****

**本 科 生 毕 业 论 文**

题 目： 大规模机器学习算法的并行实现

院 系： 软件学院

专 业： 软件工程（计算机应用）

学生姓名： 盘振坚

学 号： 10389441

指导教师： 潘炎（副教授）

二〇一四 年 三 月

**摘 要**

**(\* 中文摘要在500-1000字。**

在摘要中简要描述你所解决问题的背景、工程目的、工程技术与方法和最终结论。主要突出自己的工作，最后介绍论文成果是否投入使用或试用，以及应用前景如何等。关键是让读者能从摘要中看出你在论文阶段完成的主要工作。

一般使用第三人称，被动语句；避免大量背景、必要性、意义的描述，可简要提及；不要过分使用形容词；

摘要应具有独立性和自含性，即不阅读报告、论文的全文，就能获得必要的信息，摘要中有数据、有结论，是一篇完整的短文，可以独立使用，可以引用，可以用于推广。摘要的内容应包含与毕业论文同等量的主要信息，供读者确定有无必要阅读全文，也可供文摘等二次采用，摘要一般应说明研究工作的目的、实验方法、结果和最终结论等，而重点是结论和结果。

除非确实需要，否则在摘要中一般不要用图表、化学结构式、非公知公用的符号和术语。\*)

**关键词：** (\* 关键词1；关键词2；……

关键词一般为论文中出现的重要的、描述主题的词；

中英文关键词不少于三个，不多于七个。\*)

Abstract

(\* 英文摘要不少于500单词，严格使用英文标点符号。中、英文摘要意思要基本相同。在撰写完英文摘要后，请执行一次自动拼写检查，以减少英文拼写错误的可能性。\*)

**Keywords:**  (\* Keywords 1; Keywords 2; ... ...

中英文关键词要严格对译。\*)

目 录

(\* 目录放在正文之前，中英文摘要之后。目录至少有二级目录，即包含正文的章和节及其页码；最好有三级目录，即包含正文的章、节和小节及其页码。通常工程文档层次比较多，论文不应该采用四节标题，可改用1,2…小标题。目录还要包含参考文献及其页码。中英文摘要及目录本身及其页码可不出现在目录中。请使用由Word等排版软件自动生成目录，不要自己手工做目录。\*)

第一章 前言 1

第二章 技术与原理 2

2.1 CART 2

2.1.1 特征选择 3

2.1.2 树的生成 3

2.2 Boosting 3

2.2.1 Boosting模型 4

2.2.2 Boosting算法 4

2.3 MPI 4

2.3.1 MPI的简单使用 5

2.3.2 MPI的简单通信 6

2.4 ××××××× 7

2.5 ××××× 7

第三章 Boosting算法实现与模块设计 8

3.1 整体框架 8

3.1.1 数据读取模块 8

3.1.2 特征评估模块 9

3.1.3 训练与预测模块 9

3.2 数据读取模块 10

3.2.1 TR\_Data的.csv数据读入 10

3.2.2 理 10

3.3 特征评估模块 10

3.3.1 VAR\_Measurer的measure()内部原理 10

3.3.2 ××××× 10

3.4 训练与预测模块 11

3.4.1 CART\_Predictor建树过程 11

3.4.2 Boosting\_Predictor训练过程 11

第四章 并行化 12

4.1 ×××××× 12

4.2 ××××× 12

4.2.1 ×××× 12

4.2.2 ××××× 12

4.3 ×××× 12

4.3.1 ×××× 12

4.3.2 ×××××× 12

4.4 ×××××× 12

4.4.1 ××× 13

4.4.2 ××× 13

4.4.3 ×××× 13

4.5 ×××××× 13

4.5.1 ××× 13

4.5.2 ××××× 13

4.6 ×××××× 13

第五章 模块设计 14

5.1 14

5.2 14

第六章 部署与应用 15

6.1 15

6.2 15

第七章 结论 16

致谢 17

参考文献 18

附录 20

# 前言

(\* 正文不少于10千字；或使用小四字体、1.5倍行距、A4纸版式排版时不少于10页纸。正文须有页码，从第1页开始编页码。正文采用章、节、小节组织。章的标题使用“第一章”等字样开头，节的标题采用“1.1”等字样开头，表示第一章的第一节，小节的标题采用“1.1.1”等字样开头，表示第一章的第1.1小节。正文章、节、小节标题与正文段落使用不同的字体，并且之间有适当的间距。正文段落要统一缩进两个汉字。论文主体的第一部分通常是相关领域综述。

