**4.4 度量空间上的聚类算法**

本节介绍求解聚类问题的基于抽样的算法设计与分析。聚类是将给定的数据集合划分为一组子集合，使得每个子集合仅包含“相似”的数据。聚集问题在很多应用领域存在，如数据挖掘、数据压缩、生物信息学、模式识别、模式分类等。下边，我们分别从聚类问题的定义、经典的聚类算法、基于抽样的聚类算法的设计与分析等方面介绍基于抽样的度量空间上的聚类算法。

**4.4.1 聚类问题的定义**

直观地，有限集合*S*上的聚类操作是把*S*划分为*k*个子集合，每个子集合称为一个类，包含距离相近的*S*中的元素。目前，存在四种聚类问题，即*k-*中心聚类问题（*k-median*）、*k-*均值聚类问题（*k-means*）、最小和*k-*聚类问题（*min-sum k-clustering*）、均衡*k-*中心聚类问题（*balanced* *k-median*）。

聚类问题一般定义在度量空间上。下边的定义4.4.1给出了度量空间的定义。

**定义4.4.1.** 一个度量空间是一个二元组(*V*, *d*)，其中，*V*是度量空间的点集合，*d*: *V*×*V*→*N*是非负距离函数，满足对称性和三角不等式，并且*d*(*x*, *y*)=0当且仅当*x*=*y*，*N*是一个数值集合。

下边是四种聚类问题的定义。

**定义4.4.2.** 设(*V, d*)是度量空间，*S*是*V*的加权有限子集，|*S*|=*n*，*w*: *S*→[*1*, *M*]是权函数。*S*的*k*-中心聚类问题定义如下：

输入：*S*, *d*, *w, k*, *0*<*k*≤*n*;

输出：*K*⊆*S*，满足：|*K*|=*k*且最小化*cost*(*S*, *K*)=Σ*j*∈*S w*(*j*) min*i*∈*K*{*d*(*i*, *j*)}。

定义4.4.2中的*K*称为*S*的*k-*中心聚类问题的解，其元素*i*∈*K*称为*i*所在类的中心，*cost*(*S, K*)称为*K*的代价。

**定义4.4.3.** 设(*V, d*)是度量空间，*S*是*V*的加权有限子集，|*S*|=*n*，*w*: *S*→[*1*, *M*]是权函数。*S*的*k*-均值聚类问题定义如下：

输入：*S*, *d*, *w, k*, *0*<*k*≤*n*;

输出：*K*⊆*S*，满足：|*K*|=*k*且最小化*cost*(*S*, *K*)=Σ*i*∈*S w*(*i*)*d2*(*i*, *K*)。

**定义4.4.4.** 设(*V, d*)是度量空间，*S*是*V*的加权有限子集，|*S*|=*n*。*S*的最小和*k-*聚类问题定义如下:

输入：*S*, *d*, *w, k*, *0*<*k*≤*n*;

输出：{*Si* ⊆*S* | *1*≤*i*≤*k* }，满足: ∪*1*≤*i*≤*k Si=S*且最小化Σ*1*≤*i*≤*k* Σ*x,y*∈*Si w*(*i*)*d*(*x*, *y*)。

**定义4.4.5.** 设(*V, d*)是度量空间，*S*是*V*的加权有限子集，|*S*|=*n*，*w*: *S*→[*1*, *M*]是权函数。*S*的均衡*k-*中心问题定义如下：

输入：*S*, *d*, *w, k*, *0*<*k*≤*n*;

输出：*K*⊆*S*和{*Si* | *1*≤*i*≤*k*, *Si*⊆*S*的中心为*ci*∈*K*}，使得∪*1*≤*i*≤*k Si=S*，

而且最小化Σ*1≤i≤k* (*|Si|*Σ*x*∈*Si w*(*x*)*d*(*x*, *ci*))。

度量空间上的*k-*中心聚类问题的研究工作最多。限于篇幅，本节仅讨论度量空间上的*k-*中心聚类问题求解算法的设计与分析。度量空间上*k-*中心聚类问题是NP-难问题，求解其(*1+2/e*)-近似解也是NP-难的。为此，人们提出了一些求解度量空间上*k-*中心聚类问题的多项式时间*O*(*1*)-近似算法[2, 5, 6, 20]。进一步，人们又提出了一些求解度量空间上*k-*中心聚类问题的亚线性时间*O*(*1*)-近似算法[16, 26]。最近，基于抽样方法，人们又提出了计算时间独立于问题输入大小的近似算法[27, 28]，以很高的概率给出度量空间上*k-*中心聚类问题的*O*(*1*)-近似解。

本节将讨论求解度量空间上*k-*中心聚类问题的基于抽样的近似算法的设计与分析。在下边的讨论中，我们称*S*的任意一个包含*k*个点的集合为*k*-点集。任意一个*k*-点集都可以视为*k-*中心聚类问题的一个近似解。我们用*K\**表示*k-*中心聚类问题的优化解，*cost*(*S*, *K\**)为*K\**的代价。如果存在一个常数*c* ≥ *1*，使得*k*-点集*K*的代价*cost*(*S*, *K*)≤*c*×*cost*(*S*, *K\**)，则称*K*是*k-*中心聚类问题的*c*-近似解。

4.4.2节介绍求解*k*-中心聚类问题的一个时间复杂性为*O*(*n2*)的*8-*近似算法。4.4.3节利用这个*8-*近似算法设计一个基于抽样的近似算法，并证明算法的时间复杂性独立于输入大小。

**4.4.2*O*(*n2*)时间*8*-近似算法**

本小节介绍一个求解度量空间上*k*-中心聚类问题的时间复杂性为*O*(*n2*)的*8*-近似算法。我们首先讨论*k*-中心聚类问题的{*0,1*}-线性规划模型，然后讨论近似算法的设计，最后对近似算法进行理论分析。

**1. *k*-中心聚类问题的{0,1}-线性规划模型**

设(*V, d*)是度量空间，*S*是*V*的加权有限子集，|*S*|=*n*，*w*: *S*→[*1*, *M*]是权函数。对于*S*中每个元素*i*∈*S*，定义{0,1}-变量*yi*: 如果*i*被选作中心，则*yi=*1，否则*yi=*0。对于任意两个元素*i*, *j*∈*S*，定义0-1变量*xij*：如果“选择元素*i*作一个类中心”且“元素*j*被聚类到中心为*i*的类中”，则*xij=*1表示，否则*xij=*0。于是，*k-*中心聚类问题可以被表述为如下的{0,1}-线性规划问题*k*-Median-(0,1)-LP，

|  |  |
| --- | --- |
| 最小化∑*i, j*∈*S w*(*j*)*d*(*i,* *j*)*xij*, | (1) |
| 对于∀*j*∈*S*，Σ*i*∈*S xij=1*, | (2) |
| 对于∀*i,* *j*∈*S*，*xij* ≤ *yi*, | (3) |
| Σ*i*∈*S yi* =*k*, | (4) |
| 对于∀*i,* *j*∈*S*，*xij*∈{*0,* *1*}, | (5) |
| 对于∀*i*∈*S*，*yi*∈{*0,* *1*}. | (6) |

约束(2)保证了*S*中每个元素*j*都被聚类到一个且仅一个类中。约束(3)保证了，对*S*中每个元素*i*，只要存在某个元素被指派给了*i*，那么元素*i*就被选为中心。约束(4)保证*k*个元素被选为中心。上述{0,1}-整数规划的一个优化解(,)等价于如下的*k-*中心聚类问题的优化解：

