Spatial Scan Statistics for Graph Clustering

Bei Wang [*] Jeff M. Phillips [†] Robert Schreiber [‡] Dennis Wilkinson [§]
Nina Mishra [¶] Robert Tarjan [‖]

Abstract
In this paper, we present a measure associated with detection and inference of statistically anomalous clusters of a graph based on the likelihood test of observed and expected edges in a subgraph. This measure is adapted from spatial scan statistics for point sets and provides quantitative assessment for clusters. We discuss some important properties of this statistic and its relation to modularity and Bregman divergences. We apply a simple clustering algorithm to find clusters with large values of this measure in a variety of real-world data sets, and we illustrate its ability to identify statistically significant clusters of selected granularity.

1 Introduction.

Numerous techniques have been proposed for identifying clusters in large networks, but it has proven difficult to meaningfully and quantitatively assess them, especially from real-world data whose clustering structure is *a priori* unknown. One of the key challenges encountered by previous clustering methods is rating or evaluating the results. In large networks, manual evaluation of the results is not feasible, and previous studies have thus turned to artificially created graphs with known structure as a test set. However, many methods, especially those in which the number of clusters must be specified as an algorithm parameter, give very poor results when applied to real-world graphs, which often have a highly skewed degree distribution and overlapping, complex clustering structure [18, 33].

1.1 Prior Work on Measures.

Modularity. The problem of assessment was partially solved by the introduction of modularity [12], a global objective function used to evaluate clusters that rewards existing internal edges and penalizes missing internal edges. Non-overlapping clusters, or partitions of a graph, are obtained by maximizing the distance from what a random graph would predict, either by extremal optimization [15], fast greedy hierarchical algorithms [31, 37], simulated annealing [38] or spectral clustering [32].

However, modularity cannot directly assess how unexpected and thus significant individual clusters are. Additionally it cannot distinguish between clusterings of different granularity on the same network. For example, comparable overall modularities were reported for hard clusterings of the same scientific citation graph into 44, 324, and 647 clusters [37], results which are clearly of varying usefulness depending on the application.

Spatial Scan Statistics. Scan statistics [19] measure densities of data points for a sliding window on ordered data. The densest regions under a fixed size window are considered the most anomalous. This notion of a sliding window has been generalized to neighborhoods on directed graphs [36] where the neighborhood of a vertex is restricted to vertices within some constant number of edges in the graph. The number of neighbors is then compared to an expected number of neighbors based on previous data in a time-marked series.

Spatial scan statistics were introduced by Kulldorff [23] to find anomalous clusters of points in 2 or greater dimensions without fixing a window size. These statistics measure the surprise of observing a particular region by computing the log-likelihood of the probability of the most likely model for a cluster versus the probability of the most likely model for no cluster. Kulldorff argues that the region with the largest spatial scan statistic is the most likely to be generated by a different distribution, and thus is most anomalous. This test was shown to be the most powerful test [27] for finding a region which demonstrates that the data set is not generated from a single distribution. Kulldorff [23] derived expressions for the spatial scan statistic under a Poisson and Bernoulli model. Agarwal et.al. [2] generalized this derivation to 1-parameter exponential families, and Kulldorff has studied various (see [26]) other forms of this statistic. Many techniques exist for computing the statistic quickly for anomaly detection for point sets [30, 25, 1].

1.2 Previous Algorithmic Work.

Clustering is well-established as an important method of information

[*]Duke University : beiwang@cs.duke.edu
[†]Duke University : jeffp@cs.duke.edu
[‡]HP Labs : rob.schreiber@hp.com
[§]HP Labs : dennis.wilkinson@hp.com
[¶]Visiting Search Labs, Microsoft Research; University of Virginia : nmishra@cs.virginia.edu
[‖]Princeton University; HP Labs : robert.tarjan@hp.com

extraction from large data sets. Hard clustering divides data into disjoint clusters while soft clustering allows data elements to belong to more than one cluster. Existing techniques include MCL [41], Ncut [40], graclus [13], MCODE [5], iterative scan [9], k-clique-community [37], spectral clustering [35, 32], simulated annealing [28], or partitioning using network flow [34], edge centrality [18], and many other techniques e.g. [42].

Several statistically motivated graph clustering techniques exist [20, 39, 22]. Itzkovitz et. al. discussed distributions of subgraphs in random networks with arbitrary degree sequence, which have implications for detecting network motifs [20]. Sharan et. al. introduced a probabilistic model for protein complexes taking conservation into consideration [39]. Koyuturk et. al. identified clusters by employing a min-cut algorithm where a subgraph was considered to be statistically significant if its size exceeded a probabilistic estimation based on a piecewise degree distribution model [22]. These techniques are all different from our approach as our model based on spatial scan statistics has properties essential for detecting statistically significant clusters.

A general clustering framework using Bregman divergences as optimization functions has been proposed by Banerjee *et.al.* [14, 6, 7]. This approach is of note because the optimization function we use can be interpreted as a Bregman divergence, although our theoretical and algorithmic approaches are completely different (as explained in Section 3).

1.3 Our Contribution. Our main contribution is the generalization of spatial scan statistics from point sets to graphs. Our statistic, *Poisson discrepancy*, offers a metric that determines how significant the clusters are using a normalized measure of likelihood. By comparison to random graphs with the same expected degree sequence as the network under consideration, a p-value based on the Poisson discrepancy is associated to each cluster found, providing a problem independent measure of its statistical significance. The statistic is correctly normalized to provide an absolute measure of cluster significance on an individual level, allowing comparison between clusters from different networks or from different clusterings of the same network.

We also implement two simple greedy algorithms which seek to maximize the Poisson discrepancy. Our method includes a tunable parameter which we show is strongly correlated to the average size of the found clusters. This parameter allows the user to vary the expected size of the clusters found by the algorithms, which may be useful for various applications. In addition, because clusters of different size may be meaningfully compared with each other, our method provides a way to identify important levels of granularity within a network. In fact, our results using real-world data show that statistically significant clusters of varying sizes can exist in a given neighborhood of a network, implying that real-world networks display granularity on a number of different levels. Finally, our method allows data points to fit in any number of clusters, including zero. This makes it applicable to real-world networks in which some vertices may not have any kind of strong association.

In Sections 2 and 3 of this paper, we present the theoretical foundation of our clustering method. The algorithm and variations are described in Section 4. Bipartite extensions of our algorithm are explained in Section 5. In Section 6, we empirically evaluate our objective function and algorithms on real and synthetic datasets. We calculate the algorithms' runtime and power; we evaluate other clustering algorithms using Poisson discrepancy; and we demonstrate that the clusters found using Poisson discrepancy with our algorithm are meaningful, are scaled reliably with a tuning parameter γ, and can overlap with each other.

2 Definitions.

Graphs. Let $G = (V, E)$ be an undirected graph with vertex set V and edge set E, such that E is a multiset of elements in $[V]^2$, where $[V]^2$ denotes a set of 2-element multisets of V. That is $[V]^2 = \{\{u, v\}|u, v \in V\}$, where u and v are not necessarily distinct. G allow self-loops and multiple edges between a pair of vertices. A *cluster* $Z = (V_Z, E_Z)$ is a subgraph of G induced by subset $V_Z \subseteq V$. The collection of clusters $Z \subseteq G$ is denoted as $\mathcal{Z}$. The subgraph in G not in Z is $\bar{Z} = (V, E \setminus E_Z)$. Let $c(Z)$ be the number of observed edges in cluster $Z \subseteq G$. $c(G)$ is the total number of observed edges in G. $c^*(x)$ is the observed number of edges between a pair of vertices $x = \{v_i, v_j\} \in [V]^2$. The *degree* k_i of a vertex v_i is the number of edges that contain v_i. A self-loop at v_i is counted twice in the degree.

