

**本科生毕业设计[论文]**

**一种社交网络中最短路径计算方法**

院 系 计算机科学与技术学院

专业班级 计算机1210班

姓 名 刘明亮

学 号 U201215107

指导教师 谢夏

2016年 5 月 27日

**学位论文原创性声明**

本人郑重声明：所呈交的论文是本人在导师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外，本论文不包括任何其他个人或集体已经发表或撰写的成果作品。本人完全意识到本声明的法律后果由本人承担。

作者签名： 年 月 日

**学位论文版权使用授权书**

本学位论文作者完全了解学校有关保障、使用学位论文的规定，同意学校保留并向有关学位论文管理部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权省级优秀学士论文评选机构将本学位论文的全部或部分内容编入有关数据进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本学位论文属于 1、保密囗，在 年解密后适用本授权书

2、不保密囗 。

（请在以上相应方框内打“√”）

作者签名： 年 月 日

导师签名： 年 月 日

摘要

当今互联网高速发展，社交网络也越来越受到人们的欢迎，在线社交网络中活跃的用户逐渐构成了一张很大的社交网络。对社交网络的研究也逐渐兴起，询问两个用户结点之间的距离成为了社交网络研究中重要的一环。不止社交网络，结点距离的查询计算在其他大型网络中也有很重要的作用，例如，生物蛋白质的反应，交通网络。这些网络图的结点规模达到上百万，因此经典的精确算法，如BFS等显然不能很好地适应这些网络图。

虽然现有的基于全局标志结点估计结点距离的方法能够快速地估计结点的距离，但是结果的误差还是有一定的改进空间。在利用全局标志结点计算结点距离的时候，不是直接计算，而是重新计算一个本地标志结点，然后根据标志结点计算结点路径，这样能够有效改进结果的准确度，实现较好的全局准确度。

本文介绍了一种基于本地标志结点的最短路径估计方法，在具体计算结点对的最短距离的时候，不直接用全局标志结点的距离进行上界估计，而是将最近公共祖先作为局部标志结点，代替全局标志结点做估计。并对算法占用的内存空间进行了优化，并发现了网络图可以多次压缩的特点，并对实际的准确度进行了实验，针对不同的全局结点选取方法下结果误差进行了比较，最后将本地标志结点的方法应用到社交网络的搜索当中去，进行实验检验准确度。

**关键词：**算法；社交网络；最短路；最短路径树；图；标志结点

Abstract

**Key Words:**Algorithm; Social Networks; Shortest Path Distance; Graph; Landmarks

**目录**

[**摘要** I](#_Toc11570)

[**Abstract** II](#_Toc32631)

[**1 绪论** 1](#_Toc21862)

[1.1 课题研究的背景和意义 1](#_Toc28624)

[1.2 国内外研究现状 3](#_Toc1183)

[1.3 本课题研究的内容 4](#_Toc12283)

[1.4 论文的组织结构 4](#_Toc28347)

[**2 关键理论与技术研究** 5](#_Toc13683)

[2.1 图像分割 5](#_Toc18451)

[2.2 K-means算法 7](#_Toc10421)

[2.3 MapReduce编程模型 9](#_Toc10806)

[2.4 BWP指标 11](#_Toc19229)

[2.5 本章小结 12](#_Toc25095)

[**3 基于K-means的遥感图像分割算法** 13](#_Toc21210)

[3.1 算法简介 13](#_Toc14813)

[3.2 算法描述 13](#_Toc2508)

[3.3 参数的选择 15](#_Toc14025)

[3.4 算法的特点 17](#_Toc5732)

[3.5 本章小结 17](#_Toc19729)

[**4 基于Hadoop平台的K-means图像分割算法** 19](#_Toc19711)

[4.1 K-means算法在MapReduce中的实现 19](#_Toc13317)

[4.2 BWP指标计算在MapReduce中的实现 22](#_Toc17949)

[4.3 基于RGB显著值的K-means分割算法在MapReduce中的实现 24](#_Toc2345)

[4.4 本章小结 27](#_Toc433)

[**5 实验结果及分析** 28](#_Toc24625)

[5.1 实验环境 28](#_Toc17463)

[5.2 初始点选择对比实验 29](#_Toc28039)

[5.3 基于RGB显著值的分割效果 33](#_Toc13726)

[5.4 MapReduce运行时间分析 34](#_Toc1206)

[5.5 结论 34](#_Toc1796)

[**6 总结和展望** 36](#_Toc16455)

[**致谢** 38](#_Toc26411)

[**参考文献** 39](#_Toc228)

# 绪论

## 课题研究的背景和意义

社交网络（Social Networks），是随着互联的高速发展，因为逐渐在网络上兴起的社交网站如Facebook，Renren而形成的人际关系网络。

近年来随着社交网络受到越来越多的人追捧，用户规模达到上千万，甚至上亿，这样一个大型社交网络的形成，引发了研究者的研究兴趣。对大型社交网络性质的研究，在了解社交网络的变化有重要的作用。类似的社交网络，如交通网络，生物蛋白质交互反应网络中，了解网络的性质也是必要的。

结点之间距离是大型网络图上最基本的性质之一。衡量社交网络中用户结点之间的亲密度关系最直接的方式就是结点之间的距离，社交网络搜索中，搜索引擎往往会根据距离将结果中用户进行排序，从而提供更加准确的搜索结果，在网页排序方面也有类似的应用[[[1]](#endnote-1)]。

然而随着社交网络的结点不断增多，不能简单地应用经典的最短路径计算方法。精确的最短路径可以利用宽度优先搜索（BFS），Dijkstra和A\*算法进行计算，在大型的网络图上面，直接应用经典算法计算最短距离显然是不太现实的。对于节点数为n，边的数目为m的大型网络图，一次Dijkstra算法的时间复杂度是 ，采用堆优化的Dijkstra算法时间复杂度为，但是这个数量级的复杂度显然是无法接受的，因为实际应用时，n的规模达到上百万，m的规模达到上千万，算法应该在线高效处理所有的询问。另外如果提前预处理所有的结点对情况，将需要花费空间存储个结果，这样的空间消耗也是巨大的，因此需要一种高效的结点最短距离计算方法。

## 国外研究现状

结点最短路径计算是大型网络图中研究的重要部分，也是社交网络分析研究和应用的基础。随着社交网络的不断发展壮大，引发了大家的广泛研究兴趣，已经有一系列的社交网络Facebook[[2]](#endnote-2)，Twitter[[3]](#endnote-3)，Livejournal[[4]](#endnote-4),Orkut[[5]](#endnote-5)被人研究其性质。

论文[[[6]](#endnote-6)]提出了一种将网络图中结点映射到欧氏坐标空间中的点来计算结点最短路径长度的方法，[[[7]](#endnote-7)]同样提出了将网络图中的结点映射到双曲空间的方法。这两种方法都是基于空间映射的方法。

在论文[[[8]](#endnote-8)]中，提出了基于全局标志结点的估计方法，利用三角不等式的上界估计。[[[9]](#endnote-9)]在全局标志结点的基础上进行了改进，利用本地标志结点将结果准确度提升到更好的效果。

