官方媒体的社交网络社区发现

陶雪然（201911180118）

北京师范大学统计学院

摘要：现实世界的社交网络中往往存在社区结构，本文主要目的为探究 Facebook 网站上各官方页面的社交特征与影响力，挖掘该社交网络中存在的社区结构。首先，对诸多社区发现算法进行了初步比较，利用带有标签的基准数据集，分析了各个算法的表现，发现 Louvain 算法、谱聚类算法、SCORE 算法在社交网络社区发现中有较好的表现。其次，运用网络结构数据的描述分析方法，挖掘出该网站上影响力较大的官方用户。接下来，基于网络节点度大小，提取出了原社交网络中的核心网络。之后，使用数据驱动的方法，通过对比模块度大小、网络密度这些社区发现评估指标，选择了最优的算法 SCORE 算法和最优的社区数 13 个。最后，对提取出的核心网络进行社区发现，选择了结构较为简单的社区 10 和社区 13 进行了相应的分析及可视化。

关键词：社区发现，模块度，Louvain 算法，谱聚类

一、 引 言

1、问题背景

随着互联网技术的日益发达与大数据时代的来临，社交媒体在人们生活中的地位举足轻重，成为了人们一种主流的社交工具。正因如此，越来越多的官方媒体选择在社交网站上创建主页，以此传播信息，并扩大自身的影响力。

对于社交网站运营者而言，分析用户社交网络的特征，将会有助于社交网站制定更加合理且高效的个性化推荐，从而达到提升流量、增强用户粘性的目的。而官方账号作为社交网站用户的重要组成部分，是成千上万个体用户进行信息消费的主要来源之一。所以对官方账号的关联网络进行分析就显得更加重要和具有代表性。

为探究社交网站上各官方页面的社交特征与影响力，挖掘该社交网络中存在的社区结构。本报告以 2017 年 11 月 Facebook 网站上的官方页面数据为例进行社交网络分析。

2、数据介绍

Facebook[1]官方页面关联网络数据属于网络数据，由 22470 个节点和 171002 条连边

构成。 该社交网络中包括 Facebook 中的 4 个类型的官方账户页面，分别是政客、政府组织、电视节目和公司。

网络中的节点代表 Facebook 经过验证的官方页面，而边则是代表网站之间存在相互点赞的关系，即有社交联系。节点特征是从页面所有者对该页面的描述中提取出来的。不论两个网页之间进行了几次点赞，都仅仅视为有关联，因此该网络的边的权重均为 1，属于**无权网络**。因为两个主页之间的联系是相互的、没有方向的，所以该主页关联网络也是**无向网络**。

二、 建模和算法

1、图数据定义

在对社交网络分析之前，需要对网络数据（图数据）这一特殊数据类型进行定义。

定义 1.1 网络数据结构

图 G 记作 $G = (V, E, X, Y)$

其中，节点集为 V，边集为 E，节点特征为 X，边特征为 Y。

定义 1.2 节点度（Degree）

节点度用于描述某个节点与其他所有节点的连边数。

定义 1.3 网络密度

网络的密度是指图 G 中，实际出现的连边频数和所有可能出现的边数之比。

$$\mathrm{den}(G) = \frac{|E|}{|V|(|V|-1)/2}$$

节点的中心性特征：

节点的中心性特征反映了节点在网络中的“重要性”，从而有助于研究者发现社交网络中的核心结构所在与问题重点。

定义 1.4.1 接近中心性

$$C_{cl}(i) = \frac{1}{\sum_{j \in V} \mathrm{dist}(i,j)}$$

定义 1.4.2 介数中心性

$$C_B(i) = \sum_{s \neq t \neq i \in V} \frac{\sigma(s,t|i)}{\sigma(s,t)}$$

2、社区发现建模

社区发现（community detection）是用来揭示网络聚集行为的一种技术，实际就是一种网络聚类的方法，这里的“社区”并没有一种严格的定义，可以将其理解为一类具有相同特

性的节点的集合。

在社交网络中，往往具有鲜明的社区结构：具有共同兴趣或共同朋友的用户是同一个社区的成员。社区具有的特征是：社区内的节点连接紧密，不同社区之间节点连接稀疏。

定义 2.1　社区

给定一组社区$C = \{C_1, C_2 \ldots C_k\}$，每个社区$C_k$都是 G 的一个分割，依然保持它的局部特征和聚类特征，且满足$\mathrm{C_k} \cap \mathrm{C_{k'}} = \emptyset\ (\forall \mathrm{k}, \mathrm{k}')$，则$C$是一个不重叠的社区。

定义 2.2 社区发现

对于图中的每一个节点$\mathrm{v_i}$，将其依据一定准则划分到社区$\mathrm{C_k}$，每个节点有一个社区标签。

由于社区发现问题是无监督学习的问题，不存在真实标签。所以根据社区结构“社区内部连接紧密、不同社区之间连接稀疏”的思想，对社区发现问题的优化目标函数作如下定义：

定义 2.3 模块度（Modularity）

模块度是用来衡量一个社区的划分是否优良的目标函数。一个好的划分结果其表现形式是：在社区内部的节点相似度较高，而在社区外部节点的相似度较低。它的物理含义是社区内节点的连边数与随机情况下的边数之差，模块度越大，挖掘的社区内部连接越紧密，效果越好。它的取值范围是 [-0.5,1)，当取值在 0.3~0.7 之间时，则社区发现效果较好。

