2007 年 6 月 12 日(火)
1031060322 松本　亮介

変化点検出エンジンを利用した インシデント検知システムの構築

0 はじめに

※この資料は，参考文献を元に，参考文献の概略を作成したものです. この研究を私が実際に行ったわけではありませんので注意してください.

1 まえがき

情報通信研究機構（NICT）セキュリティ高度化グループでは，インターネット上で起こるセキュリティインシデントを分析する「インシデント分析システム」の構築を行っている．それに関する重要な要件の 1 つに，各種ログデータからのインシデント候補のリアルタイム検知がある．これを実現するために，時系列データの急激な増加などを高速高精度に検出する変化点検出エンジン ChangeFinder を応用した検知システムを構築する．検知機能には複数の異なる手法を適用するが，本稿では時系列データの変化点をリアルタイムに検出する変化点検出エンジンを応用したシステムについて述べる.

2 手法

2.1 インシデント分析

インシデント分析システムは以下の要件を満たすことを目標としている.

(1) 日本全域にまたがる ISP やエッジユーザなどからのイベントを分析対象とすること.
(2) イベントデータの収集から, インシデント候補のピックアップまでのタイムラグを 2 分以内とするリアルタイム性を確保すること.
(3) 上記実時間分析と並行し，詳細分析を行うこと.
(4) インシデントの検知のみでなく, 検知したインシデントに対する有効な対策を導出すること.

これらの要件に要求される機能は，日本全域に仕掛けられたイベント収集装置が集めるログデータをリアルタイムに分析し，インシデントの候補となる異常を，逸早く発見することであり，多量のログデータのリアルタイム処理というデータマイニングの課題として捉えることができる．この考えに基づき，変化点分析システムを，汎用の変化点検出エンジンとして開発された ChangeFinder を主要コンポーネントとして構築している.

2.2 ChageFinder

ChangeFinder の機能は，数値データの時系列を入力とし，各時点に対応して変化点らしさを示す「変化点スコア」を出力するものである．これは，イベント収集装置が集める各種ログを，数値的な統計量に集計する前処理を行う．つまり，人間が観察するのが不可能なほど多数の統計量を並列で処理し，インシデントの候補となる変化の発生を知らせることができる．以下に ChangeFinder のアルゴリズムの原理を示す.

第一段階学習では，まず各時刻 t において，AR モデルを SDAR アルゴリズムによって学習する. SDAR モデルとは，過去の情報を少しずつ忘却しながら学習する学習手法である．学習した密度関数を $P_t(x)$ とする. 各時刻 t についてデータ x_t の外れ値スコアを対数損失 $-\ln q_{t-1}(x_t)$ として計算する.

次に，平滑化では，一定サイズ T のウィンドウ内のデータの外れ値スコアの平均を計算し，ウィンドウをスライドすることによって移動平均スコアの時系列 $y_t : t = 1,2,\ldots$ を構成する．以下に外れ値スコアの平均を求める式を書いておく.

$$y_t = (1/T)\left(\sum_{i=t-T}^{t-1}(-\ln q_i(x_i))\right)$$

第二段階学習では，この時系列 $y_t : t = 1,2,\ldots$ に対して，再度 AR モデルをあてはめ，これを学習して，その系列を $q_t(x)$ とする．各時点 t において対数損失 $-\ln q_{t-1}(y_t)$ を時刻 t の変化点スコアとして計算する. 変化点スコアが大きいほど，t が変化点である度合いが高い.

ChangeFinder の特徴は，第一段階学習では時系列中の外れ値しか検出できないところを，外れ値スコ

アの平滑化を通じて，本質的なモデルの変動を検出しているところにある．計算量に関しても，データ数 n に対して，統計的検定に基づく方式が $O(n^2)$ であるのに対して，ChangeFinder の計算量は $O(n)$ で済むため，明らかに効率がよいことがわかる．

さらに, ChangeFinder は平均値の変化だけでなく，AR モデルのパラメータ（AR 係数や分散）の変化も原理的には検出できる．実際に，分散が突然変化する場合でも，十分な効果が得られるという報告がある．

