Regularizing tensor decomposition methods by optimizing pseudo-data

Omer Gottesman [1] **Finale Doshi-Velez** [1]

1. Introduction

Tensor decomposition methods have recently gained popularity as ways of performing inference for latent variable models (Anandkumar et al., 2014). The interest in these methods is motivated by the fact that they come with theoretical global convergence guarantees in the limit of infinite data (Anandkumar et al., 2012; Arora et al., 2013). However, a main limitation of these methods is that there is no easy way to enforce prior information on model parameters to improve inference when the amount of data is limited.

Previous works attempted to alleviate this drawback by modifying existing tensor decomposition methods to incorporate specific constraints such as topic sparsity (Sun et al., 2015) or incorporate modeling assumptions such as the existence of anchor words (Arora et al., 2013; Nguyen et al., 2014). All these methods require an ad-hoc modification of the algorithms to incorporate the specific structure of the prior information. Furthermore, many of these methods impose hard constraints on the learned model, which may be unhelpful or even detrimental when we have a lot of data—framed in the context of Bayesian intuition, when we have a lot of data, we want our methods to allow the evidence to overwhelm our priors.

We propose an alternative approach which addresses both of these issues. It is easily generalizable to any structure of prior information on the model parameters, and applies prior information to inference only when the data is insufficient. We adopt the common view of Bayesian priors as representing "pseudo-observations" of artificial data which biases our learned model parameters towards our prior belief (Bishop, 2006). We apply the tensor decomposition method described in Anandkumar et al. (2014) to data sets comprised of the actual data and an artificial set of pseudo-data. We use automatic differentiation (Baydin et al., 2015; Maclaurin et al., 2015) to optimize our pseudo-data such that it minimizes a cost function balancing two terms - one for imposing our prior knowledge on the inferred model parameters, and one for keeping the pseudo-data as similar as possible to the actual data.

We describe a straightforward way to use our method to impose arbitrary regularizers on the inferred models, and demonstrate it for several synthetic and real world examples.

2. Background

Tensor decomposition methods Tensor decomposition methods (TDM) learn the parameters of a latent variable model given as a matrix $A \in \mathbb{R}^{D\times K}$, where D is the dimensionality of the data and K the number of latent variables. The columns of A could represent the means of Gaussian mixtures or feature assignment probabilities in topic models such as Latent Dirichlet Allocation (LDA (Blei et al., 2003)), in which case A is referred to as the topics matrix. TDM works by leveraging the relationship between the empirical moments of the data and the latent parameters of the model. Specifically, A is learned by matching the theoretical moments of the model,

$$M_2 = \sum_{k=1}^{K} \beta_k a_k a_k^T, \tag{1}$$

$$M_3 = \sum_{k=1}^{K} \gamma_k a_k \otimes a_k \otimes a_k, \tag{2}$$

with their empirical estimates, which can be computed from data. Here a_k is the k^{th} column of A.

The decomposition itself is performed in two stages. First, we compute a whitening matrix W such that $W^T M_2 W = I$, and use W to project M_3 to a $\mathbb{R}^{K\times K\times K}$ tensor which has an orthogonal decomposition. We then use the tensor power method to decompose the reduced third order tensor. (For more details we refer the reader to Anandkumar et al. (2014)).

Priors and regularization of latent variables Quite often, we have some knowledge or expectation regarding the structure of the latent topics that we are trying to learn. In the Bayesian setting, such prior knowledge is encoded as the distribution from which the hidden parameters are drawn. Such priors can be useful to inform us of which learned

[1] Paulson School of Engineering and Applied Sciences, Harvard University. Correspondence to: Omer Gottesman <gottesman@fas.harvard.edu>.

Workshop on Modern Trends in Nonconvex Optimization for Machine Learning. The 35th International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

hidden parameters are more likely when data is too limited to make confident predictions, but would play a smaller roll in inference as more data is collected.