The *neighborhood* of a vertex set $U \subseteq V$ is the set of vertices $\{v \mid \{v, u\} \in E$ for $u \in U$ and $v \in V \setminus U\}$, denoted $N(U)$. The size of a set V is denoted $|V|$. We define the distance between two vertex sets U and V as $d(U, V) = |U| + |V| - 2|U \cap V|$.

Poisson Random Graph Model. Many large real-world graphs have diverse, non-uniform, degree distributions [8, 4, 3] that are not accurately described by the classic Erdös and Rényi random graph models [16]. We consider a Poisson random graph model here that captures some main characteristics of real-world graphs, specifically, allowing vertices to have different expected degrees. Notice that this model is different from models used in [10, 12].

The model generates random graphs with a given expected degree sequence $k = \langle k_1, k_2, ..., k_{|V|}\rangle$ for the vertex set $V = \langle v_1, v_2, ..., v_{|V|}\rangle$, where vertex v_i has expected degree k_i. Let $k_V = \sum_{i=1}^{|V|} k_i$ and $m = k_V/2$. The model chooses a total of m pairs of vertices as edges through m steps, with replacement. At each step, each of the two end-points of an edge is chosen among all vertices through a Poisson proces, proportional to their degrees. The probability that each end point of a chosen edge contains vertex v_i is k_i/k_V. The expected degree of vertex v_i after this process is $m(k_i/k_V + k_i/k_V) = m(k_i/2m + k_i/2m) = k_i$. We refer to this model as the *Poisson random graph model.* It fixes the total number of edges and allows multiple edges between any pair of vertices.

A bipartite multigraph example captured by a Poisson random graph model is an internet bulletin board. There is a set of users who post under different threads. A post in a thread would constitute an edge. Of course a single user can post multiple times to a single thread, and some users post more often and some threads generate more posts.

3 Spatial Scan Statistics for Graphs.

In this section we generalize the notion of a spatial scan statistic [23] to graphs. We also highlight some important properties of this statistic, as well as its relation to local modularity and Bregman divergences. It is this spatial scan statistic for graphs that we use to provide quantitative assessment of significant clusters.

3.1 Spatial Scan Statistics Poisson Model. Given $G = (V, E)$, its edge set describes a degree sequence $k = \langle k_1, k_2, ..., k_{|V|}\rangle$ for the vertex set $V = \langle v_1, v_2, ..., v_{|V|}\rangle$, where k_i is the degree of vertex v_i. Let $k_V = \sum_{i=1}^{|V|} k_i$ and $c(G) = k_V/2$.

According to the Poisson random graph model with k as the given expected degree sequence and $c(G)$ as the total number of expected edges, define $\mu^*(x)$ as the expected number of edges connecting the pair $x = \{v_i, v_j\} \in [V]^2$. For $i \neq j$, $\mu^*(x) = k_i k_j/2c(G)$. For $i = j$, $\mu^*(x) = k_i^2/4c(G)$. For a subgraph $A = (V_A, E_A) \subseteq G$, define $\mu(A)$ as the expected number of edges in A. Let $K_{V_A} = \sum_{v_i \in V_A} k_i$, $\mu(A) = \sum_{x\in[V_A]^2} \mu^*(x) = k_{V_A}^2/4c(G)$. Notice that $\mu(G) = \sum_{x\in[V]^2} \mu^*(x) = c(G)$. A simple example is shown in Figure 1 where cluster Z is induced by dark vertices in the graph, $c(G) = \mu(G) = 10$, $c(Z) = 6$ and $\mu(Z) = 13^2/(4 \times 10) = 169/40$.

In the *spatial scan statistics Poisson model,* we assume edges in G are generated by a Poisson process N with intensity λ. The probability that there are exactly

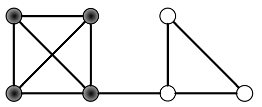
Figure 1: Example graph.

n occurrences in the process is denoted as $Poi(n, \lambda) = e^{-\lambda}\lambda^n/n!$. N assumes the same vertex set and expected degree sequence as G and generates edges according to the Poisson random graph model. Let $\mathcal{N}(A)$ be the random variable describing the number of edges in the subgraph $A \subseteq G$ under such a process. $\mathcal{N}(A)$ follows a Poisson distribution, denoted by $\mathcal{N}(A) \sim Poi(n, \lambda)$.

Under such a model, consider a single cluster $Z = (V_Z, E_Z) \subset G$, such that edges in Z are generated with rate p, while edges in $\bar{Z}$ are generated with rate q. The null hypothesis $H_0 : p = q$ assumes complete spatial randomness [21], that is, edges inside and outside the cluster are generated under the same rate. The alternative hypothesis $H_1 : p > q$ assumes that edges inside the cluster are generated at a higher rate than those outside the cluster. μ therefore represents a known underlying intensity that generates edges under H_0 [24]. Therefore, under H_0, the number of edges in any given cluster $A \subseteq G$ is Poisson distributed, $\mathcal{N}(A) \sim Poi(n, p\mu(A))$ for some value p. Under H_1, $\forall A \subseteq G$, $\mathcal{N}(A) \sim Poi(n, p\mu(A \cap Z) + q\mu(A \cap \bar{Z}))$ [23].

Likelihood Ratio Test. The spatial scan statistics for graphs are based on the likelihood ratio test derived from [23].

The spatial scan statistics Poisson Model assumes under the null hypothesis H_0 that all edges in G are generated at the same rate. That is, under H_0, the probability of $c(G)$ edges being observed in G is $Poi(c(G), p\mu(G))$. The density function $f(x)$ of a specific pair of vertices $x \in [V]^2$ being an actual edge is $f(x) = \frac{p\mu^*(x)}{p\mu(G)} = \frac{\mu^*(x)}{\mu(G)}$.

Incorporating the density of each edge, the likelihood function under H_0 is

$$\begin{aligned}\mathbf{L_0} &= \max_{p=q} Poi(c(G), p\mu(G)) \prod_{x\in E} f(x) \\ &= \max_{p=q} \frac{e^{-p\mu(G)}(p\mu(G))^{c(G)}}{c(G)!} \prod_{x\in E} \frac{\mu^*(x)}{\mu(G)} \\ &= \max_{p=q} \frac{e^{-p\mu(G)}p^{c(G)}}{c(G)!} \prod_{x\in E} \mu^*(x)\end{aligned}$$

For a fixed Z, $\mathbf{L_0}$ takes its maximum when $p = q = \frac{c(G)}{\mu(G)}$.

