

平成 18 年度

筑波大学第三学群情報学類

卒業研究論文

題目

データベースに内在するネットワークを可視化するための
対話型インタフェース

主専攻 情報科学主専攻

著者名 小池諭

指導教員 田中二郎 三末和男 志築文太郎 高橋伸

要 旨

関係を表すデータにはネットワーク構造を持つものが多く、その構造を把握するには視覚的な表現が不可欠である。しかしながら、データの管理ツールとして普及しているスプレッドシートとネットワーク可視化ツールは十分に連携しているわけではなく、データベースにあるデータを視覚的に把握するには、データの抽出、変換などの煩雑な作業を利用者に強いることになる。本研究では、データベースに内在するネットワークの抽出、可視化の支援を目的として、データベースに内在するネットワークとその抽出法について考察し、表形式のデータからネットワーク図を描画するための情報を対話的に抽出・獲得するインタフェースを開発する。本研究のシステムを利用することでスプレッドシートから容易にネットワーク図を作成できるようになり、ネットワーク図によるデータの分析・利用がより身近になると考えられる。

目次

図目次.....	3
表目次.....	4
第1章 序論	5
1.1 ネットワークの可視化.....	5
1.1.1 関係を表すデータ.....	5
1.1.2 ネットワークとネットワーク図.....	5
1.2 関係データの管理・可視化技術の現状.....	6
1.2.1 スプレッドシート.....	6
1.2.2 ネットワーク可視化ツール.....	6
1.3 現状の問題点.....	6
1.4 研究の目的と構想.....	7
1.5 本論文の構成.....	7
第2章 データベースに内在するネットワークとその抽出・可視化手法	8
2.1 データベースとそこに内在するネットワーク.....	8
2.1.1 データベース.....	8
2.1.2 データベースに内在するネットワーク.....	9
2.2 ネットワークの抽出法.....	10
2.2.1 関係性に基づくネットワークの抽出法.....	10
2.2.2 類似度に基づくネットワークの抽出法.....	12
2.3 ネットワークの可視化手法.....	16
2.4 ネットワークの抽出・可視化の流れ.....	16
第3章 ネットワークを可視化するための対話型インタフェース	17
3.1 ネットワーク抽出における問題点.....	17
3.2 ウィザード.....	17
3.3 ネットワーク可視化のためのウィザード.....	18

第 4 章 ウィザードの設計と実装	19
4.1 ウィザードの設計	19
4.1.1 ウィザードが獲得すべき情報	19
4.1.2 ウィザードのインタフェース	19
4.1.3 情報を提示する順序	20
4.1.4 データ変換	20
4.2 可視化手法	22
4.3 ウィザードの実装	23
4.3.1 システムの構成	23
4.3.2 入力フォームの実装	24
4.4 使用例	28
第 5 章 評価と考察	32
5.1 評価.....	32
5.2 考察.....	33
5.3 今後の課題	33
第 6 章 関連研究	34
第 7 章 結論	35
謝辞	36
参考文献	37

図目次

図 1 : 表形式で表したデータベース	8
図 2 : 関係表の構成図	10
図 3 : 関係表の別の視点	12
図 4 : フィールドを 2 つ配置した関係表	13
図 5 : 類似度表	14
図 6 : ネットワークの抽出・可視化の流れ	16
図 7 : 入力・選択のためのインタフェース	19
図 8 : GraphML によるグラフの記述例	21
図 9 : CSV によるグラフの記述例	22
図 10 : システム構成図	23
図 11 : データ抽出フォーム	24
図 12 : 「可視化対象」フレームの変化	24
図 13 : 「抽出フィールド」フレームの変化	25
図 14 : 「エッジの重み」フレームの変化	25
図 15 : 可視化手法選択フォーム	26
図 16 : 「レイアウト手法」フレームの変化 (1)	26
図 17 : 「レイアウト手法」フレームの変化 (2)	27
図 18 : 属性付加フォーム	27
図 19 : スプレッドシートで管理された購買履歴のデータベース	28
図 20 : 「ネットワーク抽出」ツールバー	28
図 21 : データ抽出フォームでの選択	29
図 22 : 可視化手法選択フォームでの選択	30
図 23 : メッセージボックス	30
図 24 : 「購入者と商品カテゴリーの関係性」の可視化結果	31

表目次

表 1：作業ステップの比較（関係性）	32
表 2：作業ステップの比較（類似度）	33

第1章 序論

1.1 ネットワークの可視化

1.1.1 関係を表すデータ

知人関係（人脈）は「人と人との関係」を表す情報であり、商品の購買情報は「顧客の年代とその年代に買われている商品との関係」や「商品とそれが売れる時間帯との関係」などを表す情報である。これらの関係情報はビジネスにおける貴重な情報源であり、様々なシーンで利用されている。例えば、コンビニエンスストアでは、購買履歴から年代別や時間別の商品の売れ行きなどの購買傾向を読み取り、営業戦略に役立てている。また、インターネットのオンラインストアでは顧客の商品購買履歴から顧客の買い物の共通点を調べて顧客の嗜好の類似度を読み取り、顧客に商品を紹介、推薦するサービスを提供している。

データベースによるデータの保存・管理技術が発展した現在、様々なデータがデータベースに保存されている。前述した関係情報は、データベースに保存されているデータ群からデータを抽出することでも得られる。本論文では、関係情報を含むデータ群を「関係データ」と呼ぶ。データベースから関係情報などの有益な情報を抽出する技法はデータマイニング技法のひとつとして研究がなされ、その技法も様々なものが開発されている。

関係データは表形式で保存・管理されることが多い。これは、新しいデータが書き加えられていくデータベースの管理にはデータの蓄積・管理が容易な表形式が適しているためと考えられる[9]。

1.1.2 ネットワークとネットワーク図

関係データはネットワーク構造を成すことが多い。関係データのネットワーク構造を把握することで有益な情報を発見しやすくなる。しかし、表形式で表現された関係データからネットワーク構造を把握することは容易でない。関係データのネットワーク構造を把握するには視覚的な表現が不可欠である。ネットワークを視覚的に表現する手法としてネットワーク図が挙げられる。ネットワーク図を用いることでネットワークの構造を視覚的にとらえることができる。

1.2 関係データの管理・可視化技術の現状

1.2.1 スプレッドシート

データベースに蓄積されたデータの管理・閲覧にはスプレッドシートがしばしば利用される。スプレッドシートは表形式のデータを表示、編集するためのツールであり、データベース管理のための機能を多く備えていることから、現在広く利用されている。また、表計算機能だけでなく表の作成機能やグラフ描画機能も備えており、管理しているデータから表やグラフを作成することも容易である。

スプレッドシートは既に開発・実用化されており、多くの製品がリリースされている。

1.2.2 ネットワーク可視化ツール

これまでネットワークの可視化に関する研究は数多く行われてきた。ネットワーク図の自動描画法についても研究され、コンピュータでネットワーク図を描くためのアルゴリズムも数多く開発されている[1]。

現在、多くのネットワーク可視化ツールが開発されている。Yeeらは従来のグラフレイアウト手法を発展させ、同心円や螺旋に基づく配置を行う手法を提案している[13]。J.Heerらは、ネットワーク図の描画における効果的な視覚表現を数多く取り入れた情報可視化のためのツールキットとして *prefuse* を実装している[2]。

1.3 現状の問題点

スプレッドシートで管理されている関係データのネットワークを可視化するには、表形式で表されているデータからネットワークを構成する関係情報を抽出し、可視化ツールに対応するデータ形式へと変換する作業が必要になる。しかしながら、現在これらの作業はシステムで十分にサポートされているわけではなく、作業としては煩雑で、作業者の知識や技術に頼る部分が多い。そのため、ネットワーク図による可視化はネットワーク構造の把握に有効な手段であるにもかかわらず、広く普及しているとは言い難い。

1.4 研究の目的と構想

本研究は、データベースに内在するネットワークの抽出・可視化を支援することを目的とする。本研究ではこの目的に対して、ネットワーク図による可視化技術を利用しやすい環境を作るといふというアプローチをとった。本研究では、スプレッドシート上で簡単な操作を行うだけでネットワーク図を描画できる対話型のヒューマンインタフェースを開発することを目指す。

本研究ではまず、データベースに内在するネットワークの種類や抽出法を考察し、抽出に必要な情報を整理する。次に、それらの情報を対話的に獲得するインタフェースを設計、実装し、評価と考察を行う。

1.5 本論文の構成

本論文の構成は以下の通りである。本章ではネットワーク可視化の背景と問題点を説明し、研究の目的と構想を述べた。第2章ではデータベースに内在するネットワークの抽出法について述べ、第3章でネットワーク抽出を対話的に行うインタフェースを提案する。第4章ではシステムの設計と実装について述べる。第5章で評価と考察を行い、今後の課題などについて議論する。第6章で関連研究を紹介し、第7章でまとめる。

