

# A Novel Sensitivity Analysis for Dynamic Models

Ahmed F. Yehia<sup>a</sup>, Mohamed Saleh<sup>a</sup>, Ayman Taha<sup>b</sup> and Hisham El-Shishiny<sup>c</sup>

a Operations Research & Decision Support Department, Faculty of Computers and Information,  
Cairo University

b Information System Department, Faculty of Computers and Information, Cairo University

c Advanced Technology and Center for Advanced Studies, IBM Cairo Technology Development  
Center, Cairo – Egypt.

## Abstract:

*This paper proposes a novel statistical sensitivity analysis for dynamic models, which is based on an enhanced maximal information coefficient (MIC) method. We enhanced the MIC method to handle multivariate sensitivity analysis; rather than just univariate analysis. The main motivation of this enhancement is overcoming the research gaps in the current state of the art of eigenvalue analysis. We postulate that this novel sensitivity analysis represents a solid foundation to study the multivariate complex nonlinear non-monotonic relationships between behavior modes – expressed by eigenvalues – and the model parameters. The experiments conducted corroborate our postulation.*

**Keywords:** *Sensitivity Analysis, Eigenvalue Analysis, and Maximal Information Coefficient.*

---

## 1. Introduction and Theoretical Background:

Eigenvalue analysis studies the sensitivity of behavior modes (eigenvalues) – or their associated weights – to the gains of feedback loops. In general, the gains of feedback loops are functions of the gains of links; which in turn are functions of parameters, in the model. Hence one can link parameters (independent variables) to a certain eigenvalue (the dependent variable). Eigenvalues are computed from the Jacobian matrix  $\mathbf{J}$ . Moreover, for each eigenvalue,  $\lambda_i$ , there is a distinct sensitivity matrix,  $\mathbf{S}_i$ , which is equal to the product of the left-eigenvector and the transpose of the right-eigenvector.

$$S_i = \begin{bmatrix} \frac{\partial \lambda_i}{\partial J(1,1)} & \cdot & \cdot & \frac{\partial \lambda_i}{\partial J(1,n)} \\ \cdot & \cdot & \cdot & \cdot \\ \frac{\partial \lambda_i}{\partial J(n,1)} & \cdot & \cdot & \frac{\partial \lambda_i}{\partial J(n,n)} \end{bmatrix}$$

The majority of previous works in this area (Forrester 1983; Güneralp 2006; Kampmann and Oliva 2008; Gonçalves 2009; Saleh, Oliva et al. 2010; etc...) utilized the traditional univariate sensitivity (or elasticity) measure, which is based on the first order partial derivative of the eigenvalue with respect to an independent variable. This univariate measure represents only the marginal contribution of the independent variable – assuming that all other independent variables are constants. However, as Sterman put it: "*In nonlinear systems, the sensitivity of a system to variations in multiple parameters is not a simple combination of the response to the parameters varied alone*" (Sterman, 2000, p.561).

Some scholars adopted multivariate linear regression analysis (Hekimoğlu and Barlas 2010; Tøndel, Vik et al. 2012). Regression analysis is under the umbrella of multivariate analysis; i.e. can study simultaneous changes in more than one parameter (Esbensen, Guyot et al. 2002). Linear regression analysis aims to find a linear relationship between the dependent variable and independent variables. The core algorithm of linear regression is the least squares errors algorithm, which is used in data fitting (Rencher and Christensen 2012). Yet, the main limitation here is the assumption of linear relationship between the dependent variable and independent variables.

Nonlinear regression analysis can be used when linear regression fails. However, the analyst must try different functions; e.g. exponential, logarithmic, etc ... In addition initial values for the coefficients are needed; and in general, the solution changes according to initial settings. That is,

there is no guarantee to reach the global minimum (least squares errors); as there might be several local minima. Moreover, the solution of complex nonlinear regression equations might not converge. Finally, nonlinear multivariate regression is not well suited to handle non-monotonic functions (Rencher and Christensen 2012).

To finalize this section, we present the summary table about different methods of sensitivity analysis, provided by Bier (Bier, 2011)

<b>Method</b>	<b>What is measured</b>	<b>Comparison and implications</b>	<b>Limitations</b>
<i>Scatterplots</i>	Subjective relationship between inputs and outputs	<ul style="list-style-type: none"> <li>•Good first method for identifying patterns</li> <li>•No very obvious patterns</li> </ul>	No quantitative measure
<i>Correlation Coefficients</i>	Strength of linear (or monotonic) relationship	<ul style="list-style-type: none"> <li>•Useful in ranking inputs</li> <li>•Static results similar for different types of CC, dynamic results differed for one input</li> </ul>	univariate analysis; i.e. not multivariate sensitivity analysis
<i>Stepwise Regression</i>	Coefficients for linear model that best predicts output	<ul style="list-style-type: none"> <li>•Most inputs were significant</li> <li>•Results similar to correlation coefficients</li> </ul>	Assumes linearity
<i>Elementary Effects</i>	Average derivative when one input is perturbed over different points in its domain	<ul style="list-style-type: none"> <li>•Little variation in <math>\mu^*</math> between inputs</li> </ul>	univariate analysis that only measures the marginal contribution; it does not measure the strength of the relationship (i.e. the total contribution)
<i>Sensitivity Indices</i>	Proportion of output variance attributed to input variance	Interactions were significant Especially important at the beginning of the simulation	only measures the marginal contribution; it does not measure the strength of the relationship (i.e. the total contribution)

Table 1: Different methods of sensitivity analysis; Source: (Bier, 2011, p.17)

The solution proposed, in this paper, overcomes all the above limitations; i.e. it is a multivariate sensitivity analysis that simultaneously measures the total contribution of several variables. The solution proposed is the enhancement of the MIC method to handle multivariate sensitivity analysis; rather than just univariate analysis. This will be explained further in the next section.

The rest of this paper is organized as follows: In section 2, we explain the proposed solution. In section 3, we illustrate the experiments conducted to test the proposed solution. Finally, we conclude, in section 4.

