

筑波大学大学院博士課程

システム情報工学研究科修士論文

大規模 Web サイトにおけるユーザの行動分析の
ための可視化手法の開発

武田 慎之介

修士(工学)

(コンピュータサイエンス専攻)

指導教員 三末 和男

2015 年 3 月

概要

複数の機能を提供している大規模な Web サービスにおいて、ユーザがどのような機能をどのような順で使用してサイト上を行動しているかという、ユーザ全体の傾向を視覚的に分析することを支援するツールを開発した。大規模な Web サイトは、各機能を木構造に整理することが出来ることが多く、その木構造の各階層で集約して行動の流れを視覚的に表現する。この木構造の各階層を任意に切り替えて視覚的に表現することで、サイト全体でのユーザ行動の俯瞰的観察と、各機能にフォーカスをして詳細に観察することの双方を可能とする。本論文では既に収集されている各ユーザのアクセスログから木構造を持つ各機能ごとにユーザの行動を集計し、これを基にした視覚的な表現を作成することでサイト全体でのユーザの行動分析を支援するツールを開発した。

目次

第1章	はじめに	1
1.1	用語の説明	1
1.2	問題提起	2
1.3	目的	2
1.4	Web サイトの階層構造のグラフとページ遷移のグラフ	2
1.4.1	Web サイトの階層構造のグラフ	2
1.4.2	ページ遷移の有向グラフ	3
1.4.3	Web サイトの階層構造とページ遷移の混合グラフ	5
1.5	Web サイトの階層的な分析	5
1.6	本論文の貢献	6
第2章	関連研究	7
2.1	階層構造を持つグラフの可視化に関する関連研究	7
2.2	Web のリンク構造の可視化に関する研究	7
2.3	ユーザの行動分析に関する関連研究	8
2.4	本論文の位置づけ	9
第3章	要件の検討	10
3.1	分析者への要望のヒアリング	10
3.2	階層構造の出来るだけ多くのノードの表示	11
3.3	階層構造における各階層の分析	11
3.4	全ての流量の表示	11
第4章	開発した可視化手法	12
4.1	階層構造の表現	12
4.2	各階層ごとの遷移の表現	14
4.2.1	ノードの表現	15
4.2.2	incoming ノードと outgoing ノード	15
4.2.3	エッジの表現	15
4.2.4	ノードの配置	16
	自動的な配置	16
	手作業による配置の可能性	17

4.3	階層構造とユーザの遷移の混合グラフ全体での表現	19
	全体の観察と各階層の観察のためのインタラクション	19
第 5 章	可視化手法の実装	22
5.1	対象データ	22
5.2	処理の流れ	24
5.2.1	データの収集	24
5.2.2	データの集計	24
5.2.3	データの可視化	25
5.3	可視化の実装	25
5.3.1	視覚変数への割り当て	25
5.3.2	実装したインタラクション	26
	ズームとパン	26
	全体表示領域の移動	26
	エッジの表示と非表示の切り替え	26
	ノード位置の固定	26
第 6 章	ケーススタディ	29
6.1	全体の観察	29
6.2	階層にズームした観察	31
6.2.1	pages ノードへの注目	31
6.2.2	category ノードへの注目	31
6.3	考察	32
第 7 章	結論	34
	謝辞	35
	参考文献	36

図目次

1.1	クックパッドのフッターに表示された種類別のページ一覧	3
1.2	クックパッドのサイトの階層構造	4
1.3	Web サイトの階層構造とユーザの遷移の混合グラフ	5
2.1	本論文の位置づけ	9
4.1	サイトの階層構造の矩形の入れ子による表現	13
4.2	A というノードの階層での連結図によるグラフ表現の例	14
4.3	全てのノード間にエッジがあり、かつノードやエッジの重みが等しい時の配置例	18
4.4	全てのノード間にエッジがあり、エッジの重みのみを変えた時の配置例	18
4.5	全てのノード間にエッジがあり、ノードの重みを変えた時の配置例	19
4.6	複数の階層を組み合わせた全体の例	20
4.7	図 4.6 の中心付近をズームしたもの	21
5.1	<i>controller</i> を一階層目、 <i>action</i> を二階層目とした階層構造	23
5.2	可視化までの処理の流れ	25
5.3	図 4.7 の、上位の階層のエッジを非表示にした例	27
6.1	手作業で 1 階層目を配置した全体図	30
6.2	図 6.1 の「pages」ノードにズームした図	31
6.3	図 6.1 の「category」ノードにズームした図	32

第1章 はじめに

1.1 用語の説明

本節では、本論文中で用いる幾つかの用語について予め説明する。

大規模な Web サイト

ページ数が 100 を超えるような規模の Web サイトを大規模な Web サイトと呼ぶ。

ユーザ

Web サイトにアクセスする人のことをユーザと呼ぶ。本論文で開発した可視化手法を利用する使用者では無い。

ユーザ属性

Web サイトを利用するユーザの、例えば性別や年齢、アクセスに用いた端末のような属性情報のことを指す。

分析者

主に Web サイトを作成・運営する人のうち、特にユーザの行動の分析を行う人のことを分析者と呼ぶ。本論文は、この分析者が実際の分析作業を行う時に利用することを目指して可視化手法を開発する。

ページ

Web サイトを構成する Web ページのことを指す。

PV 数

サイト上のページのページビュー数のことを指す。本文中ではアクセス数と同義で用いる。

流量

サイト上のページ間を遷移するユーザの数を流量と呼ぶ。

1.2 問題提起

現在の Web サイトは非常に規模が大きくなってきている。ここで言う「規模」は様々な意味がある。一つの Web サイトにアクセスするユーザ数の規模が数千万人を超えることは現在では珍しくなく、またページ数の規模も数百ページ以上を有する Web サイトが増えている。更には、一つの Web サイトを数十人といった大人数で管理することもある。このように、様々な意味で規模が大きくなっている。Web サイト上のユーザの行動を把握することは Web サイトの運営や改善に有用であることが知られている [17] が、規模が大きくなるに連れて、Web サイト上をユーザがどのように行動しているかを制作者や運営者が把握することが非常に困難になりつつある。また、Web サイトの行動分析において、仮説を予め立てて、それを検証する形での分析手法から、先に仮説を立てずに探索的に分析する手法に移り変わりつつある。探索的な分析は、Web サイト全体でのページ数が少ない場合であれば、全てのページをひとつずつ調べるということが可能であったが、大規模な Web サイトでは、対象とするページが多く少人数で分析することが困難であるという現状がある。

1.3 目的

本論文の目的は、ページ数の規模が大きい Web サイト上におけるユーザの行動の、探索的な分析を支援することである。ページ数が大規模な Web サイト上のユーザの行動を把握するために、Web サイト全体のユーザのページ遷移を階層的に分析することが出来る可視化手法を本論文では開発する。

1.4 Web サイトの階層構造のグラフとページ遷移のグラフ

1.4.1 Web サイトの階層構造のグラフ

大規模な Web サイトは、よくそのサイトのページのフッター部に表示されているように、種類別に複数のページがまとめて管理されている。クックパッド¹の例では、サイトのトップページのフッター部に図 1.1 のように、種類別に複数のサービスやページが示されている。このフッターは、例えば図中の A で指し示している「プロのレシピ」であれば、リンク先はクックパッドのサブサービスの一つである「プロのレシピ」のトップページへと遷移し、更にその先に「プロのレシピ」サービスがサイトの構造として連なる。これはつまり、クックパッドというサイトは種類別にサブサービスが分けられ、更にその下層にサブサービスごとにページが連なっていて、さらに下層にページが連なるという階層構造を持つ。この階層構造は図示すると図 1.2 のようになる。

¹<http://cookpad.com/>

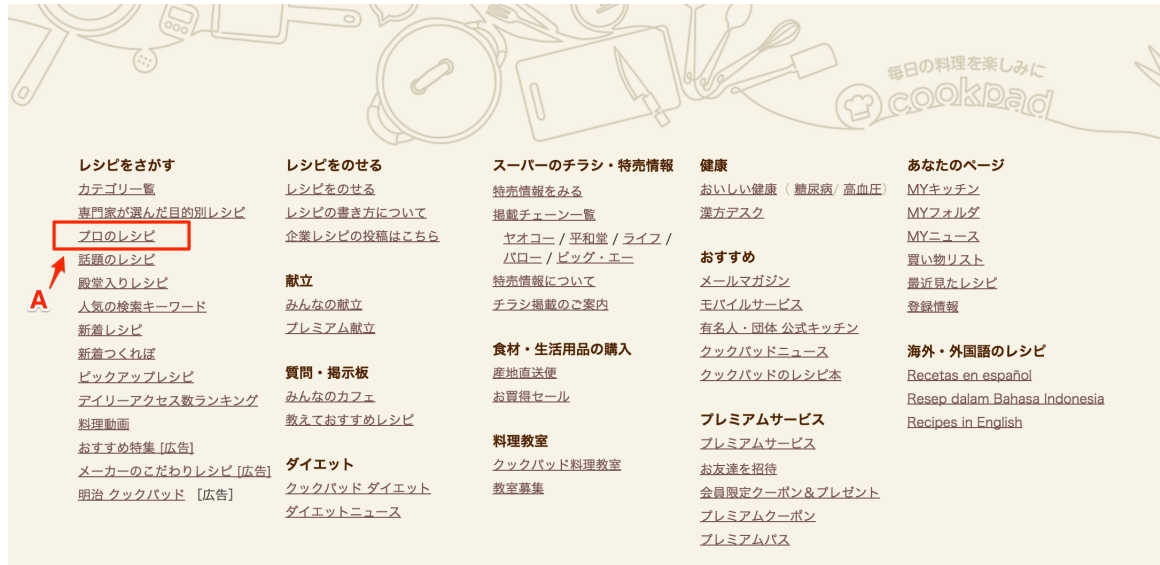


図 1.1: クックパッドのフッターに表示された種類別のページ一覧

この階層構造は、各階層のページやページをまとめた概念であるノードの集合を V 、ノード間の内包関係を表すエッジの集合を F とすると、階層構造を表すグラフ D_h は

$$D_h = (V, F)$$

と表現出来て、 V の要素である u, v が存在し、エッジ $e = (u, v) \in F$ が存在する時、 u は v を内包すると呼ぶ。またこの時の、 u に内包される v の集合を u の子ノードと呼び、子ノードの集合を $Ch(u)$ とする。逆に、 u を親ノードと呼び、 $pa(v)$ とする。

階層構造のルートノードを v_{root} と呼ぶ。また、 v_{root} からノード $u \in V$ に至るまでのエッジの個数を階層構造の深さと呼び、 $dep(v)$ で表す。

ノード $v \in V$ として、 v が Web ページの時は、 $Ch(v) = \phi$ であり、 v がページを階層的にまとめた概念である場合は、 $Ch(v) \neq \phi$ である。

ノード $v \in V$ が子ノードを持たない、すなわちページである時はそのページへのアクセス数をノードの重みとして $w(v)$ と書く。ノード v が子ノードを持つときは子ノードの重みの和を、重みとして同様に $w(v)$ とする。

