OddBall: Spotting Anomalies in Weighted Graphs

Leman Akoglu Mary McGlohon Christos Faloutsos

Carnegie Mellon University, School of Computer Science
{lakoglu,mmcgloho,christos}@cs.cmu.edu

Abstract. Given a large, weighted graph, how can we find anomalies? Which rules should be violated, before we label a node as an anomaly? We propose the OddBall algorithm, to find such nodes. The contributions are the following: (a) we discover several new rules (power laws) in density, weights, ranks and eigenvalues that seem to govern the so-called "neighborhood sub-graphs" and we show how to use them for anomaly detection; (b) we carefully choose features, and design OddBall, so that it is scalable and it can work un-supervised (no user-defined constants) and (c) we report experiments on many real graphs with up to *1.6 million* nodes, where OddBall indeed spots unusual nodes that agree with intuition.

1 Introduction

Given a real graph, with weighted edges, which nodes should we consider as "strange"? Applications of this setting abound: For example, in network intrusion detection, we have computers sending packets to each other, and we want to know which nodes misbehave (e.g., spammers, port-scanners). In a who-calls-whom network, strange behavior may indicate defecting customers, or telemarketers, or even faulty equipment dropping connections too often. In a social network, like FaceBook and LinkedIn, again we want to spot users whose behavior deviates from the usual behavior, such as people adding friends indiscriminately, in "popularity contests".

The list of applications continues: Anomalous behavior could signify irregularities, like credit card fraud, calling card fraud, campaign donation irregularities, accounting inefficiencies or fraud [7], extremely cross-disciplinary authors in an author-paper graph [31], network intrusion detection [30], electronic auction fraud [12], and many others.

In addition to revealing suspicious, illegal and/or dangerous behavior, anomaly detection is useful for spotting rare events, as well as for the thankless, but absolutely vital task of data cleansing [14]. Moreover, anomaly detection is intimately related with the pattern and law discovery: unless the majority of our nodes closely obey a pattern (say, a power law), only then can we confidently consider as outliers the few nodes that deviate.

Most anomaly detection algorithms focus on clouds of multi-dimensional points, as we describe in the survey section. Our goal, on the other hand, is

to spot strange nodes in a *graph*, with weighted edges. What patterns and laws do such graphs obey? What features should we extract from each node?

We propose to focus on neighborhoods, that is, a sphere, or a ball (hence the name `OddBall`) around each node(the *ego*): that is, for each node, we consider the induced sub-graph of its neighboring nodes, which is referred to as the *egonet*. Out of the huge number of numerical features one could extract from the egonet of a given node, we give a carefully chosen list, with features that are effective in revealing outliers. Thus, every node becomes a point in a low-dimensional feature space.

Main contributions of this work are:

1. *Egonet patterns*: We show that egonets obey some surprising patterns (like the *Egonet Density Power Law* (*EDPL*), *EWPL*, *ELWPL*, and *ERPL*), which gives us confidence to declare as outliers the ones that deviate. We support our observations by showing that the *ERPL* yields the *EWPL*.
2. *Scalable algorithm*: Based on those patterns, we propose `OddBall`, a scalable, un-supervised method for anomalous node detection.
3. *Application on real data*: We apply `OddBall`[1] to numerous real graphs (DBLP, political donations, and other domains) and we show that it indeed spots nodes that a human would agree are strange and/or extreme.

Of course, there are numerous types of anomalies - we discuss several of them in our technical report [3], but, for brevity, we focus on only the following major types (see Fig.1 for examples and Section 2 for the dataset description):

