

华 东 师 范 大 学

2012 年国家大学生创新训练计划项目

研 究 报 告

个人邮件智能挖掘及可视化

Intelligent Mining and Visualization of Personal Email

姓 名： 黄一夫

学 号： 10092510437

班 级： 软件工程 094 班

指导教师姓名： 罗远哉

指导教师职称： 副教授

2013 年 5 月

目 录

摘 要.....	I
ABSTRACT.....	II
一、 选题背景.....	1
(一) 研究目的	1
(二) 国内外研究现状	1
(三) 研究内容	1
二、 实施进程.....	2
(一) 邮件预处理	2
(二) 邮件分类	6
(三) 通讯录管理	9
(四) 邮件网络分析及可视化	12
三、 成果内容.....	15
(一) 邮件预处理	15
(二) 邮件分类	15
(三) 通讯录管理	17
(四) 邮件网络分析及可视化	18
(五) 后期工作	19
四、 创新点.....	21
五、 成果应用情况.....	22
六、 收获与体会.....	22
参考文献.....	23
致谢.....	25

摘 要

每天我们都会收到大量的电子邮件，但如何有效地查看它们却变成越来越头疼的问题。管理邮件通讯录是一个很重要的问题，主要表现在如何管理联系人信息和如何有效地发送邮件。根据邮件收发关系，可以构建出个人邮件社交网络，并进一步挖掘出更多信息。

针对以上问题，本文分别设计并实现了邮件分类，通讯录管理和邮件网络分析及可视化。邮件分类不仅探讨垃圾邮件过滤，而且进一步向用户推荐重要邮件，这样可以将个人邮件进行合理的分类，从而极大地提升邮件用户的收件效率，特别是邮件特别多的用户。通讯录管理从邮件地址和邮件签名中提取信息完善联系人资料，帮助用户自动化地维护通讯录，并提高用户的检索需求。在邮件用户发件时根据已给出的收件人预测推荐额外的收件人，因为邮件经常需要同时发给一些人，邮件收件人推荐可以减轻邮件用户的输入和验证，从而提升邮件用户发件效率，并且对邮件泄漏也有一定的预防作用。邮件网络分析及可视化从网络的结构特点和链接特点入手，计算多种系数值来衡量节点，并提供可视化展示，从而让邮件用户理清社交关系。个人邮件社交网络分析有助于发现重要人物，可视化也为用户提供了查看邮件和通讯录的新手段。

关键词：邮件，分类，信息抽取，网络分析，可视化

Abstract

We receive a mass of emails every day, but it becomes a headache how to check email effectively. It is an important task that managing mail box, for example how to manage user contact and how to send email effectively. According to the “to, cc and bcc”, we can build our own email social network and mine more information further.

For these tasks, I design and implement email classification, contact management and email network analysis and visualization. Email classification not only discusses spam filtering but also recommends important email for user which can improve user's efficiency especially user who have a mass of incoming emails. Contact management extracts information from email address and email signature to help user integrate his contacts, and also improves the efficiency of contact retrieval. Email recipient recommendation helps user add relative receiver, reduce user's mistake and prevent email leakage. Email network analysis and visualization analyzes email network in the aspects of node and link, and also provides visualization to make user more clear about his own email network and find important people. It is also a novel way for user to browser emails and contacts.

Keywords: email, classification, information extraction, network analysis, visualization

一、 选题背景

(一)研究目的

每天我们都会收到大量的电子邮件，但如何有效地查看它们却变成越来越头疼的问题。对此该项目不仅探讨垃圾邮件过滤，而且进一步向用户推荐重要邮件，这样可以将个人邮件进行合理的分类，从而极大地提升邮件用户的收件效率，特别是邮件特别多的用户。

管理邮件通讯录是一个很重要的问题，主要表现在如何管理联系人信息和如何有效地发送邮件。该项目从邮件地址和邮件签名中提取信息完善联系人资料，帮助用户自动化地维护通讯录，并提高用户的检索需求。在邮件用户发件时根据已给出的收件人预测推荐额外的收件人，因为邮件经常需要同时发给一些人，邮件收件人推荐可以减轻邮件用户的输入和验证，从而提升邮件用户发件效率，并且对邮件泄漏也有一定的预防作用。

根据邮件收发关系，可以构建出个人邮件社交网络，并进一步挖掘出更多信息。该项目分别从网络的结构特点和链接特点入手，计算多种系数值来衡量节点，并提供可视化展示，从而让邮件用户理清社交关系。个人邮件社交网络分析有助于发现重要人物，可视化也为用户提供了查看邮件和通讯录的新手段。

(二)国内外研究现状

现有的垃圾邮件过滤主要有基于规则和基于统计两种方式，基于规则方法推广性好但时效性差，基于统计方法时效性好但推广性差。CCERT^[1]致力于中文垃圾邮件的过滤，已经总结出较好的规则集，并且与多种开源邮件客户端，邮件过滤器相结合，取得了不错的结果，但其规则集早已停止更新。Gmail 最近推出的 Priority Inbox^[2]提供的重要邮件分类，其主要基于收发关系，关键字等，目前正处于初步阶段。

Vitor 等人专注于邮件收件人推荐^[3]，签名提取^[4]等工作。他们基于文本和收发关系来推荐邮件收件人，开发出相关插件与开源邮件客户端相结合，并基于正则匹配提取邮件签名，但还未就此整合到通讯录中，而且对其他邮件域信息的提取也尚为欠缺。

社交网络的研究也是目前的热点。Freeman^{[5][6]}用介数等值从网络结构上来描述节点的重要性，PageRank^[7]，HITS^[8]等则从网络链接上分析网络。因为个人邮件涉及隐私，所以这方面的研究比较缺乏，将相应的技术应用到个人邮件网络上，并以可视化的形式展现出来也是相当有意义的。

(三)研究内容

有效地过滤垃圾邮件，并且进一步向用户推荐重要邮件，即邮件重要性分类。如何用机器学习的方法对垃圾邮件，重要邮件进行高准确度的分类。

正确地排序联系人，向用户推荐潜在的收件人，并从邮件中提取更多联系人信息来完善通讯录，即邮件通讯录管理。如何从历史收发关系及文本中智能地提取信息，来推荐邮件收件人，整合通讯录。

深入地分析邮件社交网络，挖掘出重要联系人，并将这些信息可视化出来，即个人邮件社交网络分析及可视化。如何评判邮件社交网络中的重要联系人，并提供合理的可视化。

二、 实施进程

(一)邮件预处理

1. 邮件获取

邮件获取指的是将个人邮箱里的邮件下载到本地。一般来说，较为流行的邮件服务商提供批量导出邮件的功能，但是考虑到此方法与邮件服务商耦合性较大并且无法自动化，本系统基于 JavaMail^[9]编写程序进行邮件获取。

JavaMail，顾名思义，提供给开发者处理电子邮件相关的编程接口。它是 Sun 发布的用来处理 email 的 API。它可以方便地执行一些常用的邮件传输。JavaMail 同时支持 POP3 和 IMAP 协议，这样为编写邮件获取程序提供了很高的便利性。

