

無閉路有向グラフ可視化のための画面配置とカテゴリ描画の一手法

理学専攻 情報科学コース 東原 真希

1 はじめに

階層型データは身近に存在するデータ構造であり、その可視化には多くの視点から研究が進められている。一方で身の回りには DAG (Directed Acyclic Graph: 無閉路有向グラフ) 構造を構成するデータも多数あり、その可視化にはまだ議論の余地が多い。本論文では、空間充填型の大規模階層型データ可視化の一手法「平安京ビュー」[1] を拡張し、DAG 構造を持つ階層型データの可視化を試みる。以後、DAG 構造を持つ階層型データのことを DAG と略する。

2 関連研究

2.1 平安京ビュー

本研究では、大規模階層型データに対する空間充填型の可視化手法「平安京ビュー」[1] を使用する。平安京ビューでは、葉ノードを長方形のアイコン、親ノードを長方形の枠で描画し、階層構造を親ノードの枠の入れ子で表現している。それらすべての長方形を空間充填モデルに基づいて配置することによって、大規模階層型データの全体を一画面で表示する。また、葉ノードに持つ値を高さとして持たせることができるので、二次元でデータ構造を表現し、画面を回転することで要素の値を比較することができる。可視化結果においてノードが格子状に配列される様が平安京の地図のように並ぶことから命名された。平安京ビューはデータの葉ノードと親ノードの階層構造よりも、葉ノード群を一望することに主眼を置いた手法である。

2.2 FRUITSNET

FRUITSNET(Framework and User Interface for Tangled Segments Network)[2] とは、アイテム情報が付加されたノードを有する階層型ネットワーク構造の可視化の一手法である。FRUITSNET は平安京ビュー同様、葉ノードをアイコンで、親ノードを長方形の枠で表現するが、平安京ビューとの大きな違いはその配置手法にある。平安京ビューの配置は長方形の空間充填モデルにのみ基づいていたのに対し、FRUITSNET では力学モデルと空間充填モデルを併用している。それらを併用することによって、リンクで連結されたノードや共通のアイテム情報を有するノードを画面上で近くに配置し、同時描画面積の提言やノードの重なり回避を実現する。

3 研究概要

ここで従来手法での DAG 構造を持つ階層型データの処理について簡単な図を用いて説明する。図 1(左)に簡単な DAG を示す。この例では色のついた葉ノード 2 が親ノード A と親ノード B に所属している。これを平安京ビューで表示するには図 1(右)のように、葉ノード 2 を所属する親ノードの数だけ複数回描画する必要がある。しかし、この時平安京ビューで可視化すると、一つの葉ノード 2 から複数の親ノードの存在が読み取れない、全体からも複数の親ノードを持つノ

ードが存在することが読み取れない、また、葉ノードの数が本来の葉ノード数と異なるなど、本来のデータ構造を読めなくなってしまう。そこで本手法では図 2(左)のように、新たに親ノードを作成し、その下に葉ノードを格納することを考える。

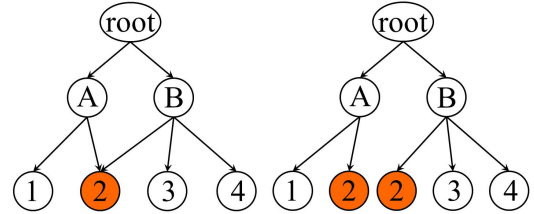


図 1: (左) 本来の DAG 構造 (右) 平安京ビューで DAG を可視化する際の一般的な木構造への変換

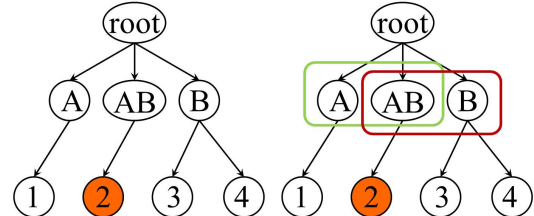


図 2: (左) 親ノード AB を作成 (右) カテゴリの追加

新たに親ノード AB を作成することによって葉ノード 2 を参照した時、複数の親ノード、葉ノード 2 を参照した時、複数の親ノードに属することが読み取れるようになる。しかし、ここで親ノード A に所属する葉ノードが葉ノード 1 と葉ノード 2 の 2 つが存在するにも関わらず、葉ノード 1 のみ親ノード A に所属すると誤読される恐れがある。親ノード B に関しても同様のことが言える。そこで図 2(左)における親ノード間の関連を明示するために、図 2(右)に示す緑色と赤色の枠のように親ノードに関する情報を付加する。この枠は図 1(左)の親ノードに対応する。本論文ではこれを「カテゴリ」と称する。以上の処理によって、DAG 本来の構造を可視化することを考える。

4 提案手法における DAG の画面配置

カテゴリ描画を実現するにあたり、ノードの画面配置は非常に重要である。特に同一のカテゴリに属している親ノードは画面上で近くに配置されることが強く望まれる。平安京ビューが採用する空間充填モデルは画面配置を小さくすることに有効であるが、関連のある親を近くに配置するという要求にはそのままでは対応しない。この問題を解決するために 2.2 節で紹介した FRUITSNET の画面配置アルゴリズムを適用する。本手法では、同一カテゴリに属する親ノード間を架空のリンクで接続し、このリンクが適切な長さを保つような力学モデルを適用する。その結果を理想位置とす