本科生工程论文应该注重标准与规范，首先要注意以下问题：（1）工程论文不是软件文档的堆砌；（2）不应包含代码、大量界面等细节。

建议论文选择一个方向最为重点内容：（1）问题建模、设计模型；（2）软件开发过程中实践某过程、技术或方法的实践，需要有流程、测量等，如基于XX方法的质量控制实践。

\*)

(\*本章建议结构

1.1 项目的背景和意义（阐明项目来源、开发系统的动机、系统研究开发的意义等）；

1.2 研究开发现状分析（领域分析。通过对来自问题领域、技术领域和两个领域的产品开发和技术运用现状的分类、分析，总结现有成果的优缺点，在此基础上，阐明项目研究开发的必要性）；

1.3 项目的目标和范围（确立项目的目标、要解决的关键问题和范围，可通过系统地描述所有外部对象与系统间的输入/输出来界定范围）；

1.4 论文结构简介（简单介绍论文各章的主要内容）。

\*)

# 技术与原理

(\* 论文主体的第二部分主要介绍自己的工作。这是论文最主要的部分，在这一部分表达你自己在论文部分的工作，而不像第一部分综述那样是介绍别人的工作。千万不要大量列举你的源代码（部分核心算法可使用伪码或流程图等形式表示），否则给人的感觉是你的论文没有内容而靠这些来充数。\*)

(\* 介绍项目中将要用到的主要技术或理论（2-4个技术点或原理方法）。这些原理、方法、图表应来源于经典工程教科书、官方技术网站、国内核心期刊、ACM/IEEE论文与期刊。

注意：第一、二章是引用文献最集中的章节，但篇幅不要超过全文的1/3，后续章节尽量不要再出现大段引用的现象；

\*)

1. CART
2. Boosting
3. MPI

## CART

CART（classification and regression tree**[1]**, 分类与回归树），该模型由Breiman等人在1984年提出，是应用广泛的决策树学习方法。CART决策树算法主要分成三个步骤：特征选择，树的生成，剪枝。

CART决策树算法既可以用于分类问题，也可以用于回归问题。CART是在给定输入随机变量X条件下输出随机变量Y的条件概率分布的学习方法。CART假设决策树是二叉树，内部结点特征的取值为“是”和“否”，左分支是取值为“是”的分支，右分支是取值为“否”的分支。这样的决策树等价于递归地二分每个特征，将输入空间即特征空间划分为有限个单元，并在这些单元上确定预测的概率分布，也就是在输入特定的条件下输出的条件概率分布**[2]**。

在使用决策树学习方法时，绝不能忽略的问题是——如何避免过拟合。CART决策树方法采用的剪枝过程来避免过拟合。在本论文中，使用Boosting提升方法来取代CART决策树的剪枝过程来避免过拟合。Boosting提升方法会在下一节（2.2节）再作介绍，接下来，介绍一下本论文所涉及的CART决策树的两个过程：特征选择和树的生成。

### 特征选择

CART决策树方法的特征选择过程，就是选择哪一个特征作为决策结点的过程。特征选择在于选取对训练数据具有分类能力的特征，这样可以提高决策树学习的效率。

CART在处理分类问题和回归问题中，对于每一次最优特征的选取，采用的是不同的评判准则：分类问题中采用基尼指数（Gimi index）最小化，回归问题采用平方误差（Square error）最小化。

### 树的生成

## Boosting

Boosting，又名提升方法，是一种常用的统计学习方法，应用广泛且有效，在分类问题中，它通过改变训练样本的权重，学习多个分类器，并将这些分类器进行线性组合，提高分类的性能**[2]**。

提升方法基于这样一种思想：对于一个复杂任务来说，将多个专家的判断进行适当的综合所得出的判断，要比其中任何一个专家单独的判断好。实际上，就是“三个臭皮匠顶个诸葛亮”的道理。

