Vol.2017-CSEC-77 No.11
Vol.2017-IOT-37 No.11
2017/5/26

スパース構造学習によるサーバの異常検知

小泉 成司[1,a)]　鮫島 正樹[1]　菅野 裕介[1]　松下 康之[1]

概要：
本稿では，データセンタにおけるサーバのリソース使用量を自動で収集し，サーバの異常を検知することを目的とする．従来の異常検知手法として主成分分析や確率的主成分分析を用いた手法が提案されているが，データセンタにおけるサーバのリソース使用量には相関がみられることから，相関を利用することで異常検知の精度を改善する方法を提案する．提案手法では，リソース使用量の相関を条件付分布で表現し，観測済みのリソース使用量を用いて学習する．弱い相関を学習することによって異常が未検出となる悪影響を防止するため，条件付分布の学習にはスパース構造学習を用いる．評価実験では，リソース使用量として CPU 使用率とパケット量を対象とし，疑似的に発生させた異常を検知した．スパース構造学習を用いない従来手法と比較し，スパース構造学習を用いた提案手法によって，異常検知の精度が改善していることを確認した．

Server Anomaly Detection by Sparse Structure Learning

Koizumi Seiji[1,a)]　Masaki Samejima[1]　Yusuke Sugano[1]　Yasuyuki Matsushita[1]

Abstract: This paper addresses automatic anomaly detection on servers in a data center based on resource usages. On top of conventional methods based on Principal Component Analysis (PCA) or Probabilistic Principal Component Analysis (PPCA), we exploit correlation between resource usages of servers for improving the anomaly detection. The correlation between resource usages is modelled as a conditional probability distribution, and parameters of the distribution are learnt with observed resource usages. We use sparse structure learning to disregard weak correlation that leaves anomalies undetected. In the evaluation experiment, we apply the proposed method to CPU usage and packet amount data that include simulated anomalies. The proposed method based on sparse structure learning can detect anomalies more accurately than the conventional methods.

1. はじめに

情報サービスの多くがインターネットを介したクラウドサービスとして提供される昨今において，クラウドサービスを支えるデータセンタに障害が発生すると，多数のユーザが不利益を被るおそれがある．データセンタの障害の予兆として，サーバのリソース使用量の異常を検知し，障害を未然に防止する試みがなされている [1]．例えば，ソフトウェアに性能上のバグがあれば，CPU 使用率やメモリの使用量が増大することがあり，これらは障害の予兆として検知される．しかし，大規模なデータセンタにおいて，リソース使用量を人手で監視し，異常を検知することは大きな負担である．本研究では，データセンタにおけるサーバのリソース使用量を収集し，自動で異常を検知することを目的とする．

一般的に，異常が発生する頻度は極めて低く，異常に関する事前知識なしに異常を検知する方法が求められている [2]．そこで，観測済みのデータが正常であると仮定して，正常データが従う分布を推定し，新たに観測されたデータの異常を検知することが行われている [3]．単純な方法として，各データが正規分布に従うと仮定して，観測データに対する尤度を求め，尤度が低い場合，異常として検知する方法がある．また，データセンタのように複数の

[1] 大阪大学大学院情報科学研究科
Graduate School of Information Science and Technology, Osaka University
[a)] koizumi.seiji@ist.osaka-u.ac.jp

サーバが存在する場合，複数のデータ間の関係を考慮することにより，異常検知の精度向上を期待できる [4], [5]．主成分分析を用いた方法では，観測済みのデータに対する次元削減によって正常部分空間を求め，再構成したデータと観測データの誤差 (再構成誤差) が大きい場合に異常を検知する [6], [7]．また，観測データごとに異なる分布を仮定する確率的主成分分析を用いることで，異常検知の精度が改善するとの報告がある [3], [8]．しかし，リソース使用量に生じる異常が小さい場合，主成分分析等によって計算される再構成誤差に有意な差が表れず，異常を検知することが困難である．

本稿では，従来の確率的主成分分析を用いた上で，データセンタにおけるサーバのリソース使用量の間に相関があること [9] に着目し，異常検知の精度を向上させる．確率的主成分分析によって得られる再構成誤差にも相関が表れることから，再構成誤差間の相関を条件付分布として表現し，観測済みの再構成誤差から条件付分布のパラメータの最尤値を推定する．有意な相関のみ推定するため，スパース構造学習 [10] を用いて条件付分布のパラメータを推定し，観測データに対する異常検知を行う．

2. データセンタにおけるサーバの異常検知

2.1 異常検知の概要

本稿で対象とするデータセンタにおけるクラウドサービス運用のモデルを図 1 に示す． ユーザからサービスに対するリクエストを受けると，サービスを提供するサーバが処理を開始する．サービスは複数のサーバを介して処理されることがある．例えば，ユーザのリクエストを受け付ける Web サーバ，サービスに必要なデータを保管する DB サーバ，サービスの応答を効率化するためのキャッシュサーバなどがある．

これらのサーバの CPU 使用率，メモリ使用量，ディスクの I/O，ネットワーク使用量などのリソース使用量における異常は，障害の予兆を示す可能性がある．障害の種類

