MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Data-Driven Estimation of Reachable and Invariant Sets for Unmodeled Systems via Active Learning

Chakrabarty, A.; Raghunathan, A.U.; Di Cairano, S.; Danielson, C.

TR2018-187 January 10, 2019

Abstract

Ensuring control performance with state and input constraints is facilitated by the understanding of reachable and invariant sets. While exploiting dynamical models have provided many set-based algorithms for constructing these sets, set-based methods typically do not scale well, or rely heavily on model accuracy or structure. In contrast, it is relatively simple to generate state trajectories in a data-driven manner by numerically simulating complex systems from initial conditions sampled from within an admissible state space, even if the underlying dynamics are completely unknown. These samples can then be leveraged for reachable/invariant set estimation via machine learning, although the learning performance is strongly linked to the sampling pattern. In this paper, active learning is employed to intelligently select batches of samples that are most informative and least redundant to previously labeled samples via submodular maximization. Selective sampling reduces the number of numerical simulations required for constructing the invariant set estimator, thereby enhancing scalability to higherdimensional state spaces. The potential of the proposed framework is illustrated via a numerical example.

IEEE Conference on Decision and Control (CDC)

Data-Driven Estimation of Reachable and Invariant Sets for Unmodeled Systems via Active Learning

Ankush Chakrabarty†, Arvind Raghunathan, Stefano Di Cairano, Claus Danielson*

***Abstract*—Ensuring control performance with state and input constraints is facilitated by the understanding of reachable and invariant sets. While exploiting dynamical models have provided many set-based algorithms for constructing these sets, set-based methods typically do not scale well, or rely heavily on model accuracy or structure. In contrast, it is relatively simple to generate state trajectories in a data-driven manner by numerically simulating complex systems from initial conditions sampled from within an admissible state space, even if the underlying dynamics are completely unknown. These samples can then be leveraged for reachable/invariant set estimation via machine learning, although the learning performance is strongly linked to the sampling pattern. In this paper, active learning is employed to intelligently select batches of samples that are most informative and least redundant to previously labeled samples via submodular maximization. Selective sampling reduces the number of numerical simulations required for constructing the invariant set estimator, thereby enhancing scalability to higher-dimensional state spaces. The potential of the proposed framework is illustrated via a numerical example.**

***Index Terms*—Machine learning; submodular maximization; imbalanced learning; design of experiments; domain of attraction; nonlinear systems.**

I. Introduction

Estimating domains of attraction, reachable, and invariant sets for nonlinear dynamical systems in a computationally efficient manner is a fundamental challenge in stability analysis [1] and constrained control, with particular applications in model predictive control [2], [3]. Convex programming formulations such as sum-of-squares (SOS) and semi-definite programming (SDP) have paved the way for efficacious construction of such sets using quadratic and polynomial Lyapunov functions for a sub-class of nonlinear dynamical systems [4]–[6].

For systems that do not conform to structured representations or when underlying models are not readily available, numerical simulations offer a fast and scalable alternative to collecting reachability or invariance information. That is, one can obtain samples on the admissible state space, generate system trajectories considering these samples as initial conditions, and, based on the state trajectory, indicate whether the sample lies in the reachable/invariant set or not. Such a data-driven approach to obtaining reachability/invariance information can then be posed as a binary classification problem. Thus, supervised learning methods can be leveraged to obtain estimates of reachable/invariant regions with high computational efficacy [7]–[10]. Furthermore, approximation properties of advanced supervised learning algorithms like support vector machines and neural networks enhance generalizability to large classes of dynamical systems, while a data-driven approach eliminates the need to exploit specific system structure. However, a potential weakness of this method is that the estimate of the invariant set relies strongly on the sampling pattern employed, and convergence guarantees (usually in a Hausdorff sense) are asymptotic: that is, a large number of samples may be required to yield good estimates of the set of interest. To mitigate this issue, we adopt an active learning formalism for selecting the most informative samples (in some sense) instead of sampling arbitrarily.

Active learning (also referred to as 'optimal experiment design' or 'directed sampling') is a fundamental tenet in artificial intelligence that allows a learner to select the most informative sample(s) from a pool of unlabeled samples based on its current knowledge of the learning problem, thereby restricting the training set size without degrading performance. In this paper, we restrict our attention to 'pool-based batch selection', which means that a batch of subsequent samples will be queried for labeling by an oracle—in our case, numerical simulations of the system under consideration—from a pool of unlabeled samples. This generally prevents prohibitive numerical simulations typical in single-instance selection methods. While batch selection can be performed based on single objectives such as distances from margins [11] or choosing samples that exhibit maximum uncertainty [12], recent methods have shifted towards a multi-criterion framework where samples are selected to maximize informativeness while minimizing redundancy [13]–[15]. Unfortunately, the consequent learning problems are usually NP-hard, and these methods seldom guarantee solution quality. In this paper, we propose an active learning method that poses the batch selection procedure as a submodular maximization problem for which near-optimal solutions can be obtained using a polynomial time greedy approach [16]. Previous utilization of submodular functions for subset selection can be found in [17], [18], although these approaches are either not multi-criterion frameworks, requires specific classes of learners, or may prove computationally more taxing due to inversion of Fisher information matrices.

