Large-Scale Principal Component Analysis on LiveJournal Friends Network[*]

Miklós Kurucz   András A. Benczúr   Attila Pereszlényi
Data Mining and Web search Research Group, Informatics Laboratory
Computer and Automation Research Institute of the Hungarian Academy of Sciences
{realace, benczur, peresz}@ilab.sztaki.hu

ABSTRACT

Principal Component Analysis (PCA) is a general means of unsupervised exploration that can be used to find basic motives and organizational themes, the guidance in friends network formation. The applications of PCA include Kleinberg's ranking algorithm as well as spectral graph partitioning. We extend the applicability of PCA to very large scale social networks by handling the abundance of small size communities that hide the higher level structure. Strongest communities, that are still small themselves, take over the first principal axes and the analysis leaves a giant mass in the all-zeroes coordinate. In a combination of heuristics that involve the removal of community cores as well as the contraction of tentacles we are able to find meaningful high level components that characterize countries, regions, age or interest in polarized topics. Our experiments are run on a 3.5M user snapshot of the LiveJournal Friends network where our algorithm outperforms all previous methods for power law graph partitioning both in speed and cluster quality. In particular, our heuristics promise similar or better performance than semidefinite relaxation in much shorter running time.

Categories and Subject Descriptors

J.4 [**Computer Applications**]: Social and Behavioral Sciences; G.2.2 [**Discrete Mathematics**]: Graph Theory—*Graph algorithms*; G.1.3 [**Mathematics of Computing**]: Numerical Analysis—*Numerical Linear Algebra*

Keywords

Communities, Principal Component Analysis, Singular Value Decomposition, Spectral Clustering

1. INTRODUCTION

In this paper we concentrate on the technical challenges and the possible power of Principal Component Analysis (PCA) of large scale social networks. PCA appears in HITS ranking [27], in spectral clustering [17] as well as used directly by e.g. in [37] who show how certain topics such as conservatism or humor appears along principal axes for a very small scale post network.

We extend the applicability of PCA to the social network of bloggers by introducing heuristics that make PCA scalable by preventing low level communities from overtaking the first principal axes. Our methods make it possible to interpret PCA results for top-level properties of the large scale blogger networks analyzed by several authors [29, and references therein]; in particular for the LiveJournal Friends network defined by the friend lists of users, a network examined by Backstrom et al. [5], Kumar et al. [30] and many others.

Our method is based on the combination of the removal of Tightly Knit Communities (TKC) [33] and the contraction of long tentacles [32]. We build on the recent findings of Xu et al. [45] who identify bridges across TKCs as the main reason for the failure of graph partitioning methods. While their method is based on the identification of community cores as a partitioning of the network, similar to several other core finder methods [18, 19] the cores identified are of small size on the global scale and cannot yield information on the global structure. These methods however can be used prior to PCA to remove a large number of cores that act as TKCs by attracting a large number of principal vectors. When combining core removal and tentacle contraction, we obtain high level distinctive characteristics of the friends network that include geography, religion and age.

Our experiments are performed on the LiveJournal Friends of more than three million users. Recently several results appeared on the structure of the post network [36] that are able to detect interesting phenomenon on a smaller scale such as the spread of certain information in the network. In contrast large scale methods are required to mine the latent information within the friends network; however this network is more robust in time and our methods give access to patterns persistent on longer time scale.

1.1 Applications of PCA

PCA is a basic ingredient of social network mining with applications to ranking and clustering that can be further deployed in viral marketing [15, 41], in user segmentation by selecting communities with desired or undesired properties as e.g. interest in certain topics or high recent churn rate, or in detecting factors that keep bloggers active [34]. In particular the Friends list of a blog can be used for social filtering, i.e. reading posts that their friends write or recently read [35].

Principal Component Analysis is also related to the HITS ranking algorithm [27]; in fact the hub and authority ranking is defined by the first left and right singular vectors and the use of higher dimensions is suggested already by [27] and analyzed in detail in

[*]Support from grants OTKA NK 72845 and *ASTOR* NKFP 2/004/05

[40]. Several authors use HITS for measuring authority in mailing lists [48] or blogs [28], the latter result observing a strong correlation of HITS score and degree, indicating that the first principal axis will contain no high-level information but simply order by number of friends.

We demonstrate that HITS-style ranking can be used but with special care due to the Tightly Knit Community (TKC) effect that result in communities that are small on a global level grabbing the first (or, as we show, even the first many) principal axes. Lempel et al. [33] are the probably the first who identify the TKC problem in the HITS algorithm, their algorithmic solution (SALSA) however turns out to merely compute in- and out-degrees [8]. In contrast we keep PCA as the underlying matrix method and filter the relevant high-level structural information by removing TKCs and concentrating the network by contracting long tentacles.

Principal Component Analysis is also the main ingredient of spectral clustering, a technique we demonstrate to be desirable for clustering large social networks. Prior to our work, spectral clustering was known to fail for large power law graphs with several partly successful attempts [32, 31]. While spectral methods are key in top-down clustering, as a different possibility agglomerative strategies are used for bottom-up clustering [3]. These latter methods are however known to be instable [22], in particular for the blogger network where small communities are in abundance while the interpretation of a next layer of supercommunities over communities is missing. We show that the top-down approach is probably the right choice to analyze very large scale social networks.

The applicability of spectral methods to graph partitioning is observed in the early 70's [17, 16]. The methods are then rediscovered for netlist partitioning, an area related to circuit design, in the early 90's [12, 2, 4, 3] and a large number of results appeared in the "Spectral Clustering Golden Age" [14; 38; 47, etc] 2001.

Prior to our work, the only known large scale formation of the LiveJournal blogger network was the Russian user group [21, 46]. By our methods we reveal clusters arranged by location, age and certain types of interest such as religion.

Our network forms the largest power law graph attacked by unsupervised methods with (after cleansing) 2.3M nodes and over 14M bidirectional edges. This graph is larger by orders of magnitude compared to earlier experimentation on social networks [20, **?**] where hierarchical community structures or even the graphs themselves could easily be visualized. The largest graph partitioning benchmark has only 448K nodes and 3.3M edges; Kevin Lang [32] considered the Yahoo IM graph with less than 10M edges.

