LS-SVM based solutions to differential equations

Siamak Mehrkanoon

: siamak.mehrkanoon@maastrichtuniversity.nl

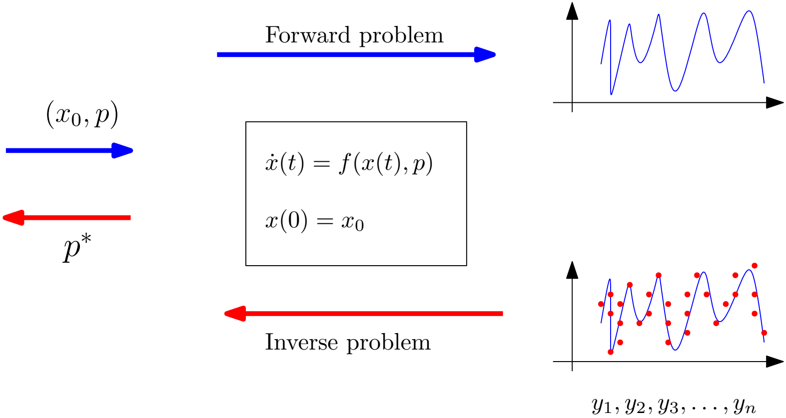
https://sites.google.com/view/siamak-mehrkanoon/homepage

Maastricht University, DKE

SMAI 2019, France

Outline

Problem Statements:

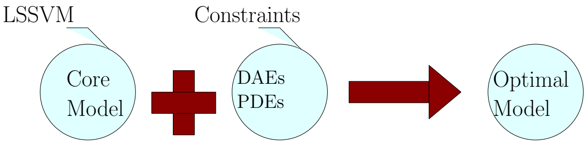


Dynamical Systems

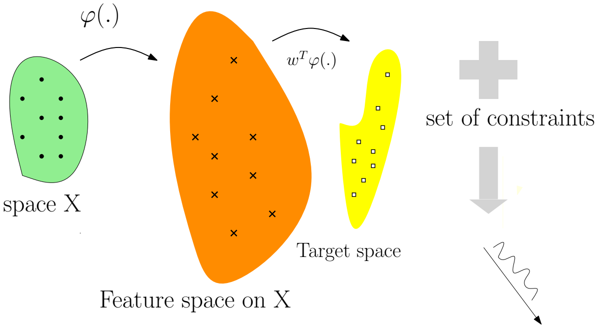
- Arise frequently in numerous applications including mathematical modeling and control theory.
- Numerical methods must be applied.

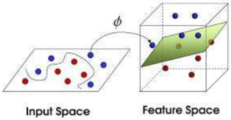
Existing numerical approaches

- Provide discrete solutions (Runge Kutta, Explicit-Implicit schemes, FDM among others).
- Require a discretization of the domain via meshing (higher dimension can potentially be a problem)
- Depend on index reduction techniques for lowering the index of a DAE system.
- Neural networks based approaches suffer from local minima solutions.



- Closed form solution
- Optimal representation of the solution
- Potentially can be used for high dimensional PDEs
- Does not require index reduction technique (high index DAEs)





- RKHS
- Gaussian process (probabilistic setting)
- LSSVM (optimization setting)

The primal LS-SVM: [1]

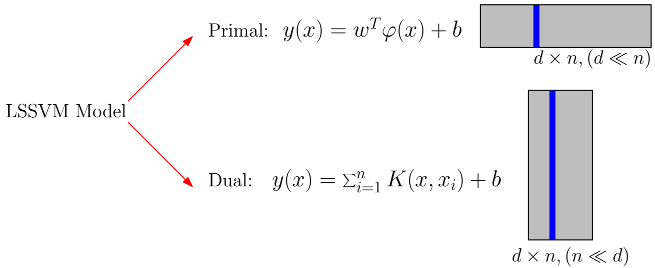
$$\underset{w,b,e}{\text{minimize}} \quad \frac{1}{2}w^T w + \frac{\gamma}{2}e^T e$$
$$\text{subject to} \quad y_i = w^T\varphi(x_i) + b + e_i, \quad i = 1, ..., n$$

The dual LS-SVM:

$$\left[\begin{array}{c|c} \Omega + I_n/\gamma & 1_n \\ \hline 1_n^T & 0 \end{array}\right] \left[\begin{array}{c} \alpha \\ \hline b \end{array}\right] = \left[\begin{array}{c} y \\ \hline 0 \end{array}\right]$$

where $\Omega_{ij} = K(x_i, x_j) = \varphi(x_i)^T\varphi(x_j)$.

[1] J. A. K. Suykens et al. *Least Squares Support Vector Machines*. World Scientific, Singapore, 2002.



- Fixed Size LSSVM [see[2]]
- Fixed Size semi-supervised KSC based model [see[3]]

[2] J. A. K. Suykens et al. *Least Squares Support Vector Machines*. World Scientific, Singapore, 2002.

[3] Siamak Mehrkanoon and Johan AK Suykens. "Large scale semi-supervised learning using KSC based model". In: *IEEE International Joint Conference on Neural Networks (IJCNN)*. 2014.

Forward Problem: PDEs

Aim

We propose a kernel based method in the LS-SVM framework [4]. The formulation is derived using the primal-dual setting.

- In primal: the solution is in terms of the feature map.
- In dual: Kernel based representation of the solution.

[4] Siamak Mehrkanoon and Johan AK Suykens. "Learning solutions to partial differential equations using LS-SVM". . In: *Neurocomputing* 159 (2015), pp. 105–116.

Forward Problem: PDEs

One dimensional PDEs

We consider the PDE of the form:

$$\begin{cases} \mathscr{L}u(\boldsymbol{x}) = f(\boldsymbol{x}), & \boldsymbol{x} \in \Sigma \in \mathbb{R}^2, \\ \mathscr{B}u(\boldsymbol{x}) = g(\boldsymbol{x}), & \boldsymbol{x} \in \partial\Sigma \end{cases} \tag{1}$$

- Σ is a bounded domain, which can be either rectangular or irregular,
- $\partial\Sigma$ represents its boundary.
- $\mathcal{B}$ and $\mathscr{L}$ are differential operators.

