EigenSpokes: Surprising Patterns and Scalable Community Chipping in Large Graphs*

B. Aditya Prakash[1], Ashwin Sridharan[2], Mukund Seshadri[2], Sridhar Machiraju[3], and Christos Faloutsos[1]

[1] Computer Science Department, Carnegie Mellon University
{badityap,christos}@cs.cmu.edu
[2] Sprint Applied Research Labs, California
{ashwin.sridharan,mukund.seshadri}@sprint.com
[3] Google Inc., California
machiraju@acm.org

Abstract. We report a surprising, persistent pattern in large sparse social graphs, which we term *EigenSpokes*. We focus on large Mobile Call graphs, spanning about *186K* nodes and *millions* of calls, and find that the singular vectors of these graphs exhibit a striking *EigenSpokes* pattern wherein, when plotted against each other, they have clear, separate lines that often neatly align along specific axes (hence the term "spokes"). Furthermore, analysis of several other real-world datasets *e.g.,* Patent Citations, Internet, *etc.* reveals similar phenomena indicating this to be a more fundamental attribute of large sparse graphs that is related to their community structure.

This is the first contribution of this paper. Additional ones include (a) study of the conditions that lead to such *EigenSpokes*, and (b) a fast algorithm for spotting and extracting tightly-knit communities, called *SpokEn*, that exploits our findings about the *EigenSpokes* pattern.

1 Introduction

Given a large phone-call network, how can we find communities of users? While the behavior of users in landline networks has been examined before [4], we study here the phone call network of mobile users in cellular networks. The analysis of mobile phone graphs is interesting for multiple reasons as mobile phones are ubiquitous and are a key conduit for Internet access too. Several recent studies have used mobile call graph data to examine and characterize the social interactions of cell phone users, with a focus on understanding the structural properties of the graph [13,11,21], the evolution of social groups and the spread of new products and services [14].

* This material is based upon work supported by the National Science Foundation under Grants No. CNS-0721736 and CNS-0721889 and a Sprint gift. Research partly done during a summer internship by the first author at Sprint Labs. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation, or other funding parties.

M.J. Zaki et al. (Eds.): PAKDD 2010, Part II, LNAI 6119, pp. 435–448, 2010.
© Springer-Verlag Berlin Heidelberg 2010

Our objective in this paper is to identify if and to what extent do well-defined social groups of callers exist in such networks. We emphasize that understanding the *entire* graph structure is *not* our goal. Indeed, in large social networks, not *every* node can be expected to belong to a community. Hence, extraction of community-like structures, which can be independently analyzed, is the focus of this paper rather than graph-partitioning. This approach based on chipping off communities is also supported by recent studies [10] that have shown the presence of small communities loosely connected with the remaining "core" of the graph. Furthermore, when we applied well-known graph clustering techniques on our datasets, none of them provided much insight into chipping off interesting community structures for further analysis, since these techniques are geared towards partitioning the *entire* graph. What was surprising, though, was our discovery of the '*spokes*' (or *EigenSpokes*) phenomenon (see Figure 1(b)): the singular vectors of the Mobile Call graph, when plotted against each other, often have clear separate lines, typically aligned with axes. We term such plots EigenEigen (or EE) plots.
We concentrate on three key questions in this paper:

1. **Cause:** What causes these *spokes*?
2. **Ubiquity:** Do they occur across varied datasets to be worth studying?
3. **Community Extraction:** How can we exploit them, to chip off meaningful communities from large graphs?

We answer these three questions on graphs of Table 1. Our primary dataset in this paper is an anonymized social graph based on mobile calls made from/to callers located within a geographically contiguous urban area. In this social graph, callers are represented as nodes, and edges represent calls between nodes. This Mobile Call graph captures activity over the duration of a month (millions of successfully completed calls) and consists of about $186,000$ nodes and $464,000$ edges. We also investigate similar Mobile Phone graph datasets obtained from other geographic areas, to support our findings. Since these graphs are disconnected, we focus on the largest connected component. Our graphs exhibit characteristics such as degree distributions and generative processes similar to those of other mobile call graphs [11,21].

Table 1. Graph datasets used in this paper

Name	Description	Nodes	Edges
Mobile Call graph	Calls between callers/callees	186,000	464,000
Patent Citations	Citations between patents	3,774,768	14,970,767
Internet routers	Network links between routers	124,651	207,214
Dictionary words	Words are connected if they differ by a single letter	52,652	178,076

In addition to the above, we also investigate several other datasets (Table 1) in the public domain[1] which allow us to determine the generality of our observations

[1] `http://www.cise.ufl.edu/research/sparse/matrices/`

and the underlying phenomenon. Also, they have meta-information that helps us demonstrate that our algorithm chips-off meaningful communities.

