Hamiltonian Monte Carlo for Regression with High-Dimensional Categorical Data*

Szymon Sacher
Columbia University

Laura Battaglia
Barcelona Graduate School of Economics

Stephen Hansen
Imperial College London

July 17, 2021

Abstract

Latent variable models are becoming increasingly popular in economics for high-dimensional categorical data such as text and surveys. Often the resulting low-dimensional representations are plugged into downstream econometric models that ignore the statistical structure of the upstream model, which presents serious challenges for valid inference. We show how Hamiltonian Monte Carlo (HMC) implemented with parallelized automatic differentiation provides a computationally efficient, easy-to-code, and statistically robust solution for this problem. Via a series of applications, we show that modeling integrated structure can non-trivially affect inference and that HMC appears to markedly outperform current approaches to inference in integrated models.

*Hansen gratefully acknowledges financial support from ERC Consolidator Grant 864863. The authors also thank the NumPyro development team for their outstanding work.

1 Introduction

As the amount of digitally recorded unstructured data continues to grow rapidly, empirical work in economics is increasingly incorporating it. The leading example of such data is text, which numerous papers in a variety of fields have recently used (Gentzkow et al. 2019), but also includes others such as survey responses, images, and audio recordings. The most relevant feature of unstructured data for statistical modeling is that observations typically have an enormous number of independent dimensions of variation. Moreover, this variation is often expressed in terms of integer counts rather than continuous variables.[1] Effectively handling such high-dimensional categorical data is therefore a major challenge in extracting information from unstructured data.

One common approach in the literature is to specify a statistical (typically Bayesian) model that projects each observation onto a low-dimensional latent space that captures the important variation in the high-dimensional feature space. In the natural language context, a popular latent variable model is *latent Dirichlet allocation* (LDA, Blei et al. 2003). In LDA the latent space represents "topics" and each document is a mixture over different topics. A recent selection of applications of LDA include macroeconomic forecasting (Larsen and Thorsrud 2019, Bybee et al. 2020, Thorsrud 2020, Ellingsen et al. 2021); conflict forecasting (Mueller and Rauh 2018); asset pricing (Hanley and Hoberg 2019, Lopez Lira 2019); political deliberation (Hansen et al. 2018, Stiglitz and Caspi 2020); central bank communication (Hansen and McMahon 2016, Hansen et al. 2019, Dieijen and Lumsdaine 2019); corporate finance (Adams et al. 2021); and media economics (Nimark and Pitschner 2019, Bertsch et al. 2021). Latent variable models for other data types closely related to LDA have also been adapted for survey data (Bandiera et al. 2020, Munro and Ng 2020, Draca and Schwarz 2021) and for network data (Nimczik 2017, Olivella et al. 2021).

In these and other applications in economics and finance, obtaining a latent representation of observations is typically not an end in itself but rather the first step in a larger econometric strategy. The latent variable model serves to transform unstructured data into a tractable, numeric form that is then plugged into a second-step regression model in which it is effectively treated as given data. Two issues arise with this approach. First, uncertainty in the latent representation from the first step is ignored in the second step, which invalidates standard inference procedures. Furthermore, the regression model usually imposes dependencies between latent representations and covariates that are ignored in the first step. This implies a potential loss of information in the first step,

[1] For example, one of the simplest representations of a textual corpus is the *bag-of-words* model in which each document is represented as a vector of integer counts over the unique vocabulary terms in the corpus. Even relatively small corpora contain thousands of unique dimensions. Moreover, the dimensionality grows even further as one consider richer linguistic units than individual words.

as assumptions about the relationship between data and covariates are ignored in the construction of the latent space.

The most natural way of overcoming these problems is to jointly specify the latent variable model and associated regression model within a single, integrated data generating process. While formulating such integrated models is relatively straightforward, conducting inference for them has to date been anything but, and required researchers to derive and code complex posterior inference algorithms every time they specified a new model. Applied economists do not normally have training in these methods, which in our view is the main obstacle that prevents the direct modeling of dependencies of interest in unstructured datasets.

We approach this problem with Hamiltonian Monte Carlo (HMC, MacKay 2003, Neal 2012), a Markov Chain Monte Carlo (MCMC) algorithm that uses information on the gradient of a joint distribution to sample from it. HMC is the basis for MCMC estimation in the popular probabilistic programming language Stan (Carpenter et al. 2017), which has been previously used in applied Bayesian econometrics (e.g. Meager 2019). But for posterior distributions with thousands of parameters or more and very high-dimensional observations, which is typical for unstructured data, the computational burden of computing gradients has until recently not been feasible. This has changed with the advent of highly efficient algorithms for automatic differentiation that utilize the massive parallelization capacity of modern computer hardware, in particular Graphical Processing Units (GPUs) and Tensor Processing Units (TPUs). These computational advances make possible gradient-based inference in large, complex Bayesian models including those for unstructured data.

The main contribution of our paper is to evaluate HMC implemented via parallelized automatic differentiation as a means of conducting inference in latent variable models for high-dimensional categorical data. To do so, we use the NumPyro package (Bingham et al. 2018, Phan et al. 2019), one of whose key innovations is the efficient computation of the gradients that underlie HMC. Through a series of applications on simulated and real-world datasets, we show that HMC has several major advantages.

The first is *ease of coding*. Models that until several years ago required hundreds, or even thousands, of lines of code to estimate can now be estimated using no more than a few dozen. This radically reduces the cost of integrated modeling, and brings it well within the technical capacity of applied economists.

The second is *computational efficiency*. Particularly when estimated on graphical processing units (GPUs), one can obtain outstanding computational performance and draw a large number of effectively independent samples from complex posterior distributions relatively quickly.

The third is *quality of inference*. One of the most popular existing integrated mod-

els for text and covariates is the structural topic model (Roberts et al. 2014, 2016) in which the topic shares of documents can depend on observables.[2] We compare the R package for the structural topic model (`stm`) (Roberts et al. 2019, with over 200,000 total downloads from Rstudio's CRAN mirror)—which implements a variational inference algorithm—with an HMC-based implementation based on automatic differentiation. In a basic simulation environment, we show that our implementation appears to outperform `stm` in both point and interval estimation. Another gradient-based approach to sampling is Langevin dynamics (LD) and its stochastic variant (SGLD, Welling and Teh 2011). Munro and Ng (2020) recently applied LD to model dynamic, latent structure in economic survey data. We compare HMC against LD within the Munro and Ng (2020) model, and show that LD converges slowly (or fails to converge within reasonable time), while HMC converges quickly and again produces high quality point and interval estimates. In short, our approach not only greatly simplify the coding of complex inference procedures,[3] but also appear to yield higher-quality estimates, at least in these leading applications.

Finally, and perhaps most importantly, is the *ability to develop new models*. By shifting the focus from inference to modeling, HMC with automatic differentiation allows the researcher to rapidly prototype alternative models and to experiment with incorporating different dependencies into the data generating process. We illustrate the ease with which new models can be built by extending the analysis in Bandiera et al. (2020) to account for situations in which latent variables both depend on observables and explain outcomes.

In summary, we introduce a new approach for inference in models of high-dimensional categorical data with latent variables that depend on the economic environment in which they are generated. HMC based on automatic differentiation shows excellent performance in simulated and real-world datasets along numerous criteria, and we believe should become an important part of the applied toolkit for unstructured data modeling going forward. It allows a flexible approach to model building without sacrificing the quality of inference, which so far has been lacking in the literature.

The rest of the paper proceeds as follows. Section 2 provides background on latent variable models and HMC. Section 3 discusses HMC-based estimation of the structural topic model of Roberts et al. (2014), while section 4 discusses HMC-based estimation of the dynamic latent variable model of Munro and Ng (2020). Section 5 illustrates how to build new regression models using HMC and applies them to the time use survey of Bandiera et al. (2020). Section 6 concludes.

[2]For a recent application in economics, see Conde-Ruiz et al. (2021).

[3]Both the `stm` code (`https://github.com/bstewart/stm`) and the code for replicating Munro and Ng (2020) (`https://github.com/evanmunro/dhlvm`) contain hundreds of lines of source code. The core functions for HMC inference in these two cases contain fewer than 50 lines.

2 Background

In this section, we first review the data generating process for latent Dirichlet allocation, which forms the core model from which later models are built. We then provide illustrative output from the minutes of Federal Open Market Committee meetings. Finally we introduce a basic overview of Hamiltonian Monte Carlo.

2.1 Data generating process for LDA

We first introduce basic notation for discrete data in the case of text. Suppose there are D total documents, with an individual document indexed by d. Each unique term in the collection of documents is indexed by an integer v, and there are V total unique terms.[4] Each document is represented as a list of term indices $\mathbf{w}_d$ of length N_d with generic element $w_{d,n} \in \{1, \ldots, V\}$. One can collapse $\mathbf{w}_d$ into a vector of term counts $\mathbf{x}_d \in \mathbb{Z}_+^V$ where $x_{d,v}$ is the count of term v in document d. The matrix whose rows are $\mathbf{x}_1, \ldots, \mathbf{x}_D$ is known as the *document-term matrix*.

LDA has two basic sets of model parameters. The first is K separate topics, which are each represented by V-dimensional probability vectors $\{\boldsymbol{\beta}_k\}_{k=1}^K$, where $\beta_{k,v}$ is the probability that the a single word drawn from topic k is term v. The second is D document-specific distributions $\{\boldsymbol{\theta}_d\}_{d=1}^D$, each of dimensionality K, that represent the association between documents and topics. Formally, each term in $\mathbf{w}_d$ has a topic assignment $z_{d,n} \in \{1, \ldots, K\}$ drawn independently from $\boldsymbol{\theta}_d$. The likelihood function for document d is therefore

$$p(\mathbf{w}_d \mid \boldsymbol{\theta}_d, \{\boldsymbol{\beta}_k\}_{k=1}^K) = \prod_{n=1}^{N_d} \sum_{k'=1}^{K} p(w_{d,n} \mid z_{d,n} = k', \{\boldsymbol{\beta}_k\}_{k=1}^K) p(z_{d,n} = k' \mid \boldsymbol{\theta_d}) \tag{1}$$

$$= \prod_{n=1}^{N_d} \sum_{k'=1}^{K} \beta_{k',w_{d,n}} \theta_{d,k'} \tag{2}$$

In principle, one could estimate the model by maximum likelihood applied to (1). In practice, the large number of parameters makes this approach prone to overfitting and MLE estimate is typically not unique (Ke et al. 2021). LDA introduces Dirichlet prior distributions over model parameters to help alleviate these problems and to specify a generative process for $\boldsymbol{\theta}_d$;[5] each $\boldsymbol{\beta}_k$ is drawn independently from $\text{Dir}(\eta)$ and each $\boldsymbol{\theta}_d$ is drawn

[4] Here term refers to a discrete unit from which language is built. This may or may not correspond to recognized English words depending on how text is preprocessed. For example, stemming a corpus often results in non-grammatical words. Also, the formation of multi-word expressions results in a single term that stands in for a phrase.

[5] The non-Bayesian version of the model is known as probabilistic Latent Semantic Indexing (Hofmann 2017).

independently from Dir(α).[6] The Dirichlet distribution is conjugate to the categorical distribution which makes it feasible to estimate the model using Gibbs sampler.

One common way of representing hierarchical Bayesian models is via plate diagrams that make explicit the independence assumptions imposed by a model. Figure 7a shows the plate diagram for LDA. The plates express repeated, independent draws of the random variables in the model. So, for example, $\boldsymbol{\theta}_d$ is inside a plate marked with D, which expresses that $\boldsymbol{\theta}_d$ is drawn D separate times. Arrows pointing to random variables express the distributions from which they are drawn; in this case, $\boldsymbol{\theta}_d$ is drawn from a Dirichlet distribution with hyperparameter α, hence the arrow connecting α with $\boldsymbol{\theta}_d$. Inside the document plate lies a term plate. Conditional on $\boldsymbol{\theta}_d$, the topic allocations $z_{d,n}$ for terms in document d are drawn independently N_d times.

The presence of discrete random variables in this formulation of LDA makes gradient-based sampling methods such as Hamiltonian Monte Carlo infeasible to implement. One can eliminate the latent topic assignments by instead modeling a document as a draw from a multinomial distribution. Given the above model, by the Law of Total Probability, the probability of observing term v in any given location in document d is $\sum_k \beta_{k,v}\theta_{d,k}$. Hence one can view document d as being generated by N_d independent draws from the V-dimensional probability vector $\sum_k \boldsymbol{\beta}_k\theta_{d,k}$ so that $\mathbf{x}_d \sim \text{Multinomial}\left(\sum_k \boldsymbol{\beta}_k\theta_{d,k}, N_d\right)$. The likelihood then becomes

$$p(\mathbf{x}_d \mid \boldsymbol{\theta}_d, \{\boldsymbol{\beta}_k\}_{k=1}^K) = \prod_v \left(\sum_k \beta_{k,v}\theta_{d,k}\right)^{x_{d,v}} \tag{3}$$

which depends only on continuous latent variables. Figure 7b shows a plate diagram for the formulation of LDA with the latent topic assignment marginalized out of the model. This formulation also highlights that LDA is a Bayesian factor model for discrete data, where the factors are the topics $\{\boldsymbol{\beta}_k\}_{k=1}^K$ and the document-specific factor loadings are $\{\boldsymbol{\theta}_d\}_{d=1}^D$.