That is

$$\mathbf{L_0} = \frac{e^{-c(G)}}{c(G)!} \left(\frac{c(G)}{\mu(G)}\right)^{c(G)} \prod_{x \in E} \mu^*(x).$$

The alternative hypothesis H_1 assumes that for a particular cluster $Z = (V_Z, E_Z) \subset \mathcal{Z}$, the edges in Z are generated at a different rate than those in $\bar{Z}$. Under H_1, the probability of $c(G)$ edges being observed in G is $Poi(c(G), p\mu(Z) + q(\mu(G) - \mu(Z)))$. For $x \in E_Z$, the density function $f(x)$ is $f(x) = \frac{p\mu^*(x)}{p\mu(Z)+q(\mu(G)-\mu(Z))}$. For $x \in E \setminus E_Z$, $f(x) = \frac{q\mu^*(x)}{p\mu(Z)+q(\mu(G)-\mu(Z))}$. The likelihood function under H_1 is

$$\begin{aligned}\mathbf{L}(Z) &= \max_{p>q} Poi(c(G), p\mu(Z) + q(\mu(G) - \mu(Z)))\\ &\cdot \prod_{x \in E_Z} f(x) \prod_{x \in E \setminus E_Z} f(x)\\ &= \max_{p>q} \frac{e^{-p\mu(Z)-q(\mu(G)-\mu(Z))}}{c(G)!}\\ &\cdot (p\mu(Z) + q(\mu(G) - \mu(Z)))^{c(G)}\\ &\cdot \prod_{x \in E_Z} \frac{p\mu^*(x)}{p\mu(Z) + q(\mu(G) - \mu(Z))}\\ &\cdot \prod_{x \in E \setminus E_Z} \frac{q\mu^*(x)}{p\mu(Z) + q(\mu(G) - \mu(Z))}\\ &= \max_{p>q} \frac{e^{-p\mu(Z)-q(\mu(G)-\mu(Z))}}{c(G)!} p^{c(Z)} q^{c(G)-c(Z)}\\ &\cdot \prod_{x \in E} \mu^*(x)\end{aligned}$$

For a fixed Z, $\mathbf{L}(Z)$ takes its maximum when $p = \frac{c(Z)}{\mu(Z)}$, $q = \frac{c(G)-c(Z)}{\mu(G)-\mu(Z)}$. If $\frac{c(Z)}{\mu(Z)} > \frac{c(G)-c(Z)}{\mu(G)-\mu(Z)}$,

$$\mathbf{L}(Z) = \frac{e^{-c(G)}}{c(G)!} \left(\frac{c(Z)}{\mu(Z)}\right)^{c(Z)} \left(\frac{c(G) - c(Z)}{\mu(G) - \mu(Z)}\right)^{c(G)-c(Z)} \cdot \prod_{x \in E} \mu^*(x)$$

otherwise $\mathbf{L}(Z) = \mathbf{L_0}$.

Given a cluster $Z \in \mathcal{Z}$, we define the *likelihood ratio*, LR, as the ratio of $\mathbf{L}(Z)$ and $\mathbf{L_0}$. For the Poisson model $c(G) = \mu(G)$, and if $\frac{c(Z)}{\mu(Z)} > \frac{c(G)-c(Z)}{\mu(G)-\mu(Z)}$, we have

$$LR(Z) = \frac{\mathbf{L}(Z)}{\mathbf{L_0}} = \left(\frac{c(Z)}{\mu(Z)}\right)^{c(Z)} \left(\frac{c(G) - c(Z)}{\mu(G) - \mu(Z)}\right)^{c(G)-c(Z)}$$

Otherwise $LR = 1$.

We then define the likelihood ratio *test statistic*, Λ, as the maximum likelihood ratio over all clusters $Z \in \mathcal{Z}$,

$$\Lambda = \max_{Z \in \mathcal{Z}} LR(Z).$$

Poisson Discrepancy. Let $r(Z) = \frac{c(Z)}{c(G)}$ and $b(Z) = \frac{\mu(Z)}{\mu(G)}$, we define the *Poisson discrepancy*, d_P, as

$$d_P(Z) = r(Z) \log \frac{r(Z)}{b(Z)} + (1 - r(Z)) \log \frac{1 - r(Z)}{1 - b(Z)}.$$

Intuitively, $r(Z)$ is the observed edge ratio and $b(Z)$ is the baseline edge ratio in Z and G.

Since $\log \Lambda = c(G) \max_{Z \in \mathcal{Z}} d_P(Z)$, for the cluster Z that maximizes d_P, $d_P(Z)$ constitutes the test statistic Λ. This means that the likelihood test based on $\max_{Z \in \mathcal{Z}} d_P(Z)$ is identical to one based on Λ. Since $0 < r(Z), b(Z) \leq 1$, from this point on, we evaluate clusters based on the Poisson discrepancy. d_P determines how surprising $r(Z)$ is compared to the rest of the distribution. Thus clusters with larger values of d_P are more likely to be inherently different from the rest of the data.

Point Set Model. It should be noted that these definitions were originally formulated where the set of all possible edges, $[V]^2$, was instead a point set, the function μ measured a baseline estimate of the population, and the function c measured reported data, such as instances of a disease. In this setting, the set of possible clusters is usually restricted to those contained in some geometrically described family of ranges. This set can often be searched exhaustively in time polynomial in the size of the point set. In our setting we have $2^{|V|}$ possible ranges (vertex sets which induce $\mathcal{Z}$) — an intractably large number. Hence, Section 4 explains how to explore these ranges effectively.

Significance of a Cluster. Although we can consider many (or all) subsets and determine the one that is most anomalous by calculating the Poisson discrepancy, this does not determine whether this value is significant. Even a graph generated according to a Poisson random graph model will have some cluster which is most anomalous. For a graph $G = (V, E)$ with degree sequence k and for a particular cluster $Z \subseteq G$ we can compare $d_P(Z)$ to the distribution of the values of the most anomalous clusters found in a large set of (say 1000) random data graphs. To create a random data graph, we fix V; then we randomly select $c(G)$ edges according to a Poisson random graph model with expected degree sequence k. If the Poisson discrepancy for the original cluster is greater than all but a .05 fraction of the most anomalous clusters from the random data sets, then we say it has a *p-value* of .05. The lower the p-value, the more significantly anomalous the range is. These high discrepancy clusters are most significant because they are the most unlikely compared to what is expected from our random graph model.

3.2 Properties of Spatial Scan Statistics. Kulldorff has proved some optimal properties for the likelihood ratio test statistic for point sets [23]. In the context of graphs, we describe those properties essential for detecting statistically anomalous clusters in terms of d_P. For details and proofs, see [23, 27]. As a direct consequence of Theorem 1 in [23], we have

THEOREM 3.1. *Let $\mathcal{X} = \{x_i | x_i \in E\}_{i=1}^{c(G)}$ be the set of edges in $G = (V, E)$ where $Z = (V_Z, E_Z)$ is the most likely cluster. Let $\mathcal{X}' = \{x'_i | x'_i \in [V]^2\}_{i=1}^{c(G)}$ be an alternative configuration of a graph $G' = (V, E')$ where $\forall x_i \in E_Z$, $x'_i = x_i$. If the null hypothesis is rejected under $\mathcal{X}$, then it is also rejected under $\mathcal{X}'$.*

Intuitively, as long as the edges within the subgraph constituting the most likely cluster are fixed, the null hypothesis is rejected no matter how the rest of the edges are shuffled around [23].

This theorem implies that:

1. $d_P(Z)$ does not change as long as its internal structure and the total number of reported edges outside Z remains the same. Intuitively, clusters defined by other subgraphs do not affect the discrepancy on Z. Formally, $d_P(Z)$ is independent of the value of $c^*(x)$ for any potential edge $x \in E \setminus E_Z$, as long as $c(\bar{Z})$ remains unchanged.

2. If the null hypothesis is rejected by d_P, then we can identify a specific cluster that is significant and implies this rejection. This distinguishes between saying "there exist significant clusters" and "the cluster Z is a significant cluster," where d_P can do the latter.

THEOREM 3.2. *d_P is individually most powerful for finding a single significant cluster: for a fixed false positive rate and for a given set of subgraphs tested, it is more likely to detect over-density than any other test statistic [30].*

This is a direct consequence of Theorem 2 in [23] and is paramount for effective cluster detection. (Its power is also explored in Section 6). It implies that:

3. We can determine the single potential edge $x \in [V]^2$ (or set of edges, such as those described by adding a vertex to the cluster) that will most increase the Poisson discrepancy of a cluster, and thus most decrease its p-value.

