2004 年情報論的学習理論ワークショップ
2004 Workshop on Information-Based Induction Sciences (IBIS2004)
Tokyo, Japan, November 8 - 10, 2004.

Web系システムからの特徴抽出とオンライン障害検知手法

Feature Extraction and Anomaly Detection in Web-based Computer Systems

井手剛* 鹿島久嗣†
Tsuyoshi Idé Hisashi Kashima

Abstract: Although several network management systems are available in the market, none of them have sufficient capabilities to detect faults in multi-tier Web-based systems with redundancy. By modeling a Web-based system as a dynamic weighted graph, where each node represents a "service" and each edge represents a dependency between services, we propose a novel anomaly detection method in Web-based computer systems.

1 研究の動機

コンピュータシステムが社会基盤の一部として定着するにつれ、その障害がもたらす社会的損失も大きなものになっている。最近の分散システムでは、人間の管理者が常駐してシステムを監視するというモデルはもはやそぐわず、何らかのオートノミックな（自律型の）管理の仕組みが必要とされている。

障害の検知を目的に、エンタープライズシステムでは、ネットワークノード管理システム（Network Node Management System; NNMS）がしばしば使われている。それらのシステムでは、各 IP（Internet Protcol）アドレスに常駐するエージェントから、SNMP（Simple Network Management Protocol）と呼ばれるプロトコルを通して、低レベルから高レベルのさまざまな情報を集められるようになっている。各エージェントは、観測値に何らかの例外が生じた場合、SNMP トラップと呼ばれる信号を発することができる。しかしコンピュータシステムの常として、局所的な観測値の変動は大きく、それが実際の障害とは結びつかないことはままある。

この限界は、とりわけ大量のトランザクションを扱う Web 系システムでは本質的問題になりえる。例として、HTTP (Hyper Text Transfer Protocol) サーバー、Web アプリケーションサーバー（WAS）、データベース（DB）サーバの 3 層構造をなすシステムを想像してみよう。このようなシステムでは、各サーバーの連係動作が現象の記述に本質的である。すなわち、トランスポート層およびそれ以下での結合に加えて、**アプリケーション層**における結合が本質的である。しかしながら、各ノードが比較的高い独立性を持っている下層での結合と違い、アプリケーション層での障害検知は、既存の NNMS では難しいとされてきた。

より具体的に、HTTP サーバーと WAS が 2 重冗長構成となっているシステムを考える。仮に一方の WAS において、想定外の重い処理が行われるなどの理由で、ある時からその活動が低下したとしよう。サーバープロセス自体は生きており、トランスポート層およびそれ以下では通信に異常は見いだされないとする。このような場合、2 台ある WAS の対称性はアプリケーション層では明らかに破れてはいるが、比較的低いトラフィックでは、ユーザから見た応答時間には異常はほとんど出ない。一方の WAS が正常に処理を行うからである。また、負荷分散装置がしばしば行う各プロセスの生存確認でも異常が出ない。しかしこのような状況は潜在的に危険である。処理能力の設計値よりもよりもはるかに低い値でシステムがダウンする可能性があるからである。

まとめると、市販の NNMS は、下記のような限界を持っている。

- 局所的な情報を単純に集約して表示することはできるが、それらを関連付けることが難しい。
- 基本的に、個別の観測地点において、閾値を測定値と単純比較することで異常が判定されるが、閾値を決めるにはかなりの専門的知識が要る。
- アプリケーション層の障害のように、ノード間の連携動作が本質的な場合には有効な障害検知を行えない。

本論文の目的は、Web 系システムのアプリケーショ

*IBM 東京基礎研究所, 242-8502 神奈川県大和市下鶴間 1623-14, e-mail goodidea@jp.ibm.com,
IBM Research, Tokyo Research Laboratory, 1623-14 Shimotsuruma, Yamato-shi, Kanagawa, Japan
†IBM 東京基礎研究所, 同, e-mail hkashima@jp.ibm.com.

ン層における障害を、教師なし学習の枠組みで自動検知することである。論点は 3 つある。

一つは、多自由度系、それも各自由度が強く絡み合った多自由度系であるところのコンピュータシステムの状態を、どのように記述するかである。その「自由度」なるものをどう定義すれば我々の目的に合うのかは自明ではない。アプリケーションのサービス品質を表すのにはたとえば応答時間やスループットが使われることがあるが、上記の例からも、適切な指標ではないことがわかる。少なくとも、局所的な観測値同士の相関をあらわに取り込んだ形の量が不可欠である。

二つは、その強相関多自由度系の全体像を、どう把握するかである。一般に、強相関の多自由度系では、取りえる状態の数がきわめて大きくなる。その一部をミクロに見ても、システム障害の判定は難しい。全体をいわば「ぼんやり」見て、何かバランスが崩れているというのを検出したいわけであり、そのための特徴量の抽出が問題になる。

