

Rails アプリ地図考

Flush Cut



@makicamel

Omotesando.rb #106 スポンサー LT

2025.02.06

自己紹介

- @makicamel / 川原万季
- Ruby 💎とビール 🍺とお酒が好き
- 好きな言葉は「エイってやってバーン」
- (株)アンドパッド 🚧



東京 Ruby 会議 12

<https://regional.rubykaigi.org/tokyo12>



Talks Schedule News Sponsors Policy Staff Access ☰ X



2025.1.18 (Sat.) 11:00～18:00

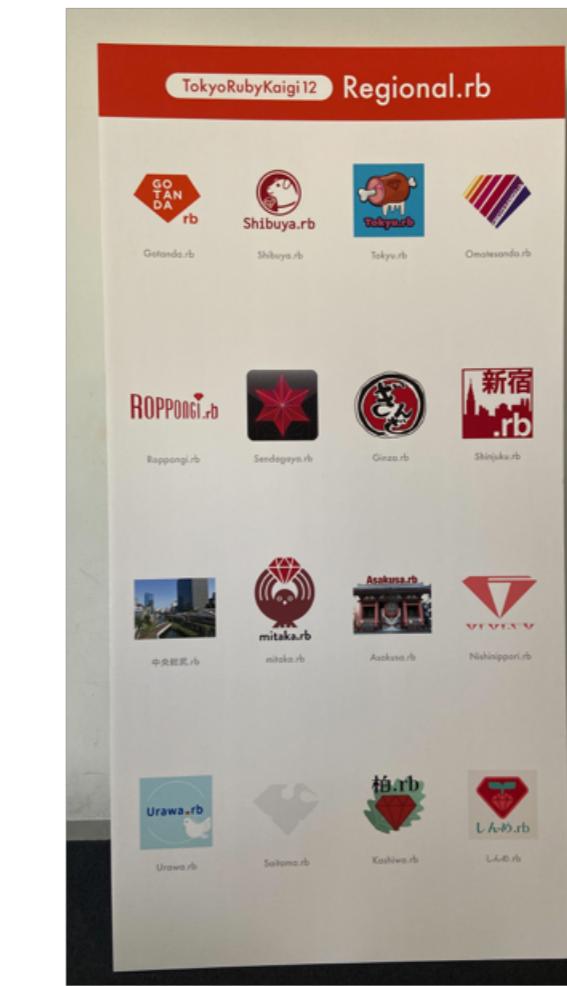
横浜市鶴見区民文化センター サルビアホール

Tsurumi Cultural Center "Salvia Hall"

Organized by the Tokyo RubyKaigi 12 Team

東京 Ruby 会議 12

- 楽しかったですね!



東京 Ruby 会議 12

- 前夜祭でトークもさせてもらいました

- <https://regional.rubykaigi.org/tokyo12/talks>



Thinking about a map for Rails applications

Rails アプリケーションではモデルで概念を表します。この概念はいくつくらいあるのでしょうか。話者が勤務する会社の rails stats ではモデルは 1,345 クラスあります（2024 年 11 月現在）。人間が全てを理解するにはつらい数ですが、全てのモデルを理解してから開発を始めるという人は多くないでしょう。わたしたちはまずアプリケーションの中心となる重要な概念を理解し、そうでない概念は後追いで理解していくからです。

ところで、日本には多くの市があります。政府統計によると 2024 年時点で 792 の市があります¹。ここで Google マップを開いてみましょう。数度ズームアウトすると日本の全体像が現れます。「東京」「大阪」「札幌」という都市が表示され、これらが重要な都市として扱われていることがわかります。ズームインすると次に重要な都市と幹線道路が現れます。

Google マップのようにズームイン・ズームアウトで重要なモデルとリレーション（つまり ER 図ですね）を見られるとそれは Rails アプリケーションの「地図」になるのではないかでしょうか。そんな Gem を作っている話をします。

[1]: <https://www.e-stat.go.jp/municipalities/number-of-municipalities>

makicamel

 @makicamel  @makicamel

東京 Ruby 会議 12

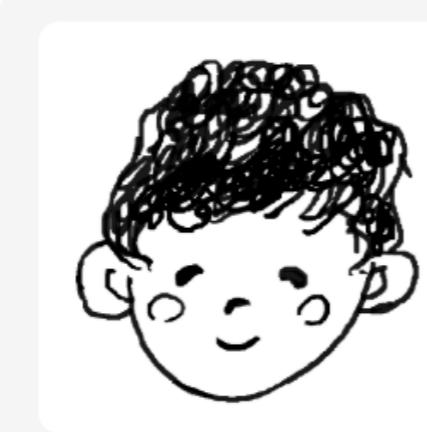
- 今日はこのトークの LT 版
- <https://regional.rubykaigi.org/tokyo12/talks>



The screenshot displays the 'Talks' section of the conference website. At the top, there are two buttons: a black one labeled '前夜祭 2025.1.17' and a grey one labeled '2025.1.18'. Below these buttons is a large, rounded rectangular area containing a speaker's profile picture on the left and a detailed talk summary on the right.

Talks

前夜祭 2025.1.17 2025.1.18

Thinking about a map for Rails applications

Rails アプリケーションではモデルで概念を表します。この概念はいくつくらいあるのでしょうか。話者が勤務する会社の rails stats ではモデルは 1,345 クラスあります (2024 年 11 月現在)。人間が全てを理解するにはつらい数ですが、全てのモデルを理解してから開発を始めるという人は多くないでしょう。わたしたちはまずアプリケーションの中心となる重要な概念を理解し、そうでない概念は後追いで理解していくからです。

ところで、日本には多くの市があります。政府統計によると 2024 年時点で 792 の市があります¹。ここで Google マップを開いてみましょう。数度ズームアウトすると日本の全体像が現れます。「東京」「大阪」「札幌」という都市が表示され、これらが重要な都市として扱われていることがわかります。ズームインすると次に重要な都市と幹線道路が現れます。Google マップのようにズームイン・ズームアウトで重要なモデルとリレーション（つまり ER 図ですね）を見られるとそれは Rails アプリケーションの「地図」になるのではないかでしょうか。そんな Gem を作っている話をします。

[1]: <https://www.e-stat.go.jp/municipalities/number-of-municipalities>

makicamel

 @makicamel  @makicamel

東京 Ruby 会議 12

- 気になった方は元資料を見たり懇親会で声をかけてもらえると嬉しいです

<https://speakerdeck.com/makicamel/thinking-about-a-map-for-rails-applications>



「地図」

地図

- 株式会社アンドパッド 秋葉原オフィスに初めて来た人 
- 地図を見てきましたね

地図

- 道しるべ^[^1]

[^1]:もちろんそれ以外のたくさんの役割があります

Rails アプリの地図

- アプリを歩くための道しるべがほしい

Rails アプリの地図

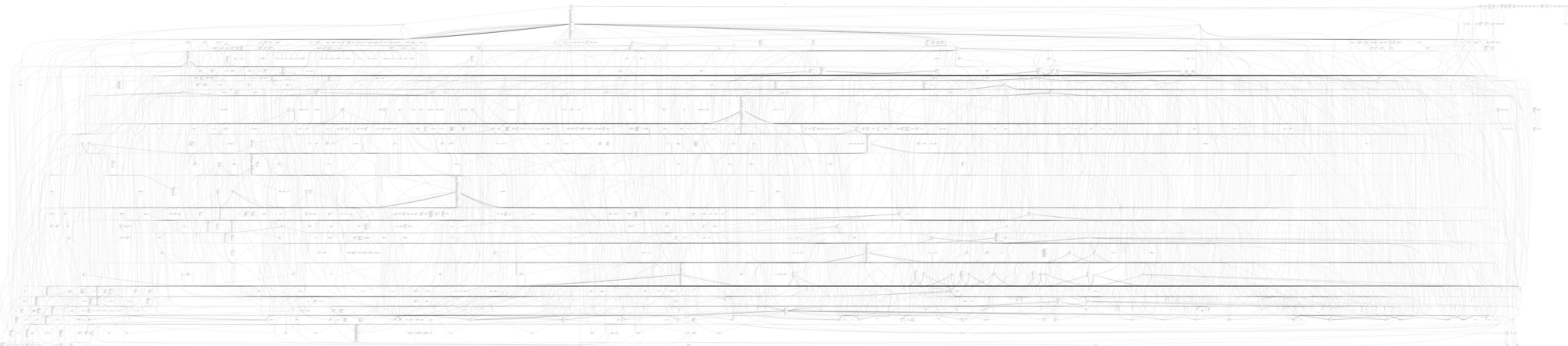
- 道しるべがないと
 -  新しいメンバー
 - オンボーディングで教わる以外のアプリの知識の獲得は手探り
 -  慣れたメンバー
 - 普段触るサービスにアプリの知識が閉じがち
 - いつの間にか知らないドメインが増えている

