

OddBall: Spotting Anomalies in Weighted Graphs

Leman Akoglu, Mary McGlohon, and Christos Faloutsos

Carnegie Mellon University, School of Computer Science
{lakoglu,mmcgho,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 these rules 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 [6], extremely cross-disciplinary authors in an author-paper graph [29], network intrusion detection [28], electronic auction fraud [10], 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 [12]. 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 [2], 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 a function of n . Extreme total weight for a given number of contacts n 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