平成24年度卒業論文

論文題目

Cl-GBI法によるふるまいのグラフの類似に基づく群れのモデルの提案

神奈川大学 工学部 電子情報フロンティア学科
学籍番号　200902799
安竹　有輝

指導担当者　木下宏揚　教授

目 次

図 目 次

第1章　序論

1.1　背景

従来のコンピュータ利用はユーザーがコンピュータのハードウェア,ソフトウェア,データなどを，自分自身で保有・管理していたのに対し，クラウドコンピューティングでは「ユーザがインターネットの向こう側からサービスを受け，サービス利用料金を払う」形になっている．ユーザーが用意すべきものは最低限の接続環境（パーソナルコンピュータ，携帯情報端末などのクライアント，それを動かす為のブラウザ，インターネット接続環境，加えてそれらサービスを利用するためのクラウドサービス利用料金）であり，とても手軽に利用できるので，広く普及している．

国内でも積極的にクラウド・コンピューティングを試行する先進的な企業や，すでに電子メールや顧客管理などでクラウドサービスを採用している企業も少なくない．クラウド・コンピューティングでは，提供される内容に応じてXaaSという言葉でその種類が表されるが，この最後のSはサービスを意味している，ここでの「サービス」にはもちろん，ITシステムの構築に必要となるハードウェアやソフトウェアを資産として購入するのではなく，利用に応じて料金を支払うという課金形式での提供形態という意味もあるが，SOAでいうところのサービス，つまり部品化されたソフトウェアという意味も含まれている．つまりSaaSが提供するアプリケーションをサービスとして利用し，企業内のシステムと連携させるということは，SaaSで提供されるソフトウェア部品を利用することに他ならない．企業もSaaSを通じてインターネット上でソフトウェアを部分的に業務に取り入れているのである[1].

このようなことからＩＴ系ではない一般企業も，ICT に詳しくない人もインターネットを何らかの形で日常的に利用していることが分かる．

1.2 問題定義

そのクラウドコンピューティングの中でもこの研究ではクラウドファイルシステムに注目した．クラウドファイルシステムとはファイルをサーバで一括管理し，特定のサーバ，ファイルに対してアクセス権限があれば利用できるというものである [2].

しかし，急速に発展し，巨大化・複雑化したインターネットが原因でアクセス権限も複雑に絡み合うようになっていて，その結果ネットワーク内では不正な情報経路が発生し，情報流出の危険性が増大してしまっている．これを covert channel と言う．

このような情報流出経路の分析法として Covert Channel 分析がある．しかしこの Covert Channel 分析には従来のように把握したコミュニティの ACL(Access Control List) のみを用いた Covert Channel の解析だけでは検出できないアクセス権の矛盾が存在する場合があるといった問題点がある．

そこでこの研究ではクラウドファイルシステムのアクセス行列と Covert Channel 分析の関係性に着目した．

1.3 これまでの研究

いままでの我が研究室の Covert Channel 分析では把握したコミュニティの ACL のみを用いた Covert Channel の解析を行ってきた．我が研究室の Covert Channel　分析の手法として Tanimoto 係数を用いた研究が挙げられる．また，外部の研究では後ろ向き推論システムなどがある [3][4].

1.4 新モデルの提案

このようにCovert Channel 分析には様々な方法でアプローチ，研究されてきたが，この研究ではCovert Channel 分析を群れ (=群知能) とアクセス行列を組み合わせてMasonと言うMulti agent simulatorを用いて行う．Covert Channel 分析の分野において，群れとアクセス行列をMasonに取り入れて行う手法はまだ着手されていないので，新しいアプローチ方法となる．

クラウドファイルシステムにおけるCovert Channelは人のふるまい，つまり人為的要因によって起こる．クラウドファイルシステムのクラウドファイルシステムのアクセスはsubject(人)，object(ファイル) 間のread，writeの連鎖である．つまりは，人のふるまいの群れである．なので，クラウドファイルシステムのアクセス行列に群れは有効活用できるという結論に至り，研究に群れを取り入れた．Masonにアクセス行列のふるまいの連鎖（ふるまいの群れ）を複数抜き出し比較し，求めた類似度をAgentの引力斥力と設定し，Covert Channel分析でCovert Channelをセパレートするのがこの研究の目的である．

第2章　基礎知識

2.1　クラウド・コンピューティング

インターネット上にグローバルに拡散したコンピューティングリソースを使って，ユーザーに情報サービスやアプリケーションサービスを提供するという，コンピュータ構成・利用に関するコンセプトのこと．

インターネットやTCP/IPネットワーク [5] は，しばしばクラウド（cloud＝雲）と表現される．

クラウドコンピューティングでは複数のコンピュータがグリッドや仮想化の技術で抽象化されネットワークで接続されたコンピュータ群が巨大な1つのコンピュータになるという，パラダイムシフトの意味が込められている．

適切な方法で“雲”＝インターネットに接続さえすれば，ユーザーは即座に各種のサービスが利用できるという点では，SaaS・ASPに近い．ただし，ASPの時代は特定のサーバファーム（データセンター）で処理を行ったが，クラウドではデータ処理が分散化される場合が想定され，このあたり“クラウド”というメタファーが使われる理由と考えられる．

クラウドコンピューティングは言葉が知られるとすぐにIT業界全体を巻き込む一大潮流となり，ITベンダ各社からさまざまな提案がなされている．クライド技術を社内システムに応用する考え方も登場しており，これはプライベートクラウドと呼ばれる．これに対して，当初のクラウドコンピューティングをパブリッククラウドということがある．

2.2 人の生活に浸透するインターネット

2.2.1 SaaS

SaaS(Software as a Service)[6][7] とは様々なソフトウェアの機能の中からユーザーが必要とする機能だけを選んでインターネットや閉域ネットワークを経由して利用できるサービスである.

2.2.2 PaaS

PaaS(Platform as a Service) とは，ソフトウェアを構築，稼働させるための基盤となるプラットフォームを，インターネット上のサービスとして利用する形態である [7].

2.2.3 Iaas

SaaS では，ユーザーはネットワーク・ハードウエア・OS・ミドルウエアのすべての要素について，自身で準備をする必要がない． ユーザーはWebブラウザを利用して，SaaS提供事業者のWebアプリケーションにアクセスし，サービスを利用する [7].