2. COMPONENTS OF THE ALGORITHM

We describe our combination of heuristics to filter out the globally relevant network structure prior to PCA. In the presentation we choose spectral clustering as the main application example that we describe in detail next. The pre-filtering heuristics (Sections 2.1 and 2.2) are however applicable in general before PCA to obtain globally meaningful principal axes as we will describe in our experiments in Section 4.

Principal component analysis is used in graph bisection heuristics and more generally in spectral clustering, a set of a heuristic algorithms all based on the overall idea of projecting onto the first few principal directions and then clustering in a low (in certain cases simply one [17]) dimensional subspace.

Graph bisection is transformed into a PCA problem via the standard Quadratic Integer Program

$$1/4x^T Lx$$

where L is the graph Laplacian and x is the ± 1 cut indicator vector. In order to avoid the trivial cut with all nodes on one side, we have $x^T e = 1$ where e is a vector of all ones. When relaxing x to arbitrary real values between -1 and +1, the optimum is known to be the second eigenvector (the Fiedler vector) of L [17]. When however we relax indicator values to be arbitrary norm 1 n-dimensional vectors, the resulting optimization problem can be solved by semidefinite programming [32].

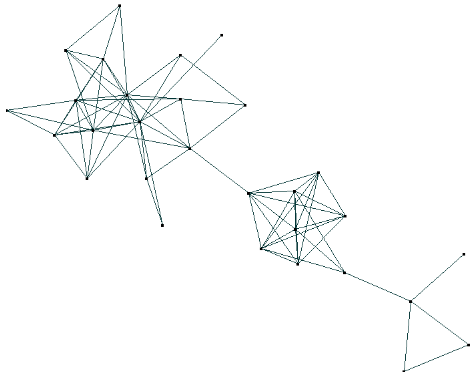


Figure 1: A 82-node subgraph of the LiveJournal Friends network, with two cores and several short tentacles.

In our experiments we use hierarchical spectral clustering algorithms that project the graph into a d-dimensional vector space [12] and divide it into more than two parts by the k-means clustering algorithm in one step, as suggested first by [47]. In order to obtain the projection we test both the SDP relaxation in d dimensions as well as the first d singular vectors. As suggested by [42, 14], for matrix L above we use the *weighted Laplacian* $L = D^{-1/2}AD^{-1/2}$ where A is the adjacency matrix and D is the diagonal matrix where the i-th entry is the total edge weight at node i. By using the weighted Laplacian we may produce better quality partitioning [31].

The two main ingredients of our algorithm consist of the removal of small dense regions and contracting long interconnecting tentacles. In Fig. 1 we see typical subgraphs of the entire network that consist of several small community cores, two of which is seen, with low degree nodes loosely connected to some of them or interconnecting pairs of them. Since PCA is unable to select from the abundance of small cores, it falls into the trap of the so-called Tightly Knit Community (TKC) effect [33] by selecting the most dominant such structure that is still very small on the scale of the entire network. We demonstrate that after the proposed preprocessing these traps are avoided and meaningful principal axes are found.

As an initial observation we show how different projection methods on the Russian cluster of the LiveJournal friends network may or may not distinguish between Russians and other nations within the Russian cluster (Ukraine, Belarus, Estonia etc.). In [31] it is observed that that direct spectral partitioning of this cluster is impossible due to singular value sequence such that even the 100th largest one is above 0.99. In accordance, the principal axes for direct SVD are non-characteristic in Fig. 4. The distinction becomes however strongly visible by using our pre-processing algorithm. Finally on the right of Fig. 4 we also see why the semidefinite relaxation outperforms SVD: since it projects nodes on a unit ball, most of the time a balanced partitioning may be constructed although it is not necessarily always the right choice. In our example locations other than Russia tend to shift to the upper left part of the projection although they strongly mix near the central dense diagonal hyperplane.

2.1 Tentacles and small component heuristics

Algorithm 1 Tentacle contraction.
input: $d_{\max}$: maximum degree of a tentacle node.
output: graph G' with all tentacles contracted.

```
while node $v$ of degree $\leq d_{\max}$ exists in $G'$ do
    Contract $v$ to its neighbor $u$ with lowest degree in $G'$
    Record $v \rightarrow u$ for tentacle set reconstruction
```

We use two heuristics for handling tentacles, one for pre and another for post-processing. The post-processing is identical to the one discussed in [31]: we test the resulting partition for small clusters and try to redistribute nodes to make each component connected. Pre-processing consists of recursively contracting all nodes that have degree below a threshold into a neighbor. In this way tentacles are eliminated and close communities are moved in proximity of each other.

In a recursive definition we say that a node belongs to a *tentacle* if its degree is not more than a prescribed value $d_{\max}$; we use $d_{\max} = 3$. As long as there are tentacle nodes in the graph, we contract them into (one of) their neighbors with smallest degree. In this way we may create new small degree nodes; the procedure may recursively continue. By recording the contractions we may also reconstruct all nodes that get contracted into a final node; such a set of nodes is called a tentacle. The procedure is described in Algorithm 1. We note that the definition of a tentacle depends on the order of contractions and hence we only use it as a preprocessing heuristic and do not use tentacles for characterizing a particular node.

Algorithm 2 `redistribute`$(C_1, \ldots, C_k)$: Small cluster redistribution

```
for all $C_i$ do
    $C'_i \leftarrow$ largest connected component of $C_i$
    if $|C'_i| <$ limit $\cdot |C_1 \cup \ldots \cup C_k|$ then
        $C'_i \leftarrow \emptyset$
Outlier $= (C_1 - C'_1) \cup \ldots \cup (C_k - C'_k)$
for all $v \in$ Outlier do
    $p(v) \leftarrow j$ with largest total edge weight $d(v, C'_j)$
for all $v \in$ Outlier do
    Move $v$ to new cluster $C_{p(v)}$
return all nonempty $C_i$
```

In addition to the tentacle removal preprocessing, in Algorithm 2 we also give a postprocessing subroutine to reject very uneven splits identical to that of [31]. Given a split of a cluster (that may be the entire graph) into at least two clusters $C_1 \cup \ldots \cup C_k$, we first form the connected components of each C_i and select the largest C'_i. We consider vertices in $C_i - C'_i$ outliers. In addition we impose a relative threshold `limit` and consider the entire C_i outlier if C'_i is below limit.

