Journal of Machine Learning Research 9 (2008) 2217-2250 Submitted 6/08; Revised 8/08; Published 10/08

Forecasting Web Page Views: Methods and Observations

Jia Li JIALI@STAT.PSU.EDU
Visiting Scientist∗
Google Labs, 4720 Forbes Avenue
Pittsburgh, PA 15213

Andrew W. Moore AWM@GOOGLE.COM
Engineering Director
Google Labs, 4720 Forbes Avenue
Pittsburgh, PA 15213.

Editor: Lyle Ungar

Abstract

Web sites must forecast Web page views in order to plan computer resource allocation and estimate upcoming revenue and advertising growth. In this paper, we focus on extracting trends and seasonal patterns from page view series, two dominant factors in the variation of such series. We investigate the Holt-Winters procedure and a state space model for making relatively short-term prediction. It is found that Web page views exhibit strong impulsive changes occasionally. The impulses cause large prediction errors long after their occurrences. A method is developed to identify impulses and to alleviate their damage on prediction. We also develop a long-range trend and season extraction method, namely the *Elastic Smooth Season Fitting (ESSF)* algorithm, to compute scalable and smooth yearly seasons. ESSF derives the yearly season by minimizing the residual sum of squares under smoothness regularization, a quadratic optimization problem. It is shown that for long-term prediction, ESSF improves accuracy significantly over other methods that ignore the yearly seasonality.

Keywords: web page views, forecast, Holt-Winters, Kalman filtering, elastic smooth season fitting

1. Introduction

This is a machine learning application paper about a prediction task that is rapidly growing in importance: predicting the number of visitors to a Web site or page over the coming weeks or months. There are three reasons for this growth in importance. First, hardware and network bandwidth need to be provisioned if a site is growing. Second, any revenue-generating site needs to predict its revenue. Third, sites that sell advertising space need to estimate how many page views will be available before they can commit to a contract from an advertising agency.

1.1 Background on Time Series Modeling

Time series are commonly decomposed into "trend", "season", and "noise":

$$X_t = L_t + I_t + N_t \ , \tag{1}$$

∗. Also, Associate Professor, Department of Statistics, The Pennsylvania State University.

where L_t is trend, I_t is season, and N_t is noise. For some prediction methods, L_t is more than a global growth pattern, in which case it will be referred to as "level" to distinguish from the global pattern often called trend. These components of a time series need to be treated quite differently. The noise N_t is often modeled by stationary ARMA (autoregressive moving average) process (Brockwell and Davis, 2002; Wei, 2006). Before modeling the noise, the series needs to be "detrended" and "deseasoned". There are multiple approaches to trend and season removal (Brockwell and Davis, 2002). In the well-known Box-Jenkins ARIMA (autoregressive integrated moving average) model (Box and Jenkins, 1970), the difference between adjacent lags (i.e., time units) is taken as noise. The differencing can be applied several times. The emphasis of ARIMA is still to predict noise. The trend is handled in a rather rigid manner (i.e., by differencing). In some cases, however, trend and season may be the dominant factors in prediction and require methods devoted to their extraction. A more sophisticated approach to compute trend is by smoothing, for instance, global polynomial fitting, local polynomial fitting, kernel smoothing, and exponential smoothing. Exponential smoothing is generalized by the Holt-Winters (HW) procedure to include seasonality. Chatfield (2004) provides practical accounts on when ARIMA model or methods aimed at capturing trend and seasonality should be used.

Another type of model that offers the flexibility of handling trend, season, and noise together is the state space model (SSM) (Durbin and Koopman, 2001). The ARIMA model can be cast into an SSM, but SSM includes much broader non-stationary processes. SSM and its computational method—the Kalman filter were developed in control theory and signal processing (Kalman, 1960; Sage and Melsa, 1971; Anderson and Moore, 1979). For Web page view series, experiments suggest that trend and seasonality are more important than the noise part for prediction. We thus investigate the HW procedure and an SSM emphasizing trend and seasonality. Despite its computational simplicity, HW has been successful in some scenarios (Chatfield, 2004). The main advantages of SSM over HW are (a) some parameters in the model are estimated based on the series, and hence the prediction formula is adapted to the series; (b) if one wants to modify the model, the general framework of SSM and the related computational methods apply the same way, while HW is a relatively specific static solution.

1.2 Web Page View Prediction

Web page view series exhibit seasonality at multiple time scales. For daily page view series, there is usually a weekly season and sometimes a long-range yearly season. Both HW and SSM can effectively extract the weekly season, but not the yearly season for several reasons elaborated in Section 4. For this task, we develop the Elastic Smooth Season Fitting (ESSF) method. It is observed that instead of being a periodic sequence, the yearly seasonality often emerges as a yearly pattern that may scale differently across the years. ESSF takes into consideration the scaling phenomenon and only requires two years of data to compute the yearly season. Experiments show that the prediction accuracy can be improved remarkably based on the yearly season computed by ESSF, especially for forecasting distant future.

To our best knowledge, existing work on forecasting Internet access data is mostly for network traffic load. For short-term traffic, it is reasonable to assume that the random process is stationary, and thus prediction relies on extracting the serial statistical dependence in the seemingly noisy series. Stationary ARMA models are well suited for such series and have been exploited (Basu et al., 1996; You and Chandra, 1999). A systematic study of the predictability of network traffic based

on stationary traffic models has been conducted by Sang and Li (2001). For long-term prediction of large-scale traffic, because trends often dominate, prediction centers around extracting trends. Depending on the characteristics of trends, different methods may be used. In some cases, trends are well captured by growth rate and the main concern is to accurately estimate the growth rate, for instance, that of the overall Internet traffic (Odlyzko, 2003). Self-similarity is found to exist at multiple time scales of network traffic, and is exploited for prediction (Grossglauser and Bolot, 1999). Multiscale wavelet decomposition has been used to predict one-minute-ahead Web traffic (Aussem and Murtagh, 2001), as well as Internet backbone traffic months ahead (Papagiannaki et al., 2005). Neural networks have also been applied to predict short-term Internet traffic (Khotanzad and Sadek, 2003). An extensive collection of work on modeling self-similar network traffic has been edited by Park and Willinger (2000).

We believe Web page view series, although closely related to network traffic data, have particular characteristics worthy of a focused study. The contribution of the paper is summarized as follows.

1. We investigate short-term prediction by HW and SSM. The advantages and disadvantages of the two approaches in various scenarios are analyzed. It is also found that seasonality exists at multiple time scales and is important for forecasting Web page view series.

2. Methods are developed to detect sudden massive impulses in the Web traffic and to remedy their detrimental impact on prediction.

3. For long-term prediction several months ahead, we develop the ESSF algorithm to extract global trends and scalable yearly seasonal effects after separating the weekly season using HW.

1.3 Application Scope

The prediction methods in this paper focus on extracting trend and season at several scales, and are not suitable for modeling stationary stochastic processes. The ARMA model, for which mature off-the-shelf software is available, is mostly used for such processes. The trend extracted by HM or SSM is the noise-removed non-season portion of a time series. If a series can be compactly described by a growth rate, it is likely better to directly estimate the growth rate. However, HW and SSM are more flexible in the sense of not assuming specific functional form for the trend on the observed series. HW and SSM are limited for making long-term prediction. By HW, the predicted level term of the page view at a future time is assumed to be the current level added by a linear function of the time interval, or simply the current level if linear growth is removed, as in some reduced form of HW. If a specifically parameterized function can be reliably assumed, it is better to estimate parameters in the function and apply extrapolation accordingly. However, in the applications we investigated, there is little base for choosing any particular function. The yearly season extraction by ESSF is found to improve long-term prediction. The basic assumption of ESSF is that the time series exhibits a yearly pattern, possibly scaled differently across the years. It is not intended to capture event driven pattern. For instance, the search volume for `Batman` surges around the release of every new Batman movie, but shows no clear yearly pattern.

In particular, we have studied two types of Web page view series: (a) small to moderate scale Web sites; (b) dynamic Web pages generated by Google for given search queries. Due to the fast changing pace of the Internet, page view series available for small to moderate scale Web sites are usually short (e.g., shorter than two years). Therefore, the series are insufficient for exploiting

yearly seasonality in prediction. The most dramatic changes in those series are often the news-driven surges. Without side information, such surges cannot be predicted from the page view series alone. It is difficult for us to acquire page view data from Web sites with long history and very high access volume because of privacy constraints. We expect the page views of large-scale Web sites to be less impulsive in a relative sense because of their high base access. Moreover, large Web sites are more likely to have existed long enough to form long-term, for example, yearly, access patterns. Such characteristics are also possessed by the world-wide volume data of search queries, which we use in our experiments.

The rest of the paper is organized as follows. In Section 2, the Holt-Winters procedure is introduced. The effect of impulses on the prediction by HW is analyzed, based on which methods of detection and correction are developed. In Section 3, we present the state space model and discuss the computational issues encountered. Both HW and SSM aim at short-term prediction. The ESSF algorithm for long-term prediction is described in Section 4. Experimental results are provided in Section 5. We discuss predicting the noise part of the series by AR (autoregressive) models and finally conclude in Section 6.

2. The Holt-Winters Procedure

Let the time series be $\{x_1, x_2, ..., x_n\}$. The Holt-Winters (HW) procedure (Chatfield, 2004) decomposes the series into level L_t, season I_t, and noise. The variation of the level after one lag is assumed to be captured by a local linear growth term T_t. Let the period of the season be d. The HW procedure updates L_t, I_t, and T_t simultaneously by a recursion:

$$\begin{aligned} L_t &= \zeta(x_t - I_{t-d}) + (1-\zeta)(L_{t-1} + T_{t-1}), && (2)\\ T_t &= \kappa(L_t - L_{t-1}) + (1-\kappa)T_{t-1}, && (3)\\ I_t &= \delta(x_t - L_t) + (1-\delta)I_{t-d} && (4) \end{aligned}$$

where the pre-selected parameters $0 \le \zeta \le 1$, $0 \le \kappa \le 1$, and $0 \le \delta \le 1$ control the smoothness of updating. This is a stochastic approximation method in which the current level is an exponentially weighted running average of recent season-adjusted observations. To better see this, let us assume the season and linear growth terms are absent. Then Eq. (2) reduces to

$$\begin{aligned} L_t &= \zeta x_t + (1-\zeta)L_{t-1} && (5)\\ &= \zeta x_t + (1-\zeta)\zeta x_{t-1} + (1-\zeta)^2 L_{t-2}\\ &\;\;\vdots\\ &= \zeta x_t + (1-\zeta)\zeta x_{t-1} + (1-\zeta)^2\zeta x_{t-2} + \cdots + (1-\zeta)^{t-1}\zeta x_1 + (1-\zeta)^t L_0. \end{aligned}$$

Suppose L_0 is initialized to zero, the above equation is an on the fly exponential smoothing of the time series, that is, a weighted average with the weights attenuating exponentially into the past. We can also view L_t in Eq. (5) as a convex combination of the level indicated by the current observation x_t and the level suggested by the past estimation L_{t-1}. When the season is added, x_t subtracted by the estimated season at t becomes the part of L_t indicated by current information. At this point of recursion, the most up-to-date estimation for the season at t is I_{t-d} under period d. When the linear growth is added, the past level L_{t-1} is expected to become $L_{t-1} + T_{t-1}$ at t. Following the same scheme of convex combination, Eq. (5) evolves into (2). Similar rationale applies to the update

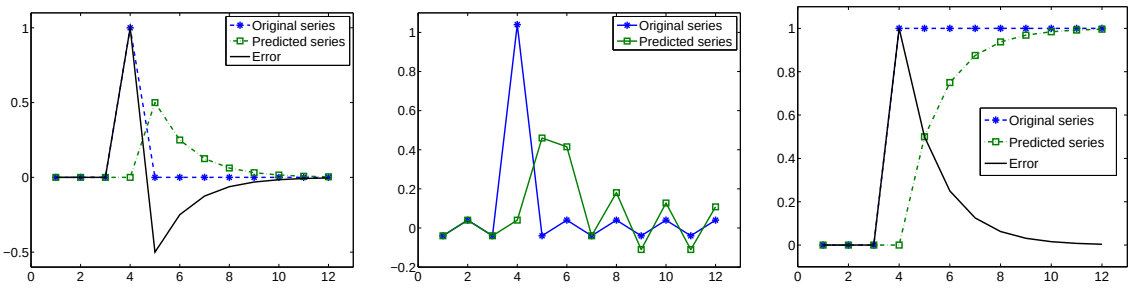


Figure 1: Holt-Winters prediction for time series with abrupt changes. (a) Impulse effect on a leveled signal: slow decaying tail; (b) Impulse effect on a periodic signal: ripple effect; (c) Response to a step signal.

of T_t and I_t in Eqs. (3) and (4). Based on past information, T_t and I_t are expected to be T_{t-1} and I_{t-d} under the implicit assumption of constant linear growth and fixed season. On the other hand, the current x_t and the newly computed L_t suggest T_t to be $L_t - L_{t-1}$, and I_t to be $x_t - L_t$. Applying convex combination leveraging past and current information, we obtain Eqs. (3) and (4).

To start the recursion in the HW procedure at time t, initial values are needed for L_{t-1}, T_{t-1}, and $I_{t-\tau}$, $\tau = 1,2,...,d$. We use the first period of data $\{x_1, x_2, ..., x_d\}$ for initialization, and start the recursion at $t = d+1$. Specifically, linear regression is conducted for $\{x_1, x_2, ..., x_d\}$ versus the time grid $\{1,2,...,d\}$. That is, x_τ and τ, $\tau = 1,...,d$, are treated as dependent variable and independent variable respectively. Suppose the regression function obtained is $b_1\tau + b_2$. We initialize by setting $L_\tau = b_1\tau + b_2$, $T_\tau = 0$, and $I_\tau = x_\tau - L_\tau$, $\tau = 1,2,...,d$.

The forecasting of h time units forward at t, that is, the prediction of x_{t+h} based on $\{x_1, x_2, ..., x_t\}$, is

$$\hat{x}_{t+h} = L_t + hT_t + I_{t-d+h \bmod d} ,$$

where *mod* is the modulo operation. The linear function of h, $L_t + hT_t$, with slope given by the most updated linear growth T_t, can be regarded as an estimation for L_{t+h}; while $I_{t-d+h \bmod d}$, the most updated season at the same cyclic position as $t+h$, which is already available at t, is the estimation for I_{t+h}.

Experiments using the HW procedure show that the local linear growth term, T_t, helps little in prediction. In fact, for relatively distant future, the linear growth term degrades performance. This is because for the Web page view series, we rarely see any linear trends visible over a time scale from which the gradient can be estimated by HW. We can remove the term T_t in HW conveniently by initializing it with zero and setting the corresponding smoothing parameter $\kappa = 0$.