Boosting是要把多个弱学习器组合成一个强学习器，即把已有的“弱学习算法”提升（boost）为“强学习算法”。“弱学习算法”与“强学习算法”来源与Kearns和Valiant所提出的两个概念：“弱可学习（weakly learning）”和“强可学习（strongly learnable）”。在概率近似正确（probably approximately correct，PAC）学习的框架中，一个概念（一个类），如果存在一个多项式的学习算法能够学习它，并且正确率很高，那么就称这个概念是强可学习的；一个概念，如果存在一个多项式的学习算法能够学习它，学习的正确率仅比随机猜测略好，那么就称这个概念时弱可学习的。后来Schapire证明强可学习与弱可学习时等价的，也就是说，在PAC学习框架下，一个概念时强可学习的充分必要条件是这个概念时弱可学习的**[2]**。以上的证明为boosting的可行性提供理论依据。

### Boosting模型

Boosting提升方法实际采用加法模型（即基函数的线性组合）与前向分步算法。在本论文中，采用CART作为boosting的基函数

(2.1)

其中，为CART的参数；M为基函数CART的个数。

### Boosting算法

Boosting算法采用前分步算法。初始基函数，第m步的模型是

(2.2)

其中，为当前模型，通过最小化误差L()来确定下一个基函数的参数，

(2.3)

## MPI

MPI (Message Passing Interface) 是一个跨语言的通讯协议，用于编写并行计算算法，支持点对点和广播；是一个消息传递应用程序接口，包括协议和语义说明，他们指明如何在各种实现中发挥其特性。MPI的目标是高性能，大规模性，和可移植性。

同时，MPI也是一个由众多并行计算机厂商、软件开发单位/组织、并行应用单位等共同维护的标准，自1994年发布以来，其标准经历了一次版本升级，从MPI-1升到了MPI-2。MPI-1实现了基本的消息通信操作，组通信操作，虚拟拓扑、可拓展数据类型管理等。MPI-2增加了外部接口，拓张的组通信操作，并行I/O，单向通信，进程管理，线程控制等，增加了域间通信能力**[4]**。

本论文所实现的Boosting并行算法通过C++语言调用MPI接口，所采用MPI实现版本是openmpi，运行环境是ubuntu，可以通过apt-get配置openmpi并行计算开发环境。

### MPI的简单使用

Hello World

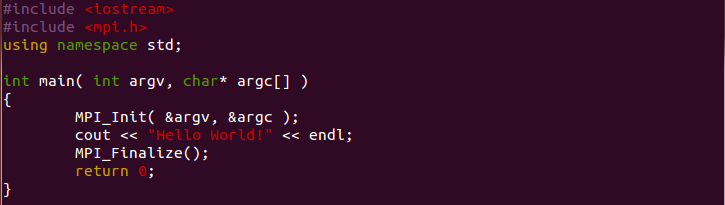


图2-1：Hello World代码实例

如上代码实例，在MPI\_Init()与MPI\_Finalize()之间的代码会被并行化，实际上，这两者之间的代码会被复制多份，并由多个线程同时执行。

键入以下指令：

mpicxx hello.cpp –o hello

编译hello.cpp

mpirun –n 4 hello

运行hello，其中，参数 -n 4 代表开始4个线程运行hello，输出如下

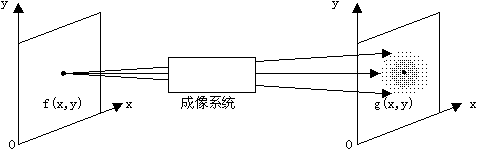


图2-2：Hello World输出样例

### MPI的简单通信

(\* 正文中有图、表和公式时必须对图、表和公式按章进行编号。例如，下图编号“图2-1”表示第二章第1个图；或者，在整篇正文中，图和表按出现的顺序依次编号，如“图1”、“图2”、“表1”、“表2”等。

图和表必须有标题（如图2-1中的“成像系统原理”），图的编号和标题放在图的下方，表的编号和标题放在表的上方。所有的图和表都必须在正文中引用，而且引用时不得采用“如下图”或“如下表”等字样，而要采用“如图2-1”或“如表2-1”等字样。排版时，可根据排版的美观性适当调整图和表在正文中的放置位置。图表中文字采用5号字。



**图2-1：成像系统原理**

**表2-1： 表的样例**

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

## ×××××××

×××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××**[1]**，此蓝色方括号上标是参考文献的引用，方括号内的数字指明前面这段文字引自第几个参考文献。

