Int. J. Nonlinear Anal. Appl. 13 (2022) 2, 2233–2246
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2021.20548.2170

A machine learning approach for solving inverse Stefan problem

Kourosh Parand[a,b,c,*], Ghazalsadat Ghaemi Javid[a], Mostafa Jani[a]

[a]Department of Computer Sciences, Faculty of Mathematical Sciences, Shahid Beheshti University, G.C. Tehran, Iran

[b]Department of Cognitive Modeling, Institute for Cognitive and Brain Sciences, Shahid Beheshti University, G.C. Tehran, Iran

[c]Department of Statistics and Actuarial Science, University of Waterloo, Waterloo, Canada

(Communicated by Madjid Eshaghi Gordji)

Abstract

In this paper, we propose a numerical scheme by using Least Squares Support Vector Regression (LS-SVR) for the simulation of the inverse Stefan problem, which has ill-posedness issues. The purpose of this paper is to express the temperature distribution in a homogeneous environment with a phase change. In the proposed machine learning approach, we apply the unconditionally stable Crank-Nicolson method to decrease the computational cost and reduce one of the dimensions. Therefore, we solve an ODE equation at each time step. The training points of the network are chosen as the Chebyshev roots, which have a normal distribution, and our constructed roots, which we describe more precisely later. In the proposed method, the regularization parameter of the SVM aims to overcome the instability issues, leading to convergent approximation. For the given method, both the primal and dual forms are investigated. The dual form of the problem is written in matrix form. Finally, some numerical examples are provided to illustrate the effectiveness and accuracy of the proposed method.

Keywords: Least squares support vector regression, Orthogonal kernel, Inverse Stefan problem, Collocation method, Chebyshev polynomials
2020 MSC: Primary 35R30, Secondary 68T01

1 Introduction

In the inverse problems, unlike the direct problems, the aim is to seek the inputs of the model. Two different inverse problems are encountered in modeling engineering problems, the input estimation problems which have the output and properties of the model and the input will be obtained, and properties estimation problems which have the output and input and the properties of the problem will be achieved [12, 5]. Inverse problems have vast applications in various fields such as imaging [8, 17], electrodynamic and elasticity [13], optics [25], damage identification [31], medical imaging [18], machine learning [1], modeling brain tumor treatment [6], and the like.
There are various Stefan inverse problems, in which the initial conditions, the boundary conditions or free boundaries, should be determined [9]. This paper is concerned with a one-dimensional one-phase inverse Stefan problem.
There are some numerical methods for inverse Stefan problem, including the meshless methods [26, 27], Galerkin method [10], collocation method [29] and the other methods such as perturbation method [24], hybrid of finite element

*Corresponding author
Email addresses: k_parand@sbu.ac.ir (Kourosh Parand), ghazal.ghaemi95@gmail.com (Ghazalsadat Ghaemi Javid), mostafa.jani@gmail.com (Mostafa Jani)

Received: June 2020 *Accepted:* November 2021

method and artificial bee colony algorithm [19], and dynamic programming [36].
Neural network methods are commonly used for differential equations like SVM and LS-SVM methods [20, 21]. Also, new approaches to solve inverse problems are neural network methods [3, 11, 2].
In this paper, we develop a numerical method for the inverse Stefan problem based on the LS-SVR model. The problem is solved in both primal and dual form. The novelty of the proposed method is to solve the one-phase inverse Stefan problem by applying the LS-SVR method with the orthogonal kernel, resulting in sparse systems. The numerical results show that the method has suitable convergence.

The rest of the paper is organized as follows: Section 3 is devoted to SVM for both classification and regression and also SVM's kernels. Section 4 is assigned to Chebyshev polynomials. Section 5 discusses our proposed method, which we solve the problem in primal and dual form. Next, we show the numerical results of our approach in Section 6. Finally, some concluding remarks are given in Conclusion.

2 Preliminaries

3 SVM for classification and regression

Machine learning algorithms are a subset of artificial intelligence that provides the appropriate output according to the task it receives as input. Typical machine learning systems are based on extracting features and classifying them. However, in deep learning procedures, the feature extraction part is eliminated, and the learning is done entirely in the network itself, i.e., end-to-end learning. One of the supervised learning methods is the support vector machines, a particular type of which is the least squares support vector machines (LS-SVM).
As aforementioned, the support vector machines are one of the supervised learning methods, which are used for classification and regression. That is, the support vector machines are a boundary sorting method that gives us the best classification and separation of data by benchmarking the support vectors. Support vectors are a set of points in the n-dimensional space of the data, based on which data is sorted and categorized, and by moving one of them, the output of the categorization may change. In 2D space, the support vectors will form a line in the 3D space form a page, and the n-dimensional space form a hyperplane. In the support vector machines, only support vectors are based on machine learning and model building, and this algorithm is not sensitive to other data points. Also, its purpose is to find the best boundary between the data as far as possible from all categories.

3.1 SVM-Classification

given training set $(x_i, y_i), i = 1, ..., N$ where x_i is training input in $\mathbb{R}^n$, and y_i is the desired output in $\mathbb{R}$. In the context of separable SVM classification, our goal is to separate two classes with a line in 2D space to have a maximum margin so that their support vectors create the widest distance. Their separator line is equal to $w^T x + b = 0$. In general, the hyperplane is $w^T x + b = f(x)$, where $w \in R^d$ and b is also a real number so the linear classifier is as follows [32, 33]

$$y(x) = sign[w^T x + b], \tag{3.1}$$

which is equivalent to

$$y_k[w^T x + b] \geq 1, \quad k = 1...N, \tag{3.2}$$

where $y_k \in \{-1, 1\}$ is class labels.