2.2 Application of LDA to FOMC minutes

As an example corpus for illustrating the output of LDA, we use the minutes of Federal Open Market Committee (FOMC) meetings from 1994-2015 inclusive, a period that includes 176 meetings in total. Minutes are released several weeks after FOMC meetings,[7] and describe the main discussion that took place during the meeting. From the minutes

[6] The original (Blei et al. 2003) model places an asymmetric Dirichlet prior over the $\boldsymbol{\theta}_d$ terms and does not place a prior over the $\boldsymbol{\beta}_k$ terms. Here we specify the model with symmetric Dirichlet distributions over all categorical distributions since this is the most common formulation in economics papers. We show below models that introduce document heterogeneity via the prior distribution over $\boldsymbol{\theta}_d$.

[7] They are publicly available at `https://www.federalreserve.gov/monetarypolicy/fomc_historical.htm`.

we extract the paragraphs that relate to the discussion of economic conditions, which total $D = 1,778$. We then perform a sequence of standard pre-processing steps[8] to obtain a final dataset with 144,612 total terms and $V = 1,582$ unique terms. The average number of terms per paragraph is 81, with a standard deviation of 34.

To estimate LDA on the FOMC minutes, we use the collapsed Gibbs sampler of Griffiths and Steyvers (2004),[9] which is a standard and popular MCMC-based inference algorithm for LDA. The conditional distributions needed for drawing samples from the posterior are simple to derive given the conjugacy between the Dirichlet and multinomial distributions, and numerous efficient implementations exist (we use the Python `lda` package). For the illustration we choose $K = 10$ topics, fix hyperparameters at $\alpha = 1$ and $\eta = 0.3$, and draw 200 samples from the Markov chain (applying a thinning interval of 10).

Figure 1a displays the estimation results. The first objects of interest are the topic-term distributions $\{\boldsymbol{\beta}\}_{k=1}^{10}$. For each topic, we display the ten most likely terms in descending order across columns based on their posterior mean probabilities in topics. By and large, the word groupings are interpretable: one observes topics related to financial markets (topic 1), investment (topic 3), labor market (topic 9), and so on. The interpretability of the output of LDA is one of the reasons for its popularity.

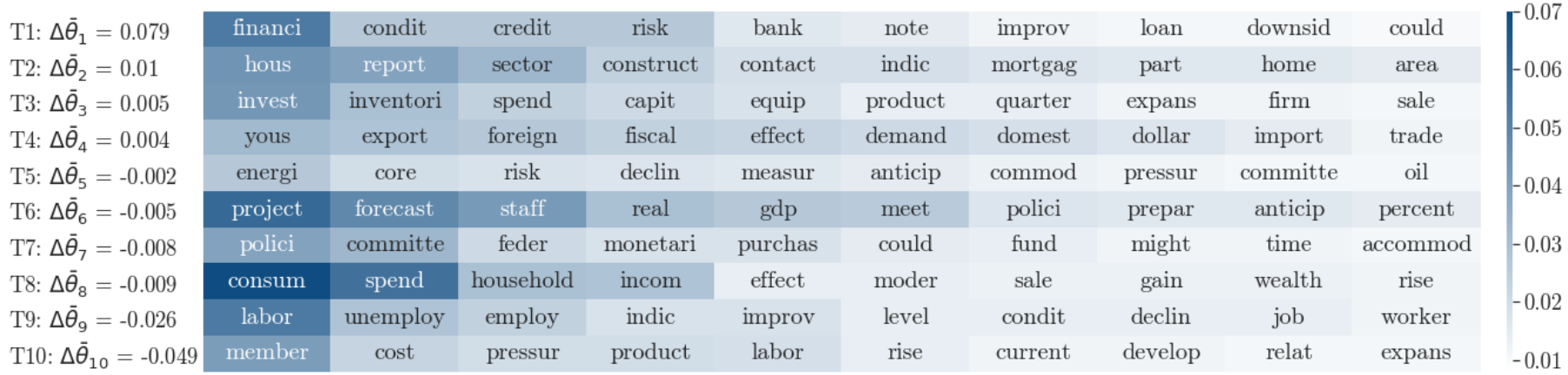
In addition, each paragraph has an estimated distribution over topics $\boldsymbol{\theta}_d$. In order to interpret variation in these, we first divide paragraphs into those generated during recessions and those generated during expansions. We then compute the average value of $\theta_{d,k}$ across all recession and expansion documents, respectively, for each topic k. The topics in Figure 1a are ordered according to the difference in these average values. For example, the words in paragraphs in recessions are estimated to be generated from topic 1 0.079 percentage points more frequently than words in paragraphs in expansions. The two most counter-cyclical topics relate to the financial and housing markets. As these markets were strongly associated with the largest downturn during our sample period, the document-level variation in topics that we estimate is natural. On the other hand, the most pro-cyclical topic relates to cost pressures, which are likely to be a salient driver of inflation during periods of growth.

2.3 Hamiltonian Monte Carlo

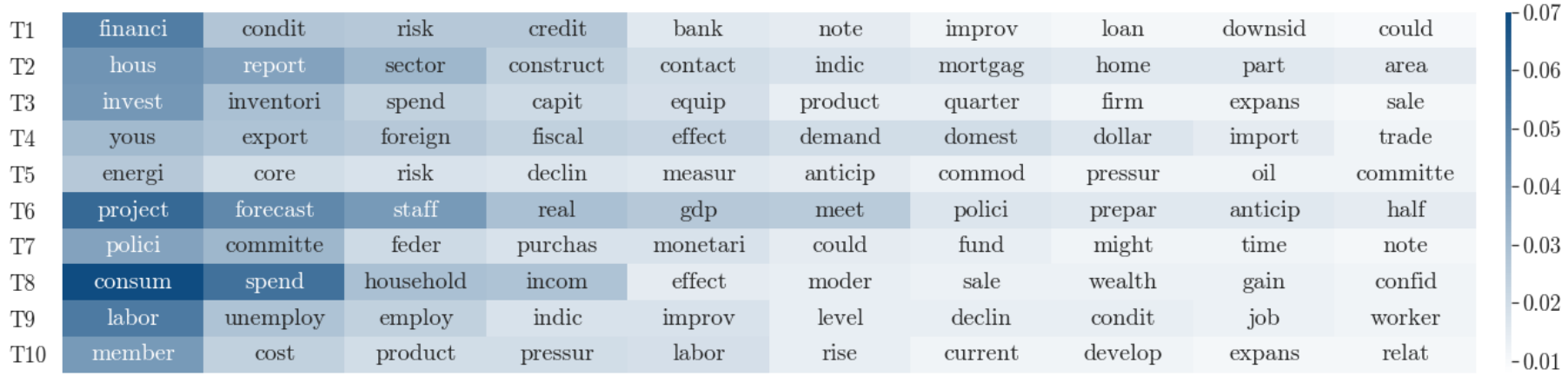
Presenting an in-depth discussion of HMC is beyond the scope of this paper, and our goal here is to give a high-level overview and an illustration of HMC applied to LDA.

[8] We expand contractions to their extended form (e.g. 'don't' becomes 'do not'); remove non-ASCII characters; lower-case all words; remove punctuation; remove numbers; remove stopwords; stem remaining words (with the Snowball stemmer); and trim stems that appear in fewer than 5 documents or more than 40% of documents.

[9] For full details of this approach, see material in Hansen et al. (2018).



(a) Collapsed Gibbs Sampler



(b) Hamiltonian Monte Carlo

Figure 1: Estimated Term-Topic Distributions for FOMC Minutes from Two Sampling Methods

Note: We estimate LDA on FOMC minutes from 1994 through 2015 using the collapsed Gibbs sampler of Griffiths and Steyvers (2004) (top panel) and using Hamiltonian Monte Carlo (bottom panel). Each row corresponds to a particular topic, and contains the ten most likely terms in the topic ordered left to right across columns, with shading indicating the magnitude of the probability $\beta_{k,v}$. Rows in the top panel are ordered top to bottom according to the average value of the $\theta_{d,k}$ across documents in recessions minus the average value of the $\theta_{d,k}$ in expansions. Topics from HMC are matched one-to-one with those from Gibbs sampling using minimum distance and are presented in the same order for comparability.

Excellent articles that cover the basic ideas that underpin HMC are Neal (2012), Hoffman and Gelman (2014), and Betancourt (2018). We are not aware of the application of HMC to LDA or related models in the literature.

Suppose $q(\mathbf{\Phi})$ is a posterior distribution over M parameters of interest $\mathbf{\Phi}$. HMC is an MCMC method for drawing samples from $q(\mathbf{\Phi})$ that uses information on its gradient. Whereas Gibbs sampling requires the conditional distributions $q(\Phi_i \mid \mathbf{\Phi}_{-i})$ to exist in closed form, HMC only requires the evaluation of the gradient of q. Hence HMC can be implemented on a much larger class of models. Moreover, by explicitly incorporating information about the shape of q, HMC explores q much more efficiently than algorithms whose proposals have random walk behavior, particularly when $\mathbf{\Phi}$ is high dimensional.

Sampling based solely on gradient information would tend to generate draws from the posterior located at the mode rather than from the entire region in which the posterior has non-negligible mass. To correct this problem, HMC introduces M auxiliary momentum variables $\mathbf{r} \sim \mathcal{N}(\mathbf{0}, I_M)$. A new sample for $\mathbf{\Phi}$ is obtained by first re-sampling the momentum variables and then following for a specified number of steps a path described by the Hamiltonian function:

$$H(\mathbf{\Phi}, \mathbf{r}) = -\log[q(\mathbf{\Phi})] + \sum_{i=1}^{M} \frac{r_i^2}{2}. \tag{4}$$

with associated dynamics

$$\frac{d\Phi_i}{dt} = \frac{\partial H}{\partial r_i} = r_i \quad \text{and} \quad \frac{dr_i}{dt} = -\frac{\partial H}{\partial q_i} = \frac{q_i(\mathbf{\Phi})}{q(\mathbf{\Phi})}. \tag{5}$$

The path described by (5) generates a stable 'orbit' around a mode of the posterior distribution. The role of the momentum variables' re-sampling is to allow the algorithm to explore all parts of the posterior that account for most of its mass (known as the *typical set*). The full HMC algorithm iteratively samples momentum variables, approximates the trajectories of Hamiltonian dynamics to generate proposals, and accepts or rejects them based on the value of the joint density at the beginning and end of the trajectory, as in the Metropolis-Hastings algorithm. Following the path allows for making distant proposals that nevertheless have a high chance of acceptance[10]. The sampled $\mathbf{\Phi}$ variables are retained and the momentum variables are discarded.

The specific variant of HMC that we use is the No-U-Turn Sampler (NUTS, Hoffman and Gelman 2014) implemented in NumPyro, a library for Python (Phan et al. 2019).

[10]The reason the Metropolis-Hastings correction is necessary is computational. If one could follow the Hamiltonian dynamics exactly, the value of the joint density of r and Φ at the beginning and the end of trajectories would be equal. However, the error introduced by approximating a continuous process with discrete steps necessitates the correction.

NUTS adds to the basic HMC algorithm a way of determining the length of the path for generating samples. The intuitive idea is to follow Hamiltonian dynamics until the resulting path begins to circle back to its starting point. This is efficient since it generates proposals relatively far from each other, thus reducing correlation between draws. The most popular implementation of NUTS is in Stan (Carpenter et al. 2017), which like NumPyro is a probabilistic programming language. The advantage of NumPyro is computational. The most costly part of implementing HMC is the computation of gradients of the log-likelihood, an operation that is amenable to parallelization. NumPyro allows users to deploy these computations to specialized hardware including Graphical Processing Units (GPUs) and Tensor Processing Units (TPUs), which for the models considered in this paper results in an order of magnitude or more reduction in computation time. All models estimated with HMC in this paper use a single Nvidia Tesla T4 GPU with 2,560 CUDA cores.

To illustrate HMC, we return to the estimation of LDA on the FOMC dataset. The basic Numpyro function for estimating LDA is displayed in section B.1. One aspect of note is the utter simplicity of the code. Estimating the model first requires the user to define the data generating process as a probabilistic function, which for LDA is expressed in nine lines. Second, only a few additional lines are needed to apply the provided NUTS sampler to this function. This is in contrast to the `lda` package above which contains hundreds of lines (`https://github.com/lda-project/lda`). To estimate LDA on the FOMC data, we use HMC to take 500 post-warmup draws from the posterior.

Since LDA is an unsupervised learning algorithm, the order of topics is not comparable across models, and we need to match topics from HMC with those estimated from the Gibbs sampler above. We form posterior mean estimates of $\{\boldsymbol{\beta}\}_{k=1}^{K}$ according to both approaches, then compute the pairwise distance between term-topic distributions from both methods. Each $\boldsymbol{\beta}_k$ distribution from HMC is minimally distant to a unique distribution from Gibbs sampling, which allows us to generate a one-to-one match based on minimum distance. Figure 1b displays the top terms in each topic estimated with HMC, and we observe very similar results to those associate with the Gibbs sampler in figure 1a. Figure 2 instead compares posterior mean estimates for $\theta_{d,k}$ and we again observe a high concordance. In summary, mean parameter estimates are nearly identical for HMC and Gibbs sampling, which suggests that one does not lose any information by relying on gradient information rather than fully specified conditional distributions.

