Sometimes Average is Best: The Importance of Averaging for Prediction using MCMC Inference in Topic Modeling

Abstract

Markov chain Monte Carlo (MCMC) approximates the posterior distribution of latent variable models of text by generating a large number of samples and averaging over them. In practice, however, it is often more convenient to cut corners, using only a single sample or following a suboptimal averaging strategy. In this paper, we systematically study different strategies for averaging MCMC samples and show empirically that performing averaging properly leads to significant improvements in prediction performance.

1 Introduction

Probabilistic topic models are powerful methods to uncover hidden thematic structures in text by projecting each document into a low dimensional space spanned by a set of *topics*, each of which is a distribution over words. Topic models such as latent Dirichlet allocation (LDA) (Blei et al., 2003) and its extensions discover these topics from text, which allows for effective exploration, analysis and summarization of the otherwise unstructured corpora (Blei, 2012; Blei, 2014).

In addition to exploratory data analysis, a typical goal of topic models is prediction. Given a set of unannotated training data, *unsupervised topic models* try to learn good topics that can generalize to unseen text. *Supervised topic models* jointly capture both the text and associated *metadata* such as a continuous response variable (Blei and McAuliffe, 2007; Zhu et al., 2009), single label (Rosen-Zvi et al., 2004; Lacoste-Julien et al., 2008; Wang et al., 2009) or multiple labels (Ramage et al., 2009; Ramage et al., 2011) to predict metadata from text.

Crucial in probabilistic topic modeling is the process of estimating the posterior distribution. Exact computation of the posterior is often intractable, which motivates various approximate inference techniques (Asuncion et al., 2009). One popular approach is Markov chain Monte Carlo (MCMC),

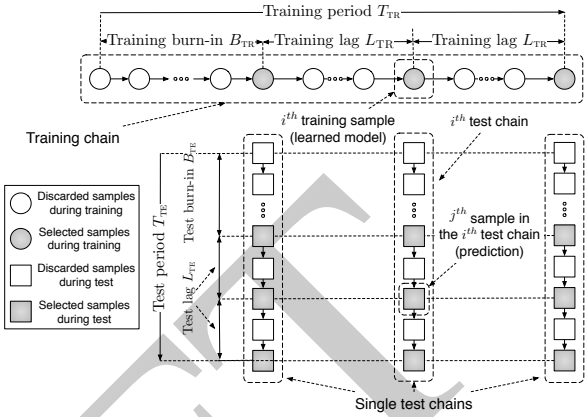

Figure 1: Training and test chains in MCMC.

a class of inference algorithms to approximate the target posterior distribution. To make prediction, MCMC algorithms generate samples on training data to estimate corpus-level latent variables, and use them to generate samples to estimate document-level latent variables for test data. The underlying theory requires averaging on both training and test samples, but in practice it is often convenient to cut corners: either skip averaging entirely by using just the values of the last sample, or use a single training sample and average over test samples.

This paper systematically studies non-averaging and averaging strategies when performing predictions using MCMC in topic modeling (Section 2). Using popular unsupervised (LDA in Section 3) and supervised (SLDA in Section 4) topic models via thorough experimentation, we show empirically that cutting corners on averaging leads to consistently poorer prediction.

2 Learning and Predicting with MCMC

While reviewing all of MCMC is beyond the scope of this paper, we need to briefly review key concepts.[1] To estimate a target density $p(x)$ in a high-dimensional space $\mathcal{X}$, MCMC generates samples $\{x_t\}_{t=1}^{T}$ while exploring $\mathcal{X}$ using Markov assumption. Under this assumption, sample x_{t+1} depends

[1] For more details please refer to (Neal, 1993; Andrieu et al., 2003; Resnik and Hardisty, 2010).

on sample x_t only, forming a *Markov chain*, which allows the sampler spends more time in the most important regions of the density. Two concepts control sample collection:

Burn-in B: Depending on the initial value of the Markov chain, MCMC algorithms take time to reach the target distribution. Thus, in practice, samples before a burn-in period B are often discarded.

Sample-lag L: Averaging over samples to estimate the target distribution requires i.i.d. samples. However, future samples depend on the current samples (i.e., the Markov assumption). To avoid autocorrelation, we discard all but every L samples.

2.1 MCMC in Topic Modeling

As generative probabilistic models, topic models define a joint distribution over latent variables and observable evidence. In our setting, the latent variables consist of corpus-level *global* variables $\boldsymbol{g}$ and document-level *local* variables $\boldsymbol{l}$; while the evidence consists of words $\boldsymbol{w}$ and additional metadata $\boldsymbol{y}$—the latter omitted in unsupervised models.