三つは、仮にそうした特徴量を得たとして、どのように異常を検知するかである。何か閾値が必要だとしても、それは系の詳細な知識を必要としないようなものであることが望ましい。願わくば、何かの妥当な確率モデルに基づいて、たとえば「0.5%危険域に入れば警報を発せよ」のように、パーセント値を閾値として採用したい。

以下、第 2 節においてコンピュータシステムのアプリケーション層のモデル化手法を説明する。次に第 3 節においてコンピュータシステムの異常検知に適した特徴抽出手法を述べる。第 4 節では、新たに開発した von Mises-Fisher 分布のオンライン更新則に基づいた異常検出手法を説明する。第 5 節では、簡単なベンチマークシステムで提案手法の実験的検証を行う。

2　コンピュータシステムのモデル化

2.1　サービス関連度行列

アプリケーション層での連係動作を記述するための最小要素として、Web 系システムの「サービス」を、下記の 4 つ組で定義する：

$$(I_s, I_d, P, Q),$$

I_s と I_d はそれぞれ、呼び出し元と呼び出し先の IP アドレス、P は呼び出し先のアプリケーションのポート番号、Q は要求の種別を表す。たとえば、図 1 のベンチマークシステムにおいて、(i_1, i_3, p_1, q_1) がサービスの例である。ここで、i_1 と i_3 はたとえば、それぞれ 192.168.0.19 と 192.168.0.53 というような IP アドレスであり、p_1 はポート番号 80、q_1 にはたとえばベンチマークアプリケーションの名前が来る。

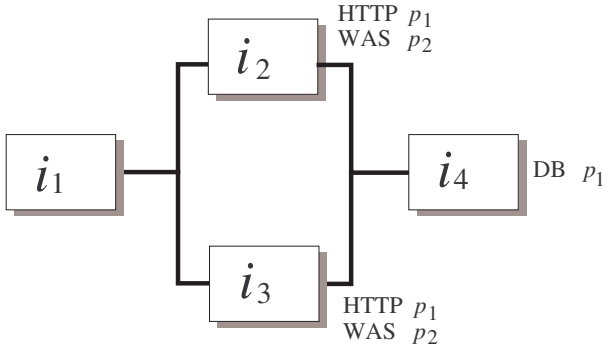

図 1: ベンチマークシステムの構成。IP アドレスとポート番号をそれぞれ i_k $(k = 1, .., 4)$ と p_j $(j = 1, 2, 3)$ で表してある。

次に、このようなサービスを頂点とするグラフを考える（図 2）。IP アドレスをふたつ含むという意味で、図 1 のような素朴な IP ネットワークよりも高次の空間を考えていることに注意されたい。この高次元化にともない、図 1 のような単純なシステムにおいても、サービスのグラフは比較的複雑になりえる。

サービス i と j を結ぶ辺には、サービスの呼び出しの出現頻度に対応した重みを与える。すなわち、このグラフの隣接行列 D の i, j 要素として、

$$\mathsf{D}_{i,j} = (1 - \delta_{i,j}) \left[f(d_{i,j}) + f(d_{j,i}) \right] + \alpha_i \delta_{i,j} \qquad (1)$$

という重みを考える。ここで、$d_{i,j}$ は、ある所定の期間においてサービス i がサービス j を呼び出した回数、$f(\cdot)$ はある単調増加関数で、ここでは

$$f(x) = \ln(1 + x)$$

とおく。また、α_i は数値計算の安定化のために導入された小さい正数、$\delta_{i,j}$ はクロネッカーのデルタである。$d_{i,j}$ の測定ないし推定の方法は自明ではないが、いくつかの既知の方法（たとえば [5]）で計算できるので、ここでは論じない。上で与えられる行列を、以降、サービス関連度行列と呼ぶ。

2.2　問題設定

本稿で考える問題を改めてまとめておく。我々は、システムの振る舞いがサービス関連度行列 D によって特徴付けられると仮定する。その次元 N と、各要素の定義は固定されているものとみなすが、その値は時間的に激しく変動する。問題は、この行列を、グラフの隣接行列と見て、その時系列から異常を検知することである。

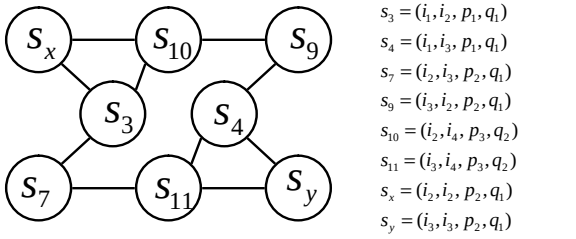

図 2: 図 1 のシステムにおけるサービス関連度グラフの例。

考えるべき点としては、隣接行列の次元はしばしば 100 のオーダーとなり、個別の行列要素もしくはそれらの恣意的な組み合わせの監視は現実的ではないこと、また、隣接行列の要素は強く時間に依存するので、個々の要素を個別に眺めるだけでは障害の検出のしようがないこと（単なるトラフィックの変動かもしれない）、障害を判定する基準からできるだけ恣意的なパラメターを排除すること、が挙げられる。

