2017 开放学术精准画像大赛 解题报告

陆承镪, 中国科学技术大学 lunar@mail.ustc.edu.cn
任震, 东南大学 Legend94rz@gmail.com
肖驰, 中国科学技术大学 xiaochi@mail.ustc.edu.cn
马莹莹, 上海科技大学 linxiaonai@outlook.com

我们队队名为【苟】，成绩第一阶段第四，第二阶段第二。成员两名来自中国科学技术大学，一名来自东南大学，一名来自上海科技大学。

1 任务一

在任务一中，我们发现，爬虫的质量（包括网络的质量）会较大影响到结果的正确性。下面除了每项学者的信息外，我们也会单独谈一下爬虫的问题。

1.1 主页 Homepage

在判断主页上，综合比较各种算法之后，我们选择了 xgboost 来做二分类。对于首页搜索结果的每条记录提取特征后进行是否是主页的分类，其中判断为主页概率最高的记录会被选择为学者的主页。在具体实现上，由于正负样本的不均衡（9：1），对负样本进行负采样可以使结果有较大的提升。特征方面我们主要使用了如下特征：

- 网站在搜索结果中的排名
- 学者所在机构是否是大学/学校
- 该条搜索结果中是否包含引用信息
- 搜索结果标题/摘要以及学者所在机构的文本长度
- url 中是否包含给定关键词，包括 *edu,org,gov,linkedin*
- 搜索结果摘要中是否包含 *mail,address*
- 搜索结果的标题/摘要中的学者名字得分。这里我们将学者名字分词为 n 份，如果其中 m 份在文本中出现，那么名字得分就是 m/n，同时，若名字缩写在文本中出现，得分加 0.2 分。

在如上处理后，我们在训练集和测试集上都能取得 0.7 左右的准确率，考虑到数据噪音以及搜索结果中多个正确主页的存在，真实准确率估计为 0.8 左右。

1.2 性别 Gender

性别上面我们主要通过学者的名字来判断。我们用 8000 个名字训练了一个朴素贝叶斯分类器，其中的用到的特征是名字后 2 位到后 7 位字母，前 2 位字母以及名字长度。如果学者名字长度不足的，用空格填满。结果上能取得接近 0.9 的准确率，但是发现对中国人的拼音名字的判断准确较低。

1.3 邮件 Email

邮件识别方面，我们主要通过正则表达式来识别。在看了许多数据后，我们写了一个非常复杂的正则表达式，虽然运行速度较慢，但是基本上能够识别出 80% 多的电子邮箱（如果网页中有的话）。为了处理网页中非该学者的电子邮箱，如机构的联系邮箱，院校的邮箱等等，我们采用了关键词过滤。针对邮件是以图片方式存在的情况，我们也用关键词过滤出了可能的邮件图片，如果文本中没有检测到邮件地址，我们就将该图片 url 设为邮件信息。

1.4 照片 Pic

照片处理中，我们使用正则表达式将网页中所有可能图片地址（首先用关键词过滤到无用图片）提取出来并下载。然后用 tensorflow 构建了一个人脸识别器，将其中包含且只包含一张人脸的照片作为学者的照片。

1.5 职位职称 Position

这里基本上都是使用的关键词包含来处理，如果文本中出现了职位职称表中职位，我们就认为他拥有该职称。在实现上我们做了一些过滤，例如检测职称附近是否有学者名字，该职称是否已经过时，职称附近有没有正向关键词出现等等。

1.6 国家 Location

此处要求提取的位置信息只有国家，所以我们可以认为该信息只会存在在一行文本中。对于每行文本，如果文本包含国家的信息，我们会依据其包含关键词的情况来给分，如果分数超过阈值，则判定为结果。主要关键词有 *mail, address, phone, tel, fax, room, office, road, street, avenue* 以及学者姓名。

1.7 爬虫

由于学者网站服务器质量良莠不齐，在采集信息时常常会遇到采集不到的情况。为了在决赛的 24 小时内尽快采集完信息供之后应用，我们写了并行化的爬虫，并通过各类手段进行网络加速。对于有蜜罐陷阱和脚本内容的网站，使用无头浏览器模拟用于访问来获取信息。

2 任务二: 学者兴趣标签预测

程序利用学者发表的论文标题信息、论文的引用和学者投稿期刊信息来预测学者的兴趣标签。程序先利用四个简单模型分别计算学者的兴趣得分，然后按照权重求出学者兴趣的加权平均得分，最后选择得分最高的 5 个兴趣作为最终预测结果。