Specific **contributions** of this paper are as follows: (i) we introduce a batch-mode active learning procedure to select

†Corresponding author: Ankush Chakrabarty, Phone: +1 (617) 758-6175.
*All authors {chakrabarty, raghunathan, dicairano, danielson}@merl.com are affiliated with Mitsubishi Electric Research Laboratories, Cambridge, MA, USA.

relevant samples for learning reachable/invariant set boundaries in a sampling-based, data-driven manner even when a dynamical model representation is unknown; (ii) we pose the multi-criterion optimization procedure within the active learning method as a submodular maximization problem for which a greedy approach is near-optimal and operates in polynomial time; (iii) we demonstrate the flexibility of this method with multiple classifiers and illustrate its potential in tackling highly imbalanced learning problems without exorbitant sampling.

II. Motivation

Although we present our discussion using a discrete-time dynamical system, the proposed method is identical for continuous-time systems. Consider a generic nonlinear system

$$x_{t+1} = f(x_t, u_t), \tag{1}$$

where $t \in \mathbb{N}$ denotes the time-step, $x_t \in \mathbb{X} \subset \mathbb{R}^{n_x}$, $u_t \in \mathbb{U} \subset \mathbb{R}^{n_u}$ are the state and input constraints at time-step t. We assume that the constraint sets $\mathbb{X}, \mathbb{U}$ are compact, convex, and contain the origin in their relative interiors. We further assume that the origin is a unique equilibrium state-input pair for the dynamics (1); that is, $f(0,0) = 0$. Other than this, we impose no structure on the nonlinearity $f(\cdot,\cdot)$; in fact *f does not need to be known*.

We begin with the following standard definitions.

Definition 1 (T-step Reachable Set)**.** *Let T be a positive integer and $\Omega \subset \mathbb{X}$. A non-empty set $\mathcal{R}_T(\Omega)$ within the admissible state space $\mathbb{X}$ is a T-step reachable set for the system* (1) *if it contains all initial states $x_0 \in \mathbb{X}$ that can be driven to a state $x_T \in \Omega$ under the closed-loop dynamics* (1) *within T time-steps.*

Definition 2 (Positive Invariant Set)**.** *A non-empty set $\Omega(\mathbb{X})$ within the admissible state space $\mathbb{X}$ is a positive invariant set for the closed-loop system* (1) *under a control law $u = \kappa(x)$ if, for every initial condition $x_0 \in \Omega$, all subsequent states $x_t \in \Omega,\ \forall t > 0$.*

The satisfaction of state and input constraints is imperative to ensuring correct and reliable operation of the closed-loop system in safety-critical applications, making the estimation of reachable and invariant sets a cogent problem in control system design. We begin with the following critical assumption.

Assumption 1. *There exists a reliable (albeit perhaps expensive to query) oracle $\mathcal{O} : \mathbb{X} \to \{-1, 1\}$ that returns whether $(+1)$ or not (-1) an initial condition $x_0 \in \mathbb{X}$ lies in a reachable/invariant set of the system* (1)*.*

Example 1. Consider the ε norm-ball $\Omega = \{x \in \mathbb{X} : \|x\|_2 \leq \varepsilon\}$ and suppose $u_t = \kappa(x_t)$ is a controller that stabilizes the system (1) to the origin. Then an oracle that provides information about the reachable set $\mathcal{R}(\Omega)$ is given by

$$\mathcal{O}_1(x_0) = \begin{cases} +1, & \text{if } x_1 \in \Omega \\ -1, & \text{otherwise.} \end{cases}$$

■

Example 2. Consider an oracle that provides feasibility information (this is a positive invariant set) for a nonlinear model predictive controller (NMPC). Let $N \in \mathbb{N}$ be the prediction horizon of an NMPC. Then, at time-step t, the following cost function

$$J := F(\bar{x}_{N|t}) + \sum_{k=0}^{N-1} \ell(\bar{x}_{k|t}, \bar{u}_{k|t}), \tag{2a}$$

is minimized, subject to the state and input constraints

$$\bar{u}_{k|t} \in \mathbb{U}, \quad \bar{x}_{k|t} \in \mathbb{X}, \quad \forall\, k = \{0, 1, \ldots, N-1\}, \tag{2b}$$

$$\bar{x}_{k+1|t} = f(\bar{x}_{k|t}, \bar{u}_{k|t}),\ \forall\, k = \{0, 1, \ldots, N-1\}, \tag{2c}$$