Next we redistribute outliers and check if the resulting clustering is sensible. In one step we schedule a single vertex v to component C_j with $d(v, C_j)$ maximum where $d(A, B)$ denotes the number of edges with one end in A and another in B. Scheduled vertices are moved into their clusters at the end so that the output is independent of the order vertices v are processed. By this procedure we may be left with less than k components; we will have to reject clustering if we are left with the entire input as a single cluster. In this case we either try splitting it with modified parameters or completely give up forming subclusters.

2.2 Tightly knit communities and the SCAN algorithm

The second main ingredient of our algorithm consists of the removal of community cores seen in Fig. 1 or, in another terminology, tightly knit communities (TKC) before singular value decomposition. Several authors observe difficulties caused by the TKCs: Lempel and Moran [33] investigate hyperlink based ranking on the Web; very recently [45] identifies hubs that bridge between several TKCs as the main difficulty in network partitioning.

Several algorithms are proposed to identify community cores. Flake et al. use network flows [18] or min-cut trees [19]; Xu et al. [45] uses an agglomerative method that prefers core nodes and avoids bridges that connect more than one TKC. All these methods however suffer from the abundance of very small communities with no superimposed larger scale structure that network flow based heuristics could exploit.

Our heuristic solution is based on the Structural Clustering Algorithm for Networks (SCAN) algorithm [45]; however instead of using moderate parameters to build large clusters directly as community cores, we use SCAN with restrictive values and remove 1–5% of the nodes that belong to TKC prior to PCA.

The assumption of Xu et al. [45] is that hub vertices bridge many clusters. Therefore they define the SCAN algorithm that selects pairs of vertices with a concentration of common neighbors as candidate intra-cluster nodes limited by parameter ϵ. Hubs, as opposed to intra-cluster nodes, are then characterized by the distraction of neighbors. Finally cores are formed by nodes that have at least μ neighbors within the core.

The key step in the SCAN algorithm is the selection of edges between pairs of nodes whose neighborhood similarity is above a threshold ϵ. In the original algorithm of Xu et al. [45], with $\Gamma(u)$ denoting the neighbors of u, the similarity is measured as

$$\sigma(u, v) = |\Gamma(u) \cap \Gamma(v)| / \sqrt{|\Gamma(u)||\Gamma(v)|}.$$

For power law graphs, in particular for the Web graph in our experiments, however the running time for computing $\sigma(u, v)$ is very large due to the huge neighborhood sets $\Gamma(u)$ involved. Hence we use the Jaccard similarity

$$\text{Jac}(u, v) = |\Gamma(u) \cap \Gamma(v)| / |\Gamma(u) \cup \Gamma(v)|$$

that we approximate by 100 min-hash fingerprints [10].

The modified SCAN Algorithm 3 proceeds as follows. First it discards edges that connect pairs of dissimilar nodes below threshold ϵ; these edges may bridge different dense regions [45]. Then nodes with more than μ remaining edges are considered as community cores; we use $\mu = 4$ in our experiments. Finally connected components of cores along remaining edges augmented by neighboring non-core nodes. The resulting components $\mathcal{C}$ may overlap at these augmented vertices that are considered hubs in [45].

Our main Algorithm 4 combines the previous three heuristics. First community cores $Q_1, \ldots, Q_s$ are identified by the SCAN algorithm and discarded from the graph. Then tentacles are contracted prior to the actual SVD procedure. SVD is performed on the normalized Laplacian. The singular vectors are normalized before the actual partitioning by k-means. The SVD and normalization steps can also be replaced by solving the semidefinite relaxation. Finally as the last heuristic we feed all k-means clusters and SCAN

Algorithm 3 Modified SCAN.
input: ϵ: similarity threshold of neighbors within same core, μ: size threshold of neighborhood within core
output: list of communities

```
for all edges $uv$ do
    compute approximate Jac$(u, v)$ by min-hash fingerprints
$E' \leftarrow \{(uv) : \sigma(u, v) \geq \epsilon\}$
$V' \leftarrow \{u : \deg_{E'}(u) \geq \mu\}$
compute the connected components $\mathcal{C}$ of $V'$ with edges $E'$
for all components $C$ of $\mathcal{C}$ do
    Add all vertices to $C$ that are connected to $C$ by edges of $E'$
return $\mathcal{C}$
```

Algorithm 4 Spectral Clustering.
input: k: desired branching factor of the cluster hierarchy.
output: hierarchical clustering

```
while desired number or cluster size is not reached do
    Select largest cluster $C_0$ and induced subgraph $G$
    $Q_1, \ldots, Q_s \leftarrow$ cores given by SCAN$(G, \epsilon, \mu)$
    $G' \leftarrow G - \bigcup Q_i$
    $G'' \leftarrow$ Contract all tentacles in $G'$
    $A \leftarrow$ adjacency matrix of $G''$
    Project $D^{-1/2}AD^{-1/2}$ into first $d$ eigenvectors
    For each node $i$ form vector $v'_i \in R^d$ of the projection
    $v_i \leftarrow v'_i/||v'_i||$
    $(C_1, \ldots, C_k) \leftarrow$output of $k$-means$(v_1, \ldots, v_{|C_0|})$
    Call redistribute$(C_1, \ldots, C_k, Q_1, \ldots, Q_s)$
    Discard $C_0$ if $C_0$ remains a single cluster
return all discarded and remaining clusters
```

cores to the small component redistribution procedure that merges the SCAN cores into the final components.

For SVD we use the Lanczos code of `svdpack` [6] and for SDP we use Burer and Monteiro's solver [11].

