

TIME SERIES FORECASTING VIA NOISY CHANNEL REVERSAL

Pejman Khadivi, Prithwish Chakraborty, Ravi Tandon, Naren Ramakrishnan

Discovery Analytics Center, Department of Computer Science, Virginia Tech, Blacksburg, VA

ABSTRACT

Developing a precise understanding of the dynamic behavior of time series is crucial for the success of forecasting techniques. We introduce a novel communication-theoretic framework for modeling and forecasting time series. In particular, the observed time series is modeled as the output of a noisy communication system with the input as the future values of time series. We use a data-driven probabilistic approach to estimate the unknown parameters of the system which in turn is used for forecasting. We also develop an extension of the proposed framework together with a filtering algorithm to account for the noise and heterogeneity in the quality of time series. Experimental results demonstrate the effectiveness of this approach.

1. INTRODUCTION

With recent advances in data analytics and machine learning, scientists and policy makers are now able to use time-series analysis in important applications such as epidemiological [1] and financial forecasting [2]. Classical time-series methods such as the autoregressive model (AR), ARMA, and ARIMA are widely popular [3]. When the modeled process is highly nonlinear, methods such as Gaussian and Dirichlet processes are more popular [4].

In traditional time series analysis, it is assumed that the behavior of time series in the future will follow the same dynamics as in the past [5]. However, in real-world time series noise and the unfolding dynamic behavior of data can have a significant effect on the accuracy of predictions.

In this paper, we propose a framework based on the similarities of time series regression and communication systems, to address the aforementioned issues by making explicit assumptions about the noise and behavior of time series. We will show that noise and deviations from the predetermined behavior of time series is similar to the noise of a communication channel and thus, use communication-theoretic methods to address this deviation to some level. We model the noisy communication channel between the future and past values of time series as an additive Gaussian noise channel. Channel parameters are estimated from the observed data together with an out-of-band filtering algorithm to boost the signal-to-noise (SNR) ratio. The estimates of this noisy channel process are subsequently used for forecasting the future time series data via regression. We perform extensive experiments with real-world and synthetic datasets to study the performance of the proposed approach. Results show that the proposed model can efficiently improve the forecasting performance in terms

of accuracy. Thus, our contributions are:

- Developing a framework to directly address the noise, instabilities, and deviations of time series behavior from its known behavior using communication theory concepts.
- Improving the accuracy of regression through a frequency domain out-of-band noise filtering.
- Developing analytic models to estimate the amount of noise in a specific time series.

2. PROBLEM FORMULATION

Let us assume that we have a time series y_1, y_2, \dots , which in the rest of the paper, we denote by Y . We denote the sequence of samples between time points t_1 and t_2 by $\mathbf{Y}_{t_1}^{t_2}$, i.e. $\mathbf{Y}_{t_1}^{t_2} = [y_{t_1} \dots y_{t_2}]^T$. If we have observed \mathbf{Y}_1^{t-k} and we desire to forecast the value of Y at time t , y_t , we have:

$$\hat{y}_t = \varphi(\mathbf{Y}_1^{t-k}) \quad (1)$$

where $\varphi(\cdot)$ is the estimation function and \hat{y}_t is the estimated value of y_t . As an example, in the classic AR(m) approach, y_t is forecasted based on the following equation:

$$\hat{y}_t = \beta + \sum_{j=0}^{m-1} \alpha_j y_{t-k-j} = \beta + \mathbf{A}^T \mathbf{Y}_{t-k-(m-1)}^{t-k} \quad (2)$$

where m is the order of autoregressive model, α_j 's and β are the regression weights, and $\mathbf{A} = [\alpha_{m-1} \dots \alpha_0]^T$.

In many practical applications, due to inaccuracies in data acquisition, forecasts should be made using unstable and noisy time series. Most of the times, this is more serious for the newer values since older samples may be corrected and stabilized over time. This non-uniform imperfection of the data causes heterogeneity in the quality of time series. Regression is highly sensitive to outliers [6] and hence, this noise is considered as a source of estimation error. To model heterogeneity, we use \dot{y}_j to denote a stable sample at time j and $\dot{\mathbf{Y}}_{t_1}^{t_2}$ to denote all the stable samples between t_1 and t_2 . Each sample of time series Y can be modeled as follows:

$$y_j = \dot{y}_j + \nu_j \quad (3)$$

where \dot{y}_j is the stable part of the sample and ν_j is a Gaussian random variable with zero mean and variance ζ_j^2 , $\nu_j \sim$

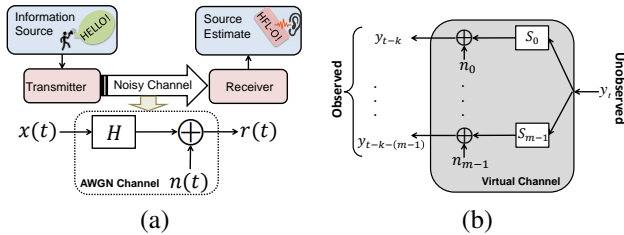


Fig. 1. (a) General model of a communication channel, (b) Multiple-output virtual channel model.

