解説

Syslog分析による大規模ネットワーク故障診断技術
――ネットワーク機器ログデータの活用――

Analyzing Syslog Data for Diagnosing Large-scale Network Failures

木村達明

Abstract

様々なサービスが展開される大規模ネットワークでは従来以上の高度な故障対応業務が求められる．ネットワーク機器のログデータであるSyslogは機器の詳細な情報を含むため，ネットワーク運用上有用であるが，膨大かつ非定形なデータであるため十分に活用されていないのが現状である．本稿では，こうしたネットワークログデータから機械学習技術を用いて故障診断において有益な情報を抽出する技術を紹介する．

キーワード：ネットワーク監視，Syslog，ログデータ，機械学習

1. はじめに

近年の大規模IPネットワークでは，IPTVやVoIPなど様々なサービスが展開され，従来以上の高度なネットワークの監視・運用業務が求められている．サービスの多様化に伴いネットワーク上で発生するイベント間の関係は複雑化し，加えてレイヤ間の相互関係やトンネリング等の技術が，監視業務を更に困難化している．一般に大規模ネットワークではNetwork Management System（NMS）を利用しアラーム（警報）を契機とした故障対応する形態が取られる．アラームはSimple Network Management Protocol（SNMP）のトラップなど，事前設定されたルールに従って発生する．しかし，これらのNMSに事前定義された警報ルールはハードウェアの致命的な故障など，明らかに単体で異常と分かる事象が多く，サービスに対して一時的に影響を持つイベントや故障の予兆となるような未知の軽微な変化，そして今日のネットワークで発生するレイヤ・サービスをまたがる複雑な故障を捉えることができない．加えて，今後更に多様化，大規模化していくネットワークにおいては，オペレータの専門知識と経験に基づく従来の運用法ではスケールしないため，複雑なネットワークオペレーションを効率化する技術への要望が高まっている．

ネットワーク機器のログデータ，SyslogやNMSから通知されるアラームなどは，機器に関する詳細な情報を含み，ネットワークの故障対応を行う上での重要な情報である．しかし，サービスや機器の増大に伴うログの大規模化や，ベンダ依存の非定形なテキストメッセージであることから，十分には活用されていない．加えて，ネットワークで発生するイベント間の関係の複雑化のため，ログデータの中から，今何がネットワークで発生しているのかの把握すら難しく，ログの分析そのものが課題となっている．

本稿ではこれらの課題を受け，ネットワーク機器から一元的に収集された，大規模かつ非定形なネットワークログデータから，ネットワークの故障診断に有用な情報を自動的に抽出する技術について紹介する．

2. ログデータについて

Syslogなどのネットワーク機器のログデータは，故障そのものからセキュリティに関わるメッセージ，コンソールの履歴まで様々な情報を含む．表1にログデータの例を示す．一般に，ログデータは日時，ホスト識別子（IPアドレスなど），そしてメッセージ部分の三つのフィールドから成る．通常これらのログデータは各ネットワーク機器からNMSなどの収集装置に一元的に集められ，ネットワークオペレータはこれらの情報を必要に応じ閲覧する．表1から分かるように，ログメッセージのフォーマットはベンダやサービスによって異なり，定

木村達明　正員　日本電信電話株式会社NTTネットワーク基盤技術研究所
E-mail　kimura.tatsuaki@lab.ntt.co.jp
Tatsuaki KIMURA, Member (NTT Network Technology Laboratories, NIPPON TELEGRAPH AND TELEPHONE CORPORATION, Musashino-shi, 180-8585 Japan).
電子情報通信学会誌　Vol.98 No.9 pp.823-828 2015年9月

