大規模クラスタリング結果に対する探検型可視化手法のサーバクライアントモデルでの実装

平山 健太† 井上 悦子‡ 中川 優‡

†和歌山大学 大学院システム工学研究科 〒640-8510 和歌山県和歌山市栄谷 930
‡和歌山大学 システム工学部 〒640-8510 和歌山県和歌山市栄谷 930
E-mail: †s111043@sys.wakayama-u.ac.jp, ‡{etsuko, nakagawa}@sys.wakayama-u.ac.jp

あらまし 本研究では，大規模クラスタリング結果の任意の部分領域をグラフで表現する探検型可視化手法を，複数ユーザから利用できるようサーバクライアントモデルで実装した．クライアントでは部分領域を閲覧するグラフ表示と領域操作を行い，サーバとの通信により表示を変化させる要素を取得し，グラフを連続的に変化させる．複数回の領域操作に伴うグラフの変化要素を事前取得することに加え，グラフの変化要素のうちクライアント側で生成可能な分は，サーバに問い合わせることなくクライアント側で生成することで通信回数を低減し，クライアントでの遅延のないグラフ閲覧を可能とした．また，本モデルで実装した試作システムにおいて，クライアントのメモリの使用量を計測し，クライアントの負荷の検討を行った．

キーワード クラスタリング， 探検型可視化手法，サーバクライアントモデル

A Server-Client Implementation of Interactive Graph-Based Visualization Method for Large Hierarchical Data

Kenta HIRAYAMA† Etsuko INOUE‡ and Masaru NAKAGAWA‡

†Graduate School of Systems Engineering, Wakayama University
930 Sakaedani, Wakayama-shi, Wakayama, 640-8510 Japan
‡Faculty of Systems Engineering, Wakayama University
930 Sakaedani, Wakayama-shi, Wakayama, 640-8510 Japan
E-mail: †s111043@sys.wakayama-u.ac.jp, ‡{etsuko, nakagawa}@sys.wakayama-u.ac.jp

1. はじめに

商品販売サイトでは，Amazon や楽天市場などで見られるように，商品の購入履歴や閲覧履歴を利用して利用者間の行動の類似や商品間の類似あるいは関連度を決定し，商品の推薦を行っている．また，Pandora, Last.fm のようなインターネットラジオでは，利用者の楽曲に対する評価などに基づいて各利用者の嗜好に合った楽曲の推薦を行っている．

このような情報推薦は，利用者の選んだアイテムに対し，関連度の高いアイテムを提示することで，利用者の嗜好に合致した未知のアイテムとの出会いの演出や，そのアイテムを購入してもらうことが期待できる．これは，アイテム自体の持つ情報の類似性や，多数の利用者の行動履歴やその嗜好の類似に着目した協調フィルタリングなどの技術を用いて実現される[1][2].

しかしながら，特定のアイテムに注目した場合に，関連度が極めて高い少数のアイテムのみを提示するのが一般的であるため，限られたアイテムとしか利用者は出会うことができない．関連度がある程度低いものも含めて，どのようなアイテムが存在するかの大まかな情報だけでも同時に俯瞰できるようにすれば，より多くの出会いを提供できるのではないだろうか．

そこで，我々はこのような大規模なアイテム集合を，注目するアイテムに関連の強いアイテム集合を詳細に，同時に関連の弱いアイテム集合についても大まかにどういったものがあるのかの概要を提示することで，利用者が興味を持ったアイテムを中心に，表示粒度を変更しながら他にどのようなアイテムが同じ集合に属するのか，他にどのような集合が存在するのかを自由に眺めながらアイテムを探索できるようなシステムの実現を目指す．

注目要素とそのごく周辺の要素を詳細に，それ以外の要素を概要のみに集約して提示する手法として，大規模なクラスタリング結果の部分領域をグラフとして

可視化する探検型可視化手法[1]がある．先ほど述べたようにアイテム間に関連度や類似度が算出可能なアイテム集合の場合，それらに基づく階層化クラスタリングが可能である．そこで，探検型可視化手法を用いてアイテム探索システムの実現を目指す．

