

Rekomendacje oparte na grafach

Systemy Rekomendacyjne 2021/2022

Reprezentacja danych jako graf

- Wierzchołkami grafu mogą być np.
 - użytkownicy
 - treści (np. filmy, utwory)
 - gatunki, albumy, autorzy
 - playlisty utworzone przez użytkowników
- Krawędziami są (niemal) dowolne relacje między wierzchołkami

Reprezentacja danych jako graf

- Graf zazwyczaj jest skierowany (np. użytkownik polubił film, utwór reprezentuje gatunek)
- Graf może być ważony (np. gdy użytkownik ocenia film w skali 1-5)
- Jeśli nie mamy relacji między tymi samymi klasami wierzchołków (np. między użytkownikami), to graf jest N-dzielny
- Do reprezentacji w grafie nie nadają się skalary (np. cechy wyrażone jako liczby z przedziału $[0, 1]$)

Reprezentacja danych jako graf

Vertex embeddings

Przypomnienie - Word2Vec

Później mówiono, że człowiek ten nadszedł od północy

Przypomnienie - Word2Vec

Deep Walk

- <https://dl.acm.org/doi/abs/10.1145/2623330.2623732>
- Modyfikacja Word2Vec
- Generujemy (bardzo, bardzo dużo) losowych ścieżek w grafie
- Ścieżki traktujemy jak zdania w korpusie
- Na wygenerowanych ścieżkach uruchamiamy Word2Vec
- Autorzy sugerują 30-60 ścieżek wygenerowanych z każdego z wierzchołków oraz okienko na 10+10 sąsiednich wierzchołków

Deep Walk

Deep Walk dla grafów ważonych

- Klasyczny algorytm dawał każdemu z sąsiadów równe prawdopodobieństwo wylosowania do ścieżki
- W grafach ważonych prawdopodobieństwo to powinno być proporcjonalne do wagi krawędzi
- Dodatkowo warto rozważyć zmniejszenie wag krawędzi prowadzących do wierzchołków o wysokich stopniach
 - Redukuje problem z nadreprezentacją takich wierzchołków

Node2Vec

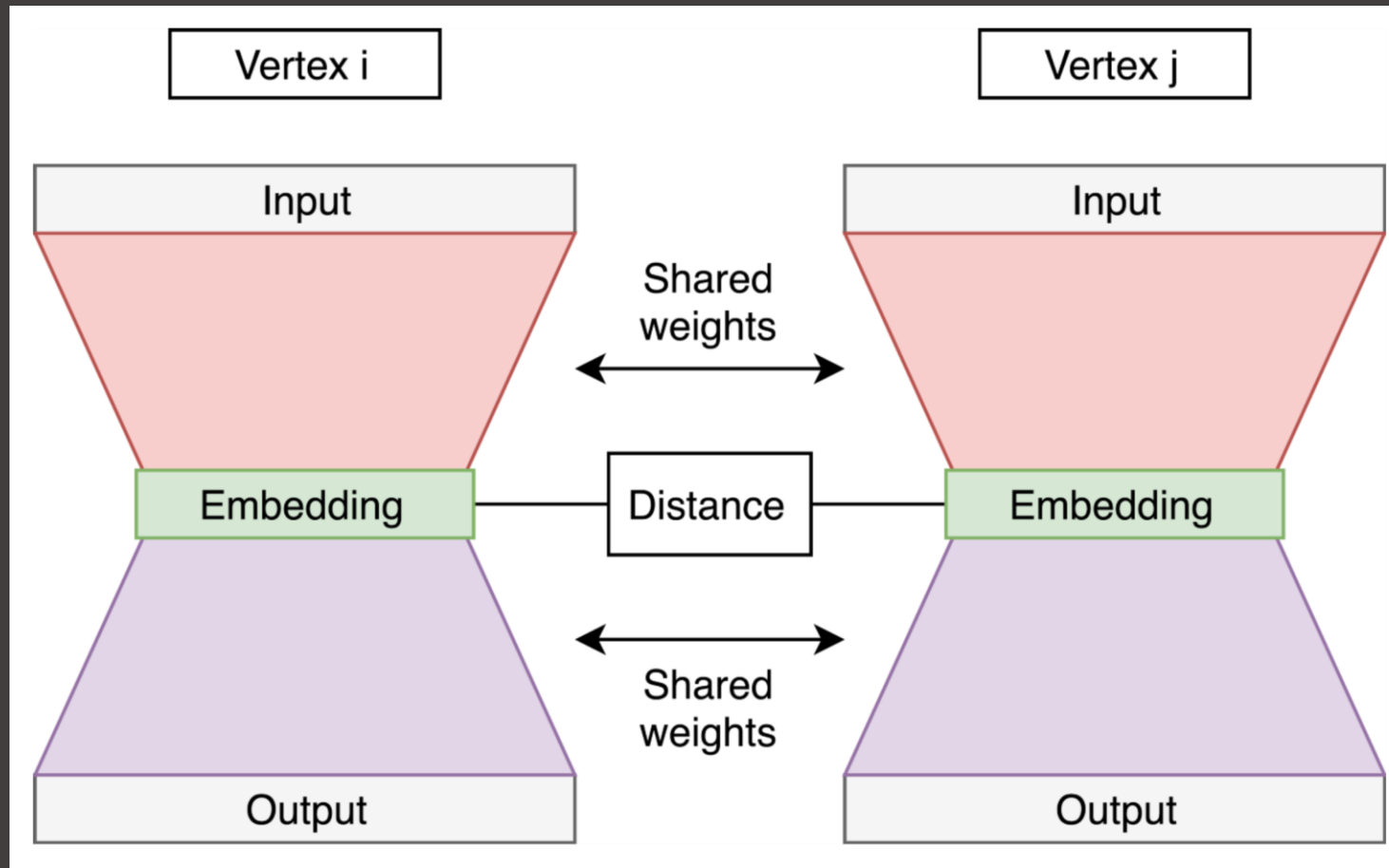
- Modyfikacja Deep Walk rozwiązująca problem braku lokalności
- Dwa parametry:
 - P – steruje prawdopodobieństwem odwiedzenia już wcześniej odwiedzonych wierzchołków
 - Odpowiada za odzwierciedlenie lokalnych zależności
 - Q – steruje prawdopodobieństwem odwiedzenia jeszcze nieodwiedzonych wierzchołków
 - Odpowiada za uwzględnienie kształtu całego grafu

