Journal of Physics: Conference Series

PAPER • OPEN ACCESS

Solving inverse boundary value problem of Poisson equation by LS-SVM

To cite this article: Minghao Huang *et al* 2024 *J. Phys.: Conf. Ser.* **2852** 012005

Journal of Physics: Conference Series **2852** (2024) 012005 doi:10.1088/1742-6596/2852/1/012005

Solving inverse boundary value problem of Poisson equation by LS-SVM

Minghao Huang[1], Ziku Wu[2] and Xiaoming Han[3,4]

[1] Department of Electrical and Electronic Engineering, Hong Kong Polytechnic University, Hong Kong, China;
[2] Science and Information College, Qingdao Agricultural University, Qingdao, China;
[3] School of Electrical Engineering, YanShan University, Qinhuang Dao, China.

[4] Corresponding author's e-mail: hanxiaoming@ysu.edu.cn

Abstract. In this paper, a new method based on least squares support vector machines (LS-SVM) is presented for solving the inverse boundary value problem of Poisson equation. The closed form analytical solution is obtained by optimizing the regression parameters. The core problem is to transform the parametric regression problem into a quadratic programming problem. To demonstrate the efficiency of the proposed algorithm, numerical experiments are conducted. The proposed method is found to be feasible for the inverse boundary value problem of Poisson equation.

1. Introduction

Some physical phenomena and engineering problems can be expressed by Poisson equation [1], for instance, electrostatic interactions in biological and chemical systems at molecular level [2,3], and the design of semiconductor devices at the nanoscale [4]. Many researchers have devoted to find an effective method for solving inverse boundary value problem of Poisson equation, such as analytical method, finite element method (FEM) and finite difference method (FDM). However, those methods are computational complexity since they require the mesh degeneration that obtains the accurate solutions. To solve the Cauchy problem for Poisson equation, the reference [5] investigated logarithmic convexity for discrete harmonics function, but the computational complexity can't be obviously reduced.

As LS-SVM method introduced regularization technology, it can overcome some limitations of the above methods to some extent. LS-SVM method has shown strong practicability since its appearance, and now it has been widely used in various problems, such as pattern recognition, classification and approximation [6,7]. Recently, LS-SVM is used to solve differential equations, which including ordinary differential equation, partial differential equation and inverse problem. Compared with traditional methods (FEM and FDM), LS-SVM can obtain closed form approximate analytical solutions [8]. The aim of this paper is to propose a method based on LS-SVM to solve the inverse boundary value problem of Poisson equation.

2. Introduction to the approximate method based on LS-SVM with mixture kernels function

Consider the following problem:

$$u_{xx} + u_{yy} = S(x, y), \quad (x, y) \in \overset{o}{\Omega} \tag{1}$$

$$u(\bar{x}_i, \bar{y}_i) = f_i,\ (\bar{x}_i, \bar{y}_i) \subset \Gamma \subset \partial\Omega \tag{2}$$

$$\left.\frac{\partial u}{\partial n}\right|_{(\bar{x}_i, \bar{y}_i)} = g_i,\ (\bar{x}_i, \bar{y}_i) \subset \Gamma \subset \partial\Omega \tag{3}$$

where S is a known function, Γ is part of the boundary $\partial\Omega$. Formulas (2) and (3) are Dirichlet condition and Neumann condition, respectively. However, only a part of boundary conditions be known. Therefore, the problem is an inverse problem. In fact, the numerical solution of the inverse problem is an approximate to the analytic solution. In this paper, we will introduce a function approximation method which based on LS-SVM with mixture kernels function, and then employ it to solve the inverse problems. Let $u(x, y)$ be a continuous and differentiable function on the bounded domain $\Omega \subset R^2$, which known on some interior points and partial boundary. Without loss of generality, we denote it as follows:

$$u(p_j) = u_j,\ p_j = (x_i, y_j) \in V_D \tag{4}$$

$$u(\bar{p}_j) = f_j,\ \bar{p}_j = (\bar{x}_i, \bar{y}_j) \in \bar{V}_B \subset \Gamma \subset \partial\Omega \tag{5}$$

$$\left.\frac{\partial u}{\partial n}\right|_{\hat{p}_i} = g_i,\ \hat{p}_i = (\hat{x}_i, \hat{y}_j) \in \bar{V}_B \subset \Gamma \subset \partial\Omega \tag{6}$$