3. DATA SET

For our experiments we use the LiveJournal friends network downloaded in a two-week period of November 2007[1]. The total number of users is 3,583,332 with 44,913,072 directed edges, out of which 14,286,827M is reciprocal. In contrast, the data set of Backstrom et al. [5] has 4.2M users with no major reason for difference between the two collections. By manual analysis we observed certain users missing due to timeouts, some users renamed, also some friends changed. The union of the two collections has 4,720,668 users, less than 28% of the 14 million listed by LiveJournal as of November 2007.

Since we downloaded the Friends network starting from a single user, our collection consists of a giant strongly connected component (SCC) as well as nodes reachable from the SCC (OUT). The collection of [5] is started from community listings, hence the union of the two data sets partly reveal the bow-tie structure of Fig. 3 with 197,325 nodes not reachable from SCC but from which SCC can be reached (IN), and 31,157 users either disconnected (ISLAND) or reachable from IN or reach OUT but not in IN or OUT (TUNNEL). The number of strongly connected components is 768351 with a single giant one, leaving tiny pieces for others. The bow tie structure observed first for Web pages by [9], then by [48] for mailing lists is depicted in Fig. 3; the relation of the size of the strongly and weakly connected component of the post network is also described by [43].

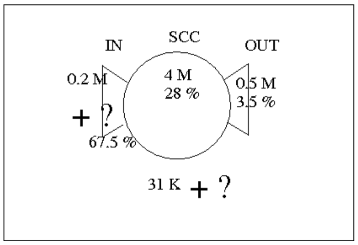


Figure 3: Left: Right:The Bow Tie structure of the LiveJournal friends network.

In our analysis below we rely solely on our crawl since no user data is collected by [5]. We keep only bidirectional edges; this procedure leaves us with a giant component with 2,379,267 nodes and 14,286,827 reciprocal edges. Since PCA requires a connected graph, we discard all other nodes.

LiveJournal user metadata is provided via the XML interface as seen in a sample example in Fig. 2. The available metadata and the percentage of users who provide the information is summarized in Table 1.

Clustering follows a distribution of the combination of location, interest and (least significantly) of age [29] with 29% friendship remaining unexplained by these three factors.

Distribution of countries is analyzed by several authors [29, 23, 24]. Geographic distribution for Spaces and Blogspot bloggers [23] differ. More detailed analysis, including language detection, in [24].

Bhagat et al. [7] observe age as a characteristic feature of blog linkage; unlike for other blogs however they see stable linkage of LiveJournal bloggers across ages. Similar to their method, we cut off the 12% users of age above 30 and below 16.

4. EXPERIMENTS

We measure clustering quality by the entropy and purity of geographic location or other external property within the cluster. By using the notation of the previous subsection let $N_{i,k}$ denote the cluster confusion matrix, the number of elements in cluster k with attribute i and let $p_{i,k} = N_{i,k}/N_k$ denote the ratio within the cluster. Let there be m clusters; then the *entropy* E and *purity* P [26] (the latter also called *accuracy* in [13]) are defined as

$$E = (-1/\log m)\sum_k (N_k/N)\sum_i p_{i,k}\log p_{i,k} \quad \text{and}$$

$$P = \frac{1}{N}\sum_k \max_i N_{i,k},$$

where the former is the average entropy of the distribution of the attribute (e.g. country or age group) within the cluster while the latter measures the ratio of the "best fit" within each cluster.

In addition we use graph-only quality measures as well, all based on the number of edges inside and across clusters. First we considered *modularity*, a measure known to suit social networks well [45] defined as follows:

$$Q = \sum_{\text{clusters } s}\left[\frac{|E(C_s, C_s)|}{|E|} - \left(\frac{|E(C_s, \overline{C_s})|}{2|E|}\right)^2\right]. \qquad (1)$$

where E is the set of all edges and $E(X, Y)$ is the set of edges with tail in X and head in Y.

[1] Available for research purposes upon request from the second author, `benczur@sztaki.hu`

```
<foaf:Person>
  <foaf:nick>sample_nick</foaf:nick>
  <foaf:name>Nick Sample</foaf:name>
  <foaf:openid rdf:resource="http://sample_nick.livejournal.com/" />
  <ya:country dc:title="US" rdf:resource="http://www.livejournal.com/directory.bml?opt_sort=ut&amp;s_l
  <ya:city dc:title="chicago" rdf:resource="http://www.livejournal.com/directory.bml?opt_sort=ut&amp;s
  <ya:blogActivity>
    <ya:Posts>
      <ya:feed rdf:resource="http://sample_nick.livejournal.com/data/foaf"
                                        dc:type="application/rss+xml" />
      <ya:posted>38</ya:posted>
    </ya:Posts>
  </ya:blogActivity>
  <foaf:weblog rdf:resource="http://sample_nick.livejournal.com/"/>
  <foaf:knows>
    <foaf:Person>
      <foaf:nick>sample_nicks_friend1</foaf:nick>
      <foaf:member_name>...</foaf:member_name>
      <foaf:tagLine>...</foaf:tagLine>
      <foaf:image>http://userpic.livejournal.com/nnnnnnnn/nnnnnnn</foaf:image>
      <rdfs:seeAlso rdf:resource="http://sample_nicks_friend1.livejournal.com/data/foaf" />
      <foaf:weblog rdf:resource="http://sample_nicks_friend1.livejournal.com/"/>
    </foaf:Person>
  </foaf:knows>
  <foaf:knows>
    <foaf:Person>
      <foaf:nick>sample_nicks_friend2</foaf:nick>
  ...
```

Figure 2: Sample XML output of the LiveJournal interface.

Country	Age	Interest	School
76.03	39.79	62.82	47.31

Table 1: Availability of metadata over the LiveJournal friends network.

location	runtime	entropy	purity	n.mod.	c.ratio
SVD redist only	1980m	0.105	0.812	2339	2
SVD all preproc	150m	0.073	0.853	2561	8
SDP no preproc	1755m	0.111	0.857	272	6
SDP all preproc	675m	0.072	0.854	2537	4

Table 2: The running time, entropy, purity, normalized modularity and cluster ratio over the entire network. Cluster ratio is shown multiplied by 10^6. We test four algorithms: SVD with small component redistribution heuristic only, with all heuristics, semidefinite relaxation (SDP) and SDP with core and tentacle removal.