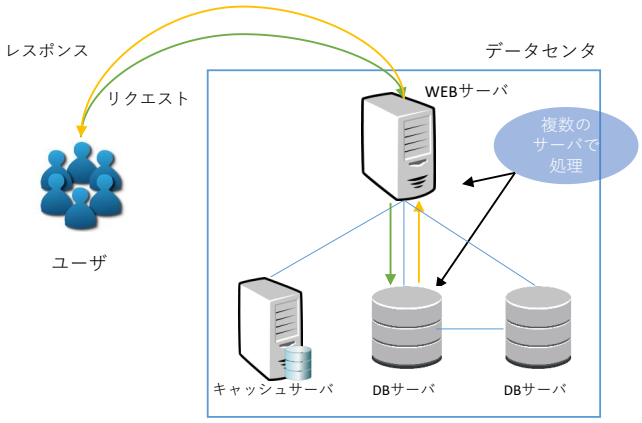


図 1 クラウドサービス運用のモデル．ユーザがリクエストを送ると複数のサーバで処理される．

に応じて異常は異なるが，障害の初期の予兆を示す Point Anomaly の異常を検知することを本研究の目的とする．Point Anomaly の異常検知の概要を図 2 に示す．

図 2 に示すように，ある時刻 $t \in \mathbb{N}$ において，$N \in \mathbb{N}$ 個のサーバの異常を検知するために，異常検知前の期間 $[t-1, t-M]$ に観測済みのリソース使用量を利用する．観測済みのリソース使用量 $\mathbf{D} \in \mathbb{R}^{M \times N}$ を以下のように表す．

$$\mathbf{D} = (\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots, \mathbf{x}^{(N)}) \tag{1}$$

$\mathbf{x}^{(i)} \in \mathbb{R}^M$ はサーバ i の観測済みのリソース使用量である．この観測済みのデータを用いて，新たな観測データ $\mathbf{x}' = (x'^{(1)}, x'^{(2)}, \cdots, x'^{(N)}) \in \mathbb{R}^N$ に対する異常検知を行う．異常検知は，各観測データ $x'^{(i)}$ が正常か異常かを判定する 2 値分類問題として定式化され，異常検知器を $f : \mathbb{R}^N \to [0,1]^N$, 異常検知器による検知結果を $\hat{\mathbf{y}} \in [0,1]^N$, 真の検知結果を $\mathbf{y} \in [0,1]^N$ とすると

$$\text{Max.} \quad \frac{1}{\|\mathbf{y}\|_1} \mathbf{y}^\top \hat{\mathbf{y}} \tag{2}$$

$$\text{Min.} \quad \frac{1}{\|\mathbf{1} - \mathbf{y}\|_1} \mathbf{y}^\top (\mathbf{y} - \hat{\mathbf{y}}) \tag{3}$$

$$\text{s. t.} \quad \hat{\mathbf{y}} = f(\mathbf{x})$$

となるような異常検知器を決定することが目的となる．検知結果 $y^{(i)} = 1$ のときサーバ i は異常であることを示し，$y^{(i)} = 0$ のときサーバ i は正常であることを示す．式 (2) は True Positive Rate (TPR) とよばれ，異常なリソース使用量に対して，正しく異常と判定した割合を表す．式 (3) は False Positive Rate (FPR) とよばれ，異常なリソース使用量に対して，誤って正常と判定した割合を表す．確からしい異常のみを検知すると TPR を最大化できるが，正常との判別が難しい異常は正常と検知されるため FPR も大きくなる．すなわち，TPR の最大化と FPR の最小化を表す 2 つの目的関数はトレードオフの関係にあり，双方において優れた異常検知器を決定する必要がある．サーバのリソース使用量に異常が発生する確率はきわめて低いため [2], [11]，異常時のリソース使用量を収集して教師あり学習を行うのではなく，教師なし学習によって異常検知器 f を決定することが求められる．

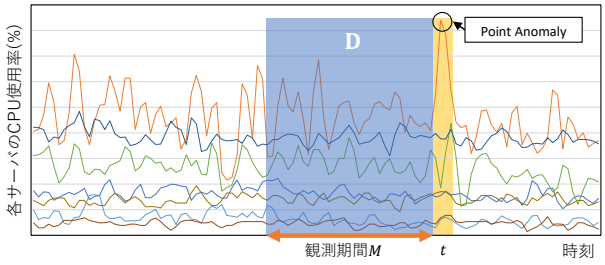


図 2 異常検知の概要．事前観測データ $\mathbf{D}$ を用いて異常を検知する．

Vol.2017-CSEC-77 No.11
Vol.2017-IOT-37 No.11
2017/5/26

3. 関連研究

従来手法として，主成分分析 (Principal Component Analysis, PCA) と確率的主成分分析 (Probabilistic PCA) を用いた異常検知手法 [3], [6], [12] について以下で述べる．

主成分分析とは，主成分空間と呼ばれる低次元の線形空間上へデータ点を直交射影するものであり，この直交射影は射影されたデータの分散が最大化されるように定める [13]．主成分分析は，観測済みのデータ $\mathbf{D}$ に対する特異値分解 $\mathbf{D} = \mathbf{USV}^\top$ において，$\mathbf{V}$ の列ベクトルを求めることと一致する．主成分分析の基底を m ($\leq N$) 個選択するときは，$\mathbf{\Sigma}$ の固有ベクトルから固有値の大きい順に m 個選択する．