The above problem can be modeled as a quadratic minimization problem with linear constraints, as shown below

$$\begin{aligned} \underset{w}{\text{minimize}} \quad & \frac{1}{2} w^T w \\ \text{subject to} \quad & y_k f(x_k) \geq 1. \end{aligned} \tag{3.3}$$

Now by applying the Lagrange multiplier, the problem is as follows

$$\begin{aligned} \max_{\alpha_k} \quad \min_{w,b} \quad & \frac{1}{2} w^T w - \sum_{k=1}^{N} \alpha_k (y_k[w^T x + b] - 1), \\ \text{subject to} \quad & y_k f(x_k) \geq 1. \end{aligned} \tag{3.4}$$

This is equivalent to solving the following linear system

$$w = \sum_{k=1}^{N} \alpha_k y_k x_k, \tag{3.5}$$

$$\sum_{k=1}^{N} \alpha_k y_k = 0. \tag{3.6}$$

which the following equation will be obtained

$$y(x) = [\sum_{k=1}^{N} \alpha_k y_k x_k^T x_k + b]. \tag{3.7}$$

By obtaining the unknowns, w, b, and α_k, this optimization problem is solved, and the optimized hyperplane for data classification will be obtained.
The LS-SVM method is a set of supervised learning methods, which is the least squares of SVM methods. The least squares support vector machines for data classification were first proposed by Suykens and Vandewalle [32]. It is also used for data regression and has wide applications, including reliability engineering [7], thermal processing system [16], prediction of gas consumption [34], to name but a few.
The purpose of this method is to minimize the following statement

$$\min J_2(w, b, e) = \frac{\mu}{2} w^T w + \frac{\gamma}{2} \sum_{i=1}^{N} e_i^2, \tag{3.8}$$

subject to

$$y_i[w^T \phi(x_i) + b] = 1 - e_i, \;\; i = 1, .., N. \tag{3.9}$$

3.2 SVM-Regression

For a dataset $(x_i, y_i), i = 1, ..., N$, where the training points x_i and output points y_i are in $\mathbb{R}$, in 2D space linear regression, the goal is to predict the $y(x) = w^T x + b$ model to minimize the loss function in some sense, which is the sum of squared error, that we use the kernel trick. That is, instead of x, replace $\phi(x)$, since $\phi(x)$s are our orthogonal kernels. The choice of these kernels is optional, which we will discuss in the next section. So the predicted model will be $y(x) = w^T \phi(x) + b$.
There are several methods to solve the regression problem, which is here solved by the LS-SVR method. The primal form of LS-SVR is as follows, derived from the Lagrange multiplier [20, 32]

$$\begin{aligned} \underset{w,b,e}{\text{minimize}} \quad & \frac{1}{2} w^T w + \frac{\gamma}{2} e^T e \\ \text{subject to} \quad & y_i = w^T \phi(x_i) + b + e_i, \; i = 1, ..., N. \end{aligned} \tag{3.10}$$

where $y \in R^+$, $b \in R$ and $w \in R^h$. $\phi(.) : R \to R^h$, which is our feature map and h is the feature space dimension.
The dual form of LS-SVR is as follows [20]

$$\left[\begin{array}{c|c} \boldsymbol{\Omega} + I_N/\gamma & \mathbf{1_N} \\ \hline \mathbf{1_N}^T & 0 \end{array}\right] \left[\begin{array}{c} \boldsymbol{\alpha} \\ \hline b \end{array}\right] = \left[\begin{array}{c} \mathbf{y} \\ \hline 0 \end{array}\right] \tag{3.11}$$

which $\boldsymbol{\Omega}_{i,j} = k(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ is the ij-th element of the positive definite kernel matrix, $\boldsymbol{\alpha} = [\alpha_1, ..., \alpha_N]^T$, $\mathbf{y} = [y_1, ..., y_N]^T$, $\mathbf{1_N}$ is a column vector of 1s and I_N is an identity $N \times N$ matrix.
The above linear system gives the following equation

$$y(x) = \sum_{i=1}^{N} \alpha_i k(x, x_i) + b. \tag{3.12}$$

3.3 Kernel

Kernel functions are introduced in support vector machines as a tool to describe the nonlinear classification and regression using the linear case. These kernels can be written in terms of basic orthogonal functions such as Chebyshev, Legendre, and Laguerre polynomials. The advantages of these orthogonal kernels are their sparse matrices, so their derivation will be convenient and fast. Furthermore, due to the sparsity, our approach propels to the well-conditioned system. plus, because of the orthogonality of Chebyshev polynomials, data in feature space have the least duplication [22].
In this paper, we make use of Chebyshev kernels in presenting a least squares SVR collocation, as described in Section 5, to solve the inverse Stefan problem.
Kernel functions are defined as [22]

$$k(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle, \tag{3.13}$$

where $\phi(.)$ is transformation function.
Different kernels have been introduced and used to solve mathematical and physics problems in literature. The most common kernels are expressed as follows [22, 14]

$$k(\mathbf{x}, \mathbf{z}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{z}\|^2}{2\sigma^2}\right), \tag{3.14}$$

which is *Gaussian kernel* function and σ is kernel parameter. It is also known as RBF kernel, which has also used in the numerical solution of differential equations [20, 21].

$$k(\mathbf{x}, \mathbf{z}) = \left(\frac{\langle \mathbf{x}, \mathbf{z} \rangle + 1}{\beta}\right)^n, \tag{3.15}$$

which is *Polynomial kernel* function, n and β are kernel parameter, and the scaling parameter, respectively. Also, the *Wavelet kernel* function is as follows

$$k(\mathbf{x}, \mathbf{z}) = \prod_{j=1}^{N} \left(\cos\left(1.75\frac{x_j - z_j}{a}\right) \exp\left(-\frac{\|x_j - z_j\|^2}{2a^2}\right)\right). \tag{3.16}$$

In this paper, we employed the orthogonal Chebyshev kernel in our method; therefore, we will describe Chebyshev kernels here. Ye *et al.* proposed the Chebyshev kernel for inputs (x, z) as follows [22, 23]

$$k(x, z) = \frac{\sum_{i=0}^{n} T_i(x) T_i(z)}{\sqrt{1 - xz}}, \tag{3.17}$$

where $T_i(x)$ is the i-th degree Chebyshev polynomial of the first kind, and $\sqrt{1 - xz}$ is the weight function $w(x, z)$, furthermore by applying the equation above for dataset $(x_j, z_j), j = 1, ..., M$ including vector $\mathbf{x}$ and $\mathbf{z}$, it turns into

$$k(\mathbf{x}, \mathbf{z}) = \prod_{j=1}^{M} \frac{\sum_{i=0}^{n} T_i(x_j) T_i(z_j)}{\sqrt{1 - x_j z_j}}. \tag{3.18}$$

