Detecting Aberrant Linking Behavior in Directed Networks

Dorit S. Hochbaum, Quico Spaen and Mark Velednitsky
Unversity of California, Berkeley, U.S.A.

Keywords: Aberrant Linking Behavior, Classification, Markov Random Fields, Directed Networks, Modularity, Spam Detection, Fake News, Parametric Minimum Cut.

Abstract: Agents with aberrant behavior are commonplace in today's networks. There are fake profiles in social media, malicious websites on the internet, and fake news sources that are prolific in spreading misinformation. The distinguishing characteristic of networks with aberrant agents is that normal agents rarely link to aberrant ones. Based on this manifested behavior, we propose a directed Markov Random Field (MRF) formulation for detecting aberrant agents. The formulation balances two objectives: to have as few links as possible from normal to aberrant agents, as well as to deviate minimally from prior information (if given). The MRF formulation is solved optimally and efficiently. We compare the optimal solution for the MRF formulation to existing algorithms, including PageRank, TrustRank, and AntiTrustRank. To assess the performance of these algorithms, we present a variant of the modularity clustering metric that overcomes the known shortcomings of modularity in directed graphs. We show that this new metric has desirable properties and prove that optimizing it is NP-hard. In an empirical experiment with twenty-three different datasets, we demonstrate that the MRF method outperforms the other detection algorithms.

1 INTRODUCTION

Agents with aberrant behavior are commonplace in today's networks. There are malicious websites on the internet, fake profiles in social media, and fake news sources prolific in spreading misinformation. There is considerable interest in detecting such aberrant agents in networks. Across contexts, the unifying theme is that normal agents rarely link to aberrant ones. We call this property *aberrant linking behavior*. This behavior occurs in a number of different contexts.

Aberrant linking behavior was observed in web graphs in the early days of search engines. In these web graphs, the goal is to separate informative websites (normal) from spam or malicious websites (aberrant). Informative websites typically link to other relevant and informative websites, whereas spam websites link to either informative websites or other spam websites. Spam websites linking to each other creates "link farms" (Wu and Davison, 2005; Castillo et al., 2007). From the linking behavior of the websites, the expected structure of normal and aberrant agents, with aberrant linking behavior, arises. In one empirical study, this structure of normal and spam sites is verified in a Japanese web graph with 5.8 million sites and 283 million links (Saito et al., 2007).

In social networks, the goal is to separate real profiles (normal) from fake profiles (aberrant). On Facebook, it is estimated that nearly 10% of accounts are either fake or otherwise "undesirable" (intentionally spreading misinformation) (Fire et al., 2014). In an empirical study of $40,000$ fake Twitter accounts, it was observed that most users of authentic social media accounts avoid following fake accounts (Ghosh et al., 2012). A similar study of $250,000$ fake accounts on LinkedIn corroborated this result (Xiao et al., 2015). It is evident that the expected structure of normal and aberrant agents, with aberrant linking behavior, arises in these settings.

In the context of fake news, the goal is to separate credible sources (normal) from dubious ones (aberrant) (Shu et al., 2017; Törnberg, 2018; Shu et al., 2019). Credible sources typically link to other credible sources and will not link to dubious ones. Thus, the expected structure of normal and aberrant agents, with aberrant linking behavior, arises. In (Shu et al., 2017), the authors observe that the credibility of a news event is highly related to the credibility of the sources it references. In (Shu et al., 2019) and (Törnberg, 2018), the authors use the colloquial term "echo chamber" to describe small networks of agents that amplify the spread of false information by repeating it.

Hochbaum, D., Spaen, Q. and Velednitsky, M.
Detecting Aberrant Linking Behavior in Directed Networks.
DOI: 10.5220/0008069600720082
In *Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2019)*, pages 72-82
ISBN: 978-989-758-382-7

The problem of aberrant agent detection is formalized here as a classification problem on a directed graph, where each vertex represents an agent and each arc represents a link from one agent to another. In contrast to an agent-based approach that classifies an agent based on its individual features (Ntoulas et al., 2006; Webb et al., 2008; Erdélyi et al., 2011), we use information about the graph links to perform the classification (Becchetti et al., 2006; Gan and Suel, 2007). These agent-based and link-based approaches are synergistic (Becchetti et al., 2008; Roul et al., 2016). For example, the output of an agent-based learning algorithm can be incorporated in a link-based approach as prior scores for the agents. Multiple link-based techniques have been proposed in the literature, including spectral methods and random walks.

Spectral techniques for classifying aberrant agents typically optimize a symmetric objective function, which is the sum of a measure of disagreement between adjacent agents (Von Luxburg, 2007; Wu and Chellapilla, 2007; Zhou et al., 2007). Such methods are sometimes called "graph regularization methods", since they interpolate missing labels from known ones (Abernethy et al., 2010). In other work, the objective functions are NP-hard cut problems, such as normalized cut (Shi and Malik, 2000) or generalized weighted cut (Meilă and Pentney, 2007). Since these cut problems are NP-Hard, they are approximated with spectral methods on an appropriately chosen symmetric matrix. After the spectral algorithm returns a partition, refinements may be made with pairwise swaps until a local optimum is reached (Malliaros and Vazirgiannis, 2013).

Random-walk based approaches include PageRank (Page et al., 1999), TrustRank (Gyöngyi et al., 2004), AntiTrustRank (Krishnan and Raj, 2006), and several domain-specific variants which have been proposed over the years (Gori and Pucci, 2006; Wu and Chellapilla, 2007; Liu et al., 2009; Rosvall and Bergstrom, 2008; Sayyadi and Getoor, 2009; Shu et al., 2019). If a set of agents is highly internally connected and minimally externally connected, a random walk starting in that set is expected to spend a lot of time inside it before exiting. Thus, in a graph with aberrant linking behavior, a random walk is expected to visit the normal agents more frequently than the aberrant agents (Rosvall and Bergstrom, 2008). In some cases, the behavior of random walks is equivalent to optimizing an explicit objective function. In other cases, no objective function is specified. We describe the mechanics of these algorithms in detail in section 2.