[[10]](#endnote-10)利用2-hop的数据结构保存每个结点的索引，初始时从每个结点出发BFS，BFS的过程中利用Pruned Landmark Labeling来缩小搜索范围，同时减小索引的空间复杂度及BFS时间复杂度。最后能够在较高效率内回答结点距离的询问。[[11]](#endnote-11)利用6中提到的索引，能够支持动态图上面的点和边的更新操作。每一次更新操作需要从增加的边两端结点重新BFS，但是提出的ResumeBFS算法能够降低BFS的复杂度，使得更新操作时间复杂度可以接受。同时提出了另外一种索引建立方法，使得可以类似的支持更新操作，能够获取之前快照上结点间的路径以及结点间最短路径变化的时间点。

## 国内研究现状

大规模网络图的分析对智能交通网络资源分配及社交网络用户亲密度的研究有重要作用。

大规模交通网络图作为复杂网络图的一种也引起了研究者的广泛研究兴趣，[[12]](#endnote-12)对现有的交通网络进行了研究，根据当前网络状况，计算交通最短路，从而更好地分配交通资源，对于缓解城市交通紧张拥堵有一定的作用。

[[13]](#endnote-13)提出了一种基于构造层次网络的最短路径近似计算方法，在每个层次网络中，通过计算结点到中心结点的距离来估计全局网络中结点之间的距离。

## 课题研究的内容

本文介绍了一中基于本地标志结点的最短距离估计算法，利用具体的社交网络数据测试准确度和效率。

具体内容如下：

1. 介绍了基于全局的标志结点的最短路径估计算法的思想，适用范围及可改进之处。
2. 引入了基于局部的标志结点的最短路径算法的思想，利用全局结点计算最短路径树，将最近公共祖先应用到局部结点的最短路径估计中。
3. 发现社交网络图数据可以压缩处理的可能性，并进一步改善准确度。
4. 利用4个社交网络数据对基于全局和局部的标志结点的算法进行测试，比

较测试结果

1. 将基于局部标志结点的距离估计算法应用到搜索排序中，实验测试结果。

## 论文的组织结构

本文剩下章节的结构主要有：

第二章说明了本文的理论基础及基于全局标志结点的距离估计算法；第三章详细阐述了基于局部标志的最短路径估计算法及图压缩优化；第四章详细描述了局部标志结点查询算法的编程思路，并在第五章在不同的数据集上利用不同的标志结点选取策略对算法进行测试；第六章将局部标志结点估计算法应用到社交网络中的搜索排名；最后第七章是全文的总结和展望。

# 理论基础与技术研究

## 社交网络与图论

社交网络是由不同的用户或数据对象之间的相互关系构成的一种关系网络，例如Facebook不同用户之间的好友关系，网页之间的相互链接关系等。下面介绍社交网络中的相关概念。

**结点** 在大型复杂网络中，结点往往是其中具有代表性的对象或物体。在社交网络中，结点可以是网络用户账号，网络空间中的网页，或者是大型交通网络中的公路连接点，公共交通站。

**关系** 常常指网络结点中的某种联系，用户之间的好友关系，网页之间的连接关系等，一旦两个结点建立起这样的关系之后，我们都可以在他们代表的结点中连接一条边。

**社区** 大量用户形成的具有紧密联系的用户群体，在网络图中常常反映为比较紧凑，边数比较稠密的联通分量。社交网络中社区的形成一个原因可能是用户之间共同拥有某项兴趣，网络空间中网页也容易形成这样的紧密联系的网页空间。通过社区的研究能够有效地找到网络图中有影响力的结点[[14]](#endnote-14)。

在本文中，我们将社交网络等这类型的网络进行建模，用经典图论中概念表示如下：

1. 表示图的顶点集合为V，边的集合为E，为方便下文，定义

，表示结点个数，表示边的个数，表示u和v之间的边的权值；

1. **，*，*其中**，满足条件且，表示u和v之间的长度为的一条路径，表示u和v之间的最短路径长度，当权值时，路径长度为经过的各条边之和。

在下文不做说明的情况下，研究的社交网络类型均指无向图。如图2.1是社交网络图的一个例子。

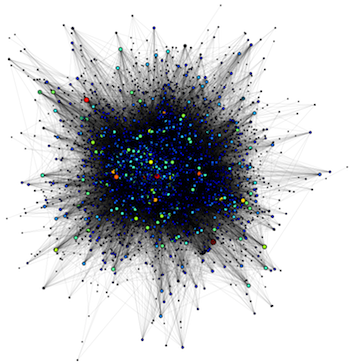


图2-1 社交网络关系图

## 社交网络中的三角不等式

如图2-2对于图中的任何两个结点u和v以及另外选定的点s，之间的距离将会满足如下公式：

 (2-1)  (2-2)

在下文会用到这些公式进行最短路径的上下界估计。

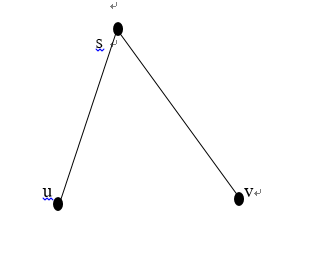


图2-2 三角不等式

## 全局标志结点估计算法

### 算法概括

通过选定的全局标志结点集合，计算每一个全局结点到图中所有结点的最短距离，然后利用三角不等式（2-1）和（2-2）进行界限估计。

### 算法主要思想

算法的基本流程如下[5]：

输入：K值以及待聚类样本集

输出：K个聚类中心点

1. 在总数据样本集N中随机选取K个初始中心点；
2. 计算每个样本X与K个中心点的距离，找出最近的中心点C n，并将其 划入聚类n；
3. 计算每个聚类的均值作为新的中心点；
4. 如果中心点收敛，或者迭代次数达到上限，或者各聚类的平方误差小于 某个阈值，则算法结束，得到聚类中心，否则，返回（2）。

### 距离公式

在计算距离时，通常使用明考斯基度量（Minkowski）表示欧几里得两点之间的距离，公式如下：

 （3-1）

其中，为是向量的距离，为向量的总维度，为第个向量的第个分量。

*r*取不同值的时候，可以得到不同的距离公式[9]，如：

1. *r* = 1，即曼哈顿距离（Manhattan Distance）

 （3-2）

1. *r* = 2，即欧式距离（ Euclidean distance），这是最常用的距离计算方法

 （3-3）

1. ，即切比雪夫距离（Lmax范数），表示两个向量之间的最大距离。

一般地，距离函数满足以下条件：

1. 当且仅当时，；
2. 非负性，任意*i*和*j*，恒有；
3. 对称性，任意*i*和*j*，恒有；
4. 满足三角不等式，任意*i*，*j*和*k*，恒有。

### 算法的性能

算法主要优点：

1. 简单易行、快速、原理便于理解，时间复杂度接近于线性，表示总数据集的样本个数；
2. 处理大数据集时，该算法具有可伸缩性和时效性，效率较高，如Mini Batch K-Means算法；
3. 当数据集满足球状或团状密集性时，分类效果很好。

