A Sparse and Adaptive Prior for Time-Dependent Model Parameters

Dani Yogatama
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
dyogatama@cs.cmu.edu

Bryan R. Routledge
Tepper School of Business
Carnegie Mellon University
Pittsburgh, PA 15213, USA
routledge@cmu.edu

Noah A. Smith
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
nasmith@cs.cmu.edu

Abstract

We consider the scenario where the parameters of a probabilistic model are expected to vary over time. We construct a novel prior distribution that promotes sparsity and adapts the strength of correlation between parameters at successive timesteps, based on the data. We derive approximate variational inference procedures for learning and prediction with this prior. We test the approach on two tasks: forecasting financial quantities from relevant text, and modeling language contingent on time-varying financial measurements.

1 Introduction

When learning from streams of data to make predictions in the future, how should we handle the timestamp associated with each instance? Ignoring timestamps and assuming data are i.i.d. is scalable but risks distracting a model with irrelevant “ancient history.” On the other hand, using only the most recent portion of the data risks overfitting to current trends and missing important time-insensitive effects. In this paper, we seek a general approach to learning model parameters that are overall sparse, but that adapt to variation in how different effects change over time.

Our approach is a prior over parameters of an exponential family (e.g., coefficients in linear or logistic regression). We assume that parameter values shift at each timestep, with correlation between adjacent timesteps captured using a multivariate normal distribution whose precision matrix is restricted to a tridiagonal structure. We (approximately) marginalize the (co)variance parameters of this normal distribution using a Jeffreys prior, resulting in a model that allows smooth variation over time while encouraging overall sparsity in the parameters. (The parameters themselves are not given a fully Bayesian treatment.)

We demonstrate the usefulness of our model on two tasks, showing gains over alternative approaches. The first is a text regression problem in which an economic variable (volatility of returns) is forecast from financial reports [Kogan et al., 2009]. The second forecasts text by constructing a language model that conditions on highly time-dependent economic variables.

Notation is given in §2. Our prior distribution is presented in §3. We draw connections to related work in §4.

§5 presents our inference algorithm and §6 our experimental results.

2 Notation

We assume data of the form \( \{(x_n, y_n)\}_{n=1}^N \), where each \( x_n \) includes a timestamp denoted \( t \in \{1, \ldots, T\} \). The aim is to learn a predictor that maps input \( x_{N+1} \), assumed to be at timestep \( T \), to output \( y_{N+1} \). In the probabilistic setting we adopt here, the prediction is MAP inference over r.v. \( Y \) given \( X=x \) and a model parameterized by \( \beta \in \mathbb{R}^I \). Learning is parameter estimation to solve:

\[
\arg\max_{\beta} \log p(\beta) + \sum_{n=1}^N \log p(y_n | x_n, \beta) \tag{1}
\]

The focus of the paper is on the prior distribution \( p(\beta) \). Throughout, we will denote the task-specific log-likelihood (second term) by \( L(\beta) \) and assume a generalized linear model such that a feature vector function \( f \) maps inputs \( x \) into \( \mathbb{R}^I \) and \( f(x)^T \beta \) is “linked” to the distribution over \( Y \) using, e.g., a logit or identity. We will refer to elements of \( f \) as “features” and to \( \beta \) as “coefficients.” We assume \( T \) discrete timesteps.

\(^1\)In this work we assume timestamps are discretized.
3 Time-Series Prior

Our time-series prior draws inspiration from the probabilistic interpretation of the sparsity-inducing lasso (Tibshirani 1996) and group lasso (Yuan & Lin 2007). In non-overlapping group lasso, features are divided into groups, and the coefficients within each group \( m \) are drawn according to:

1. Variance \( \sigma_m^2 \sim \text{an exponential distribution} \)
2. \( \beta_m \sim \text{Normal}(0, \sigma_m^2 \mathbf{I}) \).

We seek a prior that lets each coefficient vary smoothly over time. A high-level intuition of our prior is that we create copies of \( \beta \), one at each timestep: \( (\beta_1, \beta_2, \ldots, \beta_T) \). For each feature \( i \), let the sequence \( (\beta_i^{(1)}, \beta_i^{(2)}, \ldots, \beta_i^{(T)}) \) form a group, denoted \( \beta_i \). Group lasso does not view coefficients in a group as explicitly correlated; they are independent given the variance parameter. Given the sequential structure of \( \beta_i \), we replace the covariance matrix \( \Sigma \) with the precision matrix \( \Lambda \) to capture autocorrelation. Specifically, we assume the vector \( \beta_i \) is drawn from a multivariate normal distribution with mean zero and a \( T \times T \) precision matrix \( \Lambda \) with the following tridiagonal form:

\[
\Lambda = \frac{1}{\lambda} A = \frac{1}{\lambda} \begin{bmatrix}
1 & \alpha & 0 & 0 & \ldots \\
\alpha & 1 & \alpha & 0 & \ldots \\
0 & \alpha & 1 & \alpha & \ldots \\
0 & 0 & \alpha & 1 & \ldots \\
\vdots & \vdots & \vdots & \vdots & \ddots
\end{bmatrix}
\]