In this paper, we propose a new approach that formulates the aberrant agent detection problem for the first time as a Markov Random Field (MRF) problem (Geman and Geman, 1984). Others have considered an MRF formulation for the related problem of detecting campaign promoters in social media, but used loopy belief propagation instead of solving for an optimal solution (Li et al., 2014). Our formulation balances obeying any given prior information with minimizing the number of links from normal to aberrant agents. The optimal solution to the formulation is obtained efficiently with known algorithms (Hochbaum, 2001). One advantage of the proposed formulation is its ability to be given prior labels for the agents with various degrees of confidence.

In an extensive empirical study, we compare MRF with three well-known algorithms from the literature: PageRank (Page et al., 1999), TrustRank (Gyöngyi et al., 2004), and AntiTrustRank (Krishnan and Raj, 2006). We also use a random classifier as a baseline algorithm. We find that MRF outperforms its competitors. The average score of MRF is approximately 25% higher than the next-best algorithm.

The main contributions of this work are:

1. We formulate the problem of detecting aberrant linking behavior as a directed Markov Random Field (MRF) problem. The formulation is solved optimally and efficiently. This is the first time that MRF is used to model the detection of aberrant agents and the first use of MRF for directed graphs.
2. We develop a new variant of the modularity metric (Newman and Girvan, 2004) that addresses its known shortcomings on directed graphs. We show that our metric has desirable properties and prove that optimizing it is NP-hard.
3. We present an extensive empirical study in which we compare MRF with three well-known algorithms from the literature.

The rest of this paper is organized as follows: in section 2, we present preliminaries, including the details of the existing algorithms. In section 3, we present our MRF formulation. In section 4, we describe the methods we used to evaluate the quality of the results of the algorithms. In section 5 and section 6 we describe the experimental setup and results. Finally, we conclude the manuscript in section 7.

2 PRELIMINARIES

Notation. We represent the network as a directed graph $G = (V, A)$, where V is the set of vertices and A is the set of arcs. Each node in the graph G represents an agent, and an arc represents a link from one agent to another. Each arc $(i, j) \in A$ has an associated $w_{ij} \in \mathbb{R}^+$, which represents the number of links from agent i to

agent j. We use d_i^{out} and d_i^{in} to denote the weighted out-degree and in-degree of vertex i, respectively.

We will ultimately partition V into two sets: C_0, the set of "normal" vertices and C_1, the set of "aberrant" vertices. The notation W_{pq} denotes the total weight of arcs from C_p to C_q. That is,

$$W_{pq} = \sum_{i \in C_p, j \in C_q} w_{ij}.$$

We also define $W = \sum_{(i,j) \in A} w_{ij}$ to denote to the total sum of weights.

Priors. All the algorithms described here will take as input a graph in which some or all of the vertices have prior values associated with them. We use $V_{\text{prior}} \subseteq V$ to denote the set of vertices which have priors, and we use $c_i \in [0,1]$ to denote the prior value of vertex i. These priors could represent information about known vertices, the ratings of human judges or the output of another algorithm. The prior values do *not* represent "ground truth". The priors may have various degrees of confidence and may occasionally be imprecise or unreliable. Besides the structure of the graph itself, the priors are the only information we have on which to base our final classifications.

Output. All algorithms output a continuous score, $x_i \in [0,1]$, for every vertex $i \in V$, where closer to 0 means more likely normal and closer to 1 means more likely aberrant. From these continuous scores, we decide how to partition V into C_0 and C_1.

Existing Algorithms We consider the three existing algorithms that are most prevalent in the literature: PageRank, TrustRank, and AntiTrustRank. These algorithms and their variants are actively used (Tagarelli and Interdonato, 2014, 2018).

In PageRank (Page et al., 1999), a Markov chain is defined over the vertices in the graph. The trust score of vertex $i \in V$ is equal to the stationary probability that the Markov chain is in state i. The state transitions from vertex i to vertex j with probability

$$P_{ij} = \alpha \frac{w_{ij}}{d_i^{out}} + (1-\alpha) r_j \quad \forall (i,j) \in V \times V.$$

Here, $\alpha \in [0,1]$ is a hyperparameter known as the *attenuation factor* and $r_j \in [0,1]$ is the probability of *restarting* the Markov chain from vertex j. The values of r_j must sum to 1: $\sum_{j \in V} r_j = 1$.

In PageRank, $r_j = \frac{1}{n} \forall j \in V$. The unique eigenvector with eigenvalue 1, π, is computed with the Power method. The score, x_i, returned by PageRank, is equal to $1 - \pi_i$.

TrustRank (Gyöngyi et al., 2004) is a modification of PageRank where the probability for (re)starting at vertex j is proportional to $1 - c_j$ (a measure of how "trusted" the vertex is):

$$r_j^{trust} = \frac{1 - c_j}{\sum_{i \in V_{\text{prior}}} (1 - c_i)} \quad \forall j \in V_{\text{prior}}.$$

The intuition behind TrustRank is that from trusted nodes you should only reach other trusted nodes. Like PageRank, the returned score is $1 - \pi_i$, where π_i is the probability that the Markov chain is in state i.

AntiTrustRank (Krishnan and Raj, 2006) is similar to TrustRank, but differs in two aspects: The Markov chain traverses the graph in the reverse direction, and the (re)start distribution is proportional to c_j (a measure of how "distrusted" the vertex is):

$$r_j^{anti} = \frac{c_j}{\sum_{i \in V_{\text{prior}}} c_i} \quad \forall j \in V_{\text{prior}}.$$

The underlying idea is that from normal vertices you should rarely reach aberrant vertices, and thus if you follow arcs in the reverse direction, then from aberrant vertices you should reach mostly aberrant vertices.

In contrast to PageRank and TrustRank, in AntiTrustRank the eigenvector with eigenvalue 1 represents the *distrust* of the vertices. The score, x_i, returned by AntiTrustRank for vertex $i \in V$ is equal to π_i.

3 MARKOV RANDOM FIELD (MRF) MODEL

The MRF model (Geman and Geman, 1984) is defined for a directed graph $G = (V, A)$, where each vertex $i \in V$ has an associated decision variable x_i. For given *deviation* functions G_i and *separation* functions F_{ij}, the MRF model is defined as:

$$\begin{aligned} \min \quad & \lambda \sum_{i \in V} G_i(x_i, c_i) + \sum_{(i,j) \in A} F_{ij}(x_i - x_j) \\ \text{s.t.} \quad & l_i \le x_i \le u_i \quad \forall i \in V \end{aligned}$$

The deviation function $G_i(\cdot,\cdot)$ penalizes a deviation of the variable x_i away from the prior value c_i, whereas the separation function $F_{ij}(\cdot)$ penalizes the difference between the values assigned to neighboring vertices in the graph. The *trade-off* parameter $\lambda \ge 0$ determines the trade-off between the deviation and separation penalties. When $G_i(\cdot,\cdot)$ and $F_{ij}(\cdot)$ are convex, this problem is solved optimally and efficiently in either continuous or integer variables (Hochbaum, 2001; Ahuja et al., 2003). The problem is NP-hard otherwise.

