Density-dependent quantized kernel least mean square with desired smoothing

To cite this article: Qing Lin et al 2019 J. Phys.: Conf. Ser. 1423 012068

View the article online for updates and enhancements.
Density-dependent quantized kernel least mean square with desired smoothing

Qing Lin¹, Yulei Huang¹ and Yan Li¹

¹ Intelligent Science and Information Engineering College, Xi’an Peihua University, Xi’an 710125, China

Corresponding author and e-mail: Qing Lin, 530610232@qq.com

Abstract. Kernel least mean square algorithm is a simple and effective adaptive algorithm, but it is constrained by its infinite network size. Many schemes have been proposed to reduce the size of the network, but the distribution of input data and the desire output are seldomly considered together. In this paper, we present a density correlation quantization scheme in input space and a method of smoothing the desired output by using the average value of the neighbourhood in output, and call it the DQKLMS-S algorithm. The good performance of DQKLMS-S is verified by experiments.

1. Introduction

The kernel method uses the Mercer kernel. In most cases, the rich expressiveness of kernel Gaussian function in mapping low-dimensional input space to high-dimensional feature space has made great achievements in batch kernel learning, such as support vector machine [1,2], kernel principal component analysis [3], online learning kernel (OKL) [4]. OKL incrementally learns the target function, especially when the data arrives in sequence.

Kernel adaptive filter [5] is one of the most famous OKL, including the kernel least mean square (KLMS) [6-7]. However, a common problem with kernel adaptive filters is that the network size grows as new data arrives. The growing network size presents computing and memory challenges. More importantly, overfitting can also occur if the network size is too large, making the algorithm too sensitive to outliers. Increasing the sparsity of the kernel adaptive filter is helpful to solve the complex calculation and overfitting problems.

Up to now, in order to restrain the growth of network size, various sparse technologies have been applied to the kernel adaptive filter. These technologies use sparse standards and accept only important input data as the new centers. Typical sparsity criteria include novelty criteria [8], predictive variance criteria [9], surprise criteria [10], and approximate linear correlation criteria [11]. A novel quantization scheme proposed [12] limits the network size and applies to the quantized kernel adaptive filter. The quantization algorithm greatly reduces the network size and even has better performance than KLMS.

In addition to network size, the representations of the input data and the output also affect generalization performance. In QKLMS, the distribution of quantization centers is approximately uniform in the input field. In fact, the distribution of the input data is often not uniform distributed. In order to improve the learning performance, the quantization technique should quantify the input data according to the distribution of the input data. Another problem in QKLMS is that the expected output of the quantified data is considered equal to the output of the nearest center. In many cases, the...
difference between the outputs of the neighborhood may be very large, especially when the underlying system is disturbed by impulsive noise. In order to deal with the above two problems, we consider a density correlation quantization scheme in input space and a method of smoothing the desired output by using the average value of the neighborhood in output, and call it the DQKLMS-S algorithm.

The rest of this article is organized as follows. The second part briefly reviews QKLMS. The new DQKLMS-S is proposed in section 3. In the fourth part, simulations on chaotic time series prediction are presented. Finally, we conclude the paper in the section 5.

2. Quantized kernel least mean square

The purpose of the adaptive filter is to learn a continuous input/output mapping $d(i) = f(u(i))$ based on the input/output sequence. $u(i) \in U$ is an $m$-dimensional input vector. The output is assumed to be one-dimensional $d(i)$.

The least mean square (LMS) [13] is an incremental linear algorithm that updates weight vectors using the following steps

$$
\begin{align*}
    f_0 &= 0 \\
    e(i) &= d(i) - f_{i-1}(u(i)) \\
    f_i &= f_{i-1} + \eta e(i)u(i)
\end{align*}
$$

where $e(i)$ is the prediction error, $\eta$ represents the step size, and the $f_i$ denotes the learned mapping at iteration $i$. The output of LMS in iteration $i$ is $d'(i) = f_{i-1}(u(i)) = w(i)^T u(i)$. Since LMS is a linear combination of input samples, LMS will fail if the mapping between inputs and outputs is non-linear.

In order to overcome the limitation of linearity while maintaining the simplicity of LMS, the input $u(i)$ is converted to the high-dimensional feature space [6] by the kernel induction mapping $\phi(u(i))$.

If the Mercer condition is satisfied in the feature space, the inner product can be expressed by positive definite kernel function, which is expressed as

$$
\kappa(u, u') = \exp \left( -\frac{\|u - u'\|^2}{2\sigma^2} \right)
$$

where $\sigma$ denotes the kernel size.

Similar to the learning rules in the original input space, KLMS in the feature space can be expressed as

$$
\begin{align*}
    f_0 &= 0 \\
    e(i) &= d(i) - f_{i-1}(u(i)) \\
    f_i &= f_{i-1} + \eta e(i)\kappa(u(i), \cdot)
\end{align*}
$$

As can be seen from the above equation, one input $u(i)$ corresponds to one coefficient $\eta e(i)$, resulting in the KLMS network being a growing radial-basis function network with the same size as the training set, which makes the calculation burden and memory demand very large.