整个邮件获取程序的逻辑流程如下图所示：

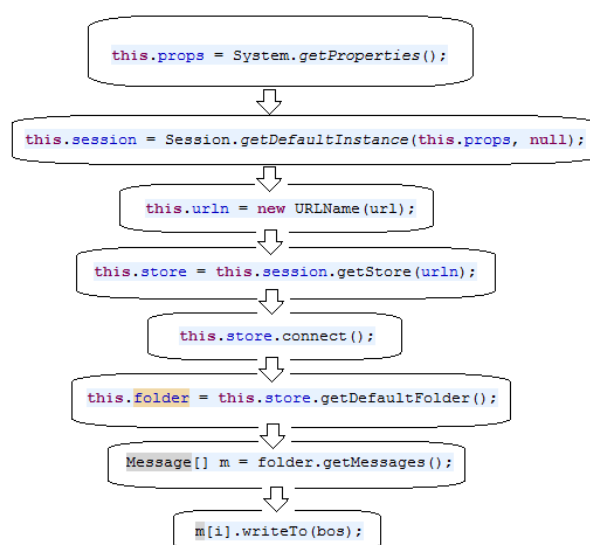


图 2-1 邮件获取程序流程

Figure 2-1 Email Receiver Process

在与邮件客户端建立连接后，需要从根文件夹进行递归地遍历，从而获取到所有邮件的路径，所使用到的关键代码如下：

```
// 递归遍历 IMAP 文件夹来获取邮件
private void dumpFolder(Folder folder, boolean recurse, String tab,
    String path) throws Exception
{
    if ((folder.getType() & Folder.HOLDS_MESSAGES) != 0)
    {
        File f = new File(path + "\\\" + folder.getName());
        f.mkdir();
        folder.open(Folder.READ_ONLY);
        Message[] m = folder.getMessages();
        int count = 0;
        BufferedOutputStream bos = null;
        for (int i = 0; i < m.length; i++)
        {
            File temp = new File(path + "\\\" + folder.getName() + "\\\" + i);
            bos = new BufferedOutputStream(new FileOutputStream(temp));
            m[i].writeTo(bos);
            count++;
        }
    }
    if ((folder.getType() & Folder.HOLDS_FOLDERS) != 0)
    {
        if (recurse)
        {
            Folder[] f = folder.list();
            for (int i = 0; i < f.length; i++)
                dumpFolder(f[i], recurse, tab + "\\t", path);
        }
    }
}
```

2. 邮件解析

根据 RFC^[10]的官方文档定义，通过邮件获取程序下载的邮件为 MIME 格式的文本文件，如图 2-2 所示。该文件存在一定的结构格式和编码，不能直接用于信息的挖掘，因此需要先对邮件进行解析。邮件解析指的是通过正则表达式进行关键域匹配，通过解码器进行解码之后获取到的字段，如发件人，收件人，抄送人，时间，主题，正文等等。

Date: Sun, 28 Apr 2013 09:18:09 +0800
From: "=?GBK?B?y++6o90i?=" <hysun@sei.ecnu.edu.cn>
Subject: "=?GBK?B?y028/s/uxL+53MDttfe/zs2o1qo=?"
To: "100925104" <100925104@sei.ecnu.edu.cn>
Cc: "=?GBK?B?varE/r+1?=" <nkjiang@sei.ecnu.edu.cn>,
"=?GBK?B?0e7K59Xq?=" <szyang@sei.ecnu.edu.cn>
Message-Id: <130428091809ac524a219e89d563c6e8a129b15f2343@sei.ecnu.edu.cn>
MIME-Version: 1.0
X-Mailer: eYou WebMail 8.1.0.1
X-Eyou-Client: 219.228.60.67
Content-Type: multipart/alternative;
boundary="2da5alb44cd236elf6cba843f08c0072"
Content-Transfer-Encoding: 7bit
X-Eyou-Sender: <hysun@sei.ecnu.edu.cn>

--2da5a1b44cd236e1f6cba843f08c0072

Content-Type: text/plain;
charset="GBK"

Content-Transfer-Encoding: base64

uPfu0u82s0aejrLTzvNK6w606DQogICAgICAgICDXUwjPIIa0oz8LW3M7l06nG8LXE
obbI7bz+z+7Ev7ncw02ht7/0s8y199X71sHDv9bczuXPws7nMbXjv6rKvK0stdi149Ta
w02/xsKlQjIx0K0sx+vNrNGrw8e7pc/g16q45qGjyOfTOM7KzOKjrMfrvLDKsbrNztK5
tc2oDQrL77qj06INCg==

```
--2da5a1b44cd236e1f6cba843f08c0072
```

Content-Type: text/html;
charset="GBK"

Content-Transfer-Encoding: base64

uPf0u82s0aejrLTzvNK6w606PGJyIC8+DQombmJzcdsmbmJzcdsmbmJzcdsmbmJzcdsm
bmJzcdsmbmJzcdsmbmJzcdsmbmJzcdsg19Q11MIzyNWjqM/C1tzo5a0pxvC1xKG2yO28
/s/uXl+53MDtoBe/zrPMtfFv+9bBw7/W3M71PHN0cm9uZz7Pws7nMbXjv6rKvDwwc3Ry
b25rPq0sPHN0cm9uZz612LXj1NrA7b/GwqVCMje4PC9zdHJvbmc+o6zH682s0afDx7ul
z+DXqrjmoePI59PQzsrM4Qosx+u8sMqxus300rm1zag8YnlGLz4NCsvvuqPTojxicAv
PgOK

--2da5a1b44cd236e1f6cba843f08c0072--

图 2-2 MIME 文件示例

Figure 2-2 MIME File Example

在实现的过程中采用适配器设计模式，如图 2-3 所示。以 Javamail 中类 `MimeMessage` 为内核进行封装实现 `JavamailImpl`，用于配合多种 Client 调用，方便后续不同挖掘功能对邮件数据的需求。

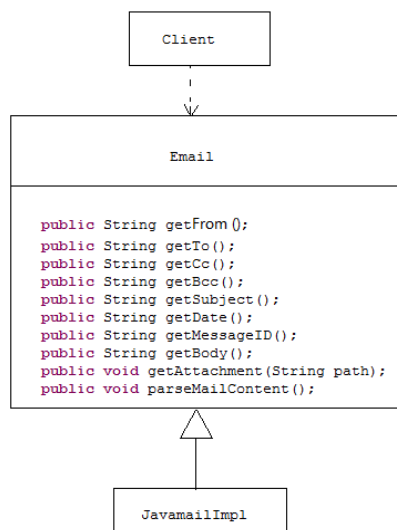


图 2-3 适配器设计模式

Figure2-3 Adapter Design Pattern

在对邮件主体进行解析时，需要对 `ContentType` 字段进行判断，然后进行相应的操作，其类似于邮件获取程序，也是递归解析的逻辑，关键代码如下：

```
//解析邮件主体
private void getMailContent(Part part)
{
    // 得到part的标签，然后进行分类
    String contenttype = part.getContentType();
    // 判断是否需要解码
    int nameindex = contenttype.indexOf("charset");
    boolean conname = true; // 标志part标签里是否含有name
    if (nameindex != -1)
    {
        conname = false;
    }
    // text/plain段并且需要解码
    if (part.isMimeType("text/plain") && !conname)
    {
        bodytext.append((String) part.getContent());
    }
    // text/html段并且需要解码
    else if (part.isMimeType("text/html") && !conname)
    {
        bodytext.append((String) part.getContent()); // 添加到body字段
    }
    else if (part.isMimeType("multipart/*")) // 复合部分，分别递归
    {
        Multipart multipart = (Multipart) part.getContent();
```

```
int counts = multipart.getCount();
for (int i = 0; i < counts; i++)
{
    getMailContent(multipart.getBodyPart(i));
}
}
else if (part.isMimeType("message/rfc822")) // 邮件格式，递归
{
    getMailContent((Part) part.getContent());
}
else
{
    // com.sun.mail.util.BASE64DecoderStream
}
}
```

(二)邮件分类

1. 垃圾邮件过滤

本系统基于 SVM^[11]进行垃圾邮件过滤。支持向量机 SVM(Support Vector Machine)作为一种可训练的机器学习方法，依靠小样本学习后的模型参数进行导航星提取，可以得到分布均匀且恒星数量大为减少的导航星表。