第2章 データベースに内在するネットワークとその抽出・可視化手法

本章では、データベースに保存された関係データから抽出できるネットワークとその抽出、可視化手法について述べる。本章での議論を基に、次章でそれらを効果的に実現するインタフェースを提案する。

2.1 データベースとそこに内在するネットワーク

2.1.1 データベース

本論文で扱うデータベースを以下のように定式化する。

$$\text{レコード } r_i = [f_{i,1}, f_{i,2}, \dots, f_{i,n}]$$

$$\text{データベース } D = [r_1, r_2, \dots, r_m]$$

このデータベースを表形式で表す場合、データベースの中にあるそれぞれの要素をセルに配置した形となる（図1）。

	f_1	f_2	\dots	f_n
r_1	$f_{1,1}$	$f_{1,2}$	\dots	$f_{1,n}$
r_2	$f_{2,1}$	$f_{2,2}$	\dots	$f_{2,n}$
\vdots	\vdots	\vdots	\ddots	\vdots
r_m	$f_{m,1}$	$f_{m,2}$	\dots	$f_{m,n}$

図 1：表形式で表したデータベース

2.1.2 データベースに内在するネットワーク

データベースにおいて同一のレコードに記録された要素の間には関係があると言える。その関係に基づいてネットワークが構成される。本論文では、データベースに内在するネットワークとして「関係性に基づいて抽出されるネットワーク」と「類似度に基づいて抽出されるネットワーク」の2つに着目した。

関係性に基づいて抽出されるネットワーク

データベースの異なる2つのフィールドの要素をノードとし、レコードごとの2つのフィールドの要素間の関係をエッジとしたネットワークが存在する。

データベース D の2つのフィールド f_p と f_q との関係性に基づくネットワークは以下のように表せる。

グラフ $G_r = (V, E, relation)$

ノード $V = F_p \cup F_q, F_p : p$ 番目のフィールドに現れる要素全体の集合

エッジ $E \subseteq F_p \times F_q$

$relation : E \rightarrow R$

$relation$ はエッジの重みを求める関数である。

このグラフは $F_p \cap F_q = \phi$ のとき2部グラフとなり、 $F_p \cap F_q \neq \phi$ のとき有向グラフとなる。

類似度に基づいて抽出されるネットワーク

データベースの1つのフィールドの要素をノードとし、要素間の類似度をエッジとしたネットワークが存在する。

データベース D のフィールド f_p の要素間の類似度を表すネットワークは以下のように表せる。

グラフ $G_s = (V, E, similarity)$

ノード $V = F_p$

エッジ $E \subseteq F_p \times F_p$

$similarity : E \rightarrow R$

$similarity$ はエッジの重みを求める関数である。

類似度は要素間のすべての組み合わせについて計算されるので、すべてのノードの間にエッジが存在する。よって、類似度を表すグラフは完全グラフとして表せる。

2.2 ネットワークの抽出法

2.2.1 関係性に基づくネットワークの抽出法

フィールドの関係性に基づくネットワークの多くは2部グラフを構成する。2部グラフは、第1行と第1列のセルに2種類のノードを並べ、中のセルにエッジの重みを並べた表として表すことができる。ノード間にエッジがない場合、エッジの重みは0である。本論文ではこの表を関係表と呼ぶ。

2つのフィールド f_p, f_q の関係性に基づくネットワークを関係表で表した例を図2に示す。

$$F_p = \{a_{1,p}, a_{2,p}, \dots, a_{s,p}\}, (s \leq m)$$

$$F_q = \{b_{1,q}, b_{2,q}, \dots, b_{t,q}\}, (t \leq m)$$

$$e_{i,j} = (a_{i,p}, b_{j,q})$$

$$W = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,t} \\ w_{2,1} & w_{2,2} & \dots & w_{2,t} \\ \vdots & \vdots & \ddots & \vdots \\ w_{s,1} & w_{s,2} & \dots & w_{s,t} \end{bmatrix}, w_{i,j} = \text{relation}(e_{i,j})$$

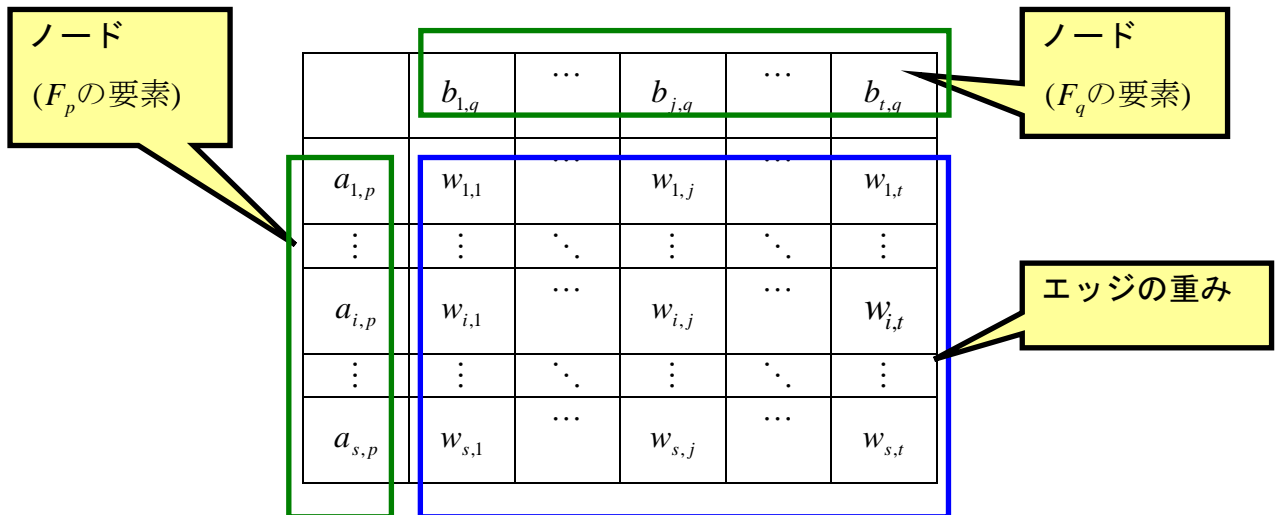


図 2 : 関係表の構成図

このとき、エッジの重みを関係の強さとすることで関係の強さを表すネットワークが得られる。エッジの重みを求める関数 *relation* には以下のものが考えられる。

同一レコードに記録された回数は関係の強さを表すパラメータと言える。同一レコードに記録された回数をエッジの重みとする場合、重みを求める関数 *relation* を以下のように定義する。

$$relation(e_{i,j}) = |K_{i,j}| \quad \text{ただし、} K_{i,j} = \{k \mid f_{k,i} = a_{i,p} \wedge f_{k,j} = b_{j,q}\}$$

同一レコードに記録された数値フィールドの要素を重みに付加することができる。さらに、付加するときの計算方法には合計、平均などが考えられる。データベースの数値フィールド f_r の要素を重みとする場合、*relation* を以下のように定義する。計算方法は合計と平均を採用した。

$$\begin{aligned} \text{合計} \quad relation(e_{i,j}) &= \sum_{k \in K_{i,j}} f_{k,r} \\ \text{平均} \quad relation(e_{i,j}) &= \frac{\sum_{k \in K_{i,j}} f_{k,r}}{|K_{i,j}|} \end{aligned}$$

