Is Your Anchor Going Up or Down? Fast and Accurate Supervised Topic Models

Thang Nguyen
iSchool and UMIACS
University of Maryland
and National Library of Medicine,
National Institutes of Health
daithang@umiacs.umd.edu

Jordan Boyd-Graber
Computer Science
University of Colorado Boulder
Jordan.Boyd.Graber
@colorado.edu

Jeff Lund,
Kevin Seppi, Eric Ringger
Computer Science
Brigham Young University
{jefflund,kseppi}@byu.edu
ringger@cs.byu.edu

Abstract

Topic models provide insights into document collections, and their supervised extensions also capture associated document-level metadata such as sentiment. However, inferring such models from data is often slow and cannot scale to big data. We build upon the "anchor" method for learning topic models to capture the relationship between metadata and latent topics by extending the vector-space representation of word-cooccurrence to include metadata-specific dimensions. These additional dimensions reveal new anchor words that reflect specific combinations of metadata and topic. We show that these new latent representations predict sentiment as accurately as supervised topic models, and we find these representations more quickly without sacrificing interpretability.

Topic models were introduced in an unsupervised setting (Blei et al., 2003), aiding in the discovery of topical structure in text: large corpora can be distilled into human-interpretable themes that facilitate quick understanding. In addition to illuminating document collections for humans, topic models have increasingly been used for automatic downstream applications such as sentiment analysis (Titov and McDonald, 2008; Paul and Girju, 2010; Nguyen et al., 2013).

Unfortunately, the structure discovered by unsupervised topic models does not necessarily constitute the best set of features for tasks such as sentiment analysis. Consider a topic model trained on Amazon product reviews. A topic model might discover a topic about vampire romance. However, we often want to go deeper, discovering facets of a topic that reflect topic-specific sentiment, e.g., "buffy" and "spike" for positive sentiment vs. "twilight" and "cullen" for negative sentiment. Techniques for discovering such associations, called supervised topic models (Section 2), both produce interpretable topics and predict metadata values. While unsupervised topic models now have scalable inference strategies (Hoffman et al., 2013; Zhai et al., 2012), supervised topic model inference has not received as much attention and often scales poorly.

The anchor algorithm is a fast, scalable unsupervised approach for finding "anchor words"—precise words with unique co-occurrence patterns that can define the topics of a collection of documents. We augment the anchor algorithm to find supervised sentiment-specific anchor words (Section 3). Our algorithm is faster and just as effective as traditional schemes for supervised topic modeling (Section 4).

1 Anchors: Speedy Unsupervised Models

The anchor algorithm (Arora et al., 2013) begins with a $V \times V$ matrix $\bar{Q}$ of word co-occurrences, where V is the size of the vocabulary. Each word type defines a vector $\bar{Q}_{i,\cdot}$ of length V so that $\bar{Q}_{i,j}$ encodes the conditional probability of seeing word j given that word i has already been seen. Spectral methods (Anandkumar et al., 2012) and the anchor algorithm are fast alternatives to traditional topic model inference schemes because they can discover topics via these summary statistics (quadratic in the number of *types*) rather than examining the whole dataset (proportional to the much larger number of *tokens*).

The anchor algorithm takes its name from the idea

of anchor words—words which unambiguously identify a particular topic. For instance, "wicket" might be an anchor word for the cricket topic. Thus, for any anchor word a, $\bar{Q}_{a,\cdot}$ will look like a topic distribution. $\bar{Q}_{\text{wicket},\cdot}$ will have high probability for "bowl", "century", "pitch", and "bat"; these words are related to cricket, but they cannot be anchor words because they are also related to other topics.

Because these other non-anchor words could be topically ambiguous, their co-occurrence must be explained through some combination of anchor words; thus for non-anchor word i,

$$\bar{Q}_{i,\cdot} = \sum_{g_k \in \mathcal{G}} C_{i,k} \bar{Q}_{g_k,\cdot}, \tag{1}$$

where $\mathcal{G} = \{g_1, g_2, \ldots, g_K\}$ is the set of K anchor words. The coefficients $C_{i,k}$ of this linear combination correspond to the probability of seeing a topic *given a word*, from which we can recover the probability of a word *given a topic* (represented in a matrix A) using Bayes' rule. In our experiments, we follow Arora et al. (2013) to first estimate $\bar{Q}$ based on the training data and then recover the C matrix

$$C^*_{i,\cdot} = argmin_{C_{i,\cdot}} D_{KL}(\bar{Q}_{i,\cdot} \,||\, \sum_{g_k \in \mathcal{G}} C_{i,k} \bar{Q}_{g_k,\cdot}),$$

where $D_{KL}(x, y)$ denotes the Kullback-Leibler divergence between x and y.