	Purity	Entropy	Norm. Mod.
	(location)		
No heuristics	0.501557	1.182126	3.556151
Tentacle + SCAN	0.504375	0.634847	11.781545

Table 3: The quality of subpartitioning the first cluster.

Unfortunately this measure is not balanced by the cluster size, so we use *normalized network modularity* [44]:

$$Q_{norm} = \sum_{\text{clusters } s} \frac{N_s}{N} \left[\left(\frac{|E(C_s, C_s)|}{|E|} - \frac{|E(C_s, \overline{C_s})|}{2|E|} \right)^2 \right]. \tag{2}$$

The larger the normalized modularity, the more edges remain within the same cluster and the less connect different clusters.

We also measure *cluster ratio* defined as follows. Let there be N users with N_k of them in cluster k for $k = 1, \ldots, m$. The *cluster ratio* is the number of edges between different clusters divided by $\sum_{i \neq j} N_i \cdot N_j$. Smaller values correspond to better clustering.

Our first observation is that direct spectral partitioning of the non-Russian LiveJournal is impossible due to singular value sequence such that even the 100th largest one is above 0.99 [31]. In accordance, the principal axes are non-characteristic. For example in Fig. 4 we observe no difference between Russians and other nations within the Russian cluster (UA, BY, EE etc.) without preprocessing; this distinction becomes however strongly visible by using our algorithm. In Table 4 we see that 15 dimensions suffice for a balanced partitioning only if all preprocessing heuristics are applied.

The running times and cluster quality measures are summarized in Table 2. The most important observation is the huge running time difference between SDP and SVD with only minor differences in cluster quality. As for the reliability of the measures, since entropy and purity relate the clustering to a ground truth, we may take them more reliable measures as the pure graph based ones. In this sense cluster ratio apparently does not form a reliable comparison probably due to its dependence on the number of clusters. The remaining three measures, although noisy in certain cases, do not show major differences in judgement.

The quality improvement for clustering is justified by observing that our method makes PCA easier while not destroying the essential network properties. When comparing the quality of the partitions given by different algorithms, we observe that SVD with the heuristics works always nearly as good as SDP; in fact for LiveJournal, the hardest instance it outperforms SDP in entropy and

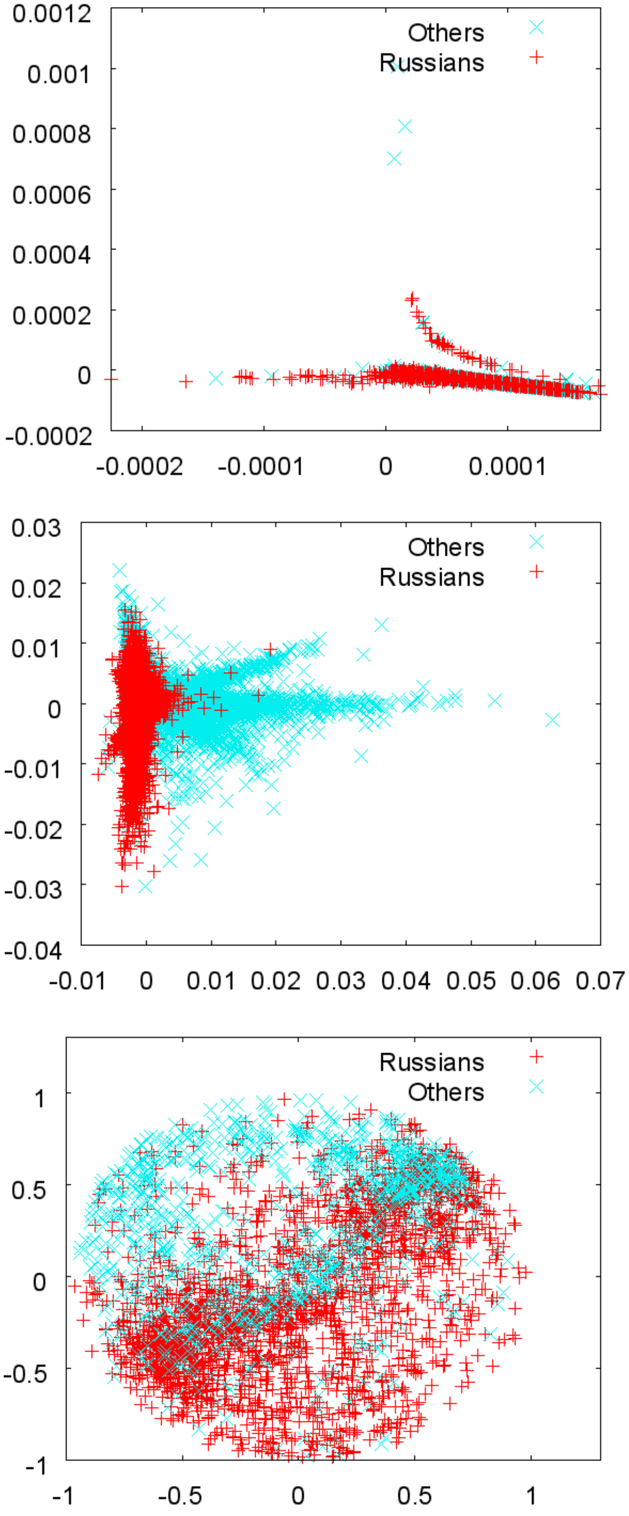
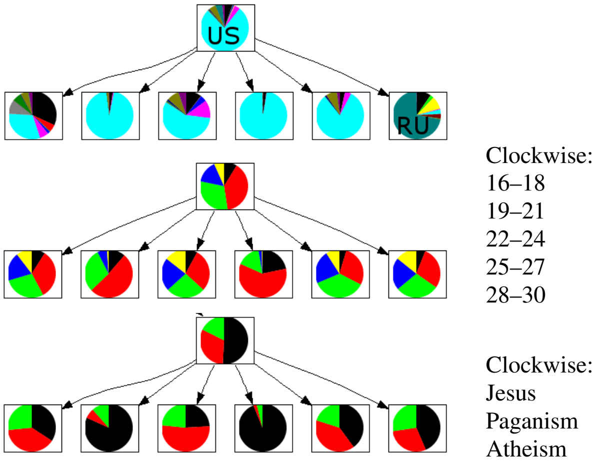


Figure 4: Principal axes 4 and 5 within the Russian cluster before (top) and after (middle) the removal of cores and tentacles as well as two dime nsions of the SDP relaxation solution (bottom).



