

Segmentacja użytkowników

Systemy Rekomendacyjne 2023/2024

Algorytmy bandytów

- Generują rekomendacje dla całych grup użytkowników
- Potrzebują dużych objętości danych
 - Ale nie aż tyle, żeby nie podzielić użytkowników na kilka grup
- Same z siebie w żaden sposób nie różnicują rekomendacji między użytkownikami

Dane

- Historia użytkowania
- Reakcje, oceny
- Analiza treści (tekstów)

Podejście naiwne

- Każdy materiał ma przypisany jeden lub kilka tematycznych tagów (np. polityka, ekonomia, sport, życie gwiazd)
- Definiujemy kilka(-naście/-dziesiąt) segmentów na podstawie tagów (np. dla zainteresowanych F1, informacjami z okolic Zielonej Góry oraz życiem osobistym Maryli Rodowicz)
- Dla każdego użytkownika znajdujemy te tagi, które go najbardziej interesują
- Przypisujemy użytkowników do segmentów

Podejście naiwne i dlaczego nie działa

- Każdy materiał ma przypisany jeden lub kilka tematycznych tagów (np. Polityka, ekonomia, sport, życie gwiazd)
 - Tagi przypisane ręcznie są problematyczne, a proces automatyczny jest trudny i zależny od języka
- Definiujemy kilka(-naście/-dziesiąt) segmentów na podstawie tagów (np. dla zainteresowanych F1, informacjami z okolic Zielonej Góry oraz życiem osobistym Maryli Rodowicz)
 - W jaki sposób sprawiedliwie podzielić przestrzeń tagów na segmenty?
 - Jak wykryć tak egzotyczne połączenia jak w przykładzie?
 - Utrzymanie takiego zestawu reguł wymaga dużo pracy i jeszcze więcej eksperckiej wiedzy
- Dla każdego użytkownika znajdujemy te tagi, które go najbardziej interesują
 - Jak zdefiniować "najbardziej interesują"?
- Przypisujemy użytkowników do segmentów
 - Jedyne stosunkowo prosty krok, ale i tak trzeba zdefiniować kryterium przynależności do segmentu

Meta-algorytm segmentacji

- Oblicz osadzenia (*embeddings*) użytkowników (i czasem materiałów)
- Podziel populację użytkowników na klastry
- (opcjonalnie) Przeprowadź postprocessing, żeby segmentacja była zrozumiała dla śmiertelników

Embeddings

Czym jest embedding?

- Chcemy reprezentować użytkowników lub elementy rekomendowane jako wektory
- Spójna reprezentacja wektorowa znacznie ułatwia dalszą analitykę i proces rekomendacji

Collaborative filtering

Collaborative filtering

- Algorytm collaborative filtering zwraca już wytrenowany model reprezentacji użytkowników (i elementów)

