Supplementary appendix to "Learning to Extract International Relations from Political Context" (ACL 2013)

Brendan O'Connor, Brandon Stewart, Noah A. Smith
Contact: brenocon@cs.cmu.edu
Web: http://brenocon.com/irevents

May 16, 2013

1 Lexical scale impurity at the type-level

As noted in the paper the measure we want is a posterior expectation defined for instance pairs, which we can reformulate at the type level as follows. Let i and j index over instances, and w and v index over types. Consider an expectation using a single sample to represent the posterior,

$$E\left[\, |g(w_i) - g(w_j)| \mid z_i = z_j \;\&\; w_i \neq w_j \;\&\; w_i, w_j \in M\right] = \frac{Q}{N} \tag{1}$$

where N is the number of instance pair comparisons satisfying the conditional, and Q is,

$$\begin{aligned}
Q &= \sum_{ij} 1\{z_i = z_j\}\, 1\{w_i \in M\}\, 1\{w_j \in M\}\, 1\{w_i \neq w_j\}\, d_{ij} && (2)\\
&= \sum_k \sum_{ij} 1\{z_i = k\}\, 1\{z_j = k\}\, 1\{w_i \in M\}\, 1\{w_j \in M\}\, 1\{w_i \neq w_j\}\, d_{ij} && (3)\\
&= \sum_k \sum_i 1\{z_i = k\}\, 1\{w_i \in M\} \sum_j 1\{z_j = k\}\, 1\{w_j \in M\}\;\; 1\{w_i \neq w_j\}\, d_{ij} && (4)\\
&= \sum_k \sum_{w \in M, v \in M, w \neq v} n_{wk}\, n_{vk}\, d_{wv} && (5)
\end{aligned}$$

where $d_{ij} = |g(w_i) - g(w_j)|$, $d_{wv} = |g(w) - g(v)|$, and n_{wk} and n_{vk} are from the collapsed Gibbs sampling count tables, i.e. $n_{wk} = \sum_i 1\{w_i = w\}\, 1\{z_i = k\}$.

The denominator is

$$N = \sum_k \sum_{w \in M, v \in M, w \neq v} n_{w,k} n_{v,k}$$

To properly compute a posterior expectation using multiple samples, Q/N should be re-evaluated on several complete samples and then averaged. However, we found little variation between samples so used only one. We also tried evaluating a single Q/N where n_{wk} and n_{vk} are *averaged* counts from multiple samples—using this corresponds to a factored, mean-field-like approximation to the posterior—but it also was very similar to using a single sample.

The implementation is in *verbdict/score.py*.

2 TABARI lexicon matching

Two additional notes.

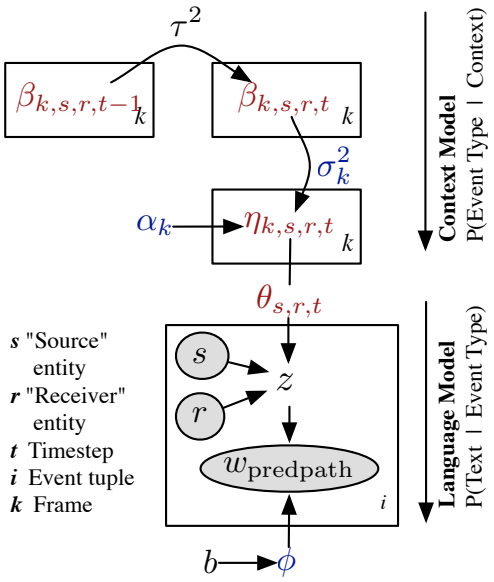
(1) There were a number of patterns in the TABARI lexicon that had multiple conflicting codes. See *verbdict/conflicting_codes.txt*.

(2) As described in the paper, the dependency paths are traversed from source to receiver, creating the corresponding word sequence. Prepositions are un-collapsed and put into the sequence. There is special handling of *xcomp*'s, which sometimes represent an infinitival 'to' and sometimes do not; we generate two versions, with and without 'to'; if either one matches to a TABARI pattern then that counts as a match.

