An Active Learning Framework for Set Inversion

Binh T. Nguyen[a,d], Duy M. Nguyen[a], Lam Si Tung Ho[b], Vu Dinh[c]

[a] *University of Science, Vietnam*
[b] *Dalhousie University, Halifax, Nova Scotia, Canada*
[c] *University of Delaware, USA*
[d] *Inspectorio Research Lab, Vietnam*

Abstract

Set inversion is a classical problem in control theory that has many vital applications in various fields of science and engineering. The state-of-the-art method for solving this problem, Set Inverter Via Interval Analysis (SIVIA), usually does not work well in high dimensions and often fails to recover sets with complicated structures. In this work, we propose a new approach to the problem of set inversion, which employs techniques from machine learning to resolve these issues. Our algorithm can handle problems in high dimensions and achieve the same level of accuracy with fewer data points compared to SIVIA. We illustrate the performance of our method in various simulation studies and apply it to investigate the dynamics of the 17th-century plague in Eyam village, England.

Keywords: set inversion, machine learning, active learning

1. Introduction

In many problems in science and engineering, it is often necessary to find the subset of the state space (parameter space) that satisfies specific desired properties. For example, when an acute inflammatory response to an infection is triggered, the initial amount of pathogens and anti-inflammatory mediators are essential to the survival of the tissue. Therefore, it is crucial to know the combinations of these two parameters that lead to a potentially lethal amount of tissue damage [1]. Such questions can be framed as a problem of set inversion, in which we want to find the pre-image $V = F^{-1}(U)$ for a set $U \in \mathbb{R}^t$ (which characterizes the desired properties) and a smooth function F (which describes the model) mapping the state space $\Omega \in \mathbb{R}^s$ to $\mathbb{R}^t$.

During the past decades, much effort has been made to develop methods for the set inversion problem. One primary direction is to rely on expert's knowledge about the geometric structure of the pre-image, which works well for the cases where V has a simple shape such as ellipsoids [2, 3], parallelotopes, or zonotopes [4]. When the pre-image has complicated geometry, Set Inverter Via Interval Analysis (SIVIA) proposed by Jaulin and Walter [5] is the most popular method. SIVIA approximates the pre-image by enclosing it between internal and external unions of boxes. However, the computational cost of this method increases exponentially concerning the dimension of the state space because the number of required boxes for covering a high dimensional pre-image is massive.

In the short version of this paper [6], we demonstrate that SIVIA fails when the dimension of the state space is 5 or higher, even when the shape of the pre-image is simple. To resolve this issue, we introduce a new machine learning-based framework, OASIS, which recasts set inversion as a problem of classifying the state space into two sets: the inside and the outside of the pre-image. OASIS uses an active learning mechanism to choose a sequence of informative samples and employ the Support Vector Machine (SVM) algorithm to construct the boundary of V. We show that this method achieves the same level of accuracy with fewer data points compared to SIVIA and inherits two important qualities from SVM:

working well with high dimension and having fast prediction times.

In this paper, we generalize OASIS to make it applicable to a wide range of classification methods and investigate the impact of such methods to OASIS. In particular, we consider four popular classification methods: SVM [7], *k*-nearest neighbors (KNN) [8], random forest (RF) [9], and multilayer perceptron (MLP) [10]. Also, our algorithm initializes the learning process using an "exploration" component that queries examples according to a random/near-random strategy (uniform sampling, Latin hypercube (LHS) [11], or Sobol sequences [12]). The simulation results show that our method works well for high dimensional problems which SIVIA cannot handle, and is slightly better at prediction compared to SIVIA for low dimensional problems. All combinations of classifiers and sampling methods have high prediction accuracy, but SVM and MLP perform better than the other two methods. Most notably, MLP can capture the shape of the boundary of the pre-image well while SVM achieves the highest prediction accuracy for high dimensional problems. Finally, we apply our proposed method to study the dynamics of the 17th-century plague in Eyam, an English village in the Derbyshire Dales District, United Kingdom. The source codes of our simulations and analysis are available online [1].

2. An active learning framework for set inversion

Nowadays, machine learning finds many real life applications including extracting brain tissues from high-resolution magnetic resonance images [13], human identification using eye movement [14], handwritten characters recognition

[1] https://github.com/BinhMisfit/active-learning-set-inversion

[15], automatic music generation [16], named entity recognition task from tweet streams [17], predicting employment [18], and food recognition [19]. In this section, we describe how to utilize machine learning to solve a set inversion problem. Recall that set inversion is a problem of finding the pre-image $V = F^{-1}(U)$ when directly computing F^{-1} is infeasible.

The main idea is to label a point in the state space as 1 if it is inside of the pre-image V and -1 otherwise. Then, finding V can be formulated as constructing the decision boundary of a classification problem. Specifically, we assume that for any $x \in \mathscr{X}$, we can evaluate the forward function F and check whether $F(x)$ belongs to the target set U. Our task is to approximate the following classifier

$$\phi(x) = \begin{cases} 1 & \text{if } F(x) \in U \\ -1 & \text{if } F(x) \notin U \end{cases} \tag{1}$$

using supervised learning methods.

In many practical applications, evaluating the forward function can be costly and time-consuming. To remedy this problem, we incorporate an active learning module into the algorithm to reduce the number of function evaluations. Our active learning procedure aggressively looks for informative samples for requesting labels. The algorithm avoids overfitting by using an “exploration” component that queries samples according to a random/near-random strategy to initialize the learning process. It is worth noticing that for set inversion problems, the evaluations of the forward functions are usually assumed to have no error. Therefore, the labels in our framework are always correct, which is an ideal scenario for an aggressive active learning scheme [20]. Even when there are errors in evaluating the forward function, our “exploration” component acts as a safety net for this situation.

Our general active learning framework for set inversion can be described as follows:

Algorithm 2.1.