3 関連度グラフからの特徴抽出

3.1 活動度ベクトルの定義

サービス関連度行列は、ある離散的な時刻 t=1,2,... において時系列的に得られるものとする。サービス関連度グラフの特徴ベクトル $\boldsymbol{u}$ として、下記のようなものを考える。

$$\boldsymbol{u}(t) \equiv \arg\max_{\tilde{\boldsymbol{u}}} \left\{ \tilde{\boldsymbol{u}}^T \mathsf{D}(t) \tilde{\boldsymbol{u}} \right\} \tag{2}$$

ただし $\tilde{\boldsymbol{u}}^T \tilde{\boldsymbol{u}} = 1$ とし、さしあたり関連度グラフが単一の連結成分しか持っていないと仮定しておく。ベクトルは列ベクトルとする。T は転置を表す。行列 D は非負なので、上式の目的関数を最大化するには、大きい行列要素に対応するサービスにおいて、特徴ベクトルの要素が大きくなっていなければならない。すなわち、もしあるサービス s が活発に他のサービスを呼び出していれば、特徴ベクトルの第 s 成分は大きいはずである。この意味で、この特徴ベクトルを、**活動度ベクトル**と呼ぶ。

ラグランジュ係数 λ を導入すれば式 (2) は下記のように書き直せる。

$$\frac{d}{d\tilde{\boldsymbol{u}}} \left[\tilde{\boldsymbol{u}}^T \mathsf{D}(t) \tilde{\boldsymbol{u}} - \lambda \tilde{\boldsymbol{u}}^T \tilde{\boldsymbol{u}} \right] = 0,$$

すなわち

$$\mathsf{D}(t) \tilde{\boldsymbol{u}} = \lambda \tilde{\boldsymbol{u}}. \tag{3}$$

この方程式は $\mathsf{D}(t)$ の任意の固有ベクトルに対して成り立つが、活動度ベクトルは主固有ベクトル（最大固有値に属する固有ベクトル）として定義される。

ひとつ大切な点は、式 (3) が $\tilde{\boldsymbol{u}}$ について同次形であることである。このため、たとえば、トラフィックの急増により、すべてのサービス関連度が一斉に k 倍されたとしても、その変化はもっぱら固有値に吸収され、活動度ベクトルの方向には影響しない。

活動度ベクトルは、運動方程式が

$$\boldsymbol{x}(\tau + 1) = \mathsf{D}(t) \boldsymbol{x}(\tau)$$

で与えられる離散時間の線形力学系の定常状態と結びつけて解釈することもできる。ここで τ は、実時間 t とは別のある仮想時間を表し、$\boldsymbol{x}$ は $\boldsymbol{u}$ と、$\boldsymbol{u}$=$\boldsymbol{x}$/||$\boldsymbol{x}$|| という関係で結ばれる。$\mathsf{D}(t)$ は定義より対称で、また、少なくとも $\alpha > 0$ に対してはフルランクなので、すべての固有値は実である。対応する固有ベクトルを用いて $\boldsymbol{x}(0)$ を展開することで、

$$\boldsymbol{x}(\infty) = \lim_{n\to\infty} [\mathsf{D}(t)]^n \boldsymbol{x}(0) = \lim_{n\to\infty} \sum_{i=1}^{N} [\lambda_i(t)]^n c_i(t) \boldsymbol{u}_i(t),$$

ただし $c_i(t)$ は対応する展開係数である。明らかに、$n \to \infty$ において、

$$\boldsymbol{u}(t) = \boldsymbol{x}(\infty)/||\boldsymbol{x}(\infty)||$$

を得る。すなわち、この力学系の状態ベクトルは、無限回の遷移の後に活動度ベクトルに収束する。このことから、コンピュータシステムでは、この定常状態は、ある仮想時刻 τ において、システムのコントロール・トークンを、あるサービスが保持する確率振幅として解釈できることがわかる。

3.2 活動度ベクトルの諸性質

まず、これは前節ですでに述べたが、規格化された固有ベクトルとしての活動度ベクトルは、全体的なトラフィック変動に対し頑強である：

性質 1 $\boldsymbol{u}$ の方向は、変換 $\mathsf{D}(t) \to k\mathsf{D}(t)$ に対して不変である。ただし、k は任意の正数とする。

我々のサービス関連度行列は非負行列であるため、Perron-Frobenius の定理 [12] により次の性質が成り立つ。

性質 2 主固有ベクトルは正ベクトルである（すなわち、すべての要素を正にとれる）。

性質 3 主固有値には縮退がない。

性質 2 から、各サービスの活動度の値は負になることはなく、それゆえ、活動度としての解釈に不整合は生じない。固有ベクトルの縮退は、微小な揺らぎに対する不安

定性をもたらすが、性質 3 によればその心配もないことがわかる。

一般には、時系列的に得られるグラフは非連結グラフとなり、厳密には Perron-Frobenius の定理の成立条件を満たさない。しかしその場合は、最大の固有値を持つ連結成分（これを主固有クラスター [9] と呼ぶ）の主固有ベクトルを、全系の活動度ベクトルと定める。幸いなことに、Web 系のシステムでは、HTTP サーバーが Web アプリケーションサーバーを呼び出すというような重要なサービスは主固有クラスターに集中しているので、単純なルールで主固有クラスターを識別することができる。[1]

