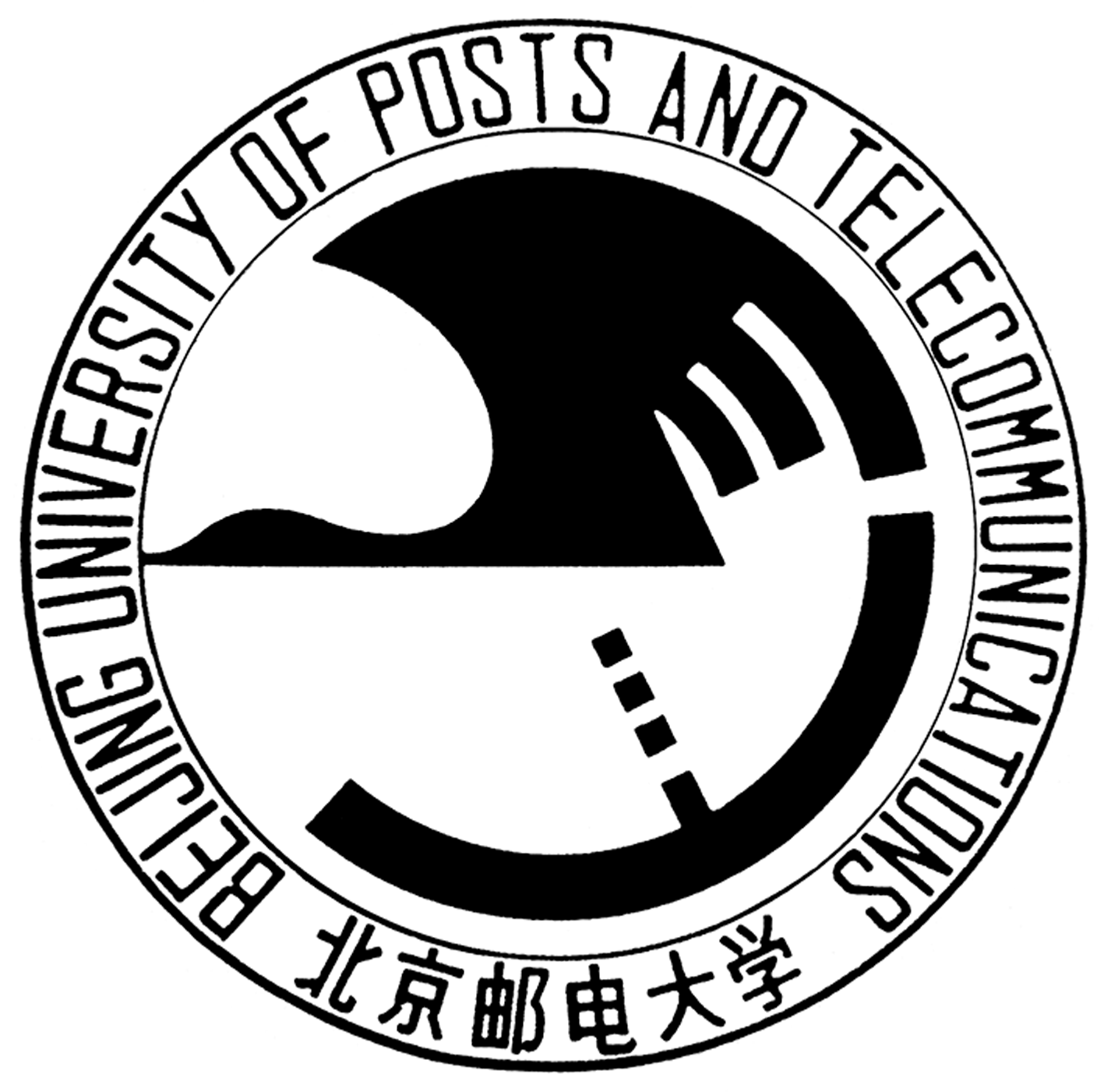


**本 科 毕 业 设 计（论文）**



**题目: 多社交网络关联融合系统的设计与实现**

**姓 名 赵莹莹**

**学 院 计算机学院**

**专 业 网络工程**

**班 级 2014211311**

**学 号 2014211446**

**班内序号 24**

**指导教师 林荣恒**

**2018 年 6 月**

**多社交网络关联融合系统的设计与实现**

**摘　要**

用户在多个社交网络中注册，并且同时使用多个社交网络已经成为常态。多个社交网络为用户带来方便的时候，给研究人员带来了新的挑战。一方面多个社交网络的用户信息不尽相同，另外一方面社交网络的用户、网络结构也不相同，这些为研究人员创建融合社交网络带来了困难。本毕业设计的主要任务是设计并实现多社交网络关联融合子系统，实现在异构社交网络间进行对齐的关键算法。

本文首先介绍了当前进行多社交网络关联融合对于现实的意义以及多社交网络进行部分对齐的问题描述。接着介绍多社交网络关联融合系统的所使用的相关技术和算法，提出通过监督学习和稳定匹配进行关键算法的研究。按照软件工程的流程对系统进行设计与实现，首先进行需求分析，明确系统的基本应用场景和使用角色，使用用例图和用例说明进行描述，接着进行系统的概要设计和详细设计，介绍了多社交网络关联融合系统的系统架构图和功能结构框图，通过静态类图和系统顺序图详细描述系统各模块功能。

本文根据关于设计网络对齐的一些已有研究，通过改进分类算法、稳定匹配算法以及加入迭代框架，实现本系统中的关键算法ISS（Iterative Supervised framework with Strict stable matching），通过在现实社交网络twitter和foursquare上进行实验并和已有研究方法比较得出该算法在对齐问题上的有效性。最后对所有的工作进行总结，展示已完成的系统对各功能的实现，分析关键算法有待改进的部分以及多社交网络关联融合子系统有待扩展的部分。

**关键词** 部分对齐 异构社交网络 严格稳定匹配 迭代框架

**Design and Implementation of the system for muti-social network convergence**

**ABSTRACT**

It is normal for users to register in multiple social networks and use multiple social networks at the same time. When multiple social networks bring convenience to users, they bring new challenges to researchers. On the one hand, the user information of multiple social networks is not the same. On the other hand, the users and network structure of social networks are also different, which has created difficulties for researchers to create a social network. The main task of this graduation project is to design and implement a multi-social network convergence system, and to implement key algorithms for aligning heterogeneous social networks.

This paper first introduces the current significance of multi-social network convergence to the reality and the problem of partial alignment of multiple social networks. Next, the related technologies and algorithms used in the multi-social network convergence system are introduced, and key algorithms are proposed through supervised learning and stable matching. In accordance with the software engineering process, the system is designed and implemented. Firstly, the requirements analysis is performed, the basic application scenarios and roles of the system are defined, the use case diagram and the use case description are described, and then the system's detailed design are introduced. The system architecture diagram and functional block diagram of the network-association fusion system describe the functions of the system modules in detail through the static class diagram and the system sequence diagram.

Based on some existing researches on designing network alignment, this paper implements the key algorithm ISS (Iterative Supervised framework with Strict stable matching) in this system by improving the classification algorithm, stable matching algorithm, and adding iterative framework. The experiments on foursquare and the comparison with the existing research methods show that the algorithm is effective in the alignment problem. Finally, it summarizes all the work, shows the completed system's implementation of each function, analyzes the parts of the key algorithm that need to be improved, and the multi-social network association fusion subsystem needs to be expanded.

**KEY WORDS** partially aligned, heterogeneous social network, Strict Stable matching, iterative framework

**目 录**

[第一章　绪论 1](#_Toc514767986)

[1.1　选题的背景及意义 1](#_Toc514767987)

[1.2　课题任务及主要研究工作 1](#_Toc514767988)

[1.3　论文组织结构 2](#_Toc514767989)

[第二章　相关技术简介 3](#_Toc514767990)

[2.1　Vue.js简介 3](#_Toc514767991)

[2.2　Django简介 3](#_Toc514767992)

[2.3　分类算法简介 3](#_Toc514767993)

[2.4　稳定婚姻匹配算法简介 4](#_Toc514767994)

[第三章　系统需求分析 5](#_Toc514767995)

[3.1　基本应用场景 5](#_Toc514767996)

[3.2　基本功能说明 6](#_Toc514767997)

[3.3　用例图及用例说明 6](#_Toc514767998)

[3.3.1　用例图 6](#_Toc514767999)

[3.3.2　用例说明 8](#_Toc514768000)

[3.4　本章小结 14](#_Toc514768001)

[第四章　系统设计与实现 16](#_Toc514768002)

[4.1　系统架构图 16](#_Toc514768003)

[4.2　系统功能结构框图 16](#_Toc514768004)

[4.2.1　系统模块结构图 16](#_Toc514768005)

[4.2.2　系统接口说明 18](#_Toc514768006)

[4.2.3　各模块交互关系 19](#_Toc514768007)

[4.3　系统算法设计与实现 23](#_Toc514768008)

[4.3.1　多社交网络对齐问题定义 23](#_Toc514768009)

[4.3.2　ISS的算法流程 24](#_Toc514768010)

[4.3.3　算法实现及结果分析 27](#_Toc514768011)

[4.4　系统模块设计与实现 28](#_Toc514768012)

[4.4.1　多社交网络融合算法模块 28](#_Toc514768013)

[4.4.2　算法实例管理模块 32](#_Toc514768014)

[4.4.3　社交网络数据集解析与管理模块 34](#_Toc514768015)

[4.4.4　网络拓扑可视化模块 36](#_Toc514768016)

[4.4.5　用户管理监控模块 38](#_Toc514768017)

[4.4.6　系统日志记录模块 40](#_Toc514768018)

[4.5　数据库设计与实现 42](#_Toc514768019)

[4.6　本章小结 44](#_Toc514768020)

[参考文献 45](#_Toc514768021)

第一章　绪论

1.1　选题的背景及意义

近年来，提供​​各种特色的在线社交网络服务已经成为我们生活中不可或缺的一部分，这些社交网络实质是异构的，社交网络中会存在不同的结构与关系，为了享受更多的社交网络服务，多数用户会加入不同的社交平台，比如，Foursquare, twitter和Facebook，但是同一用户在不同网络中的多个账号之间大多是相互隔离的，彼此之间没有任何关联，而异构网络中的用户信息的融合对于更好得理解完整得社交关系、全方位刻画用户、复杂社交链路预测、跨领域推荐等具有重要作用，异构社交网络对齐，即多社交网络关联融合，社交网络分析已经由最初的社会学问题，转变成计算机领域重要的研究方向，而随着web2.0的兴起，大型社交网络网站为社交网络分析提供了实际的大叔级。在计算机领域的相关研究中，将同一用户在不同网络账号间的联系称为锚链接，对应的用户称为锚用户，异构社交网络的对齐问题也可以转化成锚链接识别问题。同时，研究人员所获得的社交网络数据一般不完整，或由于隐私等原因隐去了重要的信息，所以就也就需要根据不完整的信息构建完整的社交网络。

另外从现实中的社交网络情况出发，本文选题的核心在于异构社交网络的部分对齐，也就是，并不是所有的用户都在不同的社交网络中拥有账号，所以现实中锚用户和非锚用户在社交网络中同时存在，如何准确得识别出锚用户和非锚用户是对现实异构社交网络进行分析的基础。

1.2　课题任务及主要研究工作

结合图论和监督学习的相关知识，设计并实现多社交网络关联融合系统。该系统包括社交网络数据集和规模的选取，社交网络拓扑结构展示，算法参数的选择，算法执行，已训练完成的算法演示执行过程展示以及预测结果和相关评价指标的比较与展示等功能。

结合已有的关于多社交网络对齐的研究，根据分类训练和稳定匹配，进行优化得出结合监督学习和严格稳定匹配的迭代框架（ISS）用于预测多社交网络中的锚链接，在迭代过程中不断完善社交网络信息与特征，使得严格稳定匹配得出的结构相关指标得已提升，即更准确地完成多社交网络部分对齐的任务。

根据已有的原始社交网络数据，明确文件解析方法和生成的社交网络数据结构，确定社交网络数据集的gpickle存储格式，结合多社交网络关联融合系统的应用场景，实现对于每个节点的已知信息的查看以及已知锚用户数量的调整，并进行以上信息的网络拓扑图的可视化展示，用户可在网页端根据相关的调整按钮实现自定义算法或者本文中提出的关键算法ISS选择，执行算法，并且可以查看预测结果和相关指标。

理解使用Django作为后端服务器提供resful api和前端框架Vue.js的MVVM的开发模式，Vue.js是一个构建数据驱动的web界面的渐进式框架，在视图层实现数据的双向绑定，实现在多社交网络关联融合子系统前端对于用户请求的处理和展示。采用Django中提供的models.py编写类，通过ORM在Mysql中操作数据库。了解Vue.js的视图组件化element-ui以及网络拓扑结构可视化的vis.js在Vue.js中的使用，使之满足用户选择参数的社交网络拓扑结构的可视化展示。

1.3　论文组织结构

本论文结构如下：

第一章：绪论。本章引入多社交网络对齐概念，并总结了系统需要完成的任务和待解决的问题。

第二章：相关技术。本章简要介绍了多社交网络关联融合系统用到的相关框架和算法。

第三章：系统需求分析。本章对多社交网络关联融合系统主要的使用场景结合用例进行分析、说明。

第四章：系统设计与实现。本章介绍了系统的功能结构框图和所包括的模块，以及模块与模块之间的交互联系和系统接口并进一步介绍了各个模块内部的系统序列图。

第五章：系统测试。本章列举了多社交网络关联融合系统各个模块的测试用例，结合系统各个模块进行功能与非功能测试。

第六章：展望与总结。对本系统的设计与实现过程进行总结。分析系统中需要改进的部分

第二章　相关技术简介

本章主要简单介绍了系统使用的框架以及算法。主要包含Django、Vue.js框架以及研究算法过程中用到的监督分类算法和稳定匹配算法。

2.1　Vue.js简介

Vue.js是JavaScript MVVM库，它是以数据驱动和组件化思想构建的，Vue的数据驱动是数据改变驱动了视图的自动更新，不需要手动改变DOM来改变试图，这也就是MVVM思想的实现，视图的组建化是将整一个网页拆分成一个个区块，每个区块可以看作成一个组件，网页可以由多个组件拼接或者嵌套做成。并且相比于Angular.js，提供了更加简洁、更加易于理解的API。并且View Model是Vue.js的核心，它是一个Vue实例，Vue实例是作用在某一个HTML元素上的，这个元素可以是body元素也可以是任何一种元素。

为了满足我们当前web的开发需求，使用Vue.js作为前端框架，实现前后端的完全分离，首先需要新建Vue.js的前端应用项目，并且使用webpack打包项目，进行页面组件的编写调整以及页面跳转路由，前端展示数据对象等逻辑处理。

2.2　Django简介

Django是一个基于MVC构造的框架，由python写成。在Django的一般项目中，控制器接受用户输入的部分由框架自行处理，所以 Django 里更关注的是模型（Model）、模板(Template)和视图（Views），称为 MTV模式。