1.4.2 ページ遷移の有向グラフ

各階層ごとの、ユーザのページ遷移による有向グラフを考える。先ほどと同様に、ページやページをまとめた概念をノードとして、その集合を V 、ユーザのページ遷移を表す概念であるエッジの集合を E とすると、ページ遷移による有向グラフ D_b は

$$D_b = (V, E)$$

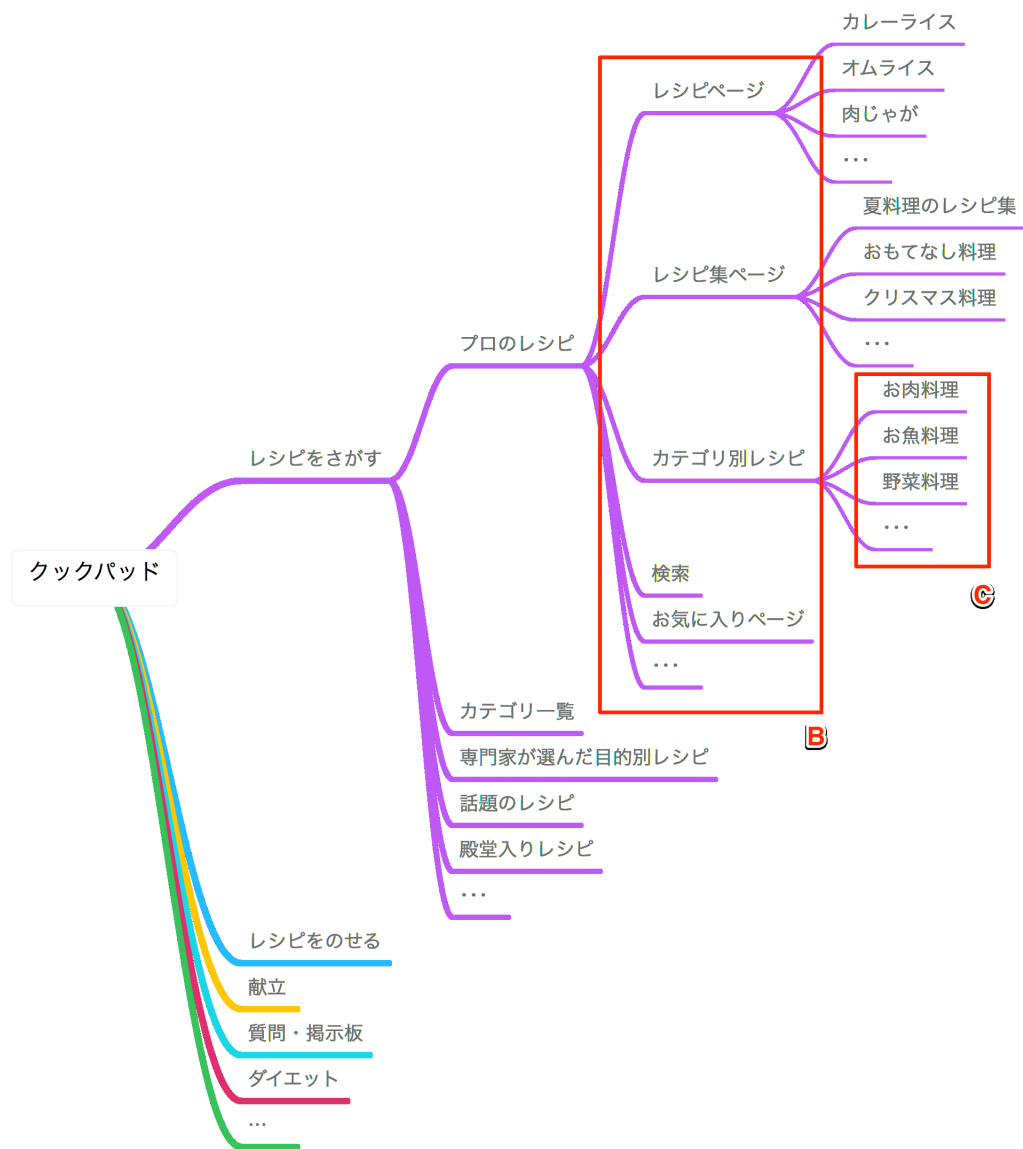


図 1.2: クックパッドのサイトの階層構造

と表現で出来て、 V の要素である u, v が存在し、エッジ $e = (u, v) \in E$ が存在する時、 u と v には隣接関係があり、 u から v へのリンク関係があり、ユーザのノード間の遷移があると
する。

エッジ $e \in E$ は、そのエッジが表現するユーザのノード間の遷移の量を重みとして、 $w(e)$ と書く。

1.4.3 Web サイトの階層構造とページ遷移の混合グラフ

本論文で対象とするデータの構造は、1.4.1 で述べた Web サイトの階層構造のグラフと、1.4.2 で述べたページ遷移の有向グラフの混合グラフであり次のように書ける。

$$D = (V, E, F)$$

また、これを図示したものを 1.3 に示す。この図中の丸がノード集合 V の要素であり、青線が階層構造を表すエッジ集合 F の要素であり、赤線がユーザの遷移を表すエッジ集合 E の要素である。

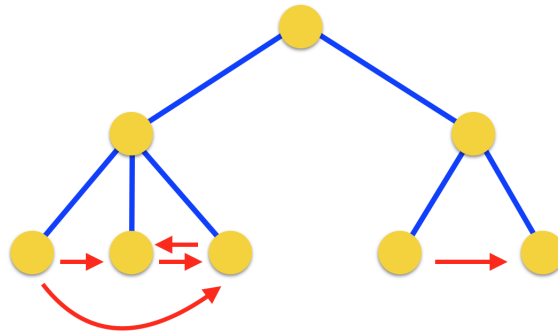


図 1.3: Web サイトの階層構造とユーザの遷移の混合グラフ

1.5 Web サイトの階層的な分析

大規模な Web サイトのユーザの行動を分析する方法として、Web サイトの階層構造の上位の階層や下位の階層を、視点を自由に移動しながら分析する方法がある。本論文では、この視点を階層の上位や下位に移動しながら Web サイト全体を分析することを階層的な分析と呼び、階層構造を持つ大規模な Web サイト上におけるユーザの行動を把握するために、この階層的な分析を行う可視化手法を開発する。

1.6 本論文の貢献

本論文の貢献は以下の三点である。

1. 大規模な Web サイト上をユーザがどのように行動しているかを探索的に分析するために、矩形の入れ子構造による階層の表現と連結図によるグラフ表現を組み合わせで応用した新たな可視化手法を開発した。Web サイトの階層構造を矩形の入れ子構造で表現し、ユーザの行動は連結図で表現することで、探索的な分析を支援した。
2. 開発した可視化手法を実装した。特に、実際に運用されている Web サイトを対象とした分析が行えるように、実装を行った。
3. 実装したプログラムを用いて、実際に運用されている Web サイトのアクセスログデータを用いてケーススタディを行った。本論文で開発した可視化手法により階層的な分析が出来て、サイト全体でのユーザ行動の把握に役立つことを確かめた。

第2章 関連研究

2.1 階層構造を持つグラフの可視化に関する関連研究

グラフを可視化する研究は活発に研究されており、Herman らは多くのグラフの可視化手法を体系的にまとめた [7]。その中でも、Sugiyama らは構造的な情報を混合有向グラフの自動描画することで可視化する手法を開発した [22]。また、Frishman らは、クラスタ化された複数のグラフを自動配置する手法を開発した [6]。これは、各クラスタ内とクラスタ外のそれぞれにおいて、力指向レイアウトを用いてノードを自動的に配置する。Sugiyama ら、Frishman らの研究は共に階層構造を持つようなグラフを自動描画することに主眼が置かれた研究である。また、特にエッジの表現にも注目すると、階層的なグラフの描画で特にエッジの隣接関係も表現する手法を Holten は開発した [8]。

2.2 Web のリンク構造の可視化に関する研究

Web のリンク構造を可視化する研究も以前より活発に行われてきた。Turetken らは Web のリンク構造の可視化手法をまとめた [24]。本論文でも用いた連結図を用いる手法でリンク構造を可視化したものには、Pitknow らの手法がある [19]。Pitknow らは各 Web ページとリンクされているページ、及びそれらのページに参照されている画像ファイルなどのリソースファイルのリンク構造を、連結図を用いたグラフの視覚的表現手法を用いて可視化する手法を開発した。Pitknow らの手法では、対象となる Web サイトのページ数は高々数十の規模を想定している。また、WWW の階層を持つようなリンク構造を可視化したものとして、Munzner らによるリンク構造を 3 次元の双曲線空間上に表現する手法がある [15]。Munzner らの手法は、大きな規模の Web サイトのリンク構造のみを可視化することを目的としていて、実際に数百の規模のページを持つような WWW のリンク構造を可視化している。他にも、複雑なハイパーメディアの構造を可視化のための研究として、Freire らは、複雑なハイパーメディアをどのようにフィルタリングしつつ視覚表現していくかということを、連結図をベースとしてインタフェースを含めた可視化手法のフレームワークとして提案した [5]。Lai らは、WWW の複雑なリンク構造を可視化するために、各ページ間の隣接関係でページをクラスタリングして、そのクラスタリングされた結果と共にリンク関係を連結図を用いて視覚的に表現、また、重要でないようなものはフィルタリングするインタラクションも含めた可視化手法を開発した [14]。リンク構造から自動的にクラスタリングして可視化するものには、他にも Durand らの開発した手法もある [2]。Durand らの手法は、大規模な Web サイトをクロールして

ページを収集し、各ページをクラスタリングした結果を可視化することに主眼がおかれ、リンク関係はクラスタリングの課程では考慮されるが、可視化した表現上は、重要なリンクのみしか可視化しない。Fluit らもまたクラスタリングを行って複数の Web ページを可視化する手法を開発した [4]。Fluit らは、リンク構造を用いたクラスタリングではなく、セマンティックな Web の意味的に近いページをクラスタリングして、その結果を可視化した。階層的な Web サイトの全体を可視化しようとしたものには、Nation らの手法がある [16]。Nation らの手法は、インタラクティブに操作することを含めたもので、テーブルレイアウトで Web サイトの階層がまとめられたものをクリックして展開していくような手法で、例えば連結図のような視覚的な表現手法は用いていない。Web サイトの階層構造を視覚的に表現するために、山口らは矩形の入れ子によって Web サイトの階層構造を表現する手法を開発した [25]。これは本論文で用いた階層構造の表現方法に非常に近い。山口らの手法は、矩形の入れ子を如何にして配置していくかという点に主眼が置かれている。

2.3 ユーザの行動分析に関する関連研究

Web サイト上におけるユーザ行動の分析には Google Analytics¹ が非常によく使われている。Google Analytics は基本的には各ページごとであったりサイト全体のアクセスの傾向を Bar chart で観察するような使われ方が多いが、ユーザ行動の分析も見ることが出来て、Sankey diagram を用いた視覚的表現によってユーザのページ遷移の行動を観察することが出来る。Sankey diagram を用いた手法は、サイトへの流入からサイトからの離脱までの流れを俯瞰的に観察しやすいが、一方向レイアウトであるために、同一のノードが二度出現したり、また、そのために画面全体に描画されるノードの数が重複も含み多くなってしまうので、重複をのぞいたノードは少数しか表現出来ない。これを改良するために、Sankey diagram をインタラクティブに操作可能とし、ズーム操作等を導入することでより多くのノード、より複雑な経路を可視化する手法を Riehmann らは開発した [20]。

Kawamoto らは、先の述べた山口らによる矩形の入れ子構造を用いた Web サイトの階層構造を表現しながら、ユーザの行動を分析する手法を開発した [12]。Kawamoto らの手法は、本研究に近い目的であるが、Kawamoto らは各ページのリンク構造やユーザの遷移を全て見せるような方法ではなく、アクセスパターンをアクセスログから機械的に導出して、その結果のみを可視化する。Kawamoto らの手法の他にも、アクセスパターンを機械的に導出した上で分析するものとしては、機械学習によってアクセスパターンを解析する Zaiane らの手法 [26] や、生物学的な挙動を模してアクセスパターンを解析する Labroche らの手法がある [13]。本研究は、予めユーザの行動の仮説を立てた上で機械学習や様々な手法で仮説検証するものではなく、極力全てのページ遷移を可視化してそれを分析者が観察して探索的に分析が出来ることを目的としている。

