Multi-task Multi-modal Models for Collective Anomaly Detection

Tsuyoshi Idé
IBM Research
T. J. Watson Research Center
Email: tide@us.ibm.com

Dzung T. Phan
IBM Research
T. J. Watson Research Center
Email: phandu@us.ibm.com

Jayant Kalagnanam
IBM Research
T. J. Watson Research Center
Email: jayant@us.ibm.com

***Abstract*—This paper proposes a new framework for anomaly detection when collectively monitoring many complex systems. The prerequisite for condition-based monitoring in industrial applications is the capability of (1) capturing multiple operational states, (2) managing many similar but different assets, and (3) providing insights into the internal relationship of the variables.**

To meet these criteria, we propose a multi-task learning approach based on a sparse mixture of sparse Gaussian graphical models (GGMs). Unlike existing fused- and group-lasso-based approaches, each task is represented by a sparse mixture of sparse GGMs, and can handle multi-modalities. We develop a variational inference algorithm combined with a novel sparse mixture weight selection algorithm. To handle issues in the conventional automatic relevance determination (ARD) approach, we propose a new ℓ_0-regularized formulation that has guaranteed sparsity in mixture weights. We show that our framework eliminates well-known issues of numerical instability in the iterative procedure of mixture model learning. We also show better performance in anomaly detection tasks on real-world data sets. To the best of our knowledge, this is the first proposal of multi-task GGM learning allowing multi-modal distributions.

1. Introduction

Keeping good operational conditions of industrial equipment is a major business interest across many industries. Although detecting indications of system malfunctions from noisy sensor data is sometimes challenging even to seasoned engineers, statistical machine learning has a lot of potential to automatically capture major patterns of normal operating conditions for condition-based monitoring (CbM). In a typical setting, physical sensor data from multiple sensors are taken as the input, and *anomaly scores*, numerical values representing the degree of anomalousness of the operational state, are computed. Then human administrators decide to take actions to mitigate the risk of *e.g.* service interruptions.

We are interested in the scenario where there is a collection of many assets that are similar but not identical, and we wish to develop a comprehensive monitoring system by leveraging the commonality of those assets while paying attention to the individuality of each. This is a frequently encountered problem in Internet-of-Things (IoT) applications. For example, a car company may wish to build a fault detection model for thousands of electric vehicles in a certain area. Since the occurrence of vehicle malfunction is quite rare, it is tempting to combine information from individual vehicles to get some common insights. On the other hand, since driving conditions can be significantly different for each driver, the model should capture the individuality of the individual vehicles.

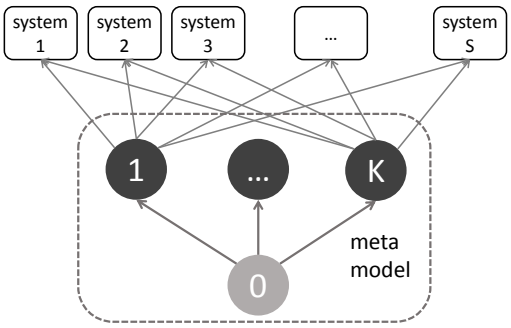

Figure 1. Overall model structure of the multi-task multi-modal model for collective condition-based monitoring.

To formalize the task of anomaly detection for a collection of systems, we leverage the framework of *multi-task learning* (MTL) [1]. In practical CbM scenarios, firstly, anomaly detection models must not be a black box. A detection model has to provide quantitative insights into the individual role of each variable, based on which human operators can see what is really happening in the system. Secondly, a detection model must handle a variety of normal operating conditions, *i.e. multi-modality*. For example, sensor data from an electric vehicle may have drastically different statistical natures between when starting the engine and when cruising on highways.

To this end, we focus on multi-task learning of Gaussian graphical models (GGMs). Thanks to sparsity-enforcing regularization techniques [2], [3], GGMs are known to be a powerful tool in anomaly detection from the viewpoint of interpretability and robustness to the noise [4], [5]. To learn GGMs in the multi-task setting, there have been proposed mainly three techniques so far: group-lasso-based [6], [7],

[8], fused-lasso-based [9], [10], and Bayesian methods [11]. However, most of the existing studies aim at learning a single common graph across the tasks and are unable to handle multi-modal natures of the real world.

The main motivation of this paper is to extend existing work to be able to handle multi-modalities and to propose a practical framework for collective CbM. As illustrated in Fig. 1, our model lets all the S tasks (or systems) share the K sparse GGMs as a "pattern dictionary." The individuality of each task is represented by the mixture weights over those K patterns. The mixture weights and the K GGMs are learned from data based on a Bayesian formulation.

The contribution of this paper is threefold:

- The first proposal of a multi-task multi-modal GGM learning model.
- The first derivation of a variational Bayes algorithm having a guaranteed sparsity in *both* variable relationship *and* mixture weights.
- The first proposal of a practical CbM framework for a fleet of assets.

Regarding the second point, we propose a novel ℓ_0-regularized formulation for mixture weight determination. This indeed sheds a new mathematical light on the traditional notion of automatic relevance determination (ARD) [12], [13], [14] for Bayesian mixture models.

2. Problem setting

2.1. Data and notations

We are given a training data set $\mathcal{D} = \mathcal{D}^1 \cup \ldots \cup \mathcal{D}^S$, where $\mathcal{D}^s$ is the data set for the s-th system or *task* (the term task is used interchangeably with system in this paper). $\mathcal{D}$ is assumed to be collected under the normal conditions of the systems. Each $\mathcal{D}^s$ is a set of N^s samples as

$$\mathcal{D}^s = \{\boldsymbol{x}^{s(n)} \in \mathbb{R}^M \mid n = 1, \ldots, N^s\}, \tag{1}$$

where M is the dimensionality of the samples (or the number of sensors), which is assumed to be the same across the tasks. We let S be the total number of tasks and $N = \sum_{s=1}^{S} N^s$ be the total number of samples. We use the superscript to represent the sample and task indexes. Vectors are represented with the bold face, *e.g.* $\boldsymbol{x}^{s(n)} = (x_1^{s(n)}, \ldots, x_i^{s(n)}, \ldots, x_M^{s(n)})^\top$, and matrices are represented with the sans serif face, *e.g.* $\mathsf{\Lambda}^k = (\Lambda^k_{i,j})$. The elements of vectors and matrices are denoted with the subscripts. As outlined in Introduction, we use a mixture model to capture multi-modalities. Each mixture component is indexed typically by k (and sometimes l), which appears either as the super- or subscript (see Sec. 3 for the detail).

2.2. Anomaly score

Our goal is to compute the anomaly score for a (set of) new sample(s) observed in an arbitrary task. For a new sample $\boldsymbol{x}$ in the s-th task, following [15], we define the overall anomaly score as

$$a^s(\boldsymbol{x}) = -\ln p^s(\boldsymbol{x} \mid \mathcal{D}), \tag{2}$$

up to unimportant additive and multiplicative constants, where $p^s(\cdot|\mathcal{D})$ is the predictive distribution of the s-th task, which is to be learned based on the training data $\mathcal{D}$ (eventually given by Eq. (34)).

In addition to the overall anomaly score, we also define the variable-wise anomaly scores using the negative log conditional predictive distribution as

$$a_i^s(\boldsymbol{x}) = -\ln p^s(x_i \mid \boldsymbol{x}_{-i}, \mathcal{D}), \tag{3}$$

where a_i^s denotes the anomaly score for the i-th variable at the s-th task, and $\boldsymbol{x}_{-i} \equiv (x_1, \ldots, x_{i-1}, x_{i+1}, x_M)^\top$. To compute this, we need detailed information on the variable dependency. Unlike other outlier detection methods such as one-class support vector machines [16], GGMs provides a clear-cut way of computing the predictive conditional distribution. This is a major reason why we focus on GGM-based anomaly detection methods in this paper. As long as using GGM as a basic building block of the model, the conditional distribution can easily be obtained via the standard partitioning formula of Gaussians [13].