算法的主要缺点：

1. 参数K需要事先人为确定，这是K-means算法的一个致命的缺点，有时由于缺乏图像的先验知识，无法给出合适的K值，由机器给出K值更加困难；
2. 初始点对聚类效果的影响很大，虽然聚类中心都会收敛，但是不恰当的初始中心可能会导致错误的结果；
3. 距离函数的选定也非常关键，不同函数的收敛方向是不一样的；
4. K-means对噪声和孤立点也非常敏感，每个点的分布都对聚类均值产生影响，所以中心可能会偏移。

## MapReduce编程模型

### MapReduce简介

Hadoop[13]的框架最核心的设计就是HDFS（[分布式文件系统](http://baike.baidu.com/view/771589.htm)）和MapReduce。顾名思义，该模型由两个核心过程组成：Map（映射）和Reduce（归约），具体功能由用户根据自己的需求设计实现，多用于处理文本，计算的输入和输出都是一系列的key/value对。运用这个模型，我们可以在普通PC机上组成高性能的大型集群进行分布式计算。

### 执行概览

MapReduce框架[22]先将输入数据自动划分为M个片段（Splits），每个片段一般为16MB到64MB，用户也可以配置参数控制。这使得Map Task可以在多个机器上并行执行。

Mapper阶段从输入Split中解析出key/value对，并将其传递给用户自定义的map函数处理，产生新的key/value对，并缓存在内存中。

被划分函数（Partition）分为R个片段，然后Reducer将根据key值对其进行排序，并把拥有相同key值得value组织在一起，形成。再调用用户定义的reduce函数处理数据得到结果。执行过程如图2-2所示。



图2-2 MapReduce执行概览

其中，集群中的计算机有一台会作为master来对执行过程进行任务分配和调度。值得注意的是，task和worker并不是一对一的关系，一个worker可能执行多个task。

## BWP指标

上文中提到K-means算法一个严重的缺点就是K值及初始中心点需要人为事先确定。BWP[11]是衡量K值的优劣的一个指标，它综合考虑了类间距离和类内距离，对单个样本的聚类结果进行了有效分析，通过计算遥感图像每个像素点的BWP指标值，得到该图像的BWP和，其值越大则聚类效果越好，K值就越佳，所以我们可以在指定的范围内计算BWP指标和，最大BWP和所对应的K值即为最佳。相关定义[11]如下：

令为总数据集，聚类数为，数据总数为*n*。

定义1 表示聚类中第个样本的最小类间距离，即该样本到其他聚类的类间距离的最小值，即：

 （3-4）

其中，表示聚类*k*的样本数，表示聚类中的第个样本。

定义2 表示聚类中第个样本的类内距离，即该样本到自身聚类的距离，即：

 （3-5）

定义3 表示聚类中第个样本的类内距离和最小类间距离之和，即：

 （3-6）

定义4 表示聚类中第个样本的类内距离和最小类间距离之差，即：

 （3-7）

定义5 （）表示聚类中第个样本的类内类间划分指标，为该样本的聚类距离和聚类离差距离的比值即：

 （3-8）

从上式可以看出，类内距离越小，最小类间距离越大，聚类效果就越优，所以最佳*K*值应该使所有样本的BWP总和达到最大，即：

 （3-9）

 （3-10）

采用BWP确定最佳*K*值需要先选择*K*的搜索范围，然后循环K-means过程计算BWP总和。一般地，取2，按照经验取值为，*n*为样本总数。但是遥感图像的像素点数量是非常巨大的，并不适合采用作为最大*K*值，第4章将给出具体算法。

## 本章小结

本章主要介绍了实验所使用的关键理论技术，包括图像分割的定义与三大类主要的遥感图像分割算法、K-means算法、MapReduce编程模型以及BWP指标。

图像分割中不同的算法应用于不同特征的图像，每个算法都有各自的优点及缺陷。例如基于边缘的分割算法就比较适用于纹理比较清晰的图像；阈值化分割算法比较适用于图像灰度层次比较明显的图像；基于区域的分割算法适用于具有多种互不相同特性的图像。

K-means算法是数据挖掘中最基本、最常用的算法之一，通过不断迭代直到中心点收敛，该算法简单速度，具有伸缩性，可计算大规模的数据集，但是初始点的选择对结果的影响较大，且需要先验知识。

MapReduce的核心就是Map和Reduce，即映射和归约。将输入文件转换为一系列的对，分配给不同的worker处理，再产生新的对，各worker之间是并行无耦合的，这是Map的工作。过程将Map产生的数据中具有相同key值得value合并起来交给Reduce处理，最后输出结果，同样也是对。

BWP是衡量不同*k*值的条件下聚类效果的指标，用于选择最佳*k*值，它同时考虑了聚类结构的类间分离性和类内紧密型。这使得我们有一个标准化的指标去衡量分割效果，而不是人直观地用眼睛去判断。单用这个指标并不能解决初始值选择的问题，下一章将给出具体的方案。

# 基于K-means的遥感图像分割算法

## 算法简介

第二章我们详细介绍了K-means算法的原理及过程，本章将提出一种基于K-means的遥感图像分割算法。该算法可以解决K-means初始点及K值取定的问题，并对聚类效果进行评估。

算法首先对遥感图像的灰度信息进行多项式拟合，然后得到拟合曲线的所有为实数的个极大值点，将这些极大值点作为初始中心点，一轮迭代收敛后得到个聚类中心，并计算所有样本的BWP和。选出两个差值最小的中心点，合并为一个新中心，对的聚类中心继续进行K-means聚类过程，重复此过程，直到聚类数为2时停止。实验表明，该算法具有较好的分割效果。

## 算法描述

根据上面描述的算法思想，本文提出的分割算法流程如下：

初始化：分割数下限值，中心点收敛阈值 th=0

输入：待分割遥感图像

输出：各K值下的BWP指标值以及聚类中心

处理流程：

1. 画出图像的灰度直方图，并采用最小二乘法进行多项式拟合。在一定范围内，拟合误差会随着拟合次数的增加逐渐减小，为避免过度拟合，我们选定第一个拐点为最佳拟合次数；
2. 求出拟合曲线的实数极大值点，并设为初始中心点，聚类数；
3. 计算每个像素点到中心点的距离，并将其划入离它最近的中心点所属的类中。当所有像素点都分类完成后，计算每个聚类的平均值得到新的聚类中心。
4. 如果相邻两次迭代的聚类中心的没有变化，或差距小于阈值th，则表示中心点收敛，停止迭代更新，得到新聚类中心。否则，返回步骤（3）；
5. 计算每个样本点的最小类间距离和类内距离，并得到所有样本的BWP指标值和。
6. 合并中差值最小的两个中心点，不妨设，差距最小，则得到新的聚类中心；
7. 聚类中心；
8. 如果，返回（3）；
9. 统计不同*k*值下的BWP指标和，使其达到最小值的*kopt*值即为最佳*k*值；
10. 对遥感图像分割为*kopt*类分割，即每一个聚类分割为一部分。