where V_D is the interior points set, $\bar{V}_B$ and $\hat{V}_B$ are the boundary points sets of Γ. The members of the three sets are $|V_D| = N$, $|\bar{V}_B| = M_1$ and $|\hat{V}_B| = M_1$, respectively. Our purpose is to seek a function $u(x, y)$ with continuous partial derivative under formulas (4)-(6) constraints. According to the principles of LS-SVM, we can define $u(x, y)$ as following expression:

$$u(p) = \sum_{j=1}^{N} \alpha_j \mathrm{K}(p - p_j) + b,\ p = (x, y) \tag{7}$$

where K is the kernels function, $\alpha^{\mathrm{T}} = [\alpha_1 \alpha_2 \cdots \alpha_N]$ and b are the regression coefficients that have to be determined. In this study, K is constructed as the weighted combination of Gaussian function and polynomial function.

$$\mathrm{K}(p - p_j) = w \bullet \exp\left(-\frac{< p - p_j, p - p_j >}{2\sigma^2}\right) + (1 - w) \bullet (1 + < p, p_j >)^k \tag{8}$$

where $<, >$ denotes the inner product, w is the weight, k is the degree of the polynomial, and σ is the width of the kernels function. In order to estimate the regression coefficients, it can be transformed into the following quadratic programming problem:

$$\min_{\alpha, b} \frac{1}{2} \alpha^T \alpha + \frac{\gamma}{2} \left(e^T e + \bar{e}^T \bar{e} + \hat{e}^T \hat{e} \right) \tag{9}$$

$$s.t. \quad \begin{cases} \sum_{j=1}^{N} \boldsymbol{\alpha}_j K\left(p_i - p_j\right) + b - u_i + e_i = 0, & p_i \in V_D \\ \sum_{j=1}^{N} \boldsymbol{\alpha}_j K\left(\bar{p}_i - p_j\right) + b - f_i + \bar{e}_i = 0, & \bar{p}_i \in \bar{V}_B \\ \sum_{j=1}^{N} \boldsymbol{\alpha}_j \bar{K}\left(\hat{p}_i - p_j\right) - g_i + \hat{e}_i = 0, \hat{p}_i \in \hat{V}_B \end{cases} \tag{10}$$

where $\gamma > 0$ is the regularization factor, $\overline{K}\left(\hat{p}_i - p_j\right)$ is the outer normal derivative of $K(\overline{p}_i - p_j)$ at point $e^T = [e_1\ e_2 \cdots e_N]$, $\overline{e}^T = [\overline{e}_1\ \overline{e}_2 \cdots \overline{e}_{M_1}]$, $\hat{e}^T = [\hat{e}_1\ \hat{e}_2 \cdots \hat{e}_{M_2}]$ are introduced bias terms.

Theorem 1. Given the kernels width σ and the regularization factor γ, the solution of the above quadratic programming problem is obtained by solving the following linear equations:

$$\begin{bmatrix} m_{11}\ m_{12}\ m_{13}\ m_{14}\ m_{15} \\ m_{21}\ m_{22}\ m_{23}\ m_{24}\ m_{25} \\ m_{31}\ m_{32}\ m_{33}\ m_{34}\ m_{35} \\ m_{41}\ m_{42}\ m_{43}\ m_{44}\ m_{45} \\ m_{51}\ m_{52}\ m_{53}\ m_{54}\ m_{55} \end{bmatrix} \begin{bmatrix} \alpha \\ \lambda \\ b \\ \eta \\ \mu \end{bmatrix} = \begin{bmatrix} Y \\ U \\ 0 \\ F \\ G \end{bmatrix} \tag{11}$$

