<h1>The p-median Problem</h1>

<h2>Statement of the Problem</h2>

The p-median problem is an application of linear optimization approaches with respect to a graph. Specifically, it is a facility-location model with the purpose of locating $p$ facilities (hereafter referred to as vertices or nodes) to minimize a constraint equation based on weighted average distances between demand nodes and selection location nodes.

<p>

Stated more formally, given some graph $G=(V,E)$ where $$V=(v\_1,\ldots,v\_n: n \in \Bbb N)$$ is a set of nodes of $G$, each representing a demand weight, and where $$E=(e\_1,\ldots,e\_m: m \in \Bbb N)$$ is a set of connected nodes of $G$, the p-median problem seeks to find any number $p$ vertices such that the sum of the weighted distances from each vertex in $G$ to the closest facility is minimized. The lengths between each vertex and each facility is calculated by taking the absolute value of each member of $E$ that represents a successive path connection of nodes in between the two and computing their sum.

</p>

<p>

In this paper, we will be discussing how exactly this calculation is performed. Three sections will be covered:

<ul>

<li>Rectifying the Issue of Computational Complexity</li>

<li>Setting up the p-median Problem as a Linear Programming Problem</li>

<li>Finding a Solution to a Linear Programming Problem</li>

</ul>

</p>

<br>

<h2>Rectifying the Issue of Computational Complexity</h2>

Unfortunately, when attempting to perform optimization calculations for this problem using a general graph, the computation time is $\mathcal N \mathcal P$-hard. We can rectify this however by formulating the problem using a mathematical tree.

<p>

Given some graph $G=(V,E)$ where

$$V=(v\_1,\ldots,v\_n: n \in \Bbb N)$$ is a set of nodes of $G$, and where

$$E=(e\_1,\ldots,e\_m: m \in \Bbb N)$$ is a set of connected nodes of $G$, we call the graph $T=G$ a <b>tree</b> if it is undirected, connected, and contains no cycles. From the procedures detailed in (Benkoczi *et al.)*[[1]](#endnote-1), if $T$ is a both rooted and binary tree, then an $O(nlog^{p+2}n)$ algorithm exists; thus, a solution to the p-median problem on $T$ can be computed in polynomial time.

</p>

<p>

If $T$ is not rooted, then it can be made to be so by arbitrarily selecting a vertex to designate as its root. If $T$ is not binary, then $T$ may be made to be binary by adding a linear number of vertices and edges to that set. Because it follows that any object that is composed of a linear function of objects that can be solved in polynomial time is itself solvable in polynomial time, then it follows that a tree $T$ which is neither rooted nor binary can also be solved in polynomial time.

</p>

<p>

Thus, because the graph $G$ for the p-median problem can be formulated as the tree $T=G$, it follows that we can calculate a solution for the p-median problem in polynomial time.

</p>

<br>

<h2>Setting up the p-median Problem as a Linear Programming Problem</h2>

<p>

Now that we have established that we can find a solution to the p-median problem in polynomial time by setting up the problem as a tree, we desire to formulate the p-median problem as a linear programming problem. The form for the general linear programming problem is as follows:

<blockquote>

Minimize $$f(\mathbf X) = \sum\_{i=1}^n c\_i x\_i \tag{1}$$ subject to $$\sum\_{i=1}^n a\_{ij} x\_i = b\_j \forall \ j=1,2,\ldots,m \tag{2}$$ and $$x\_i\ge 0, \forall \ i=1,2,\ldots,n \tag{3}$$ where $$a\_{ij},c\_i \in \Bbb R \text{, } b\_j \ge 0 \text{, and } m\lt n \tag{4}$$

</blockquote>

<p>

We will first find the total cost as a function of any given collection of $p$ vertices from our set of nodes; we will refer to this function as the <b>objective function</b>. Once this is done, we will use this objective function to formulate the p-median problem in the form of the general linear programming problem. We divide these two procedures into two subsections.

</p>

<br>

<h3>Formulating the Objective Function</h3>

Let $T=G$ be as before, and let

$$F\_p=(\mathscr f\_1,\dots,\mathscr f\_p:p\leq n \in \Bbb N)$$ be a collection of some $p$ vertices in $T$ which represent a potential facility-location set (hereafter referred to as <i>median set</i>) for $T$. Let

$$W=(w\_1,\ldots,w\_n:n \in \Bbb N)$$ be such that $w\_i$ represents the weight for vertex $v\_i \in V$; similarly, define

$$A=(a\_1,\ldots,a\_m:m \in \Bbb N)$$ to be the distance between adjacent vertices such that, if

$$e\_k=(e\_{k\_1},e\_{k\_2})$$ where $e\_{k\_1},e\_{k\_2} \in V$, then

$$a\_k=\lvert e\_k\rvert=\lvert e\_{k\_1} - e\_{k\_2} \rvert,\forall \ e\_k \in E$$

<p>

Let the (tree-metric) distance between any two arbitrary vertices $v\_i,v\_j \in V$ be denoted as $d\_{ij}$, then define the mapping $\psi:v\_i \mapsto \psi(v\_i)$ to be the minimum weighted distance between vertex $v\_i$ and any facility within the potential median set; that is,

$$\psi(v\_i)=w\_i \cdot\min\_{\mathscr f\_z \in F\_p}d\_{v\_i,\mathscr f\_z}$$ Finally, let the potential cost of locating a facility at the location $\mathscr f\_z$ be defined as the mapping $\gamma:\mathscr f\_z \mapsto \gamma(\mathscr f\_z)$.

</p>

<p>

We are now ready to formulate our objective function for any given potential median set: we define the mapping $\Gamma: F\_p \mapsto \Gamma(F\_p)$ by

$$\Gamma(F\_p)=\sum\_{\mathscr f\_z \in F\_p} \gamma(\mathscr f\_z) +\sum\_{v\_i \in V} \psi(v\_i)$$

to be the objective function for any given potential median set $F\_p \subseteq V$.

</p>

<br>

<h3>Creating the General Linear Programming Problem</h3>

Because we wish to select from any arbitrary potential median set the one for which the objective function is minimized, our goal in solving the p-median problem is to find the median set $\mathscr F\_p$ such that $$\mathscr F\_p = \min\_{F\_p \subseteq V} \Gamma (F\_p)$$

We will show that, with a few adjustments in terminology and symbology, this $\mathscr F\_p$ fits the form

<blockquote>

Minimize $$f(\mathbf X) = \sum\_{i=1}^n c\_i x\_i$$

...

\*\*MISC NOTES:

**To be used later on:**

<p></p>

For example, consider the graph $G=(V,E)$ where $$V=(v\_1,v\_2,v\_3)$$ and $$E=(e\_1,e\_2)=((v\_1,v\_2),(v\_2,v\_3))$$ One can easily see that this represents a graph composed of three nodes connected to one another through a straight line. Suppose we wish to calculate the distance from the $v\_3$ node to the other two nodes on the graph. Since $v\_2$ and $v\_3$ are adjacent, we find that the distance between

# Works cited

-all pdfs in folder

<https://en.wikipedia.org/wiki/Tree_(graph_theory)>

1. “(PDF) A New Template for Solving p-Median Problems for Trees in Sub-Quadratic Time.” ResearchGate. Accessed April 22, 2019. <https://www.researchgate.net/publication/220770539_A_New_Template_for_Solving_p-Median_Problems_for_Trees_in_Sub-quadratic_Time>. [↑](#endnote-ref-1)