贴心云衣柜
——基于机器学习和社交网络的服装搭配推荐

曹佳涵[1)] 金亦凡[1)] 李鑫烨[1)] 李光耀[1)]

[1)](南京大学计算机科学与技术系 南京 210093)

摘 要 随着物质生活水平的不断提高，服装搭配已经成为一个出门前必须进行的活动。它既是展现自我个性与风采的恰当途径，又有着十分重要的社交意义。在这一背景下，本项目开发了一款基于机器学习和社交网络的服装搭配推荐 APP——贴心云衣柜（Sweet Wardrobe）。与市面上现有 APP 的差异之处在于，该 APP 不仅可以根据天气、场合等，从用户已有的衣物集当中为用户生成合理的穿衣搭配，还可以通过用户的穿衣历史信息和衣物搭配收藏，运用一种新兴的个性化推荐算法——协同过滤算法，向用户推荐社交网络中与该用户风格相近的用户以及他们的穿衣搭配收藏。文中对本 APP 的开发背景、工具及算法、实现过程、成品展示等进行了详细的阐述。

关键词 Android；服装搭配推荐；机器学习；协同过滤；社交网络

Sweet Wardrobe
—Dressing recommendations based on machine learning and social networking

Jiahan Cao [1)] Yifan Jin [1)] Xinye Li [1)] Guangyao Li [1)]

[1)](Department of Computer Science and Technology, Nanjing University, Nanjing 210093)

Abstract With the continuous improvement of material standard of living, dressing has become a necessary activity before going out. It is not only an appropriate way to show one's personality and style, but also of social significance. Sweet Wardrobe, a dressing recommendation APP based on machine learning and social network, is developed in this paper. Different from existing applications in the APP store, this APP can offer satisfactory clothing collocation for users from their existing clothing collection according to the weather conditions, wearing occasions and so on. Besides, it also recommends users with similar styles and their clothing collocation collection in social networks to the certain user based on his dressing history information and clothing collocation collection, applying a personalized recommendation algorithm, collaborative filtering algorithm. In this paper, the development background, tools and algorithms, implementation process and finished product display of the APP are described in detail.

Key words Android；dress recommendation；machine learning；collaborative filtering；social networking

1 引言

1.1 项目开发背景

在物质生活水平不断提高的现代社会，人们对于服装搭配的要求也在一天天地提高：不仅要做到舒适得体，还要用不同的穿搭应对不同的场合。可以说，服装搭配已经成为了现代人出门之前必须要解决的一个问题。然而与此同时，随着生活节奏的进一步加快，思考如何搭配着装和选择购买新衣物的时间也愈加缺乏。在这种情况下，我们不难发现，一个智能的、个性化的，能应对上述诸多问题的穿衣搭配推荐 APP，必然能够吸引一个很大的用户群体，在 APP 市场中占有一席之地。

1.2 需求分析

在项目开发的前期准备时，开发小组访问了 APP 商店，发现市场中已经存在很多有关穿衣搭配的 APP，但是这些 APP 的推荐功能较为基础，它们只是简单地处理了用户现有的衣物的搭配问题，并没有推荐系统的处理。事实上，用户更加青睐于个性化的 APP，期望 APP 能够符合自己的品味。更进一层的需求是满足大多数人都具有的融入集体的心理，在社交网络中分享自己的风格，并找到同道中人。

以网易云音乐为例，该款 APP 的成功之处并不在于其与其他音乐 APP 相同或相似的音乐播放功能或庞大的音乐库，而是在于它特有的个性化音乐推荐功能，以及完善的社交功能，包括音视频分享功能和评论功能等。用户可以通过强大的推荐功能找到更多符合自己听歌风格的歌曲，丰富个人歌单；同时用户还能够评论、分享自己喜欢的歌曲，参与到社区互动，也可以找到听歌风格相近的用户集群。这种用户体验，是一个单纯的音乐播放器所无法提供的。