In order to restrain the growth of KLMS network, QKLMS uses a simple online vector quantization method to quantify the input space. The learning rule of QKLMS is expressed as

$$
\begin{align*}
    f_0 &= 0 \\
    e(i) &= d(i) - f_{i-1}(u(i)) \\
    f_i &= f_{i-1} + \eta e(i)\kappa(Q[u(i)], \cdot)
\end{align*}
$$
where \( Q [\cdot] \) is the quantization operator in the input space, the output of \( Q[u(i)] \) is either the \( u(i) \) itself or the nearest center.

\[
Q[u(i)] = \begin{cases} 
C_j(i-1) & \text{dis}(u(i), C(i-1)) \leq \varepsilon \\
u(i) & \text{dis}(u(i), C(i-1)) > \varepsilon
\end{cases}
\]

(5)

where \( C(i) \) represents the center set, \( \text{dis}(u(i), C(i-1)) \) is the distance between \( u(i) \) and \( C(i-1) \), and \( \varepsilon \) denotes the quantization size

\[
\text{dis}(u(i), C(i-1)) = \min_{1 \leq j \leq \text{size}(C(i-1))} \| u(i) - C_j(i-1) \|
\]

(6)

where \( C_j(i-1) \) is the \( j \)-th element of the center set \( C(i) \), and

\[
f_j = \arg \min_{1 \leq j \leq \text{size}(C(i-1))} \| u(i) - C_j(i-1) \|
\]

(7)

If \( \text{dis}(u(i), C(i-1)) > \varepsilon \), a new center is added into the center set \( C(i) = \{ C(i-1), u(i) \} \), \( \alpha(i) = [\alpha(i-1), \eta \varepsilon(i)] \), if not, the coefficient of the closest center will be updated \( \alpha_j(i) = \alpha_j(i-1) + \eta \varepsilon(i) \).

Due to the fixed quantization threshold, the distribution of the center in the input data area in the dictionary is approximately consistent. Uniform distribution means that the quantization output of dense region is the same as that of sparse region, which cannot reflect the distribution of input data. What’s more, in QKLMS, the expected output of the quantified data is considered equal to the output of the nearest center. In many cases, the difference between the outputs of the neighborhood may be very large, especially when the underlying system is disturbed by impulsive noise. In order to deal with the above two problems, we consider a density correlation quantization scheme in input space and a method of smoothing the desired output by using the average value of the neighborhood in output.

3. Density-dependent quantized kernel least mean square with desired smoothing

In this section, we discuss a new algorithm based on QKLMS, which is improved in processing input quantization and smoothing output results. On this basis, a density correlation quantization scheme and a method of smoothing the desired output by using the average value of the neighborhood are proposed into QKLMS, and an input density correlation quantization kernel least mean square with desired smoothing (DQKLMS-S) can be expressed as

\[
\begin{align*}
    f_0 &= 0 \\
    e(i) &= D_j(i) - f_{i-1}(u(i)) \\
    f_i &= f_{i-1} + \eta \varepsilon(i) \kappa(Q[u(i)], .)
\end{align*}
\]

(8)

The output of \( Q[u(i)] \) is either the \( u(i) \) itself or the nearest center

\[
Q[u(i)] = \begin{cases} 
C_j(i-1) & \text{dis}(u(i), C(i-1)) \leq \varepsilon \\
u(i) & \text{dis}(u(i), C(i-1)) > \varepsilon
\end{cases}
\]

(9)

We introduce tow vectors \( N \) and \( D \). \( N \) is used to record the number of input data quantizing into each center. \( D \) denotes the smoothed desired output. If \( \text{dis}(u(i), C(i-1)) > \varepsilon_j \), we add the input as a
new center: $C(i) = \{C(i-1), u(i)\}$, $\varepsilon = [\varepsilon, \varepsilon_i]$, $\alpha(i) = [\alpha(i-1), \eta e(i)]$, $L = [L, 1]$, $N(i) = [N(i-1), 1]$, $D(i) = [D(i-1), d(i)]$. If $\frac{\varepsilon_j - \text{dis}(u(i), C(i-1))}{2} < \varepsilon_j$ and $L_j > \delta$, we update the quantization dictionary $C(i) = \{C(i-1), u(i)\}$, $\alpha(i) = [\alpha(i-1), \eta e(i)]$ and alter the quantization threshold $\varepsilon_j = \frac{\varepsilon_j}{2}$, $\varepsilon = [\varepsilon, \varepsilon_j]$ and $L_j = 1$. $L = [L, 1]$, $N(i) = [N(i-1), 1]$, $D(i) = [D(i-1), d(i)]$. Otherwise, the quantization center set remains unchanged $C(i) = C(i-1)$, and the effect of this input is added into the coefficient of the closest center: $\alpha_j(i) = \alpha_j(i-1) + \eta e(i)$, $L_j = L_j + 1$. We also update as follows: $N_j(i) = N_j(i-1) + 1$, and the smoothed desired output can be calculated as [5]

$$D_j(i) = \frac{N_j(i-1)D_j(i-1) + d(i)}{N_j(i)}$$

(10)