For the problem of detecting aberrant agents, a "good" solution is characterized by two properties.

First, there should be few links from normal to aberrant agents. Second, the difference between the score assigned to an agent and its prior value, if any, should be small. These goals naturally map to an MRF model.

For each vertex $i \in V_{prior}$, with associated prior c_i, we choose a quadratic penalty function $G_i = (x_i - c_i)^2$ to measure the deviation between the assigned x_i and the prior c_i. The remaining vertices without priors do not have an associated deviation penalty. That is, $G_i(x_i, c_i) = 0$ for $i \notin V_{prior}$. For each arc $(i, j) \in A$ with weight w_{ij}, we have a separation function $F_{ij} = w_{ij}(x_j - x_i)^+$, where $(x_j - x_i)^+ = \max\{x_j - x_i, 0\}$. This separation function results in a penalty of $w_{ij}(x_j - x_i)$ for arc $(i, j) \in A$ if the score x_i of vertex i is lower than the score x_j of vertex j, since the (more) normal vertex i links to a (more) aberrant vertex j.

The resulting optimization problem is:

$$\min \quad \lambda \sum_{i \in V_{\text{prior}}} (x_i - c_i)^2 + \sum_{(i,j) \in A} w_{ij}(x_j - x_i)^+ \qquad \text{(MRF-Detection)}$$

$$\text{s.t.} \quad 0 \le x_i \le 1 \quad \forall i \in V.$$

In the optimization problem, each agent $i \in V$ is assigned a score $x_i \in [0, 1]$. The behavior of an agent with a score of 1 is considered aberrant, whereas a score of 0 corresponds to normal behavior.

(MRF-Detection) is a special case of the MRF problem with convex deviations and bi-linear separation functions[1], which was shown by Hochbaum (2001) to be solvable with a parametric minimum cut problem in the complexity of a single minimum cut problem plus the complexity required to find the minima of the convex deviation functions. Two parametric cut algorithms, based on the pseudoflow algorithm (Hochbaum, 2008; Hochbaum and Orlin, 2013) or the push-relabel algorithm (Goldberg and Tarjan, 1988; Gallo et al., 1989), achieve this complexity. Since the deviation functions are quadratic here, the complexity of finding the minima of the deviation functions is $O(|V|)$, which is dominated by the complexity of a minimum cut problem. As a result, the complexity of solving this parametric minimum cut problem with either of these two algorithms is expressible as $O\left(mn \log \frac{n^2}{m}\right)$ where n is the number of nodes in the graph and m is the number of arcs.

These results imply that we can solve the MRF formulation for aberrant agent detection efficiently and optimally.

[1] A bi-linear function refers here to a function of the form $\max\{u^+ z, -u^- z\}$ for $u^+, u^- \ge 0$ and $z \in \mathbb{R}$.

4 PERFORMANCE EVALUATION METRICS

In the datasets available to us, there is no "ground truth" to which we can compare our results. Instead we assess the classification performance in terms of how well the resulting partition obeys the property of having few links from normal vertices to aberrant vertices.

For this purpose, we will lay out several metrics. The first are a series of ad-hoc metrics, such as the average out-degree from normal vertices to aberrant ones. We also present a directed variant of the established modularity clustering metric (Newman and Girvan, 2004).

4.1 Ad-hoc Metrics

Aberrant linking behavior implies that there should be few arcs from normal to aberrant. For that reason, a relevant metric is $\frac{W_{01}}{N_0}$, the average degree of a normal vertex to the set of aberrant vertices.

As a baseline for comparison, we also calculate $\frac{W_{11}}{N_1}$, the average degree of an aberrant vertex to the set of aberrant vertices. If our labeling has the desired property, then we expect that $\frac{W_{01}}{N_0}$ will be significantly smaller than $\frac{W_{11}}{N_1}$. For further comparison, we also compute $\frac{W_{01}}{W_{01}+W_{11}}$, the fraction of weight into aberrant vertices coming from normal vertices.

In order to normalize these values of across graphs, we divide the degree measures by the average degree of the graph, $d_{\text{avg}} = \frac{W_{00}+W_{01}+W_{10}+W_{11}}{N_0+N_1}$.

This leads to the following three metrics:

$$\frac{W_{01}/N_0}{d_{\text{avg}}}, \quad \frac{W_{11}/N_1}{d_{\text{avg}}}, \text{ and } \frac{W_{01}}{W_{01}+W_{11}}.$$

4.2 Modularity

A metric commonly used in the graph partitioning literature is *modularity* (Newman and Girvan, 2004). It measures how many edges are within clusters versus edges between clusters and compares that to a random graph with the same degree distribution (Newman and Girvan, 2004; Kim et al., 2010).

Given a weighted, undirected graph and a partition of the set of vertices into clusters $C_0, \ldots, C_k$, modularity (Newman and Girvan, 2004) is defined as

$$\frac{1}{2W} \sum_{i,j \in V} \left[w_{ij} - \frac{d_i d_j}{2W} \right] \delta(i, j). \qquad \text{(UndirMod)}$$

Here, d_i is the weighted degree of vertex i, and

$$\delta(i,j) = \begin{cases} 1 & \exists p \mid i,j \in C_p, \\ 0 & \text{otherwise.} \end{cases}$$

A larger modularity value indicates that the cluster assignment is superior since there are more edges within the clusters than in a random graph. It is NP-hard to find the set of clusters that maximize modularity in a graph (Brandes et al., 2006).