Web サイト上のユーザの行動のように人の流れの可視化に着目した研究で、出来るだけ全ての情報を可視化した上での探索的な分析を試みた研究としては、階層的な構造を持って人

¹<https://www.google.com/analytics/>

の行動のような流れのあるデータを可視化する手法を Phan らは開発した [18]。これは、例えば地図のような図形の上に流れを描画する。

2.4 本論文の位置づけ

階層構造を持つ
グラフの可視化

Webのリンク
構造の可視化

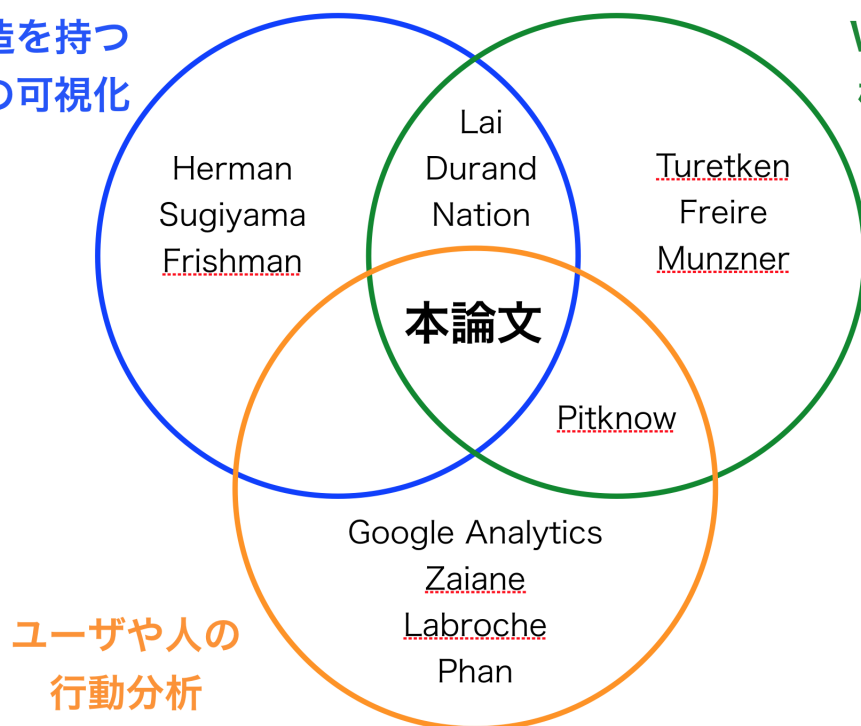


図 2.1: 本論文の位置づけ

本論文の位置づけを図 2.1 に示す。本論文は、階層構造を持つグラフの可視化を応用して Web の階層構造を矩形の入れ子により表現し、更に、グラフのエッジで重みの表現も加えることで、ユーザの行動の可視化を行う。Kawamoto らの研究も位置づけとしては本研究とほぼ同じであるが、Kawamoto らの研究ではアクセスパターンを機械的に導出した上で視覚的に表現を行い、一方で本研究は出来るだけ全ての情報を分析者に視覚的に表現し、分析者が探索的に分析を行うことを目的としている点が異なる。

第3章 要件の検討

3.1 分析者への要望のヒアリング

視覚的表現を検討するにあたり、筆者は予め Web サイトの運営において主にユーザの行動分析を業務として行う分析担当者一名と、同じ Web サイトの分析に携わりながら主に開発を担当するエンジニア一名にヒアリングを行った。ヒアリングにより次のような要望があることが分かった。

一人一人の遷移では無く、ユーザ全体の遷移の把握が出来ること

ユーザ一人一人の遷移を詳細に分析するよりは、まずはユーザ全体の傾向としてどのように遷移しているかということを知ることが出来るとよい。小数のユーザで絞って表示するより、とにかく全ての遷移が見えているような状態が理想であるとのことであった。

ユーザ属性ごとの分析では無く、全体での把握が出来ること

Web サイトにアクセスするユーザには、性別や年齢といった様々な属性情報がある。本論文では、最初はこの属性情報ごとに分析が行えるような視覚表現を検討していた。しかし、ヒアリングを行ってみると、分析者は実際にはユーザ属性ごとに最初から切り分けて分析をおこなうことは望んでいないことが分かった。理由としては、分析を主に行う人々の間の経験則として、最初から属性ごとに分けて分析を行うこともかつてはよく行っていたが、各属性ごとに分析を行っても属性ごとに大きな差があることは非常に希で効率が良くないので、現在では属性ごとに最初から分けて分析を行うことはしないということであった。そのため、属性ごとに分けるのでは無く、あくまでも全体として分析が行いたいということであった。

サイト全体の分析と、各階層ごとの分析が行えること

現在では、例えば検索結果のページからどこへの遷移が多いか、その後はどこへの遷移が多いかのように、ある特定のページに着目した分析を行うことが多い。しかしながら、Web サイトのページ数は非常に多く、もっとサイト全体を俯瞰した分析も行いたいという要望があった。一方で、各ページを階層ごとに分けた時の、各階層を詳細に分析出来ることも必要であり、各階層の分析をシームレスに行いたいという要望であった。

遷移するユーザ数が多い場所以外も分析出来ること

Web サイトにおける各ページ間を遷移するユーザ数はべき乗則に従うことが知られている。特にアクセスが多いページがどこかというようなことは、実際の業務においては既

に知られていることが多く、ヒアリングに応じてくれた分析者も、最もアクセスが多いページとそこからどのページによくユーザが遷移するかということは既知なことが多いと回答した。既に最もアクセスが多いページについては既知なことが多いので、それよりも、サイト全体と比較して例えば中程度のアクセスしか無いけれども、ユーザの多くが、分析者がまだ知らないページ遷移をしていた、というようなことを発見したいとのことであった

3.2 階層構造の出来るだけ多くのノードの表示

全体を俯瞰して分析する事を可能とするために、階層構造を構成する全てのノードを一つの画面上に表現することが要件となる。

3.3 階層構造における各階層の分析

全体の分析と各ページごとの分析をしたいという要望は、例えば図 1.2 中の B で指し示した「プロのレシピ」の下に連なる下層のノード間の流量を観察することが出来て、かつその状態から図中の C で指し示した「カテゴリ」以下の各ノード間の流量も観察することが出来れば満たすことが出来る。すなわち、Web サイトの階層構造における各階層でのノード間の流量が観察出来て、更にその子階層や親階層での流量も観察出来る事が要件となる。

3.4 全ての流量の表示

流量が多いノード間のリンクだけでは無く、より流量が少ない状態も観察出来る事が必要である。すなわち、予めデータの加工で例えば遷移の流量が少ない部分はフィルタリングするといったことをせずに、量が少ないノード間の遷移であっても、観察出来るようにすることが要件となる。

第4章 開発した可視化手法

本章では、検討した要件を踏まえて開発した可視化手法について説明する。

4.1 階層構造の表現

階層構造の可視化手法の一つとして、Tree-map[11]がある。Tree-mapは画面上の領域を再帰的に分割していくことで階層構造を視覚的に表現する。見た目が似た手法として、矩形の入れ子によって階層構造を表現する、平安京ビュー[10]やデータ宝石箱[25]がある。矩形の入れ子により階層構造を表現することは、一つの画面上でより多くのノードを表現することが出来るので、一つの画面上に出来る限り多くのノードを表現するという要件を満たすことが出来る。Tree-mapや平安京ビューを参考に、図1.2のクックパッドのサイトの階層構造を簡易的に矩形の入れ子により表現したものを、図4.1に示す。このような矩形の入れ子による表現で、階層構造を表現することとした。

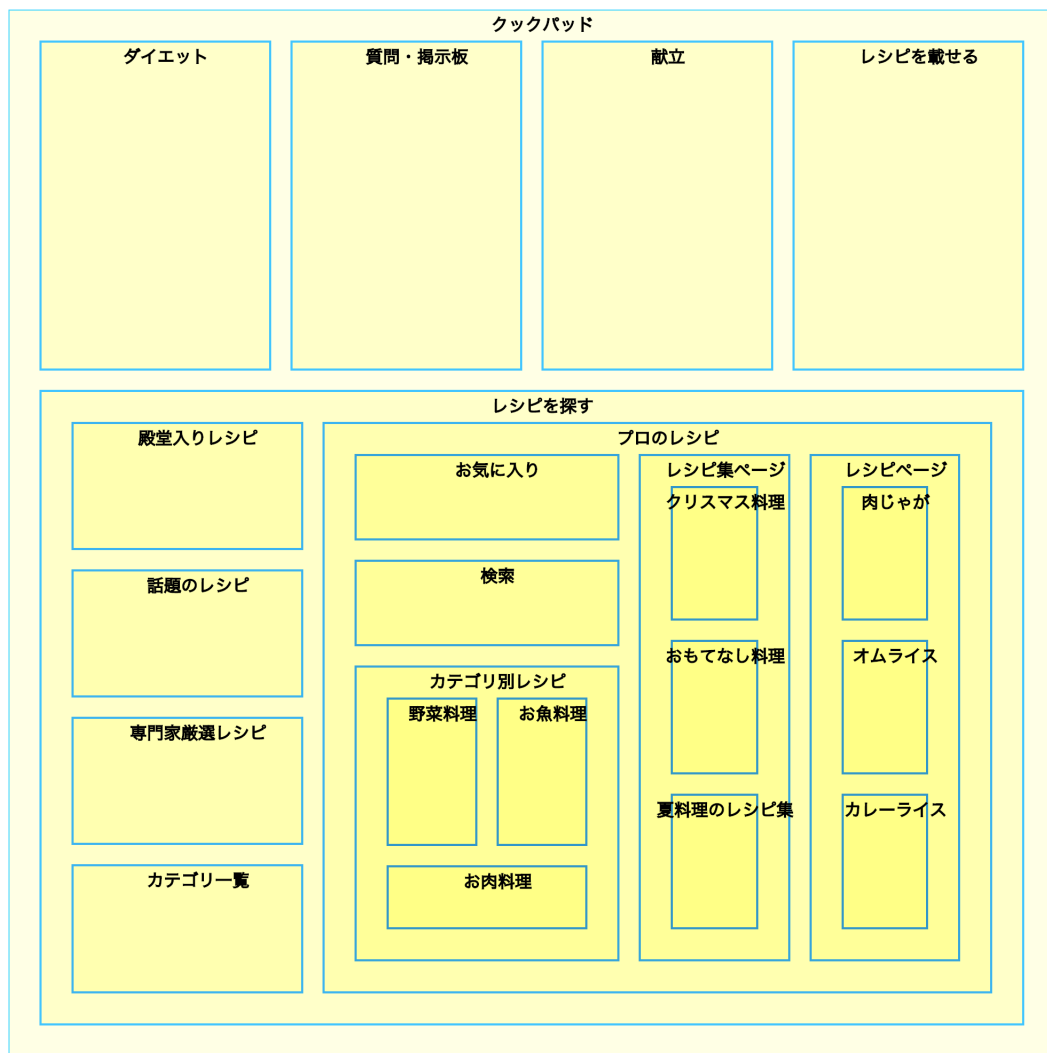


図 4.1: サイトの階層構造の矩形の入れ子による表現

4.2 各階層ごとの遷移の表現

ノード $u \in V$ があり $Ch(u) \neq \phi$ の時、ノード u が内包する階層の、ユーザのノード間の遷移を表すグラフ D_b^u は次のようになる。

$$D_b^u = (V^u, E^u) \quad (4.1)$$

$$V^u = Ch(u) \cup \{v_{outgoing}^u, v_{incoming}^u\} \quad (4.2)$$

$$E^u = \{(u, w) \in E | u \in V^u, w \in V^u\} \cup \{(v_{incoming}^u, x), (x, v_{outgoing}^u) | x \in V^u\} \quad (4.3)$$

