基于android的物流配送平台

姓名：鲍立

指导老师：王东教授

学院：信息科学与工程学院

**摘要**

时代飞速的发展，随之而来的是人们生活方式变得日益便捷。现在，不管是衣食住行等各个方面，人们的需求都能随时随地、十分方便地得到满足。不论是网购、日常点外卖、还是实际购物中，各类商场和商店都提供了送货上门服务。随着配送业务的迅速发展以及需求量的逐渐加大，如何帮助配送人员提高配送的效率便成为了一个很重要的话题。

因而，笔者认为可以通过设计一个物流配送系统来提高配送人员在配送物品的效率，这样能极大地缩短配送时间，同时更能满足用户的需求。本课题主要完成了以下几个方面的工作：

1. 基于android平台以及百度地图开放平台等，开发了可用于物流配送的app，该系统能够提供定位、路线导航等基本功能，同时结合本文提出的交通流拥堵状态预测方法，可以作为路线建议的依据；
2. 对实际交通数据获得拥堵状态的评价方法。首先使用了数据挖掘过程中比较高效的K-means聚类算法，针对一般的K-means算法对初始簇中心严重依赖、聚类结果较不稳定的问题，提出用类似于二分法的思路来获取聚类的初始化聚类中心，最终依据聚类结果得出不同交通拥堵状态的划分数值。拥堵状态是交通状态分析的重点，通过聚类方法从大量交通数据中处理得到该场景不同状态对应的划分数值，显得更加灵活和可信；
3. 对短时交通拥堵状态进行预测的过程当中，针对常见BP神经网络模型直接预测交通拥堵状态时存在有些情况预测结果不够准确的问题，提出了在神经网络的预测阶段使用BP神经网络结合统计学理论进行联合预测得到最终预测模型的短时交通拥堵状态预测方法。直接通过神经网络模型的预测正确率为95.1%，而在预测阶段结合统计学方法联合校验得到最终预测模型后进行预测的正确率为97.5%。该方法对交通拥堵状态的预测的正确率经试验结果对比证明有了一定的提升。

**关键词：百度地图；交通拥堵状态评价；短时交通拥堵预测；神经网络；统计方法**

**Abstract**

**目录**

摘要..................................................................................................

Abstract............................................................................................

第1章 绪论...................................................................................

1.1选题背景........................................................................

1.2研究的目的和意义........................................................

1.3国内外研究现状............................................................

1.4研究内容和论文结构安排............................................

第2章 关键技术研究....................................................................

2.1 Android系统...................................................................

2.2 Android开发平台简介................................. ..................

2.3百度地图开放平台.........................................................

2.4聚类方法.........................................................................

2.5 神经网络预测及优化.....................................................

2.6 本章小结.........................................................................

第3章 系统需求分析.......................................................................

3.1系统业务需求.................................................................

3.2系统功能需求.................................................................

3.3系统非功能性需求...........................................................

3.4物流配送系统总体设计....................................................

3.5本章小结............................................................................

第4章 系统详细设计及实现...........................................................

第5章 总结与展望...........................................................................

参考文献............................................................................................

致谢....................................................................................................

1. **绪论**

**第1章 选题背景**

* 1. **研究的目的和意义**
  2. **国内外研究现状**
  3. **研究内容和论文结构安排**

1. **关键技术研究**

**2.1 Android系统**

**2.1.1 Android系统介绍**

Android是一种基于[Linux](https://baike.baidu.com/item/Linux" \t "https://baike.baidu.com/item/Android/_blank)的自由及开放源代码的[操作系统](https://baike.baidu.com/item/%E6%93%8D%E4%BD%9C%E7%B3%BB%E7%BB%9F/192" \t "https://baike.baidu.com/item/Android/_blank)，主要使用于[移动设备](https://baike.baidu.com/item/%E7%A7%BB%E5%8A%A8%E8%AE%BE%E5%A4%87/9157757" \t "https://baike.baidu.com/item/Android/_blank)，如[智能手机](https://baike.baidu.com/item/%E6%99%BA%E8%83%BD%E6%89%8B%E6%9C%BA/94396" \t "https://baike.baidu.com/item/Android/_blank)和[平板电脑](https://baike.baidu.com/item/%E5%B9%B3%E6%9D%BF%E7%94%B5%E8%84%91/1348389" \t "https://baike.baidu.com/item/Android/_blank)，由[Google](https://baike.baidu.com/item/Google" \t "https://baike.baidu.com/item/Android/_blank)公司和[开放手机联盟](https://baike.baidu.com/item/%E5%BC%80%E6%94%BE%E6%89%8B%E6%9C%BA%E8%81%94%E7%9B%9F/9064338" \t "https://baike.baidu.com/item/Android/_blank)领导及开发。尚未有统一中文名称，中国大陆地区较多人使用“[安卓](https://baike.baidu.com/item/%E5%AE%89%E5%8D%93/5389782" \t "https://baike.baidu.com/item/Android/_blank)”或“[安致](https://baike.baidu.com/item/%E5%AE%89%E8%87%B4/6164281" \t "https://baike.baidu.com/item/Android/_blank)”。

Android操作系统最初由[Andy Rubin](https://baike.baidu.com/item/Andy Rubin" \t "https://baike.baidu.com/item/Android/_blank)开发，主要支持[手机](https://baike.baidu.com/item/%E6%89%8B%E6%9C%BA/6342" \t "https://baike.baidu.com/item/Android/_blank)。2005年8月由Google收购注资。2007年11月，Google与84家硬件制造商、软件开发商及电信营运商组建开放手机联盟共同研发改良Android系统。随后Google以Apache开源许可证的授权方式，发布了Android的源代码。第一部Android智能手机发布于2008年10月。Android逐渐扩展到[平板电脑](https://baike.baidu.com/item/%E5%B9%B3%E6%9D%BF%E7%94%B5%E8%84%91/1348389" \t "https://baike.baidu.com/item/Android/_blank)及其他领域上，如[电视](https://baike.baidu.com/item/%E7%94%B5%E8%A7%86/228945" \t "https://baike.baidu.com/item/Android/_blank)、[数码相机](https://baike.baidu.com/item/%E6%95%B0%E7%A0%81%E7%9B%B8%E6%9C%BA/277472" \t "https://baike.baidu.com/item/Android/_blank)、[游戏机](https://baike.baidu.com/item/%E6%B8%B8%E6%88%8F%E6%9C%BA/315328" \t "https://baike.baidu.com/item/Android/_blank)、[智能手表](https://baike.baidu.com/item/%E6%99%BA%E8%83%BD%E6%89%8B%E8%A1%A8/71070" \t "https://baike.baidu.com/item/Android/_blank)等。2011年第一季度，Android在全球的市场份额首次超过[塞班系统](https://baike.baidu.com/item/%E5%A1%9E%E7%8F%AD%E7%B3%BB%E7%BB%9F/8506777" \t "https://baike.baidu.com/item/Android/_blank)，跃居全球第一。 2013年的第四季度，Android平台手机的全球市场份额已经达到78.1%。2013年09月24日谷歌开发的操作系统Android在迎来了5岁生日，全世界采用这款系统的设备数量已经达到10亿台。

2014第一季度Android平台已占所有移动广告流量来源的42.8%，首度[超越](https://baike.baidu.com/item/%E8%B6%85%E8%B6%8A/80866" \t "https://baike.baidu.com/item/Android/_blank)iOS。但运营收入不及iOS。

**2.1.2 Android系统平台的架构**

Android系统总的来说可以分成四层架构：Linux内核层、系统运行库层、应用框架层和应用层。

1. Linux内核层

Android系统是基于Linux内核的，也就是说，Android系统的底层基于Linux系统之上。这一层为Android设备的各种硬件提供了底层的驱动，如显示驱动、音频驱动、照相机驱动、蓝牙驱动、Wi-Fi驱动、电源管理等。

但同时Android对于Linux进行了改动，比如它不包含glibc，原先用于一些便携的移动设备并没有采用glibc作为c库，而是goolge自己开发的Bionic Libc来作为代替品等，同时还增强显示驱动、键盘驱动、Flash内存驱动、照相机驱动、音频驱动、蓝牙驱动、Wi-Fi驱动、Binder IPC驱动、Power Management（电源管理），包括硬件时钟，内存分配和共享，低内存管理，kernel调试，日志设备，android IPC机制，电源管理等。

1. 系统运行库层

这一层通过一些C/C++库来为Android系统提供了主要的特性支持。如SQLite库提供了数据库的支持，OpenGL|ES库提供了3D绘图的支持，Webkit库提供了浏览器内核的支持等。