## 2. Proposed Solution

Since 2011, when Reshef et al. introduced the *Maximal Information Coefficient* (MIC), many researchers utilized it to detect any complex nonlinear and non-monotonic relationship between any two variables (Lin, Canhao et al. 2012; Zhang and Zhang 2012; Kim and Davis 2013). MIC outperforms other methods in terms of generality and equitability (Reshef, Reshef et al. 2011). Generality indicates the ability to detect a wide range of associations both functional and not; while equitability assures that similar scores are given to equally noisy relationships of different types. MIC is a univariate method that is based on the information entropy concept (Han and Kamber 2006; Saleh 2007; Gray 2011). MIC can detect all forms of associations between any pair of variables (pairwise relationship).

Reshef et al. propose the *Maximal Information Coefficient* (MIC) to measure the relationship between two variables. For a dataset  $D$  containing  $n$  observations on two variables  $X$  and  $Y$ , the values in  $X$  and  $Y$  can be partitioned into  $x$ -bins and  $y$ -bins, respectively, to create  $x$  by  $y$  grid. There are many possibilities of partitioning  $D$  into grid structure. Let  $D/G$  be the distribution of observations in  $D$  on the cells of a certain grid  $G$ , and  $I(D/G)$  denotes the mutual information of  $D/G$ . For a fixed  $D$ , different grids  $G$  results in different distributions  $D/G$ . Reshef et al. define the characteristic matrix of  $D$ ,  $M(D)_{x,y}$  as follows:

$$M(D)_{x,y} = \frac{I^*(D, x, y)}{\log \min(x, y)}$$

Where  $I^*(D; x; y) = \max (I (D/G))$  over all grids  $G$  with  $x$  columns and  $y$  rows. The characteristic matrix is then used in the calculation of MIC as

$$\text{MIC}(D) = \max_{x*y < B(n)} (MD)_{x,y}$$

Where  $B(n)$  is the upper bound on the grid size

In other words, MIC transforms the measuring association problem into a dynamic programming optimization problem; i.e. it dynamically searches for the grid distribution that maximizes the mutual information of the dataset.

Fortunately, MIC has the following advantages:

1. MIC can measure linear and nonlinear non-monotonic relationships between any two variables.
2. MIC computes association between continuous, categorical or mixed variables (where one variable is continuous and the other is categorical).
3. MIC takes values in the interval  $[0; 1]$ . A value of zero means there is no association between X and Y. However, a value of 1 means there is a perfect relationship between X and Y.
4. MIC is symmetric, that is,  $MIC(X, Y) = MIC(Y, X)$ .

Nevertheless, MIC suffers from the following disadvantages:

1. MIC is a novel association measure between a pair of variables. However, it could not be used to measure relationships among multiple variables.
2. MIC has a computational complexity problem; as the search space grows exponentially with the size of grid. For this reason, MIC must limit the size of grid.
3. MIC is sensitive to parameter values (used internally, in the MIC algorithm)

The core contribution of this paper is enhancing the MIC to become multivariate. We will denote the original MIC (developed by Reshef et al., in 2011) by  $MIC_1(Y, X_1)$ .  $MIC_1(Y, X_1)$  measures the association (dependency) level between the two variables Y and  $X_1$ ; and gives a score between 0 and 1. Now, in this paper, we define  $MIC_2(Y, X_1, X_2)$  as follows:

$$MIC_2(Y, X_1, X_2) = E[MIC_1(Y, X_1) | X_2]$$

The above expected conditional formula measures the dependency level between Y and  $X_1$  given the available information (values) about  $X_2$ ; i.e.  $MIC_2(Y, X_1, X_2)$  is the result of averaging  $MIC_1(Y, X_1) |_{X_2=x_2}$  over all possible values  $x_2$  that  $X_2$  may take. In general, the four steps procedure to compute  $MIC_2$  is as follows:

- I. Sort the variables Y,  $X_1$ ,  $X_2$  according to  $X_2$ .
- II. Split the sorted data into bins of equal percentile range (with respect to  $X_2$ ).
- III. Compute  $MIC_1(Y, X_1)$  in each bin.
- IV.  $MIC_2(Y, X_1, X_2)$  is equal to the average  $MIC_1$  score over all bins.

Similarly, in this paper, we define  $MIC_3(Y, X_1, X_2, X_3)$  as follows:

$$MIC_3(Y, X_1, X_2, X_3) = E[MIC_2(Y, X_1, X_2) | X_3]$$

And in general, we define  $MIC_n(Y, X_1, X_2, \dots, X_n)$  as follows:

$$MIC_n(Y, X_1, X_2, \dots, X_j, \dots, X_{n-1}, X_n) = E[MIC_{n-1}(Y, X_1, X_2, \dots, X_j, \dots, X_{n-1}) | X_n]$$

Where  $j = 3, \dots, n-2$

In Appendix A, we list the Matlab code of  $MIC_1$  function. Note that this function is a Matlab wrapper for the original MIC (developed by Reshef et al., in 2011). In Appendix B, we list the Matlab code of  $MIC_2$  function. And in Appendix C, we list the Matlab code of  $MIC_3$  function.

In practice,  $MIC_n$  should be computed in a stepwise fashion (like stepwise regression). The process should begin by determining which input, by itself, would lead to the highest  $MIC_1$  value. Then after the first, most important, input is found; the next most significant input is searched for and added to the model -- i.e. the one which would lead to the highest  $MIC_2$  value. The process goes on until the  $MIC_n$  value would not be significantly improved by adding any of the remaining inputs.

We postulate that by applying  $MIC_n$  to eigenvalue analysis, we can overcome the research gaps identified in the previous section. The experiments conducted, in the next section, corroborate our postulation.

Finally, we define the sensitivity of the eigenvalue,  $\lambda_i$ , to the parameter  $X_q$  as follows:

$$S(\lambda_i, X_q) = MIC_n(\lambda_i, X_1, X_2, \dots, X_j, \dots, X_{n-1}, X_n) - MIC_{n-1}(\lambda_i, X_1, X_2, \dots, X_k, \dots, X_{n-1}, X_n)$$

Where  $j = 3, \dots, n-2$

$$k = 3, \dots, n-2 \ \& \ k \neq q$$

The above sensitivity indicator measures the reduction in the association level (i.e. the MIC value), which occurs when removing parameter  $X_q$  from the set of input variables.

