Studio Projektowe 1

Wizualizacja formuł logicznych

Jakub Banach, Karol Błaszczak

Raport z wykonanych prac na dzień 12.12.2023

Hipergrafy

Obraz zawierający krąg, zrzut ekranu, sztuka

Opis wygenerowany automatycznieObraz zawierający zrzut ekranu, krąg, Wielobarwność

Opis wygenerowany automatycznieObraz zawierający mapa

Opis wygenerowany automatycznie

Utworzony kod dość dobrze realizuje podstawowe zagadnienie wizualizacji hipergrafów, jednakże w celu zwiększenia czytelności należy go wzbogacić o kolejne funkcje. Podstawowym pomysłem nasuwających się po zajęciach z Panem Profesorem jest wyeliminowanie węzłów o najniższej wartości (czyli o najjaśniejszym odcieniu niebieskiego) w celu zwiększenia przejrzystości bardziej istotnych danych.

Kolejne pomysły dotyczące rozbudowy funkcjonalności kodu to, np.:

1. Wizualizacja interaktywna, stworzona przy użyciu biblioteki takiej jak Plotly, pozwalająca użytkownikom interaktywnie eksplorować hipergraf, wraz z jego osadzeniami grafowymi i grafem skierowanym
2. Algorytmy detekcji społeczności, pozwalające dostarczyć wgląd w strukturę hipergrafu i sposób grupowania węzłów
3. Analiza osadzeń grafowych (np. przez połączenie obu tworzonych sposobów)
4. Modyfikacja hipergrafu w czasie rzeczywistym, pozwalająca użytkownikowi dynamicznie dodawać/usuwać klauzule/węzły w hipergrafie i wizualizować te zmiany w czasie rzeczywistym. Ta funkcjonalność może być przydatna do eksploracji wpływu modyfikacji na osadzenia węzłów i graf skierowany.
5. Wizualizacja podgrafów. To może być przydatne do skupienia się na konkretnych częściach hipergrafu.

Graph Embeddings

Opis metody: najpierw tworzymy przy pomocy node2vec osadzenia grafowe (korzystam z gotowej biblioteki w Pythonie: Node2Vec). Następnie konwertuję to na model word2vec, ponieważ osadzenia węzłów są wektorami numerycznymi w wielowymiarowej przestrzeni, co czyni je trudnymi do bezpośredniej wizualizacji.

Obraz zawierający zrzut ekranu, tekst, diagram

Opis wygenerowany automatycznie

Każdy punkt na powyższym wykresie reprezentuje jeden węzeł w grafie i jest podpisany odpowiednim identyfikatorem węzła. Pozycja punktu odpowiada jego dwóm składowym w przestrzeni osadzeń węzłów. Obie osie reprezentują składowe redukcji wymiarów PCA dla osadzeń węzłów.

Ten wykres ma na celu pokazanie, jak węzły są rozmieszczone w przestrzeni dwuwymiarowej na podstawie ich osadzeń. W przypadku, gdy punkty są blisko siebie, węzły te mają podobne osadzenia w przestrzeni, co może sugerować pewne podobieństwo ich struktury i zależności w grafie.

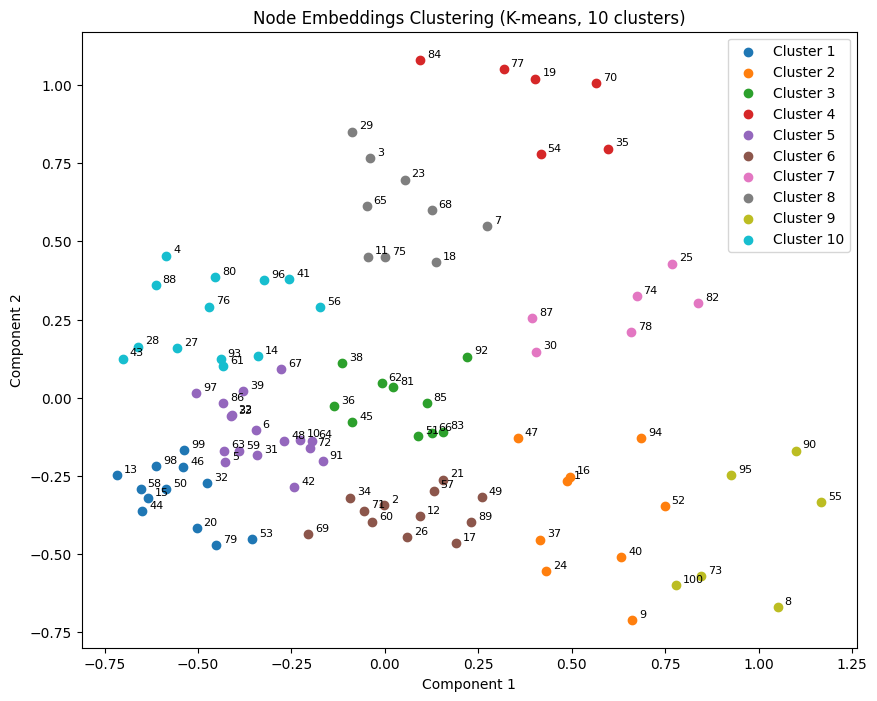
Jedną z ciekawych zależności, którą możemy uzyskać to najbardziej powiązane zmienne (czyli takie które znajdują się najbliżej w przestrzeni trójwymiarowej). Możemy to uzyskać dzięki komendzie:

“model.wv.most\_similar('NUMER\_ZMIENNEJ')”   
Uzyskujemy taką listę dla model.wv.most\_similar('77') dla powyższego wykresu:

('19', 0.8232259750366211), ('70', 0.8122387528419495), ('84', 0.7053765654563904)

Na podstawie poznania powyższych metod i ich wyglądu można wykonać kilka ciekawych zależności:

1. Użycie algorytmu kmeans, który od razu sprawnie pogrupuje poszczególne node’y



2…….