$v_{outgoing}^u$ 及び、 $v_{incoming}^u$ は outgoing ノード、incoming ノードと呼ぶ特殊なノードで、後の 4.2.2 で詳しく述べる。

階層ごとに、このグラフ D_b^u は連結図を用いて表現する。

ある $A \in V$ というノードが、 $a_1 \in Ch(A)$ と $a_2 \in Ch(A)$ というノードを内包している時の描画例を図 4.2 に示す。本節では、この図を描画するためのノードとエッジ、そしてノードの配置の説明を行う。

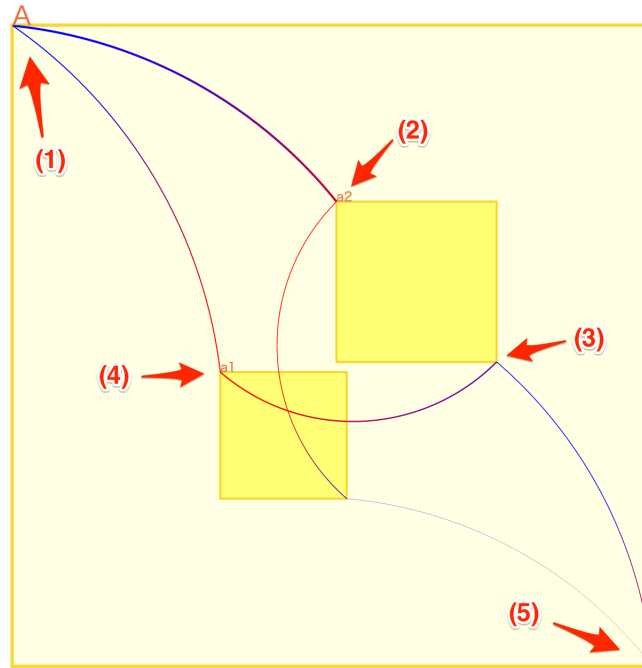


図 4.2: A というノードの階層での連結図によるグラフ表現の例

4.2.1 ノードの表現

$u \in V$ に内包されるグラフ D_b^u におけるノードは、式 4.2 のように定式化できる。グラフ内で、ノード $v \in V^u$ は正方形で表現する。そして、ノードの重み $w(v)$ は正方形の面積で表現する。ここで、グラフ D_b^u におけるノードの重みの集合を W_{vertex}^u とすると、これは次のように表現できる。

$$W_{vertex}^u = \{w(v) | v \in V^u\}$$

そして、この集合の要素のうち最大のものを $\max(W_{vertex}^u)$ とすると、ノード $v \in V^u$ を表現する正方形の一辺の長さ $l(v)$ を

$$l(v) = \alpha^u \cdot \sqrt{\frac{w(v)}{\max(W_{vertex}^u)}} \quad (4.4)$$

$$\alpha^u = \frac{l(u)}{\text{count}(Ch(u))} \quad (4.5)$$

によって定めた。なお、 $\text{count}(Ch(u))$ は集合 $Ch(u)$ に属する要素の個数、すなわちノード u の子ノードの個数とする。

ノード u を表現する正方形の中に、極力全ての子ノード $v \in V^u$ が描画出来るようにするために、式 4.5 のように α^u を定め、また、式 4.4 の後半部分によって、ノードの重みの違いが面積の違いに視覚的に割り当てられ、分析者がノードの面積を比較することでノードの重みを比較出来るようにした。

4.2.2 incoming ノードと outgoing ノード

各階層には、incoming ノードと outgoing ノードと呼ぶ、特殊なノードを一つずつ設置する。これは、上位の階層からの流入と、上位の階層への流出をそれぞれ集約するノードであり、それぞれ $v_{incoming}^u$ 、 $v_{outgoing}^u$ と表す。 $v_{incoming}^u$ は各階層の左上角、図 4.2 上の (1) で示す場所に置き、 $v_{outgoing}^u$ は図中の (5) で示す場所に置く。各階層内のノードへの上位の階層からの流入は $v_{incoming}^u$ を始点として各ノードが終点となるエッジで表現し、各階層内のノードから上位の階層への流出は各ノードを始点として $v_{outgoing}^u$ を終点とするエッジで表現する。すなわち、図 4.2 中の (2)(3) で指したノードへの上位の階層からの流入は、(1) から (2) のエッジで表現し、逆に上位の階層への流出は (3) から (5) のエッジで表現する。

4.2.3 エッジの表現

$u \in V$ に内包されるグラフ D_b^u におけるエッジ $e \in E^u$ は、図 4.2 中の (1) から (2) の曲線、(3) から (4) への曲線のように視覚的に表現する。エッジ $e = (v, w) \in E^u$ とすると、エッジの始点はノード v を表現する正方形の右下の角、終点はノード w の左上の角になるようにした。つまり、図 4.2 中の (3) と (4) を結ぶエッジは、(3) から (4) への方向を持つエッジである。同様に、(1) と (2) を結ぶエッジは、(1) から (2) へと方向がある。

エッジの方向がより視覚的に分かりやすいように、始点から終点にかけて、赤色から青色へとグラデーションを付けた。また、より方向をわかりやすくするため、右回りの弧によって線を描画する。Holten らは、先が細くなるテーパー表現が最も有向グラフのエッジ表現には適切であることを示した [9] が、本研究では、次の二点の理由により弧による表現とグラデーションによる表現の組み合わせを用いた。一つ目の理由は、次に述べるようにエッジの幅の大きさを重みを表現するが、テーパー表現は先が細くなる表現手法で、エッジ全体で幅が一定では無いので、幅を用いた重みの表現とは相性が悪いと考えた。また二つ目の理由として、本手法では重みが小さいものは 1px といった描画出来る最小の幅でエッジを描画する場合もあり、この場合はテーパー表現を用いることが出来ない。以上の理由により、エッジの表現はテーパー表現を避け、また、視覚的な混雑を避けるためにも矢印のような表現もさせて、弧による表現を用いた。

また、エッジの重み $w(e)$ はエッジの幅によって表現した。ここで、グラフ D_b^u におけるエッジの重みの集合を W_{edge}^u とすると、これは次のように表現できる。

$$W_{edge}^u = \{w(e) | e \in E^u\}$$

そして、この集合の要素のうち最大のものを $\max(W_{edge}^u)$ 、最小のものを $\min(W_{edge}^u)$ とすると、エッジ $e \in E^u$ を表現する弧の幅の大きさ $d(e)$ を

$$d(e) = (d_{max} - d_{min}) \cdot \frac{w(e)}{\max(W_{edge}^u) - \min(W_{edge}^u)}$$

によって定めた。

d_{max} 及び d_{min} は、共に定数で、それぞれ最小のエッジ幅と最大のエッジ幅であり、特に要件にある全てのエッジを表示するためには、 $d_{min} > 0$ とする。一方で、 d_{max} が大きすぎると、画面領域がエッジで埋まってしまう。この値は、対象とする Web サイトのノード数や深さに合わせて、実装に依存して定めることとする。

4.2.4 ノードの配置

自動的な配置

画面全体で大量のノードを配置する必要があるため、ノードは自動配置を行う。グラフ D_b^u のノードの自動的な配置には、ノードが一様に配置させやすい等の美的基準に優れる [23] 力指向レイアウトを用いる。

各ノード間には、エッジをバネとした力と F_{spring} と、ノードの重なりを少なくするために反発力 F_{charge} 、更にはより描画領域の中心付近にノードが集まるようにするために、描画領域の中心にむけて重力 $F_{gravity}$ を与えた。これらの力による位置の計算は、Dwyer による手法 [3] を用いた。

まず、バネの力は次の式で与えられる。

$$F_{spring} = k \cdot L$$

k はバネ定数、 L は自然長であり、この L は全てのエッジで同一の定数とした。また、 k はエッジの重みに比例するようにした。すなわち、エッジ $e \in E^u$ のバネ定数の大きさ k_e は、

$$k_e = \frac{w(e)}{\max(W_{edge}^u) - \min(W_{edge}^u)}$$

によって与える。 $\max(W_{edge}^u)$ 及び、 $\min(W_{edge}^u)$ は式 4.2.3 によるものである。

ノード間の反発力 F_{charge} は、ノードの重みを用いた。即ち、ノード $v, w \in V^u$ に隣接関係があり、この二つのノード間の反発力は

$$F_{charge} = \frac{w(v) \cdot w(w)}{distance(v, w)}$$

とした。ここで、 $distance(v, w)$ は二つのノードの描画上のユークリッド距離である。

描画領域の中心にむけた重力 $F_{gravity}$ は、物理学的に正確な意味での重力、すなわちノード間各々に働く力では無く、みせかけの重力として、領域の中心にむかっての引力である。これは、領域外にノードが描画されることを防ぐために、用いた。また、この値も全てのノードで同一の値で固定した。すなわち、

$$F_{gravity} = Const$$

である。

以上の $F = F_{spring} + F_{charge} + F_{gravity}$ によって、ノードの配置が自動的に定まる。

全てのノード同士にエッジがあつてかつ、ノードの重みが等しく、エッジの重みも全て等しいときの例を図 4.3 に示す。この時は全てのノードにおいて互いに働く F_{spring} と F_{charge} が等しいのでバランスが取れて、かつ $F_{gravity}$ によって中央に集まった形になる。

この図の状態を変えて、エッジのみにランダムに重みを持たせて、ノードの重みは均一なときの例を図 4.4 に示す。図中の (1) で指し示すノードに着目すると、(2) で指し示す incoming ノードからのエッジよりも (3) で指し示す outgoing ノードへのエッジのほうが重みがあるのでバネの力 F_{spring} が大きくなり、ノードの位置は outgoing ノード側に引き寄せられる。

次は、最初の図 4.4 を変えて、ノードのみにランダムに重みを持たせて、エッジの重みは均一にしたときの例を図 4.5 に示す。図中の (1) で囲んだ二つのノードは、双方とも重みが小さいので互いの反発力 F_{charge} が小さくなり近づく。一方で (2) で囲んだ方は、一方の重みが大きいので距離が広がる。

手作業による配置の可能性

一方で、ノードを手作業で配置出来る可能性についても特筆する。全ての階層全体で例えば 125 個のページがあったとしても、3 階層に分けてかつ各階層で均等にノードを持つように