Since sensor data of industrial applications are very noisy in general, we are often interested in averaged anomaly scores over a sliding window. If we denote the window by $\mathcal{D}_{\text{test}}$, the averaged version of the anomaly scores are defined as

$$a^s(\mathcal{D}^s_{\text{test}}) = -\frac{1}{|\mathcal{D}^s_{\text{test}}|} \sum_{\boldsymbol{x}^s \in \mathcal{D}^s_{\text{test}}} \ln p^s(\boldsymbol{x}^s \mid \mathcal{D}), \tag{4}$$

$$a_i^s(\mathcal{D}^s_{\text{test}}) = -\frac{1}{|\mathcal{D}^s_{\text{test}}|} \sum_{\boldsymbol{x}^s \in \mathcal{D}^s_{\text{test}}} \ln p^s(x_i \mid \boldsymbol{x}^s_{-i}, \mathcal{D}), \tag{5}$$

where $|\mathcal{D}^s_{\text{test}}|$ is the size of the set $\mathcal{D}^s_{\text{test}}$.

2.3. Motivating example

To fully understand the need for multi-task multi-modal models in real applications, consider a one-dimensional (1D) two-task example. To be specific, imagine we are monitoring two vehicles (two tasks) through the temperature (single variable) of a wheel axle of each car. In Fig. 2, the top row is for Vehicle 1 and the bottom row is for Vehicle 2. The histograms in the same row are all the same, showing the empirical distribution (*i.e.* ground truth) of the temperature. Since driving conditions should be different between Vehicle 1 and 2, the histogram for Vehicle 1 is different from that of Vehicle 2. Possibly due to weather conditions (rainy or not), it is likely for the temperature to have a bi-modal distribution. This is a simple example of multi-task and multi-modal situations.

To fit the empirical distribution, the figure compares three different approaches: multi-task and multi-modal (MTL-MM), non-MTL Gaussian mixture (GMM), and single-modal MTL models, corresponding to the columns.

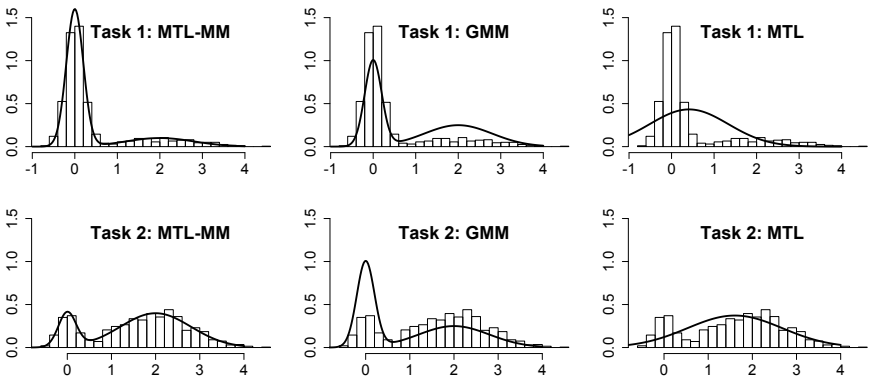

Figure 2. Example of multi-modal distributions with task-dependence. One variable ($M = 1$) and two task ($S = 2$) case is shown. The histograms show the empirical distribution (ground truth) and are fit by three different models: multi-task multi-modal (MTL-MM), standard Gaussian mixture (GMM), and multi-task learning (MTL) models. Only MTL-MM can capture the multi-modality in a task-dependent fashion.

The curves illustrate typical results of fitting. As shown in the figure, traditional non-MTL Gaussian mixture models (GMM; second column) disregard the individuality of the tasks, and existing GGM-based MTL models (third column) cannot handle the multi-modality since their goal is to find a single precision matrix on a task-wise basis. It is clear that these models lead to significant error in anomaly detection. Our goal is to develop a GGM-based MTL model that is capable of handling the multi-modality while taking advantage of task-relatedness. Although this is an illustration with a 1D model, we are interested in modeling *multivariate* systems, *i.e.*, $M > 1$.

3. Multi-task sparse GGM mixture

To capture multiple operational states of the systems, we employ a novel probabilistic Gaussian mixture model featuring double sparsity: sparsity in the dependency structure of GGM and sparsity over the mixture components. This section focuses mainly on the former along with the overall framework.

3.1. Observation model and priors

We employ a Bayesian Gaussian mixture model having K mixture components. First, we define the observation model of the s-th task by

$$p(\boldsymbol{x}^s \mid \boldsymbol{z}^s, \boldsymbol{\mu}, \Lambda) \equiv \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{x}^s \mid \boldsymbol{\mu}^k, (\Lambda^k)^{-1})^{z_k^s}, \quad (6)$$

where $\boldsymbol{\mu}$ and Λ are collective notations representing $\{\boldsymbol{\mu}^k\}$ and $\{\Lambda^k\}$, respectively. Also, $\boldsymbol{z}^s$ is the indicator variable of cluster assignment. As usual, $z_k^s \in \{0, 1\}$ for all s, and $\sum_{k=1}^{K} z_k^s = 1$.

We place the Gauss-Laplace prior on $(\boldsymbol{\mu}^k, \Lambda^k)$ and the categorical distribution on $\boldsymbol{z}$:

$$p(\boldsymbol{\mu}^k, \Lambda^k) = \mathcal{N}(\boldsymbol{\mu}^k | \boldsymbol{m}^0, (\lambda_0 \Lambda^k)^{-1}) \, \mathrm{Lap}(\Lambda^k | \rho), \quad (7)$$

$$\mathrm{Lap}(\Lambda^k | \rho) \equiv \left(\frac{\rho}{4}\right)^{M^2} \exp\left(-\frac{\rho}{2} \|\Lambda^k\|_1\right), \quad (8)$$

$$p(\boldsymbol{z}^s | \boldsymbol{\pi}^s) = \prod_{k=1}^{K} (\pi_k^s)^{z_k^s} \ \text{ s.t. } \ \sum_{k=1}^{K} \pi_k^s = 1, \ \pi_k^s \geq 0, \quad (9)$$

where $\|\Lambda^k\|_1 = \sum_{i,j} |\Lambda_{i,j}^k|$. The parameter $\boldsymbol{\pi}^s$ is determined as a part of the model while $\rho, \lambda_0, \boldsymbol{m}^0$ are given constants. From these equations, we can write down the complete likelihood as

$$P(\mathcal{D}, \mathsf{Z}, \Lambda, \boldsymbol{\mu} \mid \boldsymbol{\pi}) \equiv \prod_{k=1}^{K} p(\boldsymbol{\mu}^k, \Lambda^k) \times \prod_{s=1}^{S} \prod_{n=1}^{N^s} p(\boldsymbol{z}^{s(n)} | \boldsymbol{\pi}^s) p(\boldsymbol{x}^{s(n)} \mid \boldsymbol{z}^{s(n)}, \boldsymbol{\mu}, \Lambda), \quad (10)$$

where $\boldsymbol{z}^{s(n)}$ is the cluster assignment variable for the n-th sample in the s-th task. $\boldsymbol{\pi}$ and Z are collective notations for $\{\boldsymbol{\pi}^s\}$ and $\{\boldsymbol{z}_k^{s(n)}\}$, respectively.