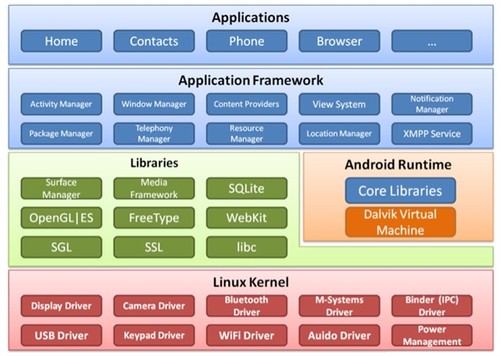
同样在这一层还有Android运行时库提供了一些核心库，能够允许开发者使用Java语言来编写Android应用。还包含了虚拟机Dalvik但之后对ART运行环境进行了部分改变，使得每一个Android应用都有自己的进程，并且都有一个属于自己的Dalvik虚拟机实例，相较于JAVA的虚拟机Dalvik是专门为移动设备定制的，针对内存和CPU性能等方面都有了优化。

1. 应用框架层

这一层主要提供了构建应用程序时可能用到的各种API，Android自带的一些核心应用就是基于这些API完成的。那什么是应用程序框架层？它是一个应用程序的核心，是一个共同使用和遵守的约定，然后在这个约定上共同扩展，但程序保持主体结构的一致，其作用是让程序保持清晰在满足不同需求的同时又不互相影响。而对于Android来说提供给应用开发者的本身就是一个框架，所有的应用开发都必须遵守这个框架的原则，同时在这个基础上进行扩展可以访问核心应用程序所使用的API框架。

1. 应用层

所有安装在手机上的应用程序都是属于这一层的，比如系统自带的联系人、短信等程序，或者从Google Play上下载的小程序，当然还包括自己开发的应用程序。



Android系统架构

**2.1.3 Android系统的优势**

Android系统是一种基于Linux的开源操作系统。从2007年11月诞生以来，Android已经走过了12个年头。在过去的12年里，Android系统发展迅速，并且一度霸占了世界上绝大部分电子设备。Android系统因为开放性、挣脱束缚、丰富的硬件等优势在中国占据越来越多的市场份额。Android系统具备以下几个显著的优点：

1. 开放性。Android的这种开放性允许任何移动终端厂商加入Android，从而可以使其拥有众多的开发者。随着用户资源和应用的日益丰富，一个新平台便会得到迅速的发展。而对于消费者这边而言，最大的受益之处是可以获取到丰富的应用资源。平台的开放同时也意味着更加激烈的竞争，如此一来，消费者便能购买到物美价廉的移动设备。
2. 挣脱运营商的束缚。在过去很长的一段时间里，尤其是在欧美等地区，手机应用等通常会受到运营商等制约，使用什么功能，接入哪种网络等，几乎完全受到运营商的控制。随着3G逐渐过渡到4G网络，手机网络的接入模式已经不再受到运营商的限制。然而，Android系统自出现就具备网络方面的特色，使得手机用户距离互联网更近。
3. 各种丰富的硬件可以选择。Android系统所具有的开放性，使得众多的手机硬件厂商会推出各种具有特色的产品。功能的不同，并不会影响到数据方面的同步，甚至是软件的兼容。
4. 不受任何拘束的开发商。Android平台给第三方开发商提供了一个十分宽泛、自由的开发环境，不会受到各种条条框框的阻扰，给第三方开发商提供及其灵活的开发环境，这样便容易出现更多优秀的应用软件。
5. 联系紧密的Google应用。如今Google已经成立20多年，从搜索引擎到移动互联网的全面渗透，Google推出的一系列服务已然成为了连接用户和互联网的重要纽带，基于Android系统的移动设备也可以无缝连接到Google提供的各种服务。

**2.2 android开发平台简介**

**2.2.1、[Android开发环境中的概念和工具介绍](https://www.cnblogs.com/xiaochao1234/p/3833558.html)**

什么是Android开发环境？Android开发环境当然是用来设计应用于移动设备的系统和软件了，由于Android本身是开源的，那么我们要做的工作就是对其进行定制以及修改，Android手机厂家工作亦是如此，定制自己的驱动（或称BSP，底层）和应用程序（上层）。

开发语言可以使用Java也可以使用C/C++语言，前者Android开发我们称做JDK（Java Development Kit）开发，后者开发我们称之为NDK（Native Development Kit）开发。这么看来的话，Android本身就是C/C++和Java的混合体，所以要学习Android编程，C语言是个基础，Java即使不会也能猜出一二，做个DEMO肯定没有问题。

Android开发环境可以搭建在目前主流系统（Mac、Windows、Linux）的任何一种上，只因为Android相关开发工具的跨平台特性。

1. Java Development Kit（JDK）

Java Development(JDK)是用于开发、编译和测试使用Java语言编写的应用程序、applet 和组件，JDK包含以下几个部分：

开发工具——指工具和实用程序，可帮助您开发、执行、调试和保存以 Java编程语言编写的程序。

运行时环境——由 JDK 使用的 Java Runtime Environment (JRE) 的实现。JRE 包括 Java 虚拟机 (JVM)、类库以及其他支持执行以 Java 编程语言编写的程序的文件。

附加库——开发工具所需的其他类库和支持文件。

演示applet 和应用程序——Java 平台的编程示例源码。

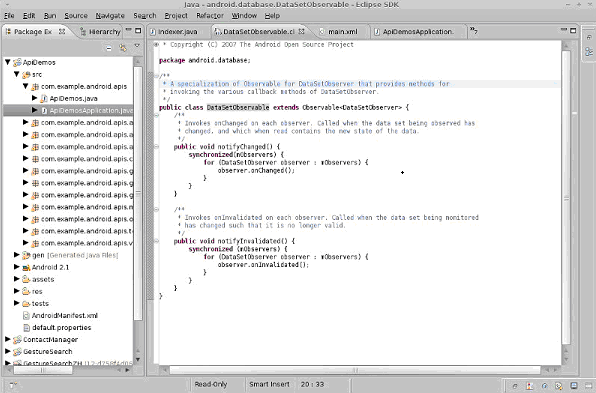
样例代码——某些 Java API 的编程样例源码。

C头文件——支持使用 Java 本机界面、JVM工具界面以及 JavaTM 平台的其他功能进行本机代码编程的头文件。

源代码——组成 Java 核心 API 的所有类的 Java源文件。

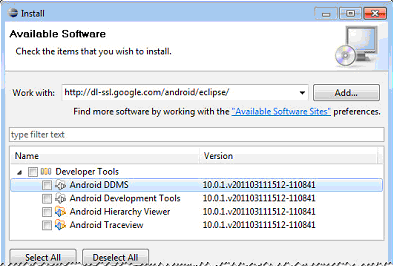
1. [Eclipse](http://www.jizhuomi.com/android/catalog.asp?tags=Eclipse" \t "https://www.cnblogs.com/xiaochao1234/p/_blank)

Eclipse最初由IBM开发的跨平台集成开发环境（IDE），后来贡献给Apache开源软件基金会。最初主要用于Java语言开发，目前可通过C++、Python、PHP等语言插件支持对应语言开发，所以，Eclipse看起来更像一个框架而已，更多工作都是交给插件或上文的JDK来完成，模块化的设计，让Eclipse的定位更清晰。



1. Android Development Tools（[ADT](http://www.jizhuomi.com/android/catalog.asp?tags=ADT" \t "https://www.cnblogs.com/xiaochao1234/p/_blank)）

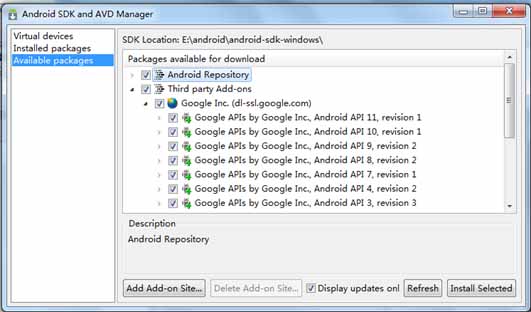
Android开发工具(ADT)，作为Eclipse工具插件，让其支持Android快速入门和便捷开发，可通过Eclipse启动菜单（启动Eclipse后，选择Help->Install New Software）安装。如下图所示。



ADT开发工具，包括Android Dalvik Debug Moniter Server(Android DDMS)和Android Development Tools（ADT），上面的插图中，还有Hierarchy Viewer和TraceView两个查看器，看名称就知道功能了。DDMS可以提供调试设备时为设备截屏，查看线程及内存信息、Logcat、广播信息、模拟呼叫、接收短消息、文件查看器等功能（该功能在安装完ADT之后，点击Eclipse工具栏右上角的那个带箭头的小窗口图标弹出菜单中打开）。 Android Development Tools（ADT）工具应该是为Eclipse支持Android项目快速编程开发和调试插件而已。