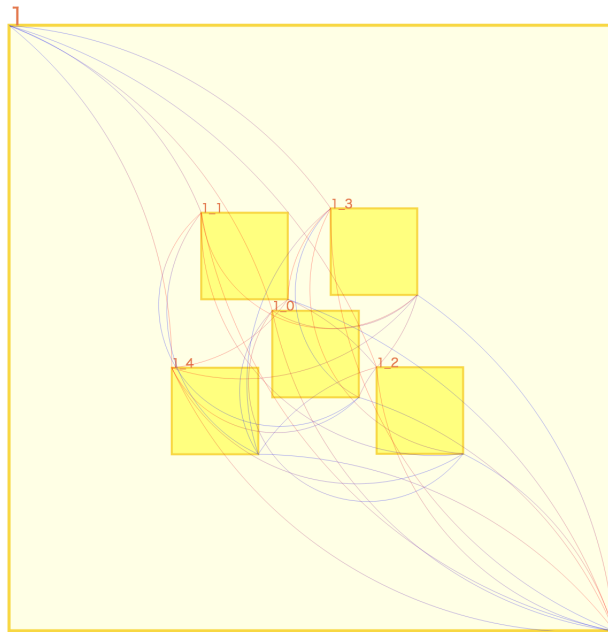


図 4.3: 全てのノード間にエッジがあり、かつノードやエッジの重みが等しい時の配置例

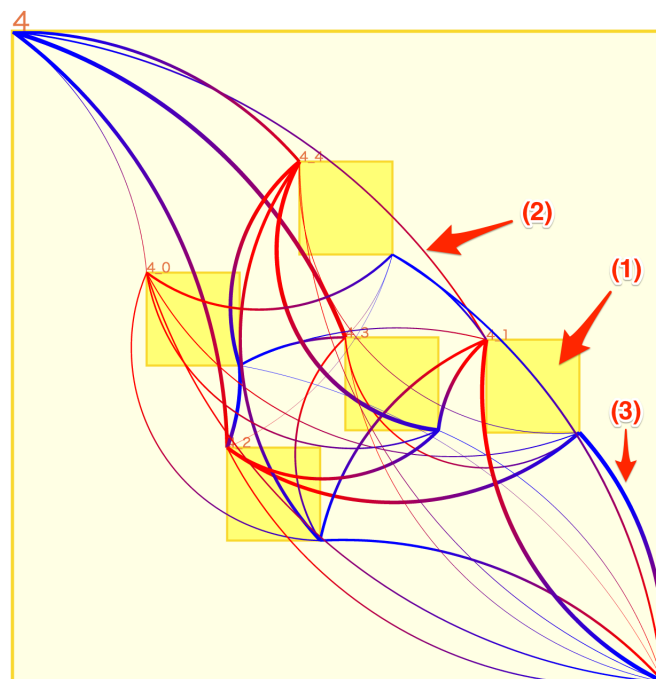


図 4.4: 全てのノード間にエッジがあり、エッジの重みのみを変えた時の配置例

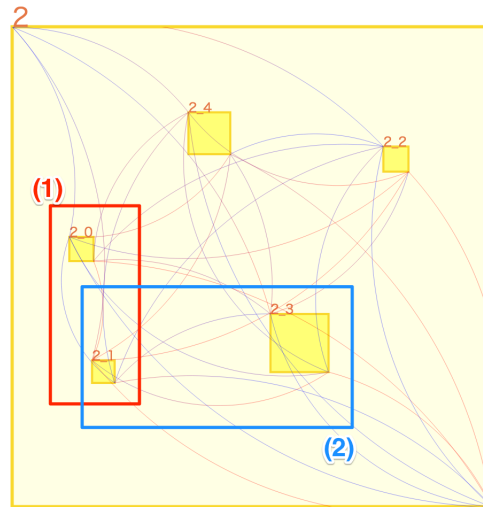


図 4.5: 全てのノード間にエッジがあり、ノードの重みを変えた時の配置例

うまく階層分けをすることが出来れば、 $\sqrt[3]{125} = 5$ であるので、各階層には5ノードだけにする事が出来る。各階層に着目するとき、5ノードだけであれば、十分に手作業での配置も可能であると考えられる。図 4.2 は手作業で二つのノードの配置を決定した。

4.3 階層構造とユーザの遷移の混合グラフ全体での表現

式 1.4.1 で示した階層構造のグラフ D_h を矩形の入れ子構造によって視覚的に表現し、また、式 1.4.2 ユーザの遷移の有向グラフ D_b を連結図によって視覚的に表現したものを組み合わせた式 1.4.3 で示した混合グラフ全体 $D = (D_h, D_b) = (V, E, F)$ を描画した例を、図 4.6 に示す。この例の図では、ノードの重みは均一にして、エッジの重みをランダムに設定した。

全体の観察と各階層の観察のためのインタラクション

図 4.6 のように全体を観察した後の、各階層をよく観察するためにはズームによるインタラクションを用いる。図 4.6 中の中心付近にあるノードにズームした図が、図 4.7 である。このようにズームインタラクションを開発した手法に取り込み、子階層の詳細な観察も行えるようにした。

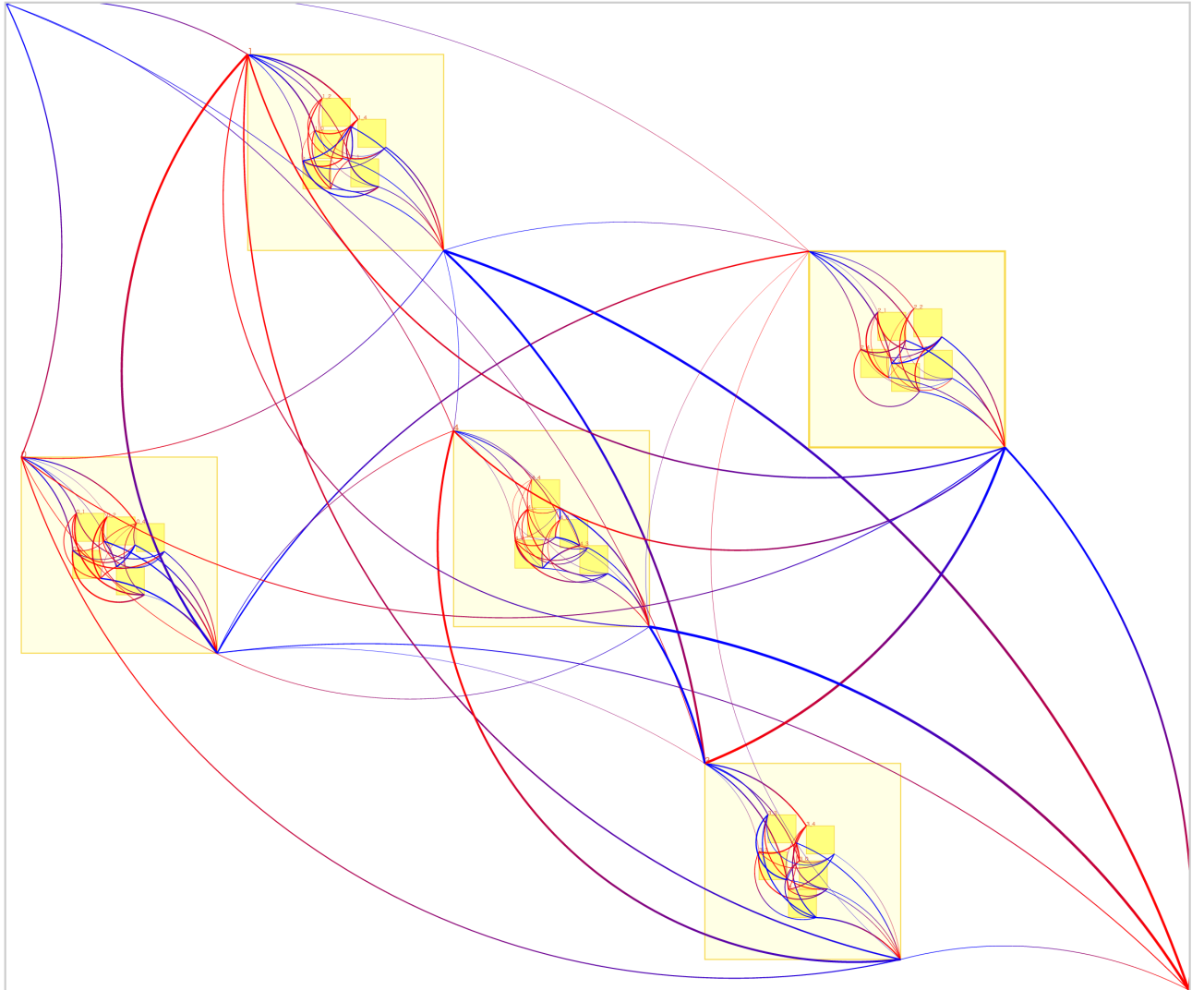


図 4.6: 複数の階層を組み合わせた全体の例

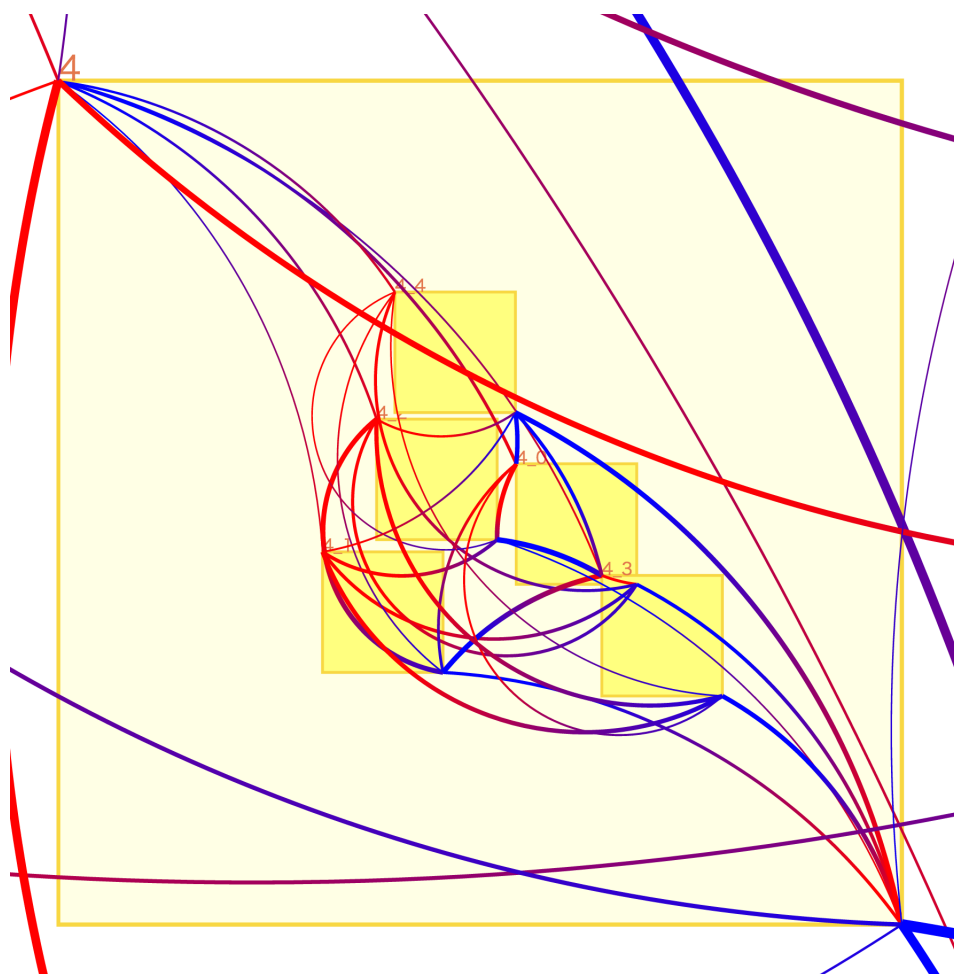


図 4.7: 図 4.6 の中心付近をズームしたもの

第5章 可視化手法の実装

筆者は、開発した可視化手法を実装した。本章では実装の詳細を述べる。

5.1 対象データ

本論文では、レシピ閲覧サイトである「クックパッド¹」の、特に「プロのレシピ²」を分析対象の Web サイトとして実装を行った。このサイトの階層構造は図 1.2 中の「プロのレシピ」以下の階層により構成される。

Web サーバのアクセスログを対象データとする。アクセスログは、ユーザが Web サーバにアクセスする度に一行ずつ増えていく。データの例を表 5.1 に示す。一つのデータには、時刻である *time* があり、