其定义如下：

$$\mathrm{Q} = \frac{1}{2\mathrm{m}} \sum_{\mathrm{i,j}} \left(\mathrm{A_{ij}} - \frac{\mathrm{k_i k_j}}{2\mathrm{m}} \right) \delta(\mathrm{c_i}, \mathrm{c_j})$$

其中，m 是网络中的连接数，i 和 j 是网络中任意的两个节点，当他们之间有连接时$\mathrm{A_{ij}} = 1$，否则为 0。$\frac{k_i k_j}{2m}$ 表示一个随机图中节点 i, j 节点可能相连的概率。$\mathrm{k_i}$代表节点 i 的度。$\delta(\mathrm{c_i}, \mathrm{c_j})$用于判断原图中 i,j 是否在一个社区内，若在同一个社区$\delta(\mathrm{c_i}, \mathrm{c_j}) = 1$，否则为 0。

定义 2.3’ 模块度简化形式

为了计算简便，可以将模块度简化为如下形式：

$$\mathrm{Q} = \sum_{\mathrm{c}} \left(\frac{\sum \mathrm{in}}{2\mathrm{m}} - \left(\frac{\sum \mathrm{tot}}{2\mathrm{m}} \right)^2 \right)$$

其中$\sum in$是社区 c 内的边数，$\sum tot$是社区 c 内节点的度数之和。

3、社区发现算法

常用的社区检测方法主要有如下几种[3]：有基于图分割的方法，如 Kernighan-Lin 算法，谱平分法(LPA)等。以及基于层次聚类的方法，如 GN 算法、Newman 快速算法等。还有基于模块度优化的方法，如 Louvain 贪婪算法、模拟退火算法等[4][5]。

由于社区发现的无监督方法无法通过训练模型选择变量和参数，在进行数据驱动的算法选择之前，应当考虑使用有实际标签的数据进行算法的评估。此时可以使用划分错误率为评价标准。本文选取了常见的社区发现算法，以及一个对谱聚类进行改进的算法。

进行算法比较时，考虑与本研究所关注的问题密切相关的社交网络基准数据集，并使用具有不同特点的网络数据，分析社区发现算法在数据上的表现。数据特征如下：

表 1 六个常用基准数据集

数据集	节点数	社区数	边数	度最小值	度最大值	平均度	数据特点
Karate	34	2	78	1	17	4.59	经典数据
Dolphins	62	2	159	1	12	5.12	经典数据
UKfaculty	79	3	552	2	39	13.97	连边较多
Football	110	11	570	7	13	10.36	社区较多
Simmons	1137	4	24257	1	293	42.67	网络异质性
Weblogs	1222	2	16714	1	351	27.35	网络较大

表 2 算法在基准数据上的社区发现错误率

数据集	LPA	GN	Louvain	Infomap	谱聚类	SCORE
Karate	5/34	1/34	1/34	1/34	0/34	0/34
Dolphins	6/62	2/62	1/62	4/62	1/62	0/62
UKfaculty	20/79	7/79	1/79	8/79	0/79	2/79
Football	29/110	7/110	21/110	9/110	5/110	5/110
Simmons	368/1137	137/1137	134/1137	237/1137	244/1137	127/1137
Weblogs	187/1222	62/1222	58/1222	69/1222	64/1222	51/1222

由预先的比较的错误率可知，Louvain 算法、SCORE 算法具有较好的表现，考虑使用其在我们后续的分析中重点使用。

(1) Louvain 算法[6]

Louvain 算法是一种贪婪优化方法，又被称为 Fast unfolding 算法。其目标是对网络的模块度进行优化，将网络划分为密集连接的节点群。在模块性和计算时间上都优于其他所有的社区发现方法。主要分为两步骤：模块化优化、社区聚集。

算法流程：

a) 将图中的每个节点看成一个独立的社区，此时社区的数目与节点个数相同。计算此时的模块度；

b) 对每个节点 i，依次尝试把节点 i 分配到其每个邻居节点所在的社区，计算分配前与分配后的模块度变化 ΔQ，并记录 ΔQ 最大的那个邻居节点，如果 $max\,\Delta Q > 0$，则把节点 i 分配 ΔQ 最大的那个邻居节点所在的社区，否则保持不变；

依据模块度定义，ΔQ 可以化简为：

$$\Delta Q = \left[\frac{\sum in + k_{i,in}}{2m} - \left(\frac{\sum tot + k_{i,in}}{2m}\right)^2\right] - \left[\frac{\sum in}{2m} - \left(\frac{\sum tot}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right]$$

化简可得：

$$\Delta Q = \left[\frac{k_{i,in}}{2m} - \frac{\sum_{tot} k_i}{2m^2}\right]$$

c) 重复 2），直到所有节点的所属社区不再变化；

d) 对图进行压缩，将所有在同一个社区的节点压缩成一个新节点，社区内节点之间的边的权重转化为新节点的环的权重，社区间的边权重转化为新节点间的边权重；

e) 重复 1）直到整个图的模块度不再发生变化。

(2) 基于谱聚类的 SCORE 算法

谱聚类算法流程：

a) 形成网络数据的邻接矩阵 A

b) 计算度矩阵 D 和拉普拉斯矩阵 L = D - A

c) 求 L 的特征值和特征向量

d) 使用 k 个最大特征值的特征向量组成一个矩阵，标准化向量

e) 对 k 维空间中的数据点使用 K-means 算法等进行聚类

SCORE 算法：

SCORE 算法的思路来源于避免度异质性的 DSBM 模型[7]。

a) 形成网络数据的邻接矩阵 A

b) 计算度矩阵 D 和拉普拉斯矩阵 L = D - A

c) 求 L 的特征值和特征向量

d) 使用 k 个最大特征值的特征向量组成一个矩阵，标准化向量

将前 k 个特征向量组成的$n \times k$矩阵变化为$n \times (k-1)$矩阵：

$$R = \left[\frac{\widehat{\xi_2}}{\widehat{\xi_1}}, \frac{\widehat{\xi_3}}{\widehat{\xi_1}} \ldots, \frac{\widehat{\xi_k}}{\widehat{\xi_1}}\right]$$