首先是人工类标标注，通过人工手动检查垃圾箱和收件箱，将误判的邮件纠正过来；然后是观察垃圾邮件，结合查询的资料（CCERT 中文规则集，SpamAssassin 等），选取出数十种特征，大多为统计特定词的词频，如表 2-1 所示。

表 2-1 垃圾邮件特征

Table2-1 Spam Feature

发件人地址长度	代开 词频	感谢 词频	商机 词频
发件人地址是否含norely	贵公司 词频	订单 词频	节能 词频
发家人地址是否含edu	抵扣 词频	商品 词频	索取 词频
主题长度	节省 词频	服务 词频	销售 词频
正文中URL数量	避税 词频	评价 词频	联系人 词频
正文中图片数量	家电 词频	反馈 词频	转帐 词频
正文中AD广告出现次数	潜在 词频	订购 词频	打扰 词频
点击这里 click here 词频	票据 词频	有奖 词频	验证 词频
unsubscribe 取消订阅 词频	特价 词频	客服 词频	钱 词频
推荐 词频	采购 词频	退订 词频	科技 词频
体验 词频	付款 词频	免费 词频	; 词频
调查 词频	机票 词频	优惠 词频	(词频
购物 词频	赚钱 词频	合作 词频	[词频
亲爱 词频	特惠 词频	发票 词频	! 词频
客户 词频	商务 词频	实业 词频	\$ 词频
您好 词频	高效 词频	缴款 词频	# 词频

对每封邮件进行特征提取，整理成 LibSVM^[12]接受的格式；一部分训练，一部分测试，所得分类器准确率为 85.3846%。整个程序流程如下所示：

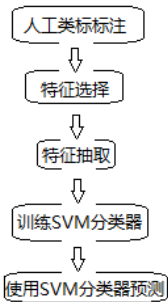


图 2-4 垃圾邮件过滤

Fufure2-4 Spam Filtering

2. 重要邮件推荐

重要邮件推荐的实现原理基于 Naïve Bayes^[13]，朴素贝叶斯分类器是一种应用基于独立假设的贝叶斯定理的简单概率分类器.更精确的描述这种潜在的概率模型为独立特征模型。

用发件箱里的邮件标注出重要邮件，由此训练出分类器，并用该分类器推荐重要邮件。首先用发件箱里的邮件自动标注出收件箱里的重要邮件，这里重要邮件的定义为回复过的邮件；然后对邮件正文进行分词，建立字典，分词实现基于 IKAnalyzer^[14]，IKAnalyzer 是一个开源的，基于 java 语言开发的轻量级的中文分词工具包；再按朴素贝叶斯概率公示训练 NB 分类器模型，建立的模型片段如图 2-5 所示；最后对待分类的邮件进行提取词项，分别求出两类得分，并归为得分高的一类。

院长	13589	5.18E-05	0.001652994
除夕	13590	5.18E-05	3.06E-04
除夕夜	13591	5.18E-05	3.06E-04
除夕过	13592	5.18E-05	1.22E-04
陪	13593	5.18E-05	3.06E-04
陪伴	13594	5.18E-05	1.84E-04
陵园	13595	5.18E-05	1.84E-04
陶	13596	5.18E-05	0.001163218
陶瓷	13597	5.18E-05	1.84E-04
隆重	13598	5.18E-05	3.06E-04
隆重	13599	5.18E-05	1.84E-04
隆重推出	13600	5.18E-05	1.84E-04
隋	13601	5.18E-05	3.06E-04
随上	13602	5.18E-05	1.22E-04
随便	13603	5.18E-05	3.06E-04
随后	13604	5.18E-05	4.90E-04
随地	13605	5.18E-05	6.12E-04
随堂	13606	5.18E-05	1.84E-04
随处	13607	5.18E-05	3.06E-04
随处可见	13608	5.18E-05	3.06E-04
随意	13609	5.18E-05	7.96E-04

图 2-5 朴素贝叶斯模型片段

Figure2-5 Naïve Bayes Model Snap Shot

整个重要邮件推荐的功能流程如下：

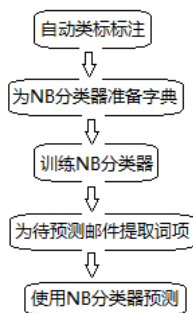


图 2-6 重要邮件推荐

Figure2-6 Important Email Recommendation

其中朴素贝叶斯分类器的训练代码如下：

```
// NB模型的训练
public void trainMultinomialNB(String Dir, String classmark, String
dic, String termnum) throws Exception
{
    // 初始化词汇表
    V = extractVocabulary(dic);
    // 初始化条件概率
    condprob = new double[V.size()][C.size()];
    // 初始化文档数
    N = countDocs(classmark);
    for (int i = 0; i < C.size(); i++)
    {
        long Nc = countDocsInClass(i, classmark);
        prior[i] = (double) Nc / (double) N;
        String TESTc = concatenateTextOfAllDocsInClass(i, classmark, Dir);
        long T[] = new long[V.size()];
        for (int j = 0; j < V.size(); j++)
        {
            if (j % 1000 == 0)
                System.out.println("countTokenOfTerm " + j + "/" + V.size());
            T[j] = countTokenOfTerm(TESTc, V.get(j));
        }
        for (int j = 0; j < V.size(); j++)
        {
            if (j % 1000 == 0)
                System.out.println("compute " + j + "/" + V.size());
            condprob[j][i] = compute(T[j], i, classmark, termnum);
        }
    }
}
```

(三)通讯录管理

1. 联系人信息抽取

联系人信息抽取主要分为两个部分：邮件地址信息抽取和邮件签名信息抽取。邮件地址信息抽取指的是，通过定义一系列的正则表达式，从邮件地址中提取出生日，手机号，部门，国家等信息，如图 2-7 所示。

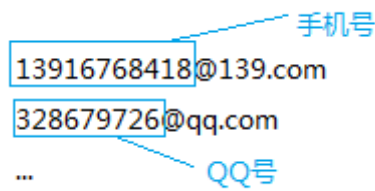


图 2-7 邮件地址信息抽取示例

Figure2-7 Email Address Information Extraction Example

邮件签名信息抽取指的是，基于隐马尔可夫模型^[15]，隐马尔可夫模型（Hidden Markov Model，HMM）是统计模型，它用来描述一个含有隐含未知参数的马尔可夫过程。其难点是从可观察的参数中确定该过程的隐含参数。然后利用这些参数来作进一步的分析，例如模式识别。使用状态表示联系人记录，使用特征值作为观察值如表 2-2 所示，通过训练产生邮件签名标注模型，然后使用该模型对未标注的邮件签名进行标注，如图 2-8 所示。

表 2-2 观察值成员域

Table2-2 Observation Field

语气词	单位		地址	电话	电子邮件	职位	图片	网址	传真
祝	学	社区	洲	phone	email	. *er\$. *\\.jpg\$	http:	fax
好	部	网	国	^[1]([3][0-9]{1} 59 58 88 89)[0-9]{8}\$	^([a-z0-9A-Z]+[- \\. ?)+[a-z0-9A-Z]{2,4}\$. *or\$. *\\.png\$	www.	[0-9]{7,8}
best	院	corporation	省	tel	邮箱	顾问	. *\\.gif\$	网址	传真
kind	司	联	州	[0-9]{7,8}	mail				
regard	technology	service	市	联系方式					
谢	institute	team	区	电话					
friendly	university	组	县	手机					
thank	务	department	镇	mobile					
warm	级	center	乡	direct line					
luck	班	基地	村	skype					
cheers	部	集团	路						
sincerely	系	办公室	号						
	中心		幢						
			单元						
			楼						
			cubic						
			road						
			地址						