Web page view series sometimes exhibit impulsive surges or dips. Such impulsive changes last a short period of time and often bring the level of page views to a magnitude one or several orders higher than the normal range. For instance, in Figure 5(a), the amount of page views for an example Web site jumps tremendously at the 404th day and returns to normal one day later. Impulses are triggered by external forces which are unpredictable based on the time series alone. One such common external force is a news launch related to the Web site. Because it is extremely difficult if possible at all to predict the occurrence of an impulse, we focus on preventing its after effect.

The influence of an impulse on the prediction by the HW procedure is elaborated in Figure 1. In Figure 1(a), a flat leveled series with an impulse is processed by HW. The predicted series attempts to catch up with the impulse after one lag. Although the impulse is over after one lag, the predicted series attenuates slowly, causing large errors several lags later. The stronger the impulse is, the slower the predicted series returns close to the original one. The prediction error consumes a positive value and then a negative one, both of large magnitudes. Apparently, a negative impulse will result in a reversed error pattern. Figure 1(b) shows the response of HW to an impulse added to a periodic series. The prediction error still yields the pattern of switching signs and large magnitudes. To reduce the influence of an impulse, it is important that we differentiate an impulse from a sudden step-wise change in the series. When a significant step appears, we want the predicted series to catch up with the change as fast as possible rather than hindering the strong response. Figure 1(c) shows the prediction by HW for a series with a sudden positive step change. The prediction error takes a large positive value and reduces gradually to zero without crossing into the negative side.

Based on the above observations, we detect an impulse by examining the co-existence of errors with large magnitudes and opposite signs within a short window of time. In our experiments, the window size is $s_1 = 10$. The extremity of the prediction error is measured relatively with respect to the standard deviation of prediction errors in the most recent past of a pre-selected length. In the current experiment, this length is $s_2 = 50$. The time units of s_1 and s_2 are the same as that of the time series in consideration. Currently, we manually set the values of $s_{1,2}$. The rationale for choosing these values is that s_1 implies the maximum length of an impulse; and s_2 balances accurate estimation of the noise variance and swift adaptation to the change of the variance over time. We avoid setting s_1 too high to ensure that a detected impulse is a short-lived, strong, and abrupt change. If a time series undergoes a real sudden rising or falling trend, the prediction algorithm will capture the trend but with a certain amount of delay, as shown by the response of HW to a step signal in Figure 1(c). In a special scenario when an impulse locates right at the boundary of a large rising trend, the measure taken to treat the impulse will further slow down the response to, but not prevent the eventual catch-up of the rise.

At time t, let the prediction for x_t based on the past series up to $t-1$ be $\hat{x}_t$, and the prediction error be $e_t = x_t - \hat{x}_t$. We check whether an impulse has started at t', $t - s_1 + 1 \le t' \le t - 1$, and ended at t by the following steps.

1. Compute the standard deviation with removed outliers, σ_{t-1}, for the prediction errors $\{e_{t-s_2}, e_{t-s_2+1}, ..., e_{t-1}\}$, which are known by time t. The motivation for removing the outliers is that at any time an impulse exists, the prediction error will be unusually large, and hence bias the estimated average amount of variation. In our experiments, 10% of the errors are removed as outliers.

2. Compute the relative magnitude of e_t by $\theta_t = \frac{|e_t|}{\sigma_{t-1}}$.

3. Examine $\theta_{t'}$ in the window $t' \in [t - s_1 + 1, t]$. If there is a t', $t - s_1 + 1 \le t' \le t - 1$, such that $\theta_{t'} > \Delta_1$ and $\theta_t > \Delta_2$ and $sign(e_{t'}) \neq sign(e_t)$, the segment $[t', t]$ is marked as an impulse. If $e_{t'}$ is positive while e_t negative, the impulse is a surge; the reverse is a dip. The two thresholds Δ_1 and Δ_2 determine the sensitivity to impulses and are chosen around 2.5.

If impulse is not detected, the HW recursion is applied at the next time unit $t+1$. Otherwise, L_t, T_t, and $I_{t'}$ for $t' \in [t - s_1 + 1, t]$, are revised as follows to reduce the effect of the impulse on the future L_τ, T_τ, and I_τ, $\tau > t$. Once the revision is completed, the HW recursion resumes at $t+1$.

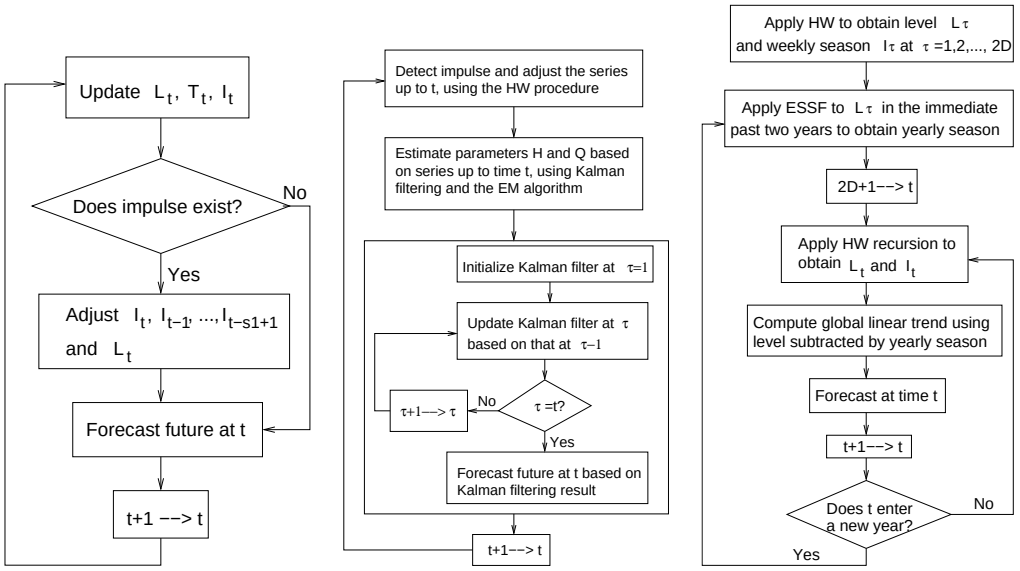


Figure 2: The schematic diagrams for the forecasting algorithms: (a) Holt-Winters with impulse detection; (b) GLS; (c) ESSF.

1. For $t' = t - s_1 + 1, \ldots, t$, set $I_{t'} = I_{t'-d}$ sequentially. This is equivalent to discarding the season computed during the impulse segment and using the most recent season right before the impulse.

2. Let $L_t = \frac{1}{2} L_{t-s_1} + \frac{1}{2}(x_t - I_t)$, where L_{t-s_1} is the level before the impulse and I_t is the already revised season at t.

3. Let $T_t = 0$.

In this paper, we constrain our interest to reducing the adverse effect of an impulse on later prediction after it has occurred and been detected. Predicting the arrival of impulses in advance using side information, for instance, scheduled events impacting Web visits, is expected to be beneficial, but is beyond our study here. A schematic diagram of the HW procedure is illustrated in Figure 2(a).

Holt-Winters and our impulse-resistant modification have the merit of being very cheap to update and predict, requiring only a handful of additions and multiples. This may be useful in some extremely high throughput situations, such as network routers. But in more conventional settings, it leads to the question: can we do better with more extensive model estimation at each time step?

3. State Space Model

A state space model (SSM) assumes that there is an underlying state process for the series $\{x_1, \ldots, x_n\}$. The states are characterized by a Markov process, and x_t is a linear combination of the states added

with Gaussian noise. In general, an SSM can be represented in the following matrix form:

$$\begin{aligned} x_t &= \mathbf{Z}_t\alpha_t + \varepsilon_t\,, \qquad \varepsilon_t \sim N(0,\mathbf{H}_t), \\ \alpha_{t+1} &= \mathbf{T}_t\alpha_t + \mathbf{R}_t\eta_t\,, \quad \eta_t \sim N(0,\mathbf{Q}_t)\,, \quad t = 1,...,n, \\ &\qquad\qquad \alpha_1 \sim N(a_1,\mathbf{P}_1) \end{aligned} \tag{6}$$

where $\{\alpha_1,\alpha_2,...,\alpha_n\}$ is the state process. Each state is an m-dimensional column vector. Although in our work, the observed series x_t is univariate, SSM treats generally p-dimensional series. The noise terms ε_t and η_t follow Gaussian distributions with zero mean and covariance matrices $\mathbf{H}_t$ and $\mathbf{Q}_t$ respectively. For clarity, we list the dimension of the matrices and vectors in (6) below.

observation	x_t	$p\times 1$	$\mathbf{Z}_t$	$p\times m$
state	α_t	$m\times 1$	$\mathbf{T}_t$	$m\times m$
noise	ε_t	$p\times 1$	$\mathbf{H}_t$	$p\times p$
noise	η_t	$r\times 1$	$\mathbf{R}_t$	$m\times r$
			$\mathbf{Q}_t$	$r\times r$
initial state mean	a_1	$m\times 1$	$\mathbf{P}_1$	$m\times m$

We restrict our interest to time invariant SSM where the subscript t can be dropped for $\mathbf{Z}$, $\mathbf{T}$, $\mathbf{R}$, $\mathbf{H}$, and $\mathbf{Q}$. Matrices $\mathbf{Z}$, $\mathbf{T}$ and $\mathbf{R}$ characterize the intrinsic relationship between the state and the observed series, as well as the transition between states. They are determined once we decide upon a model. The covariance matrices $\mathbf{H}$ and $\mathbf{Q}$ are estimated based on the time series using the Maximum Likelihood (ML) criterion.

Next, we describe the *Level with Season (LS)* model, which decomposes x_t in the same way as the HW procedure in Eq. (2)$\sim$(4), with the linear growth term removed. We discard the growth term because, as mentioned previously, this term does not contribute in the HW procedure under our experiments. However, if necessary, it would be easy to modify the SSM to include this term. We then describe the *Generalized Level with Season (GLS)* model that can explicitly control the smoothness of the level.

3.1 The Level with Season Model

Denote the level at t by μ_t and the season with period d by i_t. The LS model assumes

$$\begin{aligned} x_t &= \mu_t + i_t + \varepsilon_t, \\ i_t &= -\sum_{j=1}^{d-1} i_{t-j} + \eta_{1,t}, \\ \mu_t &= \mu_{t-1} + \eta_{2,t} \end{aligned} \tag{7}$$

where ε_t and $\eta_{j,t}$, $j = 1,2$, are the Gaussian noises.

Comparing with the HW recursion equations (2)$\sim$(4), Eq. (7) is merely a model specifying the statistical dependence of x_t on μ_t and i_t, both of which are unobservable random processes. The Kalman filter for this model, playing a similar role as Eqs. (2)-(4) for HW, will be computed recursively to estimate μ_t, i_t, and to predict future. Details on the Kalman filter are provided in Appendix A. In its simplest form, with both the linear growth and season term removed, HW reduces to exponential smoothing with recursion $L_t = \zeta x_t + (1-\zeta)L_{t-1}$. It can be shown that if we let $L_t = E(\mu_t \mid x_1,...,x_{t-1})$, the recursion for L_t in HW is the same as that derived from the Kalman

filter for the LS model without season. The smoothing parameter ζ is determined by the parameters of the noise distributions in LS. When season is added, there is no complete match between the recursion of HW and that of the Kalman filter. In the LS model, it is assumed that $\sum_{\tau=1}^{d} i_{t+\tau} = 0$ up to white noise, but HW does not enforce the zero sum of one period of the season terms. The decomposition of x_t into level μ_t and season i_t by LS is however similar to that assumed by HW.

We can cast the LS model into a time invariant SSM following the notation of (6). The matrix expansion according to (6) leads to the same set of equations in (7):

$$\alpha_t = \begin{pmatrix} i_t \\ i_{t-1} \\ \vdots \\ i_{t-d+2} \\ \mu_t \end{pmatrix}, \quad \eta_t = \begin{pmatrix} \eta_{1,t} \\ \eta_{2,t} \end{pmatrix},$$

$$\underset{d\times 1}{\mathbf{Z} = \quad (1,0,0,\cdots,0,1)}, \quad \underset{d\times 3}{\mathbf{R} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 1 \end{pmatrix}}, \quad \underset{d\times d}{\mathbf{T} = \begin{pmatrix} -1 & -1 & -1 & \cdots & -1 & 0 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 1 \end{pmatrix}}.$$

3.2 Generalized Level with Season Model

We generalize the above LS model by imposing different extent of smoothness on the level term μ_t. Specifically, let

$$\begin{aligned} x_t &= \mu_t + i_t + \varepsilon_t, \\ i_t &= -\sum_{j=1}^{s-1} i_{t-j} + \eta_{1,t}, \\ \mu_t &= \frac{1}{q}\sum_{j=1}^{q} \mu_{t-j} + \eta_{2,t}\ . \end{aligned} \tag{8}$$

Here $q \geq 1$ controls the extent of smoothness. The higher the q, the smoother the level $\{\mu_1, \mu_2, ..., \mu_n\}$. We experiment with $q = 1, 3, 7, 14$.

Again, we cast the model into an SSM. The dimension of the state vector is $m = d - 1 + q$.

$$\alpha_t = \begin{pmatrix} i_t \\ \vdots \\ i_{t-d+2} \\ \mu_t \\ \vdots \\ \mu_{t-q+1} \end{pmatrix}, \quad \eta_t = \begin{pmatrix} \eta_{1,t} \\ \eta_{2,t} \end{pmatrix}.$$

We describe **Z**, **R**, **T** by the sparse matrix format. Denote the (i, j)th element of a matrix, for example, **T**, by $\mathbf{T}(i, j)$ (one index for vectors). An element is zero unless specified.

$$\begin{aligned}
&\mathbf{Z} = [\mathbf{Z}(i,j)]_{1\times m}, && \mathbf{Z}(1) = 1,\ \mathbf{Z}(d) = 1,\\
&\mathbf{R} = [\mathbf{R}(i,j)]_{m\times 2}, && \mathbf{R}(1,1) = 1,\ \mathbf{R}(d,2) = 1,\\
&\mathbf{T} = [\mathbf{T}(i,j)]_{m\times m}, && \mathbf{T}(1,j) = -1,\ j = 1,2,...,d-1,\\
& && \mathbf{T}(1+j,j) = 1,\ j = 1,2,...,d-2,\\
& && \mathbf{T}(d,d-1+j) = \frac{1}{q},\ j = 1,...,q,\\
& && \mathbf{T}(d+j,d-1+j) = 1, j = 1,2,...,q-1, \text{ if } q > 1.
\end{aligned}$$

We compare the LS and GLS models in Section 5 by experiments. It is shown that for distant prediction, imposing smoothness on the level can improve performance.

