Calculation of "Optimal" Shape Parameters for RBF by LOOCV method

Anil Kumar Katwal
*Department of Physics and Astronomy, The University of Southern Mississippi, Hattiesburg, Mississippi, 39406 USA**

I. INTRODUCTION

The roots of Radial Basis Functions (RBFs) can be traced back to the early 1930s in the realm of numerical analysis. This period marked the emergence of a groundbreaking concept: utilizing functions that rely solely on distances for tasks like approximation and interpolation. Mathematician Hardy Multiquadrics was instrumental in introducing this concept, laying the foundation for what we now recognize as RBFs.

Moving forward into the 1960s and 1970s, there was a burgeoning interest in numerical methods and approximation techniques. During this time, early forms of RBFs, including the Thin Plate Spline, began to take shape. These functions were specifically designed for interpolating scattered data points, a task at which RBFs exhibited exceptional proficiency.

The subsequent decades, spanning from the 1980s to the 1990s, witnessed a remarkable surge in the prominence of RBFs, particularly within the burgeoning field of machine learning. This epoch saw the integration of RBFs into a revolutionary neural network architecture known as Radial Basis Function Networks. These networks harnessed the power of RBFs to transform input data into a higher-dimensional space, enabling intricate classification tasks.

Furthermore, RBFs found a pivotal role in the development of Support Vector Machines (SVMs) during this period. These powerful machine learning algorithms harnessed RBFs as kernel functions, enabling them to tackle classification and regression challenges with remarkable effectiveness. Among the various forms of RBFs, the Gaussian variant, distinguished by its capacity to non-linearly separate data, gained substantial traction.

In the ensuing years, RBFs transcended their origins in machine learning, finding application in a diverse array of fields. From numerical analysis to computational mathematics, geostatistics to computer graphics, and even computer-aided design, RBFs became indispensable tools for a wide range of scientific and engineering applications.

In conclusion, the development of RBFs is a testament to their enduring significance in various scientific and mathematical domains. From their early roots in numerical analysis to their pivotal role in machine learning and beyond, RBFs have evolved into an essential tool, contributing significantly to the advancement of numerous fields.The Radial Basis Function (RBF) is a mathematical function that plays a fundamental role in various fields, including machine learning, numerical analysis, and computational mathematics. It derives its name from its unique property of exhibiting radial symmetry around a central point. The essence of the RBF lies in its definition, which hinges on the distance between an input point and a fixed center point. This relationship is encapsulated in the mathematical expression

$$\phi(x) = \phi(\|x - c\|)$$

where x represents the input point, c signifies the center of the function, and $\|x - c\|$ denotes the Euclidean distance between the two. One of the most prevalent forms of the RBF is the Gaussian RBF, defined as

$$\phi(x) = \exp\left(-\gamma\|x - c\|^2\right)$$

where γ governs the extent of the function's spread.. The value of the function is determined by the distance between the input point and a specific center, usually in the context of Euclidean distance. The function's value is high (close to 1) when the input point is near the center and decreases as the distance increases.

Type of Radial Basis Functions

1. **Gaussian RBF**:

$$\phi(r) = e^{-\epsilon r^2}$$

2. **Multiquadric RBF**:

$$\phi(r) = \sqrt{1 + (\epsilon r)^2}$$

3. **Inverse Multiquadric RBF**:

$$\phi(r) = \frac{1}{\sqrt{1 + (\epsilon r)^2}}$$

4. **Thin Plate Spline RBF**:

$$\phi(r) = r^2 \log(r)$$

5. **Cubic RBF**:

$$\phi(r) = r^3$$

6. **Linear RBF**:

$$\phi(r) = r$$

7. **Quadratic RBF**:

$$\phi(r) = r^2$$

8. **Polyharmonic Spline RBF**:

$$\phi(r) = r^{2k+1} \quad \text{where } k \text{ is a positive integer}$$

* Anil.Katwal@usm.edu

9. **Triharmonic Spline RBF**:

$$\phi(r) = r^3$$

In our calculation,

$$\phi(r) = \sqrt{1 + (\epsilon r)^2}$$

is used.

II. PROBLEMS

The function

$$f(x, y, z) = 64x(1 - x)y(1 - y)z(1 - z)$$

is a three-dimensional "bumpy" function defined in the domain$[0, 1]^3$. Choose 500 Halton points as the interpolation points and use RBF-MQ as the basis function for volume interpolation. Try to implement LOOCV for the optimal shape parameter of MQ.

1. Choose another set of 300 Halton points in the domain as the test points and compute the maximum absolute error for the interpolation.

2. Using 40x40x40 mesh grid to plot the iso-surface corresponding to the function value 0.01 and 0.8.

3. Use slice command to plot the approximate function and absolute errors at different levels.

III. METHOD

A. Determination of optimal parameter

$$Rbf_{mq} = \sqrt{(ep \cdot r)^2 + 1} \tag{1}$$

This formula ,$Rbf_{mq} = \sqrt{(ep \cdot r)^2 + 1}$ known as a radial basis function (RBF) In our calculation following function was used to get optimal parameter ϵ [1].

```
function ceps = CostEps(ep,rbf,DM,rhs)
    A = rbf(ep,DM);
    invA = pinv(A);
    errorvector = (invA*rhs)./diag(invA);
    ceps = norm(errorvector);
end
```

Here rbf needs to provide a MATLAB function that can generate the interpolation matrix A based on a shape parameter ep and a Matrix DM of all the pairwise distance$\|x_i - x_j\|$ among the datasites.