### 3. Experiments

#### 3.1. The Experimental Plan

This section presents the comparative experiments we conducted to test our proposed solution. Specifically, we compare MIC<sub>n</sub> vs. multiple linear regression. In the rest of this section, MREG<sub>n</sub> will be used as an abbreviation for multiple linear regression with "n" independent (input) variables.

The experimental plan is as follows:

- Test the following number of input variables: 1, 2 & 3 (i.e. X<sub>1</sub>, X<sub>2</sub> & X<sub>3</sub>)
- X<sub>1</sub>, X<sub>2</sub> & X<sub>3</sub> are uniform random variables [0,1]
- Test various correlation levels between the input variables
- Test various sample sizes. As specified by the following equation, the number of samples of MIC<sub>n</sub> “#Samples\_MIC<sub>n</sub>” is linked to both the number of samples of MIC<sub>1</sub> “#Samples\_MIC<sub>1</sub>” and the number of bins “#Bins”. Recall, from the previous section, that the number of bins is a parameter required for MIC<sub>n</sub>. Moreover, from our various experiments, we concluded that the sufficient sample size of the original MIC<sub>1</sub> “#Samples\_MIC<sub>1</sub>” varies with the complexity of the relationship; however, in general, it has to be greater than 10.

$$\#Samples\_MIC_n = \#Samples\_MIC_1 * \#Bins^{n-1}$$

$$\text{i.e. } \#Bins = \sqrt[n-1]{\frac{\#Samples\_MIC_n}{\#Samples\_MIC_1}}$$

- The measures of dependency (between the output variable and the input variables) are as follows:
  - R<sup>2</sup>, in the case of MREG<sub>n</sub>
  - The MIC score, in the case of MIC<sub>n</sub>.
  - Note that “unity” is the best value for both MIC<sub>n</sub> and MREG<sub>n</sub> (i.e. full dependency)
  - While “zero” is the worst value for both methods (i.e. no dependency)

The rest of section 3 is organized as follows: In section 3.2, we conduct experiments on various nonlinear algebraic equations. Note that we have also tested several multidimensional linear equations; where we found that, in all cases, the measures of dependency for both MIC<sub>n</sub> and MREG<sub>n</sub> are equal unity (or very close to it). Finally, in section 3.3, we develop a simple

hypothetical dynamic model; then test the sensitivity of the dominant eigenvalue (associated with this simple model) to the input parameters (in the model). Note that in this model (for simplicity), the eigenvalues are always real numbers (i.e. not complex numbers) – whatever the values of the parameters.

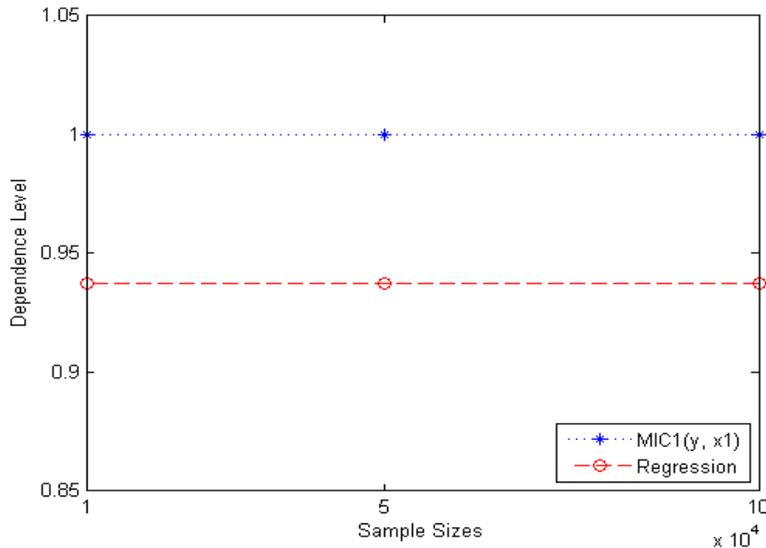
### 3.2. Experiments on Nonlinear Algebraic Equations

In this section, we describe the five experiments that we conducted on several nonlinear algebraic equations.

The first experiment has the following settings:

- 1 input variable
- The nonlinear equation used in the experiment is:  $y = 2 * x_1^2$
- Sample sizes: 10k, 50k & 100k (k=1000)
- Notice that there is no correlation between the input variables, in this experiment (as there is only one input variable)

Results of the first experiment are shown in figure 1.  $MIC_1$  outperformed  $MREG_1$ , for this simple nonlinear equation; i.e. the dependency score associated with  $MIC_1$  is greater than the dependency score associated with  $MREG_1$ . In fact, the curve of  $MIC_1$  is a horizontal line at 1 (i.e. full dependency).



**Fig. 1:** Results of Experiment #1 –  $MIC_1$  vs.  $MREG_1$

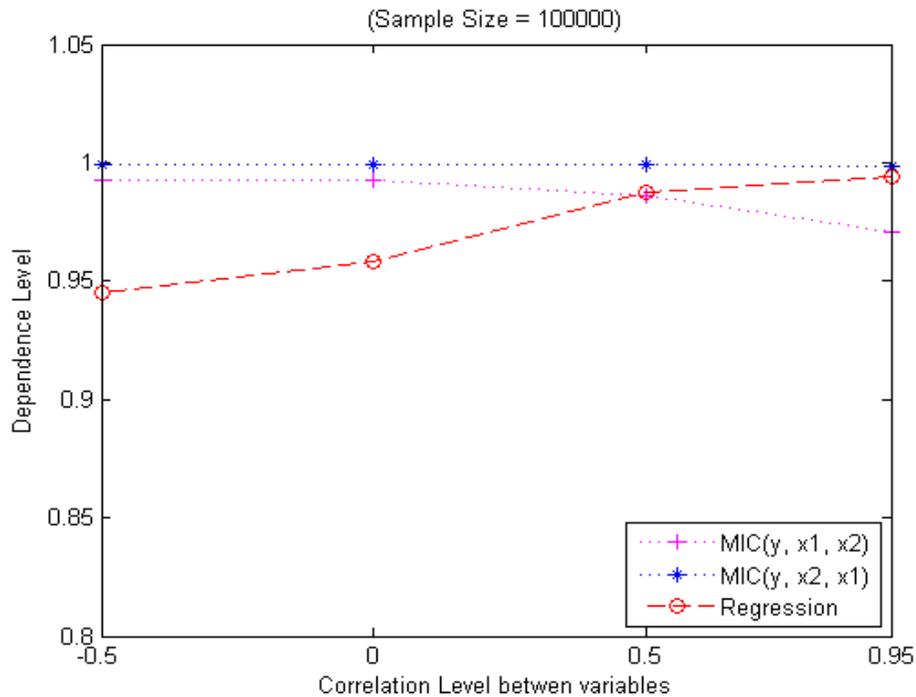
The second experiment has the following settings:

- 2 input variables
- The equation used:  $y = x_1^2 * x_2 + \sqrt{10 * x_2}$

This equation was designed to make  $x_2$  more significant than  $x_1$ ; recall that  $x_1$  is a random fraction; therefore  $x_1^2$  is always less than  $x_1$

- The sample size is fixed to 100k.
- Test various correlation levels between the input variables

Results of the second experiment are shown in figure 2. The curve of  $MIC_2(Y, X_2, X_1)$  is a horizontal line at 1 (i.e. full dependency).  $MIC_2(Y, X_2, X_1)$  slightly outperformed  $MIC_2(Y, X_1, X_2)$ ; because  $x_2$  is more significant than  $x_1$ . As we stated in section 2,  $MIC_n$  should be computed in a stepwise procedure to detect the best order for the input variables. This stepwise procedure will be tested in section 3.2 (i.e. the experimentations on the simple dynamic model). Despite the fact that  $MIC_2(Y, X_1, X_2)$  is slightly inferior to  $MIC_2(Y, X_2, X_1)$ ; nevertheless,  $MIC_2(Y, X_1, X_2)$  outperformed  $MREG_2$ , except for high correlation values. As when the correlation value (between the two input variables) is high, the two input variables tend to be almost the same variable. So, in these cases,  $MREG_2$  is simplified to  $MREG_1$ . This explanation is not applicable to  $MIC_2$ , because it is calculated via  $MIC_1$  (which only processes one input variable).

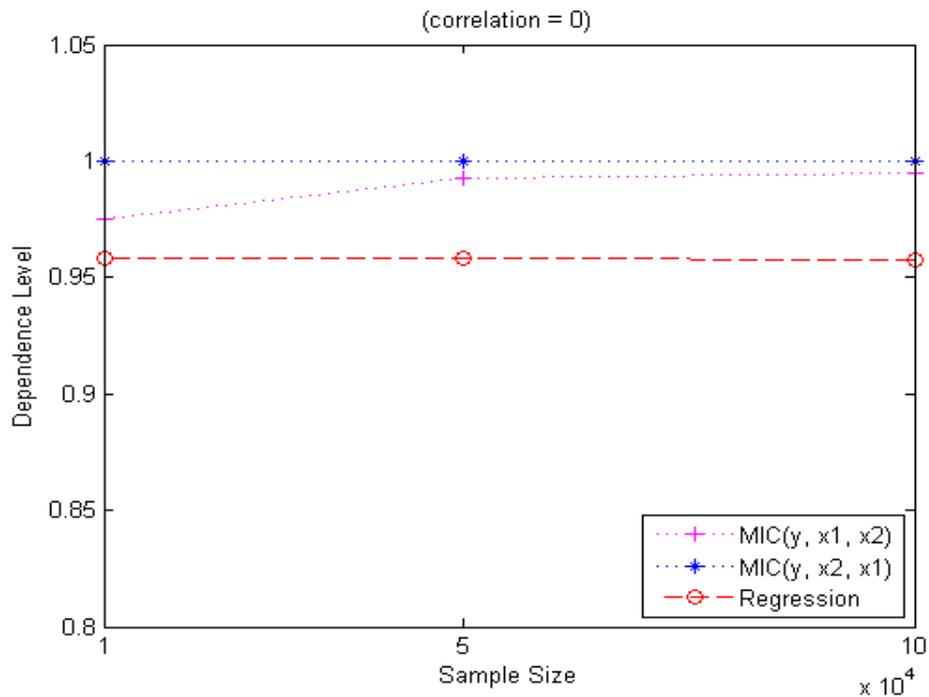


**Fig 2:** Results of Experiment #2 –  $MIC_2$  vs.  $MREG_2$  – Fixed Sample Size.

The third experiment has similar settings to the second experiment; the differences are as follows:

- The correlation level between the input variables is fixed to 0
- Sample sizes: 10k, 50k & 100k

Results of the third experiment are shown figure 3. Again the curve of  $MIC_2(Y, X_2, X_1)$  is a horizontal line at 1.  $MIC_2(Y, X_2, X_1)$  slightly outperformed  $MIC_2(Y, X_1, X_2)$ , which in turn outperformed  $MREG_2$ .



**Fig 3:** Results of Experiment #3 –  $MIC_2$  vs.  $MREG_2$  – Fixed Correlation Level

The fourth experiment has the following settings:

- 3 input variables
- The equation used is:  $y = x_1 * x_2 * x_3 + \sqrt{x_1} * x_3$

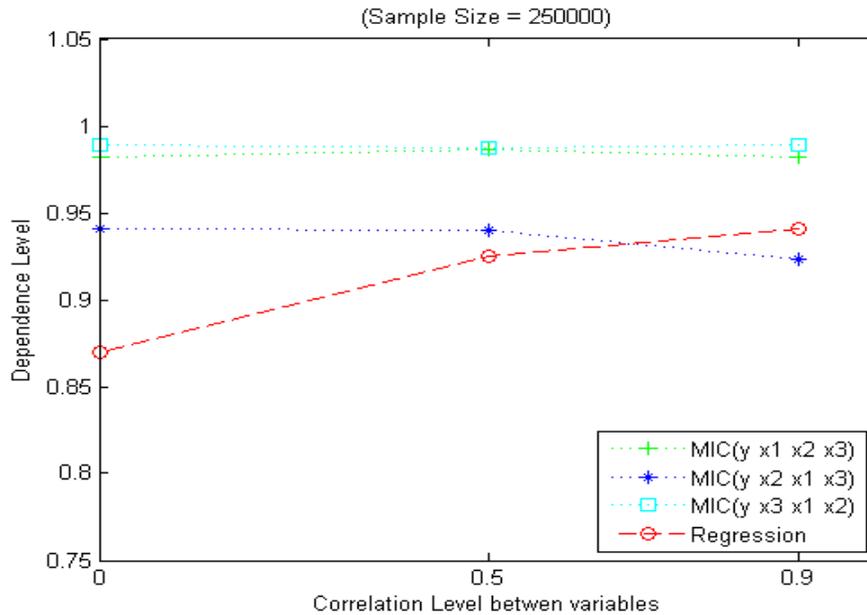
This equation was designed to make  $x_2$  the least significant input variable.