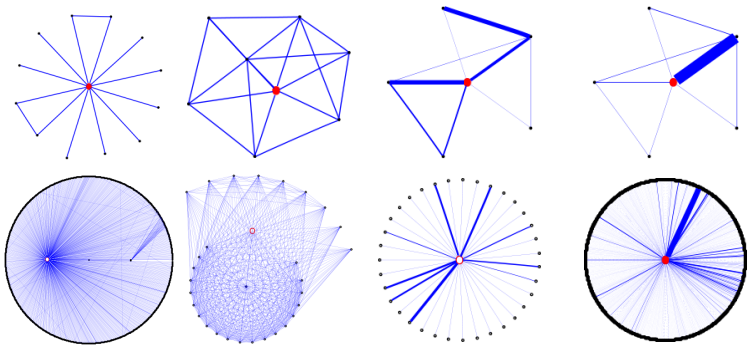
1. *Near-cliques* and *stars*: Those nodes whose neighbors are very well connected (near-cliques) or not connected (stars) turn out to be "strange": in most social networks, friends of friends are often friends, but either extreme (clique/star) is suspicious.
2. *Heavy vicinities*: If person i has contacted n distinct people in a who-calls-whom network, we would expect that the number of phone calls (weight) would be proportional to n (say, 3xn or 5xn). Extreme total weight would be suspicious, indicating, e.g., faulty equipment that forces redialing.
3. *Dominant heavy links*: In the who-calls-whom scenario above, a very heavy single link in the 1-step neighborhood of person i is also suspicious, indicating, e.g., a stalker that keeps on calling only one of his/her contacts an excessive count of times.

The upcoming sections are as follows: We describe the datasets; the proposed method and observed patterns; the experimental results; prior work; and finally the conclusions.

2 Data Description

We studied several unipartite/bipartite, weighted/unweighted large real-world graphs in a variety of domains, described in detail in Table 1. Particularly, unipartite networks include the following: *Postnet* contains post-to-post links in a

[1] Source code of our algorithm can be found at `www.cs.cmu.edu/~lakoglu/#tools`



(a) Near-star (b) Near-clique (c) Heavy vicinity (d) Dominant edge

Fig. 1. Types of anomalies that OddBall detects. Top row: toy sketches of egonets (ego shown in larger, red circle). Bottom row: actual anomalies spotted in real datasets. (a) A near-star in *Postnet*: `instapundit.com/archives/025235.php`, an extremely long post on Hurricane Katrina relief agencies with numerous links to diverse other posts about donations. (b) A near-clique in *Postnet*: `sizemore.co.uk`, who often linked to its own posts, as well as to its own posts in other blogs. (c) A heavy vicinity in *Postnet*: `blog.searchenginewatch.com/blog` has abnormally high weight w.r.t. the number of edges in its egonet. (d) Dominant edge(s) in *Com2Cand*: In FEC 2004, George W. Bush received a huge donation from a single committee: Democratic National Committee (~$87M)(!) - in fact, this amount was *spent against* him; next heaviest link (~$25M): from Republican National Committee.

set of blogs[23], *Enron* contains emails at Enron collected from about 1998 to 2002 (made public by the Federal Energy Regulatory Commission during its investigation), and *Oregon* contains AS peering information inferred from Oregon route-views BGP data. Bipartite networks include the following: *Auth2Conf* contains the publication records of authors to conferences from DBLP, and *Don2Com* and *Com2Cand* are from the U.S. Federal Election Commission in 2004[2], a public record of donations between donors and committees and between committees and political candidates, respectively.

For *Don2Com* and *Com2Cand*, the weights on the edges are actual weights representing donation amounts in dollars. For the remaining weighted datasets, the edge weights are simply the number of occurrences of the edges. For instance, if post i contains k links to another post j, the weight of the edge $e_{i,j}$ is set to k.

In our study, we specifically focused on undirected graphs, but the ideas can easily be generalized to directed graphs.

[2] Parsed dataset from all cycles can be found at `www.cs.cmu.edu/~mmcgloho/fec/data/fec_data.html`

Name	N	E	Weights	Structure	Description
Postnet	223K	217K	Yes	Unipartite	Network of posts based on citations
Auth2Conf	421K	1M	Yes	Bipartite	DBLP Author/Conference associations
Com2Cand	6K	125K	Yes	Bipartite	2004 US FEC Committee to Candidate donations
Don2Com	1,6M	2M	Yes	Bipartite	2004 US FEC Donor to Committee donations
Enron	36K	183K	No	Unipartite	Email associations at Enron
Oregon	11K	38K	No	Unipartite	AS peering connections

Table 1. Datasets studied in this work.

3 Proposed Method

Borrowing terminology from social network analysis (SNA), "ego" is an individual node.

