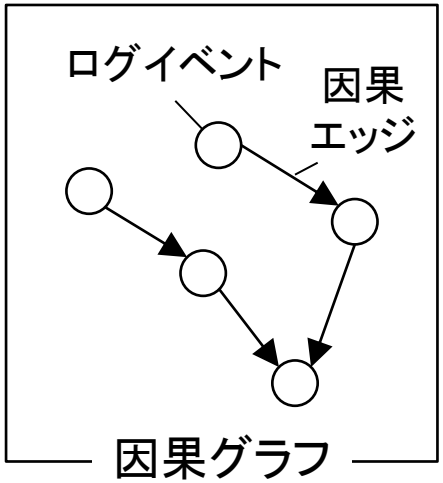
# ネットワークログの
# 対話的因果解析の検討

小林 諭[1], 石井 宏典[1], 山内 利宏[1], 明石 修[2], 福田 健介[2]
[1]岡山大学, [2]国立情報学研究所

IA研究会 2024年3月13日

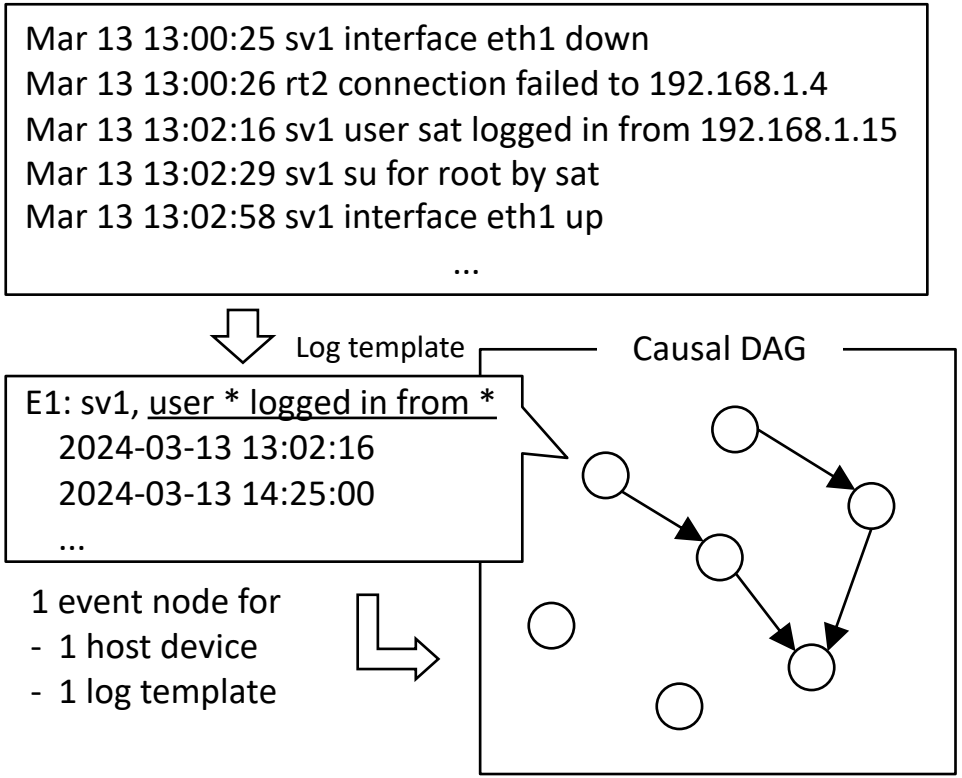
# ネットワークログの因果解析

- ネットワーク機器のシステムログ
  - 障害の予兆把握や原因究明において重要
  - 多量かつ多様に出力され、振る舞いの把握は容易でない
- ネットワークログの因果解析
  - ログイベント間の関係を因果グラフで表現
    - 短時間で直感的に把握可能
  - 擬似相関を除いて直接的な因果関係に注目可能
  - 因果関係を遡ることで原因究明に応用可能



# ログの因果解析に関する既存技術

- logdag [2]
  - ログをホスト名と出力フォーマットを用いて分類
  - 分類されたログのタイムスタンプの集合(イベント時系列)をノードとする
  - 因果探索(PCアルゴリズム)により因果グラフを推定
- システムの振る舞いを直感的に把握可能






[2] S. Kobayashi, et al. “Mining Causality of Network Events in Log Data”, IEEE Transactions on Network and Service Management, pp.53-67, vol.15, no.1, 2018

# ログ因果解析の課題とその対策

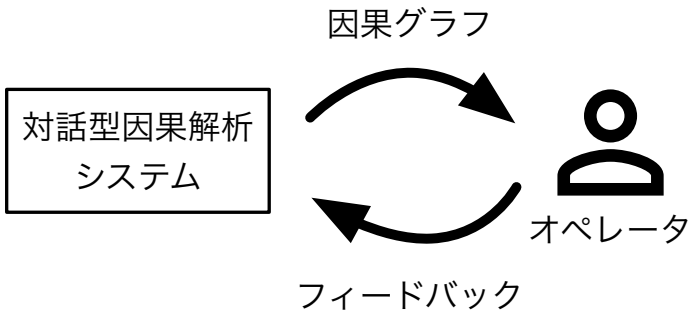
- 入力データの粒度が原因で、信頼性に難あり
  - ログ時系列は数値計測データなどより情報量が少ない
    - 数ヶ月に一度程度しか現れないログなど
  - 因果解析手法の改善のみでは対処困難