Alternatively, even without information about the distribution from which the hidden parameters are drawn, sometimes some characteristics of these parameters can make them more interpretable, such as sparsity or diversity. In the absence of enough data to have high confidence in our learned parameters, we would like to regularize the parameters to at least have these desired characteristics.

3. Pseudo-data for regularization

We wish to impose regularizers in a way which is compatible with tensor decomposition methods. The challenge in doing that is due to the fact that tensor decomposition methods only give a point estimate of the topics, rather than a posterior prediction. In methods such as MAP estimation we can use an objective function that balances a prediction term (such as log likelihood of training data) and the prior. For spectral methods, however, when we modify the inferred topics to be more consistent with our priors we have no notion of how much predictive quality we sacrifice.

To solve this problem we draw on the intuition which is often used when describing Bayesian priors, of viewing the priors as encoding pseudo-observations which match our expectation of the data. Given a prior or a regularizer, we can choose our pseudo-data in a way which will drive the inferred topics towards a form which is more in line with our expectations, and balance that with the requirement that our pseudo-data will be as likely as possible under the model we infer using only the real data.

Formally, we optimize a pseudo-dataset, $X_P \in \mathbb{R}^{D \times N_p}$, with respect to a cost function, $f(X_T, X_P)$. The function we optimize is

$$\begin{aligned} f(X_T, X_P, \lambda) = &- \log p(X_P | A(X_T)) \\ &+ \lambda f_r(A(X_{T \cup P})), \end{aligned} \tag{3}$$

where $A(X_T)$ and $A(X_{T \cup P})$ are the topic matrices learned by the TDM using either only the real training data, X_T, or a combination of both the real and pseudo-data, $X_{T \cup P}$, respectively. The cost, f_r, could be any regularizer encoding our prior knowledge of the data, and λ a parameter which denotes the relative importance of the topics conforming to the prior, compared with how unlikely we allow our pseudo-data to be.

To perform inference using this cost function we need to choose two parameters. Apart from λ, we also need to choose the number of pseudo-data points, N_p. The number of pseudo-data points N_p represents how much confidence we put in our prior knowledge of the topics structure. An advantage of this approach is that a particular choice of N_p limits how much the pseudo-data can influence the topics — as the number of training samples N_t increases, even in the limit of infinite λ, the maximum effect of the pseudo-data on the inferred topics diminishes. The dominance of the training data as $N_t \gg N_p$ represents the tendency to put less weight on the pseudo-data as more data is collected. This is analogous to the tendency of a sharply peaked likelihood to overwhelm the prior in Bayesian inference. The convergence of Algorithm 1 is formalized in Theorem 1.

Algorithm 1 Tensor Decomposition Regularization

```
Input: $X_t$, $N_p$, $\lambda$
$A(X_t) \leftarrow TDM(X_t)$ % TDM - tensor decomposition method
Draw $X_p$ from $p(X_p | A(X_t))$
while $X_p$ not converged do
  $X_p \leftarrow X_p - ADAM(\nabla_{X_p} f(X_t, X_p, \lambda))$
end while
$A(X_{t+p}) \leftarrow TDM(X_{t+p})$
Return: $A(X_{t+p})$
```

Theorem 1. *In the limit of $N_t \to \infty$, for any finite λ and any finite N_p, the results of Algorithm 1 converge to the standard TDM.*

Proof. The result of Algorithm 1 is a decomposition of the tensor estimates from $X_{T \cup P}$. These can be written as $\hat{M}_i = \frac{N_t}{N_t + N_p} \hat{M}_{i,T} + \frac{N_p}{N_t + N_p} \hat{M}_{i,P}$, where $\hat{M}_{i,T}$ and $\hat{M}_{i,P}$ are the i^{th} order ($i \in \{2, 3\}$) moments computed only from the training and pseudo-data respectively. In the limit $N_t \to \infty$, $\hat{M}_i \to \hat{M}_{i,T}$. That is, Algorithm 1 returns the TDM result of the moments computed only using the training data, thus completing the proof. □

Algorithm 1 describes the algorithm for performing the entire process of learning the regularized topics. TDM denotes the topics learned using the tensor decomposition algorithm, and ADAM is the gradient descent step based on the ADAM algorithm (Kingma and Ba, 2014). We compute the gradients of the cost function using the Python Autograd package (Maclaurin et al., 2015).