Rails アプリの地図

- といえば
- ER 図

Rails アプリの地図

- 全 ER 図

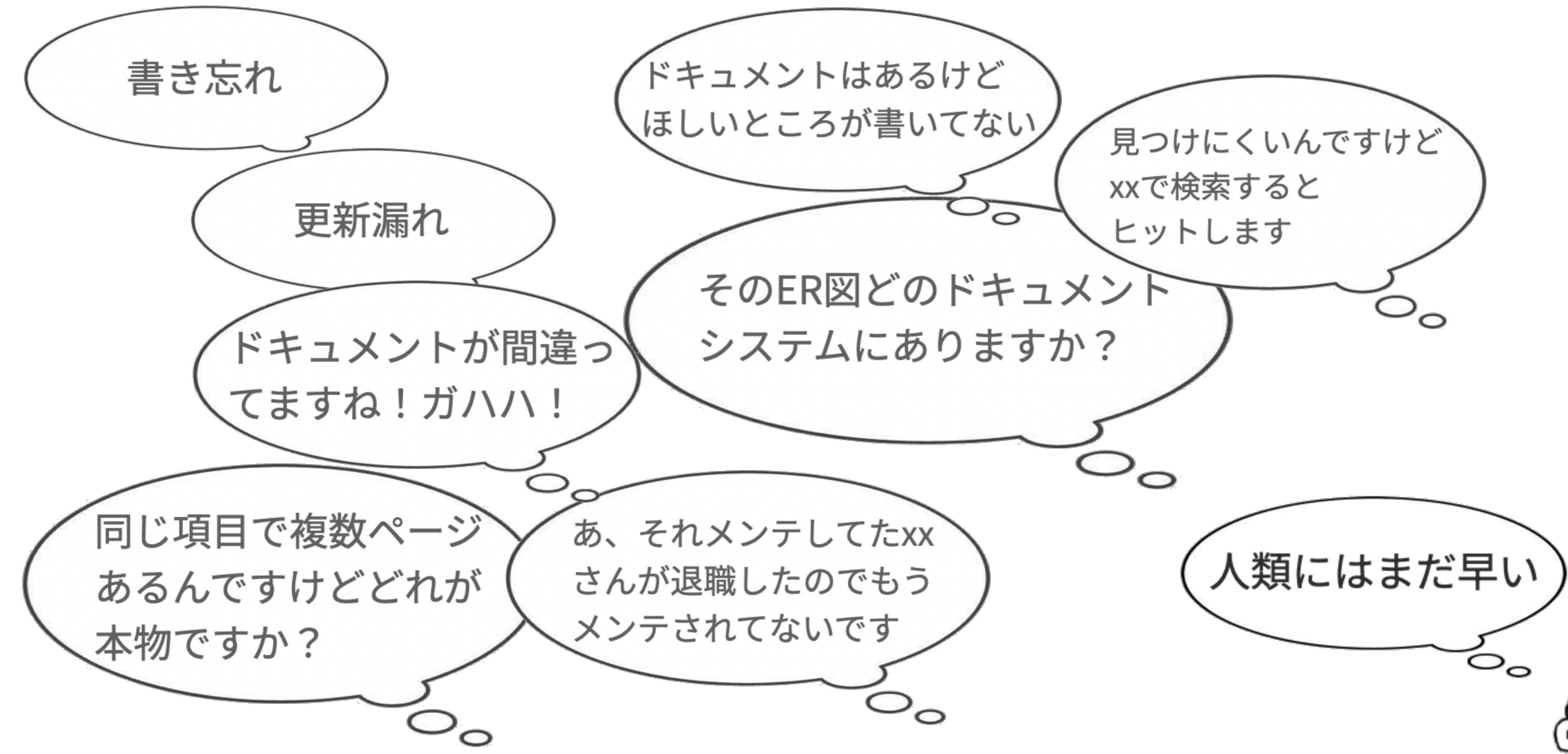


人類にはまだ早い



Rails アプリの地図

● 主要 ER 図の手動メンテ



つくった



makicamel / erd_map

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Security](#) [...](#)

Visualize, explore, and interact with your Rails application's models and associations with ease.

MIT license

Code of conduct

0 stars 0 forks 1 watching 1 Branch 1 Tag Activity

Public repository

main Go to file Code

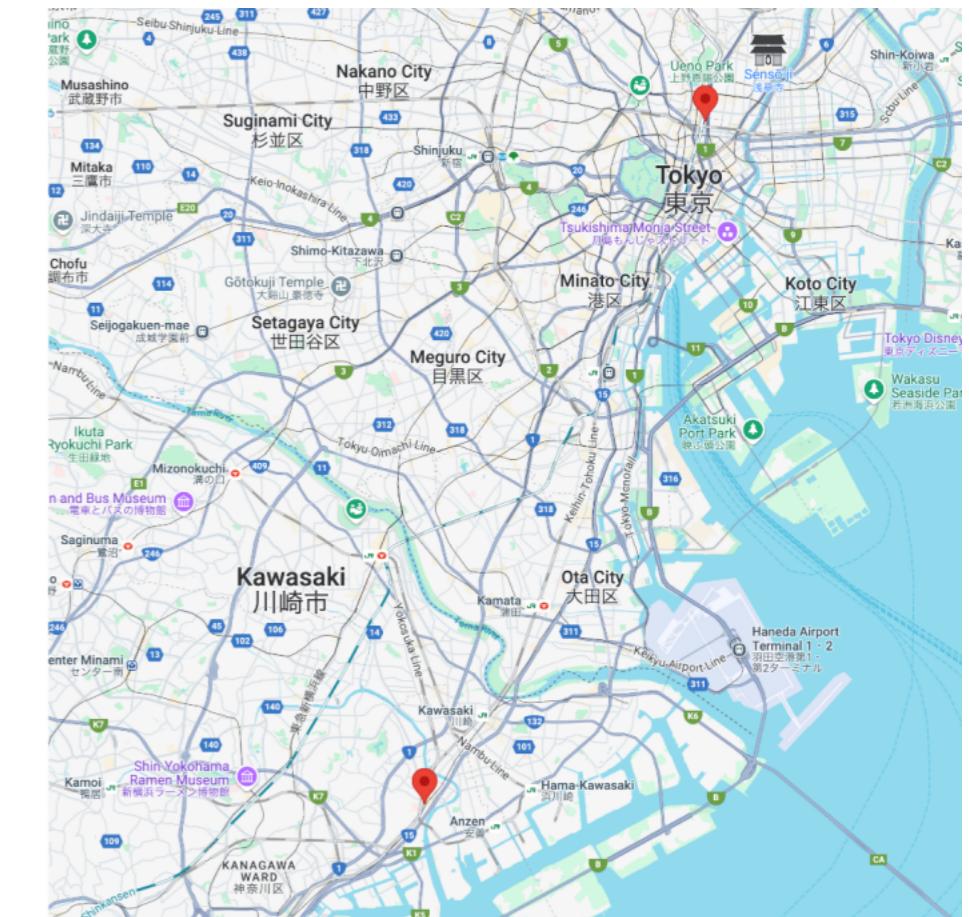
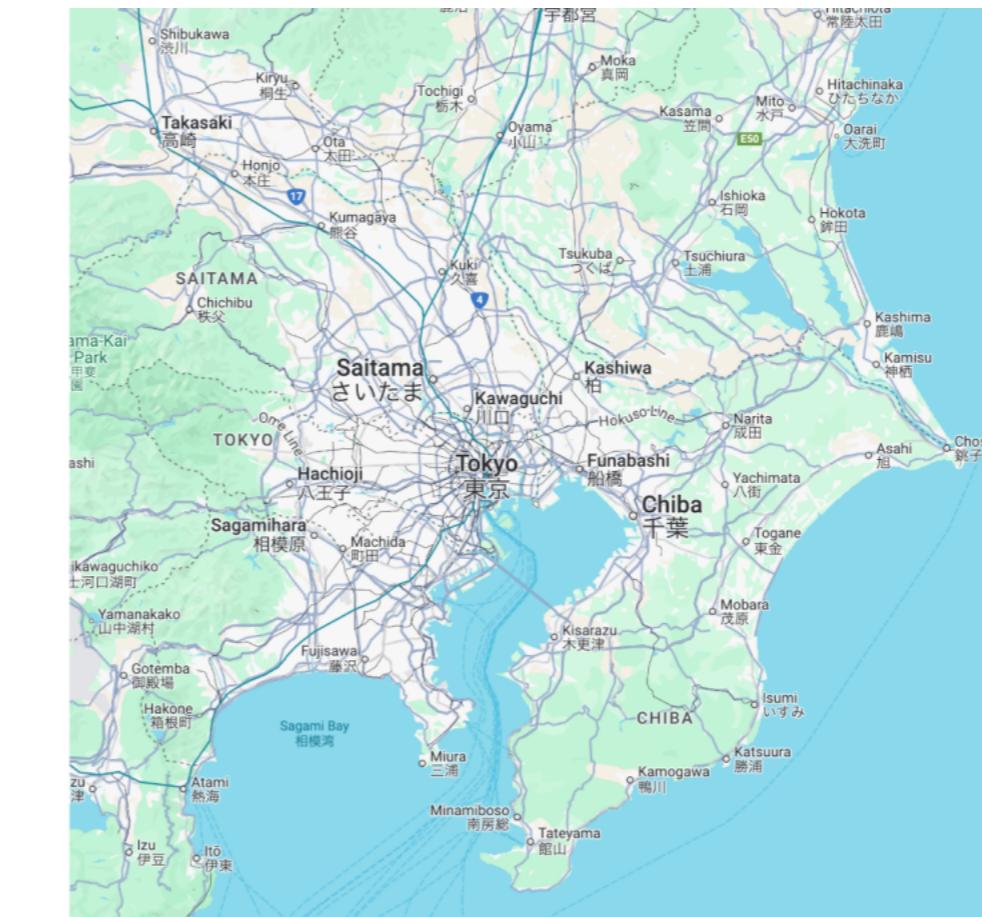
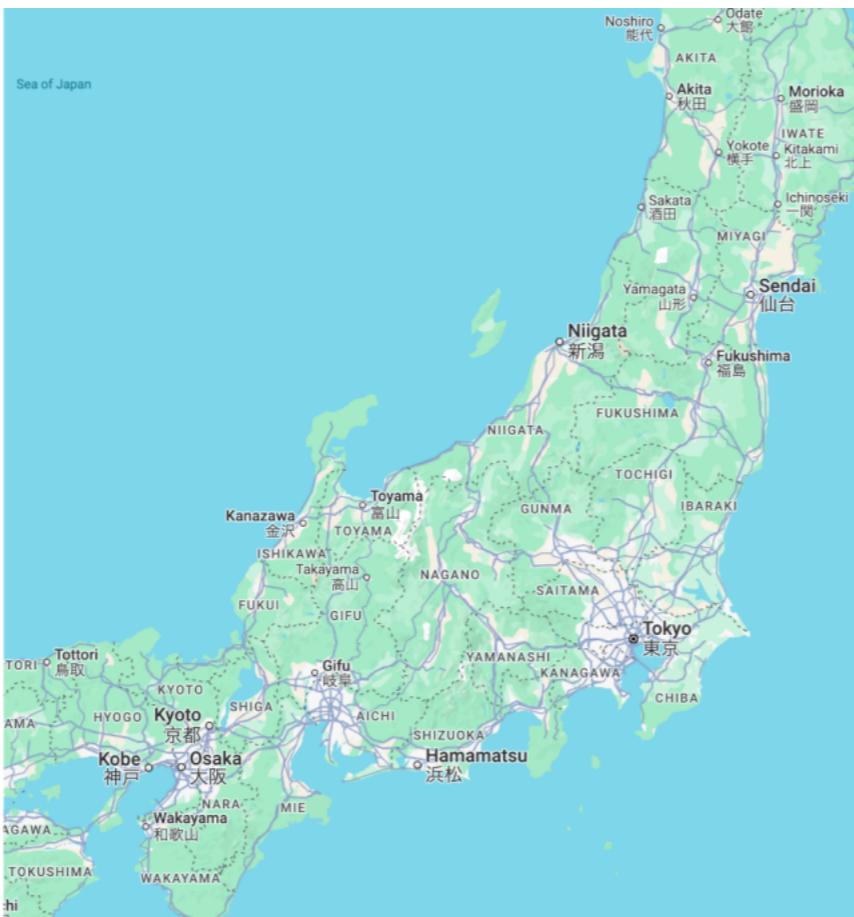
makicamel Add Changelog ✓	caa0fdf · 4 minutes ago	
.github	rails plugin new erd_map --mountable -T	last month
app	Not compute if already computed	4 hours ago