2.2.4 XaaS

情報システムの構築・運用に必要な何らかの資源 (ハードウェア，回線，ソフトウェア実行環境，アプリケーションソフト，開発環境など) をインターネットを通じてサービスとして遠隔から利用できるようにしたもの．また，そのようなサービスや事業モデル．

従来は購入したり固定的・長期的な利用契約を結んで利用してた様々な資源を，サービスとしてネットワーク越しに必要なときに必要なだけ利用し，実績に応じて代金を支払う形態を意味する．「サービスとしてのソフトウェア」(SaaS : Software as a Service) の概念を広げ，様々な要素に適用できるようにした用語である．

2.3 クラウドファイルシステム

クラウドファイルシステムとはクラウド上でファイルを管理するためのシステムである．複数のユーザーが一つのサーバーにファイルを read,write することが可能で，ファイルを共有できる [8]．クラウドファイルシステムと人の相関関係を図 2.1 に示す．

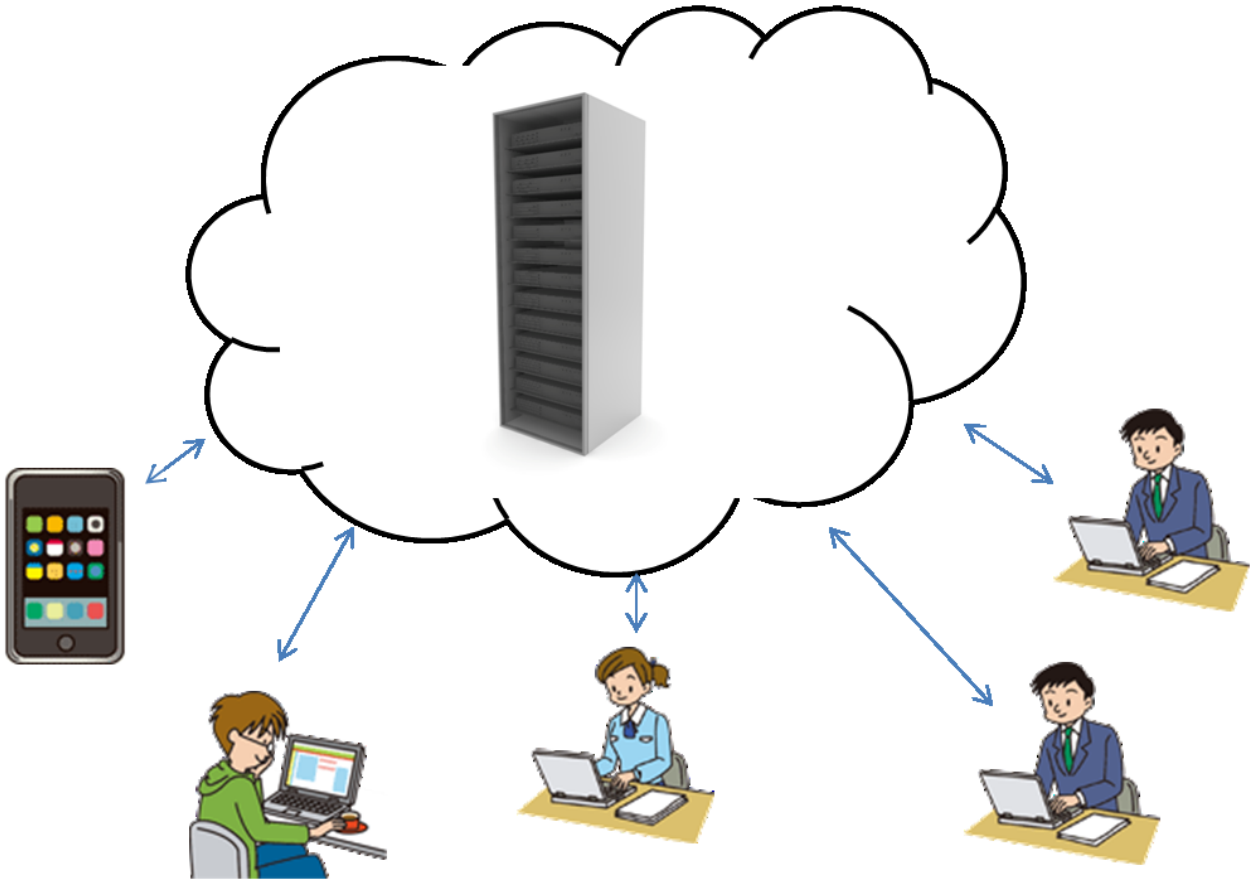

図 2.1: クラウドファイルシステムへアクセスする人たち

2.4 アクセス行列

アクセス行列とはユーザーがクラウドファイルシステムへのアクセス (read,write) の繰り返しの列を並べたものである．図 2.2 はアクセス行列の一例を示したものである．