Of course, our goal in introducing HMC for the estimation of latent variable models is not to replicate estimates that can be recovered equally well from Gibbs sampling, but to allow a more flexible approach to modeling. The rest of the paper explores how HMC can be used to estimate richer models beyond plain LDA.

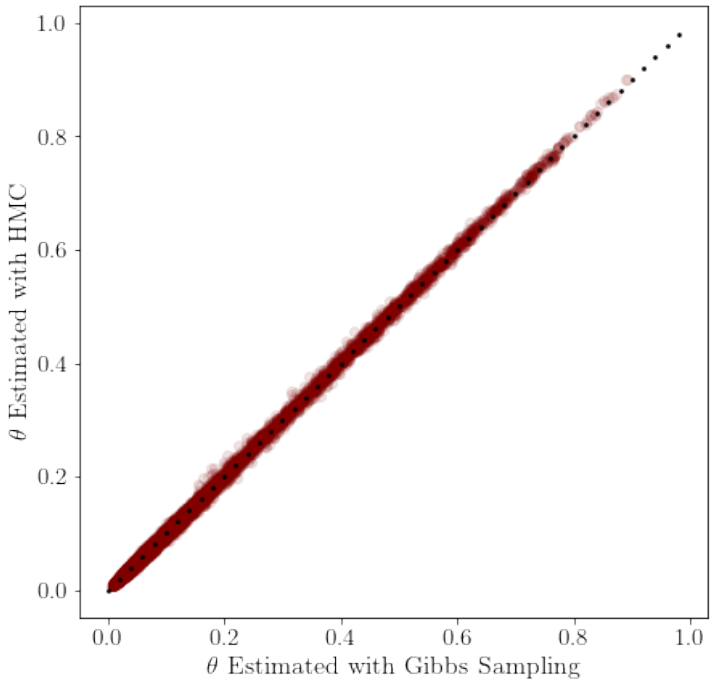


Figure 2: Comparison of Document-Term Distributions from Gibbs Sampling and HMC

Note: For each paragraph d in the FOMC data, we store the posterior mean document-term distribution $\boldsymbol{\theta}_d$ estimated by Gibbs sampling and by HMC. The ordering of topics from both approaches is given in Figures 1a and 1b, respectively. This Figure displays a scatter plot of the elements of $\boldsymbol{\theta}_d$ from both approaches. The dotted line is the 45-degree line.

3 HMC for Structural Topic Model

In the previous section, we observed that estimated topic shares in the FOMC minutes appear to move with the business cycle. To formalize this dependence, one approach would be to regress the estimated $\theta_{d,k}$ parameters from LDA on a recession indicator, and such a two-stage process is indeed common in applied econometric work that uses topic models to represent text. This strategy is a useful starting point for characterizing how documents' language varies with observables, but is problematic from the standpoint of statistical inference. The second-stage regression ignores the uncertainty in the topic shares inherited from the first stage. On one hand this makes default standard error computations invalid. On the other hand, the estimation is inefficient, since it treats all documents equally even though $\theta_{d,k}$ for some documents is far less uncertain (for example for the longer ones) and these documents should be weighted more heavily. In addition, the estimation of the topic model in the first stage assumes all documents are exchangeable, and ignores the dependence structure between words and relevant covariates assumed in the second stage. A natural solution to these problems is to model the relevant structure directly, and perform inference in the joint model.

Our starting point is the *structural topic model* (STM, Roberts et al. 2014, 2016),

which allows topic shares to depend on covariates. Here we present a simplified version of the STM to provide an initial illustration of how to jointly model text and covariates and to compare alternative approaches to inference.[11] The data generating process is similar to that of LDA, but with a different prior distribution on $\theta_{d,k}$. The relationship between document-level covariates $\boldsymbol{g}_d$ and topic shares is first modeled as

$$\tilde{\theta}_{d,k} = \boldsymbol{\gamma}_k^T \boldsymbol{g}_d + \varepsilon_{d,k} \quad \text{where} \quad \varepsilon_{d,k} \sim \mathcal{N}(0, \sigma^2), \tag{6}$$

which defines a linear regression model relating a set of real-valued, auxiliary topic-share parameters $\tilde{\theta}_{d,k}$ to the covariates. The mapping to the simplex, where topic shares live, is via the softmax function

$$\theta_{d,k} = \frac{\exp(\tilde{\theta}_{d,k})}{\sum_{k'} \exp(\tilde{\theta}_{d,k'})}. \tag{7}$$

(6) and (7) together define a logistic normal prior distribution over $\boldsymbol{\theta}_d$. This prior structure is convenient since it expresses dependencies via standard regression models in the space of real numbers which the Dirichlet is unable to capture. Since the K-dimensional simplex has $K-1$ degrees of freedom, we normalize $\tilde{\theta}_{d,k} = 0$ for some k in order to identify the remaining coefficients. The rest of the STM is as in LDA. K topic-term distributions $\{\boldsymbol{\beta}_k\}_{k=1}^K$ are each drawn independently from a Dirichlet prior distribution with parameter η, and word counts $\mathbf{x}_d$ are drawn from Multinomial $(\sum_k \boldsymbol{\beta}_k \theta_{d,k}, N_d)$.

While the replacement of the Dirichlet with the logistic normal allows the STM to be more expressive, it also substantially complicates inference since the logistic normal is not conjugate to the multinomial. This makes deriving exact conditional distributions for the STM posterior infeasible, and Gibbs sampling cannot be used as in LDA. Instead, Roberts et al. (2014) use variational inference (VI) for parameter estimation. VI assumes a simplified form for the posterior distribution, and then finds the member of the simplified class that is closest to the true posterior via an optimization problem. There are fewer statistical guarantees in VI than for MCMC methods but VI does allow the computationally efficient estimation of complex Bayesian models.

An alternative approach to inference for the STM is Hamiltonian Monte Carlo based on the NUTS algorithm. Although we are unaware of the application of HMC to inference in topic modeling (STM or otherwise), it has attractive features. Since HMC is an MCMC method, it allows one to draw samples from the true posterior rather than obtain a variational approximation. Also, formulating the optimization problem for variational inference is often not straightforward and involves extensive mathematical derivations that vary from model to model. In contrast, Appendix B.2 contains a NumPyro function needed to draw samples from the STM posterior. It is a basic extension of the LDA model

[11] The original STM model also allows the topic-term probabilities $\beta_{k,v}$ to depend on covariates, but for simplicity we ignore that dependency in this section.

coded in B.1 and builds the joint distribution in simple, transparent steps.[12] Of course, how HMC actually performs for inference is an empirical question, which we consider in the remainder of this section.

The basic question we ask is to what extent different approaches to inference are able to detect whether covariates drive topic coverage. The first approach ignores the dependency structure between covariates and language by estimating LDA with the collapsed Gibbs sampling algorithm of Griffiths and Steyvers (2004) and then treating the estimated topic shares as dependent variables in a separate regression model. The second is to use the `stm` package from Roberts et al. (2019) which implements a variational inference algorithm. The third uses HMC to sample from the posterior with the NUTS algorithm implemented in NumPyro.

3.1 Simulation exercise

We begin to answer this question with a simulation exercise. In each simulation there are 100 documents of length 25 and a potential vocabulary size of 500. We consider two topics and normalize $\tilde{\theta}_{d,2} = 0$. The share of the first topic is $\tilde{\theta}_{d,1} = \gamma_0 + \gamma_1 g_d + \varepsilon_d$ where $\gamma_0 = \gamma_1 = 1$; $\varepsilon_d \sim \mathcal{N}(0,1)$; and $g_d \sim \mathcal{N}(0, \sigma_g^2)$ where σ_g^2 is chosen so that the mean topic proportion $\theta_{d,1}$ when $g_d = 2\sigma_g$ is 0.75. This last features ensures that relatively few documents have coverage concentrated mainly on one topic. Each $\boldsymbol{\beta}$ is drawn from a Dirichlet distribution with concentration parameter $\eta = 0.2$. In total we simulate 50 datasets according to these distributions.[13] Our key question is to what extent alternative procedures produce high-quality estimates of γ_1, which determines the effect of the covariate on language and would typically be the primary object of interest in an econometric study.

Method 1: separated model For each simulation, we estimate LDA on the realized word counts by collapsed Gibbs sampling using $\alpha = 1$ and $\eta = 0.2$ with the Python `lda` package, taking 500 draws from the Markov chain (applying a thinning interval of 10 iterations). To obtain a point estimate of γ_1 we fit the linear model

$$\log\left(\frac{\hat{\theta}_{d,1}}{\hat{\theta}_{d,2}}\right) = \gamma_0 + \gamma_1 g_d + \varepsilon_d \tag{8}$$

[12] Stochastic variational inference (Hoffman et al. 2013) seeks to simplify VI via automatic differentiation, which also greatly simplifies coding requirements.

[13] The simulated data need not, and typically does not, have 500 unique words. In estimation we treat V as the realized number of unique words.

where $\hat{\theta}_{d,k}$ is the average value of $\theta_{d,k}$ across draws.[14]. The fitted value $\hat{\gamma}_1$ is the point estimate. To approximate the 95% confidence interval we follow a bootstrap procedure:

1. For every draw $s \in 1, \ldots, 500$ of the Markov chain, fit model (8) to obtain point estimate $\hat{\gamma}_1^s$ and associated standard error $\hat{\sigma}_1^s$.

2. For every draw s, draw 1,000 samples from $\mathcal{N}(\hat{\gamma}_1^s, \hat{\sigma}_1^s)$.

3. Pool all $500 * 1,000$ draws and discard the bottom and top 2.5%.

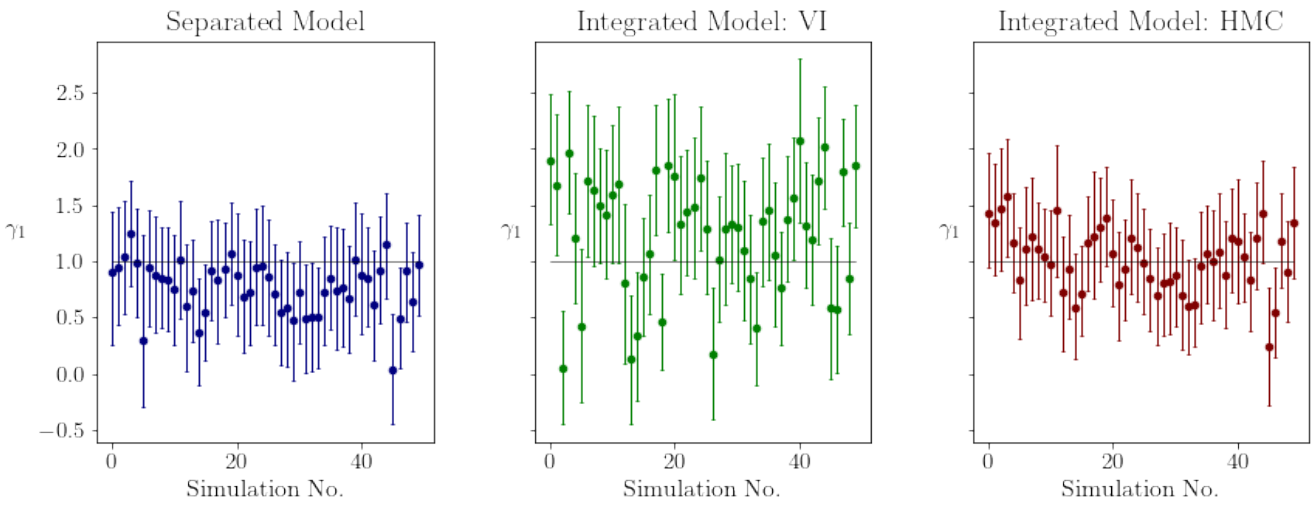
While many studies report asymptotic confidence intervals from the estimation of (8) directly, the bootstrap procedure accounts for the variation in draws around their mean values. In experiments we found that treating inferred topic shares as data may significantly overestimate the precision of coefficients γ.

Method 2: integrated model with Variational Inference For each simulation, we fit the STM with the `stm` package in R which provides a point estimate of γ_1 based on variational inference. To construct an interval estimate, we take 150 draws from the estimated variational posterior for the document-level $\boldsymbol{\theta}_d$ terms and follow the same bootstrap procedure as for Method 1.

