A Kernel-based Causal Learning Algorithm

Xiaohai Sun XIAOHAI.SUN@TUEBINGEN.MPG.DE
Max Planck Institute for Biological Cybernetics, 72076 Tübingen, Germany

Dominik Janzing JANZING@IRA.UKA.DE
Institute for Algorithms and Cognitive Systems, Universität Karlsruhe (TH), 76128 Karlsruhe, Germany

Bernhard Schölkopf BERNHARD.SCHOELKOPF@TUEBINGEN.MPG.DE
Max Planck Institute for Biological Cybernetics, 72076 Tübingen, Germany

Kenji Fukumizu FUKUMIZU@ISM.AC.JP
Institute of Statistical Mathematics, 4-6-7 Minami-Azabu, Minato-ku, Tokyo 106-8569, Japan

Abstract

We describe a causal learning method, which employs measuring the strength of statistical dependences in terms of the Hilbert-Schmidt norm of kernel-based cross-covariance operators. Following the line of the common faithfulness assumption of constraint-based causal learning, our approach assumes that a variable Z is likely to be a common effect of X and Y, if conditioning on Z increases the dependence between X and Y. Based on this assumption, we collect "votes" for hypothetical causal directions and orient the edges by the majority principle. In most experiments with known causal structures, our method provided plausible results and outperformed the conventional constraint-based PC algorithm.

1. Introduction

Until the early nineties, it was widely considered impossible to discover causal structures in purely observational data without using any controlled experiments. The seminal works of Spirtes et al. (1993) and Pearl (2000) showed that, under reasonable assumptions, it is possible to get hints on causal relationships from non-experimental data. Their well-known approach for automatically generating causal hypotheses, formalized by a directed acyclic graph (DAG), is based on the Markov condition and the faithfulness assumption: Among all graphs that contain enough causal arrows to explain *all* statistical dependences, one prefers those structures which allow *only* the conditional dependences. The best known example based on these principles is the inductive causation (IC) algorithm, which consists of three main steps:

Step 1 Connect vertices $X - Y$ if and only if no set of variables S_{XY} (excluding X, Y) can be found with $X \perp\!\!\!\perp Y \,|\, S_{XY}$, i.e. X, Y are independent given all variables in S_{XY}.

Step 2 For each substructure $X - Z - Y$ (X and Y non-adjacent), orient the edges to $X \rightarrow Z \leftarrow Y$ (a so-called v-structure), if $Z \notin S_{XY}$.

Step 3 Orient as many of undirected edges as possible subject to the condition that neither a new v-structure nor a directed cycle should be created.

A refined version of IC is the PC algorithm (after its authors Spirtes and Glymour (1991)). However, if very few or no conditional independence relations are verified, IC would have little or no chance to orient the edges in Step 2. Another disadvantage of IC is the categorical (maybe erroneously) decisions in step 1 for independence will affect all the future algorithm behavior. In addition, testing independence, especially for continuous variables, is a problem currently unsolved in its generality. The usual implementation of PC uses the partial correlations for continuous domains under the assumption of multivariate normal distribution and χ^2 tests for categorical variables. This paper tries to elaborate on these problems. First, we argue that the *kernel-based statistical independence measure* is useful, since it does not have to impose any special kinds of distributions. Second, taking the *strength* of dependences into account, a probably overdetermination of dependence (deciding dependence when there is independence) will not be so crucial for the orientation in Step 2. This requires however an appropriate measure for the strength of dependences and a reliable method to compare the strengths of

Appearing in *Proceedings of the 24^{th} International Conference on Machine Learning*, Corvallis, OR, 2007.

conditional and unconditional dependences. For this purpose, we extend a dependence measure which is based on the Hilbert-Schmidt norm of cross-covariance operators to measuring *conditional* dependences.

2. Measuring Statistical Dependences with Kernels

The idea of measuring dependences by reproducing kernel Hilbert spaces (RKHS) (Aronszajn, 1950; Schölkopf & Smola, 2002) is that statistical dependences can always be detected by correlations after data are mapped into an appropriate feature space which is implicitly given by a kernel.