Our goal is to find $\hat{u}$ that satisfies (1) on the given domain Σ:

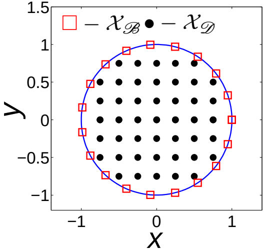
$$\begin{aligned} \underset{\hat{u}}{\text{minimize}} \quad & \|\mathscr{L}\hat{u} - f\| \\ \text{subject to} \quad & \mathscr{B}\hat{u} = g \end{aligned} \tag{2}$$

Formulation of the method

Collocation method: discretization of the domain Σ into a set of collocation points defined as follows:

$$\mathcal{X} = \left\{ \mathbf{x}^k \mid \mathbf{x}^k = (x_k, t_k),\ k = 1, \ldots, k_{end} \right\},$$

where $\mathcal{X} = \mathcal{X}_{\mathscr{D}} \cup \mathcal{X}_{\mathscr{B}}$.



Formulation of the method

One can rewrite (1) as the following optimization problem:

$$\begin{aligned} \underset{\hat{u}}{\text{minimize}} \quad & \frac{1}{2} \sum_{i=1}^{|\mathcal{X}_{\mathscr{D}}|} \left[(\mathscr{L}[\hat{u}] - f)(\mathbf{x}_{\mathscr{D}}^{i}) \right]^2 \\ \text{subject to} \quad & \mathscr{B}[\hat{u}(\mathbf{x}_{\mathscr{B}}^{j})] = g(\mathbf{x}_{\mathscr{B}}^{j}),\ \ j = 1, \ldots, |\mathcal{X}_{\mathscr{B}}|. \end{aligned} \tag{3}$$

Forward Problem: PDEs

Consider the case where $\mathscr{L}$ is defined as follows:

$$\mathscr{L} \equiv \frac{\partial^2 u}{\partial t^2} + a(x,t)\frac{\partial u}{\partial t} + b(x,t)u - c(x,t)\frac{\partial^2 u}{\partial x^2}.$$

subject to a Dirichlet boundary condition, i.e.

$$u(\boldsymbol{x}) = g(\boldsymbol{x}) \text{ for all } \boldsymbol{x} \in \partial\Sigma.$$

The approach can be summarized as follows:

Steps needed

- Assume that a general approximate solution is of the following form:

$$\hat{u}(\boldsymbol{x}) = w^T \varphi(\boldsymbol{x}) + d \tag{4}$$

where $\varphi(\cdot) : \mathbb{R}^{dim} \rightarrow \mathbb{R}^h$ is the feature map.

Forward Problem: PDEs

- Solve the optimization problem:

$$\begin{aligned}
\underset{\boldsymbol{w},d,\boldsymbol{e}}{\text{minimize}} \quad & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \frac{\gamma}{2}\boldsymbol{e}^T\boldsymbol{e} \\
\text{subject to} \quad & \boldsymbol{w}^T\Big[\varphi_{tt}(\boldsymbol{x}_{\mathscr{D}}^i) + a(\boldsymbol{x}_{\mathscr{D}}^i)\varphi_t(\boldsymbol{x}_{\mathscr{D}}^i) + b(\boldsymbol{x}_{\mathscr{D}}^i)\varphi(\boldsymbol{x}_{\mathscr{D}}^i) - \\
& \quad c(\boldsymbol{x}_{\mathscr{D}}^i)\varphi_{xx}(\boldsymbol{x}_{\mathscr{D}}^i)\Big] + b(\boldsymbol{x}_{\mathscr{D}}^i)d = f(\boldsymbol{x}_{\mathscr{D}}^i) + e_i, \; i = 1, \ldots, |\mathcal{X}_{\mathscr{D}}|, \\
& \boldsymbol{w}^T\varphi(\boldsymbol{x}_{\mathscr{B}}^i) + d = g(t_i), \; i = 1, \ldots, |\mathcal{X}_{\mathscr{B}}|.
\end{aligned}$$

Forward Problem: PDEs

Linear system [5]

$$\left[\begin{array}{c|c|c} \mathcal{K}+\gamma^{-1}I_N & S_{\mathscr{B}} & \boldsymbol{b} \\ \hline S_{\mathscr{B}}^T & \Delta_{\mathscr{B}} & \mathbf{1}_M \\ \hline \boldsymbol{b}^T & \mathbf{1}_M^T & 0 \end{array}\right] \left[\begin{array}{c} \boldsymbol{\alpha} \\ \hline \boldsymbol{\beta} \\ \hline d \end{array}\right] = \left[\begin{array}{c} \boldsymbol{f} \\ \hline \boldsymbol{g} \\ \hline 0 \end{array}\right] \quad (5)$$

[5] Siamak Mehrkanoon and Johan AK Suykens. “Learning solutions to partial differential equations using LS-SVM”. . In: *Neurocomputing* 159 (2015), pp. 105–116.

Forward Problem: PDEs

- The optimal representation in dual:

$$\hat{u}(\boldsymbol{x}) = \sum_{i=1}^{|\mathcal{X}_{\mathscr{D}}|} \alpha_i \Bigg(\Big[\nabla_{t_1^{(2)},0} K\Big](\boldsymbol{x}^i_{\mathscr{D}}, \boldsymbol{x}) + a(\boldsymbol{x}^i_{\mathscr{D}}) \Big[\nabla_{t_1,0} K\Big](\boldsymbol{x}^i_{\mathscr{D}}, \boldsymbol{x}) + b(\boldsymbol{x}^i_{\mathscr{D}}) \Big[\nabla_{0,0} K\Big](\boldsymbol{x}^i_{\mathscr{D}}, \boldsymbol{x}) - c(\boldsymbol{x}^i_{\mathscr{D}}) \Big[\nabla_{x_1^{(2)},0} K\Big](\boldsymbol{x}^i_{\mathscr{D}}, \boldsymbol{x}) \Bigg) + \sum_{i=1}^{|\mathcal{X}_{\mathscr{B}}|} \beta_i \Big[\nabla_{0,0} K\Big](\boldsymbol{x}^i_{\mathscr{B}}, \boldsymbol{x}) + d.$$

where $[\nabla_{0,0}K](t,s) = \varphi(t)^T \varphi(s)$ and $[\nabla_{t,0}K](t,s) = \frac{\partial(\varphi(t)^T \varphi(s))}{\partial t}$ are the kernel function and its derivative respectively.