Ozer and Chen afterward proposed the generalized Chebyshev kernels, which are defined as follows [22, 23]

$$k(\mathbf{x}, \mathbf{z}) = \frac{\sum_{i=0}^{n} T_i(\mathbf{x}) T_i^T(\mathbf{z})}{\sqrt{d - \langle \mathbf{x}, \mathbf{z} \rangle}}, \quad k(\mathbf{x}, \mathbf{z}) = \frac{\sum_{i=0}^{n} T_i(\mathbf{x}) T_i^T(\mathbf{z})}{\exp\left(-\gamma \|\mathbf{x} - \mathbf{z}\|^2\right)}. \tag{3.19}$$

There are also orthogonal Chebyshev kernels as the first-order, the second-order, and the like. See [35] for proof of the Chebyshev polynomial kernels construction.

4 Chebyshev polynomials

Chebyshev polynomials are considered as one of the specific cases in Jacobi polynomials $\boldsymbol{J}_n^{\alpha,\beta}$ which are defined as the eigenfunctions of the Sturm-Liouville equation [30]

$$-(1-x)^{-\alpha}(1+x)^{-\beta}\frac{d}{dx}((1-x)^{\alpha+1}(1+x)^{\beta+1}\frac{d}{dx}u(x)) = (x^2-1)\frac{d^2}{dx^2}u(x) + (\alpha - \beta + (\alpha+\beta+2)x)\frac{d}{dx}u(x), \tag{4.1}$$

which $\alpha, \beta > -1$. So the Chebyshev polynomials of the first kind $\{\boldsymbol{J}_n^{\frac{-1}{2},\frac{-1}{2}}\}$ are eigenfunctions of the Sturm-Liouville equation

$$\sqrt{1-x^2}\frac{d}{dx}(\sqrt{1-x^2}\frac{d}{dx}T_n(x)) + \lambda_n T_n(x) = 0, \tag{4.2}$$

where $\lambda_n = n^2$. The Equation (4.2) is equivalent to the below equation

$$(1-x^2)\frac{d^2}{dx^2}T_n(x) - x\frac{d}{dx}T_n(x) + n^2 T_n(x) = 0. \tag{4.3}$$

Chebyshev polynomials of the first kind of degree n for x values, which are in the interval [-1,1], are defined as follows [22, 30]

$$T_n(x) = \cos(n\theta) = \cos(n \arccos x), \quad x = \cos\theta, \quad n \geq 0, \tag{4.4}$$

which is the explicit form of the equation. However, it is computationally expensive. Therefore, for Chebyshev polynomials, the following recursive equation holds

$$T_0(x) = 1, \tag{4.5}$$
$$T_1(x) = x, \tag{4.6}$$
$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), \quad n \geq 1. \tag{4.7}$$

The orthogonality equation of the first kind Chebyshev polynomials is as follows

$$\int_{-1}^{1} T_n(x)T_m(x)w(x)dx = \frac{c_n\pi}{2}\delta_{mn}, \tag{4.8}$$

where $w(x) = \frac{1}{\sqrt{1-x^2}}$ is matched to the weight function, δ_{mn} is the Kronecker delta function and $n \geq 1, c_n = 1, c_0 = 2$. So the above equation under some cases on m and n turns into

$$\int_{-1}^{1} T_n(x)T_m(x)\frac{1}{\sqrt{1-x^2}}dx = \begin{cases} 0 & n \neq m \\ \frac{\pi}{2} & n = m \neq 0 \\ \pi & n = m = 0 \end{cases}. \tag{4.9}$$

The first derivative of the Chebyshev polynomials of the first kind is also orthogonal in regard to the weight function $\sqrt{1-x^2}$.
Chebyshev polynomials also are written by Rodrigues' formula in the following form [4]

$$T_n(x) = \frac{(-1)^n 2^n n!}{(2n)!}(1-x^2)^{\frac{1}{2}}\frac{d^n}{dx^n}(1-x^2)^{n-\frac{1}{2}}. \tag{4.10}$$

Due to every orthogonal polynomial has n real roots within the interval (a, b), Chebyshev polynomials also have n real roots in the range $(-1, 1)$.
Chebyshev polynomials of the first kind of degree $n \geq 1$ within the interval $[-1, 1]$ have n zero according to the following term

$$x_k = \cos\left(\frac{(2k-1)\pi}{2n}\right), \quad k = 1, 2, ..., n. \tag{4.11}$$

5 Collocation least squares support vector regression for inverse Stefan problem

In this section, we present an LS-SVR method for solving the inverse Stefan problem with the primal and dual form.

5.1 Primal form

By considering the one dimensional and one-phase heat equation, the problem's physical domain is as follows

$$\Gamma = \{(x,t) \in \mathbb{R} |\ 0 < x < s(t), 0 < t < T\},\ \ T > 0, \tag{5.1}$$

where T is a desired final positive time, and $s(t)$ represents a smooth function for the boundary $[0,T]$. The initial and boundary conditions and the properties of the heat-conducting body have been given [9]. In this paper, we investigate the inverse one-phase Stephen problem. So we considered the one-phase heat equation below [15]

$$\frac{\partial u(x,t)}{\partial t} = \frac{\partial^2 u(x,t)}{\partial x^2}, \quad (x,t) \in \Gamma, \tag{5.2}$$

the boundary conditions are given as follows

$$u(s(t),t) = A(t), \quad t \in [0,T], \tag{5.3}$$

$$\frac{\partial u(x,t)}{\partial x}|_{x=s(t)} = B(t), \quad t \in [0,T], \tag{5.4}$$