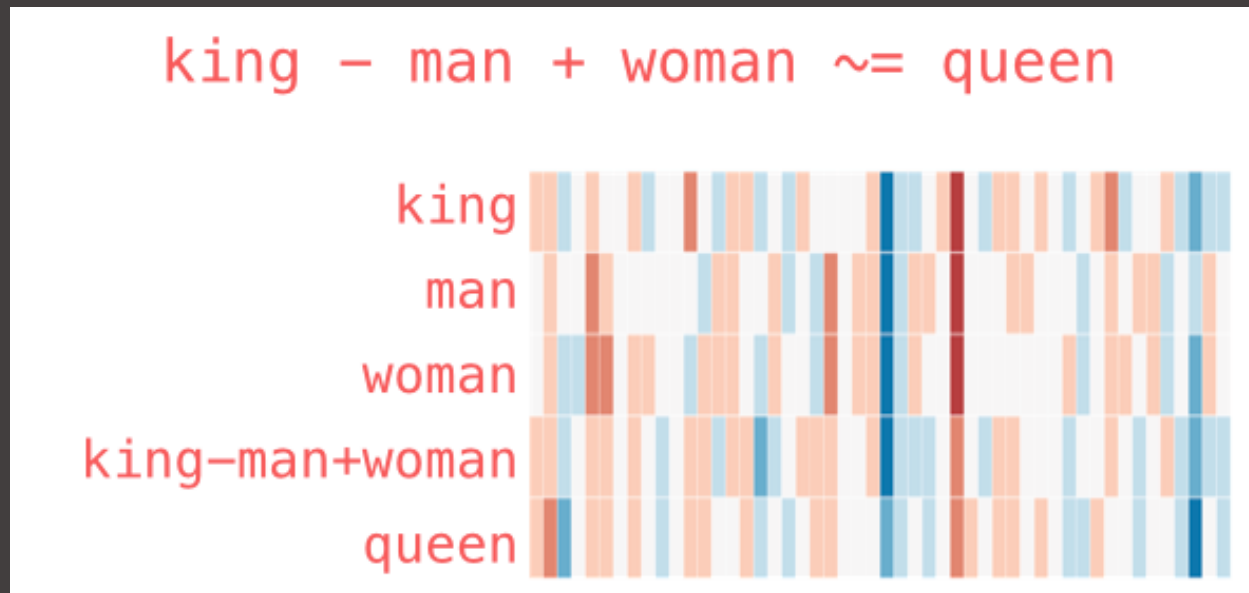
Item2Vec

Item2Vec

- Rozwinięcie algorytmu Word2Vec
 - Oryginalna publikacja: <https://arxiv.org/abs/1301.3781>
 - Przystępne omówienie: <https://jalammar.github.io/illustrated-word2vec/>

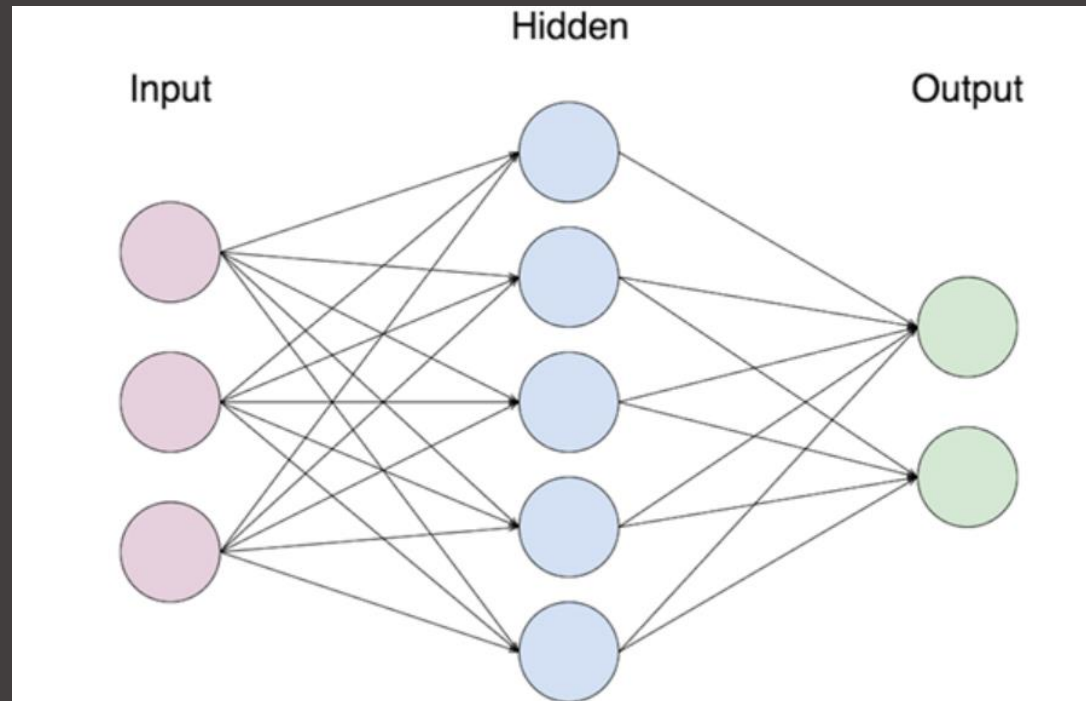
Word2Vec

- Algorytm ma na celu taką wektorową reprezentację słów (lub innych elementów), by operacje algebraiczne na wektorach odzwierciedlały relacje między elementami



Word2Vec - architektura

- Elementy (np. słowa) enkodowane są za pomocą *one-hot encoding* – każdemu elementowi odpowiada jeden neuron wejściowy (oraz jeden neuron wyjściowy)



Word2Vec – continuous bag of words

- Dla każdego elementu w sekwencji (np. słowa w zdaniu) bierzemy n elementów poprzedzających i m elementów następujących po nim
- Na wejściu aktywujemy neurony odpowiadające elementom z otoczenia, na wyjściu oczekujemy aktywacji neuronu odpowiadającego rozważanemu elementowi

