# Structured Black Box Variational Inference for Latent Time Series Models

**Robert Bamler** [1]   **Stephan Mandt** [1]

## Abstract

Continuous latent time series models are prevalent in Bayesian modeling; examples include the Kalman filter, dynamic collaborative filtering, or dynamic topic models. These models often benefit from structured, non mean field variational approximations that capture correlations between time steps. Black box variational inference with reparameterization gradients (BBVI) allows us to explore a rich new class of Bayesian non-conjugate latent time series models; however, a naive application of BBVI to such structured variational models would scale quadratically in the number of time steps. We describe a BBVI algorithm analogous to the forward-backward algorithm which instead scales linearly in time. It allows us to efficiently sample from the variational distribution and estimate the gradients of the ELBO. Finally, we show results on the recently proposed dynamic word embedding model, which was trained using our method.

## 1. Introduction

Continuous latent time series models are popular tools in Bayesian machine learning. One thereby combines multiple copies of a likelihood model through a time series prior, thus enabling the latent variables to drift over time while still sharing statistical strength accross all times (Blei & Lafferty, 2006; Wang et al., 2008; Sahoo et al., 2012; Gultekin & Paisley, 2014; Charlin et al., 2015; Ranganath et al., 2015; Jerfel et al., 2017; Bamler & Mandt, 2017).

Variational Inference (VI) enables approximate inference for complex models by solving an optimization problem (Jordan et al., 1999). One chooses a variational distribution and fits it to the posterior, where the fully-factorized mean field class is the most widely used. However, the

---

[1]Disney Research, 4720 Forbes Avenue, Pittsburgh, PA 15213, USA. Correspondence to: Robert Bamler <Robert.Bamler@disneyresearch.com>, Stephan Mandt <Stephan.Mandt@disneyresearch.com>.

standard mean field class is not a good choice when it comes to time series models. Instead, one often uses *structured* variational distributions in time (Wainwright et al., 2008; Blei & Lafferty, 2006). The perhaps simplest such choice is a multivariate Gaussian with a tridiagonal precision matrix. This can be seen as using the probabilistic model of the Kalman filter as a variational distribution (variational Kalman filter) (Blei & Lafferty, 2006), and reflects the fact that the prior is a first-order Markov process.

In this paper, we introduce an efficient VI algorithm for fitting such tridiagonal Gaussian variational models to a large class of complex latent time series models. We draw on Black Box Variational Inference (BBVI) with reparameterization gradients (Salimans & Knowles, 2013; Kingma & Welling, 2014; Rezende et al., 2014; Ruiz et al., 2016), where one forms a Monte-Carlo estimate of the variational lower bound's gradient. The problem with structured variational distributions is that computing this reparameterization gradient is expensive; a naive implementation involves a dense matrix multiplication and scales as $O(T^2)$, where $T$ is the number of time steps. In this paper, we lay out a general algorithm which gives the reparameterization gradient in $O(T)$. This algorithm can be thought of a variant of the forward-backward algorithm (Murphy, 2012) in the context of BBVI, and relies on a reparameterization procedure that never involves the inversion or multiplication of a dense matrix. We illustrate our approach on the dynamic word embedding model (Bamler & Mandt, 2017).

## 2. Background and Problem Setting

We start by specifying our generative and variational models and give an overview of the proposed algorithm.

**Generative model.** We consider models with time-dependent observations $\mathbf{x} \equiv (x_1, \ldots, x_T)$, where $T$ is the number of time steps. At each time step $t \in \{1, \ldots, T\}$, the generative process is described by an arbitrary likelihood function $p(x_t|z_t)$, and $z_t$ is a latent variable. We furthermore assume that the prior is Markovian. The joint probability thus factorizes as follows,

$$p(\mathbf{x}, \mathbf{z}) = \prod_{t=1}^{T} p(z_t|z_{t-1})\, p(x_t|z_t), \qquad (1)$$

where $p(z_1|z_0) \equiv p(z_1)$ denotes an arbitrary prior on the first latent variable. Many models fall into this class. Our goal is an efficient posterior approximation for this model class, using a structured variational approximation and black box inference techniques.