In addition to discovering topics from a given set of anchor words as described above, Arora et al. (2013) also provide a geometric interpretation of a process for finding the needed anchor words. If we view the rows of $\bar{Q}$ as points in a high-dimensional space, the convex hull of those points provides the anchor words.[1]

Equation 1 linearly combines anchor words' co-occurrence vectors $\bar{Q}_{g_k,\cdot}$ to create the representation of other words. The convex hull corresponds to the perimeter of the space of all possible co-occurrence vectors that can be formed from the set of basis anchor vectors. However, the convex hull only encodes an unsupervised view of the data. To capture topics informed by metadata such as sentiment, we need to explicitly represent the combination of words and metadata.

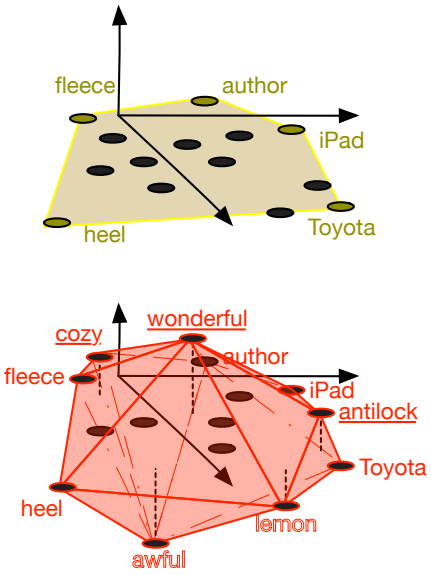

Figure 1: Graphical intuition behind supervised anchor words. Anchor words (in gold) form the convex hull of word co-occurrence probabilities in unsupervised topic modeling (top). Adding an additional dimension to capture metadata, such as sentiment, changes the convex hull: positive words appear above the original 2D plane (underlined) and negative words appear below (in outline).

One problem inherited by the anchor method from parametric topic models is the determination of the number of anchor words (and thus topics) to use. Because word co-occurrence statistics live in an extremely high-dimensional space, the number of anchor words needed to cover all of the data will be quite high. Thus, Arora et al. (2013) require a user to specify the number of anchor words *a priori* (just as for parametric topic models). They use a form of the Gram-Schmidt process to find the best words that enclose the maximum volume of points.

[1] As discussed by Arora et al. (2013), this is a slight simplification, since the most extreme points will be words that only appear infrequently. Thus, there is some nuance to choosing the anchor words. For instance, a key step for effective topic modeling is choosing a minimum number of documents a word must appear in before it can be considered an anchor word. (c.f. Figure 3).

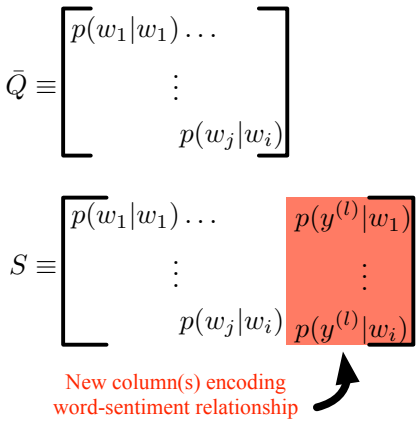

Figure 2: We form a new column to capture the relationship between words *and* each sentiment level: per entry is the conditional probability of observing a sentiment level $y^{(l)}$ given an observation of the word w_i. Adding all of these columns to $\bar{Q}$ to form an augmented matrix S.

2 Supervised Topics: Effective but Slow

Topic models discover a set of topics A. Each topic is a distribution over the V word types in the corpus. $A_{i,t}$ is the probability of seeing word i in topic t. Supervised topic models relate those topics with predictions of document metadata such as sentiment by discovering a vector of regression parameters $\vec{\mu}$ that connects topics to per-document observations y_d (Blei and McAuliffe, 2007). Blei and McAuliffe (2007) treat this as a regression: seeing one word with topic k in document d means that prediction of y_d should be adjusted by μ_k. Given a document's distribution over topics $\bar{z}_d$, the response y_d is normally distributed with mean $\vec{\mu}^\top \bar{z}_d$.[2]

Typically, the topics are discovered through a process of probabilistic inference, either variational EM (Wang et al., 2009) or Gibbs sampling (Boyd-Graber and Resnik, 2010). However, these methods scale poorly to large datasets. Variational inference requires dozens of expensive passes over the entire dataset, and Gibbs sampling requires multiple Markov chains (Nguyen et al., 2014b).

[2]We are eliding some details in the interest of a more compact presentation. The topics used by a document, $\bar{z}_d$, are based on per-token inference of topic assignments; this detail is not relevant to our contribution, and in Section 4.2 we use existing techniques to discover documents' topics.

3 Supervised Anchor Words

Because the anchor algorithm scales so well compared to traditional probabilistic inference, we now unify the supervised topic models of Section 2 with the anchor algorithm discussed in Section 1. We do so by augmenting the matrix $\bar{Q}$ with an additional dimension for each metadata attribute, such as sentiment. We provide the geometric intuition in Figure 1.