Rectangular domains

Consider the case where $\mathscr{L}$ is defined as follows:

$$\mathscr{L} \equiv \frac{\partial^2 u}{\partial t^2} + a(x,t)\frac{\partial u}{\partial t} + b(x,t)u - c(x,t)\frac{\partial^2 u}{\partial x^2}.$$

And the initial conditions of the form

$$u(x,0) + \frac{\partial u(x,0)}{\partial t} = h(x), \quad 0 \le x \le 1$$

and boundary conditions at $x = 0$ and $x = 1$ of the form:

$$u(0,t) = g_0(t), \quad u(1,t) = g_1(x), \quad 0 \le t \le T.$$

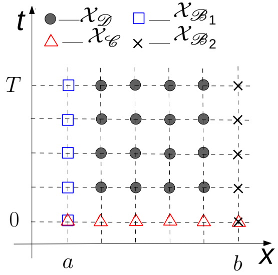


Figure:
$\mathcal{X}_{\mathscr{B}} = \mathcal{X}_{\mathscr{C}} \cup \mathcal{X}_{\mathscr{B}_1} \cup \mathcal{X}_{\mathscr{B}_2}$

Forward Problem: PDEs

The approach can be summarized as follows:

- Assume that $\hat{u}(\mathbf{x}) = w^T \varphi(\mathbf{x}) + d$, where $\varphi(\cdot) : \mathbb{R}^{dim} \to \mathbb{R}^h$.
- Solve the optimization problem:

$$
\begin{aligned}
\min_{\mathbf{w},d,\mathbf{e}} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + \frac{\gamma}{2}\mathbf{e}^T\mathbf{e} \\
\text{s.t} \quad & \mathbf{w}^T\Big[\varphi_{tt}(\mathbf{x}_{\mathscr{D}}^i) + a(\mathbf{x}_{\mathscr{D}}^i)\varphi_t(\mathbf{x}_{\mathscr{D}}^i) + b(\mathbf{x}_{\mathscr{D}}^i)\varphi(\mathbf{x}_{\mathscr{D}}^i) - c(\mathbf{x}_{\mathscr{D}}^i)\varphi_{xx}(\mathbf{x}_{\mathscr{D}}^i)\Big] \\
& \quad + b(\mathbf{x}_{\mathscr{D}}^i)d = f(\mathbf{x}_{\mathscr{D}}^i) + e_i, \; i = 1, \ldots, |\mathcal{X}_{\mathscr{D}}|, \\
& \mathbf{w}^T\Big[\varphi(\mathbf{x}_{\mathscr{C}}^i) + \varphi_t(\mathbf{x}_{\mathscr{C}}^i)\Big] + d = h(x_i), \; i = 1, \ldots, |\mathcal{X}_{\mathscr{C}}|, \\
& \mathbf{w}^T\varphi(\mathbf{x}_{\mathscr{B}_1}^i) + d = g_0(t_i), \; i = 1, \ldots, |\mathcal{X}_{\mathscr{B}_1}|, \\
& \mathbf{w}^T\varphi(\mathbf{x}_{\mathscr{B}_2}^i) + d = g_1(t_i), \; i = 1, \ldots, |\mathcal{X}_{\mathscr{B}_2}|,
\end{aligned}
$$

Forward Problem: PDEs

Linear system [6]

$$\left[\begin{array}{c|c|c} \mathcal{K}+\gamma^{-1}I_N & S & \boldsymbol{b} \\ \hline S^T & \Delta & \mathbf{1}_M \\ \hline \boldsymbol{b}^T & \mathbf{1}_M^T & 0 \end{array}\right] \left[\begin{array}{c} \boldsymbol{\alpha} \\ \hline \boldsymbol{\beta} \\ \hline d \end{array}\right] = \left[\begin{array}{c} \boldsymbol{f} \\ \hline \boldsymbol{v} \\ \hline 0 \end{array}\right]. \tag{6}$$

[6] Siamak Mehrkanoon and Johan AK Suykens. "Learning solutions to partial differential equations using LS-SVM". . In: *Neurocomputing* 159 (2015), pp. 105–116.

Forward Problem: PDEs

- The optimal representation in dual:

$$\begin{aligned}\hat{u}(\boldsymbol{x}) = {\color{red}d} + \sum_{i=1}^{|\mathcal{X}_{\mathscr{D}}|} {\color{red}\alpha_i} \Bigg(\Big[\nabla_{t_1^{(2)},0} K\Big](\boldsymbol{x}_{\mathscr{D}}^i, \boldsymbol{x}) + a(\boldsymbol{x}_{\mathscr{D}}^i)\Big[\nabla_{t_1,0} K\Big](\boldsymbol{x}_{\mathscr{D}}^i, \boldsymbol{x}) + \\ b(\boldsymbol{x}_{\mathscr{D}}^i)\Big[\nabla_{0,0} K\Big](\boldsymbol{x}_{\mathscr{D}}^i, \boldsymbol{x}) - c(\boldsymbol{x}_{\mathscr{D}}^i)\Big[\nabla_{x_1^{(2)},0} K\Big](\boldsymbol{x}_{\mathscr{D}}^i, \boldsymbol{x})\Bigg) + \\ \sum_{i=1}^{|\mathcal{X}_{\mathscr{C}}|} {\color{red}\beta_i^1} \Big[\nabla_{0,0} K + \nabla_{t_1,0} K\Big](\boldsymbol{x}_{\mathscr{C}}^i, \boldsymbol{x}) + \\ \sum_{i=1}^{|\mathcal{X}_{\mathscr{B}_1}|} {\color{red}\beta_i^2} \Big[\nabla_{0,0} K\Big](\boldsymbol{x}_{\mathscr{B}_1}^i, \boldsymbol{x}) + \\ \sum_{i=1}^{|\mathcal{X}_{\mathscr{B}_2}|} {\color{red}\beta_i^3} \Big[\nabla_{0,0} K\Big](\boldsymbol{x}_{\mathscr{B}_2}^i, \boldsymbol{x}).\end{aligned}$$

where $[\nabla_{0,0}K](t,s) = \varphi(t)^T\varphi(s)$ and $[\nabla_{t,0}K](t,s) = \frac{\partial(\varphi(t)^T\varphi(s))}{\partial t}$ are the kernel function and its derivative respectively.