因此，在分析大多数用户的需求、并与市面上的同类型或同性质的 APP 进行横纵向比较之后，开发小组明确了开发方向，即开发一款这样的 APP：它既有普通穿衣搭配 APP 的功能（包括衣物管理系统、服装搭配收藏系统以及基于用户现有衣物和天气等客观信息的搭配推荐系统），又有类似于网易云音乐的个性化推荐系统和社交系统。用户能够收到 APP 为其提供的符合该用户个性的一套或多套穿衣搭配（内含的衣物单品可能不在用户现有衣物中），也能够参加社交活动，查看好友的衣物搭配，同时在社交网络中寻找到与其风格类似的其他用户。

1.3 开发意义

从用户体验的角度分析，该 APP 能够提供当下市面上各种穿衣搭配 APP 所不能提供的一些用户体验。当用户想要在自己的衣柜中增添新衣物时，个性化的推荐功能可以更快地给用户指明方向；同时，在用户社群中找到风格类似的同道中人也能为用户带来极好的体验感，用户之间可以互相借鉴穿衣搭配，交流某种风格的穿搭心得等。

项目主要阶段以实现功能为主，在开发完成之后如果对整个项目进行优化，包括界面优化、性能优化等，那么这款 APP 成品必定能够占据一定市场，吸引一定量的用户集群。

1.4 论文结构

本文对该项目的整个开发流程都做了适当的阐述。第 2 节对项目的功能、框架及核心算法选择作了简要阐述；第 3、4 节对本项目核心功能的两大核心算法——决策树算法和协同过滤算法进行了详细的阐述；第 5 节对项目的整体功能实现作了简要阐述；第 6 节对整个项目进行了总结。

2 项目概述

2.1 项目功能简述

本项目开发的 APP 的核心功能有二：第一，根据天气等一些外在条件，在现有的衣物中为用户推荐一套适当的衣物搭配；第二，建立一个社交网络，根据该用户的穿衣风格，从社交网络中寻找与其风格相近的其他用户，并将这些用户或他们的衣物搭配收藏推荐给该用户。项目以这两个功能为核心进行架构。

除此之外，该 APP 还有一些优先级较低但同样十分必要的功能，例如天气、个人衣柜、服装搭配收藏、新衣物导入，以及一些社交网络的基本功能（好友系统、朋友圈等）。

2.2 项目框架介绍

从功能列表中，我们能够大致将所有功能分为三个大类：演示、存储和计算。演示是 APP 最基础的功能，用户的各种个人信息、为用户推荐的各种信息以及其他种种用户会想看到的信息需要在手机界面上向用户演示；用户的个人信息，例如衣物收藏等，在每次启动 APP 时均需要进行加载，故需要选择一个合适的存储方式；最后，为用户推荐信息需要进行大量的计算。

鉴于用户对 APP 的轻量化需求，开发小组决定开设一台服务器，将所有计算工作以及部分存储工作转移至服务器，以达到轻量化的目标。用户集的信息存储在服务器中，本用户的一些信息（例如衣物收藏）则存储在本地；推荐衣物搭配及用户的计算过程放在服务器端，通过 Socket 架设接口传输推荐请求和推荐结果。客户端的主要任务则是演示。

在分配好客户端和服务器各自负责的工作之后，项目整体功能的具体实现将在第 5 节阐述。

2.3 核心算法选择

在对应用于两大核心功能的推荐算法的选择上，开发小组的考虑如下：首先，对于基于已有衣物向用户进行穿衣搭配推荐这一功能，设想运用机器学习的方法。经过分析考虑，针对应用场景数据规模较小、可能存在缺失值的特点，决定使用决策树算法，这在之后的实践中被证明是一种效果较好的算法。其次，对于基于社交网络的多用户服装搭配推荐，决定使用协同过滤算法。与传统的基于内容过滤直接分析内容进行推荐不同，协同过滤分析用户兴趣，在用户群中找到与指定用户兴趣相似的用户，综合这些相似用户对某一信息的评价，形成系统对该指定用户对此信息的喜好程度预测。本文第 3-4 节将对上述两种算法进行详尽的阐述。

3 基于决策树的服装搭配推荐

3.1 应用场景分析

由没有免费午餐定理，Sliver Bullet，即在所有情形下均有效的解决方案是不存在的。对于不同的应用场景中机器学习算法的应用，需要充分考虑到其数据规模以及其他特点，不一味追求训练复杂模型。

