

Word Mover's Distance

Ксения Вальчук, Анастасия Царькова

НИУ ВШЭ

21 марта 2017 г.

Рассмотрим задачу определения степени схожести текстов.

Примеры применения:

- Классификация текстов (новостей)
- Поисковое ранжирование
- Рекомендации книг

Способы представления документов. BOW

- Bag of words(BOW) – упрощенное представление текста без учета грамматики и порядка слов, но с сохранением кратности.

Пример: John likes to watch movies and to watch tv. – ['John', 'likes', 'watch', 'movies', 'tv']

BOW = [1, 1, 2, 1, 1]

Способы представления документов. TF-IDF

- TF-IDF (term frequency-inverse document frequency) – статистическая мера для оценки важности слова в контексте документа, из некоторой коллекции документов D .

Для каждой пары (слово, текст) (t, d) вычисляется величина

$$\text{tf-idf}(t, d, D) = \text{tf}(t, d) \cdot \text{idf}(t, D)$$

, где

$$\text{tf}(t, d) = \frac{n_{td}}{\sum_{t \in d} n_{td}}$$

– количество вхождений слова t в отношении к общему числу слов,

$$\text{idf}(t, D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

– инверсия частоты, с которой слово встречается в документах коллекции.

Модель, которая сопоставляет каждому слову его векторное представление путем максимизации логарифма вероятности нахождения слова рядом со своими соседями

$$\max \frac{1}{T} \sum_{t=1}^T \sum_{j \in nb(t)} \log p(w_j | w_t)$$

- Имеет простую архитектуру
- Способна быстро обучаться на огромных датасетах
- Обучается сложным зависимостям

Пример: $\text{vec}(\text{Einstein}) - \text{vec}(\text{scientist}) + \text{vec}(\text{Picasso}) \approx \text{vec}(\text{painter})$

Пусть есть текст, который состоит из n уникальных слов. Представим его в виде нормированного мешка слов (nBOW).

$$d \in \mathbb{R}^n, \quad d_i = \frac{c_i}{\sum_{j=1}^n c_j}$$

где c_i – количество раз, которое слово i встречается в данном тексте. Так как d – нормирован, представим его как точку в $(n - 1)$ мерном симплексе.

Рассмотрим пример предложений, которые не имеют общих слов, но подразумевают схожую информацию:

- D = Obama speaks to the media in Illinois.
- D' = The President greets the press in Chicago.

$w = ['Obama', 'speaks', 'media', 'Illinois', 'President', 'greet', 'press', 'Chicago']$, тогда получаем мешки слов

$$d = [1/8, 1/8, 1/8, 1/8, 0, 0, 0, 0]$$

$$d' = [0, 0, 0, 0, 1/8, 1/8, 1/8, 1/8]$$

Word Mover's Distance

WMD определяет расстояние между двумя документами как оптимальную стоимость перемещения слов из одного документа в другой с помощью векторного представления слов.

Пусть у нас есть матрица $X \in \mathbb{R}^{d \times n}$, где $x_i \in \mathbb{R}^d$ – векторное представление i -ого слова конечной длины.

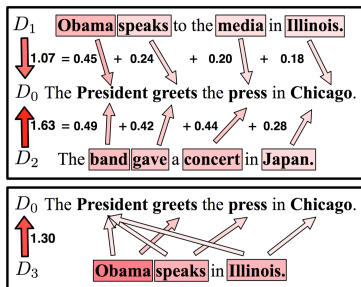
Тогда 'семантическое' расстояние между словами i и j определим как

$$c(i, j) = \|x_i - x_j\|_2$$

Flow matrix

Пусть у нас есть два текста D и D' с нормированными мешками слов d , d' соответственно.

Определим матрицу перемещений $T \in \mathbb{R}^{n \times n}$, где $T_{ij} \geq 0$ – 'количество' слова i из d которое переходит в слово j из d' . При этом должно выполняться $\sum_j T_{ij} = d_i$ и $\sum_i T_{ij} = d'_j$



Оптимизационная задача

Таким образом мы можем определить расстояние между текстами как минимальную совокупную стоимость перехода требуемую для того чтобы перевести d в d' .