$$\bar{x}_{0|t} = x(t),\ \bar{x}_{N|t} \in \mathbb{X}_f. \tag{2d}$$

Here, ℓ and F are design functions representing the stage cost and terminal penalty, respectively, and the set $\mathbb{X}_f$ is an appropriately designed terminal region. If one designs a local controller $\kappa_f : \mathbb{R}^{n_x} \to \mathbb{R}^{n_u}$ such that: (i) the control action $\kappa_f(0) = 0$; (ii) the control actions $\kappa_f(x) \in \mathbb{U}$ for all $x \in \mathbb{X}_f$; and, (iii) the closed-loop trajectories of

$$x_{t+1} = f(x_t, \kappa_f(x_t)) \tag{3}$$

reside in $\mathbb{X}_f$ for any initial condition $x_0 \in \mathbb{X}_f$ with an appropriate F and $\mathbb{X}_f$, then the origin is stabilizable by the NMPC with controls obtained by solving (2) in a receding horizon fashion [19].

One can employ an oracle of the form

$$\mathcal{O}_3(x_0) = \begin{cases} +1, & \text{if (2) has a feasible solution} \\ -1, & \text{otherwise.} \end{cases}$$

to estimate the feasible region of this NMPC. ■

Note that in both the above examples, the presence of data generated by the oracles obviates the need for knowledge of the underlying nonlinearity f. In other words, a critical advantage of Assumption 1 is that we can estimate the reachable/invariant set without access to the model structure (1). This is imperative for many applications where feasibility is determined experimentally or from archival records such as in smart buildings [20] or systems biology [21]. One can also use this method to validate 'black-box' controllers (where the controller's behaviour is completely unknown, such as for control with deep neural networks): in such cases, sampling and reachable/invariant set estimation of the closed-loop system under black-box control can generate a degree of trust in the automation framework.

Our *objective* in this paper is to estimate the reachable/invariant set of complex, possibly unmodeled, systems by employing active learning paradigms to select informative samples for training a supervised learner. Concretely, we leverage machine intelligence to iteratively select initial conditions within $\mathbb{X}$ for which the oracle of Assumption 1 should be queried for labeling. With these labeled samples, our goal is to characterize inner approximations of the reachable/invariant set for consequent use in on-line control. Due to the iterative framework of active learning algorithms, one expects gradual

improvement of the inner approximation with more samples, which is an advantage of this framework over methods that generate extremely conservative estimates of the inner approximation either due to less flexbility in the choice of fitting functions, or due to lossy convex relaxations.

III. Proposed Active Learning Algorithm

The major advantage of AL methods over traditional supervised learning is the iterative improvement of learning performance by systematically utilizing prior learners. The essential steps in typical AL algorithms are described in Algorithm 1. The AL setting is as follows. One is usually

Algorithm 1 Typical Active Learning (AL) Algorithm

Require: Initial set of labeled samples, $\mathcal{L}_0$
Require: Set of unlabeled samples, $\mathcal{S}$
Require: Final number of labeled samples, N_s
Require: Batch size, B
Require: Iteration counter, $k \leftarrow 0$
Require: Oracle, $\mathcal{O}$
while $|\mathcal{L}_k| < N_s$ **do**
  $\psi_k \leftarrow$ classifier constructed using $\mathcal{L}_k$
  Find B best (in some sense) samples from $\mathcal{S}$
  $\mathcal{L}'_k \leftarrow$ label B samples using oracle $\mathcal{O}$
  $\mathcal{L}_{k+1} \leftarrow$ append$[\mathcal{L}_k, \mathcal{L}'_k]$
  $k \leftarrow k+1$
end while
$\psi \leftarrow$ final classifier trained on $\mathcal{L}_{k+1}$

provided with a set of labeled samples denoted by $\mathcal{L}_0$ and a (much larger) set of unlabeled samples $\mathcal{S}$. The objective of AL algorithms is to select (sample-wise or batch-wise) the most informative (in some sense) $\mathcal{L}'_k \subset \mathcal{S}$ at each iteration k based on the classifier designed on the labeled data currently available (that is, $\mathcal{L}_{k-1}$). Subsequently, an oracle is queried to generate labels for the newly selected set of samples $\mathcal{L}'_k$. These labels are appended to $\mathcal{L}_{k-1}$ to form the new labeled set $\mathcal{L}_k$. The process iterates until some termination criterion is satisfied, such as a user-defined size of the labeled set.

The steps of our proposed algorithm is outlined below.

Step 1: Constructing $\mathcal{L}_0$ and $\mathcal{S}$: To begin, we need a set of labeled samples $\mathcal{L}_0$ and a set of unlabeled samples $\mathcal{S}$. The set of labeled samples may be provided from prior experiments, or using the oracle $\mathcal{O}$ to label an arbitrary subset of $\mathcal{S}$ until a minimum number of labeled samples are obtained satisfying a minimum class-imbalance ratio[1] $\nu \in (0,1)$: this ensures that a non-trivial classifier exists.