该算法的时间复杂度为，其中为像素点个数，接近于线性。时间复杂度为，其中，*M*和*N*分别为遥感图像的长和宽。流程如图3-1所示：



图3-1 算法流程图

## 参数的选择

通常，K值和初始点确定是两个特别关键的问题，在没有先验知识的情况下，人们很难知道要选取多少个类，并且K-means对初始点的依赖很大，不恰当的初始点会使聚类陷入局部最小解。遥感图像分割的初始聚类像素点应该尽量结合图像本身的特点，减少初始点的随机性。另外，彩色图像既可以根据灰度值也可以根据RGB三通道的信息进行分割，这也是一个重要的参数。

从上述流程图可以看出，采用该算法分割图像不需要事先人为指定K值，根据对图像的分析可以自动确定最佳K值以及初始点，选用拟合灰度值方图的多项式曲线的极大值点作为初始中心点是根据大量对比试验得出的结论，第五章将验证其合理性。另外，算法中采用了极大值点而不是极小值点，这也将在实验中进行证明。经典的K-means算法中，在K值确定的情况下，常用的初始点选择方法也有很多：

1. 随机选择*k*个初始点；
2. 把总样本凭直观分为*k*类，将各类的均值作为初始中心，这个方法需要有丰富的先验知识；
3. 基于数据采样的方法；
4. 用个聚类的K-means算法的结果解出*k*聚类的中心问题。即，先把总数据样本看做一个聚类，计算平均值得到中心，然后将和距离最远的样本，即作为的K-means算法的初始中心，以此类推，得到*k*聚类的初始中心。

文献[9]采用的*k*个初始点选择方法如下：

1. 将图像中的像素点根据灰度值由小到大进行排序，排序后的点集用*Y*表示；
2. 把*Y*平均分成段（），每段包含个像素点，、分别表示图像的长和宽；
3. 第位置的像素值即为第*i*个像素点，其中。

值得注意的是，以上介绍的都是在K值确定情况下的初始点选择方法，所以并没有解决K值问题，然而本文提出算法能够同时解决这两个参数问题，并且K值的确定是动态的。

如今的遥感图形多为彩色，但是灰度的图像分辨率更高，所以本文采用了基于灰度值的分割，文献[12]采用基于RGB显著性的图像分割，实验表明该显著值并不适合用于遥感图像，相关定义如下：

定义1 彩色图像包含RGB三个通道的信息，通道信息量、、表示每个通道的信息总量，、、分别表示各通道信息的平均值，即：

 （4-1）

 （4-2）

 （4-3）

 （4-4）

 （4-5）

 （4-6）

其中，， 分别表示图像和长和宽，，，分别表示像素点在R、G、B通道的像素值。

通道信息量对人的视觉作用很大，通常信息量越大，视觉效果就越突出，颜色上就占有越大的权重，所以一个点的在信息量大的通道中的像素值越大就越显著。显著值正是基于这样的原理而产生的。

定义2 RGB颜色信息显著值表示各种的显著性，即：

 （4-7）

 （4-8）

 （4-9）

定义3 各颜色通道权重表示人类视觉对于该颜色的影响程度，即：

 （4-10）

 （4-11）

 （4-12）

定义4 表示像素点的综合信息显著值，即：

 （4-13）

文献[12]对值进行聚类，实验表明，这个算法适合于颜色鲜明的彩色图像，并不适合遥感图像。

## 算法的特点

上述算法充分考虑了遥感图像自身的特点，自动提取初始中心点，不需要事先人为确定K值及初始点，不需要先验知识。相对于随机选取和文献[9]的初始点选择方案，本文选用多项式来拟合灰度直方图，同时避免过度拟合出现不必要的极大值点，这样结合遥感图像本身的像素特征选取的点更具有代表性。在得到极大值点集之后，不直接将所有的点都用于K-means分割，而是通过极大值点个数先确定*k*的范围，循环K-means过程不断合并中心点，得到不同的*k*值下的分割效果，计算出最佳*k*值，克服了K-means算法本身的缺点。而且本文采用了BWP指标来衡量分割效果，减少了人的主观性更多地用数据来判断分割优劣。使整个分割过程更加自动化、数据化、标准化。

通常人们会直观地认为根据彩色信息进行分割比根据灰度分割更准确，因为存在两个像素点灰度值一样，但颜色完全不一样的情况，如果通过RGB颜色值就可以轻易将两者分割开来。然而遥感图像不同于一般数字图像，它是通过接收远程物体的辐射的电磁波而形成的，色彩和边界都比较模糊，然而普通图像是根据物体反射的光形成的，具有的色彩更丰富也更清晰明亮。一般来讲，灰度的遥感图像比彩色的遥感图像具有更高的分辨率，本文采用的算法是根据遥感图像的灰度值进行分割，实验证明，比采用RGB显著值具有更好的分割效果。

另外，在第五章的实验中，我们可以看到此算法除了第一次迭代的次数较多，后续的迭代过程都收敛地非常迅速，这使得我们可以较快捷地得到最佳聚类。

## 本章小结

本章详细地介绍了本文所提出的基于K-means的遥感影像分割算法的流程及其特点，该算法解决了K-means算法初始参数的选择问题。还介绍了目前常用的初始点选择方法和RGB显著值，该显著值可用于分割彩色图像。下一章将对不同初始点选择方法以及基于RGB 值与灰度值的分割方法进行对比，证明本文所提出方法的合理性。

# 基于Hadoop平台的K-means图像分割算法

本文提出的算法是运行在Hadoop平台的，因为遥感图像的像素信息量较大，所以需要将K-means以及BWP计算等过程都进行编程。由于Hadoop一般都用来处理文本，在处理图片的时候需要自己设计相关数据类来读取像素值等信息。

## K-means算法在MapReduce中的实现

Hadoop是一个处理大数据的平台，核心就是HDFS和MapReduce，它的架构比较简单，能有效支持数据密集型的应用，也非常适用于并行遥感图像分析处理。但是Hadoop内置所支持的数据类型有限，且主要用于处理文本日志等，所以不能直接用处理图像数据。本文设计了一个能用于直接处理图片的数据类型，可以实现提取出图像的长度、宽度以及像素矩阵等信息。这样就使得我们在分割图像的时候能够将更多的注意力放在核心算法上，而不用过多考虑对图像本身的处理细节。

是一个重要的并行计算框架，它所计算的数据集具有这样的特点：可以被分解为许多小Data集，并且每一个小的Data集都可以完全并行地处理，互不干扰[10]。由于在K-means图像分割中，每个像素点与中心点计算距离时都是独立运行的，和其他像素点没有关联，所以可以用MapReduce模型来实现。大概的思路是：运行一次MapReduce过程，更新一次聚类中心直到收敛或迭代次数达到上限，像素数据按行存储，并且可以按行进行分片，分片数据之间没有相关性。

### 数据结构

首先，自定义ImageInputFormat类，它继承自FileInputFormat，会在Map过程之前对输入文件进行读取，并提取出相应的key和value。普通的文本文件的key为行号，value为该行的内容，ImageInputFormat读取图片的坐标值为key，灰度值为value，类图如图4-1所示：