No	Size	characteristics
1	60 681	International/Europe
2	699 199	US college
3	447 694	International/English speaking + SG, PN
4	170 257	US high school
5	729 604	US Atheist/Pagan
6	216 832	Russian

Figure 5: Partition of LiveJournal users into six, with the distribution of location (top), age (middle) and religious interest (bottom). Characteristics of the parts 1–6 (left to right) are shown in the table.

normalized modularity. The very low normalized modularity of SDP here may indicate an unfortunate split; note that the purity values are very close to .8, the fraction of US location that corresponds to a random split. Here our heuristics greatly improve SDP as well; for the other data sets however SDP performs in general better without them.

The advantages of our method become even clearer when we dig deeper into subclusters. We considered a component (No. 1 in Fig. 5) with 60,681 nodes and 228,644 edges where partitioning is possible even without our heuristics. Notice that in general comparison is not even possible since without the heuristics of Section 2 fails to produce partitions of non-negligible size. The cluster quality parameters are seen in Table 3.

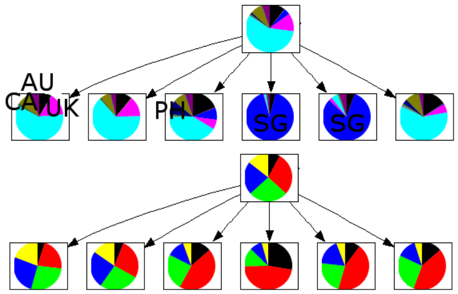


Figure 7: Subpartition of LiveJournal cluster No. 3, with the distribution of location (top) and age (bottom). Dominant location US is distributed into three clusters (#1, 2 and 6) with an age distribution moving from older groups towards a majority 16–18 from subcluster 1 to 6.

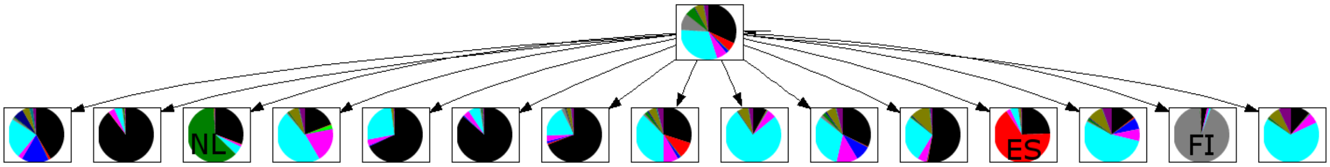


Figure 6: Location distribution of the subpartition of the International/Europe cluster. Black pie denotes locations other than the top 10. Since the parent cluster contains near 1/3 US location, several subclusters (#4, 8, 9, 13 and 15) still have a majority of US users (light pie). Certain countries have characteristic clusters; we marked the largest ones NL, ES and FI.

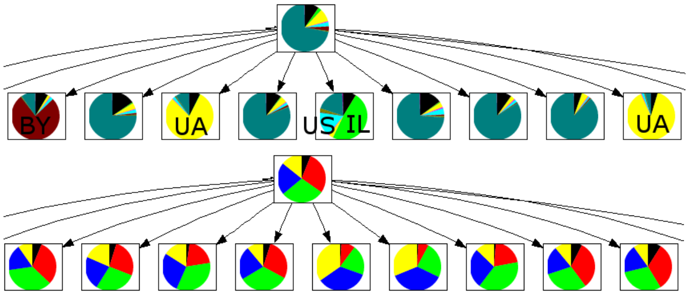


Figure 8: Subpartition of the Russian cluster, with the distribution of location (top) and age (bottom). Primarily Russian clusters (#2, 4, 6 and 8) have characteristic age distribution difference.

plain	tentacles	SCAN	both
0.999	**0.989**	**0.993**	0.987

Table 4: The 15th largest singular value for the choice of the heuristics for tentacle contraction and core removal (SCAN). Figures in boldface denote cases when no balanced partitioning is possible at the first split by Algorithm 2.

4.1 Countries and regions

Characteristic countries are those that, in a 6-level hierarchical clustering into 3000 components, constitute 20% of at least one cluster. Other than US that constitutes near 80% of the users (see also Table 5), these include

- English-speaking clusters: US, CA, UK, AU;
- Russian cluster: RU, BY, UA, EE, LT, IL;
- Non-English and Russian speaking characteristic countries: BR, DE, ES, FI, NL, PH, SG.

In the top-level partitioning (Fig. 5) we see the characteristic Russian cluster [21, 46] as well as two international clusters, one with European connection, the other with mostly English speaking countries. The European cluster splits further by location with NL, ES, FI and several other characteristic countries (Fig. 6). The English-speaking cluster (Fig. 7) consists of UK, CA and AU; in addition they are clustered together with SG and PH. Finally the Russian cluster (Fig. 8) splits again by location with a US-IL partition also dominantly appearing.

Location within US is also known to be important. Unfortunately state information is completely missing from our recent crawl. We tried to reconstruct location within US by considering interest in California, San Francisco, New York, Chicago, Los Angeles or Boston. While we see indications of effect within the US subclusters, results are less indicative due to the lack of a more-or-less complete state information.