1. *Exploration:*

   - *Select a sequence of points* $\{x_1,\dots,x_k\}$ *based on a random/near-random sampling scheme and obtain the corresponding labels* $\{y_1,\dots,y_k\}$.
   - *Construct a classifier* $\psi(x)$ *that perfectly separates the initial data according to their labels.*

2. *Active learning: Repeat the following steps until reaching the desired size of the training set:*

   - *Generate a random starting point* x_0 *on the state space* Ω *according to the uniform distribution.*
   - *Find the nearest point* x^* *to* x_0 *on the decision boundary* $\partial\psi$ *by solving the following optimization problem:*

$$\begin{aligned} &\textit{Minimize } \|x - x_0\|_2 \,. \\ &\textit{Subject to} \\ &\qquad\qquad x \in \partial\psi \end{aligned} \tag{2}$$

   - *Query the label of* x^* *and add* x^* *(with its corresponding label) to the training set.*
   - *Update* ψ *using the new training set.*

Here, the boundary $\partial\psi$ is determined as the curve which consists of all points in the state space Ω such that the predicted probability of the classifier at these points is equal to 0.5. Similar to OASIS, we use Sequential Quadratic Programming [21] to solve the optimization problem (2).

Depending on the classification method used in the algorithm, one can calibrate the corresponding hyper-parameters to reduce the empirical classification to zero. For example, OASIS tunes the parameter γ, the coefficient of the Radial Basis Function (RBF) kernel of SVM, so that the current training set are separated according to their labels. This step is done by starting with small values of γ and keep increasing it until the empirical classification error is smaller than a given threshold.

3. Numerical results

In this section, we examine the performance of our algorithm with a variety of sampling schemes (uniform sampling, Latin hypercube, Sobol sequences) and classification methods (SVM, KNN, RF, NN). The configuration of each classifier can be described as follows:

- SVM: We use the standard SVM algorithm with the RBF kernel. The value of the kernel coefficient γ is selected in the set:

  $$\left\{\frac{1}{d}, \frac{1}{d\sigma}, 0.01, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 6, 7, 8, 9, 10, 15, 20, 50, 100, 200\right\}$$

  where d is the dimension of the problem and σ is the sample standard error of the training points.

- KNN: The number of nearest neighbours of the algorithm is chosen in the set $\{\sqrt{N}, 5, 7, 10, 12, 15, 18, 20\}$ where N is the number of sampled points in the training dataset.

- MLP: We build a multi-layer perceptron classifier with the rectified linear unit (ReLU) activation function [22] with one input layer, two fully connected hidden layers, and an output layer. The number of nodes in two hidden layers are the same, which can be selected in the set $\{2, 5, 10, 15, 20\}$.

- RF: The random forest model is constructed by choosing the number of trees in the set $\{5, 10, 15, 20, 25, 30, 35, 40, 45, 50\}$.

We compare the performance of our methods with SIVIA, the state-of-the-art algorithm for set inversion. This algorithm is implemented in VSIVIA [2] [23], a vector implementation of SIVIA in MATLAB. VSIVIA can optimize the algorithm and mitigate the recursion of a large number of loops as well as function calls, therefore reduce computational time compared to the original algorithm.

In our simulations, we consider three distinct basic two-dimensional shapes (circle, doughnut, and ring) and spheres in higher dimensions (up to 8) for the preimage V. Furthermore, we demonstrate the applicability of our approach to the prevalent predator-prey problem. All experiments are performed on a computer with Intel(R) Core(TM) i9-7900X CPU, running at 3.6GHz with 128GB of RAM.

3.1. Simulations

We consider the set inversion problem in several dimensions (from 2–8). For two dimensions, we define two forward functions

$$F_1(x,y) = x^2 + y^2 \quad \text{and} \quad F_2(x,y) = x^2 + y^2 + xy$$

[2] `https://www.researchgate.net/publication/320623805_VSIVIA_V01`

to describe the following set inversion problems:

$$
\begin{aligned}
V_{\text{circle}} &= F_1^{-1}([0,2]) = \{(x,y) \in \Omega_{2D} \mid x^2+y^2 \leq 2\} \\
V_{\text{ring}} &= F_1^{-1}([1,2]) = \{(x,y) \in \Omega_{2D} \mid 1 \leq x^2+y^2 \leq 2\} \\
V_{\text{doughnut}} &= F_2^{-1}([1,2]) = \{(x,y) \in \Omega_{2D} \mid 1 \leq x^2+y^2+xy \leq 2\}
\end{aligned}
\tag{3}
$$

where $\Omega_{2D} = [-3,3] \times [-3,3]$ is the state space.

In all scenarios, our algorithm queries 100 initial points, then uses active learning to query 400 additional points. To assess the accuracy, we construct an evenly spaced grid of 601 points on each dimension and use them as the testing set. The accuracy is calculated by the fraction of correctly predicted points by the returned classifier on a testing set.

For higher dimensions, we consider problems where the pre-images are spheres. Specifically, the pre-images are defined as follows:

$$
\begin{aligned}
&V_{3D} = \{(x,y,z) \in \Omega_{3D} \mid x^2+y^2+z^2 \leq 0.5\} \\
&V_{kD} = \{(x_1,x_2,\ldots,x_k) \in \Omega_{kD} \mid \sum_{i=1}^{k} x_i^2 \leq 0.25\} \quad k = 4,5,6,7,8
\end{aligned}
$$

where

$$
\begin{aligned}
&\Omega_{3D} = [-1.5,1.5] \times [-1.5,1.5] \times [-1.5,1.5] \\
&\Omega_{4D} = [-1,1] \times [-1,1] \times [-1,1] \times [-1,1] \\
&\Omega_{kD} = [-0.6,0.6]^k, \quad k = 5,6,7,8.
\end{aligned}
$$

Again, we construct an evenly spaced grid of points and use them as the test set.

3.2. Comparing across various sampling and classification methods

The prediction accuracy of our methods is visualized in Figure 1. These experimental results suggest that there is little difference in the performances of different sampling methods. This is somewhat expected since both Latin hypercube and Sobol sequence are near-random schemes that mimic the behavior of the uniform distribution and should have the same efficiency in exploring the state space of the problems. Regarding classification methods, KNN clearly performs worse than the other three methods. On the other hand, SVM and MLP do well across all problems. MLP is the best methods in two-dimensional cases while SVM achieves the highest prediction accuracy when the dimension of the problem is 3 and higher.