而在本系统中使用了Vue.js作为前端框架，代替Django本身较为孱弱的模板引擎，Django作为服务端提供api接口，保留Django自带的对象关系映射（ORM）与数据库进行交互，使得前后端完全分离。

2.3　分类算法简介

分类算法是根据已知样本特征和标签训练模型对于未知样本的标签进行预测，单一的分类方法主要包括：决策树、贝叶斯等；另外还有用于组合单一分类方法的集成学习算法，如Bagging和Boosting等，接下来简单介绍一下本问中涉及的相关算法原理简介。

**支持向量机（SVM）:** SVM是在分类与回归分析中分析数据的监督式学习模型与相关的学习算法。给定一组训练实例，每个训练实例被标记为属于两个类别中的一个或另一个，SVM训练算法创建一个将新的实例分配给两个类别之一的模型，使其成为非概率二元线性分类器。SVM模型是将实例表示为空间中的点，这样映射就使得单独类别的实例被尽可能宽的明显的间隔分开。然后，将新的实例映射到同一空间，并基于它们落在间隔的哪一侧来预测所属类别。

除了进行线性分类之外，SVM还可以使用所谓的核技巧（一般选择rbf等核函数）有效地进行非线性分类，将其输入隐式映射到高维特征空间中。

**随机森林（Random Forest）：**鉴于决策树的过拟合的特点，随机森林是采用多个决策树的投票机制来改善决策树，类似于bagging思想，采用重采样的方式从样本集中产生n个样本，并对n个样本用建立决策树的方式获得最佳分割点，重复m次可产生m棵决策树，最后用多数投票机制要进行预测。

AdaBoosting：boosting算法的思想的大致原理是通过训练多个弱分类器组合成强分类器，提升决策能力。AdaBoosting是能够实际应用的Boosting算法，开始时，每个样本对应的权重是相同的，即 其中 n 为样本个数，在此样本分布下训练出一弱分类器 。对于 分类错误的样本，加大其对应的权重；而对于分类正确的样本，降低其权重，这样分错的样本就被突出出来，从而得到一个新的样本分布 。在新的样本分布下，再次对弱分类器进行训练，得到弱分类器。依次类推，经过 T 次循环，得到 T 个弱分类器，把这 T 个弱分类器按一定的权重叠加（boost）起来，得到最终想要的强分类器。

以上提到的分类算法都包含在python的机器学习库skicit-learn中，在系统开发时方便引用训练。

2.4　稳定婚姻匹配算法简介

1962 年，美国数学家 David Gale 和 Lloyd Shapley 发明了一种寻找稳定婚姻的策略。不管男女各有多少人，不管他们各自的偏好如何，应用这种策略后总能得到一个稳定的婚姻搭配。换句话说，他们证明了稳定的婚姻搭配总是存在的。

具体过程是第一轮，每个男人都选择自己名单上排在首位的女人，并向她表白。这种时候会出现两种情况：（1）该女士还没有被男生追求过，则该女士接受该男生的请求。（2）若该女生已经接受过其他男生的追求，那么该女生会将该男士与她的现任男友进行比较，若更喜欢她的男友，那么拒绝这个人的追求，否则，抛弃其男友。第一轮结束后，有些男人已经有女朋友了，有些男人仍然是单身。在第二轮追女行动中，每个单身男都从所有还没拒绝过他的女孩中选出自己最中意的那一个，并向她表白，不管她现在是否是单身。这种时候还是会遇到上面所说的两种情况，还是同样的解决方案。直到所有人都不在是单身。

由上述算法进行拓展，对于任何一种根据候选集进行两两配对的情况，稳定匹配算法都可以进行应用得到稳定配对结果。

本章主要介绍了构建系统需要用到的框架和算法，框架使用前后端分离的方式进行渐进式web开发，算法采用监督分类算法和稳定匹配算法。因此需要慢慢熟悉并掌握上面提到的框架及算法的实现过程，然后进行系统设计的下一步工作。

第三章　系统需求分析

多社交网络关联融合系统需求分析需要解决系统的基本应用场景以及基本的功能介绍。系统的使用者包括普通用户及管理员这两种角色。

3.1　基本应用场景

系统最终以网页呈现给用户。多社交网络数据集、用户信息、算法执行状态、系统操作日志信息都存储于数据库中。

根据具体需求，该系统的功能操作大致可以分为三个场景：

1. 用户及管理员登录系统场景

实现内容：输入用户验证信息，通过此场景进入系统。

场景描述：通过输入系统网址，跳转到登录界面，用户输入验证信息后进入下一场景。

1. 用户查看算法示例演示、执行算法、管理算法实例等信息场景

实现内容：用户实现本系统的主要操作。

场景描述：用户进入系统主页面，该系统主要由算法示例展示模块、算法执行模块、算法实例管理模块等组成。

首先用户登录成功后，系统首先根据当前用户的权限信息，确定用户是不是管理员是否拥有数据集管理和用户管理的权限。

当用户进入算法示例展示模块后，通过选择可看到不同算法示例的拓扑算法配置，对齐前后社交网络的查看（拓扑图展示），以及预测结果的相关指标评价（柱状图展示）。

当用户进入算法执行模块后，首先可选择不同的社交网络数据集以及网络规模，浏览选择完成的对齐前社交网络的可视化（拓扑图展示），接下来选择ISS算法或者自定义算法，接着进行算法相关参数的选择，提交启动算法任务，可看到算法启动成功或者失败的提示，表示算法任务线程创建成功或者失败。算法任务结果和状态需要在算法实例管理模块中查看，当用户进入算法实例管理模块后，可查看算法任务的执行状态，以及执行完成的对齐后社交网络可视化（拓扑图展示）以及相关指标查看，除此之外，用户也可以执行对相关算法任务的删除操作等。

系统日志记录模块将用户的所有操作记录在相关数据库中。

1. 系统管理员管理用户、管理数据集、算法实例管理场景

当管理员进入算法实例管理时，除了查看删除等基本操作之外，还可将相关算法实例设置成算法演示示例展示给普通用户。

当管理员进入数据集管理模块时，首先可查看现有的社交网络数据集，进行删除上传等操作。

当管理员进入用户管理模块时，能够在网页端限制用户操作，实现用户信息的编辑、新用户的添加以及用户的移除，并保存修改信息。系统将对用户的所有请求进行管理和监控，用户的一切操作必须在拥有相关权限时才可完成，比如当用户未登录时，所以页面的访问都会自动跳转到请求登录界面。

3.2　基本功能说明

1. 普通用户

账户管理：实现用户账户的登入、登出、密码修改功能。

查看算法示例：实现用户对于已有的算法示例的算法配置、拓扑结构、预测结果、相关指标的查看。

执行算法：实现用户选择相关参数，查看对齐前社交网络，通过选择算法参数启动算法任务。

算法实例管理：实现用户对于算法任务状态的查看，执行成功的对齐后结果的查看。

1. 系统管理员

账户管理：实现用户账户的登入、登出、密码修改功能。

用户管理：添加、移除用户，以及编辑用户权限，如将普通用户赋予管理员权限等。

查看算法示例：实现用户对于已有的算法示例的算法配置、拓扑结构、预测结果、相关指标的查看。

执行算法：实现用户选择相关参数，查看对齐前社交网络，通过选择算法参数启动算法任务。

算法实例管理：实现用户对于算法任务状态的查看，执行成功的对齐后结果的查看。并且可将执行成功的算法任务设置称为示例展示给用户。

数据集管理：实现管理员对于社交网络数据集的上传和删除操作。

3.3　用例图及用例说明

根据上述基本功能，管理员和普通用户的基本用例图如图3-1及3-2所示。

3.3.1　用例图

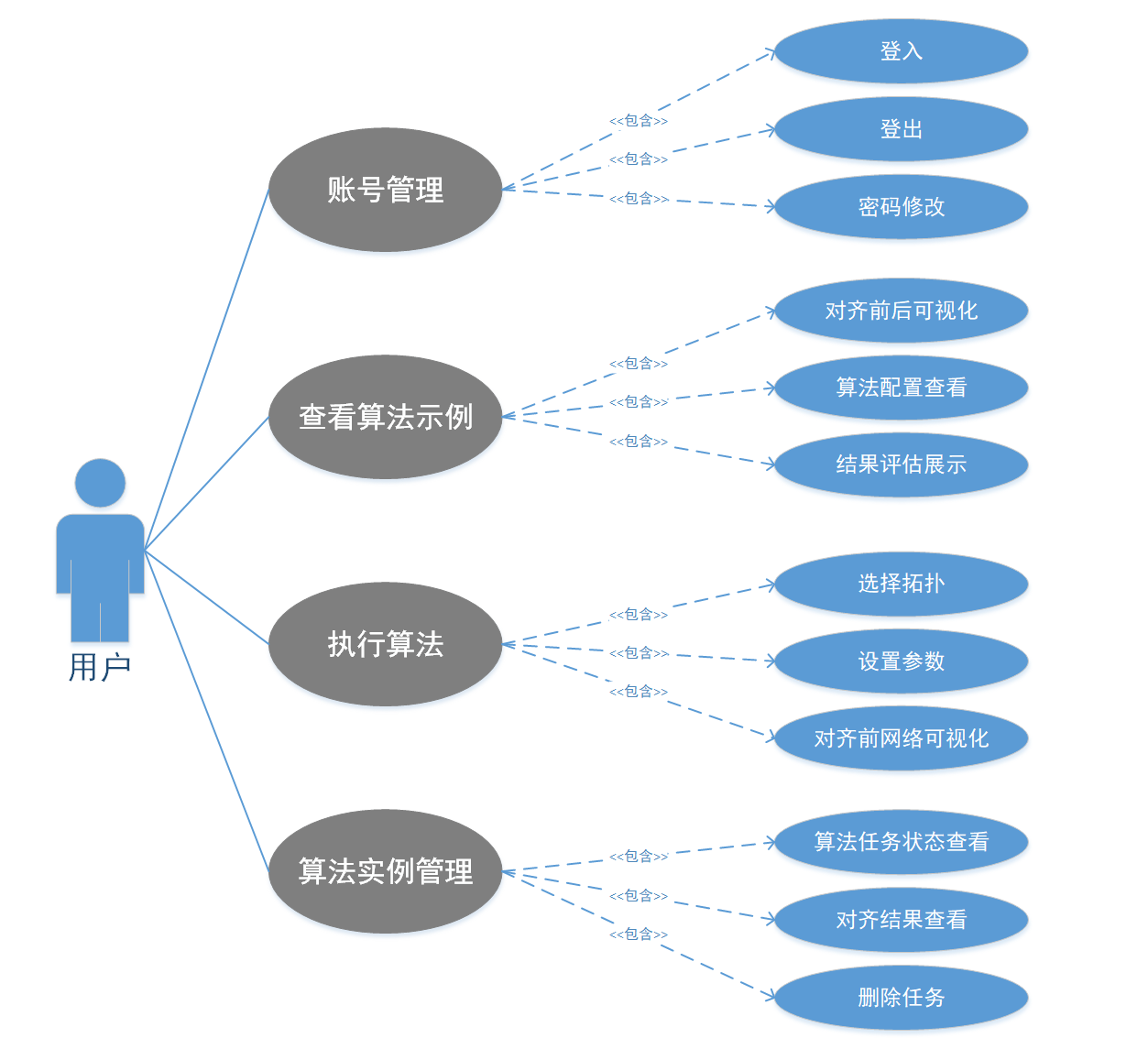


图3-1. 普通用户用例图

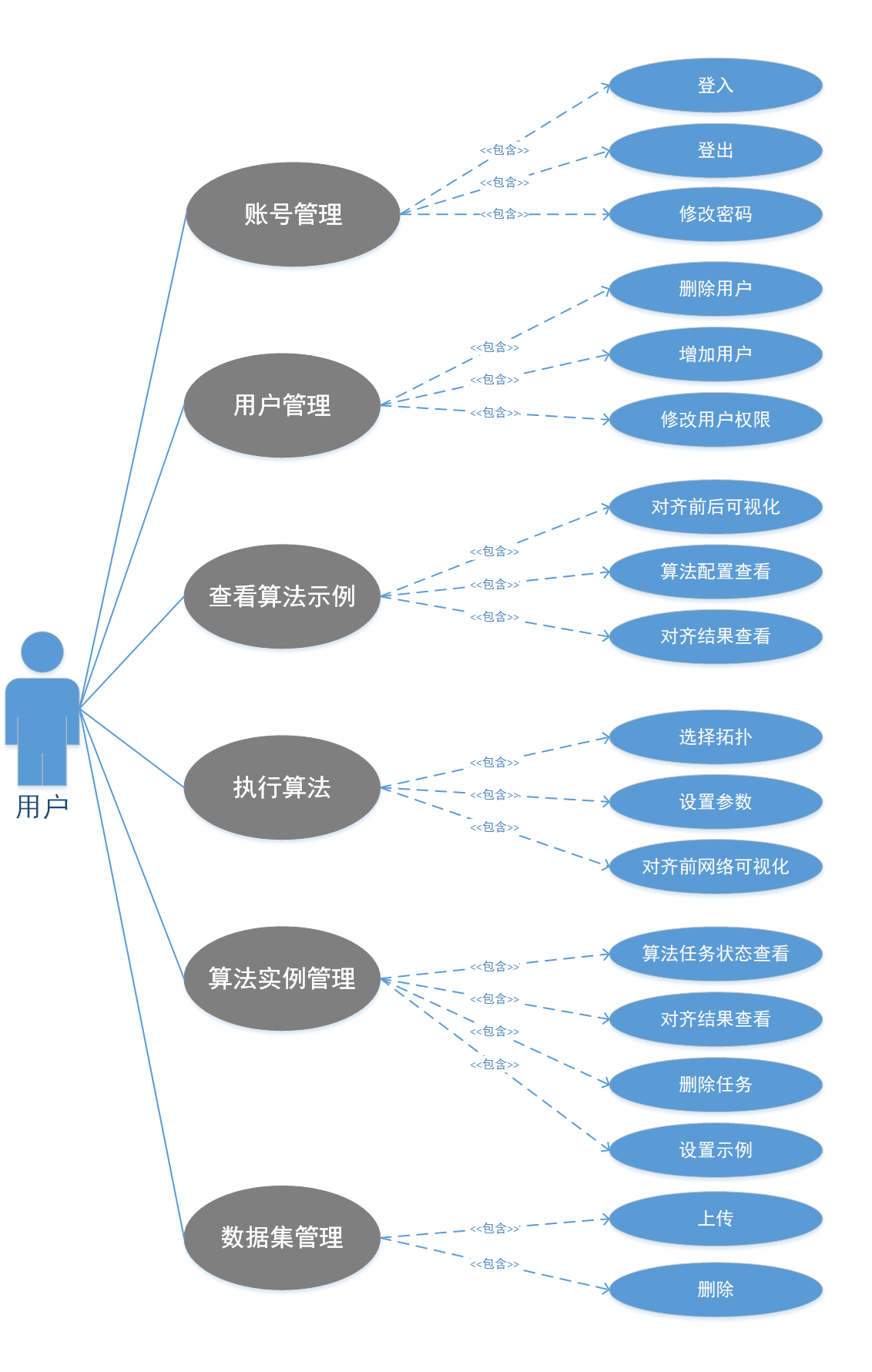


图3-2. 系统管理员用例图

3.3.2　用例说明

1. 用户用例说明

用户用例图包含账户管理、执行算法、查看算法示例、管理算法实例。

1. 账户管理（受系统管理员限制）

该用例包含登录、登出系统、修改密码三个用例。

表3-1. 用户登录系统用例表

|  |  |
| --- | --- |
| 用例编号 | User\_01 |
| 用例名称： | 登录系统 |
| 范围： | 多社交网络关联融合系统 |
| 主要参与人： | 用户 |
| 前提条件： | 用户具备登录信息 |
| 后置条件： | 用户成功登录系统 |
| 主要成功场景： | |
| 1 | 用户提交正确的用户名和密码 |
| 2 | 系统验证用户所输入的信息 |
| 3 | 系统返回验证结果 |

