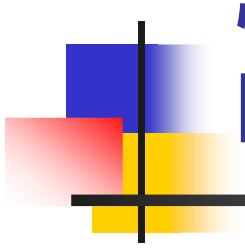
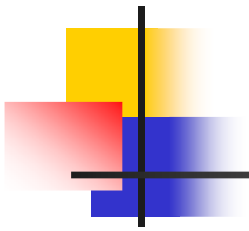


統計的手法による
時系列からの外れ値と変化点の検出

平成19年度　前期雑誌会
B4　松本　亮介



本発表

ネットワーク上で生じるような非定常な時系列データに対して、外れ値と変化点を同一のアルゴリズム内で扱うことができる手法[4]を紹介する。

⇒　外れ値による誤検出を低減

⇒　外れ値と変化点を区別して変化点を検出

---

[4] J. Takeuchi and K. Yamanishi, “A Unifying Framework for Detecting Outliers and Change Points from Time Series,” IEEE transactions on Knowledge and Data Engineering, pp.482-492, 2006.

研究目的

- データマイニングの観点
  - 外れ値と変化点を検出
  - 同一のアルゴリズムで扱う

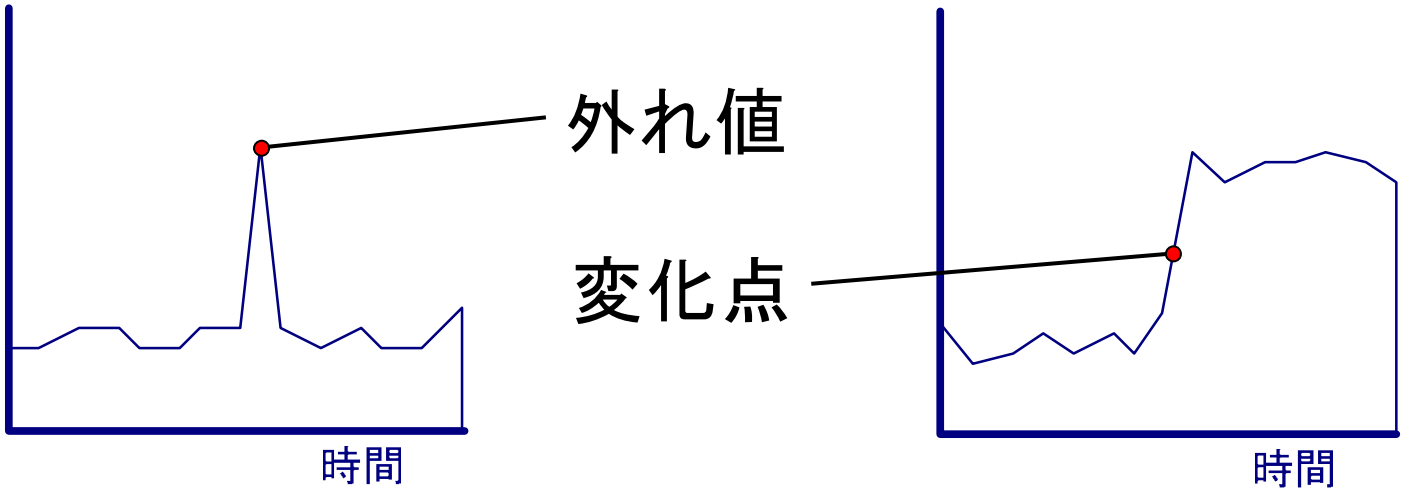
    ⇒ 明確な関係はこれまで述べられていない
- 定常状態からの変化に注目
  - 変化点検出 ⇒ データの性質が急激に変わる点を検出

■ ネットワークセキュリティに利用

⇒ ネットワーク障害の検知
⇒ サイバーテロ監視の効率化
⇒ 不正アクセスやウィルスの早期発見

外れ値と変化点

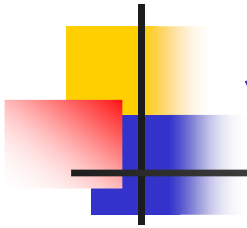
性質的に似ている点 ⇒ 同時に統計的に扱うことが難しい



外れ値・・・瞬間的なデータの増減 ⇒ 次の時点では定常に戻る
変化点・・・定常な時系列データの性質が急激に変化する時点

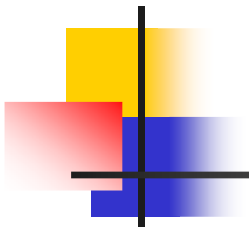
⇒ 外れ値の連続を変化点と解釈

⇒ 外れ値と変化点を同一のアルゴリズムで扱う



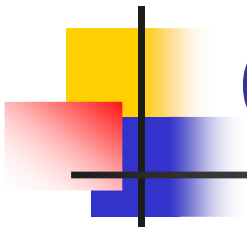
発表の流れ

- 紹介手法（ChangeFinder）の概要
- 2段階学習アルゴリズム
- 実験
- まとめ



目次

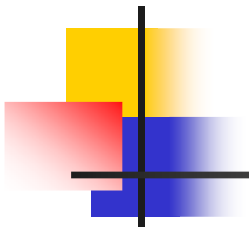
- 紹介手法（ChangeFinder）の概要
  - ChangeFinderとは
- 2段階学習アルゴリズム
- 実験
- まとめ



ChangeFinderとは

- 非定常な時系列データに対応
  - 独自の忘却型学習アルゴリズムを採用
    ⇒　ゆるやかな変化と急激な変化を区別
- リアルタイム性の確保
  - 計算量の削減（従来と比較）
- 外れ値と変化点を区別して検出
  - 2段階学習を採用