データベースから関係性のネットワークを抽出するには以下の項目を定める必要がある。

- ノードとするフィールド f_p, f_q
- エッジとするフィールド f_r
- エッジの重みの計算方法 *relation*

2.2.2 類似度に基づくネットワークの抽出法

2つのフィールド f_p, f_q の関係性を表す関係表を図 3 に示す。ここでフィールド f_p に注目すると、 F_p の各要素は「フィールド f_q の各要素との関係の強さ」という属性を持ったベクトルとして表すことができる。このベクトル間の類似度を計算することで、「フィールド f_p の各要素のフィールド f_q についての類似度」がわかる。このときフィールド f_q は「類似度計算の対象フィールド」と言える。

$$a_{i,p} = [w_{i,1}, w_{i,2}, \dots, w_{i,t}]$$

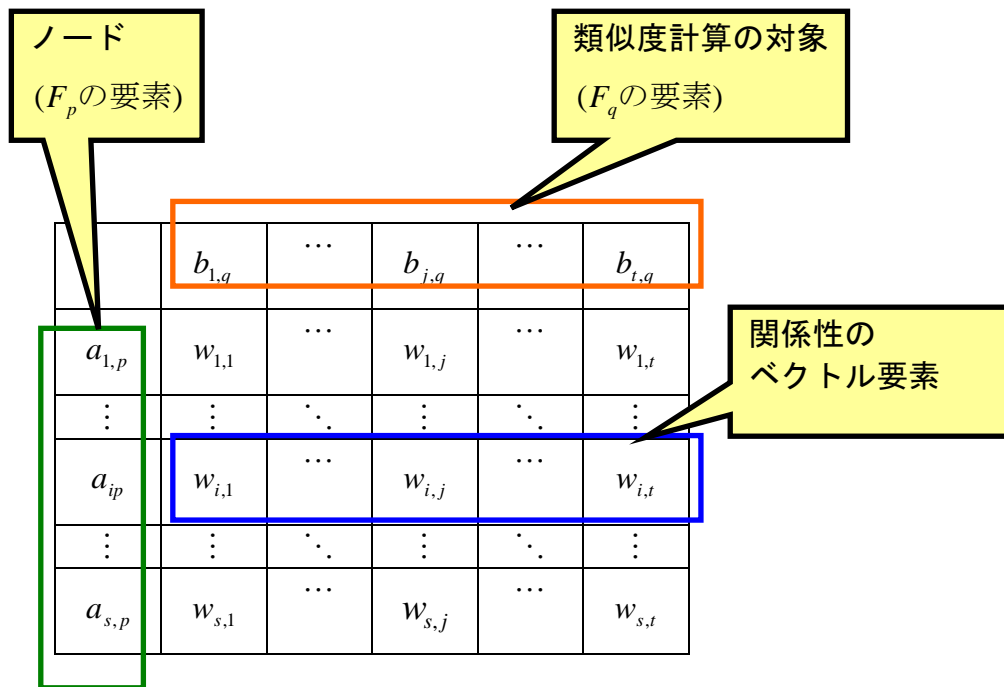


図 3 : 関係表の別の視点

複数のフィールドを類似度計算の対象フィールドに選ぶこともできる。対象フィールドとして f_q, f_v の2つを選んだ例を図4に示す。 F_p のベクトル間の類似度を計算することで、「フィールド f_p の各要素のフィールド f_q, f_v についての類似度」がわかる。

$$F_v = \{c_{1,v}, c_{2,v}, \dots, c_{u,v}\}, (u \leq m)$$

$$a_{i,p} = [w_{i,1}, w_{i,2}, \dots, w_{i,t}, w_{i,(t+1)}, w_{i,(t+2)}, \dots, w_{i,(t+u)}]$$

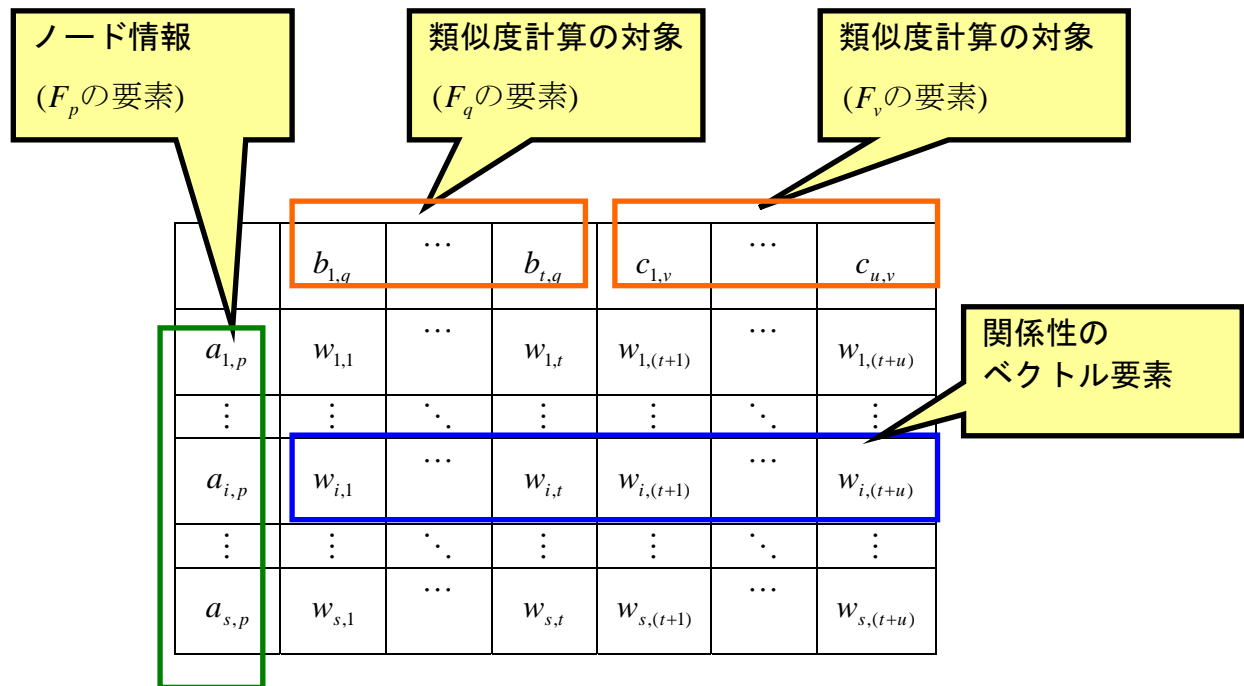


図4: フィールドを2つ配置した関係表

類似度に基づくネットワークは完全グラフを構成する。完全グラフは第1行と第1列のセルに1種類のノードを並べ、中のセルにエッジの重みを並べた表として表すことができる。本論文ではこの表を類似度表と呼ぶ。

図2の関係表を基に作成した類似度表を図5に示す。

$$F_p = \{a_{1p}, a_{2p}, \dots, a_{sp}\}$$

$$W' = \begin{bmatrix} w'_{1,1} & w'_{1,2} & \dots & w'_{1,s} \\ 0 & w'_{2,2} & \dots & w'_{2,s} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & w'_{s,s} \end{bmatrix}, w'_{ij} = \text{similarity}(a_{ip}, a_{jp})$$

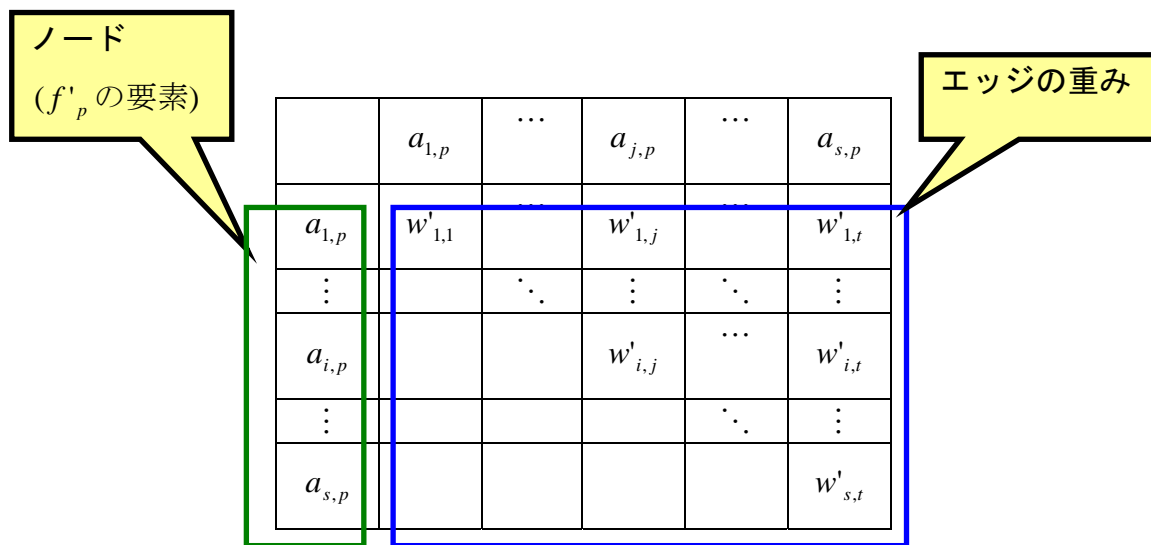


図5：類似度表

ベクトル間の類似度の計算を行う関数 *similarity* を以下のように定義する。なお、類似度の計算方法として3つの方法を採用した。

$$a_{i,p} = [w_{i,1}, w_{i,2}, \dots, w_{i,t}], a_{j,p} = [w_{j,1}, w_{j,2}, \dots, w_{j,t}], (t \leq m)$$

マンハッタン距離の逆数

$$similarity(a_{i,p}, a_{j,p}) = \frac{1}{\sum_{k=1}^t |w_{i,k} - w_{j,k}|}$$

ユークリッド距離の逆数

$$similarity(a_{i,p}, a_{j,p}) = \frac{1}{\sqrt{\sum_{k=1}^t (w_{i,k} - w_{j,k})^2}}$$