e) 对矩阵 R 使用 K-means 算法等进行聚类

4、评估指标

对于社区发现的效果评估并没有绝对的指标，常用的指标值与聚类的评估指标有相似之处，要求有高的社区内连接，社区间连接较少。

有真实标签(Ground Truth) 时，常用的评价指标有： NMI 标准互信息，Jaccard 相似度等。此外，也可以使用分类问题的评估指标如错误率、Precision，Recall，TRP，F1-score，Adjusted rand Index (ARI)。

无真实标签是常用的评价指标有：模块度、传导率(Conductance)，网络密度、内部密度等。由于本数据中没有真实标签，于是考虑使用无真实标签的几个评估指标

三、数据分析

1、网络数据的描述性统计

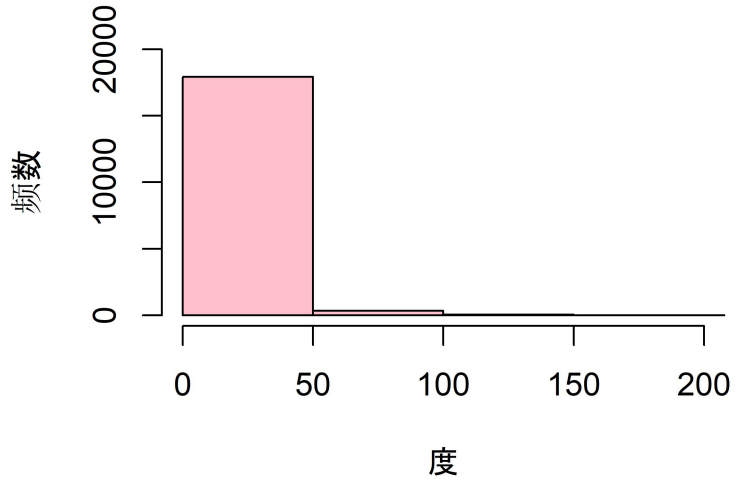
(1) 网络密度

全网络的密度为 0.00067739，即 0.06774%，网络较稀疏。

(2) 节点度

在该主页关系网络中，节点的度表示一个官方页面所关联的不同主页数量。下图为关联网络节点度分布直方图，从图中可以看出节点度分布是严重右偏的，即大部分节点的度很小，存在少部分节点的度很大。这也与社交网络的特点相符合，即存在少部分官方页面关联其它主页数量众多，大部分官方页面用户的社交互动是处于较低水平的，而这些度较大的官方页面也就是主页关系网络中的重要节点。

图 1 节点度的分布图



下表展示了主页关联网络中度最高的 10 个官方页面。排名第一的官方用户主页是 Honolulu District, U.S. Army Corps of Engineers，它的度为 472，代表与其建立关联关系的官媒主页共有 472 个。排名第三的官方页面是 Facebook 自身，对应的度为 364。

我们不难发现，Facebook 也是表 1 中唯一一个非政府机构的官方页面用户，这足以说明其在当今社交媒体的影响力是非常强的。同时，度最高的 10 个官方页面中有 9 个都是政府机构。之所以会出现这一现象，是由政府机构的工作要求所决定的：政府机构在实施某一政策时，需要与社会上各个部门进行沟通与协作。因此，它们的官方页面关联其它主页的数量会相对较多。

表 3 度排名前 10 的官方页面相关信息

节点度	页面名称	页面类型
472	Honolulu District, U.S. Army Corps of Engineers	government
365	Army Training Network (ATN)	government
364	Facebook	company
261	Defense Commissary Agency	government
247	Defense Logistics Agency (DLA)	government
237	National Park Service	government
237	Army Contracting Command, APG-Natick Contracting Division	government
226	U.S. Coast Guard	government
224	PEO Soldier	government

(3) 节点中心性

1) 接近中心性

接近中心性可以反映一个节点与其他节点是否接近，寻找社交网络中具有影响力的结点，从而分析官方账号的“意见领袖”。接近中心性高的用户与许多其他账号都有密切关联，是社交网络中的“中心”。下表展示了接近中心性最高的前 10 个主页。我们发现，与节点度不同，接近中心性排名高的页面组成类型很多元，不拘一格。其中，排名第一的官方页面是政客 André Fufuca 的主页，其接近中心性为 0.324。

表 4 接近中心性排名前 10 的官方页面相关信息

ID	接近中心性	页面名称	页面类型
704	0.3241578	André Fufuca	politician
18248	0.3174753	Shimano-MTB	company
17317	0.3174126	Queensland Corrective Services	government
10477	0.3164380	Catiuscia Marini	politician
17918	0.3035572	American Chopper	tvshow
16187	0.3020798	Roger Wicker	politician
15907	0.3020270	USEmbassyCyprus	government
15417	0.2989012	Embassy of Switzerland in Jordan	government
14483	0.2977525	Rathika Sitsabaiesan	politician
8190	0.2974687	GIORDANO	company