Informally, an ego (=node) of a given network is anomalous if its neighborhood significantly differs from those of others. The basic research questions are: (a) *what features should we use to characterize a neighborhood?* and (b) *what does a 'normal' neighborhood look like?*

Both questions are open-ended, but we give some answers below. First, let's define terminology: the "k-step neighborhood" of node i is the collection of node i, all its k-step-away nodes, and all the connections among all of these nodes – formally, this is the "*induced sub-graph*". In SNA, the 1-step neighborhood of a node is specifically known as its "*egonet*".

How should we choose the value of k steps to study neighborhoods? Given that real-world graphs have small diameter [4], we need to stay with small values of k, and specifically, we recommend k=1. We report our findings only for k=1, because using $k > 1$ does not provide any more intuitive or revealing information, while it has heavy computational overhead, possibly intractable for very large graphs.

3.1 Feature Extraction

The first of our two inter-twined questions is *which statistics/features to extract* from a neighborhood.

There is an infinite set of functions/features that we could use to characterize a neighborhood (number of nodes, one or more eigenvalues, number of triangles, effective radius of the central node, number of neighbors of degree 1, etc etc). Which of all should we use?

Intuitively, we want to select features that (a) are fast-to-compute and (b) will lead us to patterns/laws that most nodes obey, except for a few anomalous nodes. We spend a lot of time experimenting with about a dozen features, trying to see whether the nodes of real graphs obey any patterns with respect to those

features (see our technical report [3]). The majority of features lead to no obvious patterns, and thus we do not present them.

The trimmed-down set of features that *are very* successful in spotting patterns, are the following:

1. N_i: number of neighbors (degree) of ego i,
2. E_i: number of edges in egonet i,
3. W_i: total weight of egonet i,
4. $\lambda_{w,i}$: principal eigenvalue of the *weighted* adjacency matrix of egonet i.

The next question is how to look for outliers, in such an n-dimensional feature space, with one point for each node of the graph. In our case, n=4, but one might have more features depending on the application and types of anomalies one wants to detect. A quick answer to this would be to use traditional outlier detection methods for clouds of points using all the features.

In our setting, we can *do better*. As we show next, we group features into carefully chosen pairs, where we show that there are patterns of normal behavior (typically, power-laws). We flag those points that significantly deviate from the discovered patterns as anomalous. Among the numerous pairs of features we studied, the successful pairs and the corresponding type of anomaly are the following:

- E vs N: *CliqueStar*: detects near-cliques and stars
- W vs E: *HeavyVicinity*: detects many recurrences of interactions
- λ_w vs W: *DominantPair*: detects single dominating heavy edge (strongly connected pair)

3.2 Laws and Observations

The second of our research questions is *what do normal neighborhoods look like*. Thus, it is important to find patterns ("laws") for neighborhoods of real graphs, and then report the deviations, if any. In this work, we report some new, surprising patterns:

Observation 1 (*EDPL*: Egonet Density Power Law) *For a given graph $\mathcal{G}$, node $i \in \mathcal{V}(\mathcal{G})$, and the egonet $\mathcal{G}_i$ of node i, the number of nodes N_i and the number of edges E_i of $\mathcal{G}_i$ follow a power law.*

$$E_i \propto N_i^{\alpha}, \quad 1 \leq \alpha \leq 2.$$

In our experiments the *EDPL* exponent α ranged from 1.10 to 1.66. Fig. 2 illustrates this observation, for several of our datasets. Plots show E_i versus N_i for every node (green points); the black circles are the median values for each bucket of points (separated by vertical dotted lines) after applying logarithmic binning on the x-axis as in [25]; the red line is the least squares(LS) fit on the median points. The plots also show a blue line of slope 2, that corresponds to cliques, and a black line of slope 1, that corresponds to stars. All the plots are in log-log scales.

Observation 2 (*EWPL*: Egonet Weight Power Law) *For a given graph $\mathcal{G}$, node $i \in \mathcal{V}(\mathcal{G})$, and the egonet $\mathcal{G}_i$ of node i, the total weight W_i and the number of edges E_i of $\mathcal{G}_i$ follow a power law.*

$$W_i \propto E_i^{\beta}, \quad \beta \geq 1.$$

Fig. 3 shows the *EWPL* for (only a subset of) our datasets (due to space limit). In our experiments the *EWPL* exponent β ranged up to 1.29. Values of $\beta > 1$ indicate super-linear growth in the total weight with respect to increasing total edge count in the egonet.

