Document Classification by Inversion of Distributed Language Representations

Matt Taddy, Chicago Booth

Distributed language representation

$\mathcal{V}$ 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 word 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.

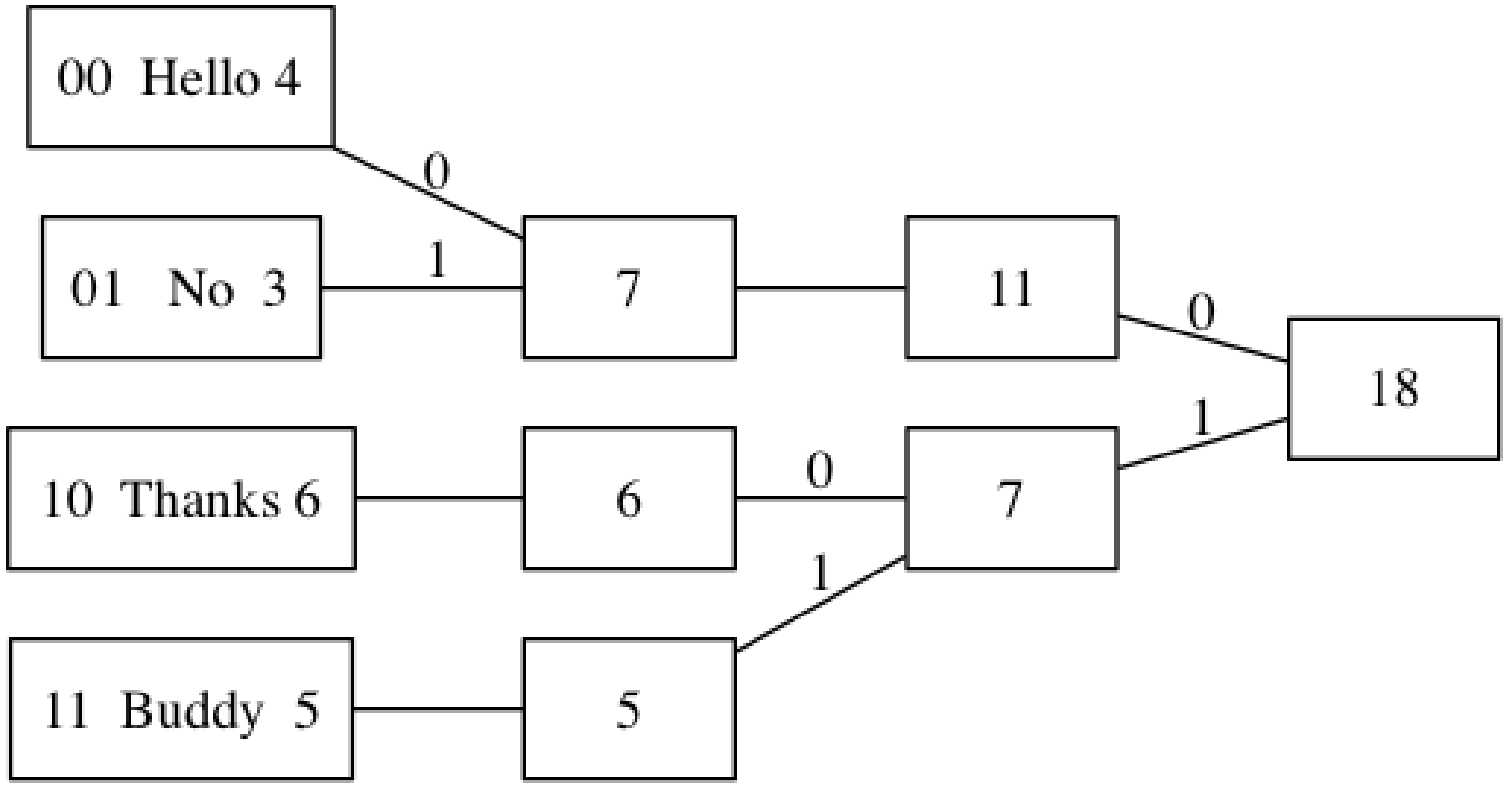
e.g., In Word2Vec (Mikolov et al. 2013)

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

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.

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

Example Huffman encoding of a 4 word vocabulary



From left to right the two nodes with lowest count are combined into a parent. Encodings are read off of the splits from right to left.

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 $\mathbf{w}$

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

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 $d = \{\mathbf{w}_1, \ldots \mathbf{w}_S\}$ 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, $y \in \{1 \ldots C\}$.