https://github.com/makicamel/erd_map

地図

<https://www.google.com/maps>

- ズームインすると詳細が見える
- ズームアウトすると全体が見通せる

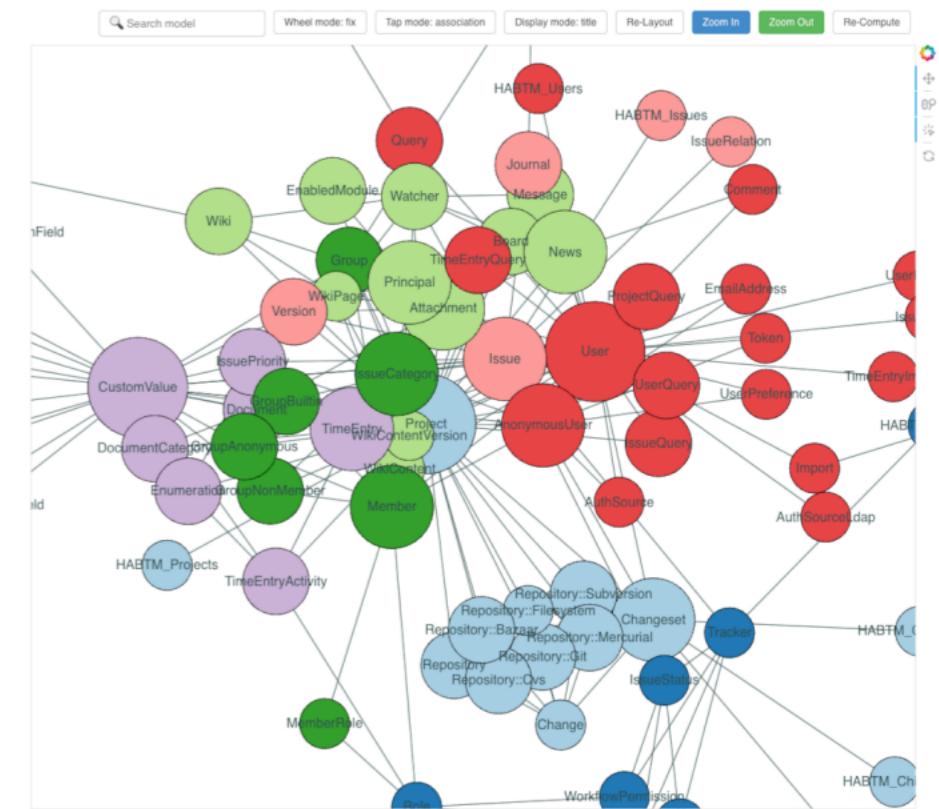
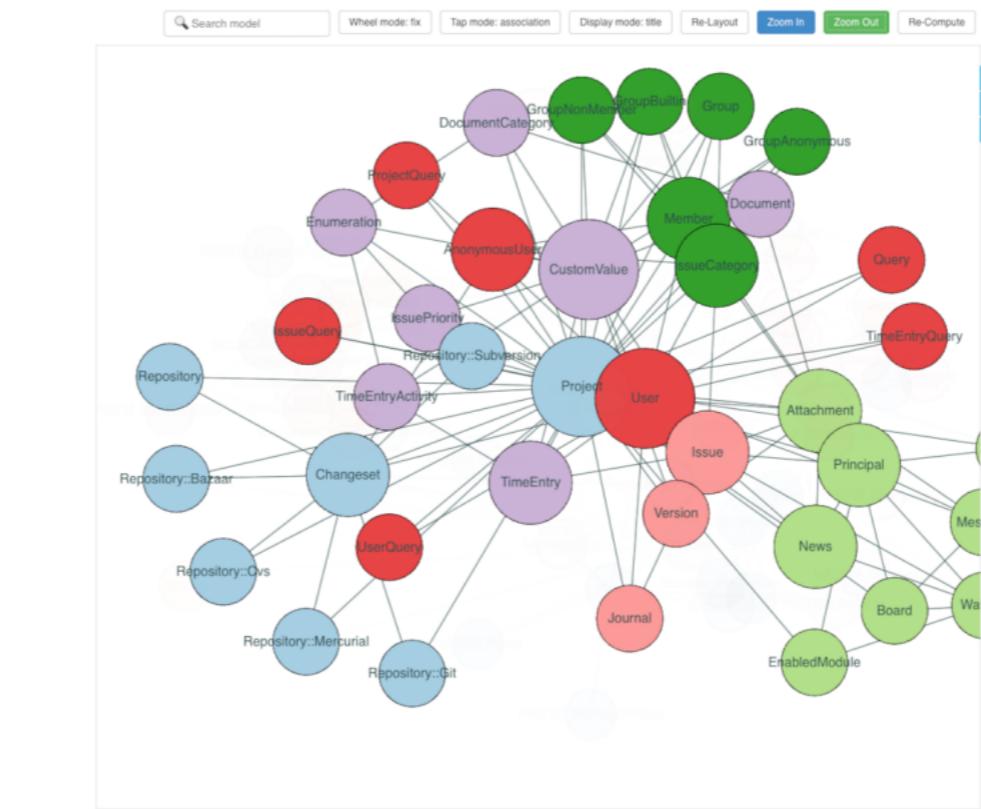
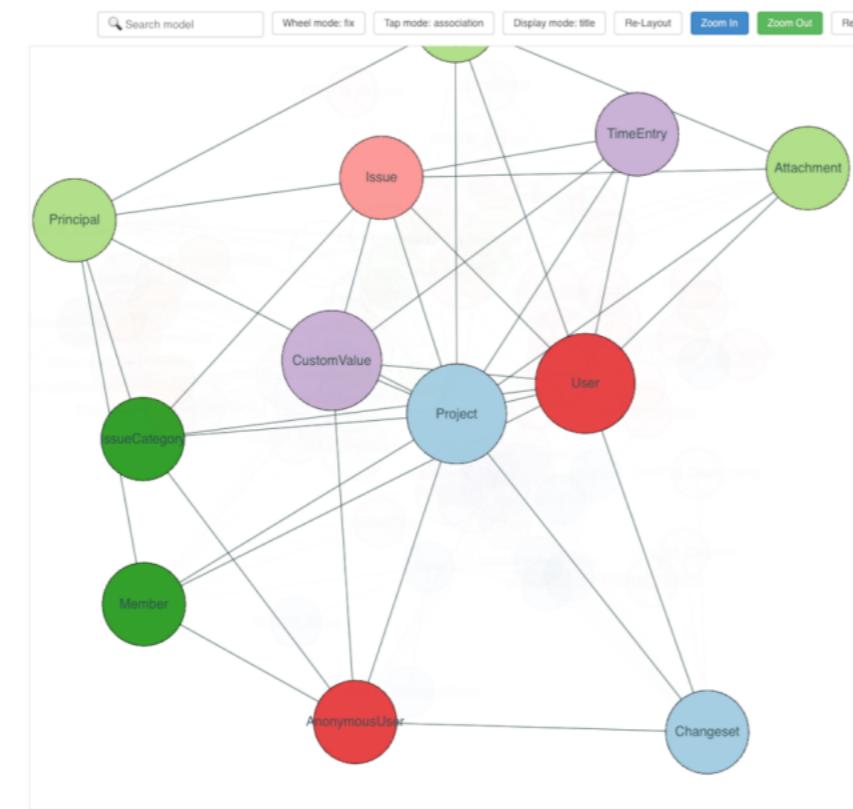
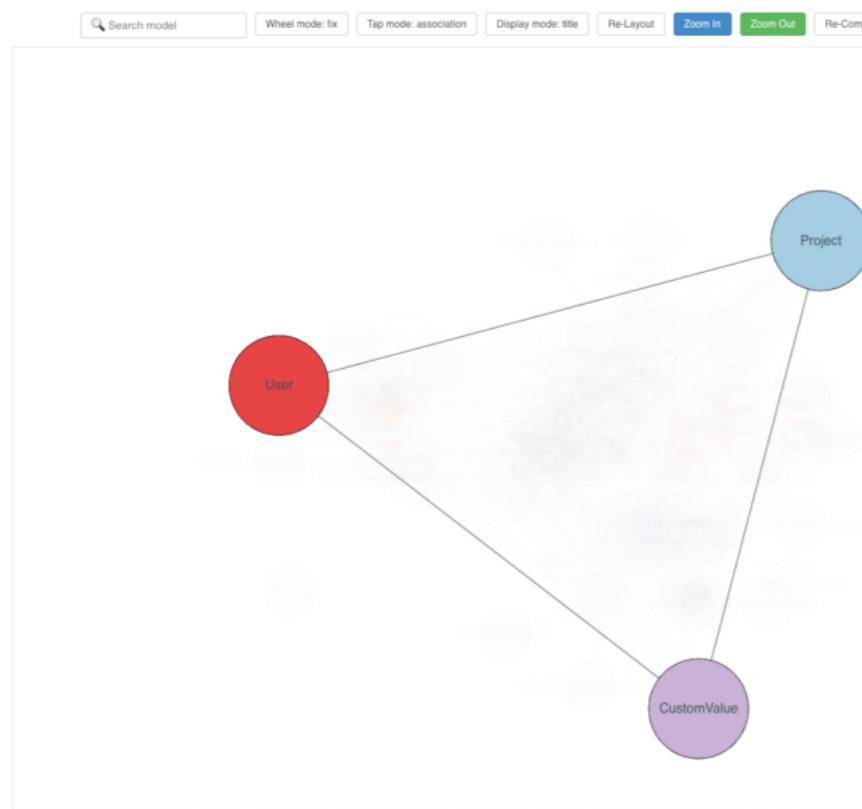