➢オペレータの知識の追加利用

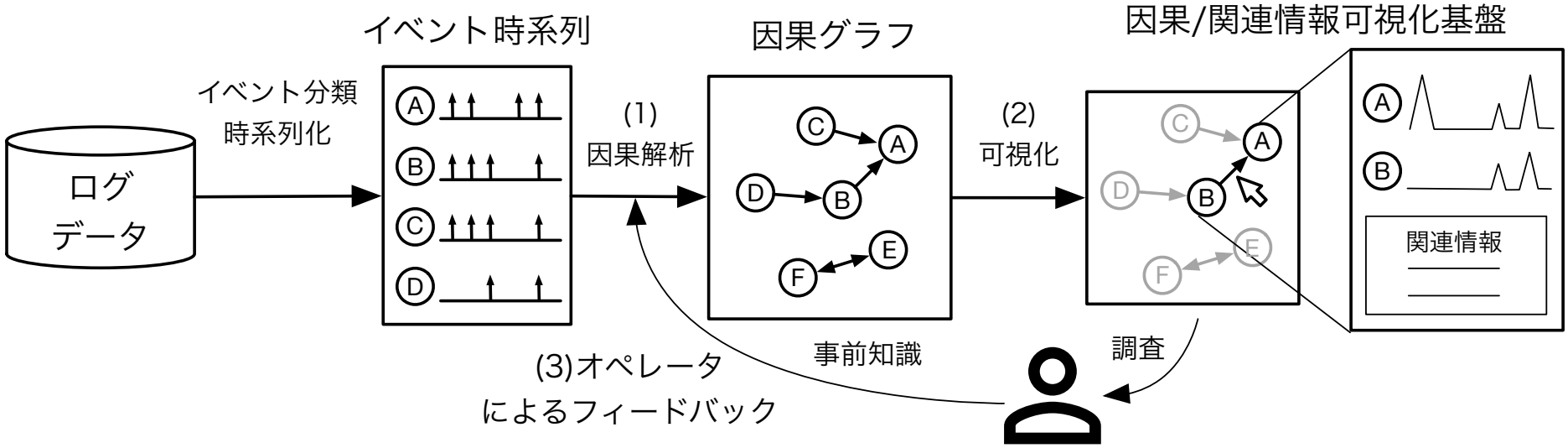
  - ログデータからは得られない情報をオペレータの知識により補完
    - システムの構成や要素技術の知識、別システムの運用経験など
  - 既存手法では、一部の知識のみヒューリスティクスとして利用

# 研究の目的

- ネットワークログの対話的因果解析
  - オペレータが因果エッジの正誤をフィードバック入力する
  - 継続的に行うことで、因果解析の事前知識を蓄積
  - 利点: スモールスタートが可能で、オペレータへの負担が小さい
- この発表の目的
  - ログの対話的因果解析の設計と要素技術の検討
  - それぞれの要素技術について課題を整理