Note that $\{\boldsymbol{\mu}^k\}$ and $\{\Lambda^k\}$ are *not* task-specific and shared by all the tasks. It is the cluster assignment probability $\boldsymbol{\pi}^s$ that reflects the individuality of the tasks. Thus $\boldsymbol{\pi}^s$ can be used as the *signature* of the s-th task.

3.2. Variational Bayes inference

A general goal of Bayesian formulations is to find the posterior distributions. We leverage the variational Bayes (VB) approach [13] to get a tractable algorithm. The central assumption of VB is that the posterior distribution has a

factorized form. In this case, we assume the categorical distribution for Z and the *Gauss-delta* distribution for $(\boldsymbol{\mu}, \Lambda)$:

$$q(\mathsf{Z}) = \prod_{s=1}^{S} \prod_{n=1}^{N^s} \prod_{k=1}^{K} (r_k^{s(n)})^{z_k^{s(n)}}, \tag{11}$$

$$q(\boldsymbol{\mu}, \Lambda) = \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{\mu}^k | \boldsymbol{m}^k, (\lambda_k \Lambda^k)^{-1}) \delta(\Lambda^k - \bar{\Lambda}^k), \tag{12}$$

where $\delta(\cdot)$ is Dirac's delta function and $\{r_k^{s(n)}, \boldsymbol{m}^k, \lambda_k, \bar{\Lambda}^k\}$ are model parameters to be learned. We combine VB analysis for $\{\mathsf{Z}, \boldsymbol{\mu}, \Lambda\}$ with point estimation for the mixture weight $\boldsymbol{\pi}^s$.

In the VB formulation, the model parameters $\{\boldsymbol{m}^k, \lambda_k, \bar{\Lambda}^k\}$ are determined so that the Kullback-Leibler (KL) divergence between $q(\mathsf{Z})q(\boldsymbol{\mu}, \Lambda)$ and $P(\mathcal{D}, \mathsf{Z}, \Lambda, \boldsymbol{\mu} | \boldsymbol{\pi})$ is minimized. It is well-known [13] that minimization of the KL divergence leads to extremely simple iterative equations:

$$\ln q(\mathsf{Z}) = \mathrm{c.} + \langle \ln P(\mathcal{D}, \mathsf{Z}, \Lambda, \boldsymbol{\mu} | \boldsymbol{\pi}) \rangle_{\Lambda, \boldsymbol{\mu}}, \tag{13}$$

$$\ln q(\Lambda, \boldsymbol{\mu}) = \mathrm{c.} + \langle \ln P(\mathcal{D}, \mathsf{Z}, \Lambda, \boldsymbol{\mu} | \boldsymbol{\pi}) \rangle_{\mathsf{Z}}, \tag{14}$$

where c. symbolically represents an unimportant constant, $\langle \cdot \rangle_{\Lambda, \boldsymbol{\mu}}$ is the expectation w.r.t. $q(\boldsymbol{\mu}, \Lambda)$, and $\langle \cdot \rangle_{\mathsf{Z}}$ is the expectation w.r.t. $q(\mathsf{Z})$.

To compute these expectations, we need to know the value of $\boldsymbol{\pi}$. In the proposed VB framework, an optimization problem to determine $\boldsymbol{\pi}$ is solved alternately with Eqs. (13) and (14) until convergence. We will discuss the details in Section 4.

3.3. VB iterative equations

Now let us find explicit expressions of the VB equations (13) and (14). Given $\{\boldsymbol{m}^k, \lambda_k, \bar{\Lambda}^k\}$ and an initialized $\boldsymbol{\pi}^s$, the first VB equation (13) gives

$$\ln r_k^{s(n)} \leftarrow \ln \left\{ \pi_k^s \, \mathcal{N}(\boldsymbol{x}^{s(n)} \mid \boldsymbol{m}^k, (\bar{\Lambda}^k)^{-1}) \right\} - \frac{M}{2\lambda_k} \tag{15}$$

$$r_k^{s(n)} \leftarrow \frac{r_k^{s(n)}}{\sum_{l=1}^{k} r_l^{s(n)}}. \tag{16}$$

To get the first equation, we calculated the expectation w.r.t. $\boldsymbol{\mu}^k$ and Λ^k using the expression of Eq. (12). The second equation is due to the normalization condition $\sum_k \pi_k^s = 1$.

To solve the second VB equation (14), we first decompose the posterior as $q(\boldsymbol{\mu}, \Lambda) = q(\boldsymbol{\mu}|\Lambda) q(\Lambda)$. For $q(\boldsymbol{\mu}^k | \Lambda^k)$, by arranging the terms of $\langle \ln P \rangle_{\mathsf{Z}}$ related to $\boldsymbol{\mu}^k$, we readily get

$$N_k \leftarrow \sum_{s=1}^{S} \sum_{n=1}^{N^s} r_k^{s(n)}, \tag{17}$$

$$\bar{\boldsymbol{x}}^k \leftarrow \frac{1}{N_k} \sum_{s=1}^{S} \sum_{n=1}^{N^s} r_k^{s(n)} \boldsymbol{x}^{s(n)}, \tag{18}$$

$$\lambda_k \leftarrow \lambda_0 + N_k, \tag{19}$$

$$\boldsymbol{m}^k \leftarrow \frac{1}{\lambda_k} (\lambda_0 \boldsymbol{m}^0 + N_k \bar{\boldsymbol{x}}^k), \tag{20}$$

given $\{\Lambda^k, r_k^{s(n)}\}$.

For $q(\Lambda)$, the VB equation does not have an analytic solution. We instead find the mode of $\ln q(\Lambda)$ by solving

$$\bar{\Lambda}^k \leftarrow \arg\max_{\Lambda^k} \left\{ \ln |\Lambda^k| - \mathrm{Tr}(\Lambda^k \mathsf{Q}^k) - \frac{\rho}{N_k} \|\Lambda^k\|_1 \right\}, \tag{21}$$

with

$$\Sigma^k \leftarrow \frac{1}{N_k} \sum_{s=1}^{S} \sum_{n=1}^{N^s} r_k^{s(n)} \boldsymbol{x}^{s(n)} \boldsymbol{x}^{s(n)\top} - \bar{\boldsymbol{x}}^k (\bar{\boldsymbol{x}}^k)^\top \tag{22}$$

$$\mathsf{Q}^k \leftarrow \Sigma^k + \frac{\lambda_0}{\lambda_k} (\bar{\boldsymbol{x}}^k - \boldsymbol{m}^0)(\bar{\boldsymbol{x}}^k - \boldsymbol{m}^0)^\top. \tag{23}$$

As shown in [2], the objective function in Eq. (21) is convex. This means that the posterior $q(\Lambda)$ is guaranteed to be unimodal, and approximating $q(\Lambda)$ by the delta function is reasonable.

As stated earlier, the VB iterative equations (15)-(23) are combined with point-estimation of $\boldsymbol{\pi}^s$. The next section discusses the details of the approach.

4. Sparse mixture weight selection

This section introduces a novel formulation to find a sparse solution for $\{\boldsymbol{\pi}^s\}$.