**Kalman smoothing revisited.** Before we describe our approach, we revisit the Kalman smoother (Rauch et al., 1965) as an efficient algorithm for a particularly simple realization of Eq. 1 where all conditional distributions are Gaussian. This is often called a Wiener process (more precisely, it is a Gauss-Markov process). The prior is Gaussian with tridiagonal precision matrix $\Lambda_0$, and the likelihood is a Gaussian centered around $z_t$ with precision $\tau$. Thus, the posterior is also Gaussian, $p(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \Lambda^{-1})$, and can be obtained analytically. One finds $\Lambda = \Lambda_0 + \tau I$ and $\boldsymbol{\mu} = \tau \Lambda^{-1} \mathbf{x}$. Obtaining the posterior modes $\boldsymbol{\mu}$ involves solving the linear system of equations $\Lambda \boldsymbol{\mu} = \tau \mathbf{x}$. For an arbitrary matrix $\Lambda$, this involves $O(T^2)$ operations. However, for the Wiener process, $\Lambda$ is tridiagonal and one can solve for $\boldsymbol{\mu}$ in linear time using a specialized algorithm.

The forward-backward algorithm (Murphy, 2012) implicitly decomposes $\Lambda = AB$ into a product of two bidiagonal matrices $A$ and $B$, where $A$ is lower triangular and $B$ is upper triangular. One starts with a forward pass through the data, in which one solves $A\tilde{\boldsymbol{\mu}} = \tau \mathbf{x}$ for the auxiliary vector $\tilde{\boldsymbol{\mu}}$ using forward substitution, followed by a backward pass in which one solves $B\boldsymbol{\mu} = \tilde{\boldsymbol{\mu}}$ for $\boldsymbol{\mu}$ using back substitution. As detailed in section 3.3, we use a similar philosophy.

**Variational model.** For a general likelihood model $p(x_t|z_t)$, the exact posterior of Eq. 1 is intractable. To circumvent this problem, we use BBVI to minimize the KL divergence between a variational distribution $q_{\boldsymbol{\lambda}}$ and the true posterior. Here, $\boldsymbol{\lambda}$ summarizes all variational parameters. We use a structured variational distribution as our variational model that is motivated by the exact posterior of the analytically solvable model discussed above. We thus consider a Gaussian with a tridiagonal precision matrix $\Lambda$,

$$q_{\boldsymbol{\lambda}}(\mathbf{z}) \equiv \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \Lambda^{-1}). \quad (2)$$

This can be understood as varying the parameters of a (fictitious) Wiener process so that its posterior approximates the posterior of the true model (Blei & Lafferty, 2006).

As we discuss below, the tridiagonal structure of $\Lambda$ allows us to fit the variational distribution efficiently. Note that the covariance matrix $\Lambda^{-1}$ is dense, thus encoding long-range correlations between any two time steps. Modelling correlations is important for many time series applications, in particular when the prior is strong e.g. due to little evidence per time step.

**Black box variational inference.** In BBVI, one fits $q_{\boldsymbol{\lambda}}(\mathbf{z})$ to the posterior $p(\mathbf{z}|\mathbf{x})$, maximizing the evidence lower bound (ELBO) using stochastic gradients. The reparameterization trick amounts to representing the ELBO as

$$\mathcal{L}(\boldsymbol{\lambda}) = \mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(0,I)}[\log p(\mathbf{x}, f(\boldsymbol{\lambda}; \boldsymbol{\epsilon}))] + H[q_{\boldsymbol{\lambda}}]. \quad (3)$$

The entropy $H[q_{\boldsymbol{\lambda}}] \equiv -\mathbb{E}_{q_{\boldsymbol{\lambda}}}[\log q_{\boldsymbol{\lambda}}(\mathbf{z})]$ is often an analytic function or can be estimated using other tricks. Here, $\boldsymbol{\epsilon}$ is a vector of $T$ independent standard-normal distributed random variables, and $\mathbf{f} \equiv (f_1, \ldots, f_T)$ denotes $T$ functions that are defined such that the random variable $\mathbf{f}(\boldsymbol{\lambda}; \boldsymbol{\epsilon})$ has probability density $q_{\boldsymbol{\lambda}}$ (see Section 3.2 below).

In order to implement an efficient BBVI algorithm, one needs to be able to estimate the gradient of $\mathcal{L}$ efficiently, i.e., in $O(T)$ time. This involves the following three tasks:

1. Efficiently evaluate the entropy $H[q_{\boldsymbol{\lambda}}]$ (Section 3.1);

2. Efficiently evaluate $\mathbf{f}(\boldsymbol{\lambda}; \boldsymbol{\epsilon})$ (Section 3.2);

3. Efficiently estimate the reparameterization gradient (Section 3.3).

All three of the above tasks can easily be solved efficiently if one chooses a mean field variational distribution, i.e., if $q_{\boldsymbol{\lambda}}(\mathbf{z})$ factorizes over all time steps. However, the mean field approximation ignores correlations between time steps, which are important in many time series models as discussed above. In Section 3 we address each one of the above tasks individually.

