Outlier Detection from Network Data with Subnetwork Interpretation

Xuan-Hong Dang, Arlei Silva, Ambuj Singh
University of California Santa Barbara
{xdang,arlei,ambuj}@cs.ucsb.edu

Ananthram Swami
Army Research Laboratory
ananthram.swami.civ@mail.mil

Prithwish Basu
Raytheon BBN Technologies
pbasu@bbn.com

***Abstract*—Detecting a small number of outliers from a set of data observations is always challenging. This problem is more difficult in the setting of multiple network samples, where computing the anomalous degree of a network sample is generally not sufficient. In fact, explaining why a given network is exceptional, expressed in the form of subnetwork, is also equally important. We develop a novel algorithm to address these two key problems. We treat each network sample as a potential outlier and identify subnetworks that help discriminate it from nearby samples. The algorithm is developed in the framework of network regression combined with the constraints on both network topology and L1-norm shrinkage to perform subnetwork discovery. Our method thus goes beyond subspace/subgraph discovery. We also show that the developed method converges to a global optimum. Empirical evaluation on various real-world network datasets demonstrates the advantages of our algorithm over various baseline methods.**

I. Introduction

Detecting and characterizing exceptional patterns is an important task in many domains ranging from fraud detection, environmental surveillance, to various health care applications [1]. This problem is often referred to as *outlier* or *anomaly* detection in the literature. Although identifying anomalous subjects has been widely studied in high dimensional data and recently extended to the network context [1], the problem remains very challenging. In the network setting, most existing works focus on searching individual nodes [10], or groups of linked nodes [8] whose structures or behaviors are irregular. Though these studies have provided intuitive concepts about outlying patterns defined in the respect of network connectivity, most results are limited to the setting of a single static network. Other recent studies have extended the scope of analysis to evolving networks [2], but the focus is on event/change detection where the temporal dimension is a key factor for defining outliers.

In this paper, we address the problem of identifying anomalous networks from a database of multiple network samples while at the same time investigating *why* a network is exceptional. An outlier is defined at the global level of an entire network sample but we use local subnetworks to explain its exceptionality. Although the outlierness of a network sample can be quantified via the outlier degree, such a single measure only bears limited explanatory information [16] since it lacks the capability of showing in what data view, i.e. local subnetworks, an anomalous network is most exceptional. Moreover, although two networks may have similar outlier degrees, the local subnetworks that make them abnormal might be quite different since the anomalous networks themselves are usually not homogeneous. For example, exploring a database of gene networks for outliers can lead to the isolation of subjects suffering from cancer. However, the gene pathway (local subnetwork) that causes the disease can vary from subject to subject due to the complexity of the disease [14], or even depending on different stages of the disease. Spotting an unhealthy subject is generally not sufficient. Figuring out what abnormal gene subnetwork leads to the disease is usually more important since it helps to develop possible and effective treatments.

We develop a novel algorithm that exploits network regression models combined with network topology regularization to concurrently address the two important problems mentioned above. Specifically, we treat each network sample as a potential outlier and determine local subnetworks that help discriminate it from nearby regular network samples. Our objective function is formulated under the framework of network regression where we first upsample the outlier candidate network in order to make the binary regression problem balanced. The objective function is then regularized by the network topology and further penalized by L1-norm shrinkage to perform subnetwork discovery. It can be shown that the combined objective function has a form closely related to the dual SVM [9,11], which can be further optimized in the primal form using Newton's method. The objective function is proven to be convex, which is key to guaranteeing the convergence of the algorithm. Our algorithm, therefore, goes beyond the simple strategy of subspaces/subgraphs examination by directly learning the most discriminative subnetworks with respect to each network sample. Consequently, the outlier degree can be appropriately computed within the space spanned by these selected subnetworks and, collectively, they form a ranking of all network samples based on the outlier scores.

II. Regression on Networks

Definition 1: A *network sample* is a triple $\mathcal{N}_k = (\mathcal{V}_k, E_k, \mathcal{F})$, where $\mathcal{V}_k = \{v_1, v_2, \ldots, v_n\}$ is a set of nodes, $E_k \subseteq \mathcal{V}_k \times \mathcal{V}_k$ is a set of undirected edges, and $\mathcal{F}$ is a function labeling each node with a real number.

Let $\mathcal{DB} = \{\mathcal{N}_1, \mathcal{N}_2, \ldots, \mathcal{N}_m\}$ be a collection of m network samples. In mining outlying networks from $\mathcal{DB}$, we aim to compute an anomaly score for each network sample in $\mathcal{DB}$ and at the same time, to uncover subnetworks that show the most exceptional properties of the network under examination.