$$u(0,t) = h_1(t), \quad t \in [0,T], \tag{5.5}$$

$$\frac{\partial u(x,t)}{\partial x}|_{x=0} = h_2(t), \quad t \in [0,T], \tag{5.6}$$

$$\int_{[0,s(t)]} u(x,t)dx = h_3(t), \tag{5.7}$$

along with the initial condition

$$u(x,0) = f(x), \quad x \in [0,s(0)]. \tag{5.8}$$

Equations from (5.2) to (5.6) with the initial condition and also equations from (5.2) to (5.5) along with Equation (5.7) and the initial state are known as the inverse Stefan problem.
Consider the main Equation (5.2). We apply the Landau's transformation in order to make the boundary condition $s(t)$ as a constant function [28]

$$y = \frac{x}{s(t)}, \quad y \in [0,1], \tag{5.9}$$

therefore we will have

$$u(x,t) = u(s(t).y,t). \tag{5.10}$$

Now the above equation is a function in terms of y

$$u(x,t) = V(y,t), \quad y \in [0,1]. \tag{5.11}$$

So by applying the Landau's transformation, the Equation (5.2) turns into

$$\frac{\partial^2 V(y,t)}{\partial y^2} - s(t)^2 \frac{\partial V(y,t)}{\partial t} + ys(t)s'(t)\frac{\partial V(y,t)}{\partial y} = 0, \quad (y,t) \in \Gamma' = (0,1) \times (0,T), \tag{5.12}$$

subject to the following boundary conditions

$$\frac{\partial V(y,t)}{\partial y}|_{y=1} = B(t)s(t), \quad t \in [0,T], \tag{5.13}$$

$$V(1,t) = A(t), \quad t \in [0,T], \tag{5.14}$$

and the initial condition

$$V(y,0) = f(ys(0)), \quad y \in [0,1]. \tag{5.15}$$

Now, first, we employ the Crank-Nicolson method to this problem, to get

$$\frac{V^{n+1}-V^n}{\Delta t} = (1-\theta)(\frac{1}{s^2(t_{n+1})}\, V_{yy}^{n+1} + \frac{s'(t_{n+1})}{s(t_{n+1})}\, y\, V_y^{n+1}) + \theta(\frac{1}{s^2(t_n)}\, V_{yy}^{n} + \frac{s'(t_n)}{s(t_n)}\, y\, V_y^{n+1}), \tag{5.16}$$

where θ is chosen in [0,1]. Now we explain the method of LS-SVR for this equation. The solution of the problem assumed in the infinite-dimensional continuous functions $C(0,1)$, is approximated by the finite-dimensional space of polynomials as follows

$$V_N^{n+1}(x) = \sum_{j=0}^{N} w_j \phi_j(x), \tag{5.17}$$

which the ϕ_j is the basic orthogonal function that we considered Chebyshev. For convenience, we use u_N instead of $V_N^{n+1}(x)$.
The primal form of the least squares support vector regression is as follows

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + \frac{\gamma}{2}\boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon} \\ \text{subject to} \quad & Lu_N(x_i) - f_i = \varepsilon_i,\ i = 1,\dots,N, \\ & u_N(1) = a, \\ & u'_N(1) = b, \end{aligned} \tag{5.18}$$

where $\mathbf{w} = [w_0, ..., w_N]^T, \boldsymbol{\varepsilon} = [\varepsilon_1, ..., \varepsilon_N]^T$ and the operator L is defined as follows

$$L = (\frac{D^0}{\Delta t} - \frac{D^2}{s^2(t_{n+1})} - \frac{s'(t_{n+1})}{s(t_n+1)}\, y\, D'). \tag{5.19}$$

The optimization Problem (5.18) may be solved by numerical optimization techniques. However, to get a computationally efficient method, we provide the dual form as a linear system of equation by using the Lagrange multiplier theorem.

5.2 Dual form

To solve the dual problem, we set

$$\mathcal{L} = \frac{1}{2}w^T w + \frac{\gamma}{2}\varepsilon^T\varepsilon - \sum_{i=0}^{N}\alpha_i(Lu_N(x_i) - f_i - \varepsilon_i) - \beta(u_N(1)-a) - \eta(u'_N(1)-b), \tag{5.20}$$

which is written as

$$\mathcal{L} = \frac{1}{2}\sum_{j=0}^{N} w_j^2 + \frac{\gamma}{2}\sum_{j=0}^{N}\varepsilon_j^2 - \sum_{i=0}^{N}\alpha_i(\sum_{j=0}^{N} w_j L\phi_j(x_i) - f_i - \varepsilon_i) - \beta(\sum_{j=0}^{N} w_j\phi_j(1) - a) - \eta(\sum_{j=0}^{N} w_j\phi'_j(1) - b). \tag{5.21}$$

The unknown variables in the formula above are: $(w_0, w_1, ..., w_N)$, $(\varepsilon_0, \varepsilon_1, ..., \varepsilon_N)$, $(\alpha_0, \alpha_1, ..., \alpha_N)$, α, β.
We differentiate $\mathcal{L}$ according to each of the variables then set the equations to zero. The following expressions are

obtained, respectively

$$\frac{\partial \mathcal{L}}{\partial w_k} = 0 \implies w_k - \sum_{i=0}^{N} \alpha_i (L\phi_k(x_i)) - \beta\phi_k(1) - \eta\phi'_k(1) = 0, \tag{5.22}$$

$$\frac{\partial \mathcal{L}}{\partial \alpha_k} = 0 \implies \sum_{j=0}^{N} w_j L\phi_j(x_k) - f_k - \varepsilon_k = 0, \tag{5.23}$$

$$\frac{\partial \mathcal{L}}{\partial \varepsilon_k} = 0 \implies \gamma\varepsilon_k + \alpha_k = 0, \tag{5.24}$$

$$\frac{\partial \mathcal{L}}{\partial \beta} = 0 \implies \sum_{j=0}^{N} w_j \phi_j(1) - a = 0, \tag{5.25}$$

$$\frac{\partial \mathcal{L}}{\partial \eta} = 0 \implies \sum_{j=0}^{N} w_j \phi'_j(1) - b = 0, \tag{5.26}$$