主成分分析で選択した m 個の正規直交基底 $\mathbf{U}_m = (\mathbf{u}_1, \mathbf{u}_2 \cdots, \mathbf{u}_m) \in \mathbb{R}^{m \times N}$ を用いて，新たに観測したリソース使用量 $\mathbf{x}'$ に対する異常を検知するとき，異常なリソース使用量は $\mathbf{U}_m$ からなる正常部分空間から乖離していると考えられる [14]．乖離している度合いを表す尺度として，正常部分空間において $\mathbf{x}'$ を再構成し，再構成したリソース使用量 $\hat{\mathbf{x}}'$ と $\mathbf{x}'$ の再構成誤差を求める．再構成誤差が大きいほど乖離が大きいとみなし，異常なリソース使用量として検知する．

再構成したリソース使用量 $\hat{\mathbf{x}}'$ は，係数ベクトル $\mathbf{c} \in \mathbb{R}^m$ を用いて，$\hat{\mathbf{x}}' = \mathbf{c}^\top \mathbf{U}_m$ として求められる．係数ベクトルは，再構成誤差が最小となるように決定する．

$$\mathbf{c} = \underset{\mathbf{c}}{\operatorname{argmin}} \|\mathbf{c}^\top \mathbf{U}_m - \mathbf{x}'\|_2^2 \tag{4}$$

サーバ i で観測したリソース使用量に対する再構成誤差を $e^{(i)}$ として，正常を仮定したリソース使用量 $\mathbf{x}^{(i)}$ に対する再構成誤差よりも大きければ異常と判定する．統計的な大小判定を行うため，正常を仮定したリソース使用量 $\mathbf{x}^{(i)}$ に対する再構成誤差が正規分布に従うと仮定する．図 3 に正規分布を仮定した再構成誤差による異常検知を示す．観測した再構成誤差 $e^{(i)}$ が，再構成誤差の正規分布の上側 α%に含まれていれば十分に大きいとみなして，サーバ i の異常として検知する．

確率的主成分分析とは主成分分析を確率的モデルに展開したものであり，m 次元の変数 $\mathbf{z}$ が潜在的に存在し，この $\mathbf{z}$ から観測量 $\mathbf{x}$ が次の確率分布により生成されると考える [3], [15]．

$$p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}|\mathbf{Wz} + \boldsymbol{\mu} + \boldsymbol{\varepsilon}, \mathbf{I}_N) \tag{5}$$

ここで，$\mathbf{W} \in \mathbb{R}^{N \times m}$ の列ベクトルは前節の正常部分空間の基底を表している．また，確率分布ならびにパラメータに関して以下の仮定を置く．

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I}_m) \tag{6}$$

$$p(\epsilon) = \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_N),\ \mathrm{Cov}(\mathbf{z}, \boldsymbol{\epsilon}) = \mathbf{0} \tag{7}$$

μ は観測量 $\mathbf{x}$ の平均値で，σ^2 は観測量 $\mathbf{x}$ が生成されるときの分散，$\mathbf{I_N}$ は N 次元の単位行列である．パラメータ $\mathbf{W}, \mu, \sigma$ は観測データ $\mathbf{D} = (\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots, \mathbf{x}^{(N)})$ に対する最尤推定によって求められる．複数回サンプリングした $\mathbf{z}$ を式 (5) に代入し $\mathbf{x}$ の値を得ることで，リソース使用量 $\hat{\mathbf{x}}$ の分布を推定できる．主成分分析の場合と同様に，推定した分布にもとづいて再構成誤差を求め，サーバ i の異常を検知する．

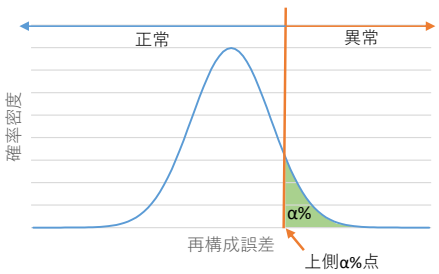


図 3 正規分布を仮定した再構成誤差による異常検知．観測した再構成誤差が再構成誤差の正規分布の上側 α%に含まれていれば十分に大きいとみなして異常とする．

4. スパース構造学習による異常検知

4.1 アプローチ

主成分分析や確率的主成分分析では，単一のサーバのリソース使用量のみに着目するため，あるサーバのリソース使用量がわずかに上昇するような異常を検知することは困難である．そこで，各サーバのリソース使用量の相関を用いて異常検知の精度を向上する．例えば，図 1 で示したデータセンタでは，WEB サーバの処理のために CPU 使用率が上昇するとともに，データベースサーバの CPU 使用率も上昇することがある．図 4 はデータセンタにおける実際の CPU 使用率の変動を示したものであるが，サーバ 1 の CPU 使用率 x_1 が上昇したときにサーバ 2 の CPU 使用率 x_2 も上昇していることが確認される．相関を考慮す

