

# A Survey of $K$ -Median and Related Algorithms

Xiaolei Li

February 14, 2002

## Abstract

We survey a number of  $k$ -median, facility-location, and related problems. These problems have application in numerous fields such as biotechnology, data compression, vector quantization, and many more. A variety of approximations have been given with each one getting better and faster. We will analyze a select group of them noting the methods, weaknesses, and strengths.

## 1 Introduction

### 1.1 Clustering

Clustering is the problem of grouping data points into clusters such that points in a particular cluster are similar to each other and separate clusters are dissimilar to each other. The similarity measure is problem-specific with Euclidean distance shown in Figure 1. Application of clustering occur in many situations: biotechnology uses it to group similar genes using their expression levels; city planners use it to plan public buildings to maximize their coverage; astronomers use it to group data being collected by satellites. In addition, clustering has applications in data mining, statistical data analysis, compression, vector quantization, etc.  $K$ -median, facility-location, and all the other problems discussed here are all clustering-related problems with different requirements or similarity measures.

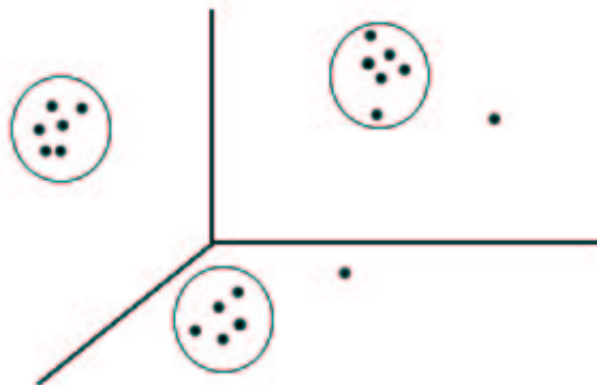


Figure 1: Clustering of data points in Euclidean space.

## 1.2 $K$ -Median

In the  $k$ -median problem, we are given  $n$  points in a metric space and a positive integer  $k$ . We want to select  $k$  *medians* such that the sum of the distances from all  $n$  points to their nearest median is minimized. In a more formal manner, the problem is to minimize  $\sum_{i=1}^n \min_{1 \leq j \leq k} d(x_i, m_j)$  where  $m_j$  are the medians and  $x_i$  are the data points. Within the  $k$ -median problem, there are two variations: the *discrete*  $k$ -median problem uses the data points as medians whereas the *continuous*  $k$ -median problem allows the medians to be chosen anywhere in  $\mathbb{R}^d$ .

Early work by Bartel [6] showed a  $\log(n)$  approximation to the problem. Lin and Vitter [19] provided a  $(1 + \epsilon)$  approximation while limiting the number of median locations to  $(1 + 1/\epsilon)(\ln(n) + 1)k$ . Furthermore, they produced another approximation of  $2(1 + \epsilon)$  while limiting the number of medians to  $(1 + 1/\epsilon)k$ . Work by Arora, Raghavan, and Rao [3] has produced a  $(1 + 1/c)$  approximation with a running time of  $O(n^{O(c+1)})$ .

## 1.3 $K$ -Means

A closely  $k$ -median related problem is the  $k$ -means problem. In this situation, we are again given a set of  $n$  data points and a positive integer  $k$ . The difference between the two is the distance measure. The goal now is to determine a set of  $k$  points, called *centers* instead of medians, such that the mean squared distance from each data point to its nearest center is minimized.

## 1.4 Facility-Location

In the *facility-location problem*, we are given a set of  $n$  data points as before *plus* a cost  $c$  for opening a facility at each of the data points. These data points are often referred to as cities. The goal is to find a set of facilities to open so as to minimize the sum of the distances from each city to its nearest facility *plus* the cost of opening the chosen facilities at the cities. There are two cases within the facility-location problem: uncapacitated facility location problems (UCFL) and capacitated facility location problems (CFL). The difference between the two is that in CFL, facilities have a limit on how many cities they may serve. Furthermore, if we set all the facility-opening costs to 0, the problem is equivalent to the  $k$ -median.

