A Consistent Method for Graph Based Anomaly Localization

Satoshi Hara **Tetsuro Morimura** **Toshihiro Takahashi** **Hiroki Yanagisawa**
IBM Research – Tokyo, Japan, {satohara, tetsuro, e30137, yanagis}@jp.ibm.com

Taiji Suzuki
Tokyo Institute of Technology & PRESTO, JST, Japan, s-taiji@is.titech.ac.jp

Abstract

The anomaly localization task aims at detecting faulty sensors automatically by monitoring the sensor values. In this paper, we propose an anomaly localization algorithm with a consistency guarantee on its results. Although several algorithms were proposed in the last decade, the consistency of the localization results was not discussed in the literature. To the best of our knowledge, this is the first study that provides theoretical guarantees for the localization results. Our new approach is to formulate the task as solving the sparsest subgraph problem on a difference graph. Since this problem is NP-hard, we then use a convex quadratic programming approximation algorithm, which is guaranteed to be consistent under suitable conditions. Across the simulations on both synthetic and real world datasets, we verify that the proposed method achieves higher anomaly localization performance compared to existing methods.

1 Introduction

Anomaly localization is the task of detecting faulty sensors by monitoring sensor values. This problem differs from the outlier detection [1, 2] and change point detection [3, 4] problems in that it requires specifying the error causes rather than evaluating whether each data point is healthy or not. The development of the anomaly localization algorithms seeks to automate the error-cause-detection procedure, which is conducted by skilled engineers in many cases.

Several anomaly localization algorithms were proposed in the last decade. Idé et al. [5, 6] proposed using the changes of the statistical dependencies between the sensor values. They used a graph to represent the inter-sensor dependency structure, and constructed anomaly localization algorithms for the graphs. Hirose et al. [7] proposed a heuristic algorithm based on the inter-sensor correlations, while Jiang et al. [8] used a PCA-based method to identify the erroneous subspace.

Despite the practical usefulness of these algorithms, the consistency of the localization results remains an open question: Can we localize faulty sensors properly when there is a sufficiently large number of samples? To the best of our knowledge, this consistency issue has not been discussed in the literature.

The contributions of this paper are twofold. First, we formulate the anomaly localization problem so that its solution is consistent. That is, the solution to the proposed problem coincides with the faulty sensors when there is a sufficiently large number of samples. The proposed formulation is based on the ideas of Idé et al. [5, 6]. We use the dependency graph among the sensors, and formulate the task as finding the sparsest subgraph on a difference graph. The resulting problem is a *sparsest subgraph problem* [9].

The second contribution is the development of a consistent polynomial-time approximation method. Since the problem is NP-hard in general [9], we use convex quadratic programming (QP) to derive an approximate solution. There are three advantages on the proposed QP approximation. First, the convex QP is polynomial-time solvable, and thus it is applicable to high dimensional problems. Second, the solution of the QP approximation is guaranteed to be consistent under suitable conditions. Third, we do not need to specify a volatile hyper-parameter.

Appearing in Proceedings of the 18^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2015, San Diego, CA, USA. JMLR: W&CP volume 38.

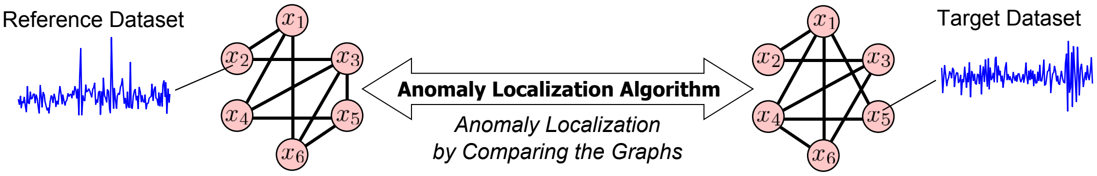


Figure 1: An overview of the graph-based anomaly localization: We transform datasets into graphs and then localize anomalies by comparing the graphs using an *anomaly localization algorithm*.

We also verified the consistency of the proposed method numerically by using synthetic datasets. Across the experiments, we found that the proposed method shows better anomaly localization performance compared to existing algorithms owing to its consistency guarantee.

2 Graph-based Anomaly Localization

The objective of the anomaly localization problem is to identify the contributions of each of d random variables (or sensor values), $\boldsymbol{x} = (x_1, x_2, \ldots, x_d)^\top \in \mathbb{R}^d$, to the differences between a reference dataset and a target dataset. In most practical cases, the reference dataset is past sensor values under no anomalies while the target dataset consists of the current observations. In the anomaly localization problem, we assume some sensor values in the target dataset are behaving differently from the ones in the reference dataset, and these values cause anomalies resulting in a distributional change. The task is to determine which sensors are and are not contributing to the changes.

We assume two conditions: (i) the number of variables in each dataset is the same, so they are both d dimensions, and (ii) the identity of each variable is the same, so the value of each corresponding element always comes from the same sensor.

2.1 Problem Definition

Across the paper, we follow the graph-based anomaly localization approach proposed by Idé et al. [5, 6]. In their approach, we use a dependency graph as the data model. Each node of the dependency graph corresponds to each of the random variables, and each edge represents a certain kind of dependency between the two variables. There are several dependency graphs we can use depending on the type of anomalies we want to detect. We discuss some representative graphs in Section 2.2. The anomaly is then defined as a structural change of this dependency graph, and the anomaly localization problem is reduced to finding a set of nodes causing the change. Figure 1 shows the overview of the graph-based anomaly localization problem.