ErdMap



https://github.com/makicamel/erd_map

- ズームインすると詳細が見える
- ズームアウトすると概要が見える





- 最初は 3 つの"重要"なモデルのみ表示
 - ズームインすると次に重要なモデルを表示
 - 重要度の高いモデルは大きく、重要度の低いモデルは小さく表示
- モデルを"コミュニティ"ごとに色分け



- ErdMap づくりに必要なこと
 - モデルの"重要"度の評価
 - "コミュニティ"分割

ErdMap のつくり方



ネットワーク分析

ネットワーク分析

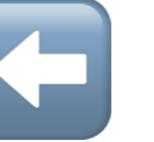
- ネットワークとは
- ノード(点)とエッジ(線)で構成されるデータ構造
- e.g. SNS のユーザー関係
 - ノード: ユーザー、エッジ: フォロー関係

ネットワーク分析

- Rails アプリをひとつのネットワークとして捉える
 - ノード: モデル
 - エッジ: 関連

ErdMap に必要なこと



- モデルの"重要"度の評価 
- "コミュニティ"分割

重要度の評価指標

- 次数中心性
- 固有ベクトル中心性
- 媒介中心性
- etc

固有ベクトル中心性を
使います



重要度の評価指標: 次数中心性

- 直接関連している他モデルの数(次数)を評価
- シンプルで高速

重要度の評価指標: 固有ベクトル中心性

- どれだけ重要なモデルと関連しているか^[^2]を評価
- 次数中心性の「数」の評価に対し「質」を評価
 - e.g. 請求データ
 - ユーザーや注文といった中核モデルと紐づくため**固有ベクトル中心性が高い**
 - 紐づくモデルが少ないため**次数中心性が低い**
 - e.g. 履歴データ
 - 中核モデルと紐づいても関連の終端になりやすく**固有ベクトル中心性が低い**
 - 多数の関連を持つため**次数中心性が高い**

[^2]: 関連する重要モデルの数ではなく相手モデルのスコアの合計

重要度の評価指標: 媒介中心性

- 他のモデル間の最短経路にどれだけ頻繁に登場するかを評価
 - e.g. 請求データ
 - ユーザーや注文といった中核モデルと紐づくため**固有ベクトル中心性が高い**
 - 他のノード間の最短経路にあまり登場しないため**媒介中心性が低い**

固有ベクトル中心性のアルゴリズム

固有ベクトル中心性のアルゴリズム

1. 初期化

- すべてのノードに初期スコアを与える

2. スコアの更新

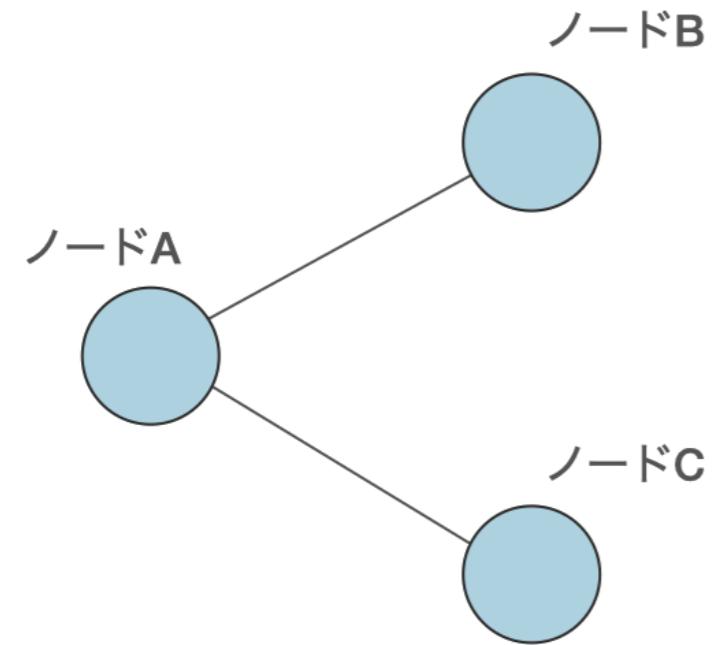
- 各ノードのスコアの「接続先ノードのスコアの合計」を算出し正規化する

3. 繰り返し

- スコアが収束するまで 2 の計算を繰り返す

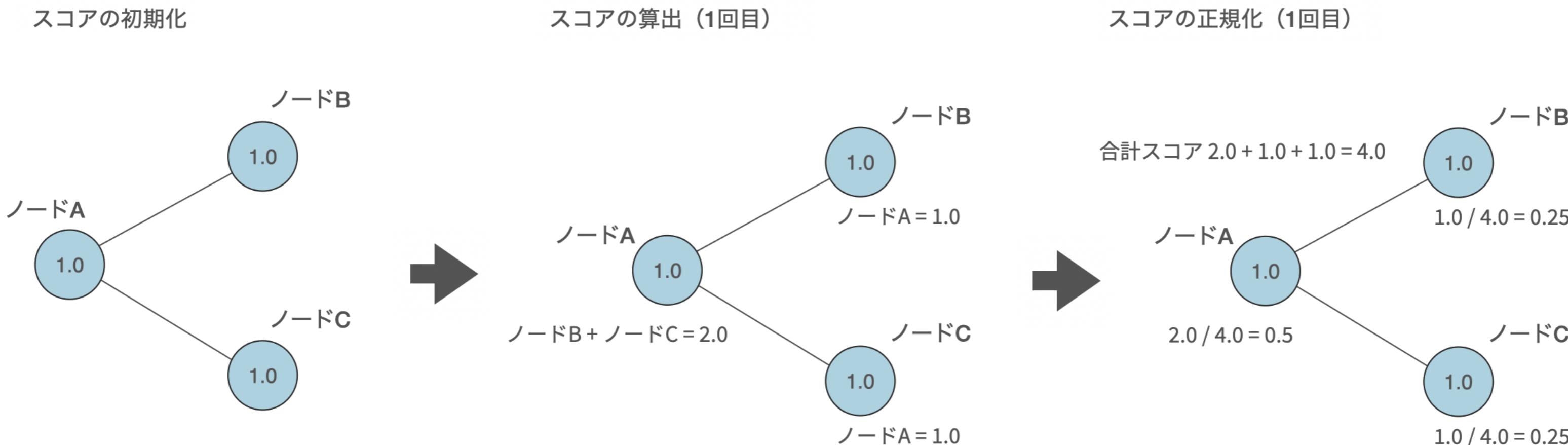
固有ベクトル中心性のアルゴリズム

- 例えば「A-B」「A-C」が接続するネットワークで考えてみる



固有ベクトル中心性のアルゴリズム

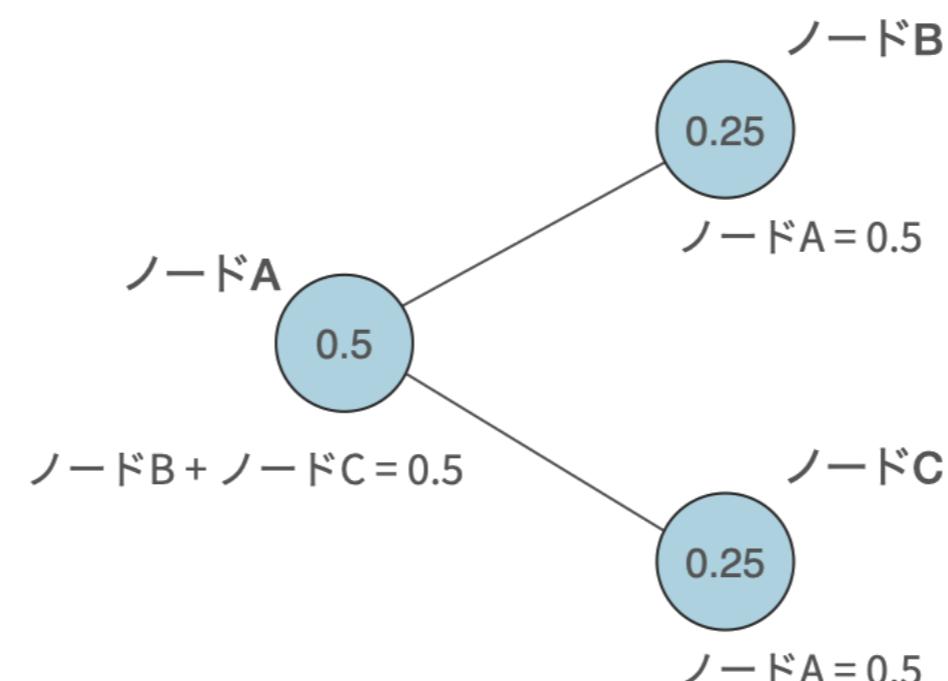
- 1. スコアの初期化
- 2. スコアの更新
 - スコアの算出 → 正規化