Investigation into this problem has yielded many results. For the UCFL problem, Hochbaum produced a  $O(\log(n))$  greedy algorithm. For the CFL problem, approximations of 3.16 (Shmoys, Tardos, and Aardal [23]), 2.41 (Guha and Khuller [13]), 1.861 (Mahdian, Markakis, Saberi, and Vazirani [21]), 1.74 (Chudak [11]), 1.728 (Charikar and Guha [9]), and 1.61 (Jain, Mahdian, and Saberi [15]) have been found. Several different methods were used to obtain these results so their speeds vary, sometimes greatly. 1.463 has been proven to be the best approximation[13].

## 1.5 $K$ -Center

Lastly, there is the  $k$ -center problem which has been approximated fairly well. The premise of the problem is similar to the previous ones. We are given  $n$  data points and a positive integer  $k$ . Except now, the goal is to minimize the maximum distance between a data point and its nearest center.

## 2 Techniques

The algorithms presented here share some common techniques and heuristics. In this section, we will examine the main motivations of these methods, reasons why they work, and their strengths and weaknesses.

### 2.1 Linear Programming

Linear programming (LP) is a common method of solving  $k$ -median and facility location problems. This is due to its ease of implementation. In most cases, one just needs to devise the linear program and let an existent implementation solve it. Once the optimal result is derived, they are rounded to obtain the integer solutions. In addition, the concept of linear-programming duality is used. This is based on the property that given a maximization or minimization problem, there exists a related minimization or maximization problem whose optimal value is identical to the original program. The original linear program is called the *primal* and the related program is called the *dual*. Often, people will refer to these type of algorithms as primal-dual solutions.

For the metric UCFL problem, Shmoys, Tardos, & Aardal [23] gave a 3.16-approximation. Subsequent improvements came from Guha & Khuller [13] with 2.41 and Chudak [11] with 1.74. For the  $k$ -median problems, Lin & Vitter presented an approximation with a factor of  $2(1 + 1/\epsilon)$  while using  $(1 + \epsilon)$  times the optimal number of medians and a factor of  $(1 + \epsilon)$  while using  $(1 + 1/\epsilon)(\ln(n) + 1)k$  times the optimal number of medians.

### 2.2 Heuristics

Heuristics are a common method to improve results. They can be applied at any step during the algorithm. Some are done as post-process refinement while others are done to the initial points before any computation. They either offer a step up in the approximation result or in the speed or sometimes both. In this section, we examine some of the heuristics used in the algorithms presented here.

#### 2.2.1 Greedy Algorithm

Greedy algorithms have the advantages that they are usually simple and fast. They have been successfully used in the UCFL in conjunction with the linear programming techniques described previously. It is either applied at each step of the LP-rounding program or as a single post-process step that improves the original result.

A greedy heuristic combined with LP has given an approximation guarantee of 1.61 with a running time of  $O(n^3)$  [15] for the UCFL. Previous work [21] had given a 1.861 approximation with a running time of  $O(m \log m)$  where  $m$  is the number of edges in the underlying complete bipartite graph connecting the cities and the facilities. In fact, it has been proven the best possible approximation is only 1.463 for the metric facility location problem [13].

### 2.2.2 Local Search

Local search makes the best small local changes based on a certain measure. This technique involves adding, deleting, or swapping medians. Given an initial solution, one of these three operations are performed until it is impossible to further improve or until a certain requirement is met.

For the CFL problem, the first algorithm based on this heuristic was by Korupolu, Plaxton, and Rajaraman [18]. They gave a polynomial algorithm which gave an approximation with a factor of  $(8 - \epsilon)$ . Two subsequent improvements came from Chudak, Williamson [12] and Arya, et al. [5]. Chudak & Williamson refined the original analysis to show a factor of  $6(1 - \epsilon)$ . Arya, et al. has since improved the gap to something between 3 and 4.

For the UCFL problem, Korupolu, et al. [18] gave an initial result of 5 while allowing those three operations mentioned above performed on single centers only. Charikar & Guha [9] gave another algorithm which allowed the deleting of multiple centers that gave an approximation of 3. The same result was shown later by Arya, et al. [5] while only allowing operations on single centers.