In Figure 2, we plot the constructed pre-images $V_{doughnut}$ of the doughnut problem. We can observe that all methods perform reasonably well with a relatively low number of observations. Visually, SVM and MLP do better than KNN and RF in reconstructing the object. Noticeably, MLP seems to be able to capture the geometric structure of the pre-image better than all other methods. This phenomenon can also be observed in all other two-dimensional problems.

3.3. Comparing with VSIVIA

First of all, we note that VSIVIA fails for problems in dimensions five or higher due to being out of memory. This is expected because SIVIA approximates the pre-image by enclosing it between internal and external unions of boxes. As a consequence, the computational cost of the algorithm increases exponentially in high-dimensional problems due to the massive number of boxes required to cover the pre-image effectively. Our algorithm, on the other hand, does not have this issue.

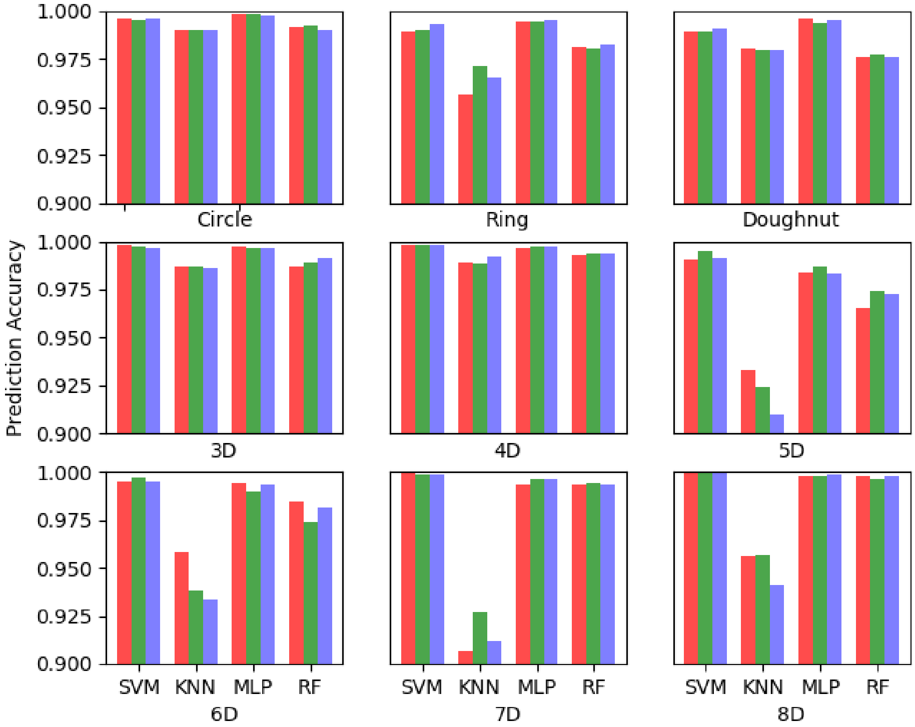

Figure 1: Prediction accuracy of different sampling schemes and classification methods. The colors of the bars correspond to the sampling methods: red corresponds to uniform sampling, green corresponds to low-discrepancy sampling by Sobol sequences and blue corresponds to Latin hypercube sampling.

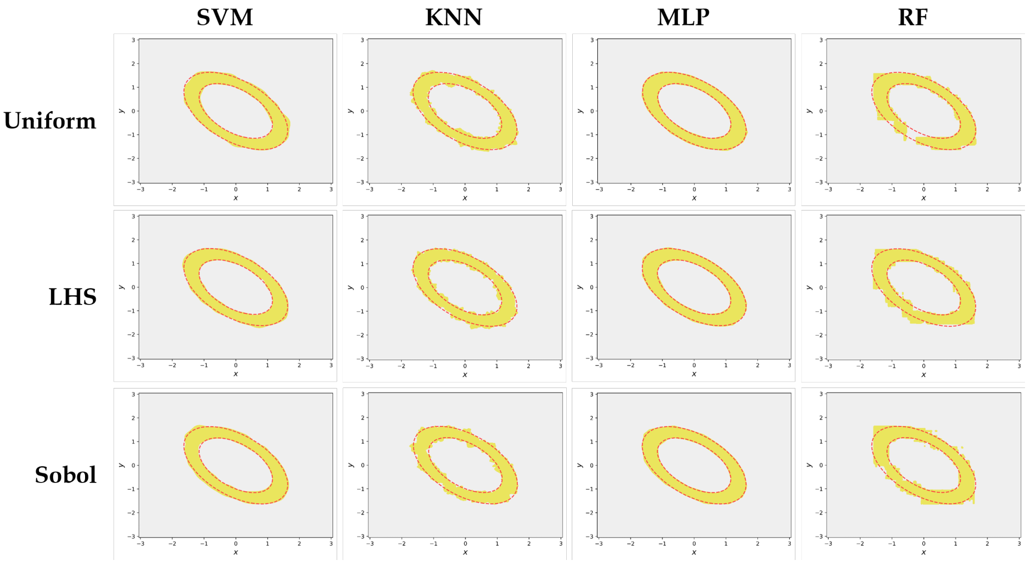

Figure 2: Doughnut problem: the red dashed line expresses the true boundary of $V_{doughnut}$, while the yellow region is the reconstructed set by our algorithms.

A comparison between our methods and VSIVIA is presented in Figure 3, for which we provide the prediction accuracy, training time, and average prediction time (to construct spheres in 2, 3 and 4 dimensions) of VSIVIA and our two best methods (SVM and MLP). The results show that our approaches are slightly better at prediction compared to VSIVIA. Although the training time of our techniques is longer than VSIVIA, our prediction time per point is noticeably faster. We want to point out that both training time and prediction time of VSIVIA seem to increase very quickly with respect to the dimension of the problem.