- The sample size is fixed to 250k.
- Test various correlation levels between the input variables

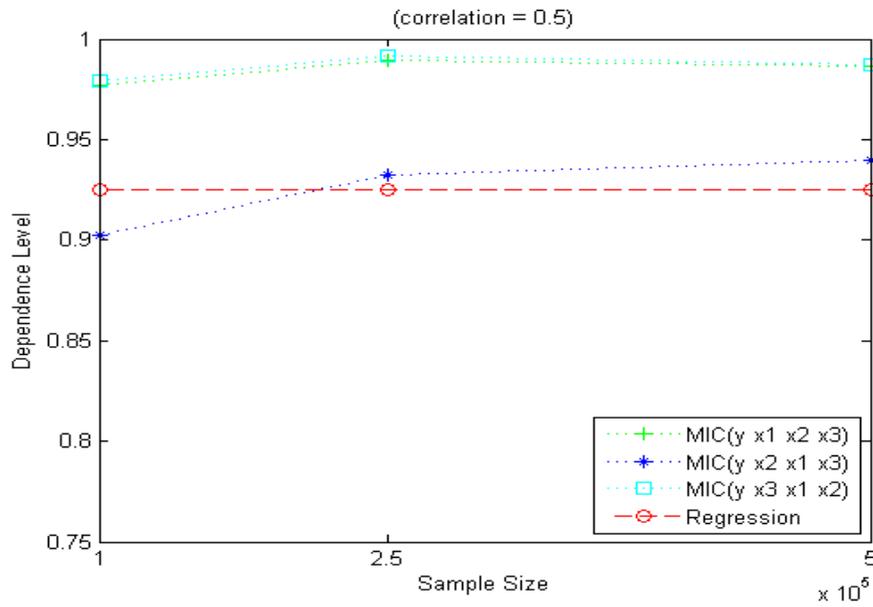
The fifth experiment has similar settings to the fourth experiment; the differences are as follows:

- The correlation level between the input variables is fixed to 0.5
- Sample sizes: 100k, 250k & 500k

Results of the fourth experiment are shown in figure 4; while results of the fifth experiment are shown in figure 5.



**Fig 4:** Results of Experiment #4 – MIC<sub>3</sub> vs. MREG<sub>3</sub> – Fixed Sample Size



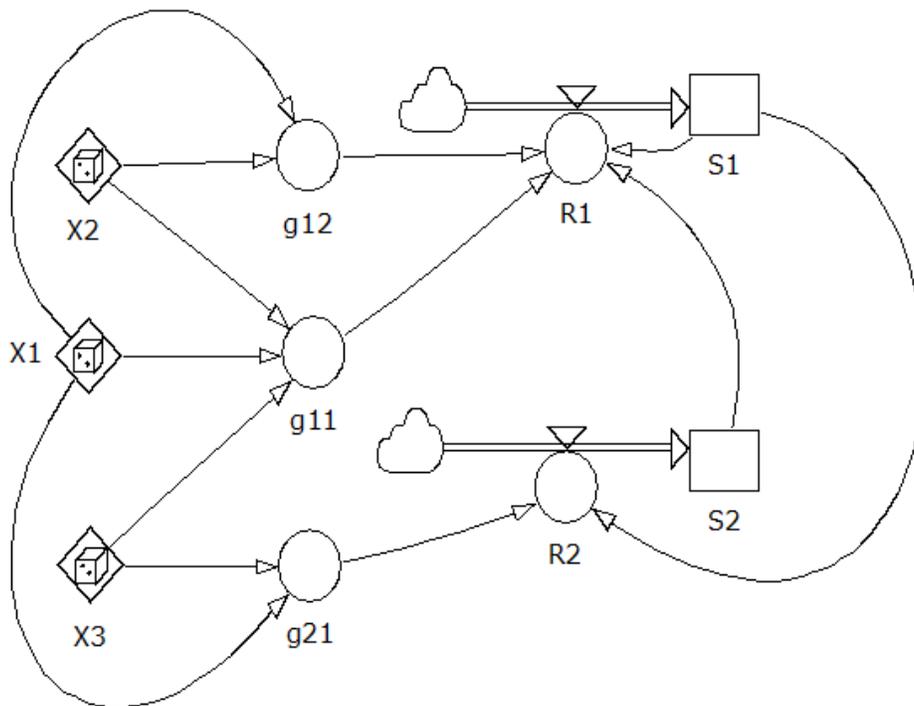
**Fig 5:** Results of Experiment #5 – MIC<sub>3</sub> vs. MREG<sub>3</sub> – Fixed Correlation Level

In both experiments (fourth & fifth),  $MIC_3(Y, X_1, X_2, X_3) \approx MIC_3(Y, X_3, X_1, X_2) \approx 1$ . And despite the fact that  $MIC_3(Y, X_2, X_1, X_3)$  is slightly inferior to the other two plotted  $MIC_3$  (because  $x_2$  is the least significant input variable); nevertheless, it outperformed MREG<sub>2</sub> – most of the time. These experiments illustrate the importance of computing  $MIC_n$  in a stepwise fashion, in order to assure obtaining the best results for  $MIC_n$ . This stepwise procedure will be demonstrated, in the next section.

