Online Instrumental Variable Regression with Applications to Online Linear System Identification

Arun Venkatraman¹, Wen Sun¹, Martial Hebert¹, J. Andrew Bagnell¹, Byron Boots²
¹Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213
²School of Interactive Computing, Georgia Institute of Technology, Atlanta, GA 30332
{arunvenk,wensun,hebert,dbagnell}@cs.cmu.edu, bboots@cc.gatech.edu

Abstract
Instrumental variable regression (IVR) is a statistical technique utilized to recover unbiased estimators when there are errors in the independent variables. Estimator bias in learned time series models can yield poor performance in applications such as long-term prediction and filtering where the recursive use of the model results in the accumulation of propagated error. However, prior work addressed the IVR objective in the batch setting, where it is necessary to store the entire dataset in memory – an infeasible requirement in large dataset scenarios. In this work, we develop Online Instrumental Variable Regression (OIVR), an algorithm that is capable of updating the learned estimator with streaming data. We show that the online adaptation of IVR enjoys a no-regret performance guarantee with respect to the original batch setting by taking advantage of any no-regret online learning algorithm inside OIVR for the underlying update steps. We experimentally demonstrate the efficacy of our algorithm in combination with popular no-regret online algorithms for the task of learning predictive dynamical system models and on a prototypical econometrics instrumental variable regression problem.

Introduction
Instrumental variable regression (IVR) is a popular statistical linear regression technique to help remove bias in the prediction of targets when both the features and targets are correlated with some unknown additive noise, usually a variable omitted from the regression due to the difficulty in observing it (Bowden and Turkington 1990). In this setting, ordinary least squares (OLS) (i.e. linear regression) from features to targets leads to a biased estimate of the dependence between features and targets. For applications where the underlying unbiased dependency is required, such as in the study of causal effects for econometrics (Miguel, Satyanath, and Sergenti 2004), epidemiology (Greenland 2000), or for the learning of dynamical system models (Söderström and Stoica 2002), IVR provides a technique to remove the correlation with the unobserved variables.

We focus in this work on the regression application of instrumental variables where the IVR process consists of multiple linear regressions steps. Prior attention on IVR has focused on the batch learning scenario: each step of regression is performed in whole with all of the data at once. However, with the ever growing prevalence of large datasets, such an approach becomes quickly infeasible due to the scaling of the memory and computational complexity with regards to the data set size and feature dimensionality. Towards this end, we propose an online version of instrumental variable regression that replaces each of the regression steps with an online learner.

Specifically, we develop an Online Instrumental Variable Regression (OIVR) procedure that can be regarded as a reduction to no-regret online learning. Under the assumption that the set of regression and instrumental variables are i.i.d., we derive a strong no-regret bound with respect to the desired objective optimized by the batch setting (batch IVR). Our theorem allows us to take advantage of any no-regret online learning procedure for the multiple regression steps in IVR. We explicitly show that OIVR allows us to introduce a new family of online system identification algorithms that can exploit no-regret online learning. This reduction extends on the initial reduction given by Hefny et. al (Hefny, Downey, and J. Gordon 2015) from batch predictive state dynamical system learning to batch IVR. Finally, we investigate the experimental performance of several popular online algorithms such as Online Gradient Descent (OGD) (Zinkevich 2003), Online Newton Step (Hazan, Agarwal, and Kale 2006) (ONS), Implicit Online Gradient Descent (iOGD) (Kulis et al. 2010), and Follow The Regularized Leader (FTRL) (Shalev-Shwartz 2011) in the context of OIVR for both dynamical system modeling and on a simple but illustrative econometrics example.

