A cat-and-mouse-game on a dataset

Security meets Machine Learning #1: https://connpass.com/event/62844/

Kenji Aiko

Outline

0. Introduction

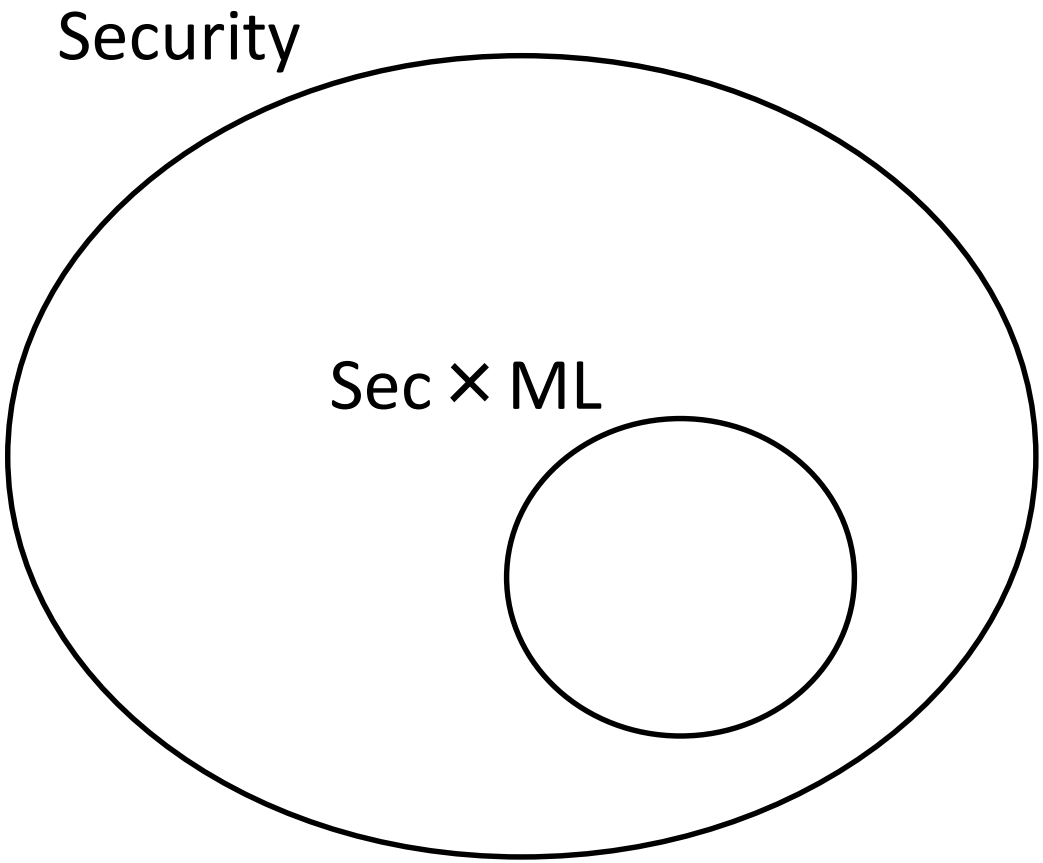
1. Dataset of Security

2. Time series

3. Automation

0. Introduction

対策 可能/不可能 問題



対策可能問題

Ex. Overflow, XSS, SQLi, etc…

```
strcpy(dst, src);
```

```
strncpy(dst, src, sizeof(dst)-1);
```

```
PrintHTML(post.data.x)
```

```
PrintHTML(htmlspecialchars(post.data.x))
```

適切な修正をすることで、100%対策できる問題
（開発者/設計者のミスによって発生）

対策可能問題

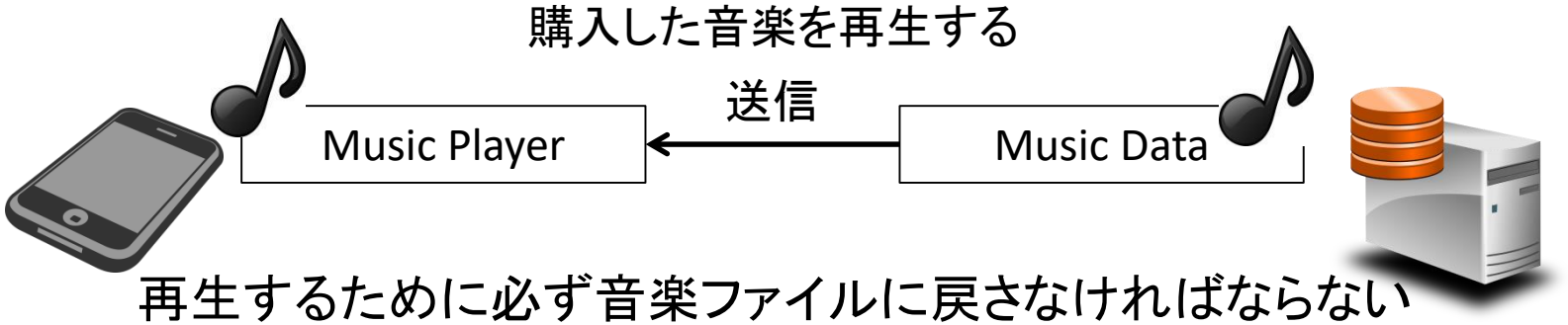
開発者（人間）はミスをする、という前提で作られる
セキュリティ機能

- StackGuard（カナリア）
- Address Space Layout Randomization（ASLR）
- Data Execution Prevention（DEP）
- yarai ZDP, Caller Checks by EMET
- Same-Origin Policy（SOP）
- X-(Content-Type/Frame/Download)-Options
- X-XSS-Protection

脆弱性があったとしても、その影響範囲を制限し、
被害を低減させる

対策不可能問題

Ex. Reversing, Crypto, etc…



再生するために必ず音楽ファイルに戻さなければならない
よって**「音楽ファイル」**の違法コピーを防げない

様々なコンピュータを経由するネットワーク通信は
“未来永劫100%”の安全性は保障できない（MITM, Cryptoanalysis, etc）

完璧な対策が存在しない問題

対策不可能問題

耐タンパー性

非正規な手段による機密データの読み書きを防ぐ機能、
解析の困難さを”耐タンパー性”と呼ぶ

Ex.

- 実行時に復号されるプログラム（or データ）
- 空気に触れると内容が消えるメモリチップ
- プローブを取り付けると動作しなくなる回路
- 解読に実現困難な計算量が必要な暗号通信