\( \lambda \geq 0 \) is a scalar multiplier whose role is to control sparsity in the coefficients, while \( \alpha \) dictates the degree of correlation between coefficients in adjacent timesteps (autocorrelation). Importantly, \( \alpha \) and \( \lambda \) (and hence \( \Lambda \) and \( \Lambda_i \)) are allowed to be different for each group \( i \).

We need to ensure that \( \Lambda \) is positive definite. Fortunately, it is easy to show that for \( \alpha \in (-0.5, 0.5) \), the resulting \( \Lambda \) is positive definite.

**Proof sketch.** To show this, since \( \Lambda \) is a symmetric matrix, we verify that each of its principal minors have strictly positive determinants. The principal minors of \( \Lambda \) are uniform tridiagonal symmetric matrices, and the determinant of a uniform tridiagonal \( N \times N \) matrix can be written as \( \prod_{n=1}^{N} \left( 1 + 2\alpha \cos \left( \frac{(n+1)\pi}{N+1} \right) \right) \) (see, e.g., Figueiredo 2002).

\( \lambda \) is driven by scalability concerns. Instead of using, e.g., a random draw from a Wishart distribution, we specify the precision matrix to have a tridiagonal structure. This induces dependencies between coefficients in adjacent timesteps (first-order dependencies) and allows the prior to scale to fine-grained timesteps more efficiently. Let \( N \) denote the number of training instances, \( I \) the number of base features, and \( T \) the number of timesteps. A single pass of our variational algorithm (discussed in [3]) has runtime \( \cO(I(N^2) + \text{space requirement } \cO(I(N + T))) \), instead of \( \cO(I(N + T^2)) \) for both if each \( \Lambda_i \) is drawn from a Wishart distribution. This can make a big difference for applications with large numbers of features (I). Additionally, we choose the off-diagonal entries to be uniform, so we only need one \( \alpha_i \) for each base feature. This design choice restricts the expressive power of the prior but still permits flexibility in adapting to trends for different coefficients, as we will see. The prior encourages sparsity at the group level, essentially performing feature selection: some feature coefficients \( \beta_i \) may be driven to zero across all timesteps, while others will be allowed to vary over time, with an expectation of smooth changes.

Note that this model introduces only one hyperparameter, \( \tau \), since we marginalize \( \alpha = \langle \alpha_1, \ldots, \alpha_I \rangle \) and \( \lambda = \langle \lambda_1, \ldots, \lambda_I \rangle \).

3.1 Generative Model

Our generative model for the group of coefficients \( \beta_i = (\beta_i^{(1)}, \beta_i^{(2)}, \ldots, \beta_i^{(T)}) \) is given by:

1. \( \lambda_i \sim \text{an improper Jeffreys prior } (p(\lambda) \propto \lambda^{-1}) \).
2. \( \alpha_i \sim \text{a truncated exponential prior with parameter } \tau \). This distribution forces \( \alpha_i \) to fall in \((-C, 0]\), so that \( \Lambda_i \) is p.d. and autocorrelations are always positive:

\[
p(\alpha \mid \tau) = \tau \exp(-\tau(\alpha + C)) \mathbb{1}_{-C < \alpha \leq 0} / \left(1 - \exp(-\tau C)\right)
\]

3. \( \beta_i \sim \text{Normal}(0, \Lambda_i^{-1}) \), with the precision matrix \( \Lambda_i \) as defined in Eq. 2.

During estimation of \( \beta \), each \( \lambda_i \) and \( \alpha_i \) are marginalized, giving a sparse and adaptive estimate for \( \beta \).

3.2 Scalability
4 Related Work

Our model is related to autoregressive integrated moving average approaches to time-series data [Box et al. 2008], but we never have access to direct observations of the time-series. Instead, we observe data (x and y) assumed to have been sampled using time-series-generated variables as coefficients (β). During learning, we therefore use probabilistic inference to reason about the variables at all timesteps together. In [4], we describe a scalable variational inference algorithm for inferring coefficients at all timesteps, enabling prediction of future data and inspection of trends.