2) 介数中心性

更进一步地，我们使用介数中心性指标来识别主页关系网络中有影响力的官方用户。介数中心性可以反映一个节点在其他节点之间起"桥梁"作用的程度。如果某个官方页面的介数中心性较高，那么在该社交网络中它对信息传播的控制力就越强。下表展示了介数中心性最高的前 10 个主页。我们发现结果与度前 10 的官方页面差别很大：政治人物以及企业的官方主页占据了榜单的大部分席位。其中，排名第一的官方页面是政客 André Fufuca 的主页，其介数中心性为 0.115。

表 5 介数中心性排名前 10 的官方页面相关信息

ID	介数中心性	页面名称	页面类型
704	0.11578961	André Fufuca	politician
10477	0.08962833	Catiuscia Marini	politician
18248	0.03981995	Shimano-MTB	company
17317	0.03980519	Queensland Corrective Services	government
13626	0.02595360	Andrea Lindholz	politician
16187	0.02269658	Roger Wicker	politician
8190	0.01955732	GIORDANO	company
17693	0.01930825	Marni	company
17918	0.01764088	American Chopper	tvshow
15907	0.01545601	USEmbassyCyprus	government

(4) 网络凝聚性特征

许多现实世界的网络的巨型组件都发现了称为小世界的著名特征。它指的是节点间最短路径的长度通常很小，但网络聚集性很高。常用的描述凝聚性指标有

1）平均路径长度：表示节点之间的最短路径的平均值。

2）最长路径：表现网络节点之间最远的距离，即网的连接性。

3）网络聚集性体现了三元组连接的存在。

在这个 Facebook 社交网络中，可以看到网络中的**平均路径长度**仅为 4.97，并且网络中**最长路径**为 15,因此网络中最短路径长度的尺度小于 log|V|，可以被认为很小。同时，**网络的聚集性**为 0.2323,网络的聚集性较大，即大约 25%的连通三元组闭合形成了三角形。

2、官方主页社交网络社区发现

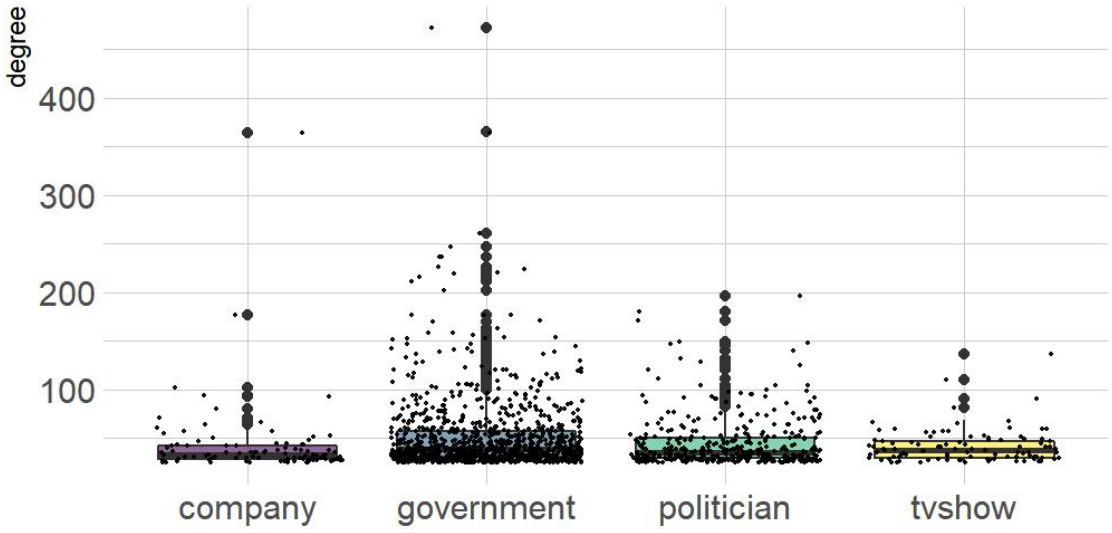
(1) 提取核心网络

为了更加细致地展示 Facebook 上官方页面的社区结构，本报告采用如下的提取方法得到了官方页面关联网络的核心网络：通过不断删除网络中度小于 25 的节点直至网络不再变化，最终得到一个由 1443 个节点，39586 条边构成的主页核心关系网络。该网络的密度是 3.81%。

