Supervised topic models for clinical interpretability

Michael C. Hughes[1], **Huseyin Melih Elibol**[1], **Thomas McCoy, M.D.**[2], **Roy Perlis, M.D.**[2], and **Finale Doshi-Velez**[1]

[1]School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA
[2]Massachusetts General Hospital, Boston, MA, USA

Abstract

Supervised topic models can help clinical researchers find interpretable cooccurence patterns in count data that are relevant for diagnostics. However, standard formulations of supervised Latent Dirichlet Allocation have two problems. First, when documents have many more words than labels, the influence of the labels will be negligible. Second, due to conditional independence assumptions in the graphical model the impact of supervised labels on the learned topic-word probabilities is often minimal, leading to poor predictions on heldout data. We investigate penalized optimization methods for training sLDA that produce interpretable topic-word parameters and useful heldout predictions, using recognition networks to speed-up inference. We report preliminary results on synthetic data and on predicting successful anti-depressant medication given a patient's diagnostic history.

1 Introduction

Abundant count data—procedures, diagnoses, meds—are produced during clinical care. An important question is how such data can assist treatment decisions. Standard pipelines usually involve some dimensionality reduction—there are over 14,000 diagnostic ICD9-CM codes alone—followed by training on the task of interest. Topic models such as latent Dirichlet allocation (LDA) (Blei, 2012) are a popular tool for such dimensionality reduction (e.g. Paul and Dredze (2014) or Ghassemi et al. (2014)). However, especially given noise and irrelevant signal in the data, this two-stage procedure may not produce the best predictions; thus many efforts have tried to incorporate observed labels into the dimensionality reduction model. The most natural extension is *supervised LDA* (McAuliffe and Blei, 2007), though other attempts exist (Zhu et al., 2012; Lacoste-Julien et al., 2009).

Unfortunately, a recent survey by Halpern et al. (2012) finds that many of these approaches have little benefit, if any, over standard LDA. We take inspiration from recent work (Chen et al., 2015) to develop an optimization algorithm that prioritizes document-topic embedding functions useful for heldout data and allows a penalized balance of generative and discriminative terms, overcoming problems with traditional maximum likelihood point estimation or more Bayesian approximate posterior estimation. We extend this work with recognition network that allows us to scale to a data set of over 800,000 patient encounters via an approximation to the ideal but expensive embedding required at each document.

2 Methods

We consider models for collections of D documents, each drawn from the same finite vocabulary of V possible word types. Each document consists of a supervised binary label $y_d \in \{0, 1\}$ (extensions to non-binary labels are straightforward) and N_d observed word tokens $x_d = \{x_{dn}\}_{n=1}^{N_d}$, with each word token an indicator of a vocabulary type. We can compactly write x_d as a sparse count histogram, where x_{dv} indicates the *count* of how many words of type v appear in document d.

2.1 Supervised LDA and Its Drawbacks

Supervised LDA (McAuliffe and Blei, 2007) is a generative model with the following log-likelihoods:

$$\log p(x_d|\phi, \pi_d) = \log \text{Mult}(x_d|N_d, \sum_{k=1}^{K} \pi_{dk}\phi_k) = \sum_{v=1}^{V} x_{dv} \log \left(\sum_{k=1}^{K} \pi_{dk}\phi_{kv} \right) \qquad (1)$$

$$\log p(y_d|\pi_d, \eta) = \log \text{Bern}(y_d|\sigma(\eta^T \pi_d)) = y_d \log \sigma(\eta^T \pi_d) + (1 - y_d) \log(1 - \sigma(\eta^T \pi_d))$$

where π_{dk} is the probability of topic k in document d, ϕ_{kv} is the probability of word v in topic k, η_k are coefficients for predicting label y_d from doc-topic probabilities π_d via logistic regression, and $\sigma(\cdot)$ is the sigmoid function. Conjugate Dirichlet priors $p(\pi_d)$ and $p(\phi_k)$ can be easily incorporated.