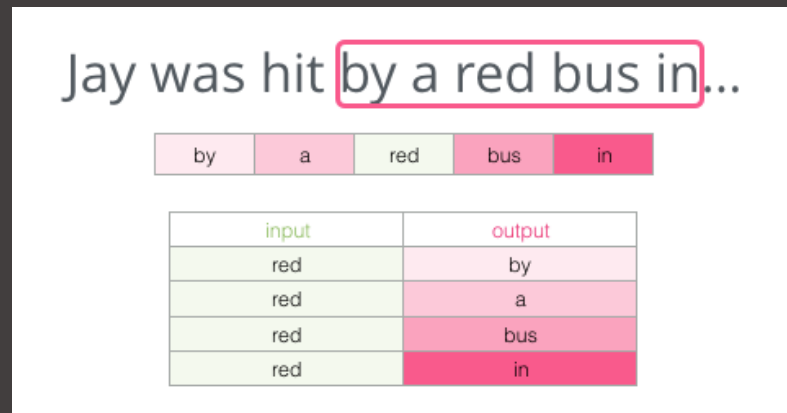
Jay was hit by a _____ bus in...

by	a	red	bus	in
----	---	-----	-----	----

input 1	input 2	input 3	input 4	output
by	a	bus	in	red

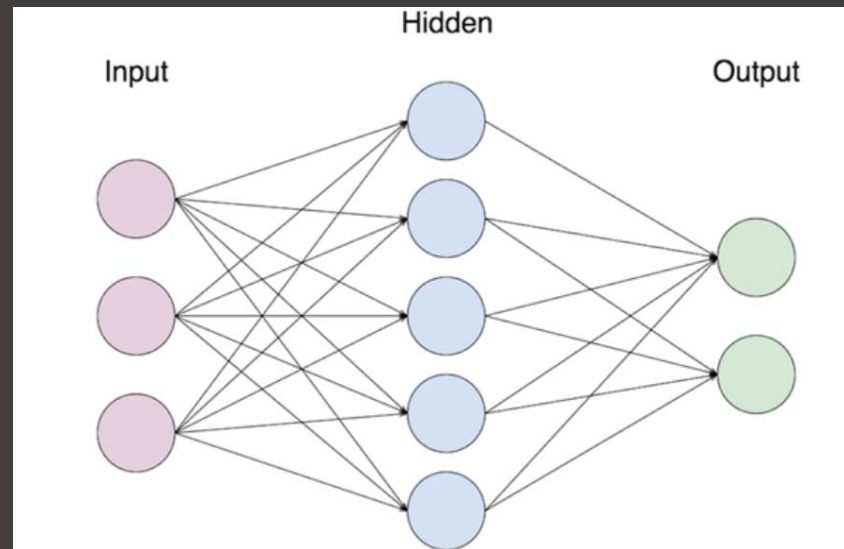
Word2Vec – skipgram

- Dla każdego elementu w sekwencji (np. słowa w zdaniu) bierzemy n elementów poprzedzających i m elementów następujących po nim
- Na wejściu aktywujemy jeden neuron z otoczenia elementu, na wyjściu oczekujemy aktywacji neuronów odpowiadających rozważanemu elementowi



Word2Vec - trening

- Przeprowadzamy klasyczny trening sieci z użyciem backpropagation
- "Odcinamy" warstwę wyjściową, embeddingi odpowiadają aktywacji dwóch pierwszych warstw wytrenowanej sieci



Od Word2Vec do Item2Vec

- Elementami ("słowami"), które będziemy osadzać w przestrzeni wektorowej są unikalne identyfikatory elementów i użytkowników
- "Zdaniami" używanymi do generowania aktywacji sieci jest historia każdego użytkownika (lista elementów, które np. kliknął)
- W wyniku treningu sieci uzyskamy embeddingi elementów (np. filmów albo artykułów); embedding użytkownika to np. średnia z embeddingów elementów, które kliknął
- Warto rozważyć, czy kolejność w historii użytkownika jest ważna

Inne metody

Inne metody obliczania embeddingów

- Embedding tekstów
 - BERT
 - Bag of words (TF/IDF)
- Złożone aktywności i zależności między elementami – graph embeddings

Clustering

Od *embeddingu* do segmentacji

- Każdy użytkownik (i każdy element) mają przypisaną jakąś wartość wektorową
- W jaki sposób możemy ustalić, które elementy są do siebie podobne, którzy użytkownicy będą zainteresowani zbliżonymi treściami?