下图展示了核心网络中四类官方页面各自度的分布情况。很显然，主页核心关系网络中的用户主体为政府组织，企业及电视节目的官方主页在核心网络中占比相对较小。同时，核

心网络中的节点度分布也呈现出了明显的右偏态势：大多数节点的度都很小，只有一小部分节点的度较大。

图 2 核心网络关于度的分组箱线图



(2) 算法选择、社区数选择

下面，通过对比模块度大小、网络密度这些社区发现评估指标，选择了最优的算法 SCORE 算法和最优的社区数 13 个。具体数据如下图所示：

图 3 不同算法、社区数取值下的模块度大小

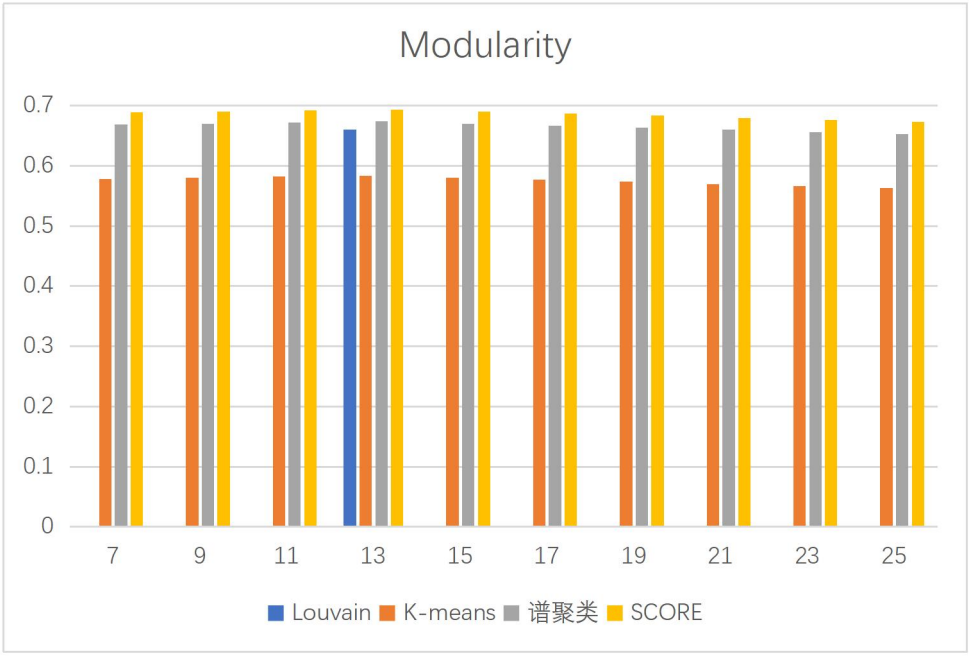
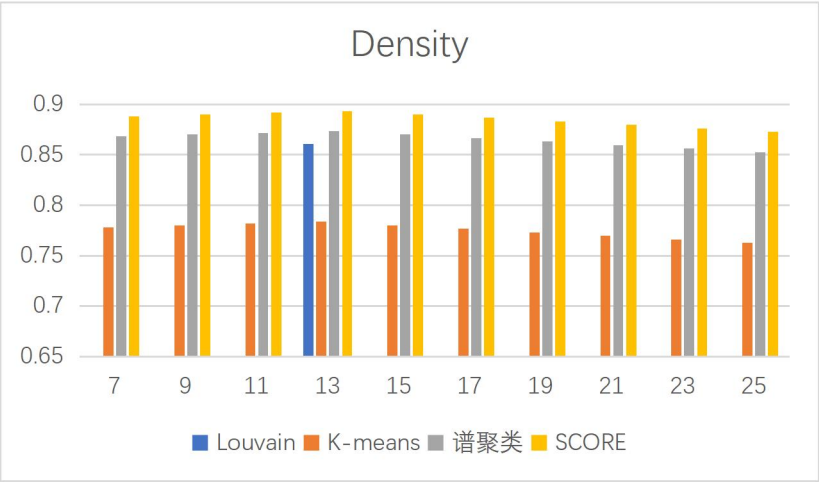


图 4 不同算法、社区数取值下的网络密度大小

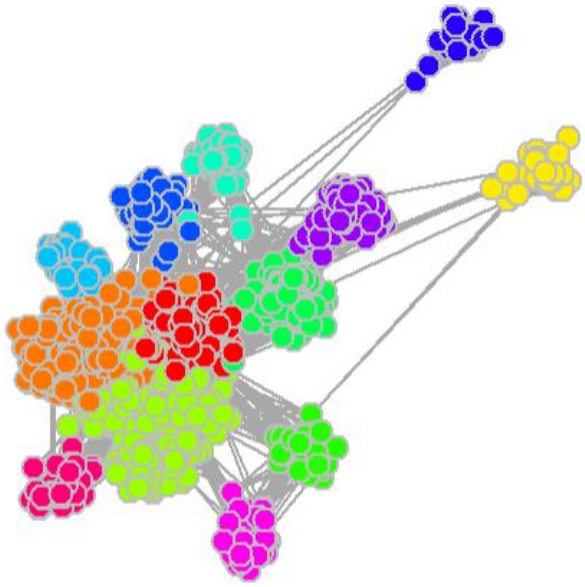


(3) 社区划分

经过算法选择、社区数选择，最终选择使用 SCORE 算法对主页核心关联网络进行社区发现，选择社区数为 13 个。最大的社区有 414 个官方页面，最小的社区涉及 36 个官方用户。本文根据社区大小对社区进行排序，最大社区为社区 2，最小社区为社区 8。

下图展示了该关联核心网络社区结构可视化的结果。其中，不同颜色代表着不同的社区。从图中可以看出节点之间呈现出一定的"社区"结构。社区内部的节点连接相对紧密，而在社区之间，节点的连接则比较稀疏。