$\mathcal{N}(0, \zeta_j^2)$. Since newer values suffer from more instability, we can assume that if $i < j$ then $\zeta_i^2 \leq \zeta_j^2$. Note that stable data samples are modeled as in Eq. 3 with $\zeta_j = 0$.

Similar to time series analysis, communication systems suffer from imperfect noisy channels and hence, the received message may be different from the sent one. The general model of a communication system is illustrated in Fig. 1(a). In this model, a transmitter sends a signal containing some information which is unknown to the receiver. On the other side of the channel, a receiver receives a signal and is aimed to determine the transmitted information. One of the most studied channel models in communication theory is the additive white Gaussian noise (AWGN) channel which is illustrated in Fig. 1(a). In this model, the transmitted signal $x(t)$ is corrupted by an additive noise, $n(t)$, which is white and comes from a Gaussian random process with zero mean, $n \sim N(0, \sigma_0^2)$. In Fig. 1(a), H is the impulse response of the channel and in its simplest form can be modeled as an attenuation. Generally, the transmitter and receiver are situated in different locations and transmission occurs before the reception.

Let us assume that a transmitter is located in a *future* time, t_F , and is sending us a message, M_{t_F} . We, as the receivers of the message, are located at the current time, $t_C < t_F$, and receive a corrupted and noisy signal, S_{t_C} . Then, the problem of communication is to estimate the message through the received signal as $\hat{M}_{t_F} = g(S_{t_C})$, where \hat{M}_{t_F} is the estimated message and $g(\cdot)$ is the estimation function. Here, since M_{t_F} is generated in the *future* and S_{t_C} is received at the present time, estimating the unknown message is a prediction problem. By mapping the t_F to t , t_C to $t - k$, M_{t_F} to y_t , and S_{t_C} to \mathbf{Y}_1^{t-k} , we can observe that the above communication system is a regression problem, defined in Eq. 1. More precisely, we can assume that a transmitter is *located* at time t and sends us an unknown message y_t that we received it in the form of \mathbf{Y}_1^{t-k} . Then, the regression problem is to estimate the transmitted message, y_t , based on the received signal. We define the noisy channel forecasting model as follows:

Definition 1 *Noisy Channel Forecasting (NCF) Model is a representation of a time series forecasting problem as a noisy communication system that assumes that the future values of the time series are unknown messages that have been sent back in time through a noisy channel reversal and received in the form of our observations.*

The NCF model aids in a better understanding of the underlying dynamic behavior of time series by leveraging tools and techniques from the communication domain. It is reminiscent of the work of Chen et al. [7] which addresses the application of communication theory and information theory through the use of equalizers to optimize information transfer and analyze causal relationships. These authors also proposed an information coupling approach for dimensionality reduction.

The NCF model can be used as a wrapper for any time series analysis approach. While various channel models can be used in NCF, in this paper we use NCF with additive noise channels, defined as follows.

Definition 2 *NCF model with additive Gaussian noise (AGN) is an NCF that models the channel as an additive Gaussian noise as $\mathbf{Y}_1^{t-k} = \mathbf{h}(y_t) + \mathbf{N}$, where $\mathbf{h}(\cdot)$ is a vector function, and \mathbf{N} is a multivariate Gaussian noise.*

Let us assume that we use the general forecasting function of Eq. 1 and the NCF model with AGN. Then, we have the following two equations that relates observed data with future values:

$$\begin{cases} \hat{y}_t = \varphi(\mathbf{Y}_1^{t-k}) \\ \mathbf{Y}_1^{t-k} = \mathbf{h}(y_t) + \mathbf{N} \end{cases} \quad (4)$$

Using the training dataset, optimum model parameters including noise, channel, and estimation function coefficients, can be estimated through an optimization problem that minimizes some sort of risk function. For instance, with an MSE risk function, the following optimization problem can be solved to find the system parameters:

$$\min (y_t - \hat{y}_t)^2 = \min (y_t - \varphi(\mathbf{h}(y_t) + \mathbf{N}))^2. \quad (5)$$

In this paper, we model linear regression problems using NCF models with AGN in order to tackle with noisy time series and reduce the effect of noise in the forecasting process to improve the accuracy of predictions. The provided framework can be easily extended to nonlinear regression algorithms.