4. Experiments

In the following section we demonstrate the tensor decomposition regularization method for two examples with semi-synthetic and real data. We focus our experiments on the LDA model, and first demonstrate that given prior knowledge of the latent topics structure, our method can be used to improve inference when data is sparse and ignore the prior knowledge when data is abundant. We then use the method to improve the interpretability of topic models for classification of medical science papers by regularizing the topics to group together articles with similar headers (where similarity is defined by distance on a hierarchy tree).

In Appendix 1 we present results for two more synthetic toy datasets, where we use pseudo-data to regularize for topic independence (orthogonality) and sparsity. We also demonstrate that the sparsity could be used to improve inference for noisy data.

4.1. Transfer learning with semi-synthetic ASD patient data

We first demonstrate that when we have prior knowledge about the latent structure of our data, our method can use it to improve inference when training data is scarce, and ignores it when the training dataset is large. We demonstrate the method on simulated autism spectrum disorder (ASD) data. The data is referred to as semi-synthetic because it is a simulated dataset, but the topics used to simulate it are learned from real data, and we therefore expect these topics to include the sparsity and correlations which are representative of the true data (Arora et al., 2013). We use electronic health records of $D = 64$ common diagnoses of children with autism (Doshi-Velez et al., 2014b). We use the real data to learn two topic matrices ($K = 4$) representing the symptoms common for two age groups, 6 to 7 and 8 to 9. We make the assumption that the symptoms of the two age groups share some similarities, and that we can transfer our knowledge about one age group to better learn the characteristics of the other. In other words, we expect the topics learned for one age group to be an informative prior for inference on the other group.

To test our method over a range of N_t, we sample observations from the LDA model using the topics matrix of ages 6 to 7. We refer to this topics matrix as A_{true}, as it represents the true topics we wish to learn. As our regularizer function, we choose $f_r = ||A(X_{T\cup P}) - A_{prior}||_2$, where A_{prior} is the topics matrix representing ages 8 to 9. In other words, we use our method to generate pseudo-data which will force our learned topics from data on ages 6 to 7 to be as similar as possible to the topics learned from ages 8 to 9. In Table 1 we demonstrate our results for different values of N_t and $N_p = 30$. We see that when training data is limited ($N_t = 100$), our method allows for tranfer learning which significantly reduces the topics reconstruction error, but has little effect on inference when $N_t = 10000$ and the training data overwhelms the pseudo-data.

Table 1. Transfer learning with ASD data

	$\|\|A(X_T) - A_{true}\|\|_2$	$\|\|A(X_{T\cup P}) - A_{true}\|\|_2$
$N_t = 100$	0.55	0.36
$N_t = 10000$	0.11	0.11

4.2. Real data: Medical Subject Headings hierarchy

The National Library of Medicine (NLM) uses a hierarchically-organized terminology of medical subject headings (MeSH) for indexing medical articles[1]. Every article is labeled using several headings, and headings are given an assignment on a hierarchical tree, in which the root represents a general topic, and headings become more specific further down the tree. An example of three generations of headings in the tree is “Adult [M01.060.116]”, “Aged [M01.060.116.100]” and “Aged, 80 and over [M01.060.116.100.080]”. Formally, each three digit number in the full heading represents a node, and the periods separating them represent edges.

Topic modeling on subject headings of papers can help in identifying publication and research trends by finding headings which occur together frequently. Because articles are hand labeled, there is sometimes significant inconsistency in labeling—for example, a particular paper could be given each of the three headings present in the example, depending on the particular person who labeled it (Doshi-Velez et al., 2014a). A useful property of the topics which could help avoid missing information due to inconsistency in labeling is pushing for topics with headings which are close on the tree.