最後にふたつほどコメントを述べておく。まず最初に、この特徴抽出手法は、D に含まれる情報の効率のよい圧縮手法となっている。すなわち、全系を表すグラフから、主固有クラスターを抽出し、さらにそこから活動度ベクトルを抽出することで、自由度を大幅に削減している。特徴量の意味づけが上記のように明確であることと併せ、素朴に全系の隣接行列の全要素を一列に並べて特徴ベクトルにする、というような手法に比べて、より利口な手法だと言える。

第二に、主固有値および主固有ベクトルを求めるための数値計算は、べき乗法 [10] と呼ばれる手法を使って非常に高速に実行できる。活動度ベクトルはオンラインに、我々の実験環境であれば 10 秒のオーダーの時間間隔の間に計算する必要があるが、N が数百程度では計算時間は全く問題にならない。

4 異常検知手法

4.1 異常度の定義

活動度ベクトルの時系列 $\{\boldsymbol{u}^{(t)}|t=1,2,...\}$ からの異常検知問題を考える。活動度ベクトルは規格化されているので、これは方向データ（L_2 ノルムが 1 に規格化されているベクトル）の時系列からの異常検知ということになる。基本的な手順は、過去の正常時のパターンをオンライン学習して、現時点の活動度ベクトルとの相違度を見る、というものである。相違度もしくは異常度としては、下記のようなものを考える。

$$z(t) \equiv 1 - \boldsymbol{r}(t-1)^T \boldsymbol{u}(t). \tag{4}$$

ここで、$\boldsymbol{r}(t-1)$ は、現在時刻 t の直前の時刻 $t-1$ における代表パターンを表す。これも規格化されていると想定する。したがって、過去の代表パターンと現在の活動度ベクトルが直交していれば異常度の値は 1、完全に一致していればゼロとなる。ここでは天下り式に z の形を与えたが、後述する確率モデルとは Fisher 核関数の意味で首尾一貫した関係にある [8]。

詳細はおいおい述べてゆくが、我々の異常検知手順を図 3 に要約しておく。

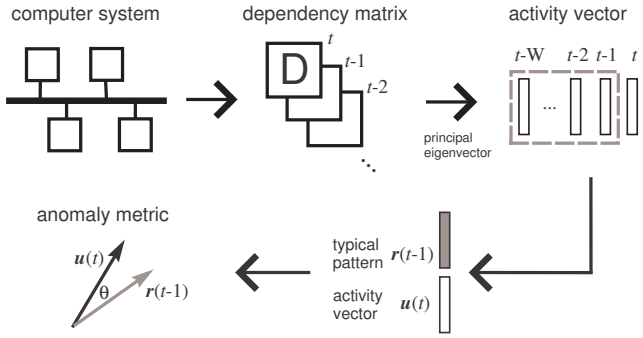

図 3: 異常検知手順の要約

4.2 過去の代表パターンの計算

式 (4) における $\boldsymbol{r}$ を求めることを考えよう。ここで便宜上、行列 $\mathsf{U}(t)$ を以下に定義する。

$$\mathsf{U}(t) = [\boldsymbol{u}(t), \boldsymbol{u}(t-1), \cdots, \boldsymbol{u}(t-W+1)],$$

ここで W は時間窓のサイズである。明らかに $\mathsf{U}(t)$ は $N \times W$ の行列となる。W の時間窓の範囲での代表パターンを、列ベクトルの 1 次結合として、

$$\boldsymbol{r}(t) = c \sum_{i=1}^{W} v_i \boldsymbol{u}(t-i+1) \tag{5}$$

のようにおく。ただし係数 $\{v_i\}$ は $\sum_{i=1}^{W} v_i^2 = 1$ を満たすとする。c は $\boldsymbol{r}$ を 1 に規格化するための係数である。単純には、$\{v_i\}$ を定数とし、$\boldsymbol{r}(t)$ を相加平均とみなすことができるが、実用上は、揺らぎの悪影響を避けるために下記のような方程式から係数ベクトル $\boldsymbol{v}^T = (v_1, v_2, ..., v_W)$ を求めるのが好ましい。

$$\boldsymbol{v}(t) \equiv \arg\max_{\tilde{\boldsymbol{v}}} \left|\left| \sum_{i=1}^{W} \tilde{v}_i \boldsymbol{u}(t-i+1) \right|\right|^2 \tag{6}$$