3. NOISY CHANNEL REVERSAL REGRESSION

In this section, we first model an autoregression problem using NCF with AGN. Then, we estimate noise and channel statistics simultaneously. Results of this estimation can be used to estimate the amount of noise in a specific time series.

3.1. Modeling time series as a Noisy Channel

Based on Definitions 1 and 2, in this section we model the AR(m) of Eq. 2 using NCF. In the general case, as depicted in Fig. 1(b), the system is modeled as a multiple-output virtual communication channel. Hence, based on Definition 2, we can formulate this channel as follows:

$$\mathbf{Y}_{t-k-(m-1)}^{t-k} = y_t \mathbf{S} + \mathbf{N} \quad (6)$$

where $\mathbf{S} = [s_{m-1} \cdots s_0]^T$, $s_j \geq 0$, represents the channel effect (attenuation) and $\mathbf{N} = [n_{m-1} \cdots n_0]^T$. Similar to Eq. 5 and using Eq. 2, the coefficients α_i 's and β , can be obtained by solving the following optimization problem:

$$\begin{aligned} \mathbf{A}^*, \beta^* &= \arg \min_{\mathbf{A}, \beta} E \left[(y_t - \hat{y}_t)^2 \right] \\ &= \arg \min_{\mathbf{A}, \beta} E \left[(\beta + \mathbf{A}^T \mathbf{N} + y_t (\mathbf{A}^T \mathbf{S} - 1))^2 \right] \end{aligned} \quad (7)$$

By the simplifying assumption that Y is a set of i.i.d. random variables, we can show that Eq. 7 depends on the mean, μ_Y , and variance, σ_Y^2 , of the time series Y . In other words,

$$\begin{aligned} \mathbf{A}^*, \beta^* &= \arg \min_{\mathbf{A}, \beta} (\beta + \mathbf{A}^T \mathbf{N})^2 + \\ &(\mathbf{A}^T \mathbf{S} - 1)^2 (\sigma_Y^2 + \mu_Y^2) + 2 (\beta + \mathbf{A}^T \mathbf{N}) (\mathbf{A}^T \mathbf{S} - 1) \mu_Y \end{aligned} \quad (8)$$

In the special case when data is available upto y_{t-1} and we model AR(1) with NCF, the optimum weights are as follows:

$$\alpha_0^* = \frac{s_0 \sigma^2}{s_0^2 \sigma^2 + \sigma_0^2}; \beta^* = \frac{\mu \sigma_0^2 - s_0 \mu_0 \sigma^2}{s_0^2 \sigma^2 + \sigma_0^2} \quad (9)$$

where $s_0 \geq 0$ is channel attenuation and n_0 is an additive Gaussian noise, $n_0 \sim N(\mu_0, \sigma_0^2)$. Furthermore, μ and σ^2 are the expectation and variance of the time series, respectively. The following lemma provides a lower bound for noise and signal parameters in the case of AR(1) with NCF model.

Lemma 1 *Considering the noise in the NCF model of AR(1) improves the accuracy if $\frac{R^2}{1-R^2} < \frac{s_0^2 \sigma^2}{\sigma_0^2}$, where $R \in [-1, +1]$ is the Pearson's product-moment coefficient of the time series.*

Proof *It can be shown that the expected Mean-Square error of the standard AR(1) is $e_{MSE} = \sigma^2(1 - R^2)$. Also, for NCF-based AR(1), error can be estimated using $e_{MSE}^{NCF} = \frac{\sigma^2 \sigma_0^2}{s_0^2 \sigma^2 + \sigma_0^2}$. We can show that $e_{MSE}^{NCF} < e_{MSE}$ holds, if we have $\sigma_0^2 < s_0^2 \sigma^2 \frac{1-R^2}{R^2}$ and with minor manipulations, this leads to $\frac{R^2}{1-R^2} < \frac{s_0^2 \sigma^2}{\sigma_0^2}$ thus proving the lemma.*

Based on Lemma 1, when $R = 0$, there is no linear relationship between y_t and y_{t-1} and for any value of s_0 , σ^2 , and σ_0^2 , the proposed method works better than the classic approach. Also, with perfect linear relationship between y_t and y_{t-1} , $R^2 = 1$, it is not possible to achieve lower error using the proposed method. Note that for random signals and noises with zero mean, $\frac{s_0^2 \sigma^2}{\sigma_0^2}$ represents the SNR at the output of the channel. This shows the importance of SNR in time series analysis.