Picture the anchor words projected down to two dimensions (Lee and Mimno, 2014): each word is a point, and the anchor words are the vertices of a polygon encompassing every point. Every non-anchor word can be approximated by a convex combination of the anchor words (Figure 1, top).

Now add an additional dimension as a column to $\bar{Q}$ (Figure 2). This column encodes the metadata specific to a word. For example, we have encoded sentiment metadata in a new dimension (Figure 1, bottom). Neutral sentiment words will stay in the plane inhabited by the other words, positive sentiment words will move up, and negative sentiment words will move down. For simplicity, we only show a single additional dimension, but in general we can add as many dimensions as needed to encode the metadata.

In this new space some of the original anchor words may still be anchor words ("author"). Other words that were near the convex hull boundary in the unaugmented representation may become anchor words in the augmented representation because they capture both topic and sentiment ("anti-lock" vs. "lemon"). Finally, extreme sentiment words might become anchor words in the new higher-dimensional space because they are so important for explaining extreme sentiment values ("wonderful" vs. "awful").

3.1 Words to Sentiment

Having explained how a word is connected to sentiment, we now elaborates on how to model that connection using the conditional probability of sentiment given a particular word. Assume that sentiment is discretized into a finite set of L sentiment levels $\{y^{(1)}, y^{(2)}, \ldots, y^{(L)}\}$ and that each document is assigned to one of these levels. We define a matrix S of size $V \times (V + L)$. The first V columns are the same as $\bar{Q}$ and the L additional columns capture the relationship of a word to each discrete sentiment

level.

For each additional column l, $S_{i,(V+l)} \equiv p(y = y^{(l)} \mid w = i)$ is the conditional probability of observing a sentiment level $y^{(l)}$ given an observation of word i. We compute the conditional probability of a sentiment level $y^{(l)}$ given word i

$$S_{i,(V+l)} \equiv \frac{\sum_d (\mathbb{1}\left[i \in d\right] \cdot \mathbb{1}\left[y_d = y^{(l)}\right])}{\sum_d \mathbb{1}\left[i \in d\right]}, \quad (2)$$

where the numerator is the number of documents that contain word type i and have sentiment level $y^{(l)}$ and the denominator is the number of documents containing word i.

Given this augmented matrix, we again want to find the set of anchor words $\mathcal{G}$ and coefficients $C_{i,k}$ that best capture the relationship between words and sentiment (c.f. Equation 1)

$$S_{i,\cdot} = \sum_{g_k \in \mathcal{G}} C_{i,k} S_{g_k,\cdot}. \quad (3)$$

Because we retain the property that non-anchor words are explained through a linear combination of the anchor words, our method retains the same theoretical guarantees of sampling complexity and robustness as the original anchor algorithm.

To facilitate direct comparisons, we keep the number of anchor words fixed in our experiments. Even so, the introduction of metadata forces the anchor method to select the words that best capture this metadata-augmented view of the data. Consequently, some of the original anchor words will remain, and some will be replaced by sentiment-specific anchor words.

4 Quantitative Comparison of Supervised Topic Models

In this section, we evaluate the effectiveness of our new method on a binary sentiment classification problem. Because the supervised anchor algorithm (**SUP ANCHOR**) finds anchor words (and thus different topics) which capture the sentiment metadata, we evaluate the degree to which its latent representation improves upon the original unsupervised anchor algorithm (Arora et al., 2013, **ANCHOR**) for classification in terms of both accuracy and speed.

4.1 Sentiment Datasets

We use three common sentiment datasets for evaluation: AMAZON product reviews (Jindal and Liu, 2008), YELP restaurant reviews (Jo and Oh, 2011), and TRIPADVISOR hotel reviews (Wang et al., 2010). For each dataset, we preprocess by tokenizing and removing all non-alphanumeric words and stopwords. As very short reviews are often inscrutable and lack cues to connect to the sentiment, we only consider documents with at least thirty words. We also reduce the vocabulary size by keeping only words that appear in a sufficient number of documents: 50 for AMAZON and YELP datasets, and 150 for TRIPADVISOR (Table 1).

4.2 Documents to Labels

Our goal is to perform binary classification of sentiment. Due to a positive skew of the datasets, the median for all datasets is four out of five. All 5-star reviews are assigned to y^+ and the rest of the reviews are assigned to y^-. Table 1 summarizes the composition of each dataset and the percentage of documents with high positive sentiment.[3]

We compare the effectiveness of different representations in predicting high-sentiment documents: unsupervised topic models (**LDA**), traditional supervised topic models (**SLDA**), the unmodified anchor algorithm (**ANCHOR**), our supervised anchor algorithm (**SUP ANCHOR**), and a traditional TF-IDF (Salton, 1968, **TF-IDF**) representation of the words.