Формально данная задача минимизации стоимости перехода из d в d' записывается следующим образом:

$$\min_{T \geq 0} \sum_{i,j} T_{ij} \|x_i - x_j\|_2$$

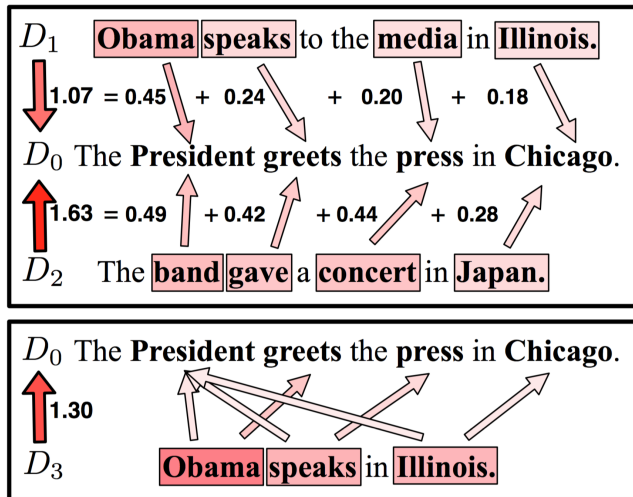
$$\sum_{j=1}^n T_{ij} = d_i \quad \forall i \in 1, \dots, n$$

$$\sum_{i=1}^n T_{ij} = d'_j \quad \forall j \in 1, \dots, n$$

Свойства метрики:

- Не имеет гиперпараметров, проста для понимания и использования
- Демонстрирует высокие показатели метрики accuracy на реальных задачах классификации текстов

Пример работы метрики



Решение данной задачи оптимизации имеет сложность $O(p^3 \log p)$, где p – количество различных слов в тексте. В больших датасетах количество уникальных слов велико – что делает вычисление метрики долгим.

Поэтому надо как то оценивать WMD, что бы уменьшить сложность алгоритма.

Попробуем оценить значение метрики снизу:

$$\begin{aligned}\sum_{i,j=1}^n T_{ij} c(i,j) &= \sum_{i,j=1}^n T_{ij} \|x_i - x'_j\|_2 = \sum_{i,j=1}^n \|T_{ij}(x_i - x'_j)\|_2 \geq \\ &\geq \left\| \sum_{i,j=1}^n T_{ij}(x_i - x'_j) \right\|_2 = \left\| \sum_{i=1}^n \left(\sum_{j=1}^n T_{ij} \right) x_i - \sum_{j=1}^n \left(\sum_{i=1}^n T_{ij} \right) x'_j \right\|_2 = \\ &= \left\| \sum_{i=1}^n d_i x_i - \sum_{j=1}^n d_j x'_j \right\|_2 = \|Xd - Xd'\|_2\end{aligned}$$

Сложность вычисления данной метрики $O(dp)$, где d – размерность пространства word2vec.

Relaxed word moving distance

Несмотря на то, что WCD можно посчитать эффективно по времени, она не очень хорошо оценивает нашу метрику.

Ослабим условия задачи

$$\min_{T \geq 0} \sum_{i,j} T_{ij} c(i,j)$$

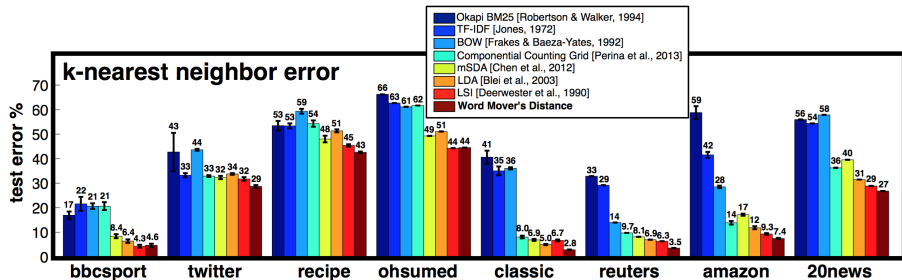
$$\sum_{j=1}^n T_{ij} = d_i \quad \forall i \in 1, \dots, n$$

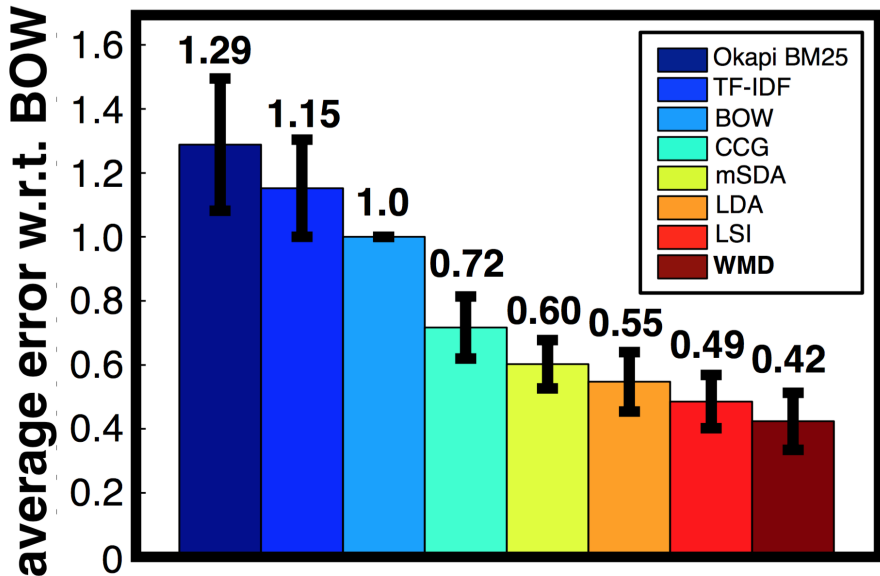
Решение полученной задачи оптимизации:

$$T_{ij}^* = \begin{cases} d_i & \text{if } j = \operatorname{argmin}_j c(i,j) \\ 0, & \text{otherwise} \end{cases}$$

$$\sum_j T_{ij} c(i,j) \geq \sum_j T_{ij} c(i,j^*) = c(i,j^*) \sum_j T_{ij} = c(i,j^*) d_i = \sum_j T_{ij}^* c(i,j)$$