膨大なアイテムとそれらの間の関連情報をサーバで管理し，利用者はクライアントマシンから自分の気になるアイテムを中心としてアイテム閲覧ができるようにできる．これまでに，性能評価のために簡易なビューアの試作はなされているが，いずれもスタンドアロンでの実装のみである．

本稿では，探検型可視化手法をどのようにサーバクライアントモデルで実現すべきかを検討し，試作システムを実装して性能評価を行った結果を報告する．サーバクライアントモデルで問題となり得るのが，クライアント側の負荷（特にメモリ使用量）と通信による遅延の影響であると考え，これらを評価するための実験を行った．

本論文の構成は以下のとおりである．第 2 章では，本研究を理解するために必要な予備知識として，既存研究である探検型可視化手法について紹介する．第 3 章では，探検型可視化手法をサーバクライアントモデルで実装するにあたって考慮点した点を説明する．第 4 章では，試作システムの構成を述べる．第 5 章では，評価実験とその結果に対する考察を述べる．最後に 6 章では，本論文のまとめを行う．

2. 探検型可視化手法

2.1. 概要

探検型可視化手法は，大規模な階層的クラスタリング結果を表現した樹状図に対し，ユーザの注目要素を中心とする部分領域をグラフの形式で可視化する手法であり，対象領域をユーザが任意に移動するのに追従してグラフを連続的に変化させることで，樹状図内の任意領域をインタラクティブに閲覧できる[3]．図 1 に本手法による可視化のイメージを示す．対象領域に含まれる要素（またはクラスタ）をノードとし，任意の要素（またはクラスタ）の組に対する類似度をリンクとするグラフで表す．

探検型可視化手法の特徴は次の 2 点である．1 つは，クラスタリング結果のデータ規模によらず対象領域に対するグラフの要素数を一定以下に保つことで，常に見易いグラフで可視化する点である．もう 1 つは，対象領域に対応するグラフ要素の計算や，領域移動に追従するためのグラフの変化要素の計算を，クラスタリング結果のデータ規模によらず高速に処理できる点である．

2.2 節では，ユーザの注目要素を中心とする部分的な対象領域を，要素数が一定以下であるグラフで可視化するための各パラメータを説明する．2.3 節では，グラフを次状態に変化させるためのグラフの差分データについて説明する．

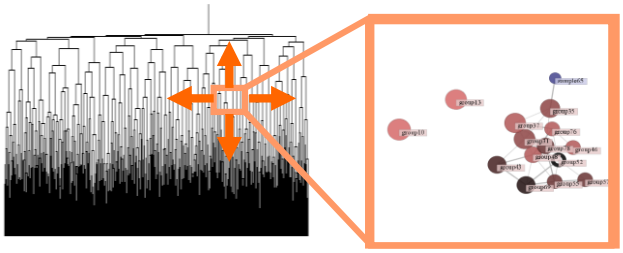

図 1 探検型可視化手法の可視化イメージ

2.2. 対象領域を決定するパラメータ

探検型可視化手法では，ユーザの注目要素を中心とする樹状図内の部分領域をグラフ化するが，本節では，この対象領域を決定するためパラメータを説明する．対象領域を決定するためのパラメータは，以下の 5 つである．

① 中心ノード
② 表示レベル
③ 中心からの表示最大距離
④ 画面あたりの最大ノード数
⑤ 画面あたりの最大リンク数

それぞれのパラメータの役割を説明する．

①は，ユーザの注目する要素である．クラスタリング結果の樹状図では，葉はクラスタリングの対象データの各要素を表し，節はクラスタリング過程で求められた各クラスタを表す．樹状図内の任意の葉または節を中心ノードに指定できる．中心ノードを変更することで，樹状図内を（主に）横方向に移動できる．