For the  $k$ -median problem, Korupolu, et al. showed a solution with  $k(1+\epsilon)$ . Arya, et al. then gave a result of 5 with single swaps and  $(3 + 2/p)$  with multiple swaps.

### 2.2.3 Refining Initial Points

This method is motivated by the weakness of iterative methods. As noted previously, iterative methods converges to a local extremum but not necessarily the global extremum. In fact, these algorithms are high sensitive to the initial data points. By refining them, we hope that the iterative algorithm will converge to a better extremum than before. This method is used mainly in real world applications where there is a high number of dimensions and an even higher number of data points. In these situations, it maybe not be possible to store all data points in memory and perform calculations on them. Thus a subsample of the data points is taken to perform the iterative calculations. If these subsamples were chosen randomly, the final result might be vastly different from the optimal. By using the refining technique, we hope to get a subsample that is best representative of the entire data set. The trick in these techniques are of course in how to choose the best representative data points.

## 2.3 Weakness of Linear Programming and Local Search

Linear programming and local search both use iterative methods in order to obtain a minimum or maximum, but it cannot differentiate between local or global extremas. In fact, it is highly likely that the algorithm will converge to a local extrema and return that as the global extrema. Of course, there are certain techniques such as Lagrangian relaxation that try to avoid this pitfall, but it is still one major weakness of these methods. In addition, iterative methods tend to run slowly. Once the algorithm is near an extrema, convergence occurs linearly.

## 2.4 Combinations

Even though the section describes these methods as though they are completely different, they are often used together. In fact, linear programming essentially is greedy in this situation. Also, these methods are sometimes combined with each other to obtain the best result; therefore, it can be hard to point to a certain algorithm and definitely say what class of methods it belongs to. For example, a certain algorithm uses a greedy algorithm as a post-process refinement of a linear program. Also, refining initial points itself is not a complete algorithm, it requires a separate algorithm to produce the final results. Lastly, the use of LP algorithms to analyze other algorithms is quite common. It is quite common to prove optimality using the dual of a certain program.

# 3 Algorithms

In this section, we shall examine some algorithms for approximating the  $k$ -median and its related problems using some of the techniques described in the previous section.

## 3.1 Arora, Ragahavan, and Rao [3]

In their approach, main concepts are borrowed from an earlier TSP paper [4] by Arora. The algorithm uses dynamic programming to build a solution of cost at most  $1 + 1/c$  times the optimum. The running time is  $O(n^{O(c+1)})$ . (This approach did not fall under the major categories described in the previous section because it is basically the only one that uses the notion of dynamic programming.) The algorithm begins by constructing a quadtree on the given points; then using the leaves of the quadtree as the basis of dynamic programming, the algorithm slowly builds up an  $m$ -light solution. An  $m$ -light solution is where all points leave and enter grids only through portals which are the 4 corners of the grid and  $m$  other equally spaced points on the edges of the grid. Also defined are  $shift(a, b)$  of quadtrees which essentially shift the entire dissection to the right by length  $a$  and up by length  $b$ .

Using the Charging Lemma, the authors show that with probability of at least  $1/2$ , an  $m$ -light solution with a random  $shift(a, b)$  will cost at most  $O(1 + O(\log L/m))$  times the optimum.  $L$  is the length of the sides of the quadtree's bounding box. With this notion, the algorithm first builds solutions in the leaf grids. Then in a bottom-up fashion, it slows

fills the parents. Each parent grid has four children and the algorithm exhaustively searches through all combinations of them for the best solution, if one even exists at all.

### 3.2 Kanungo, Mount, Netanyahu, Piatko [17]

This paper presents an implementation of Lloyd's algorithm [20]. It uses a heuristic which iteratively changes the data until it reaches a local minimum. Two major disadvantages of this approach are: local minimums do not necessarily equal global minimums and once the algorithm is sufficiently close to a minimum, convergence occurs only at a linear rate.

In this paper, a filtering algorithm is presented. It is quite easy to implement and only requires a kd-tree for the data points; however, their algorithm is data-sensitive. If data points do not naturally cluster, their algorithm performs as slow as brute-force. This is a considerable weakness, but their empirical results show that in typical situations, data points do cluster and that their algorithm performs quite well when these clusters are isolated.

