A Unified Framework for Missing Data and Cold Start Prediction for Time Series Data

Chris Xie CHRISXIE@CS.WASHINGTON.EDU
Department of Computer Science
University of Washington

Alex Tank ALEXTANK@UW.EDU
Department of Statistics
University of Washington

Emily B. Fox EBFOX@UW.EDU
Department of Statistics
University of Washington

Editor:

Abstract

Providing long-range forecasts is a fundamental challenge in time series modeling, which is only compounded by the challenge of having to form such forecasts when a time series has never previously been observed. The latter challenge is the time series version of the *cold start* problem seen in recommender systems. Time series models are also often plagued by missing data and high-dimensionality (i.e., large collections of observed series), making them ill-suited to the typical structure of big data time series. We provide a unified framework for producing long-range forecasts even when the series has missing values or was previously unobserved; the same framework can be used to impute missing values. Key to the formulation and resulting performance is (1) leveraging repeated patterns over fixed periods of time and across series, and (2) metadata associated with the individual series; both of these assets are common features of big data time series. Our formulation leverages two low-rank decompositions: one to describe low-dimensional latent time series trends underlying our repeated patterns and another to handle the weightings matrix on potentially very high-dimensional metadata. In our simulated experiments, we demonstrate the importance of these two modeling components. We then provide an analysis of these methods on web traffic in a Wikipedia dataset.

Keywords: cold start prediction; matrix factorization; time series

1. Introduction

Large collections of heterogeneous time series are now commonplace in many application domains: continuous environmental sensor data for millions of locations, product demand and purchase curves for millions of products, and web traffic over time for billions of websites. Big, messy data sets of this nature have six common features and challenges.

P1 **Prediction Challenge: Cold Start Forecasting**. Products on Amazon are introduced everyday, and old products are taken down. New websites are created every minute and old ones disappear. How can we forecast demand for this new product/website?

P2 **Prediction Challenge: Missing data**. Sensors or websites may temporarily go down. Products may be temporarily unavailable. How can we impute missing data that results from these events?

P3 **Prediction Challenge: Long-Range Forecasting**. How can we forecast demand for a product in the coming year, so as to appropriately allocate inventory?

F1 **Feature: Metadata**. Products have user reviews and product descriptions. Sensors have locations and proximities to different points of interest. Websites have both content and network information.

F2 **Feature: Shared low-dimensional structure and seasonality**. Many products, websites, and sensors show similar seasonal profiles and reactions to unobserved latent trends.

F3 **Feature: Smoothness**. Demand curves, web traffic, and environmental indications change smoothly over time.

Current approaches to smoothing and prediction in big data time series tend to tackle two or at most three of these features and challenges. Recently, a number of authors have independently proposed applying matrix factorization (MF) techniques to collections of time series (Nguyen et al., 2014; Sun and Malioutov, 2015; Li and Marlin, 2015; Yu et al., 2015; de Fréin et al., 2008). These methods focus on (P2), (F2), and (F3). For example, Nguyen et al. (2014) develops a low-rank multi-output Gaussian process model that was recently applied to handling missing data in the ICU (Li and Marlin, 2015). Yu et al. (2015) takes a similar approach, but adds penalty terms to enforce a vector autoregressive evolution of the latent time series factors. Other approaches ignore smoothness altogether, and instead focus on inferring interpretable latent seasonality patterns using non-negative matrix factorization (Sun and Malioutov, 2015; de Fréin et al., 2008). Anava et al. (2015) focuses on (P2) using online learning methods to fill in the missing data.

Prediction of time series curves from features (F1), or *covariates*, has traditionally been studied in the field of *functional data analysis* (FDA) (Ramsay and Silverman, 2005; Morris, 2014). Specifically, functional response regression predicts a smooth curve, in our case an entire time series, as a function of some covariates. A classical application of this area is in modeling growth curves as functions of treatments and other indicators (Morris, 2014). Importantly, most uses of FDA focus on single-output regression with a small number of covariates—however, in our case metadata contain feature spaces on the order of thousands of dimensions.

We present a unified approach to prediction and modeling of time series collections that naturally tackles prediction of new series and missing data imputation using both high-dimensional metadata and shared seasonality structure. To our knowledge, forcasting entire seasonal profiles for unobserved time series is novel. The key to our approach is harnessing repeated patterns over fixed periods of time (e.g., a yearly cycle). Via matrix factorization, we extract a low-dimensional representation of these repeated patterns seen across periods and time series (F2), enabling us to efficiently form long-range predictions over the next period (P3), as well as impute missing values (P2). In contrast to Sun and Malioutov (2015), we ignore non-negativity constraints since interpretability is not our goal, but rather prediction.