### 3.3. Sensitivities of Eigenvalues

In this experiment, we used the simple hypothetical dynamic model shown in the following figure. The model has two stocks:  $S_1$  and  $S_2$ .  $R_1$  is the inflow associated with stock  $S_1$ ; and  $R_2$  is the inflow associated with stock  $S_2$ . Moreover, there are three auxiliary variables:  $g_{11}$ ,  $g_{12}$ ,  $g_{21}$ . The equations of the model are as follows:

- $X_1, X_2$  &  $X_3$  are uniform random variables  $[0,1]$
- $g_{11} = X_1 * X_2 * X_3$
- $g_{12} = X_1 * X_2$
- $g_{21} = X_1 * X_3$
- $R_1 = g_{11} * S_1 + g_{12} * S_2 + 1$
- $R_2 = S_1 * g_{21}$

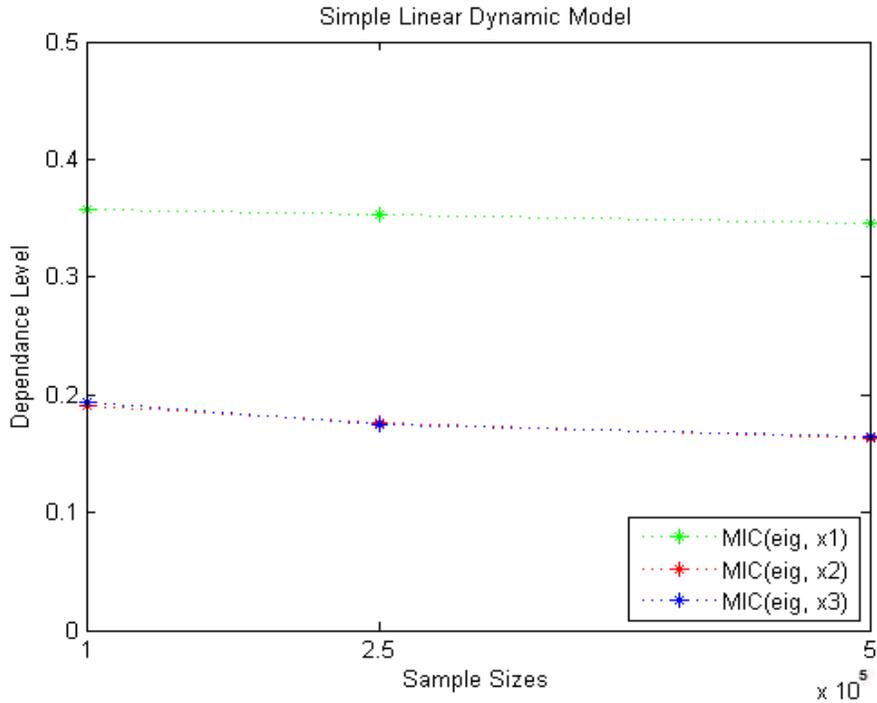


**Fig 6:** The Stock & Flow Diagram of the Model used in the Eigenvalue Experiment.

Given a set of specific values for the three random input variables ( $X_1$ ,  $X_2$  &  $X_3$ ), the two eigenvalues of this model can be computed via the Jacobian matrix. In this case, the two eigenvalues are always real numbers. Hence, the dependency between the dominant eigenvalue and the three input variables can be identified using either  $MIC_3$  or  $MREG_3$ . The settings of this experiment are as follows:

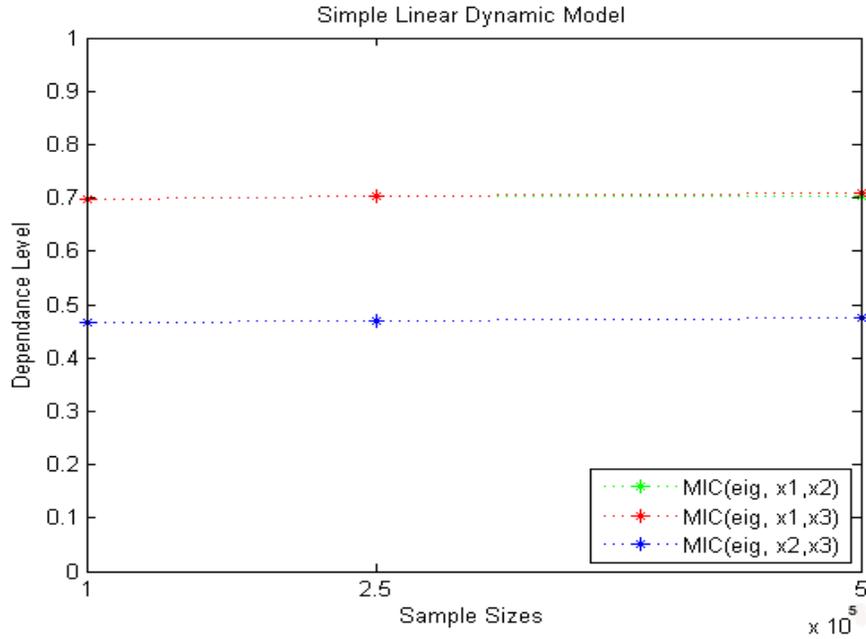
- The correlation level between the input variables is fixed to 0
- Sample sizes: 100k, 150k & 250k

To apply the stepwise procedure, we will start by plotting  $MIC_1$  associated with every input variable, as shown in the following figure; i.e.  $MIC_1(\lambda^*, X_1)$ ,  $MIC_1(\lambda^*, X_2)$  &  $MIC_1(\lambda^*, X_3)$  – where  $\lambda^*$  is the dominant eigenvalue.