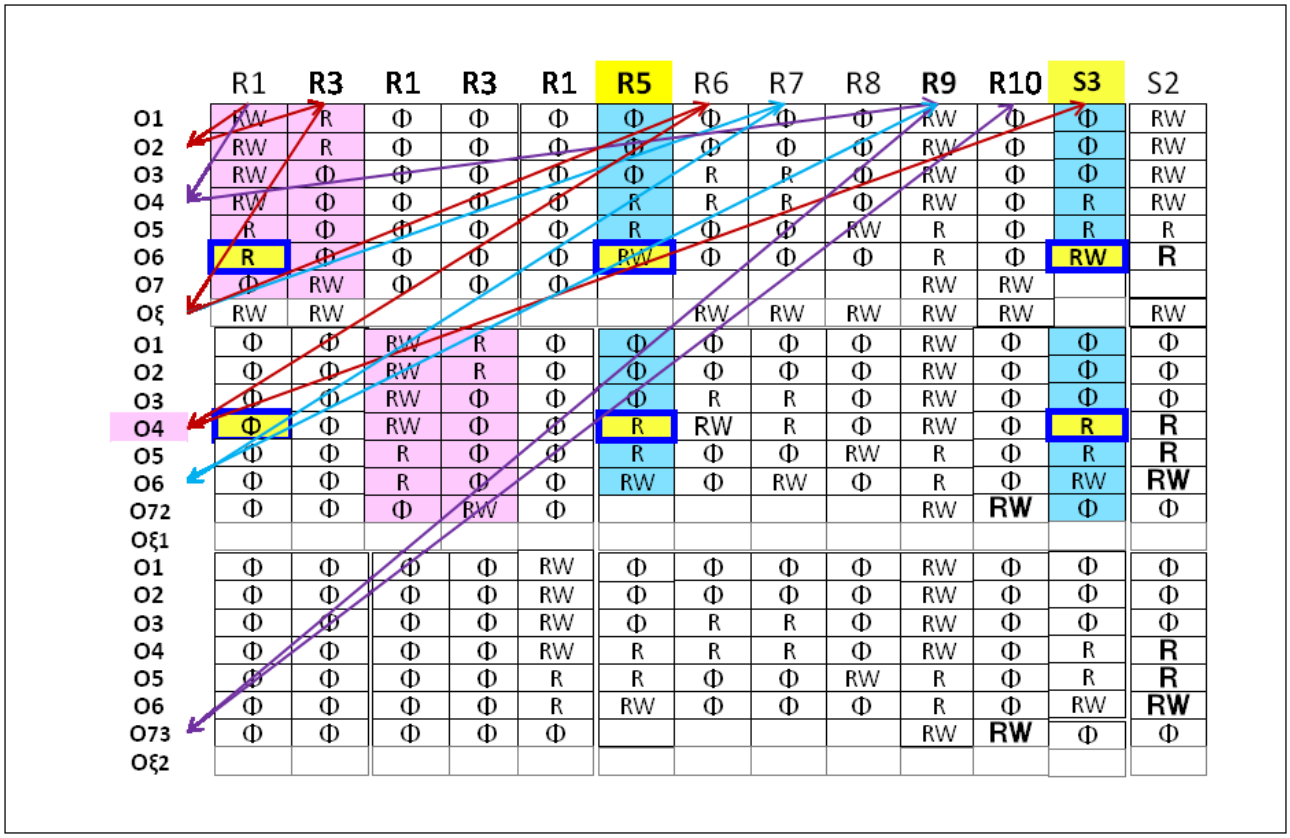

図 2.2: アクセス行列

2.5 家族的類似

ルートヴィヒ・ウィトゲンシュタインの言語ゲームで振る舞う particle は，家族的類似性によって群れを作ると定義した．家族的類似性は言語と行為の類似性を表現するものである．家族的類似性は同値関係でもないし，等価関係でもない．常に変動しつつ，少しづつ似ているエンティティの集まりである．しかし，それは自己から見れば，同値関係であってもよいし等価関係であってもよい．そのような個人個人の意味論を統合して世界を記述する意味論が無いということである．家族的類似性は不確実な世界の中の同類の定義である．家族的類似性は公的言語世界の観察視点の同類の定義である． 群知能において群れる振る舞いは家族的類似に基づいている．集まる力の源は,「家族的類似」であり，群知能のパラメータとして表現される [18].

群れる正の力

- 群れの中心に向かう力： Cohesion
- 隣人と家族的類似行為をする力： Alignment
- 行為濃度： Pheromone

2.6 Covert Channel

2.6.1 間接情報フロー

Covert Channel はアクセス行列において, Subject,Object, permission をアクセストリプルと定義したとき, その 3 点で起きる不正な情報経路である. この場合,Covert Channel はアクセス禁止のパミッションに矛盾する情報フロー（アクセス禁止のものもこのフローを使えばアクセス禁止の内容を閲覧したり, 書き換えることが出来てしまう）ともいう. （ 図 2.3 参照, 各 S=Subject, 各 O=Object,R=読みこみ可能権限,W=書き込み可能権限）図 2.3 の場合, 矢印の流れで Subject1 が本来読めないはずの Object1 を読めてしまう. Covert Channel 流出の流れは以下のようになっている. これは間接情報フローとも呼ばれる. ・始点（Subject2・Object1）Subject2 が Object1 を読み込む・中間点 1 （Subject2・Object2）Subject2 が Object2 に Object1 で読んだ内容を書き込む・中間点 2 （Subject1・Object2）Subject1 が Object2 を読む. ・終点（Object1・Subject1）Covert Channel により間接的に Object1 の内容を読めてしまうこのように不正な情報流出が発生してしまうため, アクセス制御を行う推論エンジンとしては出来るだけ発生を抑制し, 検出と訂正を的確に行えるようにするのが情報フィルタに必要な機能である [16][17].

	S1	S2
O1	Φ	R
O2	R	W

図 2.3: Covert Channel(間接情報フロー)

2.6.2 実際に発生する Covert Channel

不正な情報経路である Covert Channel を全て塞いでしまえば安全なシステムを構築することが出来るように見えるが, 単独では隠れチャンネル (Covert Channel) が存在しないようなコンピュータでもネットワークに接続されたコンピュータ群が協調することによって, 隠れチャンネルを構成できてしまう. つまり, 単独では安全なコンピュータでも, それがネットワークを構成すると安全ではなくなるような状況が簡単に存在し得るのである. このようなネットワーク構成機能の問題点が Covert Channel で利用される. 例えば以下のような例が挙げられる [16][17][19].

- 会社の機密データを社外へ持ち出したり，社外の人間（社外の PC）でも見れるようにする.mixi 等の SNS の個人データが掲示板やブログ等不特定多数へ流出
- スパイウェア等, 個人 PC から情報を持ち出すためにこれを用いて通信を行い, 検知を困難とする. WWW 等，不特定大多数が利用するネットワークでは意図しなくてもカバートチャンネルが発生してしまう恐れがあるのでそういった情報網では比較的安易に情報漏洩が起こりうる. このように Covert Channel は今のネットワーク社会にとって情報を安易に流出させてしまう存在なのである.