②の表示レベルは，対象領域に対する表示粒度を定めるパラメータである．図 2 の②に示すように，クラスタリング結果の樹状図を水平に切り出す高さを表す．各節ノードの根に近い方から 1, 2, …, N-1 の値を割り当て，その値が k である位置で樹状図を水平に切断して得られる k 個のノードは，クラスタリング対象の N 個の全データを k 個に集約した結果である．このように，表示する要素の集約度合いを決定する値を表示レベルという．表示レベルを変化させることで，樹状図の表示粒度を自由にさせることができる．

③の中心からの表示最大距離は，中心ノードを中心としてどの程度離れた範囲までを対象とするかを定めるもので，この値により対象領域の横幅を決定する．中心ノードからの距離は，図 2 の③に示すように，◎

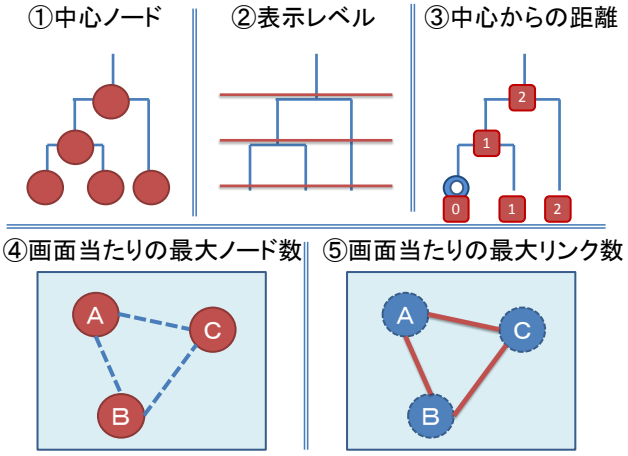

図 2 5 つのパラメータ

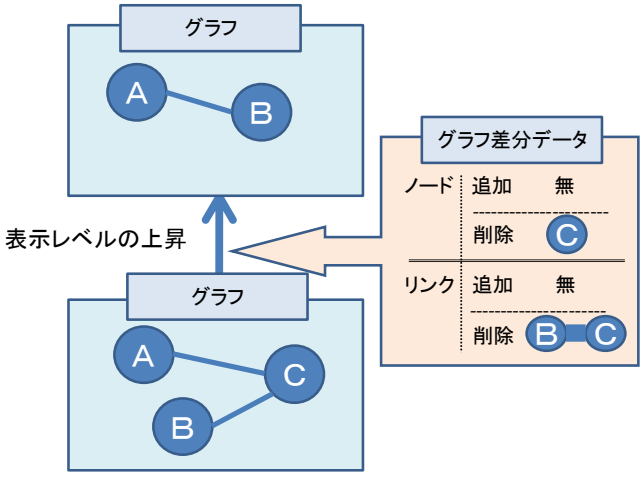

図 3 グラフ差分データ

で表した中心ノードの中心からの距離を 0 とし，その親にあたる節ノードに 1，そのまた親ノードには 2 というように，中心ノードを始点として各親ノードに 1 ずつ加算した値を中心からの距離として設定する．残りの全ノードは，それぞれの親ノードと同じ値を設定する．中心からの表示最大距離よりも中心からの距離が大きいノードは，表示対象から除外する．

④⑤の画面あたりの最大ノード数および最大リンク数は，グラフの見易さを保つために，グラフの要素数を決定する．対象領域に，多数のノードが含まれる場合は，中心からの距離が大きいノードから順に親ノードに置換する．これにより，中心ノードの近辺は表示レベルに従った粒度で，中心ノードから離れたところは粗めの表示粒度で概要を表現する．つまり，魚眼レンズのように表示粒度を歪めることで，グラフに表示するノード数を一定以下に抑えるのである．最大ノード数以下の表示ノードの任意の組合せに対する類似度がリンク候補となるが，要素数を制限するために，類似度の大きい順に最大リンク数分のリンクのみをグラフに表示する．