Nonlinear PDEs

We assume that the nonlinear PDE has the following form:

$$\frac{\partial^2 u}{\partial t^2} + \frac{\partial^2 u}{\partial x^2} + f(u) = g(\boldsymbol{x}), \quad \boldsymbol{x} \in \Sigma \in \mathbb{R}^2$$

subject to the boundary conditions of the form

$$u(\boldsymbol{x}) = h(\boldsymbol{x}), \boldsymbol{x} \in \partial\Sigma$$

where f is a nonlinear function.

Forward Problem: PDEs

$$
\begin{aligned}
\underset{\boldsymbol{w},d,\boldsymbol{e},\boldsymbol{\xi},u}{\text{minimize}} \quad & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \frac{\gamma}{2}(\boldsymbol{e}^T\boldsymbol{e} + \boldsymbol{\xi}^T\boldsymbol{\xi}) \\
\text{subject to} \quad & \boldsymbol{w}^T\left[\varphi_{tt}(\boldsymbol{x}_{\mathscr{D}}^i) + \varphi_{xx}(\boldsymbol{x}_{\mathscr{D}}^i)\right] + f(u(\boldsymbol{x}_{\mathscr{D}}^i)) \\
& = g(\boldsymbol{x}_{\mathscr{D}}^i) + e_i,\ i = 1, \ldots, |\mathcal{X}_{\mathscr{D}}|, \\
& \boldsymbol{w}^T\varphi(\boldsymbol{x}_{\mathscr{D}}^i) + d = u(\boldsymbol{x}_{\mathscr{D}}^i) + \xi_i,\ i = 1, \ldots, |\mathcal{X}_{\mathscr{D}}|, \\
& \boldsymbol{w}^T\varphi(\boldsymbol{x}_{\mathscr{B}}^i) + d = h(\boldsymbol{x}_{\mathscr{B}}^i),\ i = 1, \ldots, |\mathcal{X}_{\mathscr{B}}|.
\end{aligned}
\tag{7}
$$

Note that the second set of additional constraints is introduced to keep the optimization problem linear in $\boldsymbol{w}$.

Example 1. Consider the linear second order hyperbolic equation with variable coefficients defined on a rectangular domain:

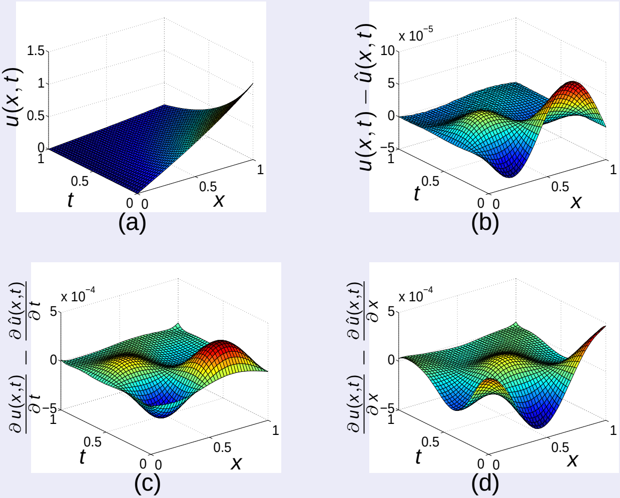
$$u_{tt} + 2e^{x+t}u_t + (\sin^2(x+t))u = (1+x^2)u_{xx} + e^{-2t}\Big(x^2 + 4e^{t+x} - \sin^2(t+x) - 3\Big)\sinh(x), \quad 0 < x < 1, 0 < t < T,$$

with exact solution $u(x,t) = e^{-2t}\sinh(x)$.

The number of collocation points (training points) inside and on the boundary of the domain are as follows:

- $|\mathcal{X}_{\mathscr{D}}| = 81$,
- $|\mathcal{X}_{\mathscr{C}}| = |\mathcal{X}_{\mathscr{B}_1}| = |\mathcal{X}_{\mathscr{B}_2}| = 10$

Experimental results



Experimental results

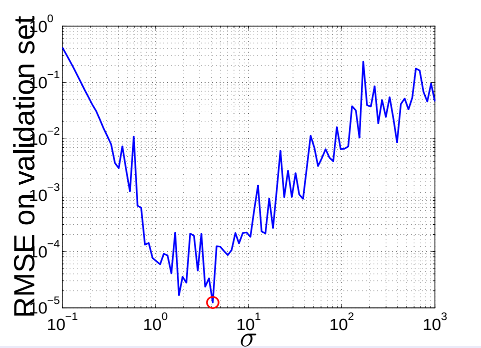


Figure: Tuning the kernel bandwidth (σ) using validation set. The red circle indicates the location of selected bandwidth.

Experimental results

Table: Numerical result of the proposed method for solving Problem 1 with time interval $[0, T]$.

Method	T	RMSE Training	RMSE Test	L_∞ Training	L_∞ Test
LSSVM	1	1.75×10^{-5}	1.94×10^{-5}	5.31×10^{-5}	6.71×10^{-5}
FDM [7]		– – – – – –	0.74×10^{-4}	– – – – – –	– – – – – –
LSSVM	2	3.18×10^{-5}	3.49×10^{-5}	1.30×10^{-4}	1.51×10^{-4}
FDM		– – – – – –	0.43×10^{-4}	– – – – – –	– – – – – –

[7] RK Mohanty. "An unconditionally stable finite difference formula for a linear second order one space dimensional hyperbolic equation with variable coefficients". In: *Applied Mathematics and Computation* 165.1 (2005), pp. 229–236.

Experimental results

Table: The effect of number of training points on the approximate solution of Problem 1 with time interval $[0, 1]$.