图 2-8 邮件签名信息抽取示例

Figure2-8 Email Signature Information Extraction Example

邮件签名信息抽取的训练算法如下所示：

```
// 签名分类器的训练
public void train(String trainPath, String labelPath)
{
    BaumWelchLearner bwl = new BaumWelchScaledLearner();
    MultistreamsHMM.trainSequences = generateSequences(trainPath);
    for (int i = 0; i < MultistreamsHMM.iterate; i++)
    {
        this.hmm = bwl.iterate(this.hmm,
MultistreamsHMM.trainSequences);
    }
    ForwardBackwardCalculator fbc = new ForwardBackwardScaledCalculator(
        MultistreamsHMM.trainSequences.get(0), this.hmm,

EnumSet.allOf(ForwardBackwardCalculator.Computation.class));
    double xi[][][] =
bwl.estimateXi(MultistreamsHMM.trainSequences.get(0),
        fbc, this.hmm);
    double gamma[][] = bwl.estimateGamma(xi, fbc);
    MultistreamsHMM.trainLabel = generateLabelSeq(labelPath);
    this.label = generateLabel(this.hmm.nbStates(),
        MultistreamsHMM.trainLabel, gamma);
    this.visit = generateVisit(this.hmm.nbStates(),
        MultistreamsHMM.trainSequences.get(0), gamma);
    this.risk = generateRisk(this.hmm.nbStates(),
        MultistreamsHMM.trainLabel,
        MultistreamsHMM.trainSequences.get(0), gamma, this.label,
        this.visit);
    this.heavyState = generateHeavyState(this.risk, this.visit,
        MultistreamsHMM.rejectBound, MultistreamsHMM.visitBound);
}
```

```

    this.riskSet = generaterejectsubset(this.risk, this.visit,
        MultistreamsHMM.rejectBound, MultistreamsHMM.visitBound);
// reference this
MultistreamsHMM nowmultihmm = this;
while (nowmultihmm.heavyState != -1)
{
    nowmultihmm.nextmultihmm = RRtrain(nowmultihmm,
        MultistreamsHMM.trainSequences,
        MultistreamsHMM.refineStateNum);
    nowmultihmm = nowmultihmm.nextmultihmm;
}
}

```

2. 收件人推荐

收件人推荐指的是，在邮件用户发件时根据已给出的收件人预测推荐额外的收件人，因为邮件经常需要同时发给一些人，邮件收件人推荐可以减轻邮件用户的输入和验证，从而提升邮件用户发件效率，并且对邮件泄漏也有一定的预防作用。

使用 Machine-Learned Ranking^[16]，考虑第一个收件人是 query，整个 addressbook 里的联系人是返回的结果，考虑正在出现在第一个收件人之后的联系人为相关，其他没有出现的联系人为不相关，这样类标问题就解决了。主要要考虑的是特征选取的问题，先暂时不考虑邮件内容的问题，从收件箱和发件箱里的伴随关系入手，如下表 2-3 所示。

表 2-3 收件人推荐特征

Table2-3 Recipient Recommendation Feature

1. 发件箱里，该邮件之前，出现的总次数
2. 发件箱里，该邮件之前，一天内出现的总次数
3. 发件箱里，该邮件之前，七天内出现的总次数
4. 发件箱里，该邮件之前，一个月出现的总次数
5. 发件箱里，该邮件之前，最近一次到此间间隔的天数，若无则用无穷大表示
6. 发件箱里，该邮件之前，伴随的总次数
7. 发件箱里，该邮件之前，伴随的一天内的次数
8. 发件箱里，该邮件之前，伴随的七天内的次数
9. 发件箱里，该邮件之前，伴随的一个月的次数
10. 发件箱里，该邮件之前，最近一次伴随到此间间隔的天数，若无伴随则用无穷大表示
11. 收件箱里，该邮件之前，出现的总次数
12. 收件箱里，该邮件之前，一天内出现的总次数
13. 收件箱里，该邮件之前，七天内出现的总次数
14. 收件箱里，该邮件之前，一个月出现的总次数
15. 收件箱里，该邮件之前，最近一次到此间间隔的天数，若无则用无穷大表示
16. 收件箱里，该邮件之前，伴随的总次数
17. 收件箱里，该邮件之前，伴随的一天内的次数
18. 收件箱里，该邮件之前，伴随的七天内的次数
19. 收件箱里，该邮件之前，伴随的一个月的次数
20. 收件箱里，该邮件之前，最近一次伴随到此间间隔的天数，若无伴随则用无穷大表示
21. 邮件域名是否相同

在设计特征的时候，主要考虑时间，频次，域名三个维度，通过解析邮件，对历史的收件箱和发件箱中邮件的收发关系进行抽取，然后基于 SVMRank API^[17]进行相应的训练和预测。

(四)邮件网络分析及可视化

1. 邮件网络分析

根据邮件收发关系，构建出个人社交网络，在结构方面，计算聚集系数，介数等值，在链接方面，计算 PageRank，HITS 等。介数反映了相应的节点或者边在整个网络中的作用和影响力，是一个重要的全局几何量，具有很强的现实意义。PageRank 根据网络节点的外部链接和内部链接的数量和质量来衡量网络节点的价值。HITS 算法通过两个评价权值——内容权威度（Authority）和链接权威度（Hub）来对网页质量进行评估。聚集系数是表示一个图形中节点聚集程度的系数，证据显示，在现实中的网络中，尤其是在特定的网络中，由于相对高密度连接点的关系，节点总是趋向于建立一组严密的组织关系。

本系统基于 JGraphT API^[18]进行实现，以下一些是计算系数的关键代码：

```
// 获取节点 vertex 的聚集系数
// 对于节点 v，如果有 k 个节点与 v 相连，则这 k 个节点称为 v 的邻居
// 这 k 个邻居之间可能存在 k(k-1) 条边
// 这 k 个邻居之间实际存在的边与总的可能边数之比定义为 v 的聚集系数
public double clusteringCoefOf(V vertex)
{
    Set<V> neigSet = neighbors(vertex);
    int k = neigSet.size();
    if (k == 1 || k == 0) // k * (k - 1) = 0
        return 0;
    return (double) numEdgesOf(neigSet) / (double) (k * (k - 1));
}
// 获取节点 vertex 的介数
/*
 * 对于节点 v，网络中包含 v 的所有最短路径的条数为 shortestv 网络中所有的最短路径的
条数为 shortest
 * shortestv/shortest 定义为 v 的介数 这里默认 shortest 为 n*(n-1)/2
 */
public double betweennessCentralityOf(
    ArrayList<DijkstraShortestPath<String, XEdge>> paths, V vertex)
{

```



```
int shortestv = 0;
Set<V> vset = this.vertexSet();
int n = vset.size();
double shortest = n * (n - 1) / 2;
for (int i = 0; i < paths.size(); i++)
{
    if (pathPassVertex(paths.get(i), vertex))
    {
        shortestv++;
    }
}
return (double) shortestv / (double) shortest;
}
// 获取节点 vertex 的凝聚度
/*
 * 对于节点 v, 从该节点到网络中其他所有节点所需路径之和为该点的凝聚度
 */
public double cohesionOf(V vertex)
{
    double cohesion = 0;
    Set<V> vset = this.vertexSet();
    for (V v : vset)
    {
        if (v.equals(vertex)) // 排除自身节点
            continue;
        @SuppressWarnings("unchecked")
        DijkstraShortestPath<String, XEdge> dPath = new
DijkstraShortestPath<String, XEdge>(
            (XDefaultGraph<String, XEdge>) this, vertex.toString(),
            v.toString());
        if (Double.isInfinite(dPath.getPathLength()))
            continue;
        cohesion += dPath.getPathLength();
    }
    return cohesion;
}
```