相関係数

$$similarity(a_{i,p}, a_{j,p}) = \frac{\sum_{k=1}^t (w_{i,k} - \overline{a_{i,p}})(w_{j,k} - \overline{a_{j,p}})}{\sqrt{\sum_{k=1}^t (w_{i,k} - \overline{a_{i,p}})^2} \sqrt{\sum_{k=1}^t (w_{j,k} - \overline{a_{j,p}})^2}}$$

データベースから類似度に基づくネットワークを抽出するには以下の項目を定める必要がある。

- ノードとするフィールド f_p
- 類似度の計算対象とするフィールド f_q, f_v, \dots
- 属性ベクトルにするフィールド f_r
- 属性ベクトルの計算方法 *relation*
- 類似度の計算方法 *similarity*

2.3 ネットワークの可視化手法

ネットワークの可視化手法に関する研究は数多くなされてきた[5]。現在、ネットワーク構造の視覚的な把握のための様々なレイアウト手法が開発されている[2,13]。データベースから抽出したネットワークに対して効果的なレイアウト手法を選ぶことで、ネットワークの構造をより把握しやすくなると考えられる。

現在、可視化手法に関する研究の成果物として多くの可視化ツールが実装されている[3,4,8]。しかしながら、これらのツールの入力データ形式はツール側で読み込みやすくすることを主眼に置いており、スプレッドシートなどで管理、編集するのは容易でない。

これらの問題に対する対応策としては、データベースの管理システムからネットワークデータを取得し、専用の変換プログラムに通すなどしてツールに対応したデータに変換をしているのが現状である。

2.4 ネットワークの抽出・可視化の流れ

ネットワークの抽出、可視化のためには図6のような手順を踏む。

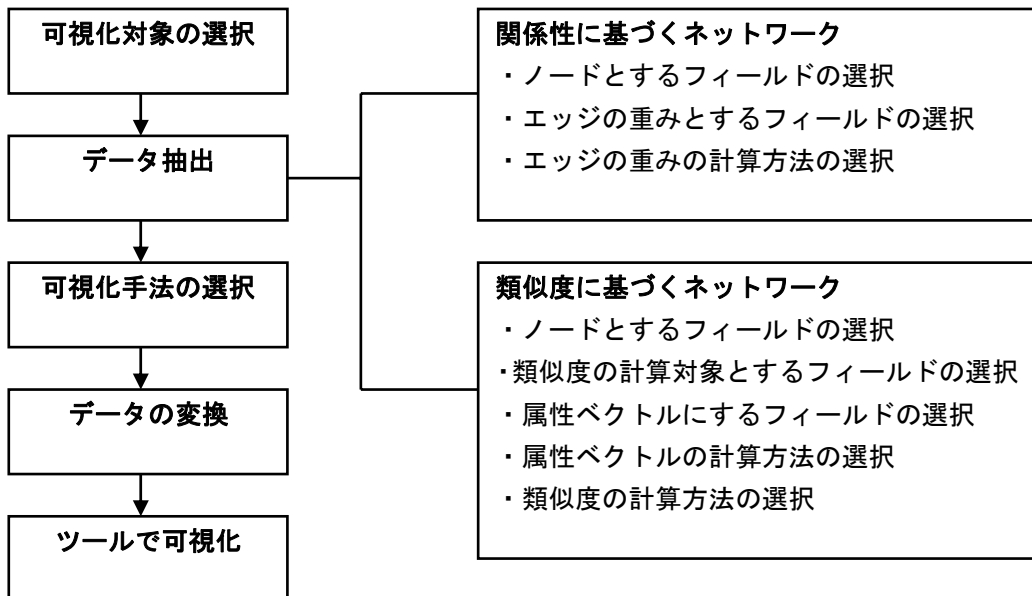


図6：ネットワークの抽出・可視化の流れ

第3章 ネットワークを可視化するための対話型 インタフェース

本章では、前章で議論したネットワーク抽出法に適するインタフェースを考察し、ネットワークの抽出・可視化のために必要な情報を対話的に獲得するインタフェースを提案する。

3.1 ネットワーク抽出における問題点

前章で述べたように、データベースからネットワークを抽出してネットワーク図で可視化するには、「可視化対象の選択」「ノード、エッジにするフィールドの選択」「エッジの重みや類似度の計算方法の指定」「可視化手法の選択」など、多くの項目を入力・設定しなければならない。また、抽出したネットワークや抽出条件によって可視化に有効な表現が異なる場合もある。さらに、抽出したネットワークをツールで可視化するには、データをツールに対応した形式に変換する必要がある。しかしながら、これらの作業を利用者がすべて行うのは煩雑な作業である。また、これらの作業はデータ抽出に関する知識やデータ変換の技術が必要であり、これらの知識・技術のない者が行うのは難しい。本研究で開発するシステムのインタフェースは、これらの作業を簡単に行えるようにサポートするものであることが望まれる。

3.2 ウィザード

複雑な操作をサポートするインタフェースとして「ウィザード」が挙げられる。ウィザードは、段階的な対話を通じてシステムの複雑な操作を行うインタフェースである。ウィザードでは、利用者はシステムからの質問に答える形で操作を行う。

ウィザードを用いる利点は2つ挙げられる。まず、操作のための知識・技術が必要とされないことである。操作に必要な情報はウィザードが質問をするので、利用者は操作に必要な知識がなくても質問に答えることで操作を行える。また、最低限の入力・選択で操作が完了することも挙げられる。ウィザードは段階的に質問をするので操作に必要な情報を絞り込むことができ、利用者は最低限の質問に答えるだけで操作を行える。

ウィザードの利用例として、アプリケーションのセットアップウィザードが挙げられる。アプリケーションのセットアップではファイルの保存場所の決定や他システムとの連携のために様々な設定が必要であるが、専門知識のない利用者がそれらをすべて行うのは難しい。ウィザードは必要な情報を得るために利用者に質問をするので、利用者はセットアップに必要な情報が何であるかを知らなくても、システムが提示する質問に答えていけばセットアップを行うことができる。

ウィザードはデータの抽出や可視化にも用いられている。スプレッドシートでは、座標系のグラフを描画するためのインタフェースとして「グラフウィザード」が実装されており、現在広く使われている。座標系グラフの描画にも、グラフの種類や可視化対象の抽出など多くの項目を入力・設定する必要があるが、システムが段階的に利用者に質問をすることでデータの抽出・可視化のために必要な情報を絞り込み、利用者に負担を強いることなくグラフを描画することを実現している。

3.3 ネットワーク可視化のためのウィザード

ネットワーク図の描画に関しても座標系グラフと同様の問題が当てはまる。座標系グラフの描画を支援するウィザードが広く利用されていることから、ネットワーク図の描画に関しても同様の解決手法が適用できると考えられる。

本研究では、データベースからネットワークを抽出するためのインタフェースとしてウィザードを提案する。このインタフェースにより、利用者はスプレッドシート上で起動したウィザードからの質問に答える形で情報を入力・選択することで、ネットワークを抽出して可視化ツールに対応したデータ形式に変換できるようになる。

第4章 ウィザードの設計と実装

本章では、ネットワーク可視化のためのウィザードの設計と実装について述べる。

4.1 ウィザードの設計

4.1.1 ウィザードが獲得すべき情報

データベースからネットワークを可視化するために必要な情報は「データの抽出に関する項目」「データの可視化に関する項目」の2つに分けられる。

データの抽出に関する項目

- ・可視化対象
- ・ノードとなるフィールド
- ・エッジとなるフィールド
- ・エッジの重みや類似度の計算方法

データの可視化に関する項目

- ・可視化手法

ウィザードはこれらの情報を項目別に提示し、利用者に各項目を入力・選択させることで可視化に必要な情報を獲得する。

4.1.2 ウィザードのインターフェース

ウィザードが操作に必要な項目の提示を行うインターフェースとして入力フォームを用いる。項目の入力・選択はフォーム内のボックスやボタンで行う（図7）。

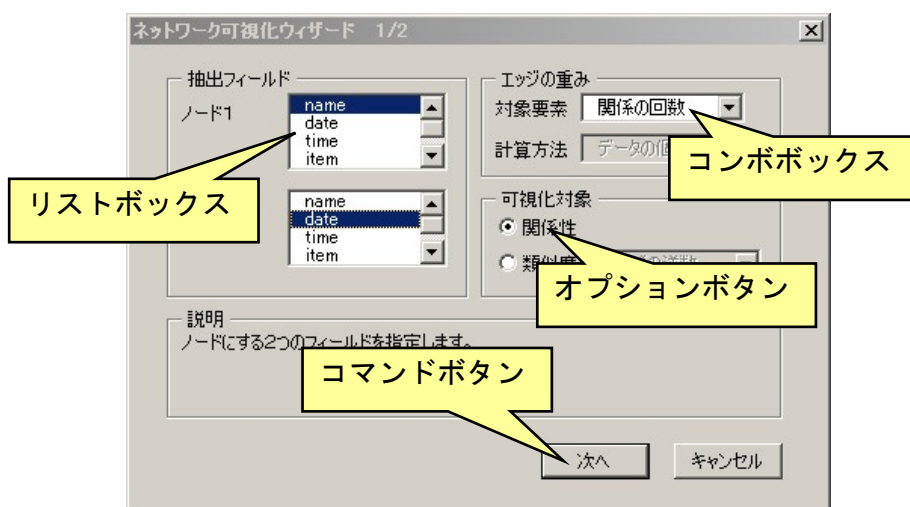


図 7：入力・選択のためのインターフェース

4.1.3 情報を提示する順序

ウィザードでは、操作に必要な項目の提示を段階的に行うことで操作に必要な情報を絞り込むことができる。利用者に余計な項目の設定を強いないために、ウィザードを「データ抽出を行うフォーム」と「可視化手法を選ぶフォーム」に分ける。