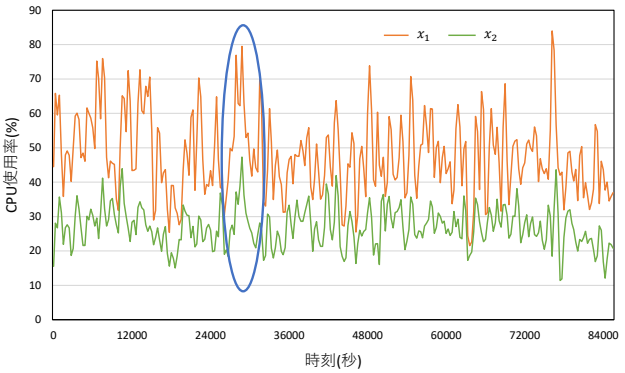


図 4 サーバの CPU 使用率の相関関係の例．x_1, x_2 はそれぞれサーバ 1，サーバ 2 の CPU 使用率を示す．

ることで，仮にサーバ 2 の CPU 使用率 x_2 が減少していれば，サーバ 1 の CPU 使用率 x_1 におけるわずかな上昇も異常とみなすことができる．

そこで従来技術に加えて，サーバのリソース使用量間の相関を利用した異常検知を行う．具体的には，新しい観測 $\mathbf{x}' = (x'_1, x'_2, \cdots, x'_N)^\top$ が与えられたとき，従来技術がリソース使用量 x'_i のみを利用してサーバ i の異常検知を行っていたのに対し，i 以外のサーバ $\mathbf{x}'_{-i} = (x'_1, \cdots, x'_{i-1}, x_{i+1}, \cdots, x_N)^\top$ を考慮した異常検知手法を提案する．

4.2 条件付分布を用いた異常検知の概要

サーバ i のリソース使用量 $x_i \in \mathbb{R}$ とサーバ i 以外のリソース使用量 $\mathbf{x}_{-i} \in \mathbb{R}^{N-1}$ の間の相関を，条件付分布 $p(x_i|\mathbf{x}_{-i}, \mathbf{D})$ でモデル化し，異常検知に利用する．条件付分布を用いた異常検知の概要を図 5 に示す．

まず，観測済みのリソース使用量 $\mathbf{D}$ を用いて条件付分布 $p(x_i|\mathbf{x}_{-i}, \mathbf{D})$ のパラメータの推定を行う．次に，求めた条件付分布にもとづいて，新たに観測したリソース使用量 x'_i と $\mathbf{x}'_{-i}$ から異常を検知する．異常検知の基準は，確率的主成分分析を用いた場合と同様に，条件付分布の上側確率を基準とする．以下では，条件付分布のパラメータ推定について述べる．

確率変数 $x_i, \mathbf{x}_{-i}$ に関する条件付分布は，その多変量正規分布 $p(\mathbf{x}|\mathbf{D})$ に対する分割公式によって求めることができる [16]．そこで，まず多変量正規分布 $p(\mathbf{x}|\mathbf{D})$ を求める方法について述べる．2.3 節で述べたように，確率的主成分分析を用いることによって潜在変数 $\mathbf{z}$ による確率分布 $p(\mathbf{x}|\mathbf{z})$ を求めることができる．さらに確率分布 $p(\mathbf{x}|\mathbf{z})$ から $\mathbf{x}$ をサンプリングし，多変量正規分布にフィッティングを行うことで $p(\mathbf{x}|\mathbf{D})$ を得る．フィッティングによって求めるべき多変量正規分布の平均 $\boldsymbol{\mu}$，共分散行列 $\boldsymbol{\Sigma}$ の最尤推定値は，$\mathbf{D}$ の標本平均と標本共分散に一致する．求めた平均 $\boldsymbol{\mu}$，共分散行列 $\boldsymbol{\Sigma}$ による多変量正規分布 $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ から，分割公式によって $x_i, \mathbf{x}_{-i}$ に関する条件付分布のパラメータを推定する．

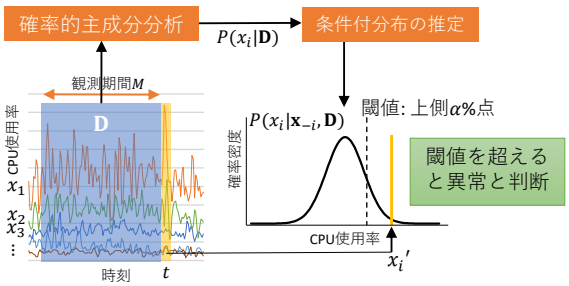


図 5 条件付分布を用いた異常検知の概要．確率的主成分分析を適用したのち，条件付分布を推定する．

確率変数 $\mathbf{x}$ が $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ に従うとして，$\mathbf{x} = [\mathbf{x}_i, \mathbf{x}_{-i}]^\top$ と分割するとき，これに対応して，$\mathbf{x}$ の平均 $\boldsymbol{\mu}$，共分散行列 $\boldsymbol{\Sigma}$，精度行列 $\boldsymbol{\Lambda}(= \boldsymbol{\Sigma}^{-1})$ を以下のように分割する．

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_i \\ \boldsymbol{\mu}_{-i} \end{pmatrix}, \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{i,i} & \boldsymbol{\Sigma}_{i,-i} \\ \boldsymbol{\Sigma}_{-i,i} & \boldsymbol{\Sigma}_{-i,-i} \end{pmatrix}, \boldsymbol{\Lambda} = \begin{pmatrix} \boldsymbol{\Lambda}_{i,i} & \boldsymbol{\Lambda}_{i,-i} \\ \boldsymbol{\Lambda}_{-i,i} & \boldsymbol{\Lambda}_{-i,-i} \end{pmatrix} \quad (8)$$