完全な対策が実現できないことを前提とした上で
解析の困難さを示す

対策不可能問題

Chain of Trust（信頼の連鎖）

非改ざんの証明を継承させて安全性を担保する仕組み
（ex. TPM: Trusted Platform Module）

One-way function（一方向性関数）

計算が容易であり、しかし逆関数の計算が困難な関数
（ex. 素因数分解 / 離散対数問題）

対策不可能問題に対する解のひとつ
（限りなく100%に近づける）

※）ちなみに一方向性関数の存在は証明されていない。また、もし存在することが証明できれば、P≠NPであることが知られている。

対策困難問題

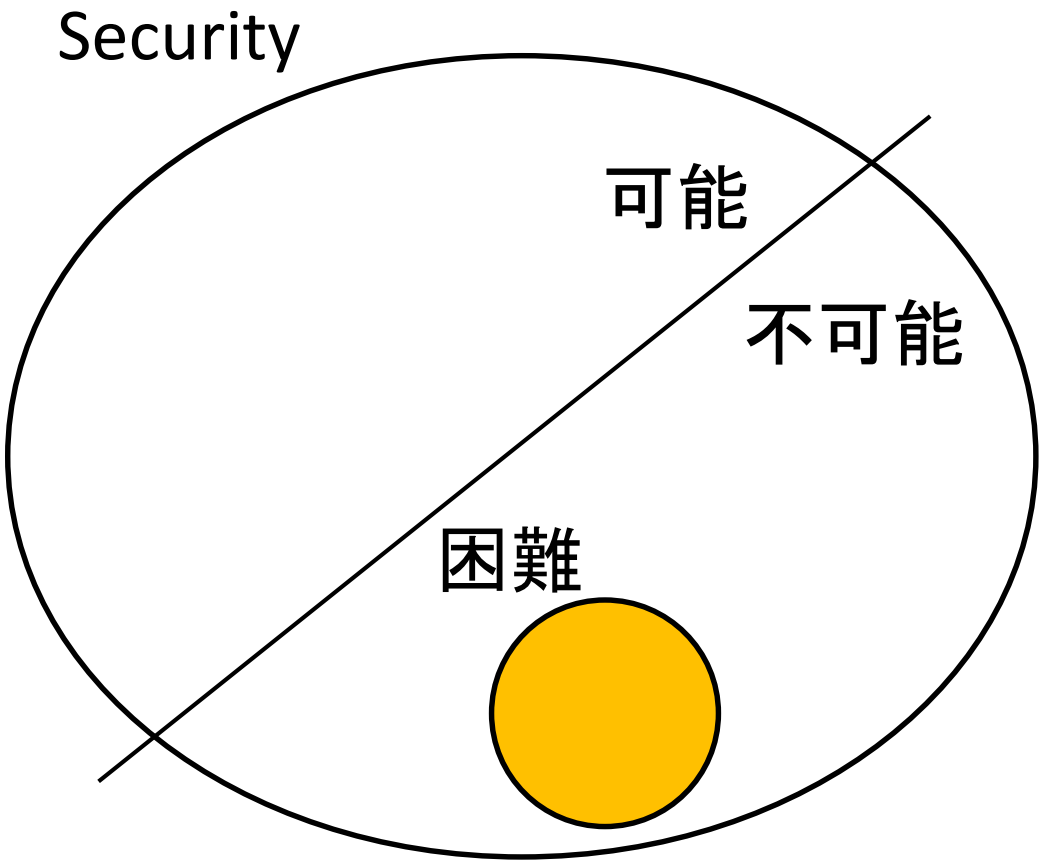
人間（or 強いA.I.）の介入がなければ
対策が続けられない（機能が劣化する）問題

- アンチウイルス
- スパム（mail/message）フィルタ
- NW攻撃検知（IPS/IDS, etc）
- オンラインゲームチート（and BOT）検知
- システム異常検知（Server etc）