表1　ログメッセージの例　一般に日時，ホスト，メッセージをフィールドとして持ち，ベンダやサービスごとにフォーマットが異なる，非定形なテキストメッセージである．

日時	ホスト	メッセージ
2015/1/1T00:00:00	HOST_X	%TRACKING-5-STATE:1 interface Fa0/0 line-protocol Up->Down
2015/1/1T00:00:01	HOST_X	%LINK-3-UPDOWN:interface FastEthernet 0/9, changed state to down
2015/1/1T00:00:01	10.1.1.2	%SYS-5-CONFIG I:Configured from console by vty2（10.11.11.11）
2015/1/1T00:00:05	10.1.1.2	［エラータイプ100］リングダウンが起こりました．ホスト名：hostA IP アドレス：10.11.1.1
2015/1/1T00:00:05	HOST_Y	［エラータイプ100］パケットロスを検出しました．host:hostB, IP アドレス：10.11.11.11
2015/1/1T00:00:10	HOST_Y2	〈優先度：低〉オペレータがログインしました．from yyyy, host:hostC, 2011/11/11
2015/1/1T00:00:11	HOST_Y	100 login:LOGIN INFORMATION:User XXX logged in form host HOST X on device X
2015/1/1T00:00:30	10.1.1.3	SNMP trap:CPU utilization exceeds threshold（96.9%>90%）

まったルールを持たない短いテキストデータである．時に Syslog は開発ベンダのデバッグ目的に設定されたログであるため，ネットワーク監視者が扱いやすい形式になっていないのが特徴である．ただし，ログメッセージ内にも幾つか共通の情報があり，イベントの種類や部位を表す箇所（%LINK-3-UPDOWN や System），状態の変化を表す部分（down，up），そしてパラメータを示す部分（IP アドレスやホスト名，プロセス ID など）が存在することが分かる．

ログデータはネットワーク機器の詳細な状態を示すため，ネットワーク運用現場において広く用いられている．ネットワークオペレータは，NMS のアラームを契機として故障対応業務を開始するが，故障の原因特定のために該当機器の直近の Syslog の発生状況を調べる．また，ネットワーク機器の健康診断では，注意すべきログメッセージの発生頻度の変化を日々監視している．しかし，これらのログ分析の多くがネットワークオペレータの知識と経験に基づいた手動による運用であるため，監視の精度や効率化が課題となっている．

用　語　解　説

Log Tensor Factorization（LTF，ログテンソル因子分解）　LTF は文献（2）において提案された NTF の拡張版であり，ログテンプレート，発生ホスト，発生時刻の三軸を持つログデータの生起を 3 階のテンソルで表現し，同時に生起しやすいログのグループを時空間情報とともに抽出する手法である．

Nonnegative Matrix Factorization（NMF，非負値行列因子分解）　全ての要素が非負の実数である入力行列を，非負値制約条件下で二つの行列の積の形に分解し，高次元データの次元削減や低ランク近似を行う手法の一種である．画像認識，音声信号処理，文書解析などの幅広い応用を持つ．

Nonnegative Tensor Factorization（NTF，非負値テンソル因子分解）　NTF は NMF のテンソルへの拡張であり，非負の実数で表現された入力のテンソルを，非負値制約条件下，ランク 1 のテンソルの和へ分解する手法である．時空間などの三軸以上のデータの解析に応用される．

3. ログデータのテンプレート化技術

ログデータは非定形なテキストメッセージであるため，そのままの状態での分析とは困難である．表 1 に示されるとおり，ログメッセージにはインタフェース名や IP アドレスなどの多くのパラメータが記述されている．オペレータが以前同じ種類のログが出現したか調べるにも，同じパラメータを持つログは二度と発生していない可能性があり，検索にも正規表現を作成するなど工夫が必要となる．また，特定のログ発生量の統計値を得るにも同様の課題がある．

こうした背景から，ログデータに対する前処理として，パラメータ部分を除いたログメッセージの主要な部分を，テンプレートとして抽出する手法が文献（1）で提案されている．抽出されたログテンプレートの例を図 1 に示す．これらのログテンプレートは，ベンダのサポートページから手に入る場合もあるが，ソフトウェアの更新や設定変更に伴いログテンプレートは日々動的に変わる可能性があり，また，全てが公開されているとも限らない．そこで，本稿で紹介する手法はログデータに関する事前知識を必要とせずに，データから自動的にログテンプレートを抽出するものである．ログテンプレート抽出法の主要なアイデアは，テンプレートまたはパラメータへのなりやすさを考慮した単語の分類とそれを用いた類似度による逐次クラスタリングである．以下ではこれについて順に説明する．