The anchor algorithm only provides the topic distribution over words; it does not provide the per-document assignment of topics needed to represent the document in a low-dimensional space necessary for producing a prediction y_d. Fortunately, this requires a very quick—because the topics are fixed—pass over the documents using a traditional topic model inference algorithm. We use the variational inference implementation for **LDA** of Blei et al. (2003)[4] to obtain $\bar{z}_d$, the topic distribution for document d.[5]

[3] Multiclass labeling for each sentiment label also works well, but binary classification simplifies the analysis and presentation.

[4] `http://www.cs.princeton.edu/~blei/lda-c/`

[5] For other inference schemes, we use native inference to apply pre-trained topics to extract DEV and TEST topic proportions.

Corpus	Train Documents	Test Documents	Tokens	Types	Percentage with Positive Sentiment
AMAZON	13,300	3,314	1,031,659	2,662	52.2%
TRIPADVISOR	115,384	28,828	12,752,444	4,867	41.5%
YELP	13,955	3,482	1,142,555	2,585	27.7%

Table 1: Statistics for the datasets employed in the experiments.

Classifiers Given a low-dimensional representation of a test document, we predict the document's sentiment y_d. We have already inferred the topic distribution $\bar{z}_d$ for each document, and we use $log(\bar{z}_d)$ as the features for a classifier. Feature vectors from training data are used to train the classifiers, and feature vectors from the development or test set are used to evaluate the classifiers.

We run three standard machine learning classifiers: decision trees (Quinlan, 1986), logistic regression (Friedman et al., 1998), and a discriminative classifier. For decision trees (hence TREE) and logistic regression (hence LOGISTIC), we use SKLEARN.[6] For the discriminative classifier, we use a linear classifier with hinge loss (hence HINGE) in Vowpal Wabbit.[7] Because HINGE outputs a regression value in $[0, 1]$, we use a threshold 0.5 to make predictions.

Parameter Tuning Parameter tuning is important in topic models, so we cross-validate: each sentiment dataset is split randomly into five folds. We used four folds to form the TRAIN set and reserved the last fold for the TEST set. All cross-validation results are averaged over the four held out DEV sets; the best cross-validation result provides the parameter settings we use on the TEST set.

For **ANCHOR** and **SUP ANCHOR**, the parameter for the document-level Dirichlet prior α is required for inferring document-topic distributions given learned topics. Despite selecting this parameter using grid search, α does not affect our final results. The same is also true for **SLDA**: its predictive performance does not significantly vary as α varies, given a fixed number of topics K.[8]

Anchor algorithms are sensitive to the value of anchor threshold M (the minimum document frequency for a word to be considered an anchor word). For each number of topics K, we perform a grid search to find the best value of M. Figure 3 shows the performance trends.

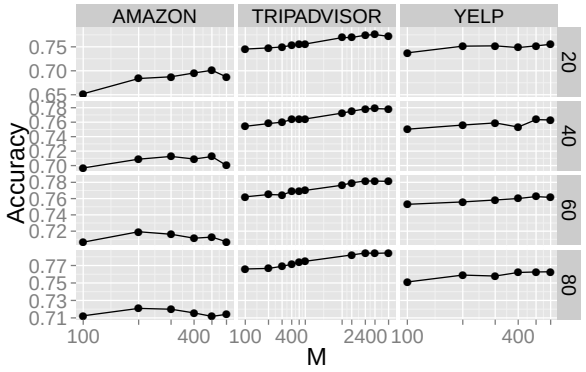

Figure 3: Grid search for selecting the word-document threshold M for **SUP ANCHOR** based on development set accuracy.

For **LDA**, we use the Gibbs sampling implementation in Mallet.[9] For training the model, we run **LDA** with 5,000 iterations; and for inference (on DEV and TEST) of document topic distribution we iterate 100 times, with lag 5 and 50 burn-in iterations. As Mallet accepts $\sum \alpha_i$ as a parameter, we always initialize $\sum \alpha_i = 1$ and only perform a grid search over different values of β, the hyper-parameter for Dirichlet prior over the per-topic topic-word distribution, starting from 0.01 and doubling until reaching 0.5.

4.3 SUP ANCHOR Outperforms ANCHOR

Learning topics that jointly reflect words and metadata improves subsequent prediction. The results for both **SUP ANCHOR** and **ANCHOR** on the TEST set are shown in Figure 4. **SUP ANCHOR** outperforms **ANCHOR** on all datasets. This trend holds consistently for LOGISTIC, TREE, and HINGE methods for sentiment prediction. For example, with twenty topics on the AMAZON dataset, **SUP ANCHOR** gives an

[6] http://scikit-learn.org/stable/

[7] http://hunch.net/~vw/

[8] We use the SLDA implementation by Chong Wang: http://www.cs.cmu.edu/~chongw/slda/ to estimate α.