1. [Android Software Development Kit](http://www.jizhuomi.com/android/catalog.asp?tags=Android+SDK" \t "https://www.cnblogs.com/xiaochao1234/p/_blank)（SDK）

一般提到SDK就会想到：API接口库、帮助文档和示例源码，Android SDK似乎也不例外，它为开发者提供相关封装API接口库文件、文档资源及一些工具包整合。当然了如果你使用Eclipse作为开发工具，那么只需要安装SDK也可以，ECLIPSE和ADT也一并省略掉。下图是ADK和AVD管理器界面，可从SDK安装目录下执行SDK Manager.exe或在Eclipse中执行Window->Android SDK and AVD manager看到。



安装后SDK目录下的帮助文档相当完善，应有尽有（位置在SDK的docs目录下）。此外，Android SDK还包含一个手机模拟器（Virtual Devices），我们开发时就可以先使用模拟器进行模拟仿真，感觉时机成熟时再下载到真机进行测试。

SDK刚下载解压后，一般还需选择安装一些必须的API接口库及平台工具，下面提到的几项都建议下载安装（至于选择哪个版本，需接口您的开发样机支持Android版本）：

       Android SDK Tools, revision x

       Android SDK Platform-tools, revision x

       Documentation for Android SDK, API xx, revision x

       SDK Platform Android x.x, API x, revision x

       Samples for SDK API x, revision x

       Android Compatibility package, revision x

1. Dalvik虚拟机

Dalvik虚拟机是Android程序的虚拟机，是Android中Java程序的运行基础。其指令集基于寄存器架构，执行其特有的文件格式——dex字节码来完成对象生命周期管理、堆栈管理、线程管理、安全异常管理、垃圾回收等重要功能。它的核心内容是实现库（libdvm.so），大体由C语言实现。依赖于Linux内核的一部分功能——线程机制、内存管理机制，能高效使用内存，并在低速CPU上表现出的高性能。每一个Android应用在底层都会对应一个独立的Dalvik虚拟机实例，其代码在虚拟机的解释下得以执行。

与Dalvik虚拟机关系最密切的非JVM莫属，在Android源码readme文档中有这样一段话：Much of the code under this directory originally came from the Apache Harmony project, and as such contains the standard Apache header comment. Some of the code was written originally for the Android project…

Dalvik VM与Apache Harmony 项目关系源远流长，因此，与JVM关系自然就密切了。然而：Dalvik VM ≠Java VM，dalvik基于寄存器，而JVM基于stack，

Dalvik执行的是特有的DEX文件格式，而JVM运行的是\*.class文件格式。

优势：

（1）在编译时提前优化代码而不是等到运行时；

（2） 虚拟机很小，使用的空间也小；被设计来满足可高效运行多种虚拟机实例；

（3）常量池已被修改为只使用32位的索引，以简化解释器。

1. Android Native Development Kit(NDK)

Android原生开发套件，这里的原生似乎就是指开发语言原生，Android平台基于Linux内核的，所以这里就是指C、C++语言，这对于很多喜欢C/C++的程序员来说或许是个好消息，使用NDK一样可以进行Android开发。

由于NDK开发编译需要GCC编译环境，如果是windows环境，还应该安装Cygwin模拟环境。NDK包含哪些内容：

用于创建基于C/C++源文件的原生代码库；

提供一种将原生库集成到应用程序包，并部署到Android设备的方法；

一系列未来Android平台均会支持的原生系统头文件和库文件；

文档，示例和教程。

**2.2.2、[Android常用集成开发工具Eclipse和Android Studio对比](https://www.cnblogs.com/xiaochao1234/p/3833558.html)**

1、Android Studio的优势

Android Studio 构建程序界面更方便：

Android Studio从一出来的就打着所见得的旗号，一迅雷不及掩耳之势占领了Android项目的开发工具的市场，其实是有道理的。在Eclipse中构建的app的项目，不仅效果和真机上的差别太大，而且速度也不一般。但是Android Studio的界面显示就非常的清楚，并且修改起来也是非常迅速。

Android Studio打印信息更详细：

打印的信息可谓是应有尽有，几乎所有的项目中遇到的问题，包括编写、设计、开发、打包、构建等得错误信息都会在控制台上打印出来，便于问题的准确发下你和定位。反观Eclipse中的打印信息则寒掺得多，除了LogCat之外及时控制台，有事布局文件中的多个逗号“，”都发现不了。

Android Studio编辑历史更详细：

在工作台上修改代码、修改布局文件或者删除文件等，记录得非常细致，每一个操作都有记录，每一个操作都能够撤销。而Eclipse中删除文件后，之前编辑记录就会被清空，及时你恢复文件你就会觉得傻眼的，之前那么多的操作如何回滚是一个问题，除非你使用了版本控制工具。

Android Studio智能识别更强大：

智能识别在Android Studio中是一个非常大的优点，比如只要你输入Fp之后，会自动向你推送含有Fp或者fp、甚至是%F（f）%P（p）的选项，中间不管隔着多少个字符，或者大小不同，系统都能够识别出来并向你推送。

2、Eclipse的优势

Eclipse创建项目更简单:

在Eclipse中创建项目只是点击几下鼠标，然后创建Android项目的问题。但是在Android Studio中可就没有那么简单了，创建一个项目需要超过20分钟的时间不说，创建过程中还可能遇到各种gradle构建的问题，可谓十分麻烦。所以许多的程序习惯从Eclipse中创建项目后导入到Android Studio中。

Eclipse中项目体积小：

在Eclipse中所有的文件必须的，没有多余的配置文件，所有的项目体积很小，毕竟一个项目几十万行的代码，顶多就是30M。但是在Android Studio中就不一样了，各种配置文件，这些文件包含了工具自身的历史文件，还有gradle的构建文件，一个项目超过90M是轻轻松松的问题。

Eclipse中配置文件无需更新：

  创建好一个项目后到项目上线，可能你都无需更新任何Eclipse的文件，这个时间的跨度有可能是一年！而Android Studio中更新gradle文件是家常便饭，昨天才更新的文件，今天启动就需要更新了。甚至在上一次运行时还好好的，这一次运行gradle就出错了，由于不能正常使用Google的原因，更新的速度也不咋滴。

Eclipse中多项目管理很方便：

  因为所有的设置都市针对Eclipse的，添加项目将沿用这些设置，所以管理多个项目是顺理成章的事情。而是Android Studio中每个项目都已每个项目特殊的设置，所以一般是一个打开项目就需要另一个打开一个主窗口。

**2.2.3、本文选择Android Studio开发工具**

总体对比而言，Android Studio更加好用： 1.颜色、图片在布局和代码中可以实时预览 2.string可以实时预览 3.多屏预览、截图带有设备框，可随时录制模拟器视频 4.可以直接打开文件所在位置 5.跨工程移动、搜索、跳转 6.自动保存，无需一直Ctrl + S 7.即使文件关闭依然可以回退N个历史 8.智能重构、智能预测报错 9.每一行文件编辑历史，可追溯到人 10、各种插件 10.例如ADB、.gitignore、sql、markdown、 11.图片可直接转.9图片,并且自带.9编辑 12.在gradle编译时使用依赖超级方便等等。

所以在本文app平台开发中选用了Android Studio开发工具。

**2.3 百度地图开放平台**

**2.3.1 百度地图开放平台概述**

百度地图开放平台面向广大政府、企业、互联网等开发者开放地图服务能力，目前开发者市场占有率约达到75%，每天支撑近50万APP、网站应用运营。

百度地图开放平台拥有六大基础地图服务开放能力：定位、影像、出行、轨迹、数据、分析，并将六大服务能力免费开放给开发者使用。

**2.3.2 百度地图提供的产品与服务**

**定位**

LBS（Location Based Service，基于位置的服务）的三大目标是：你在哪里；你和谁在一起；附近有什么资源。其中“你在哪里”是LBS服务的核心。百度智能定位服务，是为了帮助广大开发者更好解决“你在哪里”这个难题而开放的服务。支持GPS、Wi-Fi、基站融合定位，完美支持各类应用开发者对位置获取的诉求。

百度地图开放平台定位服务是广大开发者定位首选服务，每日定位请求超过300亿次。

基础定位能力——依托百度位置大数据和高精尖定位技术，提供业内最优秀的定位服务产品。具有定位精度高（综合定位精度40m）、覆盖广（完美支持全球定位）、 流量小（小于0.3K）、速度快（首次定位200ms）、内存/CPU占用率低等特点。