表3-2. 用户登出系统用例表

|  |  |
| --- | --- |
| 用例编号 | User\_02 |
| 用例名称： | 登出系统 |
| 范围： | 多社交网络关联融合系统 |
| 主要参与人： | 用户 |
| 前提条件： | 用户登录成功 |
| 后置条件： | 成功退出系统 |
| 主要成功场景： | |
| 1 | 用户提交登出请求 |
| 2 | 系统注销当前用户 |
| 3 | 重定向到登录界面 |

表3-3. 用户修改密码用例表

|  |  |
| --- | --- |
| 用例编号 | User\_03 |
| 用例名称： | 修改密码 |
| 范围： | 多社交网络关联融合系统 |
| 主要参与人： | 用户 |
| 前提条件： | 用户登录成功 |
| 后置条件： | 密码修改成功 |
| 主要成功场景： | |
| 1 | 用户提交修改密码请求 |
| 2 | 输入当前密码和新密码 |
| 3 | 返回成功修改密码提示，重定向到登录界面 |

1. 执行算法

包含选择拓扑、设置算法参数、对齐前网络可视化。

表3-4. 用户选择拓扑用例表

|  |  |
| --- | --- |
| 用例编号 | User\_04 |
| 用例名称： | 选择拓扑 |
| 范围： | 多社交网络关联融合系统 |
| 主要参与人： | 用户 |
| 前提条件： | 用户登录成功 |
| 后置条件： | 网络拓扑图显示成功 |
| 主要成功场景： | |
| 1 | 系统显示选择的数据集的信息，包括节点、边和已知锚节点 |
| 2 | 系统将拓扑图将社交网络展示出来，将已知锚节点用不同的颜色着重展示 |
| 扩展（或替代流程） | |
| 1.a | 如果用户没有自定义数据集，则选择默认数据集和规模结构 |

表3-5. 用户设置算法参数用例表

|  |  |
| --- | --- |
| 用例编号 | User\_05 |
| 用例名称： | 设置算法参数 |
| 范围： | 多社交网络关联融合系统 |
| 主要参与人： | 用户 |
| 前提条件： | 用户选择算法以及相关算法参数 |
| 后置条件： | 算法参数选择成功，启动算法任务 |
| 主要成功场景： | |
| 1 | 系统设置算法后启动算法任务成功 |
| 扩展（或替代流程） | |
| 1.a | 如果用户未设置参数，则按照默认参数执行算法 |

1. 查看算法示例

表3-6. 对齐前后可视化用例表

|  |  |
| --- | --- |
| 用例编号 | User \_06 |
| 用例名称： | 对齐前后社交网络可视化 |
| 范围： | 多社交网络关联融合系统 |
| 主要参与人： | 用户 |
| 前提条件： | 用户登录成功，选择查看的示例 |
| 后置条件： | 用户查看对齐前后社交网络的拓扑图 |
| 主要成功场景： | |
| 1 | 用户查看示例的对齐前后网络拓扑图 |

表3-7. 算法配置查看用例表

|  |  |
| --- | --- |
| 用例编号 | User \_07 |
| 用例名称： | 算法配置查看 |
| 范围： | 多社交网络关联融合系统 |
| 主要参与人： | 用户 |
| 前提条件： | 用户登录成功，选择查看的示例 |
| 后置条件： | 用户查看示例的算法配置 |
| 主要成功场景： | |
| 1 | 用户查看示例的使用的算法类型以及配置参数 |

表3-8. 结果评估展示用例表

|  |  |
| --- | --- |
| 用例编号 | User \_08 |
| 用例名称： | 结果评估展示 |
| 范围： | 多社交网络关联融合系统 |
| 主要参与人： | 用户 |
| 前提条件： | 用户登录成功，选择查看的示例 |
| 后置条件： | 用户查看算法预测结果的评估 |
| 主要成功场景： | |
| 1 | 用户查看算法示例的预测结果的评价指标比较（柱状图） |

1. 算法实例管理

表3-9. 算法任务状态查看用例表

|  |  |
| --- | --- |
| 用例编号 | User \_09 |
| 用例名称： | 算法任务状态查看 |
| 范围： | 多社交网络关联融合系统 |
| 主要参与人： | 用户 |
| 前提条件： | 用户登录成功 |
| 后置条件： | 用户查看算法任务实例列表 |
| 主要成功场景： | |
| 1 | 用户查看创建的算法任务的执行状态 |

表3-10.对齐结果查看用例表

|  |  |
| --- | --- |
| 用例编号 | User \_10 |
| 用例名称： | 对齐结果查看 |
| 范围： | 多社交网络关联融合系统 |
| 主要参与人： | 用户 |
| 前提条件： | 用户登录成功 |
| 后置条件： | 用户选择执行成功的任务实例查看结果 |
| 主要成功场景： | |
| 1 | 用户查看选定的任务的对齐后结果 |

表3-11. 删除任务用例表

|  |  |
| --- | --- |
| 用例编号 | User \_11 |
| 用例名称： | 删除任务 |
| 范围： | 多社交网络关联融合系统 |
| 主要参与人： | 用户 |
| 前提条件： | 用户登录成功 |
| 后置条件： | 用户选定删除的算法任务 |
| 主要成功场景： | |
| 1 | 用户提交删除任务请求 |
| 2 | 系统返回删除任务成功 |

1. 系统管理员用例说明（与普通用户相同的用例这里不再赘述）

系统管理员除此之外具有用户管理、算法示例管理、数据集管理的功能。

1. 用户管理

用户管理包含以下三个用例：

表3-12. 添加用户用例表

|  |  |
| --- | --- |
| 用例编号 | Admin\_01 |
| 用例名称： | 添加用户 |
| 范围： | 多社交网络关联融合系统 |
| 主要参与人： | 系统管理员 |
| 前提条件： | 系统管理员登录成功 |
| 后置条件： | 用户添加成功 |
| 主要成功场景： | |
| 1 | 管理员提交添加用户请求 |
| 2 | 管理员编辑新的用户信息 |
| 3 | 新的用户信息保存到数据库 |

表3-13. 修改用户权限用例表

|  |  |
| --- | --- |
| 用例编号 | Admin \_02 |
| 用例名称： | 修改用户权限 |
| 范围： | 多社交网络关联融合系统 |
| 主要参与人： | 系统管理员 |
| 前提条件： | 系统管理员登录成功 |
| 后置条件： | 修改用户信息 |
| 主要成功场景： | |
| 1 | 管理员提交修改用户权限请求 |
| 2 | 数据库更新用户权限 |

表3-14. 删除用户用例表

|  |  |
| --- | --- |
| 用例编号 | Admin \_03 |
| 用例名称： | 删除用户 |
| 范围： | 多社交网络关联融合系统 |
| 主要参与人： | 系统管理员 |
| 前提条件： | 系统管理员登录成功 |
| 后置条件： | 删除用户信息 |
| 主要成功场景： | |
| 1 | 管理员提交删除用户请求 |
| 2 | 管理员成功删除用户 |

1. 数据集管理

表3-15. 上传用例表

|  |  |
| --- | --- |
| 用例编号 | Admin \_04 |
| 用例名称： | 上传 |
| 范围： | 多社交网络关联融合系统 |
| 主要参与人： | 系统管理员 |
| 前提条件： | 系统管理员登录成功 |
| 后置条件： | 上传数据集信息 |
| 主要成功场景： | |
| 1 | 管理员选择上传的数据集信息 |
| 2 | 成功上传数据集至数据库 |

表3-16. 删除用例表

|  |  |
| --- | --- |
| 用例编号 | Admin \_05 |
| 用例名称： | 删除 |
| 范围： | 多社交网络关联融合系统 |
| 主要参与人： | 系统管理员 |
| 前提条件： | 系统管理员登录成功 |
| 后置条件： | 删除社交网络数据集 |
| 主要成功场景： | |
| 1 | 管理员选择删除的的数据集 |
| 2 | 成功从数据库中删除相应数据集 |

1. 算法示例管理

表3-17.设置示例用例表

|  |  |
| --- | --- |
| 用例编号 | Admin \_06 |
| 用例名称： | 设置示例 |
| 范围： | 多社交网络关联融合系统 |
| 主要参与人： | 系统管理员 |
| 前提条件： | 系统管理员登录成功 |
| 后置条件： | 设置算法示例 |
| 主要成功场景： | |
| 1 | 管理员从算法实例列表中选择设置的示例 |
| 2 | 算法示例设置成功，普通用户可在算法示例模块查看 |

3.4　本章小结

本章主要对多社交网络关联融合系统的的应用场景和基本的功能进行了描述，明确主要应用场景在于用户对于算法的执行以及算法结果的查看，在进行基本实验的基础上可得，算法一般需要一段的训练过程从而得到结果，为了方便用户对于系统的使用，所以在此基础上延伸了其它的应用场景如数据集管理、算法实例管理等，将算法启动和算法结果分开，最后通过用例图以及用例说明进行描述。

在本章明确之后，下面就可以进一步具体得对系统进行设计与实现。

第四章　系统设计与实现

首先需要对多社交网络关联融合系统的系统架构图与功能结构框图进行设计，确定了系统的总体功能，采用系统顺序图对各个模块之间的功能以及交互关系进行动态描述，并且采用类图对系统的功能进行静态描述，并且确定系统的接口。

4.1　系统架构图

如下图所示，多社交网络关联融合系统整体上由系统用户界面、服务器与数据库三个部分组成。用户界面和前端路由逻辑由Vue.js框架实现，前端向服务器发送请求，服务器通过Django提供的api接受请求后进行相关指令处理，并通过数据访问模块中读取相关数据之后返回。系统中涉及的所有的数据保存在数据库中，包括用户信息，社交网络数据集，算法实例信息，操作日志等。

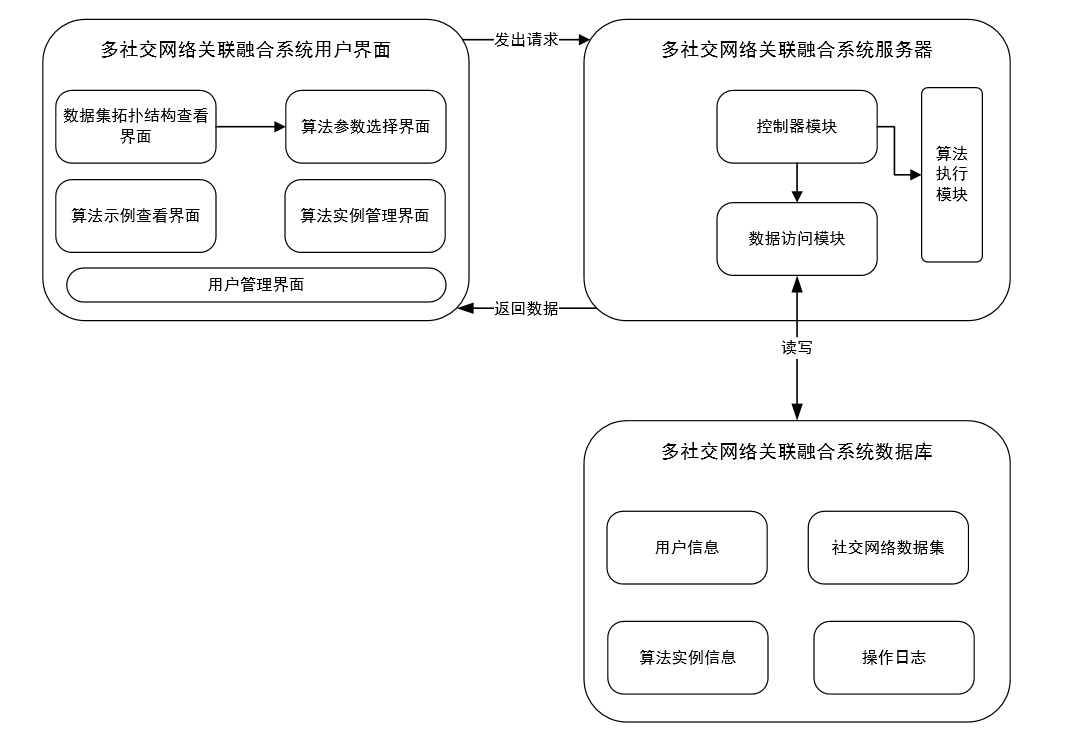


图4-1. 系统架构图

4.2　系统功能结构框图

4.2.1　系统模块结构图

多社交网络关联融合系统由用户界面模块，网络拓扑可视化模块，多社交网络融合算法执行模块，算法实例管理模块，社交网络数据集解析与管理模块，数据访问模块，用户权限管理监控模块以及系统日志记录模块构成，具体构建如下图所示。

