AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE

Przetwarzanie Języka Naturalnego
Lab 8 – LSA, LDA

Zbigniew Kaleta
`zkaleta@agh.edu.pl`

Wydział IEiT
Katedra Informatyki

06.05.2015

Model wektorowy tekstu

- każda składowa wektora to częstotliwość (lub waga) odpowiadającego jej słowa w danym tekście
- „bag of words”: nie uwzględniamy kolejności występowania wyrazów

- Wady:
  - wyrazy występujące tylko w jednym tekście niosą z sobą dużo informacji (vide prawo Zipfa), ale nie mają dużego wpływu na podobieństwo (vide np. metryka cosinusowa)
  - wyrazy występujące często nie niosą ze sobą informacji, a mają mocny wpływ na pozorne podobieństwo
  - zgodność tekstów sprowadza się do zgodności wyrazów

Poprawki do modelu wektorowego tekstu

- usuwamy *hapax legomena*
- usuwamy wyrazy, które występują w więcej niż 70% tekstów
- w macierzy term-document wartość w danej komórce zawiera wagę danego wyrazu w danym tekście

Latent Semantic Analysis (LSA)

- czasem nazywane Latent Semantic Indexing (LSI)
- metoda analizy podobieństwa między dokumentami i wyrazami oparta na tworzeniu zbioru *pojęć* (*concepts*)
- założenie: słowa bliskoznaczne pojawiają się w podobnych fragmentach tekstu
- rozkład macierzy term-document przy pomocy dekompozycji głównych składowych (ang. *singular value decomposition* - SVD)
- zmniejszona zostaje liczba wierszy (wyrazów) przy zachowaniu podobieństwa między kolumnami (dokumentami)

Latent Semantic Analysis (LSA)

A – macierz term-document o wymiarach $n \times m$
$A = U\Sigma V^T$

U – macierz pojęć o wymiarach $n \times l$ (wektory własne)
Σ – przekątniowa macierz wartości własnych o wymiarach $l \times l$
V – macierz dokumentów o wymiarach $m \times l$ (wektory własne)

Wymiary nowego układu współrzędnych wyznaczonego przez wektory własne to *pojęcia* lub *tematy* (*concepts*, *topics*).

Latent Semantic Analysis (LSA)

Wybierając k największych wartości własnych dokonujemy redukcji wymiarów:
$A' = U'\Sigma'V'^T$

Wymiary macierzy U', Σ', V' to teraz kolejno: $n \times k$, $k \times k$, $m \times k$
Możemy teraz porównywać wyrazy i dokumenty w przestrzeni o mniejszej liczbie wymiarów.

Zalety:

- oszczędność reprezentacji (k jest często rzędu setek)
- zwiększona skuteczność (usuwany jest szum)

Latent Semantic Analysis (LSA)

Wady:

- pojęcia, jako wektory, często nie mają zrozumiałej dla człowieka postaci (niejasne komponenty powstałe przy redukcji wymiarów, ujemne wartości wag, etc.), na przykład:
  - $[(auto), (motor), (kwiat)] \rightarrow [(1.34 * auto + 0.28 * motor), (kwiat)]$
  - $[(auto), (butelka), (kwiat)] \rightarrow [(1.34 * auto + 0.28 * butelka), (kwiat)]$
- ponieważ waga dla każdego słowa to pewien punkt w przestrzeni, LSA jest nieczułe na polisemię
- konsekwencje wynikające z użycia modelu „bag of words”

Latent Dirichlet Allocation (LDA)

- dokument to „mieszanka” tematów, gdzie każdy temat to zbiór słów z przypisanymi im prawdopodobieństwami
- wektor tematów dla każdego dokumentu jest bardziej zrozumiały niż w przypadku LSA
- prezentowany jest rozkład tematów w dokumencie i rozkład słów w temacie (zgodnie z rozkładem Dirichleta)

Latent Dirichlet Allocation (LDA)

Przykładowy algorytm:

1. dla każdego dokumentu każdemu słowu losowo przypisz jeden z K tematów
2. dla każdego słowa w w każdym dokumencie d:
   1. dla każdego tematu t oblicz: $P(t|d)$ i $P(w|t)$
   2. przypisz słowu w nowy temat z prawdopodobieństwem $P(t|d) * P(w|t)$

Po powtórzeniu tych kroków wiele razy otrzymujemy całkiem dobry rozkład tematów w dokumentach.

Zadania

1. zbudować modele LSA i LDA przy użyciu np. biblioteki gensim (`http://radimrehurek.com/gensim/tutorial.html`). Proszę pamiętać o sprowadzeniu wyrazów do formy podstawowej (1.5 pkt.)
2. napisać program, który dla każdej notatki wypisać najistotniejsze tematy (wg. LSA i LDA) (0.5 pkt.)
3. napisać program, znajdujący najbardziej zbliżone notatki do zadanej (1 pkt.)

Materiały:
`http://home.agh.edu.pl/~zkaleta/pjn/lab6.tar.gz`