2. 邮件网络可视化

完成对个人邮件网络结构和链接上的分析之后, 需要将结果可视化出来, 这样不

但方便展示结果，还能从可视化中挖掘出更多的信息，从而达到可视分析的效果。个人邮件社交网络分析有助于发现重要人物，可视化也为用户提供了查看邮件和通讯录的新手段。

在设计邮件网络可视化程序的时候，使用 Prefuse API^[19]进行程序的编写，以下是可视化程序设计的逻辑流程：

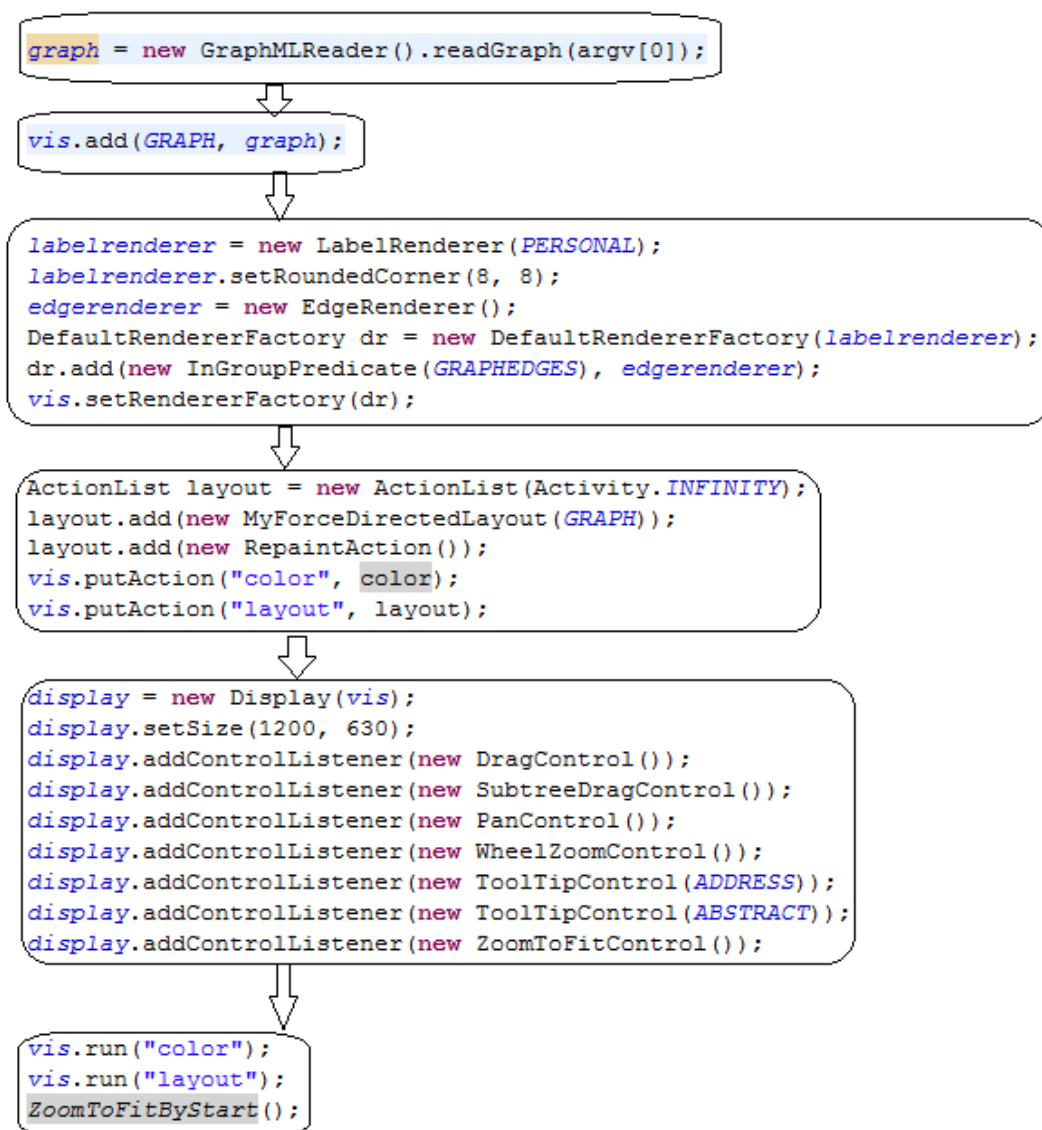


图 2-9 可视化流程

Figure2-9 Visualization Process

以下是可视化程序中计算边长度的代码：

关键代码段示例：

```

// 根据收发频次生成边的长度，频次越高，长度越短
protected float getSpringLength(EdgeItem e)

```

```

{
    float f;
    f = e.canGetFloat("weight") ? 300 / e.getFloat("weight") : 20;
    return f;
}

```

三、 成果内容

(一) 邮件预处理

1. 邮件获取

ReceiveEmailIMAP.jar 通过 IMAP 协议获取邮件

```

D:\iEmail\bin\receive>java -jar ReceiveEmailIMAP.jar
Usage: ReceiveEmailIMAP URL Storepath
URL format: imap://username:password@imaphost

```

URL: imap://邮箱帐号:邮箱密码@IMAP 主机

Storepath: 本地存放路径

2. 邮件解析

ParseEmail.jar 解析邮件，将各个域保存为文本，并提取出附件

```

D:\iEmail\bin\parse>java -jar ParseEmail.jar
Usage: ParseEmail MIMEFileDir StorePath

```

MIMEFileDir: 邮件的本地存放路径

StorePath: 邮件解析后的存放路径

(二) 邮件分类

1. 垃圾邮件过滤

ExtractEmailFeature.jar 提取垃圾邮件和非垃圾邮件特征

```

D:\iEmail\bin\filter\SUM>java -jar ExtractEmailFeature.jar
Usage: ExtractEmailFeature Dir feature.txt feature.libsvm classlabel

```

Dir: 垃圾邮件目录或非垃圾邮件目录

feature.txt: 特征文件，位于 data\filter\svm\feature.txt

feature.libsvm: 生成的 libsvm 可接受的格式

classlabel: 垃圾邮件为 1，非垃圾邮件为 0

Libsvm 分类器

```

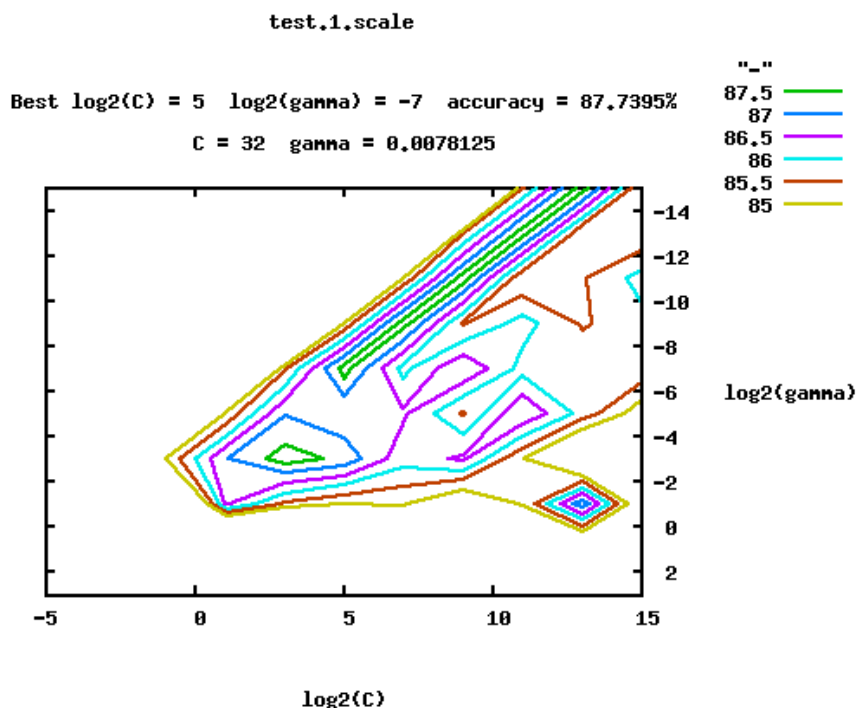
D:\iEmail\bin\filter\SUM\windows>python easy.py test.1 test.2
Scaling training data...
Cross validation...
Best c=32.0, g=0.0078125 CU rate=87.7395
Training...
Output model: test.1.model
Scaling testing data...
Testing...
Accuracy = 85.3846% (222/260) (classification)
Output prediction: test.2.predict

