Hidden State Estimation using the Correntropy Filter with Fixed Point Update and Adaptive Kernel Size

Goktug T. Cinar and José C. Príncipe

Abstract—In this paper we review the Correntropy Filter for hidden state estimation and we introduce the fixed point update rule for the Correntropy Filter instead of using gradient ascent for faster convergence. We further propose an adaptive kernel bandwidth selection algorithm. It is shown that the new filter outperforms the Kalman Filter and has no free parameters. The algorithm’s capabilities are demonstrated on a simulated experiment and a vehicle tracking problem.


I. INTRODUCTION AND BACKGROUND

It is well known that the Kalman Filter (KF) gives the optimal solution to the hidden state estimation problem in a setting where all the processes are Gaussian random processes. However because of the sub-optimal behavior of the KF in non-Gaussian settings, there was a need for a new filter that can extract higher order information from the signals. In [1] we introduced an information theoretic cost function utilizing the similarity measure Correntropy as a performance index. With this filter we have lost the recursive estimation of the hidden states, but have gained the advantage of exploiting the higher order and nonlinear statistics of the data. Previous filter have used the gradient ascent that was slower in convergence, especially with wrong step size selection. Here we propose a fixed point update rule that does not require the step size parameter, therefore does not suffer from speed of convergence. We also handle the kernel size selection in an adaptive manner, which is really critical to kernel methods. In this section we briefly review the fundamentals of the Correntropy Filter. In the following sections we will introduce the fixed point update, and suggest a kernel size adaptation rule. The elegance of the Kalman Filter lies in having an analytical solution and a recursive estimation algorithm for hidden state estimation and therefore having the optimal estimation at every iteration. The problem setup is as follows. The system equation is given as:

\[ x_k = F x_{k-1} + w_{k-1} \]  

(1)

and the measurement equation is:

\[ y_k = H x_k + v_k \]  

(2)

where \( w_k \) and \( v_k \) are independent, zero mean, Gaussian noise processes with covariance matrices \( Q_k \) and \( R_k \), respectively.

It is very important to understand the fundamental idea behind the formulation of Kalman Filter (KF). Among many ways that Kalman Filter formulation can be derived such as using orthogonality principle or the innovations approach, there is a derivation that can be achieved by the optimization of the quadratic cost function given in (3), while building a recursive estimator [2].

\[ J = \frac{1}{2} (y_k - Hx_k)^T R_k^{-1} (y_k - Hx_k) \]

\[ + \frac{1}{2} (x_k - x_k^-)^T M_k^{-1} (x_k - x_k^-) \]  

(3)

where \( M_k \) is the covariance matrix of \( (x_k - x_k^-) \) and:

\[ x_k^- = F x_{k-1} \]  

(4)

The value \( \hat{x}_k = \arg \min_{x_k} J(x_k) \) will be the best estimate. In this paper the system parameters will be assumed fixed for simplicity however all the formulation can be extended easily to time-varying case.

The formulation of KF can be derived by analytically solving (5).

\[ \frac{\partial J}{\partial x_k} = 0 \]  

(5)

Beyond the formulation that is achieved through this simple optimization process, there is an important message embedded in this cost function. It states that the main goal of the estimation process is to reduce the effect of the two sources of uncertainties in the dynamical system. The system noise (uncertainty) \( w_k \) and the measurement noise (uncertainty) \( v_k \). Basically the filter tries to account for these uncertainties and as it propagates the covariance matrix of the error through iterations, it only makes use of the second order information in the measurements.