The unlabeled set $\mathcal{S}$ is generated by sampling arbitrarily on $\mathbb{X}$: samples that are well-distributed on $\mathbb{X}$ will yield practical advantages [10]. The number of samples collected in $\mathcal{S}$ should be large because (i) they are unlabeled so the only computational overhead (offline) is in generating the samples, which is usually inexpensive; and, (ii) good coverage of $\mathbb{X}$ will ensure that there exist informative samples in close proximity to the boundary of the true reachable/invariant set that should be chosen by the AL algorithm.

[1]In binary classification, a class-imbalance ratio of ν implies that $\nu \times N_L$ samples within $\mathcal{L}_0$ are labeled as category '+1' and the rest are labeled '-1'.

Algorithm 2 Procedure for constructing $\mathcal{L}_0$

Require: Minimum size of initial labeled set, $N_L \in \mathbb{N}$
Require: Minimum class-imbalance ratio $\nu \in (0,1)$
$N_{L+} \leftarrow \text{round}(\nu \times N_L)$, minimum number of feasible samples in $\mathcal{L}_0$
$k_+, k_- \leftarrow 0$
while $k_+ + k_- < N_L$ **do**
  $x \leftarrow$ sample drawn from $\mathbb{X}$
  $l \leftarrow \mathcal{O}(x)$
  if $l = 1$ and $k_+ < N_{L+}$ **then**
    $\mathcal{L}_0 \leftarrow [\mathcal{L}_0, x]$
    $k_+ \leftarrow k_+ + 1$
  end if
  if $l = -1$ and $k_- < N_L - N_{L+}$ **then**
    $\mathcal{L}_0 \leftarrow [\mathcal{L}_0, x]$
    $k_- \leftarrow k_- + 1$
  end if
end while

Remark 1. For reachable/invariant sets that are small compared to the measure of $\mathbb{X}$, the set $\mathcal{L}_0$ may exhibit a very low class-imbalance ratio if labeled samples are generated randomly. To increase initial class-imbalance ratio ν for estimating the reachable set $\mathcal{R}(\Omega)$, one can then sample more densely around Ω to increase the likelihood of finding samples that lie in $\mathcal{R}(\Omega)$. An alternative procedure to increase ν (that is applicable to both reachable and invariant set estimation) involves a slight modification to the oracle: namely that it returns the state sequence generated during simulation, along with the label for feasible samples. That is,

$$\tilde{\mathcal{O}} = \begin{cases} +1, \{x_k\}_{k=1}^{T} & \text{if } x_0 \text{ is feasible} \\ -1, & \text{otherwise.} \end{cases} \tag{4}$$

Then, by Definitions 1 and 2, each x_0 that is labeled to lie within the T-step reachable/invariant set will generate states x_t for $0 < t \leq T$ that are all within the T-step reachable/invariant set, respectively. Thus, $\{x_k\}_{k=0}^{T}$ can be appended to the feasible class, thereby raising the value of ν. □

Remark 2. In the case of NMPC, each feasible initial state produces N feasible states, since all intermediate open-loop states $\{\bar{x}_k\}_{k=1}^{N-1}$ are feasible by the constraints in (2), along with the invariance of $\mathbb{X}_f$. Analogous to the above discussion, one can append the labeled set with these sequences if one has access to an oracle of the form (4). □

Step 2: Posing the learning problem: With $\mathcal{L}_0$ and $\mathcal{S}$ constructed, we train a classifier ψ_0 on $\mathcal{L}_0$. At each subsequent iteration, the objective is to use the prior classifier ψ_k to determine the B most informative samples within the unlabeled set $\mathcal{S}$ that simultaneously lowers redundancy with respect to the current labeled set $\mathcal{L}_k$. We pose this informativeness-redundancy trade-off idea as a cardinality constrained submodular maximization problem. This ensures that a greedy approach will provide near-optimality, as we shall see in the ensuing discussion.

Let

$$\zeta(S) \triangleq \mathcal{E}(S) + \mathcal{D}(S), \tag{5}$$

where

$$\mathcal{E}(S) = -\sum_{s\in S}\sum_{y\in\{-1,1\}} p(y|s,\psi_k)\log_2|p(y|s,\psi_k)| \quad (6)$$

denotes the Shannon entropy of the distribution $p(y|s,\psi_k)$ computed using the prior classifier ψ_k.