We view each network sample as a potential candidate outlier while comparing its properties against its K nearby networks (based on some network distance measures, e.g. cosine distance between node values [7]). Therefore, a network sample can be a *local* outlier rather than a *global* one [5], as both network distribution and the outliers themselves can be heterogeneous and one should not presume any canonical form for the distribution. Let us denote $\mathcal{N}_o$ as an outlier network candidate, and $\mathcal{N}_k$ as one of its K neighboring networks (we use the same index k as in Def.1 for simplificity). We can capture the node-values of a network sample $\mathcal{N}_k$ by a vector $\mathbf{x}_k$ in space $\mathbb{R}^n$. Under the vector format, we optimize the following regression function for each $\mathcal{N}_o$:

$$\arg\min_{\mathbf{w}} L(\mathbf{w}) = (\mathbf{x}_o^T\mathbf{w} - z_o)^2 + \sum_{k=1}^{K}(\mathbf{x}_k^T\mathbf{w} - z_k)^2 \ s.t.\ |\mathbf{w}|_1 \leq 1 \quad (1)$$

where $\mathbf{x}_o$ is the vector of local node values for network $\mathcal{N}_o$; and $z_o = -1$ while $z_k = 1$ if $\mathcal{N}_k$ is among the K neighboring networks of $\mathcal{N}_o$; $|\mathbf{w}|_1$ is the L1-norm of vector $\mathbf{w}$. The main role of $|\mathbf{w}|_1$ is to set many coefficients in $\mathbf{w}$ to zero if the corresponding nodes are less predictive. It is worth mentioning that in a conventional case, one can constrain $|\mathbf{w}|_1 \leq c$ [9] for $c > 0$. However, c is only a scalar and can be replaced by 1 by dividing both $\mathbf{w}$ and the predicted labels z_o, z_k's by c.

It is possible to see that our Eq.(1) resembles the form of Lasso regression [9]. However, there are two challenging issues in optimizing Eq.(1). First, our regression model is highly imbalanced since we have only a single outlier candidate but a large number of neighboring inliers. In dealing with this issue, we adopt a simple approach of upsampling the outlier candidate. Essentially, $(K-1)$ new samples will be generated (for the outlier class) following the normal distribution with $\mathbf{x}_o$ as the mean vector, and the covariance matrix as the one computed from the statistics of K neighboring networks. By doing so, we assure that variations at each node/dimension of the outlier class are not generated randomly but resemble the ones from the inlier class, and thus minimize the impact on the explanation quality of the outlier.

The second, more challenging, issue in optimizing Eq.(1) is that the function is not directly differentiable. The solution is at best only suboptimal using methods like sub-gradient descent [17], in which each component of $\mathbf{w}$ is optimized individually and sequentially. Moreover, such a solution is less efficient given the large number of nodes in the networks. We thus handle the L1-norm in a more general setting [17] by representing $\mathbf{w}$ using two non-negative variables $\mathbf{w}^+$ and $\mathbf{w}^-$, that are respectively defined as $\mathbf{w}^+ = \max(0, \mathbf{w})$ and $\mathbf{w}^- = -\min(0, \mathbf{w})$. Hence, $\mathbf{w} = \mathbf{w}^+ - \mathbf{w}^-$. We denote the new variable $\tilde{\mathbf{w}} = [\mathbf{w}^+; \mathbf{w}^-] \in \mathbb{R}^{2n}$. Coefficients in $\tilde{\mathbf{w}}$ are thus *all* non-negative. Now, in combination with the upsampling reasoning above, Eq.(1) can be reformulated in the matrix form:

$$\arg\min_{\tilde{w}_i \geq 0} L(\tilde{\mathbf{w}}) = \left\| [X^{(o)}, -X^{(o)}]\tilde{\mathbf{w}} - \mathbf{z} \right\|_2^2 \quad s.t. \sum_{i=1}^{2n} \tilde{w}_i \leq 1 \quad (2)$$

where $X^{(o)}$ is the matrix with the first K rows as the vectors $\mathbf{x}_k$'s, and the last K rows as $\mathbf{x}_o$ and its $(K-1)$ sampling vectors. Correspondingly, the first K entries of vector $\mathbf{z}$ are $+1$, predicting $\mathbf{x}_k$'s as inliers, while the last K entries are -1, predicting $\mathbf{x}_o$ and the upsamples as outliers.

III. Role of Network Topology

We add the network structure information as a constraint in learning w. Intuitively, if two nodes are connected in the network, their behaviors will mutually impact each other and, consequently, the corresponding values in $\mathbf{w}$ should be similar. Towards modeling this network influence, we first define a graph that generalizes the network topology of both $\mathcal{N}_o$ and its K neighboring networks.