模型中出现的符号如下：A 代表学者，A^T 表示训练集中的学者，A^U 表示待预测的学者，I 表示兴趣标签，T 表示论文标题，Id 表示论文的序号，P 表示论文的发表刊物，$S^P_{A_i}$ 代表第 i 学者投稿刊物集合，$S^I_{A_i}$ 代表第 i 位学者的兴趣集合。

2.1 模型一

模型一的基本思想是：发表论文标题内容相似的学者，他们的研究兴趣相似。

程序首先从论文集合中找出每位学者 A_i 发表的文章，即 $S^T_{A_i} = \{T^i_1, T^i_2, ..., T^i_k\}$，集合 $S^T_{A_i}$ 可以描述学者的研究内容。为了找出训练集中和待预测学者 A^U_i 研究内容相似的学者，模型利用待预测的学者 A^U_i 发表论文的集合 $S^T_{A^U_i}$ 和训练集中学者 A^T_j 发表的论文集合 $S^T_{A^T_j}$ 进行标题文本相似度计算。论文标题

集合 $S^T_{A^U_i}$ 与 $S^T_{A^T_j}$ 的相似度越大，表示学者 A^U_i 与 A^T_j 的研究内容越相近，学者 A^U_i 和学者 A^T_j 的兴趣越相似。因此学者 A^U_i 与 A^T_j 的相似度 $Sims(A^U_i, A^T_j)$ 可以定义为如下形式：

$$Sims(A^U_i, A^T_j) = Similarity(S^T_{A^U_i}, S^T_{A^T_j}) \tag{1}$$

$Sims(A^U_i, A^T_j)$ 可以作为学者 A^U_i 和 A^T_j 兴趣相似度的表示。
最终，学者 A^U_i 对兴趣 I_j 的得分可以表示为:

$$Score(I_j|A^U_i) = \sum_{k=0}^{P} Sims(A^U_i, A^T_k) \cdot Identify(I_j, S^I_{A^T_k}) \tag{2}$$

$$Identify(I_j, S^I_{A^T_k}) = \begin{cases} 1 & , I_j \in S^I_{A^T_k} \\ 0 & , I_j \notin S^I_{A^T_k} \end{cases} \tag{3}$$

根据上式，模型选取得分最高的 5 个兴趣作为学者的兴趣标签。

2.2 模型二

模型一先给待预测学者 A^U_i 在训练集中寻找研究内容相似的学者，再将学者的兴趣赋给待预测学者 A^U_i。这个模型有两个问题：

(1) 学者 A^T_j 对自己的三个兴趣 $\{I^j_1, I^j_2, I^j_3\}$ 的感兴趣程度不同；
(2) 待预测学者 A^U_i 和找出的相似学者 A^T_j，他们可能只有部分研究内容相似；

模型一使用同样的权重 $Sims(A^U_i, A^T_j)$ 将兴趣 $\{I^j_1, I^j_2, I^j_3\}$ 赋给待预测学者 A^U_i，不能真实地表示学者 A^U_i 研究兴趣。为了解决上述问题，模型二直接计算待预测学者 A^U_i 和兴趣 I_j 之间的相似度。

模型二的基本思想是，学者发表的论文标题描述了学者的研究兴趣。程序将训练集中学者发表的论文集合，按照学者的兴趣进行分类。假设学者 A^T_i 发表的论文集合 $S^T_{A^T_i}$，兴趣集合 $S^I_{A^T_i}\{I^i_1, I^i_2, I^i_3\}$，则将集合 $S^T_{A^T_i}$ 加入到集合 $S^T_{I^j_1}$、$S^T_{I^j_2}$ 和 $S^T_{I^j_3}$ 中，其中 $S^T_{I^j_1}$ 表示属于兴趣 I^j_1 的论文标题集合。

每个兴趣 I_i 都有相应的论文集合 $S^T_{I_i} = \{T^i_1, ...T^i_p\}$，集合 $S^T_{I_i}$ 可以用来描述兴趣 I_i。预测时，程序计算出集合 $S^T_{A^U_i}$ 和集合 $S^T_{I_j}$ 的相似度，将其作为学者 A^U_i 对兴趣 I_j 的得分，即：

$$Score(I_j|A^U_j) = Similarity(S^T_{A^U_i}, S^T_{I_j}) \tag{4}$$

根据上式，模型选取得分最高的 5 个兴趣作为学者的兴趣标签。

2.3 模型三

此模型的基本思想是：学者发表的文章，引用的文章和引用学者的文章可以反映出学者的研究兴趣。研究兴趣相似的学者在上述三种情况下重合度较高。学者 A_i 上述特征可以表示为集合 $S^{Id}_{A_i} = \{Id^i_1, ..., Id^i_k\}$，模型利用集合 $S^{Id}_{A_i}$ 计算学者之间的相似度。