In practice, the prediction of a future x_{t+h} based on $\{x_1, x_2, ..., x_t\}$ comprises two steps:

1. Estimate **H** and **Q** in GLS (or SSM in general) using the past series $\{x_1, ..., x_t\}$.

2. Estimate x_{t+h} by the conditional expectation $E(x_{t+h} \mid x_1, x_2, ..., x_t)$ under the estimated model.

We may not need to re-estimate the model with every new coming x_t, but update the model once every batch of data. We estimate the model by the ML criterion using the EM algorithm. The Kalman filter and smoother, which involve forward and backward recursion respectively, are the core of the EM algorithm for SSM. Given an estimated model, the Kalman filter is used again to compute $E(x_{t+h} \mid x_1, x_2, ..., x_t)$, as well as the variance $Var(x_{t+h} \mid x_1, x_2, ..., x_t)$: a useful indication for the prediction accuracy. Details on the algorithms for estimating SSM and making prediction based on SSM are provided in the Appendix. A thorough coverage on the theories of SSM and related computational methods is referred to Durbin and Koopman (2001).

Because treating impulses improves prediction, as demonstrated by the experiments in Section 5, it is conducted for the GLS approach. In particular, we invoke the impulse detection embedded in HW. For any segment of time where an impulse is marked, the observed data x_t are replaced by $L_t + I_t$ computed by HW. This modified series is then input to the GLS estimation and prediction algorithms. The schematic diagram for forecasting using GLS is shown in Figure 2(b).

4. Long-range Trend and Seasonality

Web page views sometimes show long-range trend and seasonality. In Figure 7(a), three time series over a period of four years are shown. Detailed description of the series is provided in Section 5. Each time series demonstrates apparently a global trend and yearly seasonality. For instance, the first series, namely `amazon`, grows in general over the years and peaks sharply every year around December. Such long-range patterns can be exploited for forecasting, especially for distant future. To effectively extract long-range trend and season, several needs ought to be addressed:

1. Assume the period of the long-range season is a year. Because the Internet is highly dynamic, it is necessary to derive the yearly season using past data over recent periods and usually only a few (e.g., two) are available.

2. A mechanism to control the smoothness of the long-range season is needed. By enforcing smoothness, the extracted season tends to be more robust, a valuable feature especially when given limited past data.

3. The magnitude of the yearly season may vary across the years. As shown in Figure 7(a), although the series over different years show similar patterns, the patterns may be amplified or shrunk over time. The yearly season thus should be allowed to scale.

The HW and GLS approaches fall short of meeting the above requirements. They exploit mainly the local statistical dependence in the time series. Because HW (and similarly GLS) performs essentially exponential smoothing on the level and linear growth terms, the effect of historic data further away attenuates fast. HW is not designed to extract a global trend over multiple years. Furthermore, HW requires a relatively large number of periods to settle to the intended season; and importantly, HW assumes a fixed season over the years. Although HW is capable of adjusting with a slowly changing season when given enough periods of data, it does not directly treat the scaling of the season, and hence is vulnerable to the scaling phenomenon.

In our study, we adopt a linear regression approach to extract the long-range trend. We inject elasticity into the yearly season and allow it to scale from a certain yearly pattern. The algorithm developed is called *Elastic Smooth Season Fitting (ESSF)*. The time unit of the series is supposed to be a day.

4.1 Elastic Smooth Season Fitting

Before extracting long-range trend and season, we apply HW with impulse detection to obtain the weekly season and the smoothed level series L_t, $t = 1,...,n$. Recall that the HW prediction for the level L_{t+h} at time t is L_t, assuming no linear growth term in our experiments. We want to exploit the global trend and yearly season existing in the level series to better predict L_{t+h} based on $\{L_1, L_2, ..., L_t\}$.

We decompose the level series L_t, $t = 1,...,n$, into a yearly season, y_t, a global linear trend u_t, and a volatility part n_t:

$$L_t = u_t + y_t + n_t \ , \quad t = 1,2,...,n \ .$$

Thus the original series x_t is decomposed into:

$$x_t = u_t + y_t + n_t + I_t + N_t \ , \tag{9}$$

where I_t and N_t are the season and noise terms from HW. Let $\mathbf{u} = \{u_1, u_2, ..., u_n\}$ and $\mathbf{y} = \{y_1, y_2, ..., y_n\}$. They are solved by the following iterative procedure. At this moment, we assume the ESSF algorithm, to be described shortly, is available. We start by setting $\mathbf{y}^{(0)} = 0$. At iteration p, update $\mathbf{y}^{(p)}$ and $\mathbf{u}^{(p)}$ by

1. Let $g_t = L_t - y_t^{(p-1)}$, $t = 1,...,n$. Note g_t is the global trend combined with noise, taking out the current additive estimate of the yearly season.

2. Perform linear regression of $\mathbf{g} = \{g_1, ..., g_n\}$ on the time grid $\{1,2,...,n\}$. Let the regressed value at t be $u_t^{(p)}$, $t = 1,2,...,n$. Thus for some scalars $b_1^{(p)}$ and $b_2^{(p)}$, $u_t^{(p)} = b_1^{(p)} t + b_2^{(p)}$.

3. Let $z_t = L_t - u_t^{(p)}$, $t = 1, ..., n$. Here z_t is the yearly season combined with noise, taking out the current estimate of the global trend. Apply ESSF to $\mathbf{z} = \{z_1, z_2, ..., z_n\}$. Let the yearly season derived by ESSF be $\mathbf{y}^{(p)}$.

It is analytically difficult to prove the convergence of the above procedure. Experiments based on three series show that the difference in $\mathbf{y}^{(p)}$ reduces very fast. At iteration p, $p \geq 2$, we measure the relative change from $\mathbf{y}^{(p-1)}$ to $\mathbf{y}^{(p)}$ by

$$\frac{||\mathbf{y}^{(p)} - \mathbf{y}^{(p-1)}||}{||\mathbf{y}^{(p-1)}||}, \tag{10}$$

where $||\cdot||$ is the L_2 norm. Detailed results are provided in Section 5. Because ESSF always has to be coupled with global trend extraction, for brevity, we also refer to the entire procedure above as ESSF when the context is clear, particularly, in Section 4.2 and Section 5.

We now present the ESSF algorithm for computing the yearly season based on the trend removed $\mathbf{z}$. For notational brevity, we re-index day t by double indices (k, j), which indicates day t is the jth day in the kth year. Denote the residue $z_t = L_t - u_t$ by $z_{k,j}$, the yearly season y_t by $y_{k,j}$ (we abuse the notation here and assume the meaning is clear from the context), and the noise term n_t by $n_{k,j}$. Suppose there are a total of K years and each contains D days. Because leap years contain one more day, we take out the extra day from the series before applying the algorithm.

We call the yearly season pattern $\bar{\mathbf{y}} = \{\bar{y}_1, \bar{y}_2, ..., \bar{y}_D\}$ the *season template*. Since we allow the yearly season $y_{k,j}$ to scale over time, it relates to the season template by

$$y_{k,j} = \alpha_{k,j}\bar{y}_j, \quad k = 1, 2, ..., K, \ j = 1, ..., D,$$

where $\alpha_{k,j}$ is the scaling factor. One choice for $\alpha_{k,j}$ is to let $\alpha_{k,j} = c_k$, that is, a constant within any given year. We call this scheme step-wise constant scaling since $\alpha_{k,j}$ is a step function if single indexed by time t. One issue with the step-wise constant scaling factor is that $y_{k,j}$ inevitably jumps when entering a new year. To alleviate the problem, we instead use a piece-wise linear function for $\alpha_{k,j}$. Let $c_0 = 1$. Then

$$\alpha_{k,j} = \frac{j-1}{D}c_k + \frac{D-j+1}{D}c_{k-1}, \quad k = 1, 2, ..., K, \ j = 1, ..., D. \tag{11}$$

The number of scaling factors c_k to be determined is still K. Let $\mathbf{c} = \{c_1, ..., c_K\}$. At the first day of each year, $\alpha_{k,1} = c_{k-1}$. We optimize over both the season template $\bar{y}_j$, $j = 1, ..., D$, and the scaling factors c_k, $k = 1, ..., K$.

We now have

$$z_{k,j} = \alpha_{k,j}\bar{y}_j + n_{k,j},$$

where $z_{k,j}$'s are given, while c_k, $\bar{y}_j$, and $n_{k,j}$, $k = 1, ..., K$, $j = 1, ..., D$, are to be solved. A natural optimization criterion is to minimize the sum of squared residues:

$$\min_{\bar{\mathbf{y}}, \mathbf{c}} \sum_k \sum_j n_{k,j}^2 = \min_{\bar{\mathbf{y}}, \mathbf{c}} \sum_k \sum_j (z_{k,j} - \alpha_{k,j}\bar{y}_j)^2 .$$

If the number of years K is small, $\bar{\mathbf{y}}$ obtained by the above optimization can be too wiggly. We add a penalty term to ensure the smoothness of $\bar{\mathbf{y}}$. The discrete version of the second order derivative for $\bar{y}_j$ is

$$\ddot{\bar{y}}_j = \bar{y}_{j+1} + \bar{y}_{j-1} - 2\bar{y}_j ,$$

and $\sum_j \ddot{\bar{y}}_j^2$ is used as the smoothness penalty. Since $\bar{\mathbf{y}}$ is one period of the yearly season, when j' is out of the range $[1,D]$, $\bar{y}_{j'}$ is understood as $\bar{y}_{j' \bmod D}$. For instance, $\bar{y}_0 = \bar{y}_D$, $\bar{y}_{D+1} = \bar{y}_1$. We form the following optimization criterion with a pre-selected regularization parameter λ:

$$\min_{\bar{\mathbf{y}},\mathbf{c}} G(\bar{\mathbf{y}},\mathbf{c}) = \min_{\bar{\mathbf{y}},\mathbf{c}} \sum_k \sum_j (z_{k,j} - \alpha_{k,j}\bar{y}_j)^2 + \lambda \sum_j (\bar{y}_{j+1} + \bar{y}_{j-1} - 2\bar{y}_j)^2 \,. \tag{12}$$

To solve (12), we alternate the optimization of $\bar{\mathbf{y}}$ and $\mathbf{c}$. With either fixed, $G(\bar{\mathbf{y}},\mathbf{c})$ is a convex quadratic function. Hence a unique minimum exists and can be solved by a multivariable linear equation. The algorithm is presented in details in Appendix B.

Experiments show that allowing scalable yearly season improves prediction accuracy, so does the smoothness regularization of the yearly season. As long as λ is not too small, the prediction performance varies marginally for a wide range of values. The sensitivity of prediction accuracy to λ is studied in Section 5.

A more ad-hoc approach to enforce smoothness is to apply moving average to the yearly season extracted without smoothness regularization. We can further simplify the optimization criterion in (12) by employing step-wise constant scaling factor, that is, let $\alpha_{k,j} = c_k$, $k = 1,...,K$. The jump effect caused by the abrupt change of the scaling factor is reduced by the moving average as well. Specifically, the optimization criterion becomes

$$\min_{\bar{\mathbf{y}},\mathbf{c}} \tilde{G}(\bar{\mathbf{y}},\mathbf{c}) = \min_{\bar{\mathbf{y}},\mathbf{c}} \sum_k \sum_j (z_{k,j} - c_k\bar{y}_j)^2 \,. \tag{13}$$

The above minimization is solved again by alternating the optimization of $\bar{\mathbf{y}}$ and $\mathbf{c}$. See Appendix B for details. Comparing with Eq. (12), the optimization for (13) reduces computation significantly. After acquiring $\bar{\mathbf{y}}$, we apply a double sided moving average. We call the optimization algorithm for (13) combined with the post operation of moving average the fast version of ESSF. Experiments in Section 5 show that ESSF Fast performs similarly to ESSF.

4.2 Prediction

We note again that ESSF is for better prediction of the level L_t obtained by HW. To predict x_t, the weekly season extracted by HW should be added to the level L_t. The complete process of prediction is summarized below. We assume that prediction starts on the 3rd year since the first two years have to serve as past data for computing the yearly season.

1. Apply HW to obtain the weekly season I_t, and the level L_t, $t = 1,2,...,n$.

2. At the beginning of each year k, $k = 3,4,...$, take the series of L_t's in the past two years (year $k-2$ and $k-1$) and apply ESSF to this series to solve the yearly season template $\bar{\mathbf{y}}$ and the scaling factors, c_1 and c_2 for year $k-2$ and $k-1$ respectively. Predict the yearly season for future years $k' \geq k$ by $c_2\bar{\mathbf{y}}$. Denote the predicted yearly season at time t in any year $k' \geq k$ by $Y_{t,k}$, where the second subscript clarifies that only the series before year k is used by ESSF.

3. Denote the year in which day t lies by $\nu(t)$. Let the yearly season removed level be $\tilde{L}_t = L_t - Y_{t,\nu(t)}$. At every t, apply linear regression to $\{\tilde{L}_{t-2D+1},...,\tilde{L}_t\}$ over the time grid $\{1,2,...,2D\}$. The slope of the regressed line is taken as the long-range growth term $\tilde{T}_t$.

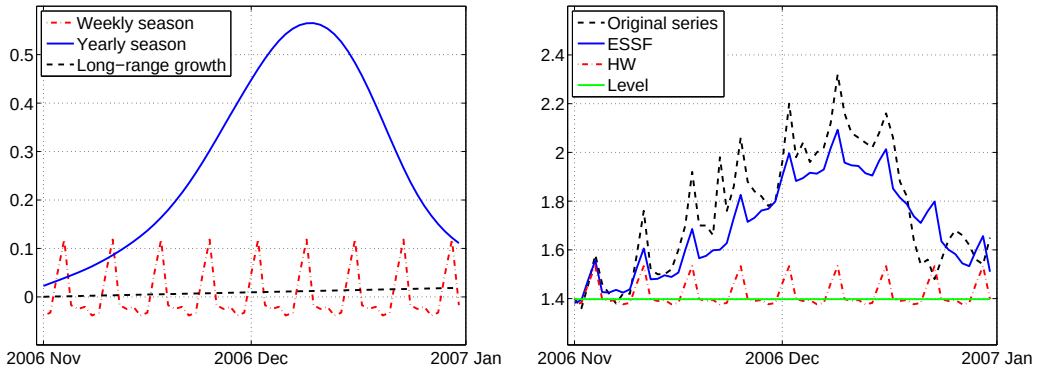


Figure 3: Decomposition of the prediction terms for the amazon series in November and December of 2006 based on data up to October 31, 2006: (a) The weekly season, yearly season, and long-range linear growth terms in the prediction; (b) Comparison of the predicted series by HW and ESSF.

Suppose at the end of day t (or beginning of day $t+1$), we predict for the hth day ahead of t. Let the prediction be $\hat{x}_{t+h}$. Also let $r(t+h)$ be the smallest integer such that $t+h-r(t+h)\cdot d \le t$ (d is the weekly period). Then,

$$\hat{x}_{t+h} = \tilde{L}_t + h\tilde{T}_t + Y_{t+h,\nu(t)} + I_{t+h-r(t+h)\cdot d}\ . \tag{14}$$

Drawing a comparison between Eqs. (14) and (9), we see that $\tilde{L}_t + h\tilde{T}_t$ is essentially the prediction for the global linear trend term u_{t+h}, $Y_{t+h,\nu(t)}$ the prediction for the yearly season y_{t+h}, and $I_{t+h-r(t+h)\cdot d}$ the prediction for the weekly season I_{t+h}. The schematic diagram for forecasting by ESSF is shown in Figure 2(c).