For $k = 0, 1, ..., N$.
So then these equations form the system of equations, that we solve with matrix form and we have the matrix A as follows

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} & A_{15} \\ A_{21} & A_{22} & A_{23} & A_{24} & A_{25} \\ A_{31} & A_{32} & A_{33} & A_{34} & A_{35} \\ A_{41} & A_{42} & A_{43} & A_{44} & A_{45} \\ A_{51} & A_{52} & A_{53} & A_{54} & A_{55} \end{bmatrix}.$$

Each A_{ij} is a matrix block that we describe below

$$A_{11} = I,$$

which matrix I is the $N+1 \times N+1$ identity matrix.

$$A_{12} = \begin{bmatrix} -L\phi_0(x_0) & \dots & -L\phi_0(x_N) \\ \vdots & \ddots & \vdots \\ -L\phi_N(x_0) & \dots & -L\phi_N(x_N) \end{bmatrix},$$

which A_{12} is a $N+1 \times N+1$ matrix.

$$A_{13} = O,$$

which O is a $N+1 \times N+1$ zero matrix.

$$A_{14} = \begin{bmatrix} -\phi_0(1) \\ \vdots \\ -\phi_N(1) \end{bmatrix},$$

$$A_{15} = \begin{bmatrix} -\phi'_0(1) \\ \vdots \\ -\phi'_N(1) \end{bmatrix},$$

which A_{14} and A_{15} are $N+1 \times 1$ vectors.

$$A_{21} = -A_{12}^T,$$

the matrix A_{21} is the negative transpose of the matrix A_{12}.

$$A_{22} = O, \quad A_{23} = -I, \quad A_{31} = O, \quad A_{32} = I,$$

A_{24} and A_{25} are $N+1 \times 1$ zero column vectors. The matrix A_{33} is

$$A_{33} = \gamma I.$$

The A_{34} and A_{35} are $N+1 \times 1$ zero column vectors.

$$A_{41} = -A_{14}^T, \quad A_{51} = -A_{15}^T,$$

A_{42}, A_{43}, A_{52}, and A_{53} are $N+1 \times 1$ zero row vectors. And A_{44}, A_{45}, A_{54}, and A_{55} are zero elements. Now the equation system is as follows

$$A \times \boldsymbol{x} = \boldsymbol{f},$$

where $\boldsymbol{x} = [w_0, ..., w_N, \alpha_0, ..., \alpha_N, \varepsilon_0, ..., \varepsilon_N, \beta, \eta]^T$ and $\boldsymbol{f} = [0, ..., 0, f_0, ..., f_N, 0, ..., 0, a, b]^T$.
The matrix A and the vector $\boldsymbol{f}$ are known, and the vector $\boldsymbol{x}$ is unknown. We can obtain the unknown vector by linear solvers.

6 Numerical Results

In this section, the numerical results are illustrated by three examples, to evaluate the effectiveness and efficiency of the proposed method.

In the following tables, the first column is stating N=M, which respectively means the number of basis functions and the number of collocation points. The method's run time in the second column vs. the number of training points is also illustrated in the tables to show the linear complexity. Besides, the third to sixth columns present the error of our approach. The term T inside these tables indicates the time steps that we calculate the maximum error on them. Additionally, the L parameter determines the number of our spatial steps.
In all of the following tables, the training points are as the Chebyshev roots and our constructed roots which is as follows

$$P_i = 1 - \left(\frac{i}{2N}\right)^2, \ i = 1, .., N+1. \tag{6.1}$$

Another point to consider is our regularization term γ, which is set to between the number 10 and 10^{18}, according to the problem behavior and maple constraints. We expect the Crank-Nicolson method to converge with the increasing number of basis functions of the problem, which means the errors will reduce. As illustrated in the tables below, by setting the value of L into 10 and 20, with the increasing number of M, the errors will be decreased to some degrees. So our method has rational convergence behavior.

Example 1. As the first example [15], we consider Equation (5.2) with boundary conditions $A(t) = 0, B(t) = \frac{-1}{\sqrt{2}}$ and moving boundary function $s(t) = \sqrt{2} - 1 + \frac{t}{\sqrt{2}}$; $0 \leqslant t \leqslant 1$. The exact solution of this problem is $u = -1 + \exp(1 - \frac{1}{\sqrt{2}} + \frac{t}{2} - \frac{x}{\sqrt{2}})$, with $(x, t) \in [0, s(t)] \times [0, 1]$.

		$L = 10, 20$		$L = 10, 20$	
		Constructed roots		Chebyshev roots	
M	time	Error	Error	Error	Error
2	0.031	9.31e-02	1.89e-01	1.27e+00	2.22e+00
4	0.047	1.52e-03	1.52e-03	6.12e-03	1.00e-02
6	0.062	1.44e-05	5.60e-05	4.70e-04	9.67e-01
8	0.125	1.08e-04	4.26e-04	1.81e-04	3.08e-01

Table 1: The numerical error of the method in $T = 1$.

		$L = 10, 20$		$L = 10, 20$	
		Constructed roots		Chebyshev roots	
M	time	Error	Error	Error	Error
2	0.032	3.44e-02	5.46e-02	5.65e-01	1.17e+00
4	0.047	5.19e-04	6.28e-04	1.03e-03	3.50e-03
6	0.062	2.70e-05	2.94e-05	8.18e-06	3.37e-02
8	0.109	3.88e-05	9.32e-05	4.97e-05	2.53e-03

Table 2: The numerical error of the method in $T = 0.75$.

		$L = 10, 20$		$L = 10, 20$	
		Constructed roots		Chebyshev roots	
M	time	Error	Error	Error	Error
2	0.031	1.90e-02	3.19e-02	2.60e-01	4.63e-01
4	0.063	2.18e-04	2.13e-04	1.95e-04	9.32e-04
6	0.094	1.36e-05	3.51e-06	1.65e-05	2.98e-05
8	0.172	1.72e-05	1.57e-05	1.90e-05	2.48e-05

Table 3: The numerical error of the method in $T = 0.5$.