考虑服装搭配推荐这一场景下，所选用机器学习算法在用户已有衣物中进行挑选搭配并推荐。现实生活中衣物数量往往十分有限，因此算法需要适用较小的训练数据集规模。同时，算法需要灵活地挑选多件衣物单品以构成具有一定整体风格的服装搭配，在此过程中可能出现推荐衣物单品时，与其搭配的衣物尚未选取或进行推荐，表现为训练数据中的缺失值。因此选取算法需对缺失值不敏感。综合这些特点，最终实现在小规模数据集上表现较好且对缺失值不敏感的，可解释性较好的决策树算法。

3.2 决策树算法

决策树是一类在分类问题中较为常见的机器学习方法。根据给定的训练数据集，构造出一个决策树模型，使它能在分类问题上对实例进行正确的分类。决策树模型本质上希望从训练集中归纳一组有效的分类规则或由训练集估计条件概率模型，以反映对象属性值与对象标签之间的映射关系。对于训练得到的决策树模型，由树节点条件判断向下遍历，最终到达叶节点对应标签值即预测结果。

决策树算法学习的过程通常是一个从训练数据集递归选择最优划分属性并根据该特征对训练数据集进行分割。这一过程本质上是对样本特征空间的划分，在具体实现中体现为决策树的构造过程。

算法核心在于如何从训练数据集特征中选择属性以划分样本特征空间，使子特征空间内样本尽量属于同一类中，尽量提高决策树子节点所包含样本的纯度。常用的衡量纯度的度量有信息熵、信息增益、基尼指数等。具体算法实现中，采用 C4.5 算法，以信息增益熵为标准为标准选择最佳划分属性，递归地划分样本数据并构建决策树，预测用户在不同的情形下对服装搭配的选择。

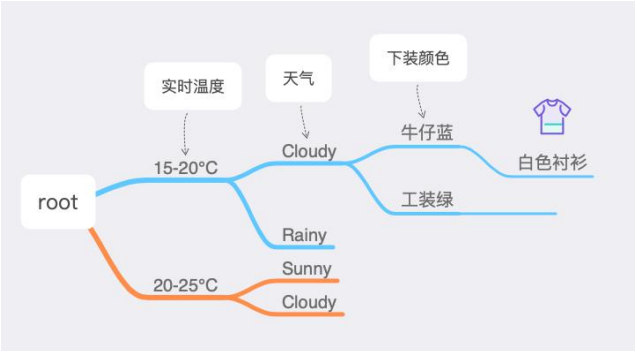

Figure1. 决策树示意图

3.3 缺失值处理

算法挑选推荐衣物单品以形成具有一定风格的搭配。在推荐衣物单品过程中，往往存在与其搭配的衣物尚未选取或尚未进行推荐。如果简单地放弃不完整样本，无缺失值的样本进行学习，显然是对不完整样本中所包含信息的浪费。若以默认值来作为数据取值进行学习，最终默认值也将在很大程度上影响推荐结果。我们最终采取只对有效值计算信息增益熵，划分样本数据时对缺失值样本进行加权的方法，来避免缺失值过多而导致默认值占优势的情形。

3.4 连续值处理

在考虑实际应用过程中，我们发现有许多用户数据，比如实时温度，都是连续值数据，如果以离散值处理方法将导致数据样本被过度分割。由于目前已知连续值变量为温度、湿度、风力这些已知变化范围的变量。在尽可能少损失信息并避免问题复杂化的条件下，我们采取了根据经验划分的方法，将值差距不大的样本划分到同一类，经过实验发现能够较好地避免过度划分样本的问题。

3.5 过拟合

在构建决策树过程中，我们发现在接近决策树叶节点的数据样本划分中，由于此时剩余属性信息增益熵都较低，与用户对衣物的选择相关性不大，容易出现完全不必要地划分出过多样本的情形。因此我们采取了剪枝的处理方法，在每次划分过程中均用性能评估函数考察此次划分能否带来泛化性能的提升，避免了依据相关性较低的属性不必要地划分样本数据。