4.1. Conventional ARD approach

To determine $\boldsymbol{\pi}$, the conventional VB formulation [13] maximizes $\langle \ln P(\mathcal{D}, \mathsf{Z}, \Lambda, \boldsymbol{\mu} | \boldsymbol{\pi}) \rangle_{\Lambda, \boldsymbol{\mu}, \mathsf{Z}}$ under the normalization condition. With Eqs. (9) and (10), we readily have

$$\langle \ln P(\mathcal{D}, \mathsf{Z}, \Lambda, \boldsymbol{\mu} | \boldsymbol{\pi}^s) \rangle_{\Lambda, \boldsymbol{\mu}, \mathsf{Z}} = \mathrm{c.} + \sum_{n=1}^{N^s} \sum_{k=1}^{K} \langle z_k^{s(n)} \rangle_{\mathsf{Z}} \ln \pi_k^s$$

as a function of $\boldsymbol{\pi}^s$. The expectation is computed using Eq. (11) as $\langle z_k^{s(n)} \rangle_{\mathsf{Z}} = r_k^{s(n)}$. Now the optimization problem we solve reads

$$\max_{\boldsymbol{\pi}^s} \sum_{k=1}^{K} c_k^s \ln \pi_k^s \quad \text{s.t.} \quad \|\boldsymbol{\pi}^s\|_1 = 1, \tag{24}$$

where $\|\boldsymbol{\pi}^s\|_1$ is the ℓ_1 norm of $\boldsymbol{\pi}^s$, and we defined

$$c_k^s \equiv \frac{1}{N^s} \sum_{n=1}^{N^s} r_k^{s(n)} \tag{25}$$

so $\sum_{k=1}^{K} c_k^s = 1$ holds.

By introducing a Lagrange multiplier for the constraint $\|\boldsymbol{\pi}^s\|_1 = 1$, it is straightforward to show that the optimal solution $\boldsymbol{\pi}^{s*}$ is given by

$$\pi_k^{s*} = c_k^s. \tag{26}$$

As discussed in [14], when combined with a small threshold value below which π_k^s is regarded as zero, the problem (24) often gives a sparse solution, which is sometimes referred to as an instance of the automatic relevance determination (ARD). However, π_k^s cannot be mathematically zero because of the logarithm function. This means that the sparsity is governed by the heuristically provided numerical threshold and the convergence of the VB algorithm may depend on chance. In fact, the conventional VB iterative algorithm is known to be sometimes numerically unstable. This can be a serious issue especially in the multi-task anomaly detection scenario since we need to manage S different anomaly detection models at once.

Keeping this fundamental limitation of the conventional VB formulation in mind, we introduce a new formulation for sparse mixture weight selection in the next subsection.

4.2. Convex mixed-integer programming approach

To achieve sparsity in a mathematically well-defined fashion in (24), *first*, we explicitly impose regularization on $\boldsymbol{\pi}^s$. Similarly to the Laplace prior on Λ^k, let us formally assume that $\boldsymbol{\pi}^s$ has a prior in the form of $p(\boldsymbol{\pi}^s) \sim \exp(-\tau\|\boldsymbol{\pi}^s\|_0/N^s)$, where $\|\cdot\|_0$ denotes the ℓ_0-norm (the number of nonzeros), and $\tau > 0$ is a constant assumed to be given. The optimization problem now looks like:

$$\max_{\boldsymbol{\pi}^s} \left\{ \sum_{k=1}^{K} c_k^s \ln \pi_k^s - \tau\|\boldsymbol{\pi}^s\|_0 \right\} \text{ s.t. } \|\boldsymbol{\pi}^s\|_1 = 1. \tag{27}$$

Obviously, the solution (26) is recovered when $\tau = 0$. Note that we cannot use the ℓ_1 norm here because of the constraint $\|\boldsymbol{\pi}^s\|_1 = 1$.

Second, we formally define the notion of ϵ-sparsity:

Definition 1. *For a given small ϵ, a vector $\boldsymbol{x}$ is called an ϵ-sparse solution if many elements satisfy $|x_i| \le \epsilon$.*

Third and finally, we modify the problem (27) into a convex mixed-integer programming (MIP) to get an ϵ-sparse solution:

$$\max_{\boldsymbol{\pi}^s, \boldsymbol{y}^s} \sum_{k=1}^{K} \{c_k^s \ln \pi_k^s - \tau y_k^s\} \text{ s.t. } \sum_{k=1}^{K} \pi_k^s = 1,$$
$$y_k^s \ge \pi_k^s - \epsilon, \;\; y_k^s \in \{0,1\} \text{ for } k = 1, \ldots, K, \tag{28}$$

where $0 < \epsilon \ll 1$ is another constant controlling the sparsity. $\boldsymbol{y}^s$ plays a role of indicator variable of $\boldsymbol{\pi}^s$. Notice that the inequality constraint $y_k^s \ge \pi_k^s - \epsilon$ grantees that $y_k^s = 1$ when $\pi_k^s > \epsilon$ and $y_k^s = 0$ when $\pi_k^s \le \epsilon$. The latter follows from the fact that $y_k^s = 1$ gives a smaller objective value and can be ignored when seeking an optimal solution. Thus we see that $\|\boldsymbol{\pi}^s\|_0$ is equal to $\sum_{k=1}^{K} y_k^s$.

We also see that the problem (28) is convex. By directly calculating the second derivative w.r.t. $\boldsymbol{\pi}^s$ and $\boldsymbol{y}^s$, we see that the Hessian is a $2K \times 2K$ diagonal matrix whose diagonal element is either $-c_k^s/(\pi_k^s)^2$ $(k = 1, \ldots, K)$ or 0. Since $c_k^s > 0$, the Hessian is negative semi-definite. Also, all decision variables are bounded in $[0, 1]$, and every constraint is linear. Thus the problem (28) is a convex mixed-integer programming with a bounded polyhedron feasible set.

Since ϵ is just an explicit representation of the threshold value that has been used heuristically [13] and the objective function is dominated by the first term when $\tau \ge 0$ is small, we conclude that the problem (28) is a mathematically well-defined surrogate of the original (24).

Let us formally summarize the above discussion:

Theorem 1 (Convex MIP mixture weight selection)**.**

(i) *The problem* (28) *is a convex mixed-integer programming with a bounded polyhedron feasible set.*
(ii) *The problem* (28) *generates an ϵ-sparse solution for a suitable selection of τ.*
(iii) *There exist small enough positive numbers τ and ϵ such that* (26) *is a solution of* (28)*.*

4.3. Solving Eq. (28)

Although solving MIP generally involves exhaustive combinatorial search and thus computationally very expensive, we can derive an efficient algorithm for the problem (28). The strategy is simple. We find a solution of (28) for each value of $\sum_k y_k^s$, and pick the best one from them. This is a practical approach since K is on the order of 10 in most CbM scenarios. In this subsection, we illustrate the outline of the approach. For proofs and more detailed discussions, the reader can refer to our companion paper [17].

Without loss of generality, we can assume that $\{c_k^s\}$ have been sorted in increasing order, $c_1^s \le \cdots \le c_K^s$. Since the objective is symmetric w.r.t. k, in order to remove duplicated solutions, we also assume $\pi_i^s \le \pi_j^s$, when $c_i^s = c_j^s$ and $i < j$. In that case, since we are solving a maximization problem, we intuitively expect that, for a given $K_0 \equiv K - \sum_k y_k^s$,

$$y_1^s = \cdots = y_{K_0}^s = 0, \;\; y_{K_0+1}^s = \cdots = y_K^s = 1 \tag{29}$$

because this choice keeps as many larger c_k^s's as possible. Based on this, we can eliminate $\boldsymbol{y}^s$ from (28) to define a K_0-specific problem:

$$\max_{\boldsymbol{\pi}^s} \sum_{k=1}^{K} c_k^s \ln \pi_k^s \text{ s.t. } \sum_{k=1}^{K} \pi_k^s = 1,$$
$$\pi_k^s \le \epsilon \text{ for } k = 1, \ldots, K_0. \tag{30}$$