*K\**={*i* | *i*∈*S*, *yi\*=1*}，

同时*K\**的代价为

*cost*(*S*, *K\**)=Σ*j*∈*S w*(*j*)min*i*∈*K*{*d*(*i*, *j*)}=Σ*i,j*∈*S w*(*j*)*d*(*i,* *j*)*xij\**。

度量空间上*k-*中心聚类问题是NP-难的。除非NP=P，我们无法在多项式时间内求出上述{0,1}-线性规划问题的优化解。为了设计多项式近似算法，常用的方法是对{0,1}-线性规划的解空间进行松弛，变量赋值不再强制为整数0或1，而后对松弛形式的优化解进行舍入，得到问题*k*-Median-(0,1)-LP的近似解。

对上述{0*,*1}-线性规划问题的解空间松弛就是将约束(*5*)和约束(*6*)替换为如下约束，

|  |  |
| --- | --- |
| 对于∀*i,* *j*∈*S*，*xij* ≥ *0*, |  |
| 对于∀*i*∈*S*，*yi* ≥ *0*. |  |

本节讨论的求解度量空间上*k-*中心聚类问题的算法分为如下两步：

第一步，首先求解如下的非整数线性规划问题*k*-Median-LP

|  |  |
| --- | --- |
| 最小化Σ*i,j*∈*S w*(*j*)*d*(*i,* *j*)*xij*, | (1) |
| 对于∀*j*∈*S*，Σ*i*∈*S xij=1*, | (2) |
| 对于∀*i,* *j*∈*S*，*xij* ≤ *yi*, | (3) |
| Σ*i*∈*S yi*≤*k*, | (4) |
| 对于∀*i,* *j*∈*S*，*xij* ≥*0*, | (5) |
| 对于∀*i*∈*S*，*yi* ≥*0*. | (6) |

第二步，将第一步得到的*k*-Median-LP的优化解(,)舍入为整数解(,)，用作问题*k*-Median-(0,1)-LP的近似解。(,)的代价为=Σ*i,j*∈*S w*(*j*)*d*(*i,j*)。由于问题*k*-Median-(0,1)-LP的优化解(,)是*k*-Median-LP的解，所以≤*cost*(*S*, *K\**)。

**2. 算法设计**

给定度量空间(*V, d*)，*S*是*V*的加权有限子集，|*S*|=*n*，*w*: *S*→[*1*, *M*]是权函数。下边讨论的算法（简记作*k*-Median-LP-Alg）将对*S*和*w*进行多次变换。为方便叙述，我们今后将使用(*S, w*)表示问题的输入实例。

算法*k*-Median-LP-Alg的第一步调用任意一个现有的非整数线性规划算法，求解问题*k*-Median-LP，得到问题*k*-Median-LP的优化解(,)。(,)的代价为=Σ*i,j*∈*S w*(*j*)*d*(*i,j*)。如前所述，≤*cost*(*S*, *K\**)。

算法*k*-Median-LP-Alg的第二步执行舍入过程，对优化解(,)进行舍入处理，将其转化为问题*k*-Median-(0,1)-LP的近似解(,)，使得(,)的代价=Σ*i,j*∈*S w*(*j*)*d*(*i,j*)满足≤8，从而≤8*cost*(*S*, *K\**)。从(,)可知，*k-*中心聚类问题的8-近似解*K*={*i*∈*S*: *yi=1*}。

算法*k*-Median-LP-Alg的第一步可以调用现有的一般线性规划问题求解算法。由于读者可以在任意一本线性规划教科书中找到这样的算法，我们不再赘述。

下边，我们详细介绍算法*k*-Median-LP-Alg的第二步，即通过舍入的方法，将问题*k*-Median-LP的优化解(,)转化为问题*k*-Median-(0,1)-LP的近似解(,)。给定问题*k*-Median的输入实例(*S,w*)和问题*k*-Median-LP的相应优化解(,)，算法*k*-Median-LP-Alg的第二步由以下六个步骤组成。

**步骤一.** 对∀*j*∈*S*，定义=Σ*i*∈*S d*(*i, j*)为元素*j*的单位代价，其中是的元素。显然，=∑ *j*∈*S w*(*j*)，其中*w*(*j*)是元素*j*的权值。我们将“元素*j*被合并到元素*i*上”定义为如下三个操作：*w*(*i*):=*w*(*i*)+*w*(*j*)、*w*(*j*):=*0*和*S*:*=S*−{*j*}。

给定问题实例(*S, w*)，将*S*中单位代价较大的元素合并到附近的单位代价较小的元素上，从而得到一个新的问题实例(*S'*, *w'*)。具体实施如下：

首先，将*S*中所有元素*j*按值递增排序。不失一般性，设*S*的*n*个元素的有序序列为*L*={*1, ..., n*}，即。

然后，从*1*到*n*依次如下处理每个元素：对于∀*j*∈*L*，顺序查找*L*，得到元素集合*Lj* ={*i*∈*L* | *w*(*i*)*>0, i*<*j*即≤*, d*(*i, j*)≤*4*}。

最后，如果*Lj*非空而且*d*(*inearest*, *j*)=min*i*∈*Lj*{*d*(*i, j*)}，将当前元素*j*合并到元素*inearest*上，即执行操作*w*(*inearest*):=*w*(*inearest*)+*w*(*j*)、*w*(*j*):=*0*和*S*:*=S*−{*j*}。令*S'*=*S*={ *j*∈*S* | *w'*(*j*)>0}，则得到(*S', w'*)。请注意，后面将证明|*S'*| *≤ 2k*。

**步骤二.** 构造问题实例(*S',w'*)的可行解(,)，使得∀*j*∈*S'*，变量*yj'*∈[*1/2*,*1*]并使得(,)的代价*C'=*Σ*i,j*∈*S' w**'*(*j*)*d*(*i,j*)*xij'*≤*2*。设{*j1, ..., jk*}是原始*S*中合并到元素*i*∈*S'*的元素集合，(,)构造如下：

首先，构造问题实例(*S',w'*)的可行解的分量：

∀*i*∈*S',* :=。

然后，构造实例(*S',w'*)的可行解的分量。对于∀*h*∈*S'*，如果> *0*，即优化解(,)将元素*h*“部分地”聚类给了元素*j*，而且元素*j*在步骤一中被合并到了某个元素*i*，则(,)不再将元素*h*“部分地”聚类给元素*j*，而是将*h*聚类到*j*的“部分”聚类给元素*i*，即*xih'*:=+，*xjh'*:=*0*。

**步骤三：**根据(,)构造(*S'*, *w'*)的一个半整数解(,)，使得∀*j*∈*S'*，∈{*1/2*,  *1*}，并且(,)的代价=Σ*i,j*∈*S' w'*(*j*)*d*(*i,j*)*≤C'=*Σ*i,j*∈*S' w'*(*j*)*d*(*i,j*)*xij'*。设在*S'*中距离元素*j*最近的元素为*δ*(*j*)，(,)构造如下：

首先，按照*S'*的每个元素*j*对应的值*w'*(*j*)×*d*(*δ*(*j*), *j*)从大到小排序*S'*中所有元素。我们在步骤一中已经提到*2k*≥|*S'*|。

然后，对*S'*中前*2k*−|*S'*|个元素中的每个元素*j*，令变量:=*1*。对其余*2*(|*S'*|−*k*)个元素每个元素*j*，令:=*1/2*。

最后，当处理完*S'*中的所有元素后，令所有:=*xij'*。

**步骤四**：根据(,)构造(*S'*, *w'*)的一个{0,1}-可行解(,)，使得∀*i,j*∈*S'*，变量和皆为*0*或*1*，并且(,)的代价=Σ*i*, *j*∈*S*′ *w*′(*j*)*d*(*i*, *j*)≤*2*具体构造如下：