完璧な対策がないという点では対策不可能問題

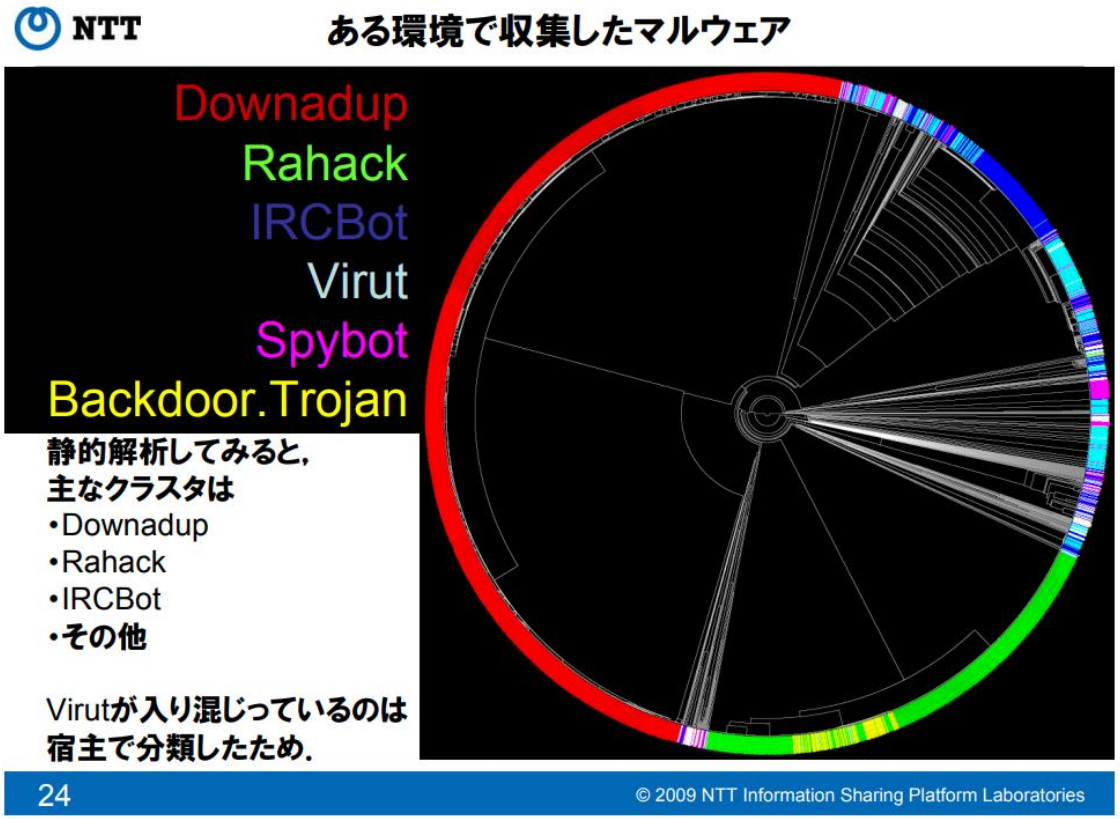
<u>強いA.I.ができれば解決する</u>

今回はこの部分の話



1. Dataset of Security

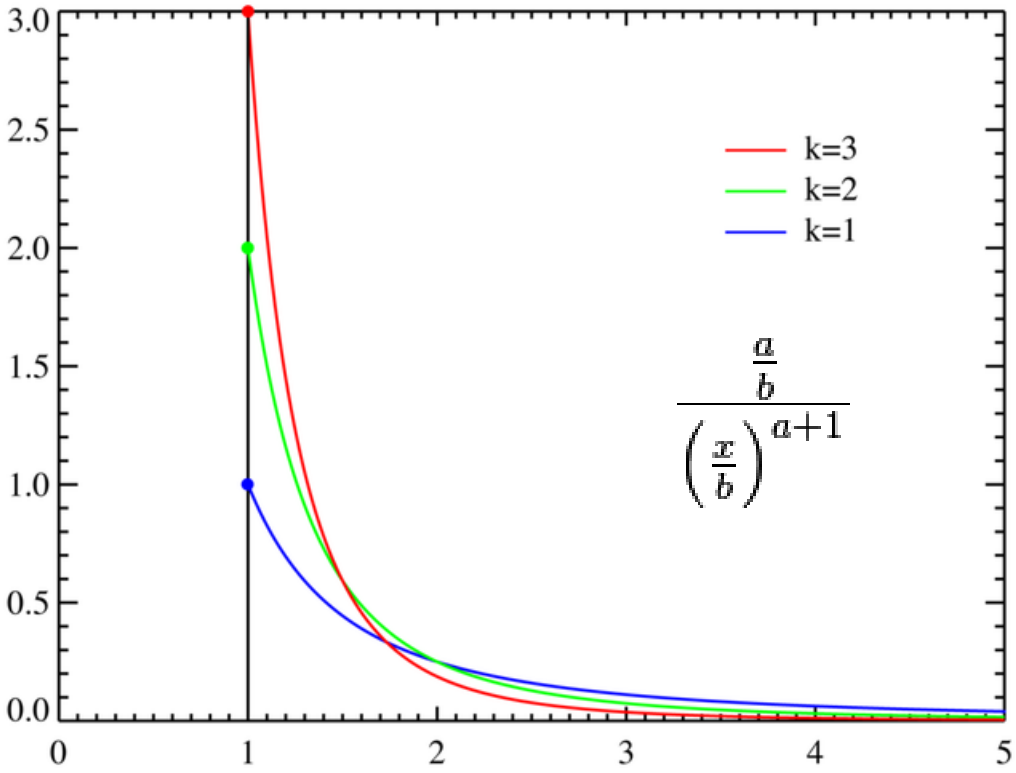
極端な分布と時間軸



機械語命令列の類似性に基づく自動マルウェア分類システム
http://www.iwsec.org/mws/2009/presentation/A8-4.pdf

極端な分布と時間軸

イタリアの経済学者ヴィルフレド・パレート (Vilfredo Pareto) が所得の分布をモデリングする分布として提唱した確率分布

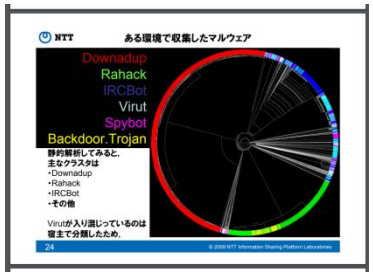


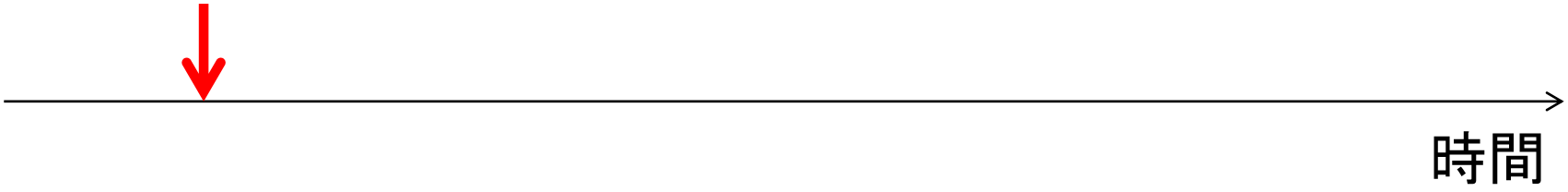
http://ja.wikipedia.org/wiki/%E3%83%91%E3%83%AC%E3%83%BC%E3%83%88%E5%88%86%E5%B8%83#mediaviewer/File:Pareto_distributionPDF.png

極端な分布と時間軸

セキュリティに関連するデータセットは
時間を固定した場合、一般にパレート分布に近似する

この分布が時間軸で変化していく





*n*年前の *Dataset* と、現在の *Dataset* は？

30年前の猫と、現在の猫に
違いはない

30年前のMalwareと、現在のMalwareは
まったく違う

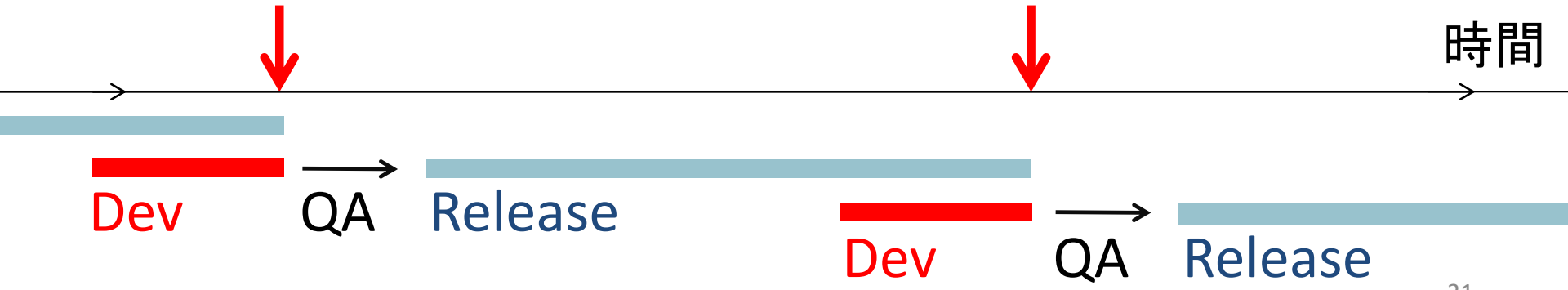
セキュリティのデータ分析とは
“変化するDatasetを分析/検知する方法”
を考えること

2. Time series

時系列変化

時系列が変化していくことに関する検知率の
”保証”を考える必要がある

- 未来の検知率を示したい（予測したい）
- アルゴリズムの有用性を定量化（評価）したい
- アップデートのタイミングを知りたい
- 学習に必要な期間を知りたい



検知率低下の予測

時系列データに基づくマルウェア検知アルゴリズムの評価
https://ipsj.ixsq.nii.ac.jp/ej/

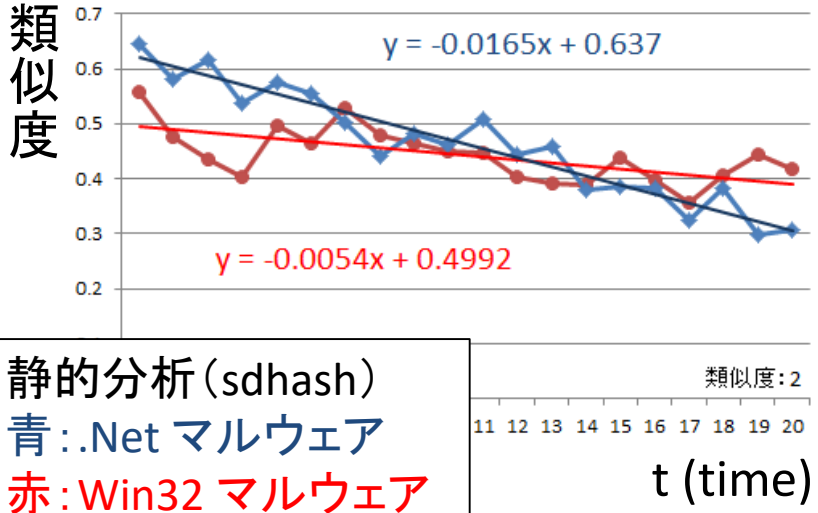
$L_{tn} = L_{1n}\ L_{2n}$ ..... L_{90n}

時間

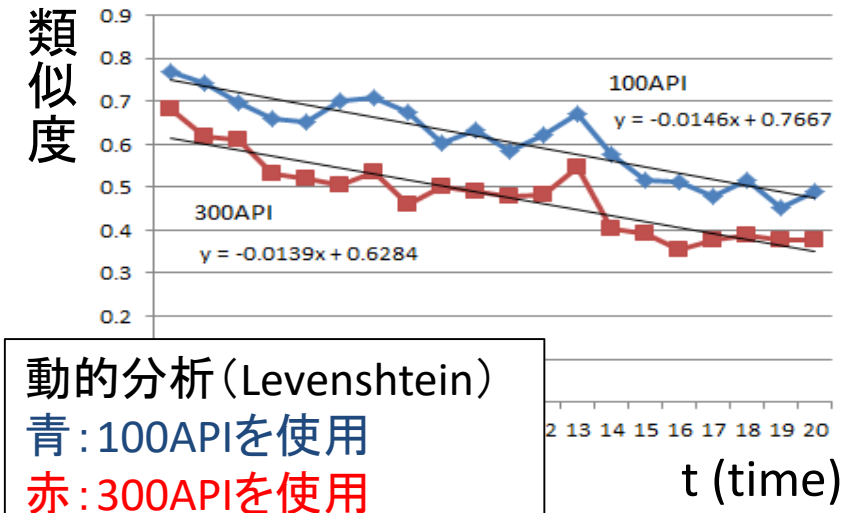
$L_{1n} \longleftrightarrow L_{tn}$
類似度を算出