where $\lambda^T = [\lambda_1\ \lambda_2 \cdots \lambda_N]$, $\eta^T = [\eta_1\ \eta_2 \cdots \eta_{M_1}]$ and $\mu^T = [\mu_1\ \mu_2 \cdots \mu_{M_1}]$ are Lagrange multipliers. The meanings of the block matrix of the system are $m_{11} = I_N$ (Identify matrix of order N), $m_{12} = (K(p_j - p_i))_{N\times N}$, $m_{13} = (0)_{N\times 1}$, $m_{14} = (K(\overline{p}_j - p_i))_{N\times M_1}$, $m_{15} = (K(\overline{p}_j - p_i))_{N\times M_2}$, $m_{21} = m_{12}^T$, $m_{22} = -\frac{1}{\gamma} I_N$, $m_{23} = (1)_{N\times 1}$, $m_{24} = (0)_{N\times M_1}$, $m_{25} = (0)_{N\times M_2}$, $m_{31} = (0)_{1\times N}$, $m_{32} = (1)_{1\times N}$, $m_{33} = 0$, $m_{34} = (1)_{1\times M_1}$, $m_{35} = (0)_{1\times M_2}$, $m_{41} = m_{14}^T$, $m_{42} = (0)_{M\times N}$, $m_{43} = (1)_{M_1\times 1}$, $m_{44} = -\frac{1}{\gamma} I_{M_1}$, $m_{45} = (0)_{M_1\times M_2}$, $m_{51} = m_{15}^T$, $m_{52} = (0)_{M\times N}$, $m_{53} = (1)_{M_2\times 1}$, $m_{54} = (0)_{M_2\times M_1}$, $m_{55} = -\frac{1}{\gamma} I_{M_2}$, $Y = (0)_{N\times 1}$, $U = (u_i)_{N\times 1}$, $F = (f_i)_{M_1\times 1}$, $G = (g_i)_{M_2\times 1}$

It is easy to prove this theorem using the Lagrange multiplier method, so it is omitted here. In order to test the approximation of the proposed method, an example is provided below. The function is $u(x, y) = \frac{x - y}{2 + \sin(xy)}$, and the domain is $\Omega = \left\{(x, y) \mid 1 \le x^2 + y^2 \le 4\right\}$. Without loss of generality, observation information is provided at the boundary $\Gamma = \left\{(x, y) \mid x^2 + y^2 = 4\right\}$. The training interior points are $\begin{cases} r = 1.05:0.05:1.95 \\ \theta = 0:9:351 \end{cases}$. It is clearly $|V_D| = 760$. Letting the boundary points $\begin{cases} r = 2 \\ \theta = 0:1:359 \end{cases}$, so we have $M_1 = M_2 = 360$. Considering the function of the regularization factor γ, we can take it as a constant. In this study, we take $\gamma = 10^8$ for all examples. Moreover, the degree of the polynomials is fixed to 3. The kernels width parameter σ is the only adjustable parameter. After a lot of experiments, we found $\sigma = 0.4$ is more suitable. Next, we set up the testing sets. The interior testing points set $T_{ia} = \{(r, \theta) \mid r = 1.01:0.01:1.99, \theta = 0:1:359\}$, the member is 35640. The maximum error (ME) and the root mean square error (RMSE) of the regression value are 2.66e-3 and 6.36e-4, respectively. We choose $T_{ib} = \{(1, \theta) \mid \theta = 0:1:359\}$ as the testing boundary. The function value and the outer normal derivative are the test indicators. The results are listed in Table 1. The approximation accuracy of the proposed method is as expected. Next, we apply it to solve the inverse boundary value problem.

3. Formation of the proposed method for inverse boundary value problem of Poisson equation

For convenience, we rewrite the inverse problems as follows:

$$Lu(p)=S(p),\ p=(x,y)\in\overset{o}{\Omega} \tag{12}$$

$$u(\bar{p}_i)=f_i,\ \bar{p}_i=(\bar{x}_i,\bar{y}_i)\in\bar{V}_B\subset\Gamma\subset\partial\Omega \tag{13}$$

$$\frac{\partial u}{\partial n}\Big|_{\bar{p}_i}=g_i,\ \hat{p}_i=(\hat{x}_i,\hat{y}_i)\in\hat{V}_B\subset\Gamma\subset\partial\Omega \tag{14}$$

where L denotes $Lu=u_{xx}+u_{yy}$. Note that the function's information is provided only at the partial boundary. The meanings of these symbols are the same as the above section. Assume the approximate solution of the problem (12)-(14) expression:

$$u(p)=\sum_{j=1}^{N}\alpha_j\mathrm{K}(p-p_j)+b \tag{15}$$

Inserting (15) into formulas (12)-(14), we can obtain the following equations

$$\sum_{j=1}^{\mathrm{N}}\alpha_j\mathrm{H}(p_i-p_j)-S_i+e_i=0,\ i\in\mathrm{N} \tag{16}$$

$$\sum_{j=1}^{\mathrm{N}}\alpha_j\mathrm{K}(\bar{p}_i-p_j)+b-f_i+\bar{e}_i=0,\ i\in\mathrm{M}_1 \tag{17}$$

$$\sum_{j=1}^{\mathrm{N}}\alpha_j\overline{\mathrm{K}}(\hat{p}_i-p_j)-g_i+\hat{e}_i=0,\ i\in\mathrm{M}_2 \tag{18}$$

where $\overline{\mathrm{K}}(\hat{p}_i-p_j)$ is the outer normal derivative of $\mathrm{K}(\bar{p}_i-p_j)$ at point $\hat{p}_i$. $\mathrm{H}(p_i-p_j)$ is the value of $L(\mathrm{K}(p-p_j))$ at the point p_i. The symbol $\tilde{N}$ denotes the set $\{1,2,\cdots,\mathrm{N}\}$.