3.3 Spatial Scan Statistics and Local Modularity. Several local versions of modularity have been used to discover local community structure [11, 29]. Specifically, local modularity introduced in [38] is used to find the community structure around a given node. The *local modularity* of $Z \subseteq G$ measures the difference between the number of observed edges $c(Z)$ and the number expected, $\mu(Z)$,

$$\mathbb{M}_\gamma(Z) = c(Z) - \gamma\mu(Z).$$

One approach to clustering is to find the cluster Z that locally maximizes $\mathbb{M}_\gamma$. The γ parameter with default value 1, is a user specified knob [38] that scales the expected number of edges within Z under a Poisson random graph model. We observe that it effectively tunes the size of the clusters which optimize $\mathbb{M}_\gamma(Z)$. For a fixed cluster Z, $\mathbb{M}_\gamma$ can be treated as a linear function of γ, where its intersection with the Y-axis is $c(Z)$, and its slope is $-\mu(Z)$. $\mathbb{M}_\gamma$ for all $Z \in \mathcal{Z}$ forms a family of linear functions whose upper envelope corresponds to clusters that maximize $\mathbb{M}$ as γ varies. It can be observed that as γ increases, $c(Z)$ is non-increasing and $\mu(Z)$ is non-decreasing for the cluster Z that maximizes $\mathbb{M}_\gamma$.

It is important to distinguish $\mathbb{M}$ from d_P. While $\mathbb{M}$ measures the edge distance from the expected random graph, d_P measures the difference in how likely the total number of edges are to occur in a general random graph and how likely they are to occur in cluster Z and its complement as separate random graph models. To summarize, $\mathbb{M}$ *calculates the distance, and spatial scan statistics measure how unexpected this distance is, given Z.*

Several properties are also shared between d_P and $\mathbb{M}$. The tuning knob γ can be used in Poisson discrepancy to scale the expected number of edges in a cluster Z.

$$d_{P,\gamma}(Z) = r(Z) \log \frac{r(Z)}{\gamma b(Z)} + (1 - r(Z)) \log \left(\frac{1 - r(Z)}{1 - b(Z)} \right)$$

Technically, the function $d_{P,\gamma}$ describes the effect of scaling by γ the expected number of edges in a cluster Z (but not outside the cluster), while not allowing q, the parameter to model the random graph outside this cluster, to reflect γ. Thus in the same way as with $\mathbb{M}_\gamma$ for large γ, clusters need to have significantly more edges than expected to have a positive d_P value. The following lemma highlights this relationship.

LEMMA 3.1. *Consider two clusters Z_1 and Z_2 such that $d_{P,\gamma}(Z_1) = d_{P,\gamma}(Z_2)$ and that $c(Z_1) > c(Z_2)$. Then for any $\delta > 0$ we know $d_{P,\gamma+\delta}(Z_1) < d_{P,\gamma+\delta}(Z_2)$.*

The same property holds for $\mathbb{M}_\gamma$ and μ in place of $d_{P,\gamma}$ and c, respectively. That is, consider two *clusters Z_1 and Z_2 such that $\mathbb{M}_\gamma(Z_1) = \mathbb{M}_\gamma(Z_2)$ and that $\mu(Z_1) > \mu(Z_2)$. Then for any $\delta > 0$ we know $\mathbb{M}_{\gamma+\delta}(Z_1) < \mathbb{M}_{\gamma+\delta}(Z_2)$.*

Proof. We can write

$$d_{P,\gamma}(Z) = d_P(Z) - r(Z)\log\gamma,$$

thus as γ increases $d_{P,\gamma}(Z_1)$ will decrease faster than $d_{P,\gamma}(Z_2)$.

We can also write

$$\mathbb{M}_\gamma(Z) = c(G)(r(Z)-\gamma b(Z)) = \mathbb{M}_1(Z)-c(G)(\gamma-1)b(Z).$$

Thus the same argument applies.

This implies that for the discrepancy measure, we should expect the size of the optimal clusters to be smaller as we increase γ, as is empirically demonstrated in Section 6.3.

Using some machinery developed by Agarwal *et.al.* [2] we can create an ε-approximation of d_P with $O(\frac{1}{\varepsilon}\log^2|V|)$ linear functions with parameters $r(Z)$ and $b(Z)$, in the sense that the upper envelope of this set of linear functions will be within ε of d_P. We can interpret $\mathbb{M}_\gamma$ as a linear function whose slope is controlled by the value of γ. Figure 2 shows how $\mathbb{M}_1$ and $\mathbb{M}_2$, respectively, approximate d_P. Thus we can find the optimal cluster for $O(\frac{1}{\varepsilon}\log^2|V|)$ values of γ and let Z be the corresponding cluster from this set which has the largest value of $d_P(Z)$. Let Z^* be the cluster that has the largest value of $d_P(Z^*)$ among all possible clusters. Then $d_P(Z) + \varepsilon \geq d_P(Z^*)$.

However, a further study of Agarwal *et.al.* [1] showed that a single linear function (which would be equivalent to $\gamma = 2$ for $\mathbb{M}_\gamma$) approximated d_P on average to within about 95% for a problem using point sets. Note in Figure 2 how $\mathbb{M}_2$ seems to approximate d_P better than $\mathbb{M}_1$, at least for a large portion of the domain containing smaller clusters.

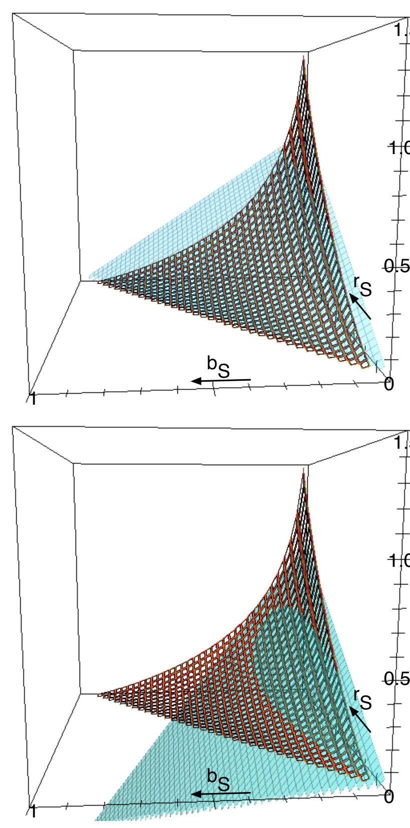

Figure 2: Comparison of d_P (gridded) to $\frac{1}{m}\mathbb{M}_1$ (transparent, upper panel) and $\frac{1}{m}\mathbb{M}_2$ (transparent, lower panel) over $(r(Z), b(Z)) \in [0,1]^2$ such that $r(Z) > b(Z)$. Recall that $r(Z)$ and $b(Z)$ are the actual and expected fraction of a graph's edges which lie in a particular cluster; for applications to large networks, a range of say $(0, 0.2)^2$ is most pertinent to clustering. For this range, $\mathbb{M}_2$ is shown to approximate d_P more closely than $\mathbb{M}_1$.

3.4 Spatial Scan Statistics and Bregman Divergences.

3.4 Spatial Scan Statistics and Bregman Divergences. Many Bregman divergences can be interpreted as spatial scan statistics. The KL-divergence, a Bregman divergence, when between two 2-point distributions, is equivalent to d_P up to a constant factor.

Banerjee *et.al.* [7] use Bregman divergences in a different way than does this paper. In the context of graph clustering, Bregman hard clustering finds a bi-partitioning and a representative for each of the partitions such that the expected Bregman divergence of the data points (edges) from their representatives is minimized [7]. For details and derivations, see [7].