××××：

列举的公式也必须按章进行编号，例如公式2.1表示第二章第1个公式。

×××××××[2]。（引用参考文献2）

××××××××[[1]](#footnote-1)

## ×××××

# 模块设计

本论文采用C++语言实现Boosting算法的并行化，本章介绍Boosting算法的模块设计。

## 整体框架

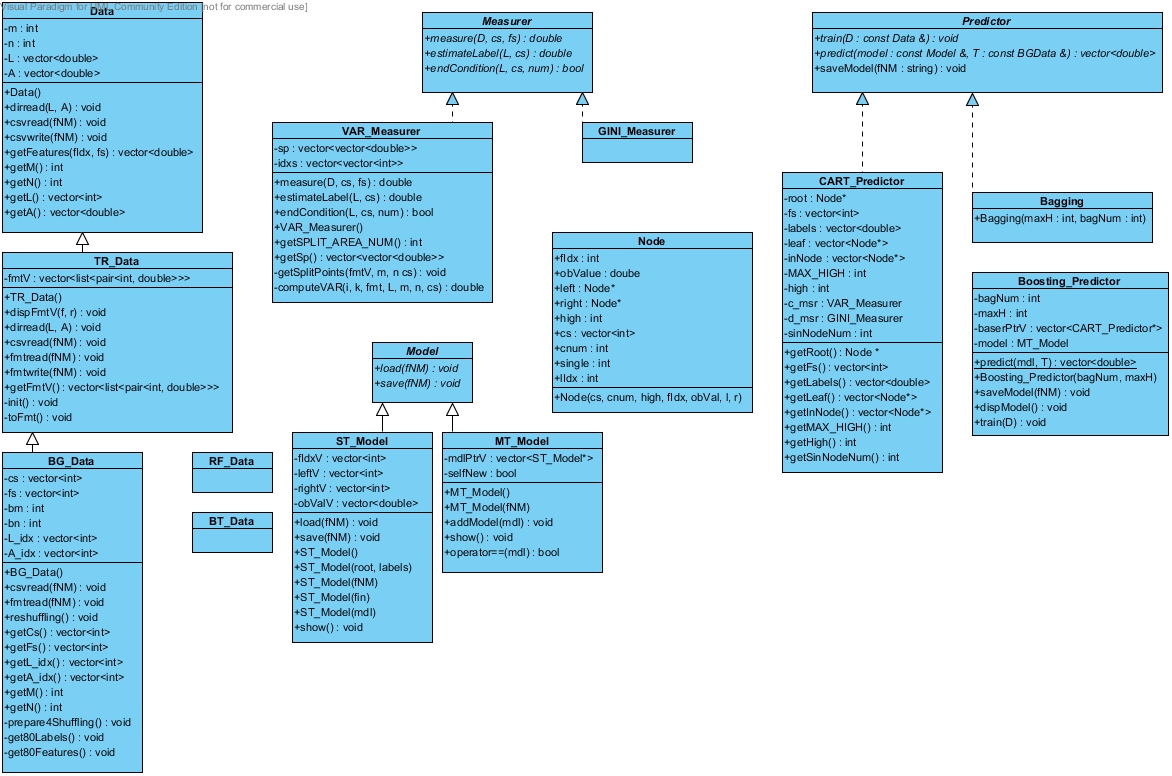


图3-1：Boosting单线程版整体框架

Boosting算法的单线程版实现主要分成三大模块：数据读取模块，特征评估模块，训练与预测器模块。

## 数据读取模块

这一部分的职责主要有3个：

1. 从文本读入并保存数据，文本可以使通用的.csv文件，也可以是方便树状模型计算的数据集格式.fmt文件。
2. 对读进来的数据进行适当的处理，方便作为训练机的输入。
3. 把数据写入文件，同时也可以完成从.csv到.fmt文件的转换

该模块主要有两层继承关系，Data和TR\_Data;

其中，Data是基类，实现.csv文件数据的读写功能，TR\_Data类拓展了Data类，采用了一种更加适合树状模型的存储格式.fmt，并对格式提供了相应的与.csv转换方法。