ここで，パラメータの役割を①②と③④⑤に大別できる．前者の①②は，クラスタリング結果内の可視化領域の移動を行うためのパラメータであり，①はクラスタリング結果内の可視化領域の横の移動，②は縦の移動と意味づけることができる．後者の③④⑤は，規模の大きなクラスタリング結果をグラフで可視化する際に，グラフが過密になるのを防ぐことが目的であり，可視化領域の大きさを決定するパラメータである．

2.3. グラフの形状変化のための差分データ

探検型可視化手法では，ユーザによる対象領域の移動に合わせて 2.2 節で説明したパラメータを変更することでグラフを連続的に変化させる．例えば，対象領域の上下移動は表示粒度の変更に対応するため，表示レベルを変化させることでグラフを変化させる．本節では，表示レベルの変更を例として，グラフの連続的な形状変化を実現するために生成するグラフ差分データについて述べる．

グラフ差分データは，表示レベルの変更前後に対応した 2 つのグラフ間の差分要素の集合である．具体的には，変更後のグラフに変形するための，追加ノード，削除ノード，追加リンク，削除リンクである．

図 3 の下のグラフから上のグラフへと変化させる場合の例を用いてグラフ差分データを説明する．下のグラフには，3 つのノード A，B，C と 2 つのリンク A-C，B-C の要素が含まれている．これに対し，上のグラフは 2 つのノード A，B と 1 つのリンク A-B で構成されている．したがって，グラフの差分データは，削除ノードがノード C，削除リンクがリンク B−C となる．

グラフ変化には，2 つの兄弟ノードから 1 つの親ノードへの結合や，1 つの親ノードから 2 つの子ノードへの分割も多いため，その場合は分割／結合前のノードが削除ノード，分割／結合後のノードが追加ノードとなる．

このように現段階の状態のグラフと次状態のグラフとのグラフ差分データを生成することで，グラフの連続的な変化を実現している

3. サーバクライアントモデルでの実装

3.1. システム構成

本章では，探検型可視化手法をサーバクライアントモデルでの実装するシステム構成を述べた後，サーバとの通信回数を低減するためのグラフ差分データの事前取得および，クライアント側でのグラム差分データの生成について説明する．

サーバクライアントモデルでの実装では，サーバ側

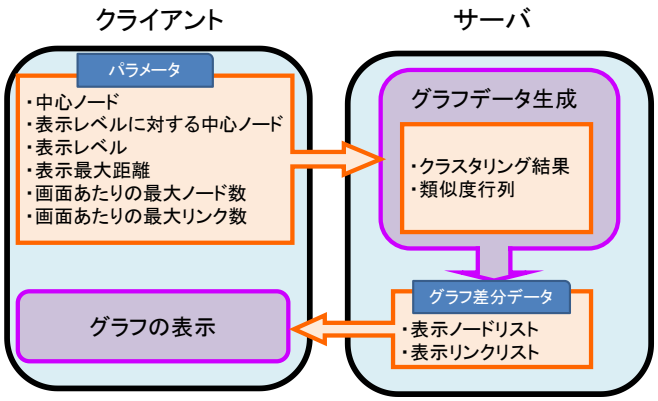

図 4 本モデルでのシステム構成

でクラスタリング結果と類似度行列の全体を保持し，クライアント側では，対象領域に対するグラフを閲覧しながら対象領域の移動操作を行う．したがって，図 4 に示すように，サーバ側ではクライアント側から送られたパラメータセットに従ってグラフ表示に必要なデータを作成し，クライアント側はサーバから送られたグラフ差分データに従ってグラフ表示を更新する．対象領域の移動があれば，移動後の領域を表わすパラメータセットに必要なグラフ差分データを表示に反映することで，ユーザによる対象領域の移動操作に追従する．