Observation 3 (*ELWPL*: Egonet λ_w Power Law) *the principal eigenvalue $\lambda_{w,i}$ of the weighted adjacency matrix and the total weight W_i of $\mathcal{G}_i$ follow a power law.*

$$\lambda_{w,i} \propto W_i^{\gamma}, \quad 0.5 \leq \gamma \leq 1.$$

Fig. 4 shows the *ELWPL* for a subset of our datasets. In our experiments the *ELWPL* exponent γ ranged from 0.53 to 0.98. γ=0.5 indicates uniform weight distribution whereas γ~1 indicates a dominant heavy edge, in which case the weighted eigenvalue follows the maximum edge weight. γ=1 if the egonet contains only one edge.

Observation 4 (*ERPL*: Egonet Rank Power Law) *the rank $R_{i,j}$ and the weight $W_{i,j}$ of edge j in $\mathcal{G}_i$ follow a power law.*

$$W_{i,j} \propto R_{i,j}^{\theta}, \quad \theta \leq 0.$$

The *ERPL* suggests that edge weights in the egonet have a skewed distribution. This is intuitive; for example in a friendship network, a person could have many not-so-close friends (light links), but only a few close friends (heavy links).

Next we show that if the *ERPL* holds, then the *EWPL* also holds. Given an egonet graph $\mathcal{G}_i$, the total weight W_i and the number of edges E_i of $\mathcal{G}_i$, let $\mathcal{W}_i$ denote the ordered set of weights of the edges, $W_{i,j}$ denote the weight of edge j, and $R_{i,j}$ denote the rank of weight $W_{i,j}$ in set $\mathcal{W}_i$. Then,

Lemma 1. *ERPL implies EWPL, that is: If $W_{i,j} \propto R_{i,j}^{\theta}$, $\theta \leq 0$, then*

$$W_i \propto E_i^{\beta} \begin{cases} \beta = 1, \text{ if } -1 \leq \theta \leq 0 \\ \beta > 1, \text{ if } \theta < -1 \end{cases}$$

Proof. For brevity, we give the proof for $\theta < -1$ – other cases are similar. Given that $W_{i,j} = cR_{i,j}^{\theta}$, $W_{min} = cE_i^{\theta}$, i.e. $c = W_{min}E_i^{-\theta}$. Then we can write W_i as

$$W_i = W_{min}E_i^{-\theta}\left(\sum_{j=1}^{E_i} j^{\theta}\right) \approx W_{min}E_i^{-\theta}\left(\int_{j=1}^{E_i} j^{\theta} dj\right)$$

$$= W_{min}E_i^{-\theta}\left(\frac{j^{\theta+1}}{\theta+1}\Big|_{j=1}^{E_i}\right) = W_{min}E_i^{-\theta}\left(\frac{1}{-\theta-1} - \frac{1}{(-\theta-1)E_i^{-\theta-1}}\right)$$

For large E_i and considering $\theta < -1$, the second term in parenthesis goes to 0. Therefore; $W_i \approx c'E_i^{-\theta}$, where $c' = \frac{W_{min}}{-\theta-1}$, and since $\theta < -1$, $\beta > 1$. □

3.3 Anomaly Detection

We can easily use the observations given in part 3.2 in anomaly detection since anomalous nodes would behave away from the normal pattern. Let us define the y-value of a node i as y_i and similarly, let x_i denote the x-value of node i for a particular feature pair $f(x, y)$. Given the power law equation $y = Cx^\theta$ for $f(x, y)$, we define the outlierness score of node i to be

$$\textit{out-line(i)} = \frac{max(y_i, Cx_i^\theta)}{min(y_i, Cx_i^\theta)} * log(|y_i - Cx_i^\theta| + 1)$$

Intuitively, the above measure is the "distance to fitting line". Here we penalize each node with both the *number of times* that y_i deviates from its *expected* value Cx_i^θ given x_i, and with the logarithm of the *amount* of deviation. This way, the minimum outlierness score becomes 0 –for which the actual value y_i is equal to the expected value Cx_i^θ.