2.3 AR モデルとその学習

ここでは，ChangeFinder で用いた時系列モデルである AR モデルについて説明する．AR モデルとは，標準的な時系列モデルの 1 つである．

今，期待値が 0 であるような定常時系列 $\{z_t : t = 1,2,\ldots\}$ が k 次の AR モデルに従って発生すると仮定すると，

$$z_t = \sum_{i=1}^{k} A_i z_{t-1} + \varepsilon$$

で表される．ただし，データ z_t は n 次元のベクトル，$A(i)(i = 1,\ldots,k)$ は n 元正方行列，ε は期待値 0，共分散行列 Σ のガウス分布 $\mathrm{N}(0,\Sigma)$ に従うノイズ項であるとする．実際に観測されるデータを $x_t = z_t + \mu$ で表す．これより，期待値が μ であるとすると， x_t の確率密度関数は， $x_{t-k}^{t-1} = (x_{t-1} \cdots x_{t-k})$ とするとき，

$$p(x_t \mid x_{t-k}^{t-1} : \theta) = \frac{1}{(2\pi)^{k/2}|\Sigma|^{1/2}} \exp\left(-\frac{\xi^T \Sigma^{-1} \xi}{2}\right) \quad (1)$$

で与えられる．ただし，

$$\xi = x_t - \left(\sum_{i=1}^{k} A(i) z_{t-i} + \mu\right)$$

であり， $\theta = (A_1,\ldots,A_k,\mu,\Sigma)$ とおいた．

AR モデルに関する通常の推定アルゴリズムについて，以下の量を定義する．

$$\hat{\mu} = \frac{1}{t-k} \sum_{i=k+1}^{t} x_i \quad (2)$$

$$C_j = \frac{1}{t-k}(x_i - \hat{\mu})(x_{i-j} - \hat{\mu})^T \quad (3)$$

(2)は μ の推定値，(3)は $x_1,\ldots,x_t$ の相関関数の推定値を表す．さらに $A(i)$ の推定値は，以下の $\overline{A}_i$ を未知数とする連立方程式を解くことで得られる．

$$C_j = \sum_{i=1}^{k} \overline{A}(i) C_{j-i} \quad (j = 1,\ldots,k) \quad (4)$$

(5) の解を $\hat{A}(i)$ とおくと， Σ の推定値は，

$$\hat{\Sigma} = C_0 - \sum_{i=1}^{k} \hat{A}(i) C_i \quad (5)$$

によって求めることができる．しかし，この手続きでは，情報源が定常であると仮定されており，いわゆるバッチ学習方式になっている．

2.4 SDAR アルゴリズム

SDAR(sequentially discount-ing AR model estimating) アルゴリズムとは, バッチ学習方式の AR モデルを改良した，逐次型学習方式である．SDAR アルゴリズムでは，逐次学習と忘却機能という 2 つのポイントがある．

逐次学習とは，新たなデータを 1 つ読み込むごとにパラメータの推定値を更新する．

忘却機能とは， i 時点前のデータの影響が $(1-r)^i$ 倍に減少するようにパラメータの推定値を更新する．これによって，非定常な情報源に対応できる．アルゴリズムのパラメータ r を忘却パラメータと呼び，$1/r$ 個程度の過去データの情報を蓄積するようにする．以下に SDAR アルゴリズムをまとめておく．

SDAR アルゴリズム $(0 < r < 1$: 所与$)$

STEP 1. 初期化

Set $\hat{\mu}, C_j, \hat{A}(j)(j = 1,\ldots,k), \hat{\Sigma}.$

STEP 2. パラメータ更新

For $t = 1,2,\ldots,$

x_t を読み込む:

$$\hat{\mu} := (1-r)\hat{\mu} + r x_t$$

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

以下の連立方程式を $A(i)$ について解く:

$$C_j = \sum_{i=1}^{k} A(i) C_{j-i} \quad (j = 1,\ldots,k). \quad (6)$$

方程式(6)の解を $\hat{A}(1),\ldots,\hat{A}(k)$ とし，以下を計算

$$\hat{x}_t := \sum_{i=1}^{k} \hat{A}(i)(x_{t-k} - \hat{\mu}) + \hat{\mu}$$

$$\hat{\Sigma} := (1-r)\hat{\Sigma} + r(x_t - \hat{x}_t)(x_t - \hat{x}_t)^T$$

本稿では，このアルゴリズムにおいて t 番目のデータまで用いて得られる確率密度関数(1)を，p_t と書く．

3 変化点検出による検知システム

2 で説明した変化点検出エンジン ChangeFinder を用いて，実際に変化点検出によるインシデント検知システムを構築する．このシステムは，以下の４つのモジュールから構成されている．