Instrumental Variable Regression
Consider the standard linear regression scenario where we wish to find $A$ given design matrices (datasets) $X = [x_1 \ x_2 \ \ldots]$ and $Y = [y_1 \ y_2 \ \ldots]$ representing our explanatory variables (features) $x_i \in \mathbb{R}^{n \times 1}$ and outputs (targets) $y_i \in \mathbb{R}^{n \times 1}$. This relationship is modeled by:

$$Y = AX + E$$  \hspace{1cm} (1)

where $E = [\varepsilon_1 \ \varepsilon_2 \ \ldots]$ are independent noise (error).
Solving this via least-squares minimization, gives us:

\[ \hat{A} = Y X^T \left( X X^T \right)^{-1} = (A X + E) X^T \left( X X^T \right)^{-1} \]

\[ = A X X^T \left( X X^T \right)^{-1} + E X^T \left( X X^T \right)^{-1} \]

\[ = A + E X^T \left( X X^T \right)^{-1} \quad \text{(2)} \]

When the number of samples \( T \) goes to infinity, by law of large number, we will have: \( E X^T / T \rightarrow E \left[ x x^T \right] \) and \( X X^T / T \rightarrow E \left[ x x^T \right] \) in probability. Normally, we assume that \( x \) and \( x \) are uncorrelated, which means \( E X^T / T \) converges to zero in probability, \( E \left[ x x^T \right] = 0 \), which yields an unbiased estimate of \( A \) from Eq. 2.

\[ \hat{A} = A + \frac{1}{T} E X^T \left( \frac{1}{T} X X^T \right)^{-1} \rightarrow A \]

However, if \( \varepsilon \) and \( x \) are correlated \( E \left[ x \varepsilon^T \right] \neq 0 \), we are only able to get a biased estimate of \( A \) through the least-squares optimization, since \( E \left[ x \varepsilon^T \right] E \left[ x x^T \right]^{-1} \neq 0 \).

On the other hand, IVR can achieve an unbiased estimate of \( A \) (Rao et al. 2008; Cameron and Trivedi 2005). In IVR, we remove this bias by utilizing an instrumental variable, denoted as \( Z = [z_1, z_2, \ldots] \) in Fig. 1. For a variable to be an instrumental variable, we need two conditions: (1) the instrumental variable \( z \) is correlated with \( x \) such that \( E \left[ z \varepsilon^T \right] \) is full row rank and (2) the instrumental variable is uncorrelated with \( \varepsilon \), i.e. \( E \left[ z \varepsilon^T \right] = 0 \).

![Figure 1: Causal diagram for IVR](image)

Instrumental variable regression proceeds as follows. IVR first linearly regresses from \( Z \) to \( X \) to get \( \hat{X} = X Z^T Z \), where \( Z^T = Z^T (Z Z^T)^{-1} \). Then, IVR linearly regresses from the projection \( X \) to \( Y \) to get an estimate of \( A \):

\[ \hat{A}_{IVR} = Y \hat{X}^T (\hat{X} \hat{X}^T)^{-1} \]

\[ = Y Z^T (Z^T Z)^{-1} \]

\[ = A X Z^T (Z^T Z)^{-1} + E Z^T (Z^T Z)^{-1} \]

\[ = A + E Z^T (Z^T Z)^{-1} \]

\[ = A + \frac{E Z^T (Z^T Z)^{-1}}{T} \]

Note that \( \hat{A}_{IVR} \rightarrow A \) in probability since \( E Z^T / T \rightarrow 0 \) in probability under the assumption that instrumental variable \( \varepsilon \) and \( x \) are uncorrelated.

The process of instrumental variable regression can be represented through the following two steps (also known as two-stage regression):

\[ M^* \leftarrow \arg \min_M \| X - M Z \|^2_F \]

\[ A^* \leftarrow \arg \min_A \| Y - A M^* Z \|^2_F \]

For shorthand, we will generally refer to the final regression stage, Eqn. 4, as the batch IVR objective.

