Takeuchi, J., and Yamanishi, K.: A Unifying Framework for Detecting Outliers and Change Points from Time Series, *IEEE Trans. on Knowledge and Data Engineering*, 18(4), pp.482–492, 2006.

2013年度論文ゼミ#9
20130621
中西 航

東京大学
大学院工学系研究科
社会基盤学専攻

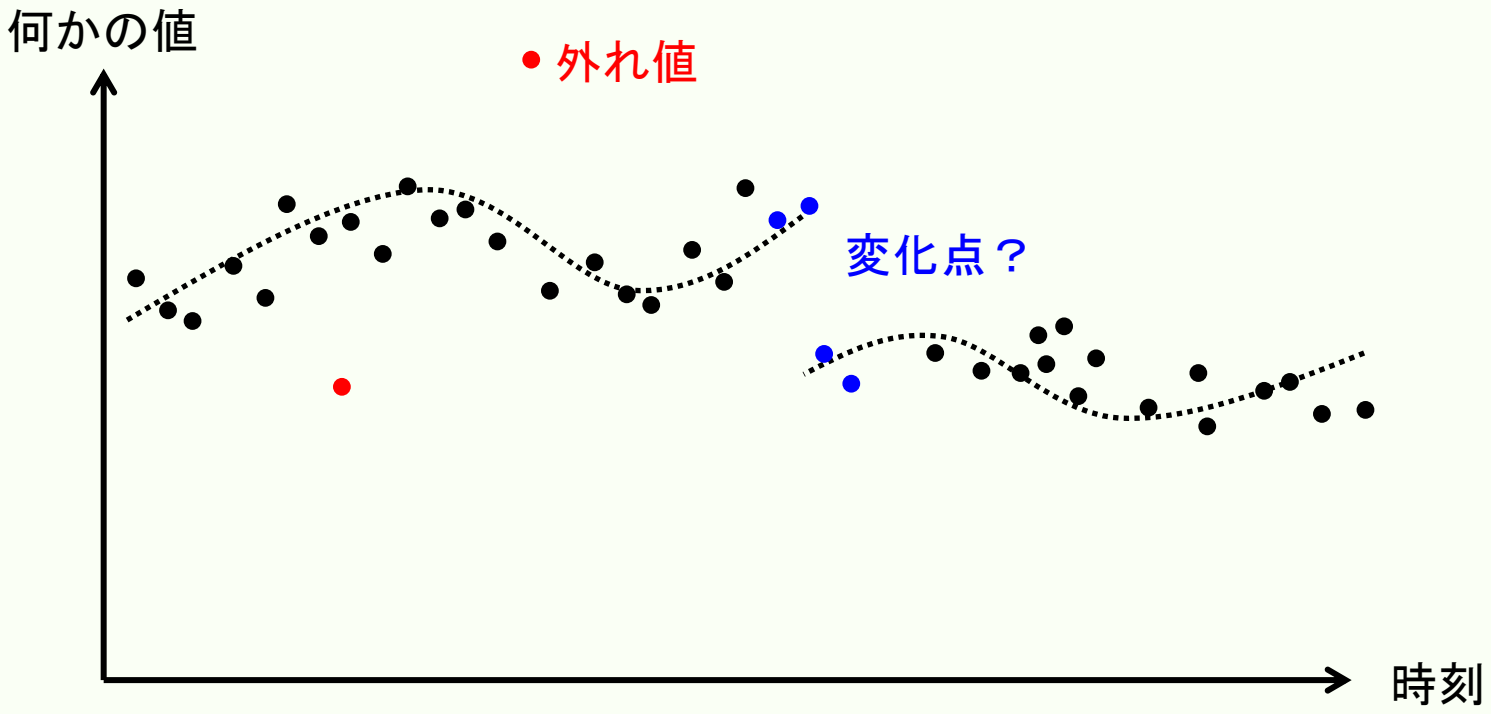
論文全体の構成

1. Introduction
2. Problem Setting
   ::本論文の主題”Change Point Detection”の位置づけ
3. Two-Stage Learning for Change Point Detection
   ::Change Point Detectionの定式化
4. Other Methods
   ::以下での比較用に別の手法を紹介
5. Simulation
   ::人工データに対する適用と分析
6. Experiment with Real Data
   ::実データに対する適用
7. Concluding Remarks

1.時系列データからの「検出」

- 時系列データの変化には、
  - 外れ値の検出
  - 変化点の検出

の2種類があり、ともに興味を引く問題となってきている



1.時系列データからの「検出」

- 時系列データの変化には、
  - 外れ値の検出
  - 変化点の検出

  の2種類があり、ともに興味を引く問題となってきている

- 例）ネットワークのアクセスログ
  - 通常状態をデータから学習し、
    新たに入ってくるデータがそこからどの程度はずれているか
    ＝外れ値＝異常検出
  - 一方で、膨大な外れ値データが入ってきた場合は、
    統計的に通常状態自体が変化しているととらえたい

- これまで、外れ値検出と変化点検出は別個に行われてきた
  →これを統一した枠組みで記述したい

1.著者らの既往研究との関係

- 従来：統計的な外れ値検出手法(時系列データに未対応)

→これに対し、

- 時系列データに対応
- 変化点の検出にも対応

の2点が本研究の手法。具体的には、

- Auto Regressionモデルを従来手法に導入する
- 外れ値検出と変化点検出を、
  2段階の「検出」ステップに分離し、逐次同時に処理する
  - より具体的には、データ単体の外れ度合いの計算と、
    データの移動平均の外れ度合いの計算を行う
  - この手法をChangeFinderと命名する

1.関連する研究(の課題)