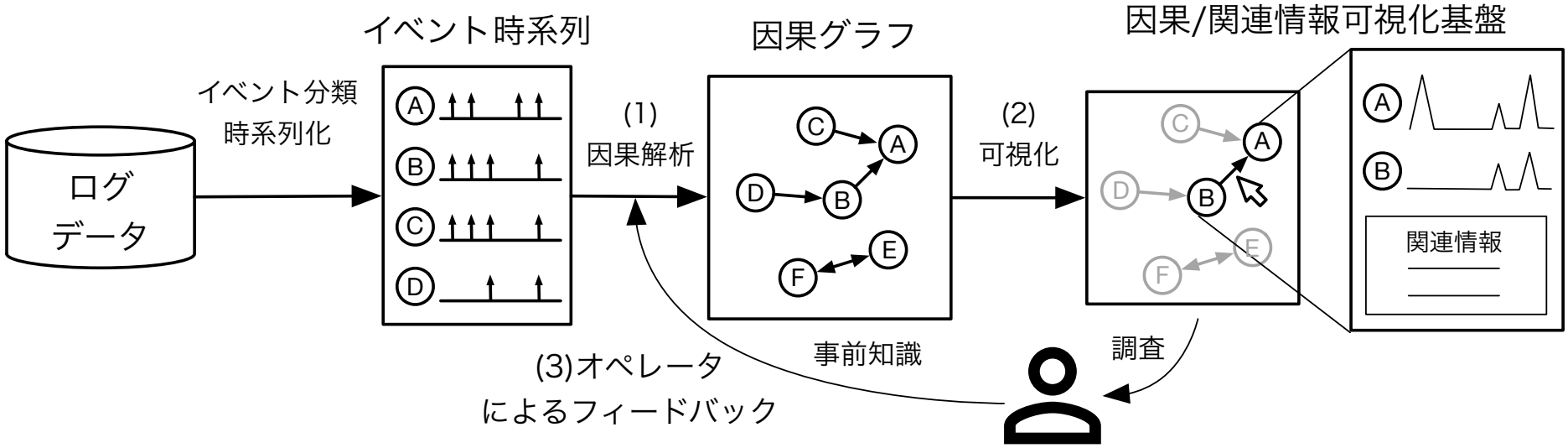
# 対話的因果解析の概要と3つの要素技術




1. 事前知識を用いたログ因果解析
2. 因果および関連する情報の可視化
3. 因果エッジに対するフィードバックの入力

3つの要素技術について
以降, 個別に検討

# 対話的因果解析の概要と3つの要素技術




1. 事前知識を用いたログ因果解析
2. 因果および関連する情報の可視化
3. 因果エッジに対するフィードバックの入力

# (1) 事前知識を用いたログ因果解析

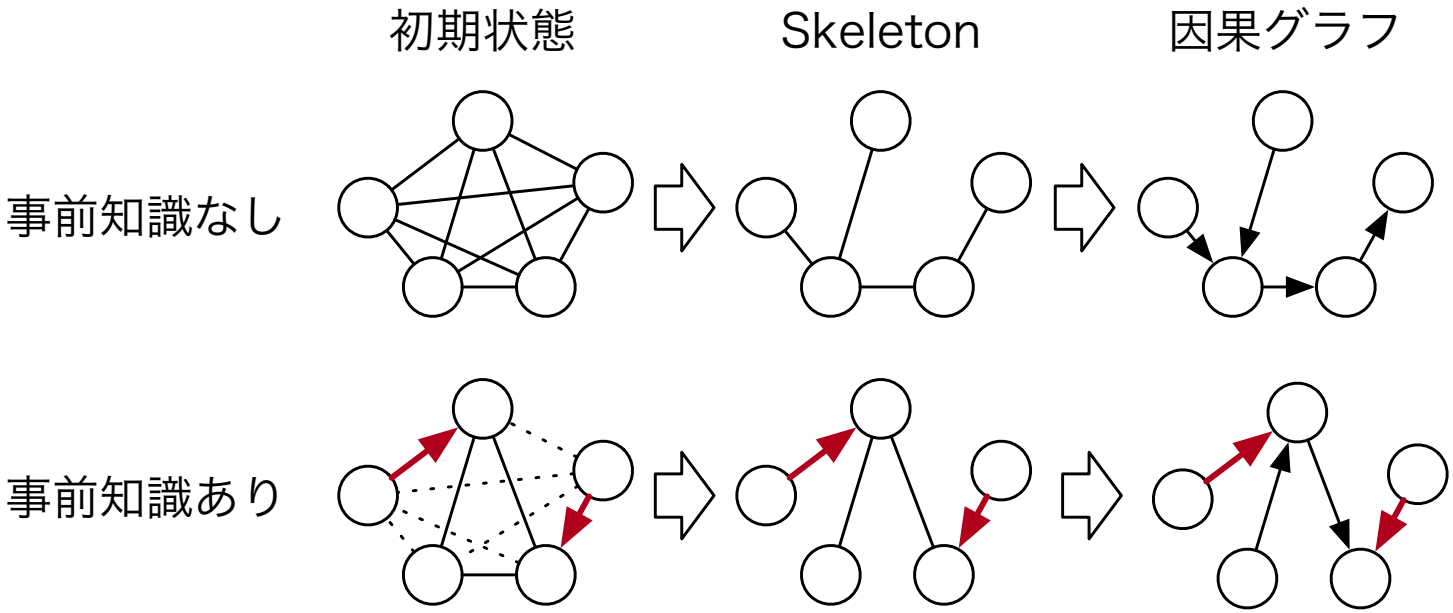
- 統計的因果推論によるログ因果解析
  - 多くのイベント間の因果関係を推定する因果探索手法を利用
  - 擬似相関を除いて直接的な因果関係に注目可能
- 事前知識を用いる因果探索
  - 与えられた事前知識を元に、一部の関係(因果エッジの有無)を固定
    - ➢事前知識が正しければ、得られる因果グラフの信頼性向上
    - ➢事前知識を多く与えると、探索空間を減らし処理時間削減

# 主な因果探索手法

既存手法(logdag)で
現在利用可能

| 手法名 | 分類 | 事前知識利用 | 実装公開 | 文献発表年 |
|---|---|---|---|---|
| GESアルゴリズム [18] | スコアベース | ✔ | ✔ | 2002 |
| DAGs with NO TEARS [21] | スコアベース | | ✔ | 2018 |
| DAG-GNN [22] | スコアベース | | ✔ | 2019 |
| DAGs with Tears [23] | スコアベース | ✔ | | 2023 |
| PCアルゴリズム [24] | 制約ベース | ✔ | ✔ | 2001 |
| MMHC [26] | スコア + 制約 | ✔ | ✔ | 2006 |
| DirectLiNGAM [28] | 非ガウス性 + 制約 | ✔ | ✔ | 2011 |
| THP [31] | Hawkes過程 + 制約 | | △ | 2021 |
| HPCI [30] | Hawkes過程 + 制約 | | | 2021 |

# PCアルゴリズムにおける事前知識の利用




因果エッジが存在する知識 (赤線)

因果エッジが存在しない知識 (破線)