Given a graph $G = (V, E)$, let x_i be a potential edge $x_i \in [V]^2$. Set $\mathcal{X} = \{x_i\}_{i=1}^{c(G)^2} \subseteq \mathbb{R}$ has probability measure μ^*. Cluster Z induces a bi-partition of G, Z and $\bar{Z}$. Let $\mu(Z)$ and $\mu(\bar{Z})$ be the induced measures on the partitions, where $\mu(Z) = \sum_{x_i\in[V_Z]^2}\mu^*(x_i)$ and $\mu(\bar{Z}) = \sum_{x_i\in\mathcal{X}\setminus[V_Z]^2}\mu^*(x_i)$. Let η_Z and $\eta_{\bar{Z}}$ denote the partition representative values.

$$\eta_Z = \sum_{x_i\in[V_Z]^2}\frac{\mu^*(x_i)}{\mu(Z)}c^*(x_i)$$

$$\eta_{\bar{Z}} = \sum_{x_i\in\mathcal{X}\setminus[V_Z]^2}\frac{\mu^*(x_i)}{\mu(\bar{Z})}c^*(x_i)$$

Define $\eta_i = \eta_Z$ for $x_i \in [V_Z]^2$, otherwise $\eta_i = \eta_{\bar{Z}}$. The Bregman clustering seeks to **minimize** the divergence between the two $c(G)^2$-point distributions

$$\{\langle c^*(x_1), c^*(x_2), \ldots, c^*(x_{c(G)^2})\rangle, \langle \eta_1, \eta_2, \ldots \eta_{c(G)^2}\rangle\}$$

We, on the other hand, **maximize** the KL-divergence between the two 2-point distributions

$$\{\langle r(Z), 1 - r(Z)\rangle, \langle b(Z), 1 - b(Z)\rangle\}.$$

But the methods do not conflict with each other. Their η_Z and $\eta_{\bar{Z}}$ variables are akin to p and q in the derivation of the scan statistic. By minimizing their Bregman divergence, they are trying to allow η_Z and $\eta_{\bar{Z}}$ to be as close to variables they represent as possible (i.e. η_Z should be close to each $c^*(x_i)$ for x_i defined on Z); and by maximizing our discrepancy we are separating p and q as much as possible, thus probabilistically representing the cluster edges and non-cluster edges more accurately with these ratios, respectively.

However, the Bregman divergence used by Banerjee *et.al.* [6, 7] typically assumes a less informative, uniform random graph model where $\mu^*(x_i) = \mu^*(x_j)$ for all i and j. Also when minimizing the KL-divergence, no edge at x_i would imply $c^*(x_i) = 0$, thus implying that the corresponding term of the KL-divergence, $c^*(x_i) \log \frac{c^*(x_i)}{\eta_i}$, is undefined. In their Bregman divergence model most similar to ours, this poses a problem as $c^*(x_i)$ can be 0 in our model; thus we do not compare the performance of these algorithms.

4 Algorithms.

In this section, we describe two bottom-up, greedy clustering algorithms. For a graph $G = (V, E)$ there are $2^{|V|}$ possible clusters (subgraphs) induced by subsets of V. That is, $|\mathcal{Z}| \leq 2^{|V|}$, where each cluster $Z = (V_Z, E_Z) \in \mathcal{Z}$ is induced by a vertex set $V_Z \subseteq V$. Clearly it is intractable to calculate discrepancy for every possible cluster through exhaustive search, as is often done with spatial scan statistics. We can, however, hope to find a locally optimal cluster. For an objective function $\Psi : 2^{|V|} \to \mathbb{R}$, define a *local maximum* as a subset $U \subseteq V$ such that adding or removing any vertex will decrease $\Psi(U)$. For some objective function Ψ and two vertex sets U and W, define

$$\partial\Psi(U, W) = \begin{cases} \Psi(U \cup W) - \Psi(U) & W \subset V \setminus U \\ \Psi(U \setminus W) - \Psi(U) & W \subset U \end{cases}$$

where $\Psi(V_Z) = d_P(Z)$ for $Z = (V_Z, E_Z) \subseteq \mathcal{Z}$ induced by V_Z. Let U^+ (resp. U^-) be the set of vertices in $N(U)$ (resp. U) such that $\partial\Psi(U, v) > 0$ for each vertex v. U_F^+ (resp. U_F^-) denotes the subset of U^+ (resp. U^-) that contains the fraction F of vertices with the largest $\partial\Psi(U, v)$ values. We now are set to describe two algorithms for refining a given subset U to find a local maximum in Ψ. Notice that both algorithms can be used to locally optimize any objective function, not limited to the Poisson discrepancy used here.

Greedy Nibble. The Greedy Nibble algorithm (Algorithm 1) alternates between an expansion phase and a contraction phase until the objective function cannot be improved. During expansion (resp. contraction) we iteratively add (resp. remove) the vertex that most improves the objective function until this phase can no longer improve the objective function.

Algorithm 1 Greedy-Nibble(U)

```
repeat
  expand = FALSE;    contract = FALSE
  $v^+ = \arg\max_{v \in N(U)} \partial\Psi(U, v)$.
  while $\partial\Psi(U, v^+) > 0$ do
    expand = TRUE
    $U = U \cup v^+$.
    $v^+ = \arg\max_{v \in N(U)} \partial\Psi(U, v)$.
  $v^- = \arg\max_{v \in U} \partial\Psi(U, v)$.
  while $\partial\Psi(U, v^-) > 0$ do
    contract = TRUE
    $U = U \setminus v^-$.
    $v^- = \arg\max_{v \in U} \partial\Psi(U, v)$.
until expand = FALSE and contract = FALSE
```

Greedy Chomp. The Greedy Chomp algorithm (Algorithm 2) is a more aggressive and faster version of the Greedy Nibble algorithm. Each phase adds a fraction F of the vertices which individually increase the Ψ value. If adding these $F|U^+|$ vertices simultaneously does not increase the overall Ψ value, then the fraction F is halved, unless $F|U^+| \leq 1$. Similar to simulated annealing, this algorithm makes very large changes to the subset at the beginning but becomes more gradual as it approaches a local optimum.

THEOREM 4.1. *Both the Greedy Nibble algorithm and the Greedy Chomp algorithm converge to a local maximum for Ψ.*

Proof. The algorithms increase the value of Ψ at each step, and there is a finite number of subsets, so they must terminate. By definition the result of termination is a local maximum.

4.1 Variations.

There are many possible heuristic variations on the above algorithms. We use Greedy Nibble because of its simplicity and Greedy Chomp because it performs similarly but faster.

In terms of initial seed selection, when time is not an issue, we recommend using each vertex as a seed. This ensures that every interesting cluster contains at least one seed. For larger graphs, randomly sampling some vertices as seeds should work comparably [38]. Clusters tend to be larger in this case, so most of them will still

Algorithm 2 Greedy-Chomp(U)

```
repeat
  expand = FALSE;  $F = 1$
  Calculate $U_F^+$
  while $\left(\partial\Psi(U, U_F^+) < 0 \text{ and } F|U^+| \geq 1\right)$ do
    $F = F/2$;  Update $U_F^+$
  while $\left(\partial\Psi(U, U_F^+) > 0\right)$ do
    expand = TRUE
    $U = U \cup U_F^+$.
    Calculate $U_F^+$;  $F = 1$
    while $\left(\partial\Psi(U, U_F^+) < 0 \text{ and } F|U^+| \geq 1\right)$ do
      $F = F/2$;  Update $U_F^+$
  contract = FALSE;  $F = 1$
  Calculate $U_F^-$
  while $\left(\partial\Psi(U, U_F^-) < 0 \text{ and } F|U^-| \geq 1\right)$ do
    $F = F/2$;  Update $U_F^-$
  while $\left(\partial\Psi(U, U_F^-) > 0\right)$ do
    contract = TRUE
    $U = U \setminus U_F^-$.
    Calculate $U_F^-$;  $F = 1$
    while $\left(\partial\Psi(U, U_F^-) < 0 \text{ and } F|U^-| \geq 1\right)$ do
      $F = F/2$;  Update $U_F^-$
until (expand = FALSE and contract = FALSE)
```

contain some seed. Alternatively, we could run another clustering algorithm to generate an initial seed and just use our greedy algorithms as a refinement.