In practice, the time to train the classifier is a one-time cost that can be computed offline while the prediction of new data must be done as quickly and as accurately as possible. For example, in computer vision, the training period using deep learning may be allowed to run for weeks, while the prediction must be done

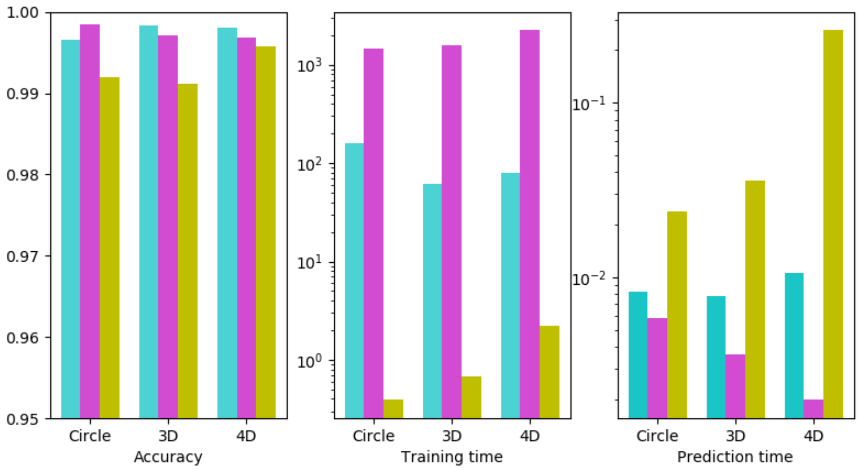

Figure 3: Prediction accuracy, (log) training time and (log) average prediction of classification methods for the problem of reconstructing spheres in 2, 3, and 4 dimensions. The colors of the bars correspond to the classification methods: cyan corresponds to SVM, magenta corresponds to MLP and yellow corresponds to VISIVIA.

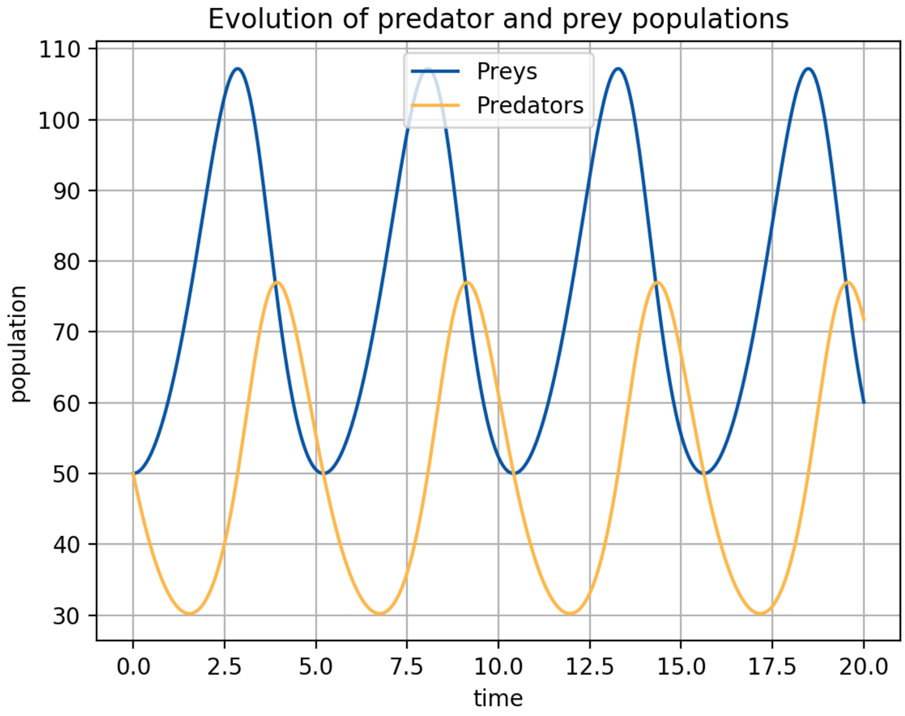

Figure 4: The dynamic of a predator-prey system under the Lotka-Volterra model with $x(0) = y(0) = 50, \alpha = 1, \beta = 0.02, \gamma = 1.5, \delta = 0.02$.

in a matter of seconds [13, 24]. Similarly, in the context of control theory, the offline-computed controller for decision-making is embedded into a device with limited memory and computational power [25, 26]. In such cases, our algorithm has the potential to make significant improvements over VSIVIA.

3.4. Predator-prey problem

Predator-prey problem considers the interaction between two different species: one is the predator while another one is the prey. The dynamics of this biological system is often modeled by the Lotka-Volterra equations:

$$\frac{dx}{dt} = \alpha x - \beta xy$$
$$\frac{dy}{dt} = \delta xy - \gamma y,$$

where x is the population of the prey, y is the population of the predator, α is the birth rate of the prey, β is the rate of a predator catching a prey, δ is the growth rate of predator due to the interaction with the prey, and γ is the death rate of the predator. Figure 4 visualizes an example of the dynamics of this system with $x(0) = y(0) = 50, \alpha = 1, \beta = 0.02, \gamma = 1.5, \delta = 0.02$.

In this simulation, we suppose that the initial populations of the predator and the prey are 50, the birth rate of the prey $\alpha = 1$, and the death rate of the predator $\gamma = 1$. We want to reconstruct the region of $(\beta, \delta) \in [0.01, 0.1] \times [0.01, 0.1]$ such that the population of the prey is always above 10 during the duration from time 0 to 20.

We use 400 initial points and 400 actively sampled points for all of our methods. The reconstructed pre-images are visualized in Figure 5, and the detailed results are given in Table 1. Again, the result is similar to what we have observed before. MLP is the most accurate method and has quick prediction times, but requires a longer training time compared to others. While SVM still performs well, it fails to recognize an important geometric feature near the edge of the state space. On the other hand, since the ground truth has a smooth boundary, it is expected that KNN and RF cannot capture this feature effectively.

