D-007

ばねモデルを用いた検索結果のグラフレイアウト手法

Graph Layout method of search result using model

垣崎　正宏†　　上原子　正利†　　小柳　滋†
Masahiro　Kakizaki　Masatoshi Kamiharako　Shigeru　Oyanagi

1．はじめに

検索結果の視覚化はデータの関係を一目で把握することができるため重要である．一般的な視覚化はデータをグラフとみなすものである．そのデータを視覚化したものから何を理解するか、視覚化の見やすさの基準は人それぞれに異なるため、どのグラフレイアウトが適切とは一概にいえない．本研究では検索結果データを類似度の尺度からグラフレイアウトして配置する．類似度の算出方法は、ベクトル空間法を用い、内積を計算するものとする．この類似度を用い、データ間の理想距離を算出して、データを配置する．アルゴリズムはばねモデルを用いた．この手法の有効性を実証するために実装を行い、検証実験を行った．

2. グラフレイアウトの概要

本章ではグラフレイアウト手法について述べる．グラフの描画では、ノードを点、円または図形で表現しそれらを結ぶ線分によってエッジを記述する場合が多い．このようなグラフ描画をグラフレイアウトと呼ぶ．グラフ描画においてこれらの要素をどのように空間上に配置するかが問題となる．

2.1 モデル的手法

モデル的手法とは力学モデルや生物モデルなどを用いてグラフレイアウトに関する制約条件を表現し、そのモデルを解く事によって最適レイアウトを求める手法である．グラフの各要素に関して様々な力学的作用を仮定し、そのエネルギー安定状態を求める手法を力指向的手法(Force-Directed Method)と呼ぶ．以下でこのモデルを用いたグラフレイアウト手法の研究例について述べる．

2.1.1　Eades の"Spring Embedder"

力指向的手法によってもっとも簡潔でよく知られたのがEades[1]による"Spring Embedder"である．このモデルでは、エッジを自然長を持ったばねと仮定し、エッジで接続したノード *i,j* 間にばね力を模した式(2.1)に示す *Fs(i;j)*、隣接しないノード同士には逆２乗則の斥力によって互いに反発しあう式(2.2)に示す力 *Fr(i;j)*を仮定する．

$$Fs_{(i,j)} = c_1 * log\left(\frac{d_{ij}}{c_2}\right) \qquad (2.1)$$

$$Fr_{ij} = \frac{c_3}{\sqrt{d_{ij}}} \qquad (2.2)$$

ただし、ここで *c1*、*c2*、*c3* は定数、*dij* はノード間の距離である．各ノードに働く力の合計を計算し、それに従ってノードを移動させる処理のくり返し計算でレイアウトを変形させることにより、ノードが近づきすぎず、隣接ノードが近くに配置され、エッジの長さが均一に近いレイアウトが得られる．

2.1.2 Kamada,Kawai(KK)モデル

Kamada,Kawai[2]は Eades のモデルから逆２乗則の斥力を取り除いた．この KK モデルでは、すべてのノード同士がばねでつながれていて、各ばねはノード間のグラフ理論的な距離（最短パス距離）によって決まる自然長を持つモデルを提案している．モデル全体のエネルギー*E* は次式によって定義される．

$$E = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{1}{2} K_{v_i v_j} (d_{v_i v_j} - l_{v_i v_j})^2 \qquad (2.3)$$

ただし、N はグラフにおけるノードの個数、*Kvivj* は頂点 *vi,vj* 間を結ぶばね定数、*lvivj* はそのばねの自然長、*dvivj* は頂点 *vi,vj* 間の距離である．このモデルは距離が近いほどノードが近くに配置されたレイアウトを得る事ができる．

KK モデルの計算方法の手順は Eades のばねモデルがグラフ全体の移動を繰り返すのに対し、１回の繰り返しで１つのノードのみを移動させる．移動させるノードは、式(2.3)のエネルギーを偏微分することによってノード *vi* のエネルギー*Ei* に基づいて決定する．移動量は Newton-Raphson 法によって決定される．