アクセス先は (*controller*, *action*, *resourceID*) の組で表現される。対象とする Web サイトは、Ruby on Rails³ によって構築されており、*controller* と *action* は、それぞれ Ruby on Rails の MVC フレームワーク上の、Controller 及びその内部のアクションを指す。また、Ruby on Rails は REST の思想により設計されているが、その REST 上でのアクションの対象となるリソースの ID が *resourceID* である。

referer はそのアクセスの前にどのページに居たか、即ちページ遷移の遷移元のページで、Web サイトの階層構造では末端のリーフに位置する Web ページである。

表 5.1: ログデータの例

time	controller	action	resource ID	referer
2014-12-16 16:14:08	pro_recipe_service/pro_recipes	show	2754883	http://cookpad.com/pro/recipe/2754883
2014-12-16 16:14:12	pro_recipe_service/search	show	nil	http://cookpad.com/pro/
2014-12-16 16:14:20	pro_recipe_service/pages	top	nil	http://cookpad.com/
2014-12-16 16:14:25	pro_recipe_service/pro_recipes	show	2762210	http://cookpad.com/category/166

本実装では、*controller* を一階層目、*action* を二階層目として実装した。これは、*controller* による区分が、ユーザがアクセスするページの区分と非常に近い設計となっているからである。*controller* を一階層目、*action* を二階層目とした階層構造を図 5.1 に示す。

¹<http://cookpad.com/>

²<http://cookpad.com/pro>

³<http://rubyonrails.org/>

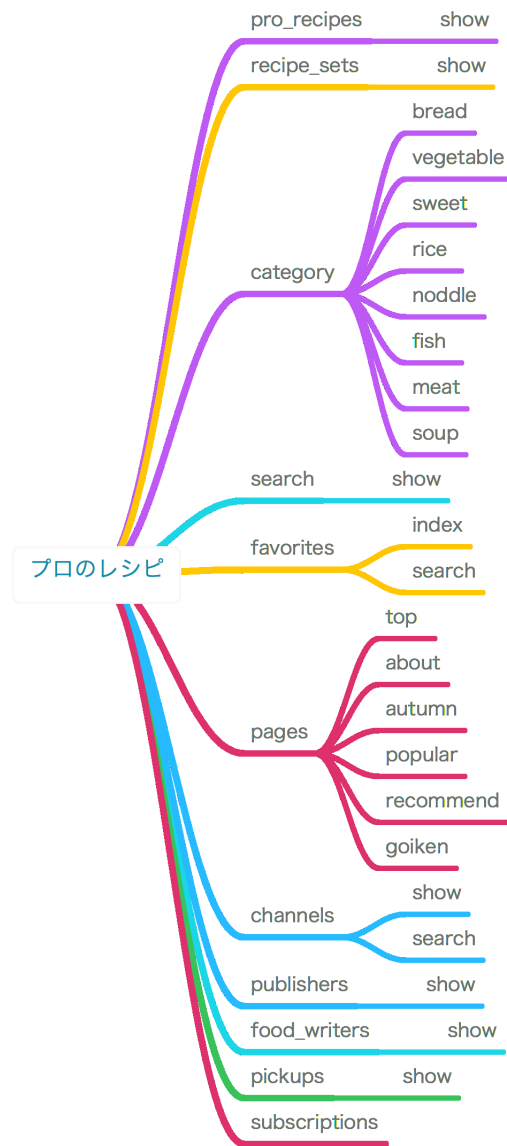


図 5.1: *controller* を一階層目、*action* を二階層目とした階層構造

5.2 処理の流れ

可視化までの処理は、先ずログデータの収集があり、それを集計し、最後に視覚的表現に変換するという流れになる。本節では、このログデータの収集から可視化までの全体の大まかな流れを説明する。

5.2.1 データの収集

対象のサイトへの全てのアクセスのログは、全ての Web サーバから Fluentd⁴ というログ収集のためのフレームワークを通して、ビッグデータを取扱うための SaaS である TreasureData⁵ に保存される。ここから、HQL という SQL ライクなクエリ言語を用いて TreasureData の Hive⁶ 上で実行されて、保存されたログデータを取り出す。Hive 上では Map/Reduce モデルで HQL が実行されるため、クエリの実行から結果を受け取るまで、少なくとも秒単位の時間がかかってしまう。

そのため、予め数日分のデータを TreasureData から HQL から取り出し、MySQL⁷ 上のデータベースに保存した。この時に、*referer* をアクセス元の *controller* と *action*、*resourceID* に変換し、取扱易くした。すなわち、表 5.1 のリファラ列を表 5.2 のように変換する。

表 5.2: 5.1 の *referer* の変換

controller	action	resource ID
pro_recipe_service/pro_recipes	show	2754883
pro_recipe_service/pages	top	nil
pro_recipe_service/pages	top	nil
category	show	166

5.2.2 データの集計

データの収集以降の、可視化までの処理の流れを図 5.2 に示す。

扱いやすく加工した上でデータベースに保存したログデータは、SQL によってデータベースサーバ上でどのノードからどのノードへどれだけの遷移の量があったかをカウントし、その結果を更に Ruby⁸ によって書かれたプログラムによって階層構造にまとめて集計し、混合グラフの構造のデータにする。混合グラフの構造にまとめたノードとエッジの情報は JSON 形式に整形する。そして、これを Ruby で書かれた Web アプリケーションを通して、可視化の描画に用いたクライアントである、Web ブラウザへと送られる。なお、Web アプリケーショ

⁴<http://www.fluentd.org/>

⁵<http://www.treasuredata.com/>

⁶<http://hive.apache.org/>

⁷<http://www.mysql.com/>

⁸<https://www.ruby-lang.org/>

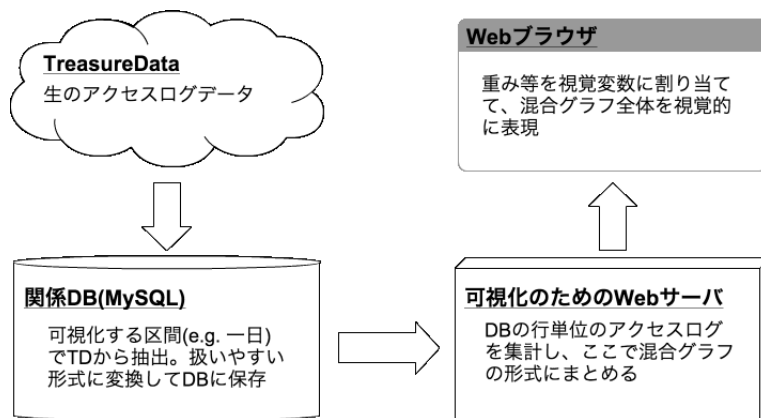


図 5.2: 可視化までの処理の流れ

ンは、フレームワークとして Sinatra⁹ を、データベースのログデータに対する O/R マッピングには ActiveRecord を用いた¹⁰。

5.2.3 データの可視化

可視化の描画処理は Web ブラウザを利用する。描画処理を行うプログラムは CoffeeScript¹¹ で作成し、JavaScript に変換した後にブラウザ上で実行する。Web ブラウザには、特に GoogleChrome¹² を利用した。

また、データの視覚変数への割り当てやレイアウトを行うためのフレームワークとして、D3.js¹³ を用いた。これは、D3.js が視覚変数の割り当ての試行錯誤を素早く行い、視覚表現を検討するのに適している [1] ので、これを利用することにした。

5.3 可視化の実装

本節では、Web ブラウザ上での可視化のための描画処理について述べる。

5.3.1 視覚変数への割り当て

視覚変数への割り当ては、ノードは 4.2.1 で説明した方法で、エッジは 4.2.3 で説明した方法でそれぞれの重みを視覚変数へと割り当てた。また、更に重みはノードの配置の位置にも割り当てられていて、4.2.4 で説明した手法によりノードを配置した。特に、力指向レイアウト

⁹<http://www.sinatrarb.com/>

¹⁰<https://github.com/rails/rails/tree/master/activerecord/>

¹¹<http://coffeescript.org/>

¹²<https://www.google.co.jp/chrome/browser>

¹³<https://github.com/mbostock/d3>

トに関しては、各力 (F_{spring} , F_{charge} , $F_{gravity}$) についてはそれぞれの力の大きさの計算をノードに対して計算し、それを D3.js に予め用意されている力指向レイアウトの枠組みにあてはめることで配置を行った。

5.3.2 実装したインタラクション

ズームとパン

4.3 で述べた、階層全体の俯瞰的観察とある階層に着目して観察を行うために、ズーム操作とパン操作ができるようにインタラクションを実装した。このズーム操作とパン操作は、マウスのスクロール操作の上下にそれぞれ割り当てた。このような割り当ては、先に GoogleMap¹⁴ をはじめとして、いくつかの先行事例があるので、ユーザのメンタルモデルにも適合していると考えられる。

全体表示領域の移動

描画領域全体の上下左右の移動は、キーボード上の矢印キーの上下左右にそれぞれ割り当てた。マウスによるドラッグ操作への割り当てを行っている例があるが、本手法においては、後述のノード配置のためのノードの移動操作にマウスのドラッグ操作を割り当てることにしたため、キーボード上の矢印キーを用いることとした。

エッジの表示と非表示の切り替え

拡大して、ある階層に着目している時に、その上の階層のエッジが重なることにより視覚的に混雑し、視認性が下がることがある。例えば図 4.7 の場合には、その注目している階層の上位のエッジも重なって表示されている。この視認性を改善するために、任意の階層のエッジの表示と非表示を切り替えられるようなインタラクションを実装した。1 番目の階層から順に「1」キー、「2」キーを押下すると、それぞれの階層のエッジの表示・非表示が切り替わるように割り当てた。図 4.7 の上位の階層のエッジを非表示にした例を図 5.3 に示す。

ノード位置の固定

4.2.4 で述べたように、手作業でノードを配置出来るようにもした。ノードをマウスでダブルクリックすることを、これに割り当てた。また、ダブルクリックで位置を固定したノードは、マウスのドラッグ操作により位置を移動して、別の場所でノードを固定することが出来るようにした。この、ダブルクリックとドラッグ操作の組み合わせにより、手作業でのノードの配置が出来る。固定されたノード以外のノードは、それが固定されていないノードであれば、力指向レイアウトであるノードを固定した状態での力計算により固定されていないノード