Odległości pomiędzy wektorami

- Metryki Minkowskiego
 - Odległość euklidesowa
 - Metryka taksówkowa
- Odległość cosinusowa

K-means

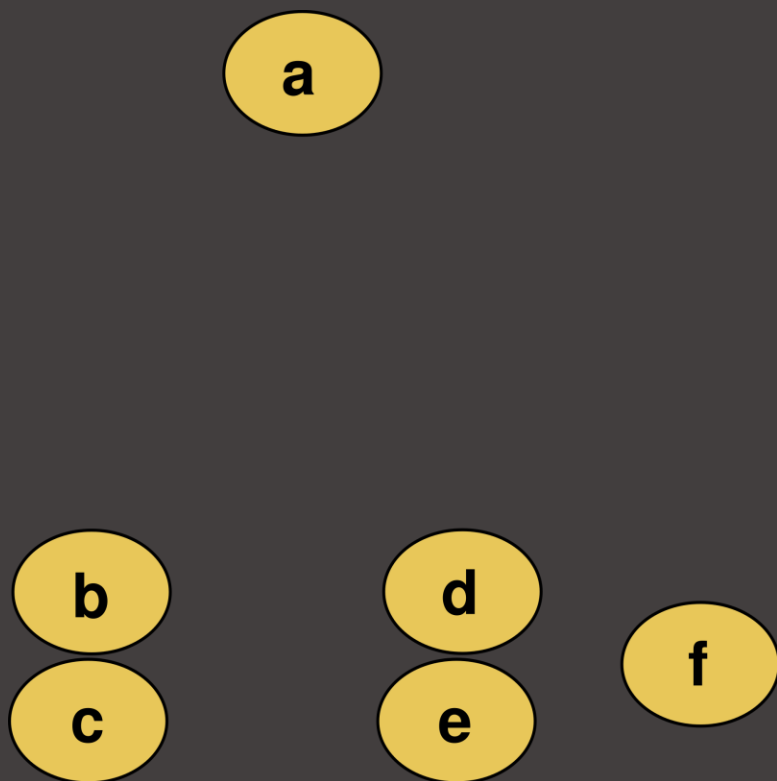
- Podziel zbiór losowo na k klastrów
- W pętli:
 - Oblicz średnią dla każdego klastra
 - Przypisz każdy element do tego klastra, którego średnia jest "najbliżej"
 - Powtarzaj, dopóki wariancja elementów w każdym z klastrów nie spadnie poniżej zakładanego poziomu

Agglomerative clustering

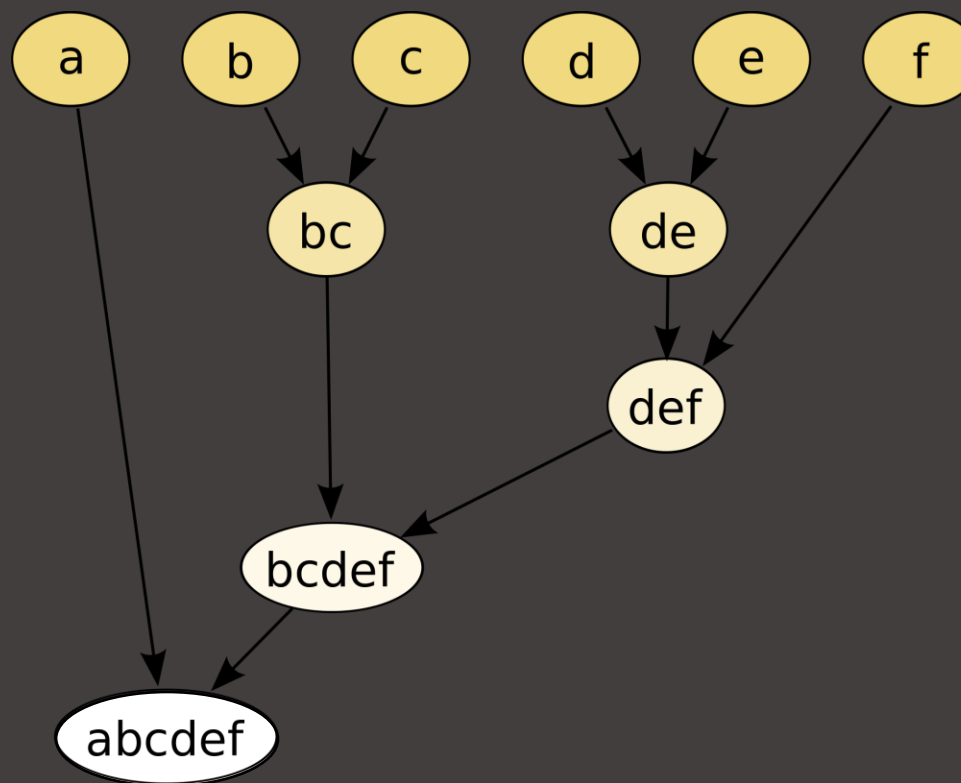
- Na początku każdy element należy do własnego jednoelementowego klastra
- W każdej iteracji kojarzymy dwa najbliższe sobie klastry w jeden większy
 - ale nie usuwamy klastrów "składowych"
- Wynikowe drzewo klastrów możemy "przyciąć" na dowolnej głębokości