模型使用 Jaccard 相似系数作为相似性度量函数：

$$Sims(A^U_i, A^T_j) = \frac{|S^{Id}_{A^U_i} \cap S^{Id}_{A^T_j}|}{|S^{Id}_{A^U_i} \cup S^{Id}_{A^T_j}|} \tag{5}$$

和模型一类似，算法将相似函数 $Sims(A^U_i, A^T_j)$ 作为学者 A^U_i 兴趣的得分。A^U_i 对兴趣 I_j 的感兴趣程度可以表示为:

$$Score(I_j|A^U_i) = \sum_{k=0}^{P} Sims(A^U_i, A^T_k) \cdot Identify(I_j, S^I_{A^T_k}) \tag{6}$$

$$Identify(I_j, S^I_{A^T_k}) = \begin{cases} 1 & , I_j \in S^I_{A^T_k} \\ 0 & , I_j \notin S^I_{A^T_k} \end{cases} \tag{7}$$

2.4 模型四

此模型的基本思想是：期刊通常收录某一领域的论文，具有一定的兴趣集合。学者将论文投到与自己研究兴趣相近的期刊。模型先计算期刊的兴趣分布，然后根据待预测学者投稿期刊的信息，预测学者的兴趣。

模型将训练集中学者的兴趣指派为其投稿期刊的兴趣，由此可以得到期刊的兴趣集合 $S^I_{P_i} = \{I^i_1, ..., I^i_k\}$，则

$$Score(I_k|P_i) = \frac{C(I_k, S^I_{P_i})}{\sum_{j=0}^{l} |(S^I_{P_i})|} \tag{8}$$

在计算待预测学者兴趣时，先统计待预测学者 A^U_i 的发表的期刊信息 $S^P_{A^U_i} = \{P^i_1, ..., P^i_k\}$。根据集合 $S^P_{A^U_i}$ 可以计算出学者对于期刊 P_i 的偏好：

$$w_i = \frac{C(P_i, S^P_{A^U_i})}{|(S^P_{A^U_i})|} \tag{9}$$

其中 $C(P_i, S_{A^U_i})$ 表示集合 $S_{A^U_i}$ 中 P_i 的个数。由此可以计算出待预测学者对兴趣 I_i 的得分：

$$Score(I_j|A^U_i) = \sum_{k=0}^{P} w_k \cdot Identify(I_j, S^I_{A^T_k}) \tag{10}$$

$$Identify(I_j, S^I_{A^T_k}) = \begin{cases} 1 & , I_j \in S^I_{A^T_k} \\ 0 & , I_j \notin S^I_{A^T_k} \end{cases} \tag{11}$$

2.5 预处理

模型一和模型二涉及计算文本相似度的计算。文本处理过程包括：

(1) 去停用词
(2) 将文本转化为词袋模型
(3) 文本转成 TF-IDF 表示
(4) LSI 模型降维 [2]
(5) 计算文本相似度

这部分主要利用了开源工具 gensim [1] 计算文本相似度。

[1] https://radimrehurek.com/gensim/

2.6 模型融合

上述四个模型利用不同特征预测学者的兴趣。为了使预测更加客观、合理，程序对四个模型进行了融合。待预测学者 A_i^U 对兴趣 I_j 的最终得分如下：

$$Score(I_j|A_i^U) = \sum_{k=0}^{4} \alpha_k Score_k(I_j|A_i^U) \tag{12}$$

3 任务三

任务三是对学者各论文未来的被引总数进行预测，给定的数据库含有 170 万的学者，以及他们发表的 300 万篇文章的信息。而其中并未包含文章的摘要部分，因此若通过知识发现或者其他基于话题的模型，得到的效果可能不令人满意，需要另辟蹊径。

3.1 概述

首先，我们根据实际情况，提出一个假设，后续的工作都基于如下假设：

假设 *3.1.* 一个学者截至给定时间 t 的被引总数，等于其发表的各论文截至时间 t 的被引总数之和。更正式地：

$$S(people, t) = \sum_{paper \in papers} s(paper, t) \tag{13}$$

其中，$papers$ 是该学者发表的（或参与发表的）文章的集合。

借助假设3.1，我们把对学者被引总数的预测，转换到求论文的被引总数。即，如果能求得各论文在未来时间 t 的被引总数，那么学者在时间 t 的被引总数就是一个简单求和过程。下面一节我们来讨论如何求论文的未来被引总数。