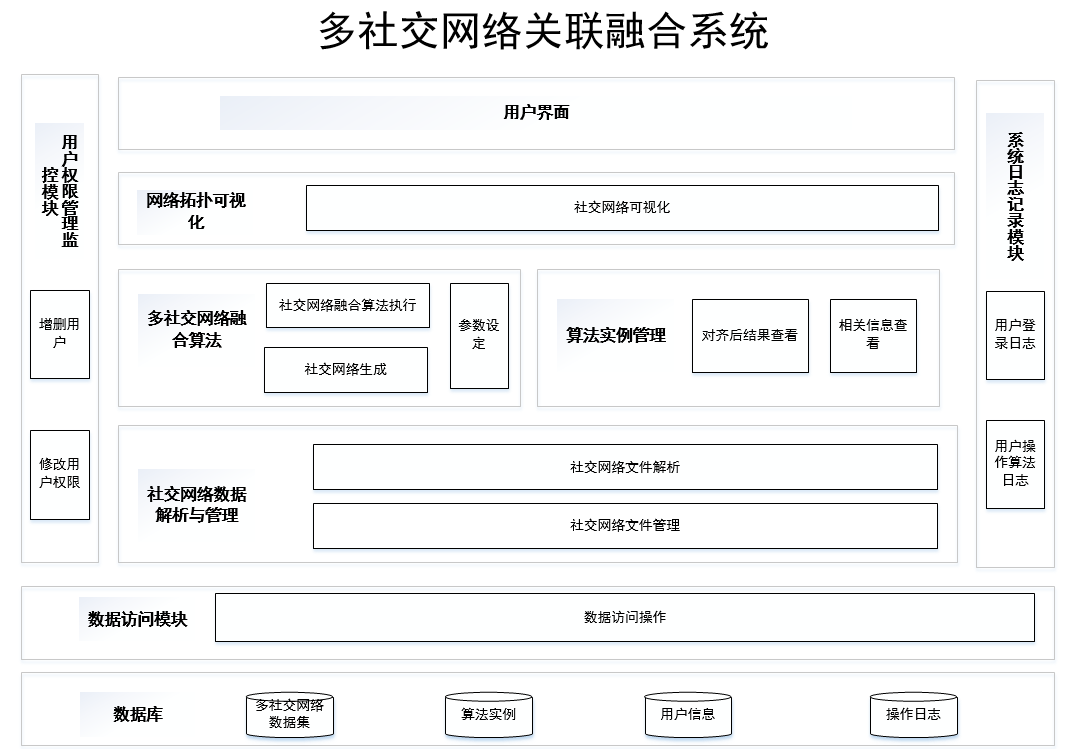


图4-2. 系统模块结构图

上图的各个功能组件的定义如下：

1. **网络拓扑可视化**

该模块负责根据指定的数据在界面上展示网络拓扑结构。

1. **社交网络融合算法**

该模块负责设定参数，并根据生成的社交网络数据结构，执行指定多社交网络融合算法。

1. **算法实例管理**

该模块根据管理员id查看相关算法实例的执行状态，包括算法执行前后网络拓扑结构，以及对齐结果和指标的查看。

1. **社交网络数据解析与管理**

该模块负责对于读入的社交网络原始数据进行解析，并生成结构化的可被算法以及可视化所使用的社交网络数据。并且该模块的功能社交网络文件管理（仅管理员可进入），提供管理可供执行算法的原始数据集的导入以及删除的功能，比如Foursquare，Twitter等。

1. **系统日志记录模块**

该模块用于记录用户的各种操作及发生时间，例如登入、登出、执行算法、查看社交网络等。根据python的logger模块进行设置，便于将所有相关的操作进行记录，便于系统监控。

1. **用户权限管理监控模块**

该模块仅限管理员进入，管理员可进行用户的增加、删除，修改用户的密码及权限。在本系统中前端Vue.js根据用户登录时返回的权限信息存储在sessionStorage中，前端页面跳转根据sessionStorage的相关权限信息进行处理。

1. **数据操作模块**

对于数据库文件等信息进行操作，作为底层模块，是与持久化数据进行交互的必经模块，在本系统中使用Django中的ORM与数据库进行交互。

用户登录系统之后，用户管理监控模块（仅管理员可进入）根据当前用户是否具有执行算法，进行数据集导入等权限在用户界面模块限制用户的相关操作。用户选择不同的数据集和参数设定，可看见设定的拓扑结构的可视化，接下来模块读取数据集信息，并根据设定参数进行融合算法，最后将融合后网络结果以及相关评价指标根据用户选择在视图层进行渲染展示；用户进入数据集管理模块（仅管理员可进入）后，选择导入的数据文件将数据集导入数据库或者进行数据集的删除操作；系统日志记录模块记录当前用户的相关操作，例如登陆、执行算法等；数据访问模块根据系统相关信息的数据库存取。

4.2.2　系统接口说明

1. 外部接口

表4-1是系统各个模块的外部接口。具体接口的功能与参数见下表。

表4-1. 系统功能外部接口表

|  |  |  |
| --- | --- | --- |
| 接口名称 | 参数说明 | 功能说明 |
| login | Str name, Str passwd | 用户登录系统接口 |
| logout | Str name | 用户登出系统接口 |

1. 内部接口

下面5个表格列出了系统各个模块的内部接口。具体接口的功能与参数如下面几个表格所示。

表4-2. 社交网络融合算法模块内部接口表

|  |  |  |
| --- | --- | --- |
| 接口名称 | 参数说明 | 功能说明 |
| get\_network | Str choosen\_param | 社交网络拓扑结构展示结果 |
| run\_al | str task\_id, str algorithm | 算法任务执行接口 |

表4-3.算法实例管理模块内部接口表

|  |  |  |
| --- | --- | --- |
| 接口名称 | 参数说明 | 功能说明 |
| get\_task\_list | Str user\_name | 用户相关算法任务数据展示 |
| get\_task\_network | Str task\_id | 算法执行结果的查看 |
| del\_task | Str task\_id | 算法执行任务的删除 |

表4-4. 社交网络数据解析与管理模块内部接口表

|  |  |  |
| --- | --- | --- |
| 接口名称 | 参数说明 | 功能说明 |
| Set\_network | Str file\_name | 根据原始文件解析出社交网络结构 |
| Upload\_dataset | Str url, Str dataset\_id | 上传社交网络数据集 |
| Del\_dataset | str datatset\_id | 删除社交网络数据集 |

表4-5. 用户管理监控模块内部接口表

|  |  |  |
| --- | --- | --- |
| 接口名称 | 参数说明 | 功能说明 |
| add\_user | str user\_name, datetime Time, Str Detail | 添加用户 |
| del\_user | str user\_name, datetime Time, Str Detail | 删除用户 |
| Update\_user | str user\_name, datetime Time, Str Detail | 更新用户权限 |

表4-6. 系统日志记录模块内部接口表

|  |  |  |
| --- | --- | --- |
| 接口名称 | 参数说明 | 功能说明 |
| logconfig | Str filepath, str detail, datetime time | 操作记录日志 |

4.2.3　各模块交互关系

根据各模块之间的功能交互得到一下的各模块之间的交互关系。

1. 多社交网络融合算法模块与社交网络数据解析模块以及数据访问模块交互



图4-3. 融合算法模块与社交网络数据解析模块以及数据访问模块的交互图

用户需要选择一定的源社交网络和目标社交网络的拓扑结构进行对齐，用户选择了数据集，融合算法需要从数据库中读取社交网络数据集并进行解析生成可使用的社交网络数据结构，再执行对齐算法。

Read\_file\_network()是数据访问模块根据参数从数据库中取出相应社交网络数据集数据返回给社交网络数据解析模块，Set\_network ()是在数据解析模块中根据请求参数解析返回社交网络数据结构返回融合算法模块。

1. 算法实例管理模块与数据访问模块交互

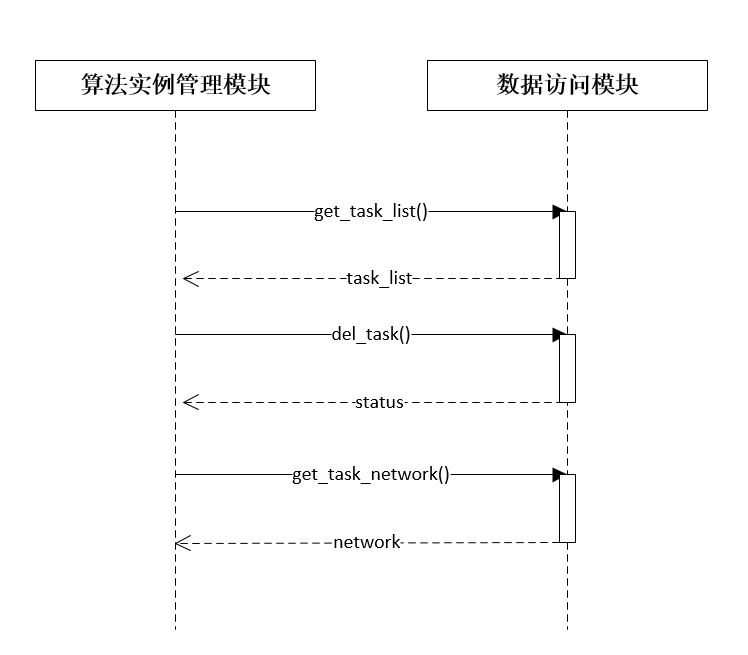


图4-4.算法实例管理与数据访问模块交互图

算法实例管理模块需要数据库中读出算法任务执行状态和相关信息，并且可以执行用户查看对齐结果和删除任务等操作。

get\_task\_list()方法返回算法执行任务列表和状态，del\_task()执行删除指定任务的执行，get\_task\_network()返回算法执行结果即对齐后社交网络的信息给算法实力管理模块。

1. 网络拓扑可视化模块与算法实例管理模块的交互



图4-5.网络拓扑可视化模块与算法实例管理模块交互图

网络拓扑可视化模块需要从算法实例中获取结果数据进行可视化显示。

get\_network\_show()是从算法实例结果中中获取用于可视化显示的数据。

1. 用户管理监控模块与数据访问模块交互

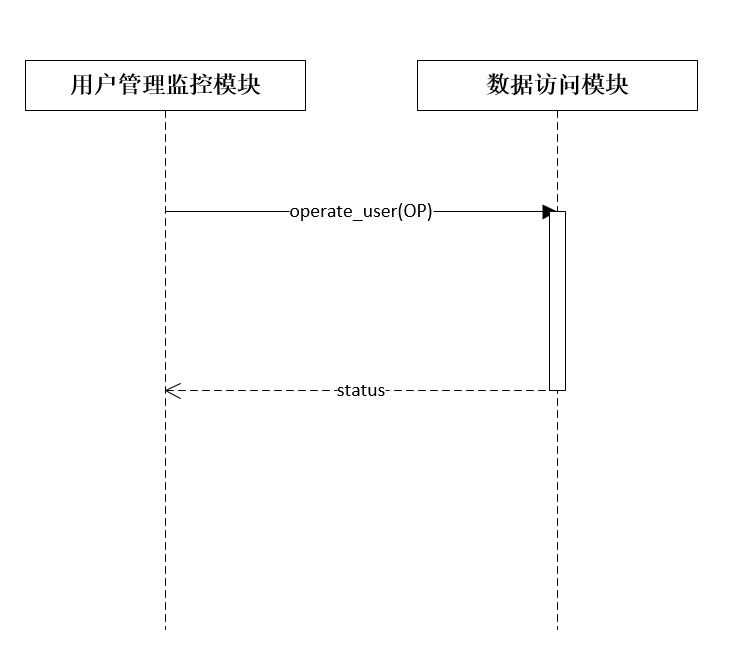


图4-6. 用户管理监控模块与数据访问模块交互图

用户管理监控模块执行增加、移除用户、编辑用户权限的操作，数据访问模块进行用户数据的同步更新。

Operate\_user(OP)方法中OP代表对用户的增删改查操作指令，该方法在数据访问模块中完成对于用户的操作，返回操作成功值。

1. 系统日志记录模块与数据访问模块交互

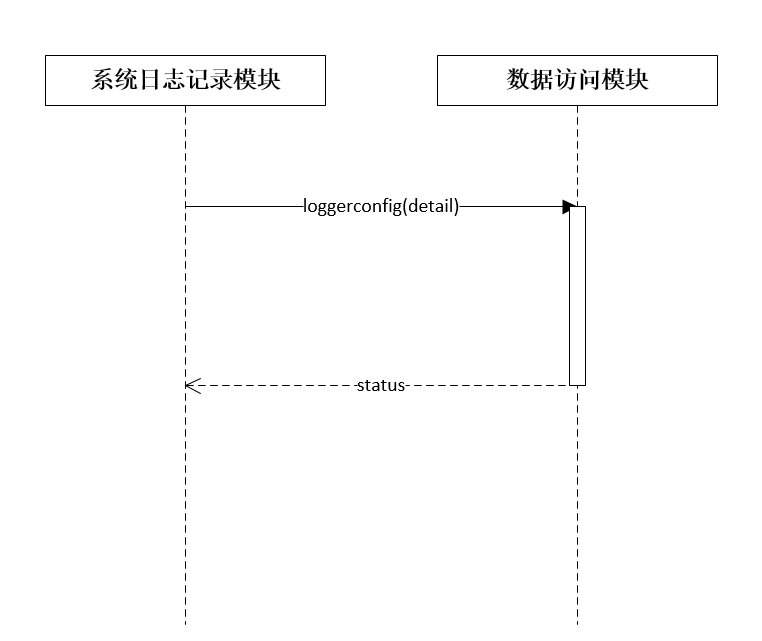


图4-7. 系统日志记录模块与数据访问模块交互图

系统日志记录模块向数据访问模块提交要保存的操作记录，数据访问模块实现操作信息的保存。

Loggerconfig(detail)方法向数据访问模块提交要保存的系统日志。

4.3　系统算法设计与实现

本系统中的算法设计和实现分为两种，用户自定义和本文中重点提出的关键算法ISS（Iterative Supervised framework with Strict stable matching）。关键步骤包括，用户对的特征提取，分类训练，最后对于训练结果选择稳定匹配算法得出最终预测结果。

4.3.1　多社交网络对齐问题定义

多社交网络融合对齐问题源自于多种社交网络提供的服务的多样性，以至于大部分的用户都会在不同的社交网络中拥有账号，根据之前关于社交网络对齐的研究[MNA]提出的概念，接下来将这些拥有不同账号的用户定义为锚用户，并且将同一用户的不同社交网络账号之间的链接称之为锚链接。但是并不是所有的用户都在不同的社交网络中都拥有账号，[PNA]提出的部分对齐社交网络也是本系统中算法研究的基础，在部分对齐社交网络中锚用户和非锚用户同时存在。为了方便展示和计算，我们专注于预测两个异构社交网络上的锚链接，以便扩展到更多社交网络。

首先将社交网络建模为，可以是任何类型的节点集，如用户、地理位置等，表示节点间的关系集，如用户间的关注关系和用户与地理位置间的签到关系等，即。将研究的两个部分对齐异构社交网络表示为和，并且，这里的拓扑图都是有向图。在本文的研究中，节点集定义为一组用户帐户，关系集表示用户间关注网络的拓扑信息，综上所述，源网络，目标网络 。