If day $t+h$ is in the same year as t, $Y_{t+h,\nu(t)} = Y_{t+h,\nu(t+h)}$ is the freshest possible prediction for the yearly season at $t+h$. If instead $\nu(t) < \nu(t+h)$, the yearly season at $t+h$ is predicted based on data more than one year ago. One might have noticed that we use only two years of data to extract the yearly season regardless of the available amount of past data. This is purely an individual choice due to our preference of using recent data. Experiments based on the series described in Section 5 show that whether all the available past data are used by ESSF causes negligible difference in prediction performance.

To illustrate the roles of the terms in the prediction formula (14), we plot them separately in Figure 3(a) for the amazon series. The series up to October 31, 2006 is assumed to have been observed, and the prediction is for November and December of 2006. Figure 3(a) shows that during these two months, the predicted yearly season is much more prominent than the weekly season and the slight linear growth. Figure 3(b) compares the prediction by ESSF and HW respectively. The series predicted by HW is weekly periodic with a flat level, while that by ESSF incorporates the yearly seasonal variation and is much closer to the original series, as one might have expected.

5. Experiments

We conduct experiments using twenty six time series. As a study of the characteristics of Web page views, we examine the significance of the seasonal as well as impulsive variations. Three relatively short series are used to assess the performance of short-term prediction by the HW and GLS approaches. The other twenty three series are used to test the ESSF algorithm for long-term prediction. In addition to comparing the different forecasting methods, we also present results to validate the algorithmic choices made in ESSF.

5.1 Data Sets

We conduct experiments based on the time series described below.

1. The `Auton` series records the daily page views of the Auton Lab, headed by Andrew Moore, in the Robotics Institute at the Carnegie Mellon University (*http://www.autonlab.org*). This series spans from August 14, 2005 to May 1, 2007, a total of 626 days.

2. The `Wang` series records the daily page views of the Web site for the research group headed by James Wang at the Pennsylvania State University (*http://wang.ist.psu.edu*). This series spans from January 1, 2006 to February 1, 2008, a total of 762 days.

3. The `citeseer` series records the hourly page views to citeseer, an academic literature search engine currently located at *http://citeseer.ist.psu.edu*. This series spans from 19 : 00 on September 6, 2005 to 4 : 00 on September 25, 2005, a total of 442 hours.

4. We acquired 23 relatively long time series from the site *http://www.google.com/trends*. This Web site provides search volumes for user specified phrases. We treat the search volumes as an indication of the page views to dynamically generated Web pages by Google. The series record daily volumes from Jan, 2004 to December 30, 2007 (roughly four full years), a total of 1460 days. The volumes for each phrase are normalized with respect to the average daily volume of that phrase in the month of January 2004. The normalization will not affect the prediction accuracy, which is measured relatively with respect to the average level of the series. We also call the series collectively the `g-trends` series.

5.2 Evaluation

Let the prediction for x_t be $\hat{x}_t$. Suppose prediction is provided for a segment of the series, $\{x_{t_0+1}, x_{t_0+2}, \ldots, x_{t_0+J}\}$, where $0 \le t_0 < n$. We measure the prediction accuracy by the error rate defined as

$$R_e = \sqrt{\frac{RSS}{SSS}}$$

where *RSS*, the *residual sum of squares* is

$$RSS = \sum_{t=t_0+1}^{t_0+J} (\hat{x}_t - x_t)^2 \tag{15}$$

and *SSS*, the *series sum of squares* is

$$SSS = \sum_{t=t_0+1}^{t_0+J} x_t^2 \,. \tag{16}$$

We call R_e the prediction error rate. It is the reciprocal of the square root of the signal to noise ratio (SNR), a measure commonly used in signal processing. We can also evaluate the effectiveness of a prediction method by comparing *RSS* to *SPV*, the *sum of predictive variation*:

$$SPV = \sum_{t=t_0+1}^{t_0+J} (x_t - \bar{x}_t)^2, \quad \bar{x}_t = \frac{\sum_{\tau=1}^{t-1} x_\tau}{t-1}.$$

We can consider $\bar{x}_t$, the mean up to time $t-1$, as the simplest prediction of x_t using past data. We call this scheme of prediction Mean of Past (MP). *SPV* is essentially the *RSS* corresponding to the MP method. The R_e of MP is $\sqrt{SPV/SSS}$. We denote the ratio between the R_e of a prediction method and that of MP by $Q_e = \sqrt{RSS/SPV}$, referred to as the error ratio. As a measure on the amount of error, in practice, R_e is more pertinent than Q_e for users concerned with employing the prediction in subsequent tasks. We thus use R_e as the major performance measure in all the experimental results. For comparison with baseline prediction methods, we also show R_e of MP as well as that of the Moving Average (MA). In the MA approach, considering the weekly seasonality, we treat Monday to Sunday separately. Specifically, if a day to be predicted is a Monday, we forecast by the average of the series on the past 4 Mondays. Similarly for the other days of a week. For the hourly page view series with daily seasonality, MA predicts by the mean of the same hours in the past 4 days.

As discussed previously, Web page views exhibit impulsive changes. The prediction error during an impulse is extraordinarily large, skewing the average error rate significantly even if impulses only exist on a small fraction of the series. The bias caused by the outlier errors is especially strong when the usual amount of page views is low. We reduce the effect of outliers by removing a small percentage of large errors, in particular, 5% in our experiments. Without loss of generality, suppose the largest (in magnitude) 5% errors are $\hat{x}_t - x_t$ at $t_0 + 1 \le t \le t_1$. We adjust RSS and SSS by using only $\hat{x}_t - x_t$ at $t > t_1$ and compute the corresponding R_e. Specifically,

$$RSS_{adj} = \sum_{t=t_1+1}^{t_0+J} (\hat{x}_t - x_t)^2, \; SSS_{adj} = \sum_{t=t_1+1}^{t_0+J} x_t^2, \; R_e^{adj} = \sqrt{\frac{RSS_{adj}}{SSS_{adj}}}.$$

We report both R_e and R_e^{adj} to measure the prediction accuracy for the series `Auton`, `Wang`, and `citeseer`. For the twenty three `g-trends` series, because there is no clear impulse, we use only R_e.

Because the beginning portion of the series with a certain length is needed for initialization in HW, SSM, or ESSF, we usually start prediction after observing $t_0 > 0$ time units. Moreover, we may predict several time units ahead for the sum of the series over a run of multiple units. The ground truth at t is not necessarily x_t. In general, suppose prediction starts after t_0 and is always for a stretch of w time units that starts h time units ahead. We call w the *window size* of prediction and h the *unit ahead*.

Let the whole series be $\{x_1, x_2, ..., x_n\}$. In the special case when $h = 1$ and $w = 1$, after observing the series up to $t-1$, we predict for x_t, $t = t_0 + 1, ..., n$. The ground truth at t is x_t. If $h \ge 1$, $w = 1$, we predict for x_t, $t = t_0 + h, ..., n$, after observing the series up to $t - h$. Let the predicted value be $\hat{x}_{t,-h}$, where the subscript $-h$ emphasizes that only data h time units ago are used. If $h \ge 1$, $w \ge 1$, we predict for

$$\tilde{x}_t = \sum_{\tau=t}^{t+w-1} x_\tau, \quad t = t_0 + h, ..., n - w + 1,$$

after observing the series up to $t-h$. The predicted value at t is

$$\hat{\tilde{x}}_t = \sum_{\tau=t}^{t+w-1} \hat{x}_{\tau,-h}\,.$$

To compute the error rate R_e, we adjust *RSS* and *SSS* in Eq. (15) and (16) according to the ground truth:

$$RSS = \sum_{t=t_0+h}^{n-w+1} (\hat{\tilde{x}}_t - \tilde{x}_t)^2\,, \quad SSS = \sum_{t=t_0+h}^{n-w+1} \tilde{x}_t^2\,.$$

For the series `Auton`, `Wang`, and `citeseer`, $t_0 = 4d$, where d is the season period. The segment $\{x_1,...,x_{4d}\}$ is used for initialization by both HW and SSM. For the `g-trends` series, $t_0 = 731$. That is, the first two years of data are used for initialization. Two years of past data are needed because the ESSF algorithm requires at least two years of data to operate.

5.3 Results

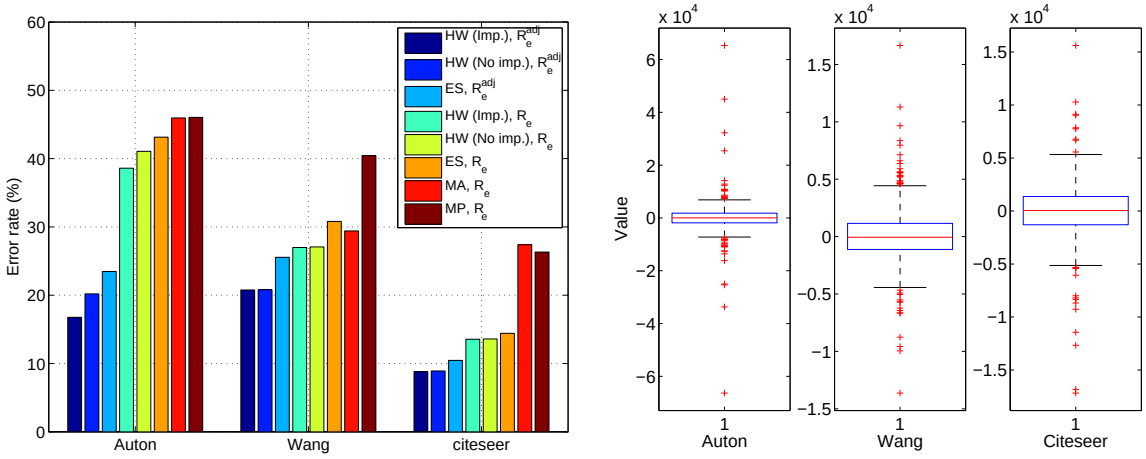
For the series `Auton`, `Wang`, and `citeseer`, we focus on short-term prediction no greater than 30 time units ahead. Because the series are not long enough for extracting long-range trend and season by the ESSF algorithm, we only test the HW procedure with or without impulse detection and the GLS approach. For the twenty three `g-trends` series, we compare ESSF with HW for prediction up to half a year ahead.

5.3.1 SHORT-TERM PREDICTION

Web page views often demonstrate seasonal variation, sometimes at multiple scales. The HW procedure given by Eq. (2)∼(4) and the GLS model specified in Eq. (8) both assume a season term with period d. In our experiments, for the daily page view series `Auton` and `Wang`, $d = 7$ (a week), while for the hourly series `citeseer`, $d = 24$ (a day). As mentioned previously, the local linear growth term in Eq. (3) is removed in our experiments because it is found not helpful. The smoothing parameters for the level and the season terms in Eq. (2) and (4) are set to $\zeta = 0.5$ and $\delta = 0.25$. Because HW has no embedded mechanism to select these parameters, we do not aggressively tune them and use the same values for all the experiments reported here.

To assess the importance of weekly (or daily) seasonality for forecasting, we compare HW and and its reduced form without the season term. Similarly as the linear growth term, the season term can be deleted by initializing it to zero and setting its corresponding smoothing parameter δ in Eq. (4) to zero. The reduced HW procedure without the local linear growth and season terms is essentially Exponential Smoothing (ES) (Chatfield, 2004). Figure 4(a) compares the prediction performance in terms of R_e and R_e^{adj} for the three series by HW and ES. Results for two versions of HW, with and without treating impulses, are provided. The comparison of the two versions will be discussed shortly. Results obtained from the two baseline methods MA and MP are also shown. For each of the three series, HW (both versions), which models seasonality, consistently outperforms ES, reflecting the significance of seasonality in these series. We also note that for the `auton` series, R_e is almost twice as large as R_e^{adj} although only 5% of outliers are removed. This dramatic skew of the error rate is caused by the short but strong impulses occurred in this series.

To evaluate the impulse-resistant measure, described in Section 2, we compare HW with and without impulse detection in Figure 4(a). Substantial improvement is achieved for the `Auton` series.



(a) (b)

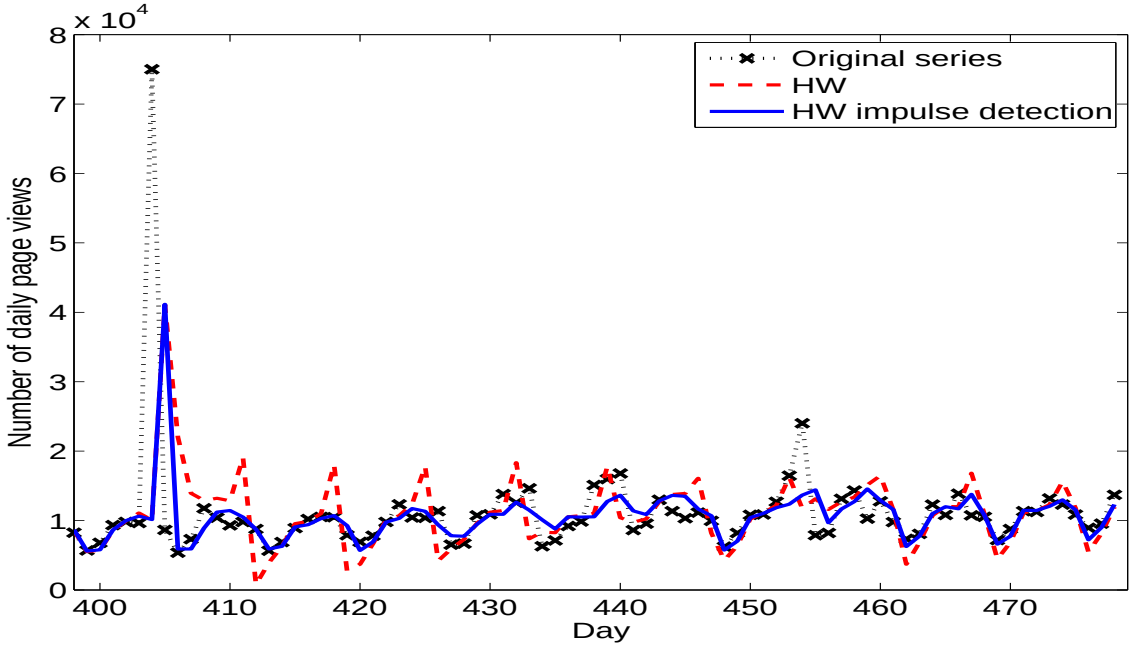
Figure 4: Compare the prediction performance in terms of R_e^{adj} and R_e on the three series `Auton`, `Wang`, and `citeseer` using different methods: HW with or without impulse detection, ES without season, MA, and MP. (a) The error rates. (b) The box plots for the differences of page views at adjacent time units.