During training, MCMC estimates the posterior $p(\boldsymbol{g}, \boldsymbol{l}^{\text{TR}} \,|\, \boldsymbol{w}^{\text{TR}}, \boldsymbol{y}^{\text{TR}})$ by generating a *training Markov chain* of T_{TR} samples.[2] Each training sample i provides a set of fully realized global latent variables $\hat{\boldsymbol{g}}(i)$, which can be used to generate samples for test data. During test time, given a learned model from training sample i, we generate a *test Markov chain* of T_{TE} samples to estimate the local latent variables $p(\boldsymbol{l}^{\text{TE}} \,|\, \boldsymbol{w}^{\text{TE}}, \hat{\boldsymbol{g}}(i))$ of test data. Each sample j of test chain i provides a fully estimated local latent variables $\hat{\boldsymbol{l}}^{\text{TE}}(i, j)$, which can be used to make prediction.

Figure 1 shows an overview. To reduce the effects of unconverged and autocorrelated samples, during training we use a burn-in period of T_{TR} and a sample-lag of L_{TR} iterations. We use $\mathcal{T}_{\text{TR}}(t) = \{i \,|\, i \in (B_{\text{TR}}, t] \wedge (i - B_{\text{TR}}) \mod L_{\text{TR}} = 0\}$ to denote the set of indices of the selected models up to training iteration t. Similarly, B_{TE} and L_{TE} are the test burn-in and sample-lag. The set of indices of selected samples in test chains is $\mathcal{T}_{\text{TE}} = \{j \,|\, j \in (B_{\text{TE}}, T_{\text{TE}}] \wedge (j - B_{\text{TE}}) \mod L_{\text{TE}} = 0\}$.

[2]We omit hyperparameters for clarity. We use TR and TE to denote "training" and "test" respectively, and use i and j respectively to index the iterations of the training and test Markov chains.

2.2 Averaging Strategies

We use $S(i, j)$ to denote the prediction obtained from sample j of the test chain i. We now discuss different strategies to obtain the final prediction:

Single Final (SF): uses the final sample in a single test chain t.

$$S_{\text{SF}}(t) = S(t, T_{\text{TE}}) \tag{1}$$

Single Average (SA): averages over multiple samples in a single test chain t. This is a common averaging strategy in which a point estimate of the global latent variables is obtained at the end of the training chain. Then, a single test chain is generated on the test data and multiple samples of this test chain are averaged to obtain the final prediction (Chang, 2012; Jiang et al., 2012; Zhu et al., 2014).

$$S_{\text{SA}}(t) = \frac{1}{|\mathcal{T}_{\text{TE}}|} \sum_{j \in \mathcal{T}_{\text{TE}}} S(T_{\text{TR}}, j) \tag{2}$$

Multiple Final (MF): averages over the last samples of multiple test chains obtained from multiple models up to iteration t in the training chain.

$$S_{\text{MF}}(t) = \frac{1}{|\mathcal{T}_{\text{TR}}(t)|} \sum_{i \in \mathcal{T}_{\text{TR}}(t)} S(i, T_{\text{TE}}) \tag{3}$$

Multiple Average (MA): averages over all the different samples of multiple test chains obtained from multiple models up to training iteration t .

$$S_{\text{MA}}(t) = \frac{1}{|\mathcal{T}_{\text{TR}}(t)|} \frac{1}{|\mathcal{T}_{\text{TE}}|} \sum_{i \in \mathcal{T}_{\text{TR}}(t)} \sum_{j \in \mathcal{T}_{\text{TE}}} S(i, j) \tag{4}$$

3 Unsupervised Topic Models

We evaluate the predictive performance of the original unsupervised topic model LDA using different averaging strategies in Section 2.

LDA: Proposed by Blei et al. in 2003, LDA posits that each document d is a multinomial distribution θ_d over K topics, each of which is a multinomial distribution ϕ_k over the vocabulary. LDA's global latent variables are topics $\{\phi_k\}_{k=1}^K$ and the local latent variables for each document d are topic proportions θ_d.

During training, we use collapsed Gibbs sampling to assign each token to a topic (Steyvers and Griffiths, 2006) and estimate topic $\hat{\phi}(i))$ at each training sample i.[3] During test, we sample the topic

[3]More details can be found in the supplementary material.