For many applications, we wish to either make predictions of y_d or inspect the topic-word probabilities ϕ directly. In these cases, point estimation is a simple and effective training goal, via the objective:

$$\max_{\phi, \pi, \eta} w_y \left(\sum_{d=1}^{D} \log p(y_d|\eta, \pi_d) \right) + w_x \left(\log p(\phi) + \sum_{d=1}^{D} \log p(x_d|\pi_d, \phi) + \log p(\pi_d) \right) \qquad (2)$$

We include penalty weights $w_x > 0, w_y > 0$ to allow adjusting the importance of the unsupervised data term and the supervised label term. Taddy (2012) gives a coordinate ascent algorithm for the totally unsupervised objective ($w_x = 1, w_y = 0$), using natural parameterization to obtain simple updates. Similar algorithms exist for all valid penalty weights.

Two problems arise in practice with such training. First, the standard supervised LDA model sets $w_x = w_y = 1$. However, when x_d contains many words but y_d has a few binary labels, the $\log p(x)$ term dominates the objective. We see in Fig. 1 that the estimated topic word parameters ϕ barely change between $w_x = 1, w_y = 0$ and $w_x = 1, w_y = 1$ under this standard training.

Second, the impact of observed labels y on topic-word probabilities ϕ can be negligible. According to the model, when the document-topic probabilities π_d are represented, the variables ϕ are conditionally independent of y. At training time the π_d may be coerced by direct updates using observed y_d labels to make good predictions, but such quality may *not* be available at test-time, when π_d must be updated using ϕ and x_d alone. Intuitively, this problem comes from the objective treating x_d and y_d as "equal" observations when they are not. Our testing scenario always predicts labels y_d from the words x_d. Ignoring this can lead to severe overfitting, particularly when the word weight w_x is small.

2.2 End-to-End Optimization

Introducing weights w_x and w_y can help address the first concern (and are equivalent to providing a threshold on prediction quality). To address the second concern, we pursue gradient-based inference of a modified version of the objective in Eq. (2) that respects the need to use the same embedding of observed words x_d into low-dimensional π_d in both training and test scenarios:

$$\max_{\phi, \eta} w_y \left(\sum_{d=1}^{D} \log p(y_d|f^*(x_d, \phi), \eta) \right) + w_x \left(\sum_{d=1}^{D} \log p(x_d|f^*(x_d, \phi), \phi) \right) \qquad (3)$$

The function f^* maps the counts x_d and topic-word parameters ϕ to the optimal unsupervised LDA proportions π_d. The question, of course, is how to define the function f^*. One can estimate π_d by solving a maximum a-posteriori (MAP) optimization problem over the space of valid $K-$dimensional probability vectors Δ^K:

$$\pi'_d = \max_{\pi_d \in \Delta^K} \ell(\pi_d), \quad \ell(\pi_d) = \log p(x_d|\pi_d, \phi) + \log \text{Dir}(\pi_d|\alpha). \qquad (4)$$

We can compute π'_d via the *exponentiated gradient* algorithm (Kivinen and Warmuth, 1997), as described in (Sontag and Roy, 2011). We begin with a uniform probability vector, and iteratively reweight each entry by the exponentiated gradient until convergence using fixed stepsize $\xi > 0$:

$$\text{init: } \pi_d^0 \leftarrow [\frac{1}{K} \cdots \frac{1}{K}]. \quad \text{until converged: } \pi_{dk}^t \leftarrow \frac{p_{dk}^t}{\sum_{j=1}^{K} p_{dj}^t}, \quad p_{dk}^t = \pi_{dk}^{t-1} \cdot e^{\xi \nabla \ell(\pi_{dk}^{t-1})}. \qquad (5)$$

We can view the final result after $T >> 1$ iterations, $\pi'_d \approx \pi_d^T$, as a *deterministic* function $f^*(x_d, \phi)$ of the input document x_d and topic-word parameters ϕ.

true topics: example docs:

-10.0 10.0 -10.0 -10.0 -1.0 -1.0 -1.0 -1.0 -1.0 0

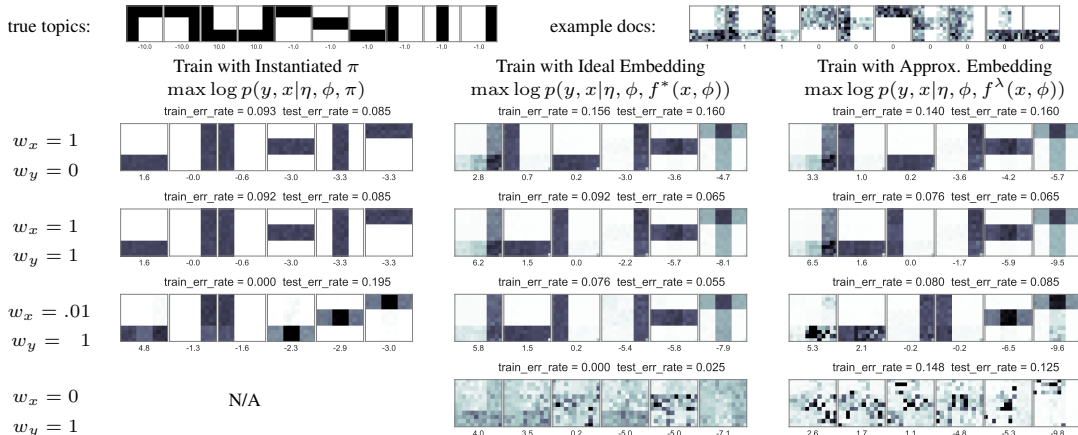|	Train with Instantiated π $\max \log p(y,x\|\eta,\phi,\pi)$	Train with Ideal Embedding $\max \log p(y,x\|\eta,\phi,f^*(x,\phi))$	Train with Approx. Embedding $\max \log p(y,x\|\eta,\phi,f^\lambda(x,\phi))$
$w_x = 1$ $w_y = 0$	train_err_rate = 0.093 test_err_rate = 0.085	train_err_rate = 0.156 test_err_rate = 0.160	train_err_rate = 0.140 test_err_rate = 0.160
	1.6 -0.0 -0.6 -3.0 -3.3 -3.3	2.8 0.7 0.2 -3.0 -3.6 -4.7	3.3 1.0 0.2 -3.6 -4.2 -5.7
$w_x = 1$ $w_y = 1$	train_err_rate = 0.092 test_err_rate = 0.085	train_err_rate = 0.092 test_err_rate = 0.065	train_err_rate = 0.076 test_err_rate = 0.065
	1.6 -0.0 -0.6 -3.0 -3.3 -3.3	6.2 1.5 0.0 -2.2 -5.7 -8.1	6.5 1.6 0.0 -1.7 -5.9 -9.5
$w_x = .01$ $w_y = 1$	train_err_rate = 0.000 test_err_rate = 0.195	train_err_rate = 0.076 test_err_rate = 0.055	train_err_rate = 0.080 test_err_rate = 0.085
	4.8 -1.3 -1.6 -2.3 -2.9 -3.0	5.8 1.5 0.2 -5.4 -5.8 -7.9	5.3 2.1 -0.2 -0.2 -6.5 -9.6
$w_x = 0$ $w_y = 1$	N/A	train_err_rate = 0.000 test_err_rate = 0.025	train_err_rate = 0.148 test_err_rate = 0.125
		4.0 3.5 0.2 -5.0 -5.0 -7.1	2.6 1.7 1.1 -4.8 -5.3 -9.8

Fig. 1: **Toy Bars Case Study.** *Top Row:* True topic-word parameters and example documents. Only the last 6 single bars were used to generate data x, but all 10 topics produce binary labels y given x. *Remainder:* Estimated topic-word parameters and prediction results for different training algorithms (columns). Each row represents a setting of the penalty weights in the objective function: w_y weights the supervised loss term, while w_x weights the unsupervised data term. We perform 3 separate initializations of each method with $K = 6$ topics and report the one with best (lowest) training error rate. Test set error rates are computed by using only the observed words x_d and the estimated topics ϕ as input for each document, then computing π_d via $f^*(x_d, \phi)$ in Eq. (5).

End-to-end training with ideal embedding. The procedure above does not directly lead to a way to estimate ϕ to maximize the objective in Eq. (3). Recently, Chen et al. (2015) developed *backpropagation supervised LDA* (BP-sLDA), which optimizes Eq. (3) under the extreme discriminative setting $w_y = 1, w_x = 0$ by pushing gradients through the exponentiated gradient updates above. We can further estimate ϕ under any valid weights with this objective. We call this "training with ideal embedding", because the embedding is optimal under the unsupervised model.