B. Error Calculation

we can estimate error by splitting the data set into two parts: one to base an approximation to the interpolant on, and the other to base an estimate for the error on.. In LOOCV one splits off only a single data point at which to compute the error, and then bases the approximation to the interpolant on the remaininng N-1 data points. This procedure is in turn repeated for each one of the N data points. In Leave-One-Out Cross-Validation (LOOCV), the error is typically calculated using a metric like Mean Squared Error (MSE) or Mean Absolute Error (MAE), depending on the nature of your problem (regression or classification).Here's a general outline of how you calculate the error in LOOCV: The mathematical expression for Leave-One-Out Cross-Validation (LOOCV) error in the context of a Radial Basis Function (RBF)can be defined as follows. Let's consider a dataset with n data points denoted as $(\mathbf{x}_i, y_i)$ for $i = 1, 2, \ldots, n$, where $\mathbf{x}_i$ represents the input vector and y_i is the corresponding target value. The LOOCV process involves iteratively leaving out one data point, training the RBF network on the remaining $n-1$ points, and then using the trained model to predict the target value for the left-out point. The prediction error for each left-out point is calculated as $\epsilon_i = (y_i - \hat{y}_i)^2$, where y_i is the true target value and $\hat{y}_i$ is the predicted target value. The LOOCV error E is then determined by summing up all the squared prediction errors: $E = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$. This mathematical expression quantifies the overall performance of the RBF network by assessing the squared prediction errors over all data points in the dataset. The objective is to minimize this error by appropriately adjusting the RBF network's parameters during training.

IV. RESULTS

we obtained an optimal shape parameter, ϵ, which was calculated to be approximately 0.91102 to 1.120 in different range.This parameter was crucial in determining the behavior of the radial basis function, and having found an optimal value meant that the function was finely tuned to the characteristics of the data.

Furthermore, we observed a maximum absolute error of approximately 2.0295×10^{-9}. This error metric provided valuable insights into how well the model fit the data. A small maximum absolute error indicated that the model closely aligned with the actual observations, suggesting a high level of accuracy.

To gain a deeper understanding, we used MATLAB to create visual representations of the exact solution and absolute error. we employed a technique known as slicing, where we cut through the three-dimensional space defined by the function to visualize it in two dimensions. Slicing provides a cross-sectional view, allowing for a detailed examination of the function's behavior at specific planes. This technique is invaluable for identifying patterns and understanding the intricate variations within the data.

In addition to slicing, we utilized iso-surfaces, which are three-dimensional surfaces that represent points of constant value within the function. These surfaces provide a comprehensive view of regions where the function exhibits similar behavior. By visualizing iso-surfaces, we could identify regions of interest and make informed decisions about parameter tuning.

These results collectively demonstrated that the radial basis function, with the chosen ϵ value, provided an accurate representation of the underlying data, making it a reliable tool for our calculations.

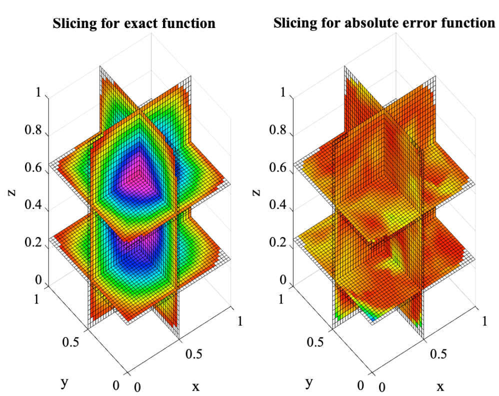

FIG. 1. exact and absolute error slice

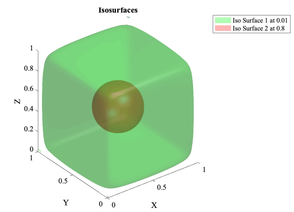

FIG. 2. Iso-surface plot Iso-surfaces at 0.01 and 0.8

The scatterplot presented in Figure 3 distinctly demonstrates that the optimal parameter range falls between

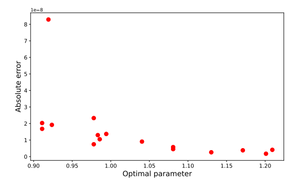

FIG. 3. A scatter plot that shows how absolute error changes with the optimal parameter value reveals an interesting pattern. It turns out that when the optimal parameter is around 1.1, the absolute error is consistently at its lowest point.

TABLE I. Table for Different range,Optimal parameter ϵ, and absolute error

Range	Parameter ϵ	absolute error
$0-1$	0.75447	$4.90X10^{-6}$
$0-2$	1.2357	$8.09X10^{-10}$
$0-3$	1.1302	$2.62X10^{-9}$
$0-4$	0.97795	$2.33X10^{-8}$
$0-5$	1.0808	$4.53X10^{-9}$
$0-6$	0.91922	$8.29X10^{-8}$
$0-7$	1.0404	$9.08X10^{-9}$
$0-8$	1.2093	$4.10X10^{-9}$
$0-9$	0.91104	$1.68X10^{-8}$
$0-10$	0.91102	$2.03X10^{-8}$
$0-11$	0.99414	$1.37X10^{-8}$
$0-12$	0.92366	$1.92X10^{-8}$
$0-13$	1.1711	$3.81X10^{-9}$
$0-14$	1.0808	$5.75X10^{-9}$
$0-15$	1.2011	$1.75X10^{-9}$
$0-16$	0.98318	$1.30X10^{-8}$
$0-17$	0.97796	$7.46X10^{-9}$
$0-18$	0.98568	$1.05X10^{-8}$

0.75447 and 1.235. Notably, when the optimal parameter ranged from above 1.0 to 1.2, a stable absolute error was consistently achieved. This observation was further corroborated by manipulating the range of maximum and minimum values. Specifically, the minimum value was set at zero, while the maximum value was varied from 1 to 18, as outlined in Table I.

[1] Fasshauer, G. E., and J. G. Zhang (2007), Numerical Algorithms **45**, 345.