ImageInputFormat ImageRecordReader

图4-1 自定义类图

ImageInputformat类中主要重写createRecordReader方法以及创建内部类ImageRecordReader，该类继承自RecordReader。

ImageRecordReader类中的方法主要重写initialize()和nextKeyValue()，前者是对输入文件进行初始化处理，本文中的方法是读取遥感图像，获取图像的长和宽等。后者是获取key和value的方法，如果还有未读取的key和value则返回true，否则，返回false，本文中的方法的key是像素点的坐标值，数据类型为Text，value为该像素点的灰度值，数据类型为IntWritable。

每一次MapReduce过程得到一次聚类结果，通过不断迭代更新中心值，直到达到停止条件。

### Map函数设计

Map函数的任务是从ImageInputFormat得到每个像素点的灰度值，计算它到中心点的距离，并将其归入最近的聚类当中。

如上文所述，Map函数输入的key为像素点的坐标，可表示为，value为灰度值，可表示为，Map阶段的主要任务是计算出每个样本所属的聚类。设中心点为，Map函数计算出每个点距离最近的中心点，则，所以Map函数的输出的key为，value为。

### Reduce函数设计

函数的任务是得到函数输出的聚类结果，重新计算每个聚类的中心，以供主函数判断中心点是否收敛或达到停止条件。

经过过程，Reduce的输入的key为，，value为相同值的集合，即，Reduce阶段的主要任务是计算出每个聚类的新中心。Reduce函数算出*list*中所有的和，并除以*list*中的元素个数，得到聚类的新中心。图4-2和图4-3分别表示K-means的程序流程图与数据流。



图4-2 K-means流程图



图4-3 MapReduce逻辑数据流图

## BWP指标计算在MapReduce中的实现

在K-means算法结束之后需要用BWP来衡量*k*值的分割效果，但是按照原来的算法每个像素点需要与其他所有像素点都算一次距离，遥感图像像素点的数量是巨大的，如果这样计算的话，运算量可想而知。并且，与每一个点都算一次距离也是没有必要的。所以我们采取抽样的算法提取像素点，例如我们计算聚类A中的点到聚类B的距离，则在聚类B中每隔*N*个像素点提取一个点与之计算距离，最后得到平均距离。

计算BWP指标之前，需要生成一些中间数据，即再跑一遍K-means的Map过程，将每个聚类以及聚类所包含的点都记录下来，放入文件Cluster，文件每一行的内容为一个像素点的信息，*ID*表示所属聚类，*gray*表示灰度值，表示坐标。还要生成一个文件TMP，它和Cluster类似，不过每一个像素点对应*k*行的信息，具体为，TMP文件将作为输入数据，其中*n*用来提示函数计算像素点与哪一个聚类的距离。

计算BWP指标和需要两轮MapReduce过程，第一轮计算每个点的最小类间距离和类内距离，第二轮计算得到每个点的BWP值。

### 计算类内及类间距离

第一轮Map之前，先在setup函数中读取Cluster文件，将数据存入变量Clusters中，其数据类型为ArrayList<ArrayList<Integer>>，即将每个聚类的像素点的灰度值存入一个数组中，Clusters记录了像素点的分类信息，后续计算需要的每个聚类的信息都从这个变量中取出。

Map的输入文件是TMP，输入格式为FileInputFormat，即每一行的行为key，内容为value，所以输入为。如果，则根据公式（3-5）计算该像素点的类内距离*dist*，输出key为，value为*dist*。否则，根据公式（3-4）计算该像素点到聚类*n*的距离*dist*，输出key为，value为*dist*。

其中在Map过程中我们已经得到了类内距离，但是还没有得到最小类间距离。Reduce的输入为及，其中包含该点到除自身聚类以外所有聚类的距离值，我们选出中的最小值即为最小类间距离。Reduce输出的key为，value为*dist*，该*dist*。数据流图如下：

图4-4 计算类间与类内距离逻辑数据流图

### 计算BWP值

第二轮轮的Map函数没有实际的计算意义，只是从上一轮输出的文本中重新识别出key和value，供Reduce使用。

Reduce函数的输入，其中为key，为value，并且中刚好包含每个像素点的类内距离及最小类间距离，易证明，像素点的类内距离不大于最小类间距离。所以其中的较小值为，较大值为。

根据公式（3-6）、（3-7）将两者相加、相减即可得到和，最后由（3-8）可得到该像素点的BWP值。数据流图如下：

图4-5 计算BWP值的逻辑数据流图

得到了所有点的BWP值后在主函数中读取输出文件将所有值相加即可得到该*k*值下的BWP和。

## 基于RGB显著值的K-means分割算法在MapReduce中的实现

### 数据结构

本文定义了一个数据类型IntegerArrayWritable，可以看做是一个Int类型的数组，用来记录像素点的RGB值。该类继承自ArrayWritable，相当于Hadoop当中的数组类型。

类似读取灰度值的ImageInputFormat，为了获得图像的RGB值，自定义类ImageInputFormatRGB，该类同样继承自FileInputFormat，与ImageInputFormat的主要不同在于nextKeyValue()的实现，它将读取图像的RGB值以及灰度值并存入一个IntegerArrayWritable类型变量中，并作为value返回。即，该输入类型返回的key是像素的坐标值，value为，长度为4。

在进行K-means聚类之前，需要先计算出每个像素点的显著值，这需要两轮MapReduce过程，分别计算出每个通道的信息量和每个像素点的显著值。

### 计算RGB通道信息量

Map过程的输入的key为，value为，输出比较简单，相当于将RGB值分开，即key为字符串，value为相应的像素值，数据类型是IntWritable。

经过过程后，Reduce输入的key/value对理论上就只有三个了，因为只有三个互不相同的key，即，value为该通道所有像素值的集合，将这些值加起来就可以得到该通道的信息量。数据流图如下：

图4-6 计算通道信息值的逻辑数据流图

### 计算像素点的显著值

在得到、、后就可以根据公式（4-4）~（4-6）计算出各通道信息的平均值、、，根据公式（4-10）~（4-12）算出权重、、。

Map的输入同样为，由于通道信息的平均值及权重都已求出，所以根据公式（4-7）~（4-8）及（4-13）求出显著值，最后将所有像素点的显著值、灰度以及坐标，即，输出保存在文件SIJ中以供后续的计算使用。

本轮计算没有Reduce过程。

### K-means的MapReduce实现

K-means的Map输入为文件SIJ，输入类型数FileInputFormat，所以key为行号，value为。但是对像素点进行聚类时用显著值来计算距离，输出的key为聚类*ID*，value为。