To find the optimality condition, we define the Lagrange function as

$$\mathcal{L}(\boldsymbol{\pi}^s, \boldsymbol{\alpha}^s, \eta^s) \equiv \sum_{k=1}^{K} c_k^s \ln \pi_k^s - \eta^s \sum_{k=1}^{K} \pi_k^s - \sum_{k=1}^{K_0} \alpha_k^s (\epsilon - \pi_k^s),$$

where $\{\alpha_k^s\}$ and η^s are Lagrange's multipliers. By differentiating $\mathcal{L}$ w.r.t. $\boldsymbol{\pi}^s$, we have the Karush-Kuhn-Tucker (KKT) conditions for the problem (30):

$$\frac{c_k^s}{\pi_k^s} = \begin{cases} \eta^s + \alpha_k^s, & k \leq K_0 \\ \eta^s, & k > K_0, \end{cases} \tag{31}$$

$$\alpha_k^s(\epsilon - \pi_k^s) = 0, \;\; \alpha_k^s \geq 0 \;\; \text{for} \;\; k \leq K_0. \tag{32}$$

This leads to the solution for the assumed K_0:

$$\pi_k^{s*}(K_0) = \begin{cases} \epsilon, & k \leq K_0, \; c_k^s \geq \epsilon\eta^s, \\ \frac{c_k^s}{\eta^s}, & \text{otherwise}, \end{cases} \tag{33}$$

where the condition $c_k^s \geq \epsilon\eta^s$ comes from $\alpha_k^s \geq 0$. The multiplier η^s is determined so $\sum_{k=1}^K \pi_k^s = 1$. It is easy to verify that Eq. (33) satisfies the KKT conditions. For more mathematical details, see our companion paper [17].

The solution $\pi_k^{s*}(K_0)$ is computed for different K_0's, and we pick the one which gives the maximum objective value of Eq. (28) (not (30)). The computational cost to find the solution is on the order of K^2 in the worst case.

4.4. Algorithm summary and remarks

Equations (15)-(23) and (28) are iteratively computed for all the components k and the tasks s until convergence. Notice that the equation for $\bar{\Lambda}^k$ preserves the original ℓ_1-regularized GGM formulation [3]. We see that the fewer samples a cluster have, the more the ℓ_1 regularization is applied due to the ρ/N^k term. This means that we do not trust samples assigned to minor clusters too much. To solve this, we can use, *e.g.*, the graphical lasso algorithm [3].

Once all the model parameters are found, with $\mathsf{A}^k \equiv \frac{\lambda_k}{1+\lambda_k}\bar{\Lambda}^k$ the predictive distribution is given by

$$p^s(\boldsymbol{x}^s|\mathcal{D}) = \sum_{k=1}^K \pi_k^s \int \mathrm{d}\boldsymbol{\mu}^k \int \mathrm{d}\Lambda^k \, \mathcal{N}(\boldsymbol{x}^s|\boldsymbol{\mu}^k, (\Lambda^k)^{-1}) q(\boldsymbol{\mu}^k, \Lambda^k),$$

$$= \sum_{k=1}^K \pi_k^s \mathcal{N}(\boldsymbol{x}^s \mid \boldsymbol{m}^k, (\mathsf{A}^k)^{-1}). \tag{34}$$

Algorithm 1 summarizes the proposed `MTL-MM` algorithm. To initialize $\{\boldsymbol{m}^k, \bar{\Lambda}^k\}$, in the context of industrial CbM, one reasonable approach is to disjointly partition each data along the time axis as $\mathcal{D}^s = \mathcal{D}_1^s \cup \mathcal{D}_2^s \ldots$ and apply the graphical lasso algorithm [3] on each. For data sets of i.i.d. samples, on the other hand, k-means clustering [13] can be used to get $\{\boldsymbol{m}^k\}$, followed by graphical lasso for $\{\Lambda^k\}$. The initial number of mixture components K should be large enough to be able to automatically find an optimal number of non-empty clusters, $K' < K$. For standardized data, $\lambda_0 = 1$ and $\boldsymbol{m}^0 = \mathbf{0}$ are a reasonable choice. For the MIP parameters, τ can be a value in $(0, 1]$ such as 0.1. Since ϵ has the meaning of minimum resolution of mixture weight (the probability to find a sample in the cluster), a value such as 10^{-5} should be reasonable. Virtually the only parameter to be determined via cross-validation is ρ. In the context of anomaly detection, ρ is determined so a performance metric such as the AUC (area-under-curve) and the F-measure is maximized.

Algorithm 1 `Multi-task multi-modal GGM`

procedure MTL-MM($\mathcal{D}$, λ_0, $\boldsymbol{m}^0$, ρ, ϵ,τ)
  Initialize $\{(\boldsymbol{m}^k, \Lambda^k)\}$
  Set $\pi_k^s = \frac{1}{K}, \lambda_k = \lambda_0 + \frac{N}{K}$ for all k, s
  repeat
    for $s \leftarrow 1, S$ **do**
      for $n \leftarrow 1, N^s$ **do**
        for $k \leftarrow 1, K$ **do**
          $r_k^{s(n)} \leftarrow$ Eq. (15)
        end for
        $r_k^{s(n)} \leftarrow r_k^{s(n)} / \sum_{l=1}^K r_l^{s(n)}$
      end for
    end for
    for $k \leftarrow 1, K$ **do**
      for $s \leftarrow 1, S$ **do**
        $\pi_k^s \leftarrow$ Eq. (28)
      end for
      $N_k \leftarrow \sum_{s=1}^S \sum_{n=1}^{N^s} r_k^{s(n)}$
      $\lambda_k \leftarrow \lambda_0 + N_k$
      $\boldsymbol{m}^k \leftarrow$ Eq. (20)
      $\bar{\Lambda}^k \leftarrow$ Eq. (21)
    end for
  until convergence
return $\{\boldsymbol{\pi}^s\}$ and $\{\boldsymbol{\mu}^k, \Lambda^k, \lambda_k\}$
end procedure

5. Related work

In the context of anomaly detection, there are three lines of research relevant to the present work: MTL for anomaly detection, Gaussian mixture models, and MTL for GGM.

As explained in Introduction, the original concept of MTL is highly tempting for anomaly detection, since anomaly samples are always limited. In fact, there are a number of studies [18], [19], [20] to attempt to pursue MTL-based anomaly detection. However, with these methods, it is not straightforward to compute variable-wise contributions and obtain insights into the internal dependency of multivariate systems, which are an integral part of the practical requirements.

Gaussian mixture models have been used in a wide variety of applications, and numerous prior studies exist *e.g.* [10], [15], [21]. However, little is known about how to extend them to MTL in the context of anomaly detection.

Finally, MTL-based sparse GGM learning has been one of the recent hot topics in the machine learning and statistics communities [6], [7], [8], [9], [10], [11]. An MTL-like setting has also been discussed in the context of anomaly detection [5]. However, few of them focus on the multi-modality, which is critical in many real industrial applications, especially in anomaly detection.

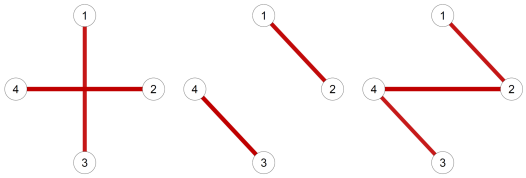

Figure 3. Ground truth precision matrices. See Sec. 6.1.