→

検知率の低下を予測
アップデートのタイミングを決める



静的分析（sdhash）
青：.Net マルウェア
赤：Win32 マルウェア



動的分析（Levenshtein）
青：100APIを使用
赤：300APIを使用

必要なデータセットの算出

時系列データに基づくマルウェア検知アルゴリズムの評価
https://ipsj.ixsq.nii.ac.jp/ej/

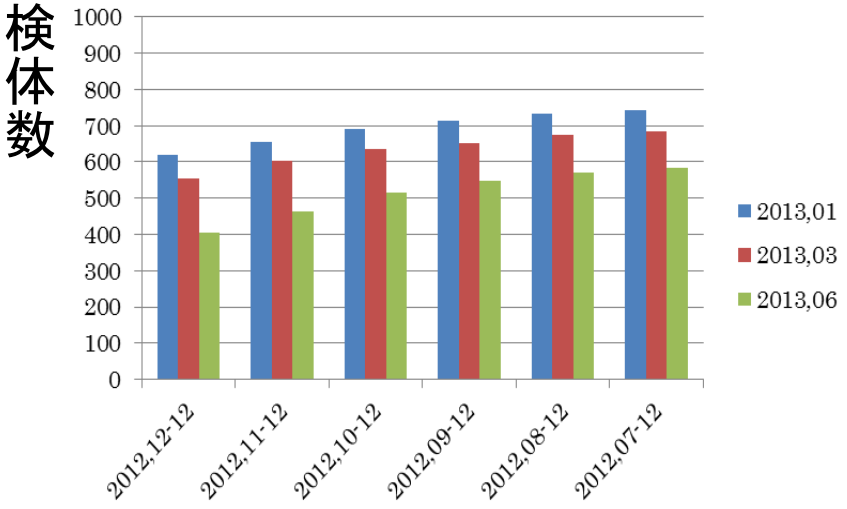
$L_{tn} = L_{1n}\ L_{2n}$ ..... L_{90n}

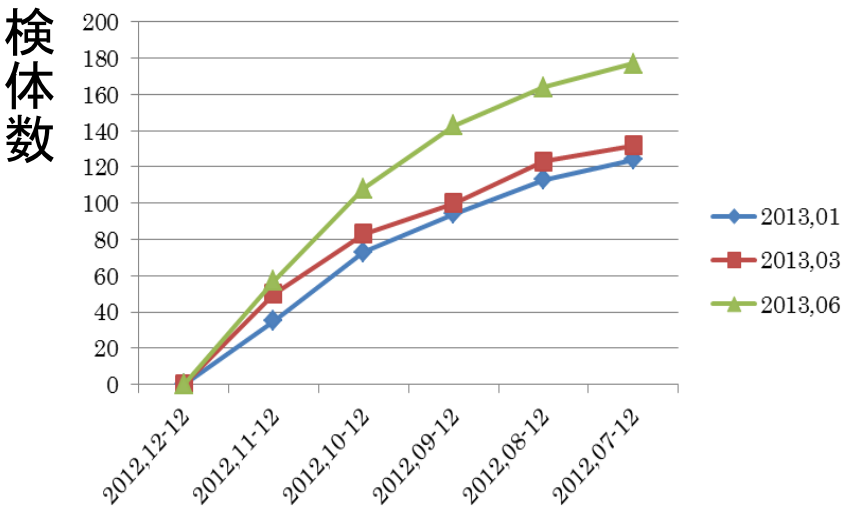
時間

$L_{xn} \longleftrightarrow \sum_{i=1}^{x-1} L_{in}$

類似度を算出

一定の検知率を担保するために
必要な学習期間を決定

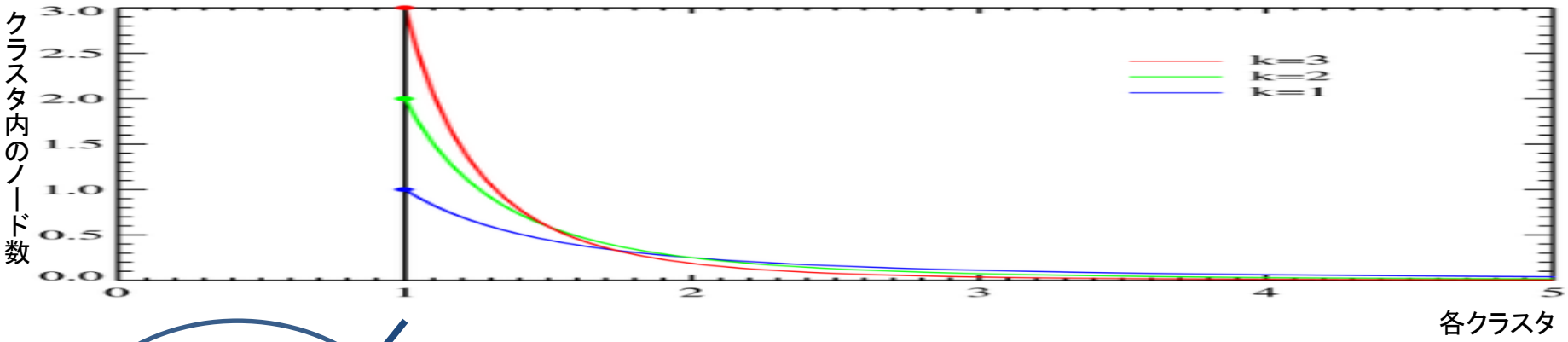


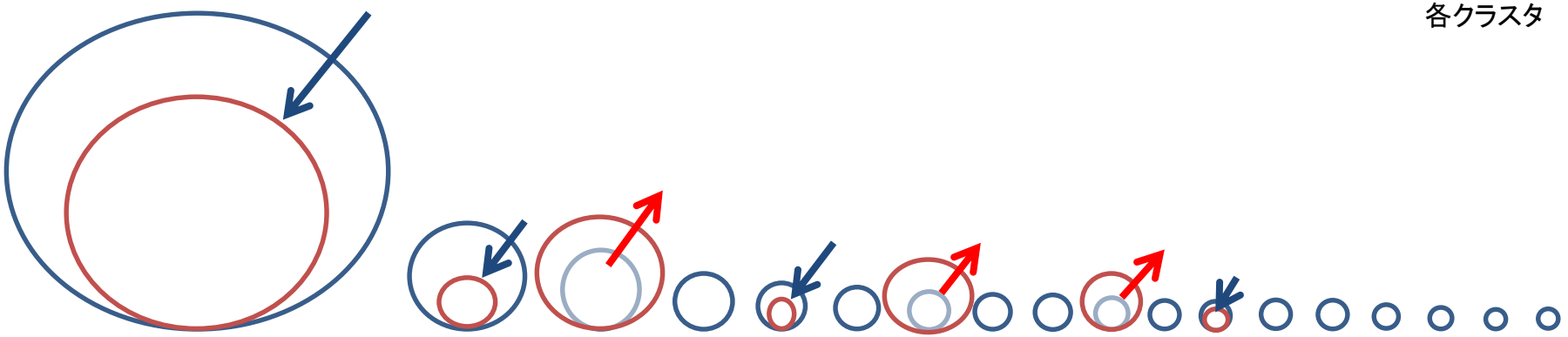


変化してきたDatasetを分析することで
これからの変化を予測できないか？

未来の予測

時間経過にともなって変化する各クラスタを
事前予測できないか？