We then turn our attention to the *cold start* challenge (P1), where we want to form long-range predictions for a brand new time series. To handle this, we incorporate metadata (F1) associated with each time series, including our new series. A challenge here is coping with the potentially high-dimensional features provided by the metadata. To this end, we examine a low-rank structure on the feature weightings matrix itself, and show that such a structure is crucial to our predictive performance.

2. Background

Our method draws from now classical approaches to matrix factorization and collaborative filtering. In particular, we draw from matrix factorization approaches to time series analysis and collaborative filtering with side information, or metadata.

2.1 Matrix Factorization for Time Series

Matrix factorization methods approximate a matrix by computing a low-rank approximation to a matrix $Y \in \mathbb{R}^{m \times n}$ as the product of two matrices $W \in \mathbb{R}^{m \times k}, H \in \mathbb{R}^{k \times n}$. This is done by solving the following problem:

$$\min_{W,H} \|Y - WH\|_F^2 \tag{1}$$

It is well known that a global solution to this problem is given by the SVD of Y and retaining only the top k singular values and singular vectors. However, when some of the entries of Y are missing, this problem can still be solved with alternating minimization methods.Nonnegative matrix factorization solves Eq (1) with the constraint that $W, H \geq 0$ component wise. Nonnegative matrix factorization can be solved with multiplicative updates or projected gradient descent (Lee and Seung, 2001; Lin, 2007).

Recent work has exploited shared seasonality patterns for time series forecasting (Sun and Malioutov, 2015). In this work, the time series data is assumed to have a specific seasonality period (e.g. a year). The data is then detrended and averaged over this period to get an average time series matrix $Y \in \mathbb{R}^{m \times n}$ where n is the number of products and m is the number of time points within a year. Their approach applies nonnegative matrix factorization to the averaged time series matrix by solving the problem

$$\begin{aligned} &\min_{W,H} \|Y - WH\|_F^2 + \alpha \left(\|W\|_F^2 + \|H\|_F^2\right) \\ &\text{s.t. } W, H \geq 0 \end{aligned} \tag{2}$$

which gives the interpretation of a latent factor loading matrix W of latent time series and latent factors H. These learned latent factors and factor loadings can be used to forecast one period ahead (e.g. a year). Other work has analyzed financial data using nonnegative matrix factorization to identify underlying trends in stock data (de Fréin et al., 2008). Nguyen et al. (2014) uses Gaussian processes as latent time series to learn a low-rank multi-output regression model. Yu et al. (2015) takes a similar approach, but adds penalty terms to enforce a vector autoregressive model of the latent time series factors.

2.2 Collaborative Filtering and the Cold Start Problem

Collaborative filtering methods aim to recommend items by intelligently filling in a matrix based on observed entries of the matrix. The most famous application of this is the Netflix Challenge, where the goal is to fill in a user-movie ratings matrix in order to recommend movies to users. By constructing a low-rank approximation of the matrix, one can interpret the model as learning latent user factors and latent movie factors that can have many explanations such as movie genres. There has been a vast amount of work that has investigated efficient ways to do this factorization using methods such as minimizing nonconvex loss functions via alternating least squares (Koren et al., 2009) and approximate matrix completion with nuclear norm regularization (Cai et al., 2010).

The cold start problem in the context of collaborative filtering occurs when a user has not rated any movies (e.g. a new user signs up for Netflix). The typical matrix factorization techniques will set the corresponding user factor to zero due to regularization, which is not desirable. Content-based filtering methods have been combined with collaborative filtering methods using user and movie metadata to add a regression component to the model