2つの入力フォームをどのような順番で提示するかであるが、以下の2つの方法が考えられる。

方法1：可視化手法の選択 → データの抽出

方法2：データの抽出 → 可視化手法の選択

方法1では、はじめに可視化手法を選び、選んだ手法で可視化できるネットワークを抽出する。一方、方法2では、はじめにデータを抽出し、抽出したデータを可視化できる手法を選ぶ。

先に紹介した Excel のグラフウィザードは方法1で実装されている。まずグラフ(図)を選び、選んだグラフを描画するために必要なデータを抽出する。これは、「数量の比較をするには棒グラフが有効である」「割合を示すには円グラフが有効である」など、可視化するデータに対して有効な図表現がわかっている場合には有効な方法である。しかしながら、関係性や類似度のネットワークを可視化する場合、抽出したネットワークに対して有効な可視化手法が何であるかを知らない場合も多いと考えられる。そこで、データを先に抽出し、抽出したデータから構成されるネットワークの種類や構造などから有効な可視化手法を提示する方式(方法2)をとることとした。

4.1.4 データ変換

抽出したネットワークを可視化するには、可視化ツールに対応したデータに変換する必要がある。本研究では、出力するデータの形式として GraphML と CSV を選んだ。

現在、XML の拡張形式としてグラフを表現することに特化した GraphML が開発されている[10][11]。GraphML は多くの可視化ツールの入力データ形式として使われるようになっており、データの適用性が高い。そこで本システムでは、GraphML を入力形式とするツールで可視化を行えるように、抽出したネットワークを GraphML ファイルとして出力する。

GraphML によるグラフの記述例を図 8 に示す。node 要素はノードを定義する。要素の id 属性はノードの ID を記録し、後にエッジを定義する際に使用する。edge 要素はエッジを定義する。source 属性と target 属性にそれぞれノードの ID を記録し、ノード間をつなぐ。ノード、エッジの子要素となる data key 要素は事前に定義しておき、子要素を追加することでノードやエッジ属性を付加する。図 4-8 の例では、ノードにラベルと色の属性、エッジに重みと色の属性を付加している。

```

<?xml version="1.0" encoding="UTF-8"?>
<graphml xmlns="http://graphml.graphdrawing.org/xmlns">
<graph edgedefault="undirected">

<key id="label" for="node" attr.name="label" attr.type="string"/>
<key id="n_color" for="node" attr.name="n_color" attr.type="string"/>
. . . . .
<key id="weight" for="edge" attr.name="weight" attr.type="double"/>
<key id="e_color" for="edge" attr.name="e_color" attr.type="string"/>
. . . . .

<node id="1">
  <data key="label">node A</data>
  <data key="n_color">red</data>
  . . . . .
</node>
<node id="2">
  <data key="label">node B</data>
  <data key="n_color">blue</data>
  . . . . .
</node>
. . . . .

<edge id="1" source="1" target="2">
  <data key="weight">3</data>
  <data key="e_color">black</data>
  . . . . .
</edge>
<edge id="2" source="1" target="3">
  <data key="weight">2</data>
  <data key="e_color">black</data>
  . . . . .
</edge>
. . . . .
</graph>

```

図 8 : GraphML によるグラフの記述例

また、抽出したネットワークを CSV ファイルとして出力できるようにした。CSV は要素をカンマで区切って並べる形のデータである。構造が単純で解析が容易であるため、Web アプリケーションなどの入力データ方式として利用されている。

CSV によるグラフの記述例を図 9 に示す。最も左の要素がノードやエッジであることを表す要素である。ノードである場合、ノードを表す要素、ノードの ID、ノードのラベル、ノードの色、. . . というようにノードの属性をカンマで区切って並べている。エッジである場合、エッジを表す要素、エッジの ID、ソースノードの ID、ターゲットノードの ID、エッジの重み、エッジの色、エッジの太さ、. . . というようにエッジの属性を並べている。

```
//(Node),(Node ID),(Node Label),(Node Color),...
"NODE",1,NodeA,red,...
"NODE",2,NodeB,blue,...
"NODE",3,NodeC,green,...
. . . . .

//(Edge),(Edge ID),(Source),(Target),(Edge Weight),(Edge Color),(Edge Width),...
"EDGE",1,1,2,3,black,4,...
"EDGE",2,2,5,2,black,4,...
. . . . .
```

図 9 : CSV によるグラフの記述例

4.2 可視化手法

ウィザードで選択できる可視化手法として以下の 3 つを選んだ。

- 1. **アンカーマップレイアウト[8]**
 2 部グラフの片方のノードの位置に制約を課したレイアウト手法。一方のノードを円状に固定して配置（アンカーノード）し、もう一方のノードをばねモデルによって配置（フリーノード）することで、関係性の強いノード同士が近くに配置される。
- 2. **アンカーマップの差分表現[12]**
 2 つの 2 部グラフの差分を算出し、1 つのアンカーマップレイアウトの中でグラフ構造の変化を視覚的に表すレイアウト手法。フリーノードの移動をアイコンで表示する。
- 3. **Animated Radial Layout[13]**
 グラフの 1 つのノードを中心に配置し、中心ノードにリンクしたノードを円状に配置するレイアウト手法。ノードをクリックすることで中心ノードを入れ替えることができる。

4.3 ウィザードの実装

本研究では、スプレッドシートとネットワーク可視化ツールを連携する対話型インタフェースをMicrosoft Excelのマクロ機能及びVBA(Visual Basic for Applications)を用いて実装した。

4.3.1 システムの構成

本システムは大きく分けて、「入力フォーム部」「データ格納部」「計算部」「データ変換・出力部」からなる(図10)。

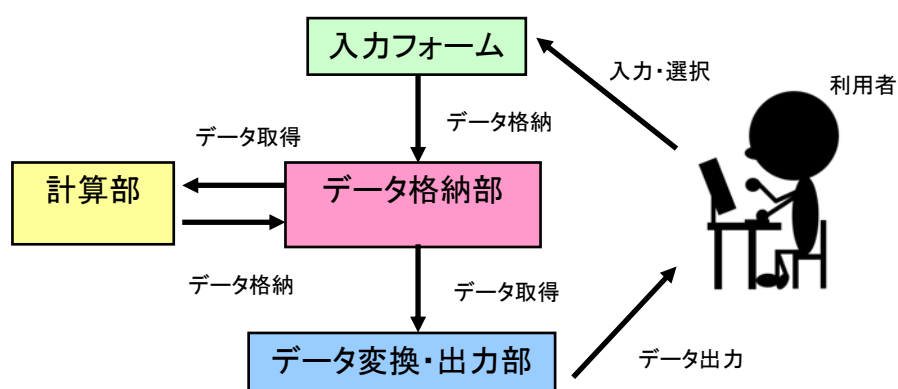


図 10 : システム構成図

入力フォーム部は利用者からの入力を受け、入力・選択された値をデータ格納部に受け渡す部分である。

データ格納部はフォームで入力・選択された値や計算部での計算結果を保持する部分である。

計算部はデータ格納部で保持されている入力情報を基に関係性、類似度を計算し、計算結果をデータ格納部に受け渡す部分である。

データ変換・出力部は、データ格納部で保持されている選択情報と関係性、類似度の計算結果を基にツールに対応したファイルを出力する部分である。

4.3.2 入力フォームの実装

入力フォームは「データ抽出フォーム」「可視化手法選択フォーム」の2つに分かれる。

データ抽出フォーム

データ抽出フォームのスクリーンショットを図 11 に示す。

図 11 : データ抽出フォーム

画面下の「説明」のフレームの中には、マウскарソルを合わせたフレームに関する簡単な説明が表示される。

画面右の「可視化対象」のフレーム内にあるオプションボタンで可視化対象を選択する。可視化対象は「関係性」と「類似度」の2種類があり、類似度を選択した場合に類似度の計算方法を指定するコンボボックスが選択可能になる(図 12)。計算方法にはマンハッタン距離の逆数、ユークリッド距離の逆数、相関係数を用いた。