Agglomerative clustering

Punkty



Klastry



Odległości między klastrami

- największa odległość między elementami

$$\max\{ d(x, y) : x \in \mathcal{A}, y \in \mathcal{B} \}.$$

- najmniejsza odległość między elementami

$$\min\{ d(x, y) : x \in \mathcal{A}, y \in \mathcal{B} \}.$$

- średnia odległość między elementami

$$\frac{1}{|\mathcal{A}| \cdot |\mathcal{B}|} \sum_{x \in \mathcal{A}} \sum_{y \in \mathcal{B}} d(x, y).$$

Postprocessing

Obliczyliśmy segmenty – i co dalej?

- Jeśli policzyliśmy je dobrze – zapewne będą działać
- Jeśli policzyliśmy je źle – zapewne nigdy się tego nie dowiemy
- Bardzo trudno przekonać kogokolwiek, by zgodził się zapłacić za magiczne czarne pudełko

Czym dokładnie zainteresowani są użytkownicy z segmentu X?

- Dla każdego segmentu możemy policzyć statystyki - liczbę artykułów oznaczonych każdym z tagów
- Na podstawie statystyk możemy obliczyć "zainteresowania" segmentów:
 - Najpopularniejsze tagi w każdym segmencie dobrze opiszą zainteresowania, ale:
 - Niektóre tagi są popularne niezależnie od segmentu
 - Opisy będą mało zróżnicowane
 - Zamiast tego, możemy np. wyznaczyć tagi najbardziej charakterystyczne dla segmentu (np. z największą odchyłką od średniej)

Podsumowanie

- Po co segmentować użytkowników?
- Metaalgorytm
 - Osadzenia
 - Clustering
 - Generowanie opisów
- Przykłady algorytmów dla poszczególnych etapów

Do poduszki

<https://tech.ringieraxelspringer.com/tag/the-one-with-all-the-personalisation-stories>



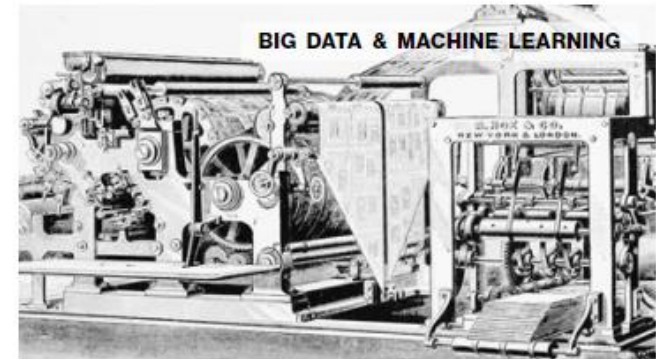
Harnessing Crowd Wisdom - Building A Scalable, Trend-Responsive User Segmentation System

In the previous post, we discussed WHY we opted for segmentation as the basis for our recommender system. In this article, we want to pre...



Why Choose User Segmentation for Your Recommender System in 2020?

We personalize content for 30 million users of Onet and other Ringier Axel Springer ventures in Europe. Here is a comprehensive and techn...



From a Steam-Powered Printing Press to the Era of Personalized Digital Publishing

Media companies must evolve to keep up with the hyper-engaging nature of social media. That's why we have built a state-of-the-art person...