We follow Yogatama et al. (2011) in creating time-specific copies of the base coefficients, so that \( \beta = (\beta^{(1)}, \beta^{(2)}, \ldots, \beta^{(T)}) \). As a prior over \( \beta \), they used a multivariate Gaussian imposing non-zero covariance between each \( \beta^{(t)}_i \) and its time-adjacent copies \( \beta^{(t-1)}_i \) and \( \beta^{(t+1)}_i \). The strength of that covariance was set for each base feature by a global hyperparameter, which was tuned on held-out development data along with the global variance hyperparameter. Yogatama et al.’s model can be obtained from ours by fixing the same \( \alpha \) and \( \lambda \) for all features \( i \). Our approach differs in that (i) we marginalize the hyperparameters, (ii) we allow each coefficient its own autocorrelation, and (iii) we encourage sparsity.

There are many related Bayesian approaches for time-varying model parameters [Belmonte et al. 2012; Nakajima & West 2012; Caron et al. 2012], as well as work on time-varying signal estimation [Angelosante & Giannakis 2009; Angelosante et al. 2009; Charles & Rozell 2012]. Each provides a different probabilistic interpretation of parameter generation. Our model has a distinctive generative story in that correlations between parameters of successive timesteps are encoded in a precision matrix. Additionally, unlike these fully Bayesian approaches that infer full posterior distributions, we only obtain posterior mode estimates of coefficients, which has computational advantages at prediction time (e.g., straightforward MAP inference and sparsity) and interpretability of \( \beta \).

As noted, our grouping together of each feature’s instantiations at all timesteps, \( (\beta^{(1)}_i, \beta^{(2)}_i, \ldots, \beta^{(T)}_i) \) and seeking sparsity, bears clear similarity to group lasso [Yuan & Lin 2007], which encourages whole groups of coefficients to collectively go to zero. A probabilistic interpretation for lasso as a two level exponential-normal distribution that generalizes to (non-overlapping) group lasso was introduced by Figueiredo (2002). He also showed that the exponential distribution prior can be replaced with an improper Jeffreys prior for a parameter-free model, a step we follow as well.

Our model is also related to the fused lasso [Tibshirani et al. 2005], which penalizes a loss function by the \( \ell_1 \)-norm of the coefficients and their differences. Our prior has a more clear probabilistic interpretation and adapts the degree of autocorrelation for each coefficient, based on the data.

Zhang & Yeung (2010) proposed a regularization method using a matrix-variate normal distribution prior to model task relationships in multitask learning. If we consider timesteps as tasks, the technique resembles our regularizer. Their method jointly optimizes the covariance matrix with the feature coefficients; we choose a Bayesian treatment and encode our prior belief to the (inverse) covariance matrix, while still allowing the learned feature coefficients to modify the matrix by posterior inference. As a result, our method allows different base features to have different matrices.

5 Learning and Inference

We marginalize \( \lambda \) and \( \alpha \) and obtain a maximum a posteriori estimate for \( \beta \), which includes a coefficient for each base feature \( i \) at each timestep \( t \). Specifically, we seek to maximize:

\[
L(\beta) + \sum_{i=1}^{l} \log \int d\alpha_i \int d\lambda_i p(\beta_i | \alpha_i, \lambda_i) p(\alpha_i | \tau) p(\lambda_i)
\]

(4)

Exact inference in this model is intractable. We use mean-field variational inference to derive a lower bound on the above log-likelihood function. We then apply a standard optimization technique to jointly optimize the variational parameters and the coefficients \( \beta \).

We introduce fully factored variational distributions for each \( \lambda_i \) and \( \alpha_i \). For \( \lambda_i \), we use a Gamma distribution with parameters \( a_i, b_i \) as our variational distribution:

\[
q_i(\lambda_i | a_i, b_i) = \frac{\lambda^{a_i-1} \exp(-\lambda_i/b_i)}{b_i^a \Gamma(a_i)}
\]

Therefore, we have \( \mathbb{E}_{q}[\lambda_i] = a_i b_i, \mathbb{E}_{q}[\lambda_i^{-1}] = ((a_i - 1) b_i)^{-1} \), and \( \mathbb{E}_{q}[\log \lambda_i] = \Psi(a_i) + \log b_i \) (\( \Psi \) is the digamma function).

For \( \alpha_i \), we choose the form of our variational distribution to be the same truncated exponential distribution as its prior, with parameter \( \kappa_i \), denoting this distribu-
This form makes it clear that the penalty depends on \( E \) and \( \alpha \). It also has some similarities to the fused lasso penalty (Tibshirani et al., 2005), which is intended to encourage sparsity in the differences between features coefficients across timesteps. Our prior, on the other hand, encourages smoothness in the differences, with additional sparsity at the feature level.

\[ \text{5.2 Variational Parameters for } \alpha \text{ and } \lambda \]

Recall that the variational distribution for \( \lambda_i \) is a Gamma distribution with parameters \( a_i \) and \( b_i \).