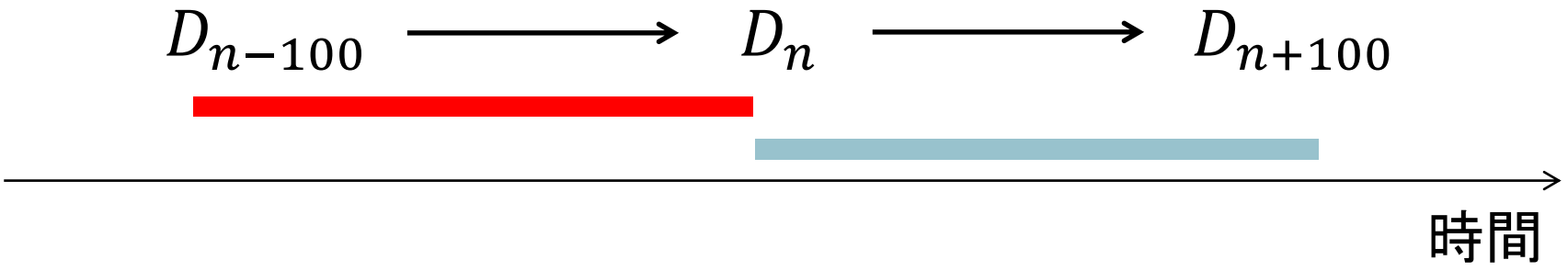




http://ja.wikipedia.org/wiki/%E3%83%91%E3%83%AC%E3%83%BC%E3%83%88%E5%88%86%E5%B8%83#mediaviewer/File:Pareto_distributionPDF.png

未来の予測

過去の変化から未来の変化を予測できないか？



極めて困難…

（失敗の許されない株価予測をしている気分）

これまでの流れ

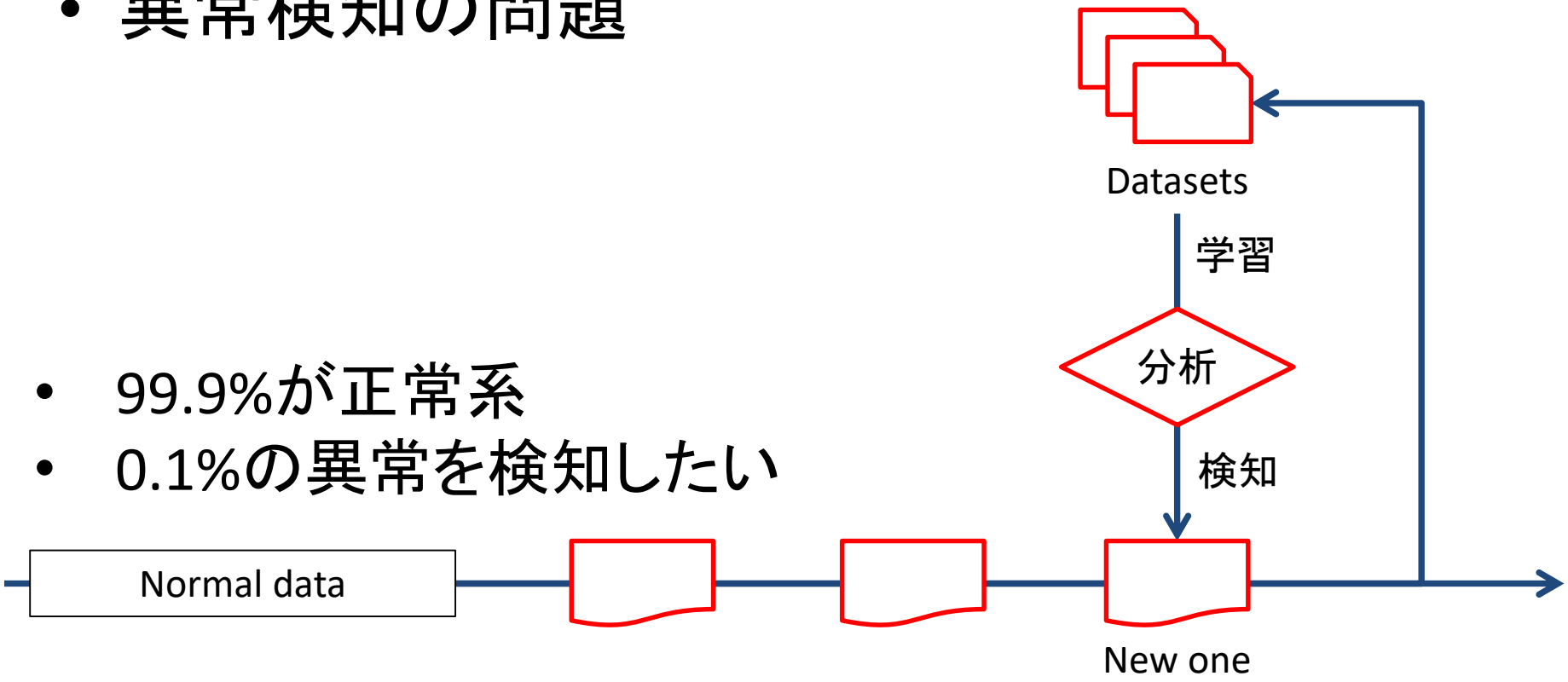
1. Datasetは極端な分布になっている
2. また時間によってそのDatasetは変化する
3. 時系列の変化を分析することは有用
4. しかし事前予測は困難

変化を”自動で”追尾するアルゴリズムで対応
人の手を借りずに、半永久的に対応が可能

3. Automation

Pattern 1

- 異常検知の問題



- 99.9%が正常系
- 0.1%の異常を検知したい

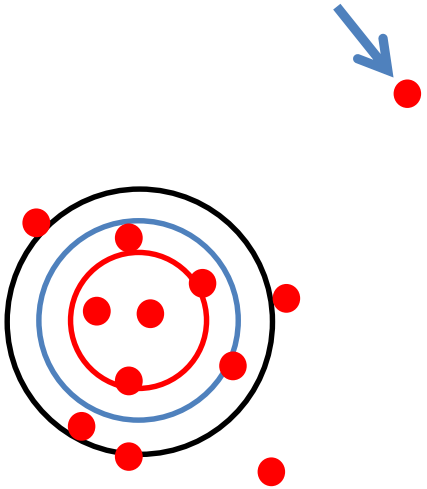
ほぼ正常のDatasetから稀に発生する異常を検知したい

Pattern 1

- 異常検知の問題
  - 正常系に共通した特徴がある（分散が小さい）
  - 誤検知を許容できる

ex:

