones that arise in practice, which are constructed for a variety of strategic reasons, so active bundling could compensate for these deficiencies more rapidly than random bundles. To simulate this we started with a set of highly correlated bundles and then compared active bundling to random bundling. We see that both improve the accuracy of the centroids but active bundling significantly outperforms random bundling. Active bundling reduces the error by up to 50% over the correlated bundles and $10 - 30\%$ over the combination of correlated and random bundles. (See Table 3.)

Errμ	ErrC	Errμ	ErrC	m	n	b	σ	ρ
Act.	Act.	Rnd	Rnd					
0.039	0.000	0.039	0.000	16	64	8	10	0
0.027	0.000	0.031	0.000	16	64	8	10	.75
3.192	0.370	3.134	0.373	16	64	8	0.2	0
2.806	0.371	2.838	0.369	16	64	8	0.2	.25
2.693	0.377	2.831	0.383	16	64	8	0.2	.5

Table 3: Selected numerical results for Active Bundle Selection and Random Bundle selection with highly correlated initial Bundles.

6 Conclusions

In this paper we have highlighted an important area for the application data mining techniques focussing on the issue of bundled data, a topic that is expected to increase in importance in the coming years. We have shown how to modify the k-means algorithm to accommodate bundled data in a robust manner without significant losses in accuracy or increased computational requirements. Lastly, we showed how one could apply the paradigm of active learning to bundled clustering and presented a simple algorithm that can find nearly optimal bundles to maximize the effectiveness of our algorithm.

Aside from the practical implications of our analysis we also view this as preliminary investigations into clustering of aggregated data and the extraction of unaggregated information from that clustering. In addition we have extended the ideas from active learning to a problem of clustering which we believe is a new direction and raises many new research questions.

Clearly there are many natural extensions of this work, including a large scale robust implementation and analysis of the BKM algorithm. In addition, the assumption of additivity which is central to our analysis needs to be weakened for many problems of interest.

References

[AR98] G.M. Allenby and P.E. Rossi. Marketing models of consumer heterogeneity. *Journal of Econometrics*, 89(1-2):57–78, 1998.

[AY76] W.J. Adams and J.L. Yellen. Commodity Bundling and the Burden of Monopoly. *Quarterly Journal of Economics*, 90(3):475–498, 1976.

[Bak01] Y. Bakos. The Emerging Landscape for Retail E-Commerce. *Journal of Economic Perspectives*, 15(1):69–80, 2001.

[BB99] Y. Bakos and E. Brynjolfsson. Bundling Information Goods: Pricing, Profits, and Efficiency. *Management Science*, 45(12):1613–1630, 1999.

[BD01] G. Baltas and P. Doyle. Random utility models in marketing research: a survey. *Journal of Business Research*, 51(2):115–125, 2001.

[BF98] P.S. Bradley and U.M. Fayyad. Refining Initial Points for K-Means Clustering. In *Proc. 15th International Conf. on Machine Learning*, volume 727. Morgan Kaufmann, San Francisco, CA, 1998.

[CGTS02] M. Charikar, S. Guha, É. Tardos, and D.B. Shmoys. A Constant-Factor Approximation Algorithm for the k-Median Problem. *Journal of Computer and System Sciences*, 65(1):129–149, 2002.

[KAR02] J. Kim, G.M. Allenby, and P.E. Rossi. Modeling Consumer Demand for Variety. *Marketing Science*, 21(3):229, 2002.

[Kau01] R.J. Kauffman. Economics and Electronic Commerce: Survey and Directions for Research. *International Journal of Electronic Commerce*, 5(4):5–116, 2001.

[KMN$^+$02] T. Kanungo, D.M. Mount, N.S. Netanyahu, C.D. Piatko, R. Silverman, and A.Y. Wu. An Efficient k-Means Clustering Algorithm: Analysis and Implementation. *Ieee Transactions On Pattern Analysis And Machine Intelligence*, pages 881–892, 2002.

[Mac65] J. MacQueen. On convergence of k-means and partitions with minimum average variance. *Ann. Math. Statist*, 36:1084, 1965.

[Mac67] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[Man77] C.F. Manski. The structure of random utility models. *Theory and Decision*, 8(3):229–254, 1977.

[McF74] D. McFadden. Conditional Logit Analysis of Qualitative Choice Behavior. *Frontiers in Econometrics*, 8:105–142, 1974.

[RA03] P.E. Rossi and G.M. Allenby. Bayesian Statistics and Marketing. *Marketing Science*, 22(3):304–328, 2003.

[SBB00] M.D. Smith, J. Bailey, and E. Brynjolfsson. Understanding Digital Markets: Review and Assessment. *Understanding the Digital Economy: Data, Tools, and Research*, 2000.

[Sch84] R. Schmalensee. Gaussian Demand and Commodity Bundling. *Journal of Business*, 57(S1):211, 1984.

[SW02] Z. Sandor and M. Wedel. Profile Construction in Experimental Choice Designs for Mixed Logit Models. *Marketing Science*, 21(4):455, 2002.