2.6.3 アクセストリプル

Subject,Object,permission をアクセストリプルと呼んでいるがそれぞれの意味についてまとめる. Subject は主体の意味でCovert Channel では主に人間（Object に対する権限がそれぞれ異なるユーザ）を指す. Object は客体の意味で Covert Channel では主にデータ（文字や画像等のことで見れるユーザ, 書きかえられるユーザがそれぞれ異なる）を指す. この Subject と Object には以上のように二項関係が成り立っていて互いの性質や関係を定義してやる必要がある. permission はファイル保護モードを言い,「READ」「WRITE」「EXE」の 3 種類の permission が,「ファイルの所有者」「グループユーザー」「その他のユーザー」のそれぞれに対して設定される権限をいう.

2.7 群れ（群知能

群知能（swarm intelligence）は，例えば鳥や昆虫の群れに見られるように，個体間の局所的な簡単なやり取りを通じて，集団として高度な動きを見せる現象（創発，等と呼ばれる）を模倣した計算手法として近年，研究が盛んになっている．全体を統御する指導者は無く，平等な立場の個体の相互作用が全体を決めるボトムアップな方法となる．進化型計算のうち，遺伝的アルゴリズムは交叉という個体間の相互作用があるので，群知能の一種と言える．群知能は進化型計算を行なうものも多いが，鳥の運動のシミュレーション等は，進化型計算ではない [11].

この研究では，“Cl-GBI 法” と “グラフ集合の特性を反映した構造類似性” を用いて類似度を求め，Mason の Agent に引力斥力として設定し，ふるまいが群れを成していることを示す.

2.7.1 ふるまい

集まる力の源は，“家族的類似” であり，群知能のパラメータとして表現される． 群れる正の力

- 群れの中心に向かう力： Cohesion
- 隣人と家族的類似行為をする力： Alignment
- 行為濃度： Pheromone

  Pheromone は行為の軌跡の重要性を表現する.

  Pheromone は揮発性である.

  濃度が濃い Pheromone は重要な行為を表す.

群れる負の力

- 群れから排除する力，群れから離れる力： Separation

2.8 Mason

- Multi Agent simulator である [12].
- 従来の Multi Agent simulator に比べて高速である.
- シミュレーションの様子を，客観的に第三者的（例えるなら神様のような）視点から観察することができる．(図 2.4 参照)
- ２次元・３次元の物理モデル・社会モデル専用である.
- 実装言語：Java
- 開発環境：なし (今回は Eclipse)
- 作成方法：Text
- 専用アプリケーション：無し

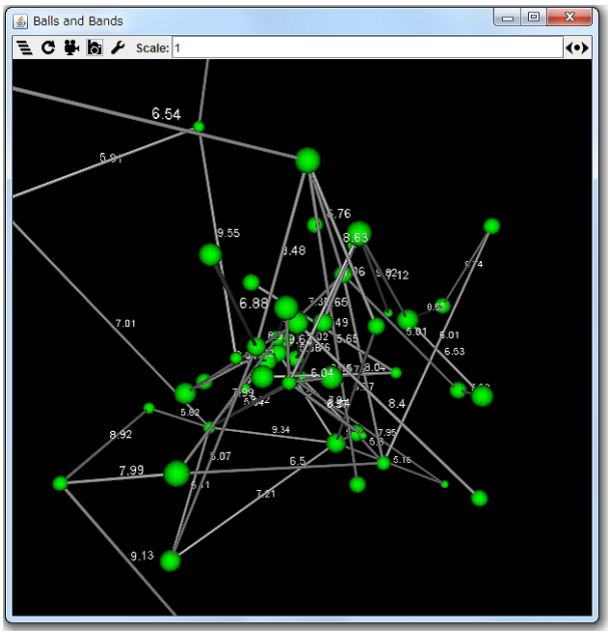

図 2.4: Mason のシュミレーション例 (ball)

2.9 Cl-GBI 法

大量に蓄積された電子データから興味深い有用な知識を獲得するデータマイニングにおいて，近年，複雑な構造を有するデータを扱うためにグラフ構造データを対象としたグラフマイニングが活発に研究されている [15]．その一手法である Graph Based Induction(GBI) 法は，ノードペアを遂次拡張 (チャンク) することにより，グラフ中に頻繁に現れる典型的なパターンを高速に発見することができる．また，GBI 法のチャンキング時の曖昧性及びチャンクすることによる探索空間の不完全性などの問題を軽減した Beam-wise GBI (B-GBI) 法も提案されている．しかしながら，GBI 法及び B-GBI 法は部分的に重複するパターンを同時に抽出できない．

Chunkingless GBI (Cl-GBI) 法では，ノードペアをチャンクせずに一つの塊として捉えること (疑似チャンキング) で重複パターンの抽出を可能とした．

2.9.1 データマイニング

データの中に潜んでいる価値ある情報や知識を掘り出すことを目的とした (大規模データに対応可能な) データ処理技術である．この研究では Cl-GBI 法という一つのデータマイニングを，大量のアクセス行列の中から制限を付けて特定のデータを抜き出すのに用いる [9].

2.9.2 グラフマイニング

コンピュータネットワーク，鉄道路線，道路交通網，神経回路などが形作る，幾つかの頂点（ノード）を結ぶ網の目状の構造を，グラフ構造と呼ぶ．例示から分かるように，この構造を持ったデータは多岐に渡り，なおかつそのネットワークの経路の最適化や構造推定，変化点の検出は，我々の生活基盤を大きく改善する可能性がある．信号の切り替えのタイミングが改善されれば，渋滞は緩和されるであ

ろうし，送電網が最適化されれば送電ロスが軽減され，電気料金は値下げされるであろう [10].