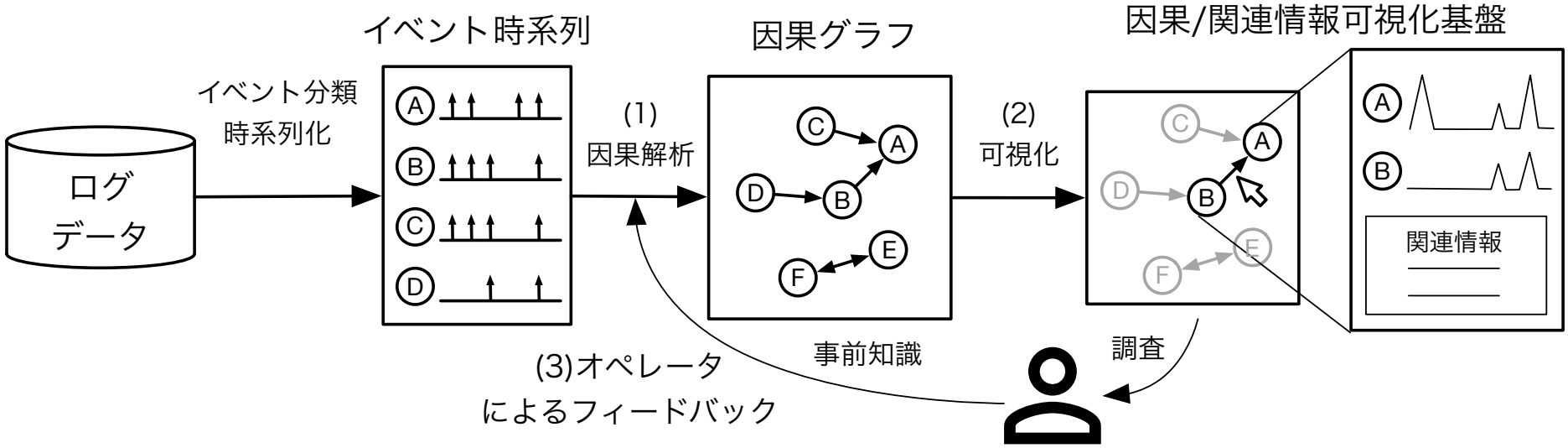
導入により、知識外の関係の推定結果も変化(改善)する

# 事前知識を用いたログ因果解析の今後の課題

- 因果探索手法の選択のため、比較評価が必要
  - 精度(信頼性)と処理時間(スケール性)の考慮が必要
  - 古い因果探索手法で事前知識を利用する場合と、新しい因果探索手法を(事前知識なしで)使う場合、どちらが良いのか?
  - 既存の比較[42]は用いたログデータが266行と極めて小さい
- ➢擬似データと実データを併用した因果探索手法の比較評価
  - 擬似データ: 正しい因果がわかるデータを取得可能、定量的評価
  - 実データ: SINETのログデータを利用予定、定性的評価

[42] V. Arya, et al. “Evaluation of Causal Inference Techniques for AIOps,” ACM SIGKDD CODS COMAD 2021, pp.188–192, 2021.