```

easy.py: 自动化训练和预测脚本

test.1: 训练数据

test.2: 测试数据



2. 重要邮件推荐

ClassMark.jar 重要邮件标记程序

```
D:\iEmail\bin\filter\NB>java -jar ClassMark.jar
Usage: ClassMark FromToCcDateReceiveName.txt FromToCcDateSend.txt ClassMark.txt
```

FromToCcDateReceiveName.txt: 经 ParseFromToCcDateName.jar 解析得到的收件箱文件

FromToCcDateSend.txt: 经 ParseFromToCcDate.jar 解析得到的发件箱文件

ClassMark.txt: 生成的重要邮件标记文件

BuildDictionary.jar 为 NB 分类准备字典

```
D:\iEmail\bin\filter\NB>java -jar BuildDictionary.jar
Usage: BuildDictionary Dir StopWordDir Dic.txt TermNum.txt
```

Dir: 经 ParseSubjectBody.jar 解析存储的目录

StopWordDir: 位于 data\filter\nb\stopword, 有中文和英文的停用词

Dic.txt: 生成字典存放路径

TermNum.txt: 生成各文件词项数存放路径

NBtrain.jar NB 训练器

```
D:\iEmail\bin\filter\NB>java -jar NBtrain.jar
Usage: NBtrain Dir ClassMark.txt dic.txt termnum.txt model.txt
```

Dir: 经 ParseSubjectBody.jar 解析存储的目录

dic.txt: BuildDictionary.jar 生成字典存放路径

termnum.txt: BuildDictionary.jar 生成各文件词项数存放路径

model.txt: 训练模型存放路径

BuildDocument.jar 为待预测邮件提取词项

```
D:\iEmail\bin\filter\NB>java -jar BuildDocument.jar
Usage: BuildDocument InputDir StopWordDir OutputDir
```

InputDir: 待预测邮件目录

StopWordDir: 位于 data\filter\nb\stopword, 有中文和英文的停用词

OutputDir: 待预测邮件提取词汇存放目录

NBpredict.jar NB 预测器

```
D:\iEmail\bin\filter\NB>java -jar NBpredict.jar
Usage: NBpredict dic.txt model.txt IncomingDir PredictDir
```

dic.txt: BuildDictionary.jar 生成字典存放路径

IncomingDir: 经 BuildDocument.jar 得到的待预测邮件提取词汇存放目录

PredictDir: 预测打分目录

预测中 0 类分值高者为重要邮件

(三)通讯录管理

1. 联系人信息抽取

GetInfoByAddress.jar 从邮件地址中抽取信息

```
D:\iEmail\bin\info>java -jar GetInfoByAddress.jar
GetInfoByAddress FromToCc.txt GetInfoByAddress.txt
```

FromToCc.txt: 经 ParseFromToCc.jar 解析得到的文件

GetInfoByAddress.txt: 按时间排序的联系人列表

ExtractSignature.jar

```
D:\iEmail\bin\info>java -jar ExtractSignature.jar
Usage: ExtractSignature Dir SignatureDir
```

Dir:挑选出来有邮件签名的目录

SignatureDir:提取出邮件签名存放的目录

SignatureClassifier.jar

```
D:\iEmail\bin\info>java -jar SignatureClassifier.jar
Usage: SignatureClassifier train.txt model.gz InputDir OutputDir
```

train.txt:人工标注好的训练数据

model.gz:模型存放文件

InputDir: ExtractSignature.jar 提取出邮件签名存放的目录

OutputDir:机器分类后结果存放的目录

2. 收件人推荐

```
0 qid:57 1:0 2:0
1 qid:57 1:2 2:217
0 qid:57 1:0 2:0
1 qid:57 1:3 2:344
0 qid:57 1:0 2:0
0 qid:57 1:0 2:0
0 qid:57 1:0 2:0
```

Figure 3. Data format of my approach

(四) 邮件网络分析及可视化

1. 邮件网络分析

ParseEmailNetwork.jar

```
D:\iEmail\bin\view>java -jar ParseEmailNetwork.jar
Usage: ParseEmailNetwork Dir U.txt E.txt
```

Dir:待可视化的邮件目录路径

V.txt:邮件网络中的节点

E.txt:邮件网络中的边

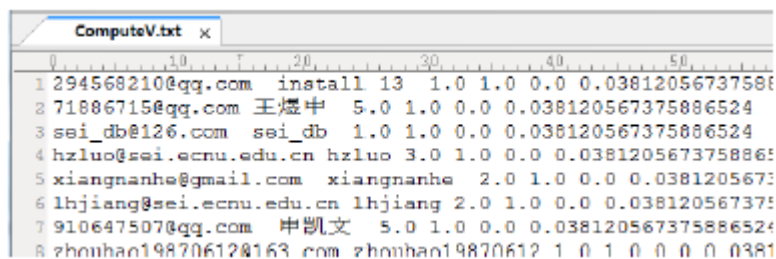
ComputeEmailNetwork.jar

```
D:\iEmail\bin\view>java -jar ComputeEmailNetwork.jar
Usage: ComputeEmailNetwork U.txt E.txt ComputeV.txt
```

V.txt: 邮件网络中的节点

E.txt: 邮件网络中的边

ComputeV.txt: 经计算后的邮件网络中的节点



ID	Email Address	Name	Value 1	Value 2	Value 3	Value 4	Value 5	Value 6	Value 7
1	294568210@qq.com	install	13	1.0	1.0	0.0	0.03812056737588		
2	71886715@qq.com	王煜中	5.0	1.0	0.0	0.038120567375886524			
3	sei_db@126.com	sei_db	1.0	1.0	0.0	0.038120567375886524			
4	hzluo@sei.ecnu.edu.cn	hzluo	3.0	1.0	0.0	0.0381205673758865			
5	xiangnanhe@gmail.com	xiangnanhe	2.0	1.0	0.0	0.0381205673			
6	lhjiang@sei.ecnu.edu.cn	lhjiang	2.0	1.0	0.0	0.038120567375			
7	910647507@qq.com	申凯文	5.0	1.0	0.0	0.038120567375886524			
8	zhonhao19870612@163.com	zhonhao19870612	1	0	1	0	0	0	0381

2. 邮件网络可视化

BuildEmailNetworkXML.jar

```
D:\iEmail\bin\view>java -jar BuildEmailNetworkXML.jar
Usage: BuildEmailNetworkXML ComputeV.txt E.txt EmailNetwork.xml
```