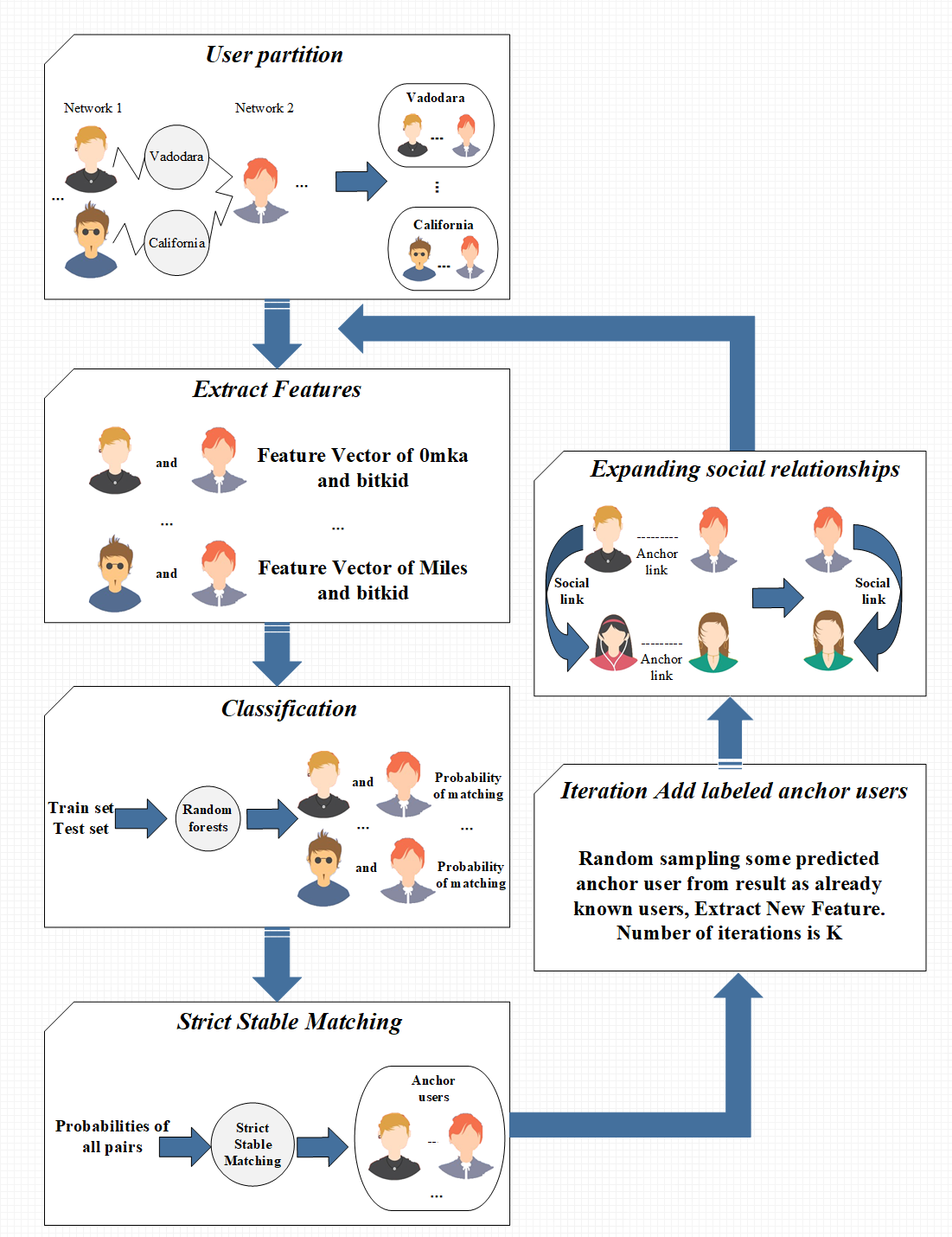
同时，在现实社交网络中，一部分用户会将自己不同社交网络中的账号进行关联，比如Foursquare的部分用户会将自己的Twitter账号标注在首页，或者用同样的手机号和邮箱地址进行注册等，这部分可被称之为已知锚链接，将源网络和目标网络间的已知锚链接定义为，表示两个用户账号属于同一个用户。同时，为了将对齐问题更能对应现实中的社交网络，问题研究的前提是，处于其中一个社交网络中的用户在另一个社交网络中有着至多一个用户账号，也就是说，没有任何两个锚链接的端点用户账号重合，问题的研究上加入了<= 一对一的限制条件。

根据以上定义，建立部分对其异构社交网络模型{ (, ), }，目标是预测出未知锚链，这个问题是一个链接识别问题，进一步得说明，是通过已知锚链接的特征，识别锚链接和非锚链接，所以对齐问题也是一个二分类的问题。综上所述，接下来将建立预测模型解决上述定义的多异构社交网络之间的部分对齐问题。

4.3.2　ISS的算法流程

基于在源网络和目标网络中的已知锚链接，现在已有的研究有。。。。。。

算法的核心是一个基于监督学习和严格稳定匹配的迭代框架，旨在解决更准确地在部分对齐社交网络中预测出锚链接。



1. 网络扩充

对于用于研究的数据，一般都会因为很多外在原因使得根据获得的数据构造的信息不完整，所以需要根据已知信息完善社交网络。算法根据已知的锚用户，基于一定的经验和规律，对于源社交网络和目标社交网络中的社会关系进行扩充，通常情况下,如果两个节点在一个网络中没有连接关系,但它们在另一个网络中对应锚节点间却存在连接关系,则可以在当前网络中给这两个节点间添加一条边。具体流程如图

1. 特征提取

为了使模型更具有泛化能力，可以在适用于不同的异构社交网络，算法选择的特征应基于网络的拓扑结构，即用户间的社交关系。根据在不同的社交网络中的用户账号可产生样本用户对，这一部分的目标在于选择计算用户对的特征，在已有的研究中，MNA提出的社交，地理时间以及语言特征使得其模型仅限于模型限制性很大，为了模型实验于Foursquare和Twitter这两个数据集但是并不仅限于这两个数据集，摒弃这些依赖于异构社交网络的独有属性产生的特征，比如地理位置等。

首先，定义在社交网络模型这一有向图中的“邻居”概念，在社交网络中普遍存在的有单向关注关系和双向关注关系，而在不同的社交网络中，强关系和弱关系

接下来利用在MNA中提出的扩展社交特征，也就是，在现实社交中关系紧密的一群人，在不同的社交网络中都有很大的可能性是联系在一起的，所以算法根据已知的锚用户，在他们各自的提出的特征概念如下。

共同邻居：

Joccard’s coefficient:

Adamic/Adar Measure:

除此之外，根据图论中的中心性（centrality）和威望（presitige），可得出第二种特征，用户（）的相似性，基于现实生活中用户社交范围的一定性，不同账号间的

作为辅助特征，在计算

1. 分类训练

在步骤（2）中提取特征后，接下来可根据已知的锚链接和非锚链接训练分类器。已知有很多监督学习的分类算法可供选择，根据之前相关研究提出的结论[Link classification]，一开始选择了SVM（rbf作为核函数）作为分类器模型进行训练，为了比较，选择了随机森林和adaBoosting集成算法进行训练。在不同规模的数据集上训练得出adaBoosting的效果比其它分类器的效果更好，这得益于adaBoosting的迭代原理，针对同一个训练集训练不同的弱分类器（决策树），通过每次训练的正确率修改样本权值，最后将每次得到得分类器融合起来作为最后得决策分类器。

但是，据上一部分对于多社交网络对齐问题的定义，本问题有<=一对一的限制条件，而分类器的直接预测的结果并不能满足这一前提条件，所以为了下一步进行稳定匹配，这一步分类器预测结果将输出用户对匹配概率。

1. 严格稳定匹配

在[MNA]和[PNA]中分别提出了稳定匹配（Stable Matching）和泛化稳定匹配（General Stable Matching）。

稳定匹配：这个匹配算法受稳定婚姻匹配算法的启发，

泛化稳定匹配：

对于以上已有的匹配算法分析如下，1.传统稳定匹配算法，只能在全对齐社交网络中有比较好的表现结果。2.对于在现实生活中普遍存在的非全对齐的社交网络，泛化稳定匹配算法中的K值的选择并不好确定到最佳的情况，并且每个用户的匹配候选链的K值越大，即配对的分数越宽容。

基于以上分析提出严格稳定匹配算法（Strict Stable Matching）

严格稳定匹配：

具体匹配过程如下图所示

1. 迭代框架

由于只有小部分数量的锚用户已知以及分布的稀疏性，在一次严格稳定匹配之后，只有一部分相对可靠的锚链接形成，剩下的用户因为特征的稀疏性将难以区分，比如特征都是0，所以使用算法中提出的迭代框架改善这种情况。

根据一轮严格稳定匹配结果的高可能性，从结果中随机抽样一部分加入已知的锚链接集A， 随机抽样的参数为γ，代表加入已知锚链接集的数量占总预测结果数量的百分比，接下来根据更新的已知条件重新提取特征，训练分类器，严格稳定匹配得出结果，这样不断用匹配输出的结果抽样加入已知锚链接集，训练配对输出预测结果，直到预测结果数量小于值d, 迭代结束，将多次预测结果取并集作为最终的预测结果。

这个迭代框架，能够使以上方法更好得适用部分社交网络的对齐，在不断的迭代和扩展中，充分利用已知的锚节点信息。再更形象地说明一下，已知John对应得锚用户，通过这个锚链接训练配对得出Jack对应的锚用户，接下来根据John和Jack这两个对应的锚链接信息，训练匹配出更多的锚链接，直到迭代结束提高，这样可以得到尽可能多的锚用户信息和结果的准确性。

4.3.3　算法实现及结果分析

为了证明所提出的方法在部分对齐的异构社交网络的锚链接预测中的有效性，接下来将在两个异构社交网络上进行实验：Twitter（社交网络）和Foursquare（基于位置的在线社交网络），这个 我们将介绍数据集和实验过程。

1. 数据集描述

下表展示了已有的两种数据集的规模，文件解析构建之后，可得到以下两种异构社交网络，其中twitter包括5226个节点和164920个有向边，foursquare包括5392个节点和76972个有向边，并且有3282个锚用户存在于这两个社交网络中。

|  | **Network** | | |
| --- | --- | --- | --- |
| ***Property*** | ***Foursquare*** | ***Twitter*** |
| # node | Users | 5392 | 5226 |
| # directed link | Neighbors | 76972 | 164920 |
| # anchor node | Users | 3282 | 3282 |

1. 实验配置

首先，需要将从两个不同的异构社交网络中产生的用户账号对进行训练，如果不加以限制，寻找其中一个网络中的用户账号的配对，将在另一个网络中的所有用户账号进行匹配，这样会导致候选集过大，搜索效率低等问题。所以在产生配对数据集前，首先进行一个用户分区操作，当两个用户账号在同一个城市有过签到足迹才组成用户账号对。即利用异构社交网络的共有属性进行候选集的修剪，这个属性可以是地理位置也可以是其它社交网络的共有属性。

接下来根据用户分区产生的用户账号对，首先将所有的3282个锚链接加入正集，而所有其它的用户账号对放入未知链接对，从未知链接对中取样作为负链集，对于正集和负集都分为5折，使用5折交叉验证进行实验，不同于其它的5折交叉验证的方法，一般现实世界的异构社交网络中只有少部分的锚链接是已知信息，所以只使用一折作为训练集，其它的作为测试集，并且从训练集中随机取样数量α的链接作为已知锚链接。

由于数据样本通常是不平衡的，这就意味着，负集可以比正集大很多倍，这样会极大影响分类训练的结果，同时，根据其他论文，为了克服这种不平衡，可以采用欠采样，采样率为**δ**，其中**δ**= 0.1表示0.1 \*（负集 - 正集）负链接随机从负链接集中移除。

在框架ISS中，**η**表示迭代次数，**γ**是每一轮结果的百分比，白哦是从结果中随机抽样，并将其添加到已知的锚链集中。

1. 算法表现评估

* 迭代次数对结果的改进
* 将ISS与其它的已有研究方法比较

4.4　系统模块设计与实现

4.4.1　多社交网络融合算法模块

该模块负责设定参数，并根据生成的社交网络数据结构，执行指定多社交网络融合算法。

模块示意图如下图所示：



图4-14. 多社交网络融合算法模块示意图

模块类图如下图所示：

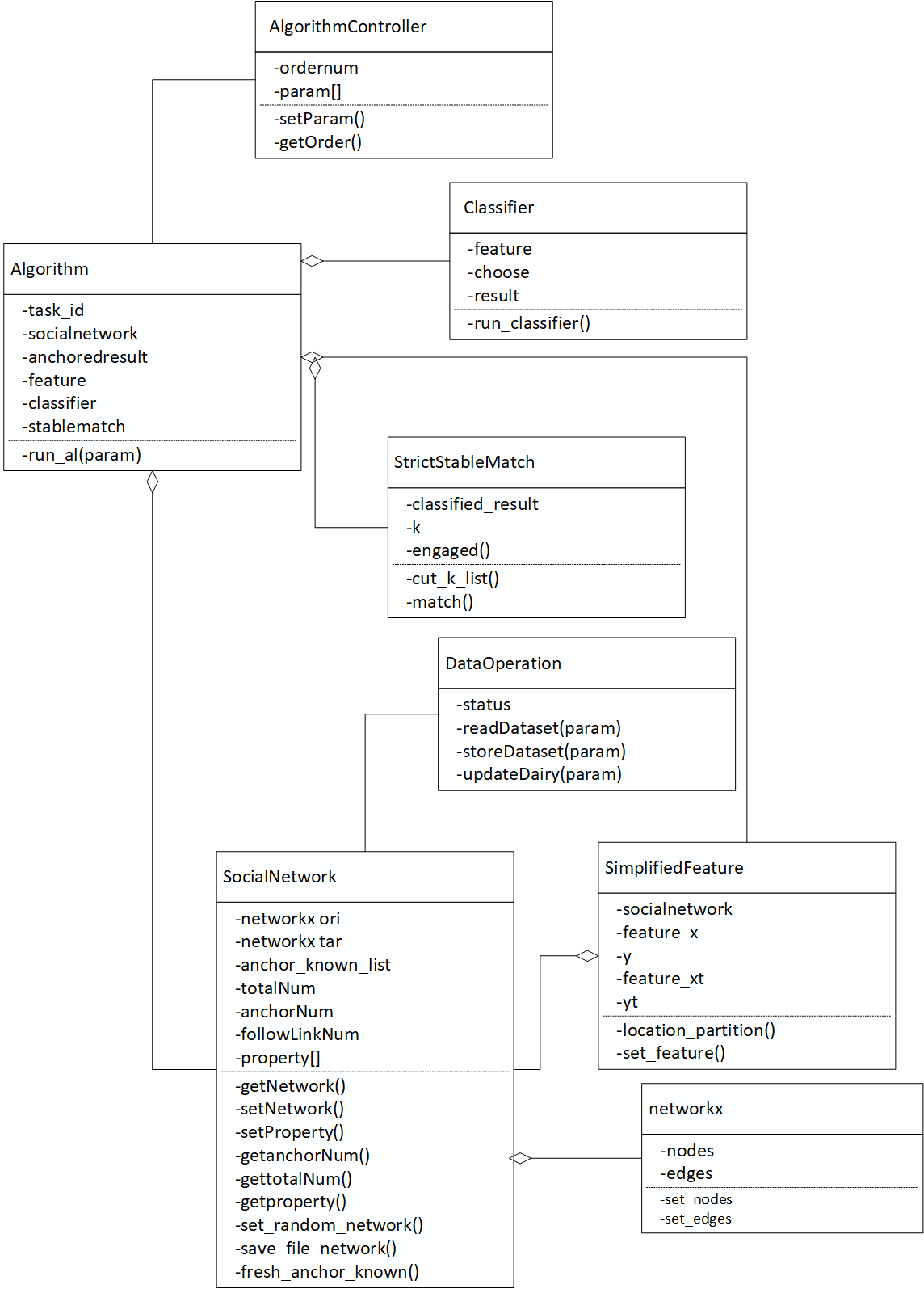


图4-15.多社交网络融合算法模块类图

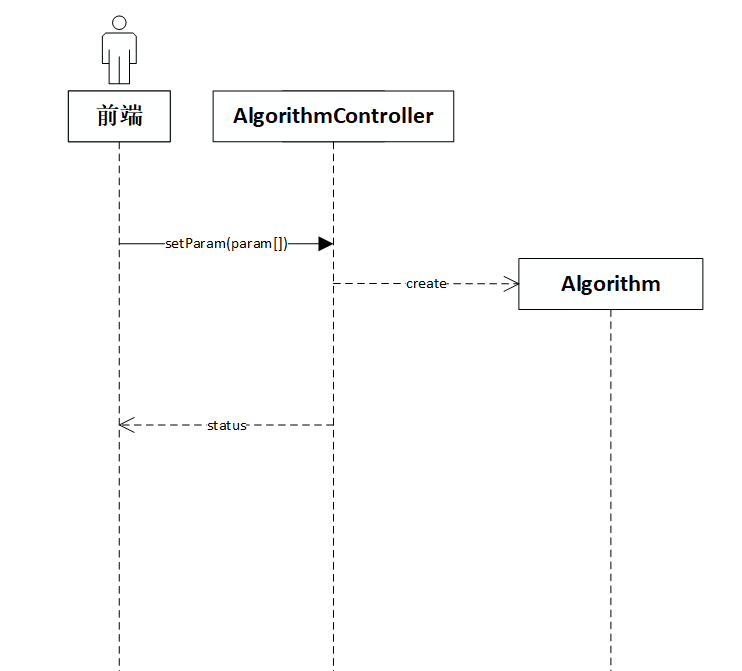
其中SocialNetwork是进行对齐的社交网络数据结构，其中networkx是使用python中networkx类对网络的拓扑结构以及相关节点信息进行存储表示，SocialNetwork中有用于对齐的源网络和目标网络，以及已知锚节点anchor\_known\_list，除此之外，社交网络数据集需要Dataoperation从数据库中读出。Algorithm是自定义的算法任务类，包括特征提取simplifiedFeatue，分类器classifier，稳定匹配算法StrictStableMatching这三个算法步骤类，可根据不同的参数进行算法结构的调整，包括ISS算法和自定义算法的形成。