Country	Number	% known	% all
US	1 463 654	76.9	40.9
CA	87 609	4.6	2.4
RU	82 801	4.3	2.3
UK	73 789	3.8	2.1
AU	32 508	1.7	0.9
SG	14 986	0.7	0.4
DE	11 329	0.6	0.3
PH	10 380	0.5	0.3
UA	10 260	0.5	0.3
JP	7 778	0.4	0.2
FI	7 104	0.4	0.2
NL	5 970	0.3	0.2
NZ	4 958	0.3	0.1
FR	3 747	0.2	0.1

Table 5: Top list of country location.

As noted in [28], correspondence across boundaries has a high value. We observe tight contact across English-speaking countries (US, UK, CA, AU) and within Europe while the US-only clusters predominantly consisting of high school or college age people.

4.2 Age

Age is closely related to interest as noted in [29]. We use their binning into ages by discarding people younger than 16 and older than 30 as in Fig. 5, middle. We see three clusters (No. 2, 3 and 5) with users predominantly from US; these clusters are distinguished by the age distribution. When considering splits in the lower level of the hierarchy, we also see clustering by age within a given location as soon as the location becomes dominant within a cluster. For example SG (Fig. 7) and RU (Fig. 8) users are already partitioned into two by age on the second level.

4.3 Polarized opinions

Polarization over political views [1] or link polarity across the same topic [25] is examined by several authors. In our experiment we manually selected interest from user profiles that may be related to polarized opinion as in Table 6. While US parties did not show effect on principal axes, religion and atheism is characteristic in the LiveJournal Friends network. We selected users expressing Jesus, Jesus Christ, Atheism and Paganism as interest.

The top-level partitioning (Fig. 5) defines three US clusters, two with predominant interest in Jesus, while the third with Jesus in minority compared to Paganism and Atheism (this last cluster is also more international). In Fig. 5 we may also notice the apparent

729093	music	26266	Jesus
480443	movies	17828	Paganism
353412	reading	9845	Theology
331027	writing	9752	Atheism
312060	friends	6834	Democrats
251376	art	5054	Jesus Christ
229519	photography	2486	Republicans
217465	books	1677	Democrat
214479	dancing	1291	Republican

Table 6: Number of users who express certain type of interest. Left: the top list. Right: polarized interest categories.

correlation with younger age and interest in Jesus. We note that certain clusters such as the Russian one are underrepresented for this type of interest.

Conclusion

We performed a top-level analysis of the LiveJournal blogger Friends network, a data set of over three million users, in near 80% from US, 6% from Western Europe and 5% from Russia and East Europe. We demonstrated that principal component analysis can be performed on such a large power law network after appropriate preprocessing heuristics and the components reveal global aspects of the network such as location, age, or religious belief. Our preprocessing steps include the removal of densely connected communities that are of small size on the global scale as well as the contraction of long "tentacles", loosely connected users that form large chains out of the center of the network.

In future work more types of interest can be analyzed and the techniques presented here can be applied to blog posts or other large social networks.

Acknowledgment

To Jon Kleinberg and Lars Backstrom for providing us with the LiveJournal friends and communities data used in [5].

5. REFERENCES

[1] Lada A. Adamic and Natalie Glance. The political blogosphere and the 2004 u.s. election: divided they blog. In *LinkKDD '05: Proceedings of the 3rd international workshop on Link discovery*, pages 36–43, New York, NY, USA, 2005. ACM.

[2] Charles J. Alpert and Andrew B. Kahng. Multiway partitioning via geometric embeddings, orderings, and dynamic programming. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 14(11):1342–1358, 1995.

[3] Charles J. Alpert and Andrew B. Kahng. Recent directions in netlist partitioning: a survey. *Integr. VLSI J.*, 19(1-2):1–81, 1995.

[4] Charles J. Alpert and So-Zen Yao. Spectral partitioning: the more eigenvectors, the better. In *DAC '95: Proceedings of the 32nd ACM/IEEE conference on Design automation*, pages 195–200, New York, NY, USA, 1995. ACM Press.

[5] Lars Backstrom, Dan Huttenlocher, Jon Kleinberg, and Xiangyang Lan. Group formation in large social networks: membership, growth, and evolution. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 44–54, New York, NY, USA, 2006. ACM Press.

[6] Michael W. Berry. SVDPACK: A Fortran-77 software library for the sparse singular value decomposition. Technical report, University of Tennessee, Knoxville, TN, USA, 1992.

[7] Smriti Bhagat, Graham Cormode, S. Muthukrishnan, Irina Rozenbaum, and Hongyi Xue. No blog is an island - analyzing connections across information networks. In *Proceedings Int. Conf. on Weblogs and Social Media (ICWSM-2007)*, 2007.

[8] Alan Borodin, Gareth O. Roberts, Jeffrey S. Rosenthal, and Panayiotis Tsaparas. Finding authorities and hubs from link structures on the world wide web. In *Proceedings of the 10th World Wide Web Conference (WWW)*, pages 415–429, 2001.

[9] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. Graph structure in the web. In *Proceedings of the 9th World Wide Web Conference (WWW)*, pages 309–320. North-Holland Publishing Co., 2000.

[10] Andrei Z. Broder. On the Resemblance and Containment of Documents. In *Proceedings of the Compression and Complexity of Sequences (SEQUENCES'97)*, pages 21–29, 1997.

[11] S. Burer and R.D.C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95(2):329–357, 2003.

[12] Pak K. Chan, Martine D. F. Schlag, and Jason Y. Zien. Spectral k-way ratio-cut partitioning and clustering. In *DAC '93: Proceedings of the 30th international conference on Design automation*, pages 749–754, New York, NY, USA, 1993. ACM Press.

[13] D Cheng, R Kannan, S Vempala, and G Wang. On a recursive spectral algorithm for clustering from pairwise similarities. Technical report, MIT LCS Technical Report MIT-LCS-TR-906, 2003.

[14] Chris H. Q. Ding, Xiaofeng He, Hongyuan Zha, Ming Gu, and Horst D. Simon. A min-max cut algorithm for graph partitioning and data clustering. In *ICDM '01: Proceedings of the 2001 IEEE International Conference on Data Mining*, pages 107–114, Washington, DC, USA, 2001. IEEE Computer Society.