In order to address instability, based on Eq. 3, we can reformulate Eq. 6 as follows:

$$\begin{aligned} \mathbf{Y}_{t-k-(m-1)}^{t-k} &= \dot{\mathbf{Y}}_{t-k-(m-1)}^{t-k} + \boldsymbol{\nu}_{t-k-(m-1)}^{t-k} \\ &= \dot{y}_t \mathbf{S} + \mathbf{N} + \boldsymbol{\nu}_{t-k-(m-1)}^{t-k} \end{aligned} \quad (10)$$

where $\boldsymbol{\nu}_{t-k-(m-1)}^{t-k} = [\nu_{t-k-(m-1)} \cdots \nu_{t-k}]^T$. Hence, we have two noise components and by independence assumption and assuming that ν_j is wide sense stationary, these noise components can be aggregated into a single noise factor, \dot{n}_j , which is also Gaussian. Then, Eq. 7 can be used to determine the regression coefficients if we substitute n_j 's with \dot{n}_j 's.

3.2. Simultaneous Estimation of Noise and Channel

Before calculating α_i 's and β , we have to first estimate the channel properties, \mathbf{S} and \mathbf{N} . There are various channel estimation methods in the literature [8]. In this paper, we develop an integrated maximum likelihood method to simultaneously estimate the noise and channel characteristics.

Let us assume that we have already observed the time series Y till time $t-k$ and an m -output noisy channel (Fig. 1(b)) is used in the NCF model. Then, using the maximum likelihood approach, we want to choose our parameters to maximize the probability of observation. In other words,

$$\mathbf{S}^*, \mathbf{M}^*, \boldsymbol{\Sigma}^* = \arg \max_{\mathbf{S}, \mathbf{M}, \boldsymbol{\Sigma}} \Pr [\mathbf{Y}_{m+k}^{t-k} | \mathbf{S}, \mathbf{M}, \boldsymbol{\Sigma}] \quad (11)$$

where $\mathbf{S} = [s_{m-1} \cdots s_0]^T$ represents channel attenuation, $\mathbf{M} = [\mu_{m-1} \cdots \mu_0]^T$ is a vector that contains noise expectations, and $\boldsymbol{\Sigma}$ is the $m \times m$ covariance matrix of noise. Note that based on the proposed model, in order to learn the system parameters, we need to relate each observed data sample, y_j to the m samples $\mathbf{Y}_{j-k-(m-1)}^{j-k}$. Therefore, in Eq. 11, the index of y begins from $m+k$. Assuming that y_j 's are i.i.d. and this assumption that noise has an m -dimensional Gaussian distribution, $\mathbf{N} \sim \mathcal{N}(\mathbf{M}, \boldsymbol{\Sigma})$, one can easily observe that Eq. 11 is equivalent to maximization of the following equation:

$$L = \prod_{j=m+k}^{t-k} \frac{e^{-\frac{1}{2}(\mathbf{Y}_{j-k-(m-1)}^{j-k} - \mathbf{M}_j)^T \boldsymbol{\Sigma}^{-1} (\mathbf{Y}_{j-k-(m-1)}^{j-k} - \mathbf{M}_j)}}{\sqrt{(2\pi)^m |\boldsymbol{\Sigma}|}}$$

where $\mathbf{M}_j = y_j \mathbf{S} + \mathbf{M}$. Using the log-likelihood, maximization of L is equivalent to the following optimization problem:

$$\mathbf{S}^*, \mathbf{M}^*, \boldsymbol{\Sigma}^* = \arg \min_{\mathbf{S}, \mathbf{M}, \boldsymbol{\Sigma}} (t - m - 2k + 1) \ln (|\boldsymbol{\Sigma}|) + \quad (12)$$

$$\sum_{j=m+k}^{t-k} (\mathbf{Y}_{j-k-(m-1)}^{j-k} - \mathbf{M}_j)^T \boldsymbol{\Sigma}^{-1} (\mathbf{Y}_{j-k-(m-1)}^{j-k} - \mathbf{M}_j)$$

For the single receiver case ($m = 1$) and when data is available upto y_{t-1} one can show that these parameters can be determined using the following set of equations:

$$\begin{cases} \sigma_0^2 = \frac{1}{t-2} \sum_{j=1}^{t-2} (y_j - s_0 y_{j+1} - \mu_0)^2 \\ s_0 \sum_{j=1}^{t-2} y_{j+1}^2 + \mu_0 \sum_{j=1}^{t-2} y_{j+1} = \sum_{j=1}^{t-2} y_{j+1} y_j \\ s_0 \sum_{j=1}^{t-2} y_{j+1} + (t-2) \mu_0 = \sum_{j=1}^{t-2} y_j \end{cases} \quad (13)$$

4. FORECASTING WITH NOISE FILTERING

Usually, real-world time series and their accompanying noise have different representations in the frequency domain. The