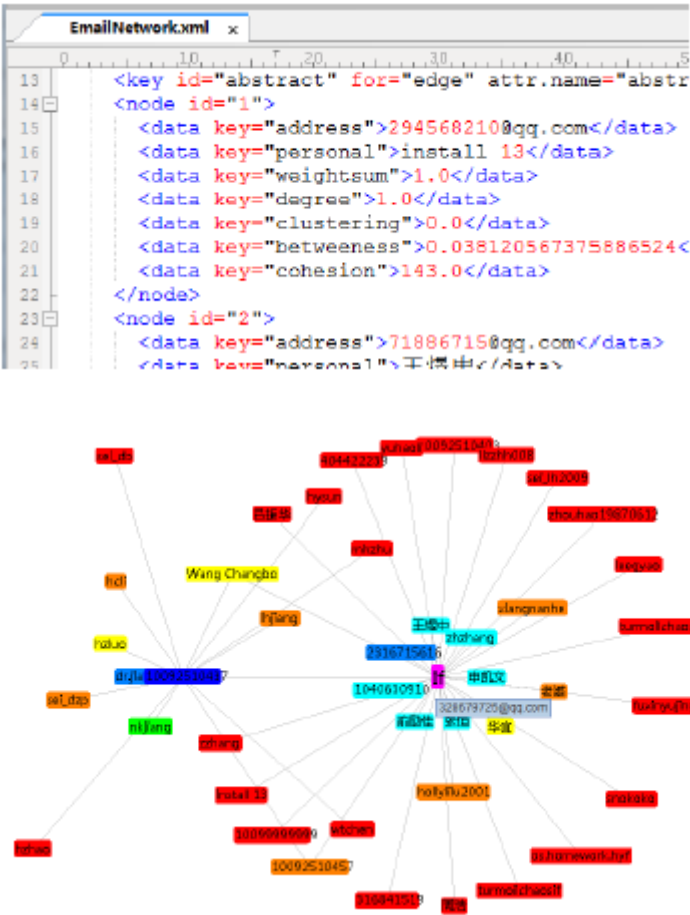
ComputeV.txt: 经计算后的邮件网络中的节点

E.txt: 邮件网络中的边

EmailNetworkView.jar

```
D:\iEmail\bin\view>java -jar EmailNetworkView.jar
Usage: EmailNetworkView EmailNetwork.xml
```

EmailNetwork.xml: 邮件网络可视化所需文件



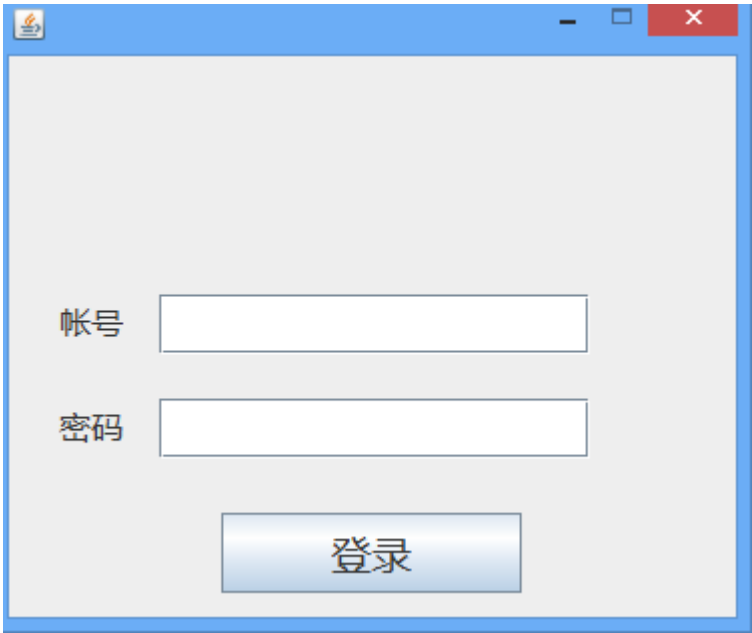
(五)后期工作

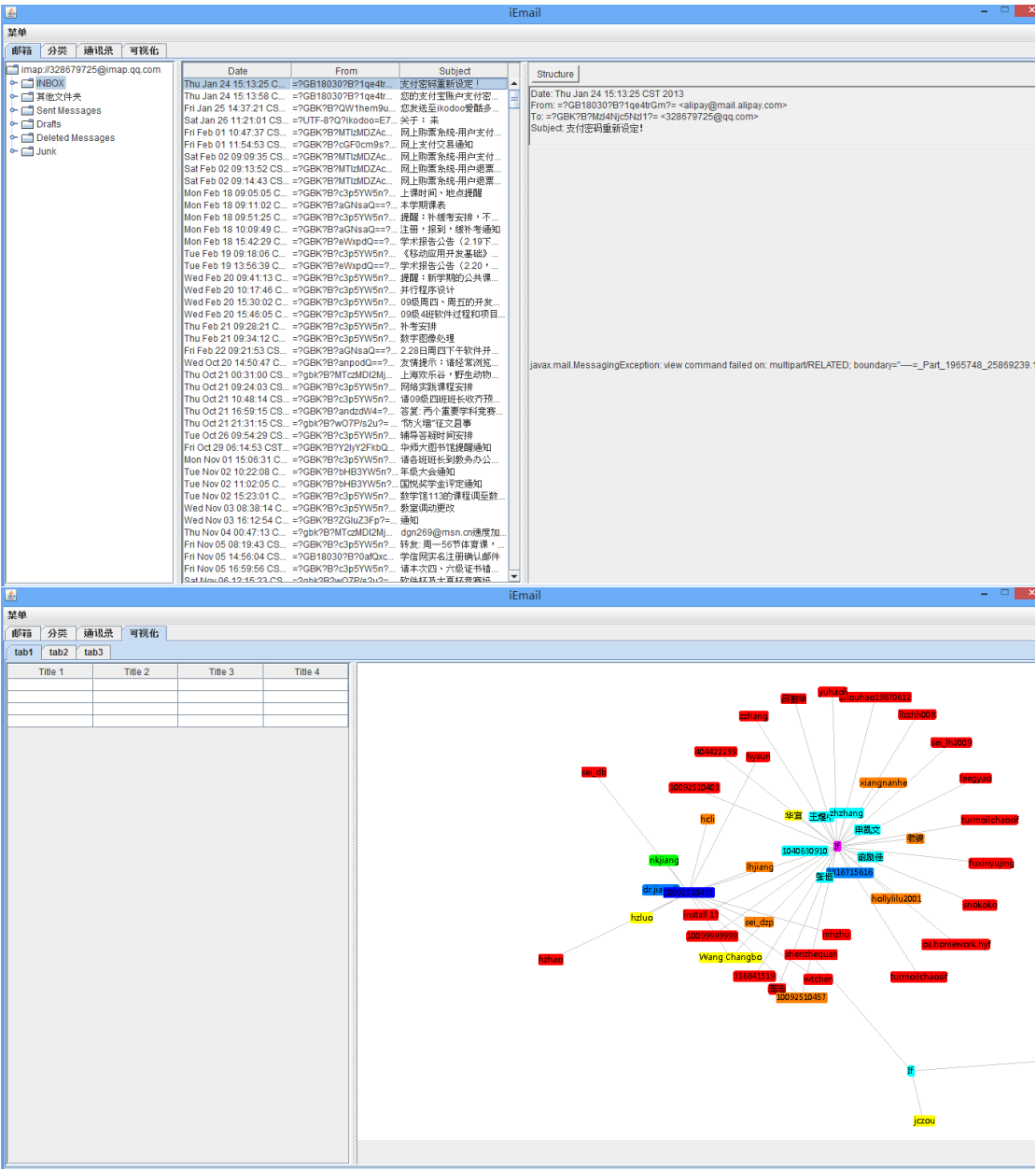
1. 代码重构

类名	描述	层次
AddressInfoExtractor	邮件地址信息抽取类	算法模型
Contact	通讯录类	数据模型
SignatureInfoExtractor	邮件签名信息抽取类	算法模型
Client	客户端调用类	控制器
Address	邮件地址类	数据模型
ComputedAddress	计算后邮件地址类	数据模型
DateSubject	日期主题类	数据模型
Email	邮件接口	数据接口
EmailImpl	Javamail 解析实现类	数据实现
EmailNetworkComputer	邮件网络计算类	算法模型
EmailNetworkParser	邮件网络解析类	算法模型
EmailNetworkXMLBuilder	邮件网络 XML 生成类	算法模型
FolderModel	邮件文件夹模型类	数据模型
FolderTreeNode	邮件文件夹节点类	数据模型
ImportantClassifier	重要邮件分类器	算法模型
ImportantFeatureExtractor	重要邮件特征抽取类	算法模型
NBpredict	朴素贝叶斯预测类	算法模型