This simple and easy-to-compute method not only helps in detecting outliers, but also provides a way to sort the nodes according to their outlierness scores. However, this method is prone to yield false positives for the following reason: Assume that there exists some points that are far away from the remaining points but that are still located close to the fitting line. In our experiments with real data, we observe that usually happens for high values of x and y. For example, in Fig. 2(a), the points marked with left-triangles ($\triangleleft$) are almost on the fitting line even though they are far away from the rest of the points.

We want to flag both types of points as outliers, and thus we propose to combine our heuristic with a density-based outlier detection technique. We used LOF [8], which also assigns outlierness scores *out-lof(i)* to data points; but any other outlier detection method would do, as long as it gives such a score. To obtain the final outlierness score of a data point i, one might use several methods such as taking a linear function of both scores and ranking the nodes according to the new score, or merging the two ranked lists of nodes, each sorted on a different score. In our work, we simply used the sum of the two normalized(by dividing by the maximum) scores, that is, *out-score(i) = out-line(i)+out-lof(i).*

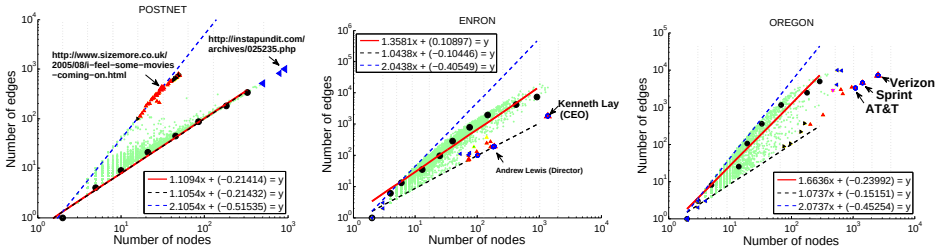
4 Experimental Results

CliqueStar Here, we are interested in the communities that the neighbors of a node form. In particular, *CliqueStar* detects anomalies having to do with near-cliques and near-stars. Using *CliqueStar*, we were successful in detecting many anomalies over the unipartite datasets (although it is irrelevant for bipartite graphs since by nature the egonet forms a "star").

In social media data *Postnet*, we detected posts or blogs that had either all their neighbors connected (cliques) or mostly disconnected (stars). We show some illustrative examples along with descriptions from *Postnet* in Fig. 1. See Fig.2a for the detected outliers on the scatter-plot from the same dataset.

In *Enron*(Fig.2b), the node with the highest anomaly score turns out to be "*Kenneth Lay*", who was the CEO and is best known for his role in the Enron scandal in 2001. Our method reveals that none of his over 1K contacts ever sent emails to each other.

In *Oregon* (Fig.2c), the top outliers are the three large ISPs ("*Verizon*", "*Sprint*" and "*AT&T*").



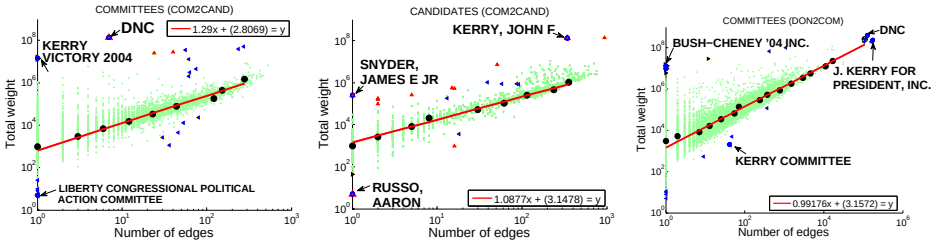
(a) *Postnet* (b) *Enron* (c) *Oregon*

Fig. 2. Illustration of the Egonet Density Power Law (*EDPL*), and the corresponding anomaly *CliqueStar*, with outliers marked with triangles. Edge count versus node count (log-log scale); red line is the LS fit on the median values (black circles); dashed black and blue lines have slopes 1 and 2 respectively, corresponding to stars and cliques. Most striking outlier: Ken Lay (CEO of Enron), with a star-like neighborhood. See Section 5.1.1 for more discussion and Fig.1 for example illustrations from *Postnet*.

HeavyVicinity In our datasets, *HeavyVicinity* detected "heavy egonets", with considerably high total edge weight compared to the number of edges. We mark the anomalies in Fig.3 for several of our datasets. See [3] for results on all the datasets and further discussions.