**Precision matrix scalar \( \lambda \).** The first derivative for variational parameters \( \alpha \) is easy to compute:

\[
\frac{\partial B}{\partial \alpha_i} = \left(-\frac{T}{2} - a_i\right) \Psi_1(a_i) + \beta_i^\top E[A_i] \beta_i + 1
\]

where \( \Psi_1 \) is the trigamma function. We can solve for \( b \) in closed form given the other free variables:

\[
b_i = \frac{\beta_i^\top E[A_i] \beta_i}{(a_i - 1)T}
\]

We therefore treat \( b \) as a function of \( a, \kappa, \) and \( \beta \) in optimization.

**Off-diagonal entries \( \alpha \).** First, notice that using Jensen’s inequality: \( \mathbb{E}[\log \det A_i^{-1}] = \mathbb{E}[-\log \det A_i] \geq -\log \det E[A_i] \) due to the fact that \( -\log \det A_i \) is a convex function. Furthermore, for a uniform symmetric tridiagonal matrix like \( A_i \), the log determinant can be computed in closed form as follows (Volpi, 2003):

\[
\log \det E[A_i] = \log \left( \prod_{i=1}^{T} 1 + 2E[\alpha_i] \cos \left( \frac{(t + 1)\pi}{T + 1} \right) \right)
\]

We therefore maximize a lower bound on \( B \), making use of the above to calculate first derivatives with respect to \( \kappa_i \):

\[
\frac{\partial B}{\partial \kappa_i} = -\tau - \frac{\partial E[\alpha_i]}{\partial \kappa_i} - \frac{1}{\kappa_i} + C + E[\alpha_i] + \frac{\partial E[\alpha_i]}{\partial \kappa_i} + \frac{C \exp(-C \kappa_i)}{1 - \exp(-C \kappa_i)} - \frac{1}{2} \frac{\partial E[\alpha_i]}{\partial \kappa_i} + \frac{1}{2} \frac{\partial \log \det E[A_i]}{\partial \kappa_i}
\]

The partial derivatives \( \frac{\partial E[\alpha_i]}{\partial \kappa_i}, \frac{\partial \log \det E[A_i]}{\partial \kappa_i}, \) and \( \frac{\partial E[\alpha_i]}{\partial \kappa_i} \) are easy to compute. We omit them for space.

\[ \text{5.3 Implementation Details} \]

A well-known property of numerical optimizers like the one we use (L-BFGS; Liu & Nocedal, 1989) is the failure to reach optimal values exactly at zero. Although theoretically strongly sparse, our prior only produces weak sparsity in practice. Future work might consider
\[
B(a, b, \kappa, \beta) \propto L(\beta) + \sum_{i=1}^{T} \left\{ \frac{1}{2} (-T \mathbb{E}_q[\log \lambda_i] \left[ -\mathbb{E}_q[\log \det A_i^{-1}] \right]) - \mathbb{E}_q[\lambda_i^{-1}] \frac{1}{2} \beta_i^\top \mathbb{E}_q[A_i] \beta_i \right\} \\
+ \sum_{i=1}^{T} \left\{ -(\mathbb{E}_q[\alpha_i] + C) \tau - \mathbb{E}_q[\log \lambda_i] \right\} - \sum_{i=1}^{T} \left\{ (a_i - 1) \mathbb{E}_q[\log \lambda_i] - \frac{\mathbb{E}_q[\lambda_i]}{b_i} - \log \Gamma(a_i) - a_i \log b_i \right\} \\
- \sum_{i=1}^{T} \left\{ \log \kappa_i - \kappa_i (\mathbb{E}_q[\alpha_i] + C) - \log(1 - \exp(-\kappa_i C)) \right\}
\]

Figure 1: The variational bound on Equation 1 that is maximized to learn \( \beta \). The boxed expression is further bounded by \(- \log \det \mathbb{E}_q[A_i]\) using Jensen’s inequality, giving a new lower bound we denote by \( B' \).

a more principled proximal-gradient algorithm to obtain strong sparsity (Bach et al., 2011; Liu & Ye, 2010; Duchi & Singer, 2009).

If we expect feature coefficients at specific timesteps to be sparse as well, it is straightforward to incorporate additional terms in the objective function that encode this prior belief (analogous to an extension from group lasso to sparse group lasso). For the tasks we consider in our experiments, we found that it does not substantially improve the overall performance. Therefore, we keep the simpler bound given in Figure 1.