Definition 2: Let $\mathcal{DB}_o = \{\mathcal{N}_o, \mathcal{N}_p, \ldots, \mathcal{N}_q\}$ be the set of networks that involves the outlier candidate network $\mathcal{N}_o$ and its K neighboring networks $\{\mathcal{N}_p, \ldots, \mathcal{N}_q\}$. We define $G^{(o)} = (\mathcal{V}, E, A)$ as a graph summarizing the network topology of $\mathcal{DB}_o$, where $\mathcal{V}$ is the union of all $K+1$ vertex sets; and E is the union of $K+1$ edge sets. Each edge $E(i,j) \in E$ is associated with a positive weight $A(i,j)$ defined as the frequency of the corresponding edge in either $\mathcal{N}_o$ or in its neighboring networks $\mathcal{N}_k$'s, i.e., $A(i,j) = \max(E_o(i,j), (1/K) \times \sum_k E_k(i,j))$ with $E_k(i,j) = 1$ if v_i connects v_j in network $\mathcal{N}_k \in \mathcal{DB}_o$.

We regularize $\mathbf{w}$ using $G^{(o)}$'s topology in order to favor subnetworks that are frequently seen in $\mathcal{N}_o$ and/or in its K neighboring networks. Let $deg(i) = \sum_{v_i \sim v_j} A(i,j)$ be the vertex degree of v_i in $G^{(o)}$. Accordingly, $L^{(o)}$ is defined as the normalized Laplacian matrix of $G^{(o)}$. This can be used as a regularization constraint on $\mathbf{w}$ by minimizing:

$$\mathbf{w}^T L^{(o)} \mathbf{w} = \sum_{v_i}\sum_{v_j} \left(\frac{w_i}{\sqrt{deg(i)}} - \frac{w_j}{\sqrt{deg(j)}} \right)^2 A(i,j) \geq 0 \quad (3)$$

In order to appropriately incorporate this network-constrained penalty into our objective function formulated in Eq.(2), we need the following lemma.

Lemma 1: Given $\tilde{\mathbf{w}} = [\mathbf{w}^+; \mathbf{w}^-]$, the following equation is satisfied:

$$\mathbf{w}^T L^{(o)} \mathbf{w} = \tilde{\mathbf{w}}^T \begin{bmatrix} L^{(o)} & -L^{(o)} \\ -L^{(o)} & L^{(o)} \end{bmatrix} \tilde{\mathbf{w}} \quad (4)$$

Lemma 1 ensures that the network constraint penalty can also be represented using the transformed variable $\tilde{\mathbf{w}}$. Following this, we recast our objective function in Eq.(2):

$$\arg\min_{\tilde{w}_i\geq 0}\mathcal{L}(\tilde{\mathbf{w}}) = \left\|[X^{(o)},-X^{(o)}]\tilde{\mathbf{w}}-\mathbf{z}\right\|_2^2 + \lambda_1\tilde{\mathbf{w}}^T\begin{bmatrix} L^{(o)} & -L^{(o)} \\ -L^{(o)} & L^{(o)}\end{bmatrix}\tilde{\mathbf{w}} \;\; s.t. \sum_{i=1}^{2n}\tilde{w}_i \leq 1 \tag{5}$$

Notice that if $|\tilde{\mathbf{w}}|_1 < 1$ in Eq.(5), then the upper bound is inactive and coefficients in $\tilde{\mathbf{w}}$ will be widely non-zero. In other words, the majority of nodes in the graph will be selected. This solution is obviously undesirable. Therefore, in order to ensure that only subnetworks with the most explanatory information are used for $\mathcal{N}_o$, this constraint should always be tight. This means that we can safely use the equality constraint $\sum_{i=1}^{2n}\tilde{w}_i = \mathbf{1}^T\tilde{\mathbf{w}} = 1$ [3,19], with $\mathbf{1}$ as the vector of all 1. In this setting, the first term in Eq.(5) can be rewritten as:

$$\begin{aligned}\left\|[X^{(o)},-X^{(o)}]\tilde{\mathbf{w}}-\mathbf{z}\right\|_2^2 &= \left\|[X^{(o)},-X^{(o)}]\tilde{\mathbf{w}}-\mathbf{z}\mathbf{1}^T\tilde{\mathbf{w}}\right\|_2^2 \\ &= \left\|[X^{(o)}-\mathbf{z}\mathbf{1}^T,-(X^{(o)}+\mathbf{z}\mathbf{1}^T)]\tilde{\mathbf{w}}\right\|_2^2 = \|[X_1,-X_2]\tilde{\mathbf{w}}\|_2^2\end{aligned} \tag{6}$$

in which X_1 and X_2 respectively denote $(X^{(o)}-\mathbf{z}\mathbf{1}^T)$ and $(X^{(o)}+\mathbf{z}\mathbf{1}^T)$. Consequently, we can combine two terms in Eq.(5) into a single quadratic form by using the following lemma.