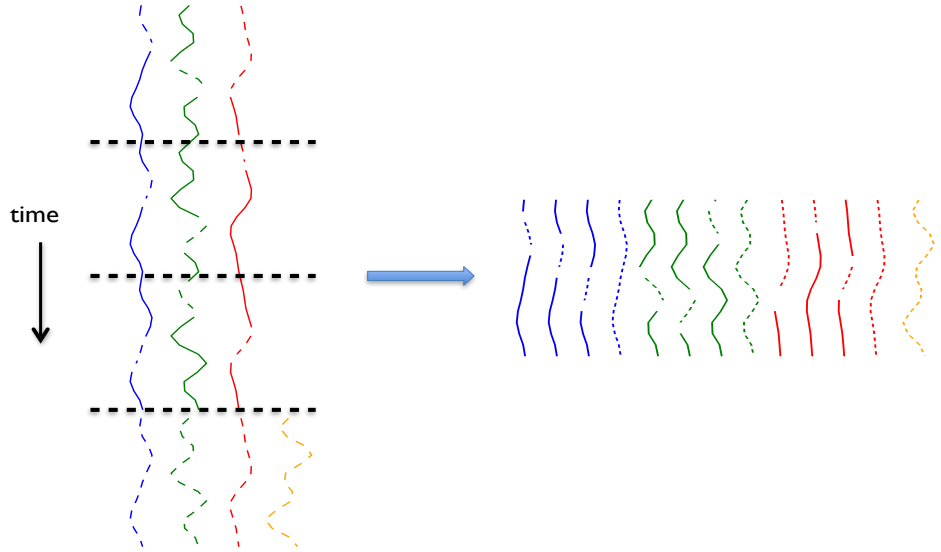


Figure 1: Transformation from raw time series to our data matrix for three observed time series (blue, green, red) and one previously unobserved series (orange). The period of interest is indicated by the horizontal dashed black lines. The solid vertical lines indicate observed data points. The dotted vertical lines are missing values corresponding to the series having different missing time points during the observed time window, and an entire period in the future we wish to predict.

to remedy the cold start problem and provide stronger prediction on the original Netflix problem (Koren et al., 2009). This typically results in a loss function

$$\min_{L,R,w} \frac{1}{2} \sum_{r_{uv} \in \Omega} \left(L_u \cdot R_v + w \cdot \phi(u,v) - r_{uv}\right)^2 + \mathcal{R}(L,R,w) \tag{3}$$

where L, R are the latent user and movie factors, respectively, w is the global regression vector, $\phi(u,v)$ is the metadata for user u and movie v, Ω is the set of observed values, and $\mathcal{R}(\cdot)$ is a regularization function. Cold start prediction in collaborative filtering has been explored extensively (Pilászy and Tikk, 2009; Schein et al., 2002; Gantner et al., 2010; Agarwal and Chen, 2009).

3. Proposed Framework

3.1 Setup

Consider a multivariate time series $[Y_1(t), \ldots, Y_N(t)], t = 1, \ldots, T$. For each univariate time series, we decompose it into a trend component, seasonal component, and a noise component: $Y_i(t) = Y_i^T(t) + Y_i^S(t) + Y_i^N(t)$. We detrend the data by computing $Y_i^{DT}(t) = Y_i(t) - Y_i^T(t)$ using the method of Cleveland et al. (1990). From this point forward, we abuse notation and let $Y_i(t)$ denote the detrended time series.

In order to leverage repeated patterns over fixed periods of time and across series (F2), we assume a period of interest and treat each period of each time series as a column in our data matrix. Thus, we form a new data matrix

$$Y(t) = \left[Y_1^1(t), \ldots, Y_1^{n_1}(t), \ldots, Y_N^1(t), \ldots, Y_N^{n_N}(t)\right] \tag{4}$$

Table 1: Model Framework

	Regression	Low-Rank Regression
No MF	$Y_i = W\phi_i + b + \varepsilon_i$	$Y_i = HU\phi_i + b + \varepsilon_i$
MF	$Y_i = LR_i + W\phi_i + b + \varepsilon_i$	$Y_i = LR_i + HU\phi_i + b + \varepsilon_i$

where $Y_i^m(t)$ denotes the m^{th} period of time series i and n_i denotes the number of years of data for series i. We perform all modeling with this data matrix. We again abuse notation and let T denote the length of the assumed period and N be the number of observed periods, i.e. $N := \sum_{i=1}^{N} n_i$. Figure 1 provides a visual representation of the stacking process. We assume that each $Y_i^m(t)$ is accompanied by the same metadata vector for $m = 1, \ldots, n_i$.