The implementation is in *verbdict/match.py*

3 Inference

The full smoothed model is:



Context model (smoothed frames):

$$\begin{aligned}
\tau^2 &\sim \text{InvGamma} \\
\sigma_k^2 &\sim \text{InvGamma} \\
\alpha_k &\sim \text{Normal} \\
\beta_{s,r,1,k} &\sim N(0, 100) \\
\beta_{s,r,t>1,k} &\sim N(\beta_{k,s,r,t-1}, \tau^2) \\
\eta_{s,r,t,k} &\sim N(\alpha_k + \beta_{k,s,r,t}, \sigma_k^2) \\
\theta_{s,r,t,*} &= \text{Softmax}(\eta_{s,r,t,*})
\end{aligned}$$

Language model:

$$\begin{aligned}
b &\sim \text{ImproperUniform} \\
\phi_k &\sim \text{Dir}(b/V) \\
z &\sim \theta_{s,r,t} \\
w &\sim \phi_z
\end{aligned}$$

The blocked Gibbs sampler proceeds on the following groups of variables. These conditionals implicitly also condition on w, s, r, t.

- Context (Politics) submodel
  - $[\alpha \mid \eta, \beta, \sigma^2]$: Exact
  - $[\beta \mid \eta, \alpha, \sigma^2]$: Exact, FFBS algorithm
- Context/Language bridge
  - $[\eta \mid \beta, \alpha, z]$: Laplace approximation Metropolis-within-Gibbs step
- Language submodel
  - $[z|\eta]$: Exact, collapsing out ϕ

- Dispersions (variances and concentrations)
  - $[\tau^2|\beta]$, $[\sigma^2|\eta, \alpha]$, $[b|z, w]$

The key step is sampling instantiations of η, which is the bottleneck between the politics and language models; given that, inference proceeds on either side of the model via well-known conjugate posterior resampling updates, each described as follows.

3.1 Language Submodel $[z|\eta]$

This is the most straightforward step in light of previous work in Bayesian language modeling. Dirichlet-Multinomial conjugacy allows Gibbs sampling to proceed on individual z's for individual tuples, collapsing out ϕ (as in Griffiths and Steyvers (2004)), though unlike that work we condition on θ):

$$p(z_i = k \mid s, r, t, w, z_{-i}, \theta, b) \;\propto\; \theta_{s,r,t} \, \frac{\#\{z = k, w\} + b/V}{\#\{z = k\} + b} \tag{6}$$

where the counts are taken from the current z setting in all corpus tuples, except tuple i. b is the Dirichlet concentration parameter, and V is the number of verb-path types.

3.1.1 Context Submodel $[\alpha, \beta \mid \eta]$

The α update is just a conjugate normal sample; see any standard Bayesian reference, e.g. §4.4.2.1 of Murphy (2012), or Gelman *et al.* (2003). Let the all-but-α residual be $r_{s,r,t,k} = \eta_{s,r,t,k} - \beta_{s,r,t,k}$, so $r \sim N(\alpha, \sigma_k^2)$. With prior $p(\alpha) \sim N(0, 100)$, then

$$p(\alpha_k \mid \eta, \beta, \sigma_k^2) = N\left(\frac{n/\sigma_k^2}{n/\sigma_k^2 + 1/100}\bar{r}_k, \;\; [1/100 + n/\sigma_k^2]^{-1}\right)$$

where $\bar{r}_k$ is the current residual empirical mean: $\bar{r}_k = \sum_{s,r,t}(\eta_{s,r,t,k} - \beta_{s,r,t,k})$, and n is the number of η emissions for this frame (i.e. the number contexts). η only exists for contexts with at least one event tuple (otherwise it is vacuous variable), the sums over (s, r, t) are only over those contexts. Still, n is very large (hundreds of thousands) so the posterior is very peaked; updating α is basically the same as an ML estimate and the prior is irrelvant.

The β update is dynamic linear model inference. Because of the emissions' diagonal covariance $diag(\sigma_1^2 \ldots \sigma_K^2)$, it decomposes into conditional independence for each frame's time series for each dyad. A single joint sample of one of these time series,

$$(\hat{\beta}_{s,r,1,k} \ldots \hat{\beta}_{s,r,T,k}) \sim p(\beta_{s,r,1,k} \ldots \beta_{s,r,T,k} \mid \alpha, \eta, \sigma_k^2, \tau^2)$$