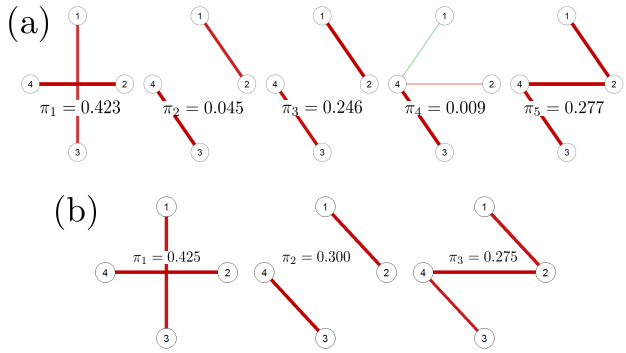

Figure 4. Learned precision matrices. (a) Conventional ARD approach. (b) Proposed MIP approach. Mixture weights are also described on the graphs. See Sec. 6.1.

6. Experiments

This section shows the utility of the proposed multi-task multi-modal framework with the convex MIP-based mixture weight selection. We first demonstrate a better convergence of the convex MIP formulation of the mixture weights. We then test performance in anomaly detection using synthetic and real-world data.

6.1. Comparison with conventional ARD approach

To test the convex MIP formulation for the mixture weight, we generated a 4-variate ($M = 4$) synthetic data set. Since Eq. (28) is solved independently for each task s, we simply set $S = 1$ in this subsection (thus the superscript ^{s} will be dropped for now). We randomly generated $N = 3\,800$ samples with three component Gaussian mixture with $\boldsymbol{\pi} = (0.4, 0.3, 0.3)^\top$. The first component has the mean $(5, 0, 0, 5)^\top$, while both of the second and third components share the same mean of $(0, 5, 5, 0)^\top$. The precision matrices for these components are shown in Fig. 3.

For initialization, as mentioned in Sec. 4.4, we split the data set into $K = 10$ disjoint blocks, and learn the precision matrix using the graphical lasso algorithm. We chose parameters as $\rho = 0.01, \tau = 0.25, \epsilon = 10^{-4}$. In the conventional ARD approach (Eq. (26)), we removed components once $\pi_k < \epsilon$ is satisfied during the iteration. In the proposed MIP method, all the K components are kept during the iteration, and those having $\pi_k < \epsilon$ are removed from the model upon convergence.

Figure 4 shows learned precision matrix (in terms of the partial correlation coefficients) and their mixture weights. We see that the proposed method precisely converged into the ground truth ($K' = 3$) in spite of the initial number of components, $K = 10$, but the conventional approach produced two spurious components. This is one manifestation of numerical instabilities of the conventional ARD method.

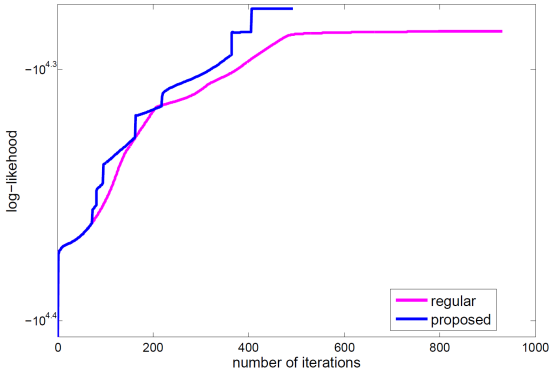

Figure 5. Log likelihood towards convergence. The conventional approach ("regular") fails to find the ground truth. See Sec. 6.1.

To get further insights, we monitored the log likelihood as a function of the number of iterations, as shown in Fig. 5. We see that the proposed MIP formulation found the optimal solution much quicker, while the conventional approach gets stuck with a local minimum. The smooth curve of the conventional approach suggests that the conventional algorithm strongly encourages convergence by forcing smaller components to be even smaller. Although Fig. 5 is just for one instance, in our repeated experiments with different random number seeds for the data, the conventional approach produced a noticeably worse solution in most cases.

In the proposed MTL framework, the most expensive step is to learn $\{\Lambda^k\}$ (Eq. (21)). Although the MIP equation (28) incurs more computational cost than the conventional one per se, the total computational cost per one iteration is dominated by Eq. (21). Thus the smaller the number iterations, the faster we reach the solution. We conclude that the proposed convex MIP-based mixture weight selection approach is faster and more stable than the conventional ARD approach.

6.2. Multi-modal graph learning: synthetic data

To illustrate how `MTL-MM` works, we compare the proposed method with two alternatives that can learn sparse and thus interpretable dependency structures in the MTL setting: the group graphical lasso (`ggl`) and fused graphical lasso (`fgl`) algorithms [9]. These methods find task-wise precision matrices $\{\Lambda^s\}$ by maximizing

$$\sum_{s=1}^{S} N^s \left\{ \ln |\Lambda^s| - \mathrm{Tr}(\hat{\mathsf{S}}^s \Lambda^s) \right\} - \sum_{s=1}^{S} \eta_1 \|\Lambda^s\|_1 - \eta_2 P(\{\Lambda^s\}),$$

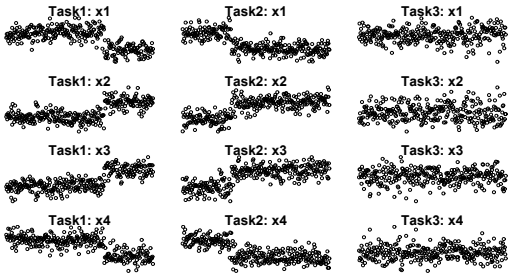

Figure 6. Synthetic data. The sample size is $N^s = 300$ for each task.

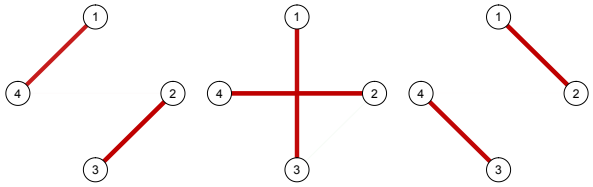

Figure 7. Learned precision matrices in terms of the partial correlation coefficients. See Sec. 6.2.

where $\hat{\mathsf{S}}^s$ is the sample covariance matrix of the s-th task, and

$$P(\{\Lambda^s\}) = \begin{cases} \sum_{i \neq j} \sqrt{\sum_{s=1}^{S} {\Lambda_{i,j}^s}^2} & (\texttt{ggl}) \\ \sum_{s' > s} \sum_{i,j} |\Lambda_{i,j}^s - \Lambda_{i,j}^{s'}| & (\texttt{fgl}) \end{cases} \quad (35)$$

Since the goal of these algorithms to find a single Gaussian graphical model for each task, the anomaly score (2) is defined as $-\ln \mathcal{N}(\cdot|\hat{\boldsymbol{\mu}}^s, \Lambda^s)$ for each task s, where $\hat{\boldsymbol{\mu}}^s$ is the sample mean on the s-th task.

We generated a three-task ($S = 3$) four-variate ($M = 4$) synthetic data. As shown in Fig. 6, the data were generated from three distinctive four-variate Gaussian distributions, say, A, B, and C. The first $\frac{2}{3}$ and $\frac{1}{3}$ of task 1 were generated by A and B, respectively. The first $\frac{1}{3}$ and $\frac{2}{3}$ of task 2 were generated by A and B, respectively. Task 3 was generated only with C. We also independently generated test data using the same pattern combinations.

To train the `MTL-MM` model, we split each of the tasks into halves and used them to initialize $\{(\boldsymbol{m}^k, \Lambda^k)\}$, resulting in $K = 6$ initial number of clusters. Upon convergence, `MTL-MM` gave $K' = 3$ non-empty clusters. We used $\rho = 0.1$, which was chosen as the minimizer of the overall anomaly score on the test data (see below).

Figures 7 and 8 show the learned precision matrices and the mixture weights, respectively. The three graphs in Fig. 7 precisely recover the pattern A, B, and C, from left to right, and the mixture weights are also consistent with the training data. This result confirms the capability of our algorithm to capture multi-modal patterns even under heavy noise. We also note that the correct number of clusters is automatically found thanks to the guaranteed sparsity formulation.