ることで、同一カテゴリに属する親ノード同士を画面上で近くに配置させ、かつ描画領域を小さく抑える。図3はあるデータをFRUITSNetの配置アルゴリズムを適用した結果である。ピンク色でハイライトされた親ノードが一つのカテゴリに属している親ノードを表しており、それらが隣接して配置されていることがわかる。このデータにはカテゴリが114あったが、そのうち94に関しては画面を四分分割した範囲内にすべての親が収まった。カテゴリ内の親が離れたところに配置された親に関してはカテゴリ描画で問題を解決する。

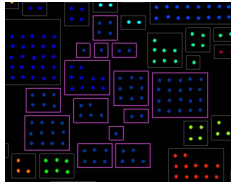


図3: FRUITSNetの配置アルゴリズムを適用したデータ(拡大図)

5 提案手法におけるカテゴリ描画

5.1 長方形での問題点

我々はカテゴリ描画実装の第一段階として、長方形でカテゴリ描画を実装した。図4はその実行例であり、指定した一つのカテゴリのみを赤い長方形で描画している。また、カテゴリに所属する親をピンク色でハイライトしている。長方形ではカテゴリの描画面積が必要以上に大きくなる上、ピンク色でハイライトされていない親までもがカテゴリ内に収まってしまい不必要な重なりが生じている。

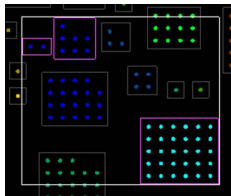


図4: カテゴリを長方形で描画した例

また、複数のカテゴリを描画する場合にもカテゴリに所属する親をハイライト表示することなく認識する描画が求められる。これらの問題点を解決するために、カテゴリ描画に関して次のようなバリエーションを考えた。一つは凸包でのカテゴリ描画である。長方形での描画の直感的なわかりやすさを保ちつつ、長方形よりは描画面積を小さくできる。しかしそれでも、カテゴリ内の親が隣接しない時、関係のない親への重なりが大きくなることもある。この問題を軽減する一手段として、枠と接続線での描画を考えた。この方法ではカテゴリ外の親への重なりを少なく抑えることができるため、親が離れて配置された時に効果がある。

5.2 カテゴリ描画結果

図5は図4と同一のカテゴリを凸包または枠と接続線で描画した例である。凸包の計算には包装アルゴリズム(Gift-wrapping Algorithm)を適用した。図5(左)

のと図4を比較すると、凸包での描画が長方形の描画に比べてカテゴリ外のノードへの重なりが減少していることがわかる。しかし未だいくつかの親ノードに関しては不必要な重なりを残す。それに対し図5(右)の枠と接続線での描画は不必要な重なりはなくなった。しかし、カテゴリ内三つの親のうち上部にある二つは配置が近すぎるために、接続線の存在に気が付かない恐れがある。また、カテゴリ内の全てのノードが隣り合わせて配置された時、接続線が煩雑になりカテゴリの読み取りが困難になることが考えられる。

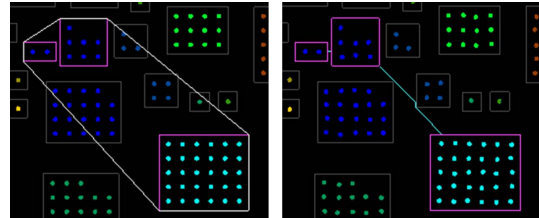


図5: (左)凸包でのカテゴリ描画例 (右)枠と接続線でのカテゴリ描画例

そこで、近くに配置された親に対してカテゴリを凸包で描画し、遠くに配置された親に対しては接続線を使って関連を示す、という二つの方法の併用を試みた。結果を図6に示す。ユーザに指定された距離以下の場合、ノード群を凸包で囲み、それらの凸包を接続線で連結している。これによりカテゴリ外の親への重なりなど上記二つのカテゴリ描画での問題点を概ね解決した。

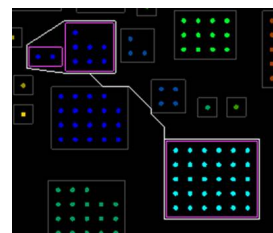


図6: 凸包と接続線を組み合わせた描画例

6 まとめ

本論文では、大規模階層型データ可視化の一手法「平安京ビュー」を拡張しDAG構造を可視化する手法を提案し、その実行例を示した。ノードの配置には平安京ビューの特性を活かしつつ、関連のあるノードを近くに配置するためにFRUITSNetの配置アルゴリズムを適用した。また、カテゴリ描画においては複数の描画を実装し、比較・検討を行った。

参考文献

- [1] 伊藤, 山口, 小山田, 長方形の入れ子構造による階層型データ可視化手法の計算時間および画面占有面積の改善, 可視化情報学会論文集, 26(2), 51-61, 2006
- [2] T. Itoh, C. Muelder, K.-L. Ma, J. Sese, A Hybrid Space-Filling and Force-Directed Layout Method for Visualizing Multiple-Category Graphs, IEEE Pacific Visualization Symposium, 121-128, 2009.