		$L = 10, 20$		$L = 10, 20$	
		Constructed roots		Chebyshev roots	
M	time	Error	Error	Error	Error
2	0.031	6.80e-03	7.90e-03	3.78e-02	1.01e-01
4	0.047	4.07e-05	5.65e-05	1.25e-05	9.20e-05
6	0.094	1.19e-07	2.22e-06	7.78e-06	2.30e-06
8	0.156	3.32e-03	3.18e-06	8.54e-07	3.81e-06

Table 4: The numerical error of the method in $T = 0.25$.

The numerical results are reported in tables 1- 4. From these tables, the convergence of the method is seen by increasing the number of training points. It is seen that the convergence behavior is greatly affected by the selection of training points. For instance, the training points with constructed roots gives better results for $L = 20$ in comparison with the Chebyshev root. Here meta-heuristic algorithm may be used for the selection of training points.

Example 2. As the second example [15], we consider Equation (5.2) with boundary conditions $A(t) = 0, B(t) = \sqrt{3 - 2t}$, $0 < t \leqslant 1$ and moving boundary function $s(t) = 2 - \sqrt{3 - 2t}$, $0 \leqslant t \leqslant 1$. The exact solution of this problem is $u = \frac{-x^2}{2} + 2x - \frac{1}{2} - t$, with $(x, t) \in [0, s(t)] \times [0, 1]$.

Numerical results in this example are illustrated in the following tables 5- 8. It can be seen that by increasing the training points, the errors are stable somehow with both constructed and Chebyshev roots. However, with the Chebyshev roots, as we can see in table 8 for $L = 10$, the errors have decreased. Also, for some time steps, training points with the Chebyshev roots are useful to apply, and for the others, the constructed roots would be a better one.

		$L = 10, 20$		$L = 10, 20$	
		Constructed roots		Chebyshev roots	
M	time	Error	Error	Error	Error
2	0.015	4.39e-05	3.86e-05	5.16e-04	2.35e-04
4	0.016	3.30e-04	4.67e-04	5.46e-04	2.30e-03
6	0.063	2.41e-04	3.50e-03	2.51e-04	7.54e-01
8	0.094	8.80e-04	1.25e-02	1.54e-03	1.87e-01

Table 5: The numerical error of the method in $T = 1$.

Example 3. As the final example, we consider Equation (5.2) with boundary conditions $A(t) = 0, B(t) = -\frac{1}{2}$, $0 < t \leqslant 1$ and moving boundary function $s(t) = 2 + \frac{t}{2} - \sqrt{3}$, $0 \leqslant t \leqslant 1$. The exact solution of this problem is

		$L = 10, 20$		$L = 10, 20$	
		Constructed roots		Chebyshev roots	
M	time	Error	Error	Error	Error
2	0.015	2.11e-05	1.47e-04	9.19e-04	6.19e-05
4	0.031	4.20e-05	1.54e-04	3.96e-04	1.74e-04
6	0.062	1.13e-04	5.44e-04	3.40e-04	1.94e-03
8	0.109	2.17e-04	1.23e-03	1.91e-04	1.80e-03

Table 6: The numerical error of the method in $T = 0.75$.

		$L = 10, 20$		$L = 10, 20$	
		Constructed roots		Chebyshev roots	
M	time	Error	Error	Error	Error
2	0.015	3.07e-05	1.05e-05	3.06e-04	8.71e-05
4	0.031	5.79e-05	1.67e-05	1.01e-04	4.17e-05
6	0.047	7.39e-05	5.90e-05	5.81e-05	4.62e-05
8	0.109	7.87e-05	9.87e-05	2.53e-05	9.03e-05

Table 7: The numerical error of the method in $T = 0.5$.

		$L = 10, 20$		$L = 10, 20$	
		Constructed roots		Chebyshev roots	
M	time	Error	Error	Error	Error
2	0.015	7.67e-07	3.13e-06	6.34e-05	9.07e-05
4	0.031	5.00e-07	6.23e-06	4.46e-06	3.13e-05
6	0.062	7.36e-07	9.13e-06	8.69e-07	3.41e-05
8	0.094	7.96e-07	1.03e-05	7.96e-07	2.36e-05

Table 8: The numerical error of the method in $T = 0.25$.

$u = \exp(-x)$, with $(x, t) \in [0, s(t)] \times [0, 1]$.

Numerical results for the final sample are shown in the following tables 9- 12. This approach channel the problem's errors into a good convergence, as seen from the tables below. For example, in all tables, the Chebyshev training roots give us better results for $L = 10$ on the contrary of the constructed roots. And vice versa for $L = 20$.

		$L = 10, 20$		$L = 10, 20$	
		Constructed roots		Chebyshev roots	
M	time	Error	Error	Error	Error
2	0.031	2.91e-02	9.26e-02	1.11e+00	1.16e+01
4	0.047	8.82e-04	1.30e-03	8.34e-04	6.67e-03
6	0.047	6.12e-04	3.13e-03	3.22e-04	3.35e-01
8	0.094	7.84e-04	7.95e-03	4.23e-04	6.01e-03

Table 9: The numerical error of the method in $T = 1$.

		$L = 10, 20$		$L = 10, 20$	
		Constructed roots		Chebyshev roots	
M	time	Error	Error	Error	Error
2	0.031	1.94e-02	1.19e-02	2.80e-01	3.51e+00
4	0.032	4.40e-04	9.66e-05	1.96e-04	4.17e-03
6	0.063	2.08e-04	5.68e-04	9.59e-05	4.17e-03
8	0.125	2.34e-04	1.05e-03	4.13e-05	2.70e-03

Table 10: The numerical error of the method in $T = 0.75$.