3.2. グラフ差分データの事前取得

サーバクライアントモデルでの探検型可視化システムでは，クライアントが対象領域を移動するごとにグラフ差分データをサーバに問い合わせて取得する必要がある．しかし，グラフ差分データを逐一問い合わせていたのでは，サーバへの通信負荷がかかり，クライアントへの応答に遅延が生じる恐れがある．そのため，このようなサーバクライアントモデルでは，一般的に，クライアント側で必要となることが予想できるデータを，あらかじめクライアント側に事前取得させることが多い．

本研究では，探検型可視化手法の対象領域の移動操作のうち，もっとも頻繁に使用される表示レベルの変更による表示粒度の変更操作に着目する．表示レベルを変更した場合，クラスタリング結果内の上下移動が起こる．他のパラメータを変更した場合に比べ，表示レベルの変更に伴うグラフ差分データに対するサーバへの要求が頻発すると考え，表示レベルの変更に伴うグラフ差分データを事前取得することで，サーバへの通信回数を減らし，グラフ変形の遅延を防ぐ．

図 5 に示すように，ある時点の表示レベルに対し，表示レベルを一定回数，上昇または下降させた場合のグラフ差分データを事前に取得する．図 5 の場合，表

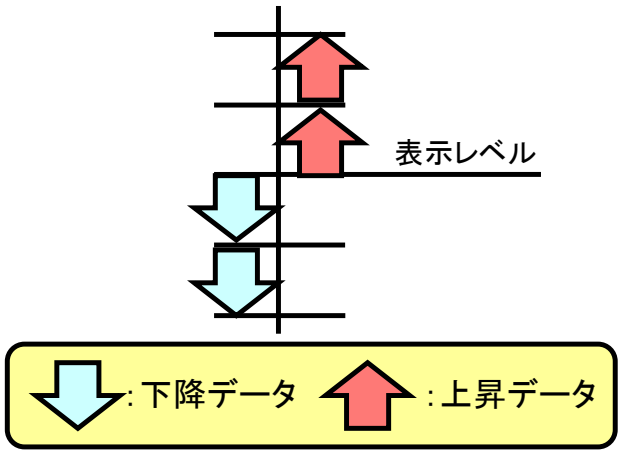

図 5 グラフ差分データの一括取得

示レベルを上下 2 段階分まで移動させても，サーバにアクセスすることなく，グラフ変化が可能となる．

表示レベルの移動を繰り返すことで，事前取得したグラフ差分データが事前取得の最低状態数を下回った時点で，サーバに対し追加の差分データを要求し，再び一定回数分のグラフ差分データを取得する．ただし，クライアント側で保持するグラフ差分データが増大しすぎるのを防ぐため，保持する状態数に閾値を設け，それを超過した分はその時点の表示レベルから離れたところから順に破棄する．

クライアント側での初期グラフ表示時に，グラフ差分データを一括取得すると，サーバ側での処理量が増加し，グラフ表示に遅延が発生する恐れがある．そこで，初期グラフの生成時には，グラフ差分データの事前取得はせずに，初期グラフの描画を開始してからサーバからの一括取得を行うものとする．

3.3. クライアント側でのグラフ差分データの生成

3.2 節では，サーバへの通信回数を低減するためのグラフ差分データを事前取得について説明した．これに加え，本研究では，クライアント側でも生成可能なグラフ差分データがあることに着目し，サーバに問い合わせることなく，クライアント側でのグラフ差分データの追加生成を行う．

