Distribution-free Prediction and Confidence Intervals for ARX Systems with Instrumental Variables

Babolcsay, Barbara
Supervisor: Csáji, Balázs Csanád

1. Introduction and Previous Work

In the first semester of this project I got to know the Sign-Perturbed Sums (SPS) method introduced by B. Cs. Csáji, M. C. Campi and E. Weyer in [2]. SPS is a statistical region-estimation method that constructs non-asymptotic and distribution-free confidence regions around a point estimate of the true model parameters.

Initially, the SPS method was applied in the case of linear regression problems where we can get an exact confidence region around the least-squares (LS) estimate of the coefficients. Using convex optimization it is also possible to efficiently create outer approximating ellipsoids, in order to reduce the computational cost [2].

Last semester I stayed within this topic and worked on numerical experiments with the SPS method in the case of least-squares and regularized least-squares problems. I also implemented the outer approximation algorithm and transformed the results from the parameter space to the model space through solving another convex optimization problem with Lagrangian multipliers.

This semester I chose to move on in the direction of time series [1] where multiple modifications and generalizations can be made. From this area I worked with the ARX model with the help of instrumental variables.

2. Predicting Scalar ARX Systems

2.1. Scalar ARX System Framework

In the ARX model we assume that we have observations in the following form:

$$Y_t = \sum_{i=1}^{d_1} a_i^* Y_{t-i} + \sum_{i=1}^{d_2} b_i^* U_{t-i} + N_t,$$

where $Y_t \in \mathbf{R}$ is the output in t (which is a discrete time index); $U_t \in \mathbf{R}$ is the input in t that can depend on the previous inputs; $[a_1^*, ..., a_{d_1}^*, b_1^*, ..., b_{d_2}^*]^T \in \mathbf{R}^d$ is the (constant) true parameter and $N_t \in \mathbf{R}$ is a noise term.

We can reformulate the ARX model using the linear regression notation system:

$$Y_t = \phi_t^T \theta^* + N_t,$$

where $\phi_t = [Y_{t-1}, ..., Y_{t-d_1}, U_{t-1}, ..., U_{t-d_2}]^T$ are the regressors, $\theta^* = [a_1^*, ..., a_{d_1}^*, b_1^*, ..., b_{d_2}^*]^T$ is the true parameter and N_t is a noise term.

We wish to estimate θ^* from a finite sample and give a confidence region around the estimate. The problem with the original version of SPS is that in this case the regressors are not exougenous - not independent from the noise terms - and that was one of the necessary assumptions before. The idea is to handle this with the help of instrumental variables (IVs).

2.2. Instrumental Variables

In the case of endogenous regressors we can introduce new variables, so called instruments [4] that meet the following two expectations:

- the IVs must be correlated with the regressors
- the IVs cannot be correlated with the noise term.

Instrumental variables (notation: $\{\psi_t\}$) can be constructed in multiple ways, for instance one can use only the previous inputs, $U_{t-1}, U_{t-2}, ...$ since they can only depend on the other inputs and some independent noise terms. Another method is to estimate the true system parameters (e.g. with LS) and generate new noise-free outputs with them. During the numerical experiments I will present both of these methods.

Similarly to LS, we can calculate the IV estimate for an ARX system by solving a modified normal equation:

$$\sum_{t=1}^{n} \psi_t (Y_t - \phi_t^T \theta) = 0, \qquad (1)$$

from which

$$\hat{\theta}_{IV} = \Big(\sum_{t=1}^{n} \psi_t \phi_t^T \Big)^{-1} \sum_{t=1}^{n} \psi_t Y_t.$$

Since the generated IVs will be independent from the noise but they will be correlated with the original regressors, we can replace our regressors with them and implement SPS in this setup.

2.3. SPS for ARX Problems

Now I would like to describe how we can apply SPS method to ARX systems based on [6]. We can assume that we have data generated by an ARX system as described above and that we already have access to the IVs we need: $\{\psi_t\}$. The additional assumptions we make are:

- $\{N_t\}$ is a sequence of independent random variables with a symmetric distribution around zero
- $\det V_n \neq 0$ a.s., where $V_n = \frac{1}{n}\sum_{t=1}^{n} \psi_t \phi_t^T$

Let us recall the main idea behind SPS method: if our θ estimate is close enough to the real parameter, θ^*, than the error term between the estimated outputs and the observed ones should be close to the noise terms, N_t. Since N_t is symmetrically distributed, if we perturb the error terms with different sequences of Rademacher variables ($\{\alpha_{i,t}\}$ where $P(\alpha_{i,t} = \pm 1) = \frac{1}{2}$) the probabilities should remain approximately the same - if θ is close enough to θ^*.