ただし、$\tilde{\boldsymbol{v}}^T \tilde{\boldsymbol{v}} = 1$ が条件である。これが意図するところは、$\boldsymbol{u}$ の合ベクトルとしての $\boldsymbol{v}$ が、$\boldsymbol{u}$ のもっとも人気のある方向に向いているようにとる、ということである。式 (5) が

$$\boldsymbol{r}(t) = c\mathsf{U}(t)\boldsymbol{v}(t) \tag{7}$$

と書けることに注意すると、上記の極値方程式は

$$\frac{d}{d\tilde{\boldsymbol{v}}} \left[\tilde{\boldsymbol{v}}^T \mathsf{U}(t)^T \mathsf{U}(t) \tilde{\boldsymbol{v}} - \mu \tilde{\boldsymbol{v}}^T \tilde{\boldsymbol{v}} \right] = 0$$

[1] 非連結なシステムに対する安定性の議論は Idé-Kashima [7] を参照。

のように書き直せる。ただし、ラグランジュ乗数を $c^2\mu$ とおいた。結局この式は

$$\left[\mathsf{U}(t)^T\mathsf{U}(t)\right]\tilde{\boldsymbol{v}} = \mu\tilde{\boldsymbol{v}} \tag{8}$$

と同じである。これは $\mathsf{U}(t)^T\mathsf{U}(t)$ の任意の固有ベクトルで満たされるが、代表パターンとしては最大固有値に属するものを採用する。式 (8) と規格化条件を用いると、

$$c = 1/\sqrt{\mu} \tag{9}$$

が直ちに導かれる。式 (7)、(8)、(9) は、$\boldsymbol{r}(t)$ が行列 $\mathsf{U}(t)$ の、最大特異値に属する左特異ベクトルであることを意味している。この場合も数値計算はべき乗法 [10] を好適に利用できる。

4.3 異常度の確率モデル

異常度の確率モデルを導くために、まず、活動度ベクトルが従う確率モデルから考えよう。方向データに対して、エントロピー最大原理の見地からもっとも自然なモデルは次の von Mises-Fisher 分布である [1]。

$$p(\boldsymbol{u}|\kappa,\boldsymbol{\mu}) = \frac{\kappa^{\frac{N}{2}-1}}{(2\pi)^{N/2}I_{\frac{N}{2}-1}(\kappa)}\exp\left(\kappa\boldsymbol{\mu}^T\boldsymbol{u}\right) \tag{10}$$

ここで $\boldsymbol{\mu}$ は平均方向で、$I_m(\cdot)$ は m 階の第 1 種の変形ベッセル関数、$1/\kappa > 0$ は角分散（angular variance）と呼ばれる定数、N は方向データ $\boldsymbol{u}$ の形式的な次元である。我々の文脈では、$\boldsymbol{\mu}$ を $\boldsymbol{r}(t-1)$ と等値し、各時刻 t において

$$\cos\theta = \boldsymbol{r}(t-1)^T\boldsymbol{u}$$

なる角度変数 $\theta \in [0,\pi]$ の分布を考えていると考えてもよい。この θ についての分布を式 (10) から周辺分布として導くために、$\boldsymbol{u}$ から N 次元球座標 $\{\theta,\theta_2,...,\theta_{N-1}\}$ へ変数変換する。ここで N 次元ユークリッド空間の単位球上の面積要素 $d^{N-1}\Omega$ が

$$d^{N-1}\Omega = d\theta d\theta_2\cdots d\theta_{N-1}\sin^{N-2}\theta\sin^{N-3}\theta_2\cdots\sin\theta_{N-2}$$

と書けることに注意すると、θ に対する周辺分布は、形式的に

$$p(\theta) = \int p(\boldsymbol{u}|\kappa,\boldsymbol{\mu})\sin^{N-2}\theta\sin^{N-3}\theta_2\cdots\sin\theta_{N-2}\prod_{i=2}^{N-1}d\theta_i$$

と表せる。今は、$|\theta| \ll 1$ を仮定できるので、

$$z(t) \simeq \frac{\theta^2}{2},\ \cos\theta \simeq 1-\frac{\theta^2}{2},\ \sin\theta \simeq \theta$$

が成り立つ。これを使うと角度変数を z に翻訳できて、結局、z についての確率分布が、

$$q(z) \propto \exp\left[-\frac{z}{2\Sigma}\right]z^{\frac{N-1}{2}-1}$$

のように求まる。ここで、$\theta d\theta = dz$ という関係を用い、また、$1/(2\kappa)$ を Σ とおいた。この関数は、本質的には、自由度 $N-1$ の χ^2 分布と同じである。

このモデルは、最大エントロピー原理の意味で、記述能力の真っ当さを保障されたモデルである。しかしこれをそのまま実験データに適用しようとすると、まったく実験と合わない。我々のモデル化の次の重要なステップは、上記の次元 N を、フィッティングパラメターとしての**有効次元** n で置き換える、というものである。すなわち、z の確率モデルとしては

$$q(z) = \frac{1}{(2\Sigma)^{(n-1)/2}\Gamma((n-1)/2)}\exp\left[-\frac{z}{2\Sigma}\right]z^{\frac{n-1}{2}-1} \tag{11}$$

を考える。ここで上と同じ近似の程度において規格化定数を求めた。Γ はガンマ関数を表す。有効次元 n は系の実質的な自由度を表す。見かけの自由度が極めて大きくても、実質的な自由度が実はかなり小さいということは、文書ベクトルの世界 [2] をはじめ実世界の多くの場所で見受けられる。有効次元については興味深い応用がさまざまあり、詳細な議論は別途論じる予定である。

