Segmentacja użytkowników

Systemy Rekomendacyjne 2024/2025

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
  - Jedyny stosunkowo prosty krok, ale i tak trzeba zdefiniować kryterium przynależności do segmentu

Meta-algorytm segmentacji

- Oblicz reprezentacje (*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)
- Dekompozycja macierzy ocen R na dwie macierze:
  I – macierz z reprezentacjami elementów
  U – macierz z reprezentacjami użytkowników
  $R \simeq I \times U$

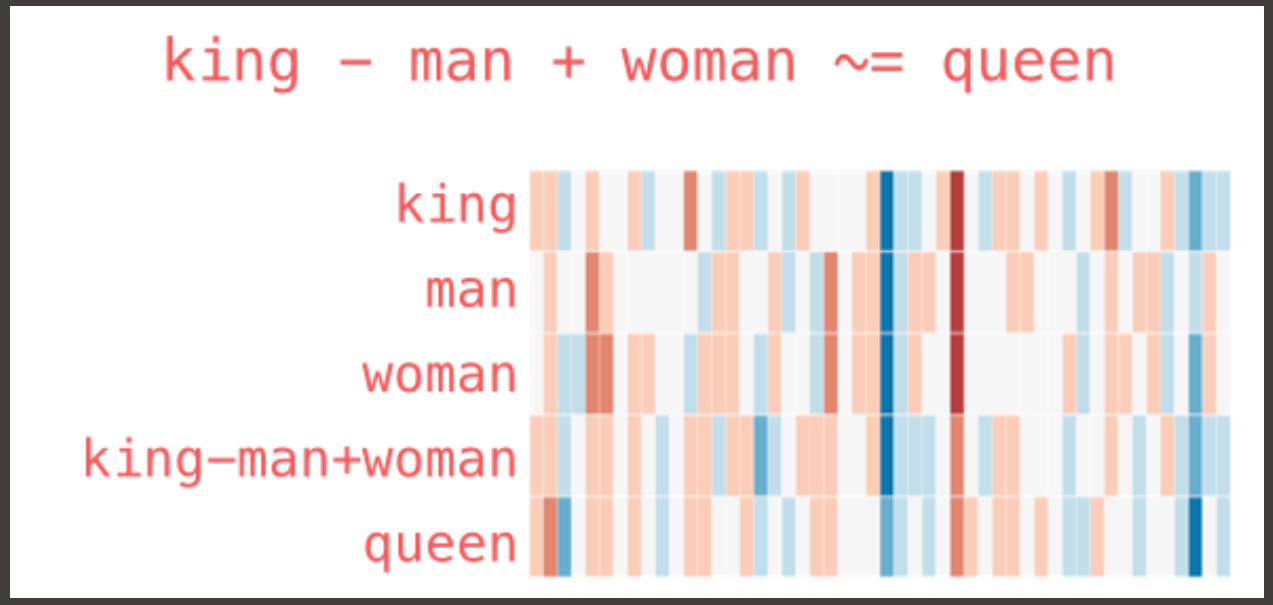
Item2Vec

Item2Vec

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

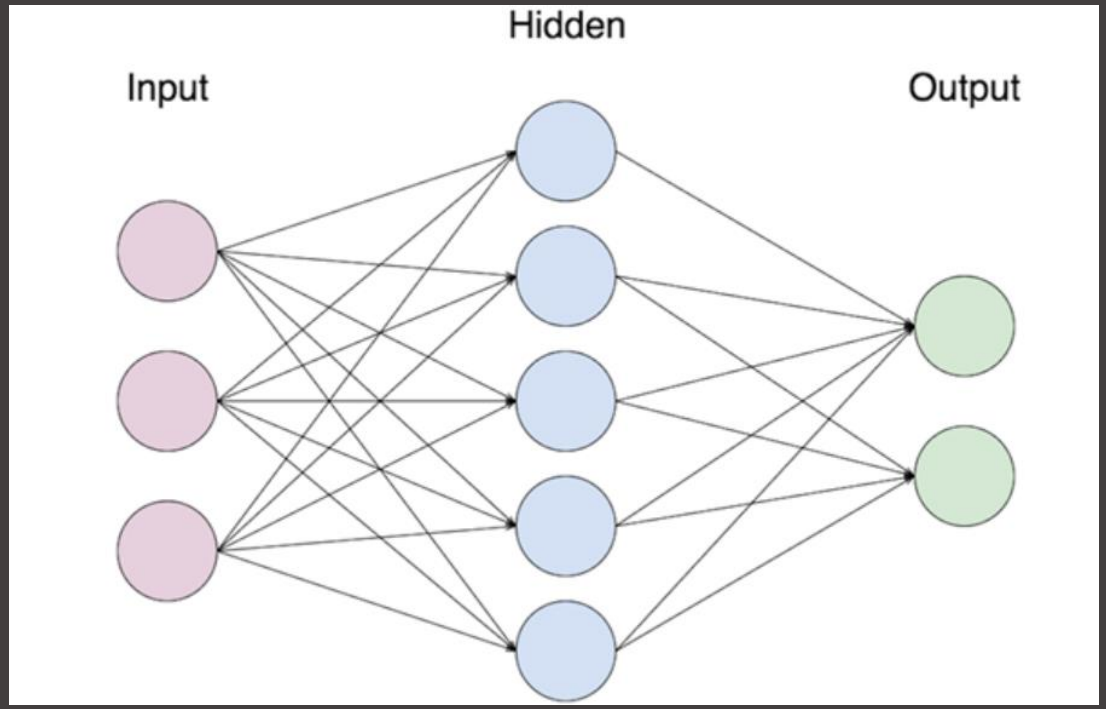
Word2Vec

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