AgorithmController负责接受用户的指令和输入的参数，生成对应的算法任务。

该模块各功能的系统顺序图如下：

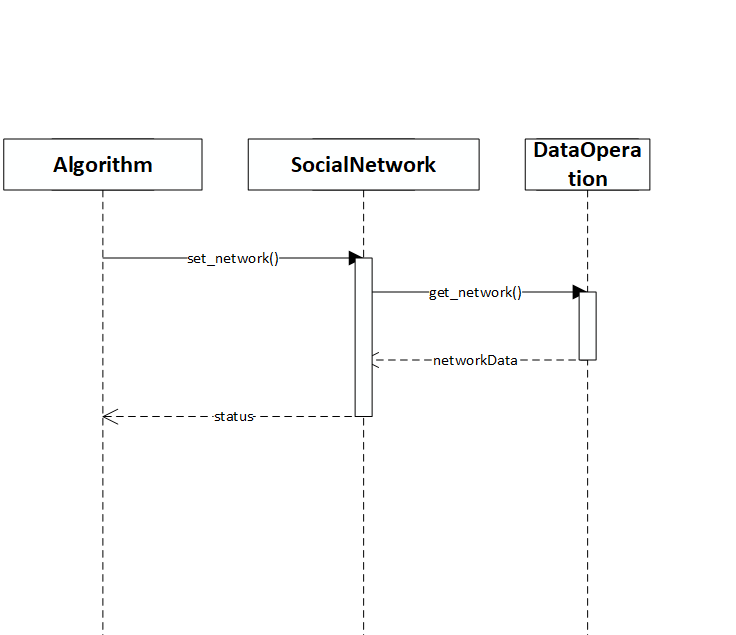
1. 设置参数

用户可在下拉框中选择不同的社交网络数据集，以及不同的算法参数，比如自定义的算法各种分类器和稳定匹配算法的选择，以及ISS的参数确定等。



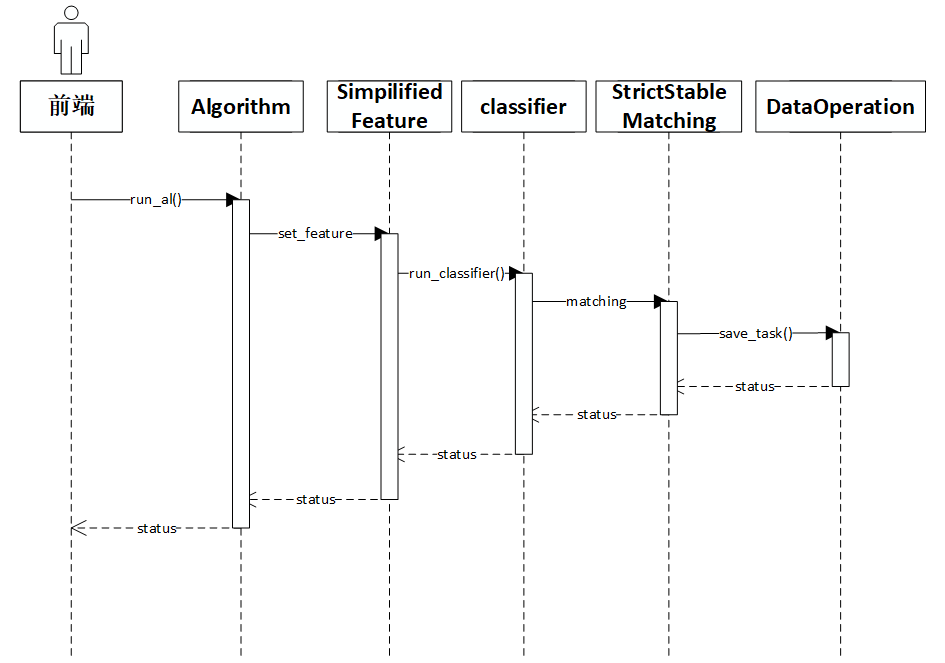
1. 社交网络生成

根据用户选择的社交网络数据集，从数据库中读取源网络和目标网络相关文件并且生成对应的社交网络数据结构。



1. 算法执行

用户选定参数后，选择算法执行，生成对应的算法任务线程，并把相关状态保存到数据库中。用户可以看到算法任务启动成功或者失败的返回信息，由于一般对齐算法都需要一定的执行时间，所以算法执行结果以及状态可在算法实例管理模块中查看。



4.4.2　算法实例管理模块

该模块根据管理员id查看相关算法实例的执行状态，包括算法执行前后网络拓扑结构，以及对齐结果和指标的查看。

模块示意图如下图所示：



图4-17.算法实例管理模块示意图

模块类图如下图所示：

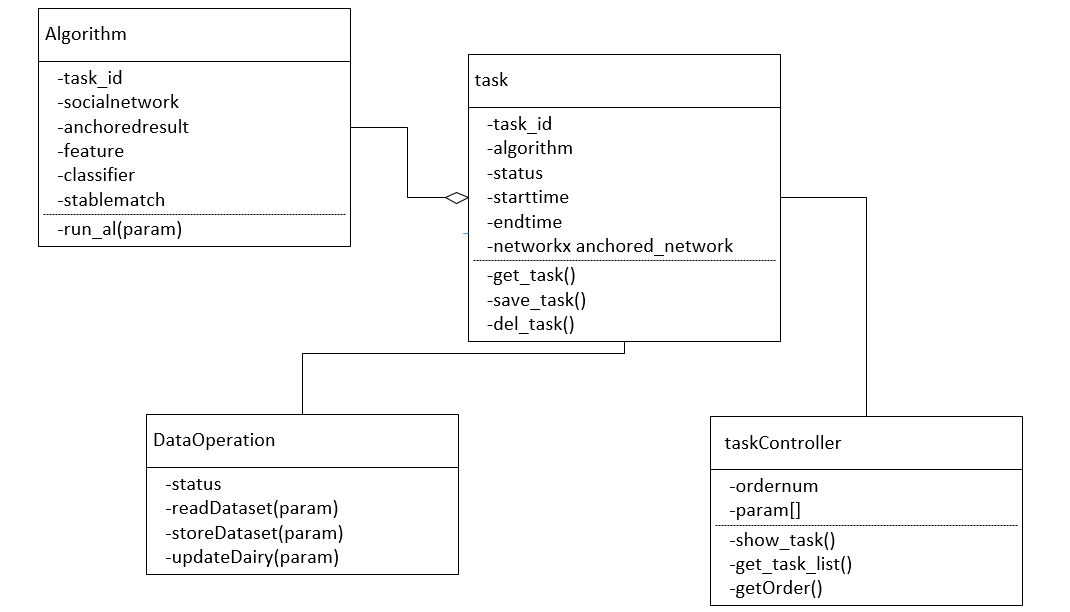


图4-18. 算法实例管理模块类图

如图4-18所示，task是算法任务类，管理算法线程的执行状态，包括对齐后结果和相关评价指标，taskController是接收用户对算法任务的删除或者查看命令，查看算法执行结果以及相关指标。

模块顺序图如下图所示：

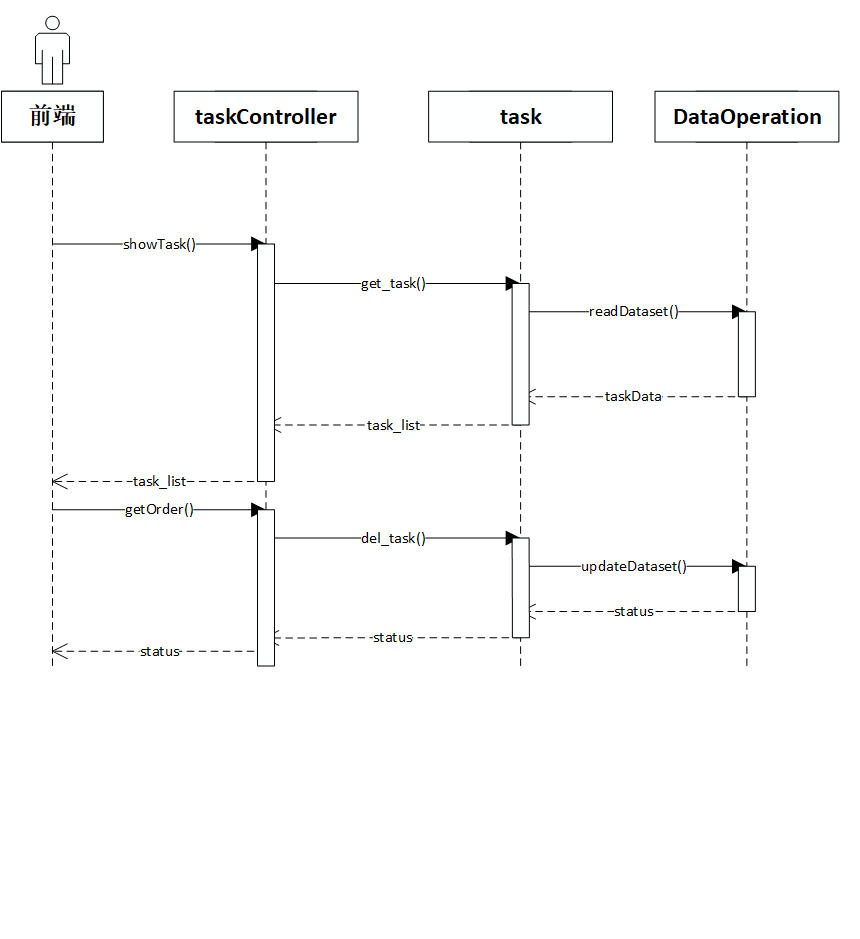


图4-19. 算法实例管理模块顺序图

用户可查看相关的算法任务列表以及其状态，有运行中和已获得结果两种状态，对于已运行完成的算法任务，可查看对齐后社交网络和算法结果的相关评价指标。另外，用户也可以对任务列表中的实例进行删除操作。

4.4.3　社交网络数据集解析与管理模块

该模块负责对于读入的社交网络原始数据进行解析，并生成结构化的可被算法以及可视化所使用的社交网络数据。并且该模块的功能社交网络文件管理（仅管理员可进入），提供管理可供执行算法的原始数据集的导入以及删除的功能，比如Foursquare，Twitter等。

模块示意图如下图所示：



图4-20. 社交网络数据集解析与管理模块示意图

模块类图如下图所示：

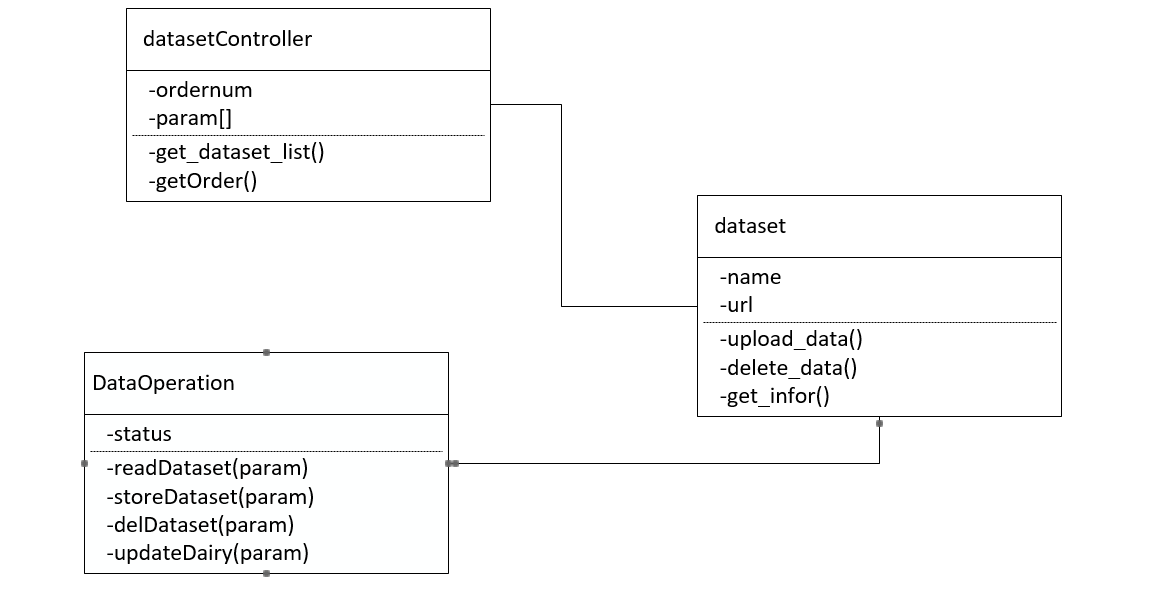


图4-21. 社交网络数据集解析与管理模块类图

如图4-21所示，datasetController用于接受用户对于数据集操作的请求，dataset类包括数据集名字和存储地址url，在读取数据集列表的时候，get\_infor()解析返回数据集信息。当用户（管理员）选择上传或者删除数据集时，dataset中的upload\_data()和delete\_data()通过数据访问层的操作数据库中的数据集存储。