3.2 预测论文的被引总数

对于单篇论文 p，它在未来时间 t_{future} 的被引总数，可以用该论文截至某历史时间 t_{his} 的被引总数乘以一个系数 ω 来近似。我们不打算给出证明，但是可以简单思考一下，假设论文 p 现在以及未来的被引总数分别是 s_{p1}、s_{p2}，那么总可以找到一个系数 ω_p，满足：

$$s_{p2} = \omega_p * s_{p1} \tag{14}$$

请注意，式14与时间没有太大关系，只要满足 s_{p1} 的时间节点早于 s_{p2} 的，并且是真实数据即可。在下文中，我们称 ω_p 为文章 p 的**放大系数**。

那么为什么说是近似，而不是“精确地等于”呢？因为上述结论并不适用于 s_{p1} 等于 0 的情况，当然，的确可以通过在式14的右端加一个常数 c，就总能找到一对常数 ω 与 c，使得上述结论成立，但是我们并不打算这样做，原因见后文。如果不加该常数，那么就只能说是“近似”。

3.3 训练

借助假设3.1与式14，我们可以把目标问题重写为：

$$S(people, t_{future}) = \sum_{paper \in papers} \omega_{paper} * s(paper, t_{his}) \tag{15}$$

这是一个线性回归问题。

我们已知 $S(people, t_{future})$，因为当 t_{future} 取 2017 年 6 月的时候，这一部分由训练数据给出。同时，当 t_{his} 取 2013 年底时，$s(paper, t_{his})$ 可由给定的数据集 $papers.txt$ 统计得出。我们仅需要对每篇文章找到最优的 ω_{paper}。

更进一步地，即是解决如下问题：

$$\boldsymbol{\omega} = \arg\min_{\boldsymbol{\omega}} \mathrm{Err}(\boldsymbol{s}_{his} \cdot \boldsymbol{\omega}, \boldsymbol{S}_{future}) \tag{16}$$

其中，$\boldsymbol{\omega}$ 是一个列向量，长度为文章数量，每个元素 $\boldsymbol{\omega}_i$ 的含义表示第 i 篇文章的放大系数；$\boldsymbol{s}_{his}$ 是一个矩阵，行数为学者数量，列数为文章数量，第 i 行第 j 列的元素表示学者 i，对于编号为 j 的文章，截至某历史时刻 his 的被引总数（若第 j 篇文章不是学者 i 发表的，则该位置为 0）；$\boldsymbol{S}_{future}$ 是一个列向量，长度为学者数量，每个元素表示学者 i 截至某未来时刻 future 的被引总数，即训练数据；$\mathrm{Err}(a, b)$ 表示 a 与 b 的误差，计算公式由比赛规则给出。

显然，该回归问题的训练集 $\boldsymbol{s}_{his}$ 会有 300 万维特征，100 万个样本，若用离线方法无论是时间复杂度还是空间复杂度都是无法忍受的，因此我们借助了在线学习的方式 [1] 来训练 $\boldsymbol{\omega}$。

同时，我们发现，优化上述问题的过程相当于求解一个含有 300 万个未知数、100 万个方程构成的方程组，这会有无数个解，我们当然希望未知数越少越好，因此，在3.2节中，我们使用近似的方式来表示 s_{p2} 而不加常数的原因就在于此，因为那样的话会使未知数的数量增加一倍，即有 600 万个，而只有 100 万个方程，这是不能接受的。

4 优化

针对实验数据，我们进行了如下优化：

(1) 由于文章 [1] 是一种单遍算法，从实验结果来看，训练一遍并不能得到最好的效果，因此我们在条件允许的情况下尽可能多地增加训练次数。
(2) 尽管在线学习是一种减少空间开销、计算复杂度的好方法，但是我们觉得仍然不令人满意。注意到式16中的 $\boldsymbol{s}_{his}$ 是一个极为稀疏的矩阵，因为一个学者不可能发表数万篇文章。矩阵中绝大部分都是 0 元素。因此在在线学习的训练过程中，对每个样本，我们只利用非 0 的有效值训练，这样可以显著提高效率。
(3) 由于比赛规则计算误差的公式不太好直接优化，若是直接用给定的公式去计算误差，那么目标问题将难以求解，因此我们采用均方误差公式来近似。从实验结果来看，这一近似也是比较成功的。

REFERENCES

[1] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research* 7, Mar (2006), 551–585.
[2] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science* 41, 6 (1990), 391.