Kalman Filter and SVR Combinations in Forecasting US Unemployment

Georgios Sermpinis, Charalampos Stasinakis, Andreas Karathanasopoulos

Kalman Filter and SVR combinations in forecasting US Unemployment

Georgios Sermpinis[1], Charalampos Stasinakis[2] and Andreas Karathanasopoulos[3]

[1] University of Glasgow Business School (E-mail: georgios.sermpinis@glasgow.ac.uk) [2] University of Glasgow Business School (E-mail: c.stasinakis.1@research.gla.ac.uk) [3] University of East London Business School (E-mail: a.karatahnasopoulos@uel.ac.uk)

Abstract. The motivation for this paper is to investigate the efficiency of a Neural Network (NN) architecture, the Psi Sigma Network (PSN), in forecasting US unemployment and compare the utility of Kalman Filter and Support Vector Regression (SVR) in combining NN forecasts. An Autoregressive Moving Average model (ARMA) and two different NN architectures, a Multi-Layer Perceptron (MLP) and a Recurrent Network (RNN), are used as benchmarks. The statistical performance of our models is estimated throughout the period of 1972-2012, using the last seven years for out-of-sample testing. The results show that the PSN statistically outperforms all models' individual performances. Both forecast combination approaches improve the statistical accuracy, but SVR outperforms substantially the Kalman Filter.

Keywords: Forecast Combinations, Kalman Filter, Support Vector Regression, Unemployment.

1 Introduction

Many applications in the macroeconomic literature aim to derive and compare information from econometric models' forecasts. For that reason, forecasting competitions of linear and non-linear architectures are common and focus on numerous time series, such as unemployment, inflation, industrial production, gross domestic product etc. The artificial NNs are computation models that researchers include in such macroeconomic forecasting schemes, because they embody promising data-adaptive learning and clustering abilities.

The motivation for this paper is to investigate the efficiency of a Neural Network (NN) architecture, the Psi Sigma Network (PSN), in forecasting US unemployment and compare the utility of Kalman Filter and Support Vector Regression (SVR) in combining NN forecasts. An Autoregressive Moving Average model (ARMA) and two different NN architectures, a Multi-Layer Perceptron (MLP) and a Recurrent Network (RNN), are used as benchmarks. The statistical performance of our models is estimated throughout the period of 1972-2012, using the last seven years for out-of-sample testing. The results show that the PSN statistically outperforms all models' individual performances. Both forecast combination approaches improve the statistical accuracy, but SVR is substantially better than the Kalman Filter.

Section 2 is a short literature review and Section 3 follows with the description of the dataset used in this application. Sections 4 and 5 give an overview of the forecasting models and the forecast combination methods implemented respectively. The statistical performance of our models is presented in Section 6. Finally, some concluding remarks are summarized in Section 7.

2 Literature Review

Forecasting unemployment rates is a very popular and well documented case study in the literature (see amongst others Rothman [16], Montgomery et al. [14] and Koop and Potter [11]). Swanson and White [20] forecast several macroeconomic time series, including US unemployment, with linear models and NNs. In their approach, NN architectures present promising empirical evidence against the linear VAR models. Johnes [9] reports the results of another forecasting competition between linear autoregressive, GARCH, threshold autoregressive and NNs models, applied to the UK monthly unemployment rate series. In his application, NNs are superior when the forecasting horizon is 18 and 24 months ahead, but fail to outperform the other models in shorter forecasting horizons.

Liang [12] applies Bayesian NNs in forecasting unemployment in West Germany and shows that they present significantly better forecasts than traditional autoregressive models. Teräsvirta et al. [22] examine the forecast accuracy of linear autoregressive, smooth transition autoregressive and NN models for 47 monthly macroeconomic variables, including unemployment rates, of the G7 economies. The empirical results of their study point out the risk for implausible NN forecasts at long forecasting horizons. Nonetheless, their forecasting ability is much improved when they are combined with autoregressive models. This idea of combining forecasts originates from Bates and Granger [1]. Newbold and Granger [15] also suggested combining rules based on variances-covariances of the individual forecasts, while Deutsch et al. [3] achieved substantially smaller squared forecasts errors combining forecasts with changing weights. Harvey [8] and Hamilton [7] both propose using state space modeling, such as Kalman Filter, for representing dynamic systems where unobserved variables (so-called 'state' variables) can be integrated within an 'observable' model. Finally, Terui and Van Dijk [23] also suggest that the combined forecasts perform well, especially with time varying coefficients.