Node2Vec

Structural Deep Network Embedding (SDNE)

- <https://academic.microsoft.com/paper/2393319904>
- Metoda oparta o dwa autoencodery (o wspólnych macierzach wag)
- Ma na celu odzwierciedlać sąsiedztwa pierwszego i drugiego rzędu
- Do funkcji błędu dodawana jest odległość pomiędzy osadzeniami wierzchołków tworzących wspólną krawędź

Structural Deep Network Embedding (SDNE)



Reprezentacja krawędzi?

- Przedstawione metody służą wyłącznie do reprezentacji wierzchołków
- Mało użytecznych algorytmów
- Algorytm naiwny – osadzenie krawędzi to średnia osadzeń wierzchołków
- Mało przydatne w rekomendacjach

Zastosowania

Znajdowanie elementów najbliższych użytkownikowi

- Wykorzystujemy fakt, że wierzchołki wszystkich typów osadziliśmy w tej samej przestrzeni
- Możemy znaleźć elementy (np. filmy, utwory, autorów, playlisty, gatunki) najbliższe użytkownikowi
 - Możemy wykorzystać odległość euklidesową lub cosinusową

Znajdowanie najbardziej podobnych użytkowników

- Znajdujemy grupę użytkowników najbliższych danemu
- Elementy "polubione" przez tych użytkowników wykorzystujemy do rekomendacji
- W ten sposób możemy próbować poszerzyć bańkę, w której funkcjonuje użytkownik

Analiza zmiany preferencji

- Możemy obliczyć *środek ciężkości* preferencji użytkownika jako średnią wektorów reprezentujących polubione/odsłuchane/obejrzane elementy
 - Warto rozważyć średnią z oknem czasowym albo ważoną czasem od ostatniej interakcji
- Analiza zmiany położenia środka ciężkości w czasie może być po pierwsze ciekawa, a po drugie możemy próbować takie zmiany przewidzieć

Rekomendacje dla grup

- Możemy uśrednić preferencje użytkowników tworzących grupę i przeprowadzić proces rekomendacji tak samo, jak dla pojedynczego użytkownika
- To nie jest tak proste, jak się wydaje

Generowanie nieskończonych rekomendacji

- Zaczynamy od jednego elementu (np. aktualnie słuchanego)
- Generujemy ścieżkę po chmurze punktów reprezentujących elementy
- Zazwyczaj warto zadbać o to, by krok w ścieżce nie był zbyt mały - by uniknąć znużenia
- Warto ograniczyć się do pewnego promienia

Ensembling

Łączenie różnych algorytmów

- Często bardzo dobre wyniki może dać połączenie kilku algorytmów rekomendacyjnych
- Algorytm oparty o graf może wytypować zbiór kandydatów do rekomendacji, a collaborative filtering albo bandyta posłuży do posortowania elementów zgodnie z (domniemanymi) preferencjami użytkownika



Bańki

Jak nie uwięzić użytkownika w złotej klatce?

- *Filter bubble* – pozytywne sygnały zwrotne dotyczące poprzednich rekomendacji ograniczają różnorodność rekomendacji przyszłych
- *Echo chamber* – powtarzające się, wybiórcze rekomendacje utwierdzają użytkownika w przekonaniu, że system rekomendacyjny odzwierciedla jego gust, a ograniczony zbiór informacji - rzeczywistość
- Oba mechanizmy, poza aspektami etycznymi, powodują także znudzenie użytkowników

Podsumowanie

- Reprezentacja wiedzy w formie grafu
- Osadzanie wierzchołków grafu w przestrzeni
 - Oparte o Word2Vec
 - Oparte o autoencodery
- Wykorzystanie osadzeń do rekomendacji
- Łączenie algorytmów
- *Filter bubbles*