4.4 確率モデルのオンライン更新則

確率モデルのパラメター推定のためには、（時に期待値最大化流の）最尤推定とそのオンライン版を考えるのが定石である [11]。しかし χ^2 分布では、ガンマ関数の存在のために、直接最尤推定を行うのが難しい。そこで、1 次および 2 次のモーメントについての χ^2 分布 (11) のよく知られた関係

$$\langle z\rangle = (n-1)\Sigma,\ \ \langle z^2\rangle = (n^2-1)\Sigma^2$$

を使うことを考える。幸運なことに、これはパラメター n と Σ について逆に解け、

$$n-1 = \frac{2\langle z\rangle^2}{\langle z^2\rangle - \langle z\rangle^2},\ \ \Sigma = \frac{\langle z^2\rangle - \langle z\rangle^2}{2\langle z\rangle}, \tag{12}$$

右辺のモーメントについては恒等式

$$\frac{1}{t}\sum_{i=1}^{t}z(t) = \left(1-\frac{1}{t}\right)\frac{1}{t-1}\sum_{i=1}^{t-1}z(i) + \frac{1}{t}z(t)$$

において、$1/t$ を忘却率 β と読み替えることにより、

$$\langle z\rangle^{(t)} = (1-\beta)\langle z\rangle^{(t-1)} + \beta z(t) \tag{13}$$

$$\langle z^2\rangle^{(t)} = (1-\beta)\langle z^2\rangle^{(t-1)} + \beta z(t)^2 \tag{14}$$

のように容易にオンライン化できる。当然 β は $0<\beta<1$ の範囲で選ぶ。これはデータ点の数の逆数に対応するので、興味ある時間スケールと行列生成間隔を元に、値を選択することができる。

以上まとめると、異常か否かを判断するオンライン計算則が以下のように与えられる。

1. 危険域の値 $0 < p_c < 1$ を与える。
2. 式 (13) と (14) により $\langle z \rangle$ と $\langle z^2 \rangle$ を時刻 t において求める。
3. $n - 1$ と Σ を式 (12) により求める。
4. $\int_{z_{\rm th}}^{\infty} dz q(z) = p_c$ を満たす $z_{\rm th}$ を数値的に求める。
5. もし $z(t) > z_{\rm th}$ なら警告を発する。

この計算則には p_c、β、W というパラメターが含まれている。β と W は、1 分なり 10 分なりといった興味ある時間スケールを自ら指定することにより決めることができる。これを決めてしまえば、システム管理者のすべきは危険域 p_c を指定することだけである。いずれにしてもシステムのふるまいに関しての詳細な知識は必要ない。

5 実験

5.1 実験構成

図 1 に示した 2 重冗長構成の Web 系システムにおいて、異常検知の実験を行った。ウェブアプリケーションサーバー（WAS）の上では、Trade3 [6] と Plants というふたつのアプリケーションが動作しているものとする [2]。両者ともクライアント数は 16、いわゆる think time は 0 から 4 秒の間でランダムに選択した。

サービス関連度行列は 20 秒間隔で生成するものとし、$d_{i,j}$ は採取されたイーサネットパケットから推定した。ループバックパケットは無視したので、図 2 における s_x と s_y のサービスは観測されていない。ただし $i_1 =$ 192.168.0.53 および $i_2 =$ 192.168.0.54 である。主固有クラスターは表 1 に定義されている。簡単のため、これらに対する摂動は無視し、活動度ベクトルはここに示されたサービスで張られる空間に限定する。表 1 において、第 0 サービスは呼び出し元と呼び出し先の最適な対が見出せない場合を表現するために人工的に導入されたもので、これを無視しても結果の解釈には影響はない。表で、DB2 は DB サーバーへのリクエストを表し、JMS は Java Messaging Service に関連した通信を表す。

5.2 確率モデルの検証

まず最初に、サービス関連度行列の行列要素の時間変化を見てみよう。図 4 に、一例として $d_{9,11}$ の挙動を示した。サービス関連度行列は 52.7 分にわたり 158 個が生成された。図より、上記実験条件の下では 20 秒間におよそ 500 回のコールがあることがわかる。揺らぎはきわめて大きく、その振幅は、ほとんど平均値と同じオーダーとなっている。対応する行列要素も、正規分布で扱えるような穏やかなものではなく、関連度行列の行列要素に直接閾値を設定してもあまり意味がないことがこの結果から分かる。