3 US Unemployment dataset

In this application, we forecast the monthly percentage change of the US unemployment rate (UNEMP), as provided by the online Federal Reserve Economic Data (FRED) database of the Federal Reserve Bank of St. Louis[1]. The forecasting perfor-

[1] Based on the description given by FRED, the US unemployment rate or civilian unemployment rate represents the number of unemployed as a percentage of the labour force. Labour force data are restricted to people 16 years of age and older, who currently reside in 1 of the

mance of our models is explored over the period of 1972 to 2012, using the last seven years for out-of-sample evaluation. The time series is seasonally adjusted and it is divided into three sub-periods as summarized in Table 1 below:

Table 1. The US Unemployment Dataset - Neural Networks' Training Dataset

PERIODS	MONTHS	START DATE	END DATE
Total Dataset	492	01//01/1972	01/12/2012
Training Dataset ***(In-sample)***	324	01//01/1972	01/12/1998
Test Dataset ***(In-sample)***	84	01/01/1999	01/12/2005
Validation Dataset ***(Out-of-sample)***	84	01/01/2006	01/12/2012

The following graph presents the US unemployment rate for the period under study:

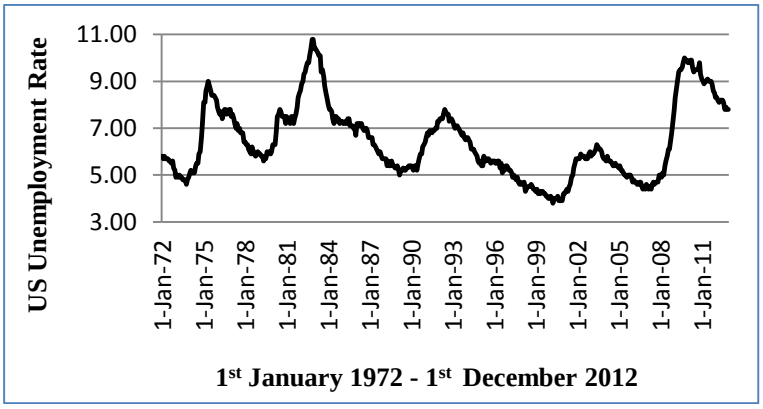

Fig. 1. The US Unemployment Rate

In the absence of any formal theory behind the selection of the inputs of a NN, we conduct some NN experiments and a sensitivity analysis on a pool of potential inputs in the in-sample dataset in order to help our decision. Finally, we select as inputs sets of autoregressive terms of UNEMP that provide the best statistical performance for each network in the test sub-period. These sets are presented in Table 2 below:

50 states or the District of Columbia, who do not reside in institutions (e.g., penal and mental facilities, homes for the aged) and who are not on active duty in the Armed Forces.

Table 2. Neural Networks' Inputs

MLP	RNN	PSN
UNEMP (1)*	UNEMP (1)	UNEMP (1)
UNEMP (2)	UNEMP (3)	UNEMP (2)
UNEMP (4)	UNEMP (4)	UNEMP (3)
UNEMP (5)	UNEMP (6)	UNEMP (6)
UNEMP (7)	UNEMP (7)	UNEMP (8)
UNEMP (10)	UNEMP (9)	UNEMP (10)
UNEMP (11)	UNEMP (11)	UNEMP (11)
UNEMP (12)	UNEMP (12)	UNEMP (12)

*UNEMPL UNEMP (1) is the first autoregressive term of the UNEMP series

4 Forecasting Models

4.1 Auto-Regressive Moving Average Model (ARMA)

In this paper an ARMA model is used to benchmark the efficiency of the NNs' statistical performance. Using as a guide the correlogram and the information criteria in the in-sample subset, we have chosen a restricted ARMA (7, 7) model, where all the coefficients are significant at the 95% confidence interval. The selected ARMA model is presented in equation (1) below:

$$\hat{Y}_t = 0.03 + 1.025Y_{t-1} - 0.293Y_{t-2} + 0.511Y_{t-4} - 0.321Y_{t-7} - 1.006\varepsilon_{t-1} + 0.463\varepsilon_{t-2} - 0.545\varepsilon_{t-4} - 0.211\varepsilon_{t-7} \quad (1)$$

where $\hat{Y}_t$ is the forecasted percentage change of the US unemployment rate.