The following sections discuss the related work, problems with traditional methods, explain the *EigenSpokes* pattern, develop the *SpokEn* algorithm and finally present many surprising communities found in the datasets.

2 Related Work

Graph partitioning is a popular approach for studying community structure in graphs. Popular methods include Spectral clustering (see [24] for a survey), a "cut-based" method for understanding graph structures, which has been successful in machine-learning and image segmentation. Similar approaches (e.g. [19,16]) use the eigenvectors of the adjacency matrix. Lastly, spectral inspired methods have been used to learn model parameters for well-separated Gaussian mixtures [22]. Alternative cut-based multilevel approaches like Metis [8] and Graclus [5,20] coarsen the graph by coalescing nodes and then apply refinement steps to recover partitions. We address both the Spectral and multi-level partitioning techniques in greater detail in § 3.

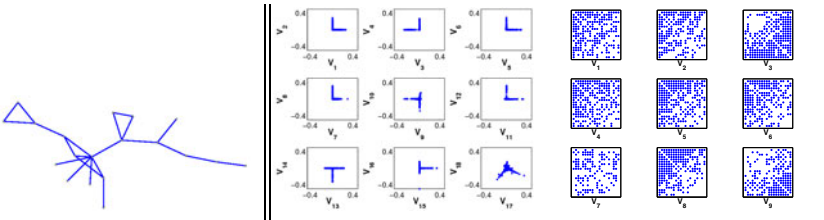
Cross-Association [2] partitions the graph so as to maximize information compression, but is limited to bi-partite structures. Co-clustering [6] trys to maximize mutual information, but like k-means, requires *a priori* information on the number of clusters. More generally in terms of clique extraction, [15] trys to extract quasi-cliques from graphs. Our focus however is on *chipping* out general community structures.

Modularity based approaches compare a graph's community structure against a random graph. Studies have proposed using modularity based Laplacian-like matrices [12,25] or greedy heuristics [3] for graph clustering. However all these techniques also partition the *entire* graph rather than extract relevant communities, which is our objective.

In terms of social network analysis, [18] extracts communities from an Instant Messenger Network by applying Co-clustering. [7] proposes flow-based techniques to identify Web-based communities. However, it identifies communities for a set of *known* nodes, while our objective is to extract all nodes that constitute communities. Also, although the focus of [26] is on randomness measures, they observed quasi-orthogonal spectral lines in context of small cavemen-like graphs. To the best of our knowledge the *EigenSpokes* pattern has not been observed in any *real*, *large* social networks.

3 Why Not Traditional Methods?

We analyzed the mobile call graph using well-known spectral clustering [24] and multi-level graph partitioning [8,5] techniques. Our goal here is to explore if these methods can help us extract communities of nodes for further analysis. Although remarkably successful in other settings like image segmentation etc., we find, as shown below, that these methods do not yield good communities in our graph.



(a) L_{rw}: **Note the long chains** (b) **EE-plot** (c) **Spy Plots of subgraph of Top 20 Nodes**

Fig. 1. (a) Typical Partition using L_{rw} (b),(c) ***EigenSpokes*** in Region 1 & Time 1

Spectral Clustering: L_{rw}**,** L_{sym}**, ...** Many "Laplacian" matrices can be defined on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ (see [24]). We applied the L_{rw} method [24] on our primary data set to obtain k-way partitions from $k = 2$ to $k = 100$. As in recent studies ([9], [10]), we found that the application of the technique yields :a) skewed partitions consisting of very small clusters and a large 'core' and b) the clusters lack internal coherence. The partitions we got from L_{rw} while lowering the N-cut value had no or little internal coherence with long chains and most nodes connected to 1 or 2 other nodes (see Figure 1(a)). We experimented with several other Laplacians including L_{sym} [24] and L_Q (based on modularity) [25] but obtained similar results of limited utility.

Graph Partitioning Methods Prevailing multilevel algorithms for graph partitioning like Metis [8] and their improvements like Graclus [5] and MCR-MCL [20] are based on repeated coarsening and refinements of nodes with emphasis on balanced cuts. To explore how well multilevel algorithms perform, we ran Graclus on the Mobile Call graph to obtain k partitions. We invoked the algorithm with various values for k from $k = 2$ to $k = 10,000$. While Graclus yields more balanced clusters than Spectral partitioning, we observed that, as before, the clusters lack internal coherence. This can be attributed, again, to the following two causes : a) these algorithms also utilize a cut-based metric and b) their objective is to partition the *entire* graph; as shown by our results as well as [10], this is not feasible when applied to social graphs that comprise of a large set of random nodes and small communities.