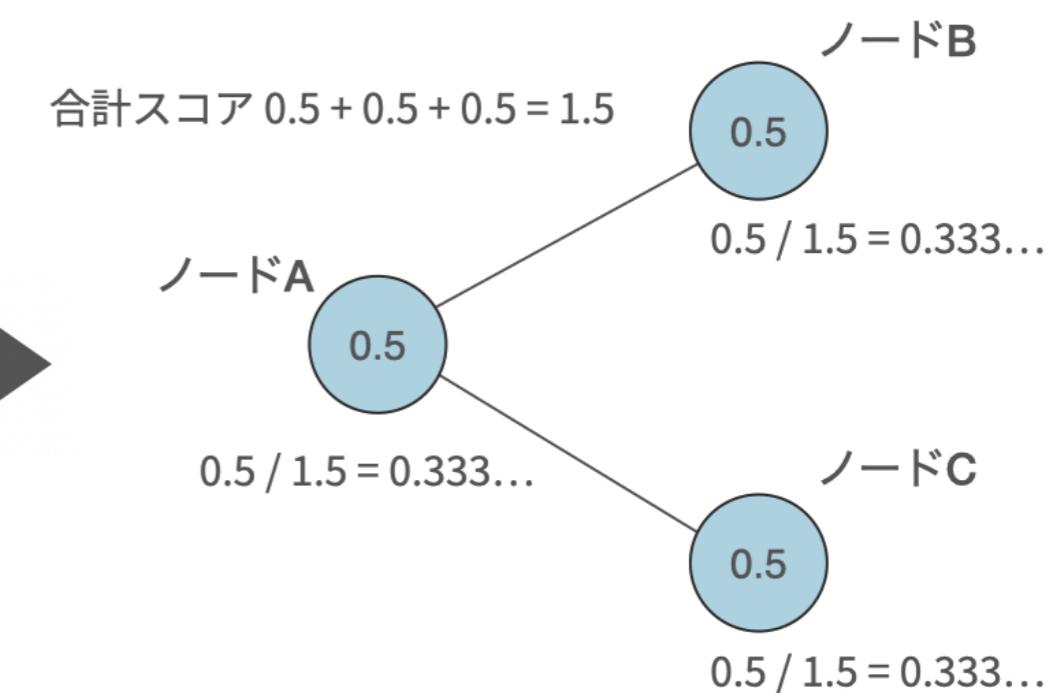
固有ベクトル中心性のアルゴリズム

● 3. スコアの更新の繰り返し

スコアの算出（2回目）

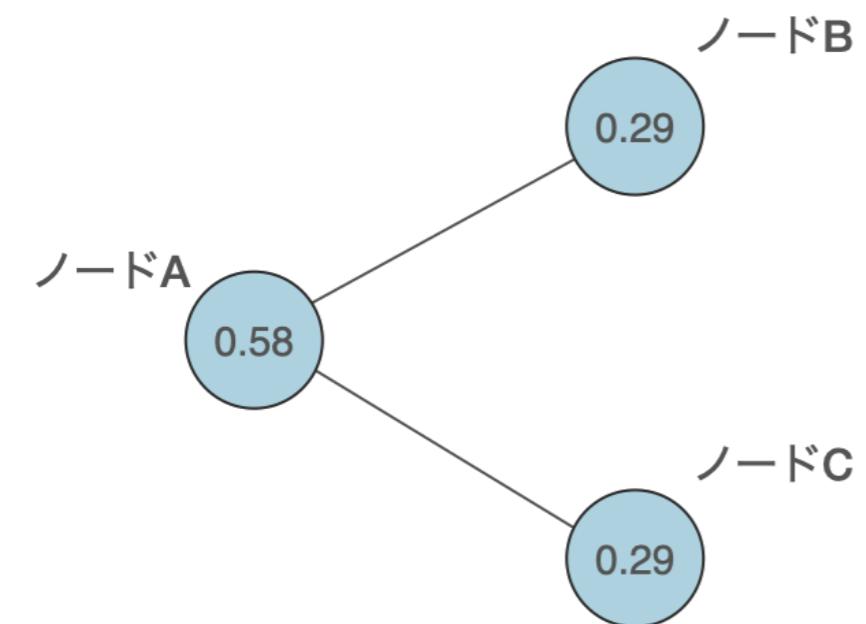


スコアの正規化（2回目）



固有ベクトル中心性のアルゴリズム

- スコアの更新を繰り返すと
スコアがほとんど変化しなくなり 最終的に収束する



ErdMap に必要なこと



- モデルの"重要"度の評価
- "コミュニティ"分割 A blue square button with a white arrow pointing to the left.