3. グラフレイアウトの実装

レイアウトするデータは検索結果データを用いる．アルゴリズムは第２章でのべた KK モデルのアルゴリズムを利用する．KK モデルのエネルギーを決定付けるため、理想距離が必要になり、理想距離は類似度を変換することによって決定する．この全体エネルギーを収束させたものを描画する．この方法で視覚化すると類似したデータが画面上で近くに配置されるため、データ関係が一目で把握できるようになる．しかし KK モデルでは１つずつノードを移動させたが、検索結果データをどれから移動させるかは決定できない．そのため、同時に動かすことによって全体的な最適なレイアウトが可能になる．

3.1　本実験で用いたレイアウト手法

頂点 *i* と頂点 *j* 間の理想距離（ばねの自然長）を *lij*、実際の頂点間の距離を *dij*、ばね定数を *kij* としたときばねに張られた頂点間の働く力は、

$$|\vec{f}_{ij}| = k_{ij}(d_{ij} - l_{ij}) \qquad (3.1)$$

となる．ただし *kij* は任意の整数とする．*kij* が大きい程そのばね全体への影響が大きくなり *f* が正の時は引力となり、負の時は斥力となる．この時全体のエネルギーの総計は、

$$E = \sum_{i} \sum_{j} \frac{1}{2} k_{ij} (d_{ij} - l_{ij})^2 \qquad (3.2)$$

となる．ある２つの頂点間の座標を(*xi,yi*),(*xj ,yj*)とすると、これらの間のエネルギー*Eij* は

$$E_{ij} = \frac{1}{2} k_{ij} \left(\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} - l_{ij} \right)^2 \qquad (3.3)$$

である．これより、

†立命館大学

$$\frac{\partial E_{ij}}{\partial x_i} = k\left((x_i - x_j) - \frac{l_{ij}(x_i - x_j)}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}\right) \quad (3.4)$$

$$\frac{\partial E_{ij}}{\partial y_i} = k\left((y_i - y_j) - \frac{l_{ij}(y_i - y_j)}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}\right) \quad (3.5)$$

となる．これを用いて、ばねの力のベクトルは

$$\vec{f}_{ij} = \left(\frac{\partial E_{ij}}{\partial x_i}, \frac{\partial E_{ij}}{\partial y_i}\right) \quad (3.6)$$

となる．したがって、各点の偏微分値はその頂点に働く力の成分に相当する．そして、各頂点の座標で偏微分した値を移動量として定義すればE(式(3.2))を最小にできる．さらにcを任意の定数として、$c\ \vec{f}_{ij}$を移動量として定義することにする．cが大きければ移動量が大きくなるため、速く収束できるが収束できない場合が多い．逆に小さければほとんど収束する．

3.2 類似度算出と理想距離への変換

類似度算出方法としてベクトル空間法を用いる．単語の重み付けは TF・IDF を用い、類似度算出には余弦を用いた．データ間の類似度を距離空間に反映するため、類似度Rijから次式のように理想距離lijを算出する．

$$l_{ij} = m\log\left(\frac{1}{R_{ij} + 0.01}\right) \quad (3.7)$$

このように類似度の逆数を理想距離にする．さらに対数をとることで頂点のかさなりすぎを防ぐ．分母の 0.01 は内積値が０の場合があるので除数が０になることを防ぐ．Mは任意の定数で空間配置しようとする画面の大きさにより調整するものである．

4 実験.