4 EigenSpokes

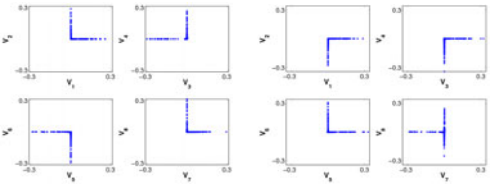
As demonstrated in the previous section (and prior work), Laplacians in certain large graphs yield communities that have a low cut but possess little internal coherence. Hence, we investigate using the adjacency matrix itself. This leads to several interesting observations, and motivates our approach for community identification.

4.1 A Surprise: Spokes

Recall that the *Singular Value Decomposition* (SVD) of an $m \times n$ matrix W is a factorization defined as: $W = U\Sigma V^T$, where U and V are $m \times m$ and

$n \times n$ size matrices respectively, and Σ is an $m \times n$ diagonal matrix comprised of the singular values. Taking the top K values of Σ yields the best rank-K approximation (w.r.t. the Frobenius norm) to the original matrix [23].

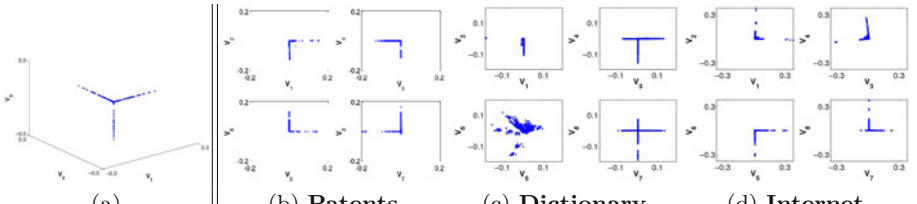
We define the *EE-plot* as the scatter plot of vector U_i and U_j, for any i and j, i.e., they plot one point (U_{in}, U_{jn}) for each node n in the graph. Surprisingly, we find that the *EE-plots* for our Mobile Call graph show clear separate straight lines that are often aligned with axes[2]! We call this the *EigenSpokes* pattern. This is demonstrated in Figure 1(b), where we plot the first $K = 18$ singular vectors pairwise. Even more striking is that, as shown in Figure 2, these spokes occur in many Mobile Call graphs collected at various points of time (separated by several months) and at various geographic regions.



(a) **Region 2, Time 2** (b) **Region 3, Time 3**

Fig. 2. *EE-plots* for Mobile Call graphs for different geographic regions and time periods: note the persistence of *EigenSpokes*

Some intuition: We delve further into the *EigenSpokes* pattern by identifying the nodes that lie on the extremities of the "spokes"; more precisely, for each of the first 9 singular vectors, we identify the 20 nodes that had the highest magnitude projection along that vector. We then plot the induced sub-graph of these nodes (see Figure 1(c)). Clearly, almost all of the induced sub-graphs contain near-cliques. *These observations hint toward a strong connection between EigenSpokes and communities, and raise the following questions : are these spokes representative of fundamental community structures; do they occur elsewhere? What is their origin and how exactly can they be used for chipping off communities?*



(a) (b) **Patents** (c) **Dictionary** (d) **Internet**

Fig. 3. (a) *EEE*-plot for Mobile Call graph and *EigenSpokes* in other Real-World Data Sets (b),(c),(d)

We answer these questions next: in § 4.2, we provide a rigorous basis for the link between *EigenSpokes* and communities; in § 4.3, we show that *EigenSpokes* can be observed in several real-world graphs; and in § 4.4, we demonstrate the various conditions that lead to the presence (and absence) of the *EigenSpokes* phenomenon.

[2] Axis-alignment is distinct from orthogonality of the vectors.

4.2 Justification and Proofs

Our focus in this paper is on undirected graphs, which implies that the adjacency matrix A is a square, symmetric matrix. For such a matrix, it is well known that the singular values are absolute values of the non-zero eigenvalues: $\sigma_i = |\lambda_i|$ and the singular vectors coincide with the non-null eigenvectors. Due to the equivalence between singular vectors and eigenvectors for the (symmetric) graphs considered in this paper, we abuse language and often refer to singular vectors as being the spectra of A.