For the task of regularizing the model we learn for the MeSH data we choose a regularizer which is more complex than simply regularizing for sparsity or diversity (see Appendix 1), as we want to use our knowledge about the hierarchical indexing structure of the data. We use the following regularizer to achieve this property—

$$f_r(A) = -\sum_{k=1}^{K}(\sum_{i\neq j} A_{ik}A_{jk}O_{ij}^{-1}). \tag{4}$$

where O_{ij} is the distance on the tree between the i^{th} and j^{th} headings. Minimizing this regularizer rewards topics with several headings which are close to each other on the tree ($-O_{ij}^{-1}$ is more negative), while simultaneously pushing for topics with at least more than one highly weighted heading. We discuss the choice of f_r further in Appendix 1.3.

We perform our experiments on a labeled dataset of research articles on statins—a group of drugs used for treating cardiovascular disease (Cohen et al., 2006). Our training data consists of $N_t = 500$ documents using the $D = 300$ most common headings, and we learn a topic model with $K = 3$ topics.

In Figure 1 (left) we plot the value of the regularizer for the learned topics with different values of N_p and λ, and

[1] https://www.nlm.nih.gov/mesh/

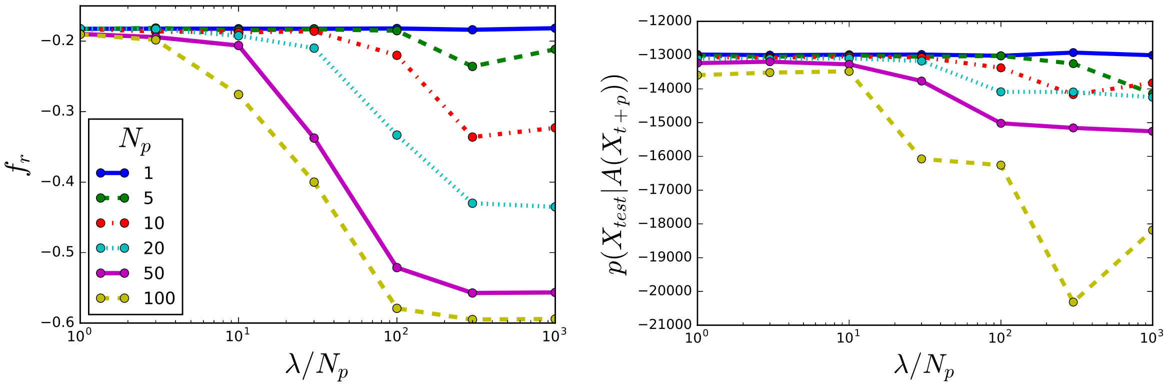

Figure 1. **Real data (MeSH) - Regularizing for interpretability.** f_r value (left) and log-likelihood of held-out test set (right) for regularized topics with different values of N_p and λ/N_p. The regularization function allows us to improve interpretability (lower f_r) at the cost of predictive power (lower log-likelihood on test data). The examples with $N_p = 20$ and $N_p = 50$ demonstrate it is possible to obtain significant improvement in interpretability with a relatively small decrease in predictive power.

observe that similarly to the previous example, a given number of pseudo-observations is limited in the effect it can have on the topics, no matter how large λ is. In Figure 1 (right) we plot the log-likelihood of a held out test set of 500 documents, using the learned topics. We see that in most cases, the interpretability of the topics comes at a relatively low cost in terms of prediction accuracy, which grows as N_p is increased.

In appendix 1.3 we analyze the topics learned with and without regularization and demonstrate that optimizing f_r indeed leads to learning more interpretable topics.

5. Discussion

The tensor decomposition regularization algorithm requires two parameters—N_p and λ. We demonstrated throughout this paper that for a given N_p, at some point increasing λ no longer influences the final regularized topics. The intuition behind this saturation is that there is a limit to how big of an effect a small fraction of the data can have on the learned topics. In practice, this means we can choose λ to be in the saturated regime (high λ), and only tune N_p to control the strength of our regularizer, reducing the number of parameter choices we are required to make.