（1）　単語の分類について

ログメッセージ内の単語の中でも，パラメータまたはテンプレートになりやすい単語には共通の性質がある．例えば，数字のみで構成された単語はプロセス ID やエラーカウンタなど，パラメータになりやすい．同様に，IP アドレスなどの数字記号混じりの単語もパラメータになる可能性が高い．一方で，down，config などのアルファベットのみで記述される単語はテンプレートを構成する可能性が高い．これらの考察に基づき，単語の性

%SYS-5-CONFIG : Configured from console by **vty2 (10.11.11.11)**
tty1 (10.0.0.1)
vty0 (192.168.0.2)
(10.1.0.2)

System : Interface **FastEthernet 0/9,** changed state to **down**
GigE 1/0/1, 2/1/1, 0/2, **up**

図1　ログテンプレートの例　似ているログメッセージのクラスタとして表現され，パラメータに該当する単語が同じ位置に重ねて示されている．

表2　単語分類の例　単語ごとにテンプレートまたはパラメータへのなりやすさにより分類される．

単語分類	例
1. 数字/記号	11, 10.1.1.2,
2. 数字＋アルファベット	Fa0/0, Ga1/0, L2TP
3. 単語（アルファベット）	interface, up, down

質に基づき，ログメッセージ内の単語を表2に従い分類する．更に，分類 i に対して重み w_i を定義し，これをログテンプレートになる傾向の重みとして設定する．

（2） ログメッセージのクラスタリング

本手法ではログテンプレートを，似た単語を持つログメッセージの一つのクラスタと考え，逐次到着するログメッセージを最も類似度の高いログテンプレートへ割り当てる方式を取る．この際，（1）で定義した単語分類ごとの重みを利用し，ログデータの性質を考慮した類似度に基づきクラスタリングを行う．今，到着したメッセージがあるとすると，これと過去に抽出したある一つのログテンプレート（クラスタ）との類似度は以下で定義される：

$$\boldsymbol{w}^{\mathrm{t}}\boldsymbol{x}/\boldsymbol{w}^{\mathrm{t}}\boldsymbol{c}_{\boldsymbol{x}}$$

ただし，$\boldsymbol{c}_{\boldsymbol{x}}$ の i 成分はログテンプレートに含まれる単語分類 i の個数を表し，$\boldsymbol{x}$ の第 i 成分は到着したログメッセージとログテンプレートの両方で一致して出現した単語分類 i の個数を表す．これにより，各単語のログテンプレートへのなりやすさを考慮した類似度を計算できる．更に，今までに抽出した全てのログテンプレートとの類似度が，あるしきい値以下である場合は，新しいログテンプレートが生成されたと考え，それ自身で新しいログテンプレートのクラスタを生成する．

4. ログデータからのイベント抽出技術

大規模ネットワーク上では日々様々なイベントが発生し，ログの扱いを困難化している．例えばルータが再起動した場合には，様々なプロセスの初期化メッセージが同時発生する．同様に，リンクフラップイベント（リンクの瞬断）はリンクアップメッセージとダウンメッセージが同時発生し，上位ルーチングレイヤのログまで波及する場合もある．更に，トポロジー上の隣接ホストに影響が及ぶ場合もある．このように，ネットワークイベントがレイヤ・サービス・トポロジーの関係により波及していくことで，複数のログメッセージが生成されている．ネットワークオペレータは知識と経験から，共起したログメッセージのグループに対して意味付けし，故障対応を行っている．

本章では，こうしたサービス・トポロジー・レイヤの関係により波及し，共起するログのグループを，大量のログデータから自動的に抽出する方法 Log Tensor Factorization（LTF）[用語],[2] を紹介する．これにより故障時の大量ログメッセージへの意味付けが容易となり，故障診断において有用である．また，抽出されたログの共起グループは新たな警報ルールとして用いることも可能である．