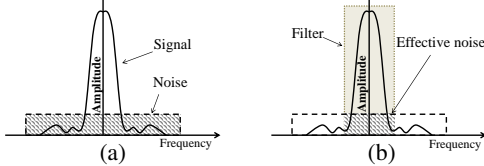


Fig. 2. (a) Example of a signal and noise in frequency domain. (b) Effect of filtering on signal and noise.

difference between the noise and signal power distribution in the frequency domain is the key intuition behind using out-of-band noise filtering in order to improve the SNR and increase the prediction accuracy. An illustrative example is shown in Fig. 2. In this figure, noise is uniformly distributed over the channel bandwidth while signal power is concentrated around zero. As it is shown in Fig. 2(b), by using a low-pass filter with a sufficient bandwidth, a significant power of noise is reduced while the effective part of signal remains unchanged.

The first step in filter design is recognition of important and non-important frequency components of the signal. For this purpose, we use FFT to find the representation of the signal in frequency domain. We define the γ -effective bandwidth of a time series as follows:

Definition 3 γ -effective bandwidth: For an arbitrary time series y_t and $\gamma \in [0, 1]$, the γ -effective bandwidth is the set of frequency-domain components, symmetrically located around the peak frequency, that carry γP_y of the time series power, where P_y is the total power of time series. Size of the γ -effective bandwidth of $y(t)$ is shown by B_y^γ .

After representing the time series in frequency domain, we aim to find the γ -effective bandwidth of the signal. Therefore, we need to identify those frequency components that carry the most part of the signal power and to choose an appropriate value for γ . Using a very small value for γ results in an unacceptable distortion in the signal while using larger values results in higher amount of noise. This step can be carried out using cross-validation. Those frequency components that carry γ portion of P_y are in the γ -effective bandwidth of the signal and the remaining ones are considered as out-of-band components. Out-of-band frequency components are forced to zero and then, the filtered version of Y , Y^f , is determined using an inverse-FFT transformation.

While filtering improves the accuracy of regression for noisy time series, it can be harmful for noiseless ones. In fact, the total energy of a noiseless time series is the signal energy and filtering any number of the frequency components results in unnecessary signal distortions which may reduce the accuracy of the regression.

Lemma 2 When noise power is uniformly distributed in frequency domain, out-of-band noise filtering with γ -effective bandwidth filter improves time series SNR if $\gamma > B_y^\gamma/B_y^1$.

Proof Let us assume that the total power of the time series is $P_{S,N} = P_S + P_N$, where P_S is the signal power and P_N is the

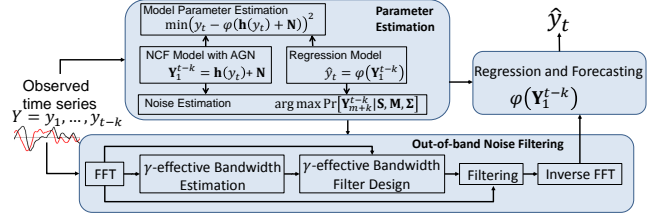


Fig. 3. Overall proposed framework.

power of noise. Obviously, SNR of the original time series is $SNR^{Original} = P_S/P_N$. Similar to Fig. 2, the noise power is uniformly distributed over all frequency components. The total bandwidth of the signal contains B_y^1 components in the frequency domain while the filtered time series contains B_y^γ components. Then, after filtering, we will have:

$$P_N^{Filtered} = \frac{B_y^\gamma}{B_y^1} P_N, \quad P_S^{Filtered} = \gamma P_{S,N} - \frac{B_y^\gamma}{B_y^1} P_N$$

$$SNR^{Filtered} = \frac{\gamma(P_S + P_N) - \frac{B_y^\gamma}{B_y^1} P_N}{\frac{B_y^\gamma}{B_y^1} P_N}$$

If we aim to improve the SNR we should have $SNR^{Filtered} > SNR^{Original}$. This means that we should have $\gamma > B_y^\gamma/B_y^1$.

4.1. Forecasting Framework

The overall view of the proposed framework is illustrated in Fig. 3. This framework has three major building blocks:

- *Parameter estimation:* where various parameters including regression coefficients and time series noise are estimated.
- *Out-of-band noise filtering:* where a filter is used to remove the out-of-band noise of the time series.
- *Regression and forecasting:* where using the filtered time series forecasting is performed.

Note that filtering should be performed if time series is noisy. Otherwise, the procedure is likely to result in lower accuracy. The main goal of the parameter estimation step is to estimate the amount of noise in the time series in terms of noise statistics and SNR to be used in the filtering block. After out-of-band noise filtering, one can use the filtered time series for forecasting.