$\mathbf{x}_{-i}$ を与えたときの $\mathbf{x}_i$ の条件付き分布は正規分布となり，それを $\mathcal{N}(x_i|\boldsymbol{\mu}_{i|-i}, \Sigma_{i|-i})$ と書くと，平均と分散は以下の式で与えられる．

$$\begin{aligned} \boldsymbol{\mu}_{i|-i} &= \mu_i + \boldsymbol{\Sigma}_{i,-i}\boldsymbol{\Sigma}_{-i,-i}^{-1}(\mathbf{x}_{-i} - \boldsymbol{\mu}_{-i}) \\ &= \mu_i - \boldsymbol{\Lambda}_{i,i}^{-1}\boldsymbol{\Lambda}_{i,-i}(\mathbf{x}_{-i} - \boldsymbol{\mu}_{-i}) \quad (9) \\ \boldsymbol{\Sigma}_{i|-i} &= \boldsymbol{\Sigma}_{i,i} - \boldsymbol{\Sigma}_{i,-i}\boldsymbol{\Sigma}_{-i,-i}^{-1}\boldsymbol{\Sigma}_{-i,i} \\ &= \boldsymbol{\Lambda}_{i,i}^{-1} \quad (10) \end{aligned}$$

4.3 スパース構造学習

3.1 節の手法では，多変量正規分布のフィッティングにおいて，最尤推定値である標本平均と標本共分散を用いた．この手法では，各サーバのリソース使用量間の共分散が非常に小さい値も含めて密に定義され，あるサーバのリソース使用量が多くのサーバのリソース使用量に条件づけられることになる．このため，サーバのリソース使用量に異常が生じたとき，共分散が小さく相関が非常に弱い多数のサーバのリソース使用量によって，異常な値を正常であるとみなしてしまい，異常を検知することができない．そこで図 6 に示すように，スパース構造学習 [16], [17] を用いることで相関が非常に弱いサーバのリソース使用量については，対応する成分が 0 となるように精度行列を決定し，異常検知の精度を向上させる．

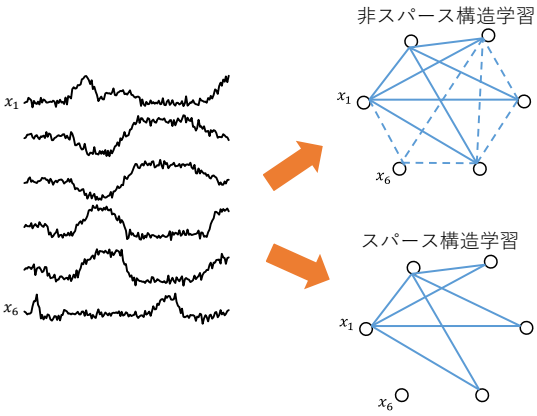


図 6 スパース構造学習を用いない例と用いる例．左図は各サーバのリソース使用量の時系列変化を，右図のノード間のエッジは相関が存在すると推定されたこと，点線のエッジは弱い相関を推定したことを表している．

以下では，スパース構造学習に関する説明を簡略化するため，従来研究と同様に多変量正規分布を以下のように表す．なお，$|\mathbf{\Lambda}|$ は $\mathbf{\Lambda}$ の行列式を表す．

$$\mathcal{N}(\mathbf{x}|\mathbf{0},\mathbf{\Lambda}) = \frac{|\mathbf{\Lambda}|^{1/2}}{(2\pi)^{2/N}}\exp\left(-\frac{1}{2}\mathbf{x}^\top\mathbf{\Lambda}\mathbf{x}\right) \tag{11}$$

この多変量正規分布においては，観測済みのリソース使用量から標本平均を引いておくことで平均 0 となるよう中心化し，精度行列 $\mathbf{\Lambda} = \mathbf{S}^{-1}$ を用いて表現している．$\mathbf{S}$ は中心化後の共分散行列である．この多変量正規分布の尤度 $J(\mathbf{\Lambda})$ は，対数をとって以下のように表現される．

$$\begin{aligned} J(\mathbf{\Lambda}) &= \frac{1}{2}\log|\mathbf{\Lambda}| - \frac{1}{2}\mathbf{x}^\top\mathbf{\Lambda}\mathbf{x} \\ &= \frac{1}{2}\log|\mathbf{\Lambda}| - \frac{1}{2}\mathrm{tr}(\mathbf{\Lambda}\mathbf{S}) \end{aligned} \tag{12}$$

スパース構造学習では，尤度を最大化しつつ，疎な精度行列 $\mathbf{\Lambda}$ を求めることが目的であり，以下の問題として定式化される [18]．

$$\mathbf{\Lambda}^* = \mathrm{argmax}\, J(\mathbf{\Lambda}) - \rho\|\mathbf{\Lambda}\|_1 \tag{13}$$

ρ は非負のパラメータであり，$\rho\|\mathbf{\Lambda}\|_1$ はスパース正則化項である．ρ が大きければ大きいほど，$\mathbf{\Lambda}$ はスパースになる．式 (13) は凸最適化問題であるため，その最適性条件は以下に示す通りである．

$$\frac{\partial J(\mathbf{\Lambda})}{\partial \mathbf{\Lambda}} = \mathbf{\Lambda}^{-1} - \mathbf{S} - \rho\mathrm{sign}(\mathbf{\Lambda}) = \mathbf{0} \tag{14}$$

ここで，行列 $\mathrm{sign}(\mathbf{\Lambda})$ は以下のように定義される．

$$\mathrm{sign}(\mathbf{\Lambda})_{i,j} = \begin{cases} 1 & (\mathbf{\Lambda}_{i,j} \geq 0) \\ -1 & (\mathbf{\Lambda}_{i,j} < 0) \end{cases} \tag{15}$$

