**Adaptive thresholding**

**in structure learning of a Bayesian network –**

**A software manual**

Michal Afek, Rafi Bojmel and Boaz Lerner

*Ben-Gurion University of the Negev*

*Introduction* This manual describes a procedure to learn a Bayesian network structure using the PC algorithm (Spirtes et al., 2000) and adaptive thresholding. To decide on independence between a pair of nodes (variables) in the graph conditioned on a set of other nodes, PC applies a threshold to test a statistical/information measure, e.g., conditional mutual information (CMI). Usually, this threshold is fixed (regardless of the problem and sample size), arbitrarily selected, and common to all tests (regardless of the order of the condition set and the cardinalities of the involved variables). We propose (Lerner et al., 2013) applying an adaptive threshold to each individual test and thereby eliminate the sensitivity of the test to the sample size, number of variables that participate in the test, and their cardinalities.

*Software Description* The procedure is based on the approach detailed in “Adaptive thresholding in structure learning of a Bayesian network”, which was published in IJCAI 2013 and may be downloaded from <http://www.ee.bgu.ac.il/~boaz/publications>.

The procedure uses codes of the BNT package developed by K. Murphy (Murphy, 2001), which may be downloaded from <https://code.google.com/p/bnt/>, and our own functions, which may be downloaded from <http://www.ee.bgu.ac.il/~boaz/software> . Following is a brief description of essential functions and their input-output of three types – C1, S1, and S2 (Lerner et al., 2013) – of adaptive thresholds (all are implemented as [Matlab](http://www.mathworks.com/products/matlab/) m-files):

1. learn\_struct\_pdag\_pc\_adaptive
	1. This function is a variant of the BNT’s learn\_struct\_pdag\_pc function. It learns a CPDAG (Chickering, 2002) based on a CMI test and an adaptive threshold.
	2. Input:
		1. Data: a data set that is used for the learning task
		2. Threshold\_flag: a number between 1 and 3 that indicates the selected adaptive threshold. 1 is for C1, 2 for S1 (alpha=0.01), and 3 for S2 (alpha=0.01) (see Lerner et al., 2013 for details)
		3. Threshold\_type: a number between 1 and 2 that indicates whether the threshold is calculated taking into consideration the condition set cardinality (2) or not (1)
	3. Output:
		1. The output is a CPDAG learned by PC with an adaptive threshold that is applied to all tests, which is similar to the output of learn\_struct\_pdag\_pc of the BNT toolbox
2. C1cond
	1. This function calculates the C1 threshold between variables X and Y given a condition set S (which may be empty) that will be applied in the CMI test for X and Y
	2. Input:
		1. Num\_x: column’s index of variable X in the dataset
		2. Num\_y: column’s index of variable Y in the dataset
		3. Num\_s: columns’ indices of the variables in the condition set S in the dataset
		4. Fulldata: the dataset
	3. Output:
		1. C1\_value: the value of the C1 adaptive threshold
3. S1cond
	1. This function calculates the S1 threshold between variables X and Y given a condition set S (which may be empty) that will be applied in the CMI test for X and Y
	2. Input:
		1. Num\_x: column’s index of variable X in the dataset
		2. Num\_y: column’s index of variable Y in the dataset
		3. Num\_s: columns’ indices of the variables in the condition set S in the dataset
		4. Fulldata: the dataset
		5. Alpha: the significance level used in the Pearson’s chi-square test
	3. Output:
		1. S1\_value: the value of the S1 adaptive threshold
4. S2cond
	1. This function calculates the S2 threshold between variables X and Y given a condition set S (which may be empty) that will be applied to the CMI test for X and Y
	2. Input:
		1. Num\_x: column’s index of variable X in the dataset
		2. Num\_y: column’s index of variable Y in the dataset
		3. Num\_s: columns’ indices of the variables in the condition set S in the dataset
		4. Fulldata: the dataset
		5. Alpha: the significance level used in the Pearson’s chi-square test
	3. Output:
		1. S2\_value: the value of the S2 adaptive threshold
5. mutual2\_decision
	1. This function calculates the CMI value (using the MI\_Log2 function) between variables X and Y given condition set S and makes a decision based on a threshold whether the variables are dependent or not.
	2. Input:
		1. Num\_x: column’s index of variable X in the dataset
		2. Num\_y: column’s index of variable Y in the dataset
		3. Num\_s: columns’ indices of the variables in the condition set S in the dataset
		4. Fulldata: the dataset
		5. Threshold: a threshold used to determine whether the variables are dependent or not
	3. Output:
		1. 1- in case the variables are independent (MI< threshold) based on the MI test, 0- otherwise
6. MI\_Log2
	1. This function calculates CMI between variables X and Y given condition set S
	2. Input:
		1. Num\_x: column’s index of variable X in the dataset
		2. Num\_y: column’s index of variable Y in the dataset
		3. Num\_s: columns’ indices of the variables in the condition set S in the dataset
		4. Fulldata: the dataset
	3. Output:
		1. CMI: The value of the calculated CMI

**References**

Chickering, D. (2002). Learning equivalence classes of Bayesian-network structures. *Journal of Machine Learning Research, 2*, 445–498.

Lerner, B., Afek, M. and Bojmel, R. (2013). [Adaptive thresholding in structure learning of a Bayesian network](http://www.ee.bgu.ac.il/~boaz/IJCAI2013LernerAfekBojmel.pdf), *23rd International Joint Conference on Artificial Intelligence* (*IJCAI-13*), Beijing, China, 1458-1464

Murphy, K. (2001). The Bayes net toolbox for Matlab. *Computing Science and Statistics, 33*, 331-350.

Spirtes, P., Glymour, C., & Scheines, R. (2000). *Causation, Prediction and Search* (2nd ed.). Cambridge: MIT Press.