When modularity is applied to directed graphs, there is no agreed-upon generalization (Newman, 2006; Rosvall and Bergstrom, 2008; Kim et al., 2010). One straightforward adaptation of modularity to directed graphs would be to calculate (Newman, 2006; Rosvall and Bergstrom, 2008; Kim et al., 2010)

$$\frac{1}{W}\sum_{i,j\in V}\left[w_{ij} - \frac{d_i^{\text{out}} d_j^{\text{in}}}{W}\right]\delta(i,j). \qquad \text{(DirMod)}$$

Several issues with this generalization have been observed. In (Malliaros and Vazirgiannis, 2013), small example graphs are shown in which certain arcs can be reversed without affecting the modularity. This is problematic given our interest in asymmetry between clusters.

In fact, we show in claim 1 that this definition of directed modularity, with 2 clusters, is proportional to the determinant of a matrix with entries W_{ij} for $i,j \in \{0,1\}$. We defer the proof to the appendix. This result implies that this definition is symmetric with respect to the cluster assignment and that the cluster labels are exchangeable without affecting the modularity.

Claim 1. *Let $G = (V,A)$ be a directed graph with vertex labels in $\{0,1\}$, defining two clusters C_0 and C_1. The modularity of the clustering assignment on G, as defined in equation* (DirMod)*, is proportional to*

$$W_{00}W_{11} - W_{01}W_{10}.$$

Corollary 1. *Let $G = (V,A)$ be a directed graph with vertex labels in $\{0,1\}$, defining a cluster assignment $\mathcal{C}$. Let $\mathcal{C}'$ be the assignment with opposite labels. Then, the modularity of cluster assignment $\mathcal{C}$ on G, as defined in equation* (DirMod)*, is the same as the modularity of assignment $\mathcal{C}'$.*

The symmetry with respect to the clustering labels is undesirable for the problem of agent detection, since only links from a normal vertex to an aberrant one should be penalized. That is, we would like to penalize arcs from C_0 to C_1 without penalizing arcs from C_1 to C_0. If the labels can be interchanged, then these penalties are necessarily symmetric. We will propose a change to the modularity metric which overcomes this deficiency.

One option might be to change the definition δ so that $\delta(1,0) = \delta(0,0) = \delta(1,1) = 1$. In other words, "rewarding" arcs from C_1 to C_0 in addition to arcs within clusters. However, repeating the calculation in Claim 1 gives a surprising result: even with the new definition of δ, the modularity metric remains proportional to $W_{00}W_{11} - W_{01}W_{10}$. For that reason, a different change is needed.

We suggest a new variant of the directed modularity metric, which captures the asymmetric nature of the relation between normal and aberrant vertices. Our metric only penalizes arcs in one direction between the two clusters. We keep the $W_{00}W_{11}$ term, but instead of subtracting off $W_{01}W_{10}$, we subtract off $\frac{3}{4}W_{01}^2$. We add the $\frac{3}{4}$ coefficient to account for the larger expected value of the term W_{01}^2 as compared to $W_{01}W_{10}$[2]. Our new definition of directed modularity in the two-cluster case is:

$$\frac{4\left(W_{00}W_{11} - \frac{3}{4}W_{01}^2\right)}{W^2}. \qquad \text{(AsymMod)}$$

Similarly to the result for undirected modularity, we establish that maximizing AsymMod is NP-hard. Our reduction is from the minimum bisection problem, which is different from the undirected case. Again, we defer the proof to the appendix.

Claim 2. *Maximizing* (AsymMod) *is NP-Hard.*

From now on, when we refer to modularity, it is assumed that we are referring to (AsymMod).

5 EXPERIMENTAL SETUP

We compare MRF against the algorithms PageRank (Page et al., 1999), TrustRank (Gyöngyi et al., 2004), AntiTrustRank (Krishnan and Raj, 2006), and a randomized baseline algorithm, which we name Random. We measure the performance of the algorithms in terms of modularity score and the ad-hoc metrics described in section 4.

Datasets. We evaluate the experimental performance of MRF on twenty-one different datasets from the KONECT project (Kunegis, 2013), as well as two Web Spam datasets (Castillo et al., 2006). The datasets in KONECT are categorized into twenty-three categories. We chose twenty-one datasets in categories in which we plausibly expected to find high-modularity clusters. The two Web Spam datasets are based on web crawls of the .uk top-level domain. The properties of

[2]The coefficient $\frac{3}{4}$ is chosen such that the terms have equal expectation when the W_{pq}/W values are drawn independently and uniformly from the unit interval.

the datasets are summarized in Table 1. We excluded from our analysis four Konect datasets for which no algorithm was able to attain a modularity score above 0.2.

Priors. Since we do not have access to datasets with prior labels, we generate priors with a simple heuristic. For each dataset, we assign priors such that $p_{prior} \leq 50\%$ percent of the vertices are marked aberrant and the same number of vertices are marked normal. The vertices are chosen according to the difference between out-degree and in-degree, since we expect that nodes with high out-degree and low in-degree are potentially aberrant. We define $\Delta_i = d_i^{out} - d_i^{in}$ for every vertex $i \in V$. The vertices are then sorted such that $\Delta_{\sigma(1)} \geq \Delta_{\sigma(2)} \geq \cdots \geq \Delta_{\sigma(n)}$. The first $n_{\text{prior}} = \lfloor p_{prior}\, n \rfloor$ vertices with the highest Δ value are considered aberrant and given the prior values $c_i = 1$. The last n_{prior} vertices with the lowest Δ value are considered normal and given the prior values $c_i = 0$.

Labeling. Each of the algorithms returns a score $x_i \in [0,1]$ for every vertex $i \in V$. We convert these scores into a binary assignment to the clusters C_0 and C_1 by means of a threshold τ. The clusters are given by:

$$C_1(\tau) = \{i \in V : x_i \geq \tau\}$$
$$C_0(\tau) = \{i \in V : x_i < \tau\}$$

We report the highest modularity obtained over a set of thresholds T:

$$\max_{\tau \in T} \quad \text{Modularity}(C_0(\tau), C_1(\tau))$$

For the MRF algorithm, the set of thresholds T is given by the set of unique values of the scores $\{x_i : i \in V\}$. For the other algorithms, the set of threshold values T consists of the $0^{\text{th}}, 5^{\text{th}}, \ldots, 100^{\text{th}}$ percentiles of the scores $\{x_i : i \in V\}$. We apply a different method for MRF since its solution typically contains only a few distinct values. In the results reported here, the median number of unique scores in the MRF solution is 13. In contrast, the node scores tend to be unique for each of the other algorithms.