The decrease in error rate for the other two series is small, a result of the fact there is no strong impulse in them. To directly demonstrate the magnitude of the impulses, we compute the differences in page view between adjacent time units, $\{x_2 - x_1, x_3 - x_2, ..., x_n - x_{n-1}\}$, and show the box plots for their distributions in Figure 4(b). The stronger impulses in `Auton` are evident from the box plots. Comparing with the other two series, the middle half of the `Auton` data (between the first and third quartiles), indicated by the box in the plot, is much narrower relative to the overall range of the data. In another word, the outliers deviate more severely from the majority mass of the data.

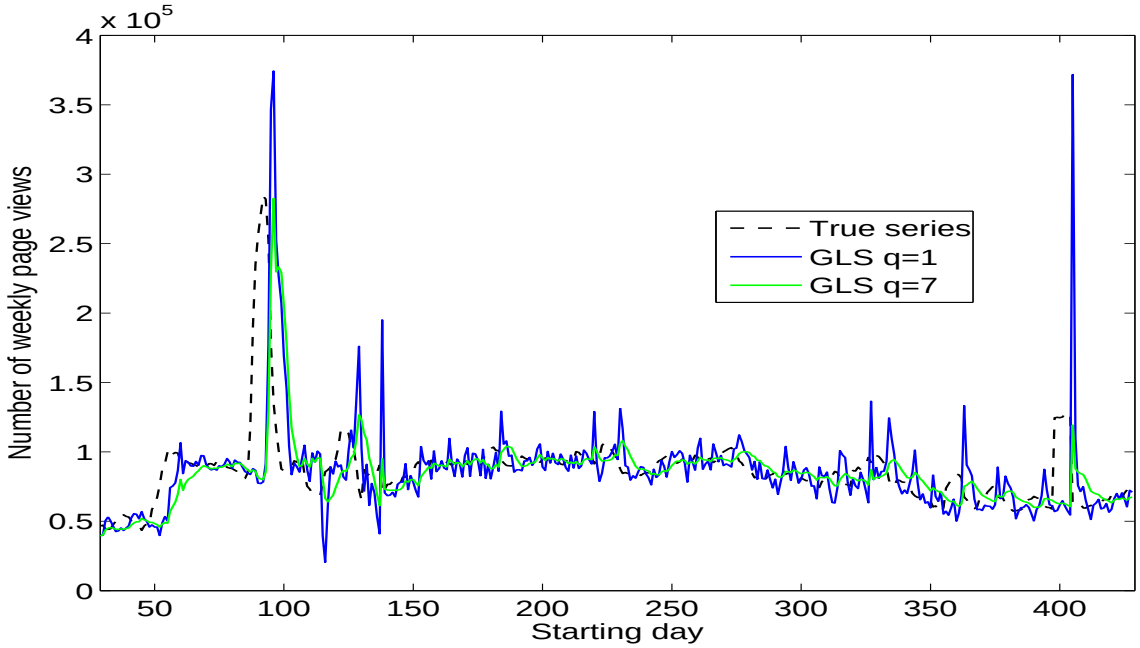
To illustrate the gain from treating impulses, we also show the predicted series for `Auton` in Figure 5(a). For clarity of the plot, only a segment of the series around an impulse is shown. The predicted series by HW with impulse detection returns close to the original series shortly after the impulse, while that without ripples with large errors over several periods afterward. In the sequel, for both HW and GLS, impulse detection is included by default.

Table 1 lists the error rates for the three series using different methods and under different pairs of (h,w), where h is the unit ahead and w is the window size of prediction. We provide the error rate R_e^{adj} in addition to R_e to show the performance on impulse excluded portion of the series. For `Auton` and `Wang`, $(h,w) = (1,1),(1,7),(1,28)$. For `citeseer`, $(h,w) = (1,1),(1,12),(1,24)$. For the GLS model, a range of values for the smooth parameter q are tested. As shown by the table, when $(h,w) = (1,1)$, the performance of HW and that of GLS at the best q are close. When predicting multiple time units, for example, $w = 7,28$ or $w = 12,24$, GLS with $q > 1$ achieves better accuracy. For `Wang` and `citeseer`, at every increased w, the lowest error rates are obtained by GLS with an increased q. This supports the heuristic that when predicting for a more distant time, smoother prediction is preferred to reduce the influence of local fluctuations.

We compare the predicted series for `Auton` by GLS with $q = 1$ and $q = 7$ in Figure 5(b). Here, the unit ahead $h = 1$, and the window size $w = 7$. The fluctuation of the predicted series obtained



(a)



(b)

Figure 5: Compare predicted series for `Auton`: (a) Results obtained by HW with and without impulse detection. The unit ahead h and window size w of prediction are 1; (b) Results obtained by GLS with $q = 1$ and $q = 7$. The unit ahead is 1, and window size is 7 (a week).

Error rate R_e (%)	HW	GLS q=1	GLS q=3	GLS q=7	GLS q=14
Auton: 1 Day	**38.60**	41.46	40.05	41.10	41.74
7 Days	34.52	36.78	34.70	30.33	**28.41**
28 Days	32.34	34.63	32.60	25.80	**21.93**
Wang: 1 Day	26.99	**26.19**	26.24	26.51	26.77
7 Days	19.95	16.19	**16.09**	16.27	16.48
28 Days	21.11	16.44	16.30	16.31	**16.05**
citeseer: 1 Hour	13.55	**13.18**	14.01	15.00	16.29
12 Hours	15.04	14.63	13.66	**12.96**	13.10
24 Hours	15.47	15.87	14.88	14.00	**13.80**

Error rate R_e^{adj} (%)	HW	GLS q=1	GLS q=3	GLS q=7	GLS q=14
Auton: 1 Day	16.76	18.02	17.45	**16.53**	18.14
7 Days	15.60	15.63	14.55	**12.94**	13.45
28 Days	17.32	17.49	16.68	**15.03**	15.31
Wang: 1 Day	20.77	**20.41**	20.64	20.98	20.99
7 Days	16.29	13.52	**13.38**	13.44	13.55
28 Days	16.83	13.65	13.49	13.45	**13.26**
citeseer: 1 Hour	8.80	**8.17**	8.95	10.38	12.16
12 Hours	12.53	10.74	**10.70**	10.97	11.45
24 Hours	12.98	12.14	11.98	11.89	**11.82**

Table 1: The prediction error rates R_e and R_e^{adj} for the three series Auton, Wang, and citeseer obtained by several methods. The window size of prediction takes multiple values, while the unit ahead is always 1. HW and the GLS model with several values of q are compared.

by $q = 1$ is more volatile than that by $q = 7$. The volatility of the predicted series by $q = 7$ is much closer to that of the true series. As shown in Table 1, the error rate R_e^{adj} achieved by $q = 7$ is 12.94%, while that by $q = 1$ is 15.63%.

Based on the GLS model, the variance of x_t conditioned on the past $\{x_1, ..., x_{t-1}\}$ can be computed. The equations for the conditional mean $E(x_t \mid x_1, ..., x_{t-1})$ (i.e., the predicted value) and variance $Var(x_t \mid x_1, ..., x_{t-1})$ are given in (19). Since the conditional distribution of x_t is Gaussian, we can thus calculate a confidence band for the predicted series, which may be desired in certain applications to assess the potential deviation of the true values. Figure 6 shows the 95% confidence band for citeseer with $(h, w) = (1, 1)$. The confidence band covers nearly the entire original series.

GLS is more costly in computation than HW. We conduct the experiments using Matlab codes on 2.4GHz Dell computer with Linux OS. At $(h, w) = (1, 28)$ for Auton and Wang, and $(h, w) = (1, 24)$ for citeseer, the average user running time for sequential prediction along the whole series is respectively 0.51, 56.38, 65.87, 72.10, and 86.76 seconds for HW, and GLS at $q = 1, 3, 7, 14$. In our experiments, the GLS models are re-estimated after every $4d$ units, where d is the period. The computation in GLS is mainly spent on estimating the models and varies negligibly for different pairs of (h, w).

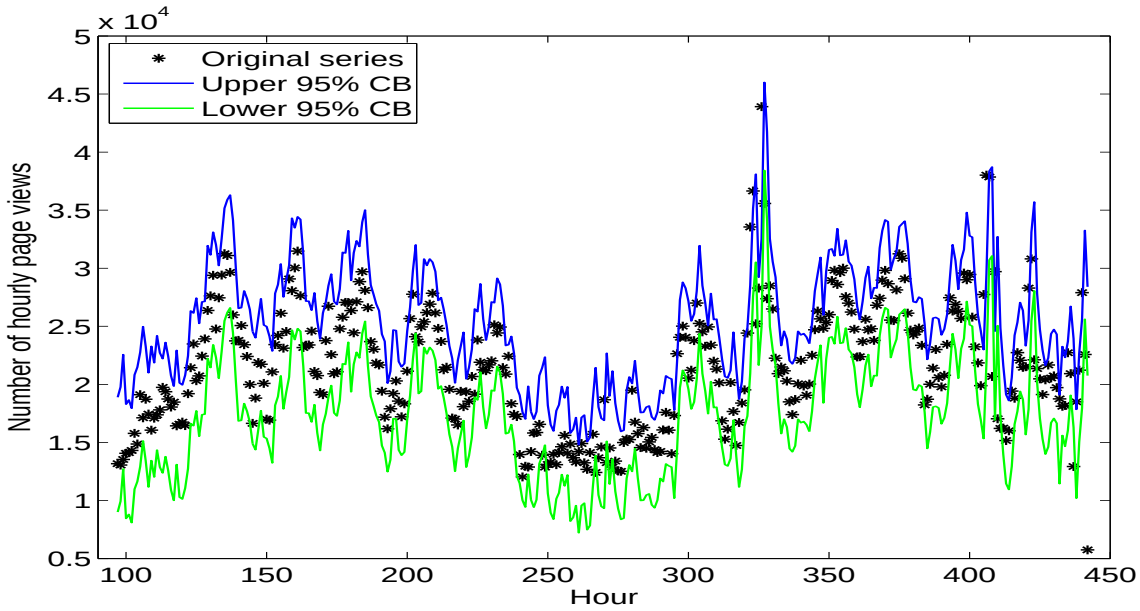


Figure 6: The predicted 95% confidence band for `citeseer` obtained by GLS with $q = 1$. The unit ahead h and window size w are 1.

5.3.2 LONG-TERM PREDICTION—A COMPREHENSIVE STUDY

We now examine the performance of ESSF based on the `g-trends` series. The first three series are acquired by the search phrases `amazon`, `Renoir` (French impressionism artist), and `greenhouse effect`, which will be used as the names for the series in the sequel. A comprehensive study with detailed results is first presented using these three series. Then, we expand the experiments to twenty additional `g-trends` series and present results on prediction accuracy and computational speed.

The original series of `amazon`, `Renoir`, and `greenhouse effect` averaged weekly are shown in Figure 7(a). Due to the weekly season, without averaging, the original series are too wiggly for clear presentation. Figure 7(c) and (d) show the yearly season templates extracted by ESSF from year 2004 and 2005 with smoothing parameter $\lambda = 0$, 1000 respectively. As expected, at $\lambda = 1000$, the yearly seasons are much smoother than those obtained at $\lambda = 0$, especially for the series `Renoir` and `greenhouse effect`. Figure 7(b) shows the scaling factors of the yearly seasons obtained by applying ESSF to the entire four years.

We compare the prediction obtained by ESSF with HW and the MA approach as a baseline. For ESSF, we test both $\lambda = 0$ and 1000, and its fast version with moving average window size 15. Prediction error rates are computed for the unit ahead h ranging from 1 to 180 days. We fix the prediction window size $w = 1$.

The error rates R_e obtained by the methods are compared in Figure 8(a), (b), (c) for `amazon`, `Renoir`, `greenhouse effect` respectively. Comparing with the other methods, the difference in the performance of HW and MA is marginal. When the unit ahead h is small, HW outperforms MA, but the advantage diminishes when h is large. For `Renoir` and `greenhouse effect`, HW becomes even inferior to MA when h is roughly above 60. ESSF with $\lambda = 1000$ and ESSF Fast perform

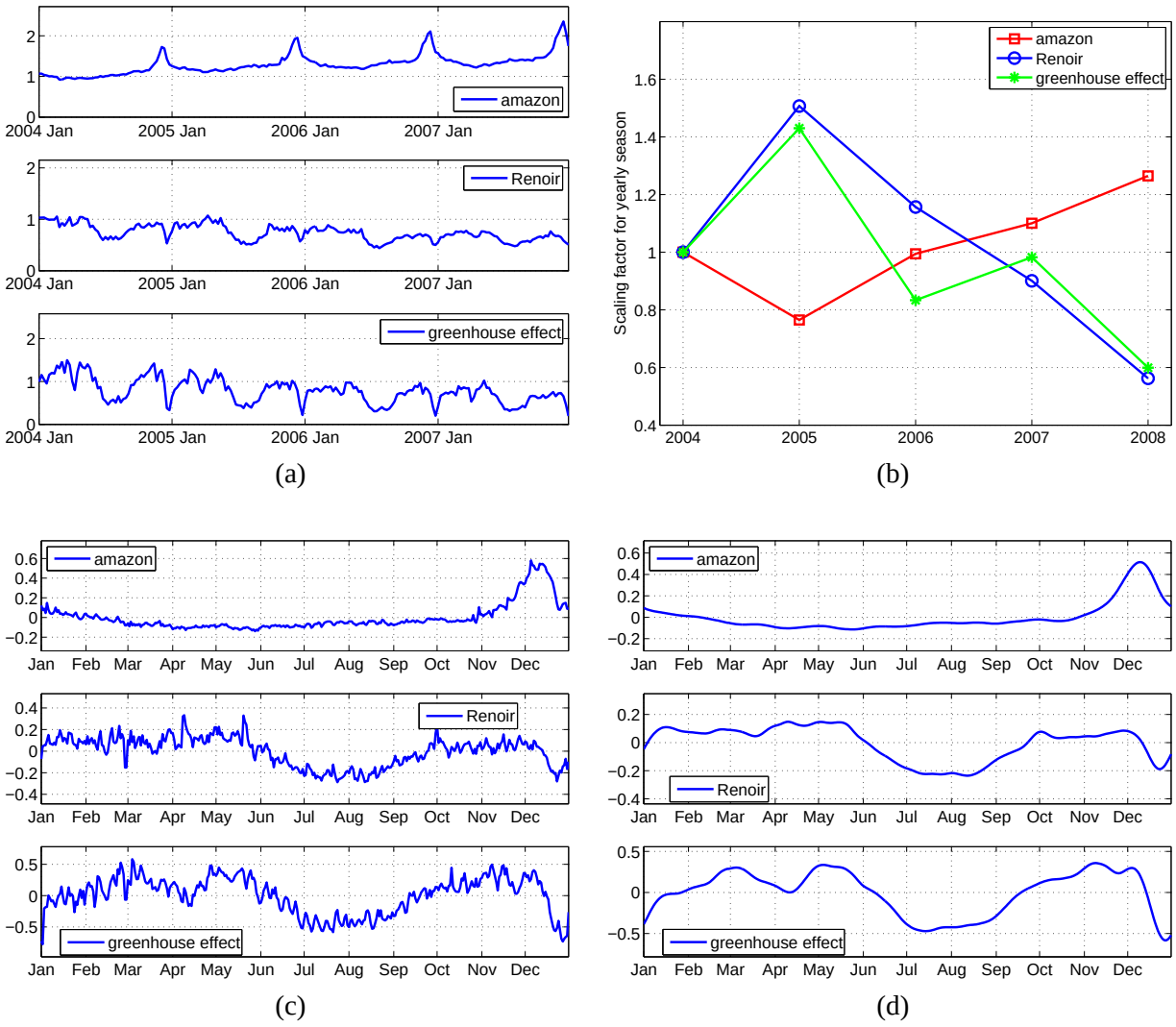


Figure 7: Extract the yearly seasons by ESSF for the `g-trends` series `amazon`, `Renoir`, and `greenhouse effect`: (a) The weekly averaged original series; (b) The scaling factor for the yearly season; (c) The yearly season extracted without smoothing at $\lambda = 0$; (d) The yearly season extracted with smoothing at $\lambda = 1000$.