4.2 Neural Networks (NNs)

Neural networks exist in several forms in the literature. The most popular architecture is the Multi-Layer Perceptron (MLP). A standard neural network has at least three layers. The first layer is called the input layer (the number of its nodes corresponds to the number of explanatory variables). The last layer is called the output layer (the number of its nodes corresponds to the number of response variables). An intermediary layer of nodes, the hidden layer, separates the input from the output layer. Its number of nodes defines the amount of complexity the model is capable of fitting. In addition, the input and hidden layer contain an extra node called the bias node. This node has a fixed value of one and has the same function as the intercept in traditional regression models. Normally, each node of one layer has connections to all the other nodes of the next layer.

The network processes information as follows: the input nodes contain the value of the explanatory variables. Since each node connection represents a weight factor, the information reaches a single hidden layer node as the weighted sum of its inputs. Each node of the hidden layer passes the information through a nonlinear activation function and passes it on to the output layer if the calculated value is above a threshold.

The training of the network (which is the adjustment of its weights in the way that the network maps the input value of the training data to the corresponding output value) starts with randomly chosen weights and proceeds by applying a learning algorithm called backpropagation of errors [18].The learning algorithm simply tries to find those weights which minimize an error function (normally the sum of all squared differences between target and actual values). Since networks with sufficient hidden nodes are able to learn the training data (as well as their outliers and their noise) by heart, it is crucial to stop the training procedure at the right time to prevent overfitting (this is called 'early stopping'). This can be achieved by dividing the dataset into three subsets respectively called the training and test sets used for simulating the data currently available to fit and tune the model and the validation set used for simulating future values. The training of a network is stopped when the mean squared forecasted error is at minimum in the test-sub period. The network parameters are then estimated by fitting the training data using the above mentioned iterative procedure (backpropagation of errors). The iteration length is optimised by maximising the forecasting accuracy for the test dataset. Then the predictive value of the model is evaluated applying it to the validation dataset (out-of-sample dataset).

4.2.1 The Multi-Layer Perceptron Model (MLP)

MLPs are feed-forward layered NN, trained with a back-propagation algorithm. According to Kaastra and Boyd [10], they are the most commonly used types of artificial networks in financial time-series forecasting. The training of the MLP network is processed on a three-layered architecture, as described above.

4.2.2 The Recurrent Neural Network (RNN)

The next NN architecture used in this paper is the RNN. For an exact specification of recurrent networks, see Elman [5]. A simple recurrent network has an activation feedback which embodies short-term memory. The advantages of using recurrent networks over feed-forward networks for modeling non-linear time series have been well documented in the past. However, as mentioned by Tenti [21], "the main disadvantage of RNNs is that they require substantially more connections, and more memory in simulation than standard back-propagation networks" (p. 569), thus resulting in a substantial increase in computational time.

4.2.3 The Psi-Sigma Neural Network (PSN)

The PSNs are a class of Higher Order Neural Networks with a fully connected feed-forward structure. Ghosh and Shin [6] were the first to introduce the PSN, trying to reduce the numbers of weights and connections of a Higher Order Neural Network. Their goal was to combine the fast learning property of single-layer networks with the mapping ability of Higher Order Neural Networks and avoid increasing the required number of weights. The price for the flexibility and speed of Psi Sigma networks is that they are not universal approximators. We need to choose a suitable order of approximation (or else the number of hidden units) by considering the estimated function complexity, amount of data and amount of noise present. To overcome this, our code runs simulations for orders two to six and then it presents the best network. The evaluation of the PSN model selected comes in terms of trading performance.[2]