离线定位技术——利用百度大数据分析挖掘能力，分析用户常驻点信息，在保证定位精准度的基础之上，提供离线定位能力。 利用离线定位能力，不仅定位速度进一步提升，同时也完美解决无网络环境下的位置获取难题。

室内高精度定位——大型商超内寻找店铺、停车场内反向寻车等业务的发展基础都是室内高精度定位。利用三角定位技术、增强Wi-Fi指纹模型技术、地磁技术、蓝牙技术等， 提供精度1-3m的室内高精度定位服务。结合百度室内地图服务，更好帮助开发者解决室内定位难题。

高精度IP定位——利用定位依据实时建库技术和百度大数据分析挖掘能力，开放高精度IP定位服务。定位综合成功率在65%以上，定位误差可有效控制在350m以内， 定位速度达到5ms。满足Web应用开发者对位置获取的需求。

位置语义化——优质的定位服务，精准性、成功率、覆盖率等是基础。更友好、智能化、人性化的描述当前用户的位置，是未来定位的发展趋势。我们所开放的定位服务， 不仅返回精准坐标，还会返回当前用户的详细地址、周边POI信息，和更人性化的位置描述信息（同时也提供自定义位置描述能力）。

**影像**

提供业内最全最新的矢量地图服务、卫星图、全景图、实时路况图、静态图和个性化地图服务。

**出行**

提供步行、骑行、公交、驾车等多种类型的线路规划方案，各方案还支持多种不同策略的检索，面向移动端设备提供导航SDK等产品。

**轨迹**

百度鹰眼是一套集轨迹追踪、存储、运算、查询的完整轨迹开放服务，可帮助开发者管理多达100万人/车轨迹。使用百度鹰眼，开发者可以轻松开发出适用于车队监控、车联网、专车计费、外业人员监管、儿童防丢领域的轨迹管理产品。

轨迹追踪——鹰眼提供Android SDK、iOS SDK、Web服务API、车载硬件解决方案和智能穿戴设备解决方案，帮助开发者轻松将手机、车联网硬件、智能穿戴设备等终端连接至鹰眼云端，持续回传轨迹。

存储并查询轨迹数据——鹰眼为轨迹数据提供长达1年的存储。同时，使用鹰眼移动端和服务端查询接口，您的应用程序可几乎无延时地查询终端实时位置和历史轨迹。

鹰眼硬件联盟成员系列——鹰眼联合优质硬件厂商组成硬件联盟，为开发者提供多种硬件产品选择。目前已开通购买硬件渠道，并欢迎广大硬件厂商加入百度鹰眼。

**数据**

基于百度地图丰富的地图数据，开放POI检索、热词推荐、地理编码等服务，通过LBS云服务实现结合个性化数据的地图功能开发，满足开发者对各类数据方面的需求。

分析

基于地理大数据、位置大数据、交通大数据和海量行为数据的商业地图产品，以领先的大数据分析和可视化技术，为政企、开发者提供更好的分析能力。

客流来源去向——从省、市、区县、商圈、街道维度精细分析客流来源、客流去向周边的分布。

客群画像——常驻居民与流动客群画像勾勒，从性别、年龄、资产状况、兴趣爱好、消费水平、消费偏好等多重维度立体化勾勒。

位置评估——分析常驻居民数量与密度分布、流动客群数量与密度分布、不同职业、年龄段人群分布、设施及场所分布。

室内客流分析——整体、楼层、店铺等客流分析、新老顾客及到店次数分析、实时客流热力分布。

室内定位——室内数据专人采集。

**分析**

基于地理大数据、位置大数据、交通大数据和海量行为数据的商业地图产品，以领先的大数据分析和可视化技术，为政企、开发者提供更好的分析能力。

客流来源去向——从省、市、区县、商圈、街道维度精细分析客流来源、客流去向周边的分布。

客群画像——常驻居民与流动客群画像勾勒，从性别、年龄、资产状况、兴趣爱好、消费水平、消费偏好等多重维度立体化勾勒。

位置评估——分析常驻居民数量与密度分布、流动客群数量与密度分布、不同职业、年龄段人群分布、设施及场所分布。

竞品研究——竞品分布热力、竞品客流指数、竞品客流属性。

室内客流分析——整体、楼层、店铺级客流分析、新老顾客及到店次数分析、实时客流热力分布。

室内定位——室内数据专人采集。

**2.3.3 百度地图提供的解决方案**

百度地图开放平台对不同行业和领域提出了有针对性的一站式服务解决方案。

**LBS游戏**

地图开放平台已提供定位、数据、地图和全景等多项服务，可为LBS游戏提供一系列的技术支持。对于游戏所需的定位、个性化地图展示、数据展示和360°全景图等功能，具有内存/CPU占用率低等特点，可满足游戏开发需求。

**O2O上门服务**

通过定位、数据、地图服务寻找周边服务，填写地址并进行下单，根据订单的位置特性，调度最合适的服务人员，规划路线，前往上门服务地点。

**物流配送**

通过对一个通用的物流配送流程进行分析，各个流程中分别用到了如下地图相关的功能：1.用户下单，即用户通过定位或place搜索确定收、寄件坐标和地址，完成下单。2.系统派单，即选择最合适的配送员进行服务： 将配送员实时位置上传至LBS云，利用LBS云检索、轻量级路线规划RouteMatrix批量路线规划和多点路径规划，计算周边实际距离最近的配送员。3.运输配送，即首先使用路径规划服务计算最优路径，预估运费。在运输过程中可使用导航SDK进行移动端实时导航。同时，通过鹰眼轨迹记录实时位置，监控运输全城，并利用地理围栏实现目的地到达提醒。4.运营分析，即在运营层面，可通过业务分布可视化实现运力和订单分布的展示，实现实时运力调度。在商业决策层面，可通过客群分析对用户群体进行用户画像，对配送站点和业务点选址进行综合分析。

**房产行业**

在地图上直接展示各POI分布，也可以通过各种组合条件查询，如价格、区域、在售状态、物业类型、新旧状态等。参考搜房网“地图搜索”；展示基础的图文信息；用全景加普通2D图尽可能为用户提供真实的数据获取周边休闲娱乐、餐饮、商场医院、学校、交通设施等信息；交通方案、距离等。

**智慧交通**

中国智慧交通云服务平台是业内云平台领导者，提供交通行业云服务，能将海量交通数据进行存储并建立海量数据搜索引擎,消除用户对基础设施与大数据使用的疑虑，帮助用户搭建全套云服务设施，给予用户0设施感，节省大量资源以及综合运维成本。目前已经建立全国交通数据仓储，辅助交通部门与行业完成数据一体化战略——从局部数据到总体数据的过程，把数据孤岛连接到网络，融合出交通出行原始资源大数据。

**商业地理**

勾勒顾客画像，展现顾客轨迹；竞品分析对比，客流来源去向，助您广拉新客，精准营销；结合不同需求，对目标区域进行全面位置评估，降低人力和时间成本。

**2.4 聚类方法**

**2.4.1 聚类方法简介**

聚类，顾名思义，聚类就是把一群没有标签的但有相同特征的数据聚在一起。广义的定义可以描述如下：把在某个方面有共性的成员聚成群组的过程。也就是，如果我们两个长得像，那么我们是一组。如果我们长得不像，我们就必须被分成两组。用一句中国的古话来描述，就是：物以类聚，人以群分

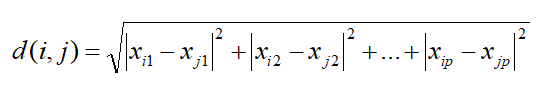
聚类是无监督学习中的典型算法，不需要对结果进行任何标记。聚类方法通常用来试图探索和发现一定的模式，进而用于发现共同的群体，按照内部存在的相似性将数据划分为多个类别使得同类内部相似性大，不同分类间相似性小。有时候作为监督学习中稀疏特征的预处理（类似于降维等方法，变成n类后，假设有6类，则每一行都可以表示为类似于000100、010000）。有时候可以作为异常值检测（反欺诈中有用）。

应用场景：新闻聚类、用户购买模式（交叉销售）、图像与基因技术等。

相似度与距离：这个概念是聚类算法中必须明白的，简单来说就是聚类就是将相似的样本聚到一起，而相似度用距离来定义，聚类是希望组内的样本相似度高，组间的样本相似度低，这样样本就能聚成类了。