- 外れ値検出は時系列データへの対応が不完全
- 時系列を考慮したARモデルはデータの定常性を仮定

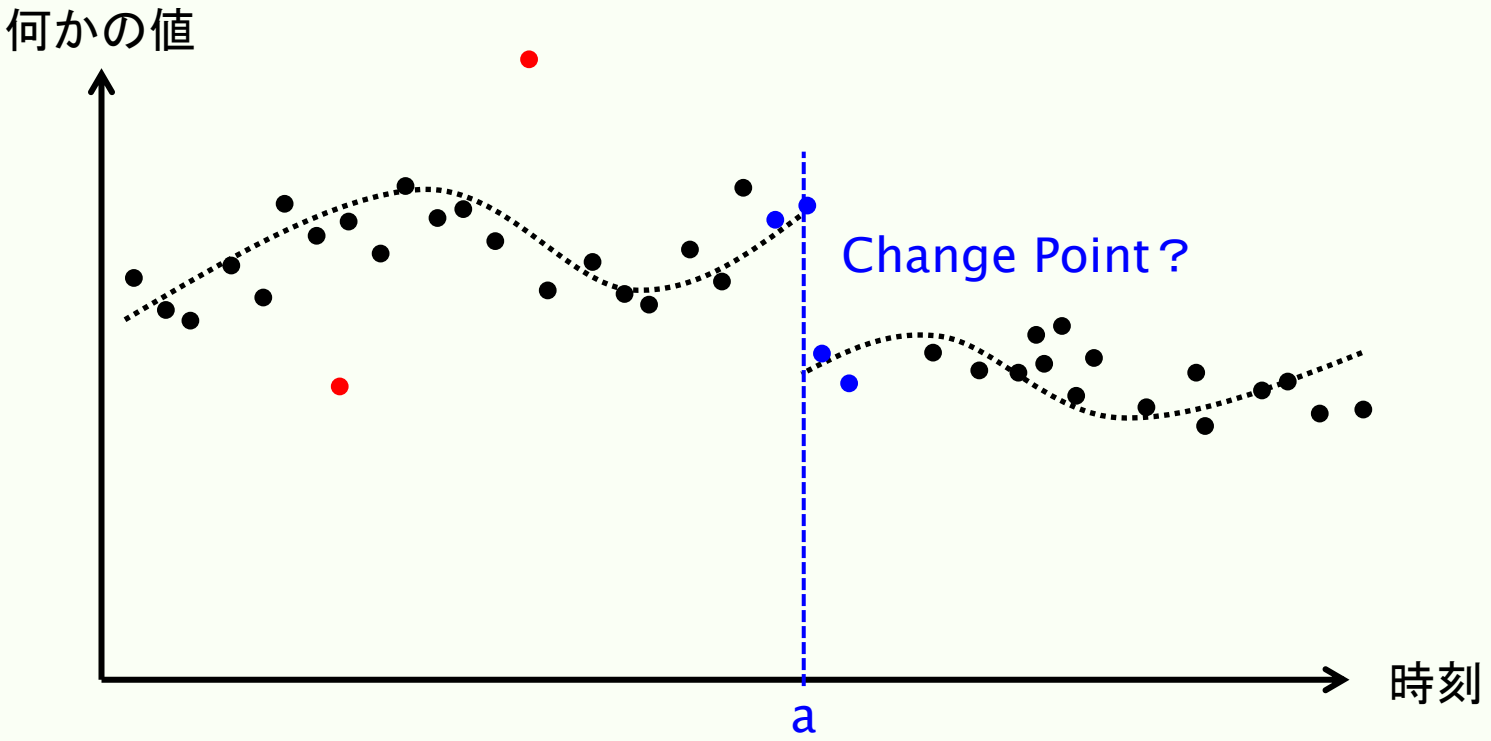
→非定常な時系列データを扱う方法として、
  - ARモデルの(共)分散に時間変化を持たせる方法
  - 割引率の導入による一種の最尤推定(本研究のアプローチ)

- 事前に変化点の個数が分かっていることを前提
- 変化後のデータがある程度蓄積されてから変化点が検出

→より一般化した適用可能性の高い手法の構築

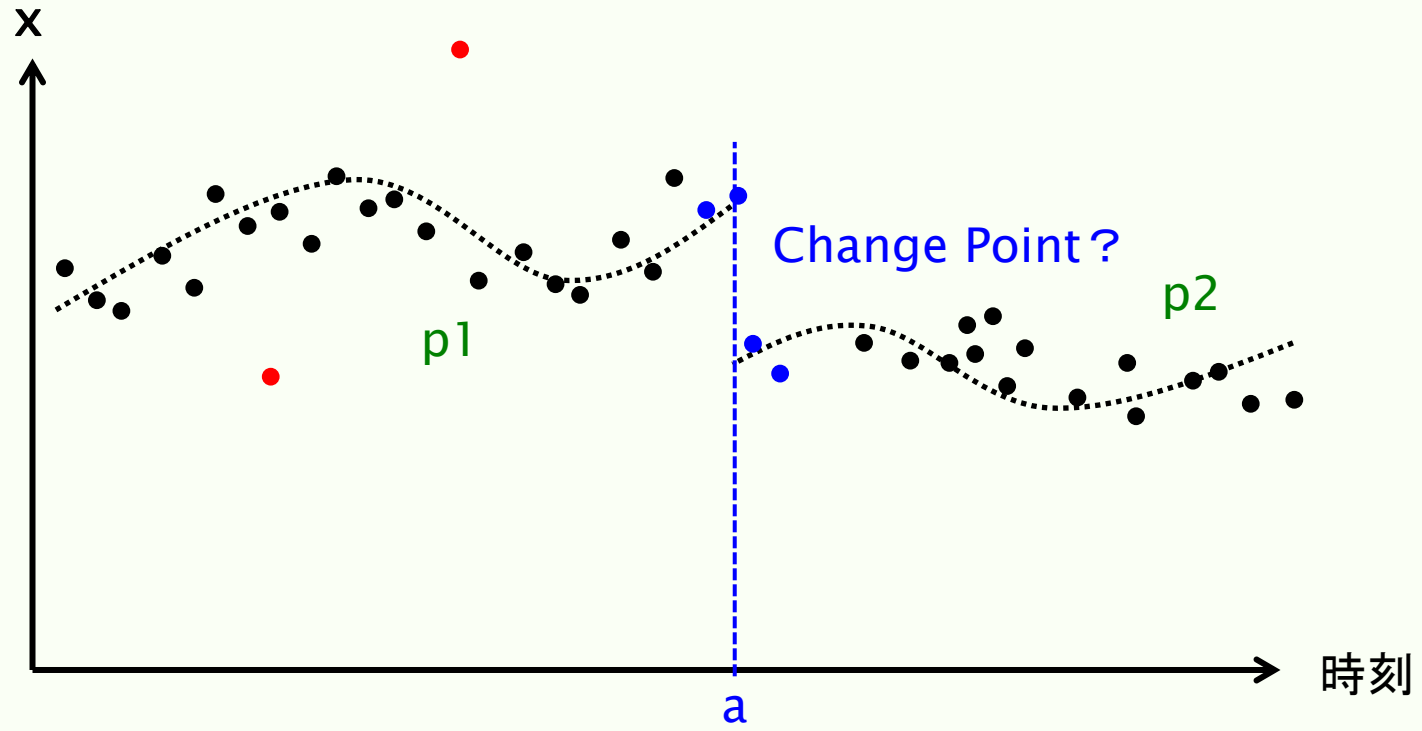
2.Change Point Detectionでは何を行うか

- Change Pointとは、データの属性が急激に変化する時間軸上の点(時刻)のことで、緩やかな変化は対象としない
- この時刻*t*=*a*を検出するのがChange Point Detection
  - (中西注：緩やかな変化に対しては、動的パラメータの逐次推定やファジィシステムの適用が行われているように思われる)