The function $\mathcal{D}(S)$ is defined by

$$\mathcal{D}(S) = \sum_{i=1}^{|\mathcal{L}_k|}\max_{s\in S} D_{is}^k, \quad (7)$$

where $D^k \in \mathbb{R}^{|\mathcal{L}_k|\times|S|}$ is a non-negative matrix whose ith row $D_i^k \geq 0$ contains the amount of mutual information between the ith labeled sample and all the unlabeled ones in S. This mutual information can be estimated by computing the relative distance between the distributions of the ith and sth samples conditional upon the classifier ψ_k. For example, one can employ the Kullback-Liebler (KL) divergence metric

$$D_{is}^k = -\sum_{y\in\{-1,1\}} p_y^i \log_2\left|\frac{p_y^s}{p_y^i}\right|.$$

To find the B best samples in $\mathcal{S}$, we solve the following optimization problem:

$$\mathcal{L}_k' = \underset{S\subseteq\mathcal{S}:|S|\leq B}{\arg\max}\ \zeta(S). \quad (8)$$

Intuitively, the problem (8) tries to maximize the quantity $\mathcal{E}(\cdot)$ that ascertains the uncertainty embedded in an unlabeled sample, while choosing samples within $\mathcal{S}$ that possess relatively low overlap with previously labeled samples, estimated via the cost $\mathcal{D}(\cdot)$. It is well known that this problem is NP-hard [22] for which no polynomial-time algorithm exists with an approximation factor better than $(1-1/e)$ (unless P=NP). In the sequel, we will show that careful selection of $\mathcal{E}$ and $\mathcal{D}$ can result in a near-optimal, polynomial time, greedy solution.

Remark 3. The cardinality of $\mathcal{S}$ is generally large to ensure well distributed samples throughout $\mathbb{X}$. For practical implementation, one may pre-select a more wieldy subset $\mathcal{S}_k \subset \mathcal{S}$ by considering the $q > B$ most informative (based on their entropy conditional upon the $(k-1)$ th classifier) samples within $\mathcal{S}$. The active samples drawn in the k th iteration will then be restricted to samples within $\mathcal{S}_k$.

Step 3: Solving (8) *to obtain the next batch:* We begin with the following definitions [16].

Definition 3. *Consider a set function $\zeta: 2^W \to \mathbb{R}$ that maps subsets of a finite set W to $\mathbb{R}$. The function ζ is:* ***normalized****: if $\zeta(\emptyset) = 0$;* ***monotone****: if $\zeta(W_1) \leq \zeta(W_2)$ for any $W_1 \subseteq W_2 \subseteq W$;* ***submodular****: if, for every $W_1 \subseteq W_2 \subseteq W$, and $w \in W \setminus W_2$, the inequality $\zeta(W_1\cup\{w\}) - \zeta(W_1) \geq \zeta(W_2\cup\{w\}) - \zeta(W_2)$ is satisfied.*

The problem (8) can be solved approximately using a greedy algorithm. The greedy algorithm involves iteratively selecting the sample $s \in \mathcal{S}$ that increases ζ the most until the number of samples selected equals the batch size B. The following theorem states that selecting the functions $\mathcal{E}$ and $\mathcal{D}$ as we have in (6) and (7) enables a greedy solution that is near-optimal in polynomial time.

Theorem 1. *Recall B is the batch size, and the set functions $\mathcal{E}$ and $\mathcal{D}$ are defined in* (6) *and* (7)*, respectively. Any solution $\mathcal{L}'_{k,\mathsf{G}}$ obtained by solving* (8) *using a greedy algorithm is no worse than a constant fraction $(1-1/e)$ away from the optimal value $\zeta^\star$ for* (8)*; that is, $\frac{\zeta(\mathcal{L}'_{k,\mathsf{G}})}{\zeta^\star} \geq 1 - \left(\frac{B-1}{B}\right)^B \geq \left(1-\frac{1}{e}\right)$.*

Remark 4 (Computational Complexity)**.** Since we are selecting B samples in a batch, for the kth active sample, to evaluate the incremental cost, we need $k(|S| - (k-1))$ operations. To find the maximum over these increments, we require $|S| - (k-1)$ operations. Thus, at kth iteration of our greedy algorithm, we have a total of $(|S|-k+1)\times(k+1)$ operations. Hence, the complexity can be bounded by $\mathbf{O}(|S|B^2)$.

Step 4: Computing the reachable/invariant set boundary: Steps 1–3 are typically operated until a termination criterion is attained. Common termination criteria include: size of the final labeled set or number of iterations. The post-termination classifier, denoted ψ_∞, will not necessarily be an inner approximation of the reachable/invariant set. To eliminate this issue, one can use the method proposed in [10] to select sub-level sets of the decision function ψ_∞ until no infeasible point is classified as feasible. Following arguments made in [10], if the boundary of the true reachable/invariant set is the zero level set of a continuous function, any classifier that can generate decision functions that are dense in the space of continuous functions can compute arbitrarily good approximations of the region of interest.

IV. Numerical Results

We illustrate our proposed approach on the following nonlinear discrete-time system studied in [23], with $\mathbb{U} = \{u \in \mathbb{R} : |u| \leq 2\}$. We define the admissible state space $\mathbb{X} = \{x \in \mathbb{R}^2 : \|x\|_\infty \leq 4\}$. The objective of expanding the state space to a region much larger than that considered in [23] is to demonstrate the utility of active learning when random sampling will result in a heavily imbalanced dataset.