与4.1.3节类似，Reduce过程计算出新的聚类中心。

通过不断循环达到停止条件后，输出收敛的聚类中心。这个过程中并没有使用到*gray*的值，但是在BWP指标的运算中将会使用到。

### 计算BWP指标

与4.2节类似，在计算指标之前需要再运算一次K-means的Map过程生成一些中间数据，即文件Cluster和TMP。但是本节的算法将做一些改动。

中间文件TMP生成的MapReduce过程的Map的输入同样为SIJ文件，对所有的像素点根据显著值进行聚类，输出的key为聚类*ID，*但是输出的value并不是，而是，*gray*对应该像素点的灰度值，*n*的意义与4.2节相同。同理，Cluster的value为。这也就是为什么SIJ文件里要记录每一个点的显著值和灰度值。可以看到这和基于灰度值的算法计算BWP时产生的中间文件完全相同，所以后续的计算也完全相同，也就是再经过两轮MapReduce过程得到BWP和。数据流图如下：



图4-7 生成中间文件TMP的逻辑数据流图

综上所述，本文所提出的遥感影像分割算法在Hadoop平台上的实现过程主要包括K-means聚类和BWP指标值计算两个过程，连接这两个运算过程的是中间文件Cluster和TMP，流程图如下：



图4-8 分割及计算总流程图

## 本章小结

本章是全文的重点之一，详细地介绍了基于RGB显著值和灰度值K-means分割算法以及BWP指标的MapReduce编程实现。K-means的实现就是一个Map和Reduce过程的不断迭代更新中心点，直到达到停止条件。BWP指标的计算比较复杂，因为计算过程较多，而且参与计算的像素点也较多。首先要产生两个中间文件TMP和Cluster，TMP是作为输入文件，为了实现分布式计算为每个worker分配任务，TMP数据的格式为，其中*ID*为key。这样*n*就可以指示worker将计算该点与那个聚类的距离，如果和*ID*相同，则计算类内距离，否则，计算类间距离，这个过程比较耗时，即使我们没有将聚类中的每一个点都参与计算。接下来的一个MapReduce过程将选出最小类间距离，于是得到了两个重要的参数和，接着在下一轮MapReduce中分别计算和并得到BWP指标值。

基于RGB值的K-means分割方法的MapReduce实现与上面基本相同，主要的区别在于聚类的时候计算的是RGB显著值的距离，也就是根据RGB显著值来分类。后续计算BWP的过程也是类似的。

Mapreduce并行计算与单机计算的时间对比，以及基于RGB显著值与灰度值的分割效果对比将在下一章介绍。

# 实验结果及分析

## 实验环境

实验采用一台机器作为Master，即NameNode和JobTracker服务节点，两台机器作为Slave，即DataNode和TaskTracker服务节点。每台机器采用相同的硬件配置，见表5-1。

表5-1 实验硬件及软件配置

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CPU | 硬盘 | 内存 | 操作系统 | 软件版本 |
| Intel CORE i5 | 250G | 2G | CentOS 5.9 | Hadoop-1.2.1, jdk-6u45 |

实验中的Hadoop集群含有三个独立的节点，在hosts文件中配置好各自的IP地址，并在master机器上创建 hadoop用户，使之拥有root权限，集群结构如下图所示：



图5-1 Hadoop集群结构图

## 初始点选择对比实验

文献[9]的提出的算法是将像素点从小到大排序，然后每隔相同距离取一个点，并将该点的灰度值作为一个初始点，实验证明，提问提出的初始点选择方法具有更好的聚类效果。

我们选用图5-2作为实验用图，图片截取自山东省全域遥感图像，来自资源卫星应用中心高分云平台项目。



图5-2 遥感分割原始影像

首先我们得到图像的灰度直方图然后对其进行多项式拟合得到所有实数极值点。从图5-3所示的直方图可以看到几个清晰的波峰，我们要对其进行多项式拟合，根据拟合函数求出这个几个极大值。

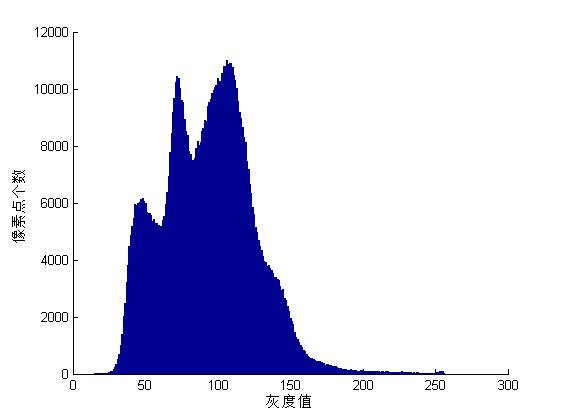


图5-3 灰度直方图

令拟合指数，利用最小二乘法拟合出相应多项式，并计算拟合函数的误差，知道出现第一个拐点，为避免过度拟合，我们取该拐点的值为拟合多项式的指数。

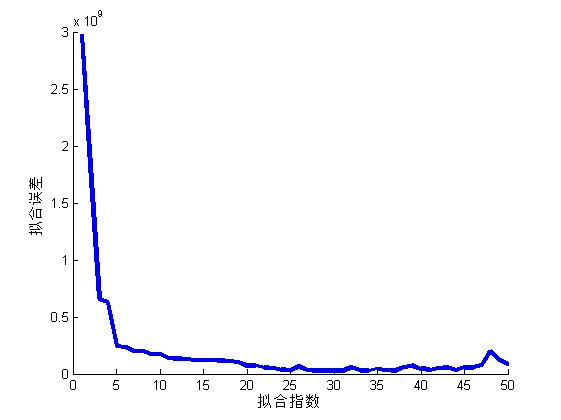


图5-4拟合指数与误差

从图5-4可以看出在指数为25左右出现了一个拐点，因此我们选择这个值为多项式的阶数对灰度直方图进行拟合得到图5-5拟合函数，对求导得到导函数和二阶导函数。最后求解出令=0，<0的所有实数解，即为初始中心点的集合，即。

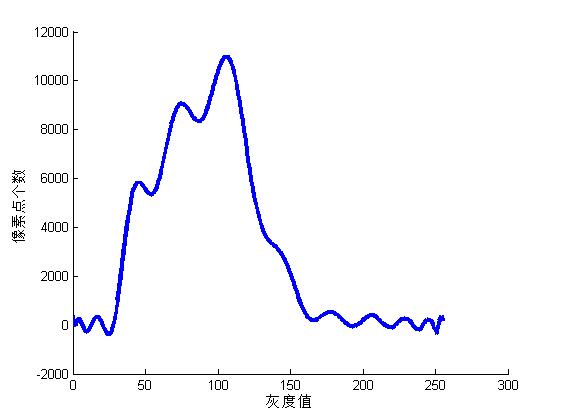


图5-5多项式拟合灰度函数

然而要得到文献[9]的初始点，还要对拟合函数进行积分，并且使之过零点(0,0)。图5-6为灰度图像的积分函数，过原点，并且是单调递增的。易证明，该函数在的值为原始图像的像素点总数值。所以将平均分为*k-1*段，令，求解出相应的*x*值即得到文献[9]的初始中心点集合。