Given the presence of *EigenSpokes*, is it reasonable to expect the nodes lying along the extremes to have similar connectivity patterns?

***EigenSpokes*, Connectivity and Communities:** The presence of spokes in *EE-plots* (axis-aligned or not) implies that nodes close to each other on a line have similar scores along *two* eigenvectors ('score' of node n along vector U_i is U_{in}). In fact, plots of the first 3 eigenvectors, or *EEE-plots* (see Figure 3) show lines too. This strongly suggests similar scores for the nodes in *many* vectors. Specifically, consider two nodes i and j whose connectivity information is represented by their rows A_i and A_j in A. If the k^{th} eigenvector is denoted by U_k, then A_iU_k and A_jU_k are the i^{th} and j^{th} components of U_k. These will be equal if $A_i = A_j$. Hence, the two nodes will have the same components along the eigenvectors. In general, we expect that *nodes with similar connectivity will have similar scores along the vectors of* U.

Is the converse also true? We can prove the following lemma to this end (note that $\langle x, y\rangle = x^T y$ denotes the dot-product of two column vectors x and y):

Lemma 1. *For any real, symmetric adjacency matrix A, if for any i and j, $\forall k,\ |\langle (A_i - A_j)^T, U_k\rangle| \leq \epsilon$, then $\forall k,\ |A_{ik} - A_{jk}| \leq (\epsilon\sqrt{N})$ as well.*

Proof. As A is real symmetric, by the Spectral Theorem, it is orthogonally diagonalizable. Hence, it is non-defective and has a full basis of eigenvectors. Consider any vector $C = \sum_k c_k U_k$ written using the basis consisting of the eigenvectors. Then,

$$\langle (A_i - A_j)^T, C\rangle \ = \ \sum_k c_k \langle (A_i - A_j)^T, U_k\rangle$$

$$\leq \ \sqrt{\sum_k c_k^2}\sqrt{\sum_k \langle (A_i - A_j)^T, U_k\rangle^2} \ \leq \ \sqrt{\sum_k c_k^2}\sqrt{N}\ \epsilon$$

where we use the Cauchy-Schwartz inequality in step 2 and given bound in step 3. Use the above inequality for $C = e_k$ (indicator vector which is zero everywhere except at index k where it is 1), for every k. Also, note that orthogonal transformations preserve the norm of a vector - hence, $\sqrt{\sum_k c_k^2} = \| C \|$, which would be equal to 1 for our choice of C's. Therefore, we get: $\forall k, |A_{ik} - A_{jk}| \leq (\epsilon\sqrt{N})$. □

Note that the above proof holds for any orthogonal basis set of vectors U; but our basis set U is also a carefully chosen set: it is the set of singular vectors. Hence, we expect the bound to be tighter in practice.

In view of the above, we expect that nodes lying close to each other on a spoke will have similar neighbor sets. But what is the link between similar neighbor sets and communities in the graph? Consider the following two (sufficient) conditions that result in similar neighbor sets:

1. Cliques (or near-cliques) result in exactly the same [3] (or similar) adjacency rows for nodes in the cliques.
2. Perfect (or near-perfect) Bipartite-cores also result in the same (or similar) adjacency rows for nodes in the two cores.

Consequently, we expect to see communities in the form of (near-)cliques or (near-)bi-partite cores among the nodes in the spokes.

Axis-alignment (or not): Recall another striking feature of the *EE-plots*: the presence of largely *axis-aligned* spokes. From the preceding discussion, given the presence of *EigenSpokes*, we should traverse the extreme points on each spoke to extract communities. This reduces to searching for 'high-scoring' nodes in each singular vector *separately*, since axis-alignment implies that the extreme points have high values only in one of the vectors. In fact, prior work [1] has already shown that nodes with scores at the extreme ends of the principal eigenvector of Erdos-Renyi graphs do belong to the same clique. However, some spokes are not axis-aligned (as in the last *EE*-plot of Figure 1(b)). This implies that some nodes have significant scores along multiple singular vectors. In the specific case of *EE-plots* the linear nature of spokes means that the scores of the nodes are *linearly correlated*. Hence, exploring dominant nodes along one singular vector should be sufficient to extract nodes of a community.