2.Change Point Detectionでは何を行うか

- ある確率過程pに基づいて、
  d次元の実数ベクトル$\mathbf{x}_t$が$t=1$から順次得られる
- $p(\mathbf{x}_t|\mathbf{x}^{t-1})$は$\mathbf{x}_1$から$\mathbf{x}_{t-1}$が得られたもとでの$\mathbf{x}_t$の分布
- このとき、ある時刻$t=a$がChange Pointならば、
  pがある時刻$t=a$を境に$p^{(1)}$と$p^{(2)}$という2つの(異なる)確率密度関数に分離されるという想定が出来る



2.Change Point Detectionでは何を行うか

- このときカルバック・ライブラー情報量

$$D\left(p^{(2)} \| p^{(1)}\right) \equiv \lim_{n\to\infty} \frac{1}{n} E_{p^{(2)}} \ln \frac{p^{(2)}\left(\mathbf{x}^n\right)}{p^{(1)}\left(\mathbf{x}^n\right)}$$

E_p：pの期待値

- たとえば、有意なChange Pointである$t=a$ではDの値がとても大きい、というような想定
- もし$p^{(1)}$と$p^{(2)}$がi.i.d.を満たす確率密度関数ならば、$p^{(1)}$のもとで$\mathbf{x}_{a+1}$から$\mathbf{x}_{a+m}$までの一連の$\mathbf{x}$が得られる確率は$p^{(2)}$のもとで得られる確率の$\exp(-mD)$倍に比例する

- ところで、上記の方法では、唯一の変化点$t=a$の存在と、その前後での$p^{(1)}$と$p^{(2)}$の定常性を仮定している

- これに対し本研究は、複数の変化点に対応し、変化点を出来る限りリアルタイムで検出することを目的としている

2.Change Point Detectionでは何を行うか

- 具体的には以下の2つを扱う
  - Jumping mean(平均の変動)
    $p^{(i)}$がi.i.d.の1次元正規分布で、平均$\mu^{(i)}$、分散ともにσ^2のとき

$$D\left(p^{(2)} \| p^{(1)}\right) = \frac{\left(\mu^{(1)} - \mu^{(2)}\right)^2}{2\sigma^2}$$

    $|\mu^{(1)} - \mu^{(2)}|$が大きいときJumping mean型の変化点

  - Jumping variance(分散の変動)
    $p^{(i)}$がi.i.d.の1次元正規分布で平均ともに0、分散が$\sigma_{(i)}^2$のとき

$$D\left(p^{(2)} \| p^{(1)}\right) = \frac{1}{2}\left(\frac{\sigma_{(2)}^2}{\sigma_{(1)}^2} - 1 - \ln\frac{\sigma_{(2)}^2}{\sigma_{(1)}^2}\right)$$

    $\sigma_{(1)}^2$と$\sigma_{(2)}^2$が大きく異なるときJumping variance型の変化点

3.ChangeFinder

- 2段階学習を行う

x_t: 時系列データ
p: xの時系列変化を表す(条件付き)確率密度関数
y_t: p_{t-1}のもとでのx_tの合致度の移動平均
q: yの時系列変化を表す(条件付き)確率密度関数

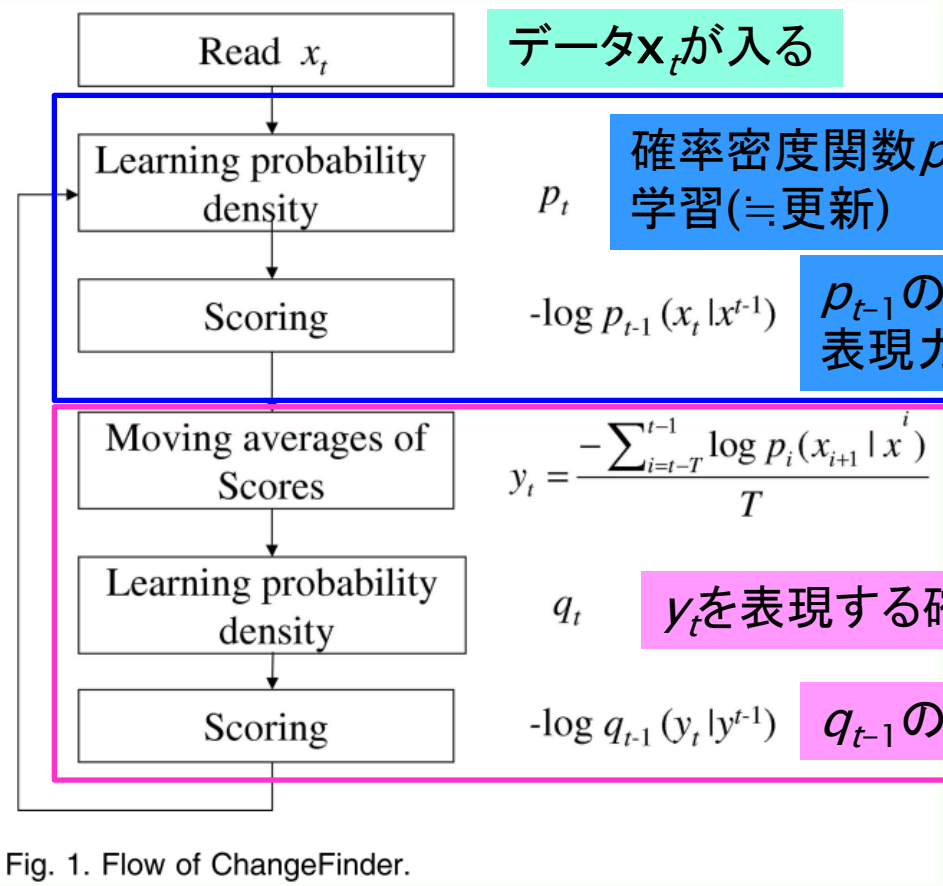
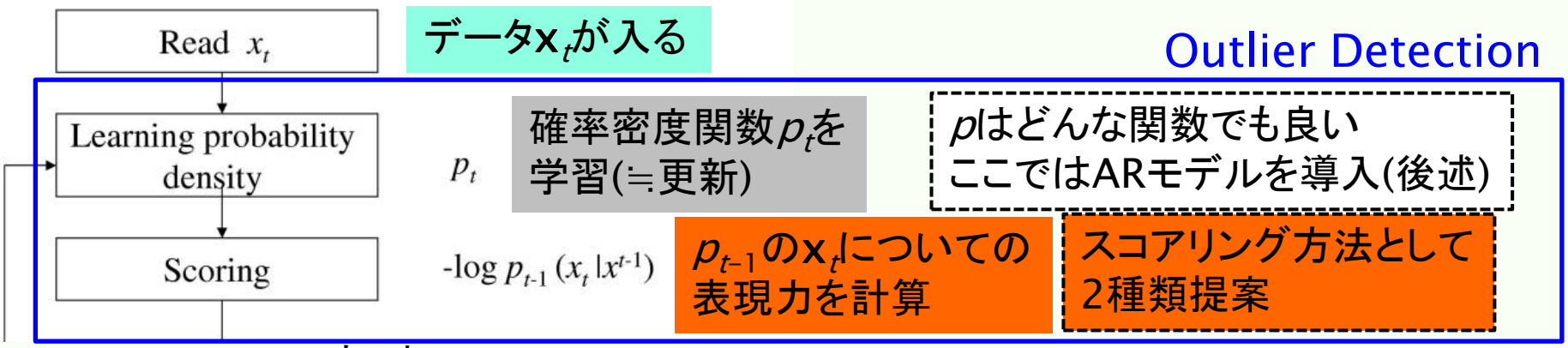


Fig. 1. Flow of ChangeFinder.