However it is reasonable to say that there would be many applications of the filter where the gaussianity condition would not hold and we would be left with the sub-optimal solution the filter produces. Therefore there was a need for a method that would use the information available in the higher-order statistics of the signals, even if the recursive solution structure is lost. One such measure is Correntropy [3].
Consider two scalar random variables $X, Y \in \mathbb{R}$. Cross-Correntropy [3] is a generalized similarity measure between two arbitrary scalar random variables $X$ and $Y$ defined by:

$$v(X, Y) = E_{XY}[\kappa(X, Y)] = \int \int \kappa(x, y)p_{X,Y}(x, y)dxdy \quad (6)$$

where $\kappa(\cdot, \cdot)$ is any continuous positive definite kernel. In our formulation the Gaussian kernel $G_\sigma(\cdot, \cdot)$ will be employed. Therefore (6) becomes:

$$v(X, Y) = E_{XY}[G_\sigma(X, Y)] = \int \int G_\sigma(x, y)p_{X,Y}(x, y)dxdy \quad (7)$$

where $\sigma$ is the kernel size or bandwidth. As we have only limited amount of data and the joint PDF is unknown, we use the sample estimator instead of the expectation operator.

$$\hat{v}_{\sigma,N}(X, Y) = \frac{1}{N} \sum_{i=1}^{N} G_\sigma(x_i - y_i) \quad (8)$$

One nice property of correntropy is that it is positive and bounded and with Gaussian kernel it reaches its maximum if and only if $X = Y$. Another important property is that for Gaussian kernel, correntropy is a weighted sum of all the even moments of the random variable $Y - X$. The proof follows from the Taylor series expansion of the Gaussian function in (7):

$$v_\sigma(X, Y) = \frac{1}{\sqrt{2\pi}\sigma} \sum_{n=0}^{\infty} \frac{(-1)^n}{2^n\sigma^{2n}n!} E[(X - Y)^{2n}] \quad (9)$$

It is seen in (9) that correntropy is the sum of all even moments when the Gaussian kernel is used. The kernel size appears as a parameter weighting the second-order moment and higher order moments. With very large $\sigma$ (compared to the dynamic range of the data) this measure approaches correlation.

Having the property of correntropy being the weighted sum of all even order moments of the random variable, we decided to utilize Correntropy in the new cost function. Therefore the cost function to be optimized in the Correntropy Filter is:

$$J_c = v(y_k, Hx_k) + v(x_k, Fx_{k-1}) \quad (10)$$

$$J_c = \frac{1}{N} \sum_{i=1}^{N} G_\sigma(\|y_i - Hx_i\|) + \frac{1}{N} \sum_{i=1}^{N} G_\sigma(\|x_i - Fx_{i-1}\|) \quad (11)$$

II. FIXED POINT UPDATE RULE

In [1] we have used gradient ascent to optimize this cost function, here we derive the fixed point update rule that is faster and also leads to an adaptive way of optimizing the cost, that does not require the step size parameter.

The formulation of fixed point Correntropy Filter can be derived by analytically solving:

$$\frac{\partial J_c}{\partial x_i} = 0 \quad (13)$$
Fig. 3. The figure shows cost surface for different kernel sizes.

\[
\frac{1}{\sigma^2} G_\sigma(\|y_i - Hx_i\|) H^T(y_i - Hx_i)
- \frac{1}{\sigma^2} G_\sigma(\|x_i - Fx_{i-1}\|)(x_i - Fx_{i-1}) = 0 \quad (14)
\]

\[
\frac{G_\sigma(\|y_i - Hx_i\|)}{G_\sigma(\|x_i - Fx_{i-1}\|)} H^T(y_i - Hx_i) = (x_i - Fx_{i-1}) \quad (15)
\]

\[
\hat{x}_i = F\hat{x}_{i-1} + \frac{G_\sigma(\|y_i - H\hat{x}_i\|)}{G_\sigma(\|\hat{x}_i - F\hat{x}_{i-1}\|)} H^T(y_i - H\hat{x}_i) \quad (16)
\]

where on the right hand side of the equation (16) \(\hat{x}_i^-\) = \(F\hat{x}_{i-1}\) and \(\hat{x}_{i-1}\) is the best estimate at time \(i - 1\). Therefore the term \(G_\sigma(\|\hat{x}_i - F\hat{x}_{i-1}\|) = G_\sigma(\|0\|)\), which simplifies (16) to:

\[
\hat{x}_i = F\hat{x}_{i-1} + e^{-\frac{1}{2\sigma^2}(y_i - H\hat{x}_i)^T H(y_i - H\hat{x}_i)} \quad (17)
\]