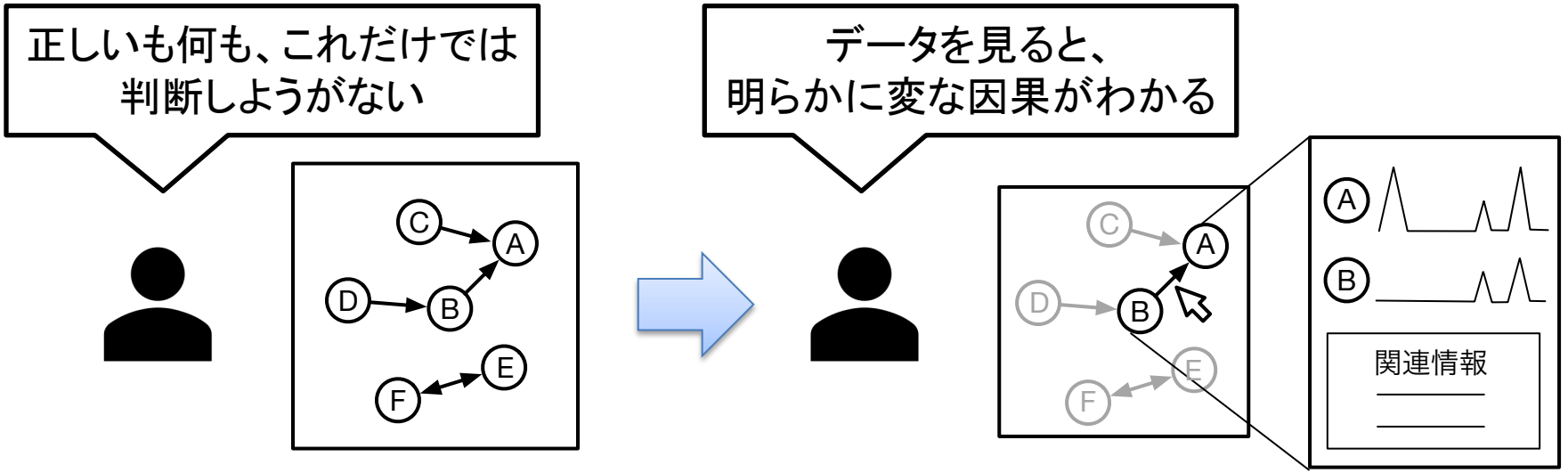
# 対話的因果解析の概要と3つの要素技術




1. 事前知識を用いたログ因果解析
2. 因果および関連する情報の可視化
3. 因果エッジに対するフィードバックの入力

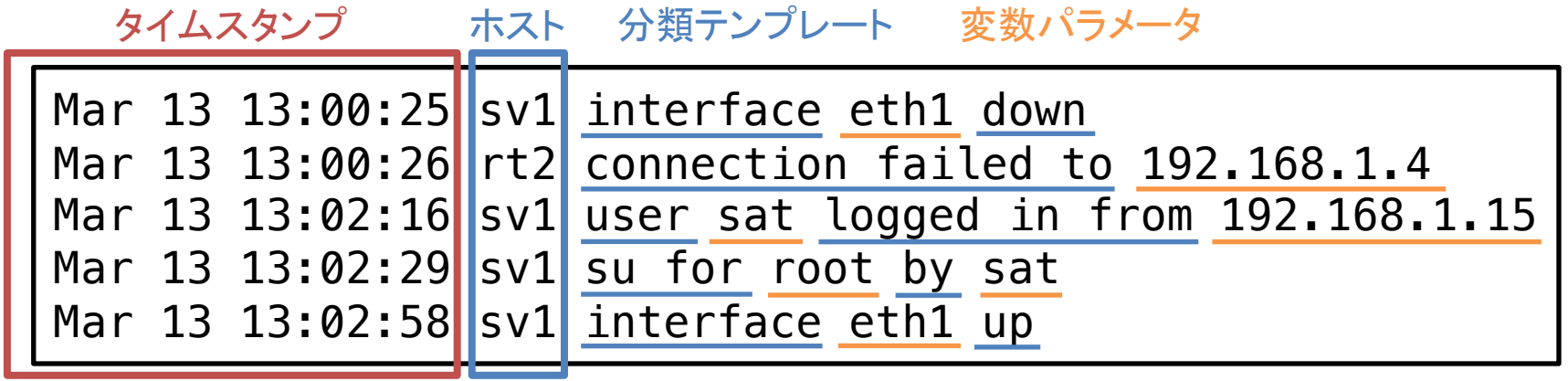
# (2) 因果および関連する情報の可視化

- 因果のみでは、システムの振る舞いを理解することは難しい
- フィードバック(因果の正誤判断)のためオペレータに情報提供



# フィードバックのためオペレータに提供すべき情報

1. ログの発生ホスト機器およびメッセージ情報
2. ログイベントの時系列上の振る舞い
3. ログメッセージ中の変数パラメータ情報



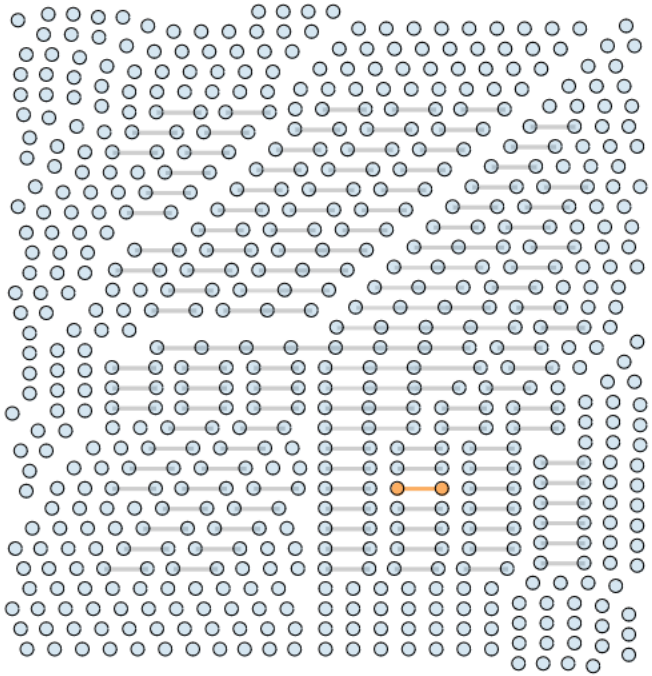
# 情報1: ログの発生ホスト機器およびメッセージ情報

- logdagの中間データから容易に取得可能
  - logdagではこれらの情報をイベント分類に用いているため
  - 因果グラフの各ノードが同一ホスト/メッセージフォーマットの集合
- 因果グラフのノードに注釈を付与
  - ノードにカーソルを乗せると表示される

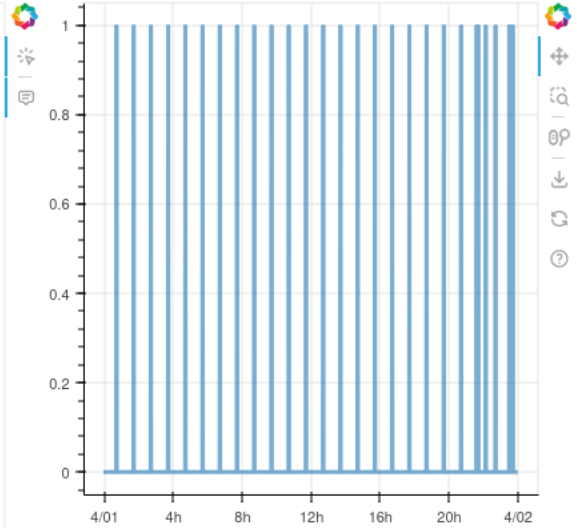

以降の可視化プロトタイプの実装には
Python向けインタラクティブ可視化ツールBokehを利用
https://bokeh.org/

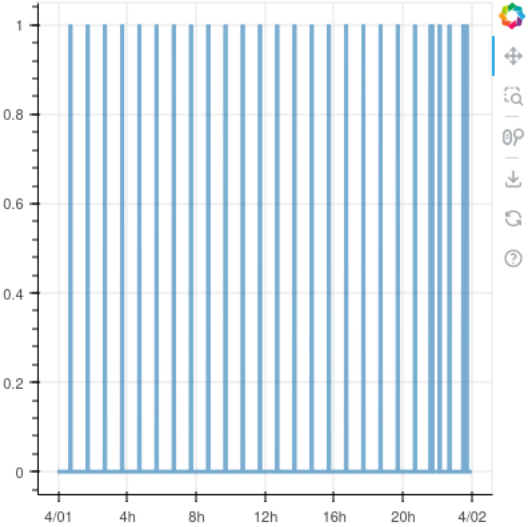
# 情報2: ログイベントの時系列上の振る舞い

- イベントの時系列を見ることで、システムの振る舞いを直感的に理解しやすくなる
- 有用なイベント時系列情報の例
  - 複数発生するイベントの発生パターン
    - 例: 一定周期で現れるもの、1度現れると繰り返すもの、ほぼランダムなもの
  - ノード間のイベント時系列の比較
    - どのような相関関係に由来して因果が見つかったのか
    - 例: 時系列上の完全な共起、一部のみの共起、偶然の共起など
- 選択されたノードの時系列をプロット表示



因果グラフ



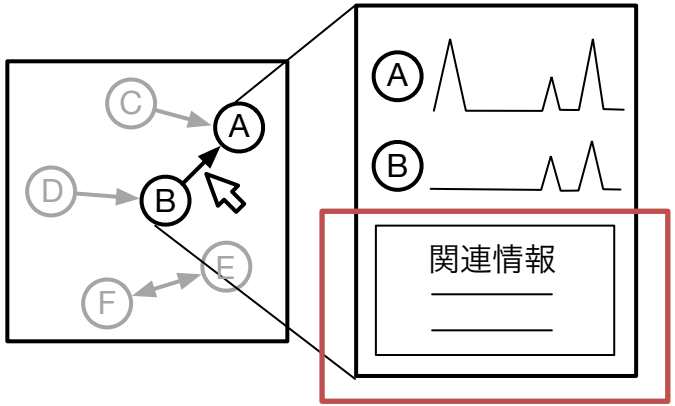




イベントの
時系列

グラフ上で因果エッジを選択すると
エッジ両端のノードの時系列が表示される

# 情報3: ログメッセージ中の変数パラメータ情報

- ログメッセージ中の変数パラメータの変化や異常値は、トラブルシューティングやフィードバックの検討に有用
  - 例: 障害発生時のみ値が異常
- まずは、メッセージの羅列で対処
  - スクロールバー付きテキストボックス
- 将来的には、パラメータ代表値や異常値の自動抽出により効率化




このあたりで
情報を提供?