グラフマイニングとは，上記のような目標を達成するために，グラフ構造が持つ性質を調べる事を指す．そのため得たい情報に対応する様々な手法が存在する．簡単な一例 (図 2.5 参照) としては，ノードのグループ分けが考えられる

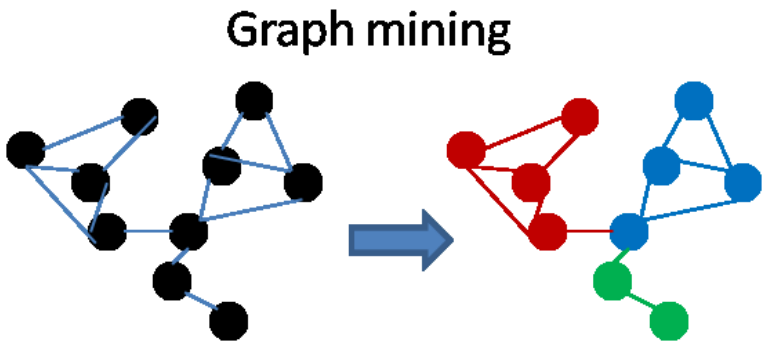

図 2.5: グラフマイニングの例

2.9.3 Cl-GBI 法のアルゴリズム

入力 : グラフデータベース D，ビーム幅 b，疑似チャングの繰り返しの最大数 N，最低支持度 θ

出力 : 典型的なパターンの集合 S(初期値は空集合)

Step1 D 中のグラフから隣接する 2 つのノードから成る全てのペアを抽出する．レベル 2 以降については，2 つのノードのうち少なくとも一方は新しく登録された疑似ノードから成るペアのすべてを抽出する．

Step2 抽出されたペアの頻度を数える．ここで，θ よりも低い頻度のペアは削除する．

Step3 “Step 1 ” で抽出されたペアの中から頻度の高い順に b 個のペアを選び，それぞれを抽出パターンとして S に加える．この時，

ペアを構成するノードが疑似ノードであれば元のパターンに復元してからSに加える．疑似チャンクすべきペアがない場合，もしくは，レベルがNの場合はここで終了する．

Step4 “Step3” で選ばれたペアにそれぞれ新しいラベルを割り当てる．ただし，グラフは書き変えない．そして，“その1” に戻る．

チャンキング過程を図2.6に示す．図2.6のSがアクセス行列のsubject，Oが Object を指し，subject，object の read，write となる．

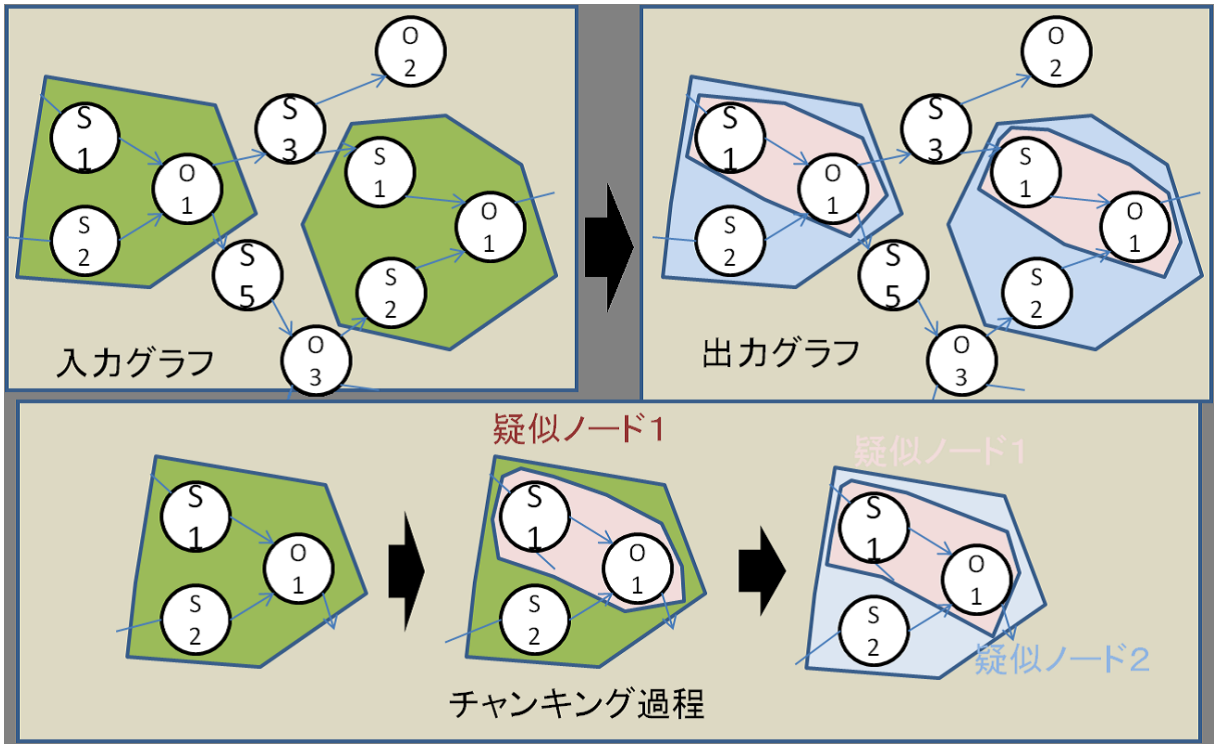

図 2.6: Cl-GBI 法を使用した例

2.9.4 深さ優先探索

深さ優先探索アルゴリズムは木の最初のノードから，目的のノードが見つかるか行き止まりまで深く下がって行く．スタートか根ノードから始まり，ずっと下ったところの葉ノードまで探索する．もし

ゴールノードが見つからなかったら引き返し，次のまだ通っていないツリーを葉に到達するまで探す [20].

第3章　モデルの提案

3.1　新モデルの流れ

このモデルはアクセス行列のグラフを抜き出し，形を比較し類似度を調べ，似ているもの同士群れを作り，そこからCovert channelをセパレートするというものである．そのモデル提案を下記に示す.

1. ふるまいの集まる力を求める.
   (a) アクセス行列のグラフに含まれるノードを用いて一致度，不一致数を求める.(図3. 1参照)
   (b) グラフのノード間類似度，ノード間相異値を求める.
   (c) グラフ間類似値，グラフ間相異値を求める.
   (d) グラフ間類似度を求める.(ふるまいが群れを成す為の，引力斥力になる.)
2. グラフの類似でふるまいが集まることをMasonで示す.
3. 群れのcovert channelを分析をする.
4. Covert channelをセパレートする.
5. 以上を行った上でも，群れが維持されていることを確認する.

3.2 Cl-GBI法で構造分布行列，グラフ，抽出パターンを求める（3.1 新モデルの流れの1.(a)から1.(d)までの詳細)

膨大なアクセス行列で制約として，一つもしくは複数抜き出したいペアを設定する (設定したペアの数分，大きな群れができる).

そこから疑似ノードが生成され，抽出パターンが抜き出される．その過程で構造分布行列，グラフが求まる.