4 基于协同过滤的多用户推荐

4.1 以用户为基础(User-based)的协同过滤

协同过滤(Collaborative Filtering)，简单来说是利用某兴趣相投、拥有共同经验之群体的喜好来推荐用户感兴趣的信息，个人透过合作的机制给予信息相当程度的回应（如评分）并记录下来以达到过滤的目的进而帮助别人筛选信息，回应不一定局限于特别感兴趣的，特别不感兴趣信息的纪录也相当重要。协同过滤又可分为评比(rating)或者群体过滤(social filtering)。其后成为电子商务当中很重要的一环，即根据某顾客以往的购买行为以及从具有相似购买行为的顾客群的购买行为去推荐这个顾客其“可能喜欢的品项”，也就是借由社群的喜好提供个人化的信息、商品等的推荐服务。

2001 年，Sarwar 提出了基于项目的协同过滤推荐算法(Item-based Collaborative Filtering Algorithms)。以项目为基

础的协同过滤方法有一个基本的假设“能够引起用户兴趣的项目，必定与其之前评分高的项目相似”，透过计算项目之间的相似性来代替用户之间的相似性。 方法步骤：

(1) 收集用户信息：收集可以代表用户兴趣的信息。一般的网站系统使用评分的方式或是给予评价，这种方式被称为“主动评分”。另外一种是“被动评分”，是根据用户的行为模式由系统代替用户完成评价，不需要用户直接打分或输入评价数据。电子商务网站在被动评分的数据获取上有其优势，用户购买的商品记录是相当有用的数据。

(2) 针对项目的最近邻搜索：先计算已评价项目和待预测项目的相似度，并以相似度作为权重，加权各已评价项目的分数，得到待预测项目的预测值。例如：要对项目 A 和项目 B 进行相似性计算，要先找出同时对 A 和 B 打过分的组合，对这些组合进行相似度计算。

(3) 对于项目来讲，它们之间的相似性要稳定很多，因此可以脱机完成工作量最大的相似性计算步骤，从而降低了在线计算量，提高推荐效率，尤其是在用户多于项目的情形下尤为显著。

4.2 隐语义模型(Latent Factor Model)

在我们的推荐环境设定下，每个用户的数据由他对各类衣物的偏好程度组成，这个偏好程度由他的历史穿衣数据以及浏览记录数据计算得到。但我们真正要从数据中提取的，是每个用户对于不同穿衣风格的偏好，由于不同种类的衣物可能会属于同一种风格，因此衣物种类之间并不是完全独立的。在理想情况下，我们希望通过一种相互独立的属性，能将不同风格的用户划分开来，而这种属性并不是显式地体现在数据中的。另一方面，由于每个用户并不是对于每种衣物都有数据，因此用户-衣物矩阵是稀疏的，很多信息因此缺失，我们需要重新计算并预测新的用户-衣物矩阵来进行用户相似度计算，因此我们采用隐语义模型(Latent Factor Model)来求解问题。

4.2.1 基本思想

核心思想: 通过隐含特征(latent factor)联系用户兴趣和物品。具体来说，就是对于某个用户，首先得到他的兴趣分类，然后从分类中挑选他可能喜欢的物品。基于兴趣分类的方法需要解决 3 个问题：

(1) 如何对物品进行分类？
(2) 如何确定物品对哪些类的物品感兴趣，以及感兴趣的程度？
(3) 对于一个给定的类，选择哪些属于这个类的物品推荐给用户，以及如何确定这些物品在一个类中的权重？

隐含语义分析技术(latent variable analysis)采取基于用户行为统计的自动聚类，可以较好解决上面提出的问题。

(1) 代表用户意见分类来自对用户行为的统计，和 ItemCF 在物品分类方面的思想类似，如果两个物品同时被多个用户喜好，那么这两个物品可能属于同一个类
(2) 控制分类粒度自定义分类个数
(3) 一个物品多分类计算出物品属于某个类的权重，因此每个物品都不是硬性地被分到某一个类中
(4) 多维度分类基于用户的共同兴趣计算出来的，如果用户的共同兴趣是某一个维度，那么 LFM 给出的类也是相同维度
(5) 物品在分类下的权重统计用户行为决定物品在某一个分类中的权重，如果某个类的用户都会喜欢某个物品，那么这个物品在这个类中的权重可能比较高