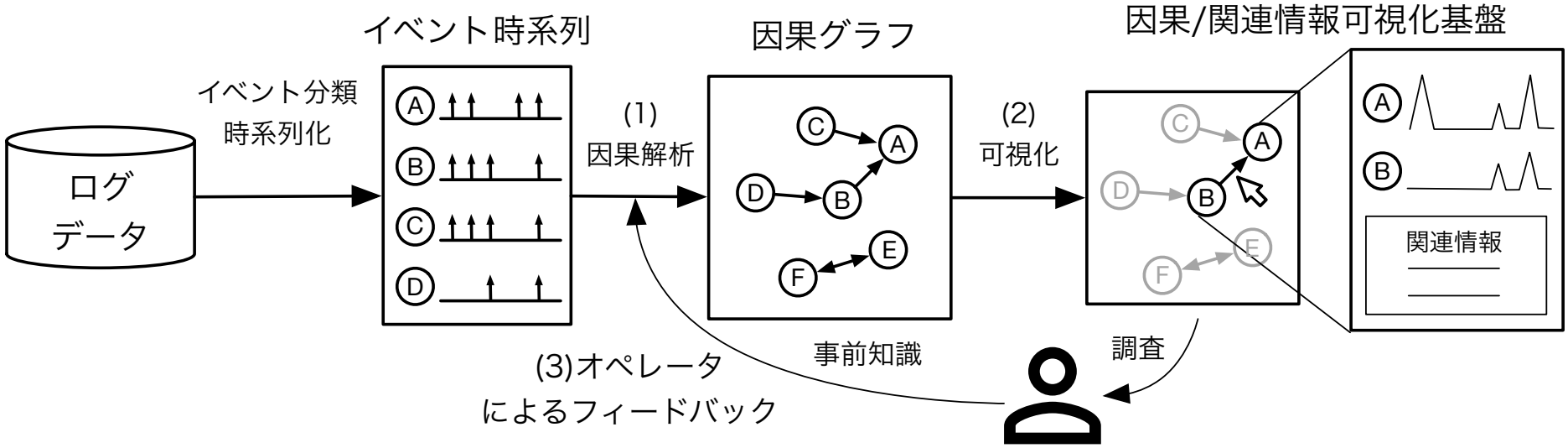
# 因果関連情報可視化における今後の課題

➢3つの関連情報について情報提供するプラットフォームを開発

1. ログの発生ホスト機器およびメッセージ情報
2. ログイベントの時系列上の振る舞い
3. ログメッセージ中の変数パラメータ情報

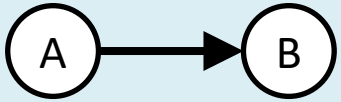
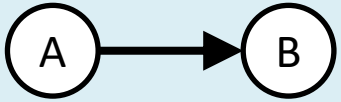
- ネットワークログ解析での利用を通して、課題等を整理

# 対話的因果解析の概要と3つの要素技術



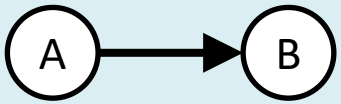

1. 事前知識を用いたログ因果解析
2. 因果および関連する情報の可視化
3. 因果エッジに対するフィードバックの入力

# (3) 因果エッジに関するフィードバックの入力

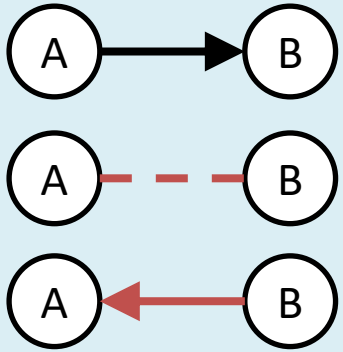
| 得られた因果グラフ | オペレータが考える実際の関係 | 5パターンのフィードバック |
| --- | --- | --- |
|  因果エッジあり |  | (1) True Positive |
| |  | (3) False Positive |
| |  | (5) False Direction |
|  因果エッジなし |  | (2) True Negative |
| |  | (4) False Negative |

# フィードバックの入力方法1 – 対話的入力

- 因果グラフ中の因果エッジを選択してフィードバック入力
  - 因果グラフに因果エッジが有るのでユーザが選択可能
- 利点: オペレータが直感的に入力可能
- 問題点: ルールの柔軟な記述はやや難しい




因果エッジあり




(1) True Positive

(3) False Positive

(5) False Direction

# フィードバックの入力方法2 – 記述型入力

- 外部ファイルにルールを記述して知識として読み込ませる
  - 因果エッジがないものはユーザの選択による入力が困難なため
- 利点: 個別のイベント組の事例ではなく一般化されたルールとして、柔軟な記述が可能
- 問題点: ユーザにとって対話的入力よりも手間がかかる