3.2.1 不一致数，一致度を “構造類似性” を用いて求める.

二つのグラフのたグラフ，抽出パターンから一致度，不一致数を求める．(図 3.1 参照)

1. 比較する 2 つのグラフから同じラベルを持つノードをそれぞれ一つずつ用意する.
2. 構造分布行列を利用し，2 つのノードの一致度 (C)，不一致数 (E) を計算する．ただし，
   (a) 各部分構造毎に二つのノードに含まれている個数の最小値にその部分構造に含まれるノード数を重みとしてかけ，その総和を一致度とする.
   (b) 2 つのノードにおいて，各部分構造の数の差を計算し，その総和を不一致数とする.
3. 同じノードラベルを持つ他のすべてのノードのペアについて (2) の処理を行う.
4. 計算された各ノードペアの中で，一致度の割合が最も大きなペアのノードを各のグラフより取り除き各数値を保存する.
5. 同じラベルを持つノードのペアが存在しなくなるまで 1)〜4) の処理を繰り返す.

6. 保存されている一致度と不一致数のそれぞれの総数から一致度の割合を求め，これを類似度とする．

なお，参考文献を元にプログラミングを行ったのでグラフ中のノードを C と O とする．C=Carbon，O=Oxygen である．

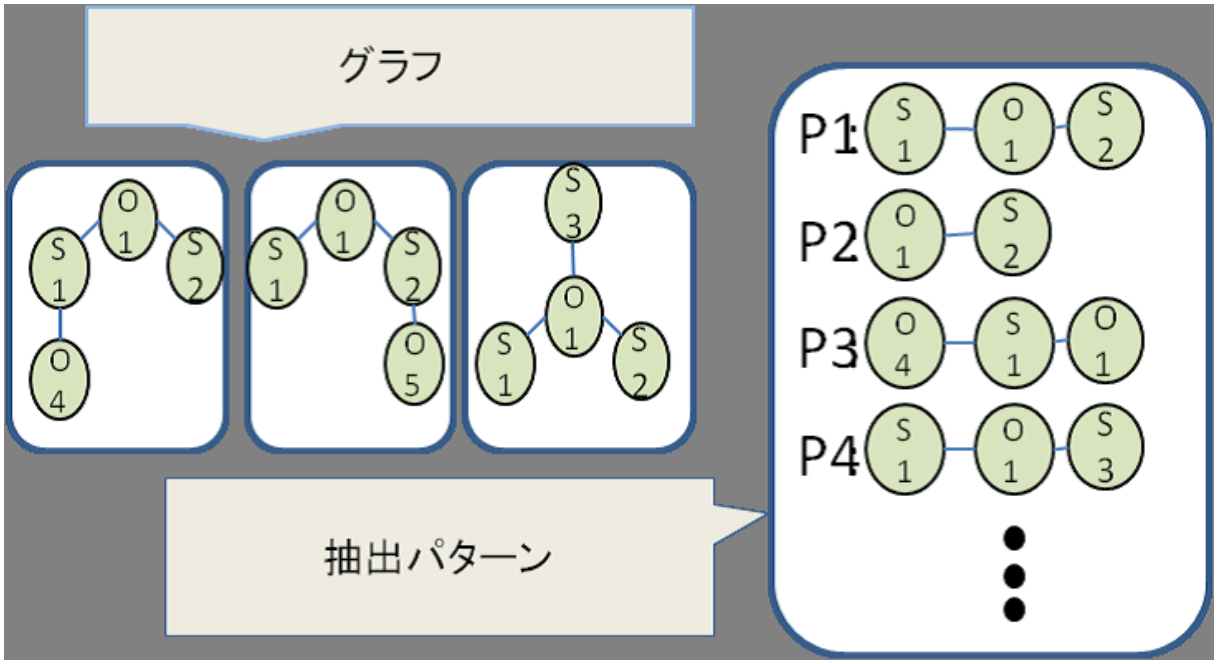

図 3.1: Cl-GBI 法を用いて求めたグラフ、抽出パターン

3.3 3.2のより詳しい解説

Cl-GBI 法により抽出された部分グラフを用いることによって，各グラフは，それを構成するノードと部分グラフとの関係に基づいて表現することができる．対象とするグラフ集合から抽出された部分グラフ集合をP，抽出された部分グラフ数をJ，ノードn_i^kを含む部分グラフ$p_j \in$ P,j =1,2,･･･，J,の数を$m_{ij}^k = m^k(n_i^k, p_j)$ とおくと，任意のグラフGk は以下に示す行列M_k で表現することができる[13][14].

$$M_k = \begin{bmatrix} m_{11}^k & m_{12}^k & \dots & m_{1J}^k \\ m_{21}^k & m_{22}^k & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ m_{I_k1}^k & \dots & \dots & m_{I_kJ}^k \end{bmatrix} \tag{3.1}$$

この行列M_kを構造分布行列と呼び，G_kの構造上の特徴が表現されているとみなす．これは，Cl-GBI 法の実行過程の情報から容易に作成することができる．

3.3.1 グラフ間類似度の算出方法

本稿では，まず，任意の 2 つのグラフ中に含まれる共通ラベルを持つノード集合ごとに，各ノード間の類似性を定義し，次に，そのノード間の類似性を用いてグラフ間の構造類似性を定義する．その際，各ノード間の類似性は，関連している部分グラフの共通数の多いノードペアから評価していくこととする．そのため，対象とするグラフ間で同一ラベルを持つノード数が異なる場合には，余分のノードは評価対象としない．たとえば，比較対象となる一方のグラフにのみ，あるノードラベルを持つノードが多数存在しても，その多くのノードは構造類似性の尺度には，直接は反映しない考え方である．しかし，一方にしか含まれないノードは，共通するノードと関連を持つ部分グラフの情報によって間接的に反映される．まず，任意のグラフ対に対して，構造類似性を表す尺度であるノード間類似度を

定義する．今，比較対象となるグラフをG_1,G_2とする．簡単のために，ノードラベルを一種類，ノード数については，G_1の方がG_2より多いとする．また，対象となるグラフ集合全体から抽出されている部分グラフの数をJ，それぞれの構造分布行列をM_1，M_2，さらに，部分グラフp_iを構成するノード数を$size(p_i)$とする．この時，グラフG_1のノードxとグラフG_2のノードyのノード間類似度r_{xy}^{12}およびノード間相異値d_{xy}^{12}を以下のように定義する[13][14].