		RMSE		L_∞	
$\lvert\mathcal{X}_{\mathscr{D}}\rvert$	σ	Training	Test	Training	Test
4	225.04	1.76×10^{-3}	2.78×10^{-3}	3.50×10^{-3}	1.01×10^{-2}
25	12.61	6.26×10^{-4}	7.57×10^{-4}	1.76×10^{-3}	2.32×10^{-3}
49	5.99	2.58×10^{-4}	2.86×10^{-4}	7.31×10^{-4}	8.93×10^{-4}
81	4.13	1.75×10^{-5}	1.94×10^{-5}	5.31×10^{-5}	6.71×10^{-5}

Experimental results

Example 2. Consider elliptic equation defined on a rectangular domain:

$$\nabla^2 u(x,y) = \exp(-x)(x - 2 + y^3 + 6y)$$

with $x, y \in [0,1]$ and the Dirichlet boundary conditions:

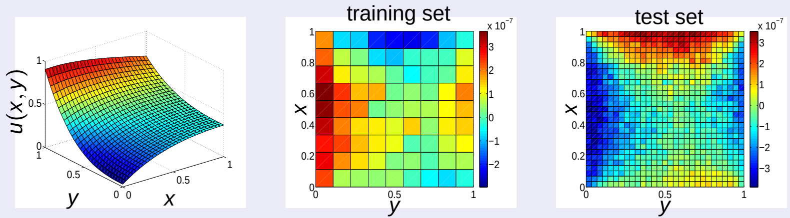
$$u(0,y) = y^3, \;\; u(1,y) = (1 + y^3)\exp(-1)$$

and

$$u(x,0) = x\exp(-x), \;\; u(x,1) = x\exp(-x)(x+1)$$

The exact solution is $u(x,y) = e^{-x}(x + y^3)$.

Experimental results



(a) (b) (c)

Figure: (b) 100 training points inside the domain $[0,1] \times [0,1]$ are used for training, (c) 900 points inside the domain $[0,1] \times [0,1]$ are used for testing.

Experimental results

Example 3. Consider the linear second order elliptic PDE:

$$\nabla^2 u(x,y) = 4x\cos(x) + (5 - x^2 - y^2)\sin(x) \tag{8}$$

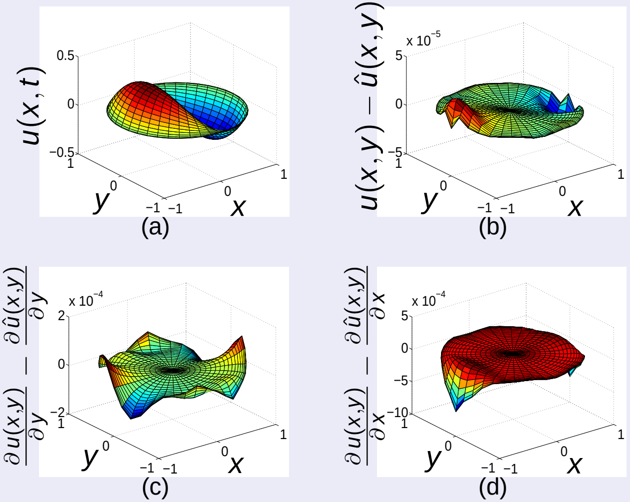
defined on a circular domain, i.e.

$$\Sigma := \left\{ (x,y) \,\middle|\, x^2 + y^2 - 1 = 0,\ -1 \le x \le 1, -1 \le y \le 1 \right\}$$

with the Dirichlet condition $u(x,y) = 0$ on $\partial\Sigma$. The exact solution is given by $u(x,y) = (x^2 + y^2 - 1)\sin(x)$.

- $|\mathcal{X}_{\mathscr{D}}| = 45$
- $|\mathcal{X}_{\mathscr{B}}| = 19$

Experimental results



Experimental results

Table: Numerical result of the proposed method for solving Problem 3

Problem	Method	MSE Training	MSE Test	L_∞ Training	L_∞ Test
3	LSSVM	5.18×10^{-11}	5.94×10^{-11}	1.91×10^{-5}	2.71×10^{-5}
	GPA [a]	− − − − −	2.04×10^{-4}	− − − − −	− − − − −

[a] András Sóbester, Prasanth B Nair, and Andy J Keane. "Genetic programming approaches for solving elliptic partial differential equations". In: *IEEE transactions on evolutionary computation* 12.4 (2008), pp. 469–478.

Experimental results

Example 4. Consider an example of nonlinear PDE

$$\nabla^2 u(x,y) + u(x,y)^2 = \sin(\pi x)\Big(2 - (\pi y)^2 + t^4 \sin(\pi x)\Big) \tag{9}$$

defined on a circular domain, i.e.

$$\Sigma := \Big\{(x,y)\,\Big|\, x^2 + y^2 - 1 = 0,\ -1 \le x \le 1, -1 \le y \le 1\Big\}$$

with the Dirichlet condition on $\partial\Sigma$. The exact solution is given by $u(x,y) = y^2 \sin(\pi x)$.

- $|\mathcal{X}_{\mathscr{D}}| = 24$
- $|\mathcal{X}_{\mathscr{B}}| = 19$.

Experimental results

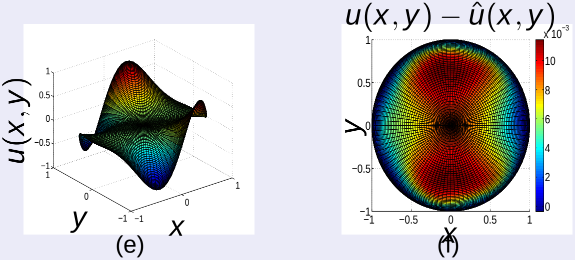


Figure: Obtained model error for problem 4.

Forward Problem: DAEs

DAEs:

Dynamical processes that are constrained e.g. by:

- conservation laws
- balance conditions
- geometric conditions

Known as descriptor, implicit or singular systems.

concentrations, populations of species, or just numbers of cells

Numerous applications in Economical, biological or chemical systems.

Forward Problem: DAEs

A semi-explicit DAE or an ODE with constraints:

$$\dot{x} = f(x, y, t)$$
$$0 = g(x, y, t).$$

- x and y are considered as differential and algebraic variables respectively.
- DAEs are characterized by their index
- If $\frac{\partial g}{\partial y}$ is nonsingular $\Rightarrow$ the index is 1

Forward Problem: DAEs

A semi-explicit DAE or an ODE with constraints:

$$\dot{x} = f(x, y, t)$$
$$0 = g(x, y, t).$$

- x and y are considered as differential and algebraic variables respectively.
- DAEs are characterized by their index
- If $\frac{\partial g}{\partial y}$ is nonsingular $\Rightarrow$ the index is 1

Forward Problem: DAEs

A semi-explicit DAE or an ODE with constraints:

$$\begin{aligned} \dot{x} &= f(x, y, t) \\ 0 &= g(x, y, t). \end{aligned}$$

- x and y are considered as differential and algebraic variables respectively.
- DAEs are characterized by their index
- If $\frac{\partial g}{\partial y}$ is nonsingular $\Rightarrow$ the index is 1

Forward Problem: DAEs

A semi-explicit DAE or an ODE with constraints:

$$\begin{aligned} \dot{x} &= f(x, y, t) \\ 0 &= g(x, y, t). \end{aligned}$$

- x and y are considered as differential and algebraic variables respectively.
- DAEs are characterized by their index
- If $\frac{\partial g}{\partial y}$ is nonsingular $\Rightarrow$ the index is 1.