Differently from the linear regression case we build the confidence region around the IV-estimate $\hat{\theta}_{IV}$ because we would like to perturb equation (1).

Let us say that we aim to construct a confidence region from n observations with probability p such that we determine if a given θ in the parameter space falls within this region. We shall set $m > q > 0$ such that $p = 1 - q/m$. Now we need to generate $n(m-1)$ random signs, $\{\alpha_{i,t}\}$, so we can perturb the reference sum of errors with them. From

$$S_0(\theta) = H_n^{-1/2} \frac{1}{n} \sum_{t=1}^{n} \psi_t (Y_t - \phi_t^T \theta)$$

we get $(m-1)$ perturbed sums:

$$S_i(\theta) = H_n^{-1/2} \frac{1}{n} \sum_{t=1}^{n} \alpha_{i,t} \psi_t (Y_t - \phi_t^T \theta),$$

where $H_n = \frac{1}{n}\sum_{t=1}^{n} \psi_t \psi_t^T$.

If we compare the 2-norm of these vectors and exclude the q largest ones we get a confidence region with the exact probability p. So we accept θ if it is not among the excluded ones in this order. Note: to make the ordering well defined we choose between two sums with the same 2-norm based on a random permutation of the indexes.

Rather than deciding about the parameters one by one it can be easier to build an ellipsoid around the exact region. Considering a larger set we guarentee that the probability of the real parameter being in the ellipsoid is greater than the expected one.

$$\hat{\Theta} = \{\theta : (\theta - \hat{\theta}) V_n^T H_n^{-1} V_n (\theta - \hat{\theta}) \leq r\}$$

As in the linear regression case we have to solve $m - 1$ convex optimization problems and the qth largest value will be the radius of our ellipsoid around the IV estimate:

$$\min \gamma$$

$$\text{s.t.} \lambda \geq 0$$

$$\begin{bmatrix} -I + \lambda A_i & \lambda b_i \\ \lambda b_i^T & \lambda c_i + \gamma \end{bmatrix} \geq 0,$$

where A_i, b_i, c_i come from $\{\phi_t\}, \{\psi_t\}, \{Y_t\}, \hat{\theta}_{IV}$ and $\{\alpha_{i,t}\}$.

2.4. A general framework for perturbation based hypothesis testing methods

In [3] a general framework is presented for similar methods to SPS where we would like to decide if a parameter is in the given confidence region or not with some perturbation of the data. The family of the hypothesis testing methods that can be built up based on this framework called Perturbed Datasets Methods (PDM). Let us introduce the notation $D := \{X_i, Y_i\} \forall i = 1, ..., n$. The general steps of these algorithms are the followings:

1. Generate m different datasets: $D^i(D, \theta)$ based on a random data perturbation setup Γ
2. Define a performance measure of θ: Z and calculate it for the generated models: Z_i
3. Give a well defined ordering of the values Z_i
4. Define the subset of the $m!$ possible orderings over which θ is accepted.

If the generated datasets are conditionally independent for some σ-algebra and in the case of $\theta = \theta^*$ they are identically distributed then it is true for Z_is too. Because of this every ordering has the same probability: $\frac{1}{m!}$, so it is indeed possible to define an exact confidence region by selecting a subset of the orderings.

It is easy to see that SPS method is a part of this family. The random perturbation setup Γ is given by the $m-1$ random sign sequences as follows:

$$W_1 = \text{Id},$$
$$W_i = \text{diag}(\alpha_i),$$

for $i = 2, ..., m$. From these we can generate the perturbed datasets: $D^i(D, \theta) = W_i N(\theta)$, where $N(\theta)$ is the noise term. In the case of $\theta = \theta^*$ the symmetric distribution of the noise guarantees the i.i.d.-ness of the perturbed datasets $D^i(D, \theta)$. The performance measure is some weighted norms in both the linear regression and the ARX case and the ordering is as described above.