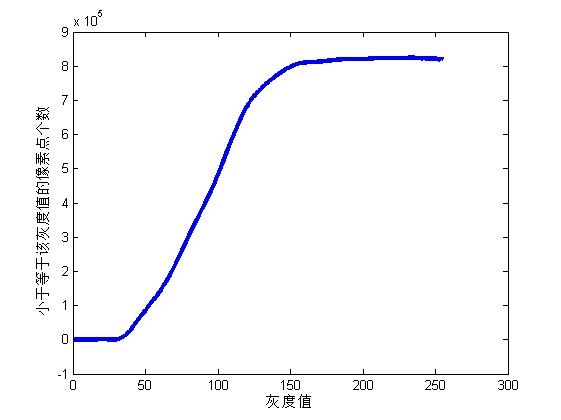


图5-6 积分函数

得到初始聚类中心后，在范围内进行K-means聚类，逐步合并中心点，计算并比较BWP和，找出最优*K*值，相关实验结果如下：

表5-2 本文提出算法的分割结果

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| K值 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 |
| 迭代次数 | 22 | 6 | 5 | 10 | 6 | 10 | 4 | 2 |
| BWP和 | 482984 | 489887 | 499759 | 491227 | 497930 | 461510 | 494585 | 499691 |

表5-3 采用极小值作为初始点的分割结果

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| K值 | 8 | 7 | 6 | 5 | 4 | 3 | 2 |
| 迭代次数 | 21 | 4 | 13 | 6 | 11 | 4 | 2 |
| BWP和 | 498602 | 490882 | 489448 | 495768 | 461993 | 494198 | 498507 |

表5-4 文献[9]所选择初始中心的分割结果

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| K值 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 |
| 迭代次数 | 5 | 5 | 5 | 6 | 8 | 6 | 6 | 8 |
| BWP和 | 469841 | 466794 | 475188 | 461595 | 483475 | 488127 | 492941 | 494110 |

从表5-2可以看出，采用本文所提出的参数选择方案，BWP和的最大值为499759，最优*k*值为7。而从表5-3可以看出，采用极小值作为初始中心的时候，最优*k*值为8，与本文算法的结果相差不大，但最大BWP值为498602，小于499759，分割效果略差。但是采用文献[9]的方法选择初始中心时，最优*k*值为2，相对来说，适用于二值分割，也就是分割前景和背景，并不适用与更复杂的遥感图像分割应用，并且BWP和的最大值为494110，同样小于499759。

所以，本文提出的算法能得到最大BWP和，从类间分离性和类内相似性上来说，能达到最好的分割效果。但是，本文所提出的算法在第一次聚类的时间迭代次数较多，后续的迭代次数将相对较少，跟另外两种情况相差不大。

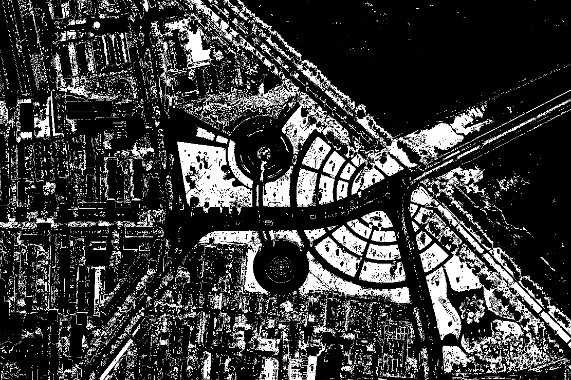


图5-7 分割聚类效果图

我们可以提取几个比较有代表性的分割块来查看分割效果：

（1） （2）



（3）

图5-8 分割块

图5-8（1）主要分离出了道路的边沿，这可以使得我们将道路和草坪等背景分割开来，像素之间连续，纹理清晰，所含的背景的比例较低。（2）分离出主干道路和部分建筑物，前景包含的比例较高，效果明显，轮廓较清晰。而（3）主要分离出草坪等背景。该分割结果相对于二值分割，能得到出更多样的分割块，这也为后续的图像处理提供了更多的信息。

## 基于RGB显著值的分割效果

对于彩色图像我们总是直观地认为根据各通道的像素信息来进行分割会更精确，文本参考了一个基于RGB显著值的方法来分割图像，并计算其分割后的BWP和，以便与文本提出的算法相对比，实验结果如下：

表5-5采用RGB显著值的分割结果

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| K值 | 7 | 6 | 5 | 4 | 3 | 2 |
| 迭代次数 | 17 | 10 | 4 | 11 | 5 | 8 |
| BWP和 | 350366 | 341568 | 332378 | 343028 | 412761 | 443381 |

从表5-5中可以看到，BWP和普遍小于本文所提出的算法的结果，最大值为443381，明显小于本文算法的最大值499759，迭代次数变化不大。并且最优*k*值为2，比较适用于前景与背景的分割，不适用于分类数较多的分割应用。图5-9为该算法的分割效果图，主要将建筑物和道路与背景分割开，与图5-7相比，分类简单，提取出的信息量较少。



图5-9 RGB显著值分割效果图

## MapReduce运行时间分析

文本对所有的计算过程都在Hadoop上进行了分布式并行处理，分布式运行时间与单机运行时间对比如表5-6所示，运算采用的是本文提出的算法，可以看出时间效率提高近一倍：

表5-6 运行时间对比

|  |  |  |
| --- | --- | --- |
| K值 | 运行时间 | |
| 单机 | Hadoop |
| 9 |  |  |
| 8 |  |  |
| 7 |  |  |
| 6 |  |  |
| 5 |  |  |
| 4 |  |  |
| 3 |  |  |
| 2 |  |  |
| 总时间 |  |  |

## 结论

综上，本文所提出的算法可以较好地解决K-means初始点选择以及最佳*k*值确定的问题，进一步完善了K-means算法，并且参数选择效果优于文献[9]所提出的方案，初始点更具有代表性。并且相对于基于RGB显著值的分割方法可以提取出更多样的分割部分，获取更多的像素信息。将BWP和作为一个衡量聚类效果的指标，避免了人为主观因素对聚类的影响，可以作为一种有效的图像分割算法。总的来说，该算法基于灰度值，增强了分割块内的相似紧密性和块与块之间的分离性。

本文还将所有的计算都进行了MapReduce编程放在了Hadoop平台上，设计了用于处理图片的自定义类ImageInputFormat，实现了直接对图像进行处理，不用讲影像数据转换为文本数据，这使得我们将更多的注意力放在核心算法处理上，避免了琐碎的图片读取过程。分布式处理图片同时这也是如今遥感影像计算的趋势，因为图像的分辨率越来越高，通道信息量越来越大，分布式计算能提高时间效率。本文实验所用的Hadoop集群共有3个节点，时间效率提升近一倍，并且能稳定运行，具有一定的扩展性，如果进一步增加处理图片的数量，我们可以将图片的像素信息存放在硬盘上进行读取，虽然处理速度有所减慢，但只要增加集群的机器数量就减小其影响。

# 总结和展望

本文设计并实现了一种基于Hadoop平台的K-means遥感图像分割算法，解决了初始点选择的问题，并且将整个计算与参数选择过程移植到了Hadoop上进行了分布式并行处理，提高了计算效率。对分割效果使用了BWP指标来衡量，减少了人为的主观性，通过各种对比实验证明了本文提出算法的合理性。