首先，根据(,)构造有向图*G*=(*V, E*)，*V*={*i* | *i*∈*S'*}，*E*={(*i*, *δ*(*i*)) | *i*∈*S'*且=*1/2*}。直观地，*1/2*意味着(,)将元素*i*的“一半代价”聚类给了自己，“另一半代价”聚类给了*δ*(*i*)，而*G*中有向边(*i, δ*(*i*))恰好表示了这*i*向*δ*(*i*)的聚类。后边我们将证明*G*中每个连通分量至多有一个有向环，并且不失一般性，可以假设环的长度为*2*。

其次，将*G*修改为有向森林：如果一个连通分量存在长度为*2*的环（设为{(*i, δ*(*i*)), (*δ*(*i*)*, i*)}），则删除环中边(*i, δ*(*i*))，从而得到一个以节点*i*为根的有向树。  
 最后，令*Odd=*{*i*∈*S'* | =*1/2*且*i*是某棵树的奇数层节点}，*Even=*{*i*∈*S' |* =*1/2* 且 *i*是某棵树的偶数层结点}。按照如下方式构造(*S',w'*)的一个{0,1}-可行解(,)：

1. 如果*=1*，则令:=*1*;
2. 如果|*Odd*|<|*Even*|，则对任意*i*∈*Odd*，令:=*1*，对任意*i*∈*Even*，令:=*0*;
3. 如果|*Odd*|≥|*Even*|，则对任意*i*∈*Even*，令:=*1*，对任意*i*∈*Odd*，令:=*0*;
4. 如果*=1*，则令:=*1*，并且对任意*j*∈*S'*，如果*j*≠*i*，则令:=*0*;
5. 如果*=0*，则令:=*1*，并且对任意*j*∈*S'*，如果*j*≠*δ*(*i*)，则令:=*0*.

**步骤五：**根据(,)构造(*S,w*)的一个{0*,*1}-可行解(,)，使得∀*i, j*∈*S*，*xij*∈{*0,1*}， *yj* ∈{*0,1*}。后面将证明(,)的代价*C=*Σ*i,j*∈*S w*(*j*)*d*(*i,j*)*xij*满足*C*≤+4。(,)的构造过程如下：

1. ∀*i*∈*S'*，令:=；
2. ∀*i*∈*S−S'*，令:=；
3. ∀*i, j*∈*S'*，令:=；
4. ∀*i*∈*S−S', j*∈*S*，:=0；
5. ∀*j*∈*S−S'*，如果元素*j*在步骤一中被合并到元素*i*，那么，∀*h*∈*S*，令:=。

**步骤六：**输出解*K*={*i*∈*S*, *yi*=*1*}。

**3. 算法近似比分析**

对于每个元素*i*∈*S*，令*w'*(*i*)为元素*i*在步骤一结束后的新权值。令集合*S'*是在步骤一中由*S*得到的元素集合，即*S'*={*i*∈*S* | *w'*(*i*)*>0*}。不失一般性，设*L*={*1, 2, ..., n*}是*S*的*n*个元素按照值递增排序的结果，即。

**引理4.3.1.** 任意元素*i, j*∈*S'*，*d*(*i, j*)>4*max*{*,*}。

**证明.** 假设存在*i, j*∈*S'*，*d*(*i, j*)≤4*max*{*,*}。由于*d*(*i, j*)≤4*max*{*,*}，则*Lj*={*i*∈*L* | *w*(*i*)*>0, i*<*j*即≤*, d*(*i, j*)≤*4*}非空，或者*Li* ={*j*∈*L* | *w*(*j*)*>0, j*<*i, d*(*i, j*)≤*4*}非空。从而，必存在*inearest*使得*d*(*inearest*, *j*)=min*i*∈*Lj*{*d*(*i, j*)}或者存在*jnearest*使得*d*(*jnearest*, *i*)=min*j*∈*Li*{*d*(*i, j*)}。根据合并规则，步骤一必将*i*合并到*jnearest*或将*j*合并到*inearest*。于是，步骤一结束后，*i*∉*S'*或*j*∉*S'*成立。与假设矛盾。**证毕**。

**引理4.3.2.** 设(,)是(*S,w*)为输入的问题*k*-Median-LP的优化解，为解(,)的代价。令=Σ*j*∈*S' w'*(*j*)，则≤。

**证明.** 根据解的代价的定义，(,)的代价=Σ*i,j*∈*S w*(*j*)*d*(*i,j*)，其中是的元素。显然，=Σ*i,j*∈*S w*(*j*)*d*(*i,j*)=Σ*j*∈*S* Σ*i*∈*S w*(*j*)*d*(*i,j*)=Σ*j*∈*S w*(*j*)[Σ*i*∈*S d*(*i,j*)]。由于∀*j*∈*S*，=Σ*i*∈*S d*(*i, j*)，因此，=Σ*j*∈*Sw*(*j*)。

我们来证明≤，即*−*= Σ*j*∈*S' w'*(*j*)*−*Σ*j*∈*Sw*(*j*)≤0。

对∀*i*∈*S'*，设集合*J*(*i*)={*i*}∪{*j*∈*S* | *j*在步骤一中被合并到了*i*}。根据步骤一的过程可知，下面三个命题成立：

1. ∀*i*∈*S'*，*w'*(*i*)=Σ*j*∈*J*(*i*)*w*(*j*)；

2.**∪***i*∈*S'J*(*i*)=*S*；

3. ∀*i*, *j*∈*S'*，如果*i*≠*j*，则*J*(*i*)∩*J*(*j*)=∅；

4. ∀*i*, *j*∈*S'*，如果*j*∈*J*(*i*)，则≤。

根据上述1、2和3，我们有

*−* =Σ*j*∈*S' w**'*(*j*)*−* Σ*j*∈*S w*(*j*)

=Σ*j*∈*S' w'*(*j*)*−* Σ*i*∈*S'*Σ*j*∈*J*(*i*) *w*(*j*)

=Σ*i*∈*S'* [*w'*(*i*)*−* Σ*j*∈*J*(*i*) *w*(*j*)]

=Σ*i*∈*S'* [Σ*j*∈*J*(*i*) *w*(*j*) *−* Σ*j*∈*J*(*i*) *w*(*j*)]

=Σ*i*∈*S'*Σ*j*∈*J*(*i*) *w*(*j*)(*−*)。

根据4可知，*−*=Σ*i*∈*S'*Σ*j*∈*J*(*i*) *w*(*j*)(*−*)≤0，即≤。**证毕。**

**引理4.3.3.** 设是(*S,w*)为输入的问题*k*-Median-LP的任意可行解，那么对于任意元素*j*∈*S*，*1/2*。

**证明.** 已知，对于每个*j*∈*S*，。如果*1/2*，则必有，矛盾。因此，下式成立

又由于，并且对于任意*i*∈*S*，有，因此

。

**证毕**。

**定理4.3.1.** 设(,)是步骤二构造的以(*S',w'*)为输入的问题*k*-Median-LP的可行解，则(,)满足

1. ∀*i*∈*S−S'*，*yi'*=0；∀*i*∈*S'*，*yi'*≥1/2；

2. (,)的代价满足=Σ*i,j*∈*S' w'*(*j*)*d*(*i,j*)*xij'*≤2。

**证明.**

1. 对于∀*i*∈*S−S'*，因为*i*∉*S*′，所以*i*必被合并到了某个元素*j*∈*S'*。于是，经过步骤二，*yi'*=0。下边，我们来证明∀*i*∈*S'*，*yi'*≥1/2。