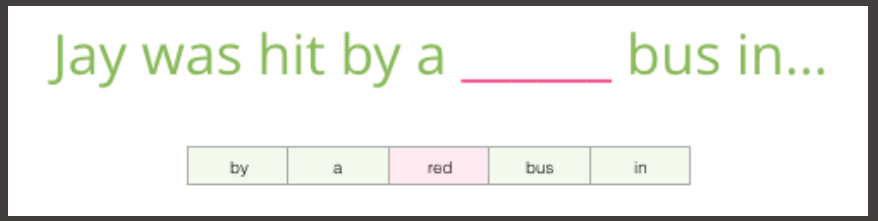
Word2Vec - architektura

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



Word2Vec – continous bag of words

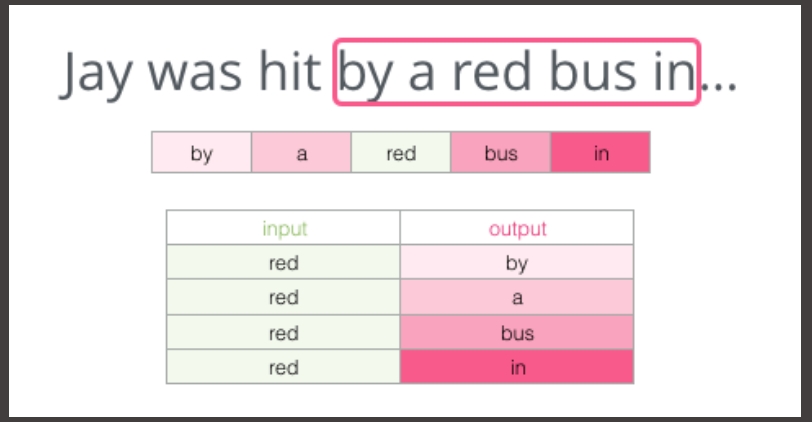
- Dla każdego elementu w sekwencji (np. słowa w zdaniu) bierzemy *n* elementów poprzedzających i *m* elementów następujących po nim
- Na wejściu aktywujemy neurony odpowiadające elementom z otoczenia, na wyjściu oczekujemy aktywacji neuronu odpowiadającego rozważanemu elementowi



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

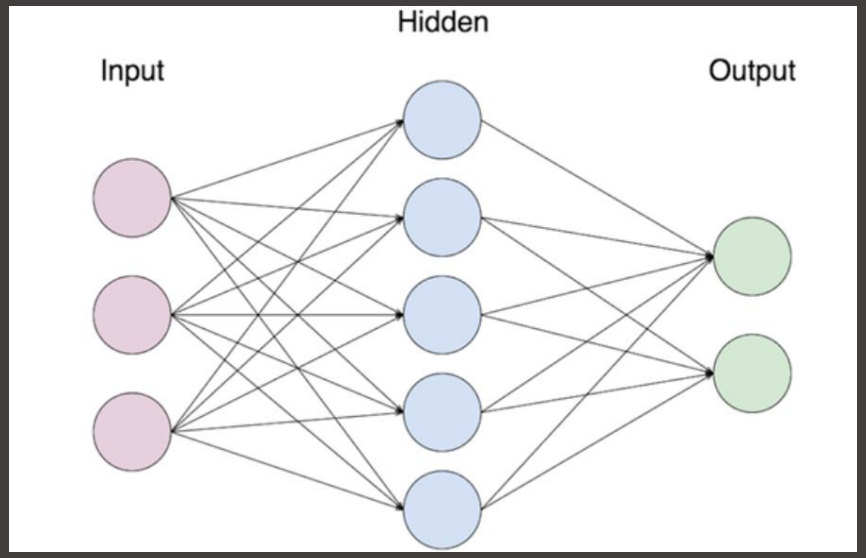
Word2Vec – skipgram

- 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
  - Modele językowe, np. BERT, GPT
  - 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