2.1. Cross-Covariance Operator and Independence

First, we introduce the cross-covariance operator (Baker, 1973) expressing correlations in the feature space and show its relation to independence of variables. Let $(\mathcal{X}, \mathcal{B}_\mathcal{X})$ and $(\mathcal{Y}, \mathcal{B}_\mathcal{Y})$ be measurable spaces and $(\mathcal{H}_\mathcal{X}, k_X), (\mathcal{H}_\mathcal{Y}, k_Y)$ be RKHSs of functions on $\mathcal{X}$ and $\mathcal{Y}$, with positive definite kernels k_X, k_Y. We consider random vector (X, Y) on $\mathcal{X} \times \mathcal{Y}$ such that the expectations $\mathrm{E}_X\left[k_X(X,X)\right], \mathrm{E}_Y\left[k_Y(Y,Y)\right]$ are finite. As presented by Fukumizu et al. (2004), there exists a unique operator Σ_{YX} (the cross-covariance operator) from $\mathcal{H}_\mathcal{X}$ to $\mathcal{H}_\mathcal{Y}$ such that

$$\begin{aligned} & \langle g, \Sigma_{YX} f \rangle_{\mathcal{H}_\mathcal{Y}} \\ = \; & \mathrm{E}_{XY}\left[f(X)g(Y)\right] - \mathrm{E}_X\left[f(X)\right]\mathrm{E}_Y\left[g(Y)\right] \\ = \; & \mathrm{Cov}\left[f(X), g(Y)\right] \quad \forall f \in \mathcal{H}_\mathcal{X}, g \in \mathcal{H}_\mathcal{Y}\,. \end{aligned}$$

Baker (1973) showed that Σ_{YX} has a representation of the form $\Sigma_{YX} = \Sigma_{YY}^{1/2} V_{YX} \Sigma_{XX}^{1/2}$ with a unique bounded operator $V_{YX}: \mathcal{H}_\mathcal{X} \to \mathcal{H}_\mathcal{Y}$ such that $\|V_{YX}\| \leq 1$. Furthermore, it is known that $\Sigma_{YX} = 0 \Leftrightarrow X \perp\!\!\!\perp Y$ for universal kernels, in the sense of (Steinwart, 2001), or for Gaussian kernels on the entire $\mathbb{R}^m$, proved by Bach and Jordan (2002).

Now, we define the conditional cross-covariance operator. Let $(\mathcal{H}_\mathcal{X}, k_X)$, $(\mathcal{H}_\mathcal{Y}, k_Y)$, $(\mathcal{H}_\mathcal{Z}, k_Z)$ be RKHSs on measurable spaces $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$, respectively, and (X, Y, Z) be a random vector on $\mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$.

$$\Sigma_{YX|Z} := \Sigma_{YX} - \Sigma_{YY}^{1/2} V_{YZ} V_{ZX} \Sigma_{XX}^{1/2}$$

is called the conditional cross-covariance operator, where V_{YZ} and V_{ZX} are the bounded operators derived from Σ_{YZ} and Σ_{ZX}. It can be shown that $\langle g, \Sigma_{YX|Z} f \rangle_{\mathcal{H}_\mathcal{Y}} = \mathrm{E}_Z[\mathrm{Cov}[f(X), g(Y)|Z]]$ for any $f \in \mathcal{H}_\mathcal{X}$ and $g \in \mathcal{H}_\mathcal{Y}$, if k_Z is universal or Gaussian. It should be stressed that we can indeed capture every conditional dependence using the cross-covariance operator if variables X and Y are "blown up", i.e. $\ddot{X} := (X, Z)$ and $\ddot{Y} := (Y, Z)$. One can show that $\Sigma_{\ddot{Y}\ddot{X}|Z} = 0 \Leftrightarrow X \perp\!\!\!\perp Y | Z$.

On the other hand, if $(X, Y) \perp\!\!\!\perp Z$, we have $\Sigma_{\ddot{Y}\ddot{X}|Z} = \Sigma_{YX} \otimes T_Z$, where T_Z is defined by $\langle h_2, T_Z h_1 \rangle := \mathrm{E}[h_1(Z)h_2(Z)]$ for arbitrary $h_1, h_2 \in \mathcal{H}_\mathcal{Z}$. For this reason, we rescale the measure by β_Z in order to obtain comparable conditional and marginal dependence values.

