Document Classification by Inversion of Distributed Language Representations

Matt Taddy, Chicago Booth

Distributed language representation

$\boldsymbol{\mathcal{V}}$ contains an embedding in $\mathbb{R}^K$ for every vocabulary word.

In a contextual language model, $\mathcal{V}$ is trained to maximize the likelihoods for each single words and its neighbors.

e.g., The skip-gram objective for word t in sentence s is

$$\max \sum_{j \neq t,\ j=t-b}^{t+b} \log \mathrm{p}_{\mathcal{V}}(w_{sj} \mid w_{st})$$

where b is the skip-gram window (truncate at ends of sentences).

Neural network language models

Local context probabilities are functions of the word embeddings.

e.g., In Word2Vec (Mikolov et al. 2013)

$$\mathrm{p}_{\mathcal{V}}(w|w_t) = \prod_{j=1}^{L(w)-1} \sigma\Big(\mathrm{ch}\left[\eta(w, j+1)\right] \mathbf{u}_{\eta(w,j)}^{\top} \mathbf{v}_{w_t}\Big)$$

where $\eta(w, i)$ is the i^{th} node in the length-$L(w)$ Huffman tree path for w and $\mathrm{ch}(\eta) \in \{-1, +1\}$ for whether η is a left or right child.

'Output' embedding $\mathbf{v}_{w_t}$ is usually the main object of interest.

From word embeddings to document modeling

Distributed representations have proven very useful for NLP tasks
next word prediction, word analogy, named entity recognition, ...

There is interest in porting this success to document modeling
author classification, sentiment prediction, attribute imputation, ...

Strategies include directly modeling the semantic composition of contexts (Socher et al. 2011) or adding latent document-location effects into the context model (as in Le + Mikolov's Doc2Vec).

This paper: composite likelihoods and Bayes rule provide a very simple way to turn local language models into document classifiers.

Composite likelihood

The local-context objectives don't correspond to a full document model, but they can be combined to form a composite likelihood.

e.g., skip-gram's pairwise-conditional composition for sentence **w**

$$\log \mathrm{p}_{\mathcal{V}}(\mathbf{w}) = \sum_{j=1}^{T} \sum_{k=1}^{T} \mathbb{1}_{[1 \leq |k-j| \leq b]} \log \mathrm{p}_{\mathcal{V}}(w_k | w_j).$$

Composite LHD approximate a full joint LHD. They are common in statistics, since Besag's pseudolikelihood $\mathrm{p}(\mathbf{w}) \approx \prod_j \mathrm{p}(w_j | \mathbf{w}_{-j})$.

Another e.g.: Jernite et al. (2015) show that CBOW Word2Vec corresponds to the pseudolikelihood for a Markov random field.

Bayesian inversion

Given sentence LHDs, document $d = \{\mathbf{w}_1, ...\mathbf{w}_S\}$ has log LHD

$$\log \mathrm{p}_{\mathcal{V}}(d) = \sum_s \log \mathrm{p}_{\mathcal{V}}(\mathbf{w}_s).$$

Suppose your documents are grouped by class label, $y \in \{1 \dots C\}$.

Then you train separate $\mathcal{V}_c$ on each sub-corpus $D_c = \{d_i : y_i = c\}$.

$\Rightarrow$ doc d has probability $\mathrm{p}_{\mathcal{V}_c}(d)$ if it came from class c, and

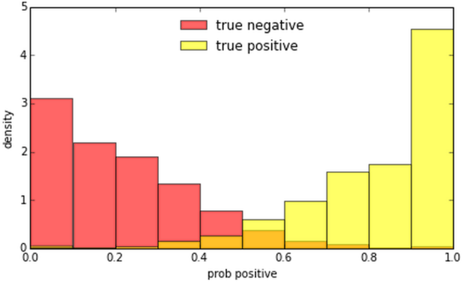
$$\mathrm{p}(y|d) = \frac{\mathrm{p}_{\mathcal{V}_y}(d)\pi_y}{\sum_c \mathrm{p}_{\mathcal{V}_c}(d)\pi_c}$$

where π_c is our prior probability on class label c (say $\pi_c = 1/C$).

Yelp reviews example

200k reviews, 2mil sentences, separate W2V for each of 1-5 stars.

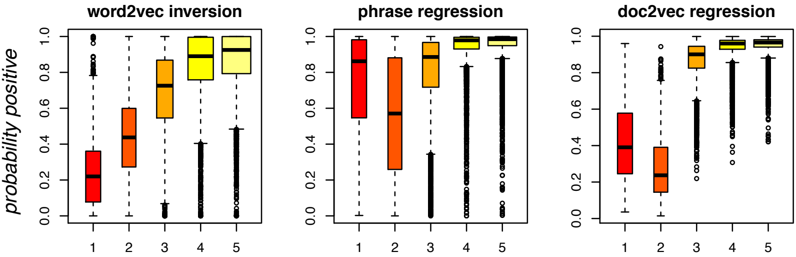
Given W2V representations $\mathcal{V}_1 \dots \mathcal{V}_5$, calculate sentiment probs as, e.g., $\mathrm{p}(\star \geq 3|d) = \left[\mathrm{p}_{\mathcal{V}_3}(d) + \mathrm{p}_{\mathcal{V}_4}(d) + \mathrm{p}_{\mathcal{V}_5}(d)\right] / \sum_{c=1}^{5} \mathrm{p}_{\mathcal{V}_c}(d)$



Everything is implemented in the `gensim` library for `python`, which now includes the `score` method to obtain $\log \mathrm{p}_{\mathcal{V}}(d)$ for fitted $\mathcal{V}$.

OOS classification performance

misclass rate	$<, \geq 3 \star$	$<, =, > 3 \star$	$1 \ldots 5 \star$
W2V inversion	.099	**.189**	.435
Phrase regression	**.084**	.200	**.410**
D2V combined	.148	. 284	.500
MNIR	.095	.254	.480
W2V aggregation	.118	.248	.461



Best or close to it, with prob(*positive*) nicely ordered by true star.

Inversion is simple, scalable, and it works

Not claiming it's a world beater, but it is an easy way to go from a context representation algorithms to document classification.

Future questions...

Can we use what we know about composite LHD to drive context models? e.g., Cox + Reid (2004): $\mathrm{p}(w_j, w_k)$ pref to $\mathrm{p}(w_j|w_k)$.

Given the local → global connection, can we apply distributed representations in new domains? e.g., consumer product choices.

THANKS!