- CPU、メモリ使用率の変化
- サーバ負荷、アクセス数の変化

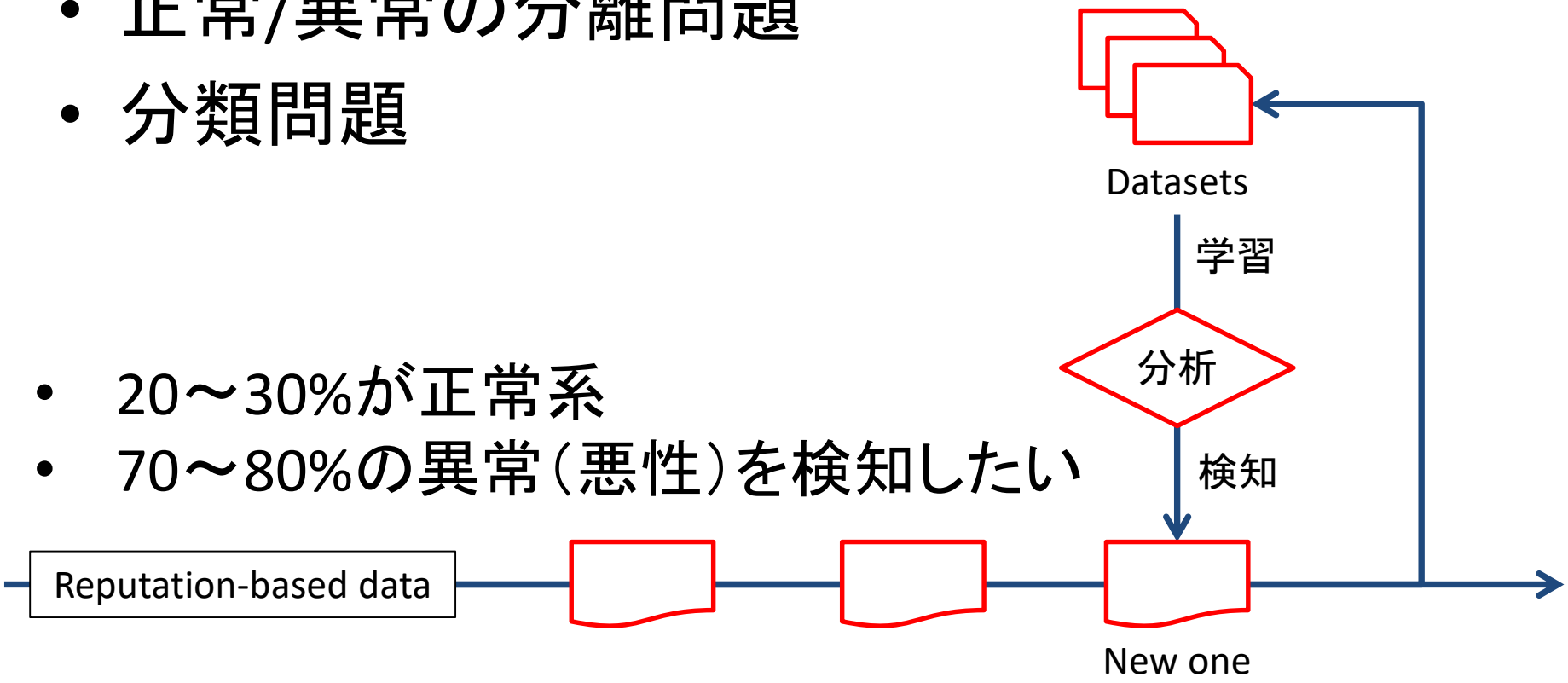


“何か問題が発生した場合にAlertを投げる”
といった使い方が一般的

Elastic Search + Kibana
https://www.elastic.co/blog/implementing-a-statistical-anomaly-detector-part-1

