Labeled Anchors and a Scalable, Transparent, and Interactive Classifier

Jeffrey Lund, Stephen Cowley, Wilson Fearn, Emily Hales, Kevin Seppi
Computer Science Department
Brigham Young University
{jefflund, scowley4, wfearn, emilyhales, kseppi}@byu.edu

Abstract

We propose Labeled Anchors, an interactive and supervised topic model based on the anchor words algorithm (Arora et al., 2013). Labeled Anchors is similar to Supervised Anchors (Nguyen et al., 2014) in that it extends the vector-space representation of words to include document labels. However, our formulation also admits a classifier which requires no training beyond inferring topics, which means our approach is also fast enough to be interactive. We run a small user study that demonstrates that untrained users can interactively update topics in order to improve classification accuracy.

In this paper, we concern ourselves with the problem of interactive and transparent text classification. The value of such a classifier can be seen in the events shortly before the 2016 US presidential election when FBI Director James Comey notified Congress that the FBI had obtained emails from candidate Hillary Clinton's private email server which potentially contained state secrets. Nearly a week later, just two days before the election, Comey announced that nothing had been found in the emails that warranted prosecution. Many speculate that the timing of these announcements may have influenced the election.

There are times when the ability to quickly analyze large quantities of text is of critical importance. In the case of the Clinton emails, manual inspection appears to have been possible in one week's time, but there could have been less controversy if the emails had been categorized in a shorter period of time. Furthermore, in future cases the data may be too large for manual analysis.

While there are many text classification algorithms, none are both interactive and transparent at scale. We require interactivity because we would like to leverage human intuition to improve classification accuracy for a specific task. Transparency not only enables interactivity, but also allows users to inspect the classifier and gain confidence in the results.

Topic models such as Latent Dirichlet Allocation (or LDA) (Blei et al., 2003) aim to automatically distill large collections of documents into topics. These topics can be used to perform document classification (Rubin et al., 2012). Furthermore, work has been done to increase the human interpretability of topics (Mimno et al., 2011). Traditionally, topic models are graphical models which typically scale poorly to large data. A faster alternative is the Anchor Words algorithm, which relies on non-negative matrix factorization to infer topics (Arora et al., 2013). Ordinarily, this factorization is NP-Hard (Arora et al., 2012), but with certain separability assumptions related to ”anchor” words which uniquely identify topics, the factorization is scalable.

The Interactive Topic Model (Hu et al., 2011) allows human knowledge to be injected into the model in order to shape the topics in some meaningful way. While this model does incorporate user feedback, it is not fast enough to be truly interactive. A more scalable alternative is Tandem Anchors, which allows users to specify anchor words in order to influence the resulting topics (Lund et al., 2017).

A separate line of topic modeling research deals with supervised topic modeling, which allows document labels to influence topic inference (Mcauliffe and Blei, 2008). The most recent work on supervised topic modeling is Supervised Anchors (Nguyen et al., 2014). This approach uses document labels to influence the selection of anchor words, which in turn affects the resulting topics. However, Supervised Anchors requires a

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 824–829
Brussels, Belgium, October 31 - November 4, 2018. ©2018 Association for Computational Linguistics

downstream classifier to be trained using topics as features.

Our main contribution combines the idea of Tandem Anchors with Supervised Anchors to produce text classification which is both interactive and transparent. Additionally, the mathematical approach we take to build this classification requires no training beyond inferring topics, unlike Supervised Anchors which requires both topic inference and significant additional time for training a downstream classifier. While Supervised Anchors requires the construction of an external classifier, our approach generates the classifier as part of topic inference. Consequently, our model is extremely fast and scalable compared to Supervised Anchors. We demonstrate that users are able to use our model to interactively improve document classification accuracy by manipulating topics.

1 Labeled Anchors

In this section we describe our approach, combining interactive and supervised topic modeling, which we call Labeled Anchors. We extend the Anchor Words algorithm (Arora et al., 2013) which takes as input a $V \times D$ matrix M of document-word counts and recovers a $V \times K$ matrix A of word probabilities conditioned by topic, where there are V word types, D documents, and K topics. Our approach extends this algorithm to incorporate L possible document labels.