图 5 官方主页关联网络核心子图社区划分

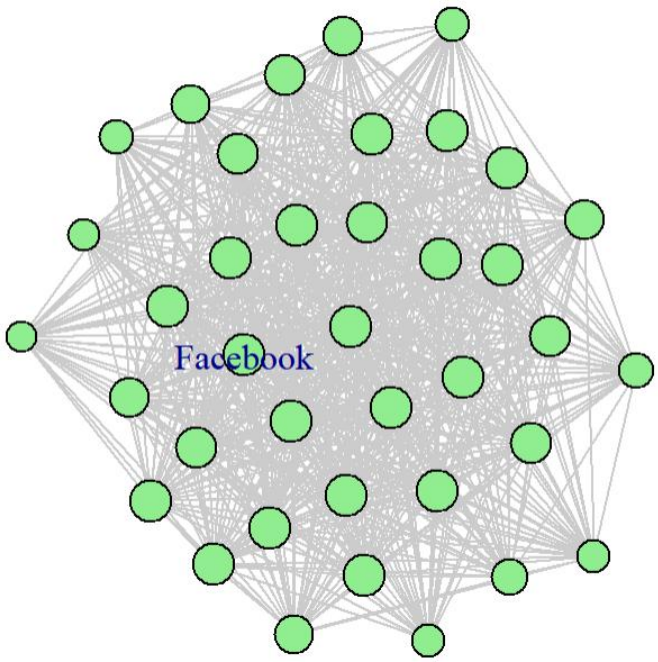


通过分析官方页面关联核心网络的社区结构，我们可以更透彻地把握官方页面间的社交模式，并挖掘出其中相对活跃的官方用户子群体。

3、社区子网络分析

下图为社区 10 的网络结构图，不难发现，在该子网络中各节点的度分布较为均匀，且网络的密度为 94.45%，聚类系数为 95.18%。这说明该社区内部各官方页面用户主体间的社交活动相对频繁，同时，社区群体对官方用户的吸引力也较强。值得注意的是，Facebook 的官方主页正隶属于该社区，而 Facebook 所对应的节点在该社交网络中的度是最大值，为 37（一共 38 个节点）；其介数中心性同样也是最大值，为 0.3326。这些都说明了 Facebook 对于该社区的维系有着举足轻重的作用。

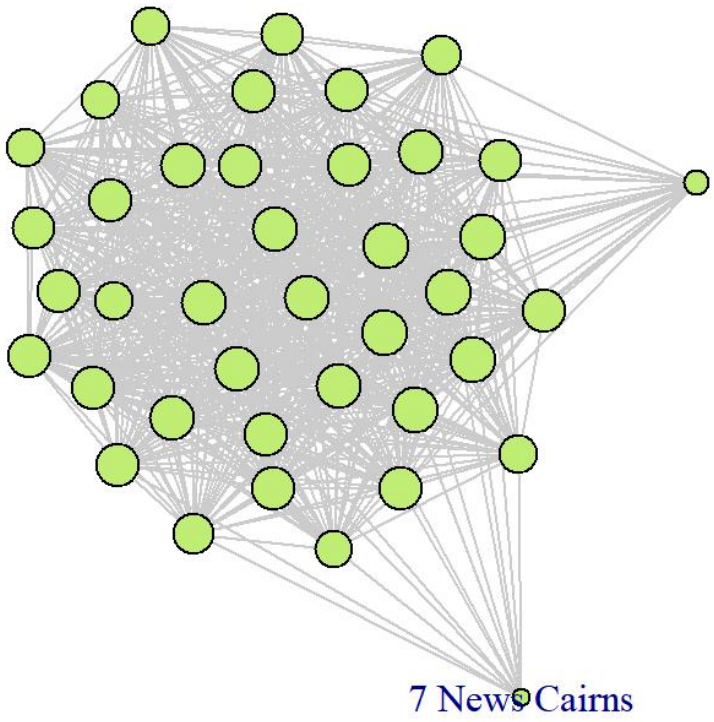
图 6 社区 10 网络结构图（含 Facebook 节点）



下图为社区 13 的网络结构图，子网络图呈现为分布式网络结构。该子网络中度的分布除去 *7 News Cairns* 所对应的节点外，大体上也较为均匀。

网络的密度为 90.3%，聚类系数为 93.3%，都处在较高的水平。在社区 13 的网络中，*7 News Cairns* 是澳洲知名新闻平台 Cairns News 在 Facebook 网站上的官方主页，这一节点的度是该社区中的最小值，为 14；其介数中心性也是最小值，为 0.03125，这些结果说明了该官方页面与社区中其它官方页面间的社交关系相对疏远。

图 7 社区 10 网络结构图（含 7 News Cairns 节点）



四、研究结果与讨论

本报告基于 Facebook 网站上官方页面间的社交网络，运用网络结构数据的分析方法，挖掘出该网站上影响力较大的官方用户。同时，借助 SCORE 算法对提取出的核心网络进行社区发现，共划分出 13 个子网络。其中，报告选择了结构较为简单的社区 10 和社区 13 进行了相应的分析及可视化。

由于精力所限，本研究还有可以后续深入的部分。例如，构造有权网络，研究各个官方页面间的联系强度。此外，还可以考虑收集 2017 年其它月份 Facebook 网站上的官方页面数据进行探究，从而得出更加深入、丰富的结论。

参考文献

[1] B. Rozemberczki, C. Allen and R. Sarkar. Multi-scale Attributed Node Embedding[J]. The Journal of Complex Networks, 2019.

[2] Blondel V D, Guillaume J L, Lambiotte R, et al. Fast unfolding of communities in large networks[J]. Journal of Statistical Mechanics Theory & Experiment, 2008.

[3] Jin D, Yu Z, Jiao P, et al. A survey of community detection approaches: from statistical modeling to deep learning[J]. IEEE Transactions on Knowledge and Data Engineering, 2021, 4(2): 1-15.