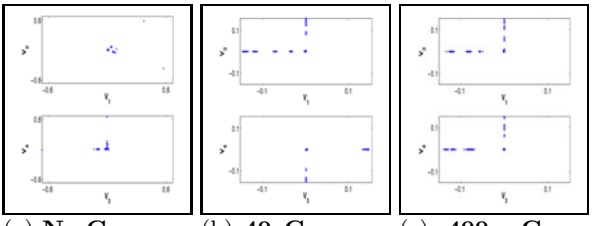
4.3 Ubiquity of Spokes

Apart from Mobile Call graphs, we have observed the *EigenSpokes* pattern with singular vectors of several other real world graphs. We show the *EE-plots* for Patents, Dictionary and Internet router connectivity in Figure 3(b), 3(c) and 3(d) ([17] contains more detailed plots). In all three cases, we see that most pairwise combinations align in a spoke pattern, with some exceptions in the Dictionary graph. We also observe that some of these spoke patterns are not axis-aligned; as discussed earlier though, the linear correlation between the scores of the nodes is preserved. Thus the *EigenSpokes* pattern is persistent across a wide array of diverse datasets.

4.4 Recreating Spokes

So far, we have provided some insight into why spokes arise. We now demonstrate exactly which features of graphs and community structure result in spokes using both synthetic and real graphs. Synthetic graphs, in particular, allow us to experiment with various parameters and characteristics, and observe their effect on their *EE*-plots. We show that the key factors responsible for these patterns are a **large** number of **well-knit** communities embedded in very **sparse** graphs.

[3] Not considering self-edges.



(a) **No Communities** (b) **40 Communities** (c) **400 Communities**

Fig. 4. Genesis of *EigenSpokes*: synthetic heavy-tailed random graphs with community structure

We started with a synthetic random heavy-tailed graph with the same number of nodes and degree distribution as our Mobile Call graph but with no community structure. The EE-plots don't exhibit any spokes pattern (Figure 4(a)) but, when we synthetically introduce 40 communities (near-cliques of sizes $31 - 50$, with a probability 0.8 of an intra-community edge) into the above random graph, in Figure 4(b), we observe the emergence of the spokes pattern. When we increase the number of communities to 400, in Figure 4(c), the spoke pattern becomes more clear, and resembles Figure 2. Further, we verified that the nodes at the extremities do indeed form the artificially embedded communities. We also found that the nature of the communities, including the level of internal connectivity, does not affect the emergence of the spokes pattern as long as such connectivity is significant. Thus we infer that one of the important causes for a spokes pattern is the presence of a **large number of tightly knit communities** in the graph.

Due to lack of space we omit details about the effect of sparsity and degree density; they are demonstrated in greater detail in [17].

5 *SpokEn*: Exploiting EigenSpokes

Based on the insights from previous sections, we now develop our community identification approach *SpokEn*, that exploits *EigenSpokes*. While developing *SpokEn*, we also use experiments with synthetically generated graphs to help us choose from the various algorithmic choices. Our approach differs from prior work on graph partitioning as for certain classes of graphs, we observe a specific structural property and exploit it. While this may not apply to all graphs, our approach is highly effective for the large sparse graphs we consider, as shown by our results later in the paper.

5.1 Designing *SpokEn*

Our proposed approach is based on the key property of *EigenSpokes* highlighted in § 4.2: the existence of *EigenSpokes* indicates the presence of well-knit communities whose nodes have a significant component in *that* singular vector. Thus an appropriate traversal of each singular vector in *isolation* can extract these communities. A good traversal should select only the nodes which belong to a coherent community. We now discuss where to start the traversal, how to *grow* the community and finally, when to stop.

Initialization: We choose the node with the score of maximum magnitude as the *seed* for the community. We multiply the given singular vector U_i by -1 if necessary to ensure that the score with the largest magnitude is positive.

Discovery: A simple algorithm for discovery is one that picks nodes in decreasing order of their scores. Such an algorithm can pick a node that is disconnected from all the nodes chosen previously. Hence, we propose the following: let C denote the set of all nodes that have been discovered so far; the next node that we select is the node with the largest score that is connected to some node in C. Formally, we augment C with a node n^* that satisfies $n^* = \arg\max_{n \in N_C} U_i(n)$, where N_C is the neighborhood of C[4]. This algorithm is intuitive and keeps C always connected.

Termination and Trimming: For termination, we need to use a metric that quantifies the quality of the community extracted so far. We propose to use a novel hybrid approach based on conductance [24] for cut and modularity (actually *relative* modularity) for coherence. The process discovers and adds nodes to the set C as long as the relative modularity increases and terminates once it reduces indicating reduction in community structure. We finally use a conductance based method to trim out the remaining false positives.