4.2.2 算法

对于一个的用户行为数据集(数据集是一个|U| × |I|的矩阵 R，|U|为用户数量，|I|为衣物种类数量，$R_{i,j}$表示第 i 个用户对于第 j 种衣物的偏好值)，使用 LFM 对其建模后，我们可以得到如下图所示的模型：(假设数据集中有 3 个用户, 4 种衣物, LFM 建模的分类数为 4)

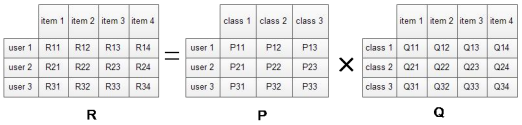

Figure2. User-Item 矩阵分解

对于一个用户来说，当计算出他对所有 item 的偏好后，就可以进行排序并作出推荐。LFM 算法设定 K 个类(Class)，将 R 矩阵表示为 P 矩阵和 Q 矩阵相乘。其中 P 矩阵是 User-Class 矩阵，矩阵值$P_{i,j}$表示的是$User_i$对$Class_j$的偏好值；Q 矩阵式 class-item 矩阵，矩阵值$Q_{i,j}$表示的是$Item_j$在$Class_i$中的权重，权重越高越能作为该类的代表。所以 LFM 根据如下公式来计算用户 u 对衣物种类 i 的兴趣度：

$$R_{ui} = P_u Q_i = \sum_{k=1}^{K} P_{u,k} Q_{k,i}$$

在计算矩阵 P 和矩阵 Q 中参数值时，我们使用最小化损失函数的方法来求参数。损失函数如下所示：

$$L = \sum_{(u,i) \in (U,I)} \left(R_{u,i} - \widehat{R_{u,i}}\right)^2$$

$$= \sum_{(u,i)\in(U,I)} (R_{u,i} - \sum_{k=1}^{K} P_{u,k}Q_{k,i})^2 + \lambda_1||P_u||^2 + \lambda_2||Q_i||^2$$

上式中最后两项是用来防止过拟合的正则化项，λ_1和λ_2为超参数，根据实验调参得到。损失函数的优化使用随机梯度下降算法：

(1) 通过求参数$P_{u,k}$和$Q_{k,i}$的偏导确定最快的下降方向；

$$\frac{\partial L}{\partial P_{u,k}} = -2\left(R_{u,i} - \sum_{k=1}^{K} P_{u,k}Q_{k,i}\right)Q_{k,i} + 2\lambda_1 P_{u,k}$$

$$\frac{\partial L}{\partial Q_{k,i}} = -2\left(R_{u,i} - \sum_{k=1}^{K} P_{u,k}Q_{k,i}\right)P_{u,k} + 2\lambda_2 Q_{k,i}$$

(2) 迭代计算不断优化参数，直到参数收敛。

$$P_{u,k} = P_{u,k} + \alpha\left(\left(R_{u,i} - \sum_{k=1}^{K} P_{u,k}Q_{k,i}\right)Q_{k,i} - \lambda_1 P_{u,k}\right)$$

$$Q_{k,i} = Q_{k,i} + \alpha\left(\left(R_{u,i} - \sum_{k=1}^{K} P_{u,k}Q_{k,i}\right)P_{u,k} - \lambda_2 Q_{k,i}\right)$$

其中，α是学习速率，α越大，迭代下降的越快。经过实验和调参，我们最终确定α的值。

5 功能实现

在确定了主要的功能和算法之后，进行项目主体框架的开发，从而实现主要功能，同时添加进其他必要的辅助功能。项目框架的开发分为三个个部分：客户端 APP 主界面，服务器端数据库/计算框架，以及用于在客户端和服务器之间交换数据的 Socket。本节将简要阐述三个部分各功能的实现，同时辅以必要的成品效果展示图。

5.1 客户端

客户端使用一款 Google 公司推出的 Android 开发 IDE，Android Studio 进行开发。

在客户端，预期实现的项目功能有如下几点：

5.1.1 主界面

主界面上提供了一些必要的功能。首先是天气功能，通过选择地区来联网获取该地区最近更新的当前天气；其次，主界面上展示了一套通过决策树推荐功能推荐出来的衣物，包括外套、上衣、下装和鞋子（决策树推荐结果由 Socket 从服务器传至客户端）。最后，在主界面上有进入其他页面的按键。