3.2 Model Framework

Given a matrix Y of these detrended and aggregated time series, we combine ideas from the collaborative filtering cold start problem and matrix factorization to consider four different models described in Table 1. Here, $\phi_i \in \mathbb{R}^m$ is a vector of metadata features for series i, $W \in \mathbb{R}^{T\times m}$ is a weight matrix, $H \in \mathbb{R}^{T\times k}, U \in \mathbb{R}^{k\times m}$ represent a low-rank decomposition of the weight matrix, $L \in \mathbb{R}^{T\times k'}, R \in \mathbb{R}^{k'\times N}$ represent the standard matrix factorization decomposition, $b \in \mathbb{R}^T$ is a bias term, and $\varepsilon_i \sim N(0, \sigma^2 I)$ is noise. k is the dimensionality of the latent structure of the weight matrix, and k' is the dimensionality of the latent structure of the matrix factorization term. With these four models, we would like to compare trade offs in prediction power for all three prediction challenges.

To predict with the matrix factorization models on prediction challenges (P1) and (P3), we only use the learned regression matrices to forecast since the latent factors R_i are uninformed by data and driven to zero by the optimization objective as described in Section 2. For prediction challenge (P2), we predict using the entire model.

3.2.1 Motivation

To fill in missing data (P2), one straightforward approach is matrix factorization as in Sun and Malioutov (2015), represented by the L, R matrices in Table 1. This approach leverages an inherent low-rank structure due to similar time series or multiple similar years of the same time series. L can be viewed as a factor loadings matrix, where each column is a latent time series. R can be viewed as a latent factor matrix where column R_i is the latent factor vector (i.e., learned weights on the latent time series) for time series i. This approach is very amenable to the missing data problem, as any missing entry can be computed with an inner product of the corresponding row of L and column of R.

However, when we encounter a new time series or want to predict the next year of an existing time series (unobserved column in the matrix Y), we have no data to inform the corresponding factor and the model will learn to predict zero as described in Section 2. To combat this, we introduce a simple regression component to the model that uses metadata to inform predictions, represented by the W matrix. Note that the regression portion alone can also be used to impute missing data.

When the metadata dimensionality is high, the W matrix becomes very large. To help reduce the parameterization, we perform another low-rank approximation to the weight matrix, resulting in the H, U matrices in Table 1. The columns of H serve as time-varying weights, while $U\phi_i$ represents a low-dimensional representation of the metadata features. In principle, the linear transformation $U\phi_i$ can be replaced by any transformation, even a nonlinear one such as a neural network. An alternative representation of this model is low-rank regression with structured noise (low-rank covariance). In this interpretation, the matrix factorization captures extra structure in the error residuals not captured by the low-rank regression, where the error residuals are defined to be the difference of the observed time series and the regression portion.

3.2.2 Optimization Objectives

Our goal is to fit these models in the missing data setting. Let Ω be the set of observed indices. To fit each of these four models, we compute regularized maximum likelihood estimates of the parameters. This amounts to solving these optimization problems:

- Regression:
$$\underset{W,b}{\operatorname{argmin}} \frac{1}{2N} \sum_{(i,j)\in\Omega} \left(Y_{ij} - W_j^\intercal \phi_i - b_j\right)^2 + \frac{\lambda_1}{2N}\|W\|_F^2$$
- Matrix factorization + regression:
$$\underset{L,R,W,b}{\operatorname{argmin}} \frac{1}{2N} \sum_{(i,j)\in\Omega} \left(Y_{ij} - L_j^\intercal R_i - W_j^\intercal \phi_i - b_j\right)^2 + \frac{\lambda_1}{2N}\|W\|_F^2 + \frac{\lambda_2}{2N}\left(\|L\|_F^2 + \|R\|_F^2\right)$$
- Low-rank regression:
$$\underset{H,U,b}{\operatorname{argmin}} \frac{1}{2N} \sum_{(i,j)\in\Omega} \left(Y_{ij} - H_j^\intercal U \phi_i - b_j\right)^2 + \frac{\lambda_1}{2N}\left(\|H\|_F^2 + \|U\|_F^2\right)$$
- Matrix factorization + low-rank regression:
$$\underset{L,R,H,U,b}{\operatorname{argmin}} \frac{1}{2N} \sum_{(i,j)\in\Omega} \left(Y_{ij} - L_j^\intercal R_i - H_j^\intercal U \phi_i - b_j\right)^2 + \frac{\lambda_1}{2N}\left(\|H\|_F^2 + \|U\|_F^2\right) + \frac{\lambda_2}{2N}\left(\|L\|_F^2 + \|R\|_F^2\right)$$

Each of these models can be solved with gradient descent or stochastic gradient descent (SGD). The regression model has an analytical solution, but is too expensive to compute in the high-dimensional setting. In our experiments, we solve these optimization problems using minibatch SGD.