LTF では，複雑なログデータの生起を背後のネットワークイベントの重ね合わせと考えることでモデル化し，因子分解アプローチにより，共起するログのグループを抽出する．例えば，ある隣接ルータ間でリンクフラップイベントが発生しており，同時に，定期監視を目的とした定期監視のジョブが動いているとする．このとき，オペレータは，隣接するホストのリンクダウンメッセージとアップメッセージ及び上位レイヤへ影響が波及したログと同時に，定期監視時に発生する自動ログインのメッセージとコマンド実行メッセージのログを，イベントの重ね合わせとして観測する．この考察に基づき，LTF はネットワークログの生起を3階のテンソル（ログテンプレート，ホスト，タイムウィンドウ）とみなし，非負値テンソル因子分解アプローチをとることで，同時に生起しやすいログテンプレートのグループ及びホスト関係を抽出する．このアプローチは非負値行列因子分解（NMF : Nonnegative Matrix Factorization）[用語],[3]

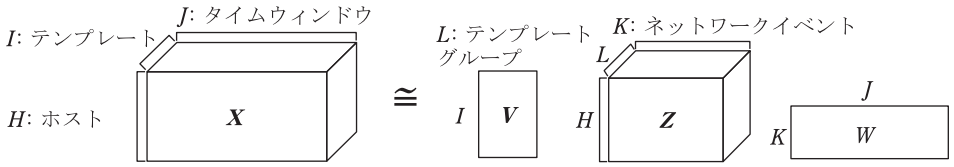

図2　ログテンソルとLTF分解イメージ

や非負値テンソル因子分解（NTF：Nonnegative Tensor Factorization）[用語],[4]と関連が深い．NMFやNTFは近年注目されている，非負制約下で混合要素の分解を行う機械学習技術であり，画像処理，音声信号処理，テキストマイニングなど様々な分野へ応用されている．以降ではまず，ログデータのテンソル表現を導入し，次にログデータから抽出するネットワークイベント情報の定義を行う．続いてLTFの詳細な説明を行い，最後に実データから抽出されたネットワークイベントの例を示す．

4.1　ログデータのテンソル表現

まず，観測されるログデータは適当なタイムウィンドウごとに分割されているものとし，一定期間で生成されたログデータを3階のテンソルで表現する．ここでは簡単のため，3階のテンソルをベクトル，及び行列の拡張と考えて構わない．ログテンプレート$i(i=1,2,\cdots,I)$，ホスト$h(h=1,2,\cdots,H)$，及びタイムウィンドウ$j(j=1,2,\cdots,J)$におけるログの発生回数を，x_{ihj}としたとき，ログテンソル$\boldsymbol{X}$は$I\times H\times J$の3階テンソルで，その(i,h,j)成分はx_{ihj}で表現される．ログテンソルのイメージを図2に示す．

4.2　ネットワークイベントの定義

続いて，LTFにおいて抽出するネットワークイベント情報を定義する．ログデータを観察して得られる知見としてまず，ログテンプレートの中には，ルータ再起動時のログのように，同時に発生しやすいログのグループが存在している．これらは機器の個々の状態を表していると考えられる．一方で，ログの背後で発生しているネットワークイベントは，これらのログテンプレートのグループをトポロジーやサービス上で関係のある複数のホストへ波及させる可能性がある．そこで，本研究ではこれらの空間的影響を考慮した相関のあるログメッセージ群を「ネットワークイベント」と考える．以下ではまず，同時生起しやすいテンプレートのグループ，「テンプレートグループ」を定義した後に，ネットワークイベントの定義を行う．以降，全てのログメッセージはログテンプレートにひも付いているとし，タイムスタンプ，ホスト名も把握されているとする．