In general, we use d_P as the objective function, but it is more prone to getting stuck in local maxima than is $\mathbb{M}$. Thus we enhance each initial seed by running the expansion phase of the algorithm with $\mathbb{M}_2$ since it closely approximates d_P as shown in Figure 2.

Since our emphasis is on the discrepancy measurement rather than clustering technique, we focus on illustrating that these simple clustering techniques based on Poisson discrepancy find locally significant clusters.

4.2 Complexity. It is difficult to analyze our algorithms precisely because they may alternate between the expansion and contraction phases many times. But Theorem 4.1 shows that this process is finite, and we notice that relatively few contraction steps are ever performed. Hence we focus on analyzing the worst case of the expansion phase in both algorithms.

Both algorithms are output dependent, where the runtime depends on the size of the final subset $|V_Z|$ and the size of its neighborhood $|N(V_Z)|$.

For Greedy Nibble we can maintain $N(V_Z)$ and calculate v^+ in $O(|N(V_Z)|)$ time since d_P only depends on the number of edges and K_{V_Z}. Thus the algorithm takes $O(|V_Z| \cdot |N(V_Z)|)$ time for each seed since v^+ needs to be calculated each iteration.

The Greedy Chomp algorithm could revert to the Greedy Nibble algorithm if F is immediately reduced to $1/|U^+|$ at every iteration. So worst case it is no faster than Greedy Nibble. In fact, each iteration takes $O(|N(V_Z)| \log |N(V_Z)|)$ time because the $\partial\Psi(U, v)$ values are sorted for all $v \in U^+$. However, in practice, a much smaller number of iterations are required because a large fraction of vertices are added at each iteration. If F were fixed throughout the algorithm, then we can loosely bound the runtime as $O(\log |V_Z| \cdot |N(V_Z)| \log |N(V_Z)|)$. Since F is generally large when most of the vertices are added, this is a fair estimate of the asymptotics.

This analysis is further evaluated empirically in Section 6.

5 Bipartite Extensions.

Many data sets which are applicable to this model (see Section 6) are bipartite graphs. Thus we derive here the key bipartite extensions of section 2 and section 3.

An undirected graph $G = (V, E)$ is *bipartite* if there is a partition $V = X \cup Y$ with X and Y disjoint and $E \subseteq \{\{u, v\} | u \in X, v \in Y\}$. A cluster is defined as a subgraph $Z = (V_Z, E_Z) = (X_Z \cup Y_Z, E_Z)$. The bipartite version of the Poisson random graph model does not allow self-loops and the total number of observed edges in G is $c(G) = \sum_{v_i \in X} k_i = \sum_{v_j \in Y} k_j$. The bipartite version of the local modularity and Poisson discrepancy follow naturally.

6 Analysis.

This section focuses on empirically exploring four aspects of this work. First, we investigate the power and runtime of our algorithms. Second, we use Poisson discrepancy as a tool to evaluate and compare different clustering algorithms. Third, we investigate properties of the clusters found by our algorithms maximizing Poisson discrepancy. Specifically, we show that varying the γ parameter can give reliable estimates of the size of clusters and we examine cases when distinct relevant clusters overlap. Finally, throughout this analysis we demonstrate that maximizing Poisson discrepancy reveals interesting and relevant clusters in real-world graphs.

6.1 Performance of Our Algorithms.

Runtime on Real-world Datasets. We demonstrate the effectiveness of our algorithm on a variety of real world datasets, the sizes of which are summarized in Table 1.

The DBR dataset describes connections between threads (the set X) and users (the set Y) of the

Duke Basketball Report message board from 2.18.07 to 2.21.07. Other datasets include Web[1] which links websites and users, Firm which links AP articles and business firms, Movie which links reviewers and movies through positive reviews, and Gene which links genes and PubMed articles. In each case, high discrepancy clusters can be used to either provide advertising focus onto a social group or insight into the structure of the dataset.

Dataset	$\|X\|$	$\|Y\|$	$\|E\|$	Nibble	Chomp
DBR	68	97	410	0.025	0.018
Web	1023	1008	4230	0.179	0.049
Firm	4950	7355	30168	9.377	0.251
Movie	1556	57153	1270473	-	32.91
Gene	6806	595036	1578537	-	242.7

Table 1: Sizes of real-world datasets and the average runtime in seconds for the Greedy Nibble and Greedy Chomp algorithms starting with singleton seeds. Runtimes for Web and Firm were generated with 100 random samples. Runtimes for Movie and Gene were generated with 50 random samples.

Power tests. The *power* of the test is the probability that the test statistic exceeds a critical value under some alternative hypothesis compared to some null hypothesis [19]. To calculate the power of our algorithm, we synthetically insert significant clusters into 100 random graphs and report the fraction of these graphs where our algorithm found the injected cluster.

In particular, we generate bipartite graphs using the Poisson random graph model such that $|X| = |Y| = 100$ and $|E| = 500$ where the expected degrees of vertices vary between 3 and 7. To inject a significant cluster, we choose a random set of vertices $V_Z = X_Z \cup Y_Z$, where $X_Z \subset X$, $Y_Z \subset Y$, and $|X_Z| = |Y_Z| = 15$. We increase the probability that an edge is chosen between two vertices in X_Z and Y_Z by a factor of ρ. We scale the probabilities on the other pairs of vertices in the graph so that each vertex retains its original expected degree. By choosing an appropriate value of ρ, we can generate graphs with the same expected degree sequence and whose injected cluster is expected to be significant. We repeat this process until we generate 100 graphs whose injected clusters have a p-value less than 0.005.

We run Greedy Nibble and Greedy Chomp using each vertex as a seed. We say that we successfully found the injected cluster Z induced by $V_Z = X_Z \cup Y_Z$ if the algorithm returns a cluster $\hat{Z} = X_{\hat{Z}} \cup Y_{\hat{Z}}$ such that $d(X_{\hat{Z}}, X_Z) \leq |X_Z|$ and $d(Y_{\hat{Z}}, Y_Z) \leq |Y_Z|$ and it either has p-value less than 0.05 or is among the top 5 clusters found.

We report the power of the algorithms in Table 2. It shows that 85% of the time Greedy Chomp locates the injected clusters. Note that we have used a relaxed criteria to determine when an injected cluster is found by our algorithm; a tighter qualification would reduce this power measurement.

Algorithm	Nibble	Chomp
Power	0.83	0.85

Table 2: Power for Greedy Nibble and Greedy Chomp tested on graphs of size 100×100 with an injected cluster of size 15×15 with p-value at most 0.005.

6.2 Algorithm Comparison.

Poisson discrepancy provides an absolute measure of cluster significance. This allows comparison between different clusterings of the same graph. We can evaluate the effectiveness of existing clustering algorithms by calculating the discrepancy of the clusters they find. Furthermore, we can enhance these clusters using Greedy Nibble or Greedy Chomp to maximize their discrepancy and evaluate how far from the local optimum these clusters used to be. We illustrate this by running the MCL algorithm [41] and the Ncut algorithm [40] on DBR. MCL generated 35 clusters and we fixed the number of clusters in Ncut to be 10. We report the top 4 clusters with the highest d_P value in Table 3. Ncut seems to find clusters with higher discrepancy. We then use clusters found by MCL and DBR as seed sets in the Greedy Chomp, further refining them in terms of their discrepancy. Ncut tends to do better than MCL in finding clusters within closer proximity of discrepancy local maxima.