¹⁴<http://maps.google.com/>

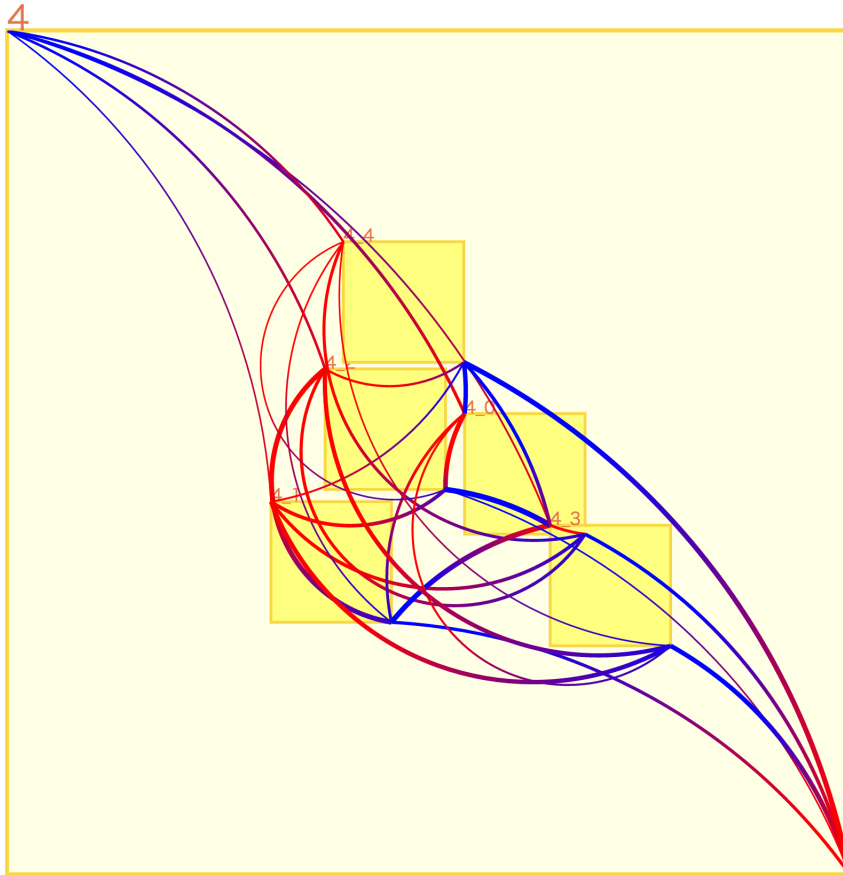


図 5.3: 図 4.7 の、上位の階層のエッジを非表示にした例

の位置が決まる。一方で、両方とも固定されていた場合には、双方の互いに対する力は無視して手作業による配置を優先するようにした。

第6章 ケーススタディ

開発した可視化手法、及びそれを実装したプログラムを用いて、ケーススタディを行った。本章では観察出来たことの詳細を述べる。

5.1 で述べたように、「クックパッド プロのレシピ」という Web サイトを対象としてプログラムを実装した。ケーススタディでは、このサイト上の 12 月下旬の平日 2 日間のアクセスログ情報を用いた。2 日間での、この対象とした Web サイトへのアクセスを記録したログは 386,917 件であった。このアクセスログを対象データに用いて、可視化を行った。

6.1 全体の観察

実装したプログラムを用いて、対象のサイトのアクセスログからユーザのページ遷移を可視化した。作成した図を 6.1 に示す。この図では、1 階層目は手作業にてノードの配置を行い、2 階層目は力指向レイアウトにより自動的にノードを配置した。手作業での配置を行った理由は、最初自動的な配置を行った時に、開発した手法では十分にノードの重なりを避けることが出来なかったためである。しかし、1 階層目のノードの数は 10 個のみであったので、1 階層目だけであれば手作業でノードを配置しなおすことが可能であり、また、そのためのインタラクションも実装してあるので手作業で見やすく配置することにした。

各階層の PV の多さがノードの大きさに対応しているので、6.1 を一目見て、先ず「pro.recipes」コントローラへのアクセスがもっとも多いことがわかる。そこへの流入はどこからが多いかについて、この「pro.recipes」ノードの左上角への流入を見ると、最も多いのが図中 (1) で指した更に上位からの流入であることがわかる。また、他のノードからも流入があるが、特に (2) で指した「search」からの流入と、(3) で指した「recipe_sets」からの流入が多いことが読み取れる。この全体図の状態で、「pages」ノードを見てみると、「pages」は複数の子ノードを持つが、特にその中でも (4) で指したエッジが目立つことに気がつく。これは、すなわち「pages」コントローラの中でも、「top」アクションから「pages」の上位であるプロのレシピ内の他への遷移が多いということを示していて、全体の階層の状態からも、このように下層のノードの特徴も読み取ることが出来る。

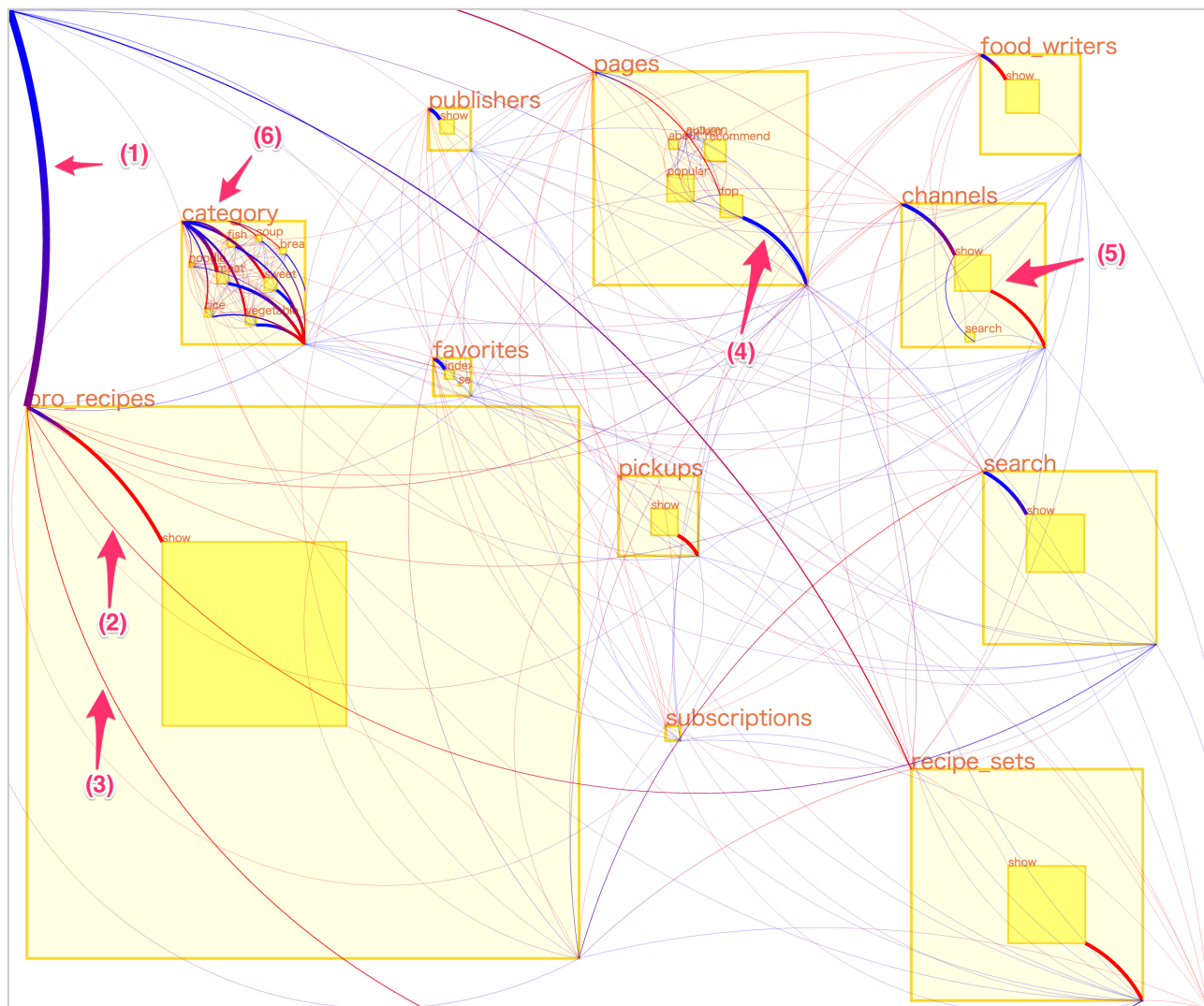


図 6.1: 手作業で 1 階層目を配置した全体図

6.2 階層にズームした観察

6.2.1 pages ノードへの注目

図 6.1 中の「pages」ノードにズームした図を 6.2 に示す。このズームした図を見ると、(1)で指した「top」ノードは先ほど全体の図で確認出来た通りであるが、同時に PV 数すなわちノードの大きさを見ると「popular」ノードが「top」よりも大きいことに気がつく。そして、「top」ノードへの流入と「popular」ノードへの流入について、特に (3) と (4) で指したエッジを比較すると、「top」へは上位の階層からの流入が多いのに対して「popular」へは少ない。すなわち、PV 数では「popular」のほうが大きいですが、そのアクセスは同じ「pages」コントロールの他のアクションから集まっていることが読み取れる。

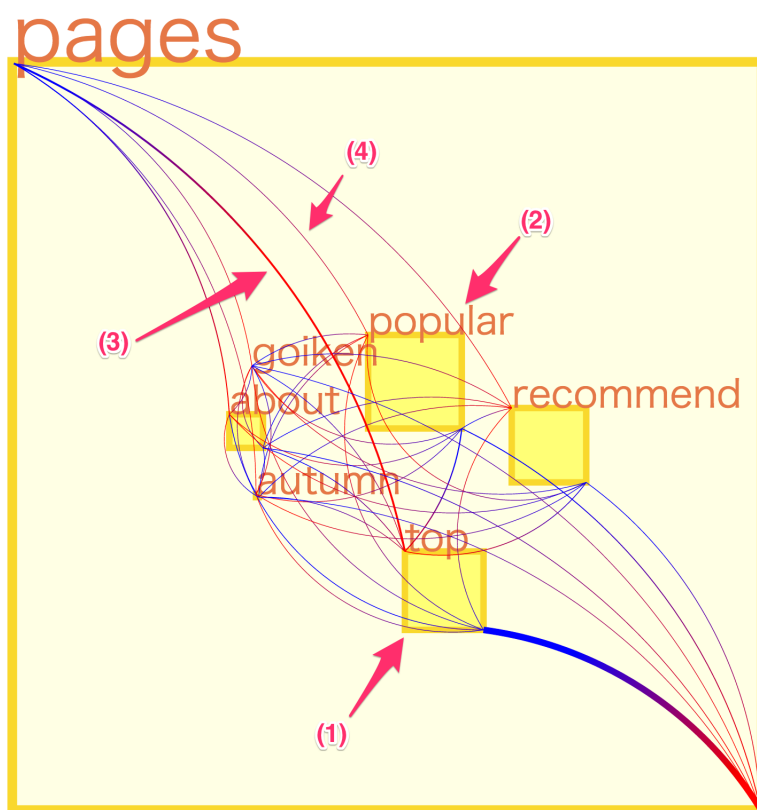


図 6.2: 図 6.1 の「pages」ノードにズームした図

6.2.2 category ノードへの注目

全体の観察を行っているとき、内部に複数のノードがある図 6.1 中の (6) で示した「category」ノードをより詳細に観察してみると、何か新たな知見があるかもしれない気がついた。図

6.1の全体での観察の時点で、それぞれの「category」の内部のノードは、左上角から流入があって右下角へそのまま抜けて行く流れが多いように見えるが、ズームして観察しても、本当に内包する子ノード同士での遷移は少ないのかを確認するために、この部分をズームして観察した。この「category」ノードをズームした図を6.3に示す。これを見ると、やはりほとんどの「category」内のノードに繋がるエッジは、左上角のincomingノードから流入する様子を示すエッジと、右下角のoutgoingノードへと流出するノードが多いことがわかる。「category」内部で他のノードへの多少は遷移もズームすると確認は出来るが、やはり流入とそのまま流出がどのノードも多いという傾向がわかり、「category」内部での回遊はあまり無さそうだということが読み取れる。