1. Minkovski距离：当p=2时，就是欧式距离： 相似性就被定义为了d的倒数，1/d，当P=1时就说城市距离（曼哈顿距离）；

欧几里德距离(用的最多的):P=2

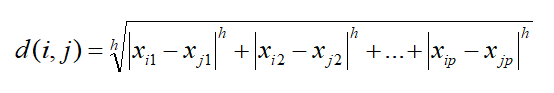


2.余弦距离：夹角的距离cosθ = (at\* b)/(|a|\*|b|) 余弦距离比较难收敛，优势是不受原来样本线性变换影响；

3. 皮尔斯相关系数：从概率论角度得到的距离 当x和y的均值为0时，皮尔森相关系数就等于余弦距离；

4. KL散度（交叉熵）：衡量两个分布之间的差异，不是传统意义上的距离，其中p（x）是真实样本分布，Q（x）是数据的理论分布，或者说是一种更简单的分布。有时候p（x）的分布很难写出，可以通过求KL散度最小而求出Q(X)。

5.闵可夫斯基距离(Minkowski Distance): 计算距离的通用的公式：



i = (xi1, xi2, …, xip) 和 j = (xj1, xj2, …, xjp) 是p维数据对象

6.曼哈顿距离(或城市块距离Manhattan distance):h=1

IMG_257

**2.4.2 常用聚类方法介绍**

1. K-Means(K均值)聚类

K-means 算法是重要的划分方法，由 J.B.MacQueen 提出，具有广泛的影响力。 其基本原理是，首先指定聚类数 k，从含有 n 个对象的数据集合中随机地选择 k 个对象作为一个簇的初始平均值或中心。计算剩余的各个对象到这些簇中心的距离，然后根据其与各个簇中心的距离，将它赋给最近的簇。 然后重新计算每个簇的平均值作为该簇新的聚类中心，如果相邻的聚类中心没有任何变化，则样本调整结束，聚类准则函数 E 己经收敛。准则函数的作用是使生成的结果簇尽可能地紧凑和独立。

算法步骤：

(1) 首先需要选择一些类/组，并随机初始化它们各自的中心点。中心点是与每个数据点向量长度相同的位置。这需要提前预知类的数量(即中心点的数量)；

(2) 计算每个数据点到中心点的距离，数据点距离哪个中心点最近就划分到哪一类中；

(3) 计算每一类中中心点作为新的中心点；

(4) 重复以上步骤，直到每一类中心在每次迭代后变化不大为止。也可以多次随机初始化中心点，然后选择运行结果最好的一个。

虽然 K-means 算法应用在很多方面，但仍然存在缺陷：(1)在 Kmeans 算法中 k 值必须事先给定，无法确定这个 K 值。 (2)K-means 聚类算法对初始质心严重依赖， 初始聚类中心直接影响着聚类结果，随机选取不同的初始聚类中心点，产生的聚类结果往往都不相同。 (3)K-means 算法的另一个缺点是通涉及到所有的变量且认为这些变量对距离影响的程度是等同的，容易引发“维数陷阱”。

2. 基于密度的聚类方法(DBSCAN)

DBSCAN是基于密度的聚类算法。

具体步骤：

（1） 首先确定半径r和minPoints. 从一个没有被访问过的任意数据点开始，以这个点为中心，r为半径的圆内包含的点的数量是否大于或等于minPoints，如果大于或等于minPoints则改点被标记为central point,反之则会被标记为noise point；

（2）重复1的步骤，如果一个noise point存在于某个central point为半径的圆内，则这个点被标记为边缘点，反之仍为noise point。重复步骤1，知道所有的点都被访问过。

优点：不需要知道簇的数量；

缺点：需要确定距离r和minPoints。

3. 层次聚类

层次聚类算法分为自上而下和自下而上两种。凝聚层级聚类(HAC)是自下而上的一种聚类算法。HAC首先将每个数据点视为一个单一的簇，然后计算所有簇之间的距离来合并簇，知道所有的簇聚合成为一个簇为止。

具体步骤：

（1）首先将每个数据点视为一个单一的簇，然后选择一个测量两个簇之间距离的度量标准。例如使用average linkage作为标准，它将两个簇之间的距离定义为第一个簇中的数据点与第二个簇中的数据点之间的平均距离；

（2）在每次迭代中，将两个具有最小average linkage的簇合并成为一个簇。

（3）重复步骤2知道所有的数据点合并成一个簇，然后选择需要多少个簇。

优点：

（1）不需要知道有多少个簇 ；

（2）对于距离度量标准的选择并不敏感。

缺点：效率低。

**2.4.3 本文提出加以改进的算法 ADAPT-means(Adaptive-means)。**

(1)质心的确定

为弥补k-means 算法的缺陷，本文在 k-means 算法的基础上先提出二分 k均值算法，即在算法初始时选出距离最远的两个对象，作为初始中心，形成两个初始簇，从这些簇中选取一个继续分裂，如此下去，直到产生 k 个簇，这种方法使得二分 k-means 不太受初始化的困扰，因为它执行每步只有两个质心。 这样就可以保证在后续的处理中同一类内的对象有极大的相似性，而不同类之间有极低的相似性。

(2)聚类个数 k 值的确定

借鉴最大、最小距离法的基本思想，来确定后续聚类的中心，同时在聚类过程中引入 Davies—BouldinIndex 指标来确定聚类个数 k。

Davies-BouldinIndex 又称为 DBI 指标，一种非模糊型的集群评估指标(clusteringvalidityIndex)，主要是利用几何原理进行运算，分别以聚类间离散程度与位于同一类内的数据对象的紧密程度作为依据，当类内各数据对象间距离越小而类间距离越大，DBI 值也就越小， 就代表各聚类内的数据对象越紧密且聚类间的差异大，表明此聚类数目下的聚类结果最佳。

具体算法步骤如下：

一、通过 (1)中的思想确定两个最佳的初始聚类中心 cl，c2；

二、查找新的聚类中心，分别计算数据集口中剩余的数据对象到cl，c2 的距离 d(c1，j)和 d(c2，j)；

三、D=max{min(d(c1，j)，d(c2，j))}，j=l，2，……，n。 得 到 xj，利 用 DBI；

公式判断是否是要找的最佳聚类中心，如何判断在下面介绍：

好的聚类结果应该是同一类间的各数据对象间相似度大，而不同类之间的相似度小，DBI 能够满足这个条件，即分子越小，分母越大时，DBI 的值越小，则代表各聚类内数据相似度大而类间的相似度小，从这个值可以确定通过 D=max{min(d(c1，j)，d(c2，j)}，查找到的 xj 是否是最佳中心，既而确定最佳的聚类数目。找到新聚类中心 xj，进行聚类，重新计算聚类的中心。 对目前形成的 k 个 聚 类 ， 计 算 其 DBI 和 前 一 次 计 算 的 DBI 进 行 比 较 ， 如 果DBInew<DBIold，则 xj 可以作为聚类中心，同时 k 在原来的基础上加1，否则算法终止。

四、如果找到的数据对象是中心，则计算 D=max{min(d(c1，j)，d(c2，

j)，d(c3，j))}，j’=1，2，…，n。 存在这样的数据对象，转到前一步。 依次类推直到跳出循环。从确定的过程可以看出，减少了人为的干预，避免了由于参数设定不当对聚类结果造成的影响。

(3)对改进 k-means 算法的描述

结合以上分析，给出改进的 k-means 算法流程如下：

输入：数据集

输出：聚类对象

l)扫描数据集 0，计算出数据集中最远的两个对象：x1，x2。

2)将数据集剩下的数据对象分别计算到 xl，x2 的距离，将其划分到距离小的那个中心的类中，标记，同时记下最小的距离。

3)划分完成后，重新计算聚类中心，得到 c1，c2。

4)借鉴最大、最小距离法的基本思想，max{min(d(c1，j)，d(c2，j))}，查

找到的 xj。

5)转入 2)，对整个数据集进行重新划分。

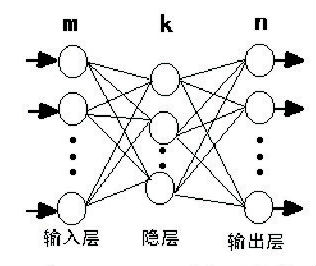
6)根据 DBI 公式，计算 DBInew 并比较，和上次计算出来的 DBIold进行比较，如果 DBInew<DBIold，也就是说找到 xj 合理的，k 值在上次的基础上增加 1， 否则就是再也找不到满足条件的新的聚类中心，聚类结束。