The filter is tested on artificial data. To create the data we have used a two dimensional rotation matrix, the system has no inputs and is driven by a Gaussian mixture noise. The mixture consists of two equally weighted Gaussian distributions centered at \([-2,-2]\) and \([2,2]\) with variances \([0.1,0;0,0.1]\) respectively. The measurement noise consists of two equally weighted Gaussian distributions centered at \(-2\) and \(2\) with variances \(0.1\) respectively. The observed signal is the sum of the hidden states. We have used a kernel size of \(\sigma = 8\). As before, the new fixed point filter outperforms the Kalman Filter. This is demonstrated in Fig. 1. Also one can observe the error pdf's of both correntropy filter and KF in Fig. 2. The fixed point version of the Correntropy Filter still places the error PDF at 0, a very desirable result. It also results in smaller error variance compared to KF.

### III. ADAPTIVE KERNEL SIZE

When the fixed point update rule is used the algorithm only has one free parameter left, the kernel size. In information theoretic learning it is always a desirable goal to find an automated/adaptive rule for kernel size selection. In this section we propose a very intuitive kernel size adaptation rule. It would be beneficial to look at how the kernel size affects the cost surface. Three sample cost surfaces at a given time instant are shown in Fig. 3 for different kernel sizes. As the kernel bandwidth gets smaller, we observe that the nonlinear nature of the cost function becomes dominant around the optimum. When the kernel size is reduced to very small values, it is observed that a valley appears in the cost surface in addition to the convex hull. Also the optimal state values do not fall into these areas either. Thus the kernel size
### TABLE I

**MEAN SQUARED ERROR COMPARISON FOR HIDDEN STATE ESTIMATION BY KALMAN FILTER, ADAPTIVE KERNEL SIZE CORRENTROPY FILTER AND FIXED KERNEL SIZE CORRENTROPY FILTER**

<table>
<thead>
<tr>
<th></th>
<th>State 1 Estimation MSE</th>
<th>State 2 Estimation MSE</th>
<th>Measurement Estimation MSE</th>
<th># of Free Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kalman Filter</td>
<td>16.9102</td>
<td>14.5245</td>
<td>4.6074</td>
<td>0</td>
</tr>
<tr>
<td>Adaptive Kernel Size</td>
<td>7.9030</td>
<td>8.1701</td>
<td>2.1727</td>
<td>0</td>
</tr>
<tr>
<td>Correntropy Filter</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fixed Kernel Size</td>
<td>6.5334</td>
<td>6.3067</td>
<td>4.5243</td>
<td>1</td>
</tr>
<tr>
<td>Correntropy Filter</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Fig. 4.** The figure shows the state estimation of both filters and the true states.

selection is a rather important issue. A rule of thumb is to start searching for a kernel size around the standard deviation of the hidden states.

There are not many values that are accessible to the user in this problem setup. Normally we only have access to the measurements and a system model \((F,H)\). Therefore it is reasonable to make use of these in the adaptation of the kernel bandwidth. During the testing stages of the algorithm authors realized that the best kernel sizes are in the range of observation errors. This makes sense as first half of the cost function has the observation error as the argument and a similar kernel size will put us in the convex hull. Therefore we heuristically choose the kernel bandwidth at each time instant to be the observation error as shown in (18).

\[
\sigma_i = \|y_i - Hx_i\| \quad (18)
\]

**Fig. 4** shows the performance of the adaptive kernel size Correntropy Filter compared to the conventional Kalman Filter. It is seen that mostly the new filter has smaller MSE for estimation. In Fig. 5 the reader can see that the error PDF is still placed at 0, however the shape is more irregular compared to the fixed kernel size Correntropy Filter. In Table I we show the overall MSE for estimation for Kalman Filter and two types of Correntropy Filters. It can be seen that the fixed kernel size filter achieves the best result, however in real life scenarios fixing the best kernel size would be almost impossible. The second best performance is obtained with the adaptive kernel size Correntropy Filter. Even though it performs slightly worse than the fixed kernel size case, all the parameters are automatically adapted, therefore this filter can be considered a plug & play filter with no user manipulation.