コミュニティ分割

- とは？

コミュニティ分割

- 重要なモデルとそれに関連するモデルを知りたい
 - ひとつのモデルはひとつの概念を表す
 - ビジネスロジックは複数のモデルから成ることが多い

アプリケーションの
構造が見えそう

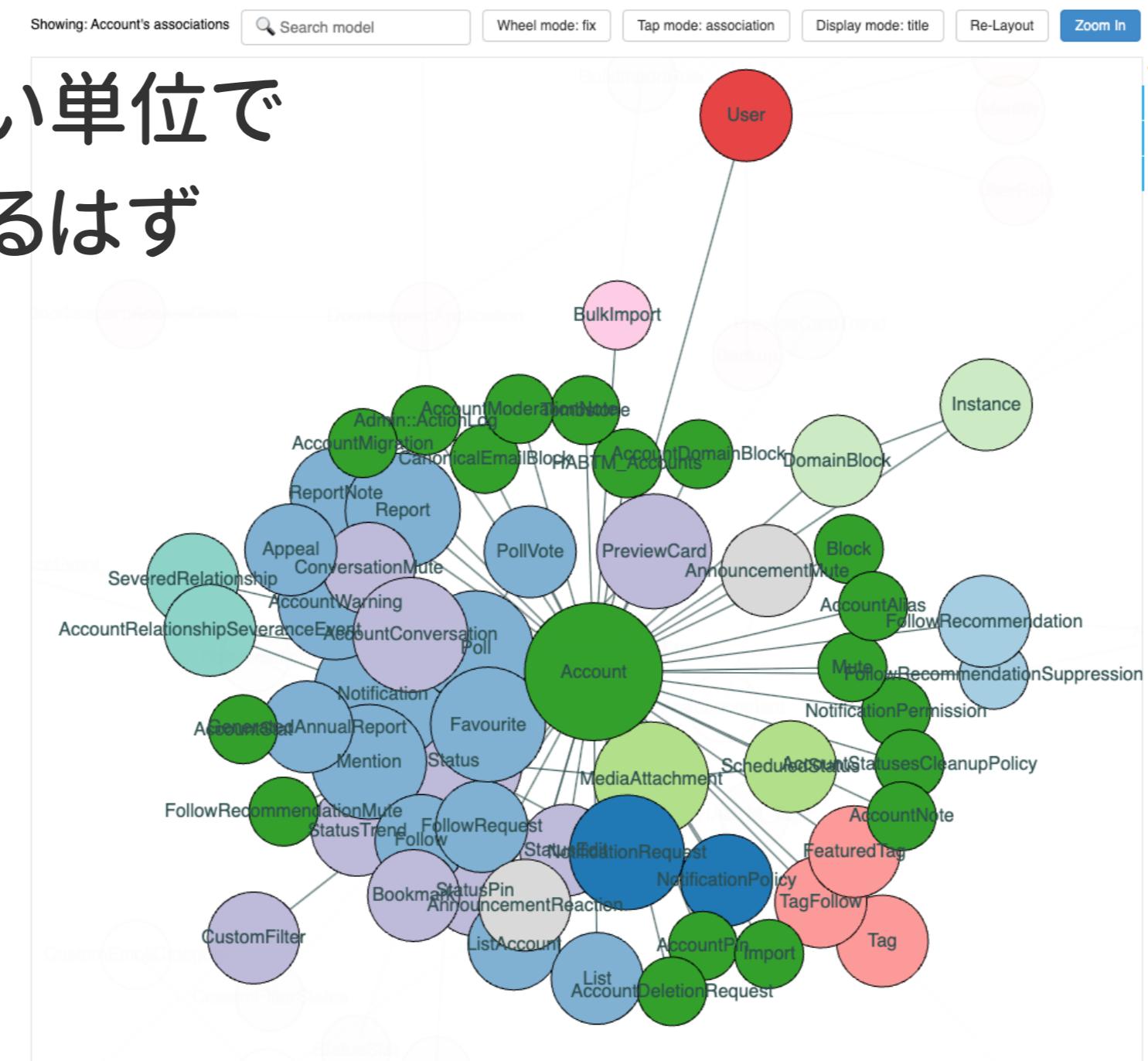


コミュニティ分割

● Mastodon の Account モデルの隣接モデル

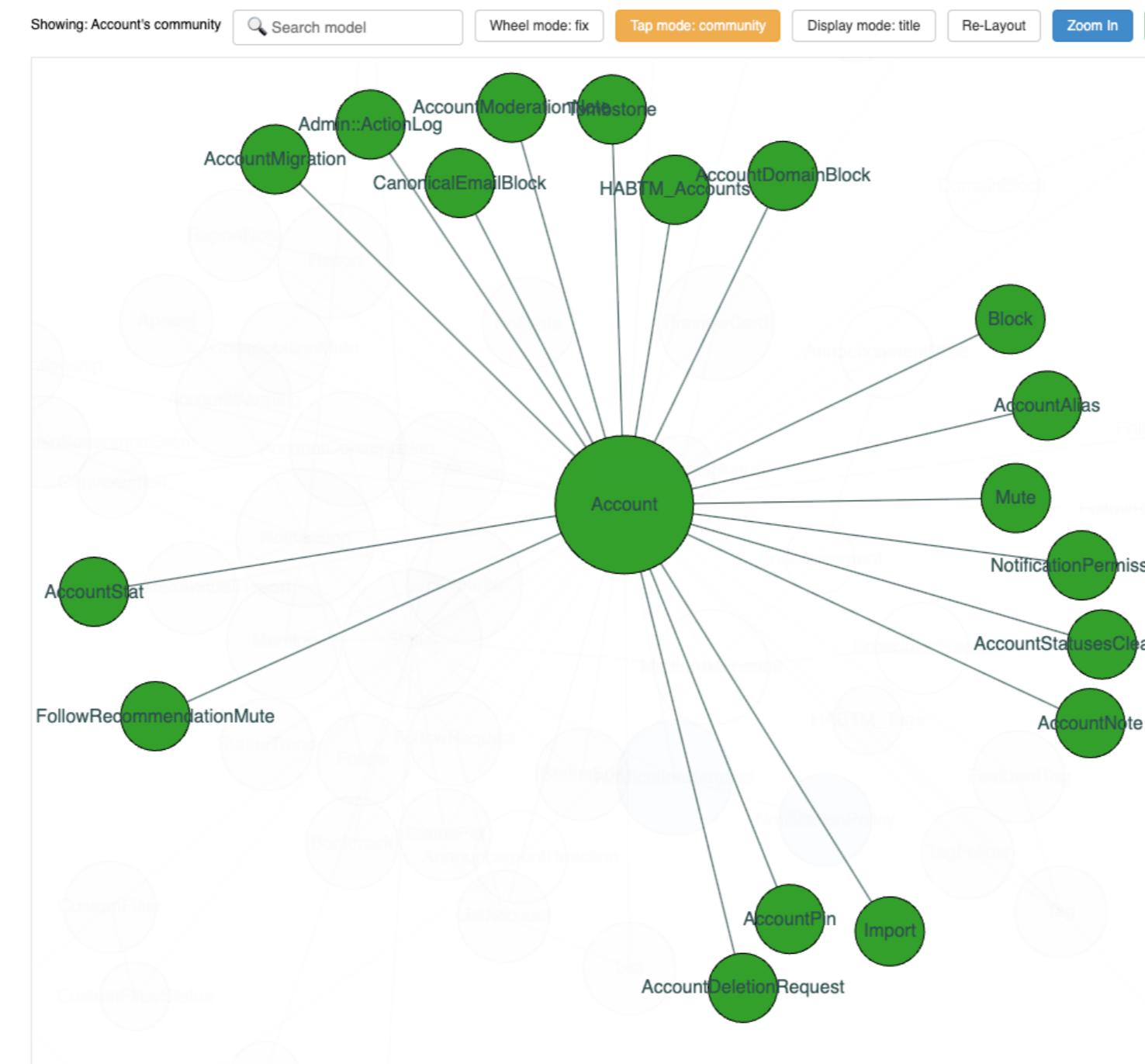
- 情報が多くて情報を読み取りづらい

- 我々はもっと細かい単位で概念を認識しているはず



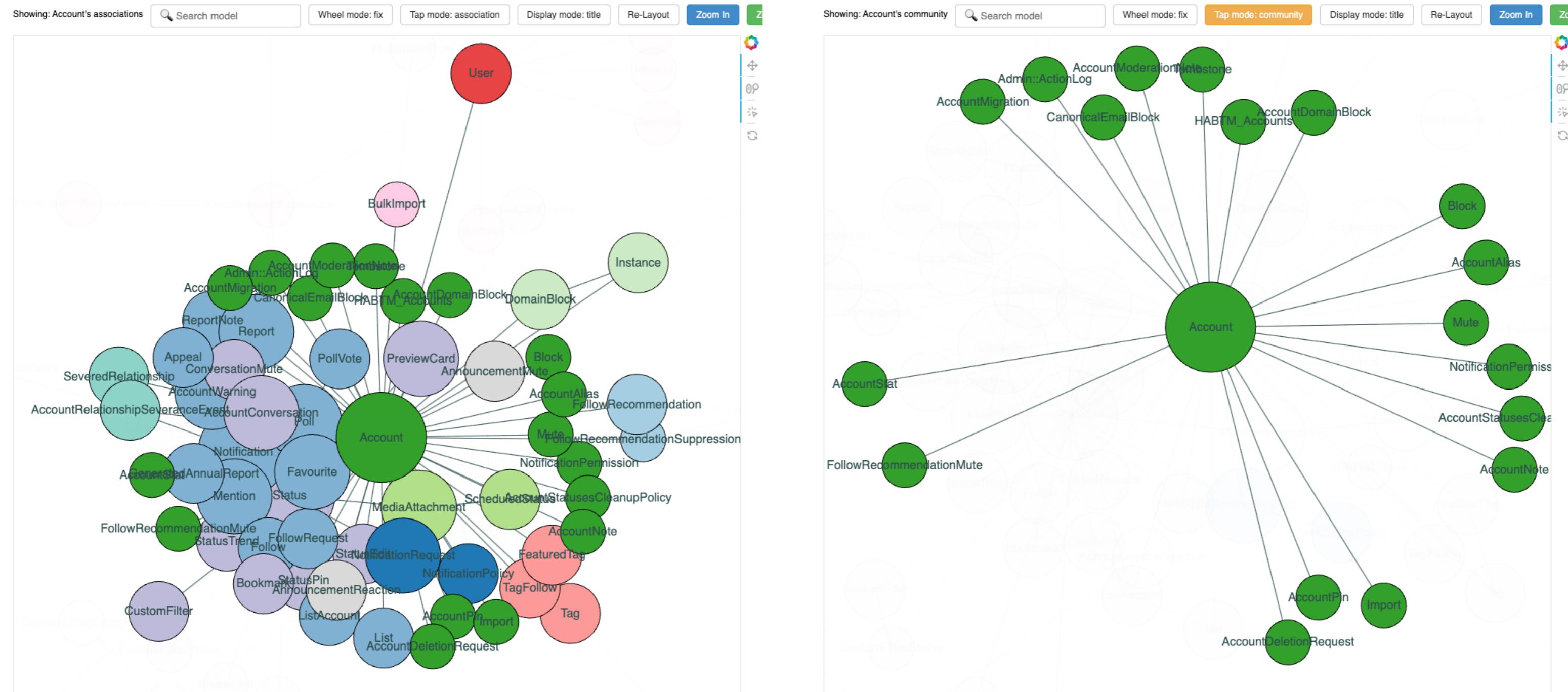
コミュニティ分割

- Mastodon のモデルの Account モデルが所属するコミュニティ
 - Account にまつわる概念が抽出されていそう



コミュニティ分割

- モデルのグループをいい感じに分割して可視化したい



コミュニティ検出

- アルゴリズム
 - Louvain 法
 - Girvan-Newman 法^[^3]
 - etc

[^3]: Louvain法は高速、Girvan-Newman法は計算コストが高いですが、Railsアプリのモデルデータくらいなら関係なさそうです

両方試して
分割結果がしつくりきた
Louvain 法を採用

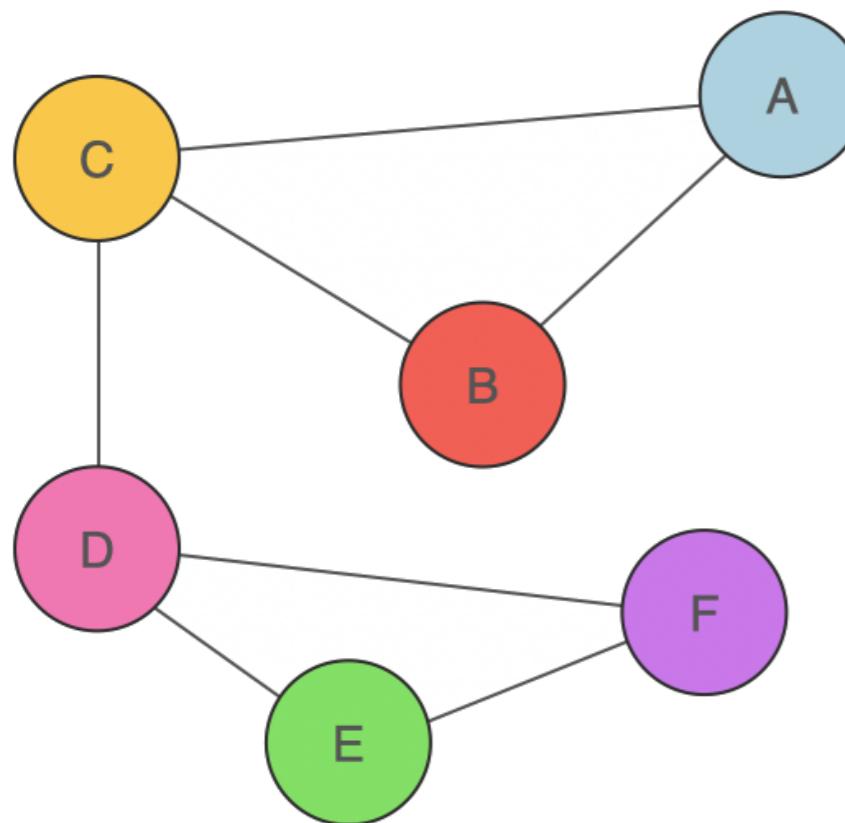


コミュニティ検出: Louvain 法

- ネットワーク内でつながりの強いコミュニティを発見するアルゴリズム
- モジュラリティを最大化するようノードを移動・統合し、最適化を繰り返すこと
で自然なまとまりに分割する
 - モジュラリティ = つながりの密度