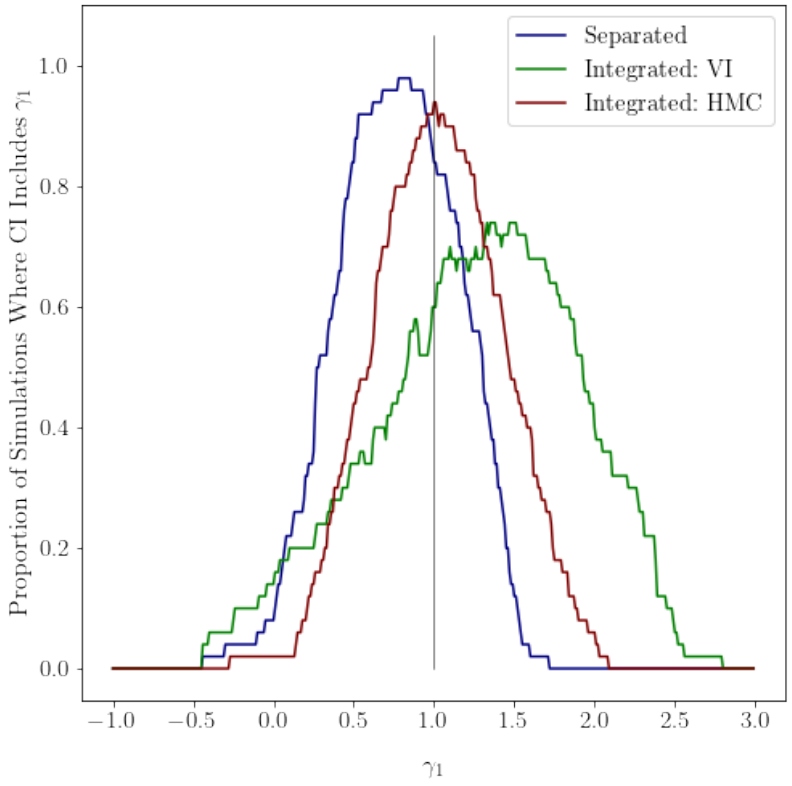
Method 3: integrated model with Hamiltonian Monte Carlo For HMC we use a $\mathcal{N}(0, 5)$ prior distribution on the γ_0 and γ_1 terms and a Dirichlet prior on the $\boldsymbol{\beta}_k$ terms with concentration $\eta = 0.2$. We fix $\sigma_g = 1$. We draw 2,000 samples after a warmup period of 1,000 draws. 95% credible intervals are constructed by discarding the bottom and top 2.5% of draws.

Figure 3 displays the results of the simulation. The top panel plots point and interval estimates produced by each method for each simulated dataset. The point estimates from the separated model consistently underestimate the strength of the effect of covariate. Intuitively, the estimation of LDA ignores the impact that g_d has on topic coverage and so is less able to separate documents on the basis of observables. On the other hand, the variational inference algorithm consistently overestimates the strength of the effect although across simulations point estimates are also quite variable. This suggests that the independence assumptions imposed by the variational approximation lead to a material loss of information in estimation. Finally, the HMC point estimates are evenly distributed around the true value and are much less variable across simulations. This shows the gain to sampling from the true posterior distribution.

[14]In the more general case with K topics and $\tilde{k}$ the normalized topic, there would be $K-1$ regressions with dependent variables $\log\left(\frac{\hat{\theta}_{d,k}}{\hat{\theta}_{d,\tilde{k}}}\right)$ for $k = 1, \ldots, \tilde{k}-1, \tilde{k}+1, \ldots, K$.



(a) Point and interval estimates



(b) Frequency of inclusion in interval estimate

Figure 3: Simulation Results

Note: The figures show results from the simulation exercise comparing three inference algorithms for estimating the γ_1 coefficient in a structural topic model, which captures the impact of variation in a covariate on topic coverage. The true value of γ_1 is 1. The algorithms were used to estimate the coefficient across 50 simulations. The top panel shows point estimates as dots and interval estimates as vertical lines around the dots. The horizontal grey line is the true value. The bottom figure displays the fraction of simulations in which different values of γ_1 appear in interval estimates.

In order to study the implications for hypothesis testing, we construct figure 3b. This shows in which fraction of simulations different values for γ_1 appear in the interval estimates. The true value of $\gamma_1 = 1$ is most likely to appear in the interval estimates produced by HMC (47 out of 50 times, consistent with our choice of 95% level for the credible intervals), followed by the separated model, followed by variational inference. The biases notable in the point estimates feed through into hypothesis tests. The separated model fails to reject the null hypothesis that γ_1 is as low as 0.5 in around 80% of the simulations, while the variational inference algorithm consistently fails to reject the null that γ_1 is as high as 1.5.

Overall we observe that HMC achieves high-quality inference in this setting, as well as being the algorithm that is easiest to code and implement.

3.2 Application on FOMC minutes

In order to illustrate the structural topic model on real data, we revisit the FOMC minutes corpus from the previous section. The STM allows us to explicitly model topic coverage as a function of the state of the economy, which we do via the regression

$$\tilde{\theta}_{d,k} = \gamma_{0,k} + \gamma_{1,k}\mathrm{R}_d + \varepsilon_{d,k} \tag{9}$$

where R_d is an indicator for whether paragraph d is generated during a US recession (as defined by the NBER). For the separated model, we use the LDA estimation from the previous section along with the inference procedure described in the simulation to obtain estimates for $\gamma_{1,k}$. For HMC and VI, we use the same prior specifications as in the simulation, but initialize values of $\boldsymbol{\beta}_k$ to the posterior means from plain LDA to maintain comparability of models. We choose the normalizing topic as the one that varies across recessions the least according to table 1a, i.e. topic 5.

Table 4 contains the results. In line with the simulation exercise, the separated model finds the most muted effects of recession on topic coverage, and identifies two topics (1 and 10) as varying significantly with recessions. HMC also estimates a significant effect on these two topics, but with much larger magnitudes. In addition, HMC identifies two additional topics (7 and 9) as also significantly related to recessions. Variational inference continues to exhibit large estimated intervals, meaning it only finds two significant effects. Hence HMC appears most able to detect dependencies between recession and topic coverage.

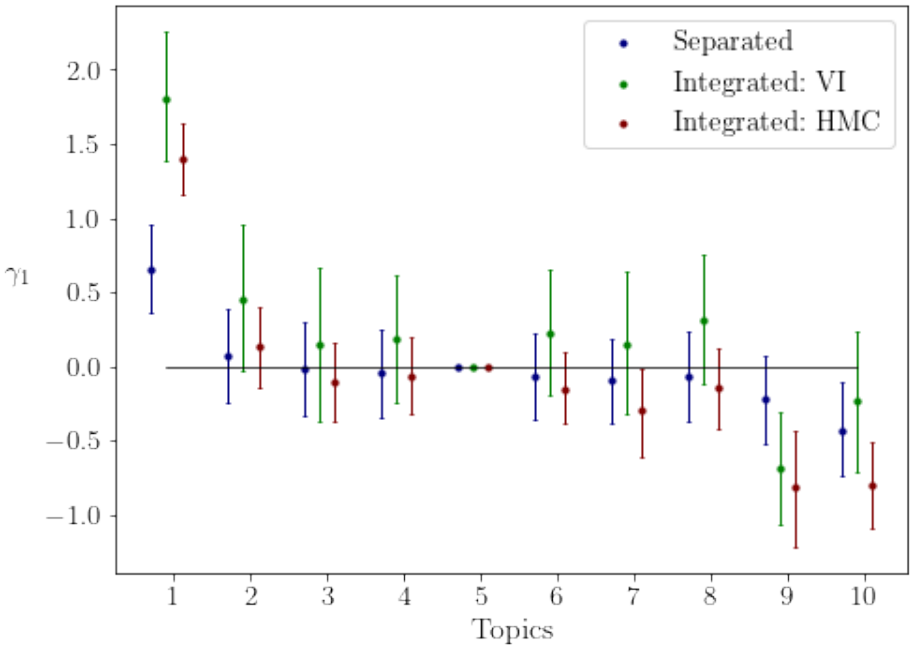


Figure 4: Estimated Effect of Recession on FOMC Topic Coverage

Note: We use a structural topic model to express the dependence of topics in FOMC minutes on a US recession indicator. This figures displays point and interval estimates for the regression coefficients on the indicator across topics. The topics are initialized at values given in table 1a.

4 HMC for Dynamic Survey Responses

The next model we consider is the dynamic model of survey responses from Munro and Ng (2020). As with text data, survey data with ordinal responses can be modeled using latent variables that capture correlation patterns in responses across individuals. Suppose we consider a survey with J total questions, where question j has L_j options. For example, respondents may be asked whether they disagree, neither agree nor disagree, or agree with a sequence of statements about political views, in which case $L_j = 3$. One would expect an 'agree' response to the question 'do you support abortion rights?' to be positively correlated with an 'agree' response to the question 'do you support gay marriage?'. One can use latent variables to capture this correlation, which here would represent political ideology. However, especially in surveys regarding beliefs about the economy, there is likely to be a dynamic component to beliefs as respondents adjust their views in response to new information about macroeconomic conditions.

Munro and Ng (2020) propose a model that admits such dynamics. Suppose that in period t there are N_t respondents to a survey with J questions; that is, the structure of the survey remains the same over time but the size of the population of respondents

potentially varies.[15] $\{\boldsymbol{\beta}^j\}_{k=1}^K$ are K separate distributions that represent response profiles for question j associated with different latent types. Person d is assigned a type $z_{d,t} \in \{1, \ldots, K\}$ and her response to question j, which we denote $x_{d,j,t} \in \{1, \ldots, L_j\}$, is drawn from $\boldsymbol{\beta}^j_{z_{d,t}}$. Here there are two notable difference compared to LDA. Whereas the case of text only requires the specification of one distribution $\boldsymbol{\beta}_k$ for each type k, here there are J separate collections of type distributions that allow each question to have a unique feature space. Second, while LDA allows different words in the same document to have different latent topic assignments, each individual in this model is assigned a single type which then determines the distributions over all questions. This feature can be relaxed when the same individual repeatedly answers the same question, as we show in the next section.

The specific data-generating process is the following:

$$
\begin{aligned}
\boldsymbol{\beta}^j_k &\sim \text{Dirichlet}(\eta^j) \\
\sigma_k &\sim \text{InverseGamma}(v_0, s_0) \\
\tilde{\theta}_{k,t} &\sim \text{Normal}(\tilde{\theta}_{k,t-1}, \sigma_k^2) \\
\boldsymbol{\theta}_t &= \text{Softmax}(\tilde{\boldsymbol{\theta}}_t) \\
z_{d,t} &\sim \text{Multinomial}(\boldsymbol{\theta}_t) \\
x_{d,j,t} &\sim \text{Multinomial}(\beta^j_{z_{d,t}})
\end{aligned}
\tag{DSR}
$$

This hierarchical model is similar to LDA, but to introduce natural dynamics into the distribution from which latent types are drawn, the Dirichlet prior on $\boldsymbol{\theta}_t$ is replaced by a logistic normal whose mean evolves according to a random walk[16]. This allows for a smoothly evolving type distribution, and is similar to models that introduce dynamics into generative models for text (Blei and Lafferty 2006). The plate diagram for the model is in figure 9 (which omits the hyperparameters of the prior distributions).

As with the structural topic model, the replacement of the Dirichlet distribution with the logistic normal complicates inference. Rather than use variational inference, however, Munro and Ng (2020) tackle the problem with an MCMC approach based on Langevin dynamics (LD), which is a special case of HMC in which gradient information is used to update the Markov chain based on a single discrete step taken along the path defined by Hamiltonian dynamics (Neal 2012).[17] For this reason, one would expect the exploration

[15] Munro and Ng (2020) ignore any panel structure and any repeat respondents are treated as separate individuals across all periods in which they participate.

[16] The random walk assumption implies that $\text{Var}(\tilde{\theta}_{k,t})$ increases with t and tends to infinity as $t \to \infty$. A more realistic model might assume a stationary AR1 process. It would be straightforward to include this structure as an extension to (DSR) with HMC, but we retain the random walk behavior for comparability with the original model.

[17] An extension of LD is stochastic gradient Langevin dynamics (SGLV, Welling and Teh 2011). In SGLD the gradient of the posterior is computed for a randomly drawn subset of observations, whereas

of the posterior distribution under LD to be less efficient than that under NUTS since NUTS traverses longer paths and so is likely to draw more uncorrelated samples. Another advantage of Numpyro's implementation of NUTS over the LD implementation of Munro and Ng (2020) is that the former computes derivatives automatically whereas the latter supplies an explicit functional form for the gradient to the sampler. This allows for much terser code, as can be seen in appendix B.3 where we supply the Numpyro function for sampling from the model defined in (DSR).

4.1 Simulation exercise

To explore the performance of LD and HMC, we again begin with an illustrative simulation exercise. In each simulation, we have 50 time periods with 10,000 individual survey respondents. In total there are eight survey questions, four of which have five responses and four of which have six responses. We specify $K = 4$ latent types, and draw $\boldsymbol{\beta}_k^j$ from a symmetric Dirichlet distribution with hyperparamter $\eta = 0.1$. We draw initial values $\tilde{\theta}_{k,0} \sim \text{Normal}(0, 1)$ and use $\sigma_k = 0.1$ for simulating the VAR.

For HMC estimation, we take 2,000 draws after discarding the first 500 as warmup. Estimation time varies from 12 to 17 minutes per simulation. For LD estimation, we use the code available at `https://github.com/evanmunro/dhlvm` and take 20,000 draws after discarding the first 5,000. Estimation time varies from 35 to 40 minutes per simulation on 3.5GHz Intel Xeon processors. While we take ten times as many draws from the Markov chain that evolves according to LD, the important question is which method produces more information about the posterior distribution which is also a function of the (lack of) autocorrelation in draws.