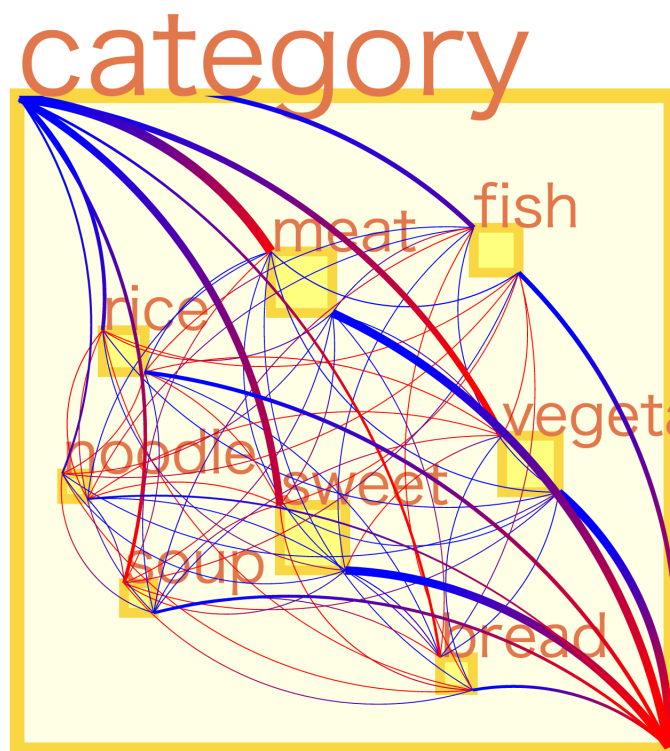


図 6.3: 図 6.1 の「category」ノードにズームした図

6.3 考察

先ず、第3章で検討した要件にある、出来るだけ多くのノードを俯瞰的に把握するという要件は満たすことが出来たと考える。図 6.1 は、図 5.1 で示した Web サイトの階層構造全体を俯瞰的に観察出来て、かつその各ノード間のユーザの遷移も観察出来た。

また、各階層に注目した分析が出来るようにするという要件も、6.2.1 や 6.2.2 で述べたよ

うに、ズーム操作によるインタラクションを用いて詳細に観察することが出来た。特に、筆者はこの「プロのレシピ」の主な開発者の一人として設計から実装まで携わり、これまでもユーザの遷移を観察することがあったが、「top」より「popular」のほうがアクセス数は多いが、その大半は pages の内部での回遊によるものであるという知見と、「category」の内部での回遊は殆ど無いという知見は、これまで行って来た仮説を先に立てて検証する方法での分析では発見することが出来なかった知見で、未知のものであった。探索的な分析を本手法で支援することにより、今回初めて発見することが出来た知見であり、本手法が有用であることが確認出来た。

更に、ユーザの遷移の流量が多い箇所だけでなく、少ない場所も遷移の量が観察出来るようにという要件は、ほぼ満たすことが出来ているが、まだ改善の余地がありそうだということがわかった。具体的には、流量が少ないエッジも、必ず表示しているので無視してしまうことは無いが、やはり流量が多い箇所が極端に目立ってしまうこと、また、流量が少ないエッジも全て表示していると視覚混雑度が高いことが課題としてあった。視覚混雑度の問題に関しては、注目していない階層のエッジの表示と非表示を素早く切り替えることが出来るインタラクションを用いることで、ある程度の対応策にはなることは分かった。

第7章 結論

本論文では、ページ数の規模が大きい Web サイトにおけるユーザの行動の、探索的な分析が困難であるという問題意識から、大規模な Web サイト上におけるユーザの行動の全体像を把握し、探索的な分析を可能とするための可視化手法を開発した。

階層的な構造を持つグラフの可視化手法や、Web のリンク構造の可視化手法、ユーザや人の行動の可視化手法は先行研究にあるが、新たにこれらを組み合わせることで、特に大規模な Web サイトにおける探索的な分析を支援したことが、本論文の貢献である。

先に行ったヒアリングを基に、開発すべき可視化手法の要件を検討した。大規模な Web サイトの全体を俯瞰的に観察出来ること、階層的な分析が出来て特に Web サイトのある階層に着目した分析が出来ること、先にフィルタリングすること無く極力全てのユーザのページ遷移を観察出来ることを開発すべき可視化手法の要件としてまとめた。

まとめた要件を基に、可視化手法を次のように開発した。大規模な Web サイトの全体を俯瞰的に観察し、階層的な分析を行うために、階層構造を矩形の入れ子構造で表現し、更に各階層のグラフを連結図で表現する手法を用いた。また、各階層での詳細な分析のためにはアクセス数やユーザの遷移の流量といった重みにより、ノードを表現する矩形の面積の大きさやエッジの幅の大きさを決定した。ノードの重みを用いて、ノードの力指向レイアウトによる自動的な配置の決定方法についても、パラメータを調整した。更には、全体の分析や任意の階層の詳細な分析を行うための、ズームやパンのインタラクションやエッジの表示・非表示の切り替えのインタラクションを、開発した可視化手法に導入した。

開発した可視化手法を、実際に運用されている Web サイト向けに実装した。そして、この実装を用いて、ケーススタディを行った。ケーススタディでは、探索的な分析により、いくつかの未知の知見を新たに発見することが出来、開発した手法が有用であることが確認出来た。

謝辞

本論文を執筆するに辺り、また、長い研究生生活において、指導教員である三末和男准教授には大変熱心かつ丁寧なご指導を賜りました。様々な面で多大なご苦勞をおかけしてしまいましたが、それでも熱心にご指導ご鞭撻を頂きました准教授に、深く感謝申し上げます。また、田中二郎教授を初め、高橋伸准教授、志築文太郎准教授、嵯峨智准教授、Simona Vasilache 助教並びに、インタラクティブプログラミング研究室の皆様には、ゼミ等で非常に参考になるご意見やご指導を頂き、大変御世話になりました。特に、NAIS チームの皆様にはゼミでの貴重な多くのご意見を初め、日常生活でも大変御世話になり、ありがとうございました。

本論文で対象とした Web サイトを開発運営しており、本論文に様々な面でご協力頂きましたクックパッド株式会社ならびにその従業員の皆様、特に、ログデータの使用や研究成果の公開をご快諾頂きました、最高技術責任者 舘野祐一執行役及び、社内の調整等にご尽力頂きました小川伸一郎副部長には、この場をお借りして深く感謝申し上げます。

最後に、長い学生生活で様々な面で支えて頂いた、家族をはじめとした友人、先輩や後輩の皆様、また、クックパッド株式会社の同僚の皆様に深く感謝申し上げます。本当にありがとうございました。

参考文献

- [1] Bostock, M., Ogievetsky, V., and Heer, J., “ D^3 data-driven documents.” In *IEEE Transactions on Visualization and Computer Graphics*, Vol. 17, No. 12, pp. 2301-2309, 2011.
- [2] Durand, D., and Kahn, P., “MAPA: a system for inducing and visualizing hierarchy in Websites” In *Proceedings of the 9th ACM Conference on Hypertext and Hypermedia*, pp. 66-76, 1998.
- [3] Dwyer, T., “Scalable, versatile and simple constrained graph layout.” In *Proceedings of the IEEE-VGTC Symposium on Visualization 2009*, Vol. 28, No. 3, pp. 991-998, 2009.
- [4] Fluit, C., Sabou, M., and Harmelen, F. V., “Ontology-based Information Visualisation” In *Visualising the Semantic Web*, Springer, pp. 45-58, 2006.
- [5] Freire, M., and Rodriguez, P., “A graph-based interface to complex hypermedia structure visualization.” In *Proceedings of the Working Conference on Advanced Visual Interfaces*, pp. 163-166, 2004.
- [6] Frishman, Y., and Tal, A., “Dynamic Drawing of Clustered Graphs” In *Proceedings of IEEE Symposium on Information Visualization 2004*, pp. 191-198, 2004.
- [7] Herman, I., Melancon, G., and Marshall, M. S., “Graph Visualization and Navigation in Information Visualization: a Survey” In *IEEE Transactions on Visualization and Computer Graphics*, Vol. 6, No. 1, pp. 24-43, 2000.
- [8] Holten, D., “Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data” In *IEEE Transactions on Visualization and Computer Graphics*, Vol. 12, No. 5, pp. 741-748, 2006.
- [9] Holten, D., and Wijk, J. J. V., “A user study on visualizing directed edges in graphs” In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2299-2308, 2009.
- [10] 伊藤 貴之, 小山田 耕二, “平安京ビュー ～ 階層型データを基盤状に配置する視覚化手法” 可視化情報学会第9回ビジュアリゼーションカンファレンス, 2003.
- [11] Johnson, B., and Shneiderman, B., “Tree-maps: A Space-filling Approach to the Visualization of Hierarchical Information Structures.” In *Proceedings of the IEEE Visualization 1991*, pp. 284-291, 1991.

- [12] Kawamoto, M., and Itoh, T., "A visualization technique for access patterns and link structures of web sites." In *Proceedings of 14th International Conference on Information Visualisation 2010*, pp. 11-16, 2010.
- [13] Labroche, N., Lesot, M. J., and Yaffi, L., "A new web usage mining and visualization tool" In *Proceedings of 19th IEEE International Conference on Tools with Artificial Intelligence*, Vol. 1, pp. 321-328, 2007.
- [14] Lai, W., Huang, X., Wibowo, R., and Tanaka, J. "An On-Line Web Visualization System with Filtering and Clustering Graph Layout" In *The IEEE Intelligent Informatics Bulletin*, Vol. 5, No. 1, pp. 11-17, 2005.
- [15] Munzner, T., and Burchard, P., "Visualizing the Structure of the World Wide Web in 3D Hyperbolic Space" In *Proceedings of the First Symposium on Virtual Reality Modeling Language*, pp. 33-38, 1995.
- [16] Nation, D. A., Plaisant, C., Marchionini, G., and Komlodi, A., "Visualizing Websites Using a Hierarchical Table of Contents Browser: WebTOC" In *Proceedings of 3rd Conference on Human Factors and the Web*, 1997.
- [17] 大塚 真吾, 喜連川 優, "Web アクセスログとその利活用 (特集 ソーシャルネットワーク時代の Web インタラクション)" *人工知能学会誌*, Vol. 21, No. 4, pp. 410-415, 2006.
- [18] Phan, D., Xiao, L., Yeh, R., and Hanrahan, P., "Flow map layout." In *Proceedings of IEEE Symposium on Information Visualization 2005*, pp. 219-224, 2005.
- [19] Pitknow, J. E., and Bharat, K. A., "Webviz: A tool for world wide web access log analysis" In *Proceedings of the First International World Wide Web Conference*, 1994.
- [20] Riehmann, P., Hanfler, M., and Froehlich, B., "Interactive sankey diagrams." In *Proceedings of IEEE Symposium on Information Visualization 2005*, pp. 233-240, 2005.
- [21] Shneiderman, B., "The eyes have it: A task by data type taxonomy for information visualizations." In *Proceedings of IEEE Symposium on Visual Languages*, pp. 336-343, 1996.
- [22] Sugiyama, K., and Misue, K., "Visualization of structural information: automatic drawing of compound digraphs" In *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 21, Issue 4, pp. 876-892, 1991.
- [23] Tamassia, R., "Constraints in Graph Drawing Algorithms" In *Constraints*, Springer, Vol. 3, Issue 1, pp 87-120, 1998.
- [24] Turetken, O., and Sharda, R., "Visualization of web spaces: state of the art and future directions" In *Newsletter of ACM SIGMIS Database*, Vol. 38, Issue 3, pp. 51-81, 2007.

- [25] 山口 裕美, 伊藤 貴之, 池畑 裕子, 梶永 泰正, “階層型データ視覚化手法「データ宝石箱」とウェブサイトの視覚化” 画像電子学会論文誌, Vol. 32, No. 4, pp. 407-417, 2003.
- [26] Zaiane, O.R., Xin, M., and Han, J., “Discovering Web access patterns and trends by applying OLAP and data mining technology on Web logs” In *Proceedings of IEEE International Forum on Research and Technology Advances in Digital Libraries*, pp. 19-29, 1998.