Remark 1. Note that (16)-(18) is slightly different from (4)-(6). Here $u(p_i)$ is unknown that to be determined.

In order to estimate the regression coefficients, it can be transformed into the following quadratic programming problem:

$$\min_{\alpha,b}\frac{1}{2}\alpha^T\alpha+\frac{\gamma}{2}[e^Te+\bar{e}^T\bar{e}+\hat{e}^T\hat{e}] \tag{19}$$

$$\text{s.}t.\ \begin{cases}\sum_{j=1}^{\mathrm{N}}\alpha_j\mathrm{H}(p_i-p_j)-S_i+e_i=0,i\in\overline{\mathrm{N}}\\ \sum_{j=1}^{\mathrm{N}}\alpha_j\mathrm{K}(\bar{p}_i-p_j)+b-f_i+\bar{e}_i=0,i\in\overline{\mathrm{M}}_1\\ \sum_{j=1}^{\mathrm{N}}\alpha_j\overline{\mathrm{K}}(\hat{p}_i-p_j)-g_i+\hat{e}_i=0,i\in\mathrm{M}_2\end{cases} \tag{20}$$

Theorem 2. Given the kernels width σ and the regularization factor γ, the solution the above quadratic programming problem (19)-(20) is obtained by solving the following linear equations:

$$\begin{bmatrix} m_{11}\ m_{12}\ m_{13}\ m_{14}\ m_{15}\\ m_{21}\ m_{22}\ m_{23}\ m_{24}\ m_{25}\\ m_{31}\ m_{32}\ m_{33}\ m_{34}\ m_{35}\\ m_{41}\ m_{42}\ m_{43}\ m_{44}\ m_{45}\\ m_{51}\ m_{52}\ m_{53}\ m_{54}\ m_{55}\end{bmatrix}\begin{bmatrix}\alpha\\ \lambda\\ b\\ \eta\\ \mu\end{bmatrix}=\begin{bmatrix}Y\\ U\\ 0\\ F\\ G\end{bmatrix} \tag{21}$$

Remark 2. Most of the symbols in formula (21) have the same meaning as those in formulas (11). The symbols with different meanings are $m_{12} = (\mathrm{H}(p_j - p_i))_{N\times N}$, $m_{14} = (\mathrm{K}(\bar{p}_j - p_i))_{N\times M_1}$, $m_{15} = (\bar{K}(\hat{p}_j - p_i)\ U = (S_i)_{N\times 1}$. Similarly to Theorem 1, we also omitted the proof.

4. Numerical examples

In this section, we will test the performance of the proposed method on two examples, denoting as Exp1 and Exp2. In Exp1: $S=0$, $u(x,y)=\frac{1}{4}(x^2-y^2-2x+8y)$, $\Omega=\{(x,y)\big|1\le x^2+y^2\le 4\}$, and the unknown boundary is $x^2+y^2=2$, $N=760$, $M_1=M_2=90$. In Exp2: $S=-2\pi^2\sin(\pi y)\sin(\pi y)$, $u(x,y)=\sin(\pi y)\sin(\pi y)$, $\Omega=\{(x,y)\big|0\le x,y\le 1\}$, and the unknown boundary is $x=0$ and $x=1$, $N=361$, $M_1=M_2=98$. We add 5% random error into the observation in order to verify the robustness of the proposed method in Exp1 (See Table 2). The RMSE of numerical results is also at the level of the noise. The formula for adding random error is as follows: $f = f + 0.1\|f\|(rand(size(f),1)-0.5)$ and $g = g + 0.1\|g\|(rand(size(g),1)-0.5)$. The optimal parameter σ for Exp1 and Exp2 is 0.6 and 0.5, respectively. The results are shown in Table 2 and Table 3.