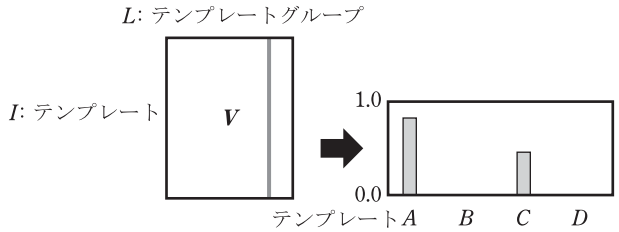

図3　テンプレートグループのイメージ　　本例では，テンプレートA及びCが同じテンプレートグループに属する

（1）　テンプレートグループ

テンプレートグループ$l(l=1,2,\cdots,L)$はある1ホストにおける同時生起しやすいテンプレートの組として定義する．テンプレートグループが表すのは，個々のホスト内で発生した事象であり，例としてルータの再起動や，リンクフラップ，コンフィグ変更に伴うログメッセージ等が挙げられる．

（2）　ネットワークイベント

ネットワークイベント$k(k=1,2,\cdots,K)$はホストとテンプレートグループの組のリスト，すなわち，(h_1,l_1)，(h_2,l_2)，…の形で表現する．ネットワークイベントはテンプレートグループの空間的拡張であると考えることができる．リンクフラップの例では，リンクフラップがあるホストで発生すれば，同様のログのグループが隣接ホストで発生する．ネットワークイベントはこのようなホスト間の関係性と，テンプレートグループの組合せにより表現される．

4.3　Log Tensor Factorization（LTF）

続いて，ログテンソルからネットワークイベント情報を抽出する手法LTFについて説明する．まず，LTFの問題設定から述べる．

［問題］　$I\times H\times J$の3階テンソル$\boldsymbol{X}$及び整数K，Lが与えられた下で，ログテンソル因子分解問題とは，以下の因子分解を満たす$v_l, z_{lk}, w_k(l=1,\cdots,L, k=1,\cdots,K)$を得ることである．

$$\boldsymbol{X}\simeq\sum_{k=1}^{K}[\sum_{l=1}^{L}\boldsymbol{v}_l\otimes\boldsymbol{z}_{lk}]\otimes\boldsymbol{w}_k.$$

ただし，v_l, z_{lk}, w_k はそれぞれ I，H，J 次元の非負1階テンソルである．

簡単のため，$I \times L$ 行列 $\boldsymbol{V}=[v_{il}]$，$L \times K \times H$ の3階テンソル $\boldsymbol{Z}=[z_{lkh}]$，$K \times J$ 行列 $\boldsymbol{W}=[w_{kj}]$ を定義する．LTF による分解のイメージを図2へ示す．

LTF の出力である $\boldsymbol{V}$，$\boldsymbol{Z}$，$\boldsymbol{W}$ はそれぞれ直感的な解釈が可能である．まず，行列 $\boldsymbol{V}$ はテンプレートグループ行列と捉えることができる．行列 $\boldsymbol{V}$ のイメージを図3へ示す．テンプレート i がテンプレートグループ l へ属するとき，v_{il} に非負の値が格納され，それ以外は0となる．すなわち，ベクトル $\boldsymbol{v}_l=(v_0, \cdots, v_I)$ は l 番目のテンプレートグループに対応しており，行列 $\boldsymbol{V}$ のサイズ L はテンプレートグループの総数を表している．

続いて，3階テンソル $\boldsymbol{Z}$ についてはネットワークイベントを表現するテンソルと捉えることができる．図4にテンソル $\boldsymbol{Z}$ のイメージを示す．k 番目のスライスである $H \times L$ 行列 $\boldsymbol{z}_k$ に注目すると，テンプレートグループ l のホスト h での生起が k 番目のネットワークイベントに属していれば，z_{lkh} に非負の値が格納され，そうでなければ0となることを表している．すなわち，スライス z_k は k 番目のネットワークイベントに対応している．

最後に，行列 $\boldsymbol{W}$ は重みを表す行列と考えることができる．各要素 w_{kj} はタイムウィンドウ j における k 番目のネットワークイベントの生起の重みを合しており，$\boldsymbol{w}_k$ により k 番目のネットワークイベントがどのタイムウィンドウで生起したかを把握することができる．