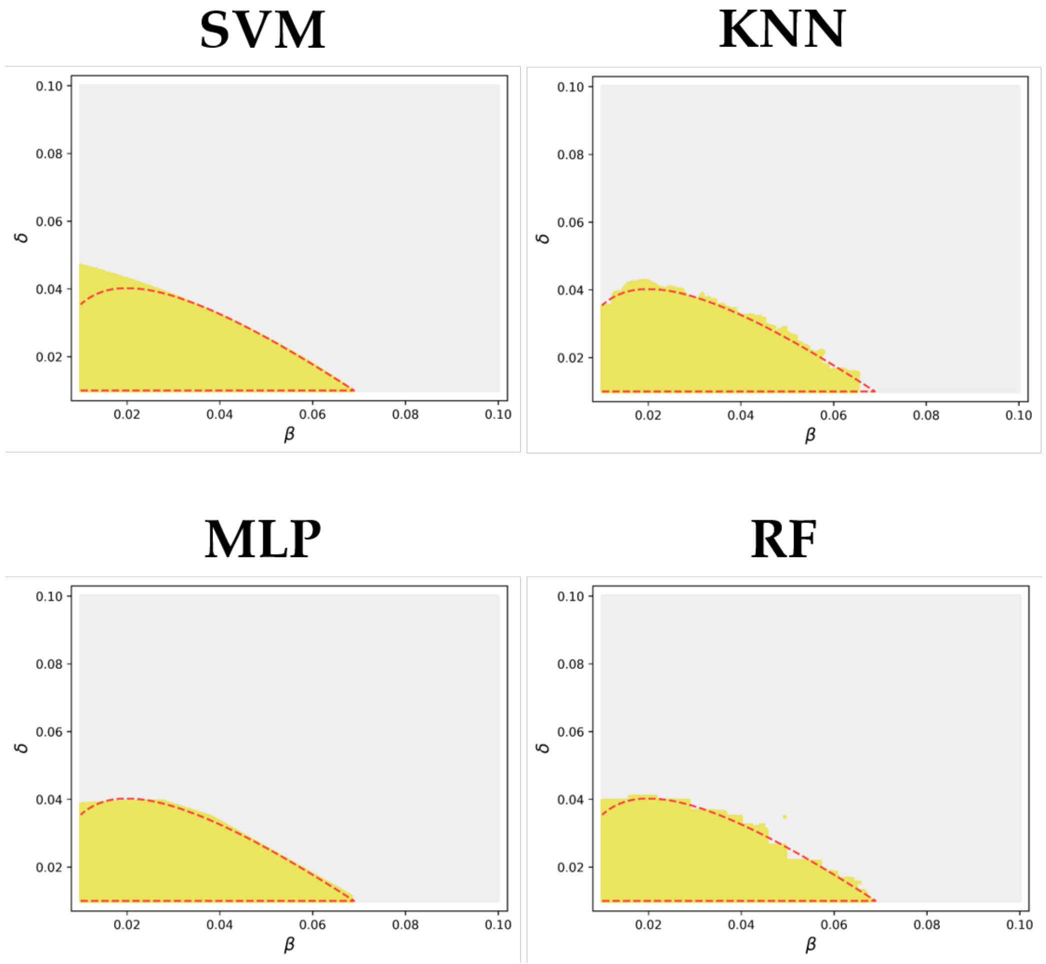

Figure 5: Reconstructed pre-images returned by our methods for the predator-prey problem. The red dashed line expresses the true boundary of V, while the yellow region is the reconstructed set by our algorithm.

Classifier	Accuracy	Training Time (s)	Predicting Time (s)
SVM	99.39%	121.63	10.95
KNN	99.38%	**33.88**	1.39
MLP	**99.56%**	5569.58	**0.33**
RF	99.50%	633.61	1.91

Table 1: Lotka-Volterra Result.

4. The 17th-century plague in Eyam

In this section, we apply our proposed method to study the dynamic of the 17th-century plague in Eyam, an English village in the Derbyshire Dales District, United Kingdom. During the epidemic, the villagers decided to isolate themselves from the outside to prevent the spread of the plague. Unfortunately, only 85 people had survived at the end of this outbreak. The total number of deaths over the period from June 18th to October 20th, 1666 can be obtained from the death list [27] (see Figure 6 for the visualization of the data). These data have been analyzed under the Susceptible-Infected-Removed (SIR) model using statistical methods such as the maximum likelihood estimator [27] and the Bayesian approach [28, 29]. Here, we re-analyze this epidemic from the set inversion point of view.

The SIR model divides the population into three groups: Susceptible (S) group includes healthy people, Infected (I) group includes infected people, and Removed (R) group includes people who are either recovered and get full immunity or die. The deterministic SIR model describes the dynamics of these groups during an