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

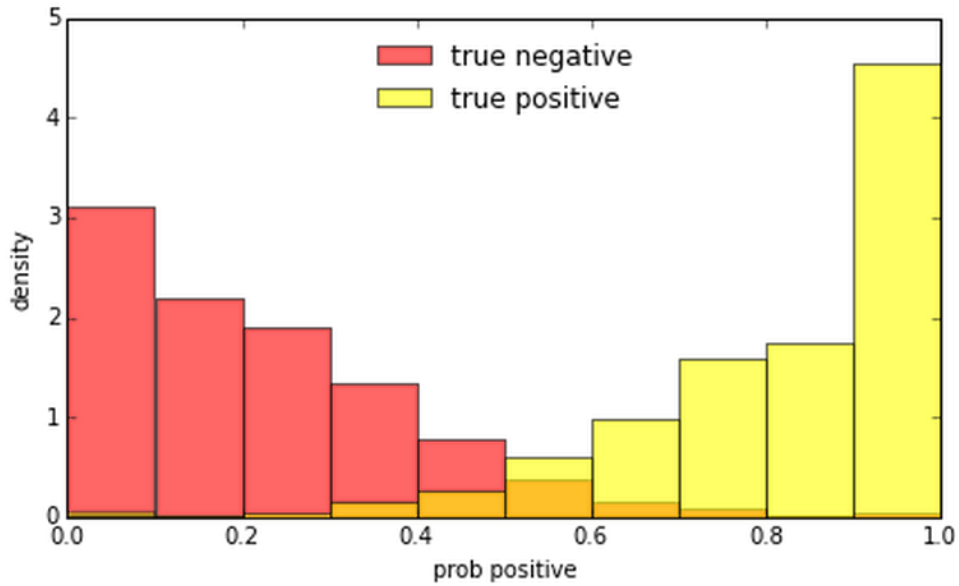
$$\mathrm{p}(y|d) = \frac{\mathrm{p}_{\mathcal{V}_y}(d)\pi_y}{\sum_c \mathrm{p}_{\mathcal{V}_c}(d)\pi_c}$$

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.

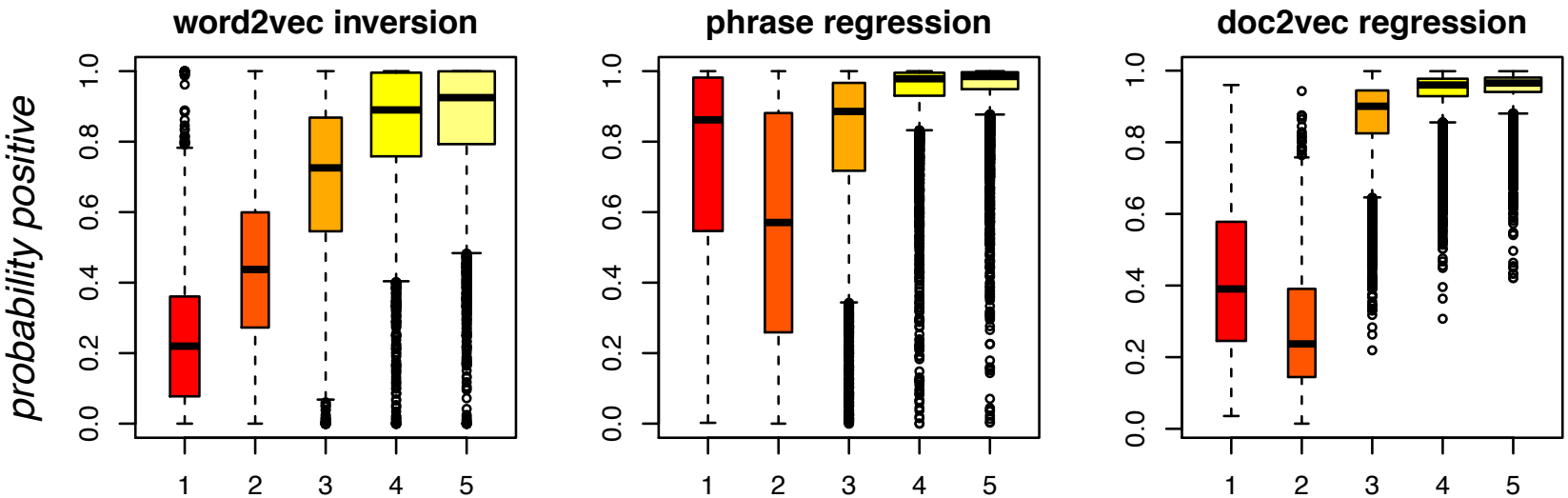
Given W2V representations $\mathcal{V}_1 \dots \mathcal{V}_5$, calculate sentiment probs as, e.g., $\mathrm{p}(\star \geq 3|d) = [\mathrm{p}_{\mathcal{V}_3}(d) + \mathrm{p}_{\mathcal{V}_4}(d) + \mathrm{p}_{\mathcal{V}_5}(d)] \,/ \sum_{c=1}^{5} \mathrm{p}_{\mathcal{V}_c}(d)$



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.

Messaging and Negotiation on eBay.com

eBay has a 'best offer' button; a buyer can use this to circumvent the auction or fixed price sale, and make a direct offer to the seller.