Here, we give a formal definition of the graph-based anomaly localization problem. Let p_A be the healthy data distribution of the reference dataset, and p_B be the faulty data distribution of the target dataset. We also let $\Lambda_\mathrm{A}, \Lambda_\mathrm{B} \in \mathbb{R}^{d\times d}$ be weighted adjacency matrices of the inter-sensor dependency graphs representing the distributions p_A and p_B, respectively (see Section 2.2 for the detail of Λ_A and Λ_B). We then model the anomaly as follows, which is a modification of the assumption discussed by Idé et al. [6, Assumption 1]:

Assumption 1 (Neighborhood Preservation) *Let $\mathcal{I}$ and $\mathcal{J}$ be the partition of $\{1, 2, \ldots, d\}$, where $\mathcal{I}$ is a set of healthy sensor indices and $\mathcal{J}$ is a set of faulty sensor indices. We assume the interactions among the healthy sensors are unchanged, which means $\Lambda_{\mathrm{A},ij} = \Lambda_{\mathrm{B},ij}$ for all $i, j \in \mathcal{I}$. In contrast, the anomalous sensors have at least one neighbor whose interaction is changing, that is, for any $i' \in \mathcal{J}$, there exists an index $j' \neq i'$ such that $\Lambda_{\mathrm{A},i'j'} \neq \Lambda_{\mathrm{B},i'j'}$.*

Intuitively, we assume the dependencies between the healthy sensors are kept constant, but this is not the case for the faulty sensors. There are some changes in the dependency structure caused by the faulty sensors. The detailed background and motivating examples of this assumption can be found in Idé et al. [5, 6]. We now define the anomaly localization problem using Assumption 1:

Problem 1 (Anomaly Localization) *Let the reference dataset $\mathcal{D}_\mathrm{A} = \{\boldsymbol{x}_\mathrm{A}^{(n)}\}_{n=1}^{N_\mathrm{A}}$ be i.i.d. draw from the healthy data distribution p_A parameterized by Λ_A, and the reference dataset $\mathcal{D}_\mathrm{B} = \{\boldsymbol{x}_\mathrm{B}^{(n')}\}_{n'=1}^{N_\mathrm{B}}$ be i.i.d. draw from the faulty data distribution p_B parameterized by Λ_B. The anomaly localization task is to identify the sets $\mathcal{I}$ and $\mathcal{J}$ from the datasets $\mathcal{D}_\mathrm{A}$ and $\mathcal{D}_\mathrm{B}$.*

Here, we import one technical assumption that the set $\mathcal{I}$ is uniquely identifiable (or otherwise the problem is ill-posed).

2.2 Graph Representation of Data

We introduce two representative graphs used by Idé et al. [5, 6]. The first one is a *covariance graph* (or *correlation graph*, defined in a similar manner), which is

Table 1: The convex QP approximation is advantageous in that it is both polynomial-time solvable and the solution is guaranteed to be consistent. Also, we do not need to specify the number of healthy sensors k.

Solution Algorithms	Time Complexity	Consistency	Need to specify k
Exact Method	Exponential	**Consistent**	Yes
Greedy Method	**Polynomial**	Not Consistent	Yes
Convex QP Approximation	**Polynomial**	**Consistent**	**No**

suitable for detecting anomalies in the sensors themselves. The adjacency matrix of the covariance graph $\Lambda \in \mathbb{R}^{d \times d}$ is defined by $\Lambda_{ii} = V_{ii}$, and $\Lambda_{ij} = |V_{ij}|$ for $i, j \in \{1, 2, \ldots, d\}$, where V_{ij} is a covariance between x_i and x_j. Suppose the error occurs on the variable x_j, that is, $\mathcal{J} = \{j\}$. The error on x_j only changes the covariance V_{ij} for all i, and thus the graph structure change occurs only in the negihbor of x_j. This is exactly what Assumption 1 expected for the set $\mathcal{J}$.

The second representation [6] is a Gaussian Graphical Model (GGM). In a GGM, a variable $\boldsymbol{x}$ is assumed to follow a Gaussian distribution $\mathcal{N}(\mathbf{0}_d, \Lambda^{-1})$ where $\Lambda \in \mathbb{R}^{d \times d}$ is a precision matrix. The weight of the node x_i is given by Λ_{ii}, and the edge weight between nodes x_i and x_j is given by Λ_{ij}. GGM is capable of capturing conditional dependency structures among variables. This property is useful when we aim to detect the changes in an underlying physical system, which may affect the observations of several sensors. This is because, even when such changes occur, the functional relationships among the healthy parts tend to remain constant and GGM is capable of capturing these consistent relationships. There are several methods available to estimate the matrix Λ from the data such as the ℓ_1-regularized maximum likelihood [10]. We can use the existing methods to derive a GGM-based graph representation (see Olsen et al. [11, Section 1]).

There are also some other representations available. In particular, the use of a GGM is instructive in that the idea can be naturally extended to some other graphical models. For instance, the nonparanormal model [12, 13], which is a generalization of a GGM, can also be used for the graph representation. Another example is the Ising model [14], which is a popular model to express the dependencies among binary random variables. More generally, we can represent data with any type of graphical model. Our proposed method accepts any type of graph representation as long as the graph is undirected, which is the case when the adjacency matrix is a symmetric matrix.

3 A Consistent Anomaly Localization Method

We now describe an anomaly localization method for solving Problem 1 with a consistency guarantee. We cast the problem as finding the most similar subgraph between the two graphs, which is formulated as the *sparsest k-subgraph problem* [9]. We show that the solution to the problem is a consistent estimator of the faulty sensor set $\mathcal{J}$. We then describe a convex Quadratic Programming (QP) approximation method to tackle the NP-hardness of the problem. In Table 1, we see the advantage of the proposed approximation. The approximate problem is polynomial-time solvable and its approximate solution is guaranteed to be consistent under suitable conditions.