## 特征评估模块

特征评估模块有两个主要的特征评估器，VAR\_Measurer和GINI\_Measurer，前者用于解决回归问题时的特征评估，后者用于解决分类问题时的特征评估；二者都继承于纯虚类Measurer，纯虚类Measurer定义了三个纯虚方法：measure()，estimateLabel()，endCondition()。

measure()：特征评估的主要功能函数，大部分计算量的核心模块所在，同时也是下一步要并行化以降低时间复杂度的地方，其返回被评估特征的一个定量的数据结构MS，见图3-2，fIdx是特征下标；obVal是切分点；msVal是特征选取的衡量值，如回归问题使用VAR（方差），分类问题使用GINI值，二者都是对一个特征被选取作为本次建树的结点是否恰当的一个定量的评估。

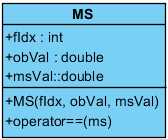


图3-2：MS类

estimateLabel()：建树完成之后，对叶子预测值的一个计算，如回归问题是被分到该叶子节点上所有样本所对应预测值的均值，分类问题是取被分到该叶子节点上比例最高的那一个类别。

endCondition()：建树过程中，判断一个节点是否应该作为叶子节点的条件。

## 训练与预测模块

该模块是整个Boosting单线程版算法的核心，该模块里面有两个主要的类CART\_Predictor和Boosting\_Predictor，分别是Boosting算法的基本学习机和主要训练预测机。为了该项目的可拓展性，二者都继承自纯虚基类Predictor，该基类定义了三个纯虚函数：train()，predict()，saveModel()，其中predict()是静态函数，只要有了model（模型），就可以直接通过类名调用predict()方法对数据进行预测。

train()：用来通过训练集训练出用来预测的model（模型），该过程比较耗时，model（模型）一旦训练完成，可以重复使用。这里的model（模型）定义了一个Model类来存在相关的信息与对应的方法，见图3-3。

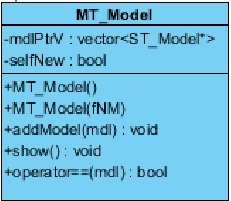


图3-3：Model类

predict()：利用训练出来的model（模型）来对数据进行预测

saveModel()：保存训练出来的model（模型）到文件，文件后缀为.mdl。

# 单线程版实现

本章首先介绍Boosting算法的基本思路，通过其单线程版的实现来从细节上介绍Boosting的每一实现步骤。接下来，举个简单的例子来说明一下特征评估选取过程。

现有如下小型训练数据集（5个样本，每个样本4个特征），储存在.csv文件：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | L | F1 | F2 | F3 | F4 |
| C1 | 1 | 0 | 0 | 0.5 | 0.4 |
| C2 | 1 | 0 | 0 | 0.5 | 0.5 |
| C3 | 1 | 0.5 | 0 | 0.5 | 0 |
| C4 | 2 | 0.7 | 0.5 | 0.5 | 0.8 |
| C5 | 2 | 0.9 | 0.5 | 0.5 | 0.9 |

图3-4：事例数据集

## Boosting算法整体流程

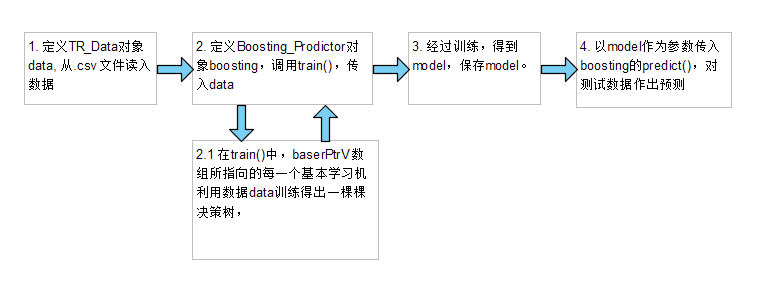


图3-5：Boosting算法整体流程

## 第一步，数据读取

## 第二步，训练

## 第三步，保存Model

## 第四步，预测

详细介绍一下VAR\_Measurer的内部详细结构

class VAR\_Measurer**:** public Measurer

**{**

public**:**

// implement virtual methods

MS measure**(** const TR\_Data **&**D**,** const vector**<**int**>** **&**cs**,** const vector**<**int**>** **&**fs **);**

double estimateLabel**(** const vector**<**double**>** **&**L**,** const vector**<**int**>** **&**cs **);**

bool endCondition**(** const vector**<**double**>** **&**L**,** vector**<**int**>** cs**,** int num **);**