End-to-end training with approximate embedding. Direct optimization of the ideal embedding function f^*, as done by Chen et al. (2015), has high implementation complexity and runtime cost. We find in practice that each document requires dozens or even hundreds of the iterations in Eq. (5) to converge reasonably. Performing such iterations at scale and back-propagating through them is possible with careful C++ implementation but will still be the computational bottleneck. Instead, we suggest an approximation: use a simpler embedding function $f^\lambda(x_d, \phi)$ which has been trained to approximate the ideal embedding. Initial experiments suggest a simple multi-layer perceptron (MLP) recognition network architecture with one hidden layer of size $H \approx 50$ does reasonably well:

$$f_k^\lambda(x_d, \phi) = \mathrm{softmax}\Big(\sum_{h=1}^H \lambda_{hk}^{\mathrm{output}} \sigma(\sum_{v=1}^V \lambda_{hv}^{\mathrm{hidden}} x_{dv} \phi_{kv}) \Big). \tag{6}$$

During training, we periodically pause our gradient descent over η, ϕ and update λ to minimize a KL-divergence loss between the approximate embedding f^λ and the ideal, expensive embedding f^*.

3   Case study: Toy bars data

To understand how different training objectives impact both predictive performance and interpretability of topic-word parameters, we consider a version of the toy bars dataset inspired by Griffiths and Steyvers (2004), but changed so the optimal ϕ parameters are distinct for unsupervised LDA and supervised LDA objectives. Our dataset has 144 vocabulary words visualized as pixels in a square grid in Fig. 1. To generate the observed words x, we use 6 true topics: 3 horizontal bars and 3 vertical bars. However, we generate label y_d using an expanded set of 10 topics, where the extra topics are *combinations* of the 6 bars. Some combinations produce positive labels, but no single bar does. We train multiple initializations of each possible training objective and penalty weight setting, and show the best run of each method in Fig. 1. Our conclusions are listed below:

Standard training that instantiates π can either ignore labels or overfit. Fig. 1's first column shows two problematic behaviors with the optimization objective in Eq. (2). First, when $w_x = 1$, the topic-word parameters are basically identical whether labels are ignored ($w_y = 0$) or included

(prevalence) DRUG	approx f^λ $w_x = 0$ $w_y = 1$	ideal f^* $w_x = 0$ $w_y = 1$	ideal f^* $w_x = 0.01$ $w_y = 1$	ideal f^* $w_x = 1$ $w_y = 1$	ideal f^* $w_x = 1$ $w_y = 0$	BoW
(0.215) citalopram	0.65	0.64	0.63	0.62	0.61	0.72
(0.135) fluoxetine	0.66	0.64	0.64	0.63	0.63	0.76
(0.133) sertraline	0.66	0.66	0.63	0.63	0.63	0.75
(0.119) trazodone	0.64	0.66	0.64	0.61	0.62	0.65
(0.115) bupropion	0.64	0.64	0.59	0.56	0.58	0.71
(0.070) amitriptyline	0.77	0.76	0.77	0.75	0.75	0.78
(0.059) venlafaxine	0.64	0.62	0.62	0.61	0.61	0.73
(0.059) paroxetine	0.68	0.73	0.74	0.76	0.75	0.76
(0.047) mirtazapine	0.70	0.69	0.70	0.71	0.70	0.67
(0.046) duloxetine	0.71	0.69	0.70	0.69	0.70	0.74
(0.041) escitalopram	0.65	0.62	0.61	0.61	0.61	0.80
(0.038) nortriptyline	0.71	0.73	0.70	0.70	0.71	0.71
(0.007) fluvoxamine	0.70	0.72	0.74	0.77	0.76	0.93
(0.007) imipramine	0.40	0.56	0.50	0.48	0.48	0.82
(0.006) desipramine	0.47	0.57	0.54	0.57	0.54	0.72
(0.003) nefazodone	0.71	0.65	0.71	0.72	0.72	0.80

Table 1: Heldout AUC scores for prediction of 16 drugs for treating depression. Each drug decision is independent, since multiple drugs might be given to a patient. "BoW" is logistic regression using x_d as features.

($w_y = 1$). Second, when the observed data is weighted very low ($w_x = 0.01$), we see severe overfitting, where the learned embeddings at training time are not reproducible at test time.