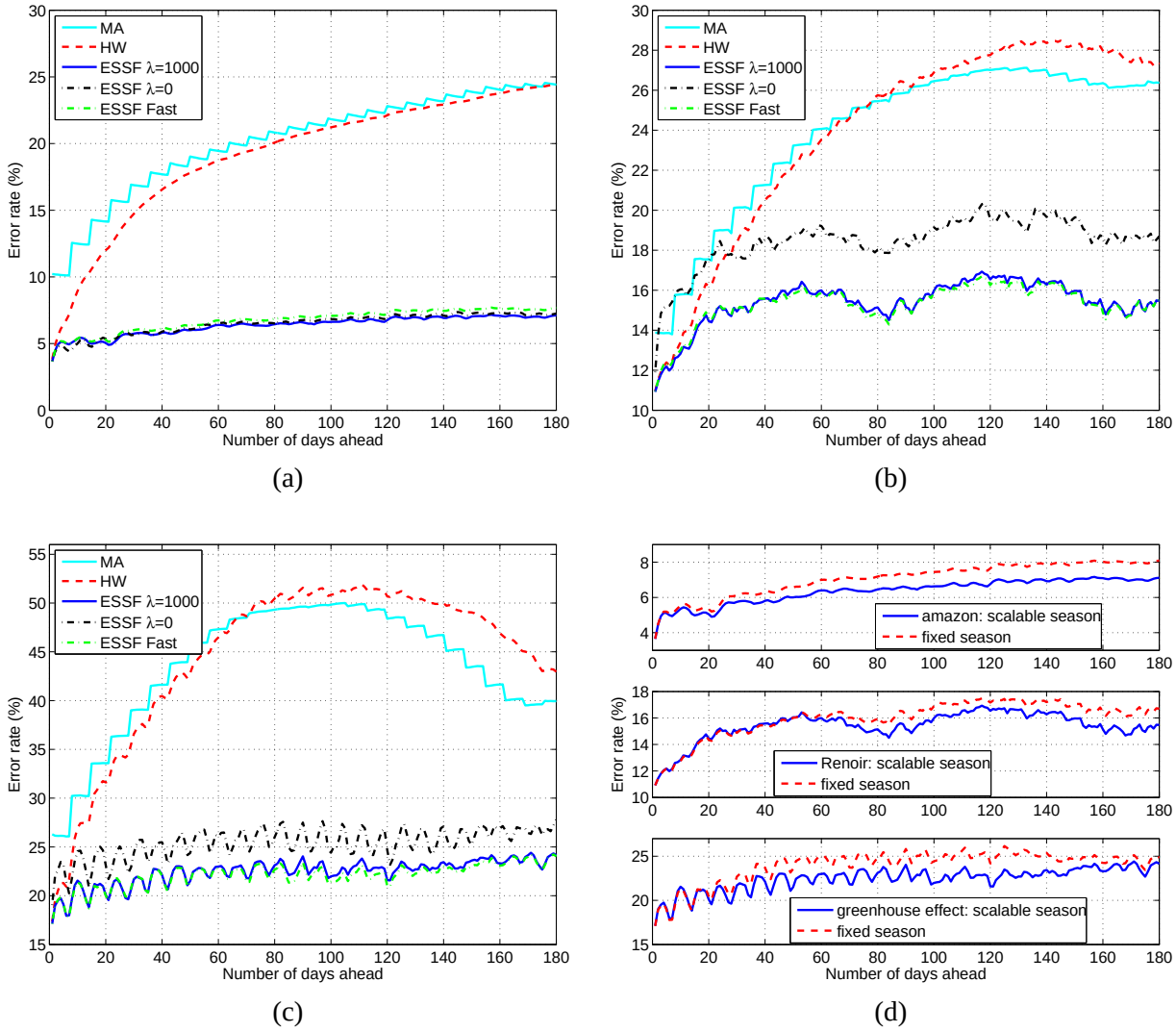
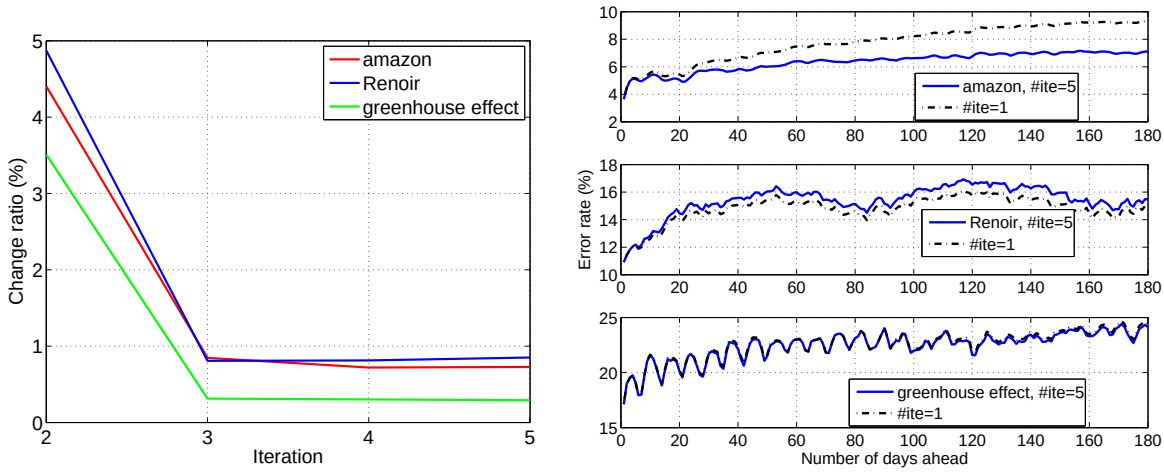


Figure 8: Compare prediction error rates R_e for three `g-trends` series using several methods. Prediction is performed for the unit ahead h ranging from 1 to 180, and a fixed the window size $w = 1$. Error rates obtained by MA, HW, ESSF with $\lambda = 1000$, 0, and the fast version of ESSF with moving average window size 15, are shown for the three series (a) `amazon`, (b) `Renoir`, (c) `greenhouse effect` respectively. The yearly season in ESSF is scalable. (d) Error rates obtained for the three series by ESSF, with $\lambda = 1000$, assuming a scalable yearly season versus fixed season.



(a) (b)

Figure 9: The effect of the number of iterations in the ESSF algorithm: (a) The change ratio in the extracted yearly season over the iterations; (b) Compare the error rates R_e obtained by ESSF with 1 iteration and 5 iterations respectively.

nearly the same, both achieving error rates consistently lower than those by HW. The gap between the error rates of ESSF and HW widens quickly with an increasing h. In general, when h increases, the prediction is harder, and hence the error rate tends to increase. The increase is substantially slower for ESSF than HW and MA. ESSF with $\lambda = 0$ performs considerably worse than $\lambda = 1000$ for `Renoir` and `greenhouse effect`, and closely for `amazon`. This demonstrates the advantage of imposing smoothness on the yearly season. We will study more thoroughly the effect of λ on prediction accuracy shortly.

Next, we experiment with ESSF under various setups and demonstrate the advantages of several algorithmic choices. First, recall that the fitting of the yearly season and the long-range trend is repeated multiple times, as described in Section 4.1. To study the effect of the number of iterations, we plot in Figure 9(a) the ratio of change in the yearly season after each iteration, as given by Eq. (10). For all the three series, the most prominent change occurs between iteration 1 and 2 and falls below 1% for any later iterations. We also compare the prediction error rates for $h = 1, \ldots, 180$ achieved by using only 1 iteration (essentially no iteration) versus 5 iterations. The results for the three series are plotted in Figure 9(b). The most obvious difference is with `amazon` for large h. At $h = 180$, the error rate obtained by 5 iterations is about 2% lower than by 1 iteration. On the other hand, even with only 1 iteration, the error rate at $h = 180$ is below 10%, much lower than the nearly 25% error rate obtained by HW or MA. For `greenhouse effect`, the difference is almost imperceptible.

In ESSF, the yearly season is not assumed simply as a periodic series. Instead, it can scale differently over the years based on the season template. To evaluate the gain, we compare ESSF with scalable yearly seasons versus fixed seasons. Here, the fixed season can be thought of as a special case of the scalable season with all the scaling parameters set to 1, or equivalently, the yearly season is the plain repeat of the season template. Figure 8(d) compares the error rates under

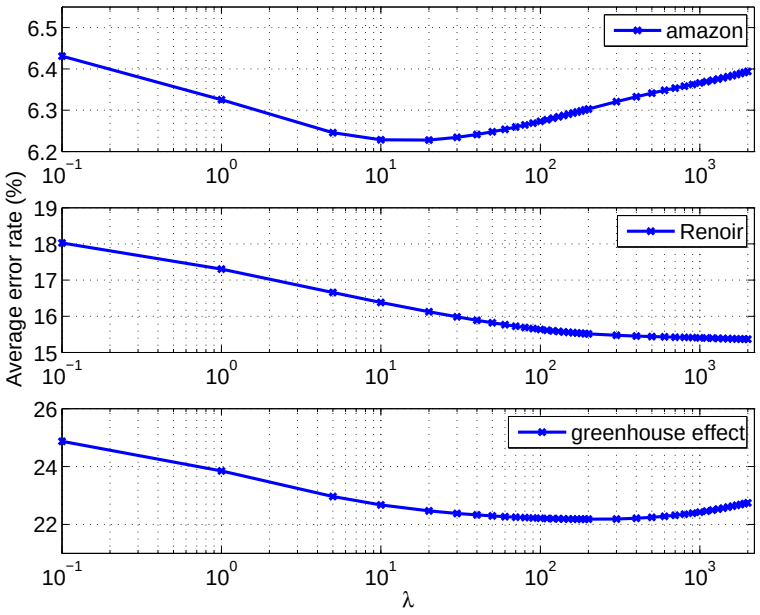


Figure 10: The effect of the smoothing parameter λ in ESSF on prediction accuracy. At each λ, the average of error rates R_e across the unit ahead $h = 1, ..., 180$ are shown.

the two schemes for the three series. Better performance is achieved by allowing scalable yearly seasons for all the three series. The advantage is more substantial when predicting the distant future.

To examine the sensitivity of the prediction accuracy to the smoothing parameter λ, we vary λ from 0.1 to 2000, and compute the error rates for $h = 1, ..., 180$. For concise illustration, we present the average of the error rates across h. Note that the results of $\lambda = 0, 1000$ at every h are shown in Figure 8, where $\lambda = 0$ is inferior. The variation of the average error rates with respect to λ (in log scale) is shown in Figure 10. For `amazon`, the error rates with different λ's lie in the narrow range of $[6.2\%, 6.45\%]$, while for `Renoir` and `greenhouse effect`, the range is wider, roughly $[15\%, 18\%]$ and $[22\%, 25\%]$ respectively. For all the three series, the decrease of the error rate is most steep when λ increases from 0.1 to 10. For $\lambda > 10$ and as large as 2000, the change in error rate is minor, indicating that the prediction performance is not sensitive to λ as long as it is not too small.

5.3.3 LONG-TERM PREDICTION—EXTENDED STUDY ON TWENTY TREND SERIES

We collect another twenty `g-trends` series with query phrases and corresponding series ID listed in Table 2. The error rates R_e achieved by the four methods: MA, HW, ESSF with $\lambda = 1000$, and ESSF Fast, over the twenty series are compared in Figure 11. The four plots in this figure each show results for predicting a single day in advance of h days, with $h = 1, 30, 60, 90$ respectively. For most series, MA is inferior to HW at every h. However, when h increases, the margin of HW over MA decreases. At $h = 1$, HW performs similarly as ESSF and ESSF Fast. At $h = 30, 60, 90$, both versions of ESSF, which achieve similar error rates between themselves, outperform HW.

To assess the predictability of the series, we compute the variation rates at $h = 1, 30, 60, 90$, shown in Figure 12(a). The variation rate at h is defined as $\sqrt{Var(x_{t+h} - x_t)} / \sqrt{Var(x_t)}$, where

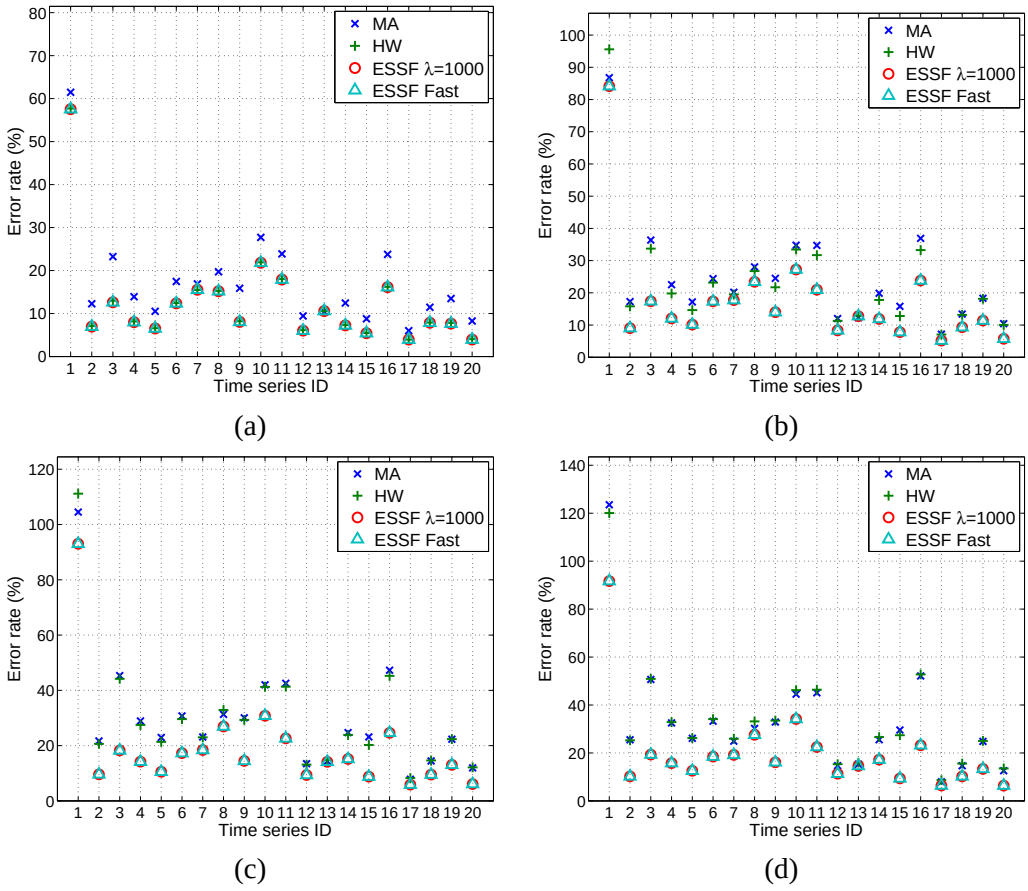


Figure 11: Compare the error rates by MA, HW, ESSF with $\lambda = 1000$, and ESSF Fast for twenty `g-trends` series. (a)-(d): The unit ahead $h = 1, 30, 60, 90$.