$$r_{xy}^{12} = \sum_{j=1}^{J} \alpha_j \min(m_{xj}^1, m_{yj}^2) \tag{3.2}$$

$$1 \leq \alpha_j \leq size(p_j)$$

$$d_{xy}^{12} = \sum_{j=1}^{J} \beta_j |m_{xj}^1 - m_{yj}^2| \tag{3.3}$$

$$1 \leq \beta_j \leq \alpha_j$$

$$m_{ij}^1 = m^1(n_i^1, p_j) \in M_1$$

$$i = 1, 2, \cdots \quad , I_1$$

$$m_{ij}^2 = m^2(ni^2, p_j) \in M_2$$

$$i = 1, 2, \cdots \quad , I_2 \; I_2 \leq I_1$$

これらを用いて，ノード間類似度s_{xy}^{12}を以下のように定義する．

$$s_{xy}^{12} = \frac{r_{xy}^{12}}{r_{xy}^{12} + d_{xy}^{12}} \tag{3.4}$$

ここで，定義されたノード間類似度がより高くなるようなノードペアを選び出す．そのための準備として，2 つのグラフに含まれる部分グラフのうち最も大きな共通部分グラフを用意し，その部分グラフと関連を持つノードを各グラフより取り出す．そして，それらのノード群の中でノード間類似度を計算し，その値がより高くなるノードペアを見つける．続いて，残りのノード群の中でノード間類似度が高くなるノードペアを見つける．これは，比較するグラフに共通する最も大きな部分グラフに関連するノード同士はより高い類似度を得る可能性が高いと考えられるからである．最終的に I_2 個のノードペアを選び，それをグラフ間類似度を評価するための比較ノードペアとして確定する．その結果グラフ G_1，G_2 から選択されたペア (x_1, y_1)，(x_2, y_2)，$\cdots$ (x_{I_2}, y_{I_2}) に対して，グラフ G_1 と G_2 のグラフ間類似値 R^{12}，グラフ間相異値 D^{12} およびグラフ間類似度 S^{12} を以下のように定義する．

$$S^{12} = \frac{R^{12}}{R^{12} + D^{12}} \tag{3.5}$$

$$R^{12} = \sum_{i=1}^{I_2} r^{12}_{x_i y_i} \tag{3.6}$$

$$D^{12} = \sum_{i=1}^{I_2} d^{12}_{x_i y_i} \tag{3.7}$$

このようにして定義したグラフ間類似度は，次のような考え方に基づいている．

1. 各ノードの類似性を評価する際には，同一の部分グラフをもつ個数を加味して評価する．したがって，部分グラフが対象とするグラフ中にどのように分布しているかという点についても，類似性評価に加味されることになる．

2. 部分グラフのサイズを類似性評価に加味することができることとしている．これは，大きな部分グラフを共有するかしないかという点が，類似性の評価に大きく影響するであろうとの考え方による．ただし，類似値については，$1 \leq \alpha_j \leq size(p_j)$，相異値については，$1 \leq \beta_j \leq \alpha_j$ とする．これは，部分グラフのサイズの反映については，類似値が相異値を下回らないという条件を置いたことによる．

3.4 制限を付けたグラフ間の類似度を計算するアルゴリズム

3.4.1 制限内容

3.3節で述べたようにグラフ間の類似度を計算するアルゴリズムがある．正し，このアルゴリズムはあるパターンが他のパターンに含まれるか解く必要がある．部分グラフ同型問題は一般にNP問題であり，制限なしで計算するのは難しい．そこで本提案では，対象となるグラフや抽出パターンについて，以下のような制限を設ける．

1. 対象となるグラフは木構造である．
2. 抽出パターンは単純パスである．
3. 各ノードは2種類に分類できる．

3.4.2 制限を付けた2つのグラフの計算法

この制限の元，図3.1に示してあるグラフ１，グラフ２について一致度，不一致数の計算例を示す．

単純パスである各パターンの両端のノードのどちらかを始点とする．このときに，始点と終点を逆にしても同じパターンであれば，これを鏡像パターンと呼ぶことにする．各パターンに関して以下の計算を行う．

ここでは，パターンP1に関してのみ具体的な計算を示す．すなわち，単純パスであるP1の両端のノードのどちらかを始点とする．P1は鏡像パターンなのでどちらを始点にしても変わらない．

グラフ1のノードの中でパターンの始点と種類が同じノードを選択する．P1の始点の種類がCなので，グラフ１から選ばれるノードはノードラベル1,2,3のノードである．

選択したそれぞれのノードから深さ優先探索を行うことで一致するパターンを見つけるここでは，ノードラベル１に関してのみ具体

例を示す．すなわち，ノードラベル１を根として木構造であるグラフ１を深さ優先探索する．

この時，深さdのノードは抽出パターンP1の始点から数えてd番目のノードと種類が一致していない場合，それより深い部分に関しては探索しない．

または，現在探索しているノードの深さdとパターンP1の長さが一致している場合，それより深い部分に関しては探索しない．このとき、根から現在探索しているノードまでのパスを抽出パターンと一致した部分として，そのパスに含まれる各ノードに関してP1に対する一致数を１カウントアップする．

ノードラベル１から探索する場合，ノードラベル２が探索される．この時，ノードラベル２の深さが２なのでパターンP1の始点から数えて２番目のノードと種類が一致しているが，現在探索している深さ２パターンP1の長さが一致しているので，それより深い部分に関しては探索しない．このときノードラベル1, ノードラベル２と言うパスが抽出パターンと一致した部分となり，ノードラベル１とノードラベル２のパターンP1に対する一致数が１カウントアップされる．