6 Experiments

We report two sets of experiments, one with a continuous \( y \), the other a language modeling application where \( y \) is text. Each timestep in our experiments is one year.

6.1 Baselines

On both tasks, we compare our approach to a range of baselines. Since this is a forecasting task, at each test year, we only used training examples that come from earlier years. Our baselines vary in how they use this earlier data and in how they regularize.

- **ridge-one**: ridge regression (Hoerl & Kennard, 1970), trained on only examples from the year prior to the test data (e.g., for the 2002 task, train on examples from 2001)
- **ridge-all**: ridge regression trained on the full set of past examples (e.g., for the 2002 task, train on examples from 1996–2001)
- **ridge-ts**: the non-adaptive time-series ridge model of Yogatama et al. (2011)
- **lasso-one**: lasso regression (Tibshirani, 1996), trained on only examples from the year prior to the test data

- **lasso-all**: lasso regression trained on the full set of past examples

In all cases, we tuned hyperparameters on a development data. Note that, of the above baselines, only ridge-ts replicates the coefficients at different timesteps (i.e., \( IT \) parameters); the others have only \( I \) time-insensitive coefficients.

The model with our prior always uses all training examples that are available up to the test year (this is equivalent to a sliding window of size infinity). Like ridge-ts, our model trusts more recent data more, allowing coefficients farther in the past to drift farther away from those most relevant for prediction at time \( T + 1 \). Our model, however, adapts the “drift” of each coefficient separately rather than setting a global hyperparameter.

6.2 Forecasting Risk from Text

In the first experiment, we apply our prior to a forecasting task. We consider the task of predicting volatility of stock returns from financial reports of publicly-traded companies, similar to Kogan et al. (2009).

In finance, volatility refers to a measure of variation in a quantity over time; for stock returns, it is measured using the standard deviation during a fixed period (here, one year). Volatility is used as a measure of financial risk. Consider a linear regression model for predicting the log volatility of a stock from a set of features (see 6.2.1 for a complete description.

\[\text{Brendan O’Connor (personal communication) has established the superiority of the lasso to the support vector regression method of Kogan et al. (2009) on this dataset; lasso is a strong baseline for this problem.}\]

\[\text{Similar to Kogan et al. (2009) and as also the standard practice in finance, we perform a log transformation, since log volatilities are typically close to normally distributed.}\]
Table 1: MSE on the 10-K dataset (various test sets). The first test year (2002) was used as our development data. Our model uses the sparse adaptive prior described in §3. The overall differences between our model and all competing models are statistically significant (Wilcoxon signed-rank test, \( p < 0.01 \)).

| year     | # examples | ridge-one | ridge-all | ridge-ts | lasso-one | lasso-all | our model |
|----------|------------|-----------|-----------|----------|-----------|-----------|-----------|
| 2002(dev)| 2,845      | 0.182     | 0.176     | 0.171    | 0.165     | 0.156     | 0.158     |
| 2003     | 3,611      | 0.185     | 0.173     | 0.171    | 0.164     | 0.176     | 0.164     |
| 2004     | 3,558      | 0.125     | 0.137     | 0.129    | 0.116     | 0.119     | 0.113     |
| 2005     | 3,474      | 0.135     | 0.133     | 0.136    | 0.124     | 0.124     | 0.122     |
| overall  | 13,488     | 0.155     | 0.154     | 0.151    | 0.141     | 0.143     | 0.139     |

of our features). We can interpret a linear regression model probabilistically as drawing \( y \in \mathbb{R} \) from a normal distribution with \( \beta^\top f(x) \) as the mean of the normal. Therefore, in this experiment: 
\[
L(\beta) = -\sum_{t=1}^{T} \sum_{i=1}^{N_i} (y_i^{(t)} - \beta^{(t)} \top f(x_i^{(t)}))^2.
\]

We apply the time-series prior to the feature coefficients \( \beta \). When making a prediction for the test data, we use \( \beta^{(T)} \), the set of feature coefficients for the last timestep in the training data.

6.2.1 Dataset

We used a collection of Securities Exchange Commission-mandated annual reports from 10,492 publicly traded companies in the U.S. There are 27,159 reports over a period of ten years from 1996–2005 in the corpus. These reports are known as “Form 10-K.” For the feature set, we downcased and tokenized the texts and selected the 101st–10,101st most frequent words as binary features. The feature set was kept the same across experiments for all models. It is widely known in the financial community that the past history of volatility of stock returns is a good indicator of the future volatility. Therefore, we also included the log volatility of the stocks twelve months prior to the report as a feature. Our response variable \( y \) is the log volatility of stock returns over a period of twelve months after the report is published.