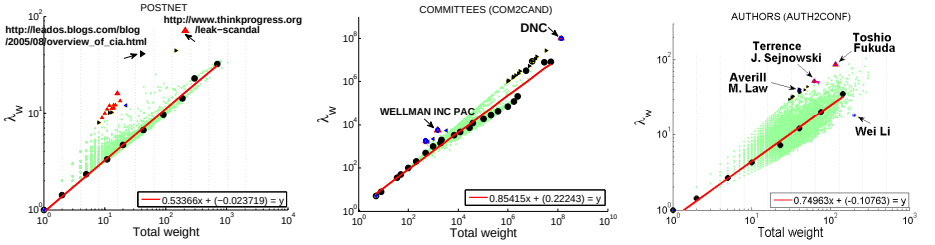
In *Com2Cand*(Fig.3a), we see that "*Democratic National Committee*" gave away a lot of money compared to the number of candidates that it donated to. In addition, "*(John) Kerry Victory 2004*" donated a large amount to a single candidate, whereas "*Liberty Congressional Political Action Committee*" donated a very small amount ($5), again to a single candidate. Looking at the *Candidates* plot for the same bipartite graph (Fig.3b), we also flagged "*Aaron Russo*", the lone recipient of that PAC. (In fact, Aaron Russo is the founder of the Constitution Party which never ran any candidates, and Russo shut it down after 18 months.)

In *Don2Com*(Fig.3c), we see that "*Bush-Cheney '04 Inc.*" received a lot of money from a single donor. On the other hand, we notice that the "*Kerry Committee*" received less money than would be expected looking at the number of checks it received in total. Further analysis shows that most of the edges in its egonet are of weight 0, showing that most of the donations to that committee have actually been returned.



(a) *Com2Cand* (b) *Com2Cand* (c) *Don2Com*

Fig. 3. Illustration of the Egonet Weight Power Law (*EWPL*) and the weight-edge anomaly *HeavyVicinity*. Plots show total weight vs. total count of edges in the egonet for all nodes (in log-log scales). Detected outliers include Democratic National Committee and John F. Kerry (in FEC campaign donations). See Section 5.2.1 for more discussions.



(a) *Postnet* (b) *Com2Cand* (c) *Auth2Conf*

Fig. 4. Illustration of the Egonet λ_w Power Law (*ELWPL*) and the dominant heavy link anomaly *DominantPair*. Top anomalies are marked with triangles and labeled. See Section 5.2.2 for detailed discussions for each dataset and Fig.1 for an illustrative example from *Com2Cand*.

DominantPair Here, we find out whether there is a single dominant heavy edge in the egonet. In other words, this method detected "bursty" if not exclusive edges.

In *Postnet*(Fig.4a) nodes such as "*ThinkProgress*"'s post on a leak scandal[3] and "*A Freethinker's Paradise*" post[4] linking several times to the "*ThinkProgress*" post were both flagged. On another note, the slope of the fitting line is close to 0.5, pointing to uniform weight distribution in egonets overall. This is expected as most posts link to other posts only once.

In *Com2Cand*(Fig.4b), "*Democratic National Committee*" is one of the top outliers. We would guess that the single large amount of donation was made to "*John F. Kerry*". Counterintuitively, however, we see that that amount was spent for an opposing advertisement against "*George W. Bush*".

[3] `www.thinkprogress.org/leak-scandal`
[4] `leados.blogs.com/blog/2005/08/overview_of_cia.html`

DominantPair flagged extremely focused authors (those publish heavily to one conference) in the DBLP data, shown in Fig.3c. For instance, "*Toshio Fukuda*" has 115 papers in 17 conferences (at the time of data collection), with more than half (87) of his papers in one particular conference (ICRA). In addition, "*Averill M. Law*" has 40 papers published to the "*Winter Simulation Conference*" and nowhere else. On the other extreme, another interesting point is "*Wei Li*", with many papers, who gets them published to as many distinct conferences, probably once or twice to each conference (uniform rather than 'bursty' distribution).

See [3] for results on all the datasets and further discussions.

5 Related Work

5.1 Outlier Detection

Outlier detection has attracted wide interest, being a difficult problem, despite its apparent simplicity. Even the definition of the outlier is hard to give: For instance, Hawkins [18] defines an outlier as "an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism." Similar, but not identical, definitions have been given by Barnett and Lewis [6], and Johnson [21].