表 1: 主固有クラスターに現れるサービスの一覧

Index	I_s	I_d	P	Q
0	0.0.0.0	0.0.0.0	0	(none)
1	192.168.0.19	192.168.0.53	80	Plants
2	192.168.0.19	192.168.0.54	80	Plants
3	192.168.0.19	192.168.0.53	80	Trade
4	192.168.0.19	192.168.0.54	80	Trade
5	192.168.0.54	192.168.0.53	5558	JMS
6	192.168.0.53	192.168.0.54	9081	Plants
7	192.168.0.53	192.168.0.54	9081	Trade
8	192.168.0.54	192.168.0.53	9081	Plants
9	192.168.0.54	192.168.0.53	9081	Trade
10	192.168.0.53	192.168.0.52	50000	DB2
11	192.168.0.54	192.168.0.52	50000	DB2

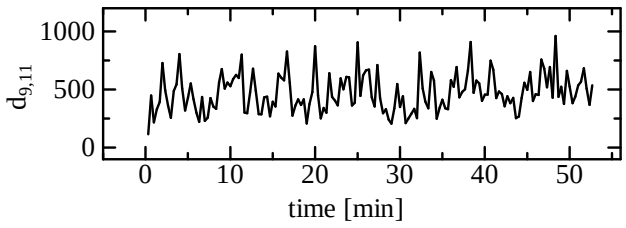

図 4: サービス 9 によるサービス 11 の呼び出し回数の時間変化.

次に、同じ条件の下、異常度 z を計算し、頻度分布としてプロットしたものが図 5 (a) である。χ^2 分布も併せて描いてある。分布のパラメターは、158 データ点を、忘却率なしで計算して求めた。結果は $n = 4.62$ および $\Sigma = 6.79 \times 10^{-5}$ である。この n の値は、先に定義した有効次元が、見かけの次元 $N = 12$ よりはるかに小さいことの一つの実例になっている。図からわかるように頻度分布と χ^2 分布の曲線の一致は良好である。念のため χ^2 分布に対する z の分位点プロットを図 5 (b) に示す。実験データは 45 度の直線上にほぼ並び、我々のモデルが実験をよく再現していることがわかる。

[2] 後者は IBM WebSphere Application Server V5 付属のサンプルアプリケーションであり、花のオンライン販売を模擬する。

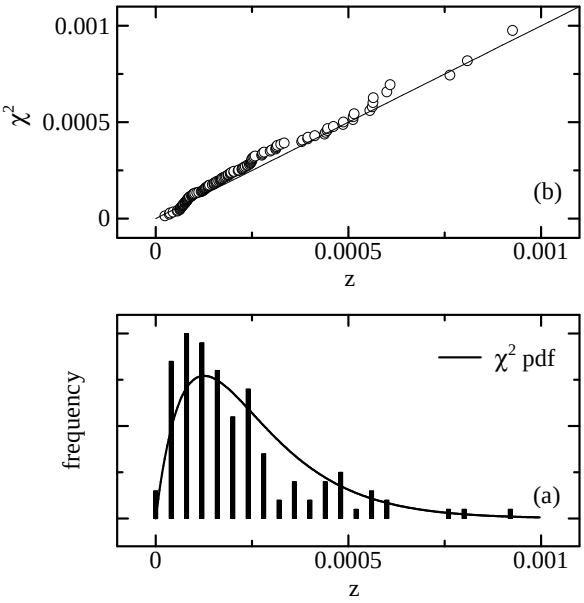

図 5: 正常時における異常度の頻度分布 (a) と、分位点プロット (b)。

5.3 異常検知

異常検知能力を調べるために、サービス 11、すなわち、192.168.0.54 における “Plants” に変調を生じさせ、DB コールを止める実験を行った。動作は不調であるが、サーバーソフトウェアのプロセス自体は動いている。しかも、WAS の一方は正常に動いているために、トラフィックがさほど大きくないうちは応答時間にもほとんど異常は出ない。そのため通常の NNMS では検出が難しい。

図 6 に、計算された活動度ベクトルを示す。縦軸が表 1 に対応するインデックス、横軸が時間である。時刻 t_A と t_B において、段差状の変化が見て取れる。これはサービス 11 の機能不全とその復旧に対応している。この区間においては、サービス 2、6、11 に活動度の減少が観察される。サービス 6 はサービス 11 の上流側にあり、11 の不調に伴い、この活動度も低下したと解釈される。サービス 2 が強く影響を受けたのは、ループバックパケットが非観測であるためである。Web 系システムのようなサービス同士の相関が強いシステムでは、ひとつのサービスの異常な挙動が、このように、システム全体にある種の相変化を生じさせる。活動度ベクトルは、そのような系全体にわたる変化を抽出するために好適な手法であると言える。図 6 は、システム全体の可視化手法という点でも興味深いものである。

この障害を自動検知するために、z をオンラインで計算した。図 7 において、z の値は、$\boldsymbol{r}(t)$ を相加平均で計算したものと、$\mathsf{U}(t)$ の特異値分解を経由したものを比較してある。図の様相は窓の大きさによりかなり変わる