Lemma 2: Let $L^{(o)}$ be decomposed into $L^{(o)} = S^TS$ and $\tilde{X} = \left(\begin{bmatrix} X_1 \\ \sqrt{\lambda_1}S\end{bmatrix}\begin{bmatrix} -X_2 \\ -\sqrt{\lambda_1}S\end{bmatrix}\right)$. Then:

$$\|[X_1,-X_2]\tilde{\mathbf{w}}\|_2^2 + \lambda_1\tilde{\mathbf{w}}^T\begin{bmatrix} L^{(o)} & -L^{(o)} \\ -L^{(o)} & L^{(o)}\end{bmatrix}\tilde{\mathbf{w}} = \tilde{\mathbf{w}}^T\tilde{X}^T\tilde{X}\tilde{\mathbf{w}} \tag{7}$$

Proof: On one hand, the expansion of the first term gives us:

$$\|[X_1,-X_2]\tilde{\mathbf{w}}\|_2^2 = \tilde{\mathbf{w}}^T\begin{bmatrix} X_1^TX_1 & -X_1^TX_2 \\ -X_2^TX_1 & X_2^TX_2\end{bmatrix}\tilde{\mathbf{w}} \tag{8}$$

On the other hand, as $L^{(o)}$ is a normalized Laplacian matrix, it can be eigen-decomposed into $L^{(o)} = U\Sigma U^T = S^TS$ where $S = \Sigma^{1/2}U^T$ and U, Σ are respectively the matrices of eigenvectors and non-negative eigenvalues of $L^{(o)}$. Therefore:

$$\begin{aligned}&\tilde{\mathbf{w}}^T\begin{bmatrix} X_1^TX_1 & -X_1^TX_2 \\ -X_2^TX_1 & X_2^TX_2\end{bmatrix}\tilde{\mathbf{w}} + \lambda_1\tilde{\mathbf{w}}^T\begin{bmatrix} L^{(o)} & -L^{(o)} \\ -L^{(o)} & L^{(o)}\end{bmatrix}\tilde{\mathbf{w}} \\ =&\tilde{\mathbf{w}}^T\begin{bmatrix} X_1^TX_1+\lambda_1S^TS & -X_1^TX_2-\lambda_1S^TS \\ -X_2^TX_1-\lambda_1S^TS & X_2^TX_2+\lambda_1S^TS\end{bmatrix}\tilde{\mathbf{w}} \\ =&\tilde{\mathbf{w}}^T\begin{bmatrix}\begin{bmatrix} X_1 \\ \sqrt{\lambda_1}S\end{bmatrix}^T \\ \begin{bmatrix} -X_2 \\ -\sqrt{\lambda_1}S\end{bmatrix}^T\end{bmatrix}\times\begin{bmatrix}\begin{bmatrix} X_1 \\ \sqrt{\lambda_1}S\end{bmatrix} & \begin{bmatrix} -X_2 \\ -\sqrt{\lambda_1}S\end{bmatrix}\end{bmatrix}\tilde{\mathbf{w}} \\ =&\tilde{\mathbf{w}}^T\tilde{X}^T\tilde{X}\tilde{\mathbf{w}}\end{aligned} \tag{9}$$

From the 1st row to the 2nd row, we have used the fact that both X_1 and X_2 have the same size of $2K\times n$ while $L^{(o)}$ has the size of $n\times n$. So, the pairwise addition of the two matrices in the 2nd row is obvious. □

Given Lemma 2 in combination with the previous results, we can rewrite Eq.(5) as follows:

$$\arg\min_{\tilde{w}_i\geq 0}\mathcal{L}(\tilde{\mathbf{w}}) = \tilde{\mathbf{w}}^T\tilde{X}^T\tilde{X}\tilde{\mathbf{w}} + \lambda_2\tilde{\mathbf{w}}^T\tilde{\mathbf{w}} \; s.t. \sum_{i=1}^{2n}\tilde{w}_i = 1 \tag{10}$$

where, like the classical ridge regression [9], we add a small amount of $L2$-norm regularization in order to improve the stability of solutions when $n \gg m$.

IV. Optimization

In solving the objective function in Eq.(10), note that it is closely related to the dual form of the SVM with the squared loss function [11,18]:

$$\arg\min_{\tilde{w}_i\geq 0} f(\tilde{\mathbf{w}}) = \tilde{\mathbf{w}}^T\tilde{X}^T\tilde{X}\tilde{\mathbf{w}} + \frac{1}{2C}\sum_{i=1}\tilde{w}_i^2 - \mathbf{1}^T\tilde{\mathbf{w}} \tag{11}$$

for any general dataset $\{\tilde{\mathbf{x}}_i\}_{i=1}^{|\mathcal{DS}|}$ of $|\mathcal{DS}|$ samples, where $\tilde{X} = [\tilde{\mathbf{x}}_1,\ldots,\tilde{\mathbf{x}}_{|\mathcal{DS}|}]\times\text{diag}(\mathbf{y})$, in which $\text{diag}(\mathbf{y})$ is the diagonal matrix whose entries are class labels (i.e., $y_i \in \{-1,1\}$), and C is the margin parameter.