7)依次类推，直到无法找到满足条件的新的聚类中心，最终输出聚类结果。

**2.5 神经网络预测及优化**

**2.5.1 BP网络模型介绍**

BP网络（Back-ProPagation Network）又称反向传播神经网络， 通过样本数据的训练，不断修正网络权值和阈值使误差函数沿负梯度方向下降，逼近期望输出。它是一种应用较为广泛的神经网络模型，多用于函数逼近、模型识别分类、数据压缩和时间序列预测等。



m×k×n 三层BP神经网络

BP网络由输入层、隐层和输出层组成，隐层可以有一层或者多层，图2是m×k×n的三层BP网络模型，网络选用S型传递函数IMG_256， 通过反传误差函数IMG_257 （ （Ti为期望输出、Oi为网络的计算输出），不断调节网络权值和阈值使误差函数E达到极小。

**基本原理：**

[人工神经网络](https://baike.baidu.com/item/%E4%BA%BA%E5%B7%A5%E7%A5%9E%E7%BB%8F%E7%BD%91%E7%BB%9C" \t "https://baike.baidu.com/item/BP%E7%A5%9E%E7%BB%8F%E7%BD%91%E7%BB%9C/_blank)无需事先确定输入输出之间映射关系的数学方程，仅通过自身的训练，学习某种规则，在给定输入值时得到最接近期望输出值的结果。作为一种智能信息处理系统，人工神经网络实现其功能的核心是算法。BP神经网络是一种按误差反向传播(简称误差反传)训练的多层前馈网络，其算法称为[BP算法](https://baike.baidu.com/item/BP%E7%AE%97%E6%B3%95" \t "https://baike.baidu.com/item/BP%E7%A5%9E%E7%BB%8F%E7%BD%91%E7%BB%9C/_blank)，它的基本思想是梯度下降法，利用梯度搜索技术，以期使网络的实际输出值和期望输出值的误差均方差为最小。

基本BP算法包括信号的前向传播和误差的反向传播两个过程。即计算误差输出时按从输入到输出的方向进行，而调整权值和阈值则从输出到输入的方向进行。正向传播时，输入信号通过隐含层作用于输出节点，经过非线性变换，产生输出信号，若实际输出与期望输出不相符，则转入误差的反向传播过程。误差反传是将输出误差通过隐含层向输入层逐层反传，并将误差分摊给各层所有单元，以从各层获得的误差信号作为调整各单元权值的依据。通过调整输入节点与隐层节点的联接强度和隐层节点与输出节点的联接强度以及阈值，使误差沿梯度方向下降，经过反复学习训练，确定与最小误差相对应的网络参数(权值和阈值)，训练即告停止。此时经过训练的神经网络即能对类似样本的输入信息，自行处理输出误差最小的经过非线形转换的信息。

**执行流程（前向传递和逆向更新）：**

在手工设定了神经网络的层数，每层的神经元的个数，学习率 η（下面会提到）后，BP 算法会先随机初始化每条连接线权重和偏置，然后对于训练集中的每个输入 x 和输出 y，BP 算法都会先执行前向传输得到预测值，然后根据真实值与预测值之间的误差执行逆向反馈更新神经网络中每条连接线的权重和每层的偏好。在没有到达停止条件的情况下重复上述过程。

其中，停止条件可以是下面这三条：

（1）权重的更新低于某个阈值的时候

（2）预测的错误率低于某个阈值

（3）达到预设一定的迭代次数

譬如说，手写数字识别中，一张手写数字1的图片储存了28\*28 = 784个像素点，每个像素点储存着灰度值(值域为[0,255])，那么就意味着有784个神经元作为输入层，而输出层有10个神经元代表数字0~9，每个神经元取值为0~1，代表着这张图片是这个数字的概率。

每输入一张图片（也就是实例），神经网络会执行前向传输一层一层的计算到输出层神经元的值，根据哪个输出神经元的值最大来预测输入图片所代表的手写数字。

然后根据输出神经元的值，计算出预测值与真实值之间的误差，再逆向反馈更新神经网络中每条连接线的权重和每个神经元的偏好。

前向传输（Feed-Forward）：

从输入层=>隐藏层=>输出层，一层一层的计算所有神经元输出值的过程。

逆向反馈（Back Propagation）：

因为输出层的值与真实的值会存在误差，我们可以用均方误差来衡量预测值和真实值之间的误差。

**优劣势：**

BP神经网络无论在网络理论还是在性能方面已比较成熟。其突出优点就是具有很强的非线性映射能力和柔性的网络结构。网络的中间层数、各层的神经元个数可根据具体情况任意设定，并且随着结构的差异其性能也有所不同。但是BP神经网络也存在以下的一些主要缺陷。

①学习速度慢，即使是一个简单的问题，一般也需要几百次甚至上千次的学习才能收敛。

②容易陷入局部极小值。

③网络层数、神经元个数的选择没有相应的理论指导。

④网络推广能力有限。

对于上述问题，目前已经有了许多改进措施，研究最多的就是如何加速网络的收敛速度和尽量避免陷入局部极小值的问题。

**应用：**

目前，在人工神经网络的实际应用中，绝大部分的神经网络模型都采用BP网络及其变化形式。它也是前向网络的核心部分，体现了人工神经网络的精华。

BP网络主要用于以下四个方面。

1)函数逼近：用输入向量和相应的输出向量训练一个网络逼近一个函数。

2)模式识别：用一个待定的输出向量将它与输入向量联系起来。

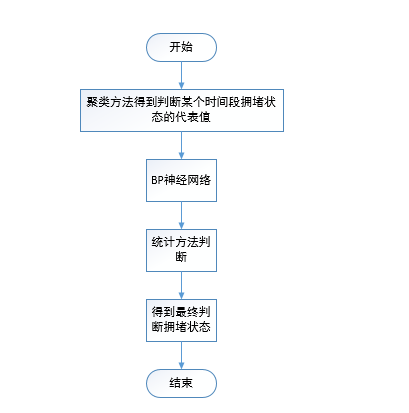
3)分类：把输入向量所定义的合适方式进行分类。

4)数据压缩：减少输出向量维数以便于传输或存储。

**2.5.2 基于BP网络模型的优化思路**

1. 方法提出

本文中神经网络简单模拟图示



本文实验框架流程图

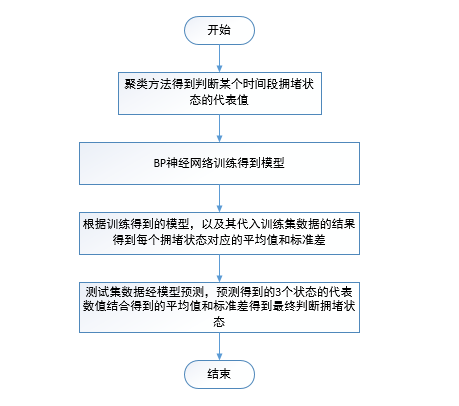
第一阶段

在第一阶段中，通过训练得到经典的BP神经网络模型，如上图所示，输入数据为4个，是4个连续单位时间内通过的车辆数量，预测结果为3个在0到1之间的数值，分别对应拥堵、一般、空闲的代表值，通过这3个数据来判断接下来一个连续单位时间内的交通拥堵状态，通过训练可得到经典的BP神经网络模型。

第二阶段

通过获得的数据集的大部分数据作为训练集，第一阶段训练得出了经典的神经网络模型。如上图所示，模型对输入会得到3个对应的输出。本文在训练出第一阶段的神经网络模型后，再次将训练集数据输入，然后得到所有训练集数据通过模型预测得到的3个代表数值和数据对应交通拥堵状态的对应值。之后再将所有同一拥堵状态下的这3个数值及其对应着的状态进行归类整合在一起。再对所有属于每个分类下的数据进行处理，获得每个状态分类下对应的最大数值的平均值和标准差。最后在对测试集的预测中，再对预测得到的3个数值进行状态归类的环节，结合这两个数据，判断这3个数值相对每一种状态的平均值的偏离程度，并以此作为辅助判断预测结果的手段。