The algorithm works as follows. First, a *balance box-decomposition tree* is built for the data points. A *kd-tree* is used in their implementation but in theory, a BBD-tree is required. Once this is done, the BBD-tree structure is kept constant throughout the algorithm. They note that this is the major difference between their implementation and others' of Lloyd's algorithm. At each iteration, the centroid of the set of data points for a center is computed and that center is moved to the centroid. This process is repeated until convergence occurs.

### 3.3 Lin, Vitter [19]

Published in 1992, this is one of the more influential works on median problems. It is based on earlier work of linear programming and rounding off solutions, but it was the first one to set variables with zero values in the fractional solution to 1. This departure was surprising at the time but resulted in great improvements. Through it all, they gave a  $(1+\epsilon)$  approximation of the median problem.

### 3.4 Ostrovsky, Rabani [22]

This approach has restricted the  $k$  in  $k$ -median to be fixed but allows for arbitrary dimensions. They observed that in most applications, the goal was to cluster data with high dimensions into a relatively small number of clusters. The algorithm uses neither sampling nor dynamic programming nor singular value decomposition, but rather reduces the dimension by random linear transformations. They show that using these transformations result only in low distortion if performed appropriately. Specifically, they are performed on embeddings into a Hamming cube or Hypercube depending on the dimensions involved. In addition, the metric space used in the problem can be more than just simple Euclidean: in a binary cube, the Hamming distance could be used and in  $\mathbb{R}^d$ ,  $L^1$ ,  $L^2$ , or square of  $L^2$  could be used.

### 3.5 Charikar, Guha, Tardos, Shmoys [10]

Their paper gives a  $6\frac{2}{3}$  approximation polynomial-time algorithm for the  $k$ -media problem. It relies on solving a natural linear programming relaxation of the problem and rounding the optimal fractional solution. Their approach is loosely based on the filtering technique described by Lin & Vitter except they relaxed its requirements by letting some centers pay a greater cost than the corresponding cost in the optimal fractional solution. While doing this, they still keep the average cost within a small increase.

The algorithm follows three major steps. Step one: a solution to the LP relaxation is found but new set of demands and consolidations of locations will be made. This modification is made to not increase cost but increase simplicity. Step two: fractional centers are “consolidated”. This step increases the cost by 2. Step three: the solution found in step two is converted back to a solution that satisfies the original linear program. Cost is increased by  $\frac{4}{3}$  in this step. Take the product of these three steps and the solution’s cost is increased by  $2\frac{2}{3}$  total. Another cost of 4 is factored in to convert this solution back to the original problem and a total cost of  $6\frac{2}{3}$  is achieved.

Their algorithm can also be extended without too much effort to support problems where centers have limited capacity, where centers have costs, and where the distance function satisfies a relaxed triangle inequality.

### 3.6 Jain, Vazirani [16]

They present algorithms to solve the metric uncapacitated facility location problem and the metric  $k$ -median problem. They give approximations of 3 and 6 with running times of  $O(m \log m)$  and  $O(m \log m(L + \log(n)))$  respectively where  $n$  and  $m$  are the number of vertices and edges in the underlying graph. Their work is based on using the primal-dual schema with some exceptions. One distinguishing feature is that in their primal and dual programs, negative coefficients are allowed in the constraint matrix, objective function vector and right hand side vector. In addition, a new procedure called forward include is conjured up for removing redundancies in the integral primal solution. Both these new mechanisms increase the complexity, but the authors claim they have simple solutions for them. This can be somewhat verified in their claimed running time, which is relatively lower than similar algorithms that use LP-rounding.

### 3.7 Mahdian, Markakis, Saberi, Vazirani [21]

In this greedy approach to the metric uncapacitated facility location problem, they present a 1.861 approximation with the running time of  $O(m \log m)$  where  $m$  is the total number of edges in the complete underlying bipartite graph connecting the cities and the facilities. The algorithm introduces a measure called cost-effectiveness. It indicated the effectiveness of the connection between a certain facility and a subset of the cities.