- インシデント分析マネージャ
- 分析プリプロセッサ
- ChangeFinder
- 分析ポストプロセッサ

分析プリプロセッサと ChangeFinder の処理を，合わせて分析ラインと呼ぶことにし，規定の分析パターンで変化点検出を行うデフォルト分析ラインと，新規の分析パターンを動的に追加して分析するための，派生分析ラインを並列で実行することにより，既知のインシデントと未知のインシデントの両方を同時に検知可能となっている．簡単にそれぞれのモジュールの説明をする．

インシデント分析マネージャは，分析対象となるデータを上位のシステムから受信し，分析ラインにデータを引き渡す機能を持つ．さらに，振る舞い分析も行うが，アクセス状況を複数の振る舞いパターンに分類する機能を持つ．これらのパターンの発生頻度を時系列として変化点分析にかけることができる．

分析プリプロセッサは，分析対象データを加工し，ChangeFinder を起動する機能を持つ．まず，上位システムから受信したデータから，ChangeFinder で用いるデータを抽出する．次に，データ抽出によってデータ系列化された情報を，変化点検出を行うための一定時間毎の統計量を作成する．そして，作成された複数の時系列の基礎統計量について，機知のインシデントのいくつかのケースについては，その数値に相関がある事が事前に判明している場合がある．そのような場合に，本処理で関連する複数のパラメータを組み合わせて，ChangeFinder にまとめて受け渡す．

ChangeFinder は，渡されたデータに対して，2 で説明したような処理を行い，分析結果を分析ポストプロセッサに渡す．

分析ポストプロセッサは，ChangeFinder が求めた変化点スコアに対して，一定の基準を設定し，インシデントの判定を行う．ワームによるパケットパターンの増加だけでなく，単にユーザ間の通信が増加したケースによって，変化点スコアが上昇することもある．その場合のために，インシデントの出力として変化点スコアをそのまま利用するのではなく，閾値による判定や，各スコアの組み合わせなどにより，総合的に判断する．

以上のシステム構成により，新規のパケットパターンが発生した場合に，そのパターンに該当するアクセス数が一定の期間内にまったく変化しない，あるいは急激に増加するなどの変化点を検出することで，未知のインシデントを早期に発見することができる．

4 ChangeFinder の検証

ChangeFinder が実際にどれほどの精度と速度を持っているかを検証するために，某組織のメールシステムログの監視と運用を行っているサイバーテロ対策チームで実証実験を行った．そのチームでは，未知ワーム対策が以前からの課題であったが，決定的な有効手段が見つかっておらず，人海戦術のようなもので対応してきていた．そのチームから特定の MessageID を有するメールの通数（1 分あたりのメールの流通量）の時系列データを，ChangeFinder を適用して，変化点を調べた．グラフを見ると，7 月 13 日に変化点スコアが異常に高くなり，アラームが上がった．実際，この日には LOVGATE と呼ばれる新種のウィルスが発生し，メールサーバの一時停止といった障害を引き起こしていた．このウィルスは受信メール内のメッセージに自身のコピーを添付して返信するワーム型ウィルスで，トラフィックの急増をもたらしていた．この日の現場が LOVGATE を発見した時刻が 10:00am であったのに対し，ChangeFinder は，8:54am にこれを検出した．Change

Finder は，7-8 月の間，他にもいくつかのアラームを上げている．その全てが，ワームやメールの大量発信を知らせる意味のアラームであった．この実証実験により，ChngeFinder の有効性が実証されている．

参考文献

[1] 竹内純一 “変化点検出エンジンを利用したインシデント検知システムの構築”