It is easy to see that our $\tilde{X}$ in Lemma 2 can also be represented in this format. Specifically, $\tilde{X} = \left(\begin{bmatrix} X_1 \\ \sqrt{\lambda_1}S\end{bmatrix}\begin{bmatrix} X_2 \\ \sqrt{\lambda_1}S\end{bmatrix}\right)\times\text{diag}(\mathbf{y})$, where $\mathbf{y}$ is the vector having first n entries as 1's and last n entries as -1's. The only difference between Eq.(11) and Eq.(10) is that the latter further requires the constraint $\mathbf{1}^T\tilde{\mathbf{w}} = 1$. However, if such a constraint is also applied to Eq.(11), then its last term becomes a constant. Indeed, this constraint simply rescales our optimal solution for $\tilde{\mathbf{w}}$ to be of unit L1-length. The sparseness property of $\tilde{\mathbf{w}}$ is obviously unchanged by such a normalization step. Similar to the dual-form SVM, we can solve Eq.(10) using techniques like coordinate descent, internal point or active set method. However, the computation often involves dealing with $2n$ inequality constraints directly. Therefore, a more practical approach is to consider such a quadratic programming problem in the primal form of an unconstrained problem [12] as follows:

$$\tilde{\mathcal{L}}(\tilde{\mathbf{w}}) = \sum_{i=1}^{2n}\sum_{j=1}^{2n}\tilde{w}_i\tilde{w}_j\tilde{\mathbf{x}}_i^T\tilde{\mathbf{x}}_j + \lambda_2\sum_{i=1}^{2n}\max(0,1-y_i\sum_j\tilde{w}_j\tilde{\mathbf{x}}_i^T\tilde{\mathbf{x}}_j)^2 \tag{12}$$

where, with the introduction of vector $\mathbf{y}$ above, we have redefined $\tilde{X} \longleftarrow \left(\begin{bmatrix} X_1 \\ \sqrt{\lambda_1}S\end{bmatrix}\begin{bmatrix} X_2 \\ \sqrt{\lambda_1}S\end{bmatrix}\right)$ with $\tilde{\mathbf{x}}_i$'s as its column vectors, and $\tilde{\mathbf{w}} \longleftarrow \text{diag}(\mathbf{y})\times\tilde{\mathbf{w}}$.

There is a flat part in this loss function (i.e., the 2nd term is 0 if $1 < y_i\sum_j\tilde{w}_j\tilde{\mathbf{x}}_i^T\tilde{\mathbf{x}}_j$), $\tilde{\mathbf{w}}$ is usually sparse. Moreover, the function is continuously differentiable, which is a great advantage. Hence, in optimizing Eq.(12), we resort to the Newton's method. In particular, let us denote $Q = \tilde{X}^T\tilde{X}$ and Q_i as the i-th column of matrix Q. The gradient of $\tilde{L}$ can be written as follows:

$$g = \frac{\partial\tilde{L}}{\partial\tilde{\mathbf{w}}} = 2Q\tilde{\mathbf{w}} - 2\lambda_2\sum_i Q_iy_i(1-y_iQ_i^T\tilde{\mathbf{w}}) \tag{13}$$

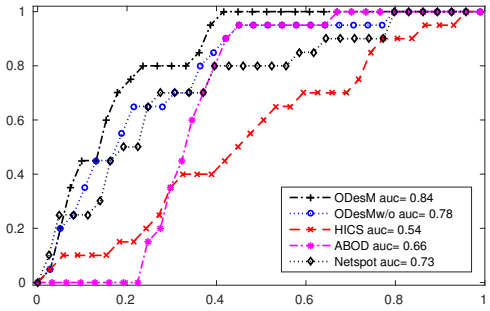

Figure 1: ROC curve performance of all algorithms on identifying outlier networks from the CMUFace graph dataset.

in which the summation in the second term is applied to $\tilde{\mathbf{x}}_i$'s for which $y_i \sum_j \tilde{w}_j \tilde{\mathbf{x}}_i^T \tilde{\mathbf{x}}_j < 1$. The Hessian is therefore:

$$H = \frac{\partial \tilde{L}}{\partial \tilde{\mathbf{w}} \partial \tilde{\mathbf{w}}^T} = 2Q + 2\lambda_2 \sum_i y_i^2 Q_i Q_i^T \tag{14}$$

At each iteration of Newton's method, we update $\tilde{\mathbf{w}}$ to $\tilde{\mathbf{w}} - \eta H^{-1} g$ where η is the learning rate found through the line search technique [3]. Given the convergence of $\tilde{\mathbf{w}}$ (thus also $\mathbf{w}$), the final subnetworks that are used as the explanations for the exceptionality of $\mathcal{N}_o$ can be identified via the non-zero entries of $\mathbf{w}$. For the outlier score of $\mathcal{N}_o$, denoted by $OS(\mathcal{N}_o)$, we follow an approach similar to [4] but compute it only in the subspace spanned by the explanatory subnetworks. The higher the value of $OS(\mathcal{N}_o)$, the more $\mathcal{N}_o$ deviates from its neighboring networks.