The oracle in this case is a nonlinear MPC with a prediction and control horizon of 8, akin to $\mathcal{O}_3$ in Example 2. The cost function J described in (2) is parameterized by $F(x) = x^\top Px$, with $P = \left[\begin{smallmatrix} 91.56 & -23.61 \\ -23.61 & 167.28 \end{smallmatrix}\right]$, and $\ell(x,u) = x^\top Qx + u^\top Ru$, where $Q = 0.1I_2$, and $R = 1$. Note that, in this case, $\mathbb{X}_f = \{x \in \mathbb{R}^2 : x^\top Px \leq 1\}$. The oracle returns $+1$ when a feasible solution is found to the problem (2) from a given initial condition in $\mathbb{X}$, and -1 otherwise.

We select a lower bound on $\mathcal{L}_0$ to be $N_L = 100$, and a minimal initial class-imbalance ratio of the labeled set to be $\nu = 0.1$. To begin, we sample uniformly distributed random points within $\mathbb{X}$. Implementing Algorithm 2 results in 97 infeasible points (due to the large size of $\mathbb{X}$ and constraints on $\mathbb{U}$) and 1 feasible point. Using the idea in Remark 2, each point on the prediction horizon for the feasible point is feasible due

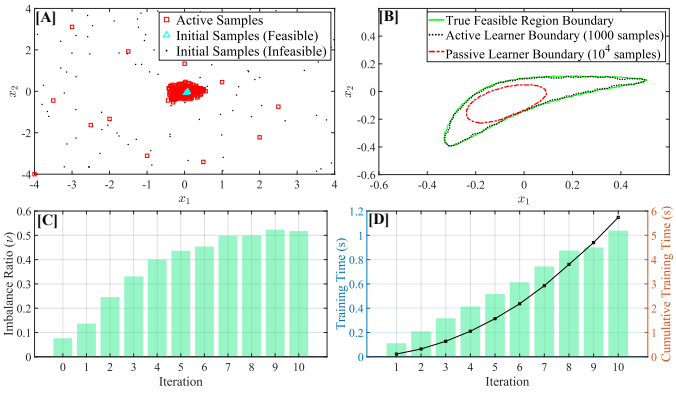

Fig. 1. [A] Distribution of initial random samples (infeasible: dots; feasible: triangles). Note that the initial distribution is heavily imbalanced towards infeasible samples because the measure of the feasible set is small. The active samples gathered over multiple iterations are shown via red squares. [B] Reconstruction of the true feasible set boundary (green continuous line) using active learning (black dotted line) and passive learning (red dashed line). Sub- and super-level sets of these reconstructions will form inner and outer approximations, respectively. [C] Imbalance ratio (ν) trend with iterations. [D] Training time for the active learning machine per iteration (bars) and cumulative training time (black line) in seconds.

to invariance, which results in $N = 8$ feasible points, making $|\mathcal{L}_0| = 105$. The initial sample set is illustrated in Fig. 1[A].

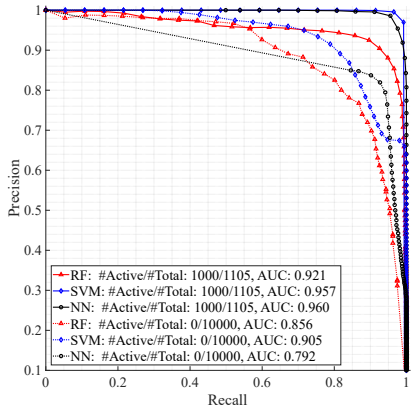

Fig. 2. Comparison of precision-recall (PR) curves for different classifiers: random forests, support vector machines, and neural networks. AUC stands for area under the PR curve.

The unlabeled set of samples is generated using low-discrepancy Halton samples [10]; $|\mathcal{S}| = 10^6$. We fix the batch size to be $B = 100$ samples. The total number of active samples to be selected is set to $N_s = 1000$. The classifier ψ_k is selected to be a support vector machine (SVM) with Gaussian radial basis function kernels and Platt scaling to generate probabilistic outputs. The learner is implemented in MATLAB R2017a via the Statistics and Machine Learning toolbox with default hyperparameters. The SVM's prediction probabilities are used to compute entropy and KL divergence as described in (6) and (7), respectively. At the kth iteration, a set $S' \subset \mathcal{S}$ is pre-selected as the effective set of unlabeled samples with $q = 2000$ samples drawn from $\mathcal{S}$ that exhibit the highest entropy, estimated by the classifier ψ_{k-1}, for reasons discussed in Remark 3. Subsequently, the submodular maximization problem is solved at each iteration via the greedy algorithm.