3.1 The Sparsest k-Subgraph Problem Formulation

Our approach is to formulate the task as finding the set of healthy variables $\mathcal{I}$, which also exposes the set of anomalous variables $\mathcal{J}$ as its complement. We start the discussion by assuming the number of healthy variables $|\mathcal{I}|$ is known to be k, which we relax to the general unknown case in Section 3.3. The problem can then be transformed to the well-known *independent set problem* [15]. Figure 2 shows an example. The central idea is the use of a graph whose adjacency matrix $\Gamma \in \mathbb{R}^{d \times d}$ is given by $\Gamma_{ij} := |\Lambda_{\mathrm{A},ij} - \Lambda_{\mathrm{B},ij}|$. From Assumption 1, for all $i, j \in \mathcal{I}$, the corresponding edge weights $\Lambda_{\mathrm{A},ij}$ and $\Lambda_{\mathrm{B},ij}$ coincide and $\Gamma_{ij} = 0$ holds. This implies that any pairs (i, j) with $i, j \in \mathcal{I}$ are disconnected in the graph specified by Γ. Such a set $\mathcal{I}$ is known as an *independent set* [15] of the graph. An independent set of size k can be found by solving the independent set problem. Hence, since the set $\mathcal{I}$ is uniquely determined, the independent set coincides with $\mathcal{I}$. This property contrasts with other anomaly localization methods where no consistency guarantees are given for the result.

We now develop this idea into a practical formulation. In practice, the problem cannot be handled as an independent set problem. This is because we only have access to the estimated adjacency matrices from $\mathcal{D}_\mathrm{A}$ and $\mathcal{D}_\mathrm{B}$, which are $\hat{\Lambda}_\mathrm{A}$ and $\hat{\Lambda}_\mathrm{B}$. For these estimated matrices, $\hat{\Gamma}_{ij} := |\hat{\Lambda}_{\mathrm{A},ij} - \hat{\Lambda}_{\mathrm{B},ij}|$ will not be zero even for the unchanged edges, but will still have some small positive value. We tackle the problem by generalizing the independent set problem into the *sparsest k-subgraph problem* [9]. The sparsest k-subgraph problem asks us to find a set of k nodes that minimizes the sum of the

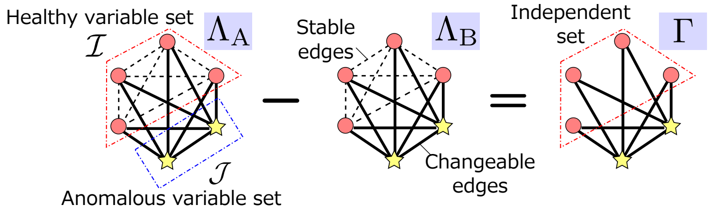


Figure 2: The healthy variable set $\mathcal{I}$ becomes an independent set in the graph specified by Γ.

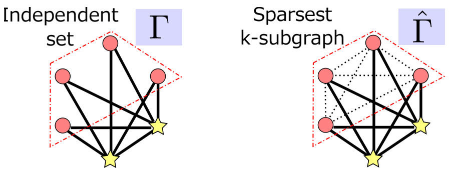


Figure 3: The sparsest k-subgraph problem allows edges to have small weights.

edge weights in the induced subgraph by the set of nodes, while the sum is constrained to be zero in the independent set problem. See Figure 3 for an example. The sparsest k-subgraph problem is formulated as

$$\hat{\mathcal{I}} = \mathrm{argmin}_{\mathcal{K}\subseteq\{1,2,\ldots,d\}} \sum\nolimits_{i,j\in\mathcal{K}} \hat{\Gamma}_{ij}, \text{ s.t. } |\mathcal{K}| = k. \quad (1)$$

The problem (1) coincides with the independent set problem when $\hat{\Gamma}_{ij} = 0$ for $i, j \in \mathcal{I}$. The next theorem guarantees that the solution $\hat{\mathcal{I}}$ coincides with $\mathcal{I}$ under suitable conditions, and the consistency guarantee follows as its corollary [1].

Theorem 1 *Let $\hat{\mathcal{I}}$ be the solution to the problem (1), and $h := \min_{\mathcal{K}\neq\mathcal{I}} \sum_{i,j\in\mathcal{K}} \Gamma_{ij} - \sum_{i',j'\in\mathcal{I}} \Gamma_{i'j'}$. If $|||\hat{\Gamma} - \Gamma|||_\infty < h/(k^2 + d^2)$, then we have $\hat{\mathcal{I}} = \mathcal{I}$ where $|||\cdot|||_\infty$ denotes an element-wise infinity norm $|||M|||_\infty := \max_{i,j} |M_{ij}|$.*

Corollary 1 *Let the dimensionality d be a fixed value. If $\hat{\Lambda}_A$ and $\hat{\Lambda}_B$ are consistent estimators of Λ_A and Λ_B, respectively, then the estimated set $\hat{\mathcal{I}}$ is a consistent estimator of $\mathcal{I}$, which means $\lim_{n_A,n_B\to\infty} P(\hat{\mathcal{I}} \neq \mathcal{I}) = 0$, where n_A and n_B are the number of samples used to estimate $\hat{\Lambda}_A$ and $\hat{\Lambda}_B$, respectively.*

We note that there are several well-known consistent estimators such as the maximum likelihood estimator of the covariance matrix and the ℓ_1-penalized maximum likelihood estimator of the precision matrix for GGM [10].