		$L = 10, 20$		$L = 10, 20$	
		Constructed roots		Chebyshev roots	
M	time	Error	Error	Error	Error
2	0.047	9.20e-03	1.07e-02	8.13e-02	6.71e-01
4	0.047	6.58e-05	1.36e-04	8.51e-05	1.07e-04
6	0.063	4.00e-05	8.15e-05	3.89e-05	2.42e-04
8	0.094	4.10e-05	1.08e-04	1.98e-05	2.15e-04

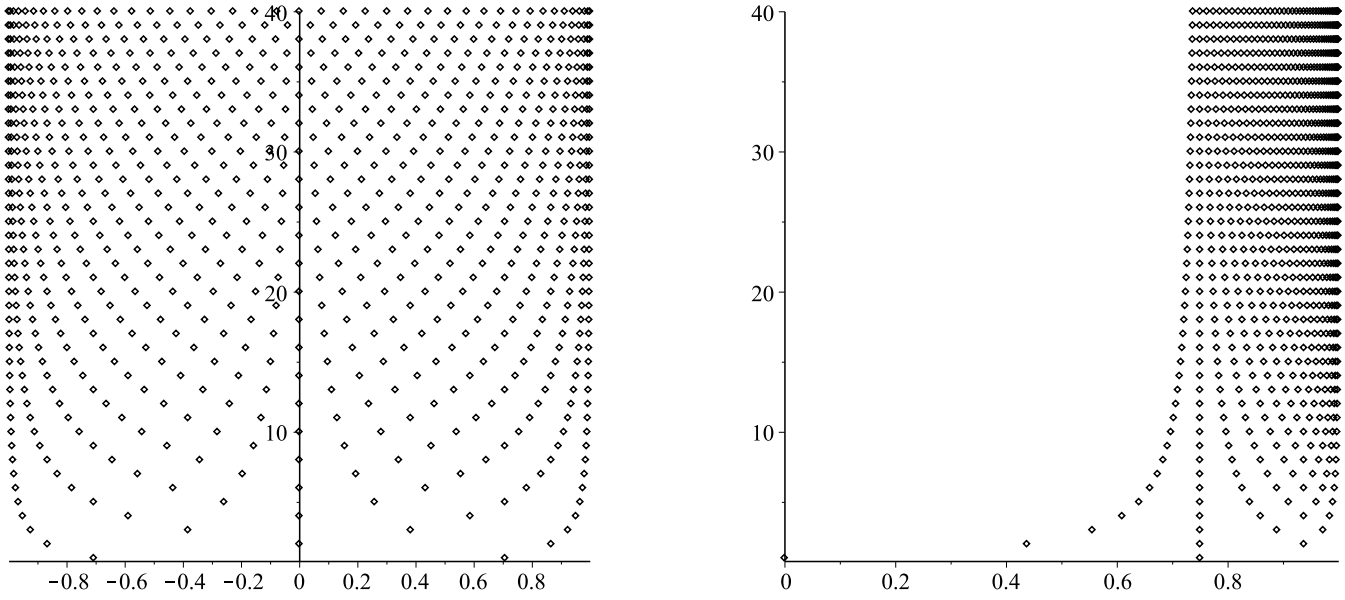
Table 11: The numerical error of the method in $T = 0.5$.

		$L = 10, 20$		$L = 10, 20$	
		Constructed roots		Chebyshev roots	
M	time	Error	Error	Error	Error
2	0.031	3.64e-03	4.30e-03	6.86e-03	6.60e-02
4	0.031	2.33e-05	2.09e-05	6.04e-06	4.42e-05
6	0.062	2.96e-06	8.54e-06	2.87e-06	4.08e-05
8	0.109	2.87e-06	8.93e-06	2.87e-06	2.92e-05

Table 12: The numerical error of the method in $T = 0.25$.

In the final analysis, we concluded that each example, based on the property of the problem we are solving, has its condition of selecting the collocation points.

Figure 1 illustrates the distribution of the training points over the domain of the inverse Stefan problem in one dimensional and one-phase case. Note that the training points are accumulated near the boundary with known conditions to ensure the accuracy of the predicted values.



(a) Chebyshev roots

(b) Constructed roots

Figure 1: Nodal distribution for training roots.

Conclusion

In this paper, we have proposed a new numerical scheme for the simulation of the Inverse Stefan problem. The method uses a hybrid of Crank-Nicolson and least squares support vector regression. The problem is ill-posed, and due to the kernel trick, the solution leads into well-posed circumstances, which we use Chebyshev polynomial kernels.

Also, the problem is first discretized in time, and then in each time step, we have an ODE equation that is solved by the proposed scheme. We solved the problem in primal and dual form. As we show in numerical results, this approach has reasonable convergence behavior.

References

[1] J. Adler and O. Öktem, *Solving ill-posed inverse problems using iterative deep neural networks*, Inverse Probl. **33** (2017), no. 12, 124007.

[2] L. Bar and N. Sochen, *Unsupervised deep learning algorithm for pde-based forward and inverse problems*, 2019.

[3] J. Berg and K. Nyström, *Neural network augmented inverse problems for pdes*, arXiv, 2018.

[4] J.P. Boyd, *Chebyshev and Fourier spectral methods*, Courier Corporation, 2001.

[5] P. Dadvand, R. Lopez, and E. Onate, *Artificial neural networks for the solution of inverse problems*, Proc. Int. Conf. Design Optim. Meth. Appl. ERCOFTAC, vol. 2006, 2006.

[6] A. Hajiollow, Y. Lotfi, K. Parand, A.H. Hadian, K. Rashedi, and J.A. Rad, *Recovering a moving boundary from Cauchy data in an inverse problem which arises in modeling brain tumor treatment: the (quasi) linearization idea combined with radial basis functions (RBFs) approximation*, Engin. Comput. (2020), 1–15.

[7] J. Ji, C. Zhang, Y. Gui, Q. Lü, and J. Kodikara, *New observations on the application of LS-SVM in slope system reliability analysis*, J. Comput. Civil Engin. **31** (2017), no. 2, 06016002.

[8] M.T. Jin, K.H.and McCann, E. Froustey, and M. Unser, *Deep convolutional neural network for inverse problems in imaging*, IEEE Trans. Image Process. **26** (2017), no. 9, 4509–4522.