[4] Su X, Xue S, Liu F, et al. A comprehensive survey on community detection with deep learning[J]. IEEE Transactions on Neural Networks and Learning Systems, 2022: 1-21.

[5] 杜楠. 复杂网络中社区结构发现算法研究及建模[D]. 北京邮电大学博士学位论文, 2009.

[6] Bickel P J, Chen A. A nonparametric view of network models and Newman–Girvan and other modularities[J]. Proceedings of the National Academy of Sciences, 2009, 106(50): 21068-21073.

[7] Holland P W, Laskey K B, Leinhardt S. Stochastic blockmodels: First steps[J]. Social Networks, 1983, 5(2): 109-137.

[8] Jin J. Fast community detection by SCORE[J]. The Annals of Statistics, 2015, 43(1): 57-89.

[9] Rohe K, Chatterjee S, Yu B. Spectral clustering and the high-dimensional stochastic blockmodel[J]. The Annals of Statistics, 2011, 39(4): 1878-1915.

[10] 高惠璇. 应用多元统计分析[M]. 北京: 北京大学出版社, 2004.

[11] 潘蕊，张妍，高天辰. 网络结构数据分析与应用[M]. 北京：北京大学出版社，2023.

附件

附件 1 数据来源网站

http://snap.stanford.edu/data/facebook-large-page-page-network.html

附件 2 程序代码

【算法比较部分】

```
dolphin<read.graph(file='https://lipn.univ-paris13.fr/~kanawati/datasets/dolphins.gml', format='gml')
karate<-read.graph(file='https://lipn.univ-paris13.fr/~kanawati/datasets/karate.gml', format='gml')
football<-read.graph(file='https://lipn.univ-paris13.fr/~kanawati/datasets/football.gml', format='gml')
polblogs<-read.graph(file='https://lipn.univ-paris13.fr/~kanawati/datasets/polblogs.gml', format='gml')
simmons<-read.graph(file='https://lipn.univ-paris13.fr/~kanawati/datasets/simmons.gml', format='gml')
weblogs<-read.graph(file='https://lipn.univ-paris13.fr/~kanawati/datasets/polblogs.gml', format='gml')



k1 = cluster_walktrap(karate)
k2 = cluster_infomap(karate)
k3 = cluster_edge_betweenness(karate)
k4 = cluster_louvain(karate)

d1 = cluster_walktrap(dolphin)
d2 = cluster_infomap(dolphin)
d3 = cluster_edge_betweenness(dolphin)
d4 = cluster_louvain(dolphin)
```

```
f1 = cluster_walktrap(football)
f2 = cluster_infomap(football)
f3 = cluster_edge_betweenness(football)
f4 = cluster_louvain(football)

compare(k1,V(karate)$value,method = "nmi")
## [1] 0.504178
compare(k2,V(karate)$value,method = "nmi")
## [1] 0.6994882
compare(k3,V(karate)$value,method = "nmi")
## [1] 0.5798278
compare(k4,V(karate)$value,method = "nmi")
## [1] 0.5866348

compare(f1,V(football)$value,method = "nmi")
## [1] 0.8873604
compare(f2,V(football)$value,method = "nmi")
## [1] 0.9005465
compare(f3,V(football)$value,method = "nmi")
## [1] 0.8788884
compare(f4,V(football)$value,method = "nmi")
## [1] 0.8549734

compare(d1,V(dolphin)$value,method = "nmi")
## [1] 0.53725
compare(d2,V(dolphin)$value,method = "nmi")
## [1] 0.593211
compare(d3,V(dolphin)$value,method = "nmi")
## [1] 0.5541605
```

```
compare(d4,V(dolphin)$value,method = "nmi")
## [1] 0.5108534
```