1.1 Vanilla Anchor Words

In order to compute the topic-word matrix A, the Anchor Words algorithm uses a $V \times V$ cooccurrence matrix $\bar{Q}$. Each entry $\bar{Q}_{i,j}$ gives the conditional probability of word j occurring after observing word i in a document. Following Appendix D.1 of Arora et al. (2013), $\bar{Q}$ is obtained by row-normalizing Q, which in turn is constructed using

$$Q = \bar{M}\bar{M}^T - \hat{M} \tag{1}$$

where $\bar{M}$ is a normalized version of the document-word matrix M giving equal weight to each document regardless of document length, and $\hat{M}$ accounts for words not cooccurring with themselves.

$\bar{Q}$ is a V-dimensional vector-space representation of each word and is used to compute a set of anchor words S. Each anchor word uniquely identifies a topic by having non-zero probability in one topic only. These anchors are computed using an adaptation of the Gram-Schmidt process from Arora et al. (2013). Once the set of anchor words S has been computed, we reconstruct the non-anchor words as a convex combination of the anchor word vectors. The coefficients of these combinations C are computed using exponentiated gradient descent to optimize

$$C_i = \underset{C_i}{argmin}\, D_{KL}(\bar{Q}_i || \sum_{k \in S} C_{i,k}\bar{Q}_k) \tag{2}$$

where i is the ith word of the vocabulary, $\bar{Q}_i$ is the vector-space representation of word i, and $D_{KL}(\cdot||\cdot)$ is Kullback-Leibler divergence.[1]

Because the occurrence pattern of each anchor word throughout the documents must mirror that of the topic it anchors, each coefficient $C_{i,j}$ gives the conditional probability of topic j occurring given word i. This is the inverse conditioning we desire in the topic-word matrix A. We can therefore compute A using Bayes' Rule by multiplying the coefficient matrix C with the empirical probability of each word to get the probability of a word given a particular topic.

1.2 Vector-Space Representations

Supervised Anchors (Nguyen et al., 2014) augments $\bar{Q}$ by appending L additional columns to $\bar{Q}$ corresponding to the probability of words cooccurring with the L possible document labels. Because this augmented vector-space representation includes dimensions corresponding to document labels, both the anchor words and the resulting topics will reflect the document labels.

Our algorithm, called Labeled Anchors, also augments the vector-space representation to include the L document labels. However, we do not directly modify $\bar{Q}$. Instead, we treat the L possible document labels as words and pretend that we observe these label pseudowords directly in each labeled document; a graphical representation of this is shown below in Figure 1.[2]

Consequently, our document-word matrix M is a $(V + L) \times D$ matrix. The first V entries of each column of M give the word counts for a particular document. The last L entries are zero, except for the entry corresponding to the label of that document.

We then construct $\bar{Q}$ using Equation 1, obtaining an order $V + L$ square matrix. As with Su-

[1] Alternatively, we can use l^2-norm in place of KL-divergence.

[2] We could add multiple such words per label, but our preliminary experiments indicate that one per label is sufficient.

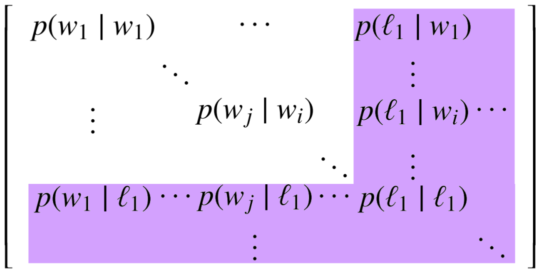
$p(w_1 \mid w_1)$
$\cdots$
$p(\ell_1 \mid w_1)$
$p(w_j \mid w_i)$
$p(\ell_1 \mid w_i)\cdots$
$p(w_1 \mid \ell_1)\cdots p(w_j \mid \ell_1)\cdots p(\ell_1 \mid \ell_1)$

Figure 1: Labeled Anchors treats labels as observed words in each labeled documents and updates $\bar{Q}$ under this assumption, creating the additional rows and columns highlighted here.

pervised Anchors, these additional L dimensions guide anchor selection to include anchors which reflect the underlying document labels. When we use Equation 2 to compute C, we also obtain an additional L rows of coefficients which each correspond to the conditional probability of a topic given a label. Finally, the first V rows of A are computed using Bayes' Rule to give us the probability of words given topics.