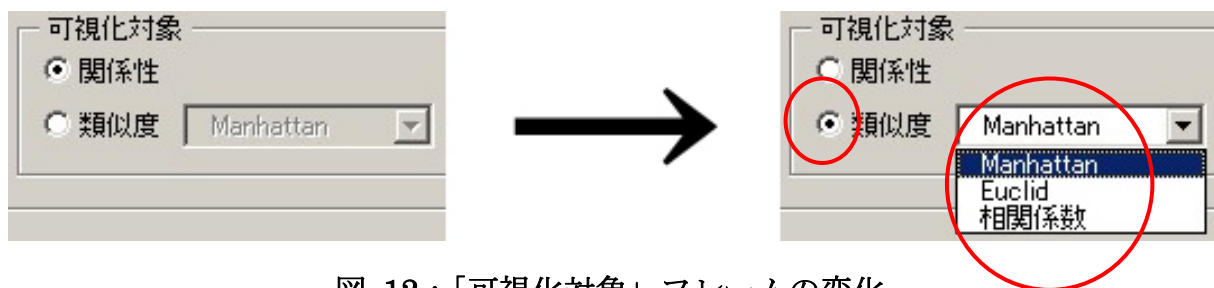


図 12 : 「可視化対象」フレームの変化

画面左の「抽出フィールド」のフレーム内の2つのリストボックスでノードにするフィールドを選択する。リストボックスの要素はスプレッドシート上のデータベースの各フィールドと対応している。可視化対象が「関係性」に設定されている場合、2つのリストボックスはノードとなる2種類のフィールドを選択するものであるが、可視化対象が「類似度」に設定されている場合、2つのリストボックスはそれぞれ「ノードとなるフィールド」「類似度の計算対象フィールド」を選択するものになる（図13）。

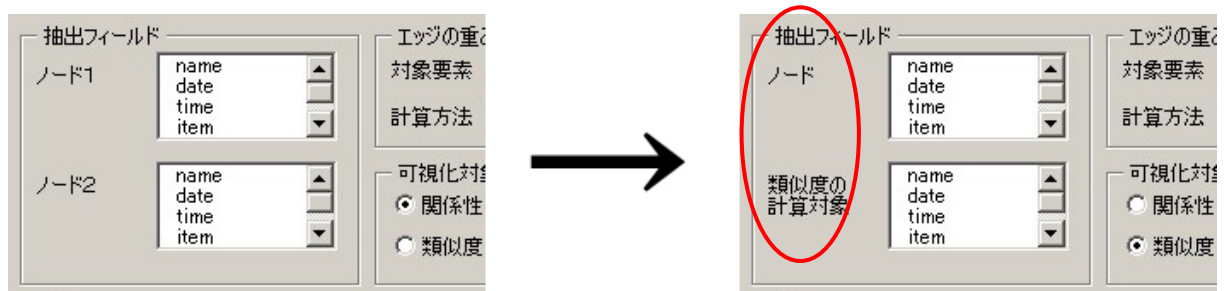


図 13 : 「抽出フィールド」フレームの変化

画面右上の「エッジ」のフレーム内にある2つのコンボボックスで、エッジの重みの対象フィールドと計算方法を選択する。重みの対象フィールドは、スプレッドシート上のデータベースの数値フィールドと対応している。重みの対象のデフォルト設定は「関係が生じた回数」であるが、重みの対象フィールドを指定した場合は重みの計算方法を指定する（図14）。計算方法は合計、平均を選んだ。



図 14 : 「エッジの重み」フレームの変化

画面下の「次へ」ボタンを押すことで、入力情報を基にした表を作成し、可視化手法選択画面へと移行する。「キャンセル」ボタンでウィザードを終了させることもできる。

可視化手法選択フォーム

可視化手法選択フォームのスクリーンショットを図 15 に示す。

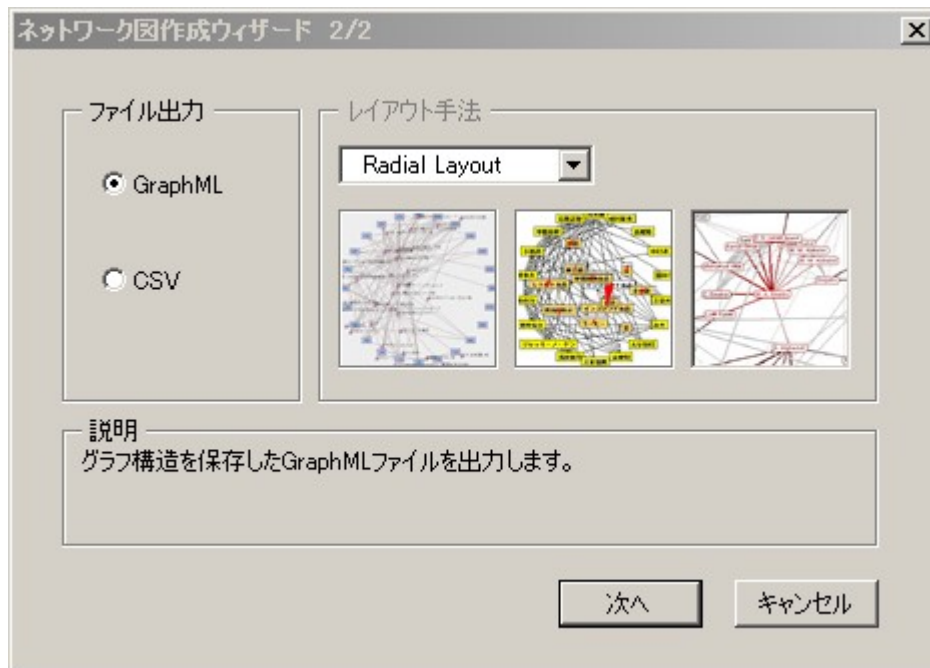


図 15 : 可視化手法選択フォーム

データ抽出部分と同じく、画面中央の「説明」のフレームの中にはマウスカーソルを合わせたフレームの説明が表示される。

画面左の「ファイル出力」のフォームの中にあるオプションボタンで出力ファイルを選択する。「CSV」を選んだ場合、出力ファイルは CSV 形式となる。「GraphML」を選んだ場合、画面左の「レイアウト手法」の中のコンボボックス及びイメージ図で可視化手法を選択する（図 16）。



図 16 : 「レイアウト手法」 フレームの変化 (1)

画面上の「レイアウト手法」のフレームの中にあるコンボボックスまたは可視化のイメージ図で可視化手法を選択する。コンボボックスの要素とイメージ図は対応しており、コンボボックスで選択した要素に対応するイメージ図が凹んで表示される。逆にイメージ図をクリックすると、コンボボックスの値が選択したイメージ図に対応したものに變更される（図 17）。

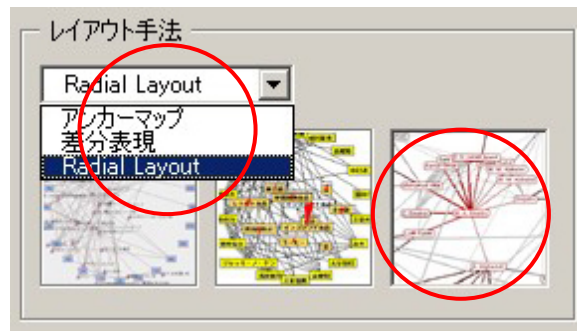


図 17: 「レイアウト手法」フレームの變化（2）

画面下の「次へ」ボタンを押すことで、フォームで選んだ形式に沿ったファイルを出力する。「CSV」を選んだ場合、ノードとエッジに属性を付加するためのフォームに移行する。「GraphML」を選んだ場合、抽出したネットワークや選択した可視化ツールの情報を基にツールに対応した GraphML ファイルを出力する。