5. CASE STUDY: SYNTHETIC DATASET

In order to study the behavior of regression algorithms under out-of-band noise filtering and to determine the accuracy of the estimations performed in previous sections, we performed various experiments on synthetic time series. In this section we provide results and discuss the observations.

To construct the synthetic dataset, we first generate a time series based on $x(t) = 3t + 2$, where t changes between 0 and 100 in steps of 0.1. Then, we construct various time series with added Gaussian noise with zero mean and different variances. For this purpose, we set $y^1 = x$ and

$$y^j = x + n^j, \quad n^j \sim \mathcal{N}(0, \sigma_j^2), \quad \sigma_j = 2^{j-2}, \quad j = 2, \dots, 9$$

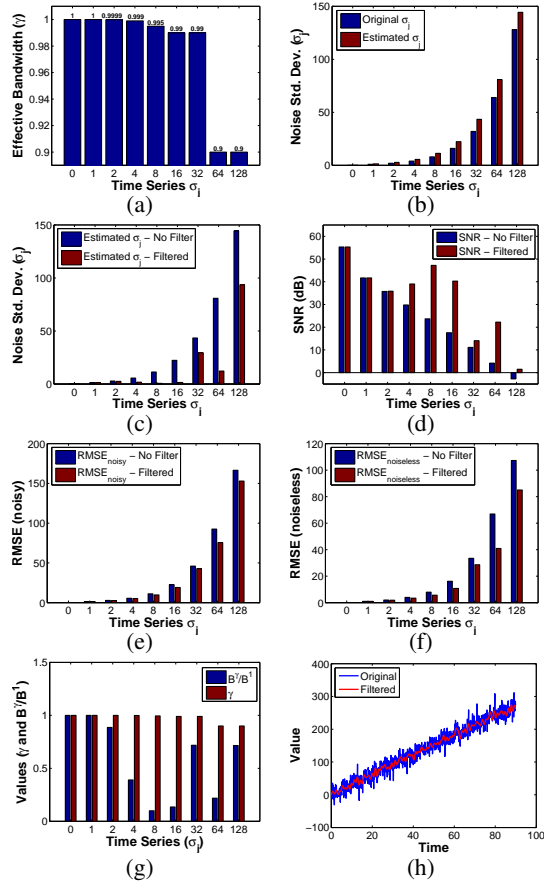


Fig. 4. Results of synthetic data: (a) selected values for γ , (b) estimated and actual values of σ_j , (c) effect of filtering on σ_j , (d) SNR of time series in dB, (e) RMS error w.r.t noisy data, (f) RMS error w.r.t noiseless data, (g) comparison of γ with $\frac{B^\gamma}{B^1}$ and (h) original and filtered time series with $\sigma_j = 16$.

From each of these time series, we generated 100 instances and for each instance, we first estimate the properties of the noise based on Eq. 13. Then, we use AR(1) with and without out-of-band noise filtering. For AR(1) with filtering, we first performed cross validation to choose the best value of γ . In addition to the regression accuracy, we also measured the signal and noise properties, before and after filtering.

Results of the synthetic time series are shown in Fig. 4. Figure 4(a) shows the best selected values for γ for each of the time series. As we expected, for smaller value of σ_j (when noise power is low), $\gamma = 1$ is the best. When noise power is low, filtering results in unnecessary signal distortions and this may be harmful for regression accuracy. However, while σ_j increases, the best value of γ decreases. Fig. 4(b) compares the original values of σ_j (the ones used in data generation) with the estimated ones, calculated based on Eq. 13. Results show that the estimated values are close to real ones. Fig. 4(c) shows σ_j of the time series before and after filtering. As we expected, filtering reduces the noise effect dramatically. The effect of filtering on SNR is illustrated in Fig. 4(d). Again,

Table 1. Effect of filtering on regression algorithms and time series. Green color shows the improvement in accuracy.