[9] http://mallet.cs.umass.edu/topics.php

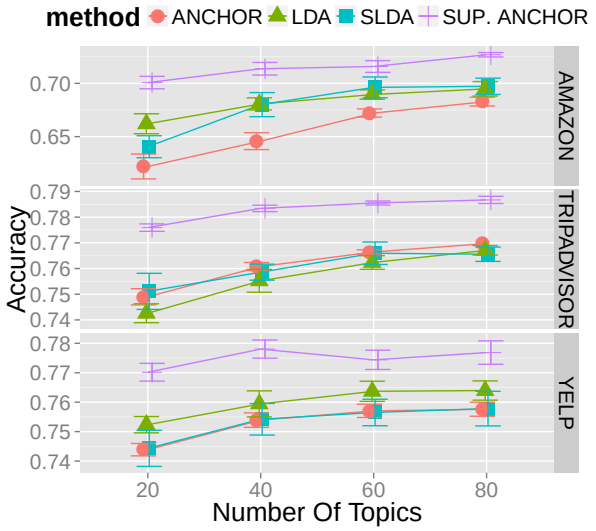

Figure 4: Results on TEST fold, **SUP ANCHOR** outperforms **ANCHOR**, **LDA**, and **SLDA** on all three datasets. We report the results based on LOGISTIC as it produces the best accuracy consistently for **ANCHOR**, **SUP ANCHOR**, and **LDA**.

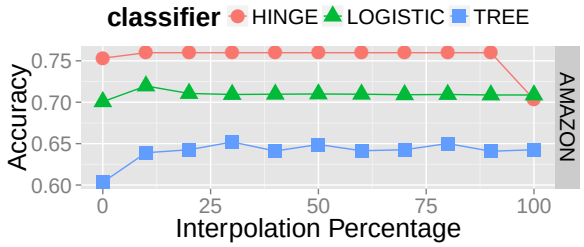

Figure 5: Accuracy on AMAZON with twenty topics. **SUP ANCHOR** produces good representations for sentiment classification that can be improved by interpolating with lexical **TF-IDF** features. The interpolation (x-axis) ranges from zero (all **TF-IDF** features) to one hundred (all **SUP ANCHOR** topic features).

accuracy of 0.71 in comparison to only 0.62 from **ANCHOR**. Similarly, with twenty topics on the YELP dataset, **SUP ANCHOR** has 0.77 accuracy while **ANCHOR** has 0.74. Our **SUP ANCHOR** model is able to incorporate metadata to learn better representations for predicting sentiment. Moreover, in Section 5 we show that **SUP ANCHOR** does not need to sacrifice topic quality to gain predictive power.

4.4 SUP ANCHOR Outperforms SLDA

More surprising is that **SUP ANCHOR** also outperforms **SLDA**. Like **SUP ANCHOR**, **SLDA** jointly learns topics and their relation to metadata such as sentiment. Figure 4 shows that this trend is consistent on all sentiment datasets. On average, **SUP ANCHOR** is 2.2 percent better than **SLDA** on AMAZON, and 2.0 percent better on both YELP and TRIPADVISOR. Furthermore, **SUP ANCHOR** is much faster than **SLDA**.

SLDA performs worse than **SUP ANCHOR** in part because **SUP ANCHOR** is able to jointly find specific lexical terms that improve prediction. Nguyen et al. (2013) show that this improves supervised topic models; forming anchor words around the same strong lexical cues could discover better topics. In contrast, **SLDA** must discover the relationship through the proxy of topics.

4.5 Lexical Features

Ramage et al. (2010) show that interpolating topic and lexical features often provides better classification than either alone. Here, we take the same approach and show how different interpolations of topic and lexical features create better classifiers. We first select an interpolation value λ in $\{0, 0.1, 0.2, \ldots, 1\}$, and we then form a new feature vector by concatenating λ-weighted topic features with $(1 - \lambda)$-weighted lexical features. Figure 5 shows the interplay between topic features and **TF-IDF** features[10] as the weight of topic features increases from zero (all **TF-IDF**) to one hundred (all **SUP ANCHOR** topic features) percent on the AMAZON dataset (other datasets are similar). Combining both feature sets is better than either alone, although the interpolation depends on the classifier.

4.6 Runtime Analysis

Having shown that **SUP ANCHOR** outperforms both **ANCHOR** and **SLDA**, in this section we show that **SUP ANCHOR** also inherits the runtime efficiency from **ANCHOR**. Table 2 summarizes the runtimes on both AMAZON and TRIPADVISOR; these results were obtained using a six-core 2.8GHz Intel Xeon X5660. On the small dataset AMAZON, **SUP ANCHOR** finishes the training within one minute, and for the larger TRIPADVISOR dataset it completes the

[10] As before, we do parameter selection on DEV data and report final TEST results.