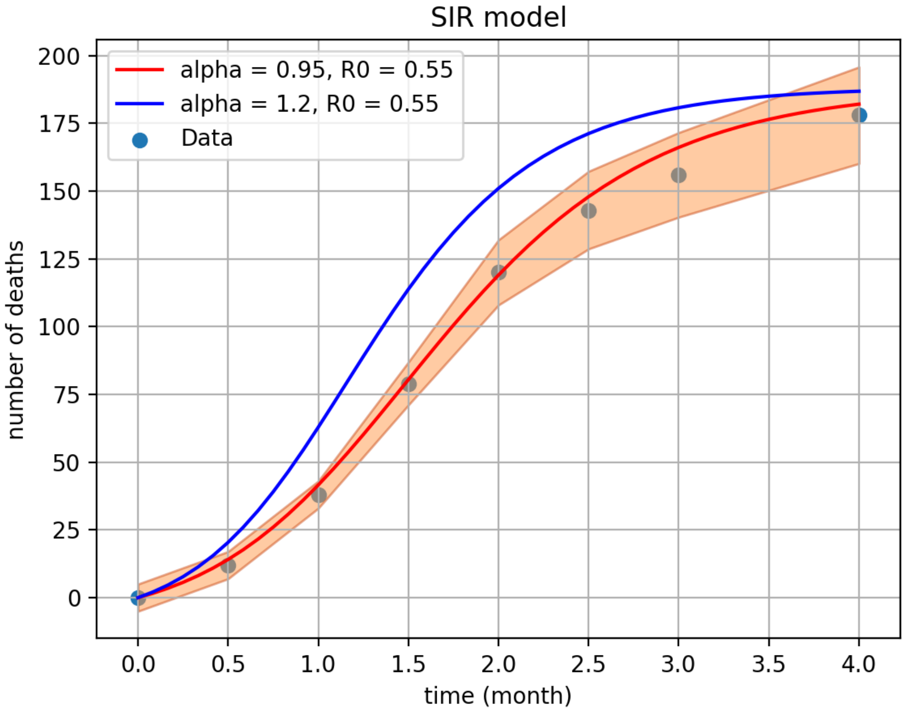

Figure 6: The error band illustrates the target set $U = [R_1 - \varepsilon_1, R_1 + \varepsilon_1] \times \cdots \times [R_7 - \varepsilon_7, R_7 + \varepsilon_7]$. The red and the blue curves represent outputs that correspond to values of $(\mathscr{R}_0, \alpha)$ that lie inside and outside the pre-image, respectively.

epidemic by the following system of ordinary differential equations

$$\begin{aligned}
&\frac{dS}{dt}(t) = -\beta S(t)I(t), \\
&\frac{dI}{dt}(t) = \beta S(t)I(t) - \alpha I(t), \text{ and} \\
&\frac{dR}{dt}(t) = \alpha I(t),
\end{aligned} \tag{4}$$

where $\beta > 0$ is the infection rate and $\alpha > 0$ is the removal rate. Note that the total population $N = S(t) + I(t) + R(t)$ remains constant under this model. An important quantity of an outbreak is the basic reproduction number $\mathscr{R}_0 = \beta N/\alpha$, which indicates whether the outbreak can become an epidemic ($\mathscr{R}_0 > 1$) or not ($\mathscr{R}_0 < 1$). Similar to [29], we opt to use $(\mathscr{R}_0, \alpha)$ as our parameters instead of (α, β). In this case, the system (4) becomes

$$\begin{aligned}
&\frac{dS}{dt}(t) = -\frac{\mathscr{R}_0\alpha}{N} S(t)I(t), \\
&\frac{dI}{dt}(t) = \frac{\mathscr{R}_0\alpha}{N} S(t)I(t) - \alpha I(t), \text{ and} \\
&\frac{dR}{dt}(t) = \alpha I(t),
\end{aligned} \tag{5}$$

In practice, it is impossible for the deterministic SIR model to fit the data perfectly. That is, there is no value for $(\mathscr{R}_0, \alpha)$ such that the solution of (5) aligns perfectly with the observations. Hence, we define the set V of possible values for $(\mathscr{R}_0, \alpha)$ given the number of deaths $(R_1, R_2, \ldots, R_7)$ at time $(t_1, t_2, \ldots, t_7)$ as follows:

$$V = \left\{ (\mathscr{R}_0, \alpha) : |R_{\mathscr{R}_0,\alpha}(t_i) - R_i| \le \max\left(5, \frac{R_i}{10}\right) \right\}$$

where $R_{\mathscr{R}_0,\alpha}(t_i)$ is R-component of the solution of (5) evaluated at time t_i with parameter values $(\mathscr{R}_0, \alpha)$ and initial value $(S_0, I_0, R_0) = (254, 7, 0)$. In other words, if we define the forward function

$$F(\mathscr{R}_0, \alpha) = [R_{\mathscr{R}_0,\alpha}(t_1), R_{\mathscr{R}_0,\alpha}(t_2), \ldots, R_{\mathscr{R}_0,\alpha}(t_7)],$$

then

$$V = F^{-1}\left([R_1 - \varepsilon_1, R_1 + \varepsilon_1] \times \cdots \times [R_7 - \varepsilon_7, R_7 + \varepsilon_7]\right)$$

where

$$\varepsilon_i = \max\left(5, \frac{R_i}{10}\right)$$

for $i = 1, 2, \ldots, 7$. That is we allow the SIR model to differ from the data about 10%. Figure 6 depicts an illustration of the target set U and examples of points from the inside and the outside of the pre-image.

We apply our proposed method to find the pre-image V on the state space $\Omega = [1.5, 2] \times [1.8, 3.7]$. The state space Ω is chosen according to the previous study of this data set in [29]. Here, we only use the MLP classifier since it has been demonstrated as the best method for two-dimensional problems. Our method achieves a high prediction accuracy (98.75%) and recovers the shape of the pre-image quite well even though the ground truth has a fairly irregular shape with multiple points of non-differentiability (see Figure 7).

5. Discussions and conclusions

In this work, we have proposed a novel active learning framework for the problem of set inversion and developed an efficient method to solve it. Our simulations show that the proposed method outperforms the current state-of-the-art program VSIVIA in low-dimensional problems, and still works well in high dimensions while VSIVIA fails to do so. The prediction time of our approach is remarkably faster than VSIVIA, which is important in scenarios where prediction needs to be made quickly with limited computational power.

We observe that the performance of the proposed algorithm does not depend strongly on the choice of sampling schemes to initialize the algorithm. On the