2.5. PEM for ARX systems with exchangeable noise

Now let us take another problem: replace the assumption about the symmetry of the noise with exchangebility. This means that we can take any permutation of the noise sequence and the joint distribution stays the same.

For this reason we can define Γ as a sequence of permutation matrices $P_1, ..., P_m$, such that $P_1 = Id$ and the others represent $m-1$ random permutations. The perturbed datasets will be $D^i(D, \theta) = P_i N(\theta)$ and the performance measure will be a weighted distance between θ and the LS or IV estimate corresponding to $D^i(D, \theta)$. The ordering stays the same as above.

It is straightforward to translate the SPS outer approximation problem to the framework of PDM and from that we get very similar convex optimization problems for the permutation version, PEM, too: instead of the $\{W_i\}$ matrices we use $\{P_i\}$ to formulate the optimization problem.

3. Modifying SPS for Multivariate ARX Systems

Another generalization direction is considering not only scalar but vector valued ARX systems. Henceforth, I will look at a simple version of this problem where the output at time t only depends on the previous one, one input and a noise term. We can describe this model as:

$$Y_t = AY_{t-1} + BU_{t-1} + N_t,$$

for $i = 1, ..., n$, where Y_t and $N_t \in \mathbf{R}^{\mathbf{d_y}}$, $U_t \in \mathbf{R}^{\mathbf{d_u}}$. The real parameters are now matrices: $A \in \mathbf{R}^{d_y \times d_y}$, $B \in \mathbf{R}^{d_y \times d_u}$.

It would be possible to reformulate each step of this system as a linear regression problem so that we could apply the previous methods on this vectorized form. But it would result in big and complex equation systems and would be hard to generalize to systems with more steps.

So an alternative way is to interpret the problem as a matrix-variate linear regression system as:

$$Y = \Phi\Theta^* + N,$$

where all the participants are matrices:

$$Y = \left[Y_1^T, ..., Y_n^T\right]^T,$$
$$\Phi = \left[\phi_0^T, ..., \phi_{n-1}^T\right]^T \text{ with } \phi_t = [Y_{t-1}^T, U_{t-1}^T]^T,$$
$$\Theta^* = \left[A^T, B^T\right]^T,$$
$$N = \left[N_0^T, ..., N_{n-1}^T\right]^T.$$

I will describe the modification of SPS to this problem (MIV-SPS) based on [5]. Firstly, some mild assumptions we need to make:

- The row vectors of N are independent and symmetrically distributed around zero
- Let the random matrix Ψ be the matrix containing the instrumental variables as rows, so Ψ and N are independent; then $\Psi^T\Phi$ is full rank almost surely.

The IV estimate we will use comes from a very similar equation as in the scalar case:

$$\Psi^T(Y - \Phi\Theta) = 0,$$

from which:

$$\hat{\Theta}_{IV} = (\Psi^T\Phi)^{-1}\Psi^T Y.$$

Now, we can use $\{W_i\}$ defined in 2.4 and generate the perturbed datasets and measure:

$$S_0(\Theta) = \frac{1}{n} H_n^{-1/2} \Psi^T (Y - \Phi\Theta)$$
$$S_i(\Theta) = \frac{1}{n} H_n^{-1/2} \Psi^T W_i (Y - \Phi\Theta),$$

where $H_n = \frac{1}{n}\Psi^T\Psi$. Since we get matrices from these calculations we compare them in Frobenius-norm with random tie-breaking and decide about Θ as previously.

3.1. *Outer approximation for MIV-SPS*

Similarly as in the scalar case we are looking for an outer approximating ellipsoid in the following form:

$$\{\Theta : \|H_n^{-1/2} V_n(\Theta - \Theta_{IV})\|_F^2 < r\}.$$

Again we can find the radius by solving $m - 1$ convex optimization problem defined by $\Phi, \Psi, Y, \hat{\Theta}_{IV}$ and W_i.

4. Numerical Experiments

4.1. *Comparing Different Methods of Generating Instrumental Variables for Scalar ARX system*

I would like to note that the methods described above are general enough to apply them on closed-loop systems as well but during this semester I only worked with open-loop ARX systems. For the experiments detailed futher I used the Python programming language and packages Numpy, CVXPy and Matplotlib.