首先用反证法证明：对于∀*i*∈*S'*，经过步骤一，在*i*的范围内的所有元素*j*∈*S−S'*，均被合并到了*i*上。假设存在元素*j*∈*S−S'*满足*d*(*i, j*)≤2，它在步骤一被合并到了另一个异于*i*的元素*i'*∈*S'*上。因为*j*要被合并到距离自己最近的元素上，所以*d*(*i', j*)≤*d*(*i, j*)≤2。由三角不等式可知，*d*(*i,i'*)≤*d*(*i,j*)+*d*(*j,i'*)≤2+2=4。由引理4.3.1知，因为*i*, *i*′∈*S*′，所以*d*(*i,i'*)>4*max*{,}，矛盾。于是，对∀*i*∈*S'*，经过步骤一，在*i*的2范围内的所有元素*j*∈*S−S'*，均被合并到了*i*上。

设*J*(*i*)={*i*}∪{*j*∈*S* | *j*在步骤一中被合并到了*i*}，*J*′(*i*)*=*{*i*}∪{*j*∈*S−S'* | *d*(*i,j*)≤}。根据上面证明的结果，*J*′(*i*)⊆*J*(*i*)。由步骤二的构造过程可知，

*yi'=*≥。

由于*J*′(*i*)*=*{*i*}∪{*j*∈*S−S'* | *d*(*i,j*)≤}=，从引理4.3.3可知，=≥*1/2*。因此，*yi'* ≥*1/2*。于是，∀*i*∈*S'*，*yi'*≥1/2。

2. 我们来证明的代价≤2。由引理4.3.2可知，≤。因此，只需证明≤2。

对∀*i*∈*S'*, *j*∈*J*(*i*)，令*H*(*j,i*)={*h*∈*S'* | *j*∈*J*(*i*)且>0}，*H*(*i*)=∪*j*∈*J*(*i*)*H*(*j,i*)。从而，

=Σ*h*∈*S' w'*(*h*)

=∑*h*∈*S' w'*(*h*)[∑*i*∈*S' d*(*i,h*)+∑*i*∈*S*−*S'd*(*i,h*)]

=∑*h*∈*S' w'*(*h*)[∑*i*∈*S'*:*h*∈*S'*−*H*(*i*) *d*(*i,h*)+∑*i*∈*S'*:*h*∈*H*(*i*)*d*(*i,h*)+∑*i*∈*S*−*S'd*(*i,h*)]

=∑*h*∈*S' w'*(*h*)∑*i*∈*S'*:*h*∈*S'*−*H*(*i*) *d*(*i,h*)+∑*h*∈*S' w'*(*h*)[∑*i*∈*S'*:*h*∈*H*(*i*)*d*(*i,h*)+∑*i*∈*S*−*S'd*(*i,h*)]

=∑*h*∈*S'*∑*i*∈*S'*:*h*∈*S'*−*H*(*i*) *w'*(*h*)*d*(*i,h*)+∑*h*∈*S'*[∑*i*∈*S'*:*h*∈*H*(*i*) *w'*(*h*)*d*(*i,h*)+∑ *j*∈*S*−*S' w'*(*h*)*d*(*j,h*)].

同时，我们有

=∑*i,j*∈*S' w'*(*j*)*d*(*i,j*)*xij'*

=∑*j*∈*S'*∑*i*∈*S' w'*(*j*)*d*(*i,j*)*xij'*

=∑*h*∈*S'*∑*i*∈*S'*: *h*∈*S'*−*H*(*i*) *w'*(*h*)*d*(*i,h*)*xih'*+∑*h*∈*S'*∑*i*∈*S'*: *h*∈*H*(*i*) *w'*(*h*)*d*(*i,h*)*xih'.*

步骤二构造结果如下：∀*h*∈*H*(*i*)，*xih'=ih*+∑*j*∈*J*(*i*)*jh*，∀*h*∈*S'*−*H*(*i*)，*xih'=ih*。于是，

=∑*h*∈*S'*∑*i*∈*S'*: *h*∈*S'*−*H*(*i*)*w'*(*h*)*d*(*i,h*)+∑*h*∈*S'*∑*i*∈*S'*: *h*∈*H*(*i*) *w'*(*h*)*d*(*i,h*)(+∑*j*∈*J*(*i*))

=∑*h*∈*S'*∑*i*∈*S'*: *h*∈*S'*−*H*(*i*)*w'*(*h*)*d*(*i,h*)+∑*h*∈*S'*[∑*i*∈*S'*: *h*∈*H*(*i*)*w'*(*h*)*d*(*i,h*)+∑*i*∈*S'*: *h*∈*H*(*i*)∑*j*∈*J*(*i*)*w'*(*h*)*d*(*i,h*)].

由于∀*j*∈*S*−*S'*都被唯一合并到某个*i*∈*S'*，因此⋃*i*∈*S' J*(*i*)=*S*−*S'*。从而，

∑*j*∈*S*−*S' w'*(*h*)*d*(*i,h*)

= ∑*j*∈⋃*i*∈*S'J*(*i*)*w'*(*h*)*d*(*i,h*)

= ∑*i*∈*S'*∑*j*∈*J*(*i*)*w'*(*h*)*d*(*i,h*)

= ∑*i*∈*S'*: *h*∈*H*(*i*)∑*j*∈*J*(*i*) *w'*(*h*)*d*(*i,h*)+∑*i*∈*S'*: *h*∈*S'*−*H*(*i*)∑*j*∈*J*(*i*) *w'*(*h*)*d*(*i,h*).

对∀*i*∈*S'*，如果*h*∈*S'*−*H*(*i*)且*j*∈*J*(*i*)，则必有=0，因为否则根据*H*(*i*)的定义可知，*h*∈*H*(*j,i*)⊆*H*(*i*)，这与*h*∈*S'*−*H*(*i*)矛盾。从而∑*i*∈*S'*: *h*∈*S'*−*H*(*i*)∑*j*∈*J*(*i*) *w'*(*h*)*d*(*i,h*)=0。因此，我们有

∑*j*∈*S*−*S' w'*(*h*)*d*(*i,h*)=∑*i*∈*S'*: *h*∈*H*(*i*)∑*j*∈*J*(*i*) *w'*(*h*)*d*(*i,h*).

从而，

=∑*h*∈*S'*∑*i*∈*S'*: *h*∈*S'*−*H*(*i*) *w'*(*h*)*d*(*i,h*)+∑*h*∈*S'*[∑*i*∈*S'*: *h*∈*H*(*i*) *w'*(*h*)*d*(*i,h*)+∑*j*∈*S*−*S' w'*(*h*)*d*(*i,h*)].

对于∀*j*∈*S*−*S'*，令*j*在步骤一中被合并到*i*∈*S'*。那么，对于∀*h*∈*H*(*j,i*)，*d*(*i,j*)≤*d*(*j,h*)，因为否则*j*不会被合并到*i*上，即*j*∉*S−S'*，矛盾。于是，对∀*h*∈*S'*与∀*j*∈*S*−*S'*，如果>0，则对于∀*i*∈{*i*∈*S'* | *j*∈*J*(*i*)}，*h*∈*H*(*j,i*)，因此*d*(*i,j*)≤*d*(*j,h*)。从而

*d*(*i,h*)≤*d*(*i,j*)+*d*(*j,h*)≤2*d*(*j,h*)。

于是，我们有

≤∑*h*∈*S'*∑*i*∈*S'*: *h*∈*S'*−*H*(*i*)*w'*(*h*)*d*(*i,h*)*ih*+∑*h*∈*S'*[∑*i*∈*S'*: *h*∈*H*(*i*)*w'*(*h*)*d*(*i,h*)*ih*+∑*j*∈*S*−*S' w'*(*h*)2*d*(*j,h*)*jh*]