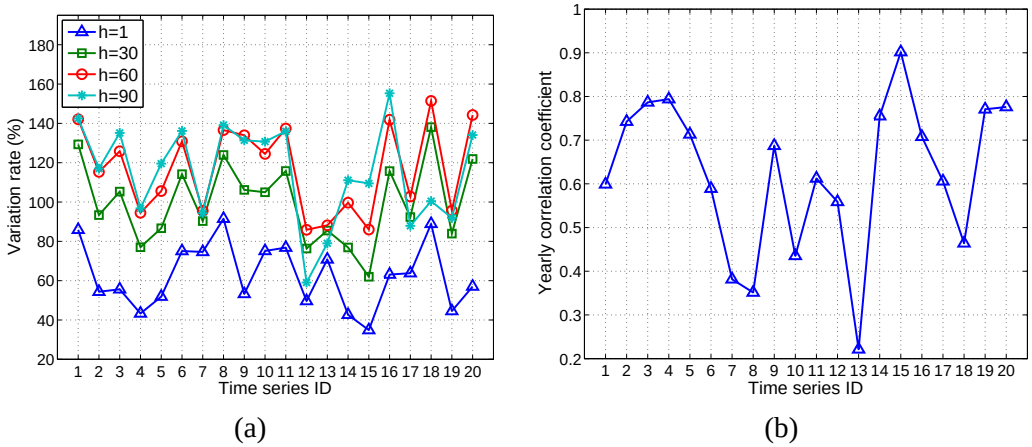


Figure 12: Predictability of twenty `g-trends` series. (a) The variation rates at $h = 1, 30, 60, 90$; (b) The average serial correlation coefficient between adjacent years.

ID	Query phrase	ID	Query phrase
1	American idol	11	human population
2	Anthropology	12	information technology
3	Aristotle	13	martial art
4	Art history	14	Monet
5	Beethoven	15	National park
6	Confucius	16	NBA
7	Cosmology	17	photography
8	cure cancer	18	public health
9	democracy	19	Shakespeare
10	financial crisis	20	Yoga

Table 2: The query phrases for twenty g-trends series and their IDs.

$Var(\cdot)$ denotes the serial variance. This rate is the ratio between the standard deviation of the change in page view h time units apart and that of the original series. A low variation rate indicates the series is less volatile and hence likely to be easier to predict. For example, Art history (ID 4) and National park (ID 15) have the lowest variation rates at $h = 1$, and they both yield relatively low prediction error rates, as shown by Figure 11. We also compute the variation rates for the page views of Web sites Auton, Wang, and citeseer at $h = 1$. They are respectively 100.0%, 88.1%, and 48.2%. This shows that the volatility of page views at these Web sites is in a similar range as that of the g-trends series.

In addition to the variation rate, the yearly correlation of the time series also indicates the potential for accurate prediction. For each of the twenty g-trends series, we compute the average of the correlation coefficients between segments of the series in adjacent years (i.e., 2004/05, 05/06, 06/07). Figure 12(b) shows the results. A series with high yearly correlation tends to benefit more in prediction from the yearly season extraction of ESSF. For instance, martial art (ID 13) has relatively low yearly correlation. The four prediction methods perform nearly the same for this series. In contrast, for NBA (ID 16) and democracy (ID 9), which have high yearly correlation, ESSF achieves substantially better prediction accuracy than HW and MA at all the values of h.

To compare the computational load of the prediction algorithms, we acquire the average user running time over the twenty g-trends series for one day ahead ($h = 1$) prediction at all the days in the last two years, 2006 and 2007. Again, we use Matlab codes on 2.4GHz Dell computer with Linux OS. The average time is respectively 0.11, 0.32, 0.59, and 3.77 seconds for MA, HW, ESSF Fast, and ESSF with $\lambda = 1000$.

6. Discussion and Conclusions

We have so far focused on extracting the trend and season parts of a time series using either HW or GLS, and have not considered predicting the noise part, as given in Eq. (1). We have argued that the variation in Web page view series is dominated by that of the trend and season. To quantitatively assess the potential gain from modeling and predicting the noise term in HW, we fit AR models to the noise. Specifically, we compute the level L_t and the season I_t by HW and let the noise $N_t = x_t - L_t - I_t$. We then fit AR models of order p to the noise series using the Yule-Walker estimation (Brockwell and Davis, 2002). We let p range from 1 to 10 and select an order p by the large-sample

motivated method described in Brockwell and Davis (1991, 2002). The fitted AR models are used to predict the noise, and the predicted noise is added to the forecasting value by HW. Suppose we want to predict x_{t+1} based on $\{x_1, x_2, ..., x_t\}$. The formula given by HW is $\hat{x}_{t+1} = L_t + I_{t+1-d}$. The predicted noise at $t+1$ given by the AR model is $\hat{N}_{t+1} = \hat{\phi}_1 N_t + \hat{\phi}_2 N_{t-1} + \cdots + \hat{\phi}_p N_{t-p+1}$, where $\hat{\phi}_j$, $j = 1, ..., p$, are estimated parameters in the AR model. We then adjust the prediction of HW by $\hat{x}_{t+1} = L_t + I_{t+1-d} + \hat{N}_{t+1}$.

In our experiments, the order of the AR model chosen for each of the three series `Auton`, `Wang`, and `citeseer` is 5, 6, 9 respectively. The error rates R_e^{adj} obtained for `Auton`, `Wang`, and `citeseer` are 17.16%, 20.30%, and 8.45%. As listed in Table 1, the error rates obtained by HW are 16.76%, 20.77%, and 8.80%. We see that the error rates for `Wang` and `citeseer` are improved via noise prediction, but that for `Auton` is degraded. For every series, the difference is insignificant. This shows that the gain from predicting the noise series is minor if positive at all. It is out of the scope of this paper to investigate more sophisticated models for the noise series. We consider it an interesting direction for future work.

To conclude, we have examined multiple approaches to Web page view forecasting. For short-term prediction, the HW procedure and the GLS state space model are investigated. It is shown that seasonal effect is important for page view forecasting. We developed a method to identify impulses and to reduce the decrease in prediction accuracy caused by them. The HW procedure, although computationally simple, performs closely to the GLS approach for predicting a small number of time units ahead. For predicting moderately distant future, the GLS model with smoother level terms tends to perform better. We developed the ESSF algorithm to extract global trend and scalable long-range season with smoothness regularization. It is shown that for predicting the distant future, ESSF outperforms HW significantly.

Acknowledgments

We thank Michael Baysek, C. Lee Giles, and James Wang for providing the logs of the Auton Lab, citeseer, and the Wang Lab. We also thank Artem Boytsov and Eyal Molad for helping us access the Google trends series, Robbie Sedgewick for suggestions on writing, and the reviewers for many insightful and constructive comments.

Appendix A. Algorithms for the State Space Model

Several major issues can be studied under the state space model:

1. Filtering: obtain the conditional distribution of α_{t+1} given X_t for $t = 1, ..., n$ where $X_t = \{x_1, ..., x_t\}$. If we consider α_t as the "true" signal, filtering is to discover the signal on the fly.

2. State smoothing: estimate α_t, $t = 1, ..., n$, given the entire series $\{x_1, ..., x_n\}$. This is to discover the signal in a batch mode.

3. Disturbance smoothing: estimate the disturbances $\hat{\varepsilon}_t = E(\varepsilon_t | y_1, ..., y_n)$, $\hat{\eta}_t = E(\eta_t | y_1, ..., y_n)$. The estimation can be used to estimate the covariance matrices of the disturbances.

4. Forecasting: given $\{x_1, ..., x_n\}$, forecast x_{n+j} for $j = 1, ..., J$.

5. Perform the Maximum Likelihood (ML) estimation for the parameters based on $\{x_1, ..., x_n\}$.

The computation methods involved in the above problems are tightly related. Filtering is conducted by forward recursion, while state smoothing is achieved by combining the forward recursion with a backward recursion. Disturbance smoothing can be easily performed based on the results of filtering and smoothing. ML estimation in turn relies on the result of disturbance smoothing. Next, we present the algorithms to solve the above problems.

A.1 Filtering and Smoothing

Recall the SSM described by Eq. (6)

$$\begin{aligned} x_t &= \mathbf{Z}_t\alpha_t + \varepsilon_t\,, \qquad \varepsilon_t \sim N(0,\mathbf{H}_t), \\ \alpha_{t+1} &= \mathbf{T}_t\alpha_t + \mathbf{R}_t\eta_t\,, \quad \eta_t \sim N(0,\mathbf{Q}_t)\,, \quad t = 1,\ldots,n, \\ &\qquad\qquad \alpha_1 \sim N(a_1,\mathbf{P}_1). \end{aligned}$$

Suppose the goal is filtering, that is, to obtain the conditional distribution of α_{t+1} given X_t for $t = 1,\ldots,n$ where $X_t = \{x_1,\ldots,x_t\}$. Since the joint distribution is Gaussian, the conditional distribution is also Gaussian and hence is uniquely determined by the mean and covariance matrix. Moreover, note that x_{t+1} is conditionally independent of X_t given α_{t+1}. Let $a_t = E(\alpha_t \mid X_{t-1})$ and $\mathbf{P}_t = Var(\alpha_t \mid X_{t-1})$. Then $\alpha_t \mid X_{t-1} \sim N(a_t,\mathbf{P}_t)$. It can be shown that a_{t+1} and $\mathbf{P}_{t+1}$ can be computed recursively from a_t, $\mathbf{P}_t$.

Let the one-step forecast error of x_t given X_{t-1} be v_t and the variance of v_t be $\mathbf{F}_t$:

$$\begin{aligned} v_t &= x_t - E(x_t \mid X_{t-1}) = x_t - \mathbf{Z}_t a_t, \\ \mathbf{F}_t &= Var(v_t) = \mathbf{Z}_t\mathbf{P}_t\mathbf{Z}_t^t + \mathbf{H}_t. \end{aligned}$$

For clarity, also define

$$\begin{aligned} \mathbf{K}_t &= \mathbf{T}_t\mathbf{P}_t\mathbf{Z}_t^t\mathbf{F}_t^{-1}, \\ \mathbf{L}_t &= \mathbf{T}_t - \mathbf{K}_t\mathbf{Z}_t. \end{aligned}$$

Then a_t, $\mathbf{P}_t$, $t = 2,\ldots,n+1$ can be computed recursively by updating v_t, $\mathbf{F}_t$, $\mathbf{K}_t$, $\mathbf{L}_t$, a_{t+1}, $\mathbf{P}_{t+1}$ as follows. It is assumed that a_1 and $\mathbf{P}_1$ are part of the model specification, and hence are known or provided by initialization. Details on initialization are referred to Durbin and Koopman (2001). For $t = 1,2,\ldots,n$,

$$\begin{aligned} v_t &= x_t - \mathbf{Z}_t a_t, \\ \mathbf{F}_t &= \mathbf{Z}_t\mathbf{P}_t\mathbf{Z}_t^t + \mathbf{H}_t, \\ \mathbf{K}_t &= \mathbf{T}_t\mathbf{P}_t\mathbf{Z}_t^t\mathbf{F}_t^{-1}, \\ \mathbf{L}_t &= \mathbf{T}_t - \mathbf{K}_t\mathbf{Z}_t, \\ a_{t+1} &= \mathbf{T}_t a_t + \mathbf{K}_t v_t, \\ \mathbf{P}_{t+1} &= \mathbf{T}_t\mathbf{P}_t\mathbf{L}_t^t + \mathbf{R}_t\mathbf{Q}_t\mathbf{R}_t^t. \end{aligned} \tag{17}$$

The above recursion is called *Kalman filter*. The dimensions for the above matrices are:

$$\begin{array}{ll} v_t & p \times 1\ , \\ \mathbf{F}_t & p \times p\ , \\ \mathbf{K}_t & m \times p\ , \\ \mathbf{L}_t & m \times m\ , \\ a_t & m \times 1\ , \\ \mathbf{P}_t & m \times m\ . \end{array}$$

We are concerned with univariate forecasting here with $p = 1$.

We now consider the *smooth estimation* $\hat{\alpha}_t = E(\alpha_t \mid x_1, x_2, ..., x_{t-1}, x_t, ..., x_n)$. Note its difference from the forward estimation $a_t = E(\alpha_t \mid x_1, x_2, ..., x_{t-1})$. The smooth estimation takes into consideration the series after t. Let the variance of the smooth estimation be $\mathbf{V}_t = Var(\alpha_t \mid x_1, x_2, ..., x_{t-1}, x_t, ..., x_n)$.

We can compute $\hat{\alpha}_t$ and $\mathbf{V}_t$ by the *backwards recursion* specified below. At $t = n$, set $\gamma_n = [0]_{m\times 1}$ and $\mathbf{N}_n = [0]_{m\times m}$. For $t = n, n-1, ..., 1$,

$$\begin{aligned} \gamma_{t-1} &= \mathbf{Z}_t^t \mathbf{F}_t^{-1} v_t + \mathbf{L}_t^t \gamma_t\ , \\ \mathbf{N}_{t-1} &= \mathbf{Z}_t^t \mathbf{F}_t^{-1} \mathbf{Z}_t + \mathbf{L}_t^t \mathbf{N}_t \mathbf{L}_t, \\ \hat{\alpha}_t &= a_t + \mathbf{P}_t \gamma_{t-1}, \\ \mathbf{V}_t &= \mathbf{P}_t - \mathbf{P}_t \mathbf{N}_{t-1} \mathbf{P}_t. \end{aligned} \tag{18}$$

Note that $\mathbf{Z}_t$, $\mathbf{F}_t$, $\mathbf{L}_t$, and $\mathbf{P}_t$ are already acquired by the Kalman filter (17). Eq. (17) and (18) are referred to as *Kalman filter and smoother*. The Kalman filter only involves forward recursion, while the smoother involves both forward and backward recursions.