5.2 Discussion

Relative Modularity: In large graphs such as ours, underlying communities are typically small ($10 \approx 100$ in a million node-graph) [10]. The equation for modularity[5] indicates that when extracting a *single* small community from a large graph, the modularity metric computed on such a highly unbalanced partition would be dominated by the larger partition and *not* the discovered community, thus rendering it useless. This was also empirically observed in extensive evaluations over our datasets.

To resolve this problem, we once again utilize the concept of scores. When traversing a singular vector (say U_i), as a pre-processing step, we construct a *new* sub-graph $G_\epsilon = (\mathcal{V}_\epsilon, E_\epsilon)$ from G wherein we discard all nodes n with values $U_i(n)$ *below* a certain threshold ϵ. The modularity computation is then conducted w.r.t. G_ϵ (hence *relative* modularity). This is justified since in the first place, we do not expect the discarded nodes to belong to the community under consideration. The removal of such nodes induces a more balanced partition and makes the modularity values more meaningful. We set $\epsilon = 10^{-4}$ in our experiments as several tests showed the results were insensitive around it.

Trimming using Conductance: As shown later in § 5.3, conductance as a termination criterion results in *premature* termination of the discovery process,

[4] N_C is the set of nodes not in C and are connected to at least one node in C.

[5] The modularity of a graph partition $\mathcal{C} = \{C_1, C_2, \ldots\}$ is $Q(C) = \frac{1}{2m} \sum_{C \in \mathcal{C}} \sum_{i,j \in C \cap \mathcal{V}} \left[A_{ij} - \frac{k_i k_j}{2m} \right]$, where k_i is the degree of a node and A_{ij} an element of the adjacency matrix.

causing several false negatives while relative modularity as a termination metric often results in *overshooting* and hence false positives. These two observations indicate that modularity and conductance are *complementary* in the role of a termination metric which is why we adopt a hybrid approach. Hence as mentioned before, after a community is extracted using relative modularity (to discover all relevant nodes at the cost of false positives), a standard spectral technique (L_{rw}) is used to trim-out false positives by further bisection to determine a better cut. We give the pseudo-code of *SpokEn* in Algorithm 1.

Algorithm 1. *SpokEn*

Require: Symmetric binary adjacency matrix A
1: U = get first several eigenvectors of A
2: Stop if U has no *EigenSpokes* pattern
3: **for all** eigenvectors $v = U_k$ **do**
4: Construct $G_\epsilon = (\mathcal{V}_\epsilon, E_\epsilon)$ such that $\mathcal{V}_\epsilon = \{i : v(i) > \epsilon\}$
5: Initialize *outputSet* using *seed* //see *Initialization*
6: //see relative modularity in *Termination*
7: **while** modularity(*outputSet*, G_ϵ) increases **do**
8: Expand *outputSet* //see *Discovery*
9: **end while**
10: C_k = trim *outputSet* using conductance //see *Triming using Conductance*
11: **end for**
12: **return** $\{C_k\}$

5.3 Empirical Results

To evaluate the performance of our discovery process and termination criterion, we applied *SpokEn* on various synthetically generated graphs with known ground truths in each case (like the ones described in § 4.4 including an ER graph embedded with *bi-partite cores*). While the detailed results are provided in [17], we found that conductance as a termination criterion undershot and hence detected fewer communities (about 60% with almost no false positives). On the other hand, modularity discovered about 80% communities but with 4% false positives. Their combination, *SpokEn*, was able to identify 76-90% of the embedded communities with almost no false positives.

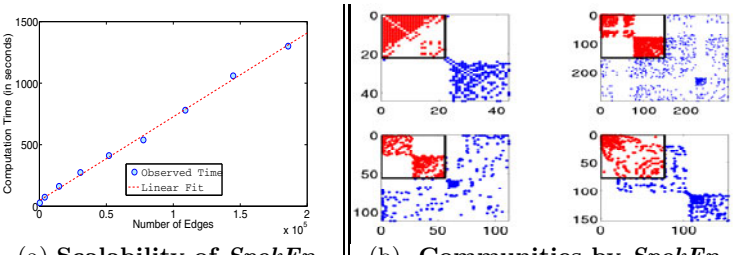
Speed : The computation time is mainly dominated by the eigenvector calculation which is linear in edges. We ran *SpokEn* on graphs of various sizes on a Dell Server with an Intel Xeon 3 GHz processor and 4 GB of RAM. In each case, we computed 100 singular vectors and mined communities from them. Figure 5(a) plots the computation time required by *SpokEn* to extract communities as a function of the number of edges of the graph. As expected, it clearly shows the processing time is *linear* in the number of graph edges, which is a key indication of scalability.