V. Experiments

A. Methodology

We name our algorithm ODeSM (Outlier Detection with Subgraph Mining), and compare its performance against techniques in both network studies and high dimensional studies: (1) Netspot [2] without temporal constraint so allowing it to uncover network regions from each individual network; (2) HiCS [13] that seeks outliers through contrast subspaces for high dimensional data; (3) ABOD [15] which discovers outliers via variance of angles between vector triples; (4) ODesMw/o, a variant of our method without exploiting network regularization. The parameter setting for ODesM and ODesMw/o follows the best-effort-approach [20].

B. CMUFace graph data

Since most network datasets (presented next) lack ground-truth subnetworks, we conduct an experiment on the CMUFace image data (http://archive.ics.uci.edu) as it allows us to evaluate the relevance of uncovered subnetworks *via visualization*. The network dataset is generated with the procedure described in [6], with the following modification: we select all networks with open-eye images as inliers, and randomly select one with sunglasses from any of 4 poses

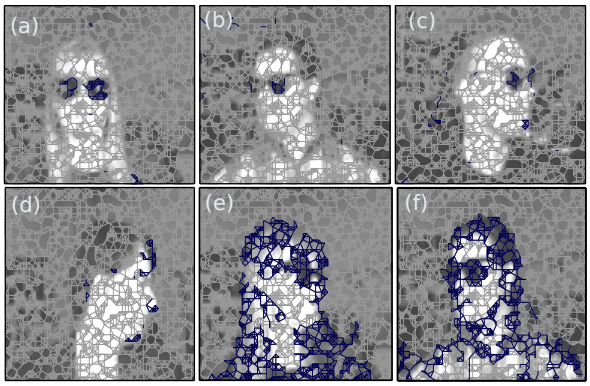

Figure 2: Subnetworks selected by ODesM ((a)-(d)) and Netspot ((e)-(f)) in CMUFace. In each figure, the network topology is shown in grey and the selected subnetworks are shown in blue, while the corresponding full image is shown in background with dimmed colors to improve the visualization.

(straight/up/left/right) as an outlier. This allows us to evaluate whether an algorithm can deal with the heterogeneity in the data. The dataset consists of 303 regular network samples and 20 anomalous ones.

Outlier identification: In Fig.1, we plot the ROC curve performance of all five algorithms. As seen from this figure, both techniques HiCS and ABOD designed for high dimensional data perform moderately well on this dataset. ABOD explores the variance over angles between an outlier candidate and every pair of other two samples, so its approach explores global outliers deviating from a single distribution of inliers. For this dataset, however, we have multiple distributions, and thus true outliers are harder determined by solely relying on the angles. HiCS, on the other hand, attempts to find most contrasting subspaces using a bottom-up approach and it starts with those of 2-dimension (from a pool of $\binom{1920}{2} = 1844160$ possible subspaces). If such low dimensional subspaces are not well sampled, the quality of contrasting subspaces found in higher dimensional subspaces can be compromised. Netspot performs better than these two techniques by relying on the p-value defined at each node. However, by converting to a p-value, Netspot also removes the contrast among node values and thus is less successful in seeking the most potential seed-nodes. ODesM performance is the best with its AUC at 0.84, as compared to 0.78 obtained by the second best ODesMw/o. This large gap in AUC also confirms the key role of network topology exploited by ODesM.

Explanatory subnetworks: We further explore the set of subnetworks discovered by ODesM as the explanation for top ranking outliers. Out of top 20 anomalous networks, 8 are true outliers. We plot in Fig.2(a-c) the three top ranked networks that are also truly labeled as outliers, and their corresponding images. As observed, despite coming from different poses, the outlier networks are still well-identified and the subnetworks located around the sunglasses are appropriately selected by ODesM, though they can vary across different outliers. Fig.2(d) shows a network sample

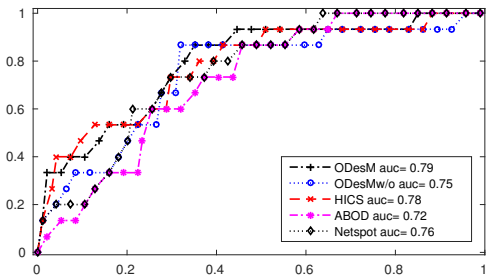

Figure 3: ROC of tested algorithms in identifying outlier networks in the Liver gene network dataset.

ranked high by ODesM but not a true outlier according to the labeling based on sunglasses.