In this first segment I would like to compare two ways of generating instrumental variables for a scalar problem in the following form:

$$Y_t = a^* Y_{t-1} + b^* U_{t-1} + N_t$$
$$U_t = c^* U_{t-1} + V_t,$$

where we aim to estimate the real parameters $\theta^* = [a^*, b^*]$ and give confidence intervals around them. For using SPS method we need the IV estimate but how can we get the IVs for that?

The first option I experimented with was using the least squares estimate of θ^*, build the predicted noise-free outputs with $\hat{\theta}$ and replace the original outputs in the regressors with this $\hat{Y}$:

$$\psi_t = [\hat{Y}_t, U_t].$$

The second version is more simple because it does not require us to calculate the least squares estimate, we just use the previous inputs:

$$\psi_t = [U_{t-1}, U_t].$$

In the folllowings I generated an $n-$long trajectory from the ARX system with $a^* = 0.7, b^* = 1, c^* = 0.75$, N as i.i.d. Gaussian and V as i.i.d. Laplacian noise. I implemented the outer approximation and stored the length of the confidence intervals given to the next, $n + 1$-th step by the two methods, IV_1 and IV_2. The figures below show the histogram of the results after 500 repeats, for $n = 100$ and 200. We can see that as we increase the size of the training data, the tails of the confidence interval lengths distribution becomes less heavy in both methods. However, IV_1 tends to be better which is not unexpected since we use the LS estimate in that method.

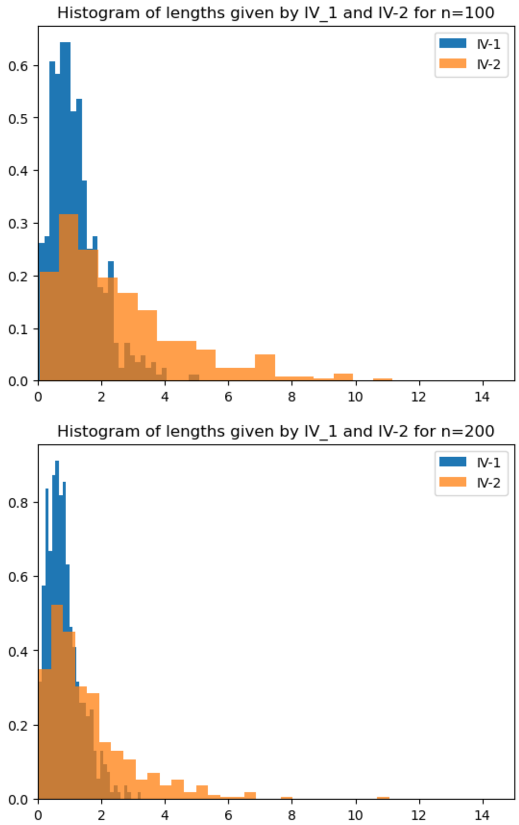

Figure 1: Lengths of the confidence intervals given by IV_1 and IV_2 for $n = 100, 200$

It is also interesting to take a look at the confidence region in the parameter space while increasing the sample size.In the figure below we can see an illustration of the outer approximated confidence regions in the parameter space. In this experiment I trained the model on a growing part of the series, from 100, 130, 160, 190, 220 and 250 outputs.

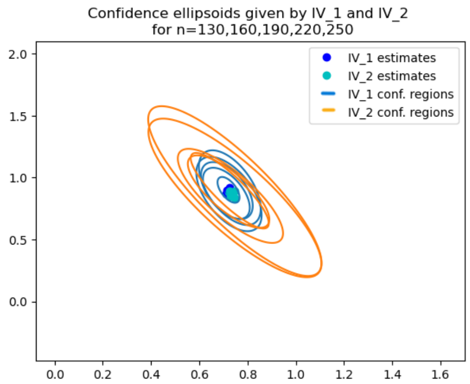

Figure 2: Confidence ellipsoids of IV_1 and IV_2 in the parameter space with increasing sample size

Again, in line with the previous results, we can see that the first model performs better. But is it also noticeable that for both models the size of the ellipses decreases with increasing sample size, as expected.