## 3. Inference

In this section, we give the details of our new black box variational inference algorithm.

### 3.1. Evaluating the Entropy

The entropy of a multivariate Gaussian with precision matrix $\Lambda$ is $H[q_{\boldsymbol{\lambda}}] = -\frac{1}{2} \log(\det \Lambda)$, up to an additive constant. Evaluating the determinant of a general $T \times T$ matrix takes $O(T^3)$ operations in practice. To avoid this expensive operation, we parameterize the precision matrix via its Cholesky decomposition,

$$\Lambda = B^\top B. \quad (4)$$

Here, $B$ is an upper triangular $T \times T$ matrix. Since we restrict $\Lambda$ to have a tridiagonal structure, $B$ is bidiagonal (Kılıç & Stanica, 2013), i.e., it has the structure

$$B(\boldsymbol{\nu}, \boldsymbol{\omega}) = \begin{pmatrix} \nu_1 & \omega_1 & & & \\ & \nu_2 & \omega_2 & & \\ & & \ddots & \ddots & \\ & & & \nu_{T-1} & \omega_{T-1} \\ & & & & \nu_T \end{pmatrix}, \quad (5)$$

with $\nu_t > 0 \ \forall t \in \{1, \ldots, T\}$. As the mapping from $B$ to $\Lambda$ is unique, it suffices to optimize the $(2T - 1)$ non-zero components of $B$ in order to find the optimal entries of $\Lambda$. It turns out that we never have to construct the matrix $\Lambda = B^\top B$ explicitly. The variational parameters $\boldsymbol{\lambda} \equiv (\boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{\omega})$ are thus the marginal means $\boldsymbol{\mu}$ (see Eq. 2) and the non-zero components of $B$. Using the relation $\det \Lambda = (\det B)^2$, we can evaluate the entropy in linear time,

$$H[q_{\boldsymbol{\lambda}}] = -\sum_{t=1}^{T} \log \nu_t + const. \tag{6}$$

### 3.2. Evaluating the reparameterization functions

In contrast to the entropy, the expected log-joint in Eq. 3 cannot be evaluated analytically for a general model. We obtain an unbiased gradient estimator $\hat{\mathbf{g}}$ of the expected log-joint by drawing $S$ independent samples $\boldsymbol{\epsilon}^{[s]} \sim \mathcal{N}(0, I)$ for $s \in \{1, \ldots, S\}$.

For what follows, let $\lambda_i$ denote any of the $(3T - 1)$ variational parameters. Using the chain rule, the estimate of the gradient of the expected log-joint with respect to $\lambda_i$ is

$$\hat{g}_i \equiv \frac{\partial}{\partial \lambda_i} \left[ \frac{1}{S} \sum_{s=1}^{S} \log p(\mathbf{x}, \mathbf{f}(\boldsymbol{\lambda}, \boldsymbol{\epsilon}^{[s]})) \right]$$
$$= \frac{1}{S} \sum_{s=1}^{S} \sum_{t=1}^{T} \gamma_t^{[s]} \frac{\partial f_t(\boldsymbol{\lambda}; \boldsymbol{\epsilon}^{[s]})}{\partial \lambda_i} \tag{7}$$

where we defined

$$\gamma_t^{[s]} \equiv \left. \frac{\partial \log p(\mathbf{x}, \mathbf{z}^{[s]})}{\partial z_t^{[s]}} \right|_{\mathbf{z}^{[s]} = \mathbf{f}(\boldsymbol{\lambda}, \boldsymbol{\epsilon}^{[s]})} \tag{8}$$
$$= \frac{\partial [\log p(z_t^{[s]}|z_{t-1}^{[s]}) + \log p(z_{t+1}^{[s]}|z_t^{[s]}) + \log p(x_t|z_t^{[s]})]}{\partial z_t^{[s]}}.$$

To further simplify Eq. 7 we specialize the reparameterization function to our variational model. Using Eqs. 2 and 4, we find

$$\mathbf{f}(\boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{\omega}; \boldsymbol{\epsilon}^{[s]}) = \boldsymbol{\mu} + B(\boldsymbol{\nu}, \boldsymbol{\omega})^{-1} \boldsymbol{\epsilon}^{[s]}. \tag{9}$$

Here, $B^{-1}$ is a dense (upper triangular) $T \times T$ matrix. Instead of computing the inverse, we evaluate the term $\mathbf{y}^{[s]} \equiv B^{-1} \boldsymbol{\epsilon}^{[s]}$ on the right-hand side of Eq. 9 by solving the linear system $B\mathbf{y}^{[s]} = \boldsymbol{\epsilon}^{[s]}$ for $\mathbf{y}^{[s]}$ using back substitution. This takes only $O(T)$ operations due to the bidiagonal structure of $B$, see Eq. 5. We can therefore evaluate Eq. 8 in $O(T)$ time for each sample $s \in \{1, \cdots, S\}$.