【数据分析部分】

```
library(igraph)
library(plyr)
library(dplyr)
library(readxl)

#网络密度
n <- length(V(g))
N <- length(E(g))
density <- N/choose(n, 2)
density

#计算节点的度
degree = ddply(data1, .(id_1), nrow)
#添加列名
colnames(degree) = c('id', 'degree')
#绘制直方图
#png(file='test.png', height=2000, width=2500,bg = "white",res = 600)
hist(degree$degree,ylab = "频数",xlab = "度",main=NULL,col = "pink",xlim =
c(0,200),ylim = c(0,20000),family = 'serif')
#dev.off()

#按度的降序排列
degree = degree[order(degree$degree,decreasing = T),]
degree_top10 <- left_join(degree[1:10,],data2)
```

```
#输出结果
knitr::kable(format(degree_top10[,-1],scientific=F),caption = "表 1 度排名前 10 的公共主页相关信息")

#接近中心性
c<-closeness(g, normalized = T)
close = data.frame(id=data2$id,clo = c)
close = close[order(close$clo,decreasing = T),]
close_top10 <- left_join(close[1:10,],data2)
knitr::kable(format(close_top10,scientific=F),caption ="表 2 接近中心性排名前 10 的公共主页相关信息")

#结点中介中心性
b<-betweenness(g, directed = F, normalized = T)
betweenness = data.frame(id=data2$id,bet = b)
betweenness = betweenness[order(betweenness$bet,decreasing = T),]
betweenness_top10 <- left_join(betweenness[1:10,],data2)
knitr::kable(format(betweenness_top10,scientific=F),caption = "表 2 中介中心性排名前 10 的公共主页相关信息")

#平均路径长度
ave.pa = average.path.length(g)
#最长路径
dia = diameter(g)
#
trans <- transitivity(g)

# 提取核心网络
W <- as_adjacency_matrix(g)
```

```
# 将"dgCMatrix"类型转化为"matrix"类型
W <- as.matrix(W)
# 提取核心网络
converg <- FALSE
old.nrow <- nrow(W)

while(!converg){
  # 计算 W 矩阵的列和
  d <- colSums(W)
  to.keep <- which(d>=25)
  # 保留列和大于等于 25 的列
  if(old.nrow==length(to.keep)){
    converg <- TRUE
  }
  old.nrow <- length(to.keep)
  W <- W[to.keep,to.keep]
}
g_core <- graph_from_adjacency_matrix(W, mode = "undirected",)

# 计算核心网络的节点数
print(paste0('核心网络的节点数为：', vcount(g_core)))
# 计算核心网络的边数
print(paste0('核心网络的边数为：', ecount(g_core)))
n_1 <- length(V(g_core))
N_1 <- length(E(g_core))
density_1 <- N_1/choose(n_1, 2)
density_1
```

```
#核心网络中四类公共主页各自度的分布情况

library(tidyverse)
library(hrbrthemes)
library(viridis)
tmp_data = merge(degree, data2, by = 'id')

tmp_data = tmp_data[order(tmp_data$degree, decreasing = T), ]
data_core = tmp_data[1:max(which(tmp_data$degree >= 25)),]

data3 = data.frame(data_core[,c(2,5)])

data3 %>%
  ggplot( aes(x=page_type, y=degree, fill=page_type)) +
  geom_boxplot() +
  scale_fill_viridis(discrete = TRUE, alpha=0.6) +
  geom_jitter(color="black", size=0.4, alpha=0.9) +
  theme_ipsum() +
  theme(
    legend.position="none",
    plot.title = element_text(size=11,hjust = 0.5)
  ) +
  ggtitle("图 2 核心网络关于度的分组箱线图") +
  xlab("")


# 社区划分
```

```
# 设置随机种子
set.seed(100)
# 使用 multilevel.community 函数对核心网络进行社区划分
com = multilevel.community(g_core)
# 展示每个社区的大小
tb = table(com$membership)[order(table(com$membership),decreasing = TRUE)]


com_color = rainbow(13)
# 为节点添加社区属性
V(g_core)$com = com$membership
g_core = igraph::simplify(g_core, remove.multiple = T,remove.loops = T)
# 不显示标签
V(g_core)$label = ""
# 设置随机种子
set.seed(100)
# 绘制
plot(g_core,
     #main = "图 3 官媒主页互关网络核心子图",
     layout=layout.fruchterman.reingold,
     # 设置节点大小
     vertex.size = 9,
     # 设置节点颜色
     vertex.color=com_color[V(g_core)$com],
     # 设置节点边框颜色
     vertex.frame.color='grey')
legend()



#子图网络分析
```

```
set.seed(1)
#detach("package:dplyr")
# 计算社区 10 网络的密度
g_com7 <- induced_subgraph(g_core, igraph::groups(com)$`10`)
# 提取社区 10 的子网络
g_com7 = igraph::simplify(g_com7, remove.multiple = T,remove.loops = T)
print(paste0('子网络的密度为：', graph.density(g_com7)))
print(paste0('子网络的聚类系数为：', transitivity(g_com7)))

# 绘制社区 10 构成的子网络
V(g_com7)$size = seq(1,14,length.out = max(degree(g_com7)))[degree(g_com7)]
# 节点大小与度成正比
set.seed(1)
par(mfrow=c(1,1), mar=c(1,1,1,1))
V(g_com7)$label = ''
V(g_com7)[V(g_com7)$name=='10395']$label <- 'Facebook'
plot(g_com7,
     #main = "图 4 基于力导向算法布局的社区 10 网络结构图",
     layout = layout.fruchterman.reingold,
     # 绘制力导向布局图
     vertex.label = V(g_com7)$label,
     # 显示节点标签
     vertex.color = 'lightgreen',
     # 设置节点颜色
     edge.color = 'grey80'
     # 设置边的颜色
)
```

```
set.seed(1)
#detach("package:dplyr")
# 计算社区 13 网络的密度
g_com8 <- induced_subgraph(g_core, igraph::groups(com)$`13`)
# 提取社区 13 构成的子网络
g_com8 = igraph::simplify(g_com8, remove.multiple = T,remove.loops = T)
print(paste0('子网络的密度为：', graph.density(g_com8)))
print(paste0('子网络的聚类系数为：', transitivity(g_com8)))

# 绘制社区 8 构成的子网络
V(g_com8)$size  =  seq(1,14,length.out  =  max(degree(g_com8)))[degree(g_com8)]
# 节点大小与度成正比
set.seed(1)
# 设置随机种子，保证同一种布局画出来的图可以重复
par(mfrow=c(1,1), mar=c(1,1,1,1))
V(g_com8)$label = ''
V(g_com8)[V(g_com8)$name=='701']$label <- '7 News Cairns'

plot(g_com8,
      #main = "图 5 基于力导向算法布局的社区 13 网络结构图",
     layout = layout.fruchterman.reingold,
     # 绘制力导向布局图
     vertex.label = V(g_com8)$label,
     # 显示节点标签
     vertex.color = '#C0EE75',
     # 设置节点颜色
     edge.color = 'grey80'
    # 设置边的颜色
```