### Algorithm 1 Batch Instrumental Variable Regression

**Input:**
- Explanatory Variable Design Matrix \( X \in \mathbb{R}^{d_x \times n} \)
- Instrumental Variable Design Matrix \( Z \in \mathbb{R}^{d_z \times n} \)
- Prediction Targets Design Matrix \( Y \in \mathbb{R}^{d_y \times n} \)

**Output:** \( A^* \in \mathbb{R}^{d_y \times d_x} \)

1. \( M^* \leftarrow \arg \min_M \| X - M Z \|^2_F \)
2. \( \hat{X} \leftarrow M^* Z \)
3. \( A^* \leftarrow \arg \min_A \| Y - A \hat{X} \|^2_F \)
4. return \( A^* \)

### Online Instrumental Variable Regression

The formulation of online algorithms yields a two-fold benefit – first, it allows us to use datasets that are too large to fit in memory by considering only one or a few data points at a time; second, it allows us to run our algorithm with streaming data, a vital capability in fields such as robotics where many sensors can push out volumes of data in seconds. In the following section, we will formulate an online, streaming-capable adaptation of the batch Instrumental Variable Regression (IVR) algorithm. We show that our online algorithm has a strong theoretical performance guarantee with respect to the performance measure of the IVR in the batch setting.

In the batch setup of IVR (Algorithm 1), we require all the datapoints \( a \) \( prior \) in order to find \( A^* \). To create an online version of this algorithm, we must instead compute estimates \( M_t \) and \( A_t \) as it receives a single set of data points, \( x_t, z_t, y_t \). To motivate our adaptation of IVR, we first consider the adaptation of OLS (i.e. linear regression) to the online setting. Given the design matrices \( X = [x_0, \ldots, x_t, \ldots] \) and \( Y = [y_0, \ldots, y_t, \ldots] \) in OLS, we optimize the following batch objective over all the data points:

\[ \beta^* = \arg \min_{\beta} \| \beta X - Y \|^2_F = \arg \min_{\beta} \sum_t \ell_t(\beta) \]

where the loss function \( \ell_t(\beta) = \| \beta x_t - y_t \|^2_2 \) is the L2 loss for the corresponding pair of data points \( (x_t, y_t) \). To formulate an online OLS algorithm, we may naturally try to optimize the L2 loss for an individual data point pair \( \| \beta x_t - y_t \|^2_2 \) at each timestep without directly considering the loss induced by other pairs. Prior work in the literature has developed algorithms that address this problem of considering losses \( \ell_t \) and predicting a \( \beta_{t+1} \) while still achieving provable performance with respect to the optimization of \( \beta \) over the batch objective (Zinkevich 2003; Hazan, Agarwal, and Kale 2006; Shalev-Shwartz 2011). The performance guarantee of these algorithms is in terms of the (average) regret, which is defined as:

\[ \frac{1}{T} \text{REGRET} = \frac{1}{T} \sum_t \ell_t(\beta_t) - \frac{1}{T} \min_{\beta} \sum_t \ell_t(\beta) \]

We say a learning procedure is no-regret if \( \lim_{T \to \infty} \frac{1}{T} \text{REGRET} = 0 \Rightarrow \text{REGRET} \in o(T) \).

Intuitively, the no-regret property tells us that the optimization of the the loss in this way gives us a solution that
is competitive with the best result in hindsight (i.e. if we had optimized over the losses from all data points). In IVR (Algorithm 1), lines 1 and 3 are each linear regression steps which are individually the same as Eqn. 5. Motivated by this, we introduce Online Instrumental Variable Regression (OIVR), in which we utilize a no-regret online learner for the individual batch linear regressions in IVR. The detailed flow of OIVR is shown in Algorithm 2.

Algorithm 2 Online Instrumental Variable Regression with No-Regret Learners

```
Input:
▷ no-regret online learning procedures LEARN1, LEARN2
▷ Streaming data sources for the explanatory variable $S_x(t): t \to x \in \mathbb{R}^{d_x}$, the instrumental variable $S_z(t): t \to z \in \mathbb{R}^{d_z}$, and the target variable $S_y(t) \to y \in \mathbb{R}^{d_y}$

Output: $\bar{A}_T \in \mathbb{R}^{d_y \times d_z}$

1: Initialize $M_0 \in \mathbb{R}^{d_z \times d_x}$, $A_0 \in \mathbb{R}^{d_y \times d_z}$
2: Initialize $M_0 \leftarrow 0 \in \mathbb{R}^{d_z \times d_x}$, $A_0 \leftarrow 0 \in \mathbb{R}^{d_y \times d_z}$
3: Initialize $t \leftarrow 1$
4: while $S_x \neq \emptyset$ and $S_z \neq \emptyset$ and $S_y \neq \emptyset$ do
5: \hspace{1em} $(x_t, z_t, y_t) \leftarrow (S_x(t), S_z(t), S_y(t))$
6: \hspace{1em} $M_t \leftarrow \text{LEARN1}(z_t, x_t, M_{t-1})$
7: \hspace{1em} $A_t \leftarrow ((t-1)M_{t-1} + M_t)/t$
8: \hspace{1em} $\bar{x}_t \leftarrow M_t z_t$
9: \hspace{1em} $\bar{A}_t \leftarrow \text{LEARN2}(\bar{x}_t, \bar{y}_t, A_{t-1})$
10: \hspace{1em} $\bar{A}_t \leftarrow ((t-1)\bar{A}_{t-1} + A_t)/t$
11: \hspace{1em} $t \leftarrow t + 1$
12: end while
13: return $\bar{A}_t$
```