Figure 2: Perplexity of LDA using different averaging strategies

assignment of each token to estimate the topic proportion $\hat{\theta}_d^{\text{TE}}(i,j)$ for each test document d. The predicted likelihood of each test token is the averaged prediction: $S(i,j) = p(w_{d,n}^{\text{TE}} \mid \hat{\theta}_d^{\text{TE}}(i,j), \hat{\phi}(i)) = \sum_{k=1}^{K} \hat{\theta}_{d,k}^{\text{TE}}(i,j) \cdot \hat{\phi}_{k,w_{d,n}^{\text{TE}}}(i)$

Setup: We use three online review datasets:

- HOTEL: 240,060 reviews of hotels from TripAdvisor (Wang et al., 2010).
- RESTAURANT: 25,459 reviews of restaurants from Yelp (Jo and Oh, 2011).
- MOVIE: 5,006 reviews of movies from Rotten Tomatoes (Pang and Lee, 2005)

For all datasets, we preprocess by tokenizing, removing stopwords, stemming, adding bigrams to the vocabulary, and we filter using tf-idf to obtain a vocabulary of 10,000 words.[4]

We use perplexity, a widely-used evaluation metric in the topic modeling community (Wallach et al., 2009). The perplexity of test documents $\boldsymbol{w}^{\text{TE}}$ is $\exp(-(\sum_d \sum_n \log(p(w_{d,n}^{\text{TE}} \mid \hat{\theta}_d^{\text{TE}}, \hat{\phi})))/N)$ where N is the total number of tokens in $\boldsymbol{w}^{\text{TE}}$. We perform 5-fold cross validation and report the averaged performance over 5 folds, and use $K = 50$ topics for all datasets.[5]

Results: Figure 2 shows the perplexity of the four averaging methods, computed at each training iteration t. SA outperforms SF, showing the benefits of averaging over multiple test samples from a single test chain. However, both multiple chain methods (MF and MA) significantly outperform these two methods.

This result is consistent with Asuncion et al. (2009), who run multiple training chains but a single test chain for each training chain and average over them. This is more costly since training chains are usually significantly longer than test chains. In addition, multiple training chains are sensitive to their initialization.

4 Supervised Topic Models

We evaluate the performance of different prediction methods using supervised latent Dirichlet allocation (SLDA) (Blei and McAuliffe, 2007) for sentiment analysis: predicting review ratings given review text. Each review text is the document $\boldsymbol{w}_d$ and the metadata $\boldsymbol{y}_d$ is the associated rating.

SLDA: Going beyond LDA, SLDA captures the relationship between latent topics and metadata by modeling each document's continuous response variable using a normal linear model, whose covariates are the document's empirical distribution of topics: $y_d \sim \mathcal{N}(\boldsymbol{\eta}^T \bar{\boldsymbol{z}}_d, \rho)$ where $\boldsymbol{\eta}$ is the regression parameter vector and $\bar{\boldsymbol{z}}_d$ is the empirical distribution over topics of document d.

For posterior inference during training, following (Boyd-Graber and Resnik, 2010), we use stochastic EM, which alternates between (1) a Gibbs sampling step to assign a topic to each token, and (2) optimizing the regression parameters using L-BFGS (Liu and Nocedal, 1989). During test time, we sample the topic assignments for all tokens in the test documents and use these assignments to estimate the response variable: $S(i,j) = \hat{y}_d^{\text{TE}}(i,j) = \hat{\boldsymbol{\eta}}(i)^T \bar{z}_d^{\text{TE}}(i,j)$.[6]

Experimental setup: We use the same data as in Section 3. For all datasets, the metadata are the 1-to-5 start review rating, which is standardized using z-normalization. We use two evaluation metrics: mean squared error (MSE) and predictive R-squared (Blei and McAuliffe, 2007).

For performance comparison, we consider two baselines: (1) multiple linear regression (MLR), which models the metada as a linear function of the features, and (2) support vector regression (SVR) (Joachims, 1999). Both baselines use the frequencies of unigrams and bigrams as features.

[4] To find bigrams, we begin with bigram candidates that occur at least 10 times in the corpus and use a χ^2 test to filter out those having a χ^2 value less than 5. We then treat selected bigrams as single word types and add them to the vocabulary.

[5] MCMC setup: $T_{\text{TR}} = 1,000$, $B_{\text{TR}} = 500$, $L_{\text{TR}} = 50$, $T_{\text{TE}} = 100$, $B_{\text{TE}} = 50$ and $L_{\text{TE}} = 5$.