5 Forecast Combination Techniques

5.1 Kalman Filter

Kalman Filter is an efficient recursive filter that estimates the state of a dynamic system from a series of incomplete and noisy measurements. The time-varying coefficient combination forecast suggested in this paper is shown below:

$$\text{Measurement Equation: } f_{c_{NNs}}^{t} = \sum_{i=1}^{3} a_i^t f_i^t + \varepsilon_t, \quad \varepsilon_t \sim NID\left(0, \sigma_\varepsilon^2\right) \tag{2}$$

$$\text{State Equation: } a_i^t = a_i^{t-1} + n_t, \quad n_t \sim NID(0, \sigma_n^2) \tag{3}$$

Where:

- $f_{c_{NNs}}^{t}$ is the dependent variable (combination forecast) at time *t*
- f_i^t $(i = 1, 2, 3)$ are the independent variables (individual forecasts) at time *t*
- a_i^t $(i = 1, 2, 3)$ are the time-varying coefficients at time *t* for each NN
- ε_t,n_t are the uncorrelated error terms (noise)

The alphas are calculated by a simple random walk and we initialized $\varepsilon_1 = 0$. Based on the above, our Kalman Filter model has as a final state the following:

$$f_{c_{NNs}}^{t} = 13.46 f_{MLP}^{t} + 16.38 f_{RNN}^{t} + 41.97 f_{PSN}^{t} + \varepsilon_t \tag{4}$$

[2] For a complete description of all the neural network models we used and their complete specifications see Sermpinis et *al.* [17].

From the above equation we note that the Kalman filtering process also favors the PSN model, which is the model that performs best individually.

5.1 Support Vector Regression (SVR)

Vapnik [24] established Support Vector Regression (SVR) as a robust technique for constructing data-driven and non-linear empirical regression models. They provide global and unique solutions and do not suffer from multiple local minima (Suykens [19]). They also present a remarkable ability of balancing model accuracy and model complexity (and Lu et al.[13]). The SVR function can be specified as:

$$f(x) = w^T \varphi(x) + b \tag{5}$$

where w and b are the regression parameter vectors of the function and φ(x) is the non-linear function that maps the input data vector x into a feature space where the training data exhibit linearity. The ε-sensitive loss Lε function finds the predicted points that lie within the tube created by two slack variables ξ_i, ξ_i^* :

$$L_\varepsilon(x_i) = \begin{cases} 0 \; if \mid y_i - f(x_i) \mid \leq \varepsilon \\ \mid y_i - f(x_i) \mid - \varepsilon \; if \; other \end{cases}, \varepsilon \geq 0 \tag{6}$$

In other words ε is the degree of model noise insensitivity and L_ε finds the predicted values that have at most ε deviations from the actual obtained values y_i. The goal is to solve the following argument[3]:

$$\text{Minimize } C\sum_{i=1}^{n}(\xi_i + \xi_i^*) + \frac{1}{2}\|w\|^2 \text{ subject to } \begin{Bmatrix} \xi_i \geq 0 \\ \xi_i^* \geq 0 \\ C > 0 \end{Bmatrix} \text{ and}$$

$$\begin{Bmatrix} y_i - w^T \varphi(x_i) - b \leq +\varepsilon + \xi_i \\ w^T \varphi(x_i) + b - y_i \leq +\varepsilon + \xi_i^* \end{Bmatrix} \tag{7}$$

The above quadratic optimization problem is transformed in a dual problem and its solution is based on the introduction of two Lagrange multipliers a_i, a_i^* and mapping with a kernel function $K(x_i, x)$:

$$f(x) = \sum_{i=1}^{n}(a_i - a_i^*)K(x_i, x) + b \quad \text{where } 0 \leq a_i, a_i^* \leq C \tag{8}$$

Support Vectors (SVs) are called all the xi that contribute to equation (8), thus they lie outside the ε-tube, whereas non-SVs lie within the ε-tube. Increasing ε leads

[3] For a detailed mathematical analysis of the SVR solution see Vapnik [24].

to less SVs' selection, whereas decreasing it results to more 'flat' estimates. The norm term $\|w\|^2$ characterizes the complexity (flatness) of the model and the term

$$\left\{\sum_{i=1}^{n}(\xi_i+\xi_i^*)\right\}$$

is the training error, as specified by the slack variables. Consequently the introduction of the parameter C satisfies the need to trade model complexity for training error and vice versa (Cherkassky and Ma [2]). In our application, the NN forecasts are used as inputs for a ε-SVR simulation. A RBF kernel4 is selected and the parameters have been optimized based on cross-validation in the in-sample dataset (ε=0.06, γ= 2.47 and C=0.103).