NBtrain	朴素贝叶斯训练类	算法模型
Parser	邮件解析类	控制器
Receiver	邮件获取类	控制器
ReceipientRecommender	收件人推荐类	算法模型
Record	收发关系记录类	数据模型
Server	邮件服务器接口	控制器接口
ServerImpl	邮件服务器实现	控制器实现
SpamClassifier	垃圾邮件分类器类	算法模型
SpamFeatureExtractor	垃圾邮件特征抽取类	算法模型
StoreTreeNode	邮件存储节点类	数据模型
XDefaultGraph	邮件网络实现类	算法实现
XEdge	邮件网络边类	数据模型
XGraph	邮件网络接口	算法接口
svm_predict	SVM 预测类	算法模型
svm_scale	SVM 放缩类	算法模型
svm_train	SVM 训练类	算法模型
ComponentFrame	组件框架类	界面
EmailNetworkViewer	邮件网络显示类	界面
FolderViewer	邮件文件夹显示类	界面
LoginViewer	系统登录类	界面
MainViewer	系统主界面类	界面
MessageViewer	邮件体显示类	界面
MultipartViewer	邮件体结构显示类	界面
TextViewer	邮件文本显示类	界面

2. 界面设计





四、 创新点

采用机器学习的方法，基于规则过滤垃圾邮件，并基于内容推荐重要邮件。这样智能的邮件分类，旨在提高邮件用户的收发效率，特别是邮件数量巨大的用户或者公司员工。

采用机器学习的方法，对邮件地址和邮件签名中的各条信息进行分类，以便格式化地完善邮件通讯录。这样智能的信息挖掘，旨在自动化地丰富邮件用户的联系人信息，提升用户检索通讯录信息的能力。

基于个人邮件社交网络的结构特点和链接特点来分析节点，并以可视化的形式展现出来，通过人机交互的手段帮助用户理清自己的社交关系。这样社交网络分析，旨在帮助用户发现重要联系人，以及联系人之间的关系。

将机器学习理论应用到实际问题中,更加智能地挖掘数据,并将结果可视化展示,有良好的人际交互接口。主要体现在垃圾邮件过滤,重要邮件推荐,邮件签名提取,邮件社交网络分析及可视化。

五、 成果应用情况

目前,市面上的邮件客户端及邮件通讯录等管理软件数量较多,但大多还不够智能。本系统成功得将机器学习和可视化的方法运用到其中,用人工智能的手段来帮助邮件用户,还能得到更佳的人机交互体验。由于我们开发系统的目的再于提升邮件用户的效率,必然会有很宽广的市场。但现在仍然处于实验室阶段,还有许多工作需要去完善,同时本系统也是产学研一体化的见证。

另外,文章中使用的各种研究方法可以很容易地应用到其他领域中。我将邮件分类部分的方法应用到图像分割中,发表了 EI 检索的论文: Yifu Huang, Haoyan Chen, Haibin Cai, Chao Peng, Linhua Jiang, " Automated Land Resource Classification of Electronic Photograph Based on Satellite CMOS Detector ", 2012 International Conference on Electrical Engineering and Computer Science(EECS 2012), LNEE 178, pp. 521-527.。我将邮件网络分析部分的方法应用到 2012 年北美交叉学科建模中,获得了三等奖。我将邮件签名抽取部分的方法应用到了我的本科毕业设计中: 基于情感分析的金融走势选择性预测。

六、 收获与体会

首先建立对项目的整体把握,然后对关键问题进行各个突破。通过阅读书籍,论文建立研究的理论基础;通过动手实验,与老师讨论建立研究的理论实践。将理论与实践相结合,达到产学研一体化。另外研究过程中使用的研究方法的适用性较广,很容易运用到其他领域中去。

参考文献

- [1]. 陈光英. CCERT 中文垃圾邮件过滤规则集 [E].
http://www.ccert.edu.cn/spam/sa/Chinese_rules.htm
- [2]. Douglas Aberdeen, Ondrej Pacovsky and Andrew Slater. The Learning behind Gmail Priority Inbox. Proc. NIPS Workshop on Learning on Cores, Clusters and Clouds (LCCC), Whistler, Canada, Dec. 2010.
- [3]. Ramnath Balasubramanyan, Vitor R. Carvalho and William W. Cohen. Cut Once: Recipient Recommendation and Leak Detection in Action [C]. AAAI-2008 EMAIL Workshop, Chicago, Jul 2008.
- [4]. Vitor R. Carvalho and William W. Cohen. Learning to Extract Signature and Reply Lines from Email. CEAS-2004 (Conference on Email and Anti-Spam), Mountain View, CA, July 2004.
- [5]. L. C. Freeman. A Set of Measures of Centrality Based Upon Betweenness. Sociometry, 40, 1977, 35-41.
- [6]. L.C. Freeman, S.P. Borgatti and D.R. White. Centrality in Valued Graphs: A Measure of Betweenness Based on Network Flow. Social Networks, 13, 1991, 141-154.
- [7]. Lawrence Page and Sergey Brin and Rajeev Motwani and Terry Winograd. The PageRank Citation Ranking: Bringing Order to the Web [R]. SIDL-WP-1999-0120.
- [8]. J. Kleinberg. Authoritative sources in a hyperlinked environment [C]. Proc. 9th ACM-SIAM Symposium on Discrete Algorithms, 1998.
- [9] JavaMail [E]. <http://www.oracle.com/technetwork/java/javamail/index.html>
- [10] Request for Comments [E]. <http://www.ietf.org/rfc.html>
- [11] Cristianini, Nello; and Shawe-Taylor, John; An Introduction to Support Vector Machines and other kernel-based learning methods [M], Cambridge University Press, 2000. ISBN 0-521-78019-5
- [12] Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines [J]. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011.
- [13] McCallum, A. and Nigam K. "A Comparison of Event Models for Naive Bayes Text Classification" [C]. In AAAI/ICML-98 Workshop on Learning for Text Categorization, pp. 41-48. Technical Report WS-98-05. AAAI Press. 1998.

- [14] IKNalyzer [E]. <http://code.google.com/p/ik-analyzer/>
- [15] Lawrence R. Rabiner, A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition [C]. Proceedings of the IEEE, 77 (2), p. 257–286, February 1989.
- [16] Tie-Yan Liu (2009), "Learning to Rank for Information Retrieval" [M], Foundations and Trends® in Information Retrieval, Foundations and Trends in Information Retrieval: Vol. 3: No 3 3 (3): 225–331, doi:10.1561/15000000016, ISBN 978-1-60198-244-5.
- [17] T. Joachims, Training Linear SVMs in Linear Time, Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD), 2006.
- [18] JGraphT [E]. <http://jgrapht.org/>
- [19] Prefuse [E]. <http://prefuse.org/>

致谢

在最后，我要感谢项目指导老师罗远哉副教授，创新实践基地的张丽老师以及其他老师同学。没有你们的帮助，我无法成功地完成该项目。