can be drawn exactly with dynamic programming, via the forward filter, backward sampling algorithm (FFBS; Harrison and West, 1997; Carter and Kohn, 1994). We leave out $\alpha, \sigma_k^2, \tau^2$ in the following equations for clarity. Here, FFBS proceeds in two steps: (1) run a Kalman filter, successively computing each $p(\beta_t \mid \eta_1 \ldots \eta_t)$ (each of which is normal), and (2) run a sampling variant of the RTS smoother, to sample successively each $\hat{\beta}_t \sim p(\beta_t \mid \hat{\beta}_{t+1}, \eta_1 \ldots \eta_t)$ (each of which is also normal). The final sequence of sampled $\hat{\beta}_t$ values is a sample from the joint sequence posterior, since $p(\beta_1 \ldots \beta_T|\eta_{1:T}) = p(\beta_T|\eta_{1:T})\; p(\beta_{T-1}|\beta_T, \eta_{1:T-1}) \;\ldots\; p(\beta_1|\beta_2, \eta_1)$.

We use μ and Σ to denote posterior beliefs about β. Let $N(\mu_{t|t-1}, \Sigma_{t|t-1})$ denote $p(\beta_t \mid \eta_1 \dots \eta_{t-1})$, and $N(\mu_t, \Sigma_t)$ denote $p(\beta_t \mid \eta_1 \dots \eta_t)$ (where Σ is just a scalar variance). The full Kalman filter is defined for much more general Gaussian state-space models: (Murphy, 2012 §18.3 notation)

$$\begin{array}{lllll} z_t & = Az_{t-1} & + & Bu_t & + & N(0, Q) \\ y_t & = Cz_t & + & Du_t & + & N(0, R) \end{array}$$

Which for us is just

$$\begin{array}{lllll} \beta_t & = \beta_{t-1} & & & + & N(0, \tau^2) \\ \eta_t & = \beta_t & + & \alpha & + & N(0, \sigma^2) \end{array}$$

The algorithm is[1]

- Filter, which takes $\eta_{1:T}$ as input.

  Initialize $\mu_{1|0} := 0,\ \Sigma_{1|0} := 100$ (and skip the prediction step on the first iteration).

  For $t = 1..T$,

  - Prediction step (infer $p(\beta_t \mid \eta_1 \dots \eta_{t-1})$):

    $\mu_{t|t-1} := \mu_{t-1}$

    $\Sigma_{t|t-1} := \Sigma_{t-1} + \tau^2$
  - Measurement step (infer $p(\beta_t \mid \eta_1 \dots \eta_t)$):

    $r := \eta_t - (\mu_{t|t-1} + \alpha)$ (residual)

    $K := \Sigma_{t|t-1}(\Sigma_{t|t-1} + \sigma^2)^{-1}$ (Kalman gain)

    $\mu_t := \mu_{t|t-1} + Kr$

    $\Sigma_t := \Sigma_{t|t-1}(1 - K)$

- Backward-sampler, which uses the filtered quantities μ_t, Σ_t as input.

  Initially sample $\hat{\beta}_T \sim N(\mu_T, \Sigma_T)$.

  For $t = (T-1)..1$,

  - Sample $\hat{\beta}_t \sim N(\mu_t + L(\hat{\beta}_{t+1} - \mu_{t+1|t}),\ \Sigma_t - L^2\Sigma_{t+1|t})$

    where $L = \Sigma_t(\Sigma_{t+1|t})^{-1}$

We have one modification to the standard DLM: while a β exists for all timesteps, there are many zero-count contexts without any event tuples. The Kalman filter is modified to skip the measurement step for those timesteps, so simply $\mu_t := \mu_{t|t-1}$ and $\Sigma_t := \Sigma_{t|t-1}$. We do not store η variables at those timesteps, since they are unnecessary for inference; but we do simulate them when creating posterior samples for analysis in the conflict detection task. (But the time-series plots of $E[\theta]$ in section 5 of the paper do not show these samples.)

We use a custom implementation of the filter and sampler that was tested via simulation in two ways: (1) comparing its inferences on simulated data to those from the *dlm* package in R (Petris, 2010), and (2) using the Cook *et al.* (2006) Bayesian software validation technique of checking the simulation distribution of inferred posterior quantiles of simulated parameters. The latter was useful for tesing other samplers as well (including the logistic normal inference algorithm described below).

[1] See also http://www.gatsby.ucl.ac.uk/~turner/Notes/1DKalmanFilter/1d_kalman_filter.pdf.