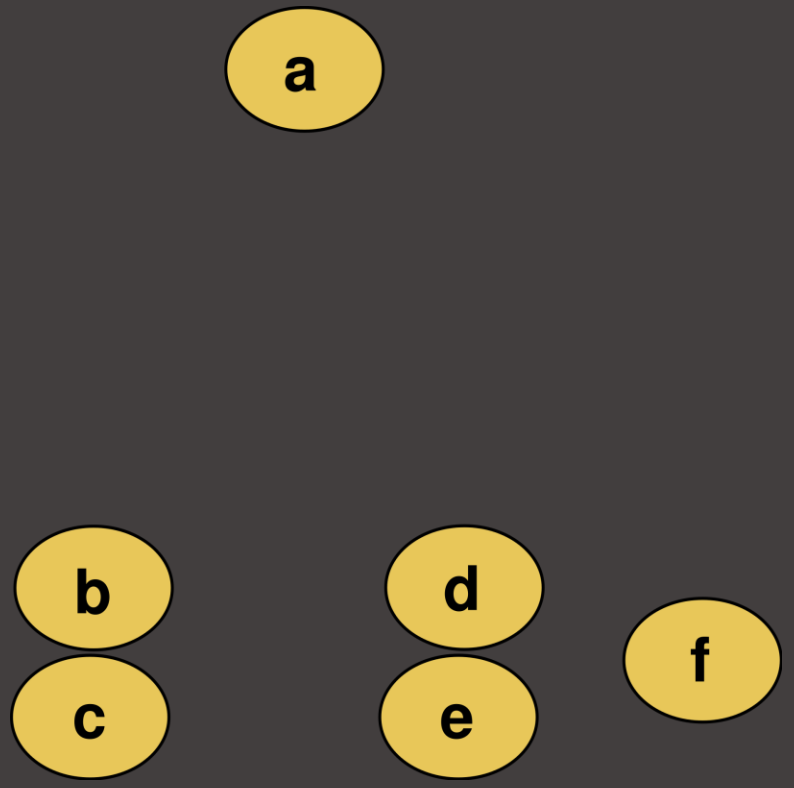
- Podziel zbiór losowo na k klastrów
- W pętli:
  - Oblicz średnią dla każdego klastra
  - Przypisz każdy element do tego klastra, którego średnia jest "najbliżej"
  - Powtarzaj, dopóki wariancja elementów w każdym z klastrów nie spadnie poniżej zakładanego poziomu

Agglomerative clustering

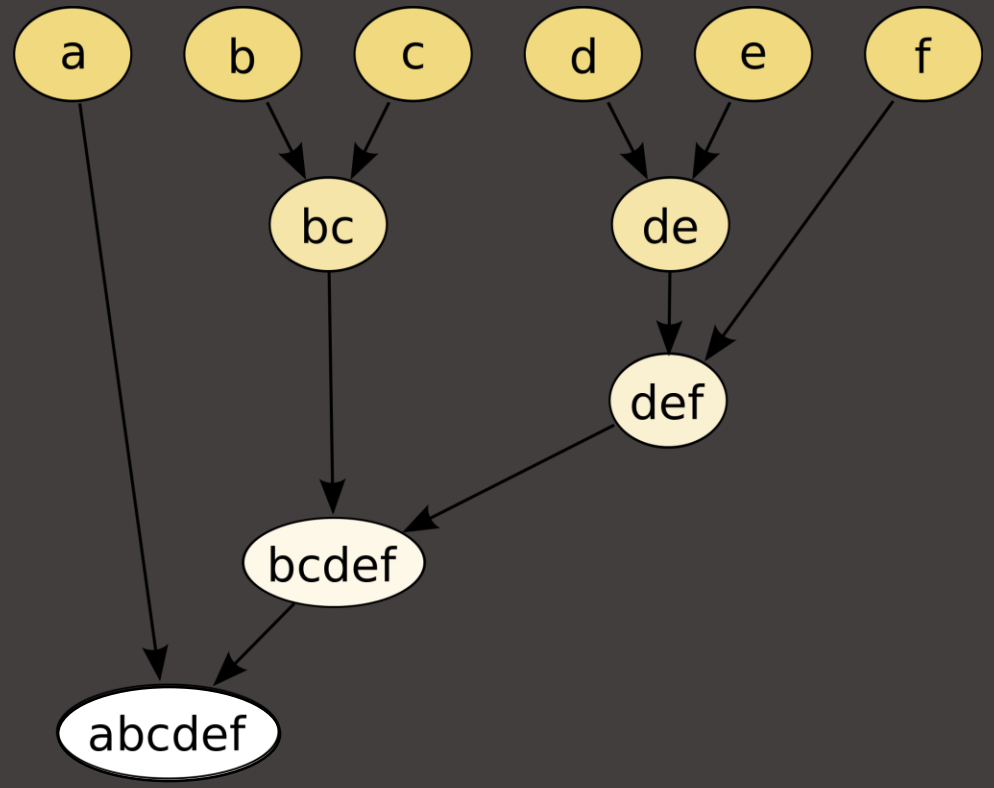
- Na początku każdy element należy do własnego jednoelementowego klastra
- W każdej iteracji kojarzymy dwa najbliższe sobie klastry w jeden większy
  - ale nie usuwamy klastrów "składowych"
- Wynikowe drzewo klastrów możemy "przyciąć" na dowolnej głębokości

Agglomerative clustering

Punkty



Klastry



Źródło: Wikipedia

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