From the definition of no-regret for the optimization on lines 6 and 9 in Algorithm 2, we get the following:

$$\frac{1}{T} \sum_t \| M_t z_t - x_t \|^2_F - \frac{1}{T} \min_M \sum_t \| M z_t - x_t \|^2_F \leq o(T)$$
$$\frac{1}{T} \sum_t \| A_t M_t z_t - y_t \|^2_F - \frac{1}{T} \min_A \sum_t \| A M_t z_t - y_t \|^2_F \leq o(T)$$

Though these regret bounds give us a guarantee on each individual regression with respect the sequence of data points, they fail to give us the desired performance bound as we get in the single OLS scenario; these bounds do not show that this method is competitive with the optimal result from batch IVR in hindsight (e.g., how close is that this method is competitive with the optimal result from get in the single OLS scenario; these bounds do not show that they fail to give us the desired performance bound as we see in the result of the below lemma.

**Lemma 1.** Given a sequence of convex loss functions $\{f_t(\beta)\}$, $1 \leq t \leq T$, and the sequence of $\{\beta_t\}$ that is generated by any no-regret online algorithm, under the assumption that $\ell_t$ is i.i.d and $\ell = E(\ell_t)$, the average $\beta$ over $T$ has the following properties:

$$E[\ell(\beta) - \ell(\beta^*)] \leq \frac{2}{\alpha T} r(T) \to 0, \quad T \to \infty, \quad (7)$$

where $r(T)$ stands for the function of the regret bound with respect to $T^2$, which is sublinear and belongs to $o(T)$ for all no-regret online learning algorithms. When $\ell$ is strongly convex with respect to $\beta$ in norm $\| \cdot \|$, we have:

$$E[\|\beta - \beta^*\|] \leq \frac{2}{\alpha T} r(T) \to 0, \quad T \to \infty \quad (8)$$

Similar online-to-batch analysis can be found in (Cesa-Bianchi, Conconi, and Gentile 2004; Littlestone 2014; Hazan and Kale 2014). For completeness, we include the proof of the lemma in the appendix.

With this, we now approach the main theorem for the regret bound on Online Instrumental Variable Regression. We explicitly assume that $x_t$, $y_t$, and $z_t$ are i.i.d, and $x = E(x_t)$, $y = E(y_t)$, $z = E(z_t)$, and $E(z_t^2)$ is positive definite.

**Theorem 2.** Assume $(x_t, y_t, z_t)$ are i.i.d. and $E(z^T z)$ is positive definite. Following any online no-regret procedure on the convex L2 losses for $M_t$ and $A_t$ and computing $\bar{M}_t$, $\bar{A}_t$ as shown in Algorithm 2, we get that as $T \to \infty$:

$$E[\|\bar{A}_T M^* z - y\|_2^2] \to E[\|A^* M^* z - y\|_2^2] \quad (9)$$
and

$$\bar{A}_T \to A^* \quad (10)$$

1Available at http://www.cs.cmu.edu/~arunvenk
2For instance, online gradient descent (Zinkevich 2003) has $r_t(T) = C \sqrt{T}$ for some positive constant $C$. 