Definition 1 *The strength of the marginal and conditional dependence can be respectively defined by*

$$\begin{aligned} \mathbb{H}_{YX} &:= \|\Sigma_{YX}\|_{\mathrm{HS}}^2 \\ \mathbb{H}_{YX|Z} &:= \beta_Z \left\|\Sigma_{\ddot{Y}\ddot{X}|Z}\right\|_{\mathrm{HS}}^2 \text{ with } \beta_Z := 1/\|T_Z\|_{\mathrm{HS}}^2\,. \end{aligned}$$

By means of rescaling in this way, the measure of conditional dependence equals that of unconditional dependence, if the conditional variable Z is independent of X and Y.

Theorem 1 *We have*

$$(X, Y) \perp\!\!\!\perp Z \quad \Longrightarrow \quad \mathbb{H}_{YX|Z} = \mathbb{H}_{YX}\,.$$

Moreover, if the kernels are universal (e.g. Gaussian kernels on compact subsets of $\mathbb{R}^m$) or Gaussian kernels on the entire $\mathbb{R}^m$,

$$\begin{aligned} \mathbb{H}_{YX} = 0 \quad &\Longleftrightarrow \quad X \perp\!\!\!\perp Y \\ \mathbb{H}_{YX|Z} = 0 \quad &\Longleftrightarrow \quad X \perp\!\!\!\perp Y \,|\, Z\,. \end{aligned}$$

For notational convenience, we will henceforth drop the double-dots on X and Y for the indices that appear in the context of *conditional* cross-covariance operators.

2.2. Empirical Estimation of Hilbert-Schmidt Dependence Measures

We consider the estimation of $\mathbb{H}_{YX}$ and $\mathbb{H}_{YX|Z}$ after finite sampling. It has been shown by Gretton et al. (2005) that

$$\widehat{\mathbb{H}}_{YX}^{(n)} := \frac{1}{(n-1)^2} \mathrm{Tr}\left(\widehat{K}_Y \widehat{K}_X\right)\,.$$

is a consistent estimator for $\mathbb{H}_{YX}$. Here $\widehat{K}$ is the centralized Gram matrix (Schölkopf et al., 1998). Fukumizu et al. (2007) showed that the estimator of the cross-covariance operator guarantees to converge in HS norm at rate $n^{-1/2}$. In some analogy to the construction of an estimator for $\Sigma_{XX|Z}$ given by Fukumizu et al. (2006) we have constructed a consistent estimator on $\mathbb{H}_{YX|Z}$ by

$$\begin{aligned} \widehat{\mathbb{H}}_{YX|Z}^{(n,\epsilon)} := \frac{\hat{\beta}_Z^{(n)}}{(n-1)^2} \mathrm{Tr}\Big(& \widehat{K}_Y \widehat{K}_X - 2\widehat{K}_Y \widehat{K}_Z (\widehat{K}_Z + \epsilon I)^{-2} \\ & \widehat{K}_Z \widehat{K}_X + \widehat{K}_Y \widehat{K}_Z (\widehat{K}_Z + \epsilon I)^{-2} \widehat{K}_Z \widehat{K}_X \widehat{K}_Z (\widehat{K}_Z + \epsilon I)^{-2} \widehat{K}_Z \Big)\,. \end{aligned}$$

Here the estimator $\hat{\beta}_Z^{(n)}$ is given by $n^2 / \sum_{ij} k_Z(z_i, z_j)^2$ and $\epsilon > 0$ a regularization constant[1] that enables inversion. If ϵ converges to zero more slowly than $n^{-1/2}$ one

[1] Our experiments showed that the empirical measures are insensitive to ϵ, if it is chosen sufficiently small, e.g. 10^{-5}.

can show that this estimator converges to $\mathbb{H}_{YX|Z}$. Note that, although it is true that kernel methods are in general inefficient for a large number of data, our kernel-based dependence measures can be computed efficiently for $1000 - 2000$ data points by employing incomplete Cholesky decomposition, as in Fine and Scheinberg (2001).