Table 1 contains the key simulation results for the population type distributions represented by the $\theta_{k,t}$ parameters. Across all such parameters and all simulations, i.e. $4(\text{latent types}) \times 50(\text{time periods}) \times 20(\text{simulations}) = 4,000$ in total, we compute the error as the difference between the posterior mean value of the parameter and its true value from the simulation. The first row of table 1 displays the mean value of the error, as well as its distribution across parameters, for both estimation methods. On average, HMC and LD produce accurate point estimates. However, in the tails of the distribution we observe substantially higher errors for LD than for HMC. Hence, from the perspective of accurate point estimation, HMC appears to outperform LD.

Since LD and HMC are both sampling methods, we can use standard Bayesian diagnostics for MCMC estimation to compare the extent to which they explore the posterior and accurately represent its shape beyond the first moment. The first statistic we report is effective sample size (ESS), which depends on the estimated autocorrelation between

the implementation of Munro and Ng (2020) computes the gradient over all observations.

Table 1: Simulations Results for Estimation of Type Probabilities

		Mean	Quantile			Frac > 1.1
			0.05	0.5	0.95	
Error	HMC	0	-0.03	0	0.03	
	LD	0.01	-0.1	0	0.16	
Effective Sample Size	HMC	424.43	165.39	402.04	747.15	
	LD	4.65	1.31	3.3	11.14	
Gelman-Rubin $\hat{R}$	HMC	1	1	1	1.01	0
	LD	1.48	1.03	1.39	2.12	0.84

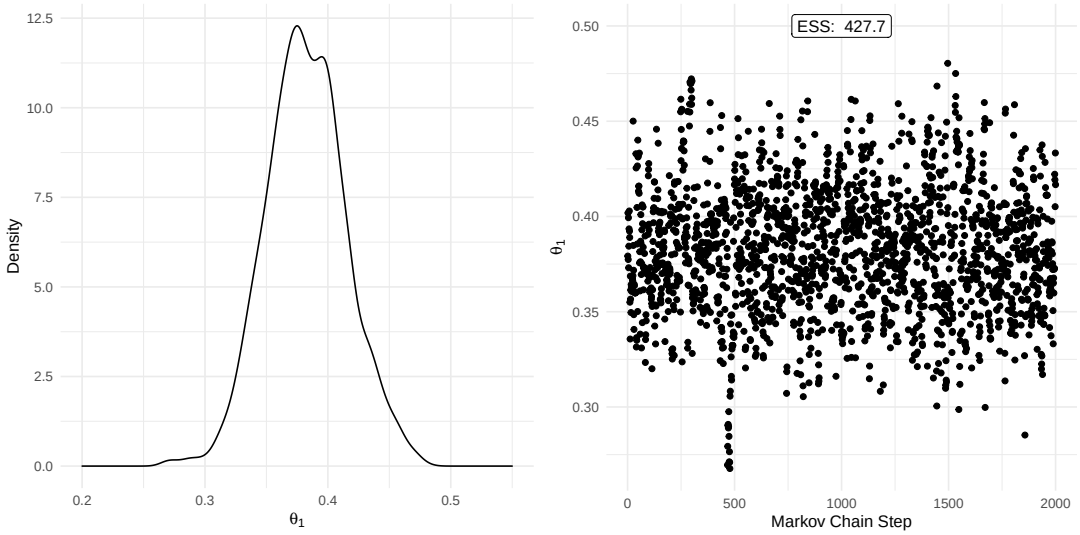
Note: In each of 20 simulations, we record draws for type probabilities $\theta_{k,t}$ in model (DSR) from Markov chains that evolve according to Hamiltonian Monte Carlo and Langevin dynamics, respectively. Error is defined as the difference between the posterior mean and the true value of a parameter. Gelman-Rubin $\hat{R}$ is a measure of convergence; the closer its value is to 1, the stronger the evidence for convergence. Values larger than 1.1 are typically taken as evidence that the Markov chain has not converged. Effective sample size accounts for autocorrelation in Markov chains and is the number of effectively independent draws from the posterior distribution. HMC was run for 2,000 post-warmup iterations, LD was run for 20,000 post-warmup iterations. Results in this table are pooled across all simulations.

draws of a Markov chain and represents the number of effectively independent draws from the posterior distribution.[18] The second statistic we report is Gelman-Rubin $\hat{R}$ (Gelman and Rubin 1992), which assesses the extent to which a Markov chain has converged to a stationary distribution. Its lowest value is 1, which indicates convergence, with higher values indicating a failure to converge.

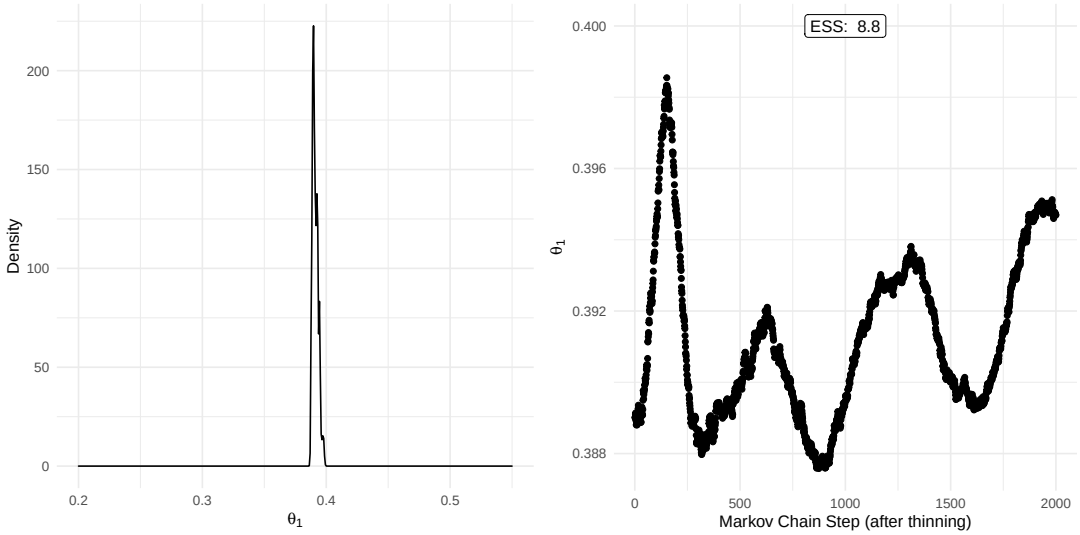
The second and third rows of table 1 display results for both diagnostics, and we observe enormous differences between methods. In spite of our drawing 20,000 *total* samples, LD produces a very limited number of *effective* samples. This suggests that LD does not effectively explore the posterior distribution and the samples remain strongly correlated for very long periods of time. In contrast, HMC produces a large number of effective samples, and along the same order of magnitude as the number of samples (2,000). This implies that one can accurately compute moments of the posterior distribution using the HMC samples, for example the credible intervals around the $\theta_{t,k}$ parameters. Consistent with a slow exploration of the posterior, we also observe that LD fails to converge for the majority of parameters, whereas $\hat{R}$ is close to 1 in all cases.

Figure 5 provides striking visual evidence of the implications of the differing abilities of LD and HMC to explore the posterior distribution. For the first simulation, we focus on a single parameter $\theta_{1,25}$, i.e. the probability of the first latent type being drawn in the period-25 population. The left-hand plots represent the estimated posterior distributions

[18] See section 11.5 of Gelman et al. (2013) for a detailed discussion; we use formula 11.8 computed via the `arviz` package in Python.



(a) Hamiltonian Monte Carlo



(b) Langevin Dynamics

Figure 5: Posterior Distribution and MCMC Time-Series for an Example Parameter.

Note: On the left, the plots display inferred posterior distribution of an example parameter ($\theta_{1,25}$) in a single simulation obtained using Langevin dynamics and Hamiltonian Monte Carlo. LD seemingly underestimates the posterior variance. On the right, the plots show the value of this parameter at different steps of the Markov Chain. For clarity, LD samples are thinned by the factor of 10. Samples obtained with LD display vary high autocorrelation which results in low effective sample size. Note that the vertical scale on the right-hand plots is not the same.

of this parameter produced by each method. While both share similar modes, LD estimates a nearly degenerate distribution while HMC produces a full distribution. While we cannot obtain the true posterior distribution, uncertainty quantification based on LD is likely to be highly misleading. The different shapes of the posterior are linked to the time series behavior of the Markov chains from which samples are drawn, and which we show in the right-hand plots. HMC produces a cloud of uncorrelated draws that vary over a significant range, while LD produces highly correlated draws that move within a narrow range (note the vertical scales in the plots differ).

The main conclusion from this analysis is that Hamiltonian dynamics are much more effective at exploring the posterior distribution of the dynamic survey response model than Langevin dynamics. While point estimates are comparable, LD generates almost no variation around those estimates which limits its effectiveness for uncertainty quantification and accurately estimating other higher-order moments. Moreover, from an implementation perspective, HMC via Numpyro is substantially easier to code since all that is required is the specification of a joint distribution in a single, compact function. We therefore see very few downsides of adopting HMC over LD in latent variable models for categorical data.

4.2 Michigan Consumer Survey

In order to assess whether these same properties hold on real data, we also use LD and HMC to estimate the model in (DSR) on the University of Michigan's Survey of Consumers on the same sample as do Munro and Ng (2020). This consists of monthly survey responses from approximately 500 participants from January 1978 through May 2019, with a total of 204,944 responses during this period. The survey consists of 14 total questions related to beliefs on economic conditions, such as 'Will you be better or worse financially a year from now?' with the possible responses 'worse', 'same', and 'better'. (See `https://data.sca.isr.umich.edu/` for additional details). Like Munro and Ng (2020) we estimate a model with $K = 4$.[19] Figure 6(a) plots posterior means and 95% credible intervals for each estimated $\theta_{k,t}$. For three of the four types, posterior means evolve in a similar way over time, and, as Munro and Ng (2020) explain, the time series variation relates strongly to various macroeconomic indicators. However, the two methods produce quite distinct time series for $\theta_{2,t}$: for HMC, the time series exhibits cyclical variation while, for LD, the time series appears less interpretable and $\theta_{2,t}$ hovers near zero for the final 15 years of the sample. Moreover, as in the simulation above, HMC produces meaningful uncertainty bands around the posterior means while LD does not.

[19] We follow Munro and Ng (2020) in not normalizing the value of a particular latent class k' to have $\tilde{\theta}_{k',t}$ for all t even though, as discussed in the previous section, K distinct values $\tilde{\theta}_{1,t}, \ldots, \tilde{\theta}_{K,t}$ are not uniquely identifiable in this model.

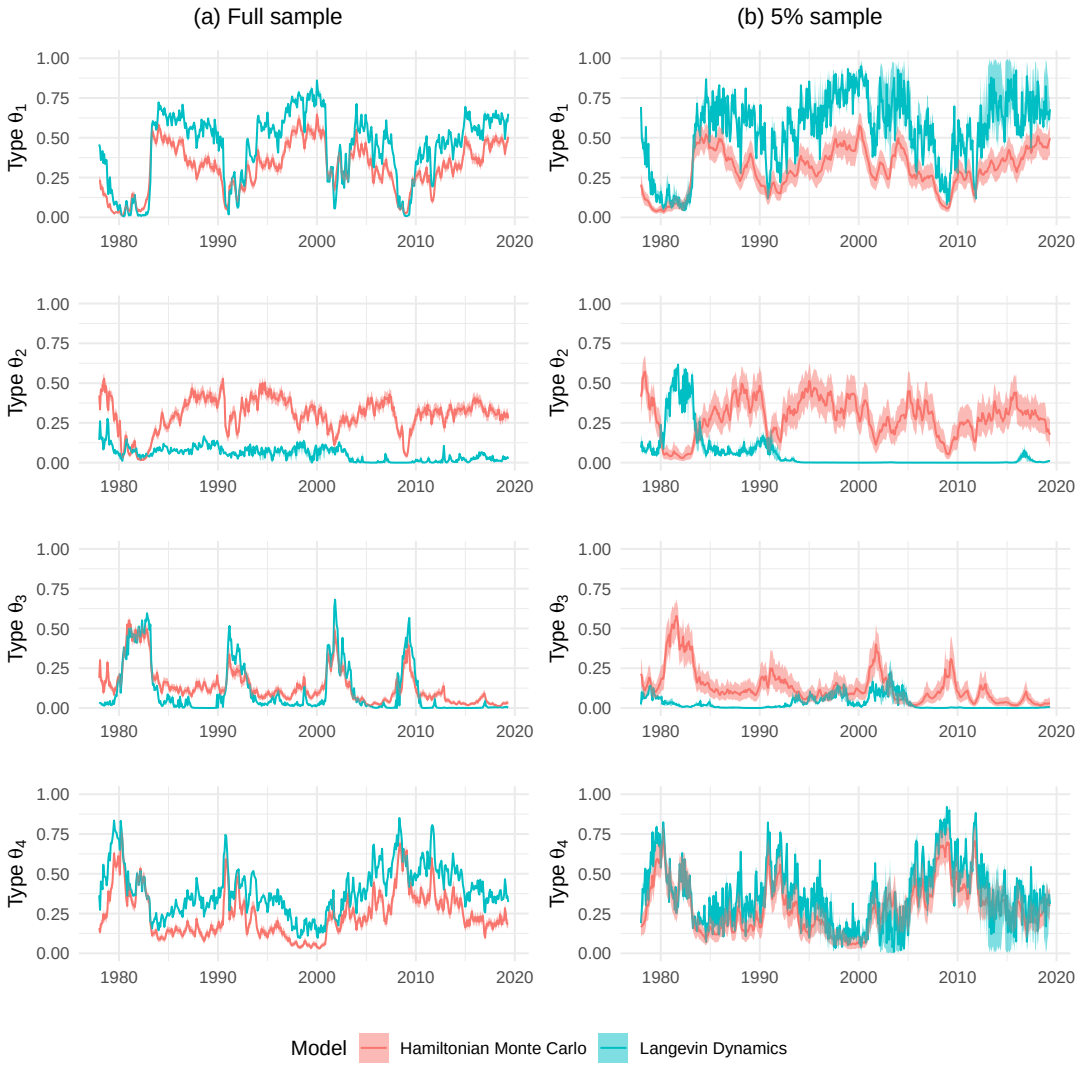


Figure 6: Michigan Data: Comparison of Belief Type Time Series

Note: This figure plots the posterior mean estimates of $\theta_{k,t}$ from model (DSR) estimated on Michigan Consumer Survey data from 1978 through 2019 using Langevin dynamics and Hamiltonian Monte Carlo, respectively. Around the posterior means, we shade 95% credible intervals.