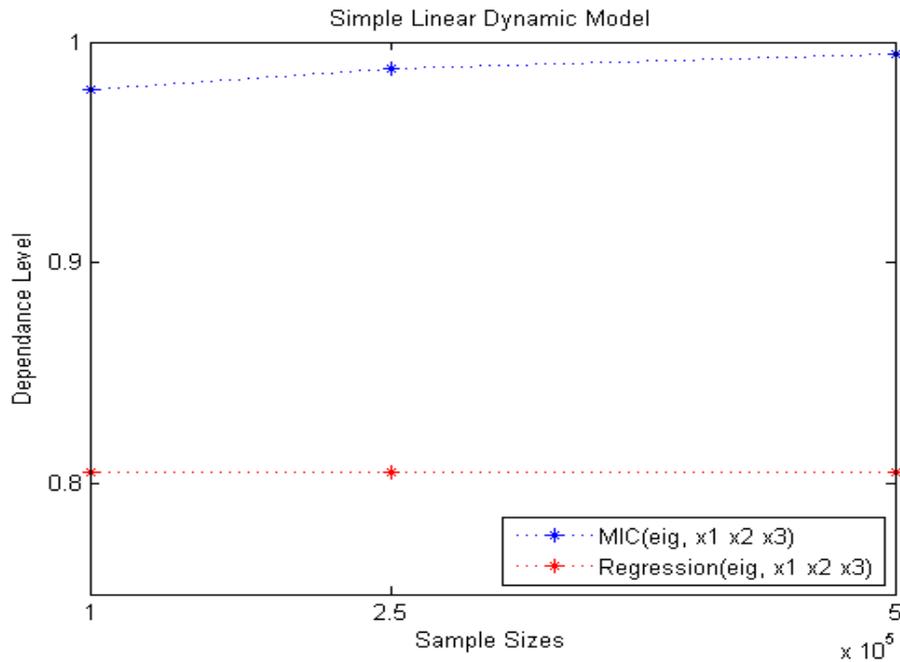
**Fig 7:**  $MIC_1$  between the dominant eigenvalue and input variables.

It is clear from the above graph that  $X_1$  has the highest impact on the dominant eigenvalue; while  $X_2$  and  $X_3$  have the same impact. Now, the coming step is to search for the next most significant input -- i.e. the one which would lead to the highest  $MIC_2$  value. Hence, in the following figure, we plotted  $MIC_2(\lambda^*, X_1, X_2)$  vs.  $MIC_2(\lambda^*, X_1, X_3)$ ; moreover, we plotted  $MIC_2(\lambda^*, X_2, X_3)$  to make sure that it is an inferior option.



**Fig 8:** MIC<sub>2</sub> between the dominant eigenvalue and input variables.

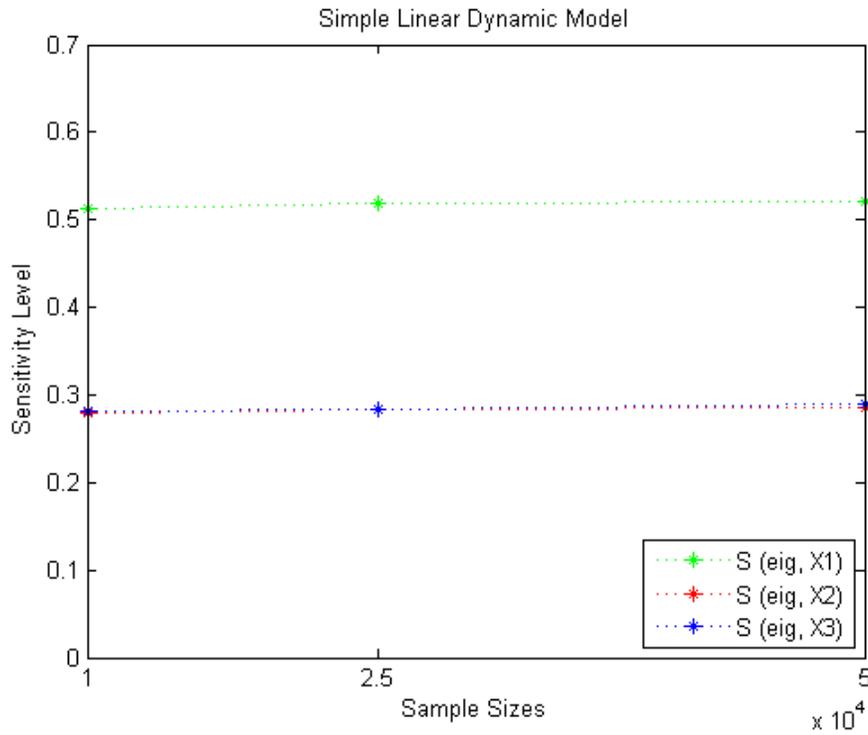
It is clear from the above graph that  $MIC_2(\lambda^*, X1, X2) \approx MIC_2(\lambda^*, X1, X3)$ . Hence, we are free to choose any order; arbitrarily, we selected the "X1, X2" order. The next step is to add the only missing input variable – i.e. X3. The following figure shows  $MIC_3(\lambda^*, X1, X2, X3)$  vs. MREG<sub>3</sub>.



**Fig 9:** MIC<sub>3</sub> between the dominant eigenvalue and input variables.

It is clear, from the above graph, that  $MIC_3$  significantly outperformed  $MREG_3$ . Moreover,  $MIC_3 \approx 1$ .

The last step is to compute the sensitivity of the dominant eigenvalue to each input variable. This can be done via subtracting the corresponding  $MIC_2$  from  $MIC_3$  – as specified in section 2. The various sensitivities are plotted, in the following figure.



**Fig 10:** Sensitivity of the dominant eigenvalue to each input variable

It is clear, from the above graph, that the dominant eigenvalue is most sensitive to X1; while X2 and X3 have the same impact on the dominant eigenvalue.

## 4. Conclusion

In this paper, we developed a novel sensitivity analysis for dynamic models, which is based on an enhanced maximal information coefficient (MIC) method. This novel sensitivity analysis overcomes the research gaps associated with rival methods. Via this novel analysis, decision-makers can rank policy parameters according to their impacts on the dominant eigenvalue.

The core contribution, in this research, is the enhancement of the MIC to handle multivariate analysis; rather than just univariate analysis. This enhancement is useful in many domains beyond simulation models. The main caveat is making sure that the required sample size is obtainable. Recall that the required sample size grows exponentially with the dimension of the multivariate MIC. In many real life applications, obtaining the required sample size may not be possible. However, in simulation models, generating a large number of samples is a normal practice. More importantly, there is no need to simulate the model. The proposed solution only interacts with a condensed matrix representation of the model; i.e. the Jacobian matrix. In each run, the process, which takes time, is the computation of eigenvalues from the Jacobian matrix; and there are algorithms that compute eigenvalues very fast.