The computational complexity of the tensor decomposition algorithm as it appears in Anandkumar et al. (2014) is $\mathcal{O}(D^3)$, where the limiting step is in computing $\hat{M}_3 \in \mathbb{R}^{D\times D\times D}$ and whitening it to the $\mathbb{R}^{K\times K\times K}$ tensor, $\hat{M}_{3,w}$. Zou et al. (2013) demonstrated that for sparse data, the tensor decomposition can be performed in $\mathcal{O}(DK + nnz(X))$ where $nnz(X)$ is the number of non-zero elements in X. Because our algorithm is based on differentiating the results of the tensor decomposition algorithm with respect to its input, if we wish for our algorithm to be flexible enough to impose any regularizer, X_P will generally not be sparse, and we cannot use the method introduced in Zou et al. (2013). Instead, we first compute $\hat{M}_2$, and use the whitening matrix, W to whiten the data. We then compute $\hat{M}_{3,w}$ directly from the whitened data X_w, and never explicitly compute $\hat{M}_3$. This makes the limiting step in the algorithm the computation and SVD of $\hat{M}_2$, and the computational complexity of the algorithm is $\mathcal{O}(D^2)$.

6. Conclusion

We introduced an algorithm to regularize tensor decomposition methods for learning latent variable model. The method is versatile and can easily be applied to a variety of models and regularizers. A strength of this method lies in its ability to use regularization and prior knowledge to improve learning when data is limited, and ignore our prior beliefs when data is abundant, much like the effect of priors in a Bayesian setting. An open question remaining is drawing a tighter connection between our method and Bayesian inference, which would lead to a more quantitative approach to selecting the algorithm parameters and formulating the regularizers as probability distributions over the latent topics.

References

Anima Anandkumar, Dean P Foster, Daniel J Hsu, Sham M Kakade, and Yi-Kai Liu. A spectral algorithm for latent dirichlet allocation. In *Advances in Neural Information Processing Systems*, pages 917–925, 2012.

Animashree Anandkumar, Rong Ge, Daniel J Hsu, Sham M Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. *Journal of Machine Learning Research*, 15(1):2773–2832, 2014.

Sanjeev Arora, Rong Ge, Yonatan Halpern, David M Mimno, Ankur Moitra, David Sontag, Yichen Wu, and Michael Zhu. A practical algorithm for topic modeling with provable guarantees. In *ICML (2)*, pages 280–288, 2013.

Atilim Gunes Baydin, Barak A Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. *arXiv preprint arXiv:1502.05767*, 2015.

Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

Aaron M Cohen, William R Hersh, K Peterson, and Po-Yin Yen. Reducing workload in systematic review preparation using automated citation classification. *Journal of the American Medical Informatics Association*, 13(2):206–219, 2006.

F Doshi-Velez, B Wallace, and R Adams. Graph-sparse lda: a topic model with structured sparsity, arxiv preprint. *arXiv*, 1410, 2014a.

Finale Doshi-Velez, Yaorong Ge, and Isaac Kohane. Comorbidity clusters in autism spectrum disorders: an electronic health record time-series analysis. *Pediatrics*, 133(1):e54–e63, 2014b.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

D Maclaurin, D Duvenaud, M Johnson, and RP Adams. Autograd: Reverse-mode differentiation of native python. http://github.com/HIPS/autograd, 2015.

Thang Nguyen, Yuening Hu, and Jordan Boyd-Graber. Anchors regularized: Adding robustness and extensibility to scalable topic-modeling algorithms. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 359–369, 2014.

Will Wei Sun, Junwei Lu, Han Liu, and Guang Cheng. Provable sparse tensor decomposition. *arXiv preprint arXiv:1502.01425*, 2015.

James Y Zou, Daniel J Hsu, David C Parkes, and Ryan P Adams. Contrastive learning using spectral methods. In *Advances in Neural Information Processing Systems*, pages 2238–2246, 2013.