3. Causal Learning Algorithm Using Dependence Measures

Empirical studies showed that the kernel-based independence measures benefit from the power of detecting nonlinear dependence and can keep type II errors (deciding independence when there is dependence) to a very low level. In the meantime, we expect a potential increase of type I errors (deciding dependence when there is independence). For this reason, it is not very surprising that sometimes so few independence relations are verified that an orientation of edges thereafter is impossible. We propose therefore the following heuristics: conditioning on a common effect has the tendency to generate dependence between the causes. This is at least true if the unconditional dependence between the causes is small. If causes X, Y are already strongly dependent, conditioning on Z can, of course, decrease the dependence. Nevertheless we assume that it will typically decrease the dependence less than conditioning on a common *cause* would do.

Based on this intuition, we introduce a voting-like procedure for orientation of edges: for any triple (X, Y, Z), one gets a vote for Z being a common effect of X and Y, if and only if $\mathbb{H}_{YX|Z} > \lambda\, \mathbb{H}_{YX}$, with an appropriate $\lambda > 0$. Counting these votes we may direct most (not always all) edges in favor of the majority. We choose λ_1 very large in the 1st run and set $\lambda_2 := \max\{\frac{\mathbb{H}_{ZX|Y}}{\mathbb{H}_{ZX}}, \frac{\mathbb{H}_{ZY|X}}{\mathbb{H}_{ZY}}\}$ in the 2nd run. The intuition behind the choice of λ_2 is to consider the one with the weakest decrease of dependence as an evidence of being a common effect. In summary, we sketch our kernel-based causal learning (KCL) algorithm as follows:

Step 1 Connect vertices $X - Y$ if and only if no set S_{xy} (excluding X, Y) can be found with $\mathbb{H}_{YX|S_{xy}} < \epsilon_0$ (ϵ_0 very small).

Step 2 Direct edges as follows: (2.1) Check for all substructures $X - Z - Y$ (X and Y not necessarily nonadjacent) whether Z is a candidate for being a common effect of X and Y with respect to λ_1. If this is the case, $X \to Z$ and $Z \leftarrow Y$ both obtain a vote. Direct all edges which obtained at least one vote (for either of both directions) according to the majority principle. If the result is balanced, leave the edge undirected. (2.2) The same procedure with λ_2.

Step 3 As IC in Section 1.

We would like to emphasize that our assumption cannot be applied to orienting the edges directly[2], but merely to collecting evidences of orientation.

4. Experiments with Toy and Real-World Data

Boolean functions, like OR/AND, are simplified models for many intuitive causal relations in real life. Our first experiment is based on models with 3 or 4 variables logically linked by noisy OR gates. An n-bit ($X_1, \ldots, X_n \in \{0, 1\}$ as inputs) noisy OR can be characterized by conditional probabilities

$$P\left(X_{n+1} = 1 \,|\, x_1, \ldots, x_n\right) = (1 - r)\left(1 - q^{x_1 + \cdots + x_n}\right) + r$$

with $q \in [0, 1]$ and $r \in [0, 1]$. If q and r vanish, the OR gate is deterministic. Here, we present 6 different OR gates. 2IndDet and 3IndDet are deterministic OR with respective 2 and 3 independent inputs; 2IndPro and 3IndPro are probabilistic OR gates with 2 and 3 independent inputs; whereas the probabilistic OR gates 2DepPro and 3DepPro were fed with 2 and 3 dependent inputs (see Table 1 for parameters). We run the experiments 1000 times for respective 200 data points sampled by each of the 6 models.

Table 1. Parameters of 6 OR gates. $P(X_i)$ is shorthand for $P(X_i = 1)$. $\mathrm{OR}_0\{X_1, \ldots, X_i\}$ denotes a deterministic OR gate with $X_1, \cdots, X_i$ as inputs; $\mathrm{OR}_{0.2}\{X_1, \ldots, X_i\}$ denotes a noisy OR gate with $q = r = 0.2$. $(1 - X_1)_{0.1}$ depicts a variable, whose value is with probability 0.1 given by an inverse of X_1 and with probability 0.9 given by the uniform noise.