Forward Problem: DAEs

Initial value problems (IVPs):

Consider a linear time varying IVPs in DAEs of the form

$$Z(t)\dot{X}(t) = A(t)X(t) + B(t)u(t), \; t \in [t_{in}, t_f], \; X(t_{in}) = X_0,$$

- $Z(t)$ is singular on $[t_{in}, t_f]$ with variable rank and the DAE may have an index that is larger than one.
- When $Z(t)$ is nonsingular, DAE can be converted to an equivalent explicit ODE system.

Forward Problem: DAEs

Assume that an approximate solution to i-th equation:

$$\hat{x}_i(t) = w_i^T \varphi(t) + d_i$$

where $\varphi(\cdot) : \mathbb{R} \to \mathbb{R}^h$ is the feature map and h is the dimension of the feature space.

Primal Problem

$$\begin{aligned} \underset{w_i, d_i, e_\ell^i}{\text{minimize}} \quad & \frac{1}{2}\sum_{\ell=1}^{m} w_\ell^T w_\ell + \frac{\gamma}{2}\sum_{\ell=1}^{m} e_\ell^T e_\ell \\ \text{subject to} \quad & Z\, W^T \Psi = A\left[W^T \Phi + D\right] + G + E, \\ & W^T \varphi(t_1) + D_{:,1} = X_0 \end{aligned}$$

Forward Problem: DAEs

The solution in dual form becomes:

$$\hat{x}_\ell(t) = \sum_{v=1}^{m}\sum_{i=2}^{N}\alpha_i^v\Big(z_{v\ell}(t_i)[\nabla_1^0 K](t_i,t) - a_{v\ell}(t_i)[\nabla_0^0 K](t_i,t)\Big) + \beta_\ell\,[\nabla_0^0 K](t_1,t) + d_\ell,\ \ell = 1,\dots,m.$$

- α, β and d follow from a square linear system.

[See[8]]

[8] Siamak Mehrkanoon and Johan AK Suykens. "LS-SVM approximate solution to linear time varying descriptor systems". In: *Automatica* 48.10 (2012), pp. 2502–2511.

Forward Problem: DAEs

Example 1 Consider the singular system of index-3

$$Z(t)\dot{X}(t) = A(t)X(t) + B(t)u(t),\ t \in [0, 20],\ X(0) = X_0$$

where $Z = \begin{bmatrix} 0 & -t & 0 \\ 1 & 0 & t \\ 0 & 1 & 0 \end{bmatrix}$, $A = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$

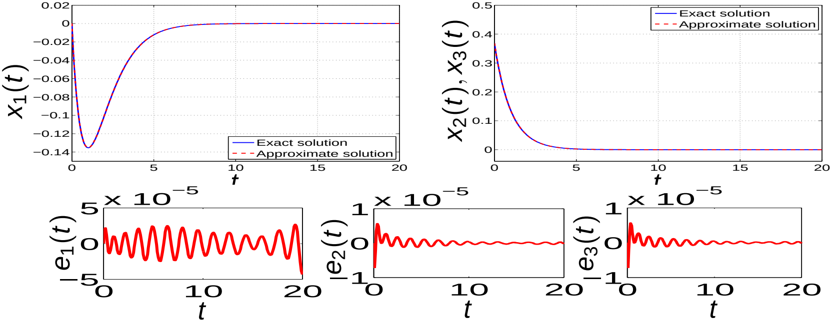
and $B(t) = 0$ with $x(0) = [0, e^{-1}, e^{-1}]^T$.

The problem is solved on domain $t \in [0, 20]$ using $N = 70$.

Forward Problem: DAEs

The exact solution is given by

$$x_1(t) = -t\exp(-(t+1)), \quad x_2(t) = x_3(t) = \exp(-(t+1)).$$



Forward Problem: DAEs

Table: Numerical results of the proposed method for solving Example 1 on time interval [0,20], with N number of collocation points.

N	MSE_{test} x_1	x_2	x_3
20	1.33×10^{-5}	4.82×10^{-8}	4.73×10^{-7}
40	1.38×10^{-8}	1.39×10^{-10}	3.14×10^{-9}
60	4.82×10^{-10}	3.54×10^{-12}	2.38×10^{-10}

Forward Problem: DAEs

BVPs in DAEs

Consider linear time varying boundary value problem in DAEs of the following from

$$Z(t)\dot{X}(t) = A(t)X(t) + g(t),\ t \in [t_{in}, t_f],$$
$$FX(t_{in}) + HX(t_f) = X_0,$$

Forward Problem: DAEs

Primal

$$\begin{aligned}
\underset{w_i, d_i, e^i_\ell}{\text{minimize}} \quad & \frac{1}{2}\sum_{\ell=1}^{m} w_\ell^T w_\ell + \frac{\gamma}{2}\sum_{\ell=1}^{m} e_\ell^T e_\ell \\
\text{subject to} \quad & Z\, W^T \Psi = A\left[W^T \Phi + D\right] + G + E, \\
& F[W^T \varphi(t_1) + D_{:,1}] + H[W^T \varphi(t_N) + D_{:,1}] = X_0
\end{aligned}$$

Dual

$$\left[\begin{array}{c|c|c} \mathcal{K} & \mathcal{U} & -F_A \\ \hline \mathcal{U}^T & \Delta & \Pi \\ \hline -F_A^T & \Pi^T & 0_{m\times m} \end{array}\right] \left[\begin{array}{c} \alpha \\ \hline \beta \\ \hline D_{:,1} \end{array}\right] = \left[\begin{array}{c} \tilde{G} \\ \hline X_0 \\ \hline 0 \end{array}\right]$$

Forward Problem: DAEs

The model in the dual form becomes:

$$\hat{x}_\ell(t) = \sum_{v=1}^{m}\sum_{i=2}^{N-1}\alpha_i^v\Big(z_{v\ell}(t_i)[\nabla_1^0 K](t_i,t) - a_{v\ell}(t_i)[\nabla_0^0 K](t_i,t)\Big) +$$
$$\sum_{v=1}^{m}\beta_v\Big([\nabla_0^0 K](t_1,t)f_{v\ell} + [\nabla_0^0 K](t_N,t)h_{v\ell}\Big) +$$
$$b_\ell,\ \ell = 1,\dots,m.$$

Here $[\nabla_0^0 K](t,s)$ and $[\nabla_1^0 K](t,s)$ are defined as previously. α_i^v and β_ℓ are Lagrange multipliers.