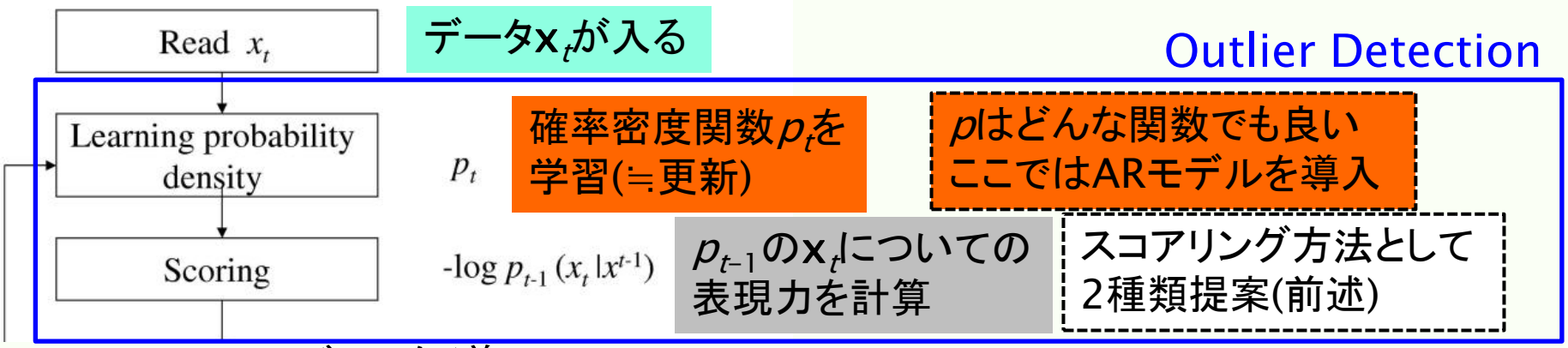
3.Outlier Detection



- Scoringの方法
  1. 対数損失 $Score(\mathbf{x}_t) = -\log p_{t-1}(\mathbf{x}_t \mid \mathbf{x}^{t-1})$
     情報理論の立場では、p_{t-1}に従って生成されるバイナリデータをもとに$\mathbf{x}_t$をエンコードするのに必要な記述長
  2. 二次損失 $Score(\mathbf{x}_t) = (\mathbf{x}_t - \hat{\mathbf{x}}_t)^2, \hat{\mathbf{x}}_t = \int \mathbf{x} p_{t-1}(\mathbf{x}_t \mid \mathbf{x}^{t-1}) d\mathbf{x}$

★*Score*(**x**)が大きいとき外れ値の可能性が高いということ

3.Outlier Detection



- pにARモデルを導入
  初期値の平均が0であるようなd次元ベクトル$\mathbf{z}_t$がk次のARモデルに従う

$$\mathbf{z}_t = \sum_{i=1}^{k} \mathbf{A}_i \mathbf{z}_{t-i} + \boldsymbol{\varepsilon}, \boldsymbol{\varepsilon} \sim N(0, \boldsymbol{\Sigma})$$

$$\mathbf{x}_t = \mathbf{z}_t + \boldsymbol{\mu}$$

- $\mathbf{x}_{t-k}$から$\mathbf{x}_{t-1}$までの系列データを$\mathbf{x}_{t-k}^{t-1}$と表記すれば、

$$p\left(\mathbf{x}_t \mid \mathbf{x}_{t-k}^{t-1} : \boldsymbol{\theta}\right) = \frac{1}{(2\pi)^{k/2} |\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}_t - \mathbf{w})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}_t - \mathbf{w})\right)$$

$$\mathbf{w} = \sum_{i=1}^{k} \mathbf{A}_i (\mathbf{x}_{t-i} - \boldsymbol{\mu}) + \boldsymbol{\mu}, \boldsymbol{\theta} = (\mathbf{A}_1, \ldots, \mathbf{A}_k, \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

3.Outlier Detection

- さきほどの式展開

$$\mathbf{x}_t = \mathbf{z}_t + \boldsymbol{\mu}$$

$$\mathbf{w} = \sum_{i=1}^{k} \mathbf{A}_i \left(\mathbf{x}_{t-i} - \boldsymbol{\mu}\right) + \boldsymbol{\mu} = \mathbf{A}_1 \left(\mathbf{x}_{t-1} - \boldsymbol{\mu}\right) + \mathbf{A}_2 \left(\mathbf{x}_{t-2} - \boldsymbol{\mu}\right) + \ldots + \mathbf{A}_k \left(\mathbf{x}_{t-k} - \boldsymbol{\mu}\right) + \boldsymbol{\mu}$$

↓

$$\mathbf{x}_t - \mathbf{w} = \mathbf{z}_t - \left(\mathbf{A}_1 \mathbf{z}_{t-1} + \mathbf{A}_2 \mathbf{z}_{t-2} + \ldots + \mathbf{A}_k \mathbf{z}_{t-k}\right) = \boldsymbol{\varepsilon} \sim N\left(0, \boldsymbol{\Sigma}\right)$$

$\mathbf{z}_t$についての回帰式

↓

$$p\left(\mathbf{x}_t \mid \mathbf{x}_{t-k}^{t-1} : \boldsymbol{\theta}\right) = \frac{1}{\left(2\pi\right)^{k/2} \left|\boldsymbol{\Sigma}\right|^{1/2}} \exp\left(-\frac{1}{2}\left(\mathbf{x}_t - \mathbf{w}\right)^T \boldsymbol{\Sigma}^{-1} \left(\mathbf{x}_t - \mathbf{w}\right)\right)$$

- Sequential Discounting AR learningによる$\boldsymbol{\theta}$の推定：
  割引率rを導入した以下の式を最大化する$\boldsymbol{\theta}$

$$\sum_{i=1}^{t} \left(1-r\right)^{t-i} \log p\left(\mathbf{x}_i \mid \mathbf{x}^{i-1}, \boldsymbol{\theta}\right)$$

3.Outlier Detection

- Sequential Discounting AR learningによる**θ**の推定量
  「:=」は代入(更新)であり、逐次推定を行うという意味

$$\begin{aligned}\hat{\mu} &:= (1-r)\hat{\mu} + r x_t, \\ C_j &:= (1-r)C_j + r(x_t - \hat{\mu})(x_{t-j} - \hat{\mu})^T \\ &\qquad (j = 0, \cdots, k).\end{aligned}$$