≤2∑*h*∈*S'*∑*i*∈*S'*: *h*∈*S'*−*H*(*i*)*w'*(*h*)*d*(*i,h*)*ih*+2∑*h*∈*S'*[∑*i*∈*S'*: *h*∈*H*(*i*)*w'*(*h*)*d*(*i,h*)*ih*+∑*j*∈*S*−*S' w'*(*h*)*d*(*j,h*)*jh*]

=2[(∑*h*∈*S'*∑*i*∈*S'*: *h*∈*S'*−*H*(*i*) *w'*(*h*)*d*(*i,h*)*ih*+∑*h*∈*S'*∑*i*∈*S'*: *h*∈*H*(*i*)*w'*(*h*)*d*(*i,h*)*ih*)+∑*h*∈*S'*∑*j*∈*S*−*S' w'*(*h*)*d*(*j,h*)*jh*]

=2[∑*h*∈*S'*∑*i*∈*S'*: *h*∈*S' w'*(*h*)*d*(*i,h*)*ih*+∑*h*∈*S'*∑*j*∈*S*−*S' w'*(*h*)*d*(*j,h*)*jh*)

=2[∑*h*∈*S'*∑*i*∈*S' w'*(*h*)*d*(*i,h*)*ih*+∑*h*∈*S'*∑*j*∈*S*−*S' w'*(*h*)*d*(*j,h*)*jh*)

=2[∑*h*∈*S'*∑*i*∈*S'**w'*(*h*)*d*(*i,h*)*ih*+∑*h*∈*S'*∑*i*∈*S*−*S' w'*(*h*)*d*(*i,h*)*ih*)

=2∑*h*∈*S'*∑*i*∈*S**w'*(*h*)*d*(*i,h*)*ih*

=2∑*h*∈*S' w'*(*h*)∑*i*∈*S**d*(*i,h*)*ih*

=2∑*h*∈*S' w'*(*h*)*h*

=2

于是，≤2≤2。**证毕。**

**推论4.3.1.** |*S'*|≤*2k*。

**证明.** 由于任意*i*∈*S'*，满足*yi'* ≥*1/2*。从问题*k*-Median-LP的约束(4)可以知道，Σ*i*∈*S yi* =*k*。由于*S'*⊆*S*，Σ*i*∈*S*′ *yi* ≤*k*。于是，|*S'*|≤*2k*。**证毕。**

**引理4.3.4.** 设为以(*S,w*)为输入实例的问题*k*-Median-LP的最优解。那么，∀*i*∈*S*，如果>*0*，则=。

**证明.** 假设∃*i*∈*S*，>*0*，≠。我们可以如下构造以(*S,w*)为输入实例的问题*k*-Median-LP的另一个解。构造如下：从矩阵中任选一个≥(−)，并令=−(−)，=+(−)。对于中所有非和项，令=。**=。**因为*d*(*i*, *i*)=0，所以=0。又由于=−(−)<，解的代价。这与“为以(*S,w*)为输入实例问题*k*-Median-LP的最优解”矛盾。从而引理成立。**证毕**。

**引理4.3.5.** 给定正整数*k*，*n*个正实数*a*1≥...≥*an*和*n*个变量*y*1*,*...*,yn*∈[*1/2,1*]。如果，那么*y*1:=*1*, …, *y2k−n*:=*1*, *y2k−n+1*:=*1/2*, …, *yn*: =*1/2*一定是使得最大化的赋值。

**证明.** 假设存在另一组赋值*y*1*',*...*,yn'*∈[*1/2,1*]满足，使得

得*A*=<*A*′=。那么，

*<0*，

即。令，。由于*a*1≥...≥*an*，所以*a*≥*b*。显然，

。

又由于，我们有，

。

又因为，从而有

。

这与矛盾。引理成立。**证毕。**

**定理4.3.2. 令**是步骤三从步骤二产生的(,)构造的以(*S',w'*)为输入实例的问题*k*-Median-LP的可行解，是的代价，*C*′=Σ*i,j*∈*S*′是(,)的代价，则。

**证明.** 不失一般性，设序列*S'=*{1,2,...,|*S'*|}为步骤三中按*w'*(*j*)*d*((*j*)*, j*)值排序的非递增序列。由引理4.3.4可知，∀*j*∈*S'*，一定满足*xjj'=yj'*≥*1/2*。直观地，一定将*j*的“*xjj'=yj'*部分”聚类给了自己，同时将“其余*1−*部分”聚类给了*S'*中距离*j*最近的元素。令函数

*C′* =Σ*i,j*∈*S' w'*(*j*)*d*(*i, j*)*xij*′

=Σ*j*∈*S'w'*(*j*)Σ*i*∈*S'd*(*i, j*)*xij'*

=Σ *j*∈*S'w'*(*j*)[Σ*i*∈*S'*: *i=j d*(*i, j*)*xij'+*Σ*i*∈*S'*: *i≠jd*(*i, j*)*xij'*]

=Σ *j*∈*S'w'*(*j*)Σ*i*∈*S'*: *i≠jd*(*i, j*)*xij'*

=Σ*j*∈*S*′

=Σ*j*∈*S*′−Σ*j*∈*S*′。

令函数*F*(***y***)=Σ*j*∈*S*′−Σ*j*∈*S*′，则*C′* =*F*()。

步骤三构造的{*1/2,1*}-可行解如下：

对于*1*≤*i*≤*2k*−|*S*′|，=1；

对于*2k*−|*S*′|≤*i*≤|*S*′|，=*1/2*；

对于*1*≤i, j≤|*S*′|，=*xij'*。

从*k*-Median-LP的约束(4)可知，Σ*j*∈*S*′≤*k*。由定理4.3.1可知∈[*1/2,1*]。由推论4.3.1知|*S'*|≤*2k*。由于全部|*S'*|个元素是按*w'*(*j*)*d*((*j*)*,j*)值从大到小排序的，因此，根据引理4.3.5可知，是最大化Σ*j*∈*S*′的赋值，即最小化*F*(***y***)。于是，步骤三构造的{*1/2,1*}-可行解的代价最小化函数*F*(***y***)。又因为*C′* =*F*()，所以。**证毕。**

**定理4.3.3.** **令**是步骤四从步骤三产生的构造的以(*S',w'*)为输入实例的问题*k*-Median-LP的{*0,1*}-可行解，是的代价，是的代价，则。

**证明.** 在步骤四从构造的有向图*G*(*V,E*)中，每个顶点至多只有一条出边，因此每个连通分量至多有*1*个有向环。我们来证明步骤四一定为*G*构造出一个有向森林。对于*G*中任意长度大于*2*的环*C*，不失一般性，设*C*的长度为*k*，即*C*={*1,2,...,k,1*}。显然，*C*中*k*个点对应的元素都在输入度量空间(*V*, *d*)中。对于*C*中的任意*i*，我们来证明*d*(*i*, *i+1*)=*d*(*i+1*, *i+2*)。因为*i+2*是*i+1*的最近点，因此*d*(*i, i+1*)≥*d*(*i+1, i+2*)。又因为*i+1*是*i*的最近点，因此*d*(*i+1, i+2*)≥*d*(*i*, *i+1*)。所以*d*(*i*, *i+1*)=*d*(*i+1*, *i+2*)。由于*i*是*C*中任意一个点，C中任意两点的距离都相等。因此，我们可以删除一条边(*i,* *j*)，再加入边，如(*j, i*)，使得*C*不再是环。于是，我们可以使得*G*中任意长度大于*2*的环变为非环，得到一个新的符合步骤四定义的有向图，且使得图中环长度至多为*2*。这样，图中的任意连通分量一定是一棵有向树，或者包含一个长度为2的环。删除长度为2的环之任一条边后，一定得到一棵有向树。因此，步骤四经过上述处理后，得到新图一定是一个有向森林。