6 Successes with Real-World Graphs

We applied *SpokEn* to our real-world datasets and found that it extracts several interesting communities that reveal useful and relevant information about the connectivity patterns.

We illustrate four typical communities extracted by *SpokEn* from our Mobile Call graph. Figure 5(b) presents the spy plots which clearly show that the communities are well-connected (the communities are red nodes on the top delineated by black boxes). To show the success of our terminating criterion, we plot additional nodes that would have been explored by the discovery process without termination. Notice that *SpokEn* does indeed typically stop at points where a community appears to have ended. In the bottom left case, however, it overshoots and combines what appear to be two near-cliques into the same community. We observe similar results for the other singular vectors as well: *SpokEn* extracts communities of nodes with good internal coherence though it sometimes clubs together two such communities into one.

The prevalence of such close-knit communities of more than 10 nodes in a Mobile Call graph was quite unexpected to us. Temporal analysis of the usage of at least a few communities leads us to believe that these communities arise from users of the same organization.



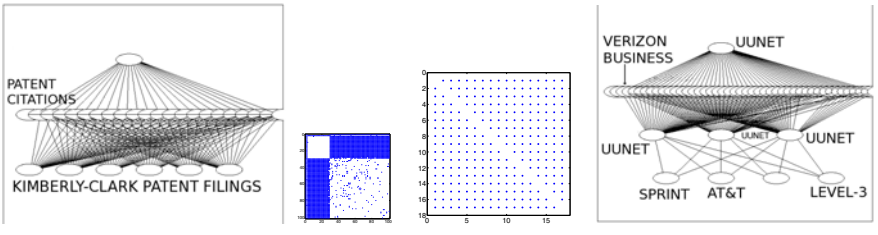
(a) **Scalability of *SpokEn*** (b) **Communities by *SpokEn***

Fig. 5. (a) *SpokEn*: Computation Time is linear in #(edges) **(b)** Four communities (delimited by black box and red dots) extracted by *SpokEn* from Mobile Call graph (spy plots). To illustrate the success of the termination criterion, we plot additional nodes in the order of discovery. Notice the near-clique (= near block diagonal) behavior.

Next, we analyze a few typical communities from other data sets. Figure 6(a) plots the connectivity between nodes of a community extracted from the Patent citation dataset. Notice the striking bipartite nature of the community. Upon further investigation, we find that the bipartite nature arises because of patents filed by the *same* organization on related topics reuse the bibliography entries. In the example shown here, one-half of the bi-partite graph comprises of about 25 patents that were filed in a period of $1998 - 1999$ by (the same) authors from Kimberly-Clark in the area of photosensitive pigments for color printers. The bi-partite nature of the graph arises because all these 25 patents cited the

same set of past references. This also illustrates a crucial aspect of *SpokEn*: it extracts communities of nodes with *similar* connectivity. This may or may not imply *mutual* connectivity.

Figure 6(b) shows a typical community extracted from the Dictionary graph. Recall that this dataset connects two words if they differ by exactly one letter. The clique shown in Figure 6(b) arises from three-letter words that all end with 'on'. We found many similar cases including words that end with 'an', 'll', etc. Finally, Figure 6(c) shows a community extracted from the router graph. The community highlights the tiering relationship typical in the Internet. The community consists of 4 UUNET back-bone routers (first and third row) from the Tier-1 layer that serve as gateways for a large community of Tier-2 Verizon Business and other small business (the second row) and are also connected to other Tier-1 routers (Sprint, AT&T *etc.,* last row).



(a) (b) (c)

Fig. 6. Structures extracted by *SpokEn* from real-world graphs. **(a)** A typical bi-partite community extracted from the Patent graph. It arises due to "cut-and-paste" bibliography generation. On the left, we plot a portion of the community for visual understanding, and the entire spy plot on the right. **(b)** A typical near-clique subgraph of words from the Dictionary graph. The words all differ by exactly one letter from each of the others. **(c)** Internet router connectivity from one provider to customers causing a bipartite community.

7 Conclusions

In answer to the questions we posed earlier in § 1, we find that:

1. **Cause:** *Spokes* can be strongly associated with the presence of well-defined communities like cliques and bi-partite cores in sparse graphs.
2. **Ubiquity:** Apart from Mobile Call graphs, they occur in a variety of datasets such as Patent citations, Dictionary and Internet.
3. **Community Extraction:** The *spokes* pattern allows us to construct an efficient and scalable algorithm "*SpokEn*" that helps us chip off communities thereby revealing several interesting structures in Mobile Call graphs as well as the other datasets.