Ideal end-to-end training can be more predictive but has expensive runtime. In contrast to the problems with standard training, we see in the middle column of Fig. 1 that using the ideal test-time embedding function f^* also during training can produce much lower error rates on heldout data. Varying the data weight w_x interpolates between interpretable topic-word parameters ϕ and good predictions. One caveat to ideal embedding is its expensiveness: Completing 100 sweeps through this 1000 document toy dataset takes about 2.5 hours using our vectorized pure Python with `autograd`.

Approximate end-to-end training is much cheaper and often does as well. We see in the far right column of Fig. 1 that using our proposed approximate embedding f^λ often yields similar predictive power and interpretable topic-word parameters when $w_x > 0$. Furthermore, it is about 3.6X faster to train due to avoiding the expensive embedding iterations at every document.

4   Case study: Predicting drugs to treat depression

We study a cohort of 875080 encounters from 49322 patients drawn from two large academic medical centers with at least one ICD9 diagnostic code for major depressive disorder (ICD9s 296.2x or 3x or 311, or ICD10 equivalent). Each included patient had an identified successful treatment: a prescription repeated at least 3 times in 6 months with no change.

We extracted all procedures, diagnoses, labs, and meds from the EHR (22,000 total codewords). For each encounter, we built x_d by concatenating count histograms from the last three months and all prior history. To simplify, we reduced this to the 9621 codewords that occurred in at least 1000 distinct encounters. The prediction goal was to identify which of 16 common anti-depressants drugs would be successful for each patient. (Predicting all 25 primaries and 166 augments is future work).

Table 1 compares each method's area-under-the-ROC-curve (AUC) with $K = 50$ topics on a held-out set of 10% of patients. We see that our training algorithm using the ideal embedding f^* improves its predictions over a baseline unsupervised LDA model as the weight w_x is driven to zero. Our approximate embedding f^λ is roughly 2-6X faster, allowing a full pass through all 800K encounters in about 8 hours, yet offers competitive performance on many drug tasks except for those like desipramine or imipramine for which less than 1% of encounters have a positive label. Unfortunately, our best sLDA model is inferior to simple bag-of-words features plus a logistic regression classifier (rightmost column "BoW"), which we guess is due to local optima. To remedy this, future work can explore better initializations such as tensor methods (Arora et al., 2013).

References

S. Arora, R. Ge, Y. Halpern, D. Mimno, A. Moitra, D. Sontag, Y. Wu, and M. Zhu. A practical algorithm for topic modeling with provable guarantees. In *International Conference on Machine Learning*, 2013.

D. M. Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.

J. Chen, J. He, Y. Shen, L. Xiao, X. He, J. Gao, X. Song, and L. Deng. End-to-end learning of LDA by mirror-descent back propagation over a deep architecture. In *Neural Information Processing Systems*, 2015.

M. Ghassemi, T. Naumann, F. Doshi-Velez, N. Brimmer, R. Joshi, A. Rumshisky, and P. Szolovits. Unfolding physiological state: mortality modelling in intensive care units. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 75–84. ACM, 2014.

T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 2004.

Y. Halpern, S. Horng, L. A. Nathanson, N. I. Shapiro, and D. Sontag. A comparison of dimensionality reduction techniques for unstructured clinical text. In *ICML workshop on clinical data analysis*, 2012.

J. Kivinen and M. K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1–63, 1997.

S. Lacoste-Julien, F. Sha, and M. I. Jordan. DiscLDA: Discriminative learning for dimensionality reduction and classification. In *Neural Information Processing Systems*, 2009.

J. D. McAuliffe and D. M. Blei. Supervised topic models. In *Neural Information Processing Systems*, 2007.

M. J. Paul and M. Dredze. Discovering health topics in social media using topic models. *PLoS One*, 9(8): e103408, 2014.

D. Sontag and D. Roy. Complexity of inference in latent dirichlet allocation. In *Neural Information Processing Systems*, 2011.

M. Taddy. On estimation and selection for topic models. In *Artificial Intelligence and Statistics*, 2012.

J. Zhu, A. Ahmed, and E. P. Xing. MedLDA: maximum margin supervised topic models. *The Journal of Machine Learning Research*, 13(1):2237–2278, 2012.