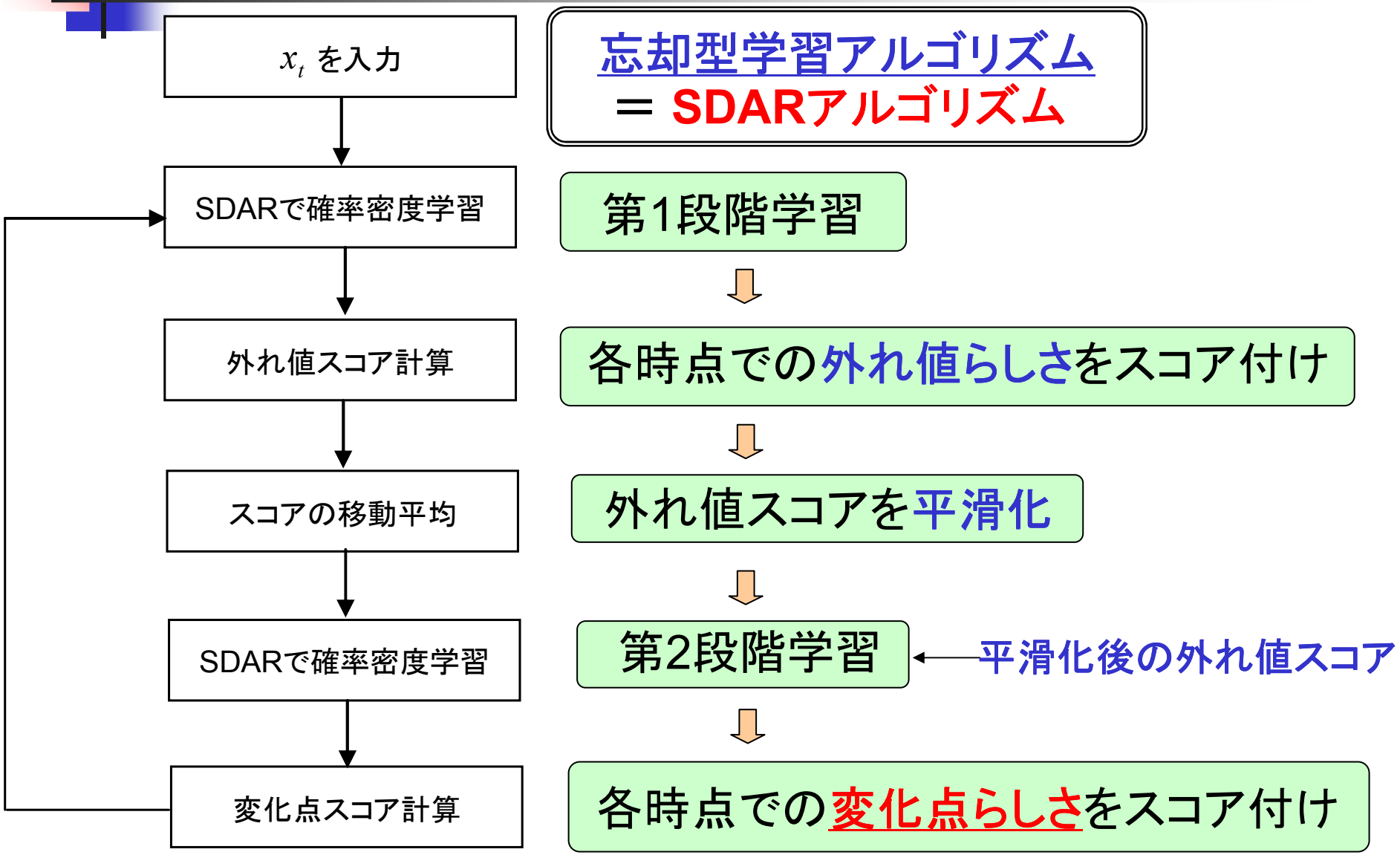
⇒　ログからDoSアタックやワーム発生検出
⇒　ネットワークシステム障害検出



目次

- 紹介手法（ChangeFinder）の概要
- 2段階学習アルゴリズム
  - 忘却型学習アルゴリズム
  - 外れ値検出
  - 外れ値スコアの平滑化
  - 変化点検出
- 実験
- まとめ

2段階学習の流れ


x_t を入力
SDARで確率密度学習
外れ値スコア計算
スコアの移動平均
SDARで確率密度学習
変化点スコア計算
忘却型学習アルゴリズム
＝ SDARアルゴリズム
第1段階学習
各時点での外れ値らしさをスコア付け
外れ値スコアを平滑化
第2段階学習
平滑化後の外れ値スコア
各時点での変化点らしさをスコア付け

忘却型学習アルゴリズム

- 統計モデル　⇒　ARモデル採用
  - 最も典型的な統計モデル
  - 時系列データを効率よく推定できるモデル
  - 問題点
    - 定常であるという仮定
    - バッチ学習方式　⇒　計算量が多い

ARモデルを改良　⇒　逐次学習
⇒　忘却機能　⇒　SDARアルゴリズム

非定常なデータに対応・計算コストの削減

忘却型学習（SDAR）アルゴリズム

そもそもARモデルとは・・・ 予測パラメータを新しいデータ毎に計算

$$x_t = A_{t-1}x_{t-1} + A_{t-2}x_{t-2} + \cdots$$

直前までのデータからパラメータを推定 ⇒ 次のデータを予測

計算量をどのように削減するかが重要

- ARモデル（バッチ学習方式）
  まとまったデータからパラメータ計算 ⇒ Σ計算 ⇒ 計算量多い
- SDARアルゴリズム（逐次学習方式）
  以前のパラメータと新たな時系列データのみで計算 ⇒ 計算量少ない