MCL	0.0376	0.0248	0.0223	0.0211
MCL+Chomp	0.0667	0.0790	0.0620	0.0698
Ncut	0.0692	0.0529	0.0527	0.0473
Ncut+Chomp	0.0757	0.0688	0.0635	0.0713

Table 3: d_P values of top 4 clusters found with MCL and Ncut on DBR and the d_P values after their refinement with Greedy Chomp.

6.3 Cluster Properties.

Cluster Overlap Analysis. Many graph clustering methods partition the data into disjoint subsets, essentially making each a cluster. Our approach finds clusters which may overlap, and it considers the rest of the graph uninteresting instead of forcing it to be a clus-

[1]We thank Neilsen//Netratings, Inc., who provided the WEB dataset to us, for permission to use its data in this investigation.

ter. We examine the top 6 clusters found from Greedy Nibble on DBR in an overlap matrix (Table 4). We use each vertex in X as a singleton seed set. The 1st, 2nd, 3rd and 5th clusters are very similar, representing a consistent set of about 13 threads on topics discussing the performance of players and strategy. The 4th cluster contains 14 threads which were posted by users who seem more interested in the site as a community and are more gossiping than analyzing. The 6th cluster contains an overlap of the above two types of topics and users: users who are interested in the community, but also take part in the analysis. The rest of the threads (about 60) deal with a wider and less focused array of topics.

C	1	2	3	4	5	6	d_P	p-value
1	13	12	12	1	12	8	0.0783	0.009
2	12	12	11	1	12	8	0.0764	0.019
3	12	11	14	0	11	7	0.0754	0.020
4	1	1	0	14	1	6	0.0749	0.020
5	12	12	11	1	13	8	0.0718	0.022
6	8	8	7	6	8	16	0.0703	0.077

Table 4: Overlap of threads among the top 6 clusters for DBR with their d_P and p-values found with the Greedy Nibble algorithm.

γ as a Resolution Scale. For our algorithm, as γ varies, we observe an inverse linear correlation between γ and the average cluster size (Figure 3). We also show that as γ varies, our algorithm locates clusters that are statistically significant on different scales, and that their contents remain meaningful.

This near-linear correlation makes γ a reliable resolution scale for our clustering algorithm. As γ goes to 0, the algorithm produces the whole graph as a cluster. As γ goes to infinity, the algorithm produces trivial singletons. The flexibility to modify γ allows a user to balance the importance of the statistical significance of the clusters found, maximized by $d_{P,\gamma}$, and their preferred size weighted by γ. This helps resolve issues (previously noted about modularity by [17]) about the preferred size of clusters which optimize d_P. For instance when searching for more focused clusters of smaller size, a reasonable γ weight can be easily inferred.

Cluster content and γ. Manual evaluation of the results show that the contents of the clusters remain meaningful and useful as γ is varied. For example, the top clusters found on the Movie dataset with $\gamma = 200$ are shown to be popular box office movies in the 90's as they are consistently reviewed favorably by various reviewers. The top two clusters found on the Gene dataset with $\gamma = 200$ are genes in the UDP glucuronosyltransferase 1 and 2 family. The 4th ranked cluster consists of genes such as MLH and PMS, both are related to DNA mismatch repair. The 8th ranked cluster for $\gamma = 200$ persists as the top ranked cluster when $\gamma = 600$; it consists of several genes for the zona pellucida glycoprotein, i.e ZP1 and ZP3A.

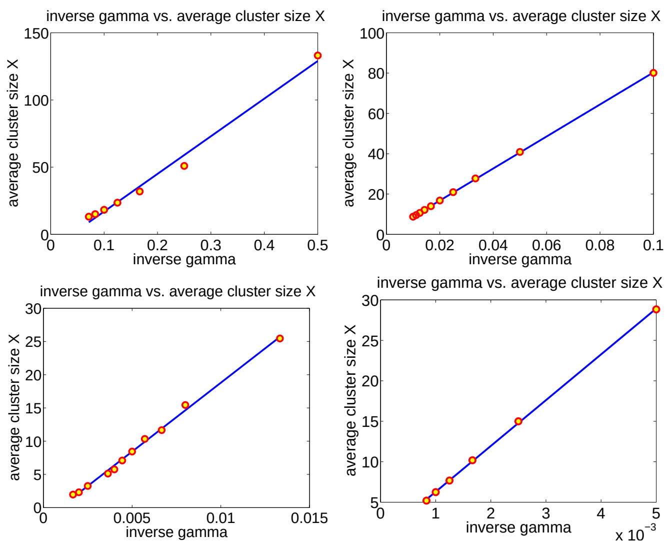

Figure 3: Plot of $1/\gamma$ vs. average cluster size on Web (top left), Firm (top right), Movie (bottom left), and Gene (bottom right).

As γ increases, the nontrivial clusters with a high $d_{P,\gamma}$-discrepancy should generally be much denser internally, since the ratio between the actual internal edges and expected edges should be greater than a given γ. On the other hand, clusters which persist as the top ranked clusters as γ increases are those that are most statistical significant in a dynamic setting. As γ increases, we would expect the number of such extremely anomalous clusters to decrease. For example, as shown in Figure 4 for the Web data set, as γ increases, the number of outlier clusters with comparatively very large discrepancy decreases. For $\gamma = 4$, many clusters seem to be significantly larger than the large component, while with $\gamma = 6$ and $\gamma = 8$ there are very few. Finally, with $\gamma = 10$ all clusters are basically in the same component.

The identification of clusters of varying size but consistently high statistical significance suggests that real-world networks are characterized by many different levels of granularity. This result is consistent with, e.g, the contrasting findings of [37] and [31], where clusters of vastly different sizes but comparable modularities are detected in the same data set. This finding calls into question the wide variety of clustering methods which are only designed to detect one cluster for a given region or group of nodes, and a further study would be of interest.

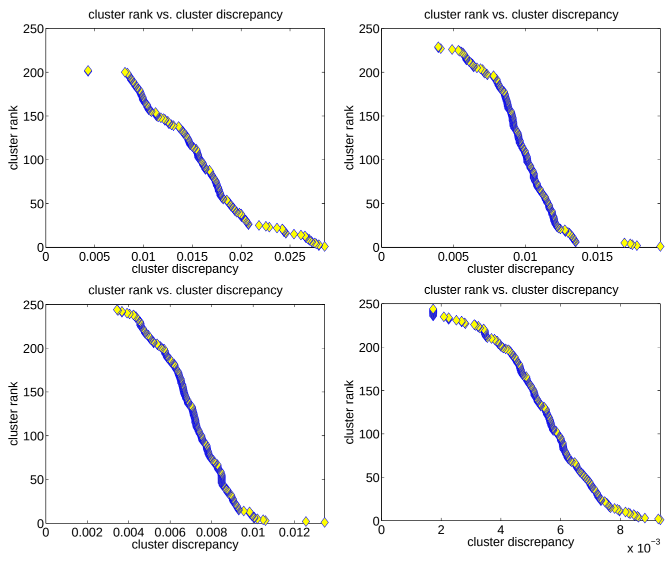

Figure 4: Web: cluster rank vs. cluster discrepancy with each X vertex used as a singleton seed set, $\gamma = 4$ (top left), $\gamma = 6$ (top right), $\gamma = 8$ (bottom left), $\gamma = 10$ (bottom right). Top ranked clusters appear at the bottom right of each figure.