We track the communication.

After controlling for parameters of the transaction (item, buyer offer, seller/buyer info, ...) what language leads to a higher probability of a seller responding with a deal or counteroffer?

We can use the results to educate buyers, give templates/examples, or generate hypotheses on bargaining behavior.

First, fit separate word2vec to those reviews with $y_i = 1$ (seller response) and with $y_i = 0$ (no seller response).

Sentences most likely in the $y = 1$ representation all showed evidence of previous deals and bundling, while those most likely in the $y = 0$ representation are driving a hard bargain:

know this low but its based on what individual [product] selling

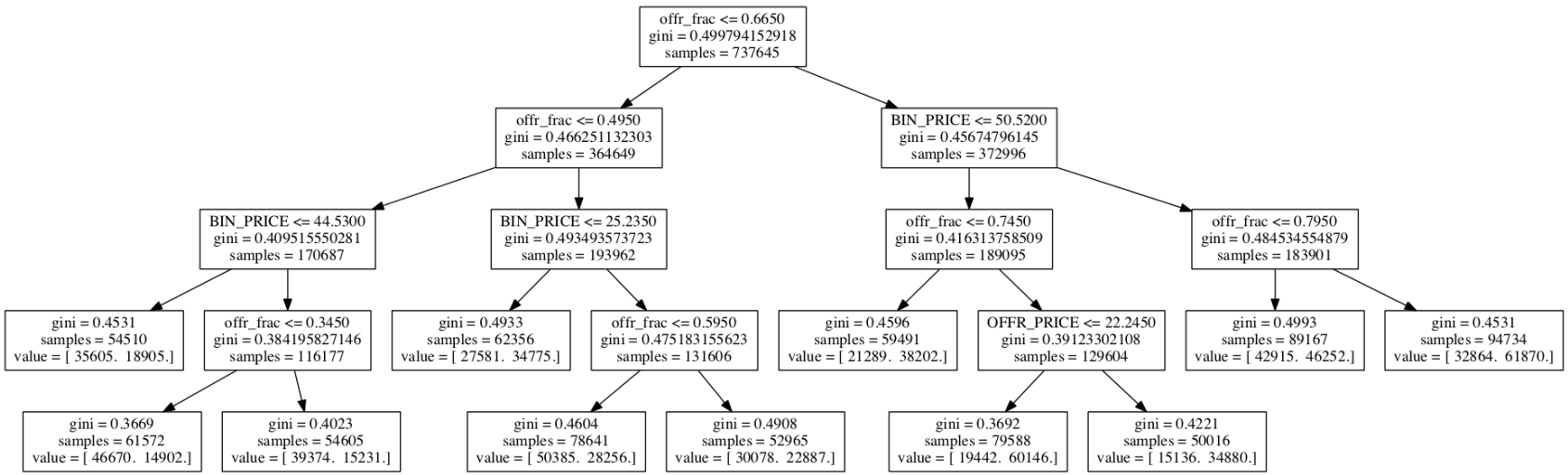
give you two cash you ship it free its very common low end [product] has broken pin good pin youd be lucky get this real its worth melt price dont believe me its really not worth listing fees that you paid

hi my offer basically what price guide lists individual [product] it may seem too low you but it never hurts place offer regards [name]

do you know that clubhouse sigs fakes worthless ridiculous fake sigs thats why other ball sold they were real sigs real sigs lot better its like gehrigs wife signing his name not worth anything

To isolate the targeted effect, we first remove what was predictable from the 0/1 response, y : 'did seller respond to buyer?'

Fit $p_i = \mathrm{p}(y_i = 1|\mathbf{x}_i)$ as an *Empirical Bayesian Forest* ($\approx$ an RF).
$\mathbf{x}$: buyer, seller, item attributes (even includes sale title topics).



We can split the reviews on the residuals – high $r_i = y_i - \hat{p}_i$ and low $r_i = y_i - \hat{p}_i$ – and fit a word2vec representation to each.

Now, with residual r_i as the variable separating our samples...

The high r_i representation still places high probability on language indicating previous deals and bundling.

But the low r_i representation now puts high probability on pleading:

know that my offer not close what you asking cant afford pay much more than my offer but really want this pin please let me know you can make counteroffer this offer not acceptabl

you dont like offer feel free tell me what lowest youll go on these [product] like you ive been collect since was im now hope we can reach deal wich fair both us by way they some awsome [product] !!

hello my friend hope my offer good enough really want [product] ... cgc going be there they grading comics also do you know how much they charge grade comic thkas greg

last one sold on dec 26th ,, can match that price im also interested [product] was wondering how much pair ?? can have money ur account tonight we can reach deal thanks your time ,,

W2V Inversion is simple, scalable, and it works

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

If you think carefully about what attributes (or even estimated quantities, or residuals) are of interest, W2V inversion is a great tool for understanding language discrepancies.

THANKS!