References

1. Alon, N., Krivelevich, M., Sudakov, B.: Finding a large hidden clique in a random graph. In: Proc. of the Ninth Annual ACM-SIAM SODA (1998)
2. Chakrabarti, D., Papadimitrou, S., Modha, D., Faloutsos, C.: Fully automatic cross-associations. In: Proc. of the tenth ACM SIGKDD (2004)
3. Clauset, A., Newman, M.E.J., Moore, C.: Finding Community Structure in Very Large Networks. Physical Review (2004)
4. Cortes, C., Pregibon, D., Volinsky, C.: Communities of interest. In: Hoffmann, F., Adams, N., Fisher, D., Guimarães, G., Hand, D.J. (eds.) IDA 2001. LNCS, vol. 2189, pp. 105–114. Springer, Heidelberg (2001)
5. Dhillon, I., Guan, Y., Kullis, B.: Weighted Graph Cuts without EigenVectors: A Multilevel Approach. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1944–1957 (November 2007)
6. Dhillon, I., Mallela, S., Modha, D.: Information-theoretic co-clustering. In: Proc. of the ninth ACM SIGKDD (2003)
7. Flake, G., Lawrence, S., Giles, L.: Efficient identification of web communities. In: Proc. of Sixth ACM KDD (2003)
8. Karypis, G., Kumar, V.: A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM Jnl. on Sci. Comp. 20(1), 359–392 (1999)
9. Lang, K.: Fixing two weaknesses of the spectral method. In: Proc. of NIPS (2006)
10. Leskovec, J., Lang, K.J., Dasgupta, A., Mahoney, M.W.: Statistical properties of community structure in large social and information networks. In: WWW (2008)
11. Nanavati, A.A., Gurumurthy, S., Das, G., Chakraborty, D., Dasgupta, K., Mukherjea, S., Joshi, A.: On the structural properties of massive telecom call graphs: findings and implications. In: Proc. of 15th ACM CIKM, pp. 435–444 (2006)
12. Newman, M.E.J.: Finding community structure in networks using the eigenvectors of matrices. Phys. Rev. E 74(036104) (2006)
13. Onnela, J.-P., Saramaäki, J., Hyvöven, J., Szabó, G., de Menezes, M.A., Kaski, K., Barabási, A.-L.: Structure and Tie Strengths in Mobile Communication Networks. New Journal of Physics 9 (2007)
14. Palla, G., Barabasi, A.-L., Vicsek, T.: Quantifying social group evolution. Nature 446(664) (2007)
15. Pei, J., Jiang, D., Zhang, A.: On mining cross-graph quasi-cliques. In: Proc. of the eleventh ACM SIGKDD (2005)
16. Perona, P., Freeman, W.T.: A factorization approach to grouping. In: Burkhardt, H.-J., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1406, pp. 655–670. Springer, Heidelberg (1998)
17. Prakash, B.A., Sridharan, A., Seshadri, M., Machiraju, S., Faloutsos, C.: Patterns and community extraction in large graphs. TR CMU-CS-09-176 (2009)
18. Resig, J., Dawara, S., Homan, C., Teredesai, A.: Extracting social networks from instant messaging populations. In: Proc. of ACM SIGKDD (2004)
19. Sarkar, S., Boyer, K.L.: Quantitative measures of changed based on feature organization: Eigenvalues and eigenvectors. Computer Vision and Image Understanding 71(1), 110–136 (1998)
20. Satuluri, V., Parthasarathy, S.: Scalable graph clustering using stochastic flows: applications to community discovery. In: Proc. of ACM SIGKDD (2009)
21. Seshadri, M., Machiraju, S., Sridharan, A., Bolot, J., Faloutsos, C., Leskovec, J.: Mobile call graphs: Beyond power-law and lognormal distributions. In: KDD (2008)

22. Shi, T., Belkin, M., Yu, B.: Data Spectroscopy: Learning Mixture Models using Eigenspaces of Convolution Operators. In: Proc. of ICML (2008)
23. Strang, G.: Introduction to Linear Algebra. Wellesley-Cambridge Press (2003)
24. von Luxburg, U.: A tutorial on spectral clustering. Statistics and Computing 17(4), 395–416 (2007)
25. White, S., Smyth, P.: A spectral clustering approach to finding communities in graphs. In: Proc. of SDM (2005)
26. Ying, X., Wu, X.: On randomness measures for social networks. In: SDM (2009)