3.2 Simple Approach: Exact and Greedy Methods

A naive way to solve the problem (1) is to use general combinatorial methods. For instance, we can use an exact method to solve the problem. The solution can be derived by solving the binary quadratic problem:

$$\hat{\boldsymbol{s}} = \mathrm{argmin}_{\boldsymbol{s}\in\{0,1\}^d} \boldsymbol{s}^\top \hat{\Gamma} \boldsymbol{s}, \text{ s.t. } \mathbf{1}_d^\top \boldsymbol{s} = k. \quad (2)$$

The set $\hat{\mathcal{I}}$ can be recovered from the solution by $\hat{\mathcal{I}} = \{i; \hat{s}_i > 0\}$. The problem can also be transformed into an integer linear programming using a well-known technique [16]. We note that integer programming is in general NP-hard. In Section 4, we find that the problem is tractable up to $d = 200$ in reasonable times by using a state-of-the-art method.

A greedy method is a popular alternative choice when the exact method is not tractable. For instance, we can use a variant of the method given by Asahiro et al. [17] for the densest k-subgraph problem. We note that the algorithm outputs only a local optima and the solution is no longer guaranteed to be consistent. The detail of the algorithm can be found in Algorithm 1. In the pseudo-code, we defined $f(\mathcal{K}) := \sum_{i,j\in\mathcal{K}} \hat{\Gamma}_{ij}$. The method terminates in $d - k$ steps and each step requires at most $\mathcal{O}(d^2)$ time complexity in a naive implementation. This time complexity could be reduced to $\mathcal{O}(d \log d)$ by using a binary heap [18], though we used a naive implementation for our simulation study in Section 4.1, since it was sufficiently fast. We can use other heuristics such as a local search method (Algorithm 2), to improve the quality of the local optima.

One shortcoming of the formulation (1) is that we need access to the number of healthy variables k. In many practical cases, we do not know the number k in advance, and we need some way to handle this problem. We present one such method in the next section.

3.3 Proposed Approximation with Convex Quadratic Programming

We now propose a convex quadratic programming (QP) approximation to solve the problem (1). There are three advantages of the proposed QP approximation (Table 1). First, convex QP is polynomial-time solvable [19], resembling the greedy method. In particular, Kozma et al. [20] recently reported that the state-of-the-art methods can solve convex QP with more than 10,000 variables in a few seconds. Second, the solution of the QP approximation is guaranteed to be consistent under suitable conditions, while this is not generally true for the greedy method. Third, we can avoid specifying the parameter k in the QP approximation, which is not true for the exact and the greedy methods.

[1] The proof of theorems in this manuscript can be found in the supplemental material.

Algorithm 1 Greedy Method

```
Input: $\hat{\Gamma} \in \mathbb{R}^{d \times d}$ and an integer parameter $k$
Output: $\hat{\mathcal{I}} \subseteq \{1, 2, \ldots, d\}$
  let $\mathcal{K} = \{1, 2, \ldots, d\}$
  repeat
    $i' = \mathrm{argmin}_{i \in \mathcal{K}} f(\mathcal{K} \setminus \{i\})$
    $\mathcal{K} \leftarrow \mathcal{K} \setminus \{i'\}$
  until $|\mathcal{K}| = k$
  return $\hat{\mathcal{I}} \leftarrow \mathcal{K}$
```

Algorithm 2 Local Search

```
Input: $\hat{\Gamma} \in \mathbb{R}^{d \times d}$, $\mathcal{K} \subseteq \{1, 2, \ldots, d\}$
Output: $\hat{\mathcal{I}} \subseteq \{1, 2, \ldots, d\}$
  repeat
    $i', j' = \mathrm{argmin}_{i \in \mathcal{K}, j \notin \mathcal{K}} f((\mathcal{K} \setminus \{i\}) \cup \{j\})$
    if $f((\mathcal{K} \setminus \{i'\}) \cup \{j'\}) < f(\mathcal{K})$ then
      $\mathcal{K} \leftarrow (\mathcal{K} \setminus \{i'\}) \cup \{j'\}$
    end if
  until $\mathcal{K}$ is not updated in this iteration
  return $\hat{\mathcal{I}} \leftarrow \mathcal{K}$
```

We first describe a convex QP approximation for the problem (2), and then present the conditions under which the approximate solution coincides with $\mathcal{I}$.

To derive the convex QP approximation, we apply three modifications to the problem (2). First, we replace the domain of $\boldsymbol{s}$ from the binary value $\{0, 1\}^d$ into the continuous interval $[0, 1]^d$. Second, we replace $\hat{\Gamma}$ with $\hat{A}_\mu := \hat{\Gamma} + \mu I_d$ where the parameter μ satisfies $\mu + \lambda_{\min}(\hat{\Gamma}) > 0$. Here, $\lambda_{\min}(\cdot)$ denotes the minimum eigenvalue of the matrix. With those two modifications, we arrive at a simple convex problem:

$$\boldsymbol{s} = \mathrm{argmin}_{\boldsymbol{s} \in [0,1]^d} \, \boldsymbol{s}^\top \hat{A}_\mu \boldsymbol{s}, \text{ s.t. } \mathbf{1}_d^\top \boldsymbol{s} = r,$$

where $r \in \mathbb{R}_+$. Similar modifications have also been considered for the densest k-subgraph problem [21]. Here, we replaced the integer k with r, since $\boldsymbol{s}$ is in the continuous domain. Since the solution $\tilde{\boldsymbol{s}}$ has at least $\lceil r \rceil$ non-zero elements, we can interpret r as the lower bound of the number of healthy variables. In the third modification, we set $r = 1$. This corresponds to using the most pessimistic assumption so that, in the worst case, there is only one healthy variable and the remaining variables may all be faulty. Although this modification may seem to be a heuristic, we provide a rigorous justification later. With $r = 1$, the constraint $\tilde{s}_i \leq 1$ automatically holds for all $i \in \{1, 2, \ldots, d\}$ and we can safely replace the constraint $\boldsymbol{s} \in [0, 1]^d$ with $\boldsymbol{s} \in \mathbb{R}_+^d$. We finally have the convex QP approximation as

$$\tilde{\boldsymbol{s}} = \mathrm{argmin}_{\boldsymbol{s} \in \mathbb{R}^d} \, \boldsymbol{s}^\top \hat{A}_\mu \boldsymbol{s}, \text{ s.t. } \mathbf{1}_d^\top \boldsymbol{s} = 1, \ \boldsymbol{s} \geq \mathbf{0}_d. \quad (3)$$