// own methods

VAR\_Measurer**();** // constructor for accurater

// getter

vector**<** vector**<**double**>** **>** getSp**()** const**;**

vector**<** vector**<**int**>** **>** getIdxs**()** const**;**

vector**<** vector**<**double**>** **>** getVars**()** const**;**

// single getter

const vector**<**int**>&** getPart1**()** const**;**

const vector**<**int**>&** getPart2**()** const**;**

int getNum1**(** int i**=-**1**,** int k**=-**1 **)** const**;**

int getNum2**(** int i**=-**1**,** int k**=-**1 **)** const**;**

private**:**

void getSplitPoints**(**vector**<**list**<**pair**<**int**,**double**>>>&** fmtV**,** vector**<**double**>&** L**,** int m**,** int n**,** vector**<**int**>** cs **);**

// will update every loop

vector**<** vector**<**double**>** **>** sp**;** // just use to test

vector**<** vector**<**int**>** **>** idxs**;** // record num for each sp

vector**<** vector**<**double**>** **>** vars**;** // for variance calculate

vector**<**int**>** part1**;**

vector**<**int**>** part2**;**

// the total value

double sums**;**

double sqSums**;**

int nums**;**

// the mid cal-value for efficiency

vector**<** vector**<**double**>** **>** sums1**;**

vector**<** vector**<**double**>** **>** sqSums1**;**

vector**<** vector**<**int**>** **>** num1s**;**

// res idx

int fIdx**;**

int spIdx**;**

double obVal**;**

**};**

图3-3：VAR\_Measurer的内部详细结构

在VAR\_Measurer内部，采用了一些技巧来提高时间的效率。众所周知，对回归问题时，建决策树过程是根据选取VAR最小的一组<特征，切分点>对来作为决策树上的一个结点的。每一组<特征，切分点>的VAR的计算公式是：

(3.1)

其中f是特征，s是切分点，M是样本总数，part1，part2是样本被切分点分成的两部分，m是part1的样本数，M-m是part2的样本数，v(part1)是计算part1样本的方差。

此处方差的计算公式是

(3.2)

其中M是样本数，x是X中的一个样本，是X的均值。

这个方差计算公式平方后再求和，其中存在着一些重复的计算，现在对这个公式进行处理，有

(3.3)

易知，(3.2)和(3.3)是等价的，

### ×××××

这里是整个算法的计算量所在，也是整个算法的关键所在，

## 训练与预测模块

### CART\_Predictor建树过程

### Boosting\_Predictor训练过程

我的层次关系

## ×××××

### ××××

### ×××××

## ××××

××

### ××××

### ××××××

## ××××××

××××××××

### ×××

×××

### ×××

### ××××

## ××××××

### ×××

### ×××××

## ××××××

# 并行化

（\*

选择2-3个模块。

只需描述主要类的说明及接口、以及主要类关键方法的算法。

界面设计，对象协作图，算法等。

算法描述可使用活动图、程序流程图或其他传统详细设计方法。

\*）

交互接口

## 

## 

# 部署与应用

（\*

1）描述系统运行环境，可以使用UML的部署图；

2）选择若干典型界面说明系统的重要输入/输出；

3）与已有系统进行多方面的对比说明先进性或特色，或与本系统投入使用前状况对比说明应用效果。

\*）

使用接口

## 

## 

# 结论

(\* 论文第三部分主体：总结、比较与展望。这一部分要总结自己工作的优点在哪里、不足之处在哪里、进一步的改进方案如何等。如果有类似应用背景的系统，还须比较你的工作与他人工作的优、劣。 \*)

# 致谢

(\* 致谢，单独占一页。对在论文工作中对你有支持、有帮助的单位或个人表示感谢。

模版在信科院论文，余阳老师工程论文指导基础上整理而成。使用者表示衷心的感谢！

\*)

# 参考文献

(\* 列出你在完成论文过程主要参阅的论文与著作。其中的篇数不宜太少，否则让人感觉到你没有参考相关工作就动手做论文；一篇本科毕业论文的典型参考文献是15至30篇，如果你所列参考文献数目不足5篇，则肯定无法通过论文初审的形式检查。列举参考文献时，按论文中引用文献的先后顺序列于此处；注意正文中必须引用此处所列的全部参考文献，而且引用顺序就是参考文献的列举顺序。参考文献的列举格式如下（也可参考《计算机学报》、《软件学报》、《电子学报》、《自动化学报》等刊物发表的论文所列参考文献的格式）：