最適性条件を示す式 (14) は解析的に解くことができないため，ブロック座標降下法を用いて解く．ブロック座標降下法とは，精度行列 $\mathbf{\Lambda}$ において x_i に関係しない要素を固定して，x_i に関係する要素について解析的に解き，このプロセスを i を変えながら反復的に解く方法である [16]．

5. 実験

5.1 実験の概要

クラウドコンピューティングのシミュレータ CloudSim[19], [20] を用いて生成した CPU 使用率と，Center for Applied Internet Data Analysis (CAIDA) が公開しているパケット量 [21] を対象とし，疑似的に発生させた異常を検知する実験を行った．CPU 使用率はサーバ 50 台に対して 5 分ごとに計 286 回観測されたものを，パケット量はサーバ 50 台に対して 1 秒ごとに計 300 回観測されたものを用いている．ランダムに時刻とサーバを選択し，Point Anomaly の異常を発生させ実験データとした．異常のモデルは，従来のサーベイ研究 [2] にもとづいて，あるサーバのリソース使用量に対して，そのリソース使用量の平均値の α 倍を加えて異常とするモデルを採用した．

異常検知のパラメータとして，観測済みのリソース使用量 $\mathbf{D}$ の観測期間 $M = 100$ とした．また，主成分分析に用いる主成分の数 K を変えながら実験を行い，異常検知の精度に対する K の影響も評価する．比較手法として以下を用いた．

- 主成分分析 (PCA) を用いた手法
- 確率的主成分分析 (PPCA) を用いた手法
- 条件付き分布を用いた手法
- スパース構造学習を用いた手法 ($\rho = 1$)

異常検知の精度を示す指標として，2.2 節で述べた TPR と FPR を用いる．TPR ならびに FPR は異常を検知する閾値に依存することから，閾値を変化させながら縦軸に TPR を，横軸に FPR をとった ROC (Receiver Operating Characteristic) 曲線を用い，ROC 曲線の下の面積 (Area Under the Curve, AUC) は異常検知システムの性能の良さとして評価する．異常を完全に分離できた場合に AUC が 1 になり，ランダムな分類の場合は 0.5 になる．

5.2 実験結果

ランダムに $\alpha = 1$ の異常を発生させた場合の実験結果について述べる．CPU 使用率を対象とした異常検知において，主成分数 $K = 5, 10$ を用いたときの ROC 曲線を図 7, 図 8 に，K を変化させたときの各手法による AUC の変化を図 9 に示す．また，パケット量を対象とした異常検知において，$K = 5, 10$ を用いたときの ROC 曲線を図 10，図 11 に，K を変化させたときの各手法による AUC の変化を図 12 に示す．

まず $K = 5$ の場合について考察すると，CPU 使用率とパケット量の双方において，主成分分析以外の手法による ROC 曲線は同等の精度を示していた．一方，$K = 10$ とすると，スパース構造学習による異常検知の精度は高いが，それ以外の手法については，$K = 5$ の場合と比べて精度が低下している．図 9 ならびに図 12 においても，スパース構造学習以外の手法では，K が増加するにつれて AUC が低下することが示されている．一方，スパース構造学習は $K \leq 30$ 程度までは AUC がほとんど変わらないが，K がさらに大きくなると AUC が大幅に低下する．従って，スパース構造学習を用いるだけではなく，確率的主成分分析によってあらかじめ次元削減を行うことで，異常検知の精度を向上できている．

CPU 使用率に対して主成分数 $K = 5, 15$ による確率的主成分分析を適用し，スパース構造学習によって得られた精度行列を図 13 に示す．青色で強調した要素には 0 以外の値が含まれている．より多くの主成分数を用いることでスパース構造が学習されている．主成分数が少ない場合，低次元な主成分空間に写像されることで，本来は相関の無

いデータも主成分空間において相関を示すことがある．そのため，主成分数が少ない場合は，再構成誤差のスパース構造は学習されない．主成分数を増加させるとスパースな精度行列を得ることができるが，主成分の個数がおよそ 30 個を超えると極端にスパースな精度行列となり，かえって異常検知の精度が悪化している．主成分の個数がおよそ 40 個に達すると，提案手法で学習される精度行列はほぼ対角行列となり，PPCA が示す精度に近づいていくことが確認される．図 14 は，図 13[2] の精度行列から，サーバ 1 からサーバ 14 までの要素を抽出した行列である．図 14 において，サーバ 4 とサーバ 10 の要素は非零であり，スパース構造学習によって相関があるとみなされている．図 15 に示すサーバ 4 とサーバ 10 の CPU 使用率の変化からは，サーバ 4 の CPU 使用率が上昇すると同時に，サーバ 10 の CPU 使用率も上昇しており，スパース構造学習によって相関が考慮されていることを確認した．

6. 結論

本稿では，データセンタにおけるサーバの異常を検知する問題を扱った．サーバのリソース使用量間に相関があることに着目し，条件付分布によって相関をモデル化し異常を検知する方法を提案した．条件付分布のパラメータを推

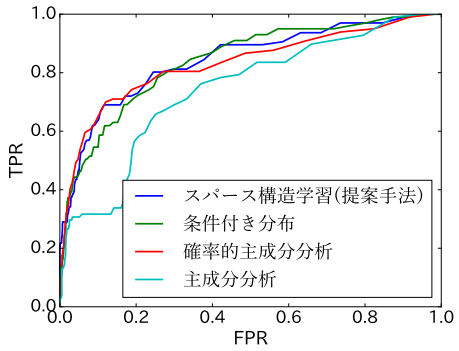


図 7 CPU 使用率にランダムに異常を発生させたときの ROC 曲線 ($K = 5$)