3.2 Logistic Normal $[\eta \mid z, \bar{\eta}]$

Next, we must resample the η variables; for every context, sample from the posterior density

$$p(\eta \mid \bar{\eta}, z) \;\propto\; N(\eta \mid \bar{\eta}, \Sigma)\;\; Mult(z \mid \theta(\eta)) \tag{7}$$

where $\bar{\eta} = \beta + \alpha$ denotes η's prior mean. This has an unnormalized log posterior density function

$$\ell(\eta) = \sum_k \left(-\frac{1}{2\sigma_k^2}(\eta_k - \bar{\eta}_k)^2 + n_k \log \theta(\eta)_k \right) \tag{8}$$

where n_k is the number of tuples in this context having frame k, and $\theta(\eta)$ is the value of θ deterministically associated with η via the softmax function.

Unfortunately, unlike the Dirichlet, a logistic normal prior on a multinomial is not conjugate; Equation 8 describes the unnormalized density, but there is no closed form for the normalized posterior (and more to the point, no known exact sampling algorithm).

As described in the paper, we use a Laplace approximation proposal—a Gaussian approximation centered at the mode, which can be justified as the second-order approximation to the log-posterior there—taking a proposed sample η^* via the steps

(1) Solve MAP $\hat{\eta} = \arg\max_\eta \ell(\eta)$

(2) Sample $\eta^* \sim N(\hat{\eta}, [H(-\ell(\hat{\eta}))]^{-1})$

where $H(-\ell(\hat{\eta}))$ denotes Hessian of the negative unnormalized log-posterior at $\hat{\eta}$.

Step #1 could be solved in a number of ways. We use a fast linear-time Newton algorithm from Eisenstein *et al.* (2011), which was faster than gradient descent methods we tried; we reproduce it below. The Newton step is

$$\eta := \eta - \lambda H^{-1} g$$

where the gradient of $-\ell$ is

$$g(\eta)_k = n\theta_k - n_k + \frac{1}{\sigma_k^2}(\eta_k - \bar{\eta}_k)$$

and the Hessian has diagonal and off-diagonal elements

$$H_{kk} = n\theta_k(1 - \theta_k) + 1/\sigma_k^2, \qquad H_{jk} = -n\theta_j\theta_k$$

where n is the number of event tuples in the context (i.e. number of individual z's). Matrix inversion is in general a cubic time algorithm, but we apply the Sherman-Morrison formula to only have to invert a diagonal matrix. For any invertible square matrix A and vectors u,v, the Sherman-Morrison formula gives an alternate expression for $(A + uv^\mathsf{T})^{-1}$ in terms of A^{-1}. For a diagonal matrix A and vectors u, v, w, we apply the Sherman-Morrison formula and configure the order of operations to avoid creating any non-diagonal matrices:

$$Z = (A + uv^\mathsf{T})^{-1} w \tag{9}$$

$$Z = A^{-1}w - [1 + v^\mathsf{T} A^{-1} u]^{-1}(A^{-1}u)(v^\mathsf{T} A^{-1} w) \tag{10}$$

$$Z_j = (A_{jj}^{-1} w_j) - \frac{1}{1 + \sum_k A_{kk}^{-1} v_k u_k}(A_{jj}^{-1} u_j) \sum_k A_{kk}^{-1} v_k w_k \tag{11}$$

where the last line shows the resulting vector for one element j.

The Hessian can be rewritten as a sum of diagonal and rank-1 matrix as $H = diag[n\theta_k + 1/\sigma_k^2] - n\theta\theta^\mathsf{T}$, thus the Newton step direction $H^{-1}g$ can be calculated in linear time by applying Eq. 10 with $A_{kk}^{-1} = (n\theta_k + 1/\sigma_k^2)^{-1},\; w = g,\; u = \sqrt{n}\theta,\; v = -\sqrt{n}\theta$.

Eisenstein *et al.* (2011) present this technique in the context of a variational inference algorithm, but actually it applies to any MAP logistic normal inference problem under diagonal covariance. We find it usually converges to an $\hat{\eta}$ estimate in only several iterations (using a line search,[2] first taking a step sized $\lambda = 1$, and if it's not an improvement, halving λ until it is.)