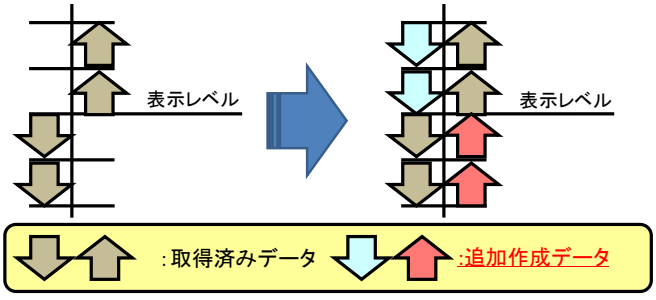

図 6 クライアント側で追加生成する
グラフ差分データ

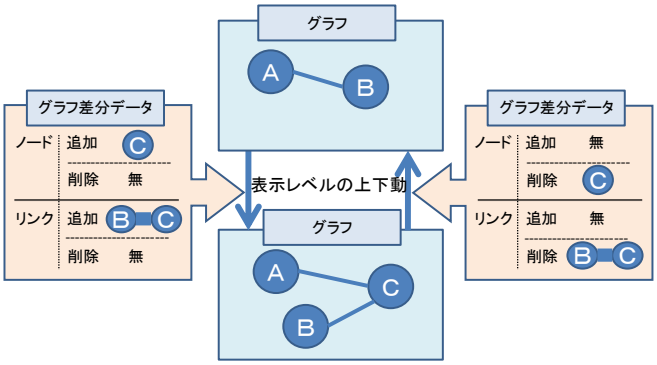

図 7 グラフ差分データの中身

表示レベルの変更対するグラフ差分データのうち，クライアント側で生成可能なグラフ差分データについて説明する．グラフ差分データの事前取得では，図 6 右に示すように，各表示レベルに対して上昇・下降させた場合のグラフ差分データが必要となる．このうち，青色と赤色の矢印で示した部分が，クライアント側で生成可能なグラフ差分データである．

クライアント側でどのようにグラフ差分データを追加生成するかを説明する．探検型可視化手法では，ノード追加や削除，リンク追加や削除によってグラフを変形する，図 7 に，表示レベルの変更を 1 状態だけ上昇または下降する操作を行う際に必要となるグラフ差分データを示す．ここで，グラフ差分データの中身に注目すると，上昇データの追加ノードは，下降データの削除ノードとなる．同様にリンクの追加，削除する要素も，上昇と下降のデータでは，追加が削除に，削除が追加に入れ替わったものがグラフ差分データとなる．このことから，上昇データまたは下降データのどちらか一方を利用し，グラフ差分データの中身を逆転することで，対となるグラフ差分データを生成することができる．

したがって，サーバからは各表示レベルに対して上昇または下降のうちいずれか一方のグラフ差分データのみを取得し，それらを利用して，ついとなるグラフ差分データはクライアント側で生成する．これにより，サーバでのグラフ差分データの算出処理と，通信するデータ量を半減することができる．

3.4. クライアント側でのグラフ差分データ生成のタイミング

サーバからグラフ差分データを事前取得した際に，クライアント側で追加生成が可能なグラフ差分データをすべて生成してしまうとクライアント側に処理の負荷がかかる．図 8 上は，サーバから事前取得するのと同時にグラフ差分データを追加生成した場合で，図 8 下は，ユーザが表示レベルの変更を要求した時点で，そ

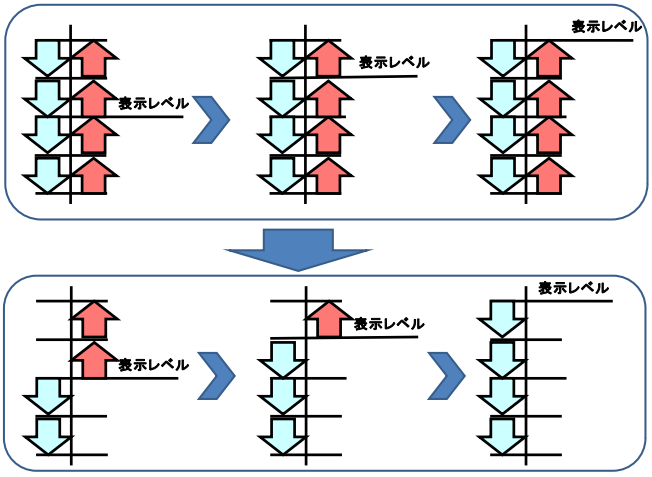

図 6 グラフ差分データを追加生成するタイミング

の変更に使用する 1 つのグラフ差分データに対して対となる 1 つのグラフ差分データを追加生成した場合を表わす．後者の場合，クライアントが保持するグラフ差分データの状態数を半分に抑えることができる．