4. Experiments

We evaluate the trade offs in performance of the proposed methods on all three challenges (P1, P2, P3) on synthetic data, and then perform an analysis of a real world dataset using our models. We describe each experimental setup for the prediction challenges:

P1 **Cold Start Forecasting**: The test set will consist of metadata and time series such that no historic seasonal period exists in the training set. Our goal is to forecast this completely new seasonal profile, corresponding to the entire dotted orange line as shown in the right hand side of Figure 1.

P2 **Missing data**: The training data contains the time series data matrix Y and all metadata. The test set will consist of missing elements of Y, and our goal is to impute these missing elements, corresponding to the portions of dotted lines embedded in the solid lines in the right hand side of Figure 1.

P3 **Long-Range Forecasting**: The training set includes the metadata and all historic seasonal periods of the series except the last, which is included in the test set. Our goal is to forecast this entire last seasonal period, corresponding to the entire dotted blue, green, and red lines as shown in the right hand side of Figure 1.

4.1 Synthetic Data

We generated synthetic data according to the matrix factorization + low-rank regression model. Each column of H and L is a sine wave of amplitude one with randomly generated periods. We sample $R_i \sim N(0, 0.015I)$, $U \sim N(0, 0.05I)$, $\varepsilon_i \sim N(0, .04I)$, and $\phi_{ik} \sim (1 - z_{ik})\delta_0 + z_{ik}\text{Exp}(5)$ where $z_{ik} \sim$ Bernoilli(.02); this is the spike and slab exponential distribution for ϕ_{ik} to emulate high-dimensional features that are sparse and positive, e.g. bag-of-words features.

Table 2: Residual sum of squares (RSS) on the synthetic test set for all 4 models and all 3 prediction challenges after running cross validation.

Model	Cold Start (P1)	Missing Data (P2)	Long-Range (P3)
Reg	7.461	1.852	6.763
Low-Rank Reg	5.997	1.416	5.945
MF + Reg	7.452	1.358	6.787
MF + Low-Rank Reg	**5.980**	**1.307**	**5.895**

We generated a synthetic dataset with dimensions $T = 300, N = 5000, m = 1000, k = k' = 20$. We train the models with minibatch SGD and run cross validation on λ_1, λ_2 over a log scale of 10 values of λ_i from 0.0001 to 100. After selecting hyperparameters, we trained each final model with 10 random restarts to avoid local minima.

The residual sum of squares (RSS) for all four models for all three prediction challenges are given in Table 2. Across all three challenges, the low-rank models perform better than their full regression counterparts, although the full regression models subsume the low-rank regression models. Furthermore, the combined matrix factorization + low-rank model performs best across all prediction challenges. In the prediction experiments (P1, P3), the improved performance shows that it pays to explicitly model the error residual structure even when cold start and long-range predictions are only based on the low-rank regression coefficients and metadata. The missing data experiments (P2) show that when the error residuals have low-rank structure combining regression with matrix factorization of the error residuals leads to better missing data imputation than regression alone.

4.2 Analysis of Wikipedia Page Traffic Dataset

We explore our framework in the context of a real world dataset collected from Wikipedia. The features associated with each article is very high-dimensional, thus we only consider the low-rank regression models due to computational bottlenecks. We focus on the long-range forecast (P3) and cold start prediction (P1) challenges. Furthermore, our experiments of Section 4.1 demonstrate the statistical importance of this structure even in moderate dimensions.

4.2.1 Dataset Collection

We collected daily page traffic counts from almost 4500 Wikipedia articles from the beginning of 2008 to the end of 2014 by querying Wiki Trends (Wikipedia, 2016). The pages were selected by obtaining a list of the 5000 most popular pages of a given week in March 2016. We use a yearly seasonality period and stack the time series data matrix as described in Section 3. Each year of article traffic has $T = 365$ days. For each article, we have anywhere between 1 to 7 years of page traffic counts starting from 2008 to 2014. After splitting each series by year, the total number of columns of Y is approximately $N = 29000$. For each series we aim to predict the seasonal pattern, rather than total traffic volume, and thus we normalize the time series to have maximum magnitude of one.

For each article, we scraped the summary of the Wikipedia page, removed stopwords, tokenized it using the Stanford CoreNLP toolkit (Manning et al., 2014), and calculated TF-IDF representations of these summaries to use as our metadata vectors. Additionally, each word used in our feature vector is present in at least two articles. After preprocessing, the dimension of the features is approximately $m = 22,000$. This matrix is extremely sparse with roughly 0.5% nonzero entries.