Algorithm Implementations. For MRF, we use the formulation as described in equation (MRF-Detection) with one modification: we normalize the trade-off parameter λ. We call this new hyperparameter the normalized trade-off parameter, λ_{norm}. It is defined by the following equation:

$$\lambda = \lambda_{norm} \frac{\sum_{(i,j) \in A} w_{ij}}{|V_{prior}|}.$$

We expect that the normalized parameter λ_{norm} is more consistent across datasets, since it corrects for the size of the graph.

We use a parametric implementation of the pseudoflow algorithm (Hochbaum, 2008) to solve the parametric minimum cut problem.

For PageRank, we use the implementation of PageRank in the NetworkX package for Python (Hagberg et al., 2008). For TrustRank and AntiTrustRank, we wrote a wrapper around the NetworkX implementation of PageRank. We use the default settings of the PageRank solver, except for the attenuation factor α, which we treat as a hyperparameter. The maximum number of iterations is set to 1000.

The Random algorithm provides baseline comparison. Each vertex is assigned a score, x_i, uniformly drawn from the unit interval. We report the average of the best modularity obtained in each of 10 trials.

Our implementations of these algorithms, as well as the code used to run the experiments, are available open-source at https://github.com/hochbaumGroup/mrf-aberrant-detection.

Hyperparameters. All algorithms, except for Random, have hyperparameters. The hyperparameters for MRF are the normalized trade-off parameter λ_{norm} and the percentage of priors p_{prior}. The hyperparameters for TrustRank and AntiTrustRank are the attenuation factor α and the percentage of priors p_{prior}. PageRank has only the attenuation factor α as its hyperparameter, since the algorithm does not utilize the prior values.

For each algorithm and dataset, we tested 200 combinations of hyperparameter values to maximize modularity. Each combination of hyperparameter values was selected with the Tree-structured Parzen estimator (TPE) algorithm (Bergstra et al., 2011) based on previous evaluations and a prior distribution for each of the parameters. The TPE algorithm uses approximate Bayesian Optimization to select a combination of hyperparameter values that has the highest expected increase in modularity score. Bayesian optimization methods, such as TPE, have been shown to outperform grid search and random search (Bergstra and Bengio, 2012) and to rival domain experts in finding good hyperparameter settings (Bergstra et al., 2011).

The prior distributions for the hyperparameters were selected as follows: For the normalized trade-off parameter λ_{norm}, we used a lognormal distribution with zero mean and a standard deviation of two. For the attenuation factor α, we used a uniform distribution on the unit interval, and for the percentage of priors p_{prior} we used a uniform distribution over the interval from 1 to 50 percent.

Table 1: Basic properties of twenty-one datasets from the KONECT project and the two datasets from the Web Spam project.

Dataset	Vertices	Arcs	Link Significance	Weights
Animal - Bison	26	314	Dominance behavior between bisons.	yes
Animal - Cattle	28	217	Dominance behavior between cattle.	yes
Animal - Hens	32	496	Pecking order among hens.	no
Citation - Cora	23166	91500	Scientific citations.	no
Citation - DBLP	12591	49728	Scientific citations in computer science.	no
Communication - Company	167	5783	Emails within manufacturing company.	yes
Communication - DNC Emails	1891	5517	Emails within committee.	yes
Communication - Slashdot	51083	130370	Responses to messages.	yes
Communication - University	1899	20296	Messages within university.	yes
Hyperlink - Blogs	1224	19022	Links between political blogs.	no
Hyperlink - Google	15763	170335	Links between internal Google pages.	no
Hyperlink - Spam 2006	11402	730774	Links between UK sites.	yes
Hyperlink - Spam 2007	114529	1771291	Links between UK sites.	yes
Infrastructure - Airports 1	3425	37594	Flights between airports.	yes
Infrastructure - Airports 2	2939	30501	Flights between airports.	no
Metabolic - Proteins (Small)	1706	6171	Interactions between proteins.	no
Social - Advogato	6539	47135	Trust relationships between users.	yes
Social - Dorm	217	2672	Friendship ratings between students.	yes
Social - High School	70	366	Friendship ratings in a high school.	yes
Social - Physicians	241	1098	Trust relationships between physicians.	no
Social - Twitter	23370	33101	Following relationships between users.	no
Trophic - FL Dry Season	128	2137	Carbon exchanges between organisms.	yes
Trophic - FL Wet Season	128	2106	Carbon exchanges between organisms.	yes

6 RESULTS

For each of the twenty-three datasets and five algorithms, we report the best modularity found after the hyperparameter search. In Table 2, in the final column we report the highest modularity score found among all the algorithms. In the remaining columns, we show the relative modularity score obtained by each algorithm as a percentage of the highest modularity score. Thus, one algorithm always achieves 100 percent.

On average, our MRF algorithm achieves the highest percentage of the maximum modularity, at 92%. The next best algorithms, TrustRank and AntiTrustRank, achieve 74% and 73% respectively. MRF achieves the highest modularity on thirteen of the twenty-three datasets.

In Table 3, we examine the properties of these partitions with maximum modularity, using the ad-hoc metrics described in section 4. Across all datasets, the median value of $\frac{W_{01}/N_0}{d_{\text{avg}}}$ for MRF is 0.04. For AntiTrustRank, it is 0.13. This suggests that MRF is indeed finding partition with fewer arcs from normal to aberrant. It attains the smallest value on twenty of the twenty-three datasets. Looking at the median value of $\frac{W_{11}/N_1}{d_{\text{avg}}}$, we see another side of the story. For AntiTrustRank, the median value is 1.18. For MRF, it is 0.60. This suggests that the vertices classified as aberrant by AntiTrustRank are more interlinked than those classified by MRF. These results make sense: whereas MRF is minimizing the links from normal to aberrant, without any stipulation about connections between aberrant vertices, AntiTrustRank is working backwards from known aberrant vertices to find aberrant clusters.

The last three columns in Table 3 corroborates this analysis. MRF outputs a labeling in which the fraction of weight to aberrant that comes from normal (as opposed to other aberrant vertices) is 0.05 (median). On the other hand, in TrustRank the same value is 0.15 and in AntiTrustRank it is 0.18. MRF attains the smallest value on twenty of the twenty-three datasets.

The running times of all algorithms, with the exception of Random, were in the same order of magnitude.