Классификация документов

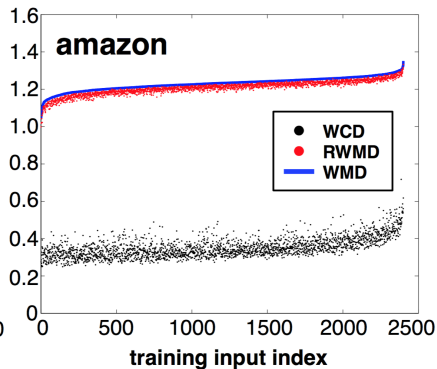
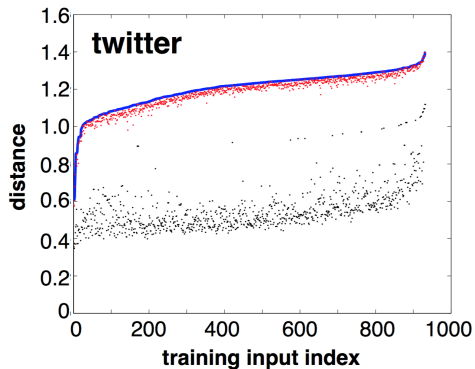




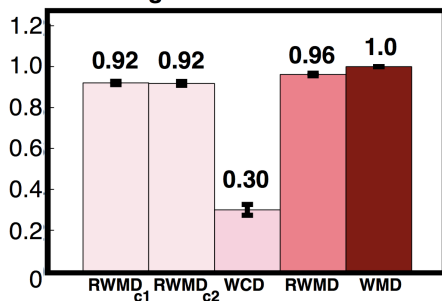
Алгоритм:

- Сортируем все документы по возрастанию метрики WCD
- Вычисляем WMD для первых k документов из предыдущего шага
- Проходимся по каждому документу с k -го по $k + m$. Если RWMD данного документа меньше WMD k -го, то добавляем его в k ближних, считаем для него WMD, сортируем все $k + 1$ и отбираем из них лучшие k .
- Повторяем предыдущий шаг, пока не перестанем находить документ лучший k -го

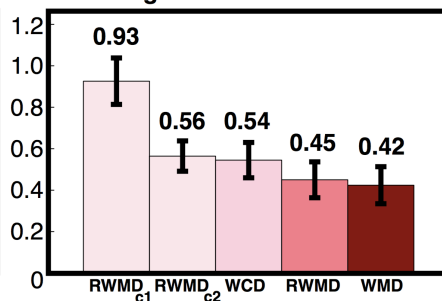
Оптимизация kNN



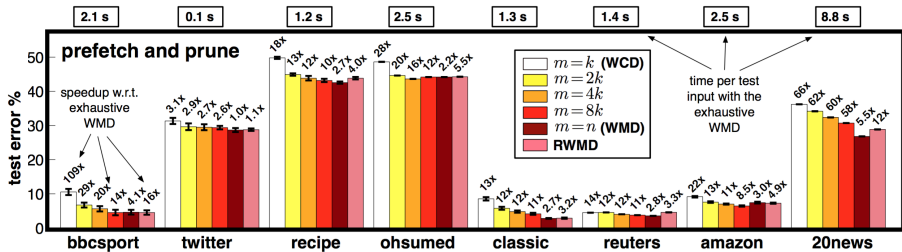
average distance w.r.t. WMD



average kNN error w.r.t. BOW



Оптимизация kNN



Добавим регуляризатор к оптимизационной задаче:

$$\min_{T \geq 0} \sum_{i,j} T_{ij} \|x_i - x_j\|_2 - \frac{1}{\lambda} h(T)$$

$$\sum_{j=1}^n T_{ij} = d_i \quad \forall i \in 1, \dots, n$$

$$\sum_{i=1}^n T_{ij} = d'_j \quad \forall j \in 1, \dots, n$$

Тексты могут быть классифицированы по теме или по автору. Это может привести к достаточно разным измерениям расхожести.

Добавим к матрице X множество меток классов Y .

Обучаем матрицу $A \in \mathbb{R}^{r \times d}$, $r \geq d$

$$d_A(x_i, x_j) = \|A(x_i - x_j)\|_2$$

Спасибо за внимание =)