Labeled Anchors inherits the run time characteristics of the original Anchor Words algorithm. As shown in Arora et al. (2013), topic recovery requires $O(KV^2 + K^2VT)$, where V is the size of the vocabulary, K is the number of anchors/topics, and T is the average number of iterations (typically around 100 in our experiments). Since $V \gg K$, adding any modest number of topics (less than 200) does not noticeably increase the runtime. Furthermore, since vocabulary size tends to grow logarithmically with respect to the size of the data (Heaps, 1978), this approach is scalable even for very large datasets.

1.3 Free Classifier

Note that once the cooccurrence matrix $\bar{Q}$ has been computed, the recovery of the topic-word matrix A scales with the size of the vocabulary, not the size of the data. However, Supervised Anchors requires topic assignments for each training document[3] for use as features for some downstream classifier. Therefore, the process of building a classifier scales linearly with the number of documents and can be time consuming compared to topic recovery.

In contrast, the formulation of Labeled Anchors allows us to construct a classifier with no additional training. To do so, rather than using LDA with fixed topics, we employ a simple model similar to Labeled LDA (Ramage et al., 2009) with the following generative story for an individual document containing N words:

1. Draw label $\ell \sim Cat(\lambda)$
2. For each $i \in [1, ..., N]$:
   (a) Draw topic assignment $z_i|\ell \sim Cat(\psi_l)$
   (b) Draw word $w_i|z_i \sim Cat(\phi_{z_i})$

The prior over document labels λ is simply the proportion of each label in the training data. We can estimate topic-label probabilities ψ using the last L rows of the coefficient matrix C, while the word-topic probabilities ϕ are the first V rows of A. Using these hyperparameters, we make predictions using the following:

$$\ell^* = \underset{\ell}{argmax}\ p(\ell|\mathbf{w}) = \underset{\ell}{argmax}\ p(\ell, \mathbf{w}) \quad (3)$$

$$= \underset{\ell}{argmax} \sum_{z_1=1}^{K} ... \sum_{z_N=1}^{K} p(\ell, \mathbf{z}, \mathbf{w}) \quad (4)$$

$$= \underset{\ell}{argmax}\ p(\ell) \prod_{i=1}^{N} \sum_{z_i=1}^{K} p(z_i|\ell)p(w_i|z_i) \quad (5)$$

$$= \underset{\ell}{argmax}\ \lambda_\ell \prod_{i=1}^{N} \sum_{z_i=1}^{K} \psi_{\ell,z_i}\phi_{z_i,w_i} \quad (6)$$

$$= \underset{\ell}{argmax}\ log\lambda_\ell + \sum_{i=1}^{N} log\left(\sum_{z_i=1}^{K} C_{\ell,z_i} A_{z_i,w_i}\right) \quad (7)$$

where Equation 4 unmarginalizes the probabilities across the word-topic assignments, Equation 5 uses the model's conditional independencies to expand and simplify the probabilities, Equation 6 explicitly uses the parameters from the generative model, and Equation 7 transitions to the matrix representations for these probabilities as found in Section 1.2. In Equation 7 we also switch to log space to mitigate numeric precision issues.

1.4 User Interaction

Assuming that $\bar{Q}$ is precomputed and fixed, Labeled Anchors is fast enough to allow interactive modification of the topics as well as interactive display of classification accuracy, even on large datasets. The final step to solving the problem of creating an interactive and transparent classifier is to allow users to inject domain specific knowledge

[3] This is typically done using LDA with fixed topics.

#Docs	#Vocab	Labeled	Supervised
39388	3406	.532s	17.1s
99955	4829	.886s	28.6s
990820	6648	1.10s	282s

Table 1: Runtime for Labeled Anchors and Supervised Anchors on various subsets of Amazon product reviews. Labeled Anchors is dramatically faster than Supervised Anchors and scales to much larger datasets.

into the topic model. To do so, we use the idea of Tandem Anchors (Lund et al., 2017), which allows users to manually select sets of words to form anchors.