Outlier detection methods form two classes, *parametric* (statistical) and *non-parametric* (model-free). The former includes statistical methods that assume prior knowledge of the underlying data distribution [6, 18]. The latter class includes *distance-based* and *density-based* data mining methods. These methods typically define as an outlier the (n-D) point that is too far away from the rest, and thus lives in a low-density area [22]. Typical methods include LOF [8] and LOCI [29]. These methods not only flag a point as an outlier but they also give outlierness scores; thus, they can sort the points according to their "strangeness". Many other *density-based* methods especially for large high-dimensional data sets are proposed in [1, 5, 13, 17]. Finally, most clustering algorithms [11, 19, 27] reveal outliers as a by-product.

5.2 Anomaly Detection in Graph Data

Noble and Cook [28] detect anomalous sub-graphs using variants of the *Minimum Description Length* (MDL) principle. Eberle and Holder [15] use MDL as well as other probabilistic measures to detect several types of anomalies (e.g. unexpected/missing nodes/edges). Frequent subgraph mining [20, 32] is used to detect non-crashing bugs in software flow graphs [24]. Chakrabarti [9] uses MDL to spot anomalous edges. Sun et al. [31] use proximity and random walks, to assess the normality of nodes in bipartite graphs. OutRank and LOADED [16, 26] use similarity graphs of objects to detect outliers.

In contrast to the above, we work with *unlabeled* graphs. We explicitly focus on nodes, while interactions are also considered implicitly as we study *neighborhood sub-graphs*. Finally, we consider both bipartite and unipartite graphs as well as edge *weights*.

5.3 Real-world graph properties

Several properties of real-world unweighted graphs have been discovered, surveyed in [10]. In our study, we explicitly concentrate on anomalous node detection in *weighted* graphs, adding to laws of weighted graphs discovered in [2, 25] and relying on these observations to detect anomalies.

6 Conclusion

This is one of the few papers that focus on anomaly detection in graph data, including weighted graphs. We propose to use "egonets", that is, the induced subgraph of the node of interest and its neighbors; and we give a small, carefully designed list of numerical features for egonets. The major contributions are the following:

1. Discovery of new patterns that egonets follow, such as patterns in density (Obs.1: *EDPL*), weights (Obs.2: *EWPL*), principal eigenvalues (Obs.3: *EL-WPL*), and ranks (Obs.4: *ERPL*). Proof of Lemma 1, linking the *ERPL* to the *EWPL*.
2. `OddBall`, a fast, un-supervised method to detect abnormal nodes in weighted graphs. Our method does not require any user-defined constants. It also assigns an "outlierness" *score* to each node.
3. Experiments on real graphs of over 1M nodes, where `OddBall` reveals nodes that indeed have strange or extreme behavior.

Future work could generalize `OddBall` to time-evolving graphs, where the challenge is to find patterns that neighborhood sub-graphs follow and to extract features incrementally over time.

References

1. Charu C. Aggarwal and Philip S. Yu. Outlier detection for high dimensional data. In *SIGMOD*, pages 37–46, 2001.
2. Leman Akoglu, Mary McGlohon, and Christos Faloutsos. RTM: Laws and a recursive generator for weighted time-evolving graphs. In *ICDM*, 2008.
3. Leman Akoglu, Mary McGlohon, and Christos Faloutsos. Anomaly detection in large graphs. In *CMU-CS-09-173 Technical Report*, 2009.
4. Reka Albert, Hawoong Jeong, and Albert-Laszlo Barabasi. Diameter of the world wide web. *Nature*, (401):130–131, 1999.
5. Andreas Arning, Rakesh Agrawal, and Prabhakar Raghavan. A linear method for deviation detection in large databases. In *KDD*, pages 164–169, 1996.
6. V. Barnett and T. Lewis. *Outliers in Statistical Data*. John Wiley and Sons, Chichester, New York, 1994.
7. Stephen Bay, Krishna Kumaraswamy, Markus G. Anderle, Rohit Kumar, and David M. Steier. Large scale detection of irregularities in accounting data. In *ICDM*, 2006.
8. Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. Lof: Identifying density-based local outliers. In *SIGMOD*, pages 93–104, 2000.