5.1.2 衣物导入

衣物导入界面用于让用户导入衣物至衣柜。通过拍照或相册读取获取需要添加的衣物图片后，用户对其进行参数选择，包含衣物的种类和颜色，然后存入衣柜。

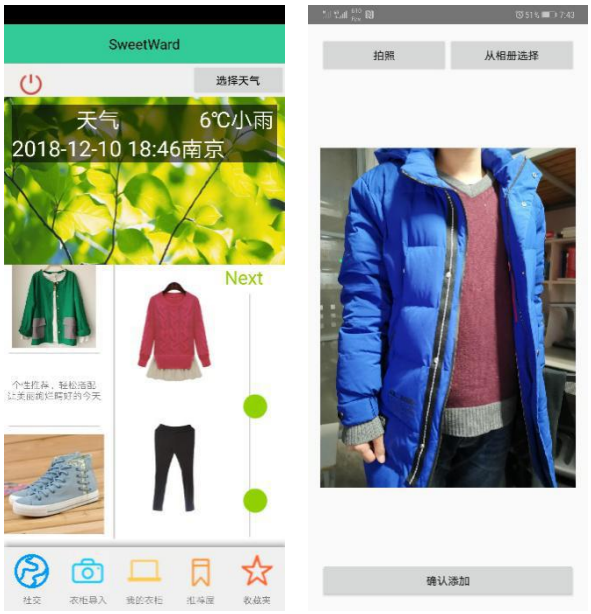

Figure3. 主界面及衣物导入界面

5.1.3 衣柜、收藏夹及推荐屋

衣柜界面展示了用户当前的所有衣物，而收藏夹界面则展示用户收藏的所有套装（上衣+下装）。推荐屋界面展示了通过协同过滤（推荐结果同样来自服务器）推荐出来的一套衣物搭配。

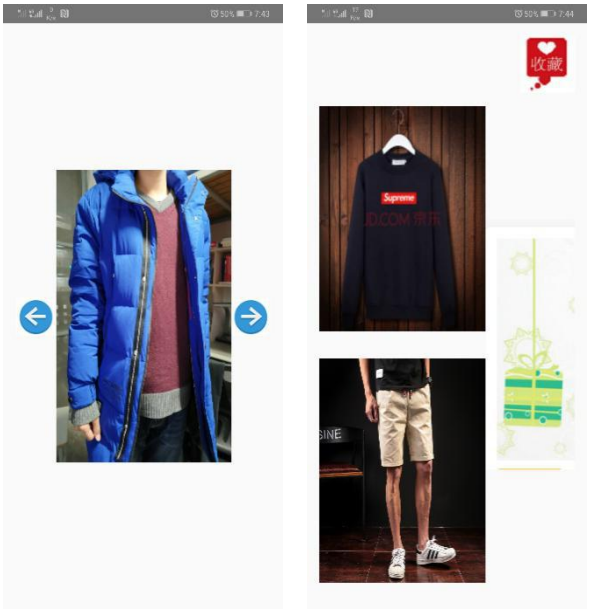

Figure4. 衣柜界面及推荐屋界面

5.1.4 社交网络

通过主界面的“社交”按钮，进入社交网络主界面，包括朋友圈、好友列表、推荐好友和个人信息四个子界面。其中朋友圈展示了其他用户的衣物图片，以朋友圈动态的方式展示；好友栏展示了该用户的好友列表；推荐好友栏通过协同过滤推荐，向用户推荐社交网络中与其穿衣品味相近的其他用户，用户可以选择向这些用户发出好友申请；个人信息页面中，用户可以对个人信息进行更改。