In the near future, we will continue the experimental work; and investigate more the time complexity of the proposed solution.

Finally, in many nonlinear dynamic models, eigenvalues depend on the current state of the model. For these cases, we plan to devise an innovative framework that links the time trajectory of the dominant eigenvalue with parameters.

## ACKNOWLEDGEMENT

This work is part of an R&D project funded by an IBM Faculty Award.

## References

- Bier, A. 2011. "Sensitivity Analysis Techniques for System Dynamics Models of Human Behavior". Proc. of the 29th Int. Conf. of the System Dynamics Society, System Dynamics Society, Washington, D.C.
- Esbensen, K., D. Guyot, et al. 2002. Multivariate data analysis-in practice: An introduction to multivariate data analysis and experimental design, Multivariate Data Analysis.
- Forrester, N. 1983. Eigen value analysis of dominant feedback loops. Plenary Session Papers Proceedings of the 1st International System Dynamics Society Conference.
- Gonçalves, P. 2009. "Behavior modes, pathways and overall trajectories: eigenvector and Eigen value analysis of dynamic systems." System dynamics review 25(1): 35-62.
- Gray, R. M. 2011. Entropy and information theory, Springer.
- Güneralp, B. 2006. "Towards coherent loop dominance analysis: progress in Eigen value elasticity analysis." System dynamics review 22(3): 263-289.
- Hadi, A. S. 1992. "Identifying multiple outliers in multivariate data." Journal of the Royal Statistical Society. Series B (Methodological): 761-771.
- Han, J. and M. Kamber 2006. Data mining: concepts and techniques, Morgan Kaufmann.
- Hekimoğlu, M. and Y. Barlas 2010. "Sensitivity Analysis of System Dynamics Models by Behavior Pattern Measures." Proc. of the 28th Int. Conf. of the System Dynamics Society, System Dynamics Society, Albany, NY.
- Jakulin, A. and I. Bratko 2003. "Quantifying and visualizing attribute interactions." arXiv preprint cs/0308002.
- Kampmann, C. E. and R. Oliva 2008. "Structural dominance analysis and theory building in system dynamics." Systems Research and Behavioral Science 25(4): 505-519.
- Kim, J. and C. Davis 2013. "Comparing the consistency and distinctiveness of speech produced in quiet and in noise." Computer Speech & Language.
- Lin, C., H. Canhao, et al. 2012. Maximal Information Coefficient for Feature Selection for Clinical Document Classification. ICML Workshop on Machine Learning for Clinical Data.
- Lutz, A. E., R. J. Kee, et al. 1988. "SENKIN: A FORTRAN program for predicting homogeneous gas phase chemical kinetics with sensitivity analysis." Sandia National Laboratories Report SAND87-8248.

- Pogaku, N., M. Prodanovic, et al. 2007. "Modeling, analysis and testing of autonomous operation of an inverter-based microgrid." *Power Electronics, IEEE Transactions on* 22(2): 613-625.
- Rencher, A. C. and W. F. Christensen 2012. *Methods of multivariate analysis*, John Wiley & Sons.
- Reshef, D. N., Y. A. Reshef, et al. 2011. "Detecting novel associations in large data sets." *science* 334(6062): 1518-1524.
- Saleh, M. 2007. "Estimating market shares in each market segment using the information entropy concept." *Applied Mathematics and Computation* 190(2): 1735-1739.
- Saleh, M., R. Oliva, et al. 2010. "A comprehensive analytical approach for policy analysis of system dynamics models." *European Journal of Operational Research* 203(3): 673-683.
- Sterman, J. 2000. *Business dynamics*, Irwin-McGraw-Hill.
- Taylor, T. and D. N. Ford 2006. "Tipping point failure and robustness in single development projects." *System Dynamics Review* 22(1): 51-71.
- Tøndel, K., J. O. Vik, et al. 2012. "Hierarchical multivariate regression-based sensitivity analysis reveals complex parameter interaction patterns in dynamic models." *Chemometrics and Intelligent Laboratory Systems*.
- Zhang, Y. and W. Zhang. 2012. A Novel Bayesian Network Structure Learning Algorithm based on Maximal Information Coefficient. *Proceedings of the Fifth International Conference on Advanced Computational Intelligence*. IEEE.

## Appendix A: Matlab Code for MIC<sub>1</sub> Function

```
function result = MIC1(v1,v2)

    fid = fopen('file_MINE.csv','w+');
    fprintf( fid , '%s, %s\n' , 'v1' , 'v2');
    dlmwrite ('file_MINE.csv', [v1,v2], '-append');
    fclose('all');

    system('java -jar MINE.jar "file_MINE.csv" 0');
    % Download MINE.jar from http://is.gd/esajuq
    % You must also install the Java Runtime Environment (JRE)

    result = csvread('file_MINE.csv,mv=0,cv=0.0,B=n^0.6,Results.csv',1,2,[1,2,1,2]);

end
```

## Appendix B: Matlab Code for MIC<sub>2</sub> Function

```
function result = MIC2(n_bins,y,x1,x2)

    M = [y x1 x2];
    Ms = sortrows(M,3);
    ys = Ms(:,1);
    x1s = Ms(:,2);

    len = length(ys);
    sum = 0;
    idx_start = 1;

    for i = 1:n_bins
        idx_end = round(i*len/n_bins);
        sum = sum + MIC1(ys(idx_start:idx_end),x1s(idx_start:idx_end));
        idx_start = idx_end+1;
    end

    result = sum/n_bins;

end
```

## Appendix C: Matlab Code for MIC<sub>3</sub> Function

```
function result = MIC3(n_bins,y,x1,x2,x3)

    M = [y x1 x2 x3];
    Ms = sortrows(M,4);
    ys = Ms(:,1);
    x1s = Ms(:,2);
    x2s = Ms(:,3);

    len = length(ys);
    sum = 0;
    idx_start = 1;

    for i = 1:n_bins
        idx_end = round(i*len/n_bins);
        sum = sum + ...
            MIC2(n_bins,ys(idx_start:idx_end),x1s(idx_start:idx_end), ...
                x2s(idx_start:idx_end));
        idx_start = idx_end+1;
    end

    result = sum/n_bins;

end
```