Segmentacja użytkowników

Systemy Rekomendacyjne 2021/2022

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 w miarę proste, ale i tak trzeba zdefiniować kryterium przynależności do segmentu

Metaalgorytm segmentacji

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

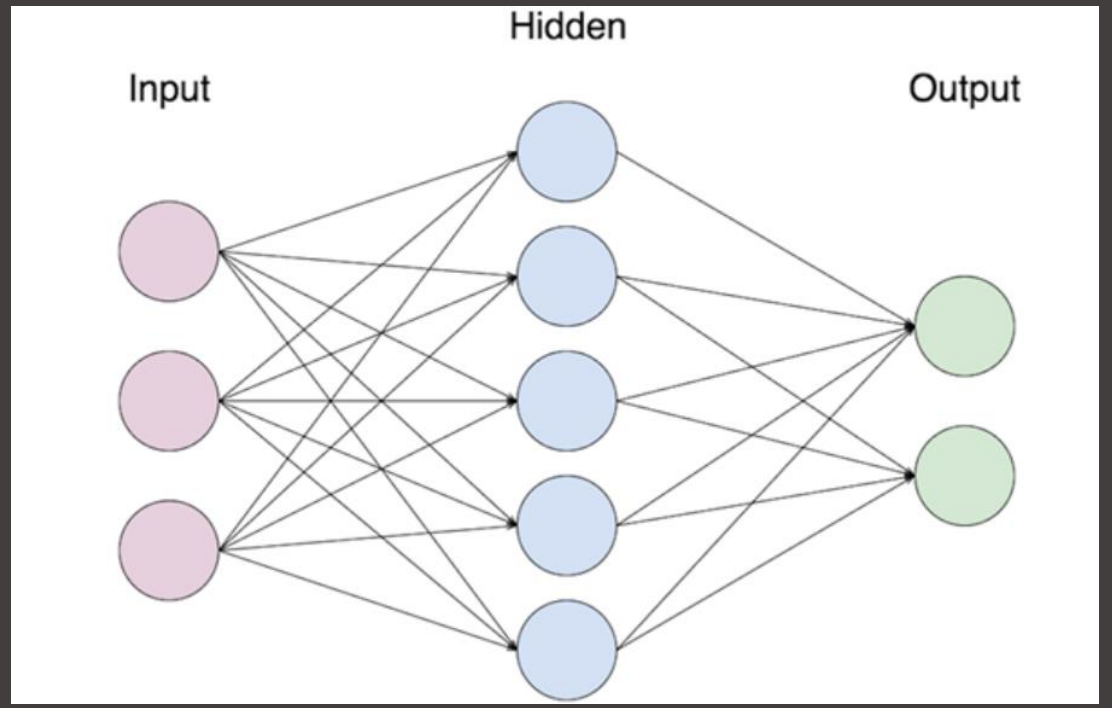
Embeddings

Collaborative filtering

- Algorytm collaborative filtering zwraca już wytrenowany model reprezentacji użytkowników

Item2Vec

- Wejście - lista artykułów przeczytanych przez użytkownika
- Wyjście - osadzenie (wektor reprezentujący użytkownika)
- https://arxiv.org/abs/1301.3781



Clustering

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

BIRCH

- Parametry:
  - liczba klastrów K,
  - maksymalna średnica klastra L
- Dodajemy elementy (użytkowników) kolejno do najbardziej pasującego klastra:
  - Jeśli nowy element leży bliżej średniej niż L – dodajemy go do segmentu
  - Jeśli nowy element leży dalej od średniej niż L – tworzymy z niego nowy klaster, a oba klastry mają wspólnego rodzica
    - Dodatkowo, jeśli przekroczymy liczbę dozwolonych klastrów N – dzielimy klaster i rozdzielamy elementy do innych

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)

TF-IDF

TF-IDF

$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

TF-IDF

Term x within document y

$tf_{x,y}$ = frequency of x in y

df_x = number of documents containing x

N = total number of documents

Segmentacja oparta na treściach

- Obliczamy wektor TD-IDF dla każdego tekstu w korpusie
- Szukamy klastrów tworzonych przez wektory – to będą nasze segmenty
- Sumujemy wektory z każdego segmentu – to będą wektory cech segmentów
- Tworzymy wektory cech dla użytkowników - sumujemy wektory wszystkich tekstów, które przeczytali
- Dla każdego użytkownika znajdujemy najpodobniejszy segment
  - Odległość cosinusowa lub euklidesowa mogą dać dobre rezultaty
- Dodatkowo – obliczamy statystyki i opisy segmentów

Podsumowanie

- Po co segmentować użytkowników?
- Metaalgorytm
  - Osadzenia
  - Clustering
  - Generowanie opisów
- Przykłady algorytmów
- Segmentacja oparta na treści

Do poduszki

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