Step #2 is to sample from the multivariate normal $N(\hat{\eta}, H^{-1})$. The simplest MVN sampling algorithm is to take K samples from $N(0, 1)$ and multiply that vector by the Cholesky root of the covariance (and add the mean). But it takes cubic time to compute a Cholesky root,[3] which becomes too expensive for large values of K. Instead, we only invert the diagonal of the Hessian (linear time), resulting in a diagonal covariance (thus each $\eta_k^* \sim N(\bar{\eta}_k,\ 1/H_{kk})$); this is only an axis-aligned MVN approximation to the posterior.[4]

So this gives a η^{new} proposal. It is possible to simply update to it directly; but it is more accurate to use it as a Metropolis-Hastings proposal. Calculate the acceptance probability

$$a = \frac{p(\eta^{\text{new}}|\bar{\eta}, z)}{p(\eta^{\text{old}}|\bar{\eta}, z)} \frac{N(\eta^{\text{old}}; \hat{\eta}, H^{-1})}{N(\eta^{\text{new}}; \hat{\eta}, H^{-1})}$$

and accept the proposal at probability $\min(a, 1)$. The ratio of true posterior densities can be calculated with the unnormalized form in Equation 8.

See also Wang and Blei (2012) which develops a Laplace approximation for variational inference for several nonconjugate models including a logistic normal topic model. The Metropolis-Hastings approach we use here is similar to Hoff (2003).

3.3 Learning concentrations and variances

There are several parameters that control the overall variability of the above quantities. The Dirichlet concentration parameter b controls the similarity between the frames' predicate-path distributions; the autoregressive variance τ^2 controls how similar a dyad's latent positions are between timesteps; and the emission variances σ_k^2 controls how similar the frame distributions are for two contexts with identical latent states.

All these prior parameters are learned, thus naturally leading the model to learn highly likely levels of sparsity and variability. This is tremendously convenient in practice, since there are no hyperparameters that need to be tuned (beyond K and data preprocessing decisions). It also helps the model learn better solutions; for example, Asuncion *et al.* (2009) finds that Dirichlet concentration learning gives much better solutions for LDA.

The symmetric Dirichlet parameter b is learned with slice sampling (Neal, 2003), under an improper uniform prior for b. (In other experiments we have found different diffuse priors for b make little difference.) Slice sampling only requires an (unnormalized) posterior density function; with a uniform prior it's just the Dirichlet-multinomial likelihood, which is, integrating out ϕ,

$$L(b) = p(w \mid z, b) = \prod_{k=1}^{K} \frac{\Gamma(b)}{\Gamma(b + n_k)} \prod_{w=1}^{V} \frac{\Gamma(b/V + n_{k,w})}{\Gamma(b/V)} \tag{12}$$

where V is the verb-path vocbaulary size, n_k is the number of event tuples having $z = k$, and $n_{k,w}$ the number having frame k and verb-path w. An implementation speedup is possible noting that

[2] e.g. http://www.cs.cmu.edu/~ggordon/10725-F12/slides/11-matrix-newton-annotated.pdf

[3] At least by the naive algorithm; is there a shortcut here?

[4] And it's not even the factored marginals of $N(\hat{\eta}, H^{-1})$, since the diagonal of a Hessian inverse is not the same thing as the inverse of a Hessian diagonal.

each frame's lexical count vector $(n_{k,1}..n_{k,V})$ is usually very sparse with mostly 0's, so those terms can be skipped in the innermost loop. (Also draw out the $\Gamma(b/V)$ denominator.) This sparsity during Gibbs sampling is a natural consequence of the sparsity of language; it can be exploited in other ways to improve sampling efficiency, e.g. Yao *et al.* (2009).

The context model's variance terms are also learned. We use a conjugate inverse-Wishart prior, in inverse chi-squared parameterization (e.g. Murphy, 2012 §4.6.2.2) of $\chi^{-2}(\text{prior strength},\ \text{prior value})$, using diffuse prior $\chi^{-2}(1,1)$. However, since the amount of data is very high, the posterior intervals are very small (often less than 10^{-3}), and sampling is nearly equivalent to ML inference.

Technically, the conjugate sampling equations are

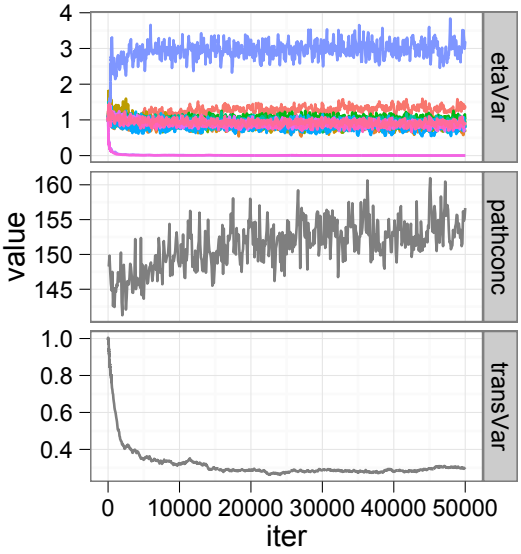
$$\tau^2 \sim \chi^{-2}\left(1+N,\ \frac{1}{1+N}\left[1+\sum_{s,r,t>1,k}(\beta_{s,r,t,k}-\beta_{s,r,t-1,k})^2\right]\right)$$