Analogous to the discrimination rule in the binary problem (2), we can interpret the value $\tilde{s}_i$ as the score of the healthiness of x_i, and we can discriminate healthy and faulty variables by introducing a threshold. Specifically, we let the estimated index sets of the healthy variables and the faulty variables be $\tilde{\mathcal{I}}_t := \{i; \tilde{s}_i > t\}$ and $\tilde{\mathcal{J}}_t := \{i; \tilde{s}_i \leq t\}$, respectively, for a threshold t.

We now provide the justification for our modifications. The next theorem and corollary guarantee that $\tilde{\mathcal{I}}_0$, the estimated set with $t = 0$, coincides with $\mathcal{I}$ under suitable conditions. In particular, $\tilde{\mathcal{I}}_0$ is a consistent estimator of $\mathcal{I}$.

Theorem 2 *Let $\tilde{\boldsymbol{s}}$ be the solution to (3) and $\tilde{\mathcal{I}}_0 := \{i; \tilde{s}_i > 0\}$. Suppose there exists $\delta > 0$ such that*

$$\min_{j \in \mathcal{J}} \textstyle\sum_{i \in \mathcal{I}} \Gamma_{ji} \geq (1 + \delta)\mu \quad (4)$$

holds where $\mu + \lambda_{\min}(\Gamma) > 0$. Then we have $\tilde{\mathcal{I}}_0 = \mathcal{I}$ if $|||\hat{A}_\mu - A_\mu|||_\infty < B_\delta \mu / d$ holds with $B_\delta = \left\{\sqrt{(2 + \delta)^2 + 8\delta(1 + \delta)} - (2 + \delta)\right\} / 4(1 + \delta)$.

Corollary 2 *Let the dimensionality d be a fixed value. Suppose $\hat{\Lambda}_{\mathrm{A}}$ and $\hat{\Lambda}_{\mathrm{B}}$ are consistent estimators of Λ_{A} and Λ_{B}, respectively, and there exists $\delta > 0$ such that the condition (4) holds. Then $\tilde{\mathcal{I}}_0$ is a consistent estimator of $\mathcal{I}$.*

The condition (4) implies that the parameter μ for relaxing the problem has to be smaller than the minimum significance of the anomaly (left hand side of (4)). The condition requires us not to modify the problem too much, or small anomalies will be overlooked.

We note that the results reveal one important fact in that, even if we do not know the correct number of healthy variables k, the solution to the problem (3) will recover the true set $\mathcal{I}$ if the specified conditions are satisfied. We note that Theorem 2 indicates that there is a trade-off on the choice of μ. The smaller μ is desirable to satisfy the condition (4), while the larger μ is desirable for the condition $|||\hat{A}_\mu - A_\mu|||_\infty < B_\delta \mu / d$. In our preliminary experiment, we found that μ does not have significant effects on the result as long as it is kept small. We therefore set $\mu = 10^{-6} - \lambda_{\min}(\hat{\Gamma})$ for all of the simulations in Section 4.

3.4 Finite Sample Consistency

Theorems 1 and 2 can be further elaborated into a finite sample consistency guarantee for each specific graph representation. Here, we introduce the GGM case as an example.

Theorem 3 *Let $\hat{\mathcal{I}}$ be the solution to the problem (1). Suppose $\hat{\Lambda}_{\mathrm{A}}$ and $\hat{\Lambda}_{\mathrm{B}}$ are graphical–Lasso type estimators [10, Eq.(11)]. Under the conditions of Theorem 1, and the conditions specified by Ravikumar et al. [10, Theorem 1], there exists a constant C_1 such that $\hat{\mathcal{I}} = \mathcal{I}$ holds with probability greater than $1 - \eta$ $\left(\eta \in \left(4d \exp\left\{-h^2 n / 2C_1^2 (k^2 + d^2)^2\right\}, 1\right)\right)$, where $n := \min\{n_{\mathrm{A}}, n_{\mathrm{B}}\}$.*

Theorem 4 *Let $\tilde{\boldsymbol{s}}$ be the solution to the problem (3) and $\tilde{\mathcal{I}}_0 := \{i; \tilde{s}_i > 0\}$. Suppose $\hat{\Lambda}_{\mathrm{A}}$ and $\hat{\Lambda}_{\mathrm{B}}$ are graphical–Lasso type estimators [10, Eq.(11)], and there exists $\delta > 0$ such that the condition (4) holds. Under the conditions of Theorem 2, and the conditions specified by Ravikumar et al. [10, Theorem 1], there exists a constant C_2 such that $\tilde{\mathcal{I}}_0 = \mathcal{I}$ holds with probability greater than $1 - \eta$ $\left(\eta \in \left(4d \exp\left\{-B_\delta^2 \mu^2 n / 2C_2^2 d^2\right\}, 1\right)\right)$.*

We can derive similar results for any other graph representation. For example, we can use the results from Ravikumar et al. [10, Lemma 1] for the covariance case and Liu et al. [13, Theorem 4.3] for the nonparanormal model case.

We note that these results may not be optimal. It will be possible to improve the convergence rate by using more sophisticated estimators for $\hat{\Lambda}_{\mathrm{A}}$ and $\hat{\Lambda}_{\mathrm{B}}$. For example, in the GGM case, the use of the node-based learning method proposed by Mohan et al. [22] would be beneficial, although the convergence rate is still unknown. The improvement of the convergence rate remains as future work.