したがって，クライアント側でのグラフ差分データ追加生成は，表示レベルの変更が生じてグラフ差分データを使用する際に行う．また，不要なグラフ差分データはその都度破棄するものとする．これにより，クライアント側のメモリの使用量が軽減できる．

4. 試作システム

4.1. システム概要

今回構築したサーバクライアントモデルでの探検型可視化システムは，探検型可視化手法に必要なパラメータと得られたグラフを表示する部分と，階層的クラスタリング結果と類似度行列を別々に保持し，通信部分にソケット利用したネットワークプログラミングで稼動する，Java アプリケーションである．グラフの描画ライブラリとして Processing[4]，TRAER.PHYSICS[5]および TRAER.ANIMATION[6]を用いている．

クライアントでは，指定されたパラメータに対応するグラフを表示する．領域の移動はユーザの操作により行われ，操作に応じて各 Java アプレット内の描画は連続的に変化する．サーバでは，起動時にクラスタリング結果と類似度行列を格納したファイルデータを読み込んで待機し，クライアントからのアクセスがあれば，受信したパラメータに対応するグラフ要素を計算し，クライアントに送信する．

4.2. システム画面

図 9 は，本モデルで実装を行った試作システムの画

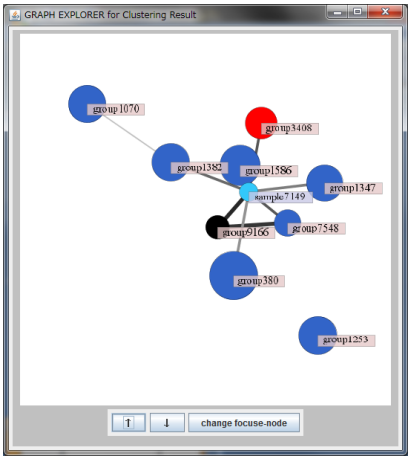

図 7 システム画面

面である．画面内のノードは要素，または，クラスタ（クラスタは含まれる子ノードの量に大きさが比例している）を表し，また，各ノードの識別を行えるようにラベルが貼られている．画面下部にあるボタンについて，「↑」および「↓」のボタンにより表示レベルの変更が行える．また，任意のノード選択後「change focuse-node」ボタンをクリックすることで，選択したノードおよびクラスタを中心ノードとしたグラフが再描画される．

5. 評価実験

5.1. 実験方法

探検型可視化手法をサーバクライアントモデルで実装したことで問題となり得るのが，クライアント側の負荷（特にメモリ使用量）と通信による遅延の影響である．そこで，クライアント側の負荷がどの程度であるかを確認するために，試作システムを用いてクライアント側のメモリ使用量を計測した．また，その結果をもとに，クライアント側で保持できるグラフ差分データの状態数の検討を行う．

実験条件を説明する．可視化対象として要素数 10,000 のクラスタリング結果データを用いて，クライアント側のパラメータセットを次のように変化させて比較する．画面あたりの最大ノード数を 25，50，100，それぞれに対して画面あたりの最大リンク数を，ノード数の 1 倍，2 倍，3 倍として，9 種類のパラメータセットで実験する．各パラメータセットに対し，クライアント側でグラフ差分データを 100 状態，200 状態保持した場合のメモリ使用量を計測する．試行回数は，全 18 パターンの各条件に対して各 100 回とする．

グラフ差分データ 200 状態での平均メモリ使用量とグラフ差分データ 100 状態での平均メモリ使用量との差を用いて，各パラメータセットでの，グラフ差分データ 1 状態あたりのメモリ使用量を求め，クラインと