コミュニティ検出: Louvain 法

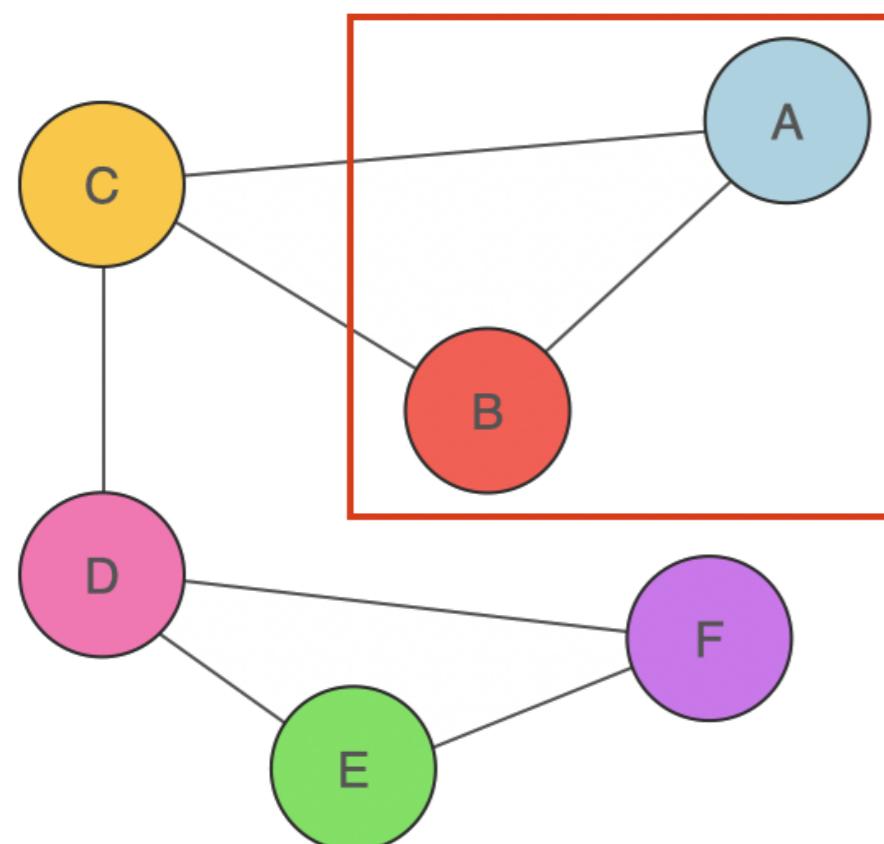
- 6つのノード、7つのエッジを持つネットワークがあるとする
- 初期状態ではすべてのノードは独立したコミュニティ