6.2.2 Results

The first test year (i.e., 2002) was used as our development data for hyperparameter tuning (\( \tau \) was selected to be 1.0). We initialized all the feature coefficients by the coefficients from training a lasso regression on the last year of the training data (lasso-one). Table 1 provides a summary of experimental results. We report the results in mean squared error on the test set: 
\[
\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2,
\]
where \( y_i \) is the true response for instance \( i \) and \( \hat{y}_i \) is the predicted response.

Our model consistently outperformed ridge variants, including the one with a time-series penalty (Yogatama et al., 2011). It also outperformed the lasso variants without any time-series penalty, on average and in three out of four test sets apiece.

One of the major challenges in working with time-series data is to choose the right window size, in which the data is still relevant to current predictions. Our model automates this process with a Bayesian treatment of the strength of each feature coefficient’s autocorrelation. The results indicate that our model was able to learn when to trust a longer history of training data, and when to trust a shorter history of training data, demonstrating the adaptiveness of our prior. Figure 2 shows the distribution of the expected values of the autocorrelation parameters under the variational distributions \( \mathbb{E}_{q[\alpha]}(\alpha) \) for 10,002 features used in our experiments (10,000 unigram features, the previous year log volatility feature, and a bias feature).

In future work, an empirical Bayesian treatment of the hyperprior \( \tau \), fitting it to improve the variational bound, might lead to further improvements.

---

8See Kogan et al. (2009) for a complete description of the dataset; it is available at http://www.ark.cs.cmu.edu/10K.
6.3 Text Modeling in Context

In the second experiment, we consider a hard task of modeling a collection of texts over time conditioned on economic measurements. The goal is to predict the probability of words appearing in a document, based on the “state of the world” at the time the document was authored. Given a set of macroeconomic variables in the U.S. (e.g., unemployment rate, inflation rate, average housing prices, etc.), we want to predict what kind of texts will be produced at a specific timestep. These documents can be written by either the government or publicly-traded companies directly or indirectly affected by the current economic situation.

6.3.1 Model

Our text model is a sparse additive generative model (SAGE; Eisenstein et al. [2011]). In SAGE, there is a background lexical distribution that is perturbed additively in the log-space. When the effects are due to a (sole) feature \(f(x)\), the probability of a word is:

\[
p(w | \theta, \beta, x) = \frac{\exp(\theta_w + \beta_w f(x))}{\sum_{w' \in V} \exp(\theta_{w'} + \beta_{w'} f(x))}
\]

where \(V\) is the vocabulary, \(\theta\) (always observed) is the vector of background log-frequencies of words in the corpus, \(f(x)\) (observed) is the feature derived from the context \(x\), and \(\beta\) is the feature-specific deviation.

Notice that the formulation above is easily extended to multiple effects with coefficients \(\beta\). In our experiment, we have 117 effects (features), each with its own \(\beta\). The first 50 correspond to U.S. states, plus an additional feature for the entire U.S., and they are observed for each text since each text is associated with a known set of states (discussed below). We assume that texts that are generated in different states have distinct characteristics; for each state, we have a binary indicator feature. The other 66 features depend on observed macroeconomics variables at each timestep (e.g., unemployment rate, inflation rate, house price index, etc.). Given an economic state of the world, we hypothesize that there are certain words that are more likely to be used, and each economic variable has its own (sparse) deviation from the background word frequencies. The generative story for a word at timestep \(t\) associated with (observed) features \(f(x^{(t)})\) is:

- Given observed real-world observed variables \(x^{(t)}\), draw word \(w\) from a multinomial distribution

\[
p(w | \theta^{(t)}, \beta^{(t)}, x^{(t)}) \propto \exp(\theta_w^{(t)} + \beta_w^{(t)} f(x^{(t)}))
\]

Our \(L(\beta)\) is simply the negative log-loss function commonly used in multinomial logistic regression:

\[
L(\beta) = \sum_{t=1}^{T} \sum_{i=1}^{N_t} \log p(w_i^{(t)} | \theta^{(t)}, \beta^{(t)}, x_i^{(t)})
\]

We use the 501st–5,501st most frequent words in the dataset. We associated the FOMC meeting transcripts with all states. The “Beige Book” texts were produced by the Federal Reserve Banks. There are twelve Federal Reserve Banks in the United States, each serving a collection of states. We associated texts from a Federal Reserve Bank with the states that it serves.