また，実際に入力のログデータから出力のテンプレートグループやネットワークイベントを得るためには，LTF は以下の最適化問題に帰着することができ，これを解くことで結果が得られる．

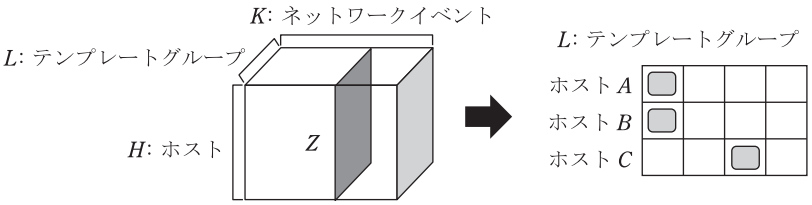

図4　ネットワークイベントのイメージ　本例では，ホスト A，B 及びホスト C で発生したテンプレートグループが一つのイベントに属する

ホスト	テンプレートグループの重み	テンプレートの重み	テンプレート
コアルータ	0.0253	0.4375	SNMP Trap: a status change for a module. Software image for the module is missing or invalid....
		0.0833	os: loader: * for * is *
		0.0833	id of requester is *
		0.0833	OsCrashDump: invalid crash record skipped, ...
		⋮	⋮
エッジルータA	0.01656	0.92805	Tunneling Virtual Path * is disconnected, hardware unavailable.
エッジルータA	0.01594	0.98308	Tunneling Virtual Path * is disconnected, hardware unavailable.
⋮	⋮	⋮	⋮

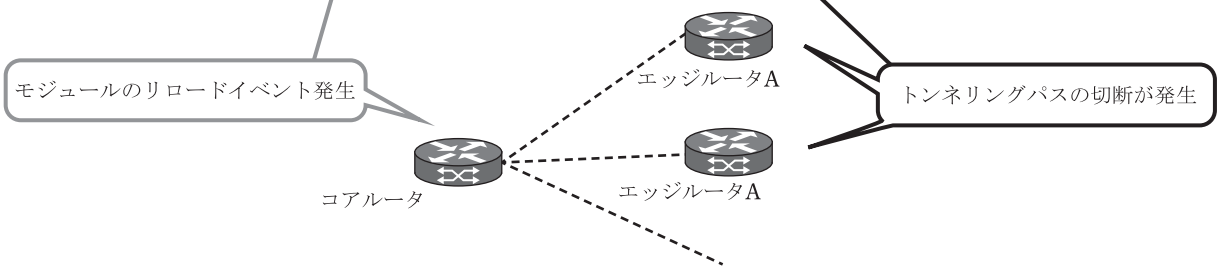

図5　ネットワークイベントの例　各ホストで発生したテンプレートグループとその発生率の重み及び，テンプレートグループに属するテンプレートとその発生率の重みが示されている．

$$\min_{\boldsymbol{V},\boldsymbol{Z},\boldsymbol{W}} \mathcal{D}(\boldsymbol{X}\|\boldsymbol{V},\boldsymbol{Z},\boldsymbol{W}),$$
$$\text{s.t.} \quad \boldsymbol{V},\boldsymbol{Z},\boldsymbol{W} \geq \boldsymbol{O}, \sum_i v_{il}=1, \sum_{l,h} z_{lkh}=1.$$

ただし，$\mathcal{D}(\cdot)$はカルバックライブラ（KL：Kullback-Leibler）情報量を表す．すなわち，

$$\mathcal{D}(\boldsymbol{X}\|\boldsymbol{V},\boldsymbol{Z},\boldsymbol{W})$$
$$\sum_{i,h,j} x_{ihj} \log \frac{x_{ihj}}{\sum_{k,l} v_{il} z_{lkh} w_{kj}} - x_{ihj} + \sum_{k,l} v_{il} z_{lkh} w_{kj},$$