Ordinarily, anchor words can be somewhat inscrutable to human users. Because anchor words must uniquely identify topics, good anchors are typically esoteric low-to-mid frequency words. Intuitive, high frequency words usually appear in multiple topics. However, if we examine Equation 2, we can see that the anchor words are just points in V-dimension space; they do not actually have to correspond to any particular word so long as that point in space uniquely identifies a topic.

Tandem Anchors allows multiple words to form a single anchor pseudoword by computing the element-wise harmonic mean of a set of words. Since the harmonic mean tends towards the lowest values, the resulting pseudoword anchor largely ignores superfluous cooccurrence patterns in the constituent words. Consequently, while individual words forming the anchor may be ambiguous, users can combine multiple ambiguous words to intuitively express a single coherent idea.

2 Experimental Results

Before running a user study to validate that Labeled Anchors works as an interactive and transparent classifier, we first run a synthetic experiment to determine the runtime characteristics of our algorithm. We take subsets of a large collection of Amazon reviews[4] to produce datasets of various sizes. Using this data, we compare the runtime of Labeled Anchors with that of Supervised Anchors. All results are obtained using a single core of an Intel Core i7-4770K.[5]

As shown in Table 1, Labeled Anchors is orders of magnitude faster than Supervised Anchors, even for moderately sized datasets. Both Labeled Anchors and Supervised Anchors require us to recover topic-word distributions, an operation which scales with the size of the vocabulary. However, Supervised Anchors also requires us to infer document-topic distributions in order to train an external classifier, an operation which scales linearly with the number of documents. Since vocabulary size typically grows logarithmically with respect to the number of documents (Heaps, 1978), Labeled Anchors scales much better than Supervised Anchors.

When a user updates the anchors, the system must reinfer the topics, create the classifier, and evaluate the development dataset, all within a few seconds. If the update is too slow, the interaction will suffer due to increased cognitive load on users (Cook and Thomas, 2005). Results from an exploratory user study confirm this: when participants are faced with update times around 10 seconds, they are not successful in their topic-based tasks.[6]

Having established that Labeled Anchors is fast enough to be interactive, we now demonstrate that participants can use our system to improve topics for classification. In order demonstrate the role of human knowledge in interactive topic modeling, we ask the users to identify sentiment (i.e. product rating) rather than product category, since the natural topics which arise from the Anchor Words algorithm tend to reflect product category instead of rating. We preprocess a set of Amazon product reviews with standard tokenization, stopword removal, and by removing words which appear in fewer than 100 documents. After preprocessing, empty documents are discarded, resulting in 39,388 documents. We use an 80/20 train/test split, with 1,500 training documents reserved as development data. We recruit five participants drawn from a university student body. The median age is 21. Three are male and two are female. None of the students have any prior familiarity with topic modeling.

We present the participants with a user interface similar to that of Lund et al. (2017). Users can view and edit a list of anchor words (or rather, sets of words which form each anchor), and they can view the top ten most probable words for each topic. We display the classification accuracy on the development data to give users an indication of

[4] http://jmcauley.ucsd.edu/data/amazon

[5] Our Python implementation is available at https://github.com/byu-aml-lab/ankura.

[6] For this reason, we do not report interactive results with Supervised Anchors.

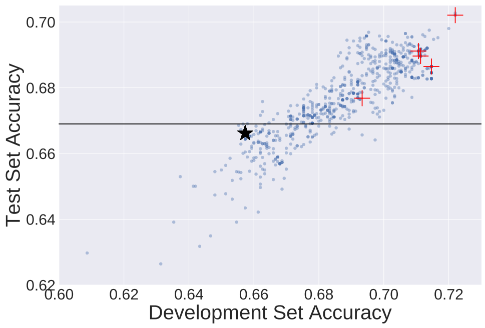

Figure 2: User study accuracy results comparing accuracy on the development set to the accuracy on the test set. The black horizontal line indicates the baseline accuracy from Supervised Anchors. The black star indicates the initial accuracy using Gram-Schmidt anchors with Labeled Anchors. The blue dots indicate various intermediate steps while editing the anchors. The red pluses are the final states after each user completes the task.

how they are doing. After a brief training on the interface, users are asked to modify the anchors to produce topics which reflect the underlying product ratings and improve the classification accuracy on the development dataset as much as possible. Participants are given forty minutes to perform this task.

