2. Word Embedding

Word Embedding

Goal: come up with a good representation of text

- Leads to good task performance
- Enables a notion of distance over texts: $d(\phi(a), \phi(b))$ is small for semantically similar texts a and b

Distance functions

Euclidean distance: for $a, b \in \mathbb{R}^d$, $d(a, b) = \sqrt{\sum_{i=1}^{d} (a_i - b_i)^2}$

Cosine similarity: $\mathrm{sim}(a, b) = \frac{a \cdot b}{||a|| ||b||} = \cos \alpha$

Count-based word embeddings

Step 1: Choose the context

- construct a co-occurence matrix (e.g. word * document, word * word)

Step 2: Reweight counts

- TFIDF: term frequency * inverse document frequency:
  $\phi_i(d) = \mathrm{count}(w_i, d) \times \log \frac{\#\text{documents}}{\#\text{documents containing } w_i}$
- Pointwise mutual information:
  $\mathrm{PMI}(x; y) \overset{\text{def}}{=} \log \frac{p(x,y)}{p(x)p(y)} = \log \frac{p(x|y)}{p(x)} = \log \frac{p(y|x)}{p(y)}$

Step 3: Dimensionality reduction: want a lower-dimensional, dense representation for efficiency

- SVD: $A_{m \times n} = U_{m \times n} \Sigma_{m \times n} V_{n \times n}^T$
  choose top-k, each row of $U_{m \times k} \Sigma_k$ corresponds to a word vector of dimension k

Prediction-based word embeddings

Goal: map each word to a vector in $\mathbb{R}^d$ such that similar words have similar word vectors

Intuition: similar words occur in similar contexts

The skip-gram model: given a word, predict its neighboring words within a window

- use softmax to predict the context words w_j from the center word w_i

$$p(w_j | w_i) = \frac{\exp[\phi_{\text{ctx}}(w_j) \cdot \phi_{\text{wrd}}(w_i)]}{\sum_{w \in \mathcal{V}} \exp[\phi_{\text{ctx}}(w_j) \cdot \phi_{\text{wrd}}(w_i)]}$$

Implementation:

- Matrix form: $\phi : w \mapsto A_{d \times |\mathcal{V}|} \phi_{\text{one-hot}}(w)$, ϕ can be implemented as a dictionary
- Learn parameters by MLE and SGD

- ϕ_{wrd} is taken as the word embedding

Negative sampling

Challenge in MLE: computing the normalizer is expensive

Key idea: get some negative samples, solve a binary classification problem instead

$p_\theta(\mathrm{real}|w, c) = \frac{1}{1+e^{-\phi_{\mathrm{ctx}}(c)\cdot\phi_{\mathrm{wrd}}(w)}}$

The continuous BoW model: predict the center word from the context words

$$p(w_j|w_i) = \frac{\exp[\phi_{\mathrm{wrd}}(w_i) \cdot \sum_{w' \in c} \phi_{\mathrm{ctx}}(w')]}{\sum_{w \in \nu} \exp[\phi_{\mathrm{wrd}}(w) \cdot \sum_{w' \in c} \phi_{\mathrm{ctx}}(w')]}$$

Semantic properties of word embeddings

Find similar words: top-k nearest neighbors using cosine similarity