A.2 Disturbance Smoothing

Let the smoothed disturbances be $\hat{\varepsilon}_t = E(\varepsilon_t | x_1, x_2, ..., x_n)$, $\hat{\eta}_t = E(\eta_t | x_1, x_2, ..., x_n)$. Suppose $\mathbf{F}_t$, $\mathbf{K}_t$, $\mathbf{L}_t$, $t = 1, ..., n$, have been obtained by the Kalman filter, and γ_t, $\mathbf{N}_t$ have been obtained by the Kalman smoother. Then we have

$$\begin{aligned} \hat{\varepsilon}_t &= \mathbf{H}_t(\mathbf{F}_t^{-1} v_t - \mathbf{K}_t^t \gamma_t), \\ Var(\varepsilon_t | x_1, x_2, ..., x_n) &= \mathbf{H}_t - \mathbf{H}_t(\mathbf{F}_t^{-1} + \mathbf{K}_t^t \mathbf{N}_t \mathbf{K}_t)\mathbf{H}_t, \\ \hat{\eta}_t &= \mathbf{Q}_t \mathbf{R}_t^t \gamma_t, \\ Var(\eta_t | x_1, x_2, ..., x_n) &= \mathbf{Q}_t - \mathbf{Q}_t \mathbf{R}_t^t \mathbf{N}_t \mathbf{R}_t \mathbf{Q}_t. \end{aligned}$$

A.3 Forecasting

Now suppose we want to forecast x_{n+j}, $j = 1, ..., J$, given $\{x_1, ..., x_n\}$. Let

$$\begin{aligned} \overline{x}_{n+j} &= E(x_{n+j} \mid x_1, x_2, ..., x_n), \\ \overline{\mathbf{F}}_{n+j} &= Var(x_{n+j} \mid x_1, x_2, ..., x_n)\ . \end{aligned}$$

First, we compute $\overline{a}_{n+j}$ and $\overline{P}_{n+j}$, $j = 1, ..., J$, by forward recursion similar to the Kalman filter in Eq. (17). The slight difference is that when $j = 1, ..., J-1$, set $v_{n+j} = 0$ and $\mathbf{K}_{n+j} = 0$. Specifically, set $\overline{a}_{n+1} = a_{n+1}$, $\overline{P}_{n+1} = \mathbf{P}_{n+1}$. The recursion for $\overline{a}_{n+j+1}$ and $\overline{P}_{n+j+1}$ for $j = 1, ..., J-1$ is:

$$\begin{aligned} \overline{a}_{n+j+1} &= \mathbf{T}_{n+j} \overline{a}_{n+j}, \\ \overline{\mathbf{P}}_{n+j+1} &= \mathbf{T}_{n+j} \overline{\mathbf{P}}_{n+j} \mathbf{T}_{n+j}^t + \mathbf{R}_{n+j} \mathbf{Q}_{n+j} \mathbf{R}_{n+j}^t\ . \end{aligned}$$

Then we forecast

$$\begin{aligned}\overline{x}_{n+j} &= \mathbf{Z}_{n+j}\overline{a}_{n+j},\\ \overline{\mathbf{F}}_{n+j} &= \mathbf{Z}_{n+j}\overline{\mathbf{P}}_{n+j}\mathbf{Z}_{n+j}^{t}+\mathbf{H}_{n+j}\,.\end{aligned}$$

A.4 Maximum Likelihood Estimation

The parameters to be estimated in the SSM are $\mathbf{H}_t$ and $\mathbf{Q}_t$, $t=1,2,...,n$. The EM algorithm is used to obtain the ML estimation. The missing data in EM in this case are the unobservable states α_t, $t=1,...,n$. Denote the parameters to be estimated collectively by ψ and the parameters obtained from the previous iteration by $\tilde{\psi}$. Let $\alpha=\{\alpha_1,\alpha_2,...,\alpha_n\}$ and $X_n=\{x_1,x_2,...,x_n\}$. The update of the EM algorithm comprises two steps:

1. Compute the expectation
$$E_{\tilde{\psi},X_n}[\log p(\alpha,X_n|\psi)]\,.$$

2. Maximize over ψ the above expectation.

It can be shown that

$$\begin{aligned}E_{\tilde{\psi},X_n}[\log p(\alpha,X_n|\psi)] &= \text{constant}-\frac{1}{2}\sum_{t=1}^{n}[\log|\mathbf{H}_t|+\log|\mathbf{Q}_{t-1}|+\\ &\quad tr[(\hat{\varepsilon}_t\hat{\varepsilon}_t^t+Var(\varepsilon_t|X_n))\mathbf{H}_t^{-1}]+\\ &\quad tr[(\hat{\eta}_{t-1}\hat{\eta}_{t-1}^t+Var(\eta_{t-1}|X_n))\mathbf{Q}_{t-1}^{-1}]\mid\psi]\end{aligned}$$

where $\hat{\varepsilon}_t$, $\hat{\eta}_{t-1}$, $Var(\varepsilon_t|X_n)$, and $Var(\eta_{t-1}|X_n)$ are computed by disturbance smoothing under parameter $\tilde{\psi}$. In the special case, when $\mathbf{H}_t=\mathbf{H}$, $\mathbf{Q}_t=\mathbf{Q}$, the maximization can be solved analytically:

$$\begin{aligned}\mathbf{H} &= \frac{\sum_{t=1}^{n}[\hat{\varepsilon}_t\hat{\varepsilon}_t^t+Var(\varepsilon_t|X_n)]}{n},\\ \mathbf{Q} &= \frac{\sum_{t=2}^{n}[\hat{\eta}_{t-1}\hat{\eta}_{t-1}^t+Var(\eta_{t-1}|X_n)]}{n-1}.\end{aligned}$$

The formula can be further simplified if $\mathbf{H}$ and $\mathbf{Q}$ are assumed diagonal. Suppose

$$\begin{aligned}\mathbf{H} &= diag(\sigma_{\varepsilon,1}^2,\sigma_{\varepsilon,2}^2,...,\sigma_{\varepsilon,p}^2),\\ \mathbf{Q} &= diag(\sigma_{\eta,1}^2,\sigma_{\eta,2}^2,...,\sigma_{\eta,r}^2)\,.\end{aligned}$$

Then

$$\begin{aligned}\sigma_{\varepsilon,j}^2 &= \frac{\sum_{t=1}^{n}[\hat{\varepsilon}_{t,j}^2+Var(\varepsilon_{t,j}|X_n)]}{n},\quad j=1,...,p,\\ \sigma_{\eta,j}^2 &= \frac{\sum_{t=2}^{n}[\hat{\eta}_{t-1,j}^2+Var(\eta_{t-1,j}|X_n)]}{n-1},\quad j=1,...,r\,.\end{aligned}$$

Appendix B. The ESSF Algorithm and Its Fast Version

To solve $\min_{\bar{\mathbf{y}},\mathbf{c}} G(\bar{\mathbf{y}},\mathbf{c})$ in Eq. (12), we iteratively optimize over $\bar{\mathbf{y}}$ and $\mathbf{c}$. Given $\mathbf{c}$, $\bar{\mathbf{y}}$ is solved by

$$\mathbf{A}_y \bar{\mathbf{y}} = \mathbf{b}_y$$

where $\mathbf{A}_y$ is a $D \times D$ matrix with non-zero entries:

$$\begin{aligned}
\mathbf{A}_y(j,j) &= \sum_k \alpha_{k,j}^2 + 6\lambda, \quad j = 1,2,...,D, \\
\mathbf{A}_y(j,j-1) &= \mathbf{A}_y(j,j+1) = -4\lambda, \quad j = 1,2,...,D, \\
\mathbf{A}_y(j,j-2) &= \mathbf{A}_y(j,j+2) = \lambda, \quad j = 1,2,...,D,
\end{aligned}$$

and the column vector $\mathbf{b}_y = (\sum_k \alpha_{k,j} z_{k,j})_j$. Recall that $\alpha_{k,j}$ is computed from $\mathbf{c}$ by Eq. (11).

Given $\mathbf{y}$, $\mathbf{c}$ is solved by

$$\mathbf{A}_c \mathbf{c} = \mathbf{b}_c$$

where $\mathbf{A}_c$ is a $K \times K$ matrix. Define $\mathbf{w}_1 = (0, \frac{1}{D}, \frac{2}{D}, ..., \frac{D-1}{D})^t$ and $\mathbf{w}_2 = (1, \frac{D-1}{D}, \frac{D-2}{D}, ..., \frac{1}{D})^t$. Let diagonal matrices $\mathbf{W}_1 = diag(\mathbf{w}_1)$, $\mathbf{W}_2 = diag(\mathbf{w}_2)$. Also define $\mathbf{z}_k = (z_{k,1}, z_{k,2}, ..., z_{k,D})^t$. The non-zero entries of $\mathbf{A}_c$ are:

$$\begin{aligned}
\mathbf{A}_c(k,k) &= (\mathbf{W}_1\bar{\mathbf{y}})^t \mathbf{W}_1 \bar{\mathbf{y}} + (\mathbf{W}_2\bar{\mathbf{y}})^t \mathbf{W}_2 \bar{\mathbf{y}}, \quad k = 1,2,...,K-1, \\
\mathbf{A}_c(K,K) &= (\mathbf{W}_2\bar{\mathbf{y}})^t \mathbf{W}_2 \bar{\mathbf{y}}, \\
\mathbf{A}_c(k,k-1) &= (\mathbf{W}_1\bar{\mathbf{y}})^t \mathbf{W}_2 \bar{\mathbf{y}}, \quad k = 2,3,...,K, \\
\mathbf{A}_c(k,k+1) &= (\mathbf{W}_1\bar{\mathbf{y}})^t \mathbf{W}_2 \bar{\mathbf{y}}, \quad k = 1,2,...,K-1,
\end{aligned}$$

and the column vector $\mathbf{b}_c$ is given by:

$$\begin{aligned}
\mathbf{b}_c(1) &= (\mathbf{W}_1\bar{\mathbf{y}})^t \mathbf{z}_1 + (\mathbf{W}_2\bar{\mathbf{y}})^t \mathbf{z}_2 - (\mathbf{W}_1\bar{\mathbf{y}})^t \mathbf{W}_2 \bar{\mathbf{y}}, \\
\mathbf{b}_c(k) &= (\mathbf{W}_1\bar{\mathbf{y}})^t \mathbf{z}_k + (\mathbf{W}_2\bar{\mathbf{y}})^t \mathbf{z}_{k+1}, \quad k = 2,3,...,K-1, \\
\mathbf{b}_c(K) &= (\mathbf{W}_1\bar{\mathbf{y}})^t \mathbf{z}_1.
\end{aligned}$$

In summary, the ESSF algorithm iterates the following two steps with initialization $\mathbf{c}^{(0)} = \mathbf{1}$. At iteration $p \geq 1$:

1. Given $\mathbf{c}^{(p-1)}$, compute $\mathbf{A}_y$ and $\mathbf{b}_y$. Let $\bar{\mathbf{y}}^{(p)} = \mathbf{A}_y^{-1}\mathbf{b}_y$.
2. Given $\bar{\mathbf{y}}^{(p)}$, compute $\mathbf{A}_c$ and $\mathbf{b}_c$. Let $\mathbf{c}^{(p)} = \mathbf{A}_c^{-1}\mathbf{b}_c$.

For the fast version of ESSF, we need to solve $\min_{\bar{\mathbf{y}},\mathbf{c}} \tilde{G}(\bar{\mathbf{y}},\mathbf{c})$ in Eq. (13). We start with $\mathbf{c}^{(0)} = \mathbf{1}$. Without loss of generality, we fix $c_1 = 1$. At iteration $p \geq 1$:

1. Given $\mathbf{c}^{(p-1)}$, compute
$$\bar{y}_j^{(p)} = \frac{\sum_k c_k^{(p-1)} z_{k,j}}{\| \mathbf{c}^{(p-1)} \|^2}, \quad j = 1,...,D.$$
2. Given $\bar{\mathbf{y}}^{(p)}$, compute
$$c_k^{(p)} = \frac{\sum_j z_{k,j} \bar{y}_j^{(p)}}{\| \bar{\mathbf{y}}^{(p)} \|^2}, \quad k = 1,...,K.$$

References

B. D. O. Anderson and J. B. Moore. *Optimal Filtering*, Prentice-Hall, Englewood Cliffs, New Jersey, 1979.

A. Aussem and F. Murtagh. Web traffic demand forecasting using wavelet-based multiscale decomposition. *International Journal of Intelligent Systems*, 16(2):215-236, 2001.

S. Basu, A. Mukherjee, and S. Klivansky. Time series models for Internet traffic. *INFOCOM '96. Fifteenth Annual Joint Conference of the IEEE Computer Societies, Networking the Next Generation*, 611-620, 1996.

G. E. P. Box and G. M. Jenkins. *Time-Series Analysis, Forecasting and Control*, San Francisco: Holden-Day, 1970.

P. J. Brockwell and R. A. Davis. *Time Series: Theory and Methods*, 2nd Edition, Springer-Verlag, New York, 1991.

P. J. Brockwell and R. A. Davis. *Introduction to Time Series and Forecasting*, 2nd Edition, Springer Science+Business Media, Inc., New York, 2002.

C. Chatfield. *The Analysis of Time Series*, Chapman & Hall/CRC, New York, 2004.

J. Durbin and S. J. Koopman. *Time Series Analysis by State Space Methods*, Oxford University Press Inc., New York, 2001.

M. Grossglauser and J.-C. Bolot. On the relevance of long-range dependence in network traffic. *IEEE/ACM Transactions Networking*, 7(5):629-640, 1999.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of Basic Engineering, Series D*, 82:35-45, 1960.

A. Khotanzad and N. Sadek. Multi-scale high-speed network traffic prediction using combination of neural networks. *Proc. Int. Joint Conf. Neural Networks*, 2:1071-1075, July 2003.

A. M. Odlyzko. Internet traffic growth: sources and implications. *Optical Transmission Systems and Equipment for WDM Networking II*, B. B. Dingel, W. Weiershausen, A. K. Dutta, and K.-I. Sato, eds.,*Proc. SPIE*, 5247:1-15, 2003.

K. Papagiannaki, N. Taft, Z.-L. Zhang, and C. Diot. Long-term forecasting of Internet backbone traffic. *IEEE Trans. Neural Networks*, 16(5):1110-1124, 2005.

K. Park and W. Willinger. *Self-Similar Network Traffic and Performance Evaluation*, John Wiley & Sons, Inc., 2000.

A. P. Sage and J. L. Melsa. *Estimation Theory with Applications to Communication and Control*, McGraw Hill, New York, 1971.

A. Sang and S. Li. A predictability analysis of network traffic. *Computer Networks*, 39(4):329-345, 2002.

W. W. S. Wei. *Time Series Analysis, Univariate and Multivariate Methods*, 2nd Edition, Pearson Education, Inc., 2006.

C. You and K. Chandra. Time series models for Internet data traffic. *Proc. 24th Annual IEEE Int. Conf. Local Computer Networks (LCN'99)*, 164, 1999.