The paper presents two algorithms. They are identical in execution but different in the description. The first one works as follows: while there are still unattached cities, choose the most cost-effective link between a facility and a subset of the cities. Once a city is connected to a facility, it is taken out of consideration. Facilities can be opened multiple times but the cost is only counted once. Repeat this process until all cities are connected to some facility. The second algorithm is a bit more involved. Both of these algorithms are analyzed by using the LP restatement.

### 3.8 Guha, Khuller [13]

This is another greedy algorithm that approximates the UCFL problem. Their approach produces a 2.408 result, but perhaps the more important result of the paper is the proof that 1.463 is the best possible result for a polynomial-time algorithm unless  $NP \in DTIME[n^{O(\log \log n)}]$ .

### 3.9 Bradley, Fayyad [7]

The authors of this paper notice that in most iterative approaches to the  $K$ -Means problem, the algorithm has the weakness of converging to a local minimum that might be vastly different than the global minimum. In addition, these algorithms are particularly sensitive to the initial starting positions. With these observations in mind, they present an algorithm which uses random subsamples of the data to find the centers. It works as follows. First, random subsamples of the data are collected and centers are found for them using any conventional iterative approach. Second, using the centers just found, another set of centers is calculated. The center point in this set with the least amount of distortion is then returned as a refined initial point.

The algorithm is tailored for large amount data. Since at any time, only a small subset of the data is needed, thus the system does not require great computational power. Experimental results in the paper also show the refined data results in a much better solution than the raw data. However, no bounds are given on the quality of the result thus the algorithm is not well suited for theoretical comparisons.

### 3.10 Arya, Garg, Khandekar, Meyerson, Munagala, Pandit [5]

In this paper, a local search heuristic is presented that solves the  $k$ -median problem with a cost of 5 and the uncapacitated facility problem with a cost of 3. Furthermore, if the  $k$ -median algorithm is allowed to swap simultaneously, the cost becomes  $3 + 2/p$  where  $p$  is the number of centers being simultaneously swapped. For the  $k$ -median problem, the algorithm starts with an arbitrary solution and swaps open centers for closed centers until it is impossible to do so. The requirement for an open center to be swappable is that the cost of itself is higher than the slightly modified cost of a closed center.

### 3.11 Charikar, Guha [9]

Approximation algorithms for the UCFL and  $k$ -median problems are presented here. Much of their work is based on improving other algorithms by *greedy improvements* or *cost scaling*. They present three different results for the UCFL problem. First is a greedy local search algorithm that produces a  $2.414 + \epsilon$  approximation with a running time of  $O(n^2/\epsilon)$  for the UCFL problem. Second is an improvement of Jain and Vazirani's [16] solution using greedy improvements and cost scaling. It achieves a 1.853 result with a running time of  $O(n^3)$ . Third is a combination of the second algorithm with the best known LP-based algorithm that produces a 1.728 approximation. Lastly, they present a 4-approximation of the  $k$ -median problem with a running time of  $O(n^3)$ .

Here are the basic concepts of the greedy improvement and cost scaling techniques deployed in their work. Cost scaling basically scales a problem to a different size that is easier than the original, solve the new, resized problem, and then scale back to get the answer for the original problem. The greedy improvement tries to lower the total of service cost and facility cost when either one is extremely high.

### 3.12 Jain, Mahdian, Saberi [15]

This paper presented greedy algorithms that achieved a 1.61 approximation for the UCFL problem with a running time of  $O(n^3)$ , a 4 approximation for the  $k$ -median problem, and a 3 approximation for the CFL problem. The UCFL algorithm uses the concept of time as introduced in [16]. As times goes on, the algorithm examines *offers* being made from cities to facilities and open facilities when the total offer amount is equal to the cost of opening the facility. While there are still unconnected cities, they keep increasing their offers to facilities until they are connected. As for their approximations of the  $k$ -median and CFL problems, they combined the use of Lagrangian relaxation and a technique by Jain and Vazirani [16].

## 4 Summary Table

Several of these works can be extended to solve other problems without too much work; however, in this table, we will only show the main problems they were designed to solve. In addition, the entries in the *Method(s)* column indicate only the distinguishing feature presented in the algorithm. Several of them use more than one.