显然，*Odd*⋃*Even*={*i*∈*S'*: =*1/2*}。我们来证明|*Odd*|+|*Even*|=*2*(*|S'|*−*k*)。令*M*=|*Odd*|+|*Even*|，则根据k-Median-LP的约束(4)，*M*×(*1/2*)*+*(|*S'*|−*M*)×*1*=*k*。从而可得|*Odd*|+|*Even*|=*2*(*|S'|*−*k*)。同理，由于*S'*−(*Odd*⋃*Even*)={*i*∈*S'*: =*1*}，我们有|*S'*−(*Odd*⋃*Even*)|=*2k*−*|S'|*。显然，(*2k*−*|S'|*)×*1+2*(*|S'|*−*k*)×*1/2*=*k*。因此，经步骤四得到的解至多选择*k*个元素作为中心，即，是以(*S',w'*)为输入实例的问题*k*-Median-LP的可行解。同时，下面不等式成立，

由于时≤1且=*1/2*，即≤*2*, 所以

。

**证毕。**

**定理4.3.4.** 设是步骤四构造的以(*S',w'*)为输入实例问题*k*-Median-LP的{*0,1*}-整数可行解，是步骤五从构造的以(*S,w*)为输入实例的问题*k*-Median-(0, 1)-LP的{*0,1*}-整数可行解。如果的代价为，的代价为，那么≤+4。

**证明.** 根据定义可知，*C*=∑*i,j*∈*S w*(*j*)*d*(*i,j*)*xij*。根据*i*和*j*的不同情况，我们有*C*=∑*i,j*∈*S'w*(*j*)*d*(*i,j*)*xij*+∑*i*∈*S', j*∈*S−S'w*(*j*)*d*(*i,j*)*xij*+∑*i*∈*S−S',j*∈*S'w*(*j*)*d*(*i,j*)*xij*+∑*i,j*∈*S−S' w*(*j*)*d*(*i,j*)*xij*。

根据步骤五的4可知，∀*i*∈*S−S'*, *j*∈*S*，*xij=0*。因此，我们有

∑*i*∈*S−S',j*∈*S'w*(*j*)*d*(*i,j*)*xij=0*且∑*i,j*∈*S−S' w*(*j*)*d*(*i,j*)*xij=0*。

从而，

*C*=∑*i,j*∈*S'w*(*j*)*d*(*i,j*)*xij*+∑*i*∈*S',j*∈*S−S'w*(*j*)*d*(*i,j*)*xij*。

替换下标有，

*C*=∑*h,i*∈*S'w*(*i*)*d*(*h,i*)*xhi*+∑*h*∈*S',j*∈*S−S'w*(*j*)*d*(*h,j*)*xhj*

=∑*h*∈*S'*∑*i*∈*S'w*(*i*)*d*(*h,i*)*xhi*+∑*h*∈*S',j*∈*S−S'w*(*j*)*d*(*h,j*)*xhj*。

对于∀*i*∈*S'*，令*J*′(*i*)={*j*∈*S* | *j*在步骤一中被合并到了*i*}。那么对于∀*h*∈*S'*，∀*j*∈*S−S'*，要么*j*∈*J*′(*h*)，要么存在*i*∈*S'−*{*h*}，*j*∈*J*′(*i*)。因此，

*C*=∑*h*∈*S'*∑*i*∈*S'w*(*i*)*d*(*h,i*)*xhi*+∑*h*∈*S'*∑*j*∈*J*′(*h*)*w*(*j*)*d*(*h,j*)*xhj*+∑*h*∈*S'*∑*i*∈*S'−*{*h*}: *j*∈*J*′(*i*)*w*(*j*)*d*(*h,j*)*xhj*。

根据步骤五的5可知，∀*j*∈*S−S'*，如果*j*∈*J*′(*i*)，那么∀*h*∈*S'*，*xhj*=*xhi*。因此，

*C* =∑*h*∈*S'*∑*i*∈*S'w*(*i*)*d*(*h,i*)*xhi*+∑*h*∈*S'*∑*j*∈*J*′(*h*)*w*(*j*)*d*(*h,j*)*xhh*+∑*h*∈*S'*∑*i*∈*S'−*{*h*}, *j*∈*J*′(*i*)*w*(*j*)*d*(*h,j*)*xhi*

=∑*h*∈*S'*∑*i*∈*S'w*(*i*)*d*(*h,i*)*xhi*+∑*h*∈*S'*[∑*i=h*, *j*∈*J*′(*i*)*w*(*j*)*d*(*h,j*)*xhi* +∑*i*∈*S'−*{*h*}, *j*∈*J*′(*i*)*w*(*j*)*d*(*h,j*)*xhi*]

=∑*h*∈*S'*∑*i*∈*S'w*(*i*)*d*(*h,i*)*xhi*+∑*h*∈*S'*[∑*i*∈*S'*, *j*∈*J*′(*i*)*w*(*j*)*d*(*h,j*)*xhi*]

=∑*h*∈*S'*∑*i*∈*S'w*(*i*)*d*(*h,i*)*xhi*+∑*h*∈*S'*∑*i*∈*S'*, *j*∈*J*′(*i*)*w*(*j*)*d*(*h,j*)*xhi*

=∑*h*∈*S'*∑*i*∈*S'w*(*i*)*d*(*h,i*)*xhi*+∑*h*∈*S'*∑*i*∈*S'*∑*j*∈*J*′(*i*)*w*(*j*)*d*(*h,j*)*xhi*。

对于∀*j*∈*J*′(*i*)，应用三角不等式*d*(*h,j*)≤*d*(*h,i*)+*d*(*i,j*)，可得

*C* ≤∑*h*∈*S'*∑*i*∈*S'w*(*i*)*d*(*h,i*)*xhi*+∑*h*∈*S'*∑*i*∈*S'*∑*j*∈*J*′(*i*)*w*(*j*)*d*(*h,i*)*xhi*+∑*h*∈*S'*∑*i*∈*S'*∑*j*∈*J*′(*i*)*w*(*j*)*d*(*i,j*)*xhi*

=∑*h*∈*S'*∑*i*∈*S'*[*w*(*i*)+∑*j*∈*J*′(*i*)*w*(*j*)]*d*(*h,i*)*xhi*+∑*h*∈*S'*∑*i*∈*S'*∑*j*∈*J*′(*i*)*w*(*j*)*d*(*i,j*)*xhi*。

根据步骤一构造可知，∀*i*∈*S'*，*w'*(*i*)=*w*(*i*)+∑*j*∈*J*′(*i*)*w*(*j*)，因此，

*C* ≤ ∑*h*∈*S'*∑*i*∈*S'w'*(*i*)*d*(*h,i*)*xhi*+∑*h*∈*S'*∑*i*∈*S'*∑*j*∈*J*′(*i*)*w*(*j*)*d*(*i,j*)*xhi*。

根据步骤五的3，∀*i, j*∈*S'*，*xij*=*ij*。因此，

*C* ≤ ∑*h*∈*S'*∑*i*∈*S' w'*(*i*)*d*(*h,i*)*hi*+∑*h*∈*S'*∑*i*∈*S'*∑*j*∈*J*′(*i*)*w*(*j*)*d*(*i,j*)*xhi*。