4 Numerical Evaluation

We investigated the anomaly localization performances of the proposed QP approximation method both on synthetic and real world datasets. For all of the simulations, we compared the performance of the proposed method against the exact method, the greedy method, and existing anomaly localization methods. We used IBM ILOG CPLEX 12.5.1 for the exact method and for the QP approximation to solve the problems (2) and (3). The greedy method and the local search method were implemented in C.

4.1 Synthetic Experiments

In the experiments, we considered two graph representations, covariance graphs and the GGMs, since they were already studied by Idé et al. [5, 6].

We prepared two different data settings with dimension $d = 100$ and 200 with the number of healthy variables $k = 90$ and 180, respectively. For each setting, we first generated the synthetic matrices Λ_{A} and Λ_{B} [2]. We then generated two datasets by i.i.d. samplings from $\mathcal{N}(\mathbf{0}_d, \Lambda_{\mathrm{A}})$ and $\mathcal{N}(\mathbf{0}, \Lambda_{\mathrm{B}})$ for the covariance graph case, and from $\mathcal{N}(\mathbf{0}_d, \Lambda_{\mathrm{A}}^{-1})$ and $\mathcal{N}(\mathbf{0}, \Lambda_{\mathrm{B}}^{-1})$ for the GGM case. We set the number of data points in each dataset to be rd where we varied the ratio r from 1 to 10 on a logarithmic scale.

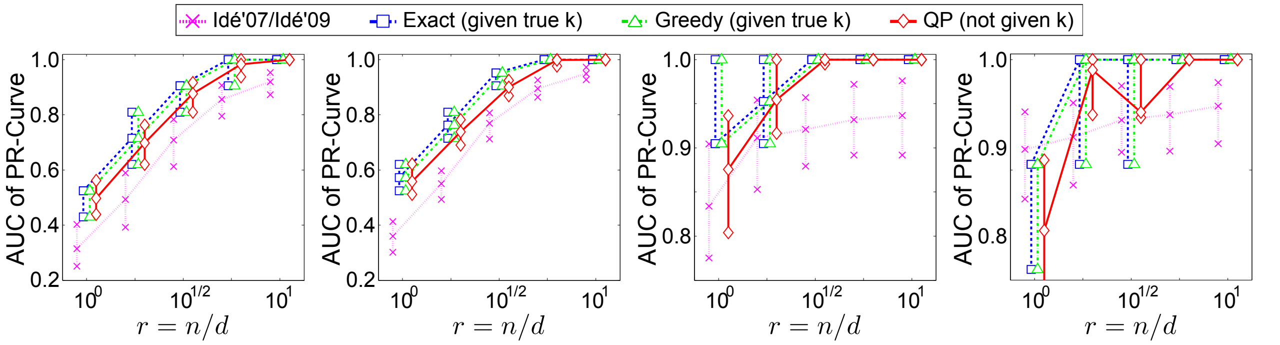
In the anomaly localization stage, we first estimated the matrices $\hat{\Lambda}_{\mathrm{A}}$ and $\hat{\Lambda}_{\mathrm{B}}$ from the datasets, and then used the QP approximation method. For $\hat{\Lambda}_{\mathrm{A}}$ and $\hat{\Lambda}_{\mathrm{B}}$, we used the empirical covariances for the covariance graph case, and the ℓ_1-regularized maximum likelihood estimators [10] for the GGM case. The regularization parameter was chosen by the 2-fold cross validation from seven values in $[10^{-4}, 10^0]$ on a logarithmic scale. We set the parameter μ in the approximate method to $\mu = 10^{-6} - \lambda_{\min}(\hat{\Gamma})$. We evaluated the anomaly localization result using the Area Under the Curve (AUC) of the precision-recall curve.

We used the exact and the greedy methods as the baseline methods. We assumed an ideal situation in which the number of healthy variables is known to be k for those methods. We also used Idé's method in ICDM'07 [5] (*Idé'07*) for the covariance graph case, and Idé's method in SDM'09 [6] (*Idé'09*) for the GGM case to contrast with. For Idé'07, we varied the degree of the graph from six values in $[d/10, d/2]$ and picked the value that maximized the AUC.

The simulation results over 100 random data realizations are shown in Figure 4. In these simulations, we found that the results of the greedy method were almost identical to those of the exact method, and therefore we did not use the local search. Figure 4(a)–(d) show the high effectiveness of the proposed QP approximation method. In particular, it showed perfect localization performance when there were a sufficiently large number of samples. This is the consistency guarantee that Corollary 2 implied. This contrasts with Idé'07 and Idé'09, which achieved the median AUC around 0.96 even at the maximum. We can also observe that, although the AUC is slightly lower, the QP approximation performs nearly as well as the exact method. We emphasize that this result was attained without specifying the number k, which shows the practical utility of the QP approximation.

Figure 5 compares the computation times of the three proposed methods. We ran all of the methods on 64-bit Windows 7 with an Intel Core i5-3320M and 8 GB of RAM. The figure shows the median runtimes for $d = 100$ and 200 with the number of samples $n = 10d$. The graph shows that the greedy method was the fastest and the QP approximation came next. The computa-

[2] The generation procedure of these matrices can be found in the supplemental material.



(a) Cov., $d = 100$ (b) Cov., $d = 200$ (c) GGM, $d = 100$ (d) GGM, $d = 200$

Figure 4: The median AUCs of the anomaly localization methods: Vertical bar extends from the 25% to the 75% quantile. We used Idé'07 for the covariance graph case ((a), (b)) and Idé'09 for the GGM case ((c), (d)).