実際のデータに対してグラフレイアウトを行う．本実験ではデータとして UCI KDD Archive[3]で公開されている NFS の Part1.zip を用いた．このデータからランダムに１０件文書をとり、P1 から P10 とする．その P1 から P10 のそれぞれのペアに対して類似度を計算する．そして、理想距離を式(3.7)を用いて算出する．初期配置はランダムで行う．実行結果を図 4 に示す．ループ回数は 1123 回であった．図中の矢印はノードの移動を示す．類似したノードが近づいているかは次章で確認する．

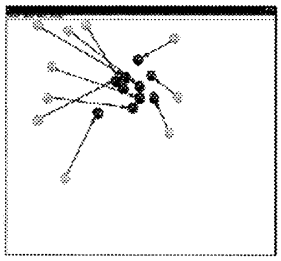

図 4:実行後の配置

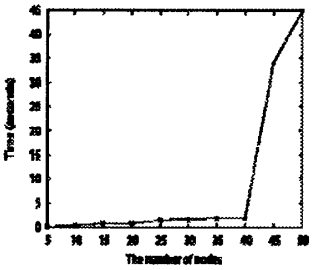

図 7：処理速度のグラフ

5. 評価と考察

前章の実験結果を評価するため、まず手法の核であるエネルギーの収束を確認する．エネルギーが０に近く収束するほど理想的配置といえる．これを確認するため、１０件のデータからなる５つのデータセットに関して実験を行った．その結果、全体エネルギー量は急速に減少し、収束している事が確認できた．これより手法が適切であった事が言える．

次に配置が適切であったかどうかの確認を行う．今回の評価では基準データから理想距離の順位に対する実際の配置上での距離による順位を示す値 A を用いる．これを次のように定義する．

$$A_i = \frac{1}{|ldr_i - adr_i| + 1}(N - rank_i) \quad (5.1)$$

ただし、Nは総データ数、$ldri$は基準データに対してのデータiの理想距離の順位、$adri$は基準データに対してのデータiの理想距離の順位である．$1/(|ldr_i - adri|+1)$は順位のずれで、ずれがないほど１に近い．$(N - ranki)$は基準データに対して順位の重みづけである．１位であればN-1の重みがつき、最下位であれば最低の１がつく．この式の最大値は$N(N-1)/2$になる．この最大値に近ければ近いほど、最適配置である．実際の結果の最大と最小を以下に示す．上は理想距離の順位順で下は処理後の順位である．

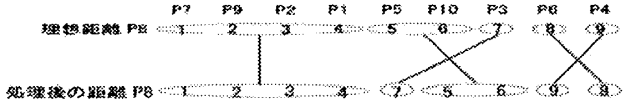

図5：P8に対しての配置順位の比較（Ap8 = 36.67）

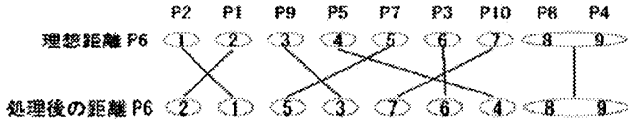

図6：P6に対しての配置順位の比較（Ap6 = 23.10）

この１０件の平均値は 28.72 となる．理想距離に近い配置ができていることがわかる．この数値を改良するには、初期配置をランダムではなく理想距離を用いて決定する事で実現できると考えられる．

最後に処理時間の計測を行う．５ノード刻みに 5~50 までの１０段階行う．処理時間は各ノード数の３回の平均を求める．この結果を図 7 に示す．x 軸にノード数、y 軸に処理終了までの時間を示す．この図からわかるように、ノード数が多くなると急激に処理時間が長くなる．40 ノードにいたってはループ回数が最大回数まで到達したが収束しなかった．この原因は、ノードの移動量が少ないため、定数 c の値を変更することによって解決できると考えられる．このアルゴリズムでは４０ノードまで１秒程度の時間で処理できる．ユーザにストレスを感じさせないためには２０程度までのノードが適切である．

6. おわりに

本論文は検索結果データにおいてグラフレイアウトできるかを検証した．検証結果より類似度の尺度からグラフ生成が可能であると言える．

7. 参考文献

[1] P. Eades. A heuristic for graph drawing. *Congressus Numerantium*, Vol. 42, pp. 149-160, 1984.
[2] T. Kamada and S. Kawai. An algorithm for drawing general undi-rected graphs. *Information Processing Letters*, pp. 7-15, 1989.
[3] . http://kdd.ics.uci.edu.
[4] 松浦：“ベクトル空間モデルにおける単語重み決定の一般化”、DEWS2007