μの推定量

C:計算上設定する変数
「自己共分散関数」と呼ばれる

Solve the following equation (C_{-i} denotes C_i^T):

$$C_j = \sum_{i=1}^{k} A_i C_{j-i} \quad (j = 1, \cdots, k).$$

←Yule-Walkerの方程式
これを解くと
Aの推定量が得られる

Letting the solution to (14) be $\hat{A}_1, \cdots \hat{A}_k$, then calculate

$$\begin{aligned}\hat{x}_t &:= \sum_{i=1}^{k} \hat{A}_i (x_{t-i} - \hat{\mu}) + \hat{\mu}. \\ \hat{\Sigma} &:= (1-r)\hat{\Sigma} + r(x_t - \hat{x}_t)(x_t - \hat{x}_t)^T.\end{aligned}$$

Aの推定量を代入すると
xとΣの推定量が得られる

3.Change Point Detection

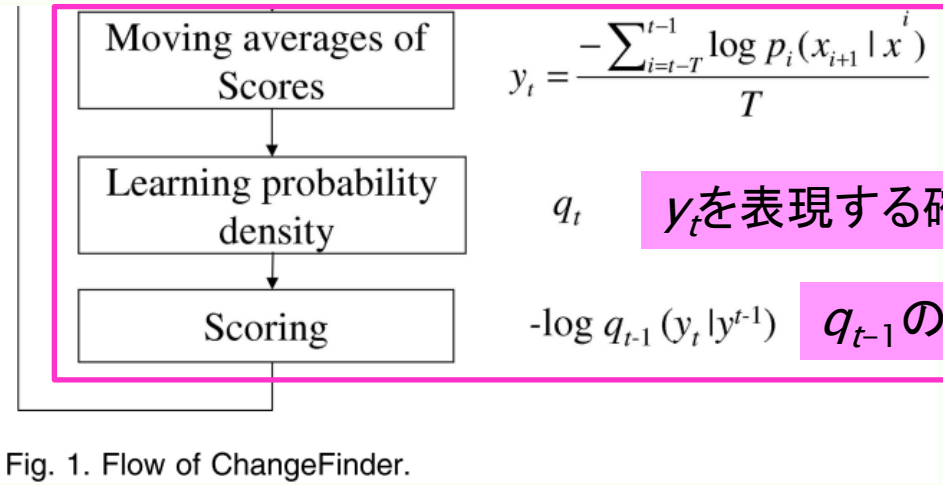


Fig. 1. Flow of ChangeFinder.

Change Point Detection

- *Score*(**x**)を元に新たな時系列データ*y*を生成 $y_t = \frac{1}{T} \sum_{i=t-T+1}^{t} Score(\mathbf{x}_i)$
  - いわゆる移動平均
- 外れ値xの前後でのyの値の変化は小さく、
  変化点tの前後でのyの値の変化は大きいことが想定される
- $\mathbf{x}_t \rightarrow p_t$と同じ要領で、$y_t$から$q_t$を生成する
- p_tと同じ要領でq_tについてScoringし、*Score*(*t*)とする

- T：小→変化点検出高速、外れ値との峻別能力低い
- T：大→変化点検出低速、検出精度高い

4.比較手法

- GS: Guralnik and Srivastava(1999)に基づく手法
  - 誤差(残差)の平方和の最小化
  - GSでは多項式による近似も行っているが今回は線形近似

- SC: GSにARモデルを導入した手法
  - 確率的コンプレキシティ(Stochastic Complexity)最小化
  - 確率密度関数 $p\left(x_t \mid x_{t-k}^{t-1}, \theta\right)$ と系列データ x_u^t が与えられたとき、
    確率的コンプレキシティ $I\left(x_u^t\right) \equiv -\sum_{i=u}^{t} \log p\left(x_t \mid x_{t-k}^{t-1}, \theta\left(x_u^{t-1}\right)\right)$
    - θは与えられたx(時刻uから$t-1$まで)のもとでの最尤推定量
  - 変化点$t=a$のあと、毎時刻$t=b$で $\frac{1}{t_b - t_a + 1}\left(I\left(x_{t_a}^{t_b}\right) - \min_{i:t_a<i<t_b}\left(I\left(x_{t_a}^{i}\right) + I\left(x_{i+1}^{t_b}\right)\right)\right)$
    を計算し、閾値以上となれば$t=i$を変化点として検出
  - 変化点が検出されたら、iをaとして繰り返す
    - $t=i$で区間を分割し、異なるθによるpを用いた方が確率的コンプレキシティが低下するとき、表現力が高いモデルだということ

5. Simulation – データ作成

- x_t：2次のARモデルに従って生成

$$x_t = a_1 x_{t-1} + a_2 x_{t-2} + \varepsilon_t$$

$$a_1 = 0.6,\ a_2 = -0.5,\ \varepsilon \sim N\left(0, \sigma^2\right)$$

- y_t：3次のARモデルに従って生成
- *Score*(*x*)：対数損失で計算
- *Score*(*y*)：二次損失で計算
- 移動平均の幅 $T=5$
- 割引率 $r=0.02$
- 変化点は $t=1000x$(x=1,2,…,9)に設ける
  →3種類のパターンで試行

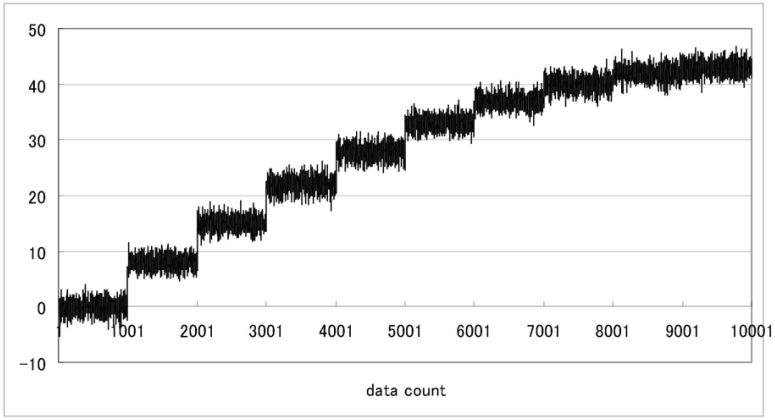
5. Simulation: Jumping mean with constant variance

- t=1000x(x=1,2,…9)の9回、平均をΔx=10−x変化(加算)
  - Δxをchange sizeと呼ぶ
- σ^2=1(一定)
- x回目の変化前後でのKL情報量は