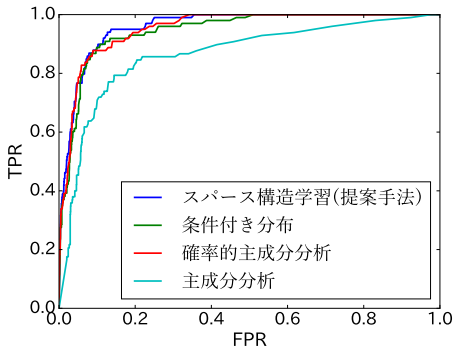


図 10 パケット量にランダムに異常を発生させたときの ROC 曲線 ($K = 5$)

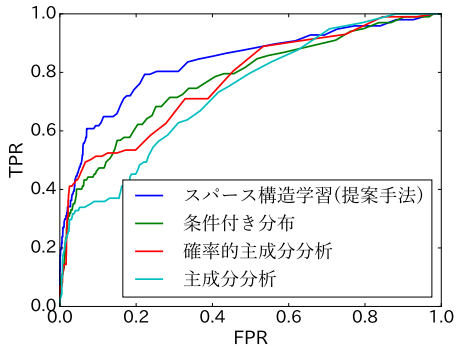


図 8 CPU 使用率にランダムに異常を発生させたときの ROC 曲線 ($K = 10$)

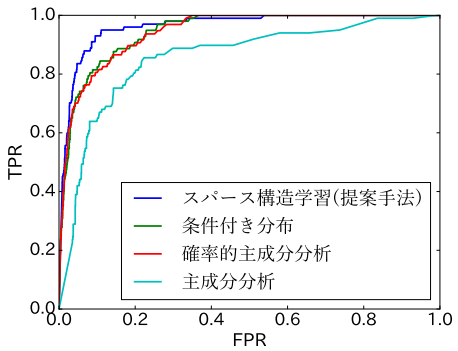


図 11 パケット量にランダムに異常を発生させたときの ROC 曲線 ($K = 10$)

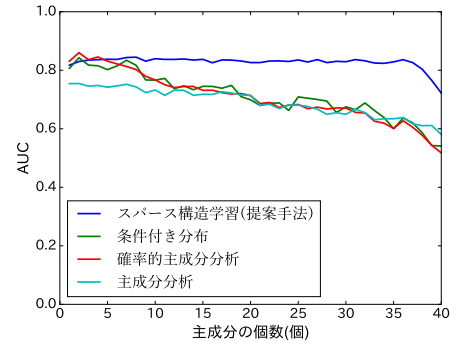


図 9 CPU 使用率にランダムに異常を発生させたときの AUC と K の関係

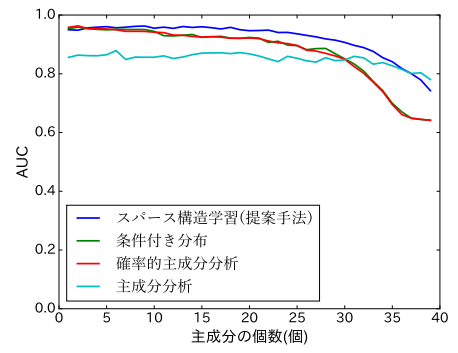
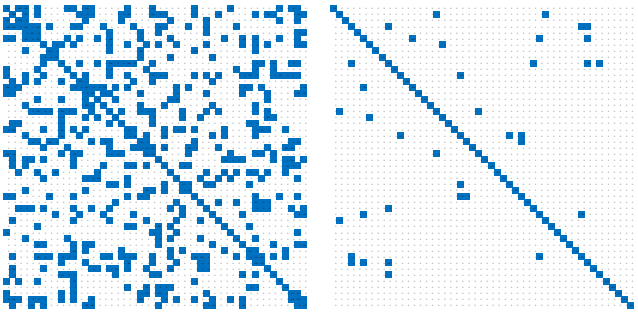


図 12 パケット量にランダムに異常を発生させたときの AUC と K の関係



[a]$K = 5$ [b]$K = 15$

図 13 [a] は主成分 K を 5 個用いたときの精度行列の一例，[b] は主成分 K を 15 個用いたときの精度行列の一例である．主成分を多く用いると精度行列はよりスパースになる．