根据定义=∑*i,j*∈*S' w'*(*j*)*d*(*i,j*)*ij*可得，

*C* ≤+∑*h*∈*S'*∑*i*∈*S'*∑*j*∈*J*′(*i*)*w*(*j*)*d*(*i,j*)*xhi*。

根据步骤一构造，由于任意一个被合并到*i*的*j*，即∀*j*∈*J*′(*i*)，必有*d*(*i,j*)≤4。带入上式可得，

*C* ≤+∑*h*∈*S'*∑*i*∈*S'*∑*j*∈*J*′(*i*) 4*w*(*j*)*xhi*。

由于(**x**,**y**)是{0,1}-整数解，因此根据k-Median-LP约束(2)可知，对于∀*i*∈*S'*，有且仅有一个*h*∈*S'*，使得*xhi*=1，且对于∀*h'*∈*S'*，如果*h'*≠*h*，则*xh'i*=0。因此，我们有

∑*h*∈*S'*∑*i*∈*S'*∑*j*∈*J*′(*i*) 4*w*(*j*)*xhi*

=∑*i*∈*S'*∑*j*∈*J*′(*i*)∑*h*∈*S'* 4*w*(*j*)*xhi*

=∑*i*∈*S'*∑*j*∈*J*′(*i*) 4*w*(*j*)[∑*h*∈*S' xhi*]

≤∑*i*∈*S'*∑*j*∈*J*′(*i*) 4*w*(*j*)

显然，**∪***i*∈*S'J*′(*i*)=*S−S'*。对于∀*i*, *j*∈*S'*，如果*i*≠*j*，则*J*′(*i*)∩*J*′(*j*)=∅。于是，我们有

∑*h*∈*S'*∑*i*∈*S'*∑*j*∈*J*′(*i*) 4*w*(*j*)*xhi*

≤∑*i*∈*S'*∑*j*∈*J*′(*i*) 4*w*(*j*)

=∑*j*∈*S−S'*4*w*(*j*)

=4∑*j*∈*S−S'**w*(*j*)

≤4∑*j*∈*S**w*(*j*)

=4。

从而，*C* ≤+4。定理成立。**证毕。**

**定理4.3.5.** 舍入算法输出的{*0,1*}-整数可行解是问题k-Median-(0,1)-LP的一个*8*-近似解。

**证明.** 由于k-Median-(0,1)-LP的最优解也是k-Median-LP的可行解，所以≤。由定理4.3.1、定理4.3.2和定理4.3.3可知，≤≤≤，从而

≤≤≤，

再由**定理4.3.4**，可知

≤+4≤8≤8。

因此，算法*k*-Median-LP-Alg输出的解是一个*8*-近似解。**证毕。**

**4. 算法**k-Median-LP-Alg**时间复杂性**

我们可以在算法k-Median-LP-Alg中调用目前最好的时间复杂度为*O*(*n4.77*)的线行规划算法[xx]求解问题k-Median-LP。这样，k-Median-LP-Alg算法求解问题k-Median-LP的时间复杂度为*O*(*n4.77*)。

下边我们来分析舍入过程的时间复杂度。

步骤一需要*O*(*n2*)的时间计算所有，需要*O*(*n*log*n*)的时间将他们排序，因此时间复杂性为*O*(*n2*)。

步骤二需要*O*(*n2*)的时间处理修改所有变量。

步骤三需要处理|*S'*|个*y*-变量。由推论4.3.1.可知，最多处理*2k*个*y*-变量，处理每个*y*-变量时，需要*O*(*n*)时间查找最近邻，因此，步骤三时间复杂性为*O*(*2kn*)。

步骤四需要*O*(*2k*)时间构造有向图，并修改为森林。而后，需要*O*(*n2*)时间构造所有变量。因此，步骤四时间复杂性为*O*(*n2*)。

步骤五时间复杂性显然为*O*(*n2*)。

综上所述，算法*k*-Median-LP-Alg的时间复杂度为*O*(*n4.77*)。

**4.4.3 时间复杂性独立于输入大小的近似算法**

为方便叙述，首先定义一个可行解的平均代价。假设*K*为*k*-中心聚类问题的任意可行解，定义*K*的平均代价为

其中，*cost*(*S*, *K*)=Σ*j*∈*S w*(*j*)min*i*∈*K*{*d*(*i*, *j*)}。

假设*K\**为*k*-中心聚类问题的优化解，*K*为*k*-中心聚类问题的一个可行解。那么，如果*K*满足，则*K*称为*k*-中心聚类问题的一个-近似解。

显然，任意一个-近似解*K*的代价满足

下面介绍一个时间独立于输入大小的随机抽样算法，它以高概率输出*k*-中心聚类问题的一个-近似解。这个算法简记作*k*-Median-Sampling-Alg。

**1. 算法*k*-Median-Sampling-Alg的设计**

算法*k*-Median-Sampling-Alg的工作步骤如下：首先，从输入集合*S*独立均匀可重复地随机采样*Y*⊆*S*；然后，调用前一节介绍的近似算法*k*-Median-LP-Alg，近似算法计算*Y*上*k*-中心聚类问题的近似解*KY*。最后，输出*KY*作为*S*的-近似解。

算法*k*-Median-Sampling-Alg的详细描述见Algorithm 4.4.1。

|  |
| --- |
| Algorithm 4.4.1：*k*-Median-Sampling-Alg  输入：度量空间 (*V*, *d*)，*V*的子集*S*，算法*k*-Median-LP-Alg,  (*V*, *d*)的直径上界Δ*>0*，误差*ε>0*，错误概率*1>δ >0*.  输出：集合*KY*.  1. 从*S*中独立均匀随机可重复地采样  *max*{*e*, , }  个元素构成样本*Y*;  2. 以*Y*为输入元素集合，执行算法*k*-Median-LP-Alg;  3. 输出算法*k*-Median-LP-Alg产生近似解*KY.* |

**2. 算法*k*-Median-Sampling-Alg的误差分析**

本节首先利用**Hoeffding不等式**证明两个引理。然后，基于这两个引理证明*k*-Median-Sampling-Alg在独立于输入大小的时间内，以至少*1−δ*的概率输出一个(32, *ε*)-近似解。

我们先来回顾一下Hoeffding不等式。设为个独立随机变量。如果存在一个常数，使得。那么，对于和任意常数，下边的Hoeffding不等式成立

**引理4.4.1.** 设常数，*Y*是算法*k*-Median-Sampling-Alg第1步从输入集合*S*随机抽取的样本。如果，则算法*k*-Median-Sampling-Alg的输出*KY*满足

。

**证明.** 令*K\**为*S*上*k*-中心聚类问题的优化解，则。设*Xi*为*Y*中元素*i*到*K\**的加权距离，即*Xi* =*d*(*i,K\**)=min*j*∈*K\**{*d*(*i,j*)}。当*Y*确定时，常量可以视为一个随机变量。显然 ==|*Y*|*costavg*(*Y, K\**)。于是，*costavg*(*Y, K\**)=。从而，

。

由于度量空间内任意两元素间距离不超过度量空间的直径，因此，上式中随机变量均满足。我们分两种情况来证明

。

(1). 如果，由于，应用**Hoeffding不等式**可知，

。

由于，并且，所以。同时，。因此，

。

从而，。

(2). 如果，则令，显然。应用Hoeffding不等式可得

。

由于，所以可得

。

从而，

给定任意元素*i*∈*S*，令*Y*(*i*)为*Y*中距离*i*最近的元素。令集合*K*={*Y*(*i*) | *i*∈*K\**}，我们来证明：

*costavg*(*Y, K*)=Σ*j*∈*Yw*(*j*)min*i*∈*K*{*d*(*i, j*)}*/*|*Y*|≤2*costavg*(*Y, K\**)=2Σ*j*∈*Yw*(*j*)min*i*∈*K\**{*d*(*i, j*)}*/|Y*|