本文的主要工作如下：

1. 介绍图像分割在遥感图像处理重要性并给出定义。调研目前主要的分割算法，即阈值化、基于边缘和基于区域的算法，分析各自的优缺点，考虑其中的提升空间。重点调研K-means算法以及初始点选择方案，并结合遥感图像的特点，另外，调研了最佳K值的确定方案。
2. 介绍MapReduce编程模型的原理和执行流程，设计适合遥感图像的数据结构，使Hadoop能直接处理图像数据，并且将K-means过程和BWP过程都进行MapReduce化。
3. 设计并实现了基于Hadoop的K-means遥感图像分割算法，并做了两方面的对比实验。一是初始点的选择方案，分别采用文献[9]的方法和灰度拟合函数的极小值作为初始点来与本文的方法，即灰度拟合函数的极大值作对比，实验表明，本文的方法更有效。二是RGB显著值与灰度聚类的对比，实验表明，灰度值聚类更适合于遥感图像。
4. 分析单机处理与Hadoop处理的时间效率，实验表明，3节点的Hadoop集群能提高近一倍的时间效率。

通过各种实验表明，本文所提出的基于K-means的遥感图像分割算法分割效果较好，可以作为一种有效的分割方法，并且能很好地运行在Hadoop平台上。但同时还发现，该算法如果进一步改进，能实现更好的分割效果。未来可以从以下几个方面考虑优化：

1. 本文采用了多项式拟合灰度直方图，但是拟合效果还有改进的空间，并且本文采用拟合函数的所有极大值点作为初始点，从实验结果可以看到有些点是无用的，所以可以在K-means之前考虑删除某些无用的极大值点。
2. 可考虑将K-means算法和边缘检测算法结合起来，因为一般遥感图像中有很多像素点的灰度值是相近的，K-means会把他们分到同一类，但实际上，他们在空间上并不属于同一类，而边缘分割就可以将他们分开来。
3. 扩展性不强，本文采用的遥感影像数据较小，如果图像数据很大，计算时间可能会很长，特别是BWP的计算时间，对后续处理也有影响。所以可考虑对指标的改进。

致谢

在论文完成之际，我首先要感谢我的指导老师谢夏副教授以及同实验室的师兄师姐的帮助。这次的毕业设计是对算法的研究，是我之前没怎么接触过的，最开始有一些畏难情绪，甚至想换一个偏工程类的题目，但后来逐渐在学习过程中克服了这些情绪。

在此期间，指导老师和师兄师姐给了我很大的帮助。每周的进度汇报邮件中，指导老师会给我提相关的建议，指出不足之处，师兄师姐也帮我找了很多论文和资料，让我有了方向感。在写论文的时候，师兄也告诉了我很多规范要求，并耐心积极地指导。

经过这几个月的实践，我也深刻地认识到了自己身上存在的很多不足，比如搭建Hadoop环境的时候就遇到了很多问题，而且表现得不耐心。虽然师兄一直在协助我，但是我总是不够细心，通常一个配置的问题就会耗费我一天的时间。这明显是经验不足的问题，我以后应该多积累这方面的经验。还有就是编程效率不高，每天完成的代码量都比较少，所以以后要认真学习编程语言，努力提高自己的编程能力，同时扩展自己眼界也非常重要。但是这期间我也收获了很多，这次毕业设计我花了大量的时间在文献阅读上，我以前从没读过这么多论文，这让我提高了不少阅读文献的能力，同时也提前熟悉了实验室环境，为今后的学习和研究打下了基础。

这次毕业设计是我第一次研究算法，由于我的知识面不是很广，研究不是很深入，时间也比较仓促，所以算法肯定存在不足和待改进之处，诚心希望各位老师，同行提出宝贵意见，我一定虚心听取，有则改之，无则加勉，希望以后能有所提高。

参考文献

1. [] X.Zhao,A.Sala,C.Wilson,H.Zheng,andB.Y.Zhao,“Orion:Shortest path estimation for large social graphs,”inProc.ofWOSN,Boston,MA, June 2010. [↑](#endnote-ref-1)
2. WILSON, C., BOE, B., SALA, A., PUTTASWAMY, K. P. N., AND ZHAO, B. Y. User interactions in social networks and their implications. In Proc. of EuroSys (April 2009). [↑](#endnote-ref-2)
3. KWAK,H.,LEE, C., PARK,H., AND MOON, S. What is twitter, a social network or a news media? In Proc. of WWW (2010). [↑](#endnote-ref-3)
4. Mislove A, Marcon M, Gummadi K P, et al. Measurement and analysis of online social networks[C]//Proceedings of the 7th ACM SIGCOMM conference on Internet measurement. ACM, 2007: 29-42. [↑](#endnote-ref-4)
5. Ahn Y Y, Han S, Kwak H, et al. Analysis of topological characteristics of huge online social networking services[C]//Proceedings of the 16th international conference on World Wide Web. ACM, 2007: 835-844. [↑](#endnote-ref-5)
6. [] X.Zhao,A.Sala,C.Wilson,H.Zheng,andB.Y.Zhao,“Orion:Shortest path estimation for large social graphs,”inProc.ofWOSN,Boston,MA, June 2010. [↑](#endnote-ref-6)
7. [] Zhao X, Sala A, Zheng H, et al. Efficient shortest paths on massive social graphs[C]//Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2011 7th International Conference on. IEEE, 2011: 77-86. [↑](#endnote-ref-7)
8. [] M. Potamias, F. Bonchi, C. Castillo, and A. Gionis. Fast Shortest Path Distance Estimation in Large Networks. In CIKM’09: Proceedings of the 18th ACM Conference on Information and Knowledge Management, pages 867–876. ACM, 2009. [↑](#endnote-ref-8)
9. Qiao M, Cheng H, Chang L, et al. Approximate shortest distance computing: A query-dependent local landmark scheme[J]. Knowledge and Data Engineering, IEEE Transactions on, 2014, 26(1): 55-68. [↑](#endnote-ref-9)
10. Akiba T, Iwata Y, Yoshida Y. Fast exact shortest-path distance queries on large networks by pruned landmark labeling[C]//Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data. ACM, 2013: 349-360. [↑](#endnote-ref-10)
11. Akiba T, Iwata Y, Yoshida Y. Dynamic and historical shortest-path distance queries on large evolving networks by pruned landmark labeling[C]//Proceedings of the 23rd international conference on World wide web. ACM, 2014: 237-248. [↑](#endnote-ref-11)
12. 马明全, 周明全, 耿国华, 等. 基于社区分析的最短路径计算[J]. 计算机应用与软件, 2008, 25(4): 177-181. [↑](#endnote-ref-12)
13. 龚诗楠. 大规模复杂网络的最短路径近似算法研究[D]. 电子科技大学, 2013. [↑](#endnote-ref-13)
14. Lancichinetti A, Fortunato S. Erratum: Community detection algorithms: A comparative analysis [Phys. Rev. E 80, 056117 (2009)][J]. Physical Review E, 2014, 89(4): 049902. [↑](#endnote-ref-14)