ABOD, ODesMw/o and HiCS are not network-based techniques. Hence, we select Netspot for comparison based on its discovered anomalous subnetwork regions. In Fig.2(e-f), we plot two typical true outliers found from 20 top networks ranked by Netspot. Unlike the subnetworks discovered by our method, it is harder to justify why the corresponding images are exceptional though the uncovered subnetworks are strongly connected. In both figures, the substructures from entire faces have been selected. This performance probably comes from the fact that, other than p-value, Netspot also relies on the adjacency of network samples to derive the time interval at which significant anomalous regions can appear. However, once the interval is set to 1 (i.e. for each individual network and no temporal development), it has limited information to justify the relevance of a network region. Thus, the p-value computed at each node is likely playing the key role. And as long as its values do not change abruptly, Netspot tends to select all of them, forming a large subnetwork structure. The patterns discovered by Netspot and our ODesM are thus fundamentally different. For this reason, we do not attempt to compare their uncovered subnetworks in subsequent experiments.

C. Biological PPI network

The second dataset we use for evaluation is the Liver metastasis in human [14] with the gene network derived from protein-protein interactions. Values associated with nodes are the gene expression values. The dataset contains $7,383$ genes and $251,916$ edges collected from 101 healthy subjects viewed as inlying network samples, and 15 diseased subjects labeled as outliers.

Outlier identification: We show in Fig.3 the ROC curve of all algorithms on the Liver dataset. The performance of our ODesM method is comparable to that of HiCS and both are better than the remaining techniques. Netspot also performs well on this dataset as indicated by its 0.76 AUC and is slightly better than ODesMw/o. Recall that each network sample of this dataset contains a large number of nodes. However, unlike the CMUFace graph data where we have multiple data distributions (each representing images from a person), here we have only a single network distribution

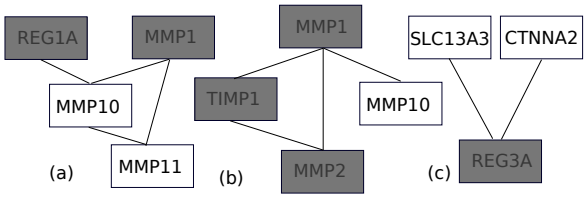

Figure 4: Subnetworks frequently discovered by ODesM in its top 15 network samples with the highest outlier scores. Shaded genes are related to liver metastasis cancer.

of healthy subjects. The outlier prediction rates of all techniques are thus not as diverse as those we have seen in the CMUFace networks. However, the results still indicate that ODesM yields the highest outlier prediction rate.

Explanatory subnetworks: In order to investigate how relevant and explanatory are the subnetworks discovered by ODesM, we compute the most frequent subnetworks found in the top 15 ranked outlying networks. In Fig.4, we plot 3 subgraphs that have the highest frequency. The first subnetwork is found in 6 networks and out of these, 4 are anomalous networks. The second subnetwork is found in 4 networks with 3 as true outliers. The last subnetwork is found in 5 network samples of which one is a true outlier. The second subnetwork in Fig.4(b) is especially interesting since 3 of the four genes are implicated in liver cancer [14].

Though the genes forming the above subnetworks are not all related to liver cancer and not all diseased network samples are ranked at the top (7 true outliers are found out of top 15), an important observation from these results is that, the frequent involvement of diseased genes in the discovered subnetworks can signal the appearance of the disease. Moreover, since diseased subjects can suffer from different stages or subtypes of the cancer, the disease-related gene pathways can possibly vary from one subject to another.

D. Road traffic networks

The last dataset we use for evaluation is LATraffic—the highway traffic network data of Los Angeles, California (http://pems.dot.ca.gov) during April 2011. Based on the distribution of the average speed computed for each network snapshot, we randomly select 300 snapshots around the mean of this distribution to label as regular networks, and other 30 snapshots from two extreme tails (15 each) to label as anomalous networks.

Outlier identification: The ROC curve performance of all algorithms on the LATraffic is shown in Fig.5. HiCS handles the subspace candidates well and its Monte-Carlo sampling based approach tends to select highly contrasting subspaces. Netspot is less successful in uncovering both types of low and high speed outliers. Among all examined techniques, ODesM is still the best performer with an AUC of 0.9.

Explanatory subnetworks: We further explore the set of subnetworks ranked top by ODesM. Fig.6 depicts the uncovered subnetworks for the top four outlier networks.

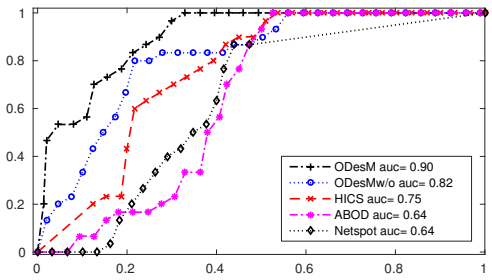

Figure 5: ROC curve performance of all algorithms on identifying outlier networks from the LATraffic network dataset.