Dataset	Measure	SUP ANCHOR	LDA	SLDA
AMAZON	Preprocessing	32	32	32
	Generating $\bar{Q}/S$	29		
	Training	33	886	4,762
	LDAC inference	38 (train), 13 (dev/test)		
	Classification	<5	<5	
TRIPADVISOR	Preprocessing	305	305	305
	Generating $\bar{Q}/S$	262		
	Training	181	8,158	71,967
	LDAC inference	830 (train), 280 (dev/test)		
	Classification	<5	<5	

Table 2: Runtime statistics (in seconds) for the AMAZON and TRIPADVISOR datasets. Blank cells indicate a timing which does not apply to a particular model. **SUP ANCHOR** is significantly faster than conventional methods.

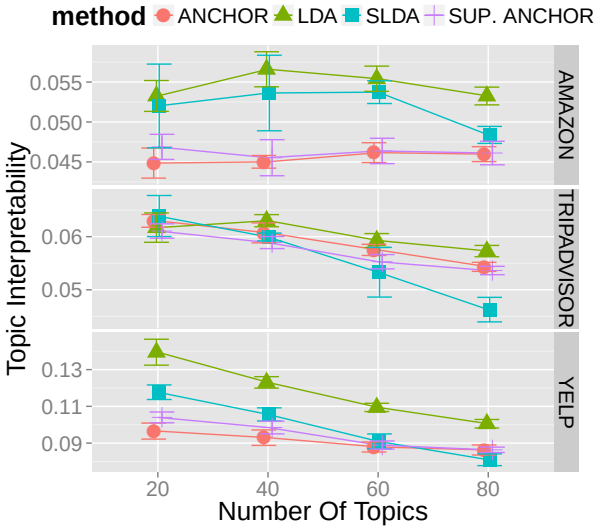

Figure 6: **SUP ANCHOR** and **ANCHOR** produce the same topic quality. **LDA** outperforms all other models and produces the best topics. Performance of **SLDA** degrades significantly as the number of topic increases.

learning in around three minutes. The main bottleneck for **SUP ANCHOR** is learning the document distributions over topics, although even this stage is fast for known topic distributions. This result is far better than the twenty hours required by **SLDA** to train on TRIPADVISOR.

5 Inspecting Anchors and their Topics

One important evaluation for topic models is how easy it is for a human reader to understand the topics. In this section, we evaluate topics produced by each model using topic interpretability (Chang et al., 2009). Topic interpretability measures how human users understand topics presented by a topic modeling algorithm. We use an automated approximation of interpretability that uses a reference corpus as a proxy for which words belong together (Newman et al., 2010). Using half a million documents from Wikipedia, we compute the induced normalized pairwise mutual information (Lau et al., 2014, NPMI) on the top ten words in topics as a proxy for interpretability.

Figure 6 shows the NPMI scores for each model. Unsurprisingly, unsupervised models (**LDA**) produce the best topic quality. In contrast, supervised models must balance metadata (i.e., response variable) prediction against capturing word meaning. Consequently, **SLDA** does slightly worse with respect to topic interpretability.

SUP ANCHOR and **ANCHOR** produce the same topic quality consistently on all datasets. Since **SUP ANCHOR** and **ANCHOR** have nearly identical runtime, **SUP ANCHOR** is better suited for supervised tasks because it improves classification without sacrificing interpretability. It is possible that regularization would improve the interpretability of these topics; Nguyen et al. (2014a) show that adding regularization removes overly frequent words from anchor-discovered topics.

The topics produced by the **ANCHOR** and **SUP ANCHOR** algorithms have many similarities. In Table 3, nearly all of the anchor words discovered by **ANCHOR** are also used by **SUP ANCHOR**. These anchor words tend to describe general food types, such as

Model	Anchor Words and Top Words in Topics	
ANCHOR and **SUP ANCHOR**	**pizza burger sushi ice garlic hot amp chicken pork french sandwich coffee cake steak beer fish**	
ANCHOR	**wine**	wine restaurant dinner menu nice night bar table meal experience
	hour	wait hour people minutes line long table waiting worth order
	late	night late ive people pretty love youre friends restaurant open
SUP ANCHOR	**favorite**	love favorite ive amazing delicious restaurant eat menu fresh awesome
	decent	pretty didnt restaurant ordered decent wasnt nice night bad stars
	line	line wait people long tacos worth order waiting minutes taco

Table 3: Comparing topics generated for the YELP dataset: anchor words shared by both **ANCHOR** and **SUP ANCHOR** are listed. Unique anchor words for each algorithm are listed along with the top ten words for that topic. For clarity, we pruned words which appear in more than 3000 documents as these words appear in every topic. The distinct anchor words reflect positive ("favorite") and negative ("line") sentiment rather than less sentiment-specific qualities of restaurants (e.g., restaurants open "late").

"pizza" or "burger", and characterize the YELP dataset well. The similarity of these shared topics explains why both **ANCHOR** and **SUP ANCHOR** achieve similar topic interpretability scores.

To explain the predictive power of **SUP ANCHOR** we must examine the anchor words and topics unique to both algorithms. The anchor words which are unique to **ANCHOR** include a general topic about wine, and two somewhat coherent topics related to time. By adding supervision to the model we get three new anchor words which identify sentiment ranging from extremely positive reviews mentioning a favorite restaurant to extremely negative reviews complaining about long waits.