**IV. VEHICLE TRACKING**

We now test both algorithms on a vehicle tracking problem and compare the results with the performance of the Kalman filter. The position data of the vehicles is taken from [4] which were hand labeled from the videos in PETS2001 dataset [5],[6].

We use Ramachandras model for vehicle tracking which is summarized below. The following equations define a linear relationship between the acceleration, velocity and position in a single dimension.

\[
x_i = x_{i-1} + \dot{x}_{i-1}T + \ddot{x}_{i-1}T^2/2 \quad (19)
\]

\[
\dot{x}_i = \dot{x}_{i-1} + \ddot{x}_{i-1}T \quad (20)
\]

\[
\ddot{x}_i = \ddot{x}_{i-1} + a_i \quad (21)
\]

where \(x, \dot{x}, \ddot{x}\) stands for position, velocity and acceleration respectively. \(T\) is the sample time and \(a_i\) is the noise term.
that accounts for maneuvers and acts as the driving force of the system. In this model we assume that the measurements are the position, and we have the same equations for both coordinates. Therefore we use the following $F,H$ matrices in our linear dynamical model.

\[
F = \begin{bmatrix}
1 & T & T^2 & 0 & 0 & 0 \\
T & 0 & 0 & 0 & 0 & 0 \\
0 & T & 0 & 0 & 0 & 0 \\
0 & 0 & T & 0 & 0 & 0 \\
0 & 0 & 0 & T & T^2 & 0 \\
0 & 0 & 0 & 0 & T & 0
\end{bmatrix}
\]  \tag{22}

\[
H = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0
\end{bmatrix}
\]  \tag{23}

Using this setup we tested the Kalman Filter, the Fixed Kernel Size Correntropy Filter and the Adaptive Kernel Size Correntropy Filter on position estimation. We will compare the estimated states corresponding to the positions in both coordinates with the hand labeled coordinates. In Fig. 6 we show that all algorithms does a good job of tracking the $y$-coordinate, and both versions of the Correntropy Filter outperformed the Kalman Filter. In Fig. 7 we show the tracking performance of all algorithms in both coordinates. It can be clearly seen that the Kalman Filter wanders off the true trajectory at some points. The error pdfs in Fig. 8 tell more about the performances of the filters. It can be clearly seen that the Fixed Kernel Size Correntropy Filter outperforms the other two algorithms. Adaptive Kernel Size Correntropy Filter outperforms the Kalman Filter in one coordinate and performs similarly in the other.

V. CONCLUSION AND DISCUSSIONS

In this paper we reviewed the Correntropy Filter for hidden state estimation and readdressed the need for such a filter. We have shown in [1] that the Correntropy Filter outperforms Kalman Filter in the presence of non-Gaussian uncertainties. We motivated the new filter by the cost function of the well-known Kalman Filter. As it is well-known the KF uses the second order statistics of the observed signals and therefore provides the optimal solution when the uncertainties in the system are Gaussian. On the other hand this assumption and second order cost function gives the possibility of coming up with an analytical solution. Therefore the convergence of the filter is very fast.

Even though we had better results than the Kalman Filter, the practicality of the filter was worse due to the presence of
two rather critical free parameters; step size and kernel band-
width. In this paper we introduced the fixed point update for Correntropy Filter instead of the gradient ascent formulation we have used previously. This helped us to get rid of the step size parameter. Later we proposed an intuitive adaptive kernel bandwidth selection and demonstrated that with this configuration the Correntropy Filter can still outperform the Kalman Filter. The best results were achieved with a pre-set and fixed kernel bandwidth, however it should be noted that in most real-life applications the user will not have a reference to set the kernel bandwidth for the best performance.

The authors are continuing to work on other adaptive kernel bandwidth selection algorithms to further improve the performance. These discussions are left for further publica-

tions.

REFERENCES

[6] (2011) Second ieee international workshop on performance evaluation of tracking and surveil-