[9] B.T. Johansson, D. Lesnic, and T. Reeve, *A method of fundamental solutions for the one-dimensional inverse Stefan problem*, Appl. Math. Modell. **35** (2011), no. 9, 4367–4378.

[10] A. Karami, S. Abbasbandy, and E. Shivanian, *Meshless Local Petrov–Galerkin Formulation of Inverse Stefan Problem via Moving Least Squares Approximation*, Math. Comput. Appl. **24** (2019), no. 4, 101.

[11] Y. Khoo and L. Ying, *Switchnet: a neural network model for forward and inverse scattering problems*, SIAM J. Sci. Comput. **41** (2019), no. 5, A3182–A3201.

[12] A. Kirsch, *An introduction to the mathematical theory of inverse problems*, vol. 120, Springer Science & Business Media, 2011.

[13] A. Kirsch and A. Rieder, *Inverse problems for abstract evolution equations with applications in electrodynamics and elasticity*, Inverse Probl. **32** (2016), no. 8, 085001.

[14] L. Liu, W. Huang, and C. Wang, *Hyperspectral image classification with kernel-based least-squares support vector machines in sum space*, IEEE J. Select. Topics Appl. Earth Observ. Remote Sens. **11** (2017), no. 4, 1144–1157.

[15] Y. Lotfi, K. Parand, K. Rashedi, and J. Amani Rad, *Numerical study of temperature distribution in an inverse moving boundary problem using a meshless method*, Engin. Comput. (2019), 1–15.

[16] X. Lu, W. Zou, and M. Huang, *A novel spatiotemporal LS-SVM method for complex distributed parameter systems with applications to curing thermal process*, IEEE Trans. Ind. Inf. **12** (2016), no. 3, 1156–1165.

[17] A. Lucas, M. Iliadis, R. Molina, and A.K. Katsaggelos, *Using deep neural networks for inverse problems in imaging: beyond analytical methods*, IEEE Signal Process. Magazine **35** (2018), no. 1, 20–36.

[18] D. Lv, Q. Zhou, J.K. Choi, J. Li, and X. Zhang, *Nonlocal TV-Gaussian prior for Bayesian inverse problems with applications to limited CT reconstruction*, Inverse Probl. Imaging **14** (2020), no. 1, 117.

[19] J. Matlak, D. Słota, and A. Zielonka, *Reconstruction of the heat transfer coefficient in the inverse Stefan problem*, Hutnik, Wiadomości Hutnicze **85** (2018), no. 1, 6–9.

[20] S. Mehrkanoon, T. Falck, and J.A. Suykens, *Approximate solutions to ordinary differential equations using least squares support vector machines*, IEEE Trans. Neural Networks Learn. Syst. **23** (2012), no. 9, 1356–1367.

[21] S. Mehrkanoon and J.A.K. Suykens, *Learning solutions to partial differential equations using LS-SVM*, Neurocomput. **159** (2015), 105–116.

[22] S. Ozer, C.H. Chen, and H.A. Cirpan, *A set of new chebyshev kernel functions for support vector machine pattern classification*, Pattern Recogn. **44** (2011), no. 7, 1435–1447.

[23] L.C. Padierna, M. Carpio, A. Rojas-Domínguez, H. Puga, and H. Fraire, *A novel formulation of orthogonal polynomial kernel functions for SVM classifiers: the Gegenbauer family*, Pattern Recog. **84** (2018), 211–225.

[24] M. Parhizi and A. Jain, *Solution of the Phase Change Stefan Problem With Time-Dependent Heat Flux Using Perturbation Method*, J. Heat Transfer **141** (2019), no. 2, 024503.

[25] L. Pilozzi, F.A. Farrelly, G. Marcucci, and C. Conti, *Machine learning inverse problem for topological photonics*, Commun. Phys. **1** (2018), no. 1, 1–7.

[26] J.A. Rad, K. Rashedi, K. Parand, and H. Adibi, *The meshfree strong form methods for solving one dimensional inverse Cauchy-Stefan problem*, Engin. Comput. **33** (2017), no. 3, 547–571.

[27] G.M.M. Reddy, M. Vynnycky, and J.A. Cuminato, *An efficient adaptive boundary algorithm to reconstruct Neumann boundary data in the MFS for the inverse Stefan problem*, J. Comput. Appl. Math. **349** (2019), 21–40.

[28] S. Sarabadan, K. Rashedi, and H. Adibi, *Boundary determination of the inverse heat conduction problem in one and two dimensions via the collocation method based on the satisfier functions*, Iran. J. Sci. Technol. Trans. A: Sci. **42** (2018), no. 2, 827–840.

[29] M.M. Sarsengeldin, S.N.N. Kharin, S. Kassabek, and Z. Mukambetkazin, *Exact Solution of the One Phase Inverse Stefan Problem*, Filomat **32** (2018), no. 3, 985–990.

[30] J. Shen, T. Tang, and L.-L. Wang, *Spectral methods: algorithms, analysis and applications*, vol. 41, Springer Science & Business Media, 2011.

[31] C.B. Smith and E.M. Hernandez, *Non-negative constrained inverse eigenvalue problems–Application to damage identification*, Mech. Syst. Signal Process. **129** (2019), 629–644.

[32] J.A. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J.P. Vandewalle, *Least squares support vector machines*, World Scientific, 2002.

[33] J.A.K. Suykens and J. Vandewalle, *Least squares support vector machine classifiers*, Neural Process. Lett. **9** (1999), no. 3, 293–300.

[34] Y.-H. Wu and H. Shen, *Grey-related least squares support vector machine optimization model and its application in predicting natural gas consumption demand*, J. Comput. Appl. Math. **338** (2018), 212–220.

[35] N. Ye, R. Sun, Y. Liu, and L. Cao, *Support vector machine with orthogonal chebyshev kernel*, 18th Int. Conf. Pattern Recog. (ICPR'06), vol. 2, IEEE, 2006, pp. 752–755.

[36] N. Zabaras and K. Yuan, *Dynamic programming approach to the inverse Stefan design problem*, Numerical Heat Transfer **26** (1994), no. 1, 97–104.