である．LTF は NMF や NTF の拡張であり，これらは multiplicative 更新式と呼ばれるシンプルな更新アルゴリズムが知られている．これらの導出と同様に，LTF は上記の最適化問題の解を得るための更新アルゴリズムを導出することができる．詳細は文献(2)及び(4)を参照されたい．

4.4 LTF の出力結果例

続いて，LTF において実際に抽出されたネットワークイベントの例を紹介する．実際に数百のホストが稼動するあるネットワークにおける 1 週間分の Syslog データを用いた実験を行った．図 5 に実際に抽出されたログメッセージのグループを示す．それぞれ，発生したホスト名と該当するテンプレートグループの内容を表内に示した．テンプレートグループの重みでは，テンプレートグループの生起重みである z_{lkh} に対応する値を，テンプレートの重みでは，テンプレートの重みである v_{li} に対応する重みの値を記載した．図のとおり，この例ではトンネリング関係にあるホストにおけるネットワークイベントを示している．まずコアルータにおけるモジュールのリロードが発生している．このリロードイベントは多くのテンプレートを発生させたため，該当するテンプレートグループ内での各テンプレートの重みが低くなっていることが分かる．また，モジュールリロードイベントは一時的なホストの停止を示し，これに伴い複数のエッジルータ A で仮想パスの切断を誘発している．なお，仮想パスを生成していたホストは多数あったため，テンプレートグループの重みがそれぞれ低くなっている．本ネットワークイベントの持続時間はおよそ 5 分程度であったが，その間に仮想パスを持つそれぞれのエッジルータ A では周期的な定期監視ジョブによるログが大量に発生しており，目視ではこれらのモジュールリロードと仮想パス切断の関係性を把握することは困難であった．このように，LTF を用いることで，大量のログメッセージからオペレーション上意味のあるログのグループを，ネットワークの空間関係を考慮した上で抽出することができた．

5. むすび

本稿では，サービスや機器構成が多様化した大規模ネットワークのログデータを分析する手法について述べた．紹介した手法により，大量に発生するログデータに対しての意味付けが容易となり，故障診断や原因解析時に有用である．今後の展望として，故障情報やネットワークトポロジーの情報といった外部情報を取り込むことで新たな知見の発見や精度向上が考えられる．

近年，計算機の高速化や機械学習などの様々な分析ツールの登場により，従来は人手で監視しきれなかったログデータの分析に注目が集まっている．日々無限にデータが生成される現在の大規模ネットワークに，今回紹介したようなデータ分析技術が広く開発・適用されることで，より高品質・高信頼なネットワークサービスが拡大していくことを願う．

文　献

（1） 木村達明，渡邉　暁，豊野　剛，西松　研，石橋圭介，塩本公平，“大規模ネットワークログ情報のオンライン・テンプレート抽出法，”信学ソ大，no. B-7-30, Sept. 2014.

（2） T. Kimura, K. Ishibashi, T. Mori, H. Sawada, T. Toyono, K. Nishimatsu, A. Watanabe, A. Shimoda, and K. Shiomoto, “Spatio-temporal factorization of log data for understanding network events,” Proc. INFOCOM, pp. 610-618, Tronto, Canada, April 2014.

（3） D.D. Lee and H.S. Seung, “Learning the parts of objects by non-negative matrix factorization,” Nature, vol. 401, no. 6755, pp. 788-791, 1999.

（4） A. Cichocki, R. Zdunek, A.H. Phan, and S.I. Amari, Nonnegative Matrix and Tensor Factorizations : Applications to Exploratory Multi-way Data Analysis and Blind Source Separation, John Wiley & Sons, 2009.

（平成 27 年 3 月 31 日受付　平成 27 年 4 月 13 日最終受付）

木村　達明（きむら　たつあき）（正員）

平 20 京大・工・情報卒．平 22 同大学院修士課程了．同年日本電信電話株式会社入社．以来，NTT ネットワーク基盤技術研究所にて，データ分析によるネットワーク運用高度化の研究に従事．2010 年度日本 OR 学会学生論文賞受賞．