1. 算法提出



算法流程图

第一部分

K-MEANS算法对交通数据进行聚类

输入：

需要聚类的个数-k，和相似度阈值-θ

输出：

　　簇和异常值（不一定存在）

算法：

　　开始每个点都是单独的聚类，根据计算点与点间的相似度，生成相似度矩阵；

　　根据相似度矩阵和相似度阈值-θ，计算邻居矩阵-A。如果两点相似度>=θ,取值1（邻居），否则取值0；

　　计算链接矩阵-L=A x A；

　　计算相似性的度量（Goodness Measure），将相似性最高的两个对象合并。回到第2步进行迭代直到形成k个聚类或聚类的数量不在发生变换。

第二部分

结合神经网络模型和统计学方法对交通流的拥堵状况进行预测的算法

具体过程如下：

输入：连续4个单位时间（5分钟）内的车流辆数据

输出：判断得到的交通拥堵状态，比如拥挤、一般和空闲。

第一步：将所有数据集数据集分为训练集和测试集两部分；

第二步：通过聚类得到的交通拥堵状态判断标准，对训练集中所有数据进行 状态的归类。

第三步：通过第二步中的归类结果，建立一般的BP神经网络模型，该模型包含4个输入和3个输出，3个输出的数值为3种状态的代表数值，并以此判断交通拥堵状态，开始进行训练；

第四步：通过训练得到的神经网络模型反过来对训练集数据进行处理得到各自状态对应的3个数值；并通过3个数值种取决定性作用的那个数值来确定每个状态对应的平均值和标准差。

第五步：最终通过下面的公式对3个数值所属于的交通拥堵状态进行判别

，，y为通过模型后得到的每种状态的代表数值，比如判断是否属于拥堵状态，在公式中通过比较预测得到的数值时候在这个区间可以进行判断，



第六步：重复前面的步骤依次对一般状态以及空闲状态进行判别，通过第五步的辅助判断标准，便更能确定各状态的判别结果。

**2.6 本章小结**

本章介绍了本文研究过程中用到的主要平台和方法，主要提出了一种思路，通过神经网络训练得到预测模型，联合统计方法校验得到预测结果的思路对交通流的拥堵状况进行预测，即在神经网络建模的基础上结合统计学的理论进行辅助判断。

1. **系统需求分析**

**3.1系统业务需求**

笔者曾通过问卷调查、面对面交谈等形式与数名多年从事一线物流相关派送工作的配送员进行了详细全面的交谈，并结合身边数十名＂网购达人＂的切身经历，总结出物流配送员在派送物件的过程中普遍遇到的一些问题。

快递员寄快递时存在的问题：

如果收件用户的群体在交通复杂的区域，尤其对于高校在校生，那么快递员在派送快递的过程中，往往是提前１到２个小时通过发短信的形式通知收件人来取件。如果收件人由于忙碌等各种原因未能及时查看短信的话就不能按时去取快递，快递员只能再通过打电话来通知收件人。这样一来比较浪费时间。如果收件人开会等不便没有及时接到快递员的电话或者因有事不能前来取快递，那么快递员这次就白跑一趟，只能下次再派送了。

如果收件用户的群体是居民小区的住户或单位的员工，快递员通常情况都不

会提前确认收件人是否在家或者在上班，而是快要到达收件人地址后再通过短信

或者电话通知收件人。如果收件人不在家或不在工作单位，那么快递员只能与收

件人另外再约定时间派送了。这样一来，通常也会使快递员白跑一趟，浪费时间不说，还无形之中增加了快递员的工作任务。

现在城市内道路交通发展迅速，各种车辆尤其是私家车的数量增加的很快，道路经常或者说很容易陷入拥堵状态，如果不能及时并且准确的提前预测各条道路接下来一段时间内的交通状态情况，配送员在面对各种交通状态是就容易陷入被动状态，也容易让自己的配送任务不能按时的完成，造成不必要的麻烦。

**3.2系统功能需求**

针对物流配送相关人员在派送各种物件过程中存在的上述问题，笔者开发

了基于Android的物流派送系统，目的是给快递员提供一系列的实用性功能，比

如快速定位、导航、实时了解各道路交通拥堵状态以及及时预测各道路接下来交通拥堵情况等，使物流配送人员在派送物件的过程中更能及时的了解到接下来要走的各种线路的交通拥堵情况，提高物件配送的效率。

如图３－１所示，是物流配送客户端app的功能结构图。下面就客户端的功能结构进行详细的说明。

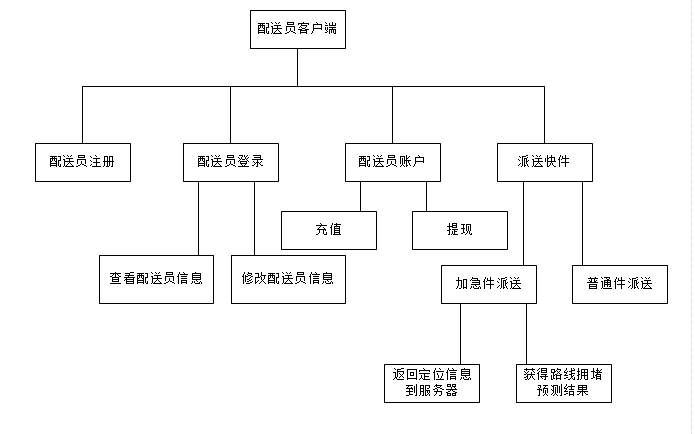


图３－１

该客户端系统从功能上主要划分为快递员注册模块、快递员登录模块、快递员账户模块、派送快件模块等主要功能模块。

该系统架构采用传统的Ｃ／Ｓ模式的三层架构。三层架构分别是显示层、业务逻辑层、数据持久层。显示层主要由基于Android的快递员端组成，业务逻辑层在服务器端实现，数据持久层则是由MySql来完成。

根据系统需求和功能模块划分，总结出系统用例图如图３－２所示:

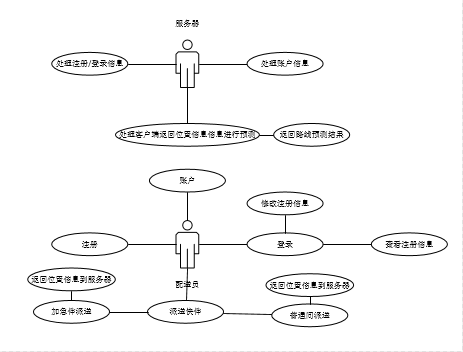


图３－２

**3.2.1 配送员注册模块**

用户注册几乎是所有应用程序的一个最基本的功能模块。配送员注册是为了使新用户注册并使用该应用程序。每个配送员只有完成注册后才能使用该应用程序。

配送员的注册信息包括手机号、登录密码、姓名、身份证号、所属快递公司、所属快递公司的快递办公网点、支付宝账号（注册的时候可先不填写，配送员登录后可以再补充完善）。

新用户注册时，首先要填写自己的手机号和登录密码，然后点击请求获取短信验证码，服务器端收到这个请求后，会检查该手机号是否己经注册过。如果该

手机号己经注册过，那么提示新用户“手机号己注册，请直接登录”。否则，服

务器给该手机号发送短信验证码。短信验证码验证通过后，新用户需要继续完善

个人信息，包括真实姓名、身份证号、所属快递公司、所在快递公司的办公网点、

支付宝账号等。

另外，系统为了推广，专口设置了＂推荐人＂这一角色。通过推荐别人使用该系统，可获得相应的奖励。所在快递员注册的时候，还可以填写推荐人的手机号，这样推荐人就可获取一定的奖励。

**3.2.2 配送员登录模块**

配送人员注册成功后登录系统后可以查看自己注册的个人信息、派送物件的列表、问题件列表、账户情况等。另外，配送人员登录系统后，还可以修改个人注册信息，比如上传或者修改自己的头像，修改支付宝账号信息等，但是姓名、手机号、身份证号、所属快递公司和快递网点不能修改。如果快遥员所属的快递公司和快递网点发生变化，配送人员可以请求客服人员进行重新认证，认证通过后系统自动更新快递员的个人信息。

**3.2.3 派送快件模块**

配送人员在完成注册并实名认证后，就可以开始派送快递的工作了。快递派送工作围绕快递单的状态展开，从快递员扫描寄件人的快递单的那一刻起，到收件人签收快递单止，快递单贯穿整个派送过程。具体功能设计如下：

**１、实时向服务器汇报自己的位置信息**

该系统集成了百度地图LBS（基于位置的服务）技术，并使用Android设备

的GPS功能实时获取快递员的位置，也就是快递员所处位置的经纬度信息。快递员从打开应用的那一刻起，就定时向服务器汇报自己的实时位置坐标。汇报位置采用的规则是：每隔３分钟向服务器汇报一次。

**２、配送员扫描快递单**

该系统集成了Google开源项——ZXing，可以用来扫描识别快递单上的条形码，扫描识别率和准确率都比较高，一般每秒可正确识别一个条形码。如果遇到由于快递单污损等原因而造成快递单上的条形码无法通过扫描来识别，该系统还提供了手动输入条形码的功能。另外，如果周围环境太昏暗而无法扫描识别快递单上的条形码，还可以打开手机内置的闪光灯来辅助照明。