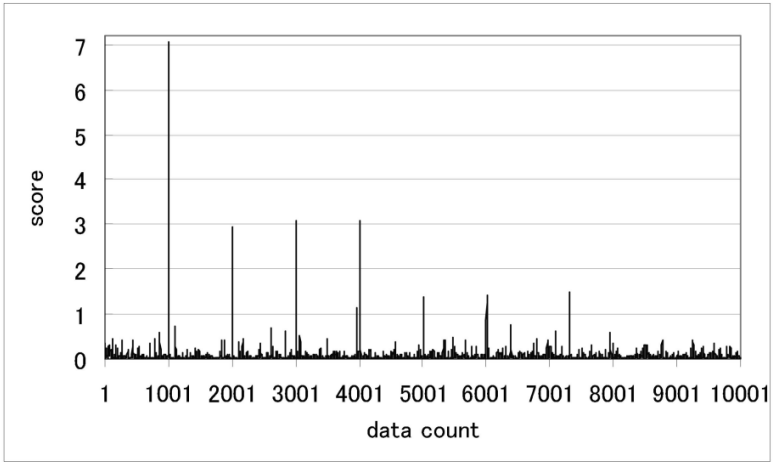
$$K(x)=\frac{\Delta(x)^2\left(1-\sum_{i=1}^{2}a_i\right)^2}{2\sigma^2}=0.405\Delta(x)^2\approx 0.4(10-x)^2$$

- t=6000まで検出できている
  このとき$K(x)$≒6.4

元データ



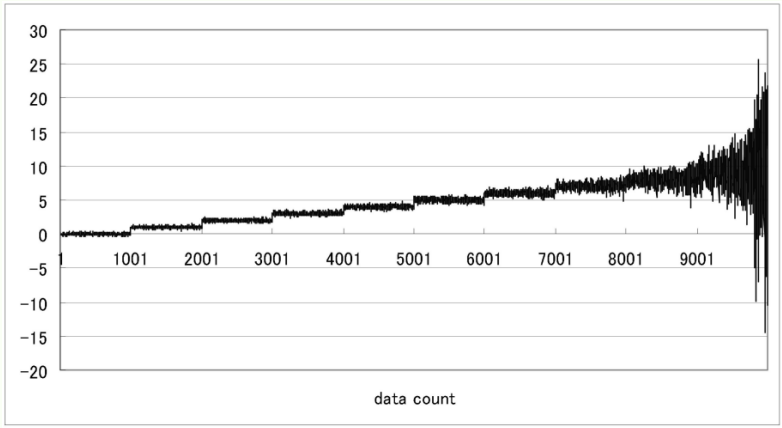
Score



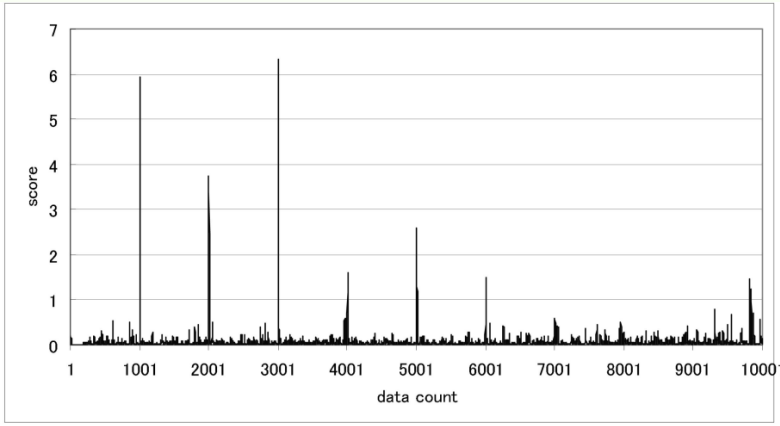
5. Simulation: Jumping mean with varying variance

- t=1000x(x=1,2,…9)の9回、平均をΔx=1変化(加算)
- $\sigma^2 = \dfrac{0.1}{\left(0.01+\left(10000-t\right)/10000\right)}$
  - 時間経過とともに0.1→10に増加
- x回目の変化前後でのKL情報量は $K(x) \approx 0.4(10-x)^2$
- t=6000まで検出できている
  このとき$K(x)$≒6.4

元データ



Score

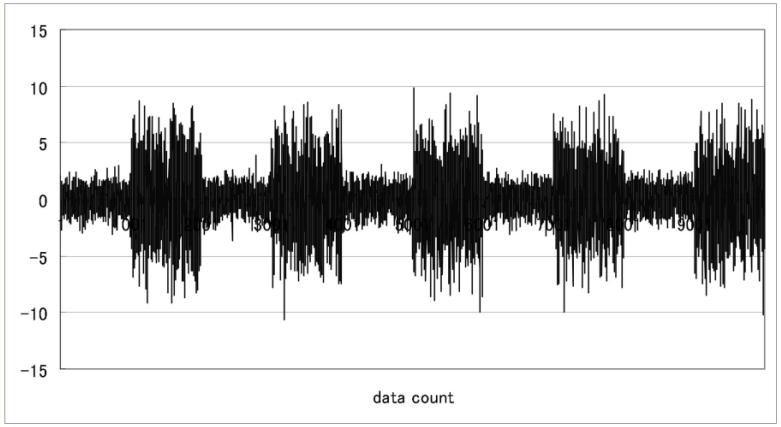


5. Simulation: Jumping variance with constant mean

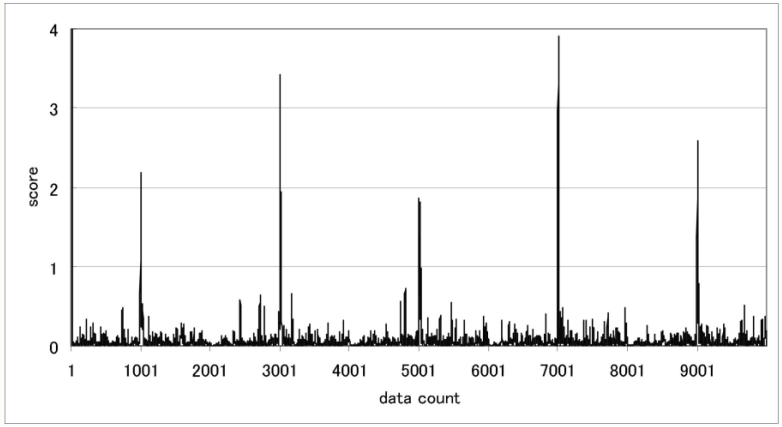
- 平均は0で一定
- t=1000x(x=1,2,…9)の9回、σ^2を1.0と9.0とで行き来させる

- 1.0→9.0のときのKL情報量≒2.90
- 9.0→1.0のときのKL情報量≒0.65
- 分散が増大するときは変化点が検出できているが、減少するときは検出できていない

