Journal of Korean
Data & Information Science Society
2004, Vol. 15, No. 2, pp. 499～506

Comparison of EKF and UKF on Training the Artificial Neural Network

Daehak Kim[1)]

Abstract

The Unscented Kalman Filter is known to outperform the Extended Kalman Filter for the nonlinear state estimation with a significance advantage that it does not require the computation of Jacobian but EKF has a competitive advantage to the UKF on the performance time. We compare both algorithms on training the artificial neural network. The validation data set is used to estimate parameters which are supposed to result in better fitting for the test data set. Experimental results are presented which indicate the performance of both algorithms.

Keywords : Artificiall neural network, Extended Kalman Filter, Jacobian, Unscented Kalman Filter

1. Introduction

For the linear dynamic system with white input and observation noise, the Kalman Filter proposed by Kalman(1960) and refined by Kalman and Bucy(1961) is well known to be an optimal algorithm. The extended version of the Kalman Filter - the Extended Kalman Filter(EKF) can be applied to the nonlinear dynamic system by linearizing the system around the current estimates of the parameters(Gelb(1974), Anderson and Moore(1979)). Singhal and Wu(1989) compared EKF and the back-propagation proposed by Rumelhart et al. (1986) on training the artificial neural network using two dimensional examples. The Unscented Kalman Filter(UKF) proposed by Julier and Uhlman(1997, 2002) is shown to outperform EKF in the nonlinear state estimation. In the parameter estimation of the artificial neural network, UKF is slightly better than the EKF with a significance advantage that it does not require the computation of neural

1) Professor, Dept. of Statistical Information, Catholic University of Daegu
E-mail : dhkim@cu.ac.kr

network Jacobian. Although EKF is computationally complicative rather than UKF, it updates parameters consistently with all previous data and usually converges in a few iterations.

On training the artificial neural network with EKF and UKF we use the validation data to find iteration number which can achieve the minimum error for the validation data. In the following sections, we introduce EKF and UKF, how they can be applied to the artificial neural network and compare their performances using the French Curve data and XOR data.

2. The Extended Kalman Filter

Let us consider the simple discrete time nonlinear dynamic system where the unobserved signal hidden state $\boldsymbol{w}_k$ is modelled as a Markov process of initial distribution $p(\boldsymbol{w}_0)$ and transition equation,

$$\boldsymbol{w}_k = \boldsymbol{w}_{k-1} + v_k \tag{1,a}$$

while the observations ,

$$y_k = h(\boldsymbol{w}_k) + u_k. \tag{1,b}$$

are assumed conditionally independent given the state $\boldsymbol{w}_k$, where $E(u_k) = E(v_k) = 0$,$Var(u_k) = U_k$ and $Var(v_k) = V_k$, U_k and V_k are assumed to be known. The process noise v_k drives the dynamic system, while the observation noise is given by u_k and they are independent of $\boldsymbol{w}_{k-1}$, y_k, respectively.

The Minimum Mean Squared Error (MMSE) estimate of the state $\boldsymbol{w}_k$ of a nonlinear discrete time system (1) satisfies conditions that the estimation error

$$e_k = \boldsymbol{w}_k - \hat{\boldsymbol{w}}_k$$

is unbiased($E[e_k] = 0$) and orthogonal to the observation y_k ($E[e_k y_k] = 0$). EKF and UKF provide an MMSE estimate of the state $\boldsymbol{w}_k$ using predictor-corrector scheme. Given the estimate of the state $\boldsymbol{w}_{k-1}$ and its covariance P_{k-1}, obtained for the set of observations up to the time step $k-1$,

$$\boldsymbol{y}^{k-1} = \{y_i, i = 1, \cdots, k-1\},$$

the filter predicts the future state using the process model and the knowledge about the process noise distribution. Predicted mean and covariance are ideally as:

$$
\begin{aligned}
\tilde{\boldsymbol{w}}_k &= E(\boldsymbol{w}_k \,|\, \boldsymbol{y}^{k-1}) = E(\boldsymbol{w}_{k-1} + v_k \,|\, \boldsymbol{y}^{k-1}) = E(w_{k-1} \,|\, \boldsymbol{y}^{k-1}) = \hat{\boldsymbol{w}}_{k-1} \\
P_k &= Cov(\boldsymbol{w}_k \,|\, \boldsymbol{y}^{k-1}) = Cov(\boldsymbol{w}_{k-1} + v_k \,|\, \boldsymbol{y}^{k-1}) \\
&= Cov(w_{k-1} \,|\, \boldsymbol{y}^{k-1}) + Cov(v_k \,|\, \boldsymbol{y}^{k-1}) = \hat{P}_{k-1} + V_{k-1}
\end{aligned}
$$