Time series	AR(1)	AR(Opt)	R-AR(1)	R-AR(Opt)	R-ARMA(1,1)	R-ARMA(Opt)
PAHO - Argentina	1.50	4.86	0.00	0.00	5.60	1.22
PAHO - Chile	0.00	2.37	9.19	0.00	15.56	0.25
PAHO - Colombia	0.00	3.83	1.04	5.18	5.21	0.00
PAHO - Costa Rica	8.88	12.00	4.15	4.36	5.76	0.35
PAHO - Ecuador	44.50	37.42	8.83	9.86	12.99	6.03
PAHO - El Salvador	2.35	3.46	8.43	1.89	7.32	2.12
PAHO - Guatemala	5.07	5.07	7.12	1.05	3.36	0.00
PAHO - Honduras	11.22	11.22	11.81	6.20	6.02	0.00
PAHO - Mexico	18.41	21.57	0.00	0.00	0.00	0.00
PAHO - Nicaragua	10.75	10.75	8.03	0.00	2.67	0.00
PAHO - Panama	1.36	1.36	1.10	0.82	7.24	0.27
PAHO - Paraguay	15.88	13.92	0.00	0.00	0.00	0.00
GST - Melanoma	4.17	0.00	8.00	6.00	2.08	2.08
GST - Volleyball	16.67	16.67	3.33	3.33	13.33	0.00
GST - Wrestling	37.04	26.09	5.26	5.26	19.05	10.53
GST - Swimming	9.30	22.50	0.00	0.00	0.00	0.00
GST - Yahoo	35.14	12.50	0.00	0.00	4.55	4.55
GST - Microsoft	66.67	100.00	0.00	0.00	0.00	0.00
GST - Facebook	14.29	15.00	10.53	5.56	5.56	0.00
GST - Youtube	14.29	10.00	10.00	5.26	5.26	0.00
GST - Hotmail	15.79	13.89	3.23	0.00	0.00	0.00
GST - Ebay	18.00	8.89	2.38	2.38	2.38	2.38
GST - Dollar	15.09	6.25	2.17	2.17	8.16	0.00
GST - Yen	26.67	24.14	11.54	4.17	26.67	0.00
GST - Pound	18.60	10.53	0.00	0.00	10.26	15.79
GST - Earthquake	21.19	11.19	4.96	0.00	6.45	0.00
GST - Hurricane	35.17	35.17	36.23	7.78	23.58	0.00
GST - Tsunami	21.92	21.92	18.46	3.51	24.66	0.00
GST - Storm	1.49	1.49	17.65	0.00	0.00	0.00
STOCK - Yahoo	5.13	5.13	5.13	5.13	7.69	7.69
STOCK - Ford	12.50	12.50	12.50	12.50	12.50	12.50
STOCK - GE	14.29	14.29	14.29	14.29	14.29	14.29
STOCK - AEP	11.11	5.88	11.11	5.88	11.11	5.88
NIST SRD - ENSO	3.21	3.31	0.00	0.00	0.00	1.62
NIST SRD - Gauss1	13.03	0.00	15.30	0.00	5.41	0.00
NIST SRD - Gauss2	22.41	10.94	25.87	12.50	10.83	5.96
NIST SRD - Gauss3	21.80	7.91	23.78	9.31	6.36	2.18
UCI - Bike Sharing (Daily)	0.57	0.00	4.40	0.00	1.70	0.00
UCI - CallT2 In (Daily sum)	2.01	0.00	1.99	0.00	4.84	0.00
UCI - Dodgers (Daily Avg)	0.00	0.00	3.60	0.00	7.64	0.00

results show that filtering improves the SNR dramatically.

Regression accuracy, before and after filtering, are compared in Fig. 4(e) and (f). In these figures, RMS error is used as a measure of accuracy. Fig. 4(e) shows the regression error when we compare the estimated value with the actual value of the noisy signal, $RMSE_{noisy}$. For each of these time series, we used the last 100 samples for test. Generally, when we aim to test the accuracy of a regression algorithm, we use $RMSE_{noisy}$. However, when the observed time series is noisy and we are using it to predict a noiseless value (such as prediction of the number of people that will have Influenza based on an observed noisy data), we need to compare our predictions with the noiseless time series. We show this error by $RMSE_{noiseless}$ and results are illustrated in Fig. 4(f).

Figure 4(g) compares the value of γ and the ratio $\frac{B^\gamma}{B^1}$. In Lemma 2, we showed that for an effective filtering, γ should be greater than $\frac{B^\gamma}{B^1}$. Results of Fig. 4(g) shows that this condition is always true. To illustrate how the amount of noise in the time series is reduced by filtering, Fig. 4(h) shows an example. This figure compares the original and filtered synthetic time series with $\sigma_6 = 16$. It is obvious that filtering has reduced the amplitude of noise in the time series dramatically which results in the improvements in regression accuracy.

6. EXPERIMENTAL RESULTS

In this section, to have a better understanding of the performance of the proposed framework, we perform experiments with 40 time series from the following datasets:

- UCI: Real-world time series from UCI Repository.

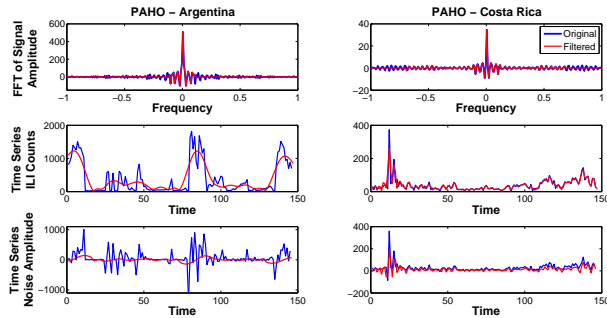


Fig. 5. The effect of filtering on two time series ($\gamma = 0.9$). The top row shows time series in frequency domain. The middle row shows the time series in time domain. The bottom row shows the estimated effective noise.