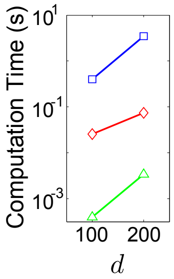
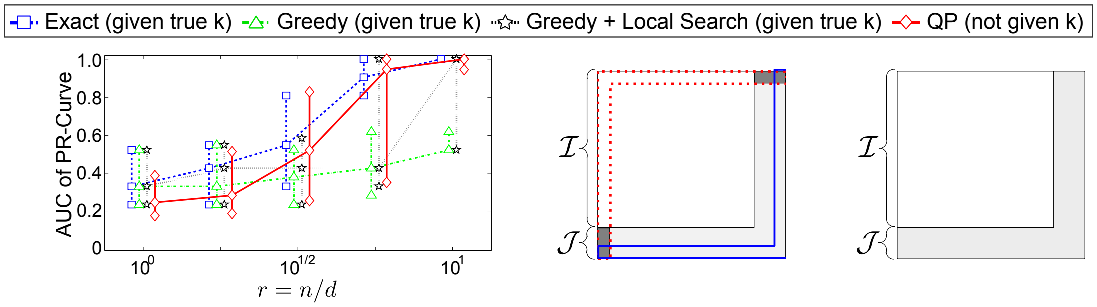


Figure 5: Median computation times (Cov.)



(a) Cov. $d = 100$ with concentrated patterns (b) Γ with a concentrated anomaly pattern (c) Γ with an uniform anomaly pattern

Figure 6: Comparison of the exact, the greedy, and the QP approximation methods (Cov., $d = 100$) with concentrated anomaly patterns: The colored parts in the figures (b) and (c) are the non-zero elements in Γ with a darker color indicating a larger value.

tion time for the exact method was around few seconds and the results show that the exact method is feasible up to $d = 200$.

We also conducted another experiment to compare two polynomial-time algorithms, the greedy method and the QP approximation. The result is shown in Figure 6(a) for the covariance graph case with $d = 100$. This time, we used the concentrated anomaly pattern as shown in Figure 6(b) instead of the uniform pattern (Figure 6(c)) used in the previous simulation. The performance of the greedy method is worse than the other methods because the greedy method tends to choose the healthy variables surrounded by the dotted line in Figure 6(b) as anomalies in the first few iterations instead of the true anomalies (solid line in Figure 6(b)). The use of the local search was helpful this time, in particular, for the case when the number of sample is large. In contrast, the QP approximation performs similarly to the exact method owing to its consistency guarantee, and the result indicates that the QP approximation would be a safer polynomial-time alternative for the exact method compared to the greedy method (and the local search).

4.2 The Sun Spot Sensor Data

Next we used the proposed methods for the real problem used by Jiang et al. [8]. The original datasets were provided by the authors. Although there are three datasets used in their paper, we used only two of them, since the true anomaly label is missing for one dataset.

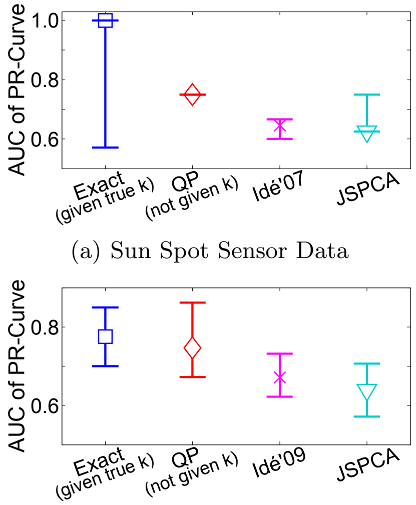
The first dataset, the Sun Spot Sensor Data, was collected in an automotive trial for transport chain security validation using seven wireless Sun Small Programmable Object Technologies (SPOTs). These SPOTs were fixed in separate boxes and loaded in the back seat of a car. The dataset was the magnitudes of the accelerations sampled by each SPOT $(a_x^2 + a_y^2 + a_z^2)^{1/2}$ where a_x, a_y, and a_z were the accelerations in x, y, and z directions, respectively. During the data collection, one of the seven sensors was removed and replaced six times. We used the sensor signals from one minute before the replacement event as the healthy data, and the signals from one minute after the event as the anomalous data, where only the removed sensor was causing an error. Since the sensor signals were sampled each 390 ms, there were 154 observations in each dataset.

We applied the approximate methods to these six events and evaluated the AUC. Since the task was to localize anomalies of the sensors, we used the covariance graph as the data representation (see Section 2.2 for the choice of the graph). We adopted the exact method, Idé'07 [5], and the PCA-based method named JSPCA [8] as the baseline methods. We did not use the greedy method since the problem size was small and the exact method was applicable. The parameter settings of the proposed methods and Idé'07 were the same as in Section 4.1. We used 2-fold cross validation for the parameter selection in JSPCA. Since JSPCA is a non-convex problem, we solved the problem 5 times with random initializations and chose the solution with the smallest objective function value.

The results are shown in Figure 7(a). The exact method achieved the best median AUC and the QP approximation came next. Although JSPCA had a comparable AUC to the QP approximation for the 75% quantile, we found that the JSPCA solution is sensitive to the choice of initial parameters because of its non-convexity. This difficulty is increased because of the need to search over several different hyper-parameters. The QP approximation does not have such an initialization sensitivity thanks to its convexity, and it has only one parameter μ to be tuned, with only minor effects on the solution. As a result, the QP approximation showed quite stable performance.