ARモデルの推定法

・パラメータの推定　⇒　時系列データの確率密度関数
・パラメータの数が多い　⇒　計算量多い　⇒　近似計算

$$P\left(x_t \mid x_{t-k}^{t-1}, A_1, \cdots, A_k, \mu, \Sigma\right)$$

k：AR次数
$j:(j=1,\cdots,k)$

■ ユールウォーカー法

定義

$$\mu = \frac{1}{t-k}\sum_{i=k+1}^{t} x_i$$

バッチ学習

$$C_j = \frac{1}{t-k}\sum_{i=k+1}^{t}\left(x_i-\mu\right)\left(x_{i-j}-\mu\right)^{\mathrm{T}}$$

ユールウォーカー方程式

$$C_0 = \sum_{i=1}^{k} A_i C_i + \Sigma \quad \cdots ①$$

$$C_j = \sum_{i=1}^{k} A_i C_{j-i} \quad \cdots ②$$

SDARアルゴリズムの推定法（1）

$$I = \sum_{i=1}^{t} (1-r)^{t-i} \log P\left(x_i \mid x^{i-1}, A_1, \cdots, A_k, \mu, \Sigma\right)$$

■ I を最大にするパラメータを推定

⇒ 最尤推定法の変形

⇒ 重みが時間 t によって指数的に減少

■ 過去のデータの影響を割り引く

忘却パラメータ ＝ $r\ (0 < r < 1)$

⇒ 過去の統計量ほど重みを減らす

⇒ リアルタイムで特徴をうまく抽出

⇒ 非定常なデータに対応

SDARアルゴリズムの推定法(2)

$$I = \sum_{i=1}^{t} (1-r)^{t-i} \log P\left(x_i \mid x^{i-1}, A_1, \cdots, A_k, \mu, \Sigma\right)$$

$$\mu := (1-r)\mu + r x_t$$
$$C_j := (1-r)C_j + r(x_t - \mu)(x_{t-j} - \mu)^{\mathrm{T}}$$

忘却パラメータ r を適用

ユールウォーカー方程式②より A_j 計算

$$C_j = \sum_{i=1}^{k} A_i C_{j-i}$$

$$\hat{x}_t := \sum_{i=1}^{k} A_i (x_{t-k} - \mu) + \mu$$
$$\Sigma := (1-r)\Sigma + r(x_t - \hat{x}_t)(x_t - \hat{x}_t)^{\mathrm{T}}$$

逐次計算を行う

- 確率が高い ⇒ 予測した値と近い
- 確率が低い ⇒ 予測した値と外れている

外れ値検出

SDARアルゴリズムで学習した確率密度関数を利用

x_t が入力

確率密度関数 P_t

- 確率が高い ⇒ 予測した値と近い
- 確率が低い ⇒ 予測した値と外れている

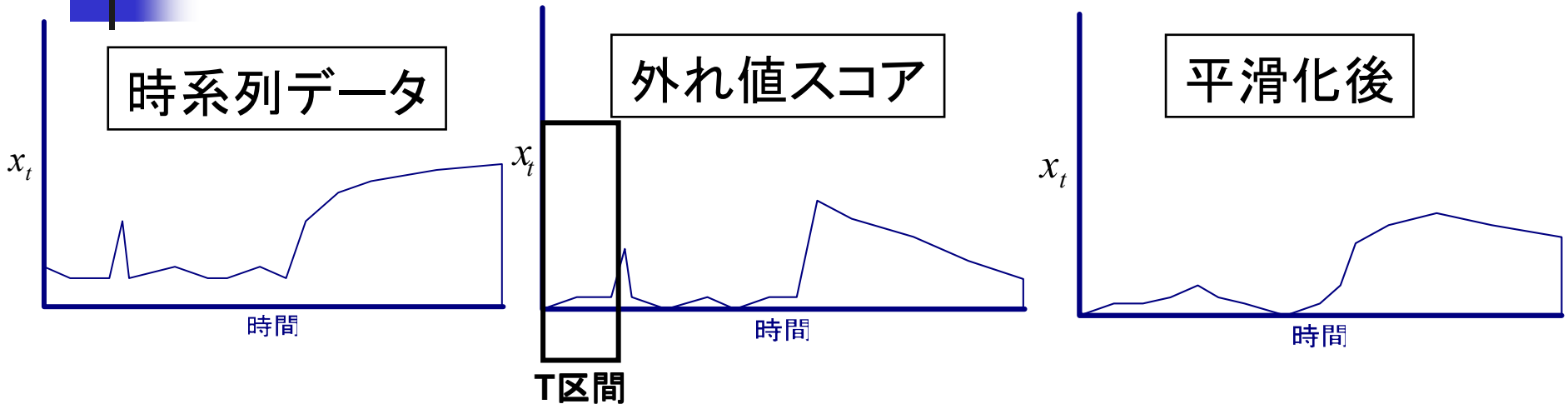
⇒ 外れ値である可能性が高い

$$\mathrm{Score}(x_t) = -\log P_{t-1}\left(x_t \mid x^{t-1}\right)$$

外れ値スコア　　対数予測損失

⇒ スコアが高いほど、外れ値である可能性が高い

外れ値スコアの平滑化



新しいデータ毎に、その時点からT時点前までの平均をとる
⇒ T区間をずらすイメージ

平滑化後の外れ値スコア ＝ 時系列データ
⇒ SDARアルゴリズムで再度学習

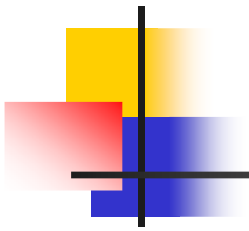
⇒ 外れ値らしさ ⇒ スコアリング ＝ 変化点スコア

変化点検出

- 変化点検出を外れ値検出に還元
- 2段階の学習と平滑化
  - 外れ値の影響を緩和
  - 外れ値による変化点の誤検出低減

⇒ 外れ値と変化点を同じ枠組みで扱うことができる

⇒ 外れ値を先に検出し、それをもとに変化点を検出



目次

- 紹介手法（ChangeFinder）の概要
- 2段階学習アルゴリズム
- 実験
  - 実データによる実験
- まとめ

実データによる実験(1)