4. Simulation results

In this section, The Mackey-Glass (MG) chaotic time series prediction is considered. The time series is generated by a delay ordinary differential equations and is expressed as follows [14]:

$$\frac{dx(t)}{dt} = -bx(t) + \frac{ax(t-\tau)}{1 + x(t-\tau)^\sigma}$$

(11)

with $b = 0.1$, $a = 0.2$ and $\tau = 30$. The sampling period of the signal is 6 seconds. The goal of MG chaotic time series prediction is to use the previous points to predict the present value. 1000 samples are served as the training data, and 100 samples are served as the test data. The time embedding dimension is 7, that is the $u(i) = [x(i-1), x(i-2), ..., x(i-7)]^T$ is selected as the input vector to predict $x(i)$. During the simulation, the training data are interfered by additive noise. We consider two kinds of noise: Gaussian noise with mean of 0 and variance of 0.04; mixed Gaussian noise with density

$$0.95 \times N(0,0.04) + 0.05 \times N(0,1)$$

(12)

where $N(\mu, \sigma^2)$ represents the Gaussian density.

We compare the performance of QKLMS and DQKLMS-S. A Gaussian kernel of width 1.0 is selected as the Mercer kernel. The step sizes of these two algorithms are selected to obtain almost the same initial convergence rate. The quantization size is set as $\varepsilon = 0.2$. Simulation results averaged over 1000 independent Monte Carlo runs. Average learning curves and network growth curves are shown in Fig. 1 and Fig. 2, respectively. The test MSEs in the final iteration is shown in Table I. It can be clearly seen that whether Gaussian noise or mixed Gaussian noise, the DQKLMS-S has better performance than QKLMS. In this example, the network size is reduced to about 200, which is very small compared to the original network size (1000).
5. Conclusions
In this paper, for improving the learning performance of the QKLMS, a density correlation quantization scheme in input space and a method of smoothing the desired output by using the average value of the neighborhood in output have been introduced into QKLMS to form a new algorithm, called DQKLMS-S. In the new algorithm, the distribution of quantization centers is consistent with the distribution of input data, and the output is smoothed by the desired signals output. Simulation results show that the algorithm is effective.

Acknowledgement
Fund Program: University-level Scientific Research Program of Xi’an Peihua University in 2019, “Research on Intelligent Tourism Service Platform Based on 5G Network” (No. PHKT19033)
References

[1] Lima M D, Costa N L and Barbosa R. Improvements on least squares twin multi-class classification support vector machine. Neurocomputing, 313(3):196-205, 2018.

[2] Miao X, Liu Y, Zhao H and Li C. Distributed Online One-Class Support Vector Machine for Anomaly Detection Over Networks. IEEE Transactions on Cybernetics, 49(4):1-14, 2019.

[3] Fan J and Chow T W S. Exactly Robust Kernel Principal Component Analysis. IEEE Transactions on Neural Networks and Learning Systems, 1-13, 2019.

[4] Ding S, Mirza B, Lin Z, Cao J, Lai X and Tam V. Kernel based online learning for imbalance multiclass classification. Neurocomputing, 277(14):139-148, 2018.

[5] Liu W, Principe J C, and Haykin S. Kernel adaptive filtering: a comprehensive introduction, volume 57. John Wiley & Sons, 2011.

[6] Liu W, Pokharel P P and Principe J C. The kernel least-mean-square algorithm. Signal Processing, IEEE Transactions on, 56(2):543–554, 2008.

[7] Bouboulis P and Theodoridis S. Extension of wirtinger’s calculus to reproducing kernel hilbert spaces and the complex kernel lms. Signal Processing, IEEE Transactions on, 59(3):964–978, 2011.

[8] Platt J. A resource-allocating network for function interpolation. Neural computation, 3(2):213–225, 1991.

[9] Csato L and Opper M. Sparse on-line gaussian processes. Neural computation, 14(3):641–668, 2002.

[10] Liu W, Park I and Principe J C. An information theoretic approach of designing sparse kernel adaptive filters. Neural Networks, IEEE Transactions on, 20(12):1950–1961, 2009.

[11] Engel Y, Mannor S and Meir R. The kernel recursive least-squares algorithm. Signal Processing, IEEE Transactions on, 52(8):2275–2285, 2004.

[12] Chen B, Zhao S, Zhu P and Principe J C. Quantized kernel least mean square algorithm. Neural Networks and Learning Systems, IEEE Transactions on, 23(1):22–32, 2012.

[13] Haykin S and Widrow B. Least-mean-square adaptive filters, volume 31. John Wiley & Sons, 2003.

[14] Xu X, Qu H, Zhao J, Yang X and Chen B. Quantised kernel least mean square with desired signal smoothing. Electronics Letters, 51(18):1457-1459, 2015.