模块顺序图如下图所示：

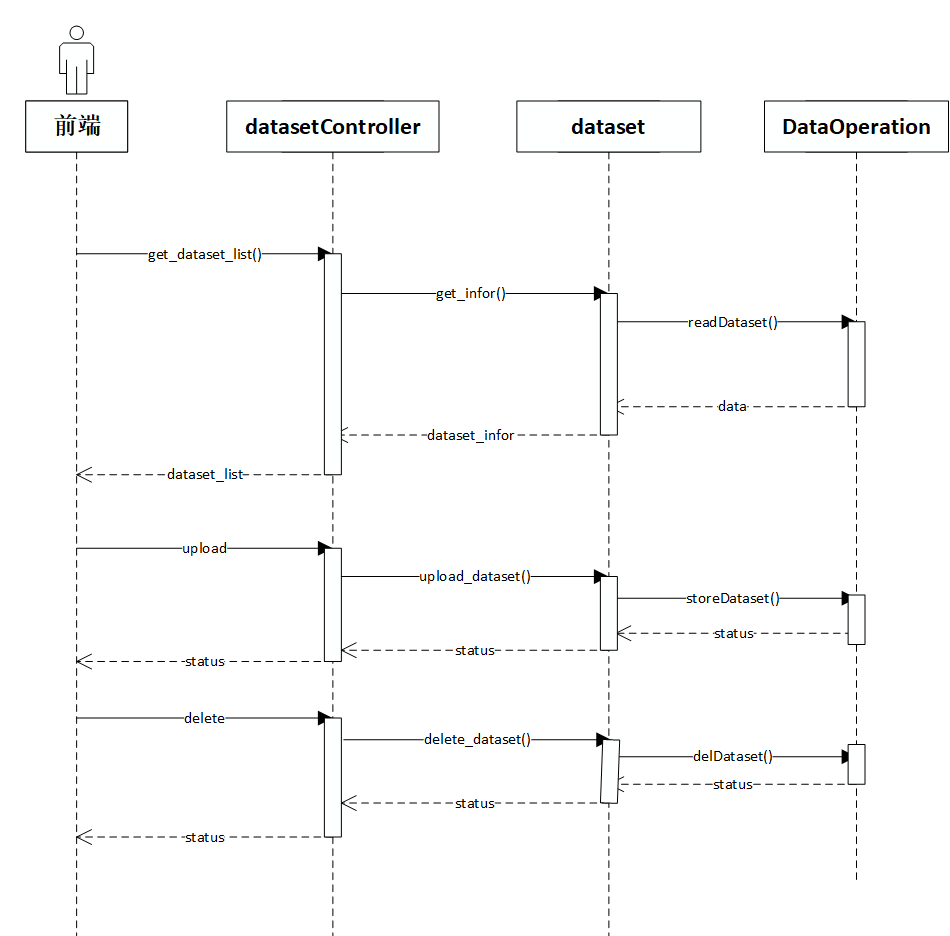


图4-22. 社交网络数据集解析与管理模块顺序图

4.4.4　网络拓扑可视化模块

该模块是处理用户对于社交网络拓扑结构查看的命令，用于系统中社交网络拓扑结构的可视化，过程是根据得到的拓扑结构数据进行用户界面的展示，这里由vis.js进行展示。

模块示意图如下图所示：



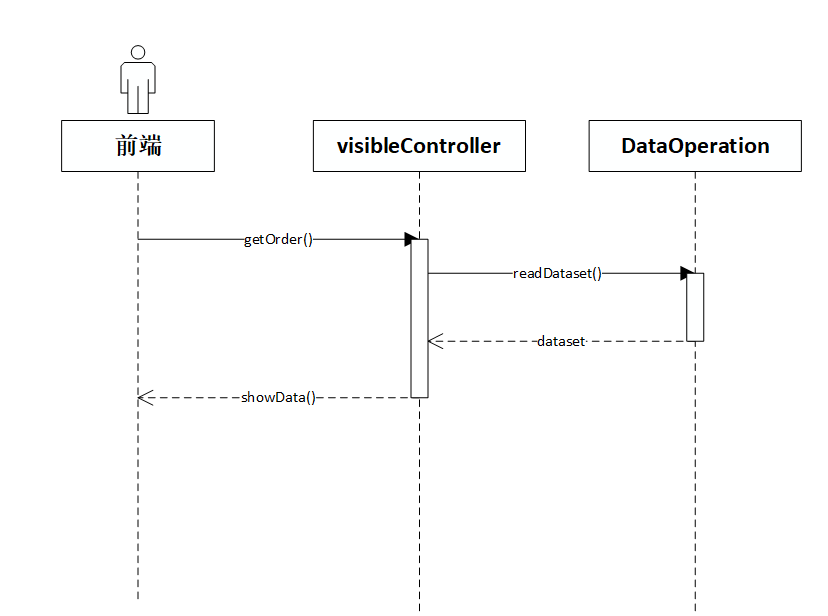
图4-23.网络拓扑可视化模块示意图

模块类图如下：



visibleController是用于处理可视化指令，order代表得到的可视化配置，get\_network\_topology()通过数据访问层得到相关展示数据，最后返回将社交网络拓扑结构（对齐前后等）展示在用户界面上。

模块顺序图如下图所示：



4.4.5　用户管理监控模块

该模块仅限管理员进入，管理员可进行用户的增加、删除，修改用户的密码及权限。在本系统中前端Vue.js根据用户登录时返回的权限信息存储在sessionStorage中，前端页面跳转根据sessionStorage的相关权限信息进行处理。

模块示意图如下图所示：



图4-24. 用户管理监控模块示意图

模块类图如下图所示：

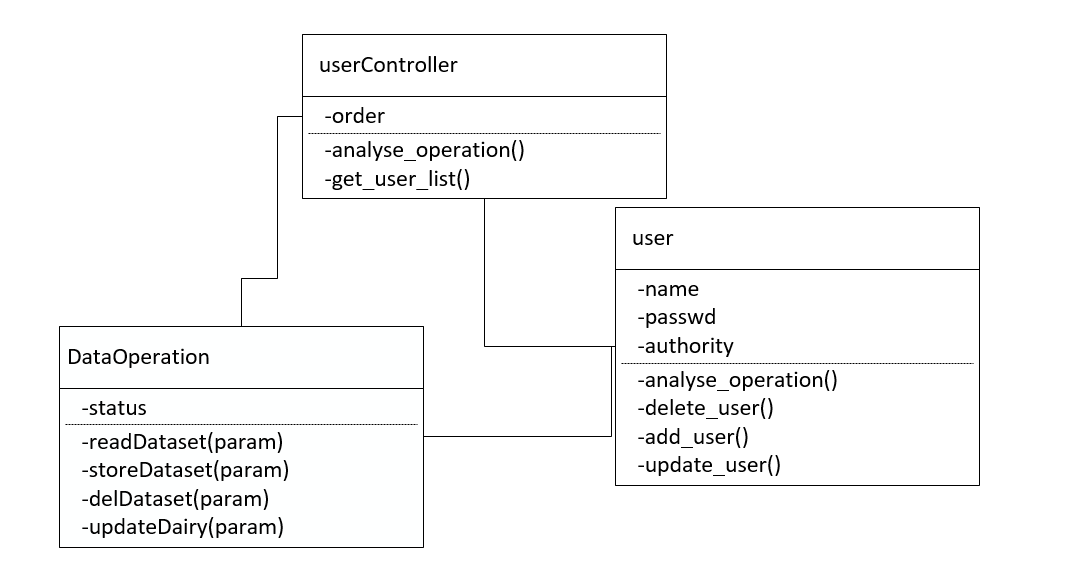


图4-24. 用户管理监控模块类图

userController接受管理员对于用户的修改命令进行操作，user类中包括用户的登录名，密码以及权限，权限分为管理员权限和普通用户权限。

模块顺序图如下图所示：

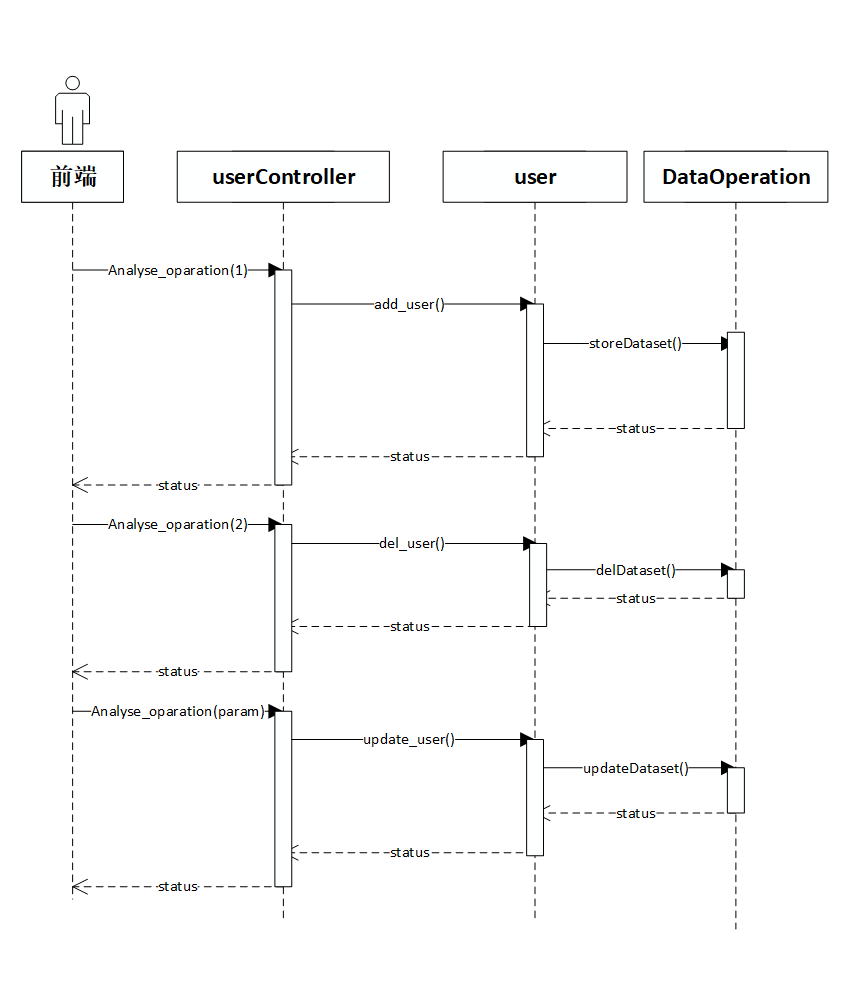


图4-25. 用户管理监控模块顺序图

管理员在用户管理界面实现用户的增加、移除以及对用户权限的改变。

4.4.6　系统日志记录模块

该模块用于记录用户的各种操作及发生时间，例如登入、登出、执行算法、查看社交网络等。根据python的logger模块进行设置，便于将所有相关的操作进行系统监控。



图4-26. 系统日志记录模块示意图

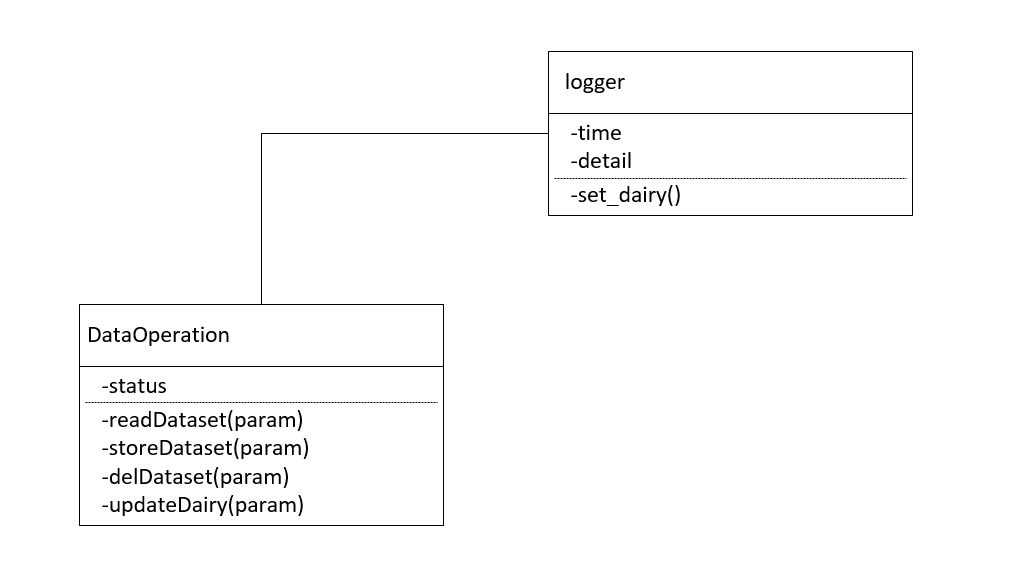


图4-27. 系统日志记录模块类图

如图4-27所示，logger类包括日志的时间和内容，通过数据访问类updateDairy()存储。

本模块为日志查询模块服务，提供系统操作的记录以管理员查看，当用户在进行任何操作时，任何控制器通过getOrder()接受到命令后，都会调用logger的set\_daity产生日志记录。

模块顺序图如下图所示：

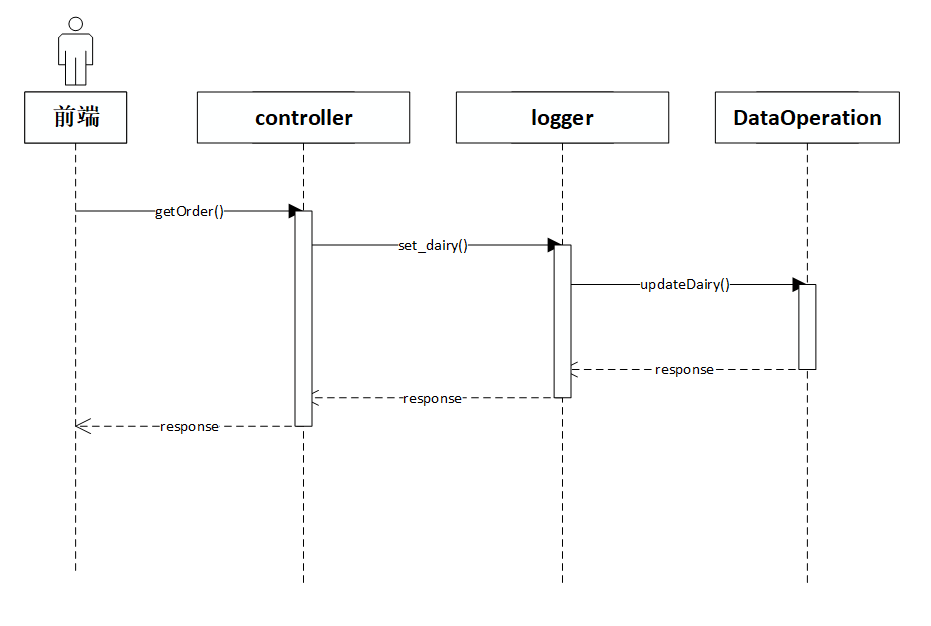


图4-28. 系统日志记录模块顺序图

4.5　数据库设计与实现

系统采用了django中的ORM实现数据库表与实体类之间的对应关系，它是通过创建 SQL 去查询和操作数据库的一个 Python 式方式，并且可以获得 Python 风格的结果。多社交网络关联融合系统数据库包含3个表：存储用户信息的user表，存储算法任务状态以及结果的task表，存储社交网络数据集以及数据路径的dataset表，具体的结构化社交网络数据集以gpickle文件格式存储，方便系统中通过networkx进行读取。在以及存储用户日志表logger，如下图所示：