3.4.3　一致度，不一致度を求め，類似度を算出する．

ノードラベル２，３に関しても同様に深さ優先探索を行うとノードラベル１，２，３のパターンP1に対する一致数はそれぞれ２，４，２である．

ここで，抽出パターンが鏡像パターンであれば，一致数は二重に数え挙げられているので，実際の一致数はその半分である．

したがってこの場合は，パターンP1が鏡像パターンなので，ノードラベル１，２，３，のパターンP1に対する実際の一致数はそれぞれ１，２，１である．グラフ２に関しても，同様に計算するとノードラベル１，２，３のパターンP1に対する実際の一致数はそれぞれ２，２，１である．グラフ１とグラフ２のノードラベル１，２，３のパターンP1に関する一致度はそれぞれのグラフのP1に関する一致数の最小値にパ

ターンP1の長さを重みとして掛けたものである．したがって2，4，2になる．グラフ1とグラフ2のノードラベル1，2，3のパターンP1に関する不一致数はそれぞれのグラフのP1に関する一致数の差である．したがって1，0，0となる．

同様にP2,P3,P4に対しての一致度，不一致数を計算し，各パターンに関する一致度の総和をグラフ1とグラフ2の一致度，各パターンに関する不一致数の総和をグラフ1とグラフ2の不一致数とする．

一致度と不一致数における一致度の割合を類似度とする．深さ優先探索については2.9.4節を参照すること．

制限を付けたグラフ間の類似度の計算結果をを図3.2に示す．

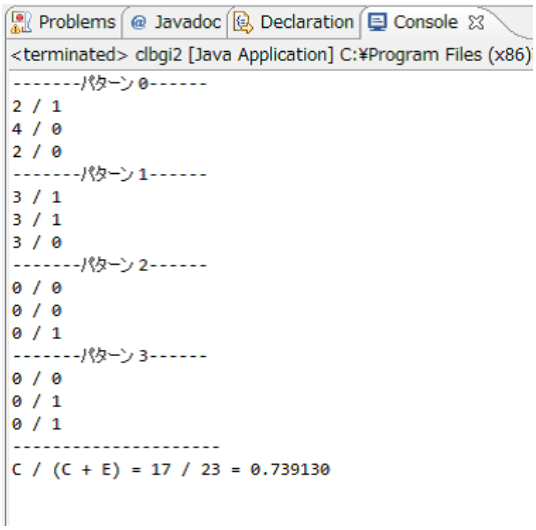

図 3.2: 木構造を用いた例

第4章　結論

図4.1のグラフ1とグラフ2を使い，プロトタイプのプログラミングを作り類似数，不一致数を求めることに成功した．今回は参考にした論文を元にプログラミングを組んだので，自分の求めたい値を思い通りに出すことができず，プログラミングの入力部を改良する必要があると感じた．Covert Channel分析をどのように群れから探し出し，セパレートするかが今後の最も大きな課題である．　本研究ではアクセス行列のグラフの “形” に重点を置いて行ったが，他に群れの中のふるまいには “メンバー”，“意味” がある．“メンバー” はアクセス行列のノードの中身に注目したふるまい(Tanimoto係数等)であり，“意味” はセキュリティーモラルに注目したふるまいである．セキュリティーモラルには「競合」,「所有」,「プライバシー」,「役割」,「階層」などがある．

今後の研究として “メンバー”，“意味” に重点を置くことで，アクセス行列を用いたセキュリティーに新しい面からアプローチできると考えている．

第５章　謝辞

本研究を行なうにあたり，終始熱心に御指導していただいた木下宏揚教授，宮田純子特別助手に心から感謝致します．また，様々な面で数多くの有益な御助言をしていただいた東洋ネットワークシステムズ株式会社の森住哲也氏，株式会社ユニテックの市瀬浩市氏に深く感謝致します．さらに，研究活動一般に様々な助言をいただきました南出氏をはじめ公私にわたり良き研究生活を送らせていただいた木下研究室の方々に感謝致します．

2013年　2月
安竹　有輝

参考文献

[1] ”クラウド・コンピューティングとSOAの関係”
http://www.enterprisecioforum.com/ja/article/

[2] ”クラウドシステム”
http://www.gms-web.com/index.php?page_no=5

[3] 鈴木久夫，臼田啓介，辻井重男，森住哲也，辻井重男：”後向き推論システムを用いたCovert Channelの検証”

[4] ”推論システム”
http://rain.ee.kanagawa-u.ac.jp/~uchida/9_22.html

[5] ”TCP/IP”
http://www5.plala.or.jp/vaio0630/tcpip/tcpip.html

[6] ”SaaS”
http://tm.softbank.jp/business/white_cloud/saas/

[7] ”SaaS/PaaS/IaaSとは”
http://itpro.nikkeibp.co.jp/article/Keyword/
20110216/357282/

[8] 秋山聖登，田又裕一，大谷真:”クラウド(GAE)上のwebファイルシステムの検討”

[9] ”データマイニングとは”
http://www.datamining.sakura.ne.jp/11haikei.html

[10] ”グラフマイニング”
http://www.eb.waseda.ac.jp/murata/research/graph

[11] ”群知能”
http://www.sist.ac.jp/~kanakubo/research/swarm_intelligence.html

[12] ”Mason”
http://cs.gmu.edu/~eclab/projects/mason/

[13] 和田　貴久，大野　博之，稲積　宏誠
”部分構造情報を用いたグラフクラスタリング手法の検討”

[14] 和田　貴久，大野　博之，稲積　宏誠
”対象グラフ集合の特性を反映した構造類似性の提案”

[15] 高林　健登，Phu Chien Nguyen，大原　剛三，元田　浩，鷲尾　隆
”グラフ構造データからの特徴的なパターン抽出における制約に基づく探索制御”

[16] 久保　直也
”検索エンジンによるCovert Channelの検出”神奈川大学　木下研究室

[17] 磯村　淳
”クラウドファイルシステム”神奈川大学　木下研究室

[18] 内山　竜佑
”多様性を実現する群知能のふるまいのモデル”神奈川大学　木下研究室

[19] 中村　峻生
”推移閉包アルゴリズムを用いたCovert Channel検出”神奈川大学　木下研究室

[20] 探索アルゴリズム
http://cis.k.hosei.ac.jp/~rhuang/Miccl/AI-0/2012-AI-0-L4-Jver.pdf

第6章　質疑応答

- Q. アクセス行列をどこで区切るのか.

A. 区切る定数を決めて，連続するアクセス行列をランダムに定数分区切る.