### 3.3. Estimating the Reparameterization Gradient

Last, we show how to efficiently estimate the reparameterization gradient in $O(T)$ time. In this subsection we describe an efficient method to evaluate the Jacobian $\partial \mathbf{f}/\partial \boldsymbol{\lambda}$ that appears on the right-hand side of Eq. 7.

A naive evaluation of all gradient estimates $\hat{g}_i$ would require $O(S \times T^2)$ operations: the sums on the right-hand side of Eq. 7 run over $S \times T$ terms, and the number of variational parameters $\lambda_i$ for which the gradient estimate has to be evaluated grows at least linearly in $T$ for a reasonably flexible variational distribution.

We use the following trick to evaluate all gradient estimates in linear time in $T$. For any invertible matrix $B$ that depends on some parameter $\lambda_i$, we have

$$0 = \frac{\partial I}{\partial \lambda_i} = \frac{\partial (BB^{-1})}{\partial \lambda_i} = \frac{\partial B}{\partial \lambda_i} B^{-1} + B \frac{\partial B^{-1}}{\partial \lambda_i}. \tag{10}$$

Solving for $\partial B^{-1}/\partial \lambda_i$, we obtain

$$\frac{\partial B^{-1}}{\partial \lambda_i} = -B^{-1} \frac{\partial B}{\partial \lambda_i} B^{-1}. \tag{11}$$

Eq. 11 expresses the derivative of the dense (upper triangular) matrix $B^{-1}$ in terms of the derivative of the bidiagonal matrix $B$. In fact, both $\partial B/\partial \nu_t$ and $\partial B/\partial \omega_t$ are sparse matrices with only a single non-zero entry, see Eq. 5. The dense matrix $B^{-1}$ still appears on the right-hand side of Eq. 11 but we avoid evaluating it explicitly. Instead, we again solve a bidiagonal linear system of equations. Combining Eqs. 7, 9 and 11, we obtain the formal expressions

$$\hat{g}_{\mu_t} = \frac{1}{S} \sum_{s=1}^{S} \gamma_t^{[s]}; \tag{12}$$

$$\hat{g}_{\nu_t} = -\frac{1}{S} \sum_{s=1}^{S} \underbrace{((\boldsymbol{\gamma}^{[s]})^\top B^{-1})_t}_{y_t'^{[s]}} \underbrace{(B^{-1} \boldsymbol{\epsilon}^{[s]})_t}_{y_t^{[s]}}; \tag{13}$$

$$\hat{g}_{\omega_t} = -\frac{1}{S} \sum_{s=1}^{S} \underbrace{((\boldsymbol{\gamma}^{[s]})^\top B^{-1})_t}_{y_t'^{[s]}} \underbrace{(B^{-1} \boldsymbol{\epsilon}^{[s]})_{t+1}}_{y_{t+1}^{[s]}} \tag{14}$$

where bold face $\boldsymbol{\gamma}^{[s]} \equiv (\gamma_1^{[s]}, \ldots, \gamma_T^{[s]})^\top$ denotes a column vector of $T$ derivatives of the log-joint as defined in Eq. 8. Instead of computing the inverse $B^{-1}$ in Eqs. 13–14, we obtain the vectors $(\mathbf{y}'^{[s]})^\top \equiv (\boldsymbol{\gamma}^{[s]})^\top B^{-1}$ and $\mathbf{y}^{[s]} \equiv B^{-1} \boldsymbol{\epsilon}^{[s]}$ by solving the linear systems $B^\top \mathbf{y}'^{[s]} = \boldsymbol{\gamma}^{[s]}$ and $B\mathbf{y}^{[s]} = \boldsymbol{\epsilon}^{[s]}$, respectively, in $O(T)$ time using the bidiagonal structure of $B$. Since $B^\top$ is lower-diagonal and $B$ is upper-diagonal these operations carry-out a forward and a backward pass through the time steps, respectively. This is similar to the forward-backward algorithm for the Wiener process with Gaussian likelihood discussed in Section 2.