属性付加フォーム

CSV ファイルに記録するグラフのノードとエッジに属性を付加するためのフォームのスクリーンショットを図 18 に示す。

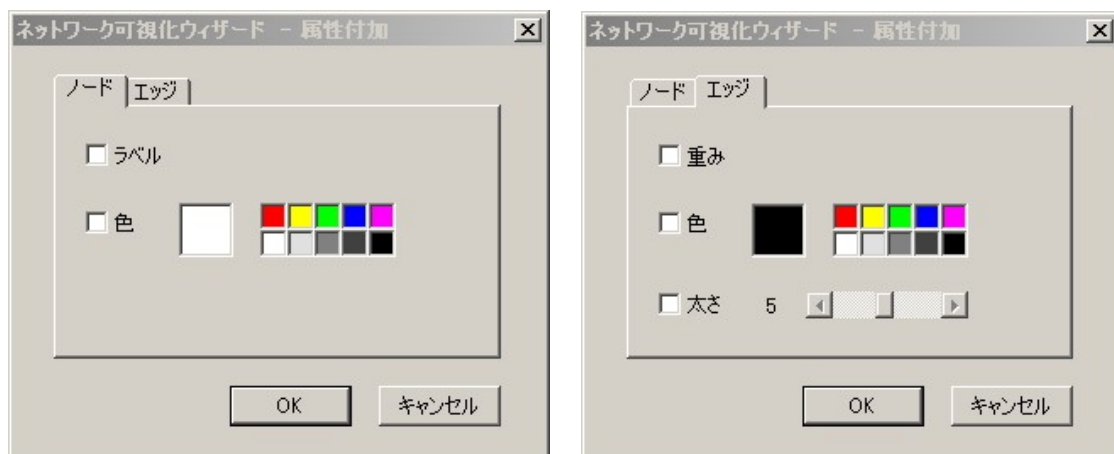


図 18: 属性付加フォーム

ノードとエッジのそれぞれのタブの中にあるチェックボックスをチェックすることで属性を付加する。ノードの属性はラベルと色、エッジの属性は重み、色、太さとした。色や重みはカラーパレットやスクロールバーで設定する。

4.4 使用例

使用例として、購買履歴のデータベースからネットワークを可視化した例を示す。購買履歴のデータベースがスプレッドシートで管理されている（図 19）。フィールド数は 5（「購入者」「日付」「時間」「商品カテゴリー」「値段」）、レコード数は 2096 である。このデータベースから「購入者と商品カテゴリーの関係」を抽出し、RadialLayout 手法で可視化した。

	A	B	C	D	E
1	name	date	time	category	price
2	Y.S	2006/6/2	22:32:24	Instant	100
3	SHUS	2006/6/5	8:21:30	Juice	100
4	H.M	2006/6/5	11:21:57	Instant	100
5	SHIS	2006/6/5	12:36:17	Instant	100
6	N.N	2006/6/5	12:52:02	Instant	100
7	K.M	2006/6/5	13:39:50	Instant	100
8	SHIS	2006/6/5	14:35:13	Tea	100
9	J.Y	2006/6/5	14:42:52	Snack	100
10	SHUS	2006/6/5	15:09:38	Coffee	100
11	H.M	2006/6/5	16:49:36	Tea	100
12	H.M	2006/6/5	16:50:14	Juice	100
13	SHUS	2006/6/5	17:59:05	Juice	100
14	Y.T	2006/6/5	19:39:50	Tea	100
15	S.K	2006/6/5	20:31:20	Tea	100
16	N.N	2006/6/5	20:37:35	Juice	100
17	M.I	2006/6/5	20:42:47	Juice	100
18	M.M	2006/6/6	10:01:10	T...	100

図 19：スプレッドシートで管理された購買履歴のデータベース

スプレッドシート上に表示されたツールバーでウィザードを起動する（図 20）。

stant	100
ea	100
nack	100
offee	100
ea	100
uice	100
uice	100

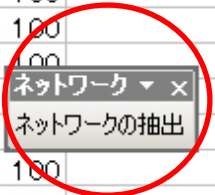


図 20：「ネットワーク抽出」ツールバー

はじめにデータ抽出フォームが表示される。ここで各項目を以下のように設定する（図 21）。

- ・可視化対象 : 「関係性」
 - ・ノード1 : 「name」
 - ・ノード2 : 「category」
 - ・エッジの重み : 「関係（ここでは購入）の回数」
- ※ 「類似度の計算方法」と「エッジの計算方法」は設定しない

図 21 : データ抽出フォームでの選択

「次へ」ボタンを押すと「可視化手法選択フォーム」へ移行する。

次に可視化手法選択フォームが表示される。ここで各項目を以下のように設定する（図 22）。

- ・ファイル出力：GraphML
- ・可視化手法：RadialLayout



図 22：可視化手法選択フォームでの選択

「OK」ボタンを押すとメッセージボックスが表示されて GraphML ファイルが出力される（図 23）。



図 23：メッセージボックス

出力した GraphML ファイルをツールで可視化した (図 24)。

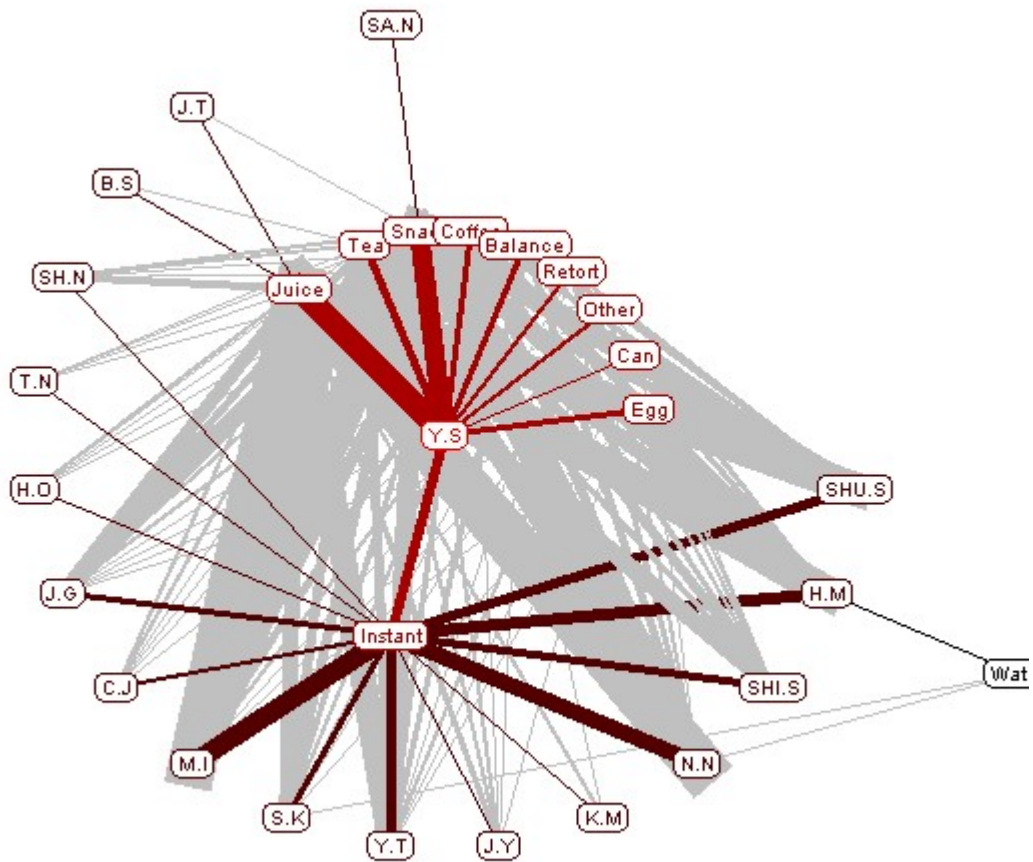


図 24 : 「購入者と商品カテゴリーの関係性」の可視化結果

このレイアウト手法では関係性の強さがエッジの太さに割り当てられている。この視覚表現により、「誰が何を多く買っているか」という購買傾向が視覚的に把握できる。

第5章 評価と考察

本研究では、開発したネットワーク可視化のための対話型インタフェースについて評価と考察を行った。

5.1 評価

商品購買データベースを使って、2つの方法で関係性と類似度のネットワークを抽出、ツールで可視化し、作業ステップの比較を行った（表1，表2）。

方法1：Excelの既存の機能などを使って手動で行う方法

方法2：開発したウィザードを使う方法

	方法1	方法2
関係性	0. CSV ファイルを GraphML ファイルに変換するプログラムを事前に準備	
	1. Excel の「ピボットテーブル」ウィザードを使ってピボットテーブル（関係表）のテンプレートを作成 1-a.作成するピボットテーブルの種類を選択 1-b.データ範囲を選択 1-c.ピボットテーブルを作成するワークシートの選択	1. 開発したウィザードを使って GraphML ファイルを出力 1-a.可視化対象、ノード、エッジ、エッジの計算方法を選択 1-b.可視化手法を選択
	2. テーブルのテンプレートにデータを入力	
	3. Excel の保存機能を使って CSV ファイルを出力	
	4. CSV ファイルを変換プログラムに通して GraphML ファイルを出力	
	5. GraphML ファイルをツールで可視化	2. GraphML ファイルをツールで可視化