4.2. *Comparing SPS and PEM on ARX system*

In 2.5 PEM is described as a method that uses the permutations of the noise sequence to generate perturbed datasets. It can be used in the case of an exchangeable noise sequence and does not require the noise distribution to be symmetric. In order to compare this method with SPS we can look at a case when both of these conditions are met: let the noise, $\{N_t\}$ be an i.i.d. sequence of variables from Gaussian distribution.

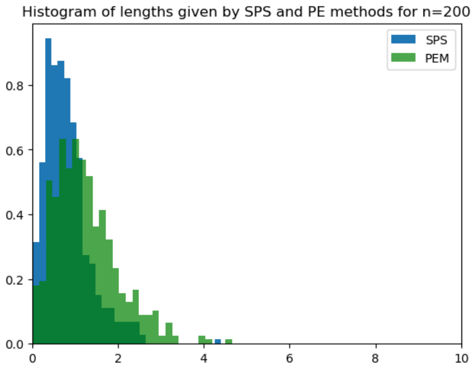

Figure 3: Lengths of SPS and PEM confidence intervals given for n=200

For the IV generation I used the first method, IV_1 and the IV-estimate we get from that to build the confidence interval around. After translating the outer approximation problem to the general framework with the W_i and P_i matrices I stored the lengths of the confidence intervals for the next observation after 200 steps. The histogram of these lengths are illustrated in Figure 3 after 500 repeats.

Examining the confidence ellipsoids in the parameter space we can notice that since we used the same IVs in both methods, the only difference will be the radius of the ellipsoids but the shape of the two will be the same in each step. In Figure 4 we can see an example where SPS stays better in each step but I would like to notice that during the experiments it occurred several times that even though PEM gave wider regions for smaller sample sizes it became better than SPS as the size increased.

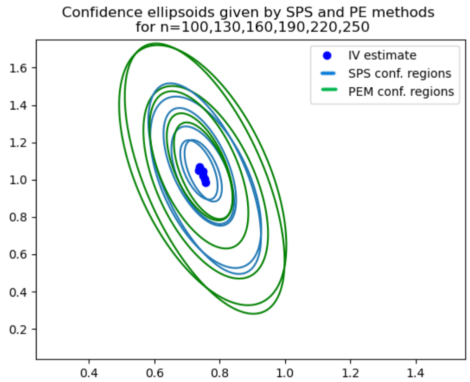

Figure 4: Confidence ellipsoids of SPS and PEM in the parameter space with increasing sample size

5. Summary and future work

During this semester I examined both scalar and vector-variate ARX system problems and different directions of generalizing SPS method for these cases. I experimented with different methods of generating instrumental variables and how well they worked compared to each other. Using the general framework of Perturbed Datasets Method I implemented PEM for systems with exchangeable noise sequence and compared it with SPS in the case of i.i.d. Gaussian noise. I also implemented the MIV-SPS method and I am planning to work on this further in the future as well as examine closed-loop systems.

References

[1] G. Box and G. Jenkins. *Time Series Analysis: Forecasting and Control*. Holden-Day series in time series analysis and digital processing. Holden-Day, 1970.

[2] B. Cs. Csáji, M. C. Campi, and E. Weyer. Sign-perturbed sums: a new system identification approach for constructing exact non-asymptotic confidence regions in linear regression models. *IEEE Trans. Signal Process.*, 63(1):169–181, 2015.

[3] S. Kolumbán, I. Vajk, and J. Schoukens. Perturbed datasets methods for hypothesis testing and structure of corresponding confidence sets. *Automatica J. IFAC*, 51:326–331, 2015.

[4] L. Ljung. *System Identification: Theory for the User*. Prentice-Hall, Upper Saddle River, 2nd edition, 1999.

[5] S. Szentpéteri and B. Cs. Csáji. Non-asymptotic state-space identification of closed-loop stochastic linear systems using instrumental variables. *Systems Control Lett.*, 178:Paper No. 105565, 11, 2023.

[6] V. Volpe, B. Cs. Csáji, A. Carè, E. Weyer, and M. C. Campi. Sign-perturbed sums (SPS) with instrumental variables for the identification of ARX systems. In *54th IEEE Conference on Decision and Control (CDC 2015)*, pages 2115–2120. IEEE, Osaka, 2015.