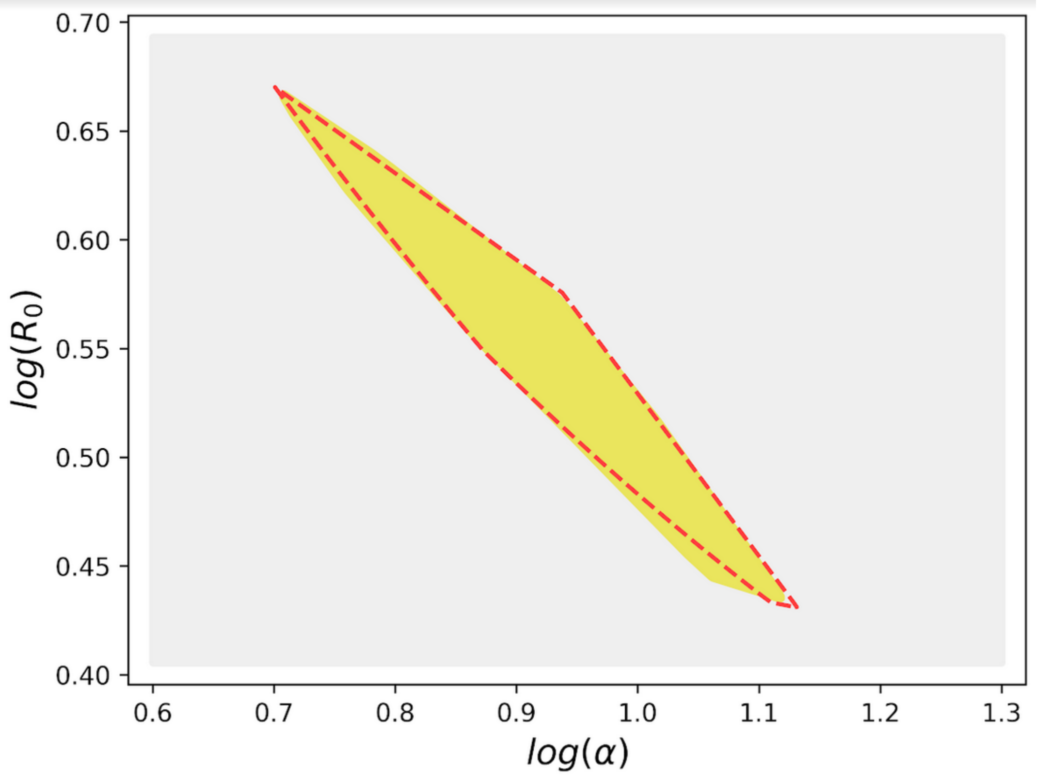

Figure 7: Reconstruction of V, the set of possible values for $(\mathscr{R}_0, \alpha)$ given the number of deaths in the plague of Eyam. The red dashed line expresses the true boundary of V, while the yellow region is the reconstructed set by our algorithm.

other hand, MLP and SVM are the better algorithms in prediction among the four classifiers we consider in this paper. The strength of SVM is having fast training time and scaling well with the dimension of the problem, while MLP can capture important geometric features of the pre-image. This provides a guideline on which classifier to choose for a specific problem in practice. For example, when applying our methods to analyze the plague in Eyam, we intentionally pick the MLP classifier because we want to retain the shape of the set of interest.

In terms of practical applicability, our proposed approach can open a new horizon for different applications of set inversion in sciences and engineerings including non-convex optimization problems [30, 31, 32], nonlinear parameter set estimation [33], localization and characterization of stability domains of dynamical systems [34, 35, 36], fault detection and identification [37], set-membership experimental design of biological systems [38], and behaviour discrimination of enzymatic reaction networks [39]. Up until this point, these applications are limited to low dimensional settings and our method provides a solution to address this bottleneck.

We note that our method requires the users to input the state space Ω manually. The choice of Ω is very important because if the state space is too large (in comparison to the pre-image), the number of samples required to explore the pre-image might increase significantly. In the future, we want to develop an automatic procedure to choose the state space Ω in an adaptive manner such that the size of the pre-image is comparable to that of the state space.

Another limitation of our approach is that the forward function can be evaluated without error. However, in a lot of applications in sciences and engineerings, sometimes it might be difficult to check if a specific combination of parameters belongs to a region of interest due to noisy measurements and model uncertainty. It is an interesting direction for future work to build an algorithm for set inversion that takes into account such uncertainty.

Finally, we have provided a new tool for studying an infectious disease epidemic. The analysis using the dataset of the 17-century plague in Eyam has confirmed the feasibility of this approach. A possible next step is to compare the performance of set inversion with traditional statistical techniques such as Maximum likelihood estimators and Bayesian methods.

Acknowledgments

LSTH was supported by startup funds from Dalhousie University, the Canada Research Chairs program, the NSERC Discovery Grant RGPIN-2018-05447, and the NSERC Discovery Launch Supplement DGECR-2018-00181. BTN was supported by the university funding T2017-05 from University of Science, Vietnam.

References

[1] V. Dinh, A. E. Rundell, G. T. Buzzard, Effective sampling schemes for behavior discrimination in nonlinear systems, International Journal for Uncertainty Quantification 4 (2014).

[2] S. Lesecq, A. Barraud, T. Dinh, Numerical accurate computations for ellipsoidal state bounding, Proceedings of MED (2003).

[3] M. Milanese, A. Vicino, Estimation theory for nonlinear models and set membership uncertainty, Automatica 27 (1991) 403–408.

[4] T. Alamo, J. M. Bravo, E. F. Camacho, Guaranteed state estimation by zonotopes, Automatica 41 (2005) 1035–1043.

[5] L. Jaulin, E. Walter, Set inversion via interval analysis for nonlinear bounded-error estimation, Automatica 29 (1993) 1053–1064.

[6] B. T. Nguyen, D. M. Nguyen, L. S. T. Ho, V. C. Dinh, OASIS: an active framework for set inversion, in: New Trends in Intelligent Software Methodologies, Tools and Techniques - Proceedings of the 17th International Conference SoMeT_18, Granada, Spain, 26-28 September 2018, 2018, pp. 883–895.

[7] C. Cortes, V. Vapnik, Support-vector networks, Machine learning 20 (1995) 273–297.

[8] N. S. Altman, An introduction to kernel and nearest-neighbor nonparametric regression, The American Statistician 46 (1992) 175–185.

[9] T. K. Ho, Random decision forests, in: Proceedings of 3rd international conference on document analysis and recognition, volume 1, IEEE, 1995, pp. 278–282.

[10] J. Friedman, T. Hastie, R. Tibshirani, The elements of statistical learning, volume 1, Springer series in statistics New York, 2001.

[11] M. D. McKay, R. J. Beckman, W. J. Conover, Comparison of three methods for selecting values of input variables in the analysis of output from a computer code, Technometrics 21 (1979) 239–245.