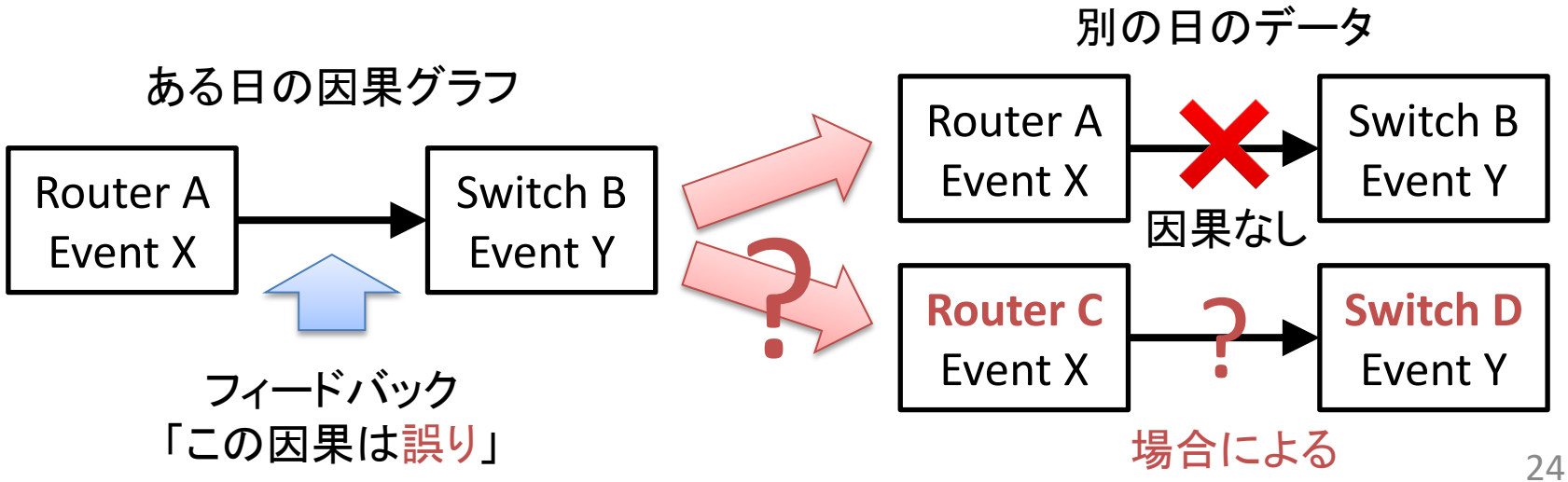

因果エッジなし


(2) True Negative


(4) False Negative

# フィードバックに関する今後の課題

- フィードバック記述ルールにおける抽象度の扱い
  - フィードバックがどの程度一般的な知識なのか? の情報が必要
  - 例: 異なるホスト間でも同じ知識/関係が成立するか?



# 議論

- フィードバック知識の誤りや競合の問題
  - システムの設定や構成の変更など、過去の知識の陳腐化
  - 複数のオペレータによるフィードバック(解釈の相違)