2103
for the $A^*, M^*$ from Batch IVR (Alg. 1).

**Proof.** For the sake of brevity, we provide the complete proof in the appendix and an abbreviated sketch below.

Since we run a no-regret online algorithm for $A_t$ on loss function $\|A_t \bar{M}_t z_t - y_t\|^2_2$, we have:

$$\sum_t \|A_t \bar{M}_t z_t - y_t\|^2_2 \leq \text{REGRET}_A + \sum_t \|A^* \bar{M}_t z_t - y_t\|^2_2$$

where $\sum_t$ denotes $\sum_{t=1}^T$.

Let $\epsilon_t = M^* - \bar{M}_t$. Then, expanding the squared norms on the left and right side of the inequality, rearranging terms, and upperbounding the terms we get:

$$\sum_t \|A_t M^* z_t - y_t\|^2_2 \leq \text{REGRET}_A$$

$$+ \sum_t \|A^* M^* z_t - y_t\|^2_2$$

$$+ \|A^*\|^2_F \|\epsilon_t\|^2_F \|z_t\|^2_2 + \|A_t\|^2_F \|\epsilon_t\|^2_F \|z_t\|^2_2$$

$$+ 2\langle A^* \epsilon_t z_t \rangle$$

$$+ 2\langle (A_t \epsilon_t - y_t) (A^* \epsilon_t z_t) \rangle$$

Assume that $\|z_t\|^2_2$, $\|y_t\|^2_F$, $\|M^*\|^2_F$, $\|A_t\|^2_F$, $\|A^*\|^2_F$ are each always upper bounded by some positive constant. Defining positive constants $C_1$ and $C_2$ appropriately and using the Cauchy-Swartz and triangle inequalities, we get:

$$\sum_t \|A_t M^* z_t - y_t\|^2_2 \leq \text{REGRET}_A$$

$$+ \sum_t \|A^* M^* z_t - y_t\|^2_2 + C_1 \|\epsilon_t\|^2_F + C_2 \|\epsilon_t\|_F$$

Since we run a no-regret online algorithm on loss $\|M_t z_t - x_t\|^2_2$ with the assumptions that $z_t$, $x_t$, and $y_t$ are i.i.d and $E[zz^T]$ is positive definite, we get as $t \to \infty$:

$$E[\|\epsilon_t\|^2_F] \leq \frac{1}{T} r_M(t) \to 0$$

and $E[\|\epsilon_t\|_F] \leq \sqrt{\frac{1}{T} r_M(t)} \to 0$,

where $E$ is the expectation under the randomness of the sequences $z$ and $x$. Considering the stochastic setting (i.e. i.i.d $z_t$, $x_t$, and $y_t$), applying Cesaro Mean (Hardy 2000) and taking $T \to \infty$:

$$\frac{1}{T} E \left[ \sum_t \|A_t M^* z - y_t\|^2_2 \right] \leq E \left[ \|A^* M^* z - y_t\|^2_2 \right]$$

Thus, we have shown the algorithm is no-regret.

Let $\bar{A}_T = \frac{1}{T} \sum A_t$. Using Jensen’s inequality, we get:

$$E \left[ \|\bar{A}_T M^* z - y_t\|^2_2 \right] \leq E \left[ \|A^* M^* z - y_t\|^2_2 \right]$$

Since the above is valid for any $A^*$, let $A^* = \arg \min_A E \left[ \|A M^* z - y\|^2_2 \right]$. Due to bounding from above and below by the objective at $A^*$, we get:

$$E \left[ \|\bar{A}_T M^* z - y\|^2_2 \right] \to E \left[ \|A^* M^* z - y\|^2_2 \right]$$

With $E[zz^T] > 0$ resulting in strongly convex objective, we get a unique minimizer for the objective:

$$\bar{A}_T \to A^*, \; T \to \infty$$

We also want to note that the regret rate of our algorithm depends on the no-regret online algorithms used. For instance, if we use OGD, which has no-regret rate of $O(\sqrt{T})$ for LEARN1 and LEARN2, then our algorithm has a no-regret rate of $O(\sqrt{T})$. The choice of learning algorithm is related to the desired trade-off between computational complexity and convergence rate. FTRL and ONS can have faster convergence, making them suitable for applications where obtaining samples is difficult: e.g., data from a physical robot. In contrast, gradient-based algorithms (e.g. iOGD, OGD) have lower computational complexity but may converge slower, making them useful for scenarios where obtaining samples is cheap, e.g., data from video games.