4.3 The Motor Current Data

The second dataset, the Motor Current Data, was 20-dimension current signals sampled from the state space simulations available at the UCR Time Series Archive [23]. In this dataset, several different types of machinery failures are simulated and we used them as the anomalies in this simulation. Specifically, the data consists of observations sampled from 21 different operating conditions; one is sampled under the completely healthy condition while the remaining 20 datasets are sampled under various specific failure modes. We generated two datasets to be contrasted. The first dataset was generated by randomly picking 200 consecutive observations from the healthy dataset. The second dataset was also 200 consecutive observations just after the first dataset, but where 10 sensor values were randomly replaced with one of the 20 faulty datasets so that they behave differently from the healthy state. The goal is to localize these 10 replaced sensors.



(a) Sun Spot Sensor Data

(b) Motor Current Data

Figure 7: The median AUCs of anomaly localization methods: Vertical bar extends from the 25% to the 75% quantile.

We used the nonparanormal model [12, 13], a generalization of the GGM, with 2-fold cross validation for the data representation to handle the non-linearity of the data. We adopted the exact method, Idé'09 [6], and JSPCA [8] with the same setting in Section 4.2 as the baseline. The results over 50 random data realizations are shown in Figure 7(b). This again shows the advantages of the proposed QP approximation method.

5 Conclusion

We proposed a consistent anomaly localization method based on the inter-sensor dependency structure. We formulated the anomaly localization problem as a sparsest k-subgraph problem, and proposed a convex QP approximation to solve the problem. We also provided consistency proofs for the proposed approximation. To the best of our knowledge, this is the first study on anomaly localization methods with theoretical justifications. Simulations on both synthetic and real world data verified that the proposed method outperforms existing methods. We also showed that the proposed approximation performs comparably with the exact method.

Acknowledgments

SH and TM were supported by CREST, JST. TS was partially supported by PRESTO and CREST, JST, and MEXT Kakenhi 25730013, 26280009 and 25120012.

References

[1] H. P. Kriegel, P. Kröger, and A. Zimak. Outlier detection techniques. *Tutorial at 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2010.

[2] C. C. Aggarwal. *Outlier analysis.* Springer, 2013.

[3] M. Basseville and I. V. Nikiforov. *Detection of Abrupt Changes: Theory and Application.* Prentice Hall, 1993.

[4] A. Lazarevic, L. Ertöz, V. Kumar, A. Ozgur, and J. Srivastava. A comparative study of anomaly detection schemes in network intrusion detection. *Proceedings of the 2003 SIAM International Conference on Data Mining*, 3:25–36, 2003.

[5] T. Idé, S. Papadimitriou, and M. Vlachos. Computing correlation anomaly scores using stochastic nearest neighbors. *Proceedings of the 7th IEEE International Conference on Data Mining*, pages 523–528, 2007.

[6] T. Idé, A. C. Lozano, N. Abe, and Y. Liu. Proximity-based anomaly detection using sparse structure learning. *Proceedings of the 2009 SIAM International Conference on Data Mining*, pages 97–108, 2009.

[7] S. Hirose, K. Yamanishi, T. Nakata, and R. Fujimaki. Network anomaly detection based on eigen equation compression. *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1185–1194, 2009.

[8] R. Jiang, H. Fei, and J. Huan. Anomaly localization for network data streams with graph joint sparse PCA. *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 886–894, 2011.

[9] R. Watrigant, M. Bougeret, and R. Giroudeau. Approximating the sparsest k-subgraph in chordal graphs. *Approximation and Online Algorithms, Lecture Notes in Computer Science*, 8447:73–84, 2014.

[10] P. Ravikumar, M. J. Wainwright, G. Raskutti, and B. Yu. High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980, 2011.

[11] P. A. Olsen, F. Öztoprak, J. Nocedal, and S. J. Rennie. Newton-like methods for sparse inverse covariance estimation. *Advances in Neural Information Processing Systems*, 25:764–772, 2012.

[12] H. Liu, J. Lafferty, and L. Wasserman. The nonparanormal: Semiparametric estimation of high dimensional undirected graphs. *The Journal of Machine Learning Research*, 10:2295–2328, 2009.

[13] H. Liu, F. Han, M. Yuan, J. Lafferty, and L. Wasserman. High-dimensional semiparametric gaussian copula graphical models. *The Annals of Statistics*, 40(4):2293–2326, 2012.

[14] S. L. Lauritzen. *Graphical Models.* Oxford University Press, 1996.

[15] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, 1979.

[16] W. P. Adams and H. D. Sherali. A tight linearization and an algorithm for zero-one quadratic programming problems. *Management Science*, 32(10):1274–1290, 1986.

[17] Y. Asahiro, K. Iwama, H. Tamaki, and T. Tokuyama. Greedily finding a dense subgraph. *Journal of Algorithms*, 34(2):203–221, 2000.

[18] A. V. Aho, J. E. Hopcroft, and J. Ullman. *Data Structures and Algorithms.* Addison-Weely, 1983.

[19] M. K. Kozlov, S. P. Tarasov, and L. G. Khachiyan. Polynomial solvability of convex quadratic programming. *Soviet Mathematics Doklady*, 20(5):1108–1111, 1979.

[20] A. Kozma, C. Conte, and M. Diehl. Benchmarking large scale distributed convex quadratic programming algorithms. *Optimization Methods and Software*, 30(1):191–214, 2015.

[21] X. T. Yuan and T. Zhang. Truncated power method for sparse eigenvalue problems. *The Journal of Machine Learning Research*, 14:899–925, 2013.

[22] K. Mohan, M. Chung, S. Han, D. Witten, S. I. Lee, and M. Fazel. Structured learning of Gaussian graphical models. *Advances in Neural Information Processing Systems*, 25:629–637, 2012.

[23] E. Keogh, Q. Zhu, B. Hu, Y. Hao, X. Xi, L. Wei, and C. A. Ratanamahatana. UCR Time Series Classification/Clustering Homepage, 2011. http://www.cs.ucr.edu/~eamonn/time_series_data/.