Table 3: Residual sum of squares (RSS) on the test set for all low-rank regression models and baselines on the Wikipedia dataset. For the (P1) baseline, we run a k-NN regression. For the (P3) baseline, we average past observed years.

Model	Cold Start (P1)	Long-Range (P3)
Baseline	3.700	3.497
Low-Rank Reg	3.703	3.022
MF + Low-Rank Reg	**3.643**	**3.000**

4.2.2 Quantitative Analysis

We compared the performance of the low-rank regression models on a test set and used a held out set to train the model parameters and select hyperparameters. To generate the test set for cold start prediction (P1), we randomly selected 25% of the articles and included the last year of page traffic in the test set. We did not include any previous years of such articles from the training set. To generate the test set for long-range forecasts (P3), we included the last year of page traffic for each article in the test set. For both experiments, we further removed 20% of values from the time series data in both the training to demonstrate the effectiveness of our prediction methods in the presence of missing training data.

To train our low-rank regression models, we used a validation set derived from the training set to select our values of λ_i, which were evenly spread on a log scale from 0.001 to 100. We trained the models with minibatch SGD with a minibatch size of 500 for 10,000 iterations. After selecting hyperparameters, we trained each final model by running 10 random restarts to avoid local minima.

Due to the novel application, there are no other existing baselines to our knowledge for (P1) and (P3) challenges. For (P1), we propose a baseline drawn from content filtering based recommender systems (Gantner et al., 2010) : use the k nearest neighbors in metadata feature space to forecast. For each nearest metadata neighbor (in Euclidean distance), we averaged its time series over all years. We then performed the prediction using a weighted average of the averaged time series where the weight was proportional to the inverse distance between metadata vectors. For long-range forecasts (P3), we use a simple baseline that predicts the future year trend using the average of the past year trends.

The results are given in Table 3. For the k-NN baseline for (P1), we compared $k = 5, 10, 15, 20$ and selected $k = 10$ due to best performance. Note that across prediction challenges the matrix factorization + low-rank regression model outperforms both baseline methods and pure low-rank regression. These prediction results suggest that the matrix factorization portion on the regression residuals helps to capture extra structure in the data that the low-rank regression portion cannot capture alone. Importantly, prediction is performed for both matrix factorization + low-rank and pure low-rank using only the regression coefficients; this implies that the matrix factorization residuals aid in improved learning of the low-rank regression coefficients.

4.2.3 Qualitative Analysis

We may visually examine the performance of the matrix factorization + low-rank regression model. In Figures 2 and 3, we plot held out time series and our predictions of them. Each subfigure has two subplots; the first one has the true observed time series in magenta with the prediction overlayed in a thick green, and the second is a zoomed in version of the predicted time series. Our model makes low-magnitude predictions due to the matrix factorization absorbing some of the magnitude during learning. Zooming in on the prediction allows one to examine the yearly pattern of the prediction.

Long-range forecasts For the predictions on the long-range forecasting experiment (P3) in Figure 2, our model captures a variety of interesting seasonal patterns. Figures 2a and 2f display a predicted yearly pattern that includes a summer and winter slump. This may be due to students visiting these articles during the school year. Figure 2b also exhibits such a structure while catching the extra bump around day 160. Note that we correctly capture weekly oscillation structure as in Figure 2d. By relying on past observed years, we can correctly predict recurring spikes such as the July 4th spike in Figure 2c. Figure 2e shows predicted increases in page traffic during what are most likely prestigious tennis tournaments.

Cold-start forecasts In the more challenging cold-start scenario, our model still learns the common summer and winter slump structure as shown in Figures 3b, 3c, and 3d. Additionally, in Figures 3a and 3f, the model predicts the locations of spikes, or sharp rises in page traffic. Specifically, in Figure 3e we predict some of the same traffic spikes due to tennis tournaments as in Figure 2e. It is interesting that we are still able to capture this spikey structure despite there being no historic time series data pertaining to "Maria Sharapova". Furthermore, in Figure 3f we predict a correct spike in the "United States presidential election" around late October. However, this prediction seems to miss an earlier, even larger spike in activity. We note that prediction of activity spikes is challenging since many activity spikes result from viral news events, rather than recurring yearly events.

While we do not leverage smoothness assumptions over time, we see that our resulting predictions are in fact smooth. This is a result of our models extracting smooth time series from the data, which exhibits repeated smooth patterns.