Quantitative U.S. macroeconomic data was obtained from the Federal Reserve Bank of St. Louis data repository (“FRED”). We used standard measures of economic activity focusing on output (GDP), employ-

6.3.2 Dataset

There is a great deal of text that is produced to describe current macroeconomic events. We conjecture that the connection between the economy and the text will have temporal dependencies (e.g., the amount of discussion about housing or oil prices might vary over time). We use three sources of text commentary on the economy. The first is a subset of the 10-K reports we used in [6,2]. We selected the 10-K reports of 200 companies chosen randomly from the top quintile of market capitalization. This gives us a manageable sample of the largest U.S. companies. Each report is associated with the state in which the company’s head office is located. Our next two data sources come from the Federal Reserve System, the primary body responsible for monetary policy in the U.S. [7] The Federal Open Market Committee (FOMC) meets roughly eight times per year to discuss economic conditions and set monetary policy. Prior to each meeting, each of the twelve regional banks writes an informal “anecdotal” description of economic activity in their region as well as a national summary. This “Beige Book” is akin to a blog of economic activity released prior to each meeting. Each FOMC meeting also produces a transcript of the discussion. For our experiments here, we focus on text from 1996–2006. As a result, we have 2,075 documents in the final corpus, consisting of 842 documents of the 10-K reports, 89 documents of the FOMC meeting transcripts, and 1,144 documents of the Beige Book summaries.

We use the 501st–5,501st most frequent words in the dataset. We associated the FOMC meeting transcripts with all states. The “Beige Book” texts were produced by the Federal Reserve Banks. There are twelve Federal Reserve Banks in the United States, each serving a collection of states. We associated texts from a Federal Reserve Bank with the states that it serves.

Quantitative U.S. macroeconomic data was obtained from the Federal Reserve Bank of St. Louis data repository (“FRED”). We used standard measures of economic activity focusing on output (GDP), employ-

---

6 For an overview of the Federal Reserve System, see the Federal Reserve’s “Purpose and Functions” document at [http://www.federalreserve.gov/pf/pf.htm](http://www.federalreserve.gov/pf/pf.htm).

7 All the text is freely available at [http://www.federalreserve.gov](http://www.federalreserve.gov). The Beige Book is released to the public prior to each meeting. The transcripts are released five years after the meetings.
Table 2: Negative log-likelihood of the documents on various test sets (lower is better). The first test year (2003) was used as our development data. Our model uses the sparse adaptive prior in [4].

| year   | # tokens | ridge-one | ridge-all | ridge-ts | lasso-one | lasso-all | our model |
|--------|----------|-----------|-----------|----------|-----------|-----------|-----------|
| 2003(dev) | 1.1      | 2.736     | 2.765     | 2.735    | 2.736     | 2.765     | 2.735     |
| 2004    | 1.5      | 2.975     | 3.004     | 2.975    | 2.975     | 3.004     | 2.974     |
| 2005    | 1.9      | 2.999     | 3.027     | 2.997    | 2.998     | 3.027     | 2.997     |
| 2006    | 2.3      | 2.916     | 2.922     | 2.913    | 2.912     | 2.922     | 2.912     |
| overall | 6.8      | 11,626    | 11,718    | 11,619   | 11,620    | 11,718    | 11,618    |

We compare our model to the baselines in §6.1. The lasso variants are analogous to the original formulation of SAGE [Eisenstein et al., 2011], except that our model directly conditions on macroeconomic variables instead of a Dirichlet-multinomial compound.

6.3.3 Results

We score models by computing the negative log-likelihood on the test dataset $\log p_{\text{test}}(y) = -\frac{1}{N} \sum_{i=1}^{N} \log p(w_i^{(T+1)} | \theta^{(T)}, \beta^{(T)}, x_i^{(T+1)})$. We initialized all the feature coefficients by the coefficients by training a lasso regression on the last year of the training data (lasso-one). The first test year (i.e., 2003) was used as our development data for hyperparameter tuning ($\tau$ was selected to be .001). Table 2 shows the results for the six models we compared. Similar to the forecasting experiments, at each test year, we trained only on documents from earlier years. When we collapsed all the training data and ignored the temporal dimension (ridge-all and lasso-all), the background log-frequencies $\theta^{(T)}$ are computed using the entire training data, which is different compared to the background log-frequencies for only the last timestep of the training data. Our model outperformed all ridge and lasso variants, including the one with a time-series penalty [Yogatama et al., 2011], in terms of negative log-likelihood on unseen dataset.

In addition to improving predictive accuracy, the prior also allows us to discover trends in the feature coefficients and gain insight. We manually examined the model from the last run (test year 2006). Examples of temporal trends learned by our model are shown in Figure 3. The plot illustrates feature coefficients for words that contain the string employ. For comparison, we also included the percentage of unemployment rate in the U.S. (which was used as one of the features $f(x)$), scaled to fit into the plot. We can see that there is a correlation between feature coefficients for the word unemployment and the actual unemployment rate. On the other hand, the correlations are less evident for other words.