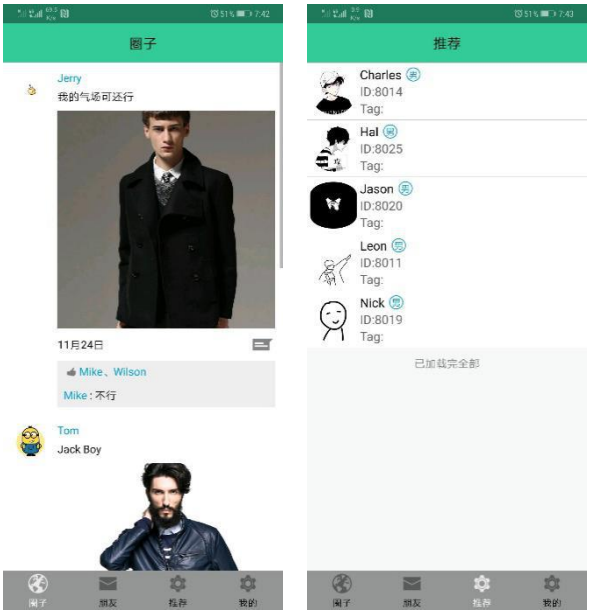

Figure5. 社交界面的朋友圈界面及推荐好友界面

5.2 Socket

通信传输模块使用 java TCP Socket 处理，建立端到端可靠连接。支持处理文本和图片(传输中都转为字节流)。通过自定义文本协议对客户端发来的请求做功能判断。将数据流交由后台数据库存储，并调用特定模块处理，返回处理结果给客户端。

5.3 服务器端

服务器端目前部署在腾讯云上。主要完成大规模的计算任务。其全部计算数据通过后台数据库进行存储与管理。

5.3.1 后台数据库管理

后台数据库使用 MySQL 搭建。主要包括用户、衣物、动态、评论表等。

5.3.2 计算推荐任务

服务器端包括决策树推荐算法、协同过滤推荐算法的实现。决策树算法返回衣物推荐列表给特定用户。协同过滤算法完成相似用户计算，并更新数据库中的用户推荐列表。

5.3.3 社交圈子的交互

社交圈子采用开源的APIJSON网络传输协议(前端可以自由定制数据，而后台提供公共接口)。其内部封装好了对数据库事物操作的 API。前端涉及社交圈子的活动全部转为对后端数据库事物操作。

6 总结

在本次项目实践中，我们项目小组对于机器学习算法以及软件项目开发有了更加深刻的理解。我们开发了一款穿衣搭配 APP——贴心云衣柜，用来满足人们对于节省穿衣搭配时间、以及进行社交活动的需求。在项目开发过程中，我们也学习了 Android Studio 的使用，后台服务器及 Socket 的搭建，以及系统化进行项目开发的软件工程方法。当前版本的贴心云衣柜实现了所有核心功能和绝大部分的辅助功能。在之后的后续开发过程中，我们会进一步对项目进行优化，例如增加运行流畅度、优化算法精准度、添加一些更加强大的辅助功能等。

致　谢　特别感谢项目指导老师——卜磊副教授自始至终对我们所给予的支持和鼓励！

参 考 文 献

[1] https://itunes.apple.com/cn/app/chuan-yi-zhu-shou-jiao-ni/id577700263?mt=8

[2] http://www.cs.cmu.edu/%7Etom/pubs/MachineLearning.pdf

[3] https://en.wikipedia.org/wiki/Netflix_Prize

[4] https://en.wikipedia.org/wiki/Collaborative_filtering#Auxiliary_Information_in_Collaborative_Filtering

[5] https://en.wikipedia.org/wiki/Decision_tree

[6] 项亮编著，2012 年 6 月，《推荐系统实践》第三章——推荐系统冷启动问题

[7] https://en.wikipedia.org/wiki/Collaborative_filtering#Data_sparsity

[8] https://en.wikipedia.org/wiki/Sparse_matrix

[9] Meyer, C. D. (2000), Matrix Analysis and Applied Linear Algebra

[10] Wall, Michael E., Andreas Rechtsteiner, Luis M. Rocha (2003). "Singular value decomposition and principal component analysis"

[11] https://console.heweather.com/my/service

[12] 郭霖编著，2016 年 12 月，《第一行代码——Android Studio》

[13] https://github.com/TommyLemon/Android-ZBLibrary

[14] Badrul Sarwar, Badrul Sarwar, Joseph Konstan, John Riedl, WWW '01 Proceedings of the 10th international conference on World Wide Web, Item-based collaborative filtering recommendation algorithms

[15] Rodolphe Jenatton, Nicolas L. Roux, Antoine Bordes, Guillaume R. Obozinski, Advances in Neural Information Processing Systems 25 (NIPS 2012), A latent factor model for highly multi-relational data