	2IndDet 	2IndPro 	2DepPro 
X_1	$P(X_1) = 0.6$	$P(X_1) = 0.6$	$P(X_1) = 0.6$
X_2	$P(X_2) = 0.5$	$P(X_2) = 0.5$	$(1 - X_1)_{0.1}$
X_3	$\mathrm{OR}_0\{X_{1,2}\}$	$\mathrm{OR}_{0.2}\{X_{1,2}\}$	$\mathrm{OR}_{0.2}\{X_{1,2}\}$
	3IndDet 	3IndPro 	3DepPro 
X_1	$P(X_1) = 0.6$	$P(X_1) = 0.6$	$P(X_1) = 0.6$
X_2	$P(X_2) = 0.5$	$P(X_2) = 0.5$	$(1 - X_1)_{0.1}$
X_3	$P(X_3) = 0.4$	$P(X_3) = 0.4$	$\mathrm{OR}_{0.2}\{X_{1,2}\}$
X_4	$\mathrm{OR}_0\{X_{1,2,3}\}$	$\mathrm{OR}_{0.2}\{X_{1,2,3}\}$	$\mathrm{OR}_{0.2}\{X_{1,2,3}\}$

In addition, we compare the results of KCL to the

[2]The way of making use of the quantitative information about the strength of dependences has some analogies to the monotone faithfulness principle proposed by Cheng et al. (2002). It states that blocking a previously active path that connects two nodes decreases the mutual information. However, Chickering and Meek (2006) show that this principle cannot generally be valid. For causal networks with many nodes one will usually find several nodes that violate it.

Table 2. The underlying true model: 2-bit OR gates (see Table 1) and the structures generated by different algorithms (see text).

	2IndDet				2IndPro				2DepPro			
True Direction	○ ○	○→○	○←○	○ − ○	○ ○	○→○	○←○	○ − ○	○ ○	○→○	○←○	○ − ○
(X_1, X_2)	**100**	0	0	0	**100**	0	0	0	0	**100**	0	0
(X_1, X_3)	0	**100**	0	0	0	**100**	0	0	0	**100**	0	0
(X_2, X_3)	0	**100**	0	0	0	**100**	0	0	0	**100**	0	0
KCL	○ ○	○→○	○←○	○ − ○	○ ○	○→○	○←○	○ − ○	○ ○	○→○	○←○	○ − ○
	96.7	0.2	0	3.1	**96.7**	0	0.4	2.9	12.6	0.1	0	**87.3**
	0	**99.7**	0	0.3	0	**95.2**	0.4	4.4	0	**98.1**	0	1.9
	0	**99.7**	0.2	0.1	0	**95.2**	0.2	4.8	0	**98.1**	0.1	1.8
PC	○ ○	○→○	○←○	○ − ○	○ ○	○→○	○←○	○ − ○	○ ○	○→○	○←○	○ − ○
	93.9	0	0	6.1	**96.5**	0	0	3.5	**96.8**	0	0	3.2
	0	**93.9**	0	6.1	0	**94.1**	0	5.9	0	**94.2**	0	5.8
	0	**93.9**	0	6.1	0	**94.1**	0	5.9	0	**94.2**	0	5.8
BN-PC	○ ○	○→○	○←○	○ − ○	○ ○	○→○	○←○	○ − ○	○ ○	○→○	○←○	○ − ○
	93.7	0	6.3	0	**96.3**	0	3.7	0	**72.0**	0.1	27.9	0
	0	**93.7**	6.3	0	0	**82.2**	17.8	0	0.1	**71.4**	28.5	0
	0	**93.7**	6.3	0	0	**82.2**	17.8	0	0	**71.4**	28.6	0
ES	○ ○	○→○	○←○	○ − ○	○ ○	○→○	○←○	○ − ○	○ ○	○→○	○←○	○ − ○
	98.5	0.4	0.5	0.6	**99.3**	0.1	0.2	0.4	**99.4**	0.1	0.2	0.3
	0	**99.2**	0.2	0.6	0	**89.0**	6.8	4.2	0	**88.9**	6.5	4.6
	0	**99.1**	0.2	0.7	0	**91.7**	3.5	4.8	0	**91.0**	4.3	4.7
GS	○ ○	○→○	○←○	○ − ○	○ ○	○→○	○←○	○ − ○	○ ○	○→○	○←○	○ − ○
	69.1	16.2	14.7	0	**81.5**	10.4	8.1	0	**80.6**	10.6	8.8	0
	0	**83.1**	16.9	0	0	**68.4**	31.6	0	0	**66.5**	31.5	0
	0	**82.7**	17.3	0	0	**70.0**	30.0	0	0	**68.1**	31.9	0
MWST+GS	○ ○	○→○	○←○	○ − ○	○ ○	○→○	○←○	○ − ○	○ ○	○→○	○←○	○ − ○
	97.2	2.5	0.3	0	**98.7**	1.3	0	0	**98.9**	1.1	0	0
	0	**99.0**	1.0	0	0	**94.6**	5.4	0	0	**94.1**	5.9	0
	0	**97.9**	2.1	0	0	**89.4**	10.6	0	0	**88.2**	11.8	0
MWST+K2	○ ○	○→○	○←○	○ − ○	○ ○	○→○	○←○	○ − ○	○ ○	○→○	○←○	○ − ○
	0	**100**	0	0	39.7	**60.3**	0	0	40.6	**59.4**	0	0
	0	**100**	0	0	0	**100**	0	0	0	**100**	0	0
	0	0	**100**	0	0	0	**100**	0	0	0	**100**	0
MCMC	○ ○	○→○	○←○	○ − ○	○ ○	○→○	○←○	○ − ○	○ ○	○→○	○←○	○ − ○
	69.1	16.2	14.7	0	**77.9**	11.3	10.8	0	**77.4**	11.1	10.5	0
	0	**83.1**	16.9	0	0	**75.9**	24.1	0	0	**75.0**	25.0	0
	0	**82.7**	17.3	0	0	**74.1**	25.9	0	0	**74.4**	25.6	0