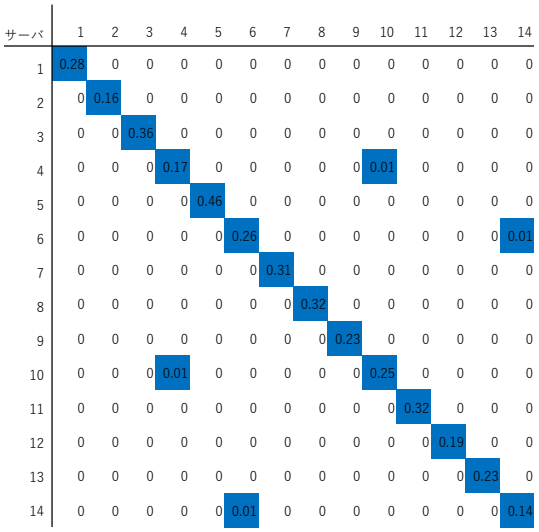


図 14 図 13[b] の精度行列の拡大図．対角成分以外で値が存在する要素に対応する 2 つのサーバは，相関が存在すると考えられる．

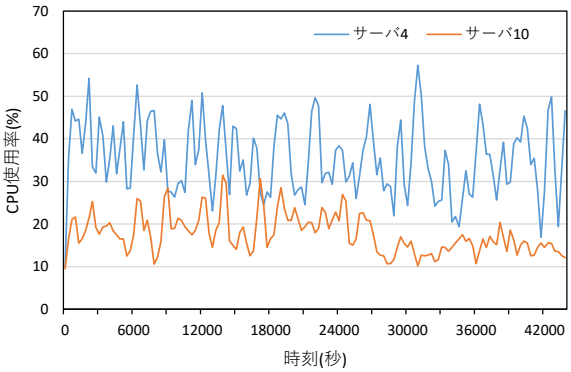


図 15 サーバ 4 とサーバ 10 の関係．2 つのサーバが同時に上昇している箇所があるため，相関が存在することがわかる．

定する方法として，確率的主成分分析で求めた多変量正規分布による最尤推定法を適用することが一般的に考えられる．しかし，最尤推定法では密な共分散行列が求まるため，リソース使用量における異常を弱い相関で表現し，異常が検知されないことがある．そこで，尤度にスパース正則化項を加えたスパース構造学習によってパラメータを推定した．

提案手法の有効性を評価するため，実際のアクセスデータにもとづく CPU 使用率とパケット量を対象として実験を行った．実験では，全サーバにランダムに異常を発生させ，スパース構造学習を用いていない手法と用いた手法との検知精度を比較した．実験の結果，スパース構造学習を用いていない手法と比較して，スパース構造学習を用いた手法によって，高い検知精度が得られることを確認した．

今後の課題として，今回対象とした Point Anomaly 以外の異常を検出することや，サーバ数に対するスケーラビリティを確保すること，時系列性を考慮することで異常検知の精度を改善することが挙げられる．

謝辞 本研究は JSPS 科研費 JP16K16047 の助成を受けたものです．

参考文献

[1] K. Adamova, D. Schatzmann, B. Plattner, and P. Smith. Network anomaly detection in the cloud: The challenges of virtual service migration. In *Proceedings of 2014 IEEE International Conference on Communications*, pp. 3770–3775, 2014.

[2] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, Vol. 41, No. 3, pp. 15–86, 2009.

[3] 井手剛. 入門 機械学習による異常検知 –R による実践ガイド –. コロナ社, 2015.

[4] H. Nguyen, Y. Tan, and X. Gu. PAL: Propagation-aware anomaly localization for cloud hosted distributed applications. In *ACM SOSP Workshiop on Managing Large-scale Systems via the Analysis of System Logs and the Application of Machine Learning Techniques*, pp. 1:1–1:8, 2011.

[5] 鮫島正樹. マルコフ転換モデルによるクラウドサービスのリソース使用量分析. 電気学会情報システム研究会資料, No. 12, pp. 7–11, 2016.

[6] H. Ringberg, J. Rexford, A. Soule, and C. Diot. Sensitivity of PCA for traffic anomaly detection. In *Proceedings of the 2007 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, Vol. 35, pp. 109–120, 2007.

[7] 乾稔, 矢入健久, 河原由伸, 町田和雄. 次元削減の再構成誤差を用いた異常検知手法の比較. 第 23 回全国人工知能学会大会, 2009.

[8] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2007.

[9] Y. Tan, H. Nguyen, Z. Shen, X. Gu, C. Venkatramani, and D. Rajan. Prepare: Predictive performance anomaly prevention for virtualized cloud systems. In *Proceedings of 2012 IEEE 32nd International Conference on Distributed Computing Systems*, pp. 285–294, 2012.

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

[11] O. Vallis, J. Hochenbaum, and A. Kejariwal. A novel technique for long-term anomaly detection in the cloud. In *6th USENIX Workshop on Hot Topics in Cloud Computing*, 2014.

[12] D. Brauckhoff, K. Salamatian, and M. May. Applying

PCA for traffic anomaly detection: Problems and solutions. In *Proceedings of IEEE Conference on Computer Communications*, pp. 2866–2870, 2009.

[13] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, Vol. 24, No. 6, pp. 417–441, 1933.

[14] 櫻田麻由, 矢入健久. オートエンコーダを用いた次元削減による宇宙機の異常検知. 第 28 回人工知能学会全国大会論文集, pp. 1–3, 2014.

[15] S. Roweis. EM algorithms for PCA and SPCA. pp. 626–632, 1998.

[16] 井手剛, 杉山将. 異常検知と変化検知. 講談社, 2015.

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

[18] 井手剛. スパース構造学習によるセンサーデータの変化点検出と異常解析. *Provision*, No. 65, pp. 71–76, 2010.

[19] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya. CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software - Practice & Experience*, Vol. 41, No. 1, pp. 23–50, 2011.

[20] S. K. Garg and R. Buyya. NetworkCloudSim: Modelling parallel applications in cloud simulations. In *Proceedings of the 2011 Fourth IEEE International Conference on Utility and Cloud Computing*, pp. 105–113, 2011.

[21] The CAIDA UCSD anonymized OC48 internet traces 2002-2003. `https://data.caida.org/datasets/oc48/oc48-original/` (2017.3.27 参照).