表 1：作業ステップの比較（関係性）

	方法 1	方法 2
ファイルの類似度	0. CSV ファイルを GraphML ファイルに変換するプログラムを事前に準備	
	1. Excel の「データ分析」ツールを使って類似度表を作成 2. Excel の保存機能を使って CSV ファイルを出力 3. CSV ファイルを変換プログラムに通じて GraphML ファイルを出力	1. 開発したウィザードを使って GraphML ファイルを出力 1-a. 可視化対象、ノード、エッジ、エッジの計算方法を選択 1-b. 可視化手法を選択
	4. GraphML ファイルをツールで可視化	2. GraphML ファイルをツールで可視化

表 2：作業ステップの比較（類似度）

5.2 考察

方法 1 では、関係性、類似度のどちらを可視化する場合にも、可視化ツールに対応した GraphML データを出力するのに CSV 形式の仲介ファイルを出力しなければならない。また、CSV ファイルから GraphML ファイルを出力するための変換プログラムを用意する必要があり、それが無い場合は手作業で GraphML ファイルへ変換しなければならない。これはデータ形式やプログラミングに関する知識がないと難しい作業である。その一方で、方法 2 ではそうした面倒な作業を一切行うことなく、ウィザードの中で可視化ツールを選ぶことでツールに対応した GraphML ファイルが出力できた。作業ステップの現象と操作の簡略化という面から、ネットワークの可視化を支援するという目的を果たすものだと言えるだろう。

5.3 今後の課題

現在のウィザードは可視化ツールに対応した GraphML ファイルを出力するが、可視化ツールは手動で起動する必要がある。今後はファイルの出力からツールの起動までの操作を自動化し、出力した GraphML ファイルを引数に自動で可視化ツールを起動できるようにしたいと考えている。また、評価実験などを行って実際にウィザードを使ってもらい、使った感想などを基に機能性、使いやすさの向上を図りたいと考えている。

第6章 関連研究

N.Henry らは、表形式とネットワーク図を連携することでネットワークの可視化を支援するシステム **MatrixExplorer** を開発した[4]。このシステムは、表形式で現されたソーシャルネットワークにおける人物間のリンク関係をネットワーク図でも同時に表し、表と図の対応などを視覚的に表しながらネットワーク構造の把握を支援するものである。この研究は表とネットワーク図を連携してネットワークの把握を支援するという点で関連が深い。本研究では表とネットワーク図の連携にウィザードを用いるという新しいアプローチを試みた。

J.Mackinlay らは、関係情報に対して有効な図表現をデザインするツール **APT(A Presentation Tool)**を開発している[7]。このツールは、関係情報の特徴にあわせて棒グラフや散布図、ネットワーク図といった図表現を自動的にデザインするものである。この研究は、関係情報の解釈や図表現の提示をシステムがサポートするという意味で関連が深い。本研究ではデータ管理を行うツールとして普及しているスプレッドシートを用いたことでネットワーク図による可視化を身近なものにした。

ネットワークの可視化に関する研究には様々なアプローチがなされた[5]。杉山は、グラフの自動描画のためのレイアウト手法およびアルゴリズムについて体系的にまとめている[1]。ネットワークの可視化ツールが数多く開発されていることは第1章でも述べた。また近年では、**Web** に存在するネットワーク構造の可視化に関する研究が行われている。J.Heer らは、オンラインソーシャルネットワーキングサービスにおける人物のリンク関係を可視化して知人関係の把握やコミュニティの発見を支援するシステム **Vizster** を開発した[3]。石原は、動的ネットワークの可視化手法として波紋表現を開発し、ウェブログのリンク関係の広がりを可視化することに応用した[12]。

第7章 結論

本研究ではまず、データベースに内在するネットワークの種類や抽出法を考案し、ネットワークの抽出に必要な情報や手順を整理した。

データベースに保存される関係データから抽出できるネットワークとして「関係性に基づくネットワーク」と「類似度に基づくネットワーク」に着目した。関係性のネットワークは多くの場合2部グラフとして表現することができ、データベースから関係性のネットワークを抽出するには、ノードとなる2種類のフィールドとエッジの重みとなるフィールド及びその計算方法を定める必要がある。類似度のネットワークを抽出するには、ノードとなるフィールドと類似度の計算対象となるフィールドとエッジの重みとなるフィールド及びその計算方法、類似度の計算方法を定める必要がある。

データベースからネットワークを抽出するのに必要な情報を取得するのに有効な手法としてウィザードで対話的に情報獲得を行う手法を提案した。この提案を基に、スプレッドシート上のデータベースに内在するネットワークを抽出し、ネットワーク可視化ツールに対応したデータに変換するウィザードを **Microsoft Excel** のマクロ機能及び **VBA** を用いて実装した。このウィザードを使うことによって、スプレッドシートから簡単にネットワーク図を描画できるようになり、ネットワーク図によるデータ分析、可視化がより身近になると考えられる。

今後の課題としては、ネットワーク可視化ツールを自動起動する機能を整備し、より少ない操作で簡単にネットワーク図を描画できるように改良を加える予定である。

謝辞

本論文を執筆するにあたり、指導教員である田中二郎先生、三末和男先生をはじめ、志築文太郎先生、高橋伸先生には研究方法から研究内容、論文執筆に至るまで丁寧なご指導と適切なお助言をいただきました。心より感謝致します。

また、IPLAB(田中研究室)の皆様にはゼミなどを通じて貴重なご意見を数多くいただきました。特に、石原正樹さん、宮崎裕久さんには研究を進める上で必要なツールを提供していただきました。また、NAIS チームの皆様にはチームミーティングでの意見のみならず研究生活全体にわたって多くのご意見とご指摘をいただきました。ここに深く感謝致します。

何よりも家族の支えなくしては、このように有意義な研究生活は送れませんでした。物心両面にわたって私を支えてくれた両親、暖かい励ましをくれた兄弟に心より感謝の意を申し上げます。

最後に、様々な面で力になってくれた多くの友人、大学生活でお世話になったすべての方々に心から感謝致します。本当にありがとうございました。

参考文献

- [1] 杉山公造. グラフ自動描画法とその応用 —ビジュアルヒューマンインタフェース—. 計測自動制御学会,1993
- [2]J.Heer, S.K.Card and J.A.Landay. Prefuse: a toolkit for Interactive Information Visualization. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems 2005*, pp. 421-430, 2005
- [3]J.Heer and D.Boyd. Vizster: Visualizing Online Social Networks. *Proceedings of the 2005 IEEE Symposium on Information Visualization(INFOVIS '05)*, pp. 33-40
- [4]N.Henry and J.D.Felete. MatrixExplorer: a Dual-Representation System to Explore Social Networks. *IEEE Transactions on Visualization and Computer Graphics, Vol.12, No.5, September/October 2006*. pp. 677-684
- [5] From Visual Data Exploration to Visual Data Mining: A Survey. *IEEE Transactions on Visualization and Computer Graphics*, pp. 378-394
- [6]Maria Cristina Ferreira de Olivia and Haim Levkowitz. From Visual Data Exploration to Visual Data Mining: A Survey. *IEEE Transactions on Visualization and Computer Graphics*, pp. 378-394, July 2003.
- [7]J.Mackinlay. Automating the Design of Graphical Presentations of Relational Information. *ACM Transactions on Graphics, Vol.5, No.2, April 1986*, pp. 110-141
- [8]Kazuo Misue. Drawing Bipartite Graphs as Anchored Maps. *Proceedings of Asia-Pacific Symposium on Information Visualization (APVIS2006)*, pp. 169-177 (February 1-3, 2006)
- [9]M.Ghoniem,J.D.Felete and P.Castagliola. A Comparison of the Readability of Graphs Using Node-Link and Matrix-Based Representations. *Proceedings of the IEEE Symposium on Information Visualization (INFOVIS'04)*, pp. 17-24
- [10]U.Brandes and Pich. GraphML Transformation. *Proceedings of Symposium on Graph Drawing 2004*. pp. 89-99
- [11]U.Brandes, M.Eiglsperger, I.Herman, M.Himsolt, and M.Scott. GraphML Progress Report. *Proceedings of Symposium on Graph Drawing 2001* pp. 501-512
- [12]石原正樹. 動的ネットワークの成長過程と差分の可視化手法. 筑波大学大学院博士課程 2006年度システム情報工学研究科修士論文
- [13]Yee, D.Fisher, R.Dhamija, and M.A.Hearsi. Animated Exploration of Dynamic Graphs with Radial Layout. *Proceedings of the IEEE Symposium on Information Visualization (INFOVIS'01)*, pp. 43-50