The networks in (a) and (d) are the true outliers with low speed while the ones in (b) and (c) are the true outliers with high speed. The sets of discovered subnetworks in both cases are quite consistent. An interesting point emerges upon closer inspection of these explanatory substructures. We would expect the explanatory subnetworks for two types of outliers to be different since one considers low speeds while the other one considers high speeds. However, it turns out that they share one large subnetwork spanned by nodes 11, 6, 9, 12 and 25. The common selection of this subnetwork may suggest that such a set of adjacent road segments is highly sensitive to traffic congestion. For monitoring purposes, these road segments should be the top candidate since they are likely to reflect the overall condition of the entire traffic network.

VI. Conclusions

In this paper, we addressed an important problem of identifying and explaining outlier network samples. A novel algorithm was developed to identify subnetworks that discriminate outlier networks from their neighboring regular network samples. The algorithm was designed in the framework of network regression combined with the constraint on the network topology and L1-norm shrinkage to perform subnetwork discovery. Our algorithm thus goes beyond both subspace learning and subgraph discovery methods by directly learning the most discriminative subnetworks to justify the exceptional properties of an anomalous network. Evaluation on various real-world network datasets demonstrated that our novel algorithm not only outperformed existing techniques, but also uncovered highly relevant and interpretable local subnetworks.[1]

Acknowledgements: Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053 (the ARL Network Science CTA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

[1]The full version of this paper is publicly available in ArXiv

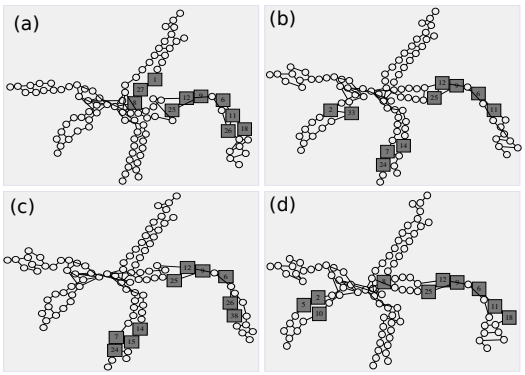

Figure 6: Top four outlier networks discovered by ODesM from the LATraffic dataset. Road segments involved in the explanatory subnetworks are shaded.

References

[1] L. Akoglu, H. Tong, and D. Koutra. Graph based anomaly detection and description: a survey. *DMKD*, 2015.

[2] P. Bogdanov et al. Netspot: Spotting significant anomalous regions on dynamic networks. In *SDM*, 2013.

[3] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[4] M. M. Breunig et al. LOF: identifying density-based local outliers. In *SIGMOD*, 2000.

[5] X. H. Dang et al. Discriminative features for identifying and interpreting outliers. In *ICDE*, 2014.

[6] X. H. Dang et al. Learning predictive substructures with regularization for network data. In *ICDM*, 2015.

[7] X. H. Dang, A. K. Singh, P. Bogdanov, H. You, and B. Hsu. Discriminative subnetworks with regularized spectral learning for global-state network data. In *ECML*, 2014.

[8] W. Eberle and L. B. Holder. Discovering structural anomalies in graph-based data. In *ICDM workshop*, 2007.

[9] T. Hastie et al. *The Elements of Statistical Learning. Data Mining, Inference, and Prediction*. Springer, 2009.

[10] K. Henderson et al. It's who you know: graph mining using recursive structural features. In *KDD*, 2011.

[11] C. Hsieh et al. A dual coordinate descent method for large-scale linear SVM. In *ICML*, 2008.

[12] S. Keerthi, O. Chapelle, and D. DeCoste. Building support vector machines with reduced classifier complexity. *JMLR*, 2006.

[13] F. Keller, E. Müller, and K. Böhm. Hics: High contrast subspaces for density-based outlier ranking. In *ICDE*, 2012.

[14] D. H. Ki et al. Whole genome analysis for liver metastasis gene signatures in colorectal cancer. *Int J Cancer*, 2007.

[15] H. Kriegel, M. Schubert, and A. Zimek. Angle-based outlier detection in high-dimensional data. In *SIGKDD*, 2008.

[16] B. Micenková, R. T. Ng, X. H. Dang, and I. Assent. Explaining outliers by subspace separability. In *ICDM*, 2013.

[17] M. W. Schmidt, G. Fung, and R. Rosales. Fast optimization methods for L1 regularization: A comparative study and two new approaches. In *ECML*, 2007.

[18] J. Taylor and S. Sun. A review of optimization methodologies in support vector machines. *Neurocomput.*, 2011.

[19] Q. Zhou et al. A reduction of the elastic net to SVM with an application to GPU computing. In *AAAI*, 2015.

[20] A. Zimek et al. A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining*, 2012.