- NIST: Time series from the NIST StRD website¹.
- STOCK: Closing stock price of different companies from Yahoo Finance.
- GST: Weekly Google Search Trends from different topics.
- PAHO: Influenza Like Illnesses statistics of various countries from the Pan American Health Organization website².

In addition to AR(m), we also perform experiments with recursive-AR(m) (R-AR) and recursive-ARMA(m, q) (R-ARMA). We report results for basic (i.e. $m = q = 1$) and optimal settings for each algorithm, where optimal settings were determined through cross validation. We also performed experiments with and without filtering. We measured the accuracy using the following relative error (RE):

$$e_{RE} = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \frac{|\hat{y}_t - y_t|}{\max(y_t, \hat{y}_t)}$$

where \mathcal{T} is the test set (in this experiment, last 15% samples).

Table 1 shows the effect of filtering on various algorithms and time series. Numbers depict the achieved improvements in accuracy due to filtering and the green color shows cases where filtering has improved the accuracy. It is obvious from the table that in most of the cases filtering has improved the accuracy of regression. Results also show that on average, AR(1) shows the highest improvement. This is due to the nature of AR(1), as other methods are more complex and less vulnerable to noise. The average error of algorithms before and after filtering over 40 time series are compared in Fig. 6.

The effect of filtering with $\gamma = 0.9$ on two example time series is illustrated in Fig. 5. The first row of the figure shows the FFT of the time series before and after filtering. This figure also shows that how filtering affects on the quality of signal and amplitude of noise. It is obvious from this figure that the noise amplitude has been reduced dramatically after filtering. The results are consistent with Table 1.

7. CONCLUSION

In this paper, we proposed the NCF model to extend forecasting methods and a new framework for time series regression,

¹<http://www.itl.nist.gov/div898/strd/nls/nls/main.shtml>

²<http://ais.paho.org/phil/viz/ed/flu.asp>

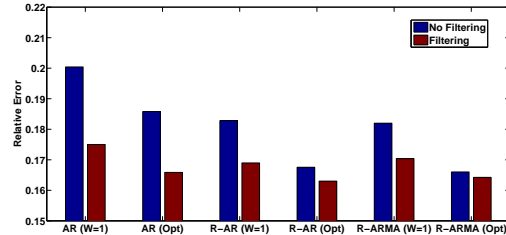


Fig. 6. Comparison between average error of different regression algorithms before and after filtering over 40 time series.

using a communication theoretic method. In this model, noise plays the role of a hidden factor in the system and we showed that under certain conditions, the proposed model performs better than the classic methods. Experimental results showed that the proposed solution results in higher accuracy than the classic approaches. Future work includes extending the NCF model to nonlinear regressions and expanding it to include other communication-theoretic techniques such as equalizers.

Acknowledgment: Supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center (DoI/NBC) contract number D12PC000337, the US Government is authorized to reproduce and distribute reprints of this work for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the US Government.

8. REFERENCES

- [1] P. Chakraborty et al., “Forecasting a moving target: Ensemble models for ILI case count predictions,” in *Proc. of SDM’14*, 2014, pp. 262–270.
- [2] N.I. Sapankevych and R. Sankar, “Time series prediction using support vector machines: A survey,” *Computational Intelligence, IEEE*, vol. 4, no. 2, pp. 24–38, 2009.
- [3] G.E.P. Box et al., *Time series analysis: forecasting and control*, John Wiley & Sons, 2013.
- [4] S.J. Gershman and D.M. Blei, “A tutorial on bayesian nonparametric models,” *J. of Math. Phys.*, vol. 56, no. 1, pp. 1–12, 2012.
- [5] J.B. Guerard and E. Schwartz, “Regression analysis and forecasting models,” in *Quantitative Corporate Finance*, pp. 277–301. Springer, 2007.
- [6] J. Stevens, “Outliers and influential data points in regression analysis,” *Psychological Bulletin*, vol. 95, pp. 334–344, 1984.
- [7] Kwang-Cheng Chen, Shao-Lun Huang, Lizhong Zheng, and H. Vincent Poor, “Communication theoretic data analytics,” *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 4, pp. 663–675, 2015x.
- [8] R.S. Ganesh and J.J. Kumari, “A survey on channel estimation techniques in mimo-ofdm mobile communication systems,” *IJSER*, vol. 4, no. 5, pp. 1851–1855, 2013.