Pattern 2

- 正常/異常の分離問題
- 分類問題

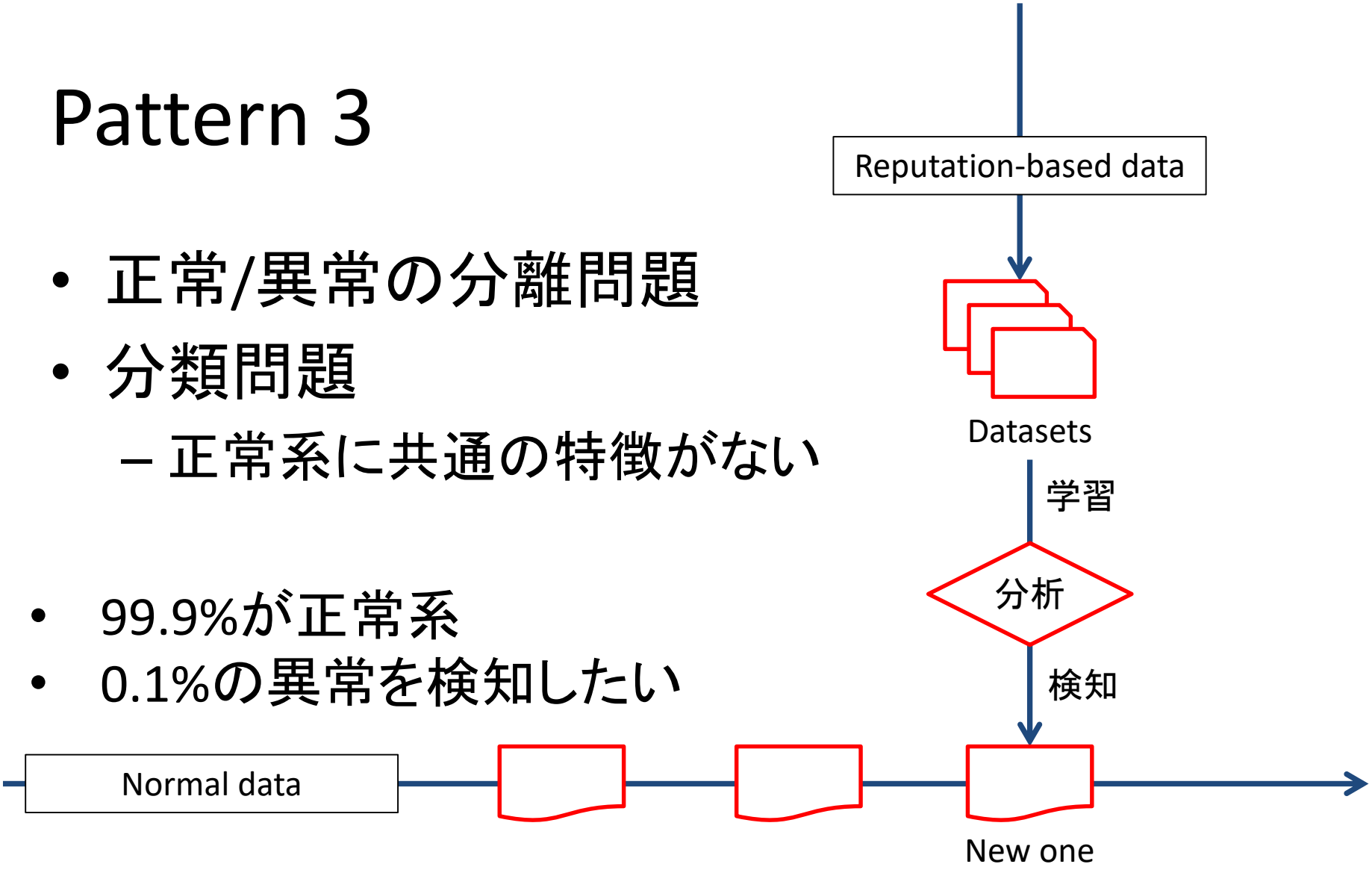


- 20～30%が正常系
- 70～80%の異常（悪性）を検知したい

ほぼ異常のDatasetから異常のDataを検知したい

Pattern 3

- 正常/異常の分離問題
- 分類問題
  - 正常系に共通の特徴がない



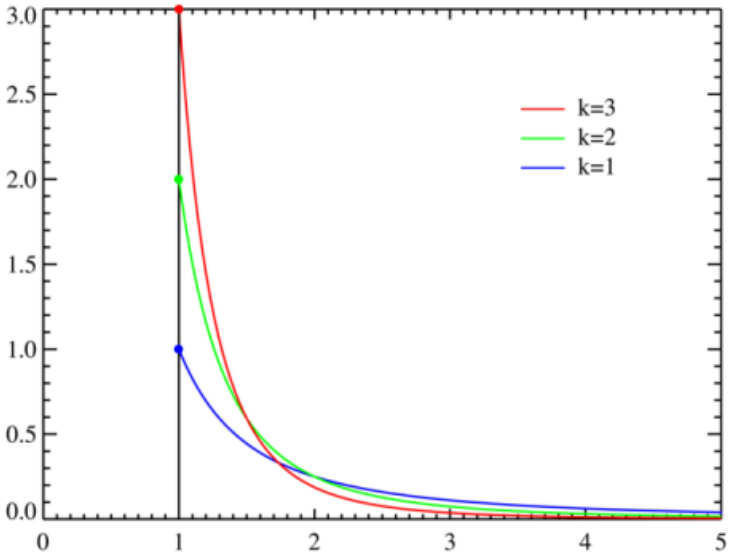
- 99.9%が正常系
- 0.1%の異常を検知したい

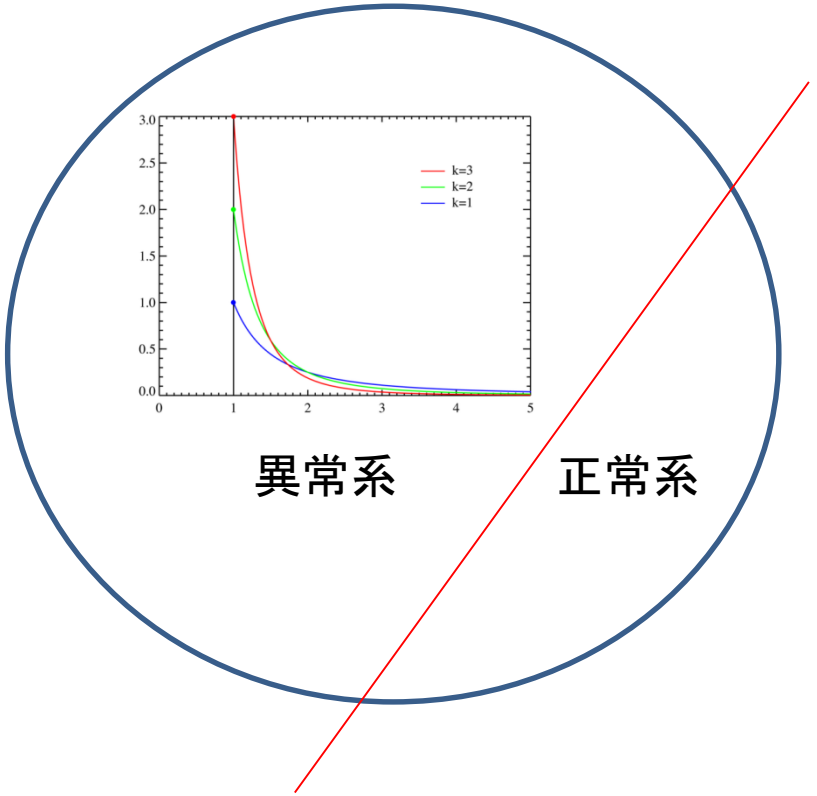
ほぼ異常のDatasetから稀に発生する異常を検知したい

正常/異常の分離

Reputation-basedのデータは
だいたい異常系:正常系が7:3の割合になる

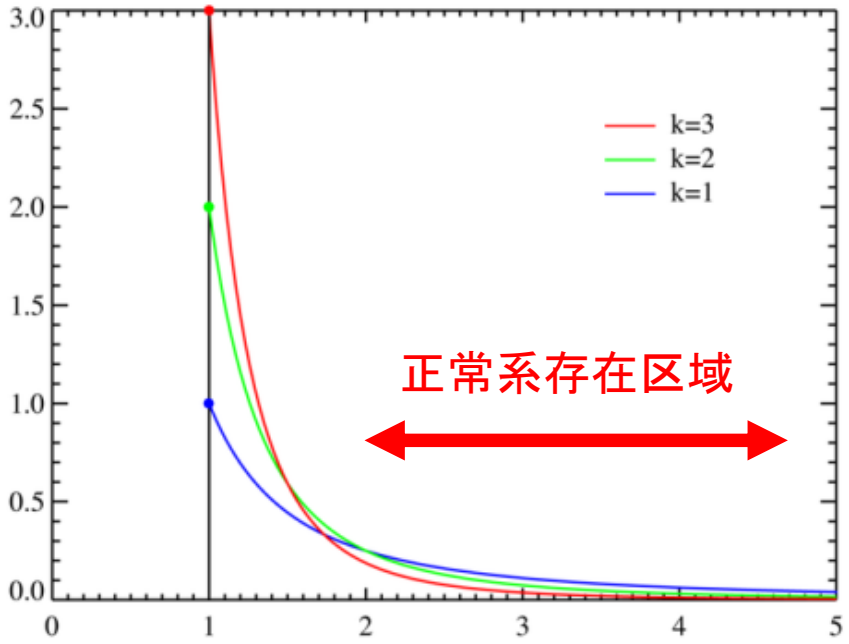
全体としての分布





正常/異常の分離