7 Conclusions.

The main contribution of this paper is the introduction of a quantitative and meaningful measure, *Poisson discrepancy*, for clusters in graphs, derived from spatial scan statistics on point sets. According to our definition, the higher the discrepancy, the better the cluster. We identify interesting relations between Poisson discrepancy, local modularity, and Bregman divergences.

To illustrate the usefulness of this statistic, we describe and demonstrate two simple algorithms which find local optima with respect to the spatial scan statistic. In the context of real-world and synthetic datasets that are naturally represented as bipartite graphs, this method has identified individual clusters of vertices that are statistically the most significant. These clusters are the least likely to occur under a random graph model and thus best identify closely-related groups within the network. Our model places no restrictions on overlapping of clusters, thus allowing a data point to be classified into two or more groups to which it belongs. As our greedy algorithms are the simplest and most intuitive approach, it remains an open problem to find more effective algorithms to explore the space of potential subgraphs to maximize the Poisson discrepancy. Notice that Poisson discrepancy can also detect regions that are significantly under-populated by requiring $p < q$ in the alternative hypothesis.

Similarly the spatial scan statistic Bernoulli model for graph clustering can be derived from the corresponding model for point sets. However, this model requires that each potential edge be chosen with equal probabilities, regardless of the degree of a vertex. Also, under this model each pair of vertices can have at most one edge.

In summary, we argue that a graph cluster should be statistically justifiable, and a quantitative justification comes from a generalization of spatial scan statistics on graphs, such as the Poisson discrepancy.

References

[1] D. Agarwal, A. McGregor, J. M. Phillips, S. Venkatasubramanian, and Z. Zhu. Spatial scan statistics: Approximations and performance study. In *12 Annual ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006.

[2] D. Agarwal, J. M. Phillips, and S. Venkatasubramanian. The hunting of the bump: On maximizing statistical discrepancy. In *Proceedings 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1137–1146, 2006.

[3] W. Aiello, F. Chung, and L. Lu. Random evolution in massive graphs. *Handbook of massive data sets*, pages 97–122, 2002.

[4] R. Albert, H. Jeong, and A. Barabasi. The diameter of the world wide web. *Nature*, 401:130, 1999.

[5] G. D. Bader and C. W.V. Hogue. An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics*, 4:2, 2003.

[6] A. Banerjee, I. S. Dhillon, J. Ghosh, S. Merugu, and D. S. Modha. A generalized maximum entropy approach to bregman co-clustering and matrix approximation. In *Proceedings 10th Annual ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004.

[7] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh. Clustering with bregman divergences. *Journal of Machine Learning*, 6:1705–1749, 2005.

[8] A. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286:509, 1999.

[9] J. Baumes, M. Goldberg, M. Krishnamoorthy, M. Magdon-Ismail, and N. Preston. Finding communities by clustering a graph into overlapping subgraphs. *International Conference on Applied Computing (IADIS 2005)*, Feb 2005.

[10] F. Chung and L. Lu. Connected components in random graphs with given expected degree sequences. *Annals of Combinatorics*, 6(2):125–145, 2002.

[11] A. Clauset. Finding local community structure in networks. *Physical Review E*, 72, 2005.

[12] A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *Phys. Rev. E*, 70:066111, 2004.

[13] I. S. Dhillon, Y. Guan, and B. Kulis. Weighted graph cuts without eigenvectors: A multilevel approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007 (to appear).

[14] I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. In *Proceedings 9th Annual ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003.

[15] J. Duch and A. Arenas. Community detection in complex networks using extremal optimization. *Physical Review E*, 72:027104, 2005.

[16] P. Erdös and A. Rényi. On random graphs. *Publ. Math. Debrecen*, 1959.

[17] S. Fortunato and M. Barthelemy. Resolution limit in community detection. *Proc. Natl. Acad. Sci. USA*, 104:36, 2007.

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

[19] J. Glaz, J. Naus, and S. Wallenstein. *Scan Statistics*. Springer-Verlag,New York, 2001.

[20] S. Itzkovitz, N. Kashtan, G. Ziv, and U. Alon. Subgraphs in random networks. *Physical Review E*, 68, 2003.

[21] J. M. Jennings, F. C. Curriero, D. Celentano, and J. M. Ellen. Geographic identification of high gonorrhea transmission areas in baltimore, maryland. *American Journal of Epidemiology*, 161(1):73–80, 2005.

[22] M. Koyuturk, W. Szpankowski, and A. Grama. Assessing significance of connectivity and conservation in protein interaction networks. *Journal of Computational Biology*, 14(6):747–764, 2007.

[23] M. Kulldorff. A spatial scan statistic. *Communications in Statistics: Theory and Methods*, 26:1481–1496, 1997.

[24] M. Kulldorff. Spatial scan statistics: models, calculations, and applications. *Scan Statistics and Applications*, pages 303–322, 1999.

[25] M. Kulldorff. Prospective time-periodic geographical disease surveillance using a scan statistic. *Journal of the Royal Statistical Society, Series A*, 164:61–72, 2001.

[26] M. Kulldorff. *SaTScan User Guide*, 7.0 edition, August 2006.

[27] M. Kulldorff, T. Tango, and P. J. Park. Power comparisons for disease clustering tests. *Computational Statistics & Data Analysis*, 42(4):665–684, April 2003.

[28] F. J. McErlean, D. A. Bell, and S. I. McClean. The use of simmulated annealing for clustering data in databases. *Information Systems*, 15(2):233–245, 1990.

[29] S. Muff, F. Rao, and A. Caflisch. Local modularity measure for network clusterizations. *Phys Rev E Stat Nonlin Soft Matter Phys*, 72(5 Pt 2), November 2005.

[30] D. B. Neill and A. W. Moore. Rapid detection of significant spatial clusters. In *Proceedings 10th Annual ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004.

[31] M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69:066133, 2004.

[32] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E*, 74:036104, 2006.

[33] G. Palla, I. Derenyi, I. Farkas, and T. Viscek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435:814–818, 2005.

[34] D. A. Plaisted. A heuristic algorithm for small separators in arbitrary graphs. *SIAM J. Comput.*, 19(2):267–280, 1990.

[35] A. Pothen, H. Simon, and K. Liou. Partitioning sparse matrices with eigenvalues of graphs. *SIAM J. Matrix Anal. Appl.*, 11(3):430–452, 1990.

[36] C. E. Priebe, J. M. Conroy, D. J. Marchette, and Y. Park. Scan statistics on enron graphs. *Computational & Mathimatical Organization Theory*, 11(3):229–247, October 2005.

[37] J. M. Pujol, J. Bejar, and J. Delgado. Clustering algorithm for determining community structure in large networks. *to appear Physical Review E*, 2006.

[38] J. Reichardt and S. Bornholdt. Statistical mechanics of community detection. *Phys. Rev. E*, 74:016110, 2006.

[39] R. Sharan, T. Ideker, B. Kelley, R. Shamir, and R. Karp. Identification of protein complexes by comparative analysis of yeast and bacterial proten interaction data. *Journal of Computational Biology*, 12(6):835–846, 2005.

[40] J. Shi and J. Malik. Normalized cutes and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, August 2000.

[41] S. von Dongen. *Graph Clustering by Flow Simulation*. PhD thesis, University of Utrecht, May 2000.

[42] D. M. Wilkinson and B. A. Huberman. A method for identifying communities of related genes. *Proc. Natl. Acad. Sci. USA*, 101(14):1073, 2004.