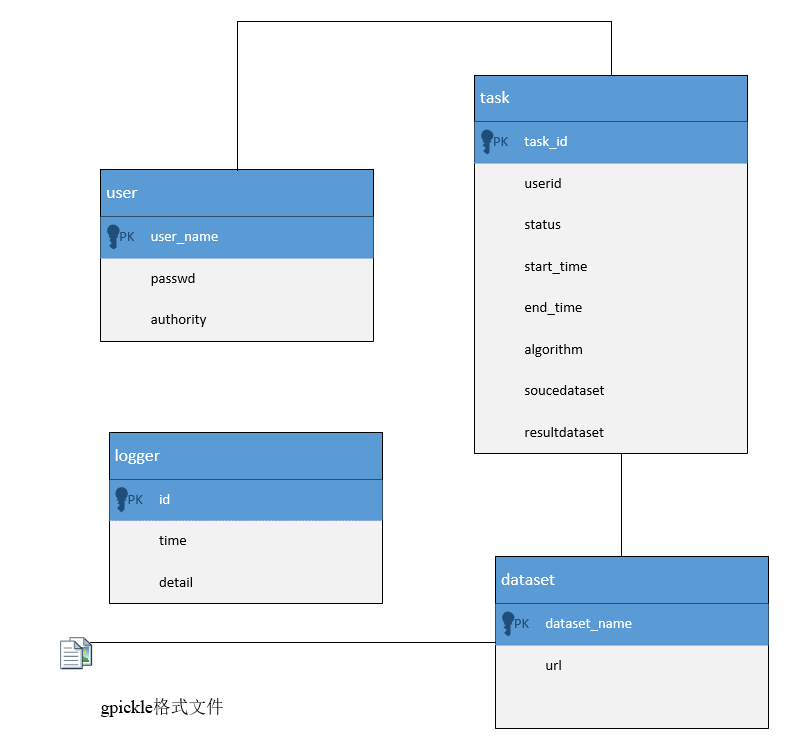


图4-29. 数据库设计图

下面对这3个表的详细设计和使用进行说明。

1. 用户信息表user

该表主要记录系统中所有用户的信息。

表4-9. 用户信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 约束 | 是否为空 | 字段描述 |
| user\_name | char(20) | 主键 | 否 | 用户名 |
| passwd | char(20) |  | 否 | 密码 |
| authority | boolean |  |  | 用户权限 |

1. 算法任务表task

该表主要记录了算法任务的执行状态以及相关数据集

表4-10. 算法任务信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 约束 | 是否为空 | 字段描述 |
| task\_id | int | 主键 | 否 | 任务id |
| userid | user | 外键 |  | 算法任务执行者 |
| start\_time | datetime |  |  | 任务开始时间 |
| end\_time | datetime |  |  | 任务结束时间 |
| Algorithm | json |  |  | 算法配置信息 |
| soucedataset | dataset | 外键 | 否 | 对齐前社交网络 |
| resultdataset | Dataset | 外键 | 否 | 对齐后结果 |

1. 社交网络数据集信息表dataset

该表记录了系统中存在的所有社交网络数据集

表4-11. 数据集信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 约束 | 是否为空 | 字段描述 |
| dataset\_name | char(20) | 主键 | 否 | 数据集名称 |
| url | char(50) |  | 否 | gpickle文件路径 |

1. 系统操作日志信息表logger

该表记录了操作详细信息。

表4-12. 系统操作日志信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 约束 | 是否为空 | 字段描述 |
| id | int | 主键 | 否 | Id |
| Detail | varchar(100) |  |  | 系统操作详情 |
| time | datetime |  | 否 | 系统操作时间 |

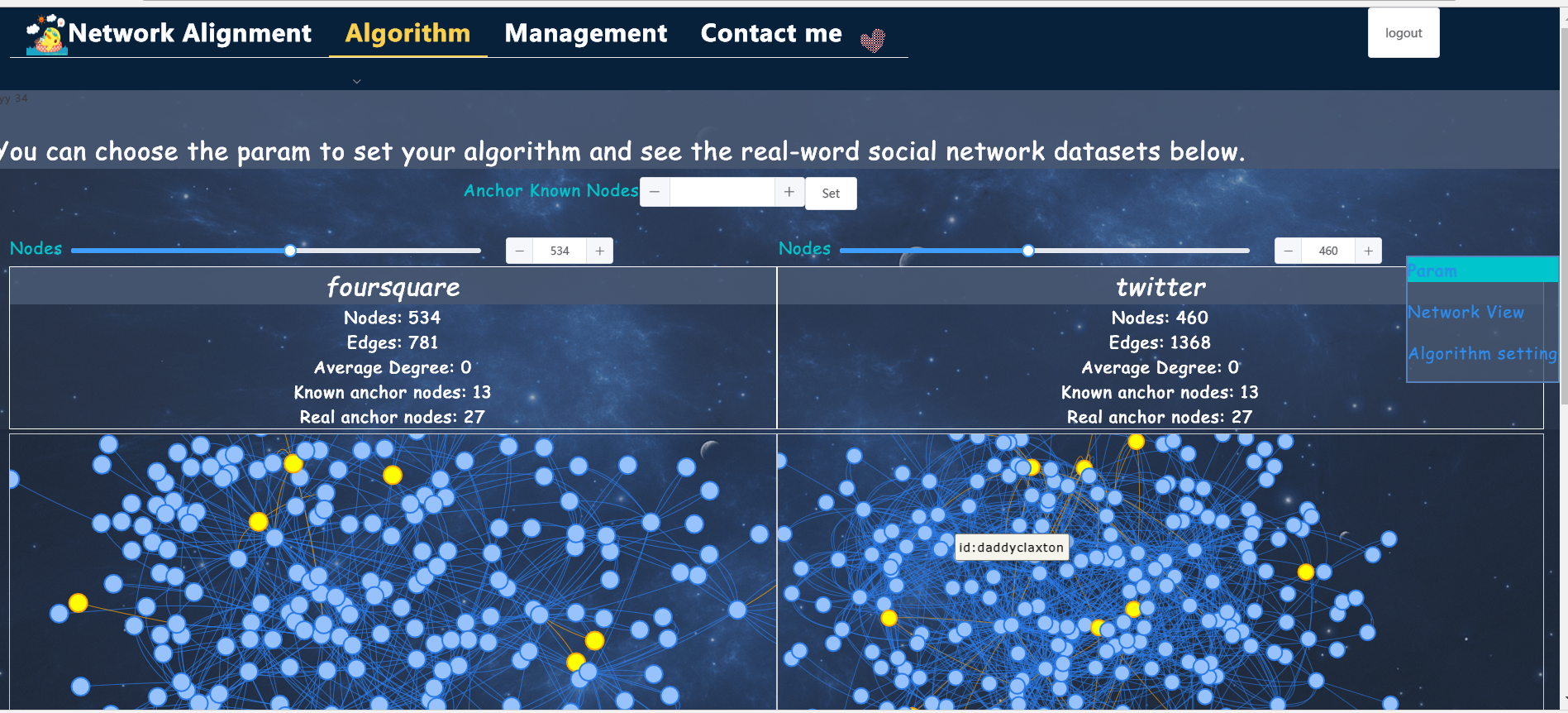
4.6　本章小结

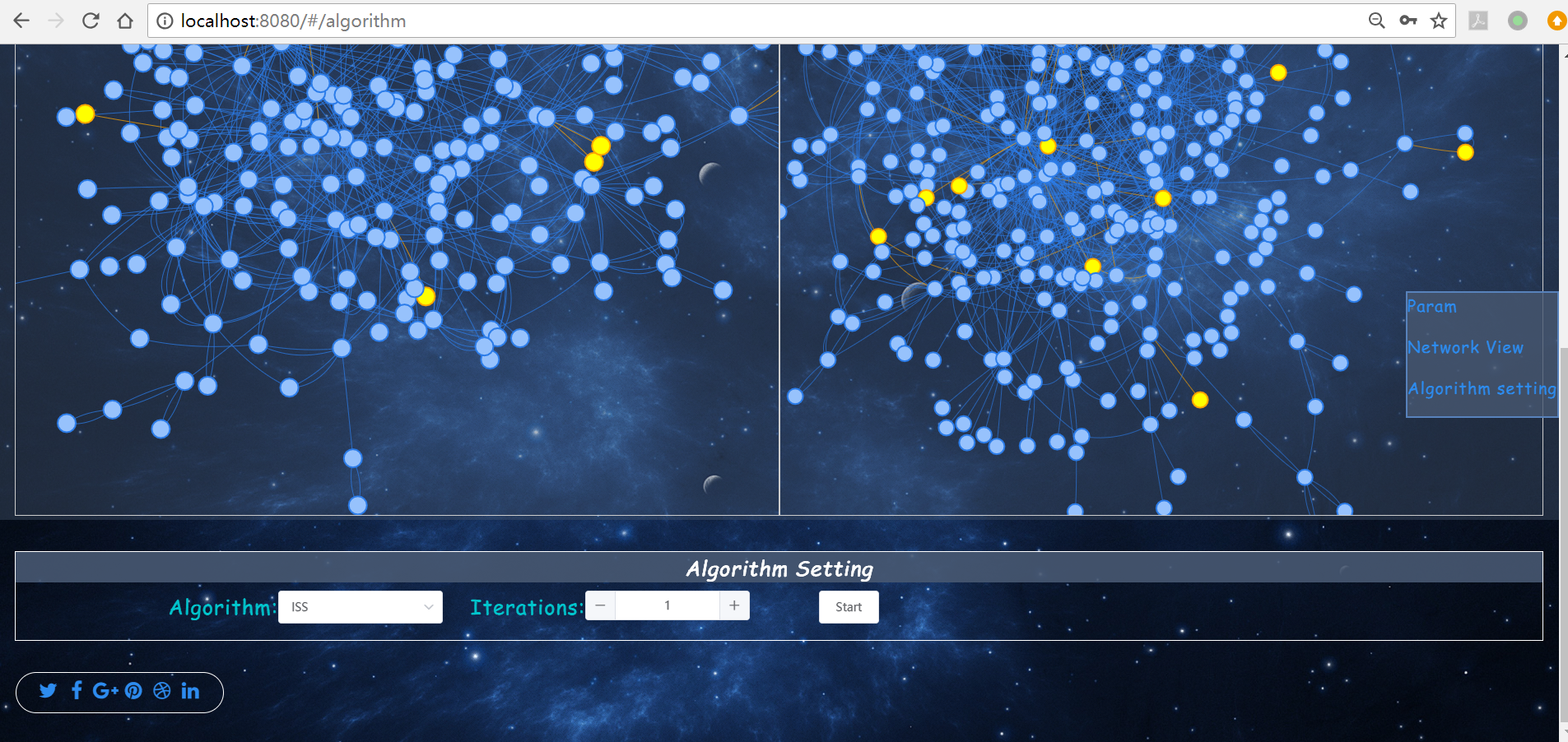
本章可称为多社交网络关联融合系统设计与实现，首先通过构建系统架构图和系统功能模块框图确定系统的模块构建与各个模块的功能，进一步确定模块间交互关系和内部接口。重点是分析模块的动态功能，通过系统顺序图绘制每个模块的功能，并随之确定模块的静态类图，为系统编码提供了清晰的思路。

本章的另一个重点在于系统关键算法的设计与实现，提出ISS（Iterative Supervised framework with Strict stable matching）的迭代框架，通过特征提取、分类器训练、严格稳定匹配进行多社交网络的对齐，重点通过分类器的改进以及迭代框架的加入准确最大化网络对齐的结果，除此之外，算法还使用了欠采样、候选集剪枝等方法在在训练过程中对于结果进行改进，并将一定的实验结果与已有的研究方法进行比较，得出ISS对于多社交网络对齐结果是有一定的改进作用。

最后各项数据的存储，将与系统需求相对应分为多个表进行存储，详细描述了数据表的各个字段的存储，系统中通过django自带的ORM进行操作。

第五章　系统测试





参考文献

1. Adamic, Lada A., and Eytan Adar. "Friends and neighbors on the web." *Social networks* 25.3 (2003): 211-230.
2. Zhang, Jiawei, and S. Yu Philip. "Integrated Anchor and Social Link Predictions across Social Networks." *IJCAI*. 2015.
3. Al Hasan, Mohammad, et al. "Link prediction using supervised learning." *SDM06: workshop on link analysis, counter-terrorism and security*. 2006.
4. Zhang, Jiawei, Philip S. Yu, and Zhi-Hua Zhou. "Meta-path based multi-network collective link prediction." *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014.
5. Tan, Shulong, et al. "Mapping Users across Networks by Manifold Alignment on Hypergraph." *AAAI*. Vol. 14. 2014.
6. Zhang, Jiawei, et al. "Pna: Partial network alignment with generic stable matching." *Information Reuse and Integration (IRI), 2015 IEEE International Conference on*. IEEE, 2015.
7. Kong, Xiangnan, Jiawei Zhang, and Philip S. Yu. "Inferring anchor links across multiple heterogeneous social networks." *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. ACM, 2013.
8. Menon, Aditya Krishna, and Charles Elkan. "Link prediction via matrix factorization." *Joint european conference on machine learning and knowledge discovery in databases*. Springer, Berlin, Heidelberg, 2011.
9. Bayati, Mohsen, et al. "Algorithms for large, sparse network alignment problems." *Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on*. IEEE, 2009.
10. Li, Chung-Yi, and Shou-De Lin. "Matching users and items across domains to improve the recommendation quality." *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014.
11. Man, Tong, et al. "Predict Anchor Links across Social Networks via an Embedding Approach." *IJCAI*. Vol. 16. 2016.
12. Dong, Yuxiao, et al. "Link prediction and recommendation across heterogeneous social networks." *Data mining (ICDM), 2012 IEEE 12th international conference on*. IEEE, 2012.
13. Lichtenwalter, Ryan N., Jake T. Lussier, and Nitesh V. Chawla. "New perspectives and methods in link prediction." *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2010.
14. Zhang, Yutao, et al. "Cosnet: Connecting heterogeneous social networks with local and global consistency." *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015.
15. Shih, Yu-Keng, and Srinivasan Parthasarathy. "Scalable global alignment for multiple biological networks." *BMC bioinformatics*. Vol. 13. No. 3. BioMed Central, 2012.
16. Zhang, Jiawei, Xiangnan Kong, and S. Yu Philip. "Predicting social links for new users across aligned heterogeneous social networks." *Data Mining (ICDM), 2013 IEEE 13th International Conference on*. IEEE, 2013.
17. Zhang, Jiawei, and S. Yu Philip. "Multiple anonymized social networks alignment." *Data Mining (ICDM), 2015 IEEE International Conference on*. IEEE, 2015.
18. Li, Chung-Yi, and Shou-De Lin. "Matching users and items across domains to improve the recommendation quality." *Proceedings of the20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014.
19. Zhang, Yutao, et al. "Cosnet: Connecting heterogeneous social networks with local and global consistency." *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015.