**Dynamical Systems as Instrumental Variable Models**

For a dynamical system, let us define state $s \in S \in \mathbb{R}^m$ and observation $o \in \mathcal{O} \in \mathbb{R}^n$. At time step $t$, the system stochastically transitions from state $s_t$ to state $s_{t+1}$ and then receives an observation $o_{t+1}$ corresponding to $s_{t+1}$. A dynamical system generates a sequence of observations $o_t$ from latent states $s_t$ connected in a chain. A popular family of algorithms for representing and learning dynamical systems are predictive state representations (PSRs) (Littman, Sutton, and Singh 2001; Singh, James, and Rudary 2004; Boots and Gordon 2012; Boots and Gordon 2011; Hefny, Downey, and J. Gordon 2011; Hefny, Downey, and J. Gordon 2015). It also has been shown in (Boots and Gordon 2011b; Hefny, Downey, and J. Gordon 2015) that we can interpret the problem of learning PSRs as linear instrumental-variable regression, which reduces the dynamical system learning problem to a regression problem.

Following (Hefny, Downey, and J. Gordon 2015), we define the predictive state $Q$ as $Q_t = E(o_{t:t+k-1} | o_{1:t-1})$ (instead of tracking the posterior distribution $P(s_t|o_{1:t-1})$ on state, we track the observable representation $Q_t$), where $o_{t:t+k-1}$ is a $k$-step time window of future observations. We also define the extended future observations as $o_{t:t+k}$, which is a $(k+1)$-step time window of future observations. The predictive state representation of extended futures is defined as $P_t = E(o_{t:t+k} | o_{1:t-1})$. Therefore, learning a dynamical system is equivalent to finding an operator $A$ that maps from $Q_t$ to $P_t$:

$$P_t = AQ_t$$

With $A$ and the initial belief $Q_0 = E(o_{0:k-1})$, we are able to perform filtering and prediction. Given the belief $Q_t$ at step $t$, we use $A$ to compute $P_t = E(o_{t:t+k} | o_{1:t-1})$. To compute $E(o_{t:t+k} | o_{1:t-1})$ (prediction), we simply drop the $o_t$ from $P_t$. For filtering, given a new observation $o_t$, under the assumption that the extended future $o_{t:t+k}$ has constant covariance, we can compute $E(o_{t:t+k} | o_{1:t})$ by simply performing a conditional Gaussian operation.
A naive approach to compute \( A \) is to use ordinary linear regression directly from futures \( o_{t:t+k-1} \) to extended futures \( o_{t:t+k} \). However, even though \( o_{t:t+k-1} \) and \( o_{t:t+k} \) are unbiased samples of \( Q_t \) and \( P_t \), they are noisy observations of \( Q_t \) and \( P_t \) respectively. The noises overlap: \( o_{t:t+k-1} \) and \( o_{t:t+k} \) share a \( k \)-step time window (Hefny, Downey, and J. Gordon 2015). Therefore, directly regressing from futures instead of building a large Hankel matrix ((Hefny, Downey, and J. Gordon 2015; Boots, Siddiqi, and Gordon 2003) requires \( O((kn)^2) \) computations at each step compared to the \( O((kn)^3) \) in the batch-based algorithms (usually due to matrix inversions).

Experiments

We demonstrate the performance of OIVR on a variety of dynamics benchmark and one illustrative econometrics problem. In Fig. 2, we show the convergence of the estimated \( \hat{A}_t \) in OIVR to the \( A^* \) computed with IVR. As an additional performance metric, we report the observation prediction error with a constant covariance Kalman filter using \( \hat{A}_t \) (Fig. 3) on a set of held out test trajectories. For computational reasons, we report the filter error after every 50 data points given to the online learner. Below we describe each of our test benches.