7 CONCLUSIONS

In this paper, we studied the problem of identifying agents with aberrant behavior in networks. Such agents frequently appear in today's networks, including malicious websites on the internet, fake profiles in

Table 2: Normalized asymmetric modularity score by algorithm for twenty-three datasets. Performance is reported as a percentage of the highest asymmetric modularity score (last column) obtained among all the algorithms.

Dataset	**MRF**	**Trust Rank**	**AntiTrust Rank**	**Page Rank**	**Random**	**Best Modularity Score (100%)**
Animal - Bison	82.4	**100.0**	94.3	84.4	36.2	0.279
Animal - Cattle	84.5	**100.0**	80.5	97.9	31.3	0.294
Animal - Hens	99.3	**100.0**	99.8	**100.0**	29.5	0.233
Citation - Cora	55.7	57.3	**100.0**	32.1	12.2	0.517
Citation - DBLP	**100.0**	89.3	41.2	58.1	16.4	0.409
Communication - Company	66.1	**100.0**	89.8	65.5	22.1	0.405
Communication - DNC Emails	**100.0**	47.4	37.7	22.3	21.2	0.402
Communication - Slashdot	93.9	69.2	**100.0**	49.9	20.7	0.321
Communication - University	**100.0**	63.1	62.1	29.7	18.1	0.342
Hyperlink - Blogs	**100.0**	70.8	81.2	57.8	23.3	0.302
Hyperlink - Google	**100.0**	71.2	94.4	53.1	15.7	0.448
Hyperlink - Spam 2006	**100.0**	53.4	91.5	37.2	12.8	0.743
Hyperlink - Spam 2007	**100.0**	47.6	63.7	45.5	19.9	0.505
Infrastructure - Airports 1	**100.0**	45.9	51.6	17.8	10.6	0.577
Infrastructure - Airports 2	**100.0**	71.2	47.9	19.9	10.7	0.591
Metabolic - Proteins (Small)	**100.0**	68.5	68.9	1.2	13.6	0.517
Social - Advogato	**100.0**	72.7	76.8	53.8	17.4	0.392
Social - Dorm	57.3	**100.0**	66.4	63.4	11.1	0.615
Social - High School	89.2	**100.0**	92.2	94.9	14.5	0.739
Social - Physicians	94.1	82.8	**100.0**	43.8	8.7	0.931
Social - Twitter	92.7	7.7	15.1	**100.0**	31.9	0.244
Trophic - FL Dry Season	**100.0**	85.3	51.1	90.4	22.1	0.581
Trophic - FL Wet Season	**100.0**	93.4	78.2	93.4	28.3	0.588
Average	**92.0**	**73.8**	**73.2**	**57.0**	**19.5**	—

Table 3: Ad-hoc metrics of the partition returned by each algorithm. The results are reported for the partition that maximizes asymmetric modularity, as in Table 2. For brevity, we exclude PageRank and Random. For a partition that satisfies aberrant linking behavior we expect the first and third metric to be small.

	$\frac{W_{01}/N_0}{d_{\text{avg}}}$			$\frac{W_{11}/N_1}{d_{\text{avg}}}$			$\frac{W_{01}}{W_{01}+W_{11}}$		
	MRF	Trust	AntiTrust	MRF	Trust	AntiTrust	MRF	Trust	AntiTrust
Animal - Bison	0.16	0.20	0.16	0.46	0.51	0.59	0.29	0.28	0.24
Animal - Cattle	0.02	0.08	0.03	0.58	0.63	0.69	0.04	0.06	0.05
Animal - Hens	0.03	0.04	0.04	0.45	0.48	0.48	0.07	0.07	0.08
Citation - Cora	0.00	0.02	0.05	0.36	0.41	0.98	0.00	0.01	0.08
Citation - DBLP	0.00	0.09	0.01	0.43	0.48	1.18	0.00	0.06	0.05
Communication - Company	0.33	0.94	0.19	0.83	0.55	1.91	0.18	0.23	0.28
Communication - DNC Emails	0.06	2.35	0.11	0.66	0.20	0.86	0.05	0.38	0.36
Communication - Slashdot	0.06	0.35	0.17	0.80	0.31	1.27	0.03	0.21	0.23
Communication - University	0.23	1.11	0.19	0.60	0.36	2.46	0.24	0.35	0.41
Hyperlink - Blogs	0.08	0.49	0.15	0.33	0.31	1.13	0.09	0.22	0.29
Hyperlink - Google	0.00	0.07	0.13	0.60	0.32	0.96	0.00	0.15	0.15
Hyperlink - Spam 2006	0.00	0.28	0.01	1.92	0.61	10.10	0.00	0.02	0.02
Hyperlink - Spam 2007	0.00	0.41	0.24	0.71	0.12	7.37	0.00	0.15	0.14
Infrastructure - Airports 1	0.24	1.84	0.19	0.87	0.25	4.15	0.09	0.45	0.30
Infrastructure - Airports 2	0.07	0.64	0.33	0.47	0.28	0.82	0.15	0.36	0.33
Metabolic - Proteins (Small)	0.19	0.69	0.22	0.52	0.37	1.25	0.27	0.38	0.34
Social - Advogato	0.11	0.75	0.13	0.39	0.29	1.86	0.10	0.31	0.28
Social - Dorm	0.19	0.21	0.25	0.55	0.65	0.86	0.29	0.17	0.23
Social - High School	0.04	0.08	0.09	0.62	0.85	0.76	0.05	0.05	0.08
Social - Physicians	0.00	0.00	0.00	0.88	0.79	1.03	0.00	0.00	0.00
Social - Twitter	0.00	0.03	0.00	0.80	0.74	1.57	0.00	0.01	0.00
Trophic - FL Dry Season	0.00	0.00	0.09	1.96	1.42	7.09	0.00	0.00	0.18
Trophic - FL Wet Season	0.00	0.35	0.05	1.70	0.34	5.80	0.00	0.06	0.14
Median	**0.04**	**0.28**	**0.13**	**0.60**	**0.41**	**1.18**	**0.05**	**0.15**	**0.18**

social media, and fake news sources prolific in spreading misinformation. The unifying property of these networks is that normal agents rarely link to aberrant ones. We call this *aberrant linking behavior*.

We formulated the detection problem in a novel way: as a directed Markov Random Field (MRF) problem. This formulation balances obeying any given prior information with minimizing the links from normal to aberrant agents. We discussed how the formulation is solved optimally and efficiently.