Fig. 1[A] shows the active samples (red squares) selected over all 10 iterations; as expected, the algorithm chooses samples that are in the region of highest uncertainty, while mostly ignoring samples near the periphery of $\mathbb{X}$ since they are more likely to be infeasible. The effectiveness of the approach is demonstrated in Fig. 1[B], where we compare an active learner with a passive (or not active) learner trained on 10^4 labeled samples. In spite of the $10\times$ increase of the training set, the estimated boundary of the passive learner (red dashed line) is far more conservative than our approach (black dotted line); our final estimate is close to the true feasible region boundary (green continuous line). Although we did not explicitly include imbalance correction as an objective, sampling around higher uncertainty regions naturally reduces imbalance in the resulting training set. This is apparent from Fig. 1[C]: the imbalance ratio (expressed as a percentage) improves from around 7% to near 50% at the end of 10 AL iterations. The performance improvement comes with a trade-off in terms of training time. As Fig. 1[D] shows, the time required to train the SVM increases with each iteration due to an increase in the number of training samples. The resultant 6 s that is required to train the SVM is $100\times$ higher than a corresponding SVM with a training set of equal cardinality (0.06 s) and $\approx 3\times$ higher than the SVM trained on 10000 samples (2.34 s).

A significant advantage of the proposed method is that it seamlessly integrates with a wide variety of supervised learning algorithms. To illustrate, we test the active learning algorithm using three different classification frameworks: random forests, support vector machines, and neural networks. Our goal is not to compare the learners themeselves (hence, hyperparameter tuning is not emphasized here), but to show that using directed sampling improves the performance of each classifier while hyperparameters are kept constant. This is illustrated by constructing the precision-recall characteristics of each of the algorithms with and without active learning. We use precision and recall instead of classification accuracy as a metric, because the dataset could be heavily imbalanced. In such cases, a naive predictor that always predicts the majority class will be very accurate owing to the imbalance in samples. The precision-recall characteristics are computed based on a uniform 300×300 grid within $[-0.75, 0.75]^2$ to test classification performance near the true decision boundary. We see in Fig. 2 that the precision-recall (PR) curves for all three classifiers with 1000 active samples (continuous lines) exhibit significantly higher AUCs in comparison with their passive counterparts (dotted lines) in spite of $10\times$ more samples for training.

Finally, we compare the performance of different AL algorithms: marginal SVM [11], sampling by uncertainty [24], batch-mode active learning using BatchRank and BatchRand algorithms [15], and the proposed method. Results from the comparative study tested on the same uniform 300×300 grid mentioned above are presented in Table I. Myopic algorithms

TABLE I

COMPARISON OF AL ALGORITHMS IN TERMS OF: AREA UNDER A PR CURVE WITH 1000 TOTAL ACTIVE SAMPLES, TIME REQUIRED TO SELECT $B = 100$ FROM A POOL OF 2000 UNLABELED SAMPLES, AND TIGHTNESS OF SOLUTION QUALITY GUARANTEES.

Algorithm	AUC	CPU Time [s]	Optimization Method	Solution Guarantees	Complexity
Marginal SVM	0.772	<1	n/a	n/a	$\mathbf{O}(\lvert\mathcal{S}\rvert \ln \lvert\mathcal{S}\rvert)$
Uncertainty Sampling	0.924	<1	n/a	n/a	$\mathbf{O}(\lvert\mathcal{S}\rvert \ln \lvert\mathcal{S}\rvert)$
BatchRank	0.955	3.85	integer QP (LP relaxation)	not on true objective function	$\mathbf{O}(\lvert\mathcal{S}\rvert^2)$
BatchRand	0.953	>200	integer QP (SDP relaxation)	not on true objective function	$\mathbf{O}(\lvert\mathcal{S}\rvert^3)$
Proposed	0.957	5.91	submodular max. (greedy)	near-optimal	$\mathbf{O}(\lvert\mathcal{S}\rvert B^2)$

(that rely solely on samples around the decision boundary) such as marginal SVM and uncertainty sampling are outperformed by algorithms that explicitly maximize diversity among selected samples. All three of the algorithms that trade-off exploitation and exploration (BatchRank, BatchRand, and our proposed method) exhibit similar areas under the PR curve. Both the BatchRank and BatchRand algorithms rely on convex relaxations of an NP-hard integer quadratic programming problem: this results in certain complications that our method is immune to. For example, the BatchRand algorithm exploits a semidefinite programming (SDP) relaxation: the solution of the SDP culminates in a spike in training times from 6 s for our method to over 200 s. BatchRank entails a linear programming relaxation whose solution is equivalent to a greedy selection algorithm; however, since the problem solved in BatchRank is a supermodular maximization, one cannot (yet) claim a near-optimal bound on the solution quality, as we can in our proposed formulation. Additionally, since $|S| \gg B$, we reduce computational complexity in our proposed formulation in spite of maintaining a multi-criterion AL framework.

V. CONCLUSIONS

In this paper, we developed an active learning framework for estimating control-relevant sets such as reachable and invariant sets in a data-driven manner. We pose the sample selection paradigm as a submodular maximization problem and leverage a greedy algorithm to compute solutions with guarantees. The potential of the approach in estimating small volumes within admissible state spaces in a data-driven manner and the critical advantage of model-free set estimation is demonstrated empirically.