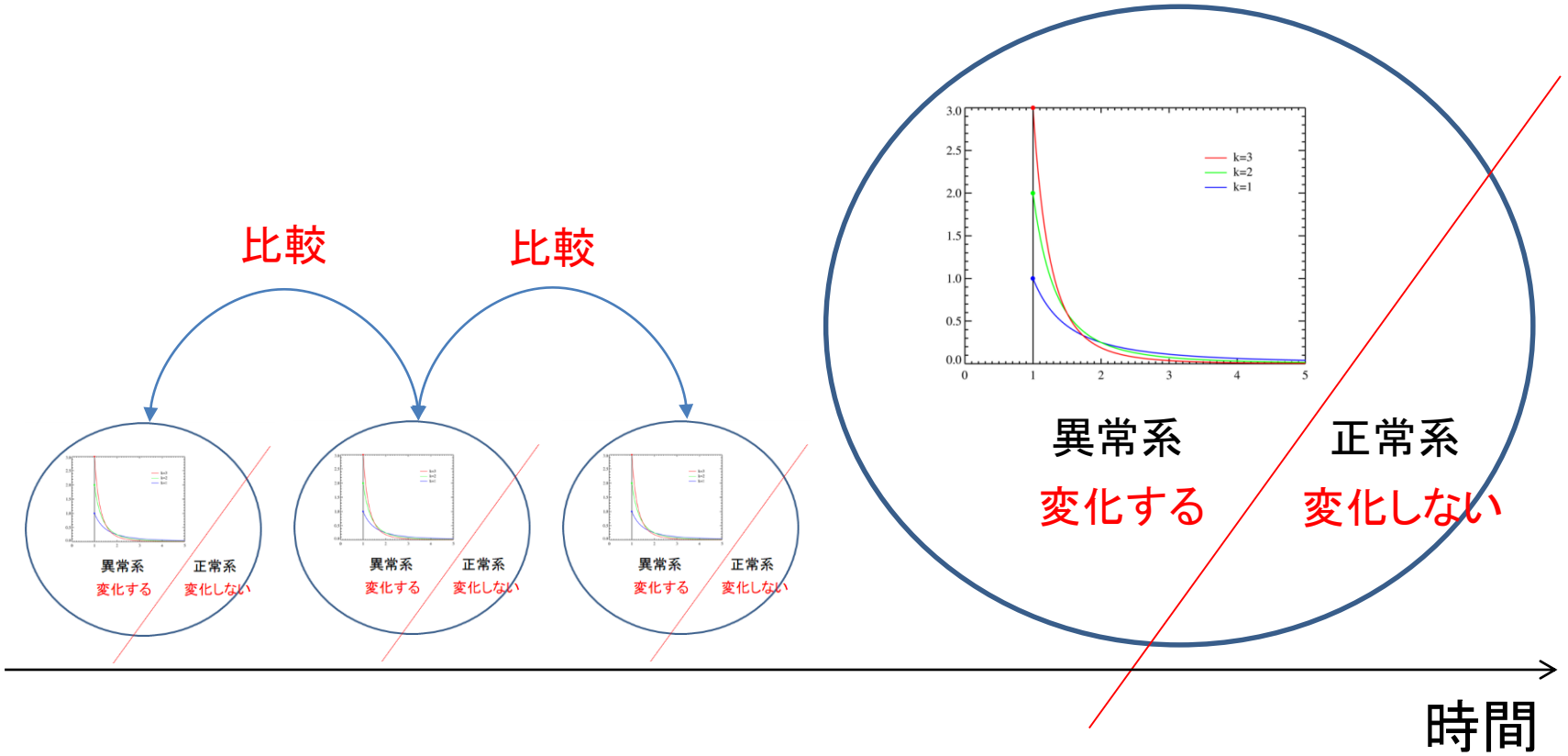
正常系は共通点がなく、
各クラスタがマイノリティのため小尾に存在



よって上位クラスタを異常系として学習すればOK

正常/異常の分離2

異常系が時系列で変化することを利用し、
変化が観測されたクラスタを異常系と判断する

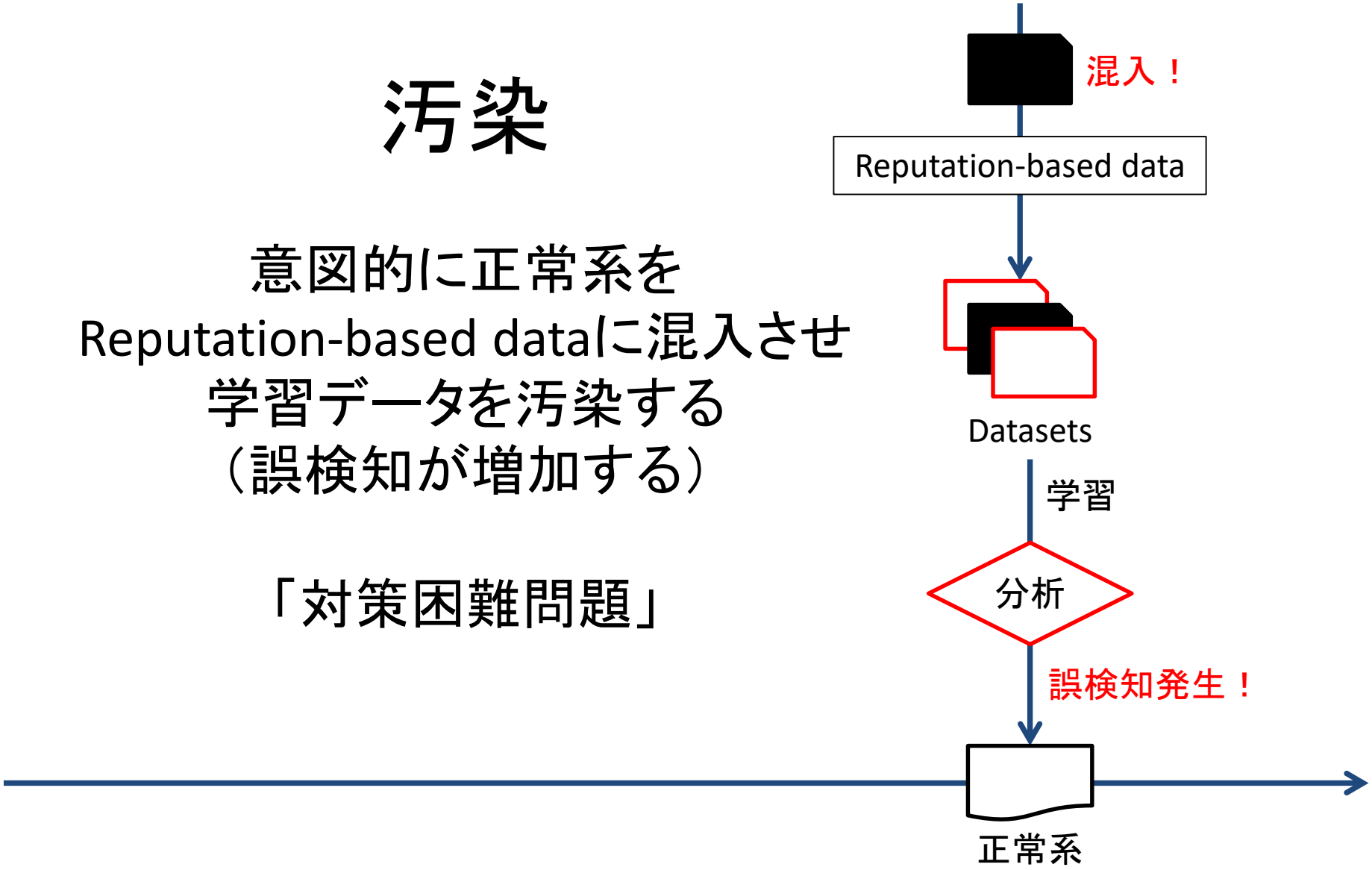


Demo
https://github.com/kenjiaiko/secml

汚染

意図的に正常系を
Reputation-based dataに混入させ
学習データを汚染する
（誤検知が増加する）

「対策困難問題」



アルゴリズムをサーバサイド（など）に置けば
BlackBoxテストと等価

まとめ

1. Datasetは極端な分布になっている
2. また時間によってそのDatasetは変化する
3. 時系列の変化を分析することは有用
4. しかし事前予測は困難
5. なら自動で学習/検知させればよい
6. いたちごっこの自動化
7. 自動化の仕組みにも脆弱性はある