■ ChageFinderをネットワークセキュリティに適用

サーバのPort135に対する接続頻度時系列
期間(2003年7月31日午後3時～8月26日午後2時)
⇒ MS.Blastウィルスが流行し始めた時期

忘却パラメータ $r = 0.02$
平滑化パラメータ $T = 5$

従来手法[2](統計モデルの比較手法)と比較
⇒ 精度とリアルタイム性の比較
Trend Micro社のレポートと比較(※参考)
⇒ リアルタイム性の確認

実データによる実験（2）

Trend Micro社のレポート

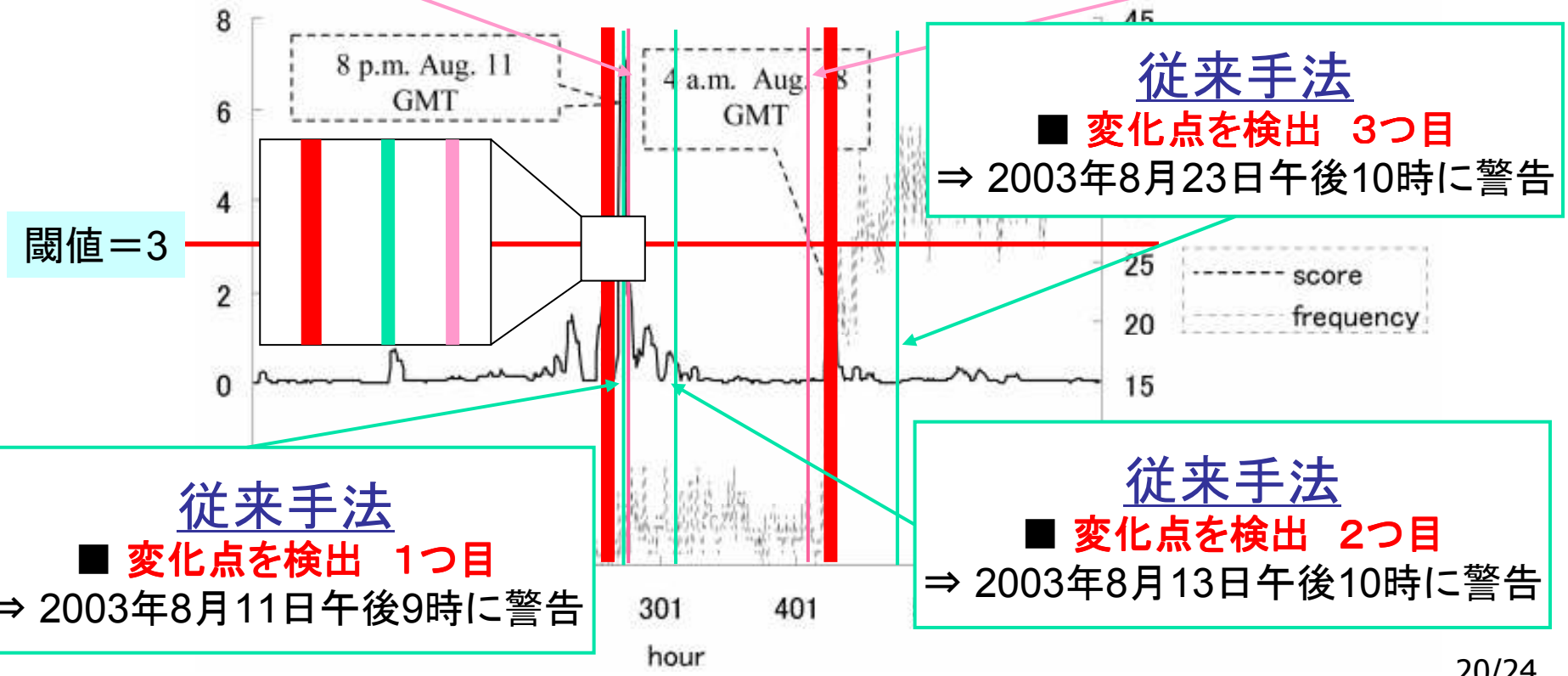
■ **MS.Blastウィルス　第1発生**

⇒ 2003年8月11日午後9時44分に発表

Trend Micro社のレポート

■ **MS.Blastウィルス　第2発生**

⇒ 2003年8月17日午後5時31分に発表



従来手法

■ **変化点を検出　3つ目**

⇒ 2003年8月23日午後10時に警告

従来手法

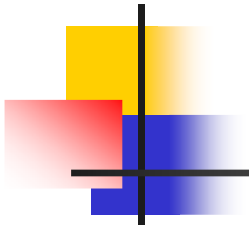
■ **変化点を検出　1つ目**

⇒ 2003年8月11日午後9時に警告

従来手法

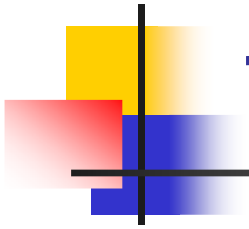
■ **変化点を検出　2つ目**

⇒ 2003年8月13日午後10時に警告



目次

- 紹介手法（ChangeFinder）の概要
- 2段階学習アルゴリズム
- 実験
- まとめ



まとめ

- 外れ値と変化点の統一的検出手法の紹介
  - 変化点検出を外れ値検出に還元
  - 忘却型学習により非定常な時系列データに対応
  - 2段階学習によって外れ値による誤検出を低減