This general trend is seen across each of the datasets. For example, **ANCHOR** and **SUP ANCHOR** both discover shared topics describing consumer goods, but **SUP ANCHOR** replaces two topics discussing headphones with topics describing "frustrating" products and "great" products. Similarly, in the TRIPADVISOR data, both **ANCHOR** and **SUP ANCHOR** share topics about specific destinations, but only **SUP ANCHOR** discovers a topic describing "disgusting" hotel rooms.

6 Related Work

Improving the scalability of statistical learning has taken many forms: creating online approximations of large batch algorithms (Hoffman et al., 2013; Zhai et al., 2014) or improving the efficiency of sampling (Yao et al., 2009; Hu and Boyd-Graber, 2012; Li et al., 2014).

These insights have also improved supervised topic models. For example, Zhu et al. (2013) train the max-margin supervised topic models **MEDLDA** (Zhu et al., 2009) by reformulating the model such that the hinge loss is included inside a collapsed Gibbs sampler, rather than being applied externally on the sampler using costly SVMs. Using insights from Smola and Narayanamurthy (2010), the samplers run in parallel to train the model. While these advancements improve the scalability of max-margin supervised topic models, the improvement is limited by the fact that the sampling algorithm grows with the number of tokens.

In contrast, this paper explores a different vein of research that focuses on using efficient representations of summary statistics to estimate statistical models. While this has seen great success in unsupervised models (Cohen and Collins, 2014), it has increasingly also been applied to supervised models. Wang and Zhu (2014) show how to use tensor decomposition to estimate the parameters of **SLDA** instead of sampling to find maximum likelihood estimates. In contrast, anchor-based methods rely on non-negative matrix factorization.

We found that a discriminative classifier did not always perform best on the downstream classification task. Zhu et al. (2009) make a comprehensive comparison between **MEDLDA**, **SLDA**, and SVM+LDA, and they show that SVM+LDA performs worse than **MEDLDA** and **SLDA** on binary classification. It could be that better feature preprocessing could improve our performance.

Bag-of-words representations are not ideal for sentiment tasks. Rubin et al. (2012) introduce Dependency LDA which associates individual word tokens with different labels; their model also outperforms linear SVMs on a very large multi-labeled corpus. Latent variable models that consider grammatical structure (Sayeed et al., 2012; Socher et al., 2011; Iyyer et al., 2014) could also be improved through efficient inference (Cohen and Collins, 2014).

7 Discussion

Supervised anchor word topic modeling provides a general framework for learning better topic representations by taking advantage of both word-cooccurrence and metadata. Our straightforward extension (Equation 2) places each word in a vector space that not only captures co-occurrence with other terms but also the interaction of the word and its sentiment, in contrast to algorithms that only consider raw words.

While our experiments focus on binary classification, the same extension is also applicable to multi-class classification.

Moreover, supervised anchor word topic modeling is fast: it inherits the polynomial-time efficiency from the original unsupervised anchor word algorithm. It is also effective: it is better at providing features for classification than unsupervised topic models and also better than supervised topic models with conventional inference.

Our supervised anchor word algorithm offers the ability to quickly analyze datasets without the overhead of Gibbs sampling or variational inference, allowing users to more quickly understand big data and to make decisions. Combining bag-of-words analysis with metadata through efficient, low-latency topic analysis allows users to have deep insights more quickly.

Acknowledgments We thank Daniel Petersen, Jim Martin, and the anonymous reviewers for their insightful comments. This work was supported by the collaborative NSF Grant IIS-1409287 (UMD) and IIS-1409739 (BYU). Boyd-Graber is also supported by NSF grants IIS-1320538 and NCSE-1422492.

References

Anima Anandkumar, Dean P. Foster, Daniel Hsu, Sham Kakade, and Yi-Kai Liu. 2012. A spectral algorithm for latent dirichlet allocation. In *Proceedings of Advances in Neural Information Processing Systems*.

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

David M. Blei and Jon D. McAuliffe. 2007. Supervised topic models. In *Proceedings of Advances in Neural Information Processing Systems*.

David M. Blei, Andrew Ng, and Michael Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3.

Jordan Boyd-Graber and Philip Resnik. 2010. Holistic sentiment analysis across languages: Multilingual supervised latent Dirichlet allocation. In *Proceedings of Empirical Methods in Natural Language Processing*.

Jonathan Chang, Jordan Boyd-Graber, Chong Wang, Sean Gerrish, and David M. Blei. 2009. Reading tea leaves: How humans interpret topic models. In *Proceedings of Advances in Neural Information Processing Systems*.

Shay B. Cohen and Michael Collins. 2014. A provably correct learning algorithm for latent-variable PCFGs. In *Proceedings of the Association for Computational Linguistics*.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. 1998. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28:2000.