Table 1. Approximate results on Γ_{ib} .

w	Function value		Outer normal derivative	
	ME	RMSE	ME	RMSE
1.00	5.53e-3	5.83e-3	1.37e-1	5.05e-2
0.80	2.08e-3	8.90e-4	4.33e-2	1.60e-2
0.70	2.53e-3	9.57e-4	3.95e-2	1.583-2
0.60	3.16e-3	1.07e-3	5.47e-2	1.76e-2
0.40	4.34e-3	1.58e-3	9.27e-2	2.73e-2

Table 2. The numerical results of Exp1.

	Training error		Testing error		Test boundary Function		Test boundary Outer normal derivative	
W	ME	RSME	ME	RSME	ME	RSME	ME	RSME
0.00	3.00e-6	9.90e-7	3.14e-6	9.95e-7	3.16e-6	1.48e-6	3.00e-6	1.45e-6
0.25	1.12e-4	5.06e-5	1.16e-4	5.09e-5	9.24e-5	4.75e-5	9.05e-5	4.55e-5
0.50	1.16e-4	4.17e-5	1.20e-4	4.19e-5	1.20e-4	5.68e-5	1.62e-4	9.92e-5
0.50*	8.32e-2	2.77e-2	9.05e-2	2.79e-2	9.22e-2	4.32e-2	1.71e-1	6.31e-2
0.75	3.23e-4	1.63e-4	3.28e-4	1.64e-4	3.26e-4	1.82e-4	3.93e-4	2.17e-4
1.00*	1.80e-2	7.83e-3	1.86e-2	7.88e-3	1.88e-2	1.53e-2	2.33e-2	1.26e-2

* With 5% rand error in the observation data

Table 3. The numerical results of Exp2.

	Training error		Testing error		Test boundary Function		Test boundary Outer normal derivative	
W	ME	RSME	ME	RSME	ME	RSME	ME	RSME
0.00	0.2981	0.1255	0.3587	0.1275	0.3811	0.1560	2.2623	1.1001
0.25	4.26e-3	1.78e-3	5.03e-3	1.84e-3	5.25e-3	3.26e-3	3.35e-2	1.67e-2
0.50	4.17e-3	1.60e-3	4.92e-3	1.65e-3	5.14e-3	2.95e-3	3.17e-2	1.55e-2
0.75	3.64e-3	1.41e-3	4.29e-3	1.45e-3	4.48e-3	2.62e-3	2.83e-2	1.40e-2
1.00	2.67e-3	1.11e-3	3.14e-3	1.14e-3	3.28e-3	2.08e-3	2.23e-2	1.13e-2

5. Conclusions

In this study, a new numerical method that based on LS-SVM is proposed to solve the inverse boundary value problem for Poisson equation. This method can inverse the information on the unknown boundary through some given Dirichlet and Neumann boundary conditions on the accessible part of the domain.

Journal of Physics: Conference Series **2852** (2024) 012005 doi:10.1088/1742-6596/2852/1/012005

In addition, the method provides a closed form analytical solution. We compare the numerical solutions with the analytical solutions. The numerical results show that the method is effective and stable. In the future, the proposed method may be extended to solve nonlinear inverse problems.

References

[1] D. N. Arnold, F. Brezzi and B. Cockburn 2001 Unified analysis of discontinuous galerkin methods for elliptic problems *Slam J. Numer.Anal.* 39 1749-79
[2] Y. Taur and T. H. Ning 1998 *Fundamentals of Modern VLSI Devices* (Cambridge University Press)
[3] W. A. Strauss 2007 *Partial Differential Equations* (Joans Wiley & Sons, Ltd, Hoboken)
[4] A. F. Dafaalla, K. Hwang and S. Ahn 2010 A new finite difference representation for Poisson's equation on R3 from a contour integral *Appl.Math.Comput.* **217** 3624-34
[5] R. S. Falk and P. B. Monk 1986 Logarithmic convexity for discrete harmonic functions and the approximation of the Cauchy problem for Poisson's equation *Math.Comput.* **47** 135-149
[6] J. A. K. Suykens, T. V. Gestel and J. D. Brabanter 2002 *Least Squares Support Vector Machines* (World Scientific, Singapore)
[7] J. A. K. Suykens and J. Vandewalle 1999 Least squares support vector machine classifiers *Neural Process Lett.* **9** 293-300
[8] X. M. Han, J. J. Wang and Z.K. Wu 2018 Learning solutions to two dimensional electromagnetic equations using LS-SVM *Neurocomputing* **317** 15-27