**３、普通件派送**

如果快递单上的收件人姓名、通讯地址、联系方式等信息完整且准确无误，那么该快递单就属于一个＂合格＂的快递单，对该快递单的派送工作就属于普通件派送。此时，快递员只需要将快递在一定的时间范围內送到收件人手中并签收，那么此单快递的派送工作就完成了

**４、加价件派送**

对于一个＂合格＂的快递单而言，如果寄件人需要在最短的时间内让快递到达收件人手中，那么寄件人可以通过和快递员协商，通过提高快递费用来完成目的。

当寄件人对他的某一快递单提交＂加价＂请求后，服务器会收到这一请求，然后通过推送技术将此请求推送给快递员。快递员对寄件人的＂加价＂请求作

出响应（＂同意＂或者＂拒绝＂）后，服务器再将快递员的＂答复＂推送给寄件人。当然，＂加价＂的具体细节寄件人和快递员可以聊天或者直接打电话来协商确定。

**５、问题件派送**

如果快递单上寄件人或者收件人的姓名、通讯地址及联系方式等信息有缺失

或者不准确，那么该快递单属于＂问题件＂。快递员在派送＂问题件＂的时候，

可以反馈给服务器，需要后台维护人员及时更正才能完成快递派送工作。

**６、实时推送快递单的状态变化信息**

快递整个派送过程就是围绕快递单展开的，从快递单的创建、被派送、签收

的全过程都应该实时地被寄件人和快递员掌握。该应用可以实时地接收到来自服务器端的推送消息，动态地掌握快递单的状态。

递派送过程更加可控、透明、时效。

**3.2.4 配送员账户管理模块**

由于寄件人在发快递的时候可以通过微信或支付宝的支付方式向快递员支

付快递费，所以快递员在派送快递的过程中可能会产生收入和消费等行为。该系统集成了第三方支付平台alipay，便于支持快递员对自己的账户进行充值和

提现操作。

１、充值

配送人员发现自己的账户余额不足完成某一项活动需求时，可Ｗ对自己的账户进行充值。

２、提现

配送员还可以随时随地提取自己账户的金额，目前本系统支持的体现最小金

额是20元。

**3.3系统非功能性需求**

通常来讲，系统的需求可以分为功能性需求和非功能性需求两部分。非功能

性需求是对功能需求的一个很好补充，对系统的开发同样至关重要。

所谓非功能性需求，是指系统为满足用户业务需求而必须具有的一些特性，

包括系效率、易用性、可靠性、可扩展性、可维护性等方面的要求。

１、一些手机应用经常会出现＂闪退＂的现象。该系统应该保证程序运行稳定，性能效率高，尽可能少出现或者不出现由于内存错误引起的崩溃、闪退等现象。另外，执行一些耗时较长的网络操作时应该尽量减少用户等待时间，避免出现＂ANR＂现象（Application Not Responding）。

２、用户界面风格设计应同时满足可使用性、灵活性和可靠性，并且符合Android系统的操作使用规范。

３、系统的设计应该符合＂高内聚低稱合＂的特点，使得系统能够有比较良好的可扩展性。

**3.4物流配送系统总体设计**

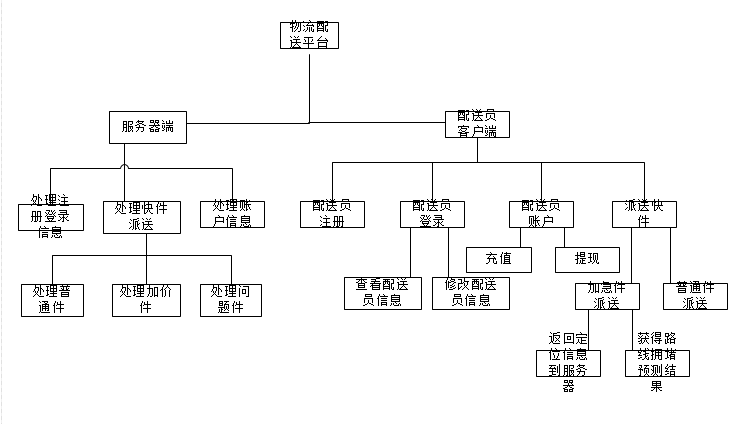


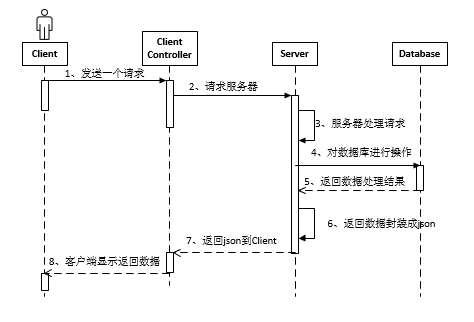
图3-3 物流配送平台总体功能结构图

整个快递派送平台主要由两个部分组成，分别是配送人员端和服务器端。服务器端是基于MVC的设计模式，采用的是目前比较流行的SSM（Spring + SpringMvc + Mybatis）架构来实现。应用服务器是Tomcat，数据库方面选用了开源的MySql数据库。

服务器端主要有两方面的功能，一方面是对系统的用户（物流配送人员）进行管理，另一方面是向客户端（物流人员客户端）提供各种调用接口的

支持，响应并处理客户端的业务请求，给客户端返回相应的结果。

如图3-4所示，为服务器响应客户端一个请求的处理流程。用户在客户端进行操作，然后客户端业务還辑控制器将操作产生的请求连同参数一起发送给服务器；服务器解析并处理请求，接着向数据库发出操作请求；数据库对服务器发来的请求作出响应并把结果返回给服务器。此时，服务器利用返回结果创建json，并把json返回给客户端，最后客户端将json进行一定的处理并显示出来。



3-4 服务器处理客户端的请求时序图

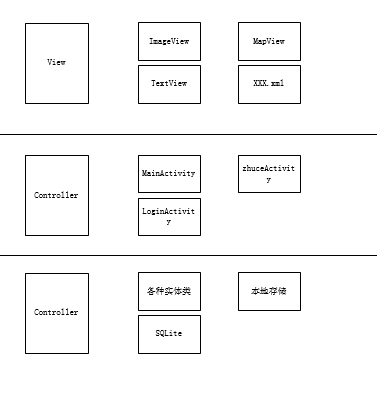
物流配送人员客户端的设计也采用MVC的设计模式。在Model层，主要包含一些实体类（如JavaBean等）、SharedPreferences、SQLite数据库（一种安卓系统内置的数据库）等，主要做一些业务逻辑处理，比如数据库存取操作、网络操作、耗时的任务等都在Model层处理；在View层，一般采用XML文件进行界

面的描述，使用的时候可以非常方便的引入，另外还可以根据业务需求自定义一

些UI组件，View通常用来显示Model层的数据结果。在Controller层，用Activity

来处理用户交互问题，Activity读取View层的数据（如读取当前EditText控件的

数据），控制用户输入（如EditText控件数据的输入），并向Model层发送数据请求（如发起网络请求等）。客户端架构图如图3-5所示。



3-5 客户端架构图

**3.5数据存储和交互**

客户端获取数据的来源主要是从服务器端得到网络数据。在该系统中，客户

端与服务器端的数据交互是通过ＨＴＴＰ协议完成的，请求方式为GET和POST

方式，并采用JSON作为服务器端返回给客户端数据的格式。

该系统之所以选择JSON作为基础框架协议，是因为JSON主要有两方面的优点，一方面移动互联网应用通常要考虑到流量限制因素，网络通信数据流量不宜太大，而JSON数据格式简单，易于读写，而且是经过压缩处理的，传输速度

快；另一方面，JSON数据封装简洁易懂，容易解析。

在数据存储方面，该系统绝大多数的数据（包括配送人员的信息、快递单信息、账户信息等）都存储在服务器端，客户端只是存储了极少量的数据。这些数据可以存储在Android系统内置的一些存储机制中，如SharedPreferes、SQLite数据库等。

**3.6本章小结**

本章第一节针对目前物流配送人员在派送快递过程中存在的一些问题进行物流派送平台的业务需求分析；第二节基于第一节的业务需求分析提出系统的功能需求，并分模块介绍系统的各主要功能；第三节提出了系统在非功能性需求方面应该达到的要求：第四节介绍了整个快递派送平台的功能划分，并简单介绍了服务器端和客户端的系统架构组成；第五节简单地描述了服务器和客户端之间进行数据交互和存储的原理及方式。

1. **系统详细设计与实现**
2. **总结与展望**

**致谢**

**参考文献**