$$
\begin{aligned}
\tilde{y}_k &= E(y_k \,|\, \boldsymbol{y}^{k-1}) = E(h(\boldsymbol{w}_k) + u_k \,|\, \boldsymbol{y}^{k-1}) = E(h(\boldsymbol{w}_k) \,|\, \boldsymbol{y}^{k-1}) \\
q_k &= Cov(y_k \,|\, \boldsymbol{y}^{k-1}) = Cov(h(\boldsymbol{w}_k) + u_k \,|\, \boldsymbol{y}^{k-1}) = Cov(h(\boldsymbol{w}_k) \,|\, \boldsymbol{y}^{k-1}) + U_k \\
S_k &= Cov(\boldsymbol{w}_k, y_k \,|\, \boldsymbol{y}^{k-1}) = Cov(\boldsymbol{w}_k, h(\boldsymbol{w}_k) + u_k \,|\, \boldsymbol{y}^{k-1}) \\
&= Cov(\boldsymbol{w}_k, h(\boldsymbol{w}_k) \,|\, \boldsymbol{y}^{k-1})
\end{aligned}
$$

The estimate $\hat{\boldsymbol{w}}_k = E[\boldsymbol{w}_k \,|\, \boldsymbol{y}^k]$ and its covariance $\hat{P}_k = Cov[\boldsymbol{w}_k \,|\, \boldsymbol{y}^k]$ are obtained by updating (correcting) the state prediction $(\tilde{\boldsymbol{w}}_k, P_k, S_k, q_k)$ with the current observation y_k as follows.

$$
\begin{aligned}
\hat{\boldsymbol{w}}_k &= E(\boldsymbol{w}_k \,|\, \boldsymbol{y}^k) = \tilde{\boldsymbol{w}}_k + S_k q_k^{-1}(y_k - \tilde{y}_k) = \tilde{\boldsymbol{w}}_k + K_k(y_k - \tilde{y}_k) \\
\hat{P}_k &= Cov(\boldsymbol{w}_k \,|\, \boldsymbol{y}^k) = P_k - S_k q_k^{-1} S_k' = P_k - K_k S_k'
\end{aligned}
\tag{2}
$$

where $K_k = S_k q_k^{-1}$ is the Kalman gain matrix. EKF uses the first order Taylor approximation of $h(\boldsymbol{w}_k)$ with respect to $\boldsymbol{w}_k = \tilde{\boldsymbol{w}}_k$. Then $\hat{\boldsymbol{w}}_k = E[\boldsymbol{w}_k \,|\, \boldsymbol{y}^k]$ and $\hat{P}_k = Cov[\boldsymbol{w}_k \,|\, \boldsymbol{y}^k]$ can be estimated by using following results.

$$
\begin{aligned}
\tilde{y}_k &= E(y_k \,|\, \boldsymbol{y}^{k-1}) = h(\tilde{\boldsymbol{w}}_k) \\
q_k &= Cov(y_k \,|\, \boldsymbol{y}^{k-1}) = A_k P_k A_k' + U_k \\
S_k &= Cov(\boldsymbol{w}_k, y_k \,|\, \boldsymbol{y}^{k-1}) = P_k A_k'
\end{aligned}
\tag{3}
$$

where A_k is Jacobian of h with respect to $\boldsymbol{w}_k = \tilde{\boldsymbol{w}}_k$.

3. The Unscented Kalman Filter

Julier and Uhlman(1997) proposed the Unscented Transformation (UT) in order to calculate the statistics of a random variable $\boldsymbol{w}$ propagated through nonlinear function $y = h(\boldsymbol{w})$. The n dimensional continuous random variable $\boldsymbol{w}_k$ with $\tilde{\boldsymbol{w}}_k = E(\boldsymbol{w}_k \mid \boldsymbol{y}^{k-1})$ and q_k and S_k are approximated by $2n+1$ sigma points W_p with corresponding weights wt_p, $p = 0, 1, \cdots, 2n$.