6 Statistical Performance

As it is standard in literature, in order to evaluate statistically our forecasts, the RMSE, the MAE, the MAPE and the Theil-U statistics are computed (see Dunis and Williams [4]). For all four of the error statistics retained the lower the output, the better the forecasting accuracy of the model concerned. In Table 3 we present the statistical performance of all our models in the in-sample period.

Table 3. Summary of the In-Sample Statistical Performance

	ARMA	**MLP**	**RNN**	**PSN**	**Kalman Filter**	**SVR**
MAE	1.9941	0.0078	0.0077	0.0073	0.0067	0.0062
MAPE	65.25%	52.78%	50.17%	47.73%	45.76%	41.52%
RMSE	2.5903	1.0671	0.9572	0.9045	0.8744	0.8256
Theil-U	0.6717	0.6142	0.5827	0.5325	0.5017	0.4549

We note that from our individual forecasts, the PSN statistically outperforms all other models. Both forecast combination techniques improve the forecasting accuracy, but SVR is the superior model regarding all four statistical criteria. Table 4 below summarizes the statistical performance of our models in the out-of-sample period.

[4] The RBF kernel equation is $K(x_i, x) = \exp(-\gamma\|x_i - x\|^2), \gamma > 0$.

Table 4. Summary of the Out-of-sample Statistical Performance

	ARMA	MLP	RNN	PSN	Kalman Filter	SVR
MAE	0.0332	0.0072	0.0071	0.0061	0.0057	0.0051
MAPE	67.45%	50.17%	48.97%	44.38%	40.21%	34.33%
RMSE	2.4043	0.9557	0.9354	0.8927	0.8549	0.8005
Theil-U	0.5922	0.5654	0.5591	0.4818	0.4657	0.4154

From the results above, it is obvious that the statistical performance of the models in the out-of-sample period is consistent with the in-sample one and their ranking remains the same. All NN models outperform the traditional ARMA model. In addition, the PSN outperforms significantly the MLP and RNN in terms of statistical accuracy. The idea of combining NN unemployment forecasts seems indeed very promising, since both Kalman Filter and SVR present improved statistical accuracy also in the out-of-sample period. Moreover, SVR confirms its forecasting superiority over the individual architectures and the Kalman Filter technique. In other words, SVR's adaptive trade-off between model complexity and training error seems more effective than the recursive ability of Kalman Filter to estimate the state of our process.

7 Concluding Remarks

The motivation for this paper is to investigate the efficiency of a Neural Network (NN) architecture, the Psi Sigma Network (PSN), in forecasting US unemployment and compare the utility of Kalman Filter and Support Vector Regression (SVR) in combining NN forecasts. An Autoregressive Moving Average model (ARMA) and two different NN architectures, a Multi-Layer Perceptron (MLP) and a Recurrent Network (RNN), are used as benchmarks. The statistical performance of our models is estimated throughout the period of 1972-2012, using the last seven years for out-of-sample testing.

As it turns out, the PSN outperforms its benchmark models in terms of statistical accuracy. It is also shown that all the forecast combination approaches outperform in the out-of-sample period all our single models. All NN models beat the traditional ARMA model. In addition, the PSN outperforms significantly the MLP and RNN in terms of statistical accuracy. The idea of combining NN unemployment forecasts seems indeed very promising, since both Kalman Filter and SVR present improved statistical accuracy also in the out-of-sample period. SVR confirms its forecasting superiority over the individual architectures and the Kalman Filter technique. In other, SVR's adaptive trade-off between model complexity and training error seems more effective than the recursive ability of Kalman Filter to estimate the state of our process. The remarkable statistical performance of SVR allows us to conclude that it can be considered as an optimal forecast combination for the models and time series under

study. Finally, the results confirm the existing literature, which suggests that nonlinear models, such as NNs, can be used to model macroeconomic series.