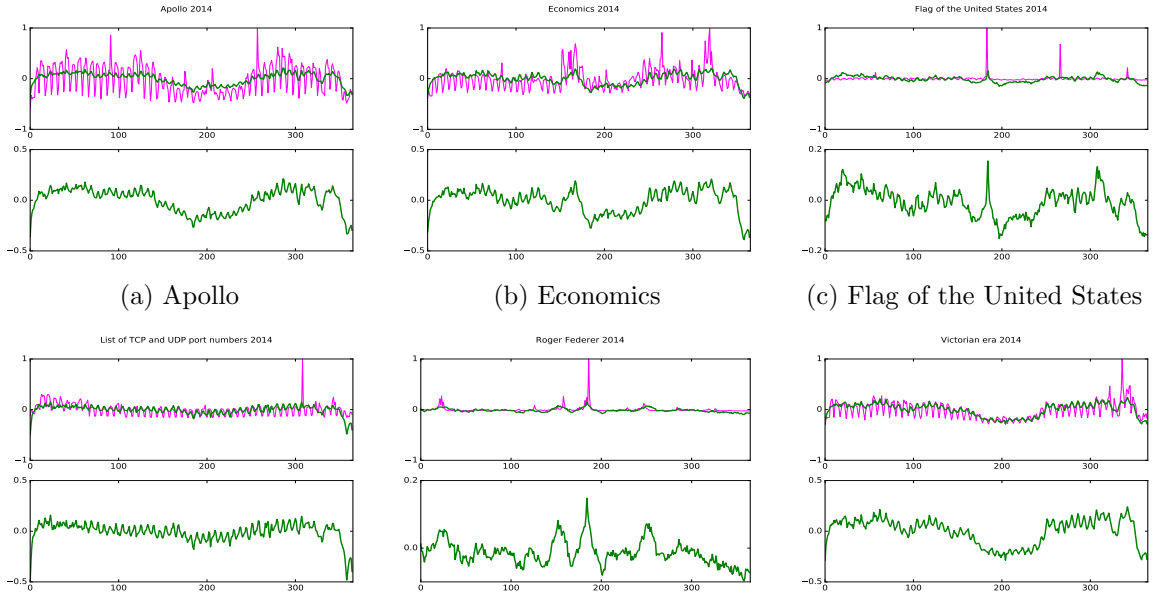
5. Discussion

We presented a unified framework for long-range forecasting applicable to big data time series. Our approach elegantly handles missing observations while providing both long-range forecasts and cold start predictions. Our formulation leverages two low-rank decompositions. One is a matrix factorization of a carefully constructed data matrix such that repeated patterns over fixed periods of time and across series can be learned as latent factor processes. The second is a decomposition of the weightings matrix on the regression term that enables us to incorporate high-dimensional metadata to aid in the cold-start problem. A benefit of our simple featurized MF approach is the computational efficiency associated with this type of structure.

In our simulated experiments, we demonstrated that the low-rank decomposition of the weightings matrix is critical to forecasting completely new time series. We also showed that the matrix factorization component is important for imputing missing data and capturing a low-rank correlation structure in the residuals. We provided an analysis of a large Wikipedia dataset where we showed that our methods can produce long-range forecasts of web traffic on totally new pages. Such long-range forecasts—in our case an entire year—are extremely challenging for standard time series models even when long histories of a process have previously been observed. In future work, we plan to investigate the benefits of functional forms such as spline bases to capitalize on smoothness assumptions (F3).