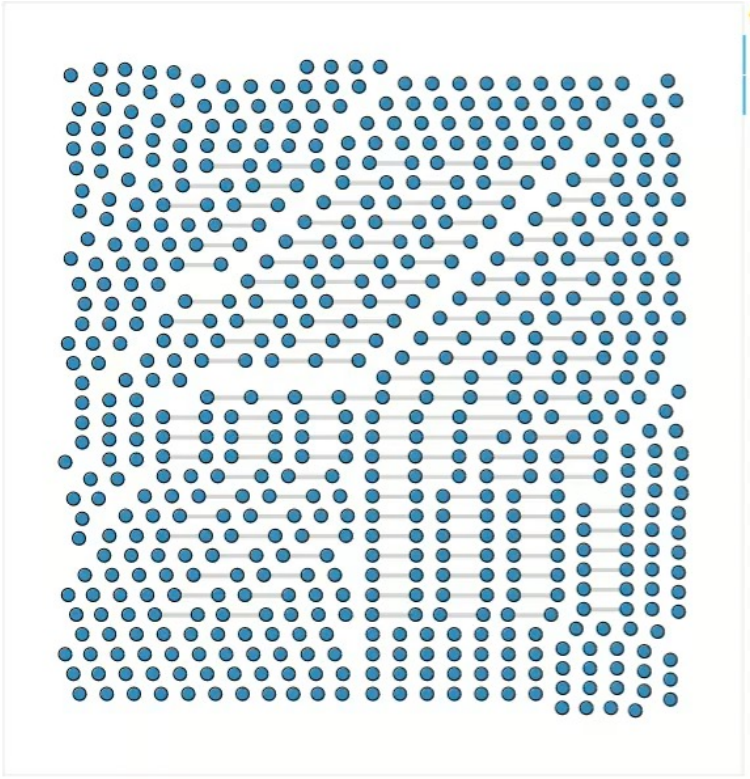
➢フィードバック知識の除去

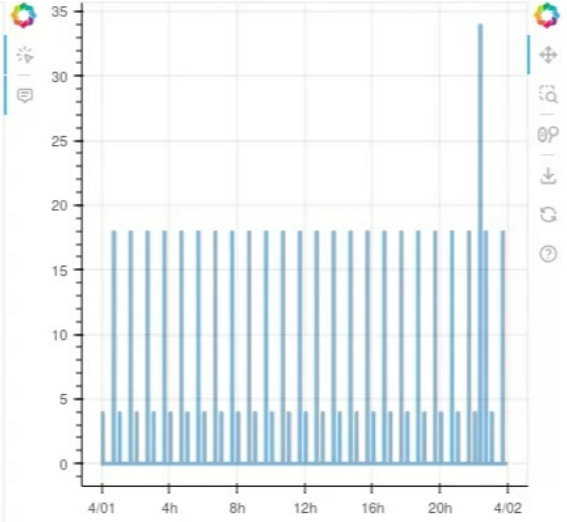
A) オペレータの操作により除去する方法
- 対話的入力/記述型入力の拡張

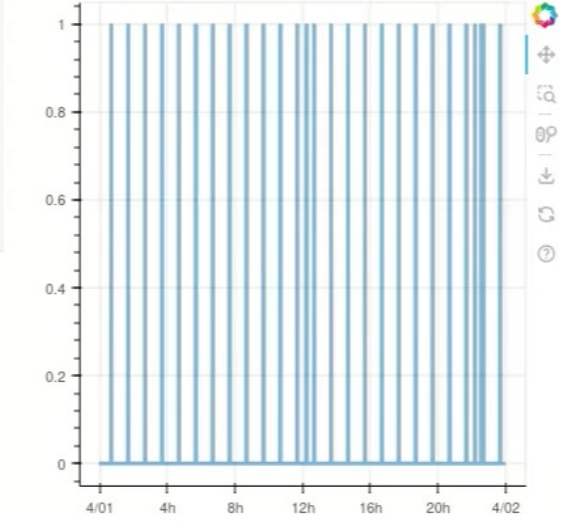
B) 一定の条件により知識を忘却させる方法
- 一定期間の経過、相反する知識の導入など

# まとめ

- ログ因果解析の信頼性を向上するため、オペレータの知識をフィードバックとして用いる対話的因果解析の仕組みを検討
- 3つの要素技術について課題を整理
  1. 事前知識を用いたログ因果解析
     - 因果探索手法選択のための比較評価
  2. 因果および関連する情報の可視化
     - 関連情報の可視化プラットフォーム開発、解析利用により課題整理
  3. 因果エッジに対するフィードバックの入力
     - 記述ルール・抽象度の扱いの検討



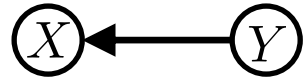

35
30
25
20
15
10
5
0
4/01
4h
8h
12h
16h
20h
4/02


1
0.8
0.6
0.4
0.2
0
4/01
4h
8h
12h
16h
20h
4/02

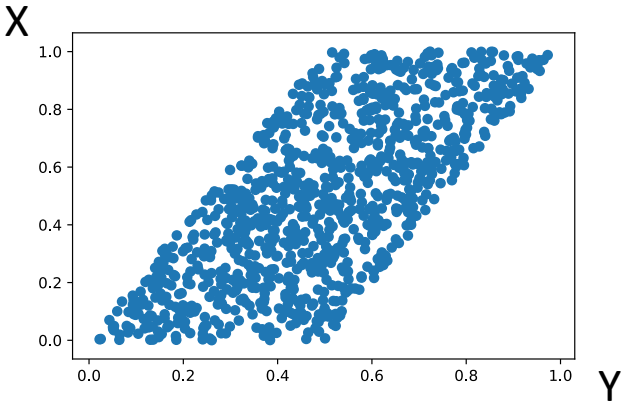
# LiNGAM: 非ガウス性と識別可能性

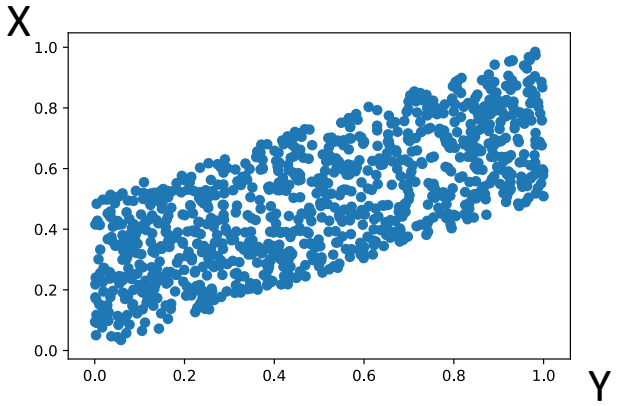

$$X = e_x$$
$$Y = 0.5X + e_y$$

eが一様分布の場合




$$X = 0.5Y + e_x$$
$$Y = e_y$$

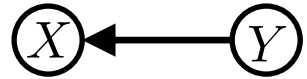


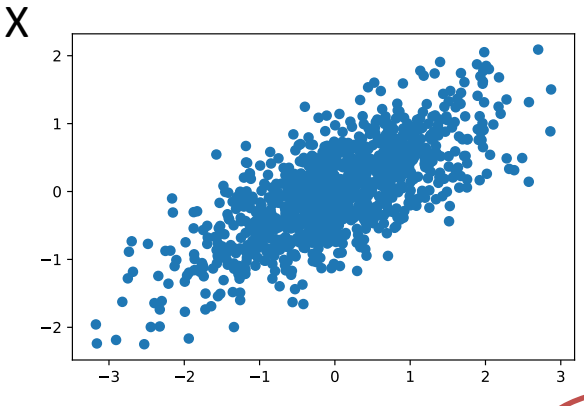
# LiNGAM: 非ガウス性と識別可能性

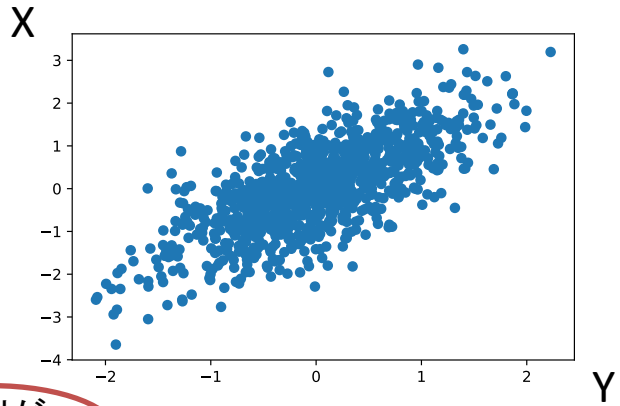

$$X = e_x$$
$$Y = 0.5X + e_y$$

eがガウス分布の場合




$$X = 0.5Y + e_x$$
$$Y = e_y$$




区別がつかない