Table 2 displays the same diagnostic statistics we report in table 1 but for estimation on the consumer survey data. We observe similar patterns whereby HMC explores the posterior much more widely than LD, whose convergence behavior on real data if anything looks worse than in the simulated data.

Table 2: Michigan Data: Convergence diagnostics for type probability variables, $\theta_{k,t}$

		Mean	Quantile			Frac > 1.1
			0.05	0.5	0.95	
Effective Sample Size	HMC	1543.17	847.76	1561.23	2225.98	
	LD	2.98	1.3	1.32	12.82	
Gelman-Rubin $\hat{R}$	HMC	1	1	1	1.01	0
	LD	1.88	1.08	2.09	2.12	0.94

Note: We record draws for type probabilities $\theta_{k,t}$ in model (DSR) estimated on Michigan Consumer Survey data from 1978 through 2019 from Markov chains that evolve according to Hamiltonian Monte Carlo and Langevin dynamics, respectively. Gelman-Rubin $\hat{R}$ is a measure of convergence; the closer its value is to 1, the stronger the evidence for convergence. Values larger than 1.1 typically mean the Markov chain has not converged. Effective sample size accounts for autocorrelation in Markov chains and is the number of effectively independent draws from the posterior distribution. HMC was run for 1000 post-warmup iterations, LD was run for 3000 post-warmup iterations, identical to reported in Munro and Ng (2020).

Finally, to investigate how the algorithms perform on differently-sized date, we reestimate the model on a random sample of 5% observations. The resulting time series are presented in Figure 6(b). We observe that HMC finds very similar mean posterior values of $\theta_{k,t}$ based on the full sample and the 5% subsample and the latter indicates substantially more uncertainty in the estimates, as expected. On the other hand, LD appears to have failed to converge. Most notably, with 5% sample LD has not been able to find the periodic behaviour present in $\theta_{3,t}$.

The findings on the Michigan consumer survey data reinforce the main message of the simulations that full Hamiltonian dynamics provide major improvements in inference quality.

5 HMC for Supervised Topic Model

So far we have shown that HMC compares favorably with other approaches for estimating existing latent variable models (the structural topic model and the dynamic survey response model). In our final application, we show that HMC also allows a researcher to quickly develop and prototype new models that capture dependencies of interest while also conducting valid hypothesis testing. By reducing the burden of developing inference algorithms, HMC helps to shift focus from computational difficulties to modeling choices.

The data we use to develop new models is on CEO time use and comes from Bandiera et al. (2020). A cross section of 916 CEOs participated in a survey that recorded features of time use in each 15-minute interval of a given week, e.g. Monday 8am-8:15am, Monday 8:15am-8:30am, and so forth. The recorded categories are 1) the type of activity (meeting, public event, etc.); 2) duration of activity (15m, 30m, etc.); 3) whether the activity is planned or unplanned; 4) the number of participants in the activity; 5) the functions of the participants in the activity (HR, finance, suppliers, etc.). In total there are 654 unique combinations of these features observed in the data. Similarly to text data, we can denote $x_{d,v}$ as the number of times feature combination v appears in the time use diary of CEO d. On average a CEO is engaged in 88.4 activities, with a minimum of 2 and a maximum of 222.

Bandiera et al. (2020) use LDA with $K = 2$ dimensions to organize the time use data.[20] The authors refer to these dimensions as *pure behaviors*, and each one gives a separate distribution over time use combinations $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$. $\theta_{d,1}$ is called the *CEO index* and is the tendency of CEO d to draw his or her time use according to pure behavior 1. The goal is to describe salient differences in executive time use, and relate these to firm and CEO characteristics as well as firm outcomes. The main regression specification takes the form

$$y_d = \gamma\theta_{d,1} + \boldsymbol{q}_d^T\boldsymbol{\zeta} + \epsilon_d \tag{10}$$

where y_d is the log of firm d sales, $\boldsymbol{q}_d$ is a vector of firm observables (e.g. labor inputs and sector of activity), and $\theta_{d,1}$ is the CEO index. The authors first estimate LDA on the time use data using the collapsed Gibbs sampler of Griffiths and Steyvers (2004), then form an estimate $\hat{\theta}_{d,1}$ based on the posterior mean. They then use this an input into the productivity regression (10).

As mentioned in Section 3, such use of inferred objects in regression is statistically problematic. In addition to the problems outlined there (inefficiency, incorrect standard errors, failure of the IID assumption), in this example the estimated $\hat{\gamma}$ is likely to be biased towards 0 due to the measurement error in $\hat{\theta}_{d,1}$. Notice, the variance of the measurement error $\varepsilon_d = (\theta_{d,1} - \hat{\theta}_{d,1})$ is going to be larger for the observations for which the CEO was involved in fewer activities, and for which these activities have similar probabilities under both pure behaviors. The integrated model presented below will reduce the bias resulting from measurement error by effectively weighting more heavily the observations whose measurement error is likely to be smaller, as indicated by the posterior density[21].

[20]The reason off-the-shelf LDA can be used in this survey data but not the survey data in the previous section is that each CEO effectively answers the same survey multiple consecutive times, i.e. in each fifteen-minute unit of time in a week. This allows one to model a distribution over combinations of question responses rather than each question separately.

[21]Intuitively, OLS regression applies a penalty equal to squared deviation between fitted value and data. Meanwhile, in our model the penalty depends on the difference in posterior density between the

To address these issues we propose to reformulate the two-step procedure (first estimating LDA and then running OLS regression) as a single, fully-specified Bayesian model. The model builds on the Supervised Topic Model (Blei and McAuliffe 2010) by adding covariates and prior distributions on regression coefficients. We believe this model is novel in the economics literature[22] and we call it Supervised LDA. The model is described by the data generating process in (S-LDA). A plate diagram representation is presented in appendix figure 10.

$$
\begin{aligned}
&\boldsymbol{\beta}_k \sim \text{Dirichlet}(\eta) && && \boldsymbol{\chi} \sim \text{Normal}(\mathbf{0}, \text{I}\sigma^{\chi}) \\
&\gamma_{0,k} \sim \text{Normal}(0, \sigma_0^{\gamma}) && && \boldsymbol{\zeta} \sim \text{Normal}(\mathbf{0}, \text{I}\sigma^{\zeta}) \\
&\boldsymbol{\theta}_d \sim \text{LogisticNormal}(\boldsymbol{\gamma}_0, \text{I}\sigma^{\theta}) && && \sigma_y \sim \text{Gamma}(s_0, s_1) \\
&\boldsymbol{x}_d \sim \text{Multinomial}\left(\sum_k \boldsymbol{\beta}_k \theta_{d,k}, N_d\right) && && y_d \sim \text{Normal}(\boldsymbol{\theta}_d^T \boldsymbol{\chi} + \boldsymbol{q}_d^T \boldsymbol{\zeta}, \sigma_y^2)
\end{aligned}
\tag{S-LDA}
$$

In essence, the model combines a topic model—described by the first column of equations in (S-LDA)—with a standard Bayesian regression with normal priors on the coefficients—described by the second column. While this formulation admits the general case with an arbitrary K, we use $K = 2$ below in line with Bandiera et al. (2020).

To estimate the model we standardize all numerical variables, set the standard deviations of the Normal priors to $\sigma_0^{\gamma} = \sigma^{\theta} = \sigma^{\chi} = \sigma^{\zeta} = 2$, use a symmetric Dirichlet concentration parameter $\eta = 1$, and set the parameters of the Gamma prior on regression errors to $s_0 = 20$ and $s_1 = 0.5$. We use log of firm sales as the regression outcome y_d, and log of employment and year and country dummies as covariates $\mathbf{q}_d$.

Turning to results, we begin by analyzing the probability distributions over activities, $\boldsymbol{\beta}_k$. To do so, we use the relative probability of various subsets of activities V^a—for example all the activities involving meeting suppliers—between pure behaviors 1 and 2, $\frac{\sum_{v \in V^a} \hat{\beta}_{1,a}}{\sum_{v \in V^a} \hat{\beta}_{2,a}}$.[23] Column (1) of table 3 reproduces these ratios from the original paper while column (2) presents the results from (S-LDA). Despite the modeling differences (including using a logistic normal distribution in place of a Dirichlet prior for $\boldsymbol{\theta}_d$) they are broadly similar. In both cases we see that a CEO that primarily exhibits pure behavior 1 conducts relatively fewer plant visits, meetings with supplier and production-related activities and relatively more meetings with C-suite executives. Bandiera et al. (2020) dubbed this behavior *leadership*, as opposed to *management*.

The main finding of Bandiera et al. (2020) was that higher inferred $\hat{\theta}_{d,1}$ is associated

mode of $\theta_{d,1}$ and the fitted value. The observations whose posterior densities of $\theta_{d,1}$ are flatter will then have less weight in determining the slope of the regression line.

[22] We were made aware that a related model is concurrently studied in the computer science literature by Ahrens et al. (2021)

[23] For details see Bandiera et al. (2020)

Activity	Bandiera et al (2020)	S-LDA	SS-LDA
	(1)	(2)	(3)
Plant Visits	0.11	0.11	0.11
Suppliers	0.32	0.43	0.34
Production	0.46	0.39	0.40
Just Outsiders	0.58	1.27	0.68
Communication	1.49	1.30	1.36
Multi-Function	1.90	1.20	1.43
Insiders and Outsiders	1.90	1.85	2.02
C-suite	33.90	11.67	19.36

Table 3: Relative probability of observing certain activities between Pure Behavior 1 and Pure Behaviour 2. The value of 1 indicates that this activity is equally likely under both Pure Behaviours. Values higher than 1 mean that this type of activity is more likely to be performed under Pure Behavior 1.

with better firm performance measured by sales and profits. In other words, the companies where CEOs are primarily *leaders* perform better. Column (1) of table 4 reports the coefficients from the regression part of S-LDA model. Consistent with the findings in the original paper we find that $\theta_{d,1}$ is positively associated with higher sales controlling for firm's employment, year and country. The magnitude is economically large and the estimate is statistically significant in a Bayesian sense.

5.1 Extension: Structural Supervised-LDA

In addition to showing that low-dimensional representations of CEO behavior are strongly associated with firm performance, Bandiera et al. (2020) also analyzed which CEO and firm characteristics are associated with behaviors. This question is investigated using OLS regressions of the form:

$$\hat{\theta}_{d,1} = \boldsymbol{g}_d^T \boldsymbol{\gamma} + \epsilon_d,$$

where $\boldsymbol{g}_d$ is a vector of CEO d characteristics and the associated firm. Noticeably, this question is very closely related to the problem studied in section 3 of this paper: what is the impact of covariates on the propensity to use certain topics.[24] As a dramatic example of flexibility of HMC we will now extend our S-SLDA by incorporating elements of STM, thereby creating Structural Supervised LDA (SS-LDA).

The data generating process assumed in this model is identical to S-LDA except for

[24] An important distinction between OLS regression of topic shares on covariates and STM in that in the latter the "dependent" variables are the "unnormalized" topics $\tilde{\theta}_d \in \mathbf{R}$. As in logistic regression, in the STM the marginal effect of covariates on topic shares is non-linear.

Table 4: Model Coefficients from S-SLDA and SS-SLDA applied to CEO data.