7 Conclusions

We presented a time-series prior for the parameters of probabilistic models; it produces sparse models and adapts the strength of temporal effects on each coefficient separately, based on the data, without an explosion in the number of hyperparameters. We showed how to do inference under this prior using variational approximations. We evaluated the prior for the task of forecasting volatility of stock returns from financial reports, and demonstrated that it outperforms other

\(^7\)For growing output series, like GDP, we calculate growth rates as log differences.

\(^8\)Returns are monthly, excess of the risk-free rate, and continuously compounded. The data are from CRSP and are available for these portfolios at http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html.

\(^9\)Out-of-vocabulary items are ignored.
competing models. We also evaluated the prior for the task of modeling a collection of texts over time, i.e., predicting the probability of words given some observed real-world variables. We showed that the prior achieved state-of-the-art results as well.

Acknowledgments

The authors thank several anonymous reviewers for helpful feedback on earlier drafts of this paper. This research was supported in part by a Google research award to the second and third authors. This research was supported in part by the Intelligence Advanced Research Projects Activity via Department of Interior National Business Center contract number D12PC00347. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

References

Angelosante, Daniele and Giannakis, Georgios B. Rls-weighted lasso for adaptive estimation of sparse signals. In Proc. of ICASSP, 2009.

Angelosante, Daniele, Giannakis, Georgios B., and Grossi, Emanuele. Compressed sensing of time-varying signals. In Proc. of ICDSIP, 2009.

Bach, Francis, Jenatton, Rodolphe, Mairal, Julien, and Obozinski, Guillaume. Convex optimization with sparsity inducing norms. MIT Press, 2011.

Belmonte, Miguel A. G., Koop, Gary, and Korobilis, Dimitiris. Hierarchical shrinkage in time-varying parameter models, 2012. Working paper.

Box, George E. P., Jenkins, Gwilym M., and Reinsel, Gregory C. Time Series Analysis: Forecasting and Control. Wiley Series in Probability and Statistics, 2008.

Caron, François, Bornn, Luke, and Doucet, Arnaud. Sparsity-promoting bayesian dynamic linear models, 2012. arXiv 1203.0106.

Charles, Adam S. and Rozell, Christopher J. Reweighted $\ell_1$ dynamic filtering for time-varying sparse signal estimation, 2012. arXiv 1208.0325.

Duchi, John and Singer, Yoram. Efficient online and batch learning using forward backward splitting. Journal of Machine Learning Research, 10:2873–2908, 2009.

Eisenstein, Jacob, Ahmed, Amr, and Xing, Eric P. Sparse additive generative models of text. In Proc. of ICML, 2011.

Figueiredo, Mario A. T. Adaptive sparseness using jeffreys’ prior. In Proc. of NIPS, 2002.

Hoerl, Arthur E. and Kennard, Robert W. Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 12(1):55–67, 1970.

Kogan, Shimon, Levin, Dimitry, Routledge, Bryan R., Sagi, Jacob S., and Smith, Noah A. Predicting risk from financial reports with regression. In Proc. of NAACL, 2009.

Liu, Dong C. and Nocedal, Jorge. On the limited memory BFGS method for large scale optimization. Mathematical Programming B, 45(3):503–528, 1989.

Liu, Jun and Ye, Jieping. Moreau-yosida regularization for grouped tree structure learning. In Proc. of NIPS, 2010.

Nakajima, Jouchi and West, Mike. Bayesian analysis of latent threshold dynamic models. Journal of Business and Economic Statistics, 2012.

Tibshirani, Robert. Regression shrinkage and selection via the lasso. Journal of Royal Statistical Society B, 58(1):267–288, 1996.

Tibshirani, Robert, Saunders, Michael, Rosset, Saharon, Zhu, Ji, and Knight, Keith. Sparsity and smoothness via the fused lasso. Journal of the Royal Statistical Society B, 67(1):91–108, 2005.

Volpi, Leonardo. Eigenvalues and eigenvectors of tridiagonal uniform matrices. 2003.

Yogatama, Dani, Heilman, Michael, O’Connor, Brendan, Dyer, Chris, Routledge, Bryan R., and Smith, Noah A. Predicting a scientific community’s response to an article. In Proc. of EMNLP, 2011.

Yuan, Ming and Lin, Yi. Model selection and estimation in regression with grouped variables. Journal of the Royal Statistical Society, Series B, 68(1):49–67, 2007.

Zhang, Yu and Yeung, Dit-Yan. A convex formulation for learning task relationships in multi-task learning. In Proc. of UAI, 2010.