对于任意*j*∈*Y*，设*K\**(*j*)为*j*在*K\**中距离*j*最近的元素，*K*(*j*)为*j*在*K*中距离*j*最近的元素。因此，我们有

*costavg*(*Y, K*)=Σ*j*∈*Yw*(*j*)min*i*∈*K*{*d*(*i, j*)*/*|*Y*|=Σ*j*∈*Yw*(*j*)*d*(*K*(*j*)*, j*)*/*|*Y*|。

*costavg*(*Y, K\**)=Σ*j*∈*Yw*(*j*)min*i*∈*K\**{*d*(*i, j*)}*/*|*Y*|=Σ*j*∈*Yw*(*j*)*d*(*K\**(*j*)*, j*)*/*|*Y*|。

显然，*Y*[*K\**(*j*)]∈*K*⊆*Y*是*K\**(*j*)在*Y*中与*K\**(*j*)最近的元素。由于*K*(*j*)是*K*中与*j*最近的元素且*Y*[*K\**(*j*)]∈*K*，因此*d*(*K*(*j*)*, j*)≤*d*(*Y*[*K\**(*j*)]*, j*)。根据三角不等式，

*d*(*Y*[*K\**(*j*)]*, j*)≤*d*(*Y*[*K\**(*j*)], *K\**(*j*))+*d*(*K\**(*j*)*, j*)。

于是，*d*(*K*(*j*)*, j*)≤*d*(*Y*[*K\**(*j*)], *K\**(*j*))+*d*(*K\**(*j*)*, j*)。因为*Y*[*K\**(*j*)]在*Y*中距离*K\**(*j*)最近，且*j*∈*Y*，所以有*d*(*K\**(*j*)*, Y*[*K\**(*j*)])≤*d*(*K\**(*j*)*, j*)。从而，

*d*(*K*(*j*)*, j*)≤*d*(*Y*[*K\**(*j*)], *K\**(*j*))+*d*(*K\**(*j*)*, j*)≤2*d*(*K\**(*j*)*, j*)。

由*j*的任意性可得，Σ*j*∈*Yw*(*j*)*d*(*K*(*j*)*, j*)≤2Σ*j*∈*Yw*(*j*)*d*(*K\**(*j*)*, j*)。于是，

*costavg*(*Y, K*)=Σ*j*∈*Yw*(*j*)*d*(*K*(*j*)*, j*)*/*|*Y*|≤2Σ*j*∈*Yw*(*j*)*d*(*K\**(*j*)*, j*)*/*|*Y*|=2*costavg*(*Y, K\**)。

于是，的优化解满足： 。由于上*k*-中心聚类问题的8-近似解，我们有

。

前面已经证明了。于是，

。

从而，。**证毕。**

**引理4.4.2.** 设，如果，则算法*k*-Median-Sampling-Alg的输出满足:

Pr[{*cost*avg(*Y*, *KY*)≤*16*(*1+α*)*cost*avg(*S*, *K\**)}∧{*cost*avg(*S*, *KY*)>*16*(*1+α*)*cost*avg(*S*, *K\**)}]≤*δ/2*.

**证明.** 如果*K*是以*S*为输入的代价为*cost*avg(*S*, *Kbad*)>*16*(*1*+*α*)*cost*avg(*S*, *K\**)的算法*k*-Median-Sampling-Alg的输出，则称*K*是*S*的劣质解，简记作*Kbad*。

我们有

Pr[{*cost*avg(*Y*, *KY*)≤*16*(*1+α*)*cost*avg(*S*, *K\**)}∧{*Kbad* ⊆*Y*}]

=Pr[{*cost*avg(*Y*, *KY*)≤*16*(*1+α*)*cost*avg(*S*, *K\**) | *Kbad* ⊆*Y*]Pr[*Kbad* ⊆*Y*]

≤Pr[{*cost*avg(*Y*, *KY*)≤*16*(*1+α*)*cost*avg(*S*, *K\**)]Pr[*Kbad* ⊆*Y*]

令随机变量为中元素到的加权距离，即

*Xi* =*d*(*i,Kbad*)=min*j*∈*Kbad*{*d*(*i,j*)}。

当*Y*确定时，常量可以视为一个随机变量。显然， ==|*Y*|*costavg*(*Y, Kbad*)。于是，*costavg*(*Y, Kbad*)=。从而，

，

由于，所以。又由于，应用Hoeffding不等式可得

。

我们先来估计下界。下边的等式成立：

由于*cost*avg(*S*, *Kbad*)>16(1+α)*cost*avg(*S*, *K\**)，从而

。

下边，我们来估计的上界。由上边的不等式可知

。

从而，

≤

。

设事件*E*为。那么只有当⊆*Y*为*S*中的某个的时候，事件*E*才会发生。此时，算法*k*-Median-Sampling-Alg必然将采样到了*Y*中。因此，

Pr[{*cost*avg(*Y*, *KY*)≤*16*(*1+α*)*cost*avg(*S*, *K\**)}∧{*cost*avg(*S*, *KY*)>*16*(*1+α*)*cost*avg(*S*, *K\**)}]

由于上的*k*-中心聚类问题的可行解总共至多有个，因此至多有个。因此，

Pr[{*cost*avg(*Y*, *KY*)≤*16*(*1+α*)*cost*avg(*S*, *K\**)}∧{*cost*avg(*S*, *KY*)>*16*(*1+α*)*cost*avg(*S*, *K\**)}]

。

下边我们证明：若，则，即。我们只需证明如下两个命题：

1. 若，则。

2. 若，则。

我们先来证明命题1。若，考虑两种情况：(*a*). 如果，则。由于，所以；(*b*). 如果，则，即。于是，。命题(1)成立。

我们再来证明命题2。令，，则可简记为函数*f(y)=k* ln*y−b y2*。再令，则。由于，，于是， 。我们只需证明，即，亦即。由于一元二次方程判别式

，

所以可求得一个根为。由于是减函数，因此*g*(*y*)=也是减函数。根据命题1，

。

由于，

。

因为*g*()=0且*g*(*y*)是减函数，所以当时，*g*(*y*)≤0, 即。

综上所述，引理成立。**证毕。**

基于引理4.4.1与引理4.4.2，可以证明下述定理4.4.1，从而保证了*k*-Median-Sampling-Alg会以至少的概率成为以*S*为输入实例的问题*k*-中心聚类的 (*32*,)-近似解。

**定理4.4.1.** 设

事件，

事件。

如果

，

则算法*k*-Median-Sampling-Alg的输出至少的概率成为以*S*为输入实例的问题*k*-中心聚类的(*32*,)-近似解，即满足。

**证明.** 我们下面证明，从而得证。设

事件*A*=，

事件*B*=。

于是，。我们分两种情证明：。

(1). 。令，可以验证。此时，

事件*A*=，

事件*B*=。

由于

事件，

事件，

因此，。于是，

。

由于，且，则根据引理4.4.1可知，。同时，根据引理4.4.2可知，。因此，

。

(2). 。此时。令，可以验证。此时，

事件*A*=，

事件*B=*。

由于

事件，

事件，

因此，。从而，

。

由于，且，则根据引理4.4.1知，。同时，根据引理4.4.2可知，。因此，

。

定理成立。**证毕。**

**3. 算法***k*-Median-Sampling-Alg**时间复杂性分析**

算法*k*-Median-Sampling-Alg所需样本*S*的大小为，即。同时，算法*k*-Median-LP-Alg的时间复杂度为。因此，以*S*为输入时，算法*k*-Median-LP-Alg的时间复杂性为。也就是说说，算法*k*-Median-Sampling-Alg的时间复杂性为为，是一个关于，和的函数，独立于输入元素集合的大小。