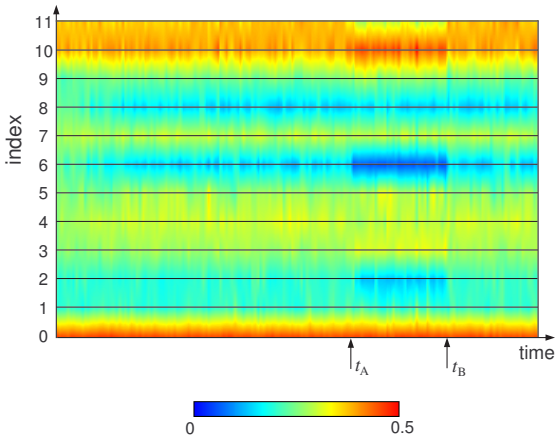

図 6: 活動度ベクトルの時間依存性。t_A から t_B までの期間が障害発生期間に対応している。

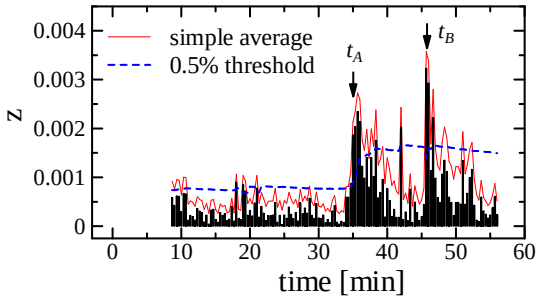

図 7: z の時間依存性

が、ここでは実験的に求めた最適値 $W = 25$ に選んだ。z に加えて、$p_c = 0.5\%$に対応する閾値 $z_{\rm th}$ をオンラインで学習し、点線でプロットしてある。β は 0.005 にとった。図では、$t = 35.0$ と 45.7 の区間で明瞭な構造が見てとれ、また区間の両端において z が $z_{\rm th}$ を上回っていることがわかる。明らかにこれは図 6 の $t_A < t < t_B$ に対応しており、障害の自動検知における本手法の有効性が確認できる。

6 要約と今後の課題

相関の強い多自由度系としての Web 系システムのアプリケーション層におけるオンライン障害検知の問題を考えた。まず、システムの状態を、サービス関連度行列として把握することを提案した。次に、サービス関連度行列からサービス活動度ベクトルという量を定義し、それが、システムの動態の縮図として機能することを述べた。活動度ベクトルは L_2 ノルムが 1 に規格化されているため、問題は、方向データからの異常検知の問題となる。我々は、von Mises-Fisher 分布から出発して、適切に定義された異常度の近似的確率分布を導き、そのオン

ライン更新式を与えた。

見方を変えると我々は、グラフ時系列からの異常検知の問題への一解法を提案したとも言える。グラフの特徴量を考える上で付きまとうのは、その高自由度がゆえの扱いにくさであったが、新たに考案した特徴抽出手法と、有効次元の概念を活用した確率モデルにより、その困難を乗り越えることにある程度成功した。

今回は小さいベンチマークシステムで実験を行ったのみであり、提案手法を実運用に適用するには、当然まだ解決すべき問題はある。それとは別に、機械学習の観点から興味深い研究テーマを二つほど挙げよう。一つは、今回の特徴抽出手法と、最近注目を集めるスペクトラル・クラスタリングとの関連の検討である [4, 3]。もう一つは、本論文で強調した有効次元の概念の展開である。

データベースの高速探索技術から出発したデータマイニングは、最近はその対象と手法を大幅に広げ、いわば知識発見工学として発展を続けている。本論文がそこに新しい何かを付け加えられたら幸いである。

参考文献

[1] A. Banerjee, I. Dhillon, J. Ghosh, and S. Sra. Generative model-based clustering of directional data. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 19–28, 2003.

[2] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[3] I. S. Dhillon, Y. Guan, and B. Kulis. Kernel k-means, spectral clustering and normalized cuts. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 551–556, 2004.

[4] C. Ding. Spectral clustering. In *Tutorial Notes of The Twenty-First International Conference on Machine Learning*, 2004.

[5] M. Gupta, A. Neogi, M. K. Agarwal, and G. Kar. Discovering dynamic dependencies in enterprise environments for problem determination. In *Proceedings of 14th IFIP/IEEE Workshop on Distributed Systems: Operations and Management*, pp. 221–233, 2003.

[6] IBM. Trade3; `http://www-306.ibm.com/software/webservers/appserv/benchmark3.html`.

[7] T. Idé and H. Kashima. Eigenspace-based anomaly detection in computer systems. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 440–449, 2004.

[8] T. Idé and H. Kashima. Effective dimensions in anomaly detection: Its application to computer systems. *Lecture Notes in Computer Science*, 2005. in press.

[9] S. Sarkar and K. Boyer. Quantitative measures for change based on feature organization: Eigenvalues and eigenvectors. *Computer Vision and Image Understanding*, 71:110–136, 1998.

[10] G. Strang. *Linear Algebra and its Applications*. Academic Press, 1976.

[11] K. Yamanishi, J. Takeuchi, G. Williams, and P. Milne. On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 320–324, 2000.

[12] 伊藤昇, 岩井斉良, 岩堀長慶, 関野薫, 高橋秀一. 経済系・工学系のための行列とその応用. 紀伊国屋書店, 1987.