表 1 グラフ差分データ 100 状態あたりのメモリ使用量

		ノード数		
		25	50	100
リンク率	1	19.407(±0.133)	19.686(±0.489)	20.174(±0.619)
	2	19.535(±0.132)	20.423(±0.413)	20.808(±0.431)
	3	19.639(±0.212)	20.242(±0.291)	20.258(±0.286)

(MB)

表 2 グラフ差分データ 200 状態あたりのメモリ使用量

		ノード数		
		25	50	100
リンク率	1	19.655(±0.253)	20.246(±0.480)	20.879(±0.506)
	2	19.758(±0.096)	20.723(±0.179)	21.150(±0.199)
	3	20.024(±0.210)	20.780(±0.542)	21.392(±0.350)

(MB)

表 3 グラフ差分データ 1 状態あたりのメモリ使用量

		ノード数		
		25	50	100
リンク率	1	2.482	5.600	7.044
	2	2.233	3.000	3.422
	3	3.850	5.379	11.333

(KB)

側で保持できるグラフ差分データの状態数を検討する．

5.2. 結果と考察

実験結果として，表 1，2 に，各パラメータセットにおいてグラフ差分データを 100 状態，200 状態に保持した場合の平均メモリ使用量を示す．また，これらの結果から算出した，グラフ差分データ 1 状態あたりのメモリ使用量を表 3 に示す．その結果，各パラメータセットにおいて，グラフ差分データ 1 状態あたりのメモリ使用量は，最大でも 12KB 程度であることがわかった．

このことから，クライアントが，グラフ差分データを 100 状態保持したとしても，クライアント側のメモリ使用は 1MB 程度の負荷に収まり，かなり多くのグラフ差分データを保持していたとしても大きな負荷にないことがわかる．

したがって，クライアント側でグラフ差分データを大量に保持しておくように設定することで，サーバに遅延が発生しても，クライアントは事前取得したグラフ差分データを使用してグラフの変形操作が可能になることがわかった．

6. おわりに

本研究では，大規模なクラスタリング結果の部分領域をグラフとして可視化する探検型可視化手法の複数ユーザを意識した，システム構築を目指すため，サーバクライアントモデルでの実装を行った，また，クライアントのグラフ変化に伴うグラフの差分データの取

得を事前取得に行うことにより通信の回数を減らすと共に，サーバへ対するクライアントからのアクセス集中時に考えうる,グラフ変形までの遅延を軽減できた.これにより，クライアントでは，探検型可視化手法によるスムーズなグラフの閲覧が可能になった.

今後の課題として，本評価は１クライアントでの評価であるが，クライアント数を増やして評価を行うことが挙げられる．また，グラフ内に表示する情報は，適用分野によって異なるが，それによってグラフ差分データも変化すると考えられるため，分野ごとの情報の要求も視野に入れ，今後検討していくことが必要である.

参　考　文　献

[1] 神嶌敏弘, “推薦システムのアルゴリズム(1)~(3)”, 人工知能学会誌, vol.22, no.6 ~ vol.23, no.2, 2007-2008.

[2] 堀幸雄, 今井慈郎, 中山堯, “協調フィルタリングを用いた共著関係の予測”, 情報知識学会誌, vol.18, no.2, pp.99-104, 2008.

[3] 井上悦子, 吉廣卓哉, 中川優, “大規模クラスタリング結果のグラフによるインタラクティブな可視化手法”, 電子情報通信学会和文論文誌 D, vol.J92-D, no3, pp.351-360, 2009.

[4] 前川峻志, 田中孝太郎, “Built with Processing”, 株式会社ビー・エヌ・エヌ新社, 2007.

[5] TRAER.PHYSICS, http://www.cs.princeton.edu/~traer/physics/ （2009/10/27 アクセス）

[6] TRAER.ANIMATION, http://www.cs.princeton.edu/~traer/animat （2009/10/27 アクセス）