■ 特徴

⇒ 従来手法よりも計算量の面で非常に効率がよい

⇒ 精度も同程度

■ 問題点

閾値やパラメータ（忘却係数、平滑化係数）の経験的決定

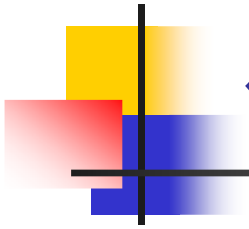
考察と今後の課題

■ 外れ値スコアの平滑化パラメータの適応的決定
⇒ 経験的に決めた値でうまくいく原因追求
⇒ 時系列データに対するパラメータの最適化

■ ARMAモデルの採用
⇒ ARモデルをさらに一般化したモデル
⇒ 時系列データに対してより柔軟に対応できる
⇒ 検出精度の向上

■ マルコフモデルとの融合
⇒ 紹介論文においてデータは全て連続値であると仮定
⇒ より一般的なデータを扱うために離散値に対応
⇒ 連続値と離散値を同時に扱うことができるモデル

ありがとうございました。



付録（１）

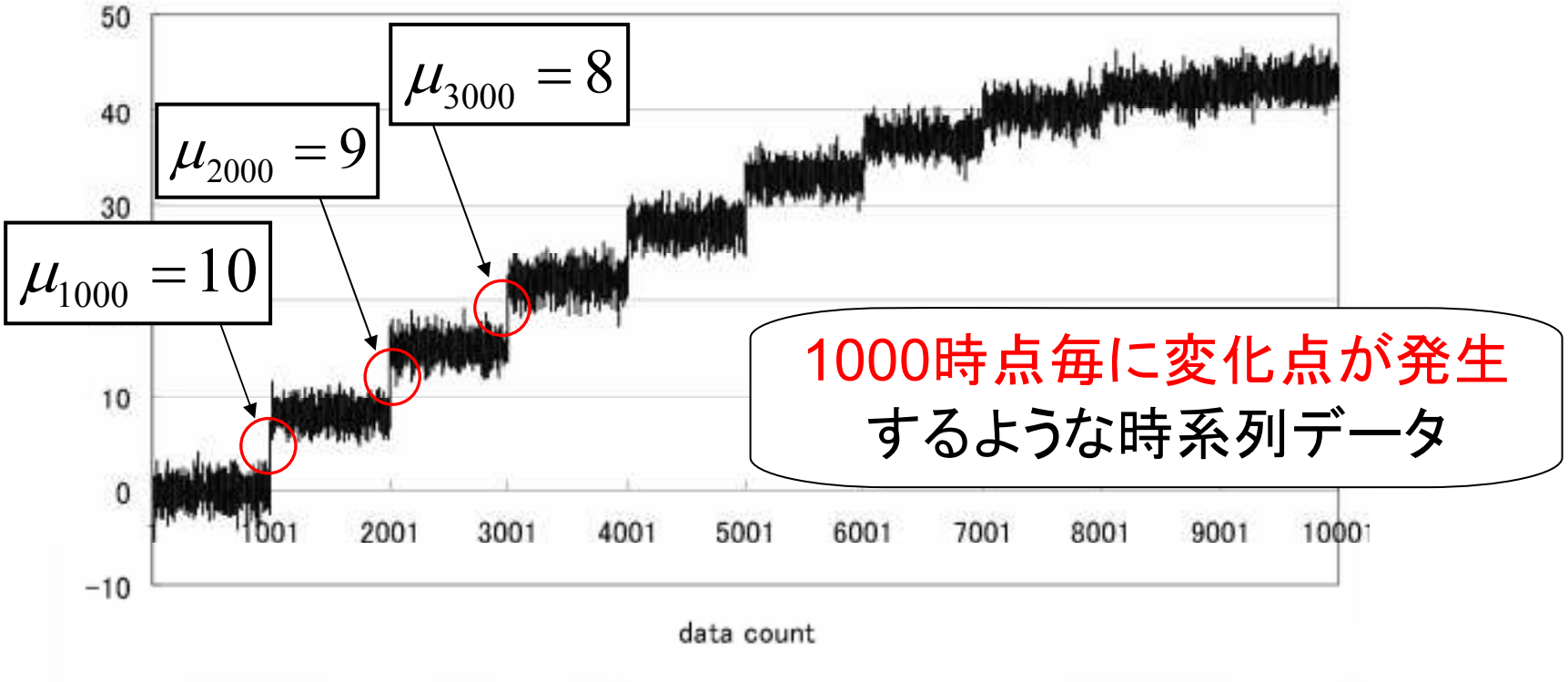
数値シミュレーション

数値シミュレーション（1）

$$x_t = z_t + \mu_t \qquad z_t = 0.6z_{t-1} - 0.5z_{t-2} + \varepsilon_t$$