元データ



Score



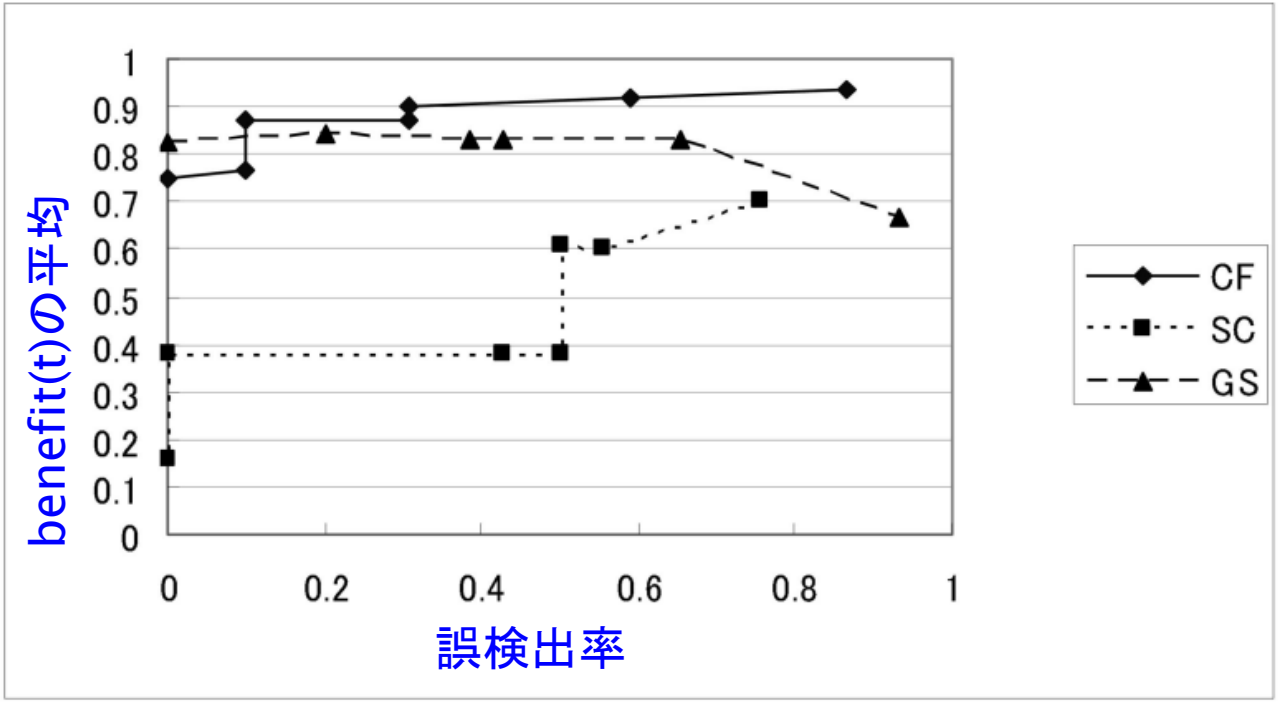
5. Simulation: Quantitative Evaluation

- Activity Monitoring[Fawcett and Provost, 1999]に基づき、検出力を検出速度と誤検出率の観点から評価

- Scoreに対して設ける閾値によって検出力が変動する
  →Effective AlarmとFalse Alarm Rateとを軸としたROC曲線のようなことを考える
  - Effective Alarm：変化点から20タイムステップ以内に検出
    t^*を真の変化時刻、tを検出時刻としたとき、$benefit(t) \equiv 1 - (t - t^*)/20$
    - Alarm Policy：
      最新のAlarmから20タイムステップ以内のAlarmは無視
    - SCやGSの場合はt^*より前の時点で検出することがあるので、
      $benefit(t) \equiv 1 - |t - t^*|/10$ とする
  - False Alarm Rate：False Alarm/全Alarm

5. Simulation: Quantitative Evaluation

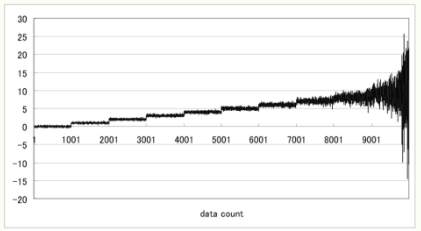
- Jumping mean with constant variance
  - データセットは前出同様、
    ただしchange sizeは5で固定している