References

1. Bates, J. M., Granger, C. W. J.: The Combination of Forecasts. Operational Research Society. 20, 451-468 (1969)
2. Cherkassky, V., Ma, Y.: Practical selection of SVM parameters and noise estimation for SVM regression. Neural Networks. 17, 113-126 (2004)
3. Deutsch, M., Granger, C.W. J., Teräsvirta, T.: The combination of forecasts using changing weights. International Journal of Forecasting. 10, 47-57 (1994)
4. Dunis, C. L., Williams, M.: Modelling and Trading the EUR/USD Exchange Rate: Do Neural Network Models Perform Better?. Derivatives Use, Trading and Regulation. 8, 211-239 (2002)
5. Elman, J. L.: Finding Structure in Time. Cognitive Science. 14, 179-211 (1990)
6. Ghosh, J., Shin, Y.: The Pi-Sigma Network: An efficient Higher-order Neural Networks for Pattern Classification and Function Approximation. In: Proceedings of International Joint Conference of Neural Networks, pp. 13-18. (1991)
7. Hamilton, J. D.: Time series analysis. Princeton University Press, Princeton (1994)
8. Harvey, A. C.: Forecasting, structural time series models and the Kalman filter. Cambridge University Press, Cambridge (1990)
9. Johnes, G.: Forecasting unemployment. Applied Economics Letters. 6, 605-607 (1999)
10. Kaastra, I., Boyd, M.: Designing a Neural Network for Forecasting Financial and Economic Time Series. Neurocomputing. 10, 215-236 (1996)
11. Koop, G., Potter, S.M.: Dynamic Asymmetries in U.S. Unemployment. Journal of Business & Economic Statistics. 17, 298-312 (1999)
12. Liang, F.: Bayesian neural networks for nonlinear time series forecasting. Statistics and Computing. 15, 13-29 (2005)
13. Lu, C.J., Lee, T.S., Chiu, C.C.: Financial time series forecasting using independent component analysis and support vector regression. Decision Support Systems. 47, 115-125 (2009)
14. Montgomery, A.L., Zarnowitz, V., Tsay, R.S., Tiao, G.C.: Forecasting the U.S. Unemployment Rate. Journal of the American Statistical Association. 93, 478-493 (1998).
15. Newbold, P., Granger, C. W. J.: Experience with Forecasting Univariate Time Series and the Combination of Forecasts. Journal of the Royal Statistical Society. 137, 131-165 (1974).
16. Rothman, P.: Forecasting Asymmetric Unemployment Rates. The Review of Economics and Statistics. 80, 164-168 (1998)
17. Sermpinis, G., Laws, J., Dunis, C.L.: Modelling and trading the realised volatility of the FTSE100 futures with higher order neural networks. European Journal of Finance. 1-15 (2012)
18. Shapiro, A. F.: A Hitchhiker's guide to the techniques of adaptive nonlinear models. Insurance: Mathematics and Economics. 26, 119-132 (2000)
19. Suykens, J. A. K., Brabanter, J. D., Lukas, L., Vandewalle, L.: Weighted least squares support vector machines: robustness and sparse approximation. Neurocomputing. 48, 85-105 (2002)

20. Swanson, N. R., White, H.: A Model Selection Approach to Real-Time Macroeconomic Forecasting Using Linear Models and Artificial Neural Networks. Review of Economics and Statistics. 79, 540-550 (1997)
21. Tenti, P. : Forecasting foreign exchange rates using recurrent neural networks. Applied Artificial Intelligence. 10, 567-581 (1996)
22. Teräsvirta, T., Van Dijk, D., Medeiros, M. C.: Linear models, smooth transition autoregressions, and neural networks for forecasting macroeconomic time series: A re-examination. International Journal of Forecasting. 21,755–774 (2005)
23. Terui, N., Van Dijk, H. K.: Combined forecasts from linear and nonlinear time series models. International Journal of Forecasting. 18, 421-438 (2002).
24. Vapnik, V. N.: The nature of statistical learning theory. Springer (1995)