[6] More details can be found in the supplementary material.

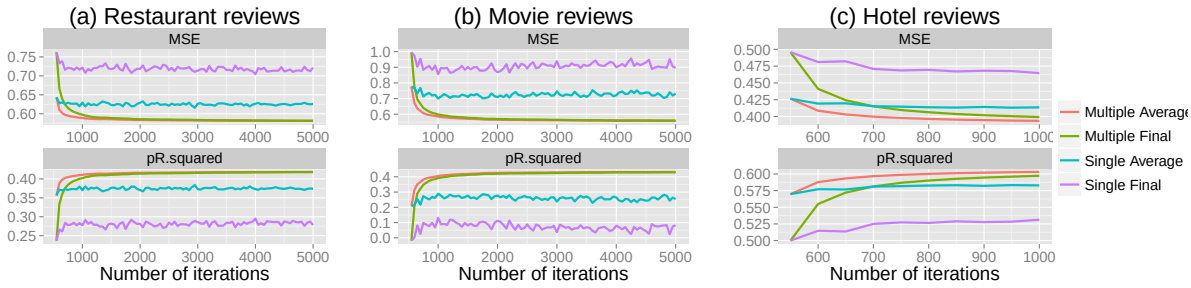

Figure 3: Performances of SLDA using different averaging strategies computed at each training iteration.

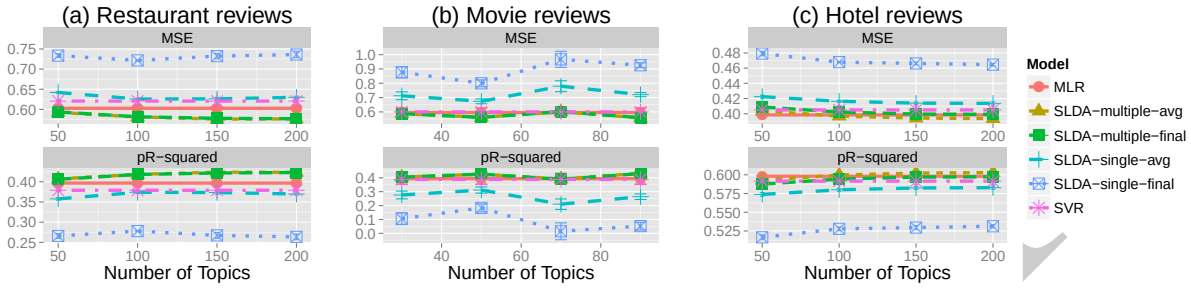

Figure 4: Performances of SLDA using different averaging strategies computed at the final training iteration T_{TR}, compared with two baselines MLR and SVR.

We perform 5-fold cross validation and report the averaged performance over 5 folds. For all models, we use a development set to tune their parameter(s) and use the set of parameters that gives best results on the development data for test.[7]

Results: Figure 3 shows the results of SLDA predicting the review rating with different averaging strategies, computed at different training iterations.[8] Consistent with the results using unsupervised method in Section 3, SA outperforms SF, but both are outperformed significantly by the two methods using multiple test chains (MF and MA).

We also compare the performance of the four prediction methods obtained at the final iteration T_{TR} of the training chain with the two baselines. The results in Figure 4 show that the two baselines (MLR and SVR) outperform significantly the SLDA using only a single test chains (SF and SA). Methods using multiple test chains (MF and MA), on the other hands, perform as well as (for HOTEL)[9] or better (for RESTAURANT and MOVIE) than the two baselines.

5 Discussions and Conclusions

MCMC relies on averaging multiple samples to approximate target densities. When used for prediction, MCMC needs to generate and average over both training samples to learn from training data and test samples to make prediction. We have shown that actually doing that averaging—not more aggressive, *ad hoc* approximations like taking the final sample (either training or test)—is not just a question of theoretical aesthetics, but an important factor in obtaining good prediction performance.

Compared with SVR and MLR baselines, SLDA using multiple test chains (MF and MA) performs as well as or better, while SLDA using a single test chain (SF and SA) falters. Thus, this simple experimental setup choice can determine whether a model improves over reasonable baselines. In addition, better prediction with shorter training is possible with multiple test chains. We conclude that averaging using multiple chains is the above-average choice.