$$\begin{aligned}
&W_0 = \tilde{\boldsymbol{w}}, wt_0 = \lambda/(n+\lambda), \lambda = \alpha^2(n+\phi) - n \text{ for } p = 1, 2, \cdots, n,\\
&W_p = \tilde{\boldsymbol{w}} + s_p\sqrt{n+\lambda},\ wt_p = 0.5/(n+\lambda)\, wt_0 = \lambda/(n+\lambda),\\
&W_{p+n} = \tilde{\boldsymbol{w}} - s_p\sqrt{n+\lambda},\ wt_{p+n} = 0.5/(n+\lambda),
\end{aligned}$$

where α determines the spread of the sigma points around $\tilde{\boldsymbol{w}}$ (usually $1.e-4 \leqq \alpha \leqq 1$) and $\phi \in R$ is the scaling parameter, usually set to 0 or $3-n$, s_p is the pth row or column of the matrix square root of $P_{\boldsymbol{w}}$. Each sigma point is instantiated through the function $h(\cdot)$ to yield the set of transformed sigma points $h(W_0)$ and the mean $\tilde{y}_k$ of a transformed distribution is estimated by

$$\begin{aligned}
\tilde{y}_k = E(y_k \mid \boldsymbol{y}^{k-1}) = &\sum_{p=0}^{2n} wt_p h(W_p)\\
= \frac{\lambda}{n+\lambda} h(\tilde{\boldsymbol{w}}_k) + &\frac{1}{2(n+\lambda)} \sum_{i=1}^{n} h(\tilde{\boldsymbol{w}}_k + s_p\sqrt{(n+\lambda)}) + h(\tilde{\boldsymbol{w}}_k - s_p\sqrt{n+\lambda}).
\end{aligned}$$

The covariance estimates obtained by unscented transform are

$$\begin{aligned}
q_k = &Cov(y_k \mid \boldsymbol{y}^{k-1}) = \sum_{p=0}^{2n} wt_p(h(W_p) - \tilde{y}_k)(h(W_p) - \tilde{y}_k)^T\\
&= \frac{\lambda}{n+\lambda}(h(\tilde{\boldsymbol{w}}_k) - \tilde{y}_k)(h(\tilde{\boldsymbol{w}}) - \tilde{y}_k)^T\\
&+ \frac{1}{2(n+\lambda)} \sum_{p=1}^{n} (h(\tilde{\boldsymbol{w}}_k + s_p\sqrt{n+\lambda}) - \tilde{y}_k)(h(\tilde{\boldsymbol{w}}_k + s_p\sqrt{n+\lambda} - \tilde{y}_k)^T\\
&+ \frac{1}{2(n+\lambda)} \sum_{p=1}^{n} (h(\tilde{\boldsymbol{w}}_k - s_p\sqrt{n+\lambda}) - \tilde{y}_k)(h(\tilde{\boldsymbol{w}} - s_p\sqrt{n+\lambda} - \tilde{y}_k)^T
\end{aligned} \tag{4}$$

and

$$\begin{aligned} S_k &= Cov(\boldsymbol{w}_k, y_k \,|\, \boldsymbol{y}^{k-1}) = \sum_{p=0}^{2n} wt_p (W_p - \tilde{\boldsymbol{w}}_k)(h(W_p) - \tilde{y})^T \\ &= \frac{\lambda}{n+\lambda} (W_p - \tilde{\boldsymbol{w}}_k)(h(\tilde{\boldsymbol{w}}_k) - \tilde{y}_k)^T \\ &\quad + \frac{1}{2(n+\lambda)} \sum_{p=1}^{n} (W_p - \tilde{\boldsymbol{w}}_k)(h(\tilde{\boldsymbol{w}}_k + s_p \sqrt{n+\lambda} - \tilde{y}_k)^T \\ &\quad + \frac{1}{2(n+\lambda)} \sum_{p=1}^{n} (W_p - \tilde{\boldsymbol{w}}_k)(h(\tilde{\boldsymbol{w}}_k - s_p \sqrt{n+\lambda} - \tilde{y}_k)^T. \end{aligned} \tag{5}$$

Then $(\hat{\boldsymbol{w}}_k, \hat{P}_k)$ are updated with the current observation y_k by substituting (4) and (5) in the update step (2).

<Figure 1> Predicted line for the test data by EKF and UKF

4. Numerical studies

We illustrate the performance of EKF and UKF on training the artificial neural network through two data sets - French curve data set and XOR data set. The logarithmic sigmoid transfer function is used for the artificial neural network for both data sets.

The French curve data set consists of 230 of input data $\boldsymbol{x}$ generated from $x_i = 3^*(2^* i - 1)/200$ for $i = 1, \cdots, 230$ and 230 of output data $\boldsymbol{y}$ generated from a normal distribution

$$N(4.26*(exp(-x_i)-4*\exp(-2*x_i)+3*\exp(-3*x_i)),0.04)$$

for $i=1,\cdots,230$. The first 100 of data are used for the training data set to estimate $\boldsymbol{w}$, 30 of data are used for the validation to find the proper iteration number and the rest of data are used for test data set to predict the true curve. For the French curve data 2 layers with 5 nodes are used then the parameters to be estimated in the network is given as $\boldsymbol{w}$ such that

$$\begin{aligned} l_1 &= x\times w(1)+w(2),\ l_2 = x\times w(3)+w(4),\ l_3 = x\times w(5)+w(6)\\ l_4 &= x\times w(7)+w(8),\ l_5 = x\times w(9)+w(10),\\ h(\boldsymbol{w}) &= w(11)/(1+exp(-l_1))+w(12)/(1+exp(-l_2))\\ &\quad +w(13)/(1+exp(-l_3))+w(14)/(1+exp(-l_4))\\ &\quad +w(15)/(1+exp(-l_5))+w(16) \end{aligned}$$