初期値0（1000時点で10、その後1000時点毎に1減少）

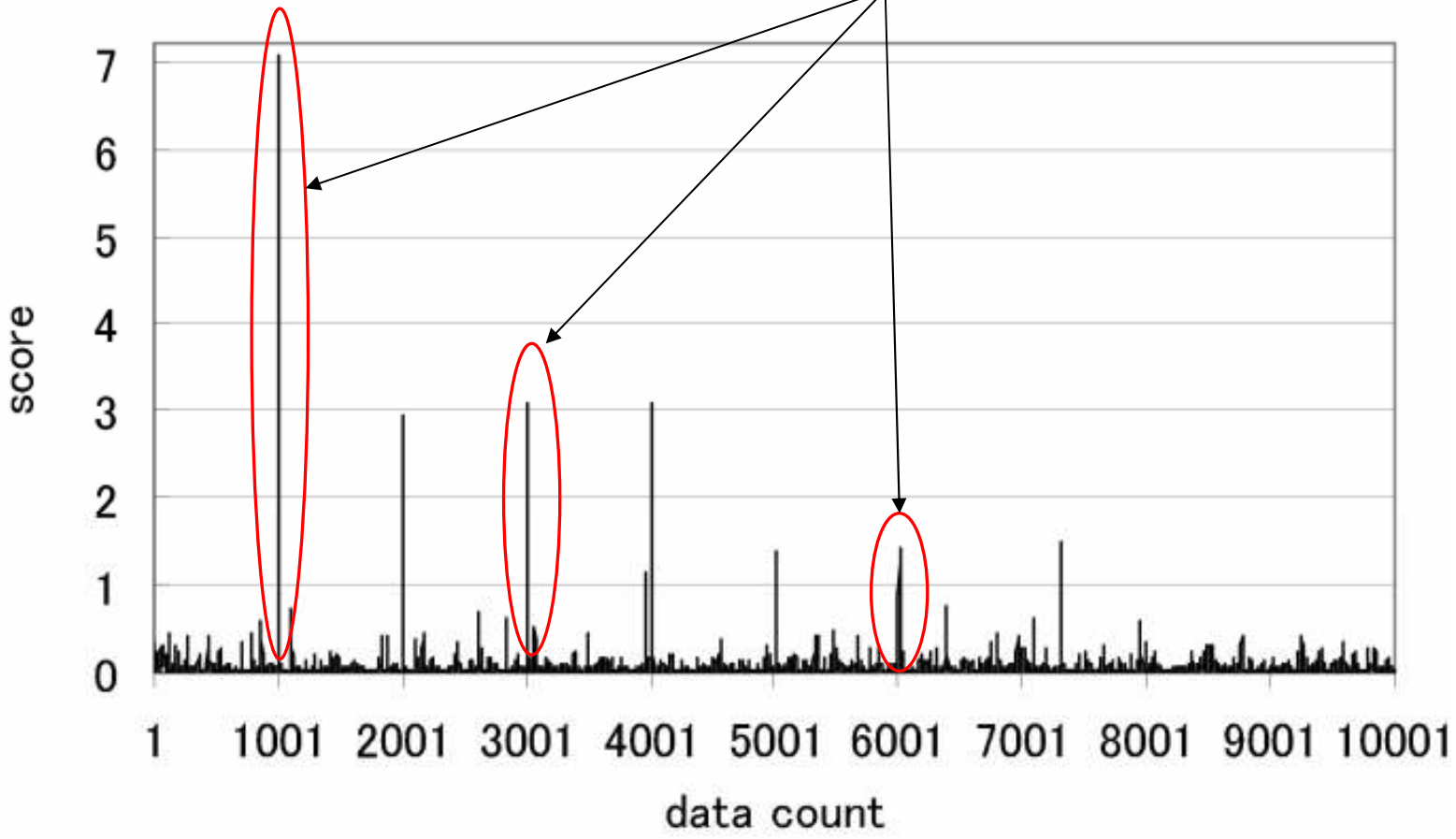
独立ガウスノイズ



1000時点毎に変化点が発生するような時系列データ

数値シミュレーション（1）

変化度合いがスコアに反映されている



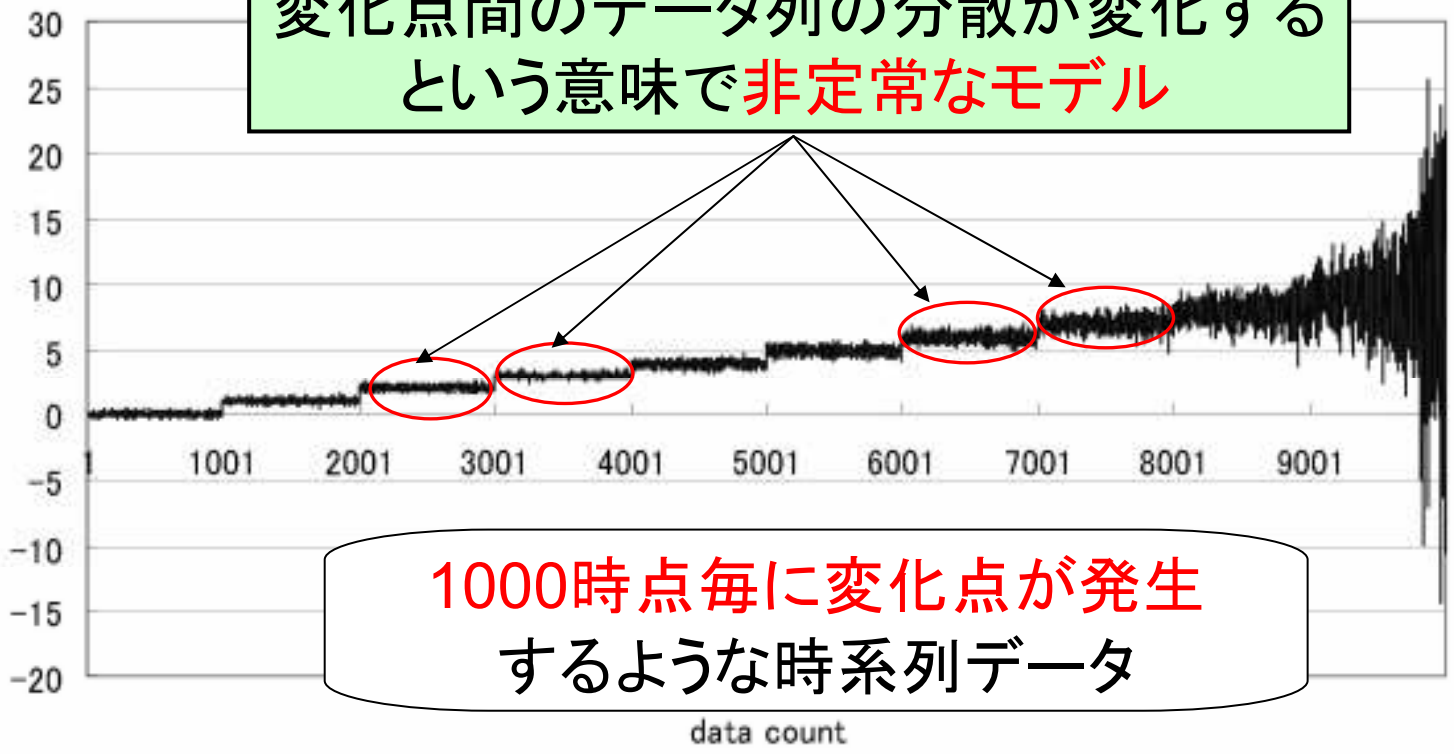
数値シミュレーション（2）

$$x_t = z_t + \mu_t \qquad z_t = 0.6z_{t-1} - 0.5z_{t-2} + \varepsilon_t$$

初期値0（1000時点毎に1増加）

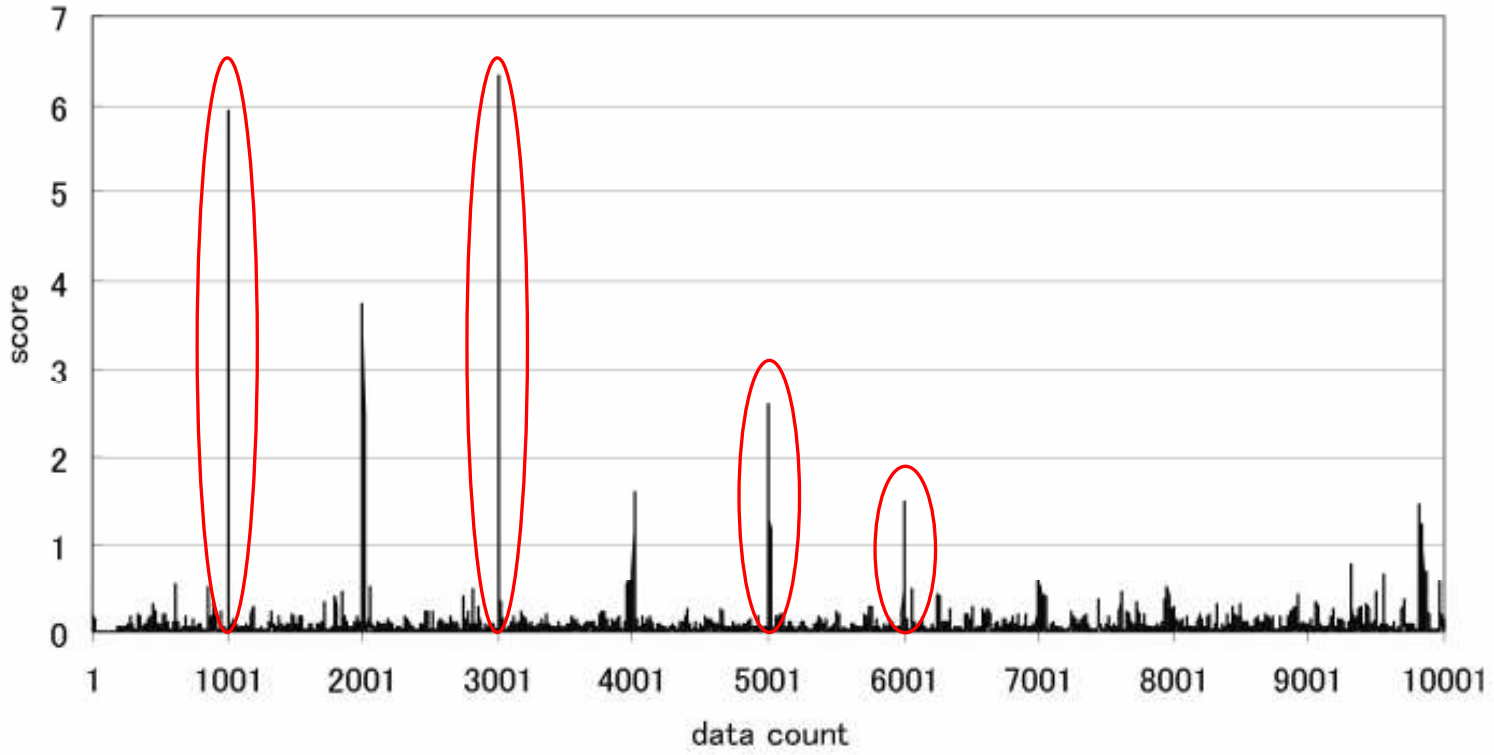
標準偏差を$1000 \times t$となるように変化させたノイズ

変化点間のデータ列の分散が変化する
という意味で非定常なモデル

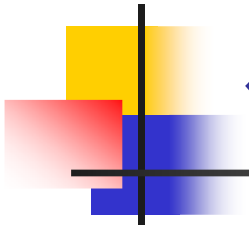


1000時点毎に変化点が発生
するような時系列データ

数値シミュレーション（2）



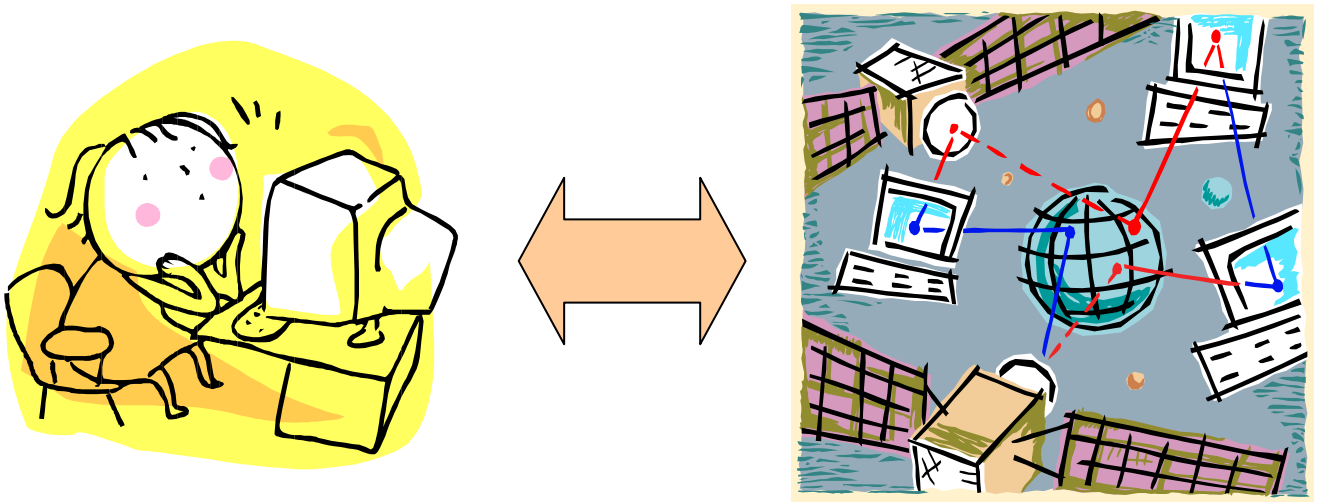