Eqs. 12–14 conclude the derivation of the gradient estimator for the expected log-joint. The gradient of the ELBO with respect to $\nu_t$ contains an additional term $-1/\nu_t$ due to the entropy, see Eq. 6.
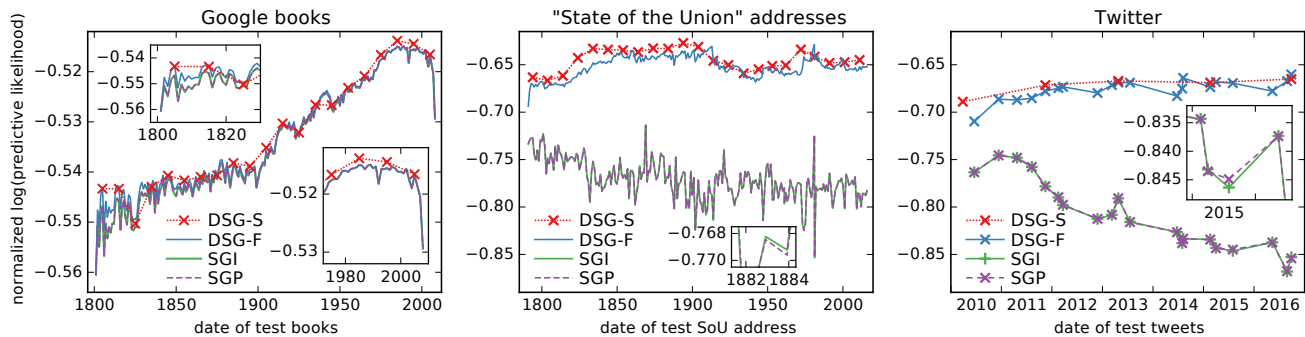
Figure 1. Predictive log-likelihoods for two proposed versions of the dynamic skip-gram model (DSG-F & DSG-S) and two competing methods SGI (Hamilton et al., 2016) and SGP (Kim et al., 2014) on three different corpora (high values are better). The proposed structured BBVI algorithm (red crosses) outperforms all baselines.

## 4. Experiments

To demonstrate the benefits of our approach, we summarize selective experimental results from work on dynamic word embeddings, which used our proposed BBVI algorithm.

**Dynamic Skip-Gram Model.** Bamler & Mandt (2017) proposed the dynamic skip-gram model as a generalization of the skip-gram model to time-stamped text corpora (Mikolov et al., 2013). The model, thus, represents words as latent trajectories in an embedding space, and is able to track how words change their meaning over time.

To train the model, one first summarizes the text corpus in terms of certain sufficient statistics, using a finite vocabulary. For each time stamp, one builds co-occurrence matrices of words and their surrounding words. In addition, the model requires so-called *nagative* examples, expressing the likelihood of co-occurrence by mere chance. The model then models these statistics in terms of the geometrical arrangement of certain latent word and context embedding vectors. The prior over these embedding vectors is a latent Ornstein-Uhlenbeck process (a Wiener process in the presence of a restoring force), enforcing a continuous drift over time. For more details, we refer to (Bamler & Mandt, 2017).

**Algorithms and Baselines.** SGI denotes the non-Bayesian skip-gram model with independent random initializations of word vectors (Mikolov et al., 2013; Hamilton et al., 2016). SGP denotes the same approach as above, but with word and context vectors being pre-initialized with the values from the previous year, as in (Kim et al., 2014). Both approaches have no dynamical priors and hence no overhead and just scale linearly with $T$. DSG-F denotes the dynamic skip-gram filtering algorithm, proposed in (Bamler & Mandt, 2017), which also runs in $O(T)$, but uses the mean field approximation and only uses information form the past. Finally, DSG-S denotes the dynamic skip-gram

smoothing algorithm. This is the $O(T)$ algorithm proposed in this paper, applied to the dynamic skip-gram model.

**Data.** We used data from the Google books corpus (Michel et al., 2011) from the last two centuries ($T = 209$). We also used the "State of the Union" (SoU) addresses of U.S. presidents, which spans more than two centuries. Finally, we used a Twitter corpus of news tweets for 21 randomly drawn dates from 2010 to 2016. Details on hyperparameters and preprocessing are given in (Bamler & Mandt, 2017).

**Results.** We focus on the quantitative results of (Bamler & Mandt, 2017), showing show that the dynamic skip-gram smoothing algorithm (described and generalized in this paper) generalizes better to unseen data than all baselines at similar run time. We thereby analyze held-out predictive likelihoods on word-context pairs at a given time $t$, where $t$ is excluded from the training set.