We trained the artificial neural network with EKF and UKF by initializing $\hat{P}_0 = I_{16}$ and setting $V_k = 0.01\times I_{16}$ and $U_k = 0.01$, respectively, and the control parameter α of UKF 0.433. For test data we obtained MSE as 0.6128 and 0.0416, respectively. Figure 1 shows the true curve and predicted curve by EKF and UKF. For the reference we train the artificial neural network with the back-propagation algorithm 500 times with each iteration number 1000. The minimum of MSE for test data is 0.0379, the maximum of MSE is 0.0503 and the average of MSE is 0.0408.

1200 XOR data are generated as follows

$$\begin{gathered} x_{1i}\sim U(-1,+1),\ x_{2i}\sim U(-1,+1)\\ y_i = -1 \text{ if } x_{1i}\times x_{2i} < 0,\ y_i = +1 \text{ if } x_{1i}\times x_{2i} \geqq 0. \end{gathered}$$

The first 500 of data are used for the training data set to estimate $\boldsymbol{w}$, 200 of data are used for the validation to find the proper iteration number and the rest of data are used for test data set to predict the true curve. For XOR data 2 layers with 4 nodes are used then the parameters to be estimated in the network is given as $\boldsymbol{w}$ such that

$$l_1 = x_1 \times w(1) + w(2) \times x_2 + w(3), l_2 = x_1 \times w(4) + w(5) \times x_2 + w(6),$$
$$l_3 = x_1 \times w(7) + w(8) \times x_2 + w(9), l_4 = x_1 \times w(10) + w(11) \times x_2 + w(12)$$
$$h(\boldsymbol{w}) = w(13)/(1 + exp(-l_1)) + w(14)/(1 + exp(-l_2))$$
$$+ w(15)/(1 + exp(-l_3)) + w(16)/(1 + exp(-l_4)) + w(17)$$

We trained the artificial neural network with EKF and UKF by initializing $\hat{P}_0 = I_{17}$ and setting $V_k = 0.001 \times I_{17}$ and $U_k = 0.001$, respectively, and the control parameter α of UKF 0.420. For test data we obtained misclassifcation rate for the test data 0.13 and 0.08, respectively. Figure 2 shows the predicted values by EKF and UKF.

For the reference we train the artificial neural network with the back-propagation algorithm 500 times with each iteration number 1000. The minimum of misclassification rate for test data is 0.032, the maximum of misclassification rate is 0.450 and the average of misclassification rate is 0.1126.

<Figure 2> The predicted value by EKF and UKF (*:y=-1, o: y=+1)

5. Remarks and Conclusions

Through the examples we showed that UKF derives more satisfying results than EKF on training the artificial neural network, but for performance time EKF is more satisfying. And for some cases UKF can provide better results than the back-propagation on training the artificial neural network.

References

1. Anderson, B.D.O. and Moore, J.B. (1979). Optimal Filtering, Prentice Hall.
2. Gelb, A. (1974). Applied Optimal Estimation, MIT Press.
3. Julier, S.J. and Uhlmann, J.K. (1997). A New Extension of the Kalman Filter to Nonlinear Systems, *Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls,* Orlando, FL.
4. Julier, S.J. and Uhlmann, J.K. (2002). Scaled Unscented Transformation, *Proceedings of IEEE American Control Conference, 8-10 May.*
5. Kalman, R.E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering, Transactions of the ASME Series D* 82, 35-45
6. Kalman, R.E. and Bucy, R. S. (1961). New Results in Linear Filtering and Prediction Theory, *Journal of Basic Engineering*, 8D: 96-108.
7. Rumelhart, D.E., Hinton, G. E. and Williams, R. J. (1986). Learning Internal Representations by Error Propagation, Parallel Distributed Processing, Cambridge, MIT Press.
8. Singhal, S. and Wu, L. (1989) Training Multilayer Perceptrons with The Extended Kalman Algorithm, *Advances in Neural Information Processing Systems 1, ed. D.S. Touretzky, Morgan Kaufmann Publishers.*

[received date : Mar. 2004, accepted date : May. 2004]