To compare the performance of the algorithms, we developed a new, asymmetric variant of the modularity metric for directed graphs, addressing a known shortcoming of the existing metric. We showed that our metric has desirable properties and proved that maximizing it is NP-hard. We also used several ad-hoc metrics to better understand properties of the solutions.

In an empirical experiment, we found that the MRF method outperforms competitors such as PageRank, TrustRank, AntiTrustRank, and Random. The solutions returned by MRF had the largest modularity score on thirteen of the twenty-three datasets tested. The modularity for MRF was, on average, 25 percent better than the modularity returned by TrustRank or AntiTrustRank.

ACKNOWLEDGEMENTS

D. S. Hochbaum was supported in part by National Science Foundation (NSF) award CMMI 1760102. M. Velednitsky was supported by the National Physical Science Consortium (NPSC).

REFERENCES

Abernethy, J., Chapelle, O., and Castillo, C. (2010). Graph regularization methods for web spam detection. *Machine Learning*, 81(2):207–225.

Ahuja, R. K., Hochbaum, D. S., and Orlin, J. B. (2003). Solving the convex cost integer dual network flow problem. *Management Science*, 49(7):950–964.

Becchetti, L., Castillo, C., Donato, D., Leonardi, S., and Baeza-Yates, R. (2008). Web spam detection: Link-based and content-based techniques. In *The European Integrated Project Dynamically Evolving, Large Scale Information Systems (DELIS): proceedings of the final workshop*, volume 222, pages 99–113.

Becchetti, L., Castillo, C., Donato, D., Leonardi, S., and Baezayates, R. (2006). Linkbased characterization and detection of web spam. In *2nd International Workshop on Adversarial Information Retrieval on the Web, AIRWeb 2006-29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2006*.

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305.

Bergstra, J. S., Bardenet, R., Bengio, Y., and Kégl, B. (2011). Algorithms for hyper-parameter optimization. In *Advances in neural information processing systems*, pages 2546–2554.

Brandes, U., Delling, D., Gaertler, M., Görke, R., Hoefer, M., Nikoloski, Z., and Wagner, D. (2006). Maximizing modularity is hard. *arXiv preprint physics/0608255*.

Castillo, C., Donato, D., Becchetti, L., Boldi, P., Leonardi, S., Santini, M., and Vigna, S. (2006). A reference collection for web spam. *SIGIR Forum*, 40(2):11–24. http://chato.cl/webspam/datasets/.

Castillo, C., Donato, D., Gionis, A., Murdock, V., and Silvestri, F. (2007). Know your neighbors: Web spam detection using the web topology. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 423–430. ACM.

Erdélyi, M., Garzó, A., and Benczúr, A. A. (2011). Web spam classification: a few features worth more. In *Proceedings of the 2011 Joint WICOW / AIRWeb Workshop on Web Quality*, pages 27–34. ACM.

Fire, M., Kagan, D., Elyashar, A., and Elovici, Y. (2014). Friend or foe? fake profile identification in online social networks. *Social Network Analysis and Mining*, 4(1):194.

Gallo, G., Grigoriadis, M. D., and Tarjan, R. E. (1989). A fast parametric maximum flow algorithm and applications. *SIAM Journal on Computing*, 18(1):30–55.

Gan, Q. and Suel, T. (2007). Improving web spam classifiers using link structure. In *Proceedings of the 3rd international workshop on Adversarial information retrieval on the web*, pages 17–20. ACM.

Garey, M. R., Johnson, D. S., and Stockmeyer, L. (1974). Some simplified np-complete problems. In *Proceedings of the sixth annual ACM symposium on Theory of computing*, pages 47–63. ACM.

Geman, S. and Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741.

Ghosh, S., Viswanath, B., Kooti, F., Sharma, N. K., Korlam, G., Benevenuto, F., Ganguly, N., and Gummadi, K. P. (2012). Understanding and combating link farming in the twitter social network. In *Proceedings of the 21st international conference on World Wide Web*, pages 61–70. ACM.

Goldberg, A. V. and Tarjan, R. E. (1988). A new approach to the maximum-flow problem. *Journal of the ACM (JACM)*, 35(4):921–940.

Gori, M. and Pucci, A. (2006). Research paper recommender systems: A random-walk based approach. In *Web Intelligence, 2006. WI 2006. IEEE/WIC/ACM International Conference on*, pages 778–781. IEEE.

Gyöngyi, Z., Garcia-Molina, H., and Pedersen, J. (2004). Combating web spam with trustrank. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 576–587. VLDB Endowment.

Hagberg, A., Swart, P., and S Chult, D. (2008). Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States).

Hochbaum, D. S. (2001). An efficient algorithm for image segmentation, markov random fields and related problems. *Journal of the ACM (JACM)*, 48(4):686–701.

Hochbaum, D. S. (2008). The pseudoflow algorithm: A new algorithm for the maximum-flow problem. *Operations research*, 56(4):992–1009.

Hochbaum, D. S. and Orlin, J. B. (2013). Simplifications and speedups of the pseudoflow algorithm. *Networks*, 61(1):40–57.

Kim, Y., Son, S.-W., and Jeong, H. (2010). Finding communities in directed networks. *Physical Review E*, 81(1):016103.

Krishnan, V. and Raj, R. (2006). Web spam detection with anti-trust rank. In *AIRWeb*, volume 6, pages 37–40.

Kunegis, J. (2013). Konect: the koblenz network collection. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 1343–1350. ACM. http://konect.uni-koblenz.de/.

Li, H., Mukherjee, A., Liu, B., Kornfield, R., and Emery, S. (2014). Detecting campaign promoters on twitter using markov random fields. In *2014 IEEE International Conference on Data Mining*, pages 290–299. IEEE.

Liu, T.-Y. et al. (2009). Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331.

Malliaros, F. D. and Vazirgiannis, M. (2013). Clustering and community detection in directed networks: A survey. *Physics Reports*, 533(4):95–142.

Meilă, M. and Pentney, W. (2007). Clustering by weighted cuts in directed graphs. In *Proceedings of the 2007 SIAM international conference on data mining*, pages 135–144. SIAM.

Newman, M. E. (2006). Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582.

Newman, M. E. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113.