Finally, we computed the averaged anomaly score Eq. (4) for each task on the test data. Since the test data follow the same generative model as the training data, the anomaly score *should be small* as long as the learned model is faithful to the ground truth distribution. In Eq. (4), the

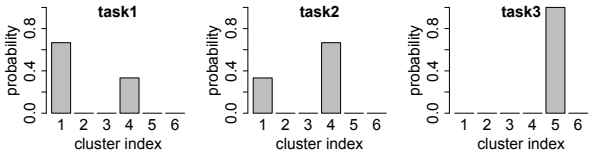

Figure 8. Learned mixture weights $\boldsymbol{\pi}^s$ for $s = 1, 2, 3$.

window size was taken as the same as the total number of samples of each task ($N^s = 300$). To train `ggl` and `fgl`, we fixed $\eta_1 = 0.1$ and computed the anomaly score as a function of η_2. Figure 9 shows the result, where the vertical axes represent the ratio to the overall anomaly scores computed by `MTL-MM`. We see that `ggl` and `fgl` give values close to one only for task 3 while having much larger anomaly scores for tasks 1 and 2, meaning that they failed to capture the underlying distribution correctly. We also see that the group lasso and fused lasso penalties do not help improve the fit of the models. This clearly shows the failure of the existing MTL-based GGM learning approaches in terms of multi-modality in such tasks as 1 and 2.

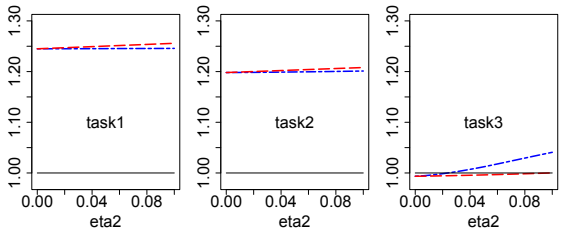

Figure 9. Averaged overall anomaly score calculated by `ggl` (dashed line) and `fgl` (dash-dotted line), relative to that of `MTL-MM` (solid line at the level of 1). Smaller is better.

6.3. Anomaly detection: London School data

The `London School` data[1] are widely used benchmark data for multi-task learning. The data contain the score of an examination of students along with other three student-specific attributes (*VRband*, *gender*, *ethnicity*) and five school-specific attributes. We picked schools with more than or equal to 200 students, ending up with $S = 11$ schools as tasks. Although the original data are not intended for anomaly detection, we held out the students whose *ethnicity* is categorized into "*others*" as anomalous samples. We also randomly picked normal samples so that a half of samples of the test data is normal. To be fair to the uni-modal alternatives, we re-labeled the categorical attributes of *VRband* and *ethnicity* so the largest categories come in the middle and the smallest categories come at the both ends. We used only the student-specific variables ($M = 4$), each of which was standardized to have zero mean and unit variance and then was added Gaussian noise of standard deviation of 0.1. As a result, we obtained 2 355 samples in total over $S = 11$ tasks in the training data, and 104 samples

1. Downloadable at http://www.bristol.ac.uk/cmm/learning/support/datasets/. See [22] for original descriptions.

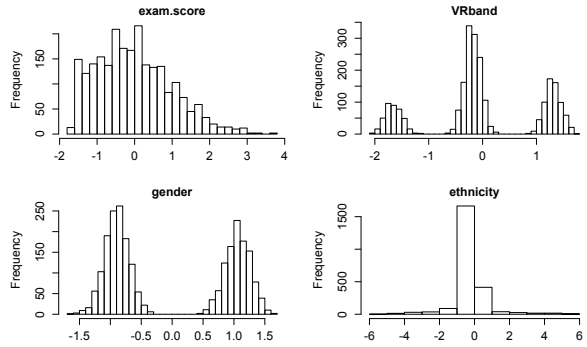

Figure 10. Variable-wise distribution of `London School` training data with Gaussian noise. Samples are aggregated over $S = 11$ schools as tasks.

in total in the test data. Figure 10 shows the distribution of the training data for each variable.

We compared the performance of anomaly detection of `MTL-MM` with `ggl` and `fgl` with the ROC (receiver operating characteristic) curve. Equation (2) was used for anomaly scoring. The regularization parameters were chosen so the AUC values are maximized on the test data: $\rho = 1.1, (\eta_1, \eta_2)_{\texttt{ggl}} = (0.25, 0.25)$, and $(\eta_1, \eta_2)_{\texttt{fgl}} = (0.19, 0.13)$. The k-means clustering (repeated five times to pick the best one) was used to initialize `MTL-MM` with two initial clusters for each task, *i.e.* $K = 22$. The other parameters were set to be their default values as described in Sec. 4.4. Upon convergence, we had $K' = 13$ non-empty clusters.

The ROC curves in Fig. 11 clearly show that the proposed multi-modal model outperforms the uni-modal alternatives. The AUC values are summarized in Table 1. As is evident from Fig. 10, the distribution of this data is multi-modal. Even to *ethnicity*, for example, `MTL-MM` assigned major weights on three Gaussians in most tasks. As a result, the uni-modal alternatives sometimes fail to detect the anomalous samples. This is a clear demonstration of the utility of the proposed mixture model.

6.4. Anomaly detection: Anuran Calls data

Next we applied `MTL-MM` to `Anuran Calls` data [23], a real-world data set collected from frog croaking sounds[2]. We picked three major species of *AdenomeraAndre*, *Ameeregatrivittata*, and *HylaMinuta* as tasks ($S = 3$), in which the first ten MFCCs (Mel-frequency cepstral coefficients) were used as the variables ($M = 10$). Although the original data are not intended for anomaly detection, we used the species of *Rhinellagranulosa* as the anomalous class. For each task, we created an equal mixture between the 68 anomalous samples and randomly picked normal samples. The remaining normal samples were used as the training data. As a result, the test data have 68×2 samples for each task, and the training data have samples of $(N^1, N^2, N^3) = (604, 474, 242)$.

2. Downloadable from UCI Archive at https://archive.ics.uci.edu/ml/.

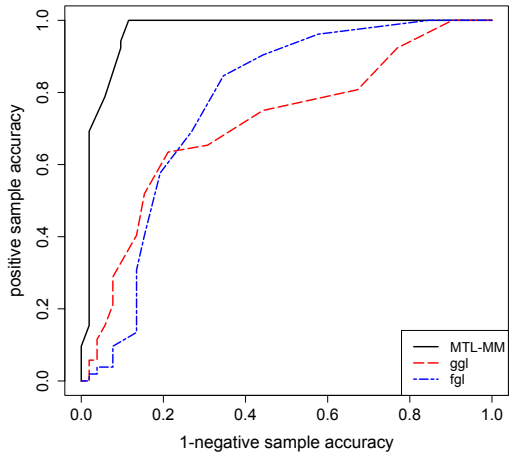

Figure 11. Comparison of anomaly detection performance on the `London School` data. The test samples are aggregated over the $S = 11$ tasks.