Forward Problem: DAEs

Example 2 Consider the linear time varying index one boundary value problem of DAE given by:

$$Z(t)\dot{X}(t) = A(t)X(t) + g(t), \ t \in [0,1],$$

where $Z = \begin{bmatrix} 1 & -t & t^2 \\ 0 & 1 & -t \\ 0 & 0 & 0 \end{bmatrix}$, $A = \begin{bmatrix} -1 & (t+1) & -(t^2+2t) \\ 0 & 1 & 1-t \\ 0 & 0 & -1 \end{bmatrix}$ with $g(t) = [0, 0, \sin(t)]^T$ and boundary conditions

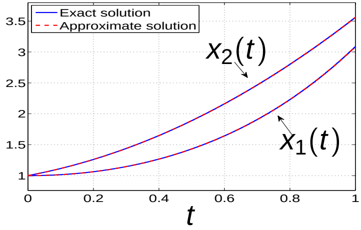
$$x_1(0) = 1, \ \ x_2(1) - x_3(1) = e.$$

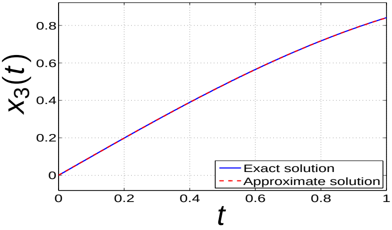
Forward Problem: DAEs

The exact solution is given by

$$x_1(t) = e^{-t} + te^{t}, \;\; x_2(t) = e^{t} + t\sin(t), \;\; x_3(t) = \sin(t).$$

The problem is solved on domain $t \in [0, 1]$ using $N = 10$.



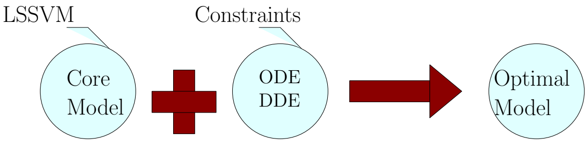


Problem Statement

We are given a dynamical system in state-space form

$$\dot{X}(t) = F(t, X(t), \theta), \tag{10}$$

The vector θ denotes unknown model parameters which can be either constant or time varying.

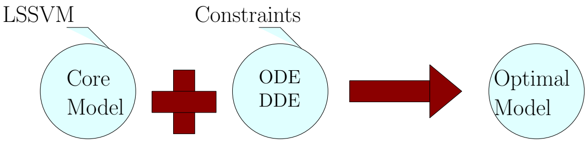


Goal

In order to estimate the unknown parameters, the state variable $X(t)$ is observed at N time instants $\{t_i\}_{i=1}^N$, so that we have

$$Y(t_i) = X(t_i) + E_i, \; i = 1, ..., N,$$

where $\{E_i\}_{i=1}^N$ are independent measurement errors with zero mean.



Goal

In order to estimate the unknown parameters, the state variable $X(t)$ is observed at N time instants $\{t_i\}_{i=1}^N$, so that we have

$$Y(t_i) = X(t_i) + E_i, \ i = 1, ..., N,$$

where $\{E_i\}_{i=1}^N$ are independent measurement errors with zero mean.

Estimating the time invariant parameters

First Step

- $\hat{x}_\ell(t) = w_\ell^T \varphi(t) + b_\ell = \sum_{i=1}^N \alpha_i^\ell K(t_i, t) + b_\ell, \; \ell = 1, ..., m,$
- $\frac{d}{dt}\hat{x}_\ell(t) = w_\ell^T \dot{\varphi}(t) = \sum_{i=1}^N \alpha_i^\ell \varphi(t_i)^T \dot{\varphi}(t) = \sum_{i=1}^N \alpha_i^\ell K_s(t_i, t), \; \ell = 1, ..., m.$

Second Step

$$\begin{aligned} \underset{\theta}{\text{minimize}} \quad & \frac{1}{2}\sum_i \|\Xi_i\|_2^2 \\ \text{subject to} \quad & \Xi_i = \frac{d}{dt}\hat{X}(t_i) - F(t_i, \hat{X}(t_i), \theta), \;\; i = 1, ..., N. \end{aligned}$$

If the system is linear in the parameters $\Rightarrow$ a convex optimization problem.

Estimating the time varying parameter

Consider the first order dynamical system of the form:

$$\frac{dx}{dt} + \theta(t)f(x(t)) = g(t),\ x(0) = x_0 \tag{11}$$

f is an arbitrary known function and $\theta(t)$ is the time varying parameter of the system and is considered to be unknown.

The state $x(t)$ has been measured at certain time instants $\{t_i\}_{i=1}^{N}$ i.e.

$$y_i = x(t_i) + e_i,\ i = 1, ..., N$$

where e_i's are i.i.d. random errors with zero mean and constant variance.

We assume an explicit LS-SVM model

$$\hat{\theta}(t) = v^T \psi(t) + b_\theta$$

as an approximation for the parameter $\theta(t)$.