Figure 2 summarizes the results of our user study. With just baseline anchors from Gram-Schmidt, the classification accuracy of Labeled Anchors is on par with that of Supervised Anchors using logistic regression as the downstream classifier. However, because Labeled Anchors is fast enough to allow interaction, participants are able to improve classification accuracy on the development set by an average of 5.31%. This corresponds to a 2.31% increase in accuracy on the test set.

We record each step of the user interactions and find a Pearson correlation coefficient of .88 between development accuracy and test accuracy. Thus, Labeled Anchors allows participants to interactively see updated classification accuracy and have confidence that held-out test accuracy will also improve.

With regard to the interaction that users had with the dataset, we observe several common strategies. Firstly, we notice that users who made more edits tend to have more success in terms of accuracy; this validates our assertion that slower update times hurt performance. Secondly, users end with a median of 21 topics, which is close to the 20 topics they start with, suggesting that either the users felt like this was an appropriate number of topics, or that they felt uneasy drastically changing the total number of topics from what they started with. Lastly, we find that users chose more single word anchors than we expected, with about 88% of anchors being single word anchors. Most of the multiword anchors users used were short 2-3 word phrases which did not have an obvious single word counterpart.

3 Conclusion

Our results demonstrate that Labeled Anchors yields a classifier that is both human-interpretable and fast. Our approach not only combines the strengths of Supervised Anchors and Tandem Anchors, but introduces a mathematical construct for producing a classifier as a by-product of topic inference. Compared to Supervised Anchors, which requires costly training of a downstream classifier in addition to topic inference, our approach is much more scalable. Using Labeled Anchors, our participants are able to adjust the classifier so as to obtain superior classification results than those produced by Supervised Anchors alone.

Returning to our original motivating problem of quickly annotating a large collection of unlabeled emails, we assert that our approach could aid in quickly labeling the entire collection. With a modest investment of manual annotation, the initial training set could be labeled, and then with the help of our system the remaining documents could be automatically labeled in a transparent and explainable fashion.

Acknowledgements

This work was supported by the NSF Grant IIS-1409739.

Thanks to Piper Armstrong, Naomi Johnson, Connor Cook and Nozomu Okuda for their invaluable help with this work.

References

Sanjeev Arora, Rong Ge, Yoni Halpern, David M. Mimno, Ankur Moitra, David Sontag, Yichen Wu, and Zhu Michael. 2013. A practical algorithm for topic modeling with provable guarantees. In *Proceedings of the International Conference of Machine Learning*.

Sanjeev Arora, Rong Ge, Ravindran Kannan, and Ankur Moitra. 2012. Computing a nonnegative matrix factorization–provably. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*.

David M. Blei, Andrew Ng, and Michael Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Kristin A. Cook and James J. Thomas. 2005. Illuminating the path: The research and development agend for visual analytics.

Harold Stanley Heaps. 1978. *Information retrieval: Computational and theoretical aspects*. Academic Press, Inc.

Yuening Hu, Jordan Boyd-Graber, Brianna Satinoff, and Alison Smith. 2011. Interactive topic modeling. In *Proceedings of the Association for Computational Linguistics*.

Jeffrey Lund, Connor Cook, Kevin Seppi, and Jordan Boyd-Graber. 2017. Tandem anchoring: A multiword anchor approach for interactive topic modeling. In *Proceedings of the Association for Computational Linguistics*.

Jon D Mcauliffe and David M Blei. 2008. Supervised topic models. *Proceedings of Advances in Neural Information Processing Systems*.

David Mimno, Hanna M Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. 2011. Optimizing semantic coherence in topic models. In *Proceedings of Empirical Methods in Natural Language Processing*.

Thang Nguyen, Jordan Boyd-Graber, Jeff Lund, Kevin Seppi, and Eric Ringger. 2014. Is your anchor going up or down? Fast and accurate supervised topic models. In *Conference of the North American Chapter of the Association for Computational Linguistics*.

Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D Manning. 2009. Labeled lda: A supervised topic model for credit attribution in multilabeled corpora. In *Proceedings of Empirical Methods in Natural Language Processing*.

Timothy Rubin, America Chambers, Padhraic Smyth, and Mark Steyvers. 2012. Statistical topic models for multi-label document classification. *Machine Learning*, 1(88):157–208.