The predictive likelihoods as a function of time are shown in Figure 1. On all three corpora, differences between the two implementations of the static model (SGI and SGP) are small, which suggests that pre-initializing the embeddings with the previous result seems to have little impact on the predictive power. Log-likelihoods for the skip-gram filter (DSG-F) grow over the first few time steps as the filter sees more data, and then saturate. Skip-gram smoothing (this paper) further outperforms skip-gram filtering.

To conclude, we stress that a structured BBVI algorithm with quadratic instead of linear run time in $T$ would be impractical. We therefore hope that our structured reparameterization trick will spur further research on complex latent time series models.

# References

Bamler, Robert and Mandt, Stephan. Dynamic word embeddings. In *International Conference on Machine Learning*, 2017.

Blei, David M and Lafferty, John D. Dynamic Topic Models. In *Proceedings of the 23rd International Conference on Machine Learning*, pp. 113–120. ACM, 2006.

Charlin, Laurent, Ranganath, Rajesh, McInerney, James, and Blei, David M. Dynamic Poisson Factorization. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pp. 155–162, 2015.

Gultekin, San and Paisley, John. A Collaborative Kalman Filter for Time-Evolving Dyadic Processes. In *Proceedings of the 2nd International Conference on Data Mining*, pp. 140–149, 2014.

Hamilton, William L, Leskovec, Jure, and Jurafsky, Dan. Diachronic word embeddings reveal statistical laws of semantic change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 1489–1501, 2016.

Jerfel, Ghassen, Basbug, Mehmet E, and Engelhardt, Barbara E. Dynamic Compound Poisson Factorization. In *Artificial Intelligence and Statistics*, 2017.

Jordan, Michael I, Ghahramani, Zoubin, Jaakkola, Tommi S, and Saul, Lawrence K. An Introduction to Variational Methods for Graphical Models. *Machine learning*, 37(2):183–233, 1999.

Kılıç, Emrah and Stanica, Pantelimon. The Inverse of Banded Matrices. *Journal of Computational and Applied Mathematics*, 237(1):126–135, 2013.

Kim, Yoon, Chiu, Yi-I, Hanaki, Kentaro, Hegde, Darshan, and Petrov, Slav. Temporal Analysis of Language Through Neural Language Models. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, pp. 61–65, 2014.

Kingma, Diederik P and Welling, Max. Auto-encoding variational Bayes. In *ICLR*, 2014.

Michel, Jean-Baptiste, Shen, Yuan Kui, Aiden, Aviva Presser, Veres, Adrian, Gray, Matthew K, Pickett, Joseph P, Hoiberg, Dale, Clancy, Dan, Norvig, Peter, Orwant, Jon, et al. Quantitative Analysis of Culture Using Millions of Digitized Books. *Science*, 331(6014):176–182, 2011.

Mikolov, Tomas, Sutskever, Ilya, Chen, Kai, Corrado, Greg S, and Dean, Jeff. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26*, pp. 3111–3119. 2013.

Murphy, Kevin P. *Machine learning: a probabilistic perspective*. MIT press, 2012.

Ranganath, Rajesh, Perotte, Adler J, Elhadad, Noémie, and Blei, David M. The Survival Filter: Joint Survival Analysis with a Latent Time Series. In *UAI*, pp. 742–751, 2015.

Rauch, Herbert E, Striebel, CT, and Tung, F. Maximum likelihood estimates of linear dynamic systems. *AIAA journal*, 3(8):1445–1450, 1965.

Rezende, Danilo Jimenez, Mohamed, Shakir, and Wierstra, Daan. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *The 31st International Conference on Machine Learning (ICML)*, 2014.

Ruiz, Francisco, Titsias, Michaelis, and Blei, David. The generalized reparameterization gradient. In *NIPS*, 2016.

Sahoo, Nachiketa, Singh, Param Vir, and Mukhopadhyay, Tridas. A Hidden Markov Model for Collaborative Filtering. *MIS Quarterly*, 36(4):1329–1356, 2012.

Salimans, Tim and Knowles, David A. Fixed-form variational posterior approximation through stochastic line ar regression. *Bayesian Analysis*, 8(4), 2013.

Wainwright, Martin J, Jordan, Michael I, et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2): 1–305, 2008.

Wang, Chong, Blei, David, and Heckerman, David. Continuous time dynamic topic models. In *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence*, pp. 579–586, 2008.