Acknowledgements

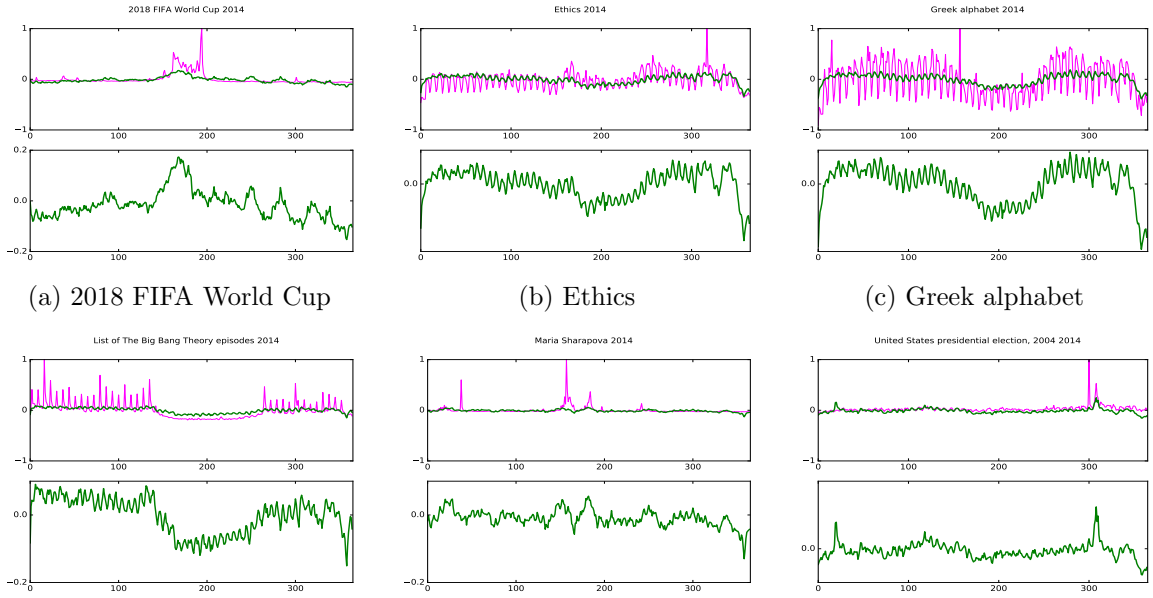
This work was supported in part by ONR Grant N00014-15-1-2380 and NSF CAREER Award IIS-1350133. Chris Xie was supported in part by an NDSEG fellowship.



(a) Apollo (b) Economics (c) Flag of the United States

(d) List of TCP and UDP port numbers (e) Roger Federer (f) Victorian era

Figure 2: Long-Range forecasts on Wikipedia page traffic dataset for year 2014



(a) 2018 FIFA World Cup (b) Ethics (c) Greek alphabet

(d) List of The Big Bang Theory episodes (e) Maria Sharapova (f) United States presidential election, 2004

Figure 3: Cold start predictions on Wikipedia page traffic dataset for year 2014.

References

Deepak Agarwal and Bee-Chung Chen. Regression-based latent factor models. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 19–28. ACM, 2009.

Oren Anava, Elad Hazan, and Assaf Zeevi. Online time series prediction with missing data. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 2191–2199, 2015.

Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

Robert B Cleveland, William S Cleveland, Jean E McRae, and Irma Terpenning. Stl: A seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics*, 6(1):3–73, 1990.

Ruairí de Fréin, Konstantinos Drakakis, Scott Rickard, and Andrzej Cichocki. Analysis of financial data using non-negative matrix factorization. In *International Mathematical Forum*, volume 3, pages 1853–1870. Journals of Hikari Ltd, 2008.

Zeno Gantner, Lucas Drumond, Christoph Freudenthaler, Steffen Rendle, and Lars Schmidt-Thieme. Learning attribute-to-feature mappings for cold-start recommendations. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 176–185. IEEE, 2010.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.

Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

Steve Cheng-Xian Li and Benjamin Marlin. Collaborative multi-output gaussian processes for collections of sparse multivariate time series. *NIPS 2015 Time Series Workshop*, 2015.

Chih-Jen Lin. Projected gradient methods for nonnegative matrix factorization. *Neural computation*, 19(10):2756–2779, 2007.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014.

J. S. Morris. Functional Regression. *ArXiv e-prints*, June 2014.

VT Nguyen, Edwin Bonilla, et al. Collaborative multi-output gaussian processes. UAI, 2014.

István Pilászy and Domonkos Tikk. Recommending new movies: even a few ratings are more valuable than metadata. In *Proceedings of the third ACM conference on Recommender systems*, pages 93–100. ACM, 2009.

James O.. Ramsay and Bernard W Silverman. *Functional Data Analysis*. Springer, 2005.

Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 253–260. ACM, 2002.

Wei Sun and Dmitry Malioutov. Time series forecasting with shared seasonality patterns using non-negative matrix factorization. In *NIPS, Time Series Workshop*, 2015.

Wikipedia. Wikipedia pageview statistics: Wiki trends. http://www.wikipediatrends.com/, 2016. URL http://www.wikipediatrends.com/.

H.-F. Yu, N. Rao, and I. S. Dhillon. High-dimensional Time Series Prediction with Missing Values. *ArXiv e-prints*, September 2015.