非定常なデータでもシミュレーション（1）のような
定常な場合とほぼ同様の結果が得られている



付録（2）

研究背景

研究背景



- 計算機・ネットワークは社会に必要不可欠
- ウィルスなどの発生　⇒　セキュリティが重要

研究背景

- 現実には様々な被害を受けている
- ネットワークセキュリティの確保
  - インシデント分析システムへの注目

⇒ 各種ログデータからのインシデント候補を検出

リアルタイム検知
（インシデントを検出しアラートを出す）

⇒ ネットワーク管理者の監視を効率化

リアルタイム検知の研究

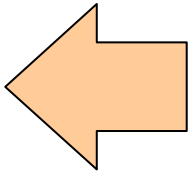
研究：時系列データからの外れ値特定による変化点検出

ネットワークセキュリティ

・不正アクセス
・ネットワーク障害
・サイバーテロ

↓

変化点



アクセス頻度の
時系列データ

■ 外れ値と変化点検出をセキュリティに利用

⇒ 変化点検出によってウィルス被害等を防ぐ

⇒ データマイニングの観点からセキュリティに応用

目的(1)

■ ネットワークセキュリティの課題

⇒ ワーム等の集中アタック

⇒ それに伴うアクセス頻度の急激な変化の検出

⇩

時系列データの変化点検出問題ととらえる

■ 要件

⇒ リアルタイムで検出可能

⇒ 時系列データの性質変化にも柔軟に対応

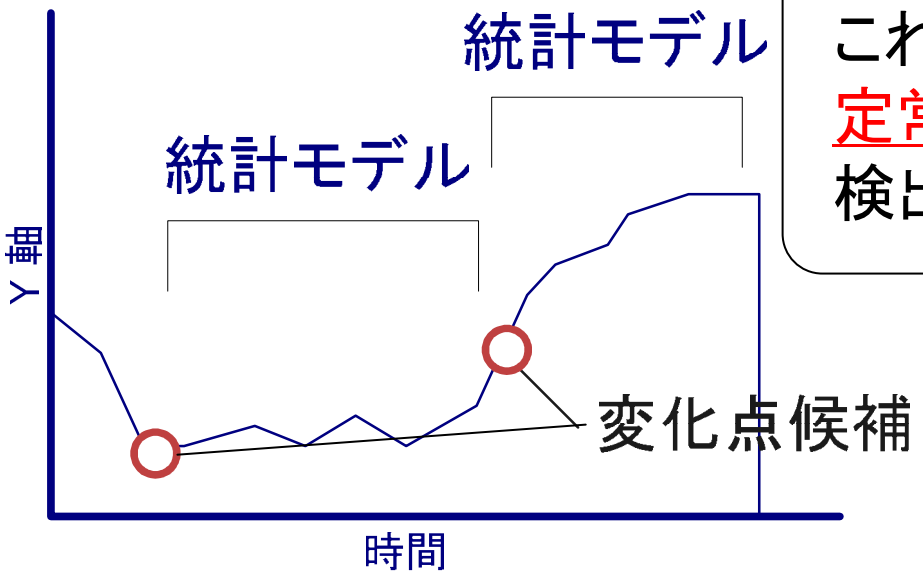
⇒ 非定常な時系列データに対応

目的(2)

■ 従来の手法

⇒ 変化点候補の前後の統計モデルの違いを検定する手法

・リアルタイムで検出できない
・非定常な時系列データに柔軟でない



これらの問題を克服し、定常・非定常に関わらず、高速に変化点検出を行う手法

ChangeFinder

付録（3）

変化点スコアの計算法

変化点検出(1)

■ 平滑化後の外れ値スコアをSDARで再度学習

⇒ 時系列データ(外れ値スコア)に対する確率密度関数

■ 新しい外れ値スコアに対して

確率低い ⇒ 急激な変化である可能性

確率高い ⇒ ゆるやかな変化である可能性

変化点である可能性が高い

この確率をスコアにしたもの ＝ 変化点スコア

$$\mathrm{Score}(y_t) = -\log Q_{t-1}\left(y_t \mid y^{t-1}\right)$$

変化点スコアが高い ⇒ 変化点である可能性が高い