Ntoulas, A., Najork, M., Manasse, M., and Fetterly, D. (2006). Detecting spam web pages through content analysis. In *Proceedings of the 15th international conference on World Wide Web*, pages 83–92. ACM.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

Rosvall, M. and Bergstrom, C. T. (2008). Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123.

Roul, R. K., Asthana, S. R., Shah, M., and Parikh, D. (2016). Spam web page detection using combined content and link features. *International Journal of Data Mining, Modelling and Management*, 8(3):209–222.

Saito, H., Toyoda, M., Kitsuregawa, M., and Aihara, K. (2007). A large-scale study of link spam detection by graph algorithms. In *Proceedings of the 3rd international workshop on Adversarial information retrieval on the web*, pages 45–48. ACM.

Sayyadi, H. and Getoor, L. (2009). Futurerank: Ranking scientific articles by predicting their future pagerank. In *Proceedings of the 2009 SIAM International Conference on Data Mining*, pages 533–544. SIAM.

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *Departmental Papers (CIS)*, page 107.

Shu, K., Bernard, H. R., and Liu, H. (2019). Studying fake news via network analysis: detection and mitigation. In *Emerging Research Challenges and Opportunities in Computational Social Network Analysis and Mining*, pages 43–65. Springer.

Shu, K., Sliva, A., Wang, S., Tang, J., and Liu, H. (2017). Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, 19(1):22–36.

Tagarelli, A. and Interdonato, R. (2014). Lurking in social networks: topology-based analysis and ranking methods. *Social Network Analysis and Mining*, 4(1):230.

Tagarelli, A. and Interdonato, R. (2018). *Mining Lurkers in Online Social Networks: Principles, Models, and Computational Methods*. Springer.

Törnberg, P. (2018). Echo chambers and viral misinformation: Modeling fake news as complex contagion. *PloS one*, 13(9):e0203958.

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416.

Webb, S., Caverlee, J., and Pu, C. (2008). Predicting web spam with http session information. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 339–348. ACM.

Wu, B. and Chellapilla, K. (2007). Extracting link spam using biased random walks from spam seed sets. In *Proceedings of the 3rd international workshop on Adversarial information retrieval on the web*, pages 37–44. ACM.

Wu, B. and Davison, B. D. (2005). Identifying link farm spam pages. In *Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 820–829. ACM.

Xiao, C., Freeman, D. M., and Hwa, T. (2015). Detecting clusters of fake accounts in online social networks. In *Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security*, pages 91–101. ACM.

Zhou, D., Burges, C. J., and Tao, T. (2007). Transductive link spam detection. In *Proceedings of the 3rd international workshop on Adversarial information retrieval on the web*, pages 21–28. ACM.

APPENDIX

Proof of Claim 1.

We multiply (DirMod) by the constant W^2:

$$\sum_{i,j\in V}\left[Ww_{ij}-d_i^{\text{out}}d_j^{\text{in}}\right]\delta(i,j)$$
$$=W\sum_{i,j\in V}w_{ij}\delta(i,j)-\sum_{i,j\in V}d_i^{\text{out}}d_j^{\text{in}}\delta(i,j).$$

Now, since $\delta(i,j)=1$ if and only if i and j are in the same cluster:

$$=W(W_{00}+W_{11})-\sum_{i\in C_0}\sum_{j\in C_0}d_i^{\text{out}}d_j^{\text{in}}-\sum_{i\in C_1}\sum_{j\in C_1}d_i^{\text{out}}d_j^{\text{in}}$$
$$=(W_{00}+W_{01}+W_{10}+W_{11})(W_{00}+W_{11})$$
$$-(W_{00}+W_{01})(W_{00}+W_{10})$$
$$-(W_{11}+W_{10})(W_{11}+W_{01})$$
$$=2(W_{00}W_{11}-W_{10}W_{01})$$

□

Proof of Claim 2. The proof is a reduction from the minimum bisection problem on an undirected and unweighted graph $G=(V,E)$. A bisection (C_0,C_1) is a partition of the set of vertices V such that $|C_0|=|C_1|=n/2$, where n is the number of nodes in the original graph. The minimum bisection problem is to find a bisection (C_0,C_1) that minimizes $W_{01}=W_{10}$. This problem is known to be NP-Hard (Garey et al., 1974).

Consider an undirected and unweighted graph G on which we would like to solve the minimum bisection problem. We create the graph G' where we copy G by turning each edge into two directed arcs and add two vertices, s and t, with an arc from s to every vertex in G and an arc from every vertex in G to t. Let the weight of all these arcs be M, for sufficiently large M (e.g. $M\geq m^2$ where m is the number of arcs in the original graph).

Consider a partition of this new graph into C_0 and C_1. We still use W_{pq} to denote the total weight *in the original graph* between clusters p and q. We break the modularity [3] calculation into four cases, depending on which clusters s and t are in:

$s\in C_0, t\in C_0$:

$$(2|C_0|M+W_{00})(W_{11})-\frac{3}{4}(|C_1|M+W_{01})^2$$

$s\in C_0, t\in C_1$:

$$(|C_0|M+W_{00})(|C_1|M+W_{11})-\frac{3}{4}(nM+W_{01})^2$$

$s\in C_1, t\in C_0$:

$$(|C_0|M+W_{00})(|C_1|M+W_{11})-\frac{3}{4}(W_{01})^2$$

$s\in C_1, t\in C_1$:

$$(W_{00})(2|C_1|M+W_{11})-\frac{3}{4}(|C_0|M+W_{01})^2$$

Expanding these expressions, we see that the coefficient of M^2 is largest when $s\in C_1$, $t\in C_0$, and $|C_0|=|C_1|=n/2$. Thus, for sufficiently large M, these are necessary conditions to maximize the modularity of the clustering in G'. The expression becomes:

$$\frac{n^2}{4}M^2+\frac{n}{2}(W_{00}+W_{11})M+W_{00}W_{11}-\frac{3}{4}W_{01}^2.$$

Assuming M is sufficiently large, an optimal solution maximizes $\frac{n}{2}M(W_{00}+W_{11})$ and thus $W_{00}+W_{11}$. However, maximizing this quantity is equivalent to minimizing $W_{01}+W_{10}=2W_{01}$. Thus, the partition which maximizes modularity in G' gives us the minimum bisection in G. □

[3] We ignore the normalization term $\frac{4}{W^2}$.