[15] Pedro Domingos and Matt Richardson. Mining the network value of customers. In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 57–66, New York, NY, USA, 2001. ACM.

[16] W. E. Donath and A. J. Hoffman. Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development*, 17(5):420–425, September 1973.

[17] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(98), 1973.

[18] Gary Flake, Steve Lawrence, and C. Lee Giles. Efficient identification of web communities. In *Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 150–160, Boston, MA, August 20–23 2000.

[19] Gary W. Flake, Robert E. Tarjan, and Kostas Tsioutsiouliklis. Graph clustering and minimum cut trees. *Internet Mathematics*, 1(4):385–408, 2003.

[20] M. Girvan and M. E. Newman. Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA*, 99(12):7821–7826, June 2002.

[21] E. Gorny. Russian livejournal: National specifics in the development of a virtual community. pdf online, May 2004.

[22] John Hopcroft, Omar Khan, Brian Kulis, and Bart Selman. Natural communities in large linked networks. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 541–546, New York, NY, USA, 2003. ACM.

[23] Matthew Hurst. 24 hours in the blogosphere. In *2006 AAAI Spring Symposium on Computational Approaches to Analyzing Weblogs*, pages 67–72, 2006.

[24] Matthew Hurst, Matthew Siegler, and Natalie Glance. On estimating the geographic distribution of social media. In *Proceedings Int. Conf. on Weblogs and Social Media (ICWSM-2007)*, 2007.

[25] Anubhav Kale, Amit Karandikar, Pranam Kolari, Akshay Java, Tim Finin, and Anupam Joshi. Modeling trust and influence in the blogosphere using link polarity. In *Proceedings Int. Conf. on Weblogs and Social Media (ICWSM-2007)*, 2007.

[26] G. Karypis. CLUTO: A clustering toolkit, release 2.1. Technical Report 02-017, University of Minnesota, Department of Computer Science, 2002.

[27] Jon Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[28] Pranam Kolari, Tim Finin, Kelly Lyons, Yelena Yesha, Yaacov Yesha, Stephen Perelgut, and Jen Hawkins. On the structure, properties and utility of internal corporate blogs. In *Proceedings Int. Conf. on Weblogs and Social Media (ICWSM-2007)*, 2007.

[29] Ravi Kumar, Jasmine Novak, Prabhakar Raghavan, and Andrew Tomkins. Structure and evolution of blogspace. *Commun. ACM*, 47(12):35–39, 2004.

[30] Ravi Kumar, Jasmine Novak, and Andrew Tomkins. Structure and evolution of online social networks. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 611–617, New York, NY, USA, 2006. ACM.

[31] Miklós Kurucz, András A. Benczúr, Károly Csalogány, and László Lukács. Spectral clustering in telephone call graphs. In *WebKDD/SNAKDD Workshop 2007 in conjunction with KDD 2007*, 2007.

[32] Kevin Lang. Fixing two weaknesses of the spectral method. In *NIPS '05: Advances in Neural Information Processing Systems*, volume 18, Vancouver Canada, 2005.

[33] Ronny Lempel and Shlomo Moran. The stochastic approach for link-structure analysis (SALSA) and the TKC effect. *Computer Networks*, 33(1–6):387–401, 2000.

[34] Thomas Lento, Howard T. Welser, Lei Gu, and Marc Smith. The ties that blog: Examining the relationship between social ties and continued participation in the wallop weblogging system. In *3rd Annual Workshop on the Weblogging Ecosystem*, 2006.

[35] Kristina Lerman. Social networks and social information filtering on digg. In *Proceedings Int. Conf. on Weblogs and Social Media (ICWSM-2007)*, 2007.

[36] Jure Leskovec, Mary McGlohon, Christos Faloutsos, Natalie S. Glance, and Matthew Hurst. Patterns of cascading behavior in large blog graphs. In *SDM*. SIAM, 2007.

[37] Mary McGlohon, Jure Leskovec, Christos Faloutsos, Matthew Hurst, and Natalie Glance. Finding patterns in blog shapes and blog evolution. In *Proceedings Int. Conf. on Weblogs and Social Media (ICWSM-2007)*, 2007.

[38] M. Meila and J. Shi. A random walks view of spectral segmentation. In *AISTATS*, 2001.

[39] M. Newman. Detecting community structure in networks. *The European Physical Journal B - Condensed Matter*, 38(2):321–330, March 2004.

[40] Andrew Y. Ng, Alice X. Zheng, and Michael I. Jordan. Link analysis, eigenvectors and stability. In *Proc. Int. Joint Conf. Artificial Intelligence, Seattle, WA*, August 2001.

[41] Matthew Richardson and Pedro Domingos. Mining knowledge-sharing sites for viral marketing. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 61–70, New York, NY, USA, 2002. ACM Press.

[42] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2000.

[43] Xiaolin Shi, Belle Tseng, and Lada Adamic. Looking at the blogosphere topology through different lenses. In *Proceedings Int. Conf. on Weblogs and Social Media (ICWSM-2007)*, 2007.

[44] Motoki Shiga, Ichigaku Takigawa, and Hiroshi Mamitsuka. A spectral clustering approach to optimally combining numerical vectors with a modular network. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 647–656, New York, NY, USA, 2007. ACM.

[45] Xiaowei Xu, Nurcan Yuruk, Zhidan Feng, and Thomas A. J. Schweiger. Scan: a structural clustering algorithm for networks. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 824–833, New York, NY, USA, 2007. ACM Press.

[46] Pavel Zakharov. Structure of livejournal social network. In *Proceedings of SPIE Volume 6601, Noise and Stochastics in Complex Systems and Finance*, 2007.

[47] Hongyuan Zha, Xiaofeng He, Chris H. Q. Ding, Ming Gu, and Horst D. Simon. Spectral relaxation for k-means clustering. In Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani, editors, *NIPS*, pages 1057–1064. MIT Press, 2001.

[48] Jun Zhang, Mark S. Ackerman, and Lada Adamic. Expertise networks in online communities: structure and algorithms. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 221–230, New York, NY, USA, 2007. ACM Press.