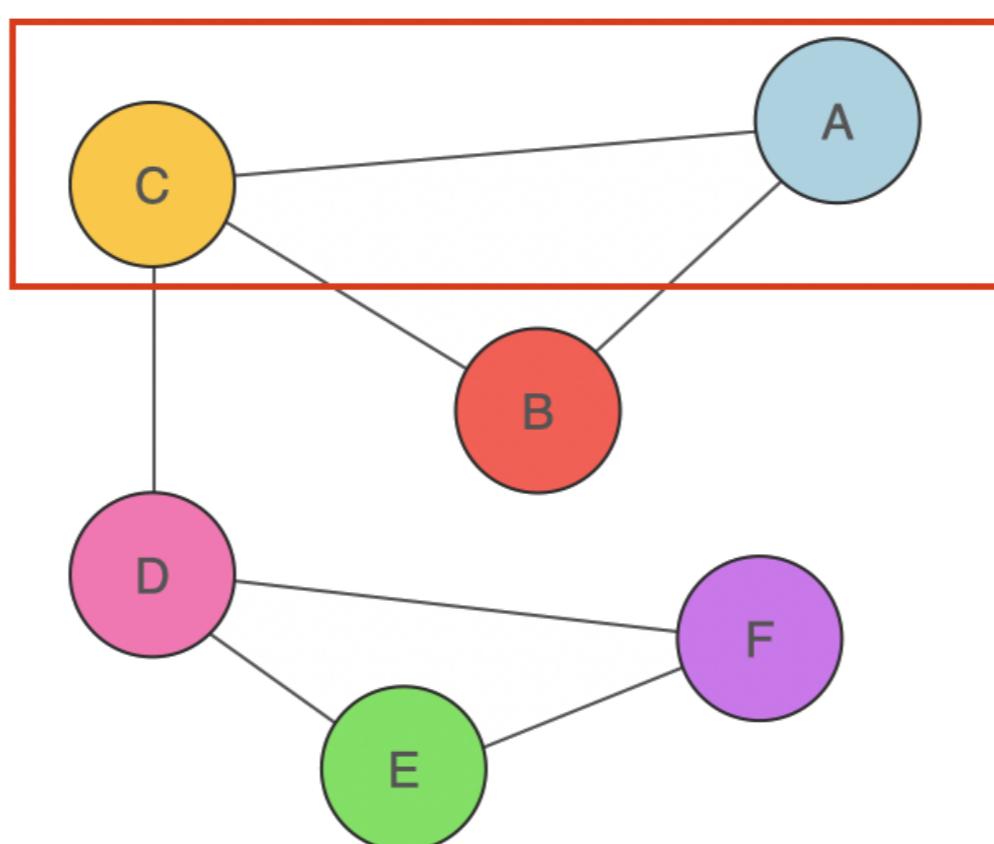
コミュニティ検出: Louvain 法

- 複数のノードを同じコミュニティにした場合のモジュラリティの変化を見て
モジュラリティが高くなるノードグループをコミュニティとして確定する

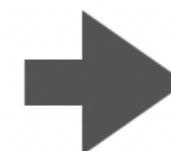
AとBをひとつのコミュニティにして
モジュラリティの変化を見る



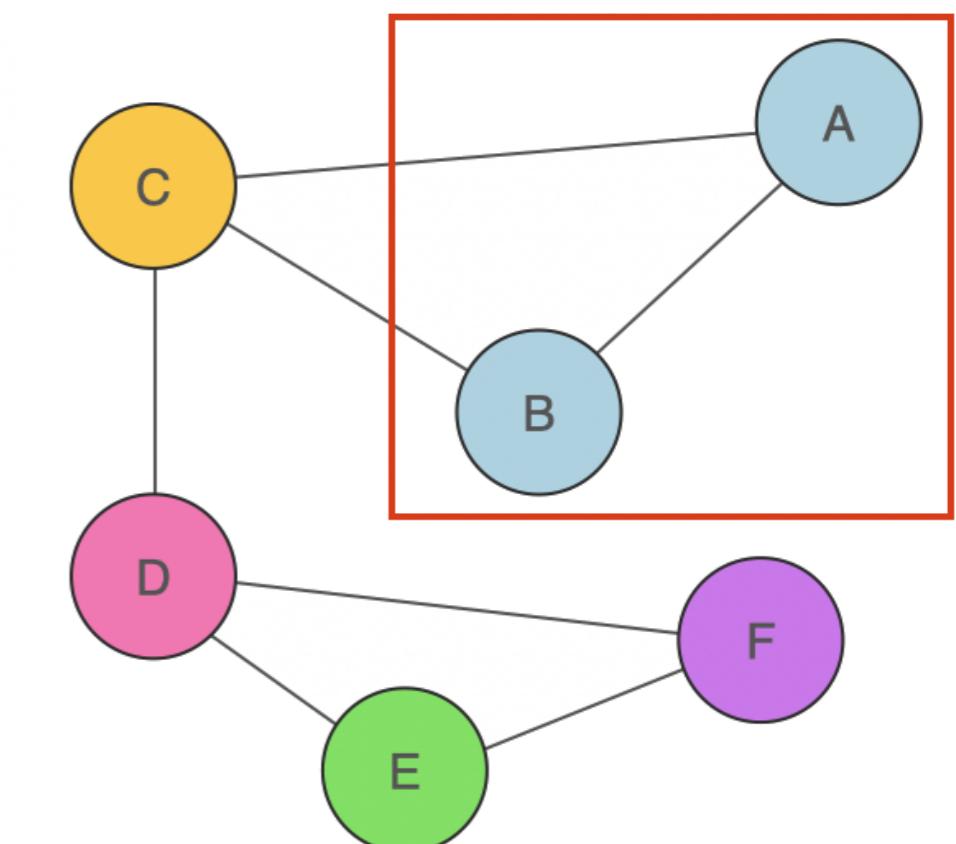
AとCをひとつのコミュニティにして
モジュラリティの変化を見る



AとBが同じ
コミュニティのほうが
モジュラリティが高い

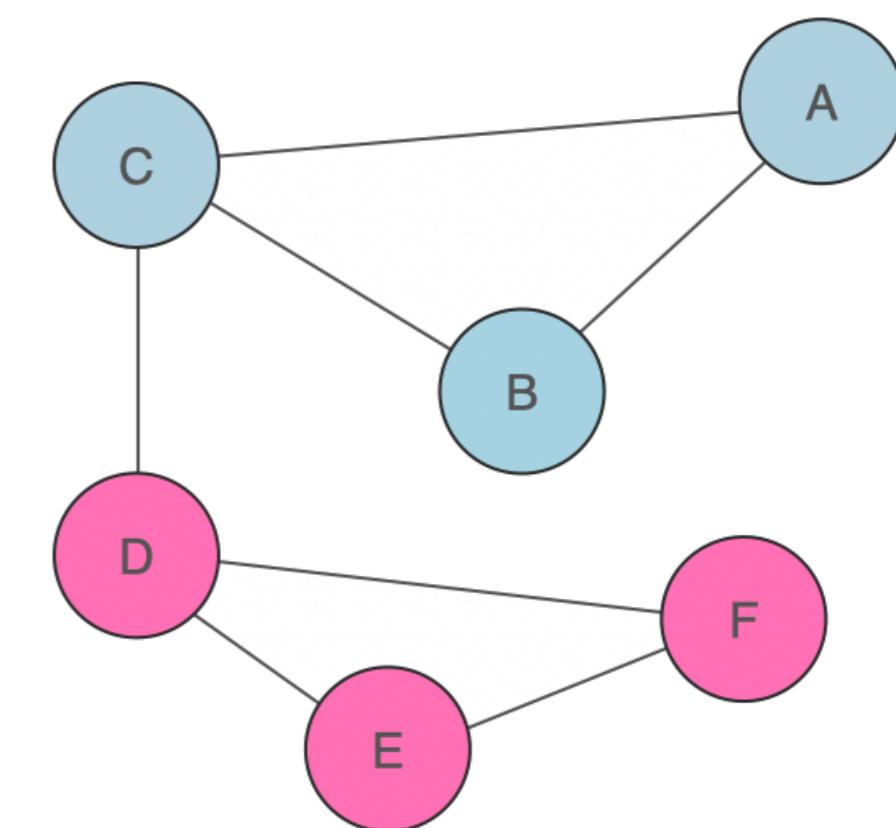


AとBを同じコミュニティに確定



コミュニティ検出: Louvain 法

- この操作を繰り返すと以下のコミュニティに分割される



実装

- 自分で実装しなくていい
- NetworkX が提供するメソッドをコールするだけ

NetworkX

- ネットワーク分析用パッケージ
- Python 製
- Rails から呼びたい

NetworkX <https://github.com/networkx/networkx>

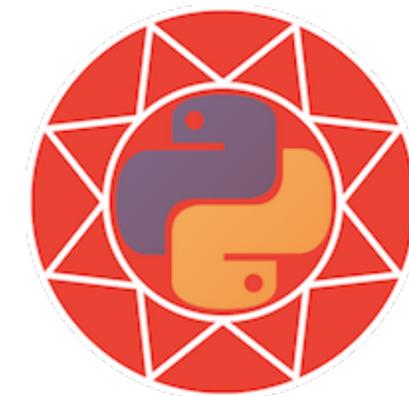


PyCall.rb

- Ruby - Python ブリッジライブラリ
 - Python オブジェクトを Ruby から触れる
 - Ruby オブジェクトを Python 側に見せられる
- mrknさん作

PyCall <https://github.com/mrkn/pycall.rb>

PyCall があれば Ruby で機械学習ができる <https://magazine.rubyist.net/articles/0055/0055-pycall.html>



固有ベクトル中心性

```
networkx = PyCall.import_module("networkx")
graph = networkx.Graph.new
# ... graphにノードやエッジを追加...
centralities = networkx.eigenvector_centrality(graph)
# => {ノード名: 中心性} のハッシュが返る
```

- これだけのコードで
固有ベクトル中心性が得られる



Louvain 法

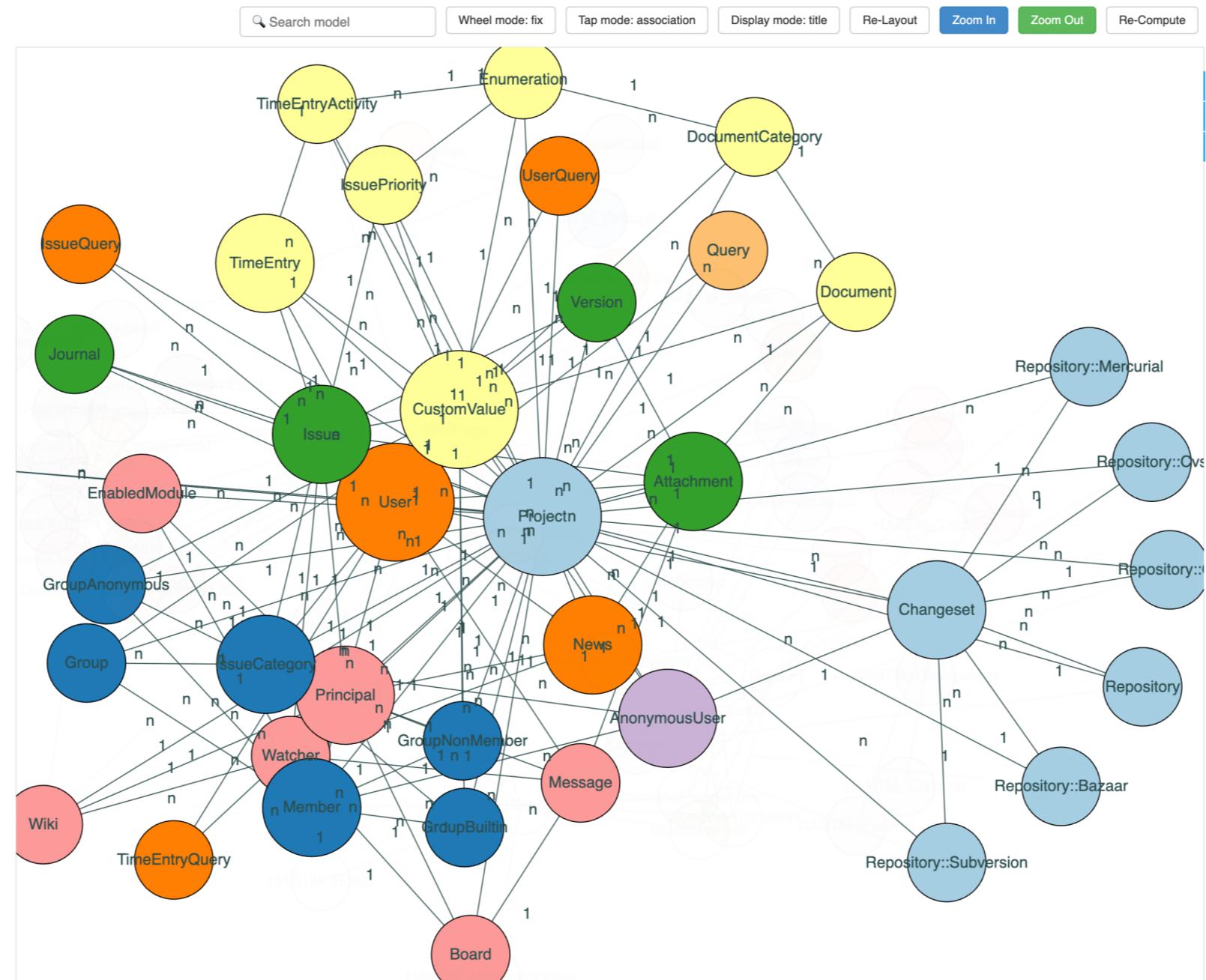
```
whole_communities = networkx_community
    .louvain_communities(whole_graph)
    .map { |communities| PyCall::List.new(communities).to_a }
```

- これだけのコードで
Louvain 法でコミュニティ分割できる





● デモ(Redmine の ErdMap)



Redmine <https://github.com/redmine/redmine>

Sample HTML https://github.com/makicamel/erd_map/blob/main/sample/redmine.html

ErdMap を使ってみて

ErdMap

- 初見の小規模アプリ(モデル数 100)の場合
 - 重要な概念がわかる
 - アプリの機能や構造・データフローのイメージが湧く

ErdMap

- 触り慣れている中規模アプリ(モデル数 1,400)の場合
 - 発見が多い
 - 最近こんなモデルが作られていたのか、とか
 - 気になっていたあのモデルはこういうものだったのか、とか
 - あのサービスの主要モデルの関連はこうなっているのか、とか

ErdMap

- 楽
 - 並列に並ぶファイル名から概念や構造を見出すのは困難
 - 重みづけされグルーピングされて可視化されているとかんたん
 - 発見がある
 - 最新の状態の簡易的な ER 図がいつでもそこにある
- 可視化面白い
 - 見えるようにすると何かが見える(かも)
 - 設計の改善点、ボトルネック...

Rails アプリの地図

- ぜひ試してみてください
- https://github.com/makicamel/erd_map

Special Thanks

@youchan



ANDPAD が支える建設DX

導入企業数
7年連続
シェア **1^{*}**
No.1



※『建設業マネジメントクラウドサービス市場の動向とベンダーシェア（ミックITリポート2024年12月号）』（デロイトトーマツ ミック経済研究所調べ）

Mission

幸せを築く人を、幸せに。

住まいをつくる。ビルや施設をつくる。街をつくる。

生活を豊かにする建築・建設業は、幸せづくりと例えられます。

私たちは、その幸せづくりをする人たちをテクノロジーの力で後押ししていきたい。心からそう考えてます。

我々はこれからもお客様の声をサービスに反映して、建設業界及び建設業従事者様の業務効率化、DX化を支援してまいります。

We're hiring!