	Dependent Variable:		
	Log(sales)		Un-normalized CEO Index
	S-LDA	SS-LDA	SS-LDA
	(1)	(2)	(3)
CEO Index, $\theta_{d,1}$	0.282	0.317	
	(0.119, 0.417)	(0.178, 0.488)	
Log Employment	0.945	0.95	0.438
	(0.902, 1.008)	(0.911, 0.985)	(0.38, 0.499)
MBA			0.346
			(0.21, 0.471)
Family CEO			-0.728
			(-0.85, -0.595)
Public Firm			-0.986
			(-1.172, -0.819)
MNE			1.081
			(0.927, 1.265)
Controls	X	X	X

Note: The point estimate is the mean posterior value of the coefficients. In parenthesis we report 95% (symmetric) Bayesian credible intervals. The Un-normalized CEO index is a real valued variable. In order to obtain the CEO Index from the Un-normalized CEO Index one needs to apply the softmax transform. MBA is a dummy indicating if a CEO has MBA degree, Family CEO takes value one if the firm is owned and managed by founding family. Public Firm take value of one if the firm is listed. MNE takes value of one if the firm is a multinational enterprise.

the fact that the CEO type is assumed to be distributed as

$$\boldsymbol{\theta}_d \sim \text{LogisticNormal}\left[(\boldsymbol{g}_d^T\boldsymbol{\gamma}_1, \ldots, \boldsymbol{g}_d^T\boldsymbol{\gamma}_K)^T, \text{I}\sigma^\theta\right]$$

and coefficients in this model are given the prior $\boldsymbol{\gamma}_k \sim \text{Normal}(\mathbf{0}, \text{I}\sigma^\gamma)$. Importantly, if $\boldsymbol{g}_d$ is a constant, SS-LDA nests S-LDA. The model code in appendix B.4 reflects it—moving from S-LDA to SS-LDA requires modifying just two lines of code.

We estimate this model with hyperparameters identical to the ones described above, with the addition of setting $\sigma^\gamma = 2$; as before we first standardize all of the numeric variables. The covariates $\boldsymbol{g}_d$ we select are: firm employment, whether or not the CEO has an MBA, whether or not a firm is owned and managed by a founding family, whether or not the firm is a multinational enterprise, and whether it is listed. Note that the variables that explain $\boldsymbol{\theta}_d$ need not coincide with those that explain y_d.

Turning to results of SS-LDA, comparing Columns (2) and (3) in Table 3 we see that with the addition of $\boldsymbol{g_d}$, the relative probabilities of different activities change somewhat, typically in the direction of increasing the differences between the pure behaviors (the

ratios are further away from 1). Looking at the effect of CEO behavior on sales in Column (2) of table 4, we continue to find a strong positive association. Finally, looking at the relationship between firm characteristics and the unnormalized, real-valued CEO index[25] we observe that larger employment and holding an MBA degree increase the CEO index, i.e. make it more likely for a CEO to be a *leader*. On the other hand, CEOs of family firms have lower values of the index, i.e. they are more likely to be *managers*. Perhaps surprisingly, the effects of being a public firm and being an MNE go in opposite directions, where the latter increases the CEO index.

Altogether, this section shows that HMC gives a researcher the ability to analyze questions of interest in an intellectually consistent framework, without relying on using moments of an inferred posterior distribution as data and appealing to the asymptotic theory of OLS. The approach we propose fully accounts for uncertainty in the inferred latent variables such as $\boldsymbol{\theta}_d$, which not only results in correct Bayesian credible regions (as opposed to incorrect OLS asymptotic confidence intervals), but also increases efficiency and reduces bias. Given the relative simplicity of adopting this methodology in NumPyro we believe applied researchers would greatly benefit from it.

6 Conclusion

In this paper, we have shown how Hamiltonian Monte Carlo deployed with efficient algorithms for automatic differentiation can be used to sample from complex posterior distributions of the type that arise in unstructured data analysis. Our results suggest that this framework is beneficial whenever a researcher wishes to jointly specify a latent variable model for dimensionality reduction together with a regression model involving those latent representations. As this encompasses the large majority of applications of latent variable models for unstructured data in economics, we believe our findings to be of broad interest. Applied researchers can now model problem-specific dependencies and conduct valid inference without resorting to deriving and coding complex algorithms. We therefore expect Hamiltonian Monte Carlo to become a core part of unstructured data analysis going forward.

It is also important to acknowledge the limitations of HMC for unstructured data modeling. One constraint is scalability. The applications we have explored in this paper do not involve vast amounts of data, nor do many in the literature. For those that do, however, HMC-based inference is likely to be infeasible due to computational burdens. Historically variational inference algorithms have been used for posterior approximation

[25]Recall, $\boldsymbol{\theta_d}$ is a point on the simplex, $\boldsymbol{\theta_d} = \text{Softmax}(\tilde{\boldsymbol{\theta}}_d)$. As before, we normalize $\tilde{\theta}_{d,K} = 0$ since the K-dimensional simplex has $K-1$ degrees of freedom. The unnormalized CEO index is then given by $\tilde{\theta}_{d,1}$.

with large data, and these too can be formulated as probabilistic programs that rely on automatic differentiation (Hoffman et al. 2013). More recently, scalable extensions of HMC based on stochastic approximations of gradients have shown initial promise (Dang et al. 2019). NumPyro supports these other approaches, and we leave for future research the question of which inference procedures best suit which big-data problems. In any case, we remain confident that formulating these procedures as probabilistic programs will be key to their widespread adoption.

Identification in LDA and related models is also an area of active research, and the choice of priors over the categorical distributions can affect inference even asymptotically (Ke et al. 2021). These issues relate more to the structure of the model than to a particular approach to inference. Moreover, by allowing researchers to focus more on modeling and less on the implementation of algorithms, the methods we introduce can also help empirically assess the impact of different prior choices on parameter estimation.

Finally, at a high level, HMC is an algorithm for sampling from joint distributions. Since structural models in economics are usually formulated as joint distributions over model parameters and data, HMC can be used for efficient structural estimation. Blei et al. (2021) integrates a latent variable model for text with a structural model of learning to recover agents' beliefs, and uses HMC for parameter estimation. HMC as a means for incorporating unstructured data into structural models is an exciting future prospect.

References

Adams, R. B., Ragunathan, V., and Tumarkin, R. (2021). Death by Committee? An Analysis of Corporate Board (Sub-) Committees. *Journal of Financial Economics*, forthcoming.

Ahrens, M., Ashwin, J., Calliess, J.-P., and Nguyen, V. (2021). Bayesian Topic Regression for Causal Inference. *unpublished manuscript*.

Bandiera, O., Prat, A., Hansen, S., and Sadun, R. (2020). CEO Behavior and Firm Performance. *Journal of Political Economy*, 128(4):1325–1369.

Bertsch, C., Hull, I., and Zhang, X. (2021). Narrative fragmentation and the business cycle. *Economics Letters*, 201:109783.

Betancourt, M. (2018). A Conceptual Introduction to Hamiltonian Monte Carlo. *arXiv:1701.02434 [stat]*.

Bingham, E., Chen, J. P., Jankowiak, M., Obermeyer, F., Pradhan, N., Karaletsos, T., Singh, R., Szerlip, P., Horsfall, P., and Goodman, N. D. (2018). Pyro: Deep Universal Probabilistic Programming. *arXiv:1810.09538 [cs, stat]*.

Blei, D. M., Hansen, S., Prat, A., and Sacher, S. (2021). A Structural Model for Estimating Beliefs from Textual Panel Data. Unpublished manuscript.

Blei, D. M. and Lafferty, J. D. (2006). Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 113–120, New York, NY, USA. Association for Computing Machinery.

Blei, D. M. and McAuliffe, J. D. (2010). Supervised Topic Models. *arXiv:1003.0783 [stat]*.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3(null):993–1022.

Bybee, L., Kelly, B. T., Manela, A., and Xiu, D. (2020). The Structure of Economic News. Technical Report w26648, National Bureau of Economic Research.

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., and Riddell, A. (2017). Stan: A Probabilistic Programming Language. *Journal of Statistical Software*, 76(1):1–32.

Conde-Ruiz, J. I., Ganuza, J.-J., García, M., and Puch, L. A. (2021). Gender Distribution across Topics in Top 5 Economics Journals: A Machine Learning Approach. Technical Report 1241, Barcelona Graduate School of Economics.

Dang, K.-D., Quiroz, M., Kohn, R., Tran, M.-N., and Villani, M. (2019). Hamiltonian Monte Carlo with Energy Conserving Subsampling. *Journal of Machine Learning Research*, 20(100):1–31.

Dieijen, M. and Lumsdaine, R. L. (2019). What Say They About Their Mandate? A Textual Assessment of Federal Reserve Speeches. SSRN Scholarly Paper ID 3455456, Social Science Research Network, Rochester, NY.

Draca, M. and Schwarz, C. (2021). How Polarized are Citizens? Measuring Ideology from the Ground-Up. SSRN Scholarly Paper ID 3154431, Social Science Research Network, Rochester, NY.

Ellingsen, J., Larsen, V., and Thorsrud, L. A. (2021). News Media vs. FRED-MD for Macroeconomic Forecasting. *Journal of Applied Econometrics*, forthcoming.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian Data Analysis*. Chapman and Hall/CRC, Boca Raton, 3rd edition edition.

Gelman, A. and Rubin, D. B. (1992). Inference from Iterative Simulation Using Multiple Sequences. *Statistical Science*, 7(4):457–472.

Gentzkow, M., Kelly, B., and Taddy, M. (2019). Text as Data. *Journal of Economic Literature*, 57(3):535–574.

Griffiths, T. L. and Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5228–5235.

Hanley, K. W. and Hoberg, G. (2019). Dynamic Interpretation of Emerging Risks in the Financial Sector. *The Review of Financial Studies*, 32(12):4543–4603.

Hansen, S. and McMahon, M. (2016). Shocking language: Understanding the macroeconomic effects of central bank communication. *Journal of International Economics*, 99:S114–S133.

Hansen, S., McMahon, M., and Prat, A. (2018). Transparency and Deliberation Within the FOMC: A Computational Linguistics Approach*. *The Quarterly Journal of Economics*, 133(2):801–870.

Hansen, S., McMahon, M., and Tong, M. (2019). The long-run information effect of central bank communication. *Journal of Monetary Economics*, 108:185–202.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). Stochastic Variational Inference. *Journal of Machine Learning Research*, 14(4):1303–1347.

Hoffman, M. D. and Gelman, A. (2014). The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(47):1593–1623.

Hofmann, T. (2017). Probabilistic Latent Semantic Indexing. *ACM SIGIR Forum*, 51(2):211–218.

Ke, S., Olea, J. L. M., and Nesbit, J. (2021). Robust Machine Learning Algorithms for Text Analysis. Unpublished manuscript.

Larsen, V. H. and Thorsrud, L. A. (2019). The value of news for economic developments. *Journal of Econometrics*, 210(1):203–218.

Lopez Lira, A. (2019). Risk Factors That Matter: Textual Analysis of Risk Disclosures for the Cross-Section of Returns. *SSRN Electronic Journal*.

MacKay, D. J. C. (2003). *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, Cambridge, UK ; New York, illustrated edition edition.

Meager, R. (2019). Understanding the Average Impact of Microcredit Expansions: A Bayesian Hierarchical Analysis of Seven Randomized Experiments. *American Economic Journal: Applied Economics*, 11(1):57–91.

Mueller, H. and Rauh, C. (2018). Reading Between the Lines: Prediction of Political Violence Using Newspaper Text. *American Political Science Review*, 112(2):358–375.

Munro, E. and Ng, S. (2020). Latent Dirichlet Analysis of Categorical Survey Responses. *Journal of Business & Economic Statistics*, pages 1–16.

Neal, R. M. (2012). MCMC using Hamiltonian dynamics. *arXiv:1206.1901 [physics, stat]*.

Nimark, K. P. and Pitschner, S. (2019). News media and delegated information choice. *Journal of Economic Theory*, 181:160–196.

Nimczik, J. S. (2017). Job Mobility Networks and Endogenous Labor Markets. Technical Report 168147, Verein für Socialpolitik / German Economic Association.

Olivella, S., Pratt, T., and Imai, K. (2021). Dynamic Stochastic Blockmodel Regression for Network Data: Application to International Militarized Conflicts. *arXiv:2103.00702 [cs, stat]*.

Phan, D., Pradhan, N., and Jankowiak, M. (2019). Composable Effects for Flexible and Accelerated Probabilistic Programming in NumPyro. *arXiv:1912.11554 [cs, stat]*.

Roberts, M. E., Stewart, B. M., and Airoldi, E. M. (2016). A Model of Text for Experimentation in the Social Sciences. *Journal of the American Statistical Association*, 111(515):988–1003.

Roberts, M. E., Stewart, B. M., and Tingley, D. (2019). Stm: An R Package for Structural Topic Models. *Journal of Statistical Software*, 91(2).

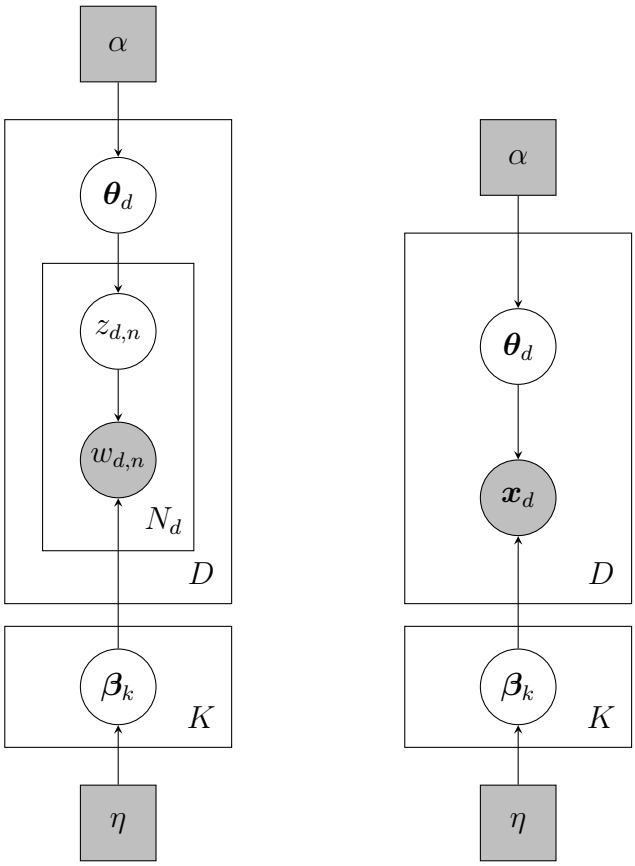
Roberts, M. E., Stewart, B. M., Tingley, D., Lucas, C., Leder-Luis, J., Gadarian, S. K., Albertson, B., and Rand, D. G. (2014). Structural Topic Models for Open-Ended Survey Responses. *American Journal of Political Science*, 58(4):1064–1082.