MG-10 The Mackey-Glass (MG) time-series is a standard dynamical modelling benchmark (Ralaivola and D’Alché-Buc 2004; Wingate and Singh 2006) generated from the nonlinear time-delay differential equation  
\[
\dot{x}(t) = -bx(t) + \frac{ax(t-\tau)}{1+x(t-\tau)^m}.
\]
This system produces chaotic behavior for larger time delays \( \tau \) (seconds).

Helicopter The simulated helicopter from (Abbeel and Ng 2005) computes its dynamics in a 21-dimensional state space with a 4-dimensional control input. In our experiments, a closed loop LQR controller attempts to bring the helicopter to hover at a fixed point from randomly chosen starting configurations. White noise is added in each state transition. The LQR controller chooses actions based on state and it poses a challenge for the learner to extract this implicit relationship governing the evolution of the system.

Airplane Flight Take Off We also consider the complex dynamics generated during a DA-42 airplane’s take off in a flight simulator, X-plane (Research 2015), a well known program for training pilots. Trajectories of observations, which include among others speed, height, angles, and the pilot’s control inputs, were collected were collected from a human expert controlling the aircraft. Due to high correlation among the observation dimensions, we precompute a whitening projection at the beginning of online learning using a small set of observations to reduce the dimensionality of the observations by an order of magnitude.

Online Learning for Dynamical Systems

Given OIVR, learning dynamical systems online becomes straightforward. To apply Alg. 2 to model dynamical systems, we maintain a \( k \)-step time window of the future \( o_{t:t+k-1} \), a \((k+1)\)-step time window of the extended future \( o_{t:t+k} \), and a \( k \)-step time window of the past \( o_{t-k:t-1} \). Matching terms to Alg. 2, we set \( x_t = o_{t:t+k-1} \), \( y_t = o_{t:t+k} \), and \( z_t = o_{t-k:t-1} \). With \( x_t, y_t \) and \( z_t \), we update \( M_t \) and \( A_t \) following lines 6 and 9. When a new observation \( o_{t+k+1} \) is received, the update of \( x_t \) and \( y_t \) and \( z_t \) to \( x_{t+1}, y_{t+1} \) and \( z_{t+1} \) is simple and can be computed efficiently (e.g., to compute \( y_{t+1} = o_{t+1:t+k+1} \), we simply drop \( o_t \) from \( x_t \) and append \( o_{t+k+1} \) at the end (i.e. circular buffer)).

By maintaining these three fixed-step time window of observations instead of building a large Hankel matrix (Hefny, Downey, and J. Gordon 2015; Boots, Siddiqi, and Gordon 2011) that stores concatenations of all the observations, we significantly reduce the required space complexity. At every online update step (lines 6 and 9 in Alg. 2), the online learning procedure usually has lower computational complexity. For instance, using Online Gradient Descent (Zinkevich 2003) requires \( O((kn)^2) \) computations at each step compared to the \( O((kn)^3) \) in the batch-based algorithms (usually due to matrix inversions).
Our experimental results show that OIVR algorithms work for Online Instrumental Variable Regression and proved strong theoretical performance bounds with regard to the traditional batch Instrumental Variable Regression setting through a connection to no-regret online algorithms. Through connections between IVR and dynamical system identification, we introduced a rich new family of online system identification algorithms. Our experimental results show that OIVR algorithms work well in practice on a variety of benchmark datasets.

Acknowledgments
This material is based upon work supported in part by: National Science Foundation Graduate Research Fellowship Grant No. DGE-1252522 and DARPA ALIAS contract number HR0011-15-C-0027. The authors also thank Geoff Gordon for valuable and insightful discussions.
References


Boots, B., and Gordon, G. 2011a. An online spectral learning algorithm for partially observable nonlinear dynamical systems. In AAAI.


Boots, B.; Siddiqi, S.; and Gordon, G. 2011. Closing the learning planning loop with predictive state representations. IJRR.


Ralaivola, L., and D’Alche-Buc, F. 2004. Dynamical modeling with kernels for nonlinear time series prediction. NIPS.