Authors	Method(s)	Problem(s)	Result	Running Time
Aardal, et. al.[1]	LP	$k$ -level UCFL	3	
Arora, et. al.[3]	Dynamic	$k$ -median	$1 + 1/c$	$n^{O(c+1)}$
Arya, et. al.[5]	Local Search	$k$ -median & UCFL	5 & 3	
Bradley, Fayyad[7]	Refining Init. Pts.	$k$ -means		
Charikar, Guha[9]	Greedy	UCFL & $k$ -median	2.414 & 4	$O(n^2/\epsilon) \& n^3$
Chudak[11]	LP	UCFL	1.736	
Chudak, Williamson[12]	Local Search	CFL	$6(1 + \epsilon)$	
Guha & Khuller[13]	Greedy	UCFL	2.408	
Jain, et. al.[15]	Greedy	UCFL	1.61	$n^3$
Jain & Vazirani[16]	LP	UCFL & $k$ -median	3 & 6	$m \log m \& m \log m(L)$
Kanungo, et. al. [17]	LP	$k$ -means		
Lin & Vitter[19]	Greedy	$k$ -median	$(1 + \epsilon)$	
Mahdian, et. al.[21]	Greedy	UCFL	1.861	$m \log m$
Shmoys, et. al.[23]	LP	UCFL	3.16	

## References

- [1] K. Aardal, F. Chudak, D. B. Shmoys. A 3-approximation algorithm for the  $k$ -level uncapacitated facility location problem.
- [2] K. Alsabti, S. Ranka, and V. Singh. An Efficient  $K$ -Means Clustering Algorithm.
- [3] S. Arora, P. Raghavan, and S. Rao. Approximation schemes for Euclidean  $k$ -medians and related problems.
- [4] S. Arora. Nearly Linear Time Approximation Schemes for Euclidean TSP and other Geometric Problems.
- [5] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit. Local Search Heuristics for  $k$ -median and Facility Location Problems.
- [6] Y. Bartal. Probabilistic Approximations of Metric Spaces and its Algorithmic Applications.
- [7] P. S. Bradley and U .S .Fayyad. Refining Initial Points for the  $K$ -Means Clustering.
- [8] A. F. Bumb, W. Kern. A simple dual ascent algorithm for the multilevel facility location problem.
- [9] M. Charikar, S. Guha. Improved Combinatorial Algorithms for the Facility Location and  $k$ -Median Problems.
- [10] M. Charikar, S. Guha, E. Tardos, and D. B. Shmoys. A constant-factor approximation algorithm for the  $k$ -median problem.
- [11] F. Chudak. Improved approximation algorithms for uncapacitated facility location.

- [12] F. Chudak and D. Williamson. Improved approximation algorithms for capacitated facility location.
- [13] S. Guha and S. Khuller. Greedy strikes back: Improved Facility Location Algorithms.
- [14] D.S. Hochbaum. Heuristics for the Fixed Cost Median Problem.
- [15] K. Jain, M. Mahdian, and A. Saberi. A new greedy approach for facility location problems.
- [16] K. Jain and V. Vazirani. Primal-Dual Approximation Algorithms for Metric Facility Location and  $k$ -Median Problems.
- [17] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. Piatko. An Efficient  $k$ -Means Clustering Algorithm: Analysis and Implementation.
- [18] M. Korupolu, C. Plaxton, and R. Rajaraman. Analysis of a Local Search Heuristic for Facility Location Problems.
- [19] J. Lin, J. S. Vitter. Approximation Algorithms for Geometric Median Problems.
- [20] S. P. Lloyd. Least squares quantization in PCM.
- [21] M. Mahdian, E. Markakis, A. Saberi, V. Vazirani. A Greedy Facility Location Algorithm Analyzed using Dual Fitting.
- [22] R. Ostrovsky and Y. Rabani. Polynomial Time Approximation Schemes for Geometric  $k$ -Clustering.
- [23] D. B. Shmoys, E. Tardos, K. Aardal. Approximation algorithms for the facility location problems.
- [24] B. Zhang, G. Kleyner, M. Hsu. A Local Search Approach to  $K$ -Clustering.