Stiglitz, E. and Caspi, A. (2020). Observability and Reasoned Discourse: Evidence from the U.S. Senate. SSRN Scholarly Paper ID 3627564, Social Science Research Network, Rochester, NY.

Thorsrud, L. A. (2020). Words are the New Numbers: A Newsy Coincident Index of the Business Cycle. *Journal of Business & Economic Statistics*, 38(2):393–409.

Welling, M. and Teh, Y. W. (2011). Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pages 681–688, Madison, WI, USA. Omnipress.

A Plate Diagrams



(a) With Latent Topic Assignments

(b) No Latent Topic Assignments

Figure 7: Plate Diagram for Latent Dirichlet Allocation

Note: Hyperparameters are denoted with shaded rectangles; random variables with unshaded circles; and data with shaded circles. Random variables that lie within a given plate are exchangeable. In the model on the left, discrete topic assignments are drawn per word $z_{d,n}$. In the model on the right, these assignments have been marginalized out of the model so that all random variables are continuous. $\boldsymbol{x}_d$ is a vector of document d word counts (a length V vector, where V is the number of vocabulary terms).

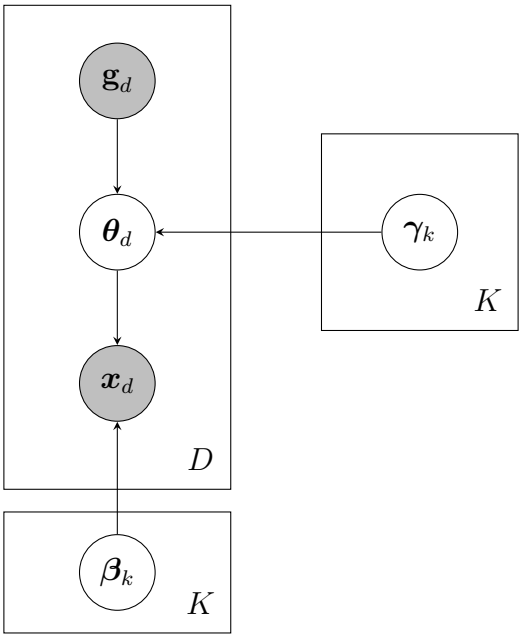


Figure 8: Plate Diagram for the Structural Topic Model

Note: In the structural topic model, document-level covariates $\boldsymbol{g}_d$ affect the prevalence of topics in documents. They combine with regression coefficients $\boldsymbol{\gamma}_k$ to generate topic shares $\boldsymbol{\theta}_d$ via a logistic normal prior distribution. The term-topic distributions $\boldsymbol{\beta}_k$ are drawn from Dirichlet distributions as in plain LDA. Here we omit prior hyperparameters from the plate diagram.

Figure 9: Plate Diagram for the Dynamic LDA.

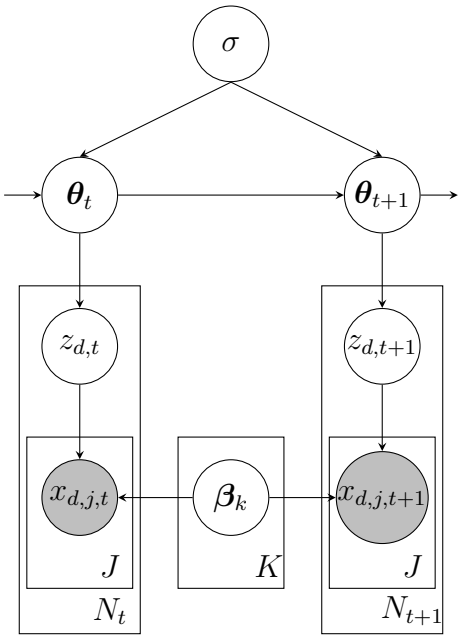


Figure 10: Plate Diagram for the Supervised LDA.

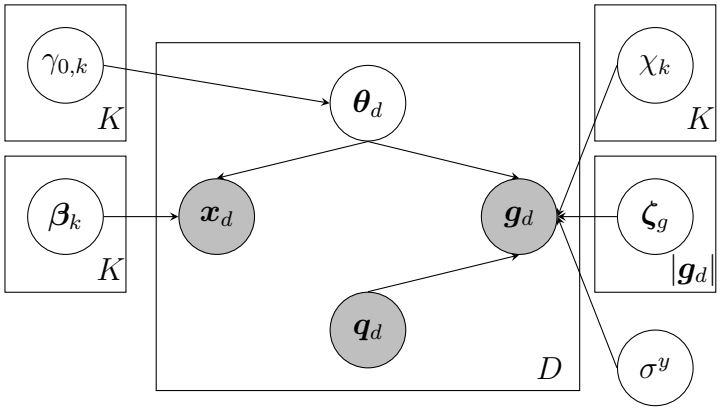
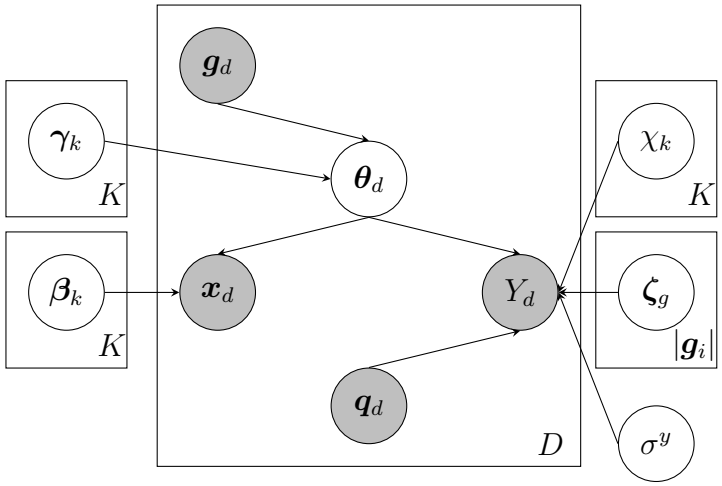


Figure 11: Plate Diagram for the Structural Supervised LDA.



B Numpyro Model Codes

B.1 Plain LDA

```
def lda(X, K, alpha, eta):
    # X: document-word matrix of BoWs
    # K: number of topics
    # alpha: Dirichlet hyperparameter for topic prevalence
    # eta: Dirichlet hyperparameter for topic concentration

    D, V = jnp.shape(X)
    N = X.sum(axis = 1)

    # document-topic distributions
    with plate("docs", D):
        theta = sample("theta", dist.Dirichlet(alpha*jnp.ones([K])))

    # topic-word distributions
    with plate("topics", K):
        beta = sample("beta", dist.Dirichlet((eta) * jnp.ones([V])))

    # likelihood
    distMultinomial = dist.Multinomial(total_count = N,
                      probs = jnp.matmul(theta, beta))
    with plate("hist", D):
        sample("obs", distMultinomial, obs = X)
```

B.2 Structural Topic Model

```python
import jax.numpy as jnp
from jax.nn import softmax
from numpyro import sample, plate, deterministic
import numpyro.distributions as dist

def stm(X, G, K, eta, norm_topic=0, sigma_gamma=2, sigma_theta=1):
    # X: document-word matrix of BoWs
    # G: matrix of covariates entering topic selection
    # K: number of topics
    # eta: Dirichlet hyperparameter for topic concentration
    # norm_topic: index for topic to be normalized to zero
    # sigma_gamma: variance for gaussian prior on topic prevalence coefficients
    # sigma_theta: variance for logistic-normal prior on topic prevalence

    try:
        D, M = G.shape
    except:
        G = jnp.expand_dims(G, axis=1)
        D, M = G.shape
    V = X.shape[1]
    N = X.sum(axis = 1)

    # document-topic distributions
    with plate("tpcs", K-1):
        with plate("characteristics", M):
            gamma = sample("gamma", dist.Normal(0, sigma_gamma))
        with plate("doc_proportions", D):
            eps = sample("eps",dist.Normal(0,1))
    try:
        _, _ = gamma.shape
    except:
        gamma= jnp.expand_dims(gamma, axis=0)
    A = jnp.matmul(G, gamma) + sigma_theta * eps
    A = jnp.hstack([A[:,0:norm_topic], jnp.zeros([D,1]) ,A[:,norm_topic:]])

    theta = softmax(A, axis = -1)
    theta = deterministic("theta", theta)

    # topic-word distributions
    with plate("topics", K):
        beta = sample("beta", dist.Dirichlet(jnp.ones([V]) * eta))

    # likelihood
    with plate("docs", D):
        sample("obs", dist.Multinomial(total_count = N,
        probs = jnp.matmul(theta, beta)), obs = X)
```

B.3 Dynamic LDA

```
import jax.numpy as jnp
import numpyro.distributions as dist
from jax import nn, jit, vmap

from numpyro import sample, deterministic, factor
from numpyro.contrib.control_flow import scan

def dynamic_lda_logLik(y, beta, phi):
    # y: array of selected answers
    # phi: array of type-probabilities
    # beta: list (of length J) of type-answer probabilities
    J = len(beta)
    logProbs = [dist.CategoricalProbs(beta[j]).log_prob(y[j]) for j in range(J)]
    logProbs = jnp.vstack(logProbs).sum(0) + jnp.log(phi)
    return nn.logsumexp(logProbs)

dynamic_lda_logLik = jit(vmap(logLik, in_axes = (0, None, 0)))

def dynamic_lda(Y, K, T, I, eta, v0 = 10, s0 = 1):
    # Y : survey answers
    # K : number of types (topics)
    # T : number of time periods
    # I : vector of time indices for each observation
    # eta: Dirichlet hyperparamter
    # v0, s0: Inverse gamma hyperparameters

    J = Y.shape[1]

    # variance of errors
    sigma_sq = sample("sigma_sq", dist.InverseGamma(v0, s0).expand([K]))

    # Standard deviation of difference in w errors
    sigma_tilde = jnp.sqrt(sigma_sq[1:K] + sigma_sq[0])

    # Initial differences in (unnormalized) type probabilities
    pi_tilde_0 = sample("pi_tilde_0", dist.Normal(0, 5 * sigma_tilde ))

    def transition(state_prev, i):
        state_cur = sample("pi_tilde", dist.Normal(state_prev, sigma_tilde))
        return state_cur, state_cur
    _ , pi_tilde = scan(transition, pi_tilde_0, jnp.ones([T]))

    # Array of time-type probabilities
    pi = nn.softmax(jnp.hstack([jnp.zeros([T,1]), pi_tilde]), axis = 1)
    pi = deterministic("pi", pi)

    # List of length J of type-answer probabilities
    beta = [sample("beta_{}".format(j), dist.Dirichlet(eta[j])) for j in range(J)]

    # Likelihood
    factor("logLik", dynamic_lda_logLik(Y, beta, phi[I]))
```

B.4 Structural Supervised LDA

```
import jax.numpy as jnp
import numpyro.distributions as dist
from numpyro import sample, plate
from jax.nn import softmax


def structural_slda(Y, X, N, Z, Q, K, eta = .1, alpha = 1):
    # Y : regression outcomes
    # X : document-word matrix of BoWs
    # N : total word counts per document
    # Z : matrix of non-text covariates
    # Q : matrix of covariates entering topic selection
    # K : number of topics
    # eta, alpha : Dirichlet hyperparamters

    D, V = X.shape
    z,q = Z.shape[1], Q.shape[1]

    #### LDA part of model

    with plate("topics", K):
        # Topic-word distributions
        beta = sample("beta", dist.Dirichlet(eta * jnp.ones(V)))

    phis = sample("phis", dist.Normal(0,2).expand([q, K-1]))

    with plate("docs", D, dim = -2):
        A = sample("A", dist.Normal(jnp.matmul(Q, phis) , alpha))

    # document-topic distributions
    theta = softmax(jnp.hstack([A, jnp.zeros([D, 1])]), axis = -1)

    distMultinomial = dist.Multinomial(total_count=N, probs = jnp.matmul(theta, beta))
    with plate("hist", D):
        sample("obs_x", distMultinomial, obs = X)

    #### Regression part of model
    gammas = sample("gammas", dist.Normal(0, 2).expand([K-1]))
    zetas = sample("zetas", dist.Normal(0,2).expand([z]))
    sigma = sample("sigma", dist.Exponential(1.))

    mean = jnp.matmul(theta[:,:(K-1)], gammas) + jnp.matmul(Z, zetas)

    with plate("y", D):
        sample("obs_y",  dist.Normal(mean, sigma), obs = Y)
```