We again compared the performance of anomaly detection with the ROC curve. To train `MTL-MM`, we initialized the cluster with the k-means scheme, where three initial clusters were generated for each task, *i.e.* $K = 9$. The parameters ρ, η_1, η_2 were chosen so the AUC is maximized, resulting in $\rho = 9.5, (\eta_1, \eta_2)_{\texttt{ggl}} = (10^{-2}, 0.02)$, and $(\eta_1, \eta_2)_{\texttt{fgl}} = (10^{-4}, 10^{-5})$. The other parameters were set to be their default values as described in Sec. 4.4. Upon convergence, we got only $K' = 2$ non-empty clusters as shown in Fig. 12. For the mixture weight between these, we obtained $\boldsymbol{\pi}^1 = (0.59, 0.41)^\top$, $\boldsymbol{\pi}^2 = (0.13, 0.87)^\top$, and $\boldsymbol{\pi}^3 = (0.00, 1.00)^\top$. These numbers suggest that the first and second tasks have significant multi-modality. In fact, this is the major reason why they mis-predicted on the test samples, as evidenced by the ROC curves in Fig. 13 and their AUC values in Table 1.

7. Conclusion

We have proposed a new framework for collective anomaly detection based on a Bayesian multi-task multi-modal sparse mixture of sparse GGMs. By combining the variational Bayes framework with (1) Laplace prior-based sparse structure learning and (2) a novel ℓ_0-based sparse mixture weight selection approach, our formulation has guaranteed dual sparsity over both variable-variable dependency and mixture components, which helps to efficiently learn multi-modal distributions that are very often observed in Internet-of-Things applications. We confirmed that our formulation successfully eliminated the well-known issue

TABLE 1. AUC VALUES.

	`MTL-MM`	`ggl`	`fgl`
`London School` (Fig. 11)	**0.967**	0.714	0.770
`Anuran Calls` (Fig. 13)	**0.849**	0.654	0.637

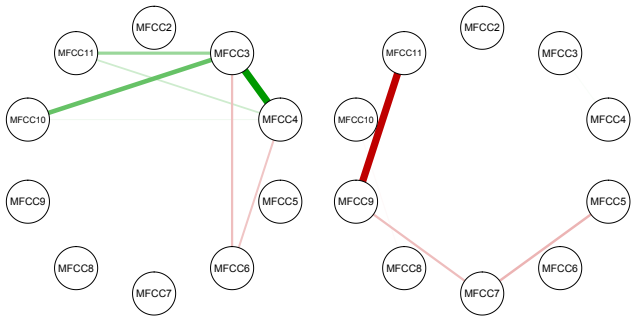

Figure 12. GGMs learned with `MTL-MM` in terms of the partial correlation coefficients. The line width is proportional to their values. The green and red lines represent positive and negative nonzero values, respectively.

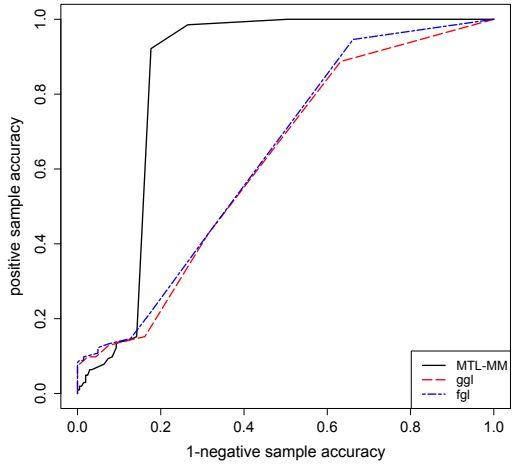

Figure 13. Comparison of anomaly detection performance on the `Anuran Calls` data. The test samples are aggregated over the $S = 3$ tasks.

of numerical instability in mixture weight learning. We also demonstrated better performance in anomaly detection thanks to the capability of handling multi-modal multi-task learning.

Acknowledgment

The authors would like to thank Dr. Minhazul Islam Sk, who contributed a lot to an initial stage of the project. We also thank Centre for Multilevel Modelling, University of Bristol, for its permission of using London School data.

References

[1] R. Caruana, "Multitask learning," *Machine Learning*, vol. 28, no. 1, pp. 41–75, 1997.

[2] O. Banerjee, L. E. Ghaoui, and G. Natsoulis, "Convex optimization techniques for fitting sparse Gaussian graphical models," in *Proc. Intl. Conf. Machine Learning*. Press, 2006, pp. 89–96.

[3] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.

[4] T. Idé, A. C. Lozano, N. Abe, and Y. Liu, "Proximity-based anomaly detection using sparse structure learning," in *Proc. of 2009 SIAM International Conference on Data Mining (SDM 09)*, 2009, pp. 97–108.

[5] S. Hara and T. Washio, "Learning a common substructure of multiple graphical gaussian models," *Neural Networks*, vol. 38, pp. 23–38, 2013.

[6] G. Varoquaux, A. Gramfort, J.-B. Poline, and B. Thirion, "Brain covariance selection: better individual functional connectivity models using population prior," in *Advances in Neural Information Processing Systems*, 2010, pp. 2334–2342.

[7] J. Honorio and D. Samaras, "Multi-task learning of gaussian graphical models," in *Proceedings of the 27th International Conference on Machine Learning*, ser. ICML 2010, 2010, pp. 447–454.

[8] J. Chiquet, Y. Grandvalet, and C. Ambroise, "Inferring multiple graphical structures," *Statistics and Computing*, vol. 21, no. 4, pp. 537–553, 2011.

[9] P. Danaher, P. Wang, and D. M. Witten, "The joint graphical lasso for inverse covariance estimation across multiple classes," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 76, no. 2, pp. 373–397, 2014.

[10] C. Gao, Y. Zhu, X. Shen, W. Pan *et al.*, "Estimation of multiple networks in gaussian mixture models," *Electronic Journal of Statistics*, vol. 10, no. 1, pp. 1133–1154, 2016.

[11] C. Peterson, F. C. Stingo, and M. Vannucci, "Bayesian inference of multiple gaussian graphical models," *Journal of the American Statistical Association*, vol. 110, no. 509, pp. 159–174, 2015.

[12] D. J. MacKay, *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[13] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer-Verlag, 2006.

[14] A. Corduneanu and C. M. Bishop, "Variational bayesian model selection for mixture distributions," in *Artificial intelligence and Statistics*, vol. 2001. Morgan Kaufmann Waltham, MA, 2001, pp. 27–34.

[15] K. Yamanishi, J. Takeuchi, G. Williams, and P. Milne, "On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms," in *Proc. the Sixth ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, 2000, pp. 320–324.

[16] D. M. Tax and R. P. Duin, "Support vector data description," *Machine Learning*, vol. 54, no. 1, pp. 45–66, 2004.

[17] D. T. Phan and T. Idé, "Demystifying automatic relevance determination in probabilistic mixture models." (submitted).

[18] M. Bahadori, Y. Liu, and D. Zhang, "Learning with minimum supervision: A general framework for transductive transfer learning," in *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, 2011, pp. 61–70.

[19] X. He, G. Mourot, D. Maquin, J. Ragot, P. Beauseroy, A. Smolarz, and E. Grall-Maës, "Multi-task learning with one-class svm," *Neurocomput.*, vol. 133, pp. 416–426, Jun. 2014.

[20] Y. Xiao, B. Liu, S. Y. Philip, and Z. Hao, "A robust one-class transfer learning method with uncertain data," *Knowledge and Information Systems*, vol. 44, no. 2, pp. 407–438, 2015.

[21] W. Zhang and P. Fung, "Sparse inverse covariance matrices for low resource speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 3, pp. 659–668, March 2013.

[22] H. Goldstein, "Multilevel modelling of survey data," *Journal of the Royal Statistical Society. Series D (The Statistician)*, vol. 40, no. 2, pp. 235–244, 1991.

[23] J. G. Colonna, M. Cristo, M. Salvatierra, and E. F. Nakamura, "An incremental technique for real-time bioacoustic signal segmentation," *Expert Systems with Applications*, vol. 42, no. 21, pp. 7367–7374, 2015.