where N is $(K-1)\times$ NumDyads $\times$ (NumTimesteps-1), and

$$\sigma_k^2 \sim \chi^{-2}\left(1+N,\ \frac{1}{1+N}\left[1+\sum_{(s,r,t)\text{ where }n_{s,r,t}>0}(\eta_{s,r,t,k}-\bar{\eta}_{s,r,t,k})^2\right]\right)$$

where N is the number of contexts with non-zero events. In both cases N is hundreds of thousands to millions, swamping the prior pseudocount value of 1.

Here is a plot of the dispersion parameters over one Gibbs sampling run (all 10 σ_k^2's, b, then τ^2). The fact that dispersions are still drifting in early iterations is an indicator the sampler has not mixed. (Indeed, even though we attained useful results at iteration 10,000, and changing the number of iterations to higher numbers made little difference to the evaluation metrics, these plots clearly indicate mixing has not been achieved at that point. The inferences can be justified only as approximations (but useful ones) of the posterior.)



3.3.1 Softmax bug

The results in the ACL paper have an anomaly where one frame often has a very low probability mass, so it is essentially a $K-1$ dimensional topic model. (This is why the $K=2$ models es-

sentially learn only one frame, and thus have a similar lexical scale purity as the random choice baseline that would come from one big cluster of all words.) This was discovered to be due to a bug in the implementation: it clamped the K'th element of η to 0 (attempting to implement an alternate version of softmax with better identifiability), but only the first $K-1$ elements of θ were used in the posterior density evaluation for the MH step, so counts of $z = K$ were ignored in the likelihood. Thus the model would eventually shift θ_K to zero and put all the probability mass on the first $K-1$ elements. (It takes a while for the bug to cause this to happen, since the MAP optimum and Laplace approximation for η, given a fixed θ, is computed correctly. But the density ratio for the MH step prefers assigning low θ_K values.) If $\theta_K = 0$, then the model is exactly the same as a fully parameterized $K-1$ model; since it is close to zero, it is very similar to that.

References

Asuncion, A., Welling, M., Smyth, P., and Teh, Y. W. (2009). On smoothing and inference for topic models. In *Uncertainty in Artificial Intelligence*, volume 100.

Blei, D. M. and Lafferty, J. D. (2007). A correlated topic model of science. *AAS*, **1**(1), 17–35.

Carter, C. K. and Kohn, R. (1994). On Gibbs sampling for state space models. *Biometrika*, **81**(3), 541–553.

Cook, S. R., Gelman, A., and Rubin, D. B. (2006). Validation of software for bayesian models using posterior quantiles. *Journal of Computational and Graphical Statistics*, **15**(3).

Eisenstein, J., Ahmed, A., and Xing, E. (2011). Sparse additive generative models of text. In *Proceedings of ICML*, pages 1041–1048.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2003). *Bayesian data analysis*. Chapman & Hall/CRC.

Griffiths, T. and Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, **101**(Suppl 1), 5228.

Harrison, J. and West, M. (1997). *Bayesian forecasting and dynamic models*. Springer Verlag, New York.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Second Edition*. Springer.

Hoff, P. D. (2003). Nonparametric modeling of hierarchically exchangeable data. *University of Washington Statistics Department, Technical Report*, **421**.

Murphy, K. P. (2012). *Machine Learning: a Probabilistic Perspective*. MIT Press.

Neal, R. M. (2003). Slice sampling. *Annals of Statistics*, pages 705–741.

Petris, G. (2010). An R package for dynamic linear models. *Journal of Statistical Software*, **36**(12), 1–16. http://www.jstatsoft.org/v36/i12/paper.

Wang, C. and Blei, D. M. (2012). Variational inference in nonconjugate models. *arXiv preprint arXiv:1209.4360*.

Yao, L., Mimno, D., and McCallum, A. (2009). Efficient methods for topic model inference on streaming document collections. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 937–946. ACM.