[编号] 作者，书名，出版社，年份

[编号] 作者，论文名称，刊物名称，卷(期): 起始页-终止页，年份

[编号] 作者，论文名称，见: 会议论文集名称，起始页-终止页，年份

请参照以下格式书写（下面：[1][2]是书籍、[3][4]是期刊论文、[5]是会议论文）\*)

1. L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. Classification and Regression Trees. Wadsworth Advanced Books and Software, Belmont, CA, 1984
2. 李航，统计学习方法，北京：清华大学出版社，2012
3. 都志辉，高性能计算之并行编程技术——MPI并行程序设计，北京：清华大学出版社，2001
4. 张武生、薛巍、李建江、郑伟民，MPI并行程序设计实例教程，北京：清华大学出版社，2009
5. 袁崇义，Petri网原理，北京：电子工业出版社，1998
6. Bloch J.，Effective Java: Programming Language Guide，Addison Wesley，2001
7. 李建中、李金宝、石胜飞，传感器网络及其数据管理的概念、问题与进展，软件学报，14(10): 1717-1727，2003
8. Goguen J. A. , Parameterized Programming. IEEE Transactions on Software Engineering, 10(5): 528-543, 1984
9. Snyder A. , Encapsulation and Inheritance in Object-Oriented Programming Languages, In: Proceedings of Annual ACM SIGPLAN Conferences on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA’86), 38-45, 1986
10. … …

# 附录

(\* 附录是正文主体的补充。下列内容可以作为附录：  
1）攻读学位期间发表的（含已录用，并有录用通知书的）与学位论文相关的学术论文。  
2）由于篇幅过大，或取材于复制件不便编入正文的材料、数据。  
3）对本专业同行有参考价值，但对一般读者不必阅读的材料。  
4）论文中使用的符号意义、单位缩写、计算机程序全文及有关说明书。  
5）附件：光盘、与论文相关的鉴定证书、获奖奖状或专利证书的复印件等。 \*)

毕业论文成绩评定记录

|  |
| --- |
| 指导教师评语：  成绩评定：    指导教师签名： 年 月 日 |
| 答辩小组或专业负责人意见：  成绩评定：    签名（章）： 年 月 日 |
| 院系负责人意见：  成绩评定：  签名（章）： 年 月 日 |

**附表一、毕业论文开题报告**

|  |
| --- |
| 论文（设计）题目： |
| （简述选题的目的、思路、方法、相关支持条件及进度安排等）  学生签名： 年 月 日 |
| 指导教师意见：  1、同意开题（ ） 2、修改后开题（ ） 3、重新开题（ ）    指导教师签名： 年 月 日 |

**附表二、毕业论文过程检查情况记录表**

|  |  |
| --- | --- |
| 指导教师分阶段检查论文的进展情况（要求过程检查记录不少于3次）：  **第1次检查**  学生总结：  指导教师意见：  **第2次检查**  学生总结：  指导教师意见：  **第3次检查**  学生总结：  指导教师意见：  **第4次检查**  学生总结：  指导教师意见：  学生签名： 年 月 日  指导教师签名： 年 月 日 | |
| **总体**  **完**  **成**  **情**  **况** | 指导教师意见：  1、按计划完成，完成情况优（ ）  2、按计划完成，完成情况良（ ）  3、基本按计划完成，完成情况合格（ ）  4、完成情况不合格（ ）  指导教师签名： 年 月 日 |

**附表三、毕业论文答辩情况**

|  |  |  |  |
| --- | --- | --- | --- |
| 答辩人 |  | 专 业 |  |
| 论文（设计）题目 |  | | |
| 答辩小组成员 |  | | |
| 答辩记录：  记录人签名： 年 月 日 | | | |

**学术诚信声明**

本人所呈交的毕业论文，是在导师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料均真实可靠。除文中已经注明引用的内容外，本论文不包含任何其他人或集体已经发表或撰写过的作品或成果。对本论文的研究作出重要贡献的个人和集体，均已在文中以明确的方式标明。本毕业论文的知识产权归属于培养单位。本人完全意识到本声明的法律结果由本人承担。

本人签名： 日期：

1. (\* 注释，为你的某段文字提供解释，如表数据的来源等 \*) [↑](#footnote-ref-1)