[12] S. I. Meerovich, On the distribution of points in a cube and the approximate evaluation of integrals, Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki 7 (1967) 784–802.

[13] D. M. H. Nguyen, H. T. Vu, H. Q. Ung, B. T. Nguyen, 3d-brain segmentation using deep neural network and gaussian mixture model, in: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), 2017, pp. 815–824.

[14] N. V. Cuong, V. Dinh, L. S. T. Ho, Mel-frequency cepstral coefficients for eye movement identification, in: 2012 IEEE 24th International Conference on Tools with Artificial Intelligence, volume 1, 2012, pp. 253–260. doi:10.1109/ICTAI.2012.42.

[15] P. Y. Simard, D. Steinkraus, J. C. Platt, Best practices for convolutional neural networks applied to visual document analysis, in: Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings., 2003, pp. 958–963. doi:10.1109/ICDAR.2003.1227801.

[16] H. K. Cao, D. T. Ly, D. M. Nguyen, B. T. Nguyen, Automatically generate hymns using variational attention models, in: H. Lu, H. Tang, Z. Wang (Eds.), Advances in Neural Networks – ISNN 2019, Springer International Publishing, Cham, 2019, pp. 317–327.

[17] V. C. Tran, N. T. Nguyen, H. Fujita, D. T. Hoang, D. Hwang, A combination of active learning and self-learning for named entity recognition on twitter using conditional random fields, Knowledge-Based Systems 132 (2017) 179 – 187. URL: http://www.sciencedirect.com/science/article/pii/

S0950705117303040. doi:https://doi.org/10.1016/j.knosys.2017.06.023.

[18] F. J. García-Peñalvo, J. Cruz-Benito, M. Martín-González, A. Vázquez-Ingelmo, J. C. Sánchez-Prieto, R. Therón, Proposing a machine learning approach to analyze and predict employment and its factors, International Journal of Interactive Multimedia and Artificial Intelligence 5 (2018) 39–45. URL: http://www.ijimai.org/journal/sites/default/files/files/2018/02/ijimai_5_2_5_pdf_12552.pdf. doi:10.9781/ijimai.2018.02.002.

[19] B. T. Nguyen, D.-T. Dang-Nguyen, T. X. Dang, T. Phat, C. Gurrin, A deep learning based food recognition system for lifelog images, in: Proceedings of the 7th International Conference on Pattern Recognition Applications and Methods - Volume 1: INDEED,, INSTICC, SciTePress, 2018, pp. 657–664. doi:10.5220/0006749006570664.

[20] S. Dasgupta, The two faces of active learning, Lecture Notes in Computer Science 5808 (2009) 35.

[21] J. Nocedal, S. Wright, Sequential quadratic programming, Springer, 2006.

[22] V. Nair, G. E. Hinton, Rectified linear units improve restricted boltzmann machines, in: Proceedings of the 27th international conference on machine learning (ICML-10), 2010, pp. 807–814.

[23] P. Herrero, P. Georgiou, C. Toumazou, B. Delaunay, L. Jaulin, An efficient implementation of sivia algorithm in a high-level numerical programming language, Reliable Computing (2012) 239–251.

[24] Y. Sun, X. Wang, X. Tang, Deep learning face representation from predicting 10,000 classes, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1891–1898. doi:10.1109/CVPR.2014.244.

[25] A. Bemporad, M. Morari, V. Dua, E. N. Pistikopoulos, The explicit linear quadratic regulator for constrained systems, Automatica 38 (2002) 3–20.

[26] A. Chakrabarty, V. Dinh, M. J. Corless, A. E. Rundell, S. H. Żak, G. T. Buzzard, Support vector machine informed explicit nonlinear model predictive control using low-discrepancy sequences, IEEE Transactions on Automatic Control 62 (2017) 135–148.

[27] G. Raggett, A stochastic model of the Eyam plague, Journal of Applied Statistics 9 (1982) 212–225.

[28] L. S. T. Ho, F. W. Crawford, M. A. Suchard, et al., Direct likelihood-based inference for discretely observed stochastic compartmental models of infectious disease, The Annals of Applied Statistics 12 (2018) 1993–2021.

[29] L. S. T. Ho, J. Xu, F. W. Crawford, V. N. Minin, M. A. Suchard, Birth/birth-death processes and their computable transition probabilities with biological applications, Journal of Mathematical Biology 76 (2018) 911–944.

[30] R. Hammer, M. Hocks, U. Kulisch, D. Ratz, C++ Toolbox for Verified Computing, Springer, 1995.

[31] R. Kearfott, Rigorous Global Search: Continuous Problems, Kluwer, Dordrecht, the Netherlands, 1996.

[32] R. Moore, Methods and Applications of Interval Analysis, SIAM, 1979.

[33] M. Kieffer, E. Walter, Interval analysis for guaranteed nonlinear parameter estimation, in: MODA 5 - Advances in Model-Oriented Data Analysis and Experimental Design, Springer, 1998, pp. 115–125.

[34] E. Colle, S. Galerne, Mobile robot localization by multiangulation using set inversion, Robotics and Autonomous Systems (2012).

[35] E. Walter, L. Jaulin, Guaranteed characterization of stability domains via set inversion, IEEE Transactions on Automatic Control 39 (1994) 886–889.

[36] V. Drevelle, P. Bonnifait, A set-membership approach for high integrity height-aided satellite positioning, GPS solutions 15 (2011) 357–368.

[37] C. Jauberthie, N. Verdière, L. Travé-Massuyès, Fault detection and identification relying on set-membership identifiability, in: Fault Detection, Supervision and Safety of Technical Processes, volume 8, 2012, pp. 1262–1267.

[38] S. W. Marvel, C. M. Williams, Set membership experimental design for biological systems, BMC Systems Biology 6 (2012) 21.

[39] A. Donzé, B. Krogh, A. Rajhans, Parameter synthesis for hybrid systems with an application to simulink models, in: Hybrid Systems: Computation and Control, Springer, 2009, pp. 165–179.