9. Deepayan Chakrabarti. Autopart: Parameter-free graph partitioning and outlier detection. In *PKDD*, pages 112–124, 2004.
10. Deepayan Chakrabarti and Christos Faloutsos. Graph mining: Laws, generators, and algorithms. *ACM Comput. Surv.*, 38(1), 2006.
11. Vineet Chaoji, Mohammad Al Hasan, Saeed Salem, and Mohammed J. Zaki. Sparcl: Efficient and effective shape-based clustering. In *ICDM*, 2008.
12. Duen Horng Chau, Shashank Pandit, and Christos Faloutsos. Detecting fraudulent personalities in networks of online auctioneers. *PKDD*, 2006.
13. Amitabh Chaudhary, Alexander S. Szalay, and Andrew W. Moore. Very fast outlier detection in large multidimensional data sets. In *DMKD*, 2002.
14. Tamraparni Dasu and Theodore Johnson. *Exploratory Data Mining and Data Cleaning*. Wiley-Interscience, May 2003.
15. William Eberle and Lawrence B. Holder. Discovering structural anomalies in graph-based data. In *ICDM Workshops*, pages 393–398, 2007.
16. Amol Ghoting, Matthew Eric Otey, and Srinivasan Parthasarathy. Loaded: Link-based outlier and anomaly detection in evolving data sets. In *ICDM*, 2004.
17. Amol Ghoting, Srinivasan Parthasarathy, and Matthew Eric Otey. Fast mining of distance-based outliers in high-dimensional datasets. *Data Min. Knowl. Discov.*, 16(3):349–364, 2008.
18. D. Hawkins. Identification of outliers. *Chapman and Hall*, 1980.
19. Tianming Hu and Sam Yuan Sung. Detecting pattern-based outliers. *Pattern Recognition Letters*, 24(16), 2003.
20. Ruoming Jin, Chao Wang, Dmitrii Polshakov, Srinivasan Parthasarathy, and Gagan Agrawal. Discovering frequent topological structures from graph datasets. In *KDD*, 2005.
21. R. A. Johnson and D. W. Wichern. *Applied Multivariate Statistical Analysis*. Prentice Hall, 1998.
22. Edwin M. Knorr and Raymond T. Ng. Algorithms for mining distance-based outliers in large datasets. In *VLDB*, pages 392–403, 1998.
23. Jure Leskovec, Mary McGlohon, Christos Faloutsos, Natalie Glance, and Matthew Hurst. Cascading behavior in large blog graphs: Patterns and a model. In *Society of Applied and Industrial Mathematics: Data Mining*, 2007.
24. Chao Liu, Xifeng Yan, Hwanjo Yu, Jiawei Han, and Philip S. Yu. Mining behavior graphs for "backtrace" of noncrashing bugs. In *SDM*, 2005.
25. Mary McGlohon, Leman Akoglu, and Christos Faloutsos. Weighted graphs and disconnected components: Patterns and a model. In *ACM SIGKDD*, 2008.
26. H. D. K. Moonesinghe and Pang-Ning Tan. Outrank: a graph-based outlier detection framework using random walk. *International Journal on Artificial Intelligence Tools*, 17(1), 2008.
27. Raymond T. Ng and Jiawei Han. Efficient and effective clustering methods for spatial data mining. In *VLDB*, pages 144–155, 1994.
28. Caleb C. Noble and Diane J. Cook. Graph-based anomaly detection. In *KDD*, pages 631–636, 2003.
29. Spiros Papadimitriou, Hiroyuki Kitagawa, Phillip B. Gibbons, and Christos Faloutsos. Loci: Fast outlier detection using the local correlation integral. In *ICDE*, 2003.
30. Karlton Sequeira and Mohammed Javeed Zaki. Admit: anomaly-based data mining for intrusions. In *KDD*, 2002.
31. Jimeng Sun, Huiming Qu, Deepayan Chakrabarti, and Christos Faloutsos. Neighborhood formation and anomaly detection in bipartite graphs. *ICDM*, 2005.
32. Xifeng Yan and Jiawei Han. gspan: Graph-based substructure pattern mining. In *ICDM*, 2002.