REFERENCES

[1] A. Vannelli and M. Vidyasagar, "Maximal Lyapunov functions and domains of attraction for autonomous nonlinear systems," *Automatica*, vol. 21, no. 1, pp. 69–80, 1985.

[2] E. C. Kerrigan and J. M. Maciejowski, "Invariant sets for constrained nonlinear discrete-time systems with application to feasibility in model predictive control," in *Proc. of the 39th IEEE Conference on Decision and Control (CDC)*, vol. 5, 2000, pp. 4951–4956.

[3] L. Magni, G. De Nicolao, L. Magnani, and R. Scattolini, "A stabilizing model-based predictive control algorithm for nonlinear systems," *Automatica*, vol. 37, no. 9, pp. 1351–1362, 2001.

[4] G. Chesi, *Domain of attraction: analysis and control via SOS programming*. Springer Science & Business Media, 2011, vol. 415.

[5] M. Korda, D. Henrion, and C. N. Jones, "Inner approximations of the region of attraction for polynomial dynamical systems," *IFAC Proceedings Volumes*, vol. 46, no. 23, pp. 534–539, 2013.

[6] D. Han and D. Panagou, "Chebyshev approximation and higher order derivatives of Lyapunov functions for estimating the domain of attraction," in *Proc. of the 56th IEEE Conference on Decision and Control (CDC)*, 2017, pp. 1181–1186.

[7] A. Chakrabarty, V. Dinh, G. T. Buzzard, S. H. Żak, and A. E. Rundell, "Robust explicit nonlinear model predictive control with integral sliding mode," in *Proc. of the American Control Conference (ACC)*, 2014, pp. 2851–2856.

[8] C. J. Ong, D. Sui, and E. G. Gilbert, "Enlarging the terminal region of nonlinear model predictive control using the support vector machine method," *Automatica*, vol. 42, no. 6, pp. 1011–1016, 2006.

[9] R. E. Allen, A. A. Clark, J. A. Starek, and M. Pavone, "A machine learning approach for real-time reachability analysis," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)*, 2014, pp. 2202–2208.

[10] A. Chakrabarty, V. Dinh, M. J. Corless, A. E. Rundell, S. H. Żak, and G. T. Buzzard, "Support vector machine informed explicit nonlinear model predictive control using low-discrepancy sequences," *IEEE Trans. on Automatic Control*, vol. 62, no. 1, pp. 135–148, 2017.

[11] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *Journal of Machine Learning Research*, vol. 2, no. Nov, pp. 45–66, 2001.

[12] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby, "Selective sampling using the query by committee algorithm," *Machine Learning*, vol. 28, no. 2-3, pp. 133–168, 1997.

[13] A. Krause and C. Guestrin, "Nonmyopic active learning of gaussian processes: an exploration-exploitation approach," in *Proc. of the 24th International Conference on Machine Learning (ICML)*. ACM, 2007, pp. 449–456.

[14] Y. Guo, "Active instance sampling via matrix partition," in *Advances in Neural Information Processing Systems (NIPS)*, 2010, pp. 802–810.

[15] S. Chakraborty, V. Balasubramanian, Q. Sun, S. Panchanathan, and J. Ye, "Active batch selection via convex relaxations with guaranteed solution bounds," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 37, no. 10, pp. 1945–1958, 2015.

[16] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions," *Mathematical Programming*, vol. 14, no. 1, pp. 265–294, 1978.

[17] S. C. Hoi, R. Jin, J. Zhu, and M. R. Lyu, "Batch mode active learning and its application to medical image classification," in *Proc. of the 23rd International Conference on Machine learning (ICML)*. ACM, 2006, pp. 417–424.

[18] K. Wei, R. Iyer, and J. Bilmes, "Submodularity in data subset selection and active learning," in *International Conference on Machine Learning*, 2015, pp. 1954–1963.

[19] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[20] D. J. Burns, C. Danielson, J. Zhou, and S. Di Cairano, "Reconfigurable model predictive control for multi-evaporator vapor compression systems," *IEEE Trans. on Control Systems Technology*, vol. PP, no. 99, pp. 1–17, 2017.

[21] J. H. Abel, A. Chakrabarty, and F. J. Doyle III, "Controlling biological time: Nonlinear model predictive control for populations of circadian oscillators," in *Emerging Applications of Control and Systems Theory*. Springer, 2018, pp. 123–138.

[22] U. Feige, "A threshold of $ln|n|$ for approximating set cover," *Journal of the ACM (JACM)*, vol. 45, no. 4, pp. 634–652, 1998.

[23] G. Pin, M. Filippo, F. A. Pellegrino, G. Fenu, and T. Parisini, "Approximate model predictive control laws for constrained nonlinear discrete-time systems: analysis and offline design," *International Journal of Control*, vol. 86, no. 5, pp. 804–820, 2013.

[24] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *Proc. of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1994, pp. 3–12.