Matthew Hoffman, David M. Blei, Chong Wang, and John Paisley. 2013. Stochastic variational inference. In *Journal of Machine Learning Research*.

Yuening Hu and Jordan Boyd-Graber. 2012. Efficient tree-based topic modeling. In *Proceedings of the Association for Computational Linguistics*.

Mohit Iyyer, Peter Enns, Jordan Boyd-Graber, and Philip Resnik. 2014. Political ideology detection using recursive neural networks. In *Proceedings of the Association for Computational Linguistics*.

Nitin Jindal and Bing Liu. 2008. Opinion spam and analysis. In *Proceedings of First ACM International Conference on Web Search and Data Mining*.

Yohan Jo and Alice H. Oh. 2011. Aspect and sentiment unification model for online review analysis. In *Proceedings of ACM International Conference on Web Search and Data Mining*.

Jey Han Lau, David Newman, and Timothy Baldwin. 2014. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *Proceedings of the European Chapter of the Association for Computational Linguistics*.

Moontae Lee and David Mimno. 2014. Low-dimensional embeddings for interpretable anchor-based topic inference. In *Proceedings of Empirical Methods in Natural Language Processing*.

Aaron Q Li, Amr Ahmed, Sujith Ravi, and Alexander J Smola. 2014. Reducing the sampling complexity of topic models. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 891–900. ACM.

David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. 2010. Automatic evaluation of topic coherence. In *Conference of the North American Chapter of the Association for Computational Linguistics*.

Viet-An Nguyen, Jordan Boyd-Graber, and Philip Resnik. 2013. Lexical and hierarchical topic regression. In *Proceedings of Advances in Neural Information Processing Systems*.

Thang Nguyen, Yuening Hu, and Jordan Boyd-Graber. 2014a. Anchors regularized: Adding robustness and extensibility to scalable topic-modeling algorithms. In *Proceedings of the Association for Computational Linguistics*.

Viet-An Nguyen, Jordan Boyd-Graber, and Philip Resnik. 2014b. Sometimes average is best: The importance of averaging for prediction using mcmc inference in topic modeling. In *Proceedings of Empirical Methods in Natural Language Processing*.

Michael Paul and Roxana Girju. 2010. A two-dimensional topic-aspect model for discovering multi-faceted topics. In *Association for the Advancement of Artificial Intelligence*.

J. R. Quinlan. 1986. Induction of decision trees. 1(1):81–106, March.

Daniel Ramage, Susan T. Dumais, and Daniel J. Liebling. 2010. Characterizing microblogs with topic models. In *International Conference on Weblogs and Social Media*.

Timothy N. Rubin, America Chambers, Padhraic Smyth, and Mark Steyvers. 2012. Statistical topic models for multi-label document classification. *Journal of Machine Learning Research*, 88(1-2):157–208, July.

Gerard Salton. 1968. *Automatic Information Organization and Retrieval.* McGraw Hill Text.

Asad B. Sayeed, Jordan Boyd-Graber, Bryan Rusk, and Amy Weinberg. 2012. Grammatical structures for word-level sentiment detection. In *North American Association of Computational Linguistics*.

Alexander Smola and Shravan Narayanamurthy. 2010. An architecture for parallel topic models. *International Conference on Very Large Databases*, 3(1-2):703–710.

Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. 2011. Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions. In *Proceedings of Empirical Methods in Natural Language Processing*.

Ivan Titov and Ryan McDonald. 2008. A joint model of text and aspect ratings for sentiment summarization. In *Proceedings of the Association for Computational Linguistics*.

Yining Wang and Jun Zhu. 2014. Spectral methods for supervised topic models. In *Proceedings of Advances in Neural Information Processing Systems*.

Chong Wang, David Blei, and Li Fei-Fei. 2009. Simultaneous image classification and annotation. In *Computer Vision and Pattern Recognition*.

Hongning Wang, Yue Lu, and Chengxiang Zhai. 2010. Latent aspect rating analysis on review text data: a rating regression approach. In *Knowledge Discovery and Data Mining*.

Limin Yao, David Mimno, and Andrew McCallum. 2009. Efficient methods for topic model inference on streaming document collections. In *Knowledge Discovery and Data Mining*.

Ke Zhai, Jordan Boyd-Graber, Nima Asadi, and Mohamad Alkhouja. 2012. Mr. LDA: A flexible large scale topic modeling package using variational inference in mapreduce. In *Proceedings of World Wide Web Conference*.

Ke Zhai, Jordan Boyd-Graber, and Shay B. Cohen. 2014. Online adaptor grammars with hybrid inference.

Jun Zhu, Amr Ahmed, and Eric P. Xing. 2009. MedLDA: maximum margin supervised topic models for regression and classification. In *Proceedings of the International Conference of Machine Learning*.

Jun Zhu, Xun Zheng, Li Zhou, and Bo Zhang. 2013. Scalable inference in max-margin topic models. In *Knowledge Discovery and Data Mining*.