[7] For MLR we use a Gaussian prior $\mathcal{N}(0, 1/\lambda)$ with $\lambda = a \cdot 10^b$ where $a \in [1, 9]$ and $b \in [1, 4]$; for SVR, we use SVM^{light} (Joachims, 1999) and vary $C \in [1, 50]$, which trades off between training error and margin; for SLDA, we fix $\sigma = 10$ and vary $\rho \in \{0.1, 0.5, 1.0, 1.5, 2.0\}$, which trades off between the likelihood of words and response variable.

[8] MCMC setup: $T_{\text{TR}} = 5,000$ for RESTAURANT and MOVIE and $1,000$ for HOTEL; for all datasets $B_{\text{TR}} = 500$, $L_{\text{TR}} = 50$, $T_{\text{TE}} = 100$, $B_{\text{TE}} = 20$ and $L_{\text{TE}} = 5$.

[9] One main reason for this is that SLDA has not converged after 1,000 training iterations (Figure 3).

References

Christophe Andrieu, Nando de Freitas, Arnaud Doucet, and Michael I. Jordan. 2003. An introduction to MCMC for machine learning. *Machine Learning*, 50(1-2):5–43.

Arthur Asuncion, Max Welling, Padhraic Smyth, and Yee Whye Teh. 2009. On smoothing and inference for topic models. In *UAI*.

David M. Blei and Jon D. McAuliffe. 2007. Supervised topic models. In *NIPS*.

David M. Blei, Andrew Ng, and Michael Jordan. 2003. Latent Dirichlet allocation. *JMLR*, 3.

David M. Blei. 2012. Probabilistic topic models. *Commun. ACM*, 55(4):77–84, April.

David M. Blei. 2014. Build, compute, critique, repeat: Data analysis with latent variable models. *Annual Review of Statistics and Its Application*, 1(1):203–232.

Jordan Boyd-Graber and Philip Resnik. 2010. Holistic sentiment analysis across languages: Multilingual supervised latent Dirichlet allocation. In *EMNLP*.

Jonathan Chang. 2012. lda: Collapsed Gibbs sampling methods for topic models. `http://cran.r-project.org/web/packages/lda/index.html`. [Online; accessed 02-June-2014].

Qixia Jiang, Jun Zhu, Maosong Sun, and Eric P. Xing. 2012. Monte Carlo methods for maximum margin supervised topic models. In *NIPS*.

Yohan Jo and Alice H. Oh. 2011. Aspect and sentiment unification model for online review analysis. In *WSDM*.

Thorsten Joachims. 1999. Making large-scale SVM learning practical. In *Advances in Kernel Methods - Support Vector Learning*, chapter 11. Cambridge, MA.

Simon Lacoste-Julien, Fei Sha, and Michael I. Jordan. 2008. DiscLDA: Discriminative learning for dimensionality reduction and classification. In *NIPS*.

D. Liu and J. Nocedal. 1989. On the limited memory BFGS method for large scale optimization. *Math. Prog.*

Radford M. Neal. 1993. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, University of Toronto.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *ACL*.

Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher Manning. 2009. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *EMNLP*.

Daniel Ramage, Christopher D. Manning, and Susan Dumais. 2011. Partially labeled topic models for interpretable text mining. In *KDD*, pages 457–465.

Philip Resnik and Eric Hardisty. 2010. Gibbs sampling for the uninitiated. Technical Report UMIACS-TR-2010-04, University of Maryland. http://drum.lib.umd.edu//handle/1903/10058.

Michal Rosen-Zvi, Thomas L. Griffiths, Mark Steyvers, and Padhraic Smyth. 2004. The author-topic model for authors and documents. In *UAI*.

Mark Steyvers and Tom Griffiths. 2006. Probabilistic topic models. In T. Landauer, D. Mcnamara, S. Dennis, and W. Kintsch, editors, *Latent Semantic Analysis: A Road to Meaning*. Laurence Erlbaum.

Hanna M. Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. 2009. Evaluation methods for topic models. In Leon Bottou and Michael Littman, editors, *ICML*.

Chong Wang, David Blei, and Li Fei-Fei. 2009. Simultaneous image classification and annotation. In *CVPR*.

Hongning Wang, Yue Lu, and Chengxiang Zhai. 2010. Latent aspect rating analysis on review text data: A rating regression approach. In *SIGKDD*, pages 783–792.

Jun Zhu, Amr Ahmed, and Eric P. Xing. 2009. MedLDA: maximum margin supervised topic models for regression and classification. In *ICML*.

Jun Zhu, Ning Chen, Hugh Perkins, and Bo Zhang. 2014. Gibbs max-margin topic models with data augmentation. *Journal of Machine Learning Research*, 15:1073–1110.