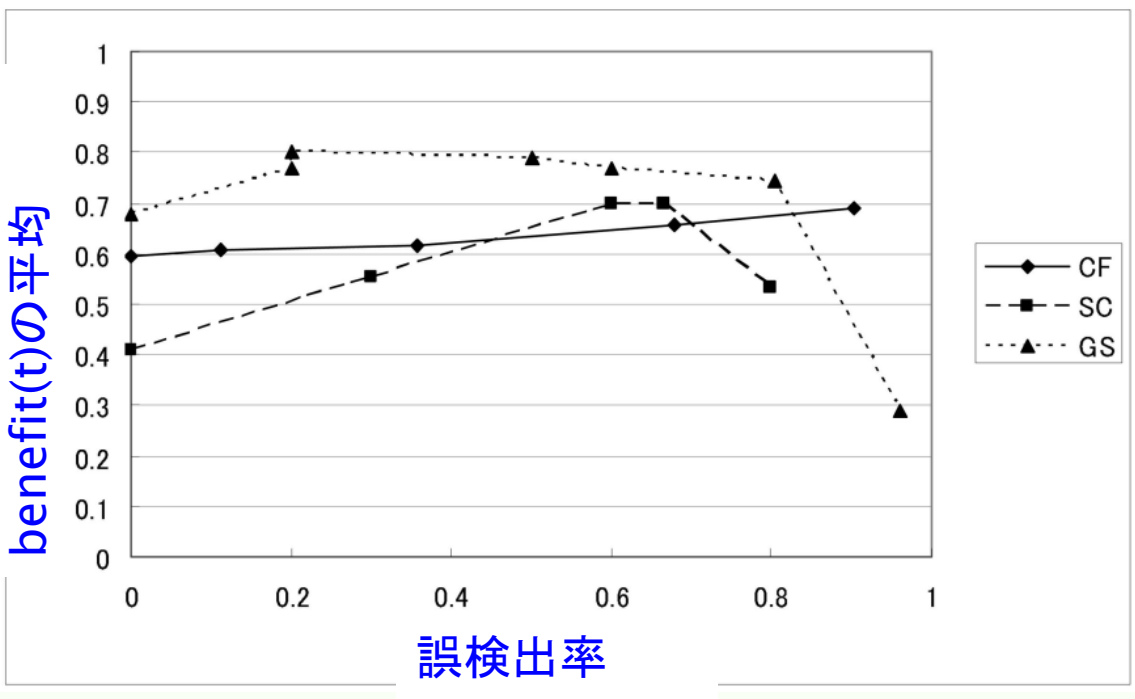


★CFの性能が高い

5. Simulation: Quantitative Evaluation

- Jumping mean with varying variance
  - データセットは前出同様



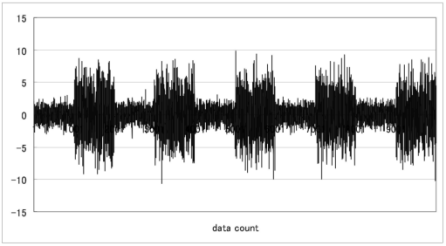


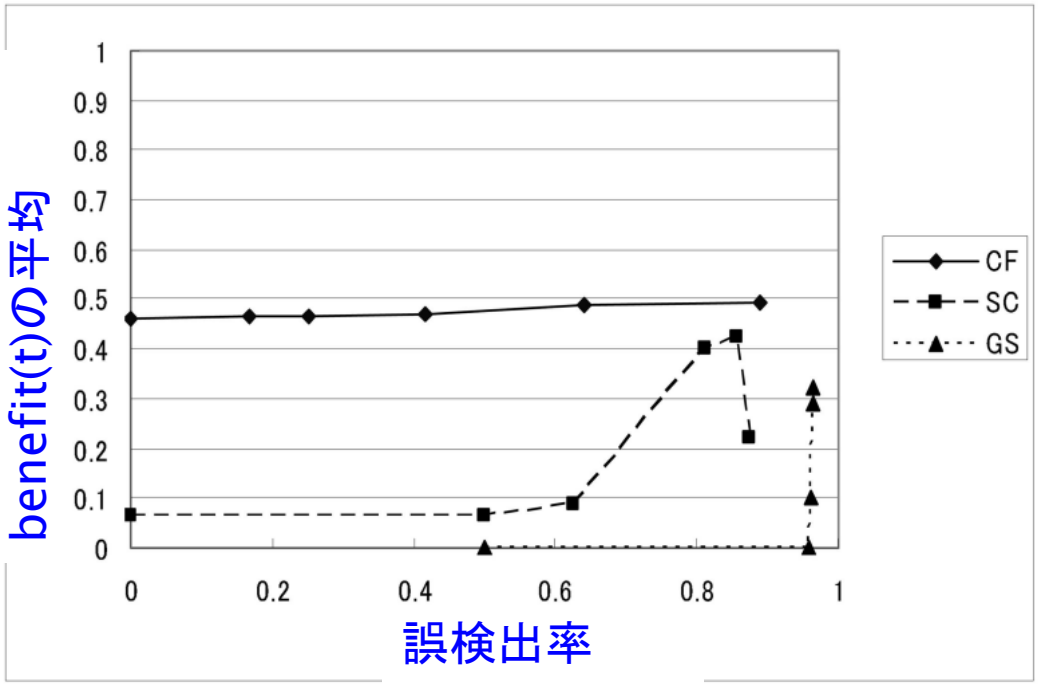
★CFよりもGSの性能が高い

- ただし、分散が徐々に縮小していくパターンを試すと、CFのみが変化点を検出できた(と書いてあるがグラフは無い)

5. Simulation: Quantitative Evaluation

- Jumping variance with constant mean
  - データセットは前出同様





★CFの性能が高い

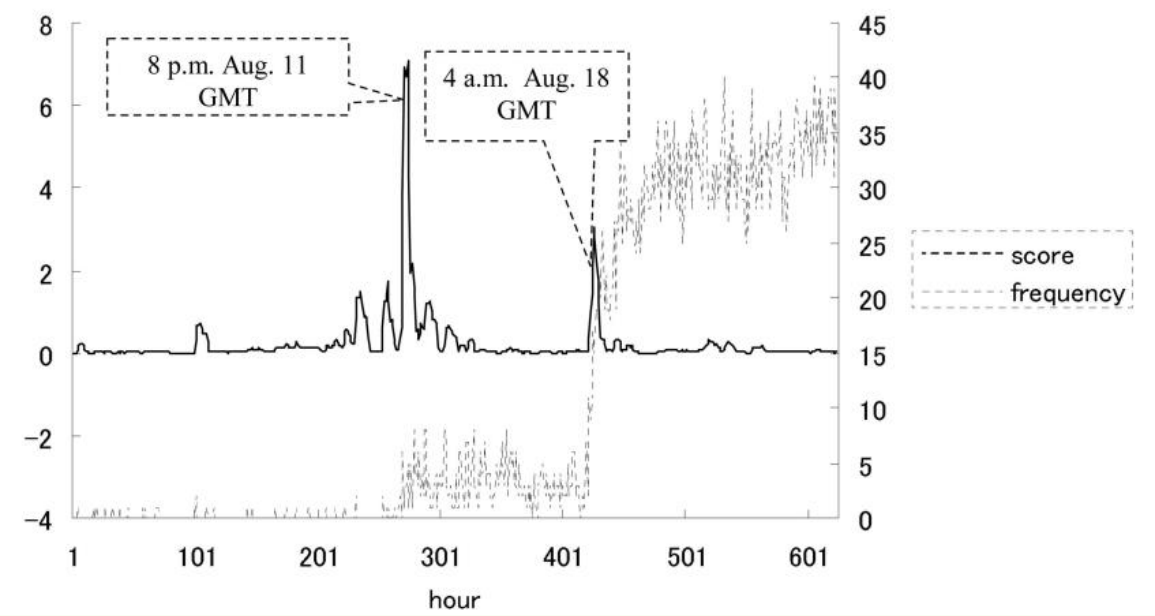
- ただし、分散が小さくなる(KL情報量が小さい)変化点を検出できていないのでbenefitが低い
- そもそもKL情報量に依存する手法であることの課題

5. Simulation

- 考察
  - 定性的には、CFはSCやGSよりも変化点検出に遅れの出にくい方法といえる。
  - SCやGSで、False Alarm Rateをあげても Benefitが低下することがある。これは、Alarmの発生が真の変化点よりも早すぎることが多くなるため。
  - CFはSCやGSに匹敵する検出力を持っている。特に3番目のデータセットと、2番目のデータセットの分散を小さくしていくパターンでは、相対的に高性能といえる。
  - 計算のオーダーは、SCやGSが$O(n^2)$であるのに対し、CFは$O(n)$であるため、リアルタイム分析にはCFが適している。

6.実データへの適用

- DoS攻撃の実例に適用
  - パラメータ設定はSimulationと同様、Scoringは対数損失



- CFにより、変化点を正しく検出できた。
  - 8月11日のAlarmは、現実のトレンドマイクロの発表より1時間44分早く、8/18のAlarmも約1日の遅れに留まっている
  - SCやGSでも変化点は検出できるが、CFより検出時刻が遅い

7.まとめ

- 成果
  - 2段階の学習プロセスからなるChangeFinderの開発
  - ChangeFinderへのARモデルとSDARモデルの適用
  - 外れ値検出と変化点検出の統合的な枠組みの構築
  - いくつかのデータによる挙動の検証

- 課題
  - *T*の最適化、データセットからの*T*自体の学習
  - 分散が減少する変化点の検出法
  - ARIMAモデルや状態空間モデルへの拡張
  - 隠れマルコフモデル等との統合