We estimate the time-varying coefficient $\theta(t)$ by solving the following optimization problem:

$$\begin{aligned} \underset{v, b_\theta, e}{\text{minimize}} \quad & \frac{1}{2} v^T v + \frac{\gamma}{2} e^T e \\ \text{subject to} \quad & \frac{d}{dt}\hat{x}(t_i) + \left[v^T \psi(t_i) + b_\theta \right] f(\hat{x}(t_i)) = \\ & \hat{g}(t_i) + e_i, \ \text{for } i = 1, ..., M. \end{aligned} \tag{12}$$

Inverse Problem: ODEs

The solution to (12) can be obtained by solving the following dual problem [see[a]]

$$\left[\begin{array}{c|c} D\Omega D + I_M/\gamma & f(\hat{x}) \\ \hline f(\hat{x})^T & 0 \end{array}\right] \left[\begin{array}{c} \alpha \\ \hline b_\theta \end{array}\right] = \left[\begin{array}{c} \hat{g} - \frac{d\hat{x}}{dt} \\ \hline 0 \end{array}\right] \qquad (13)$$

[a] Siamak Mehrkanoon, Tillmann Falck, and Johan AK Suykens. "Parameter estimation for time varying dynamical systems using least squares support vector machines". In: *IFAC Proceedings Volumes* 45.16 (2012), pp. 1300–1305.

The model in the dual form becomes

$$\hat{\theta}(t) = v^T\psi(t) + b_\theta = \sum_{i=1}^{M} \alpha_i f(\hat{x}_i) K(t_i, t) + b_\theta \qquad (14)$$

where K is the kernel function.

Example 1. Consider the following nonlinear scalar dynamical system,

$$\frac{dx}{dt} - \frac{\cos(t)}{\sin(t)+2}\cos(x(t)^2) = \cos(t), \quad x(0) = 1$$

The aim is to estimate the time varying coefficient $\theta(t) = \frac{\cos(t)}{\sin(t)+2}$ from measured data. For collecting the data:

- Matlab built-in solver ode45 over the domain of $[0, 20]$ with sampling interval $T_s = 0.1$.
- Then we have artificially introduced random noise (Gaussian white noise with noise level η) to the true solution.

Inverse Problem: ODEs

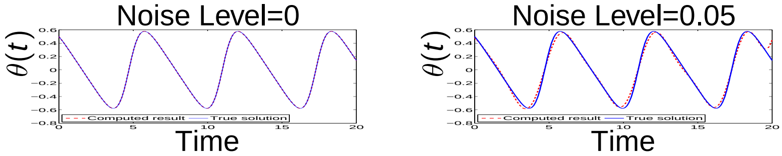


Figure: Estimation of time varying parameter of dynamical system formulated in Example 2.

Table: The influence of noise level and number of observed data on the parameter estimates. Parameter η is the std of the noise and N is the number of observed data.

N	η	**MSE**
100	0.0	8.34×10^{-5}
	0.05	3.51×10^{-3}
200	0.0	3.06×10^{-6}
	0.05	2.01×10^{-3}

Example 2. Consider the forced Van der Pol's Oscillator:

$$\dot{x}_1 = x_2,\; x_1(0) = -5,$$
$$\dot{x}_2 = \theta(1 - x_1^2)x_2 + 9x_1 = \sin(50t),\; x_2(0) = -1$$

where θ is the unknown parameter. In our study θ is taken as 1.1.

- The true solution is prepared by numerically integrating the equation on domain $[0, 10]$.
- Then the model observation data, i.e $y(t)$, is constructed using sampling interval $T_s = 0.01$ as follows:

$$y_k = x_1(t_k) + e_k.$$

Inverse Problem: ODEs

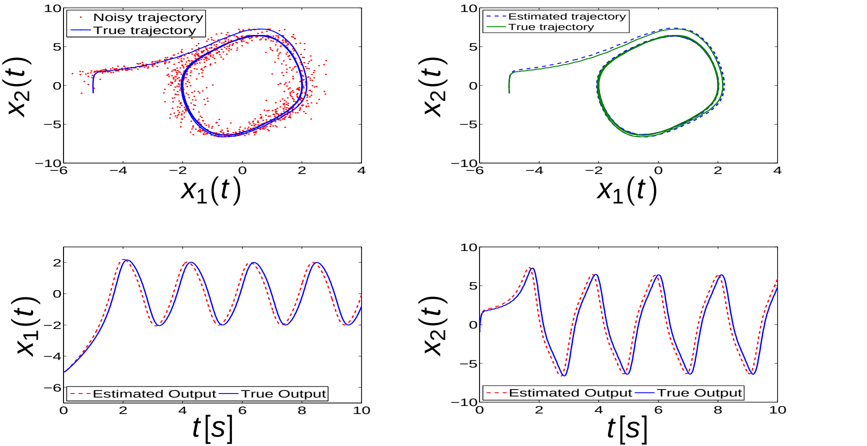


Figure: Estimation of the parameter θ for the forced nonlinear Van der Pol equation from data with observational noise generated using $\eta = 10$.

Conclusion & Future works

- Overview of LS-SVM based models for learning PDEs and DAEs solutions.
- Overview of LS-SVM based model for solving inverse problem in ODEs.
- Exploring and designing new deep architectures.
- Higher dimensional PDEs.

Demo

- Matlab demos:
  `https://sites.google.com/view/siamak-mehrkanoon/code-data`

References

- J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle, *Least Squares Support Vector Machines*, World Scientific Pub. Co., Singapore, 2002.
- S. Mehrkanoon, T. Falck and J.A.K Suykens, Approxmate solution to ordinary differential equations using least squares support vector machines0, *IEEE-TNNLS 23(9), pp. 1356-1367, 2012.*
- S. Mehrkanoon and J.A.K Suykens, LS-SVM approximate solution to linear time varying descriptor systems, *Automatica, 48, pp. 2502-2511, 2012.*
- S. Mehrkanoon, T. Falck, J.A.K. Suykens, Parameter Estimation for Time Varying Dynamical Systems using Least Squares Support Vector Machines, *SYSID 2012, Brussels, Belgium*, pp. 1300-1305.
- S. Mehrkanoon and J.A.K Suykens, Learning Solutions to Partial Differential Equations using LS-SVM, *Neurocomputing 159, pp. 105-116, 2015.*
- S. Mehrkanoon, S. Mehrkanoon, J.A.K. Suykens, Parameter estimation of delay differential equations: an integration-free LS-SVM approach, *CNSNS, vol. 19, no. 4, pp. 830-841, 2014*.
- S. Mehrkanoon, Deep Neural Kernel Blocks, *Neural Networks, 2019.*

Thank you for your attention