well-known constraint-based PC algorithm, BN-PC (an information-theory-based refinement of IC) by Cheng et al. (2002). Apart from constraint-based algorithms, there exists a great variety of Bayesian methods for structure learning, particularly in case of discrete domains. Bayesian approaches with BDe priors using exhaustive search (ES), Greedy Search/Hill-climbing (GS) by Chickering (2003), MCMC (Markov Chain Monte Carlo) by Herskovits (1991) are considered here. The well-known K2 algorithm by Cooper and Herskovits (1992) can actually not be used to find the causal structure, since an initial causal ordering of variables must already be given. K2 is then only able to decide which arrows can be dropped without violating the Markov condition. Heckerman et al. (1994) proposed to apply the maximum weight spanning tree algorithm (MWST) by Chow and Liu (1968) to initialize K2. We call it "MWST+K2". Note that an initial order can also optionally be specified for greedy search (GS). We call this combination "MWST+GS". All these methods are implemented and described in detail by Murphy[3], Leray and Francois[4]. Table 3 summarizes the resulting graph of each algorithm in the majority of cases. The detailed statistics of the 1000 runs can be found in Table 2 and 6. The entries are percentages of detected arcs between two variables (X_i, X_j). For (X_i, X_j), "○ ○" depicts the absence of an edge between X_i and X_j; "○—○" depicts a present but undirected edge between them; "○→○" and "○←○" denote "$X_i \to X_j$" and "$X_i \leftarrow X_j$", respectively. As seen from Table 3, both constraint-based and Bayesian methods have

[3]BayesNet Toolbox, http://bnt.sourceforge.net/.

[4]BNT Structure Learning Package, http://banquiseasi.insa-rouen.fr/projects/bnt-slp/.

Table 3. The first row illustrates the underlying true models (see Table 1 for parameters). Rows 2 to 9 visualize graphical results of different algorithms (see text). Each graph consists of at most 4 nodes, which are represented by circles: X_1: top left, X_2: top right, X_3: bottom left, X_4: bottom right.

	2IndDet	2IndPro	2DepPro	3IndDet	3IndPro	3DepPro
True Model						
KCL						
PC						
BN-PC						
ES						
GS						
MWST+GS						
MWST+K2						
MCMC						

Table 4. Arcs detected by KCL+K2. 400 data points are sampled from Asia network 1000 times. The entries are percentages within columns. No undirected edges are detected.

	(X_1, X_2)	(X_1, X_3)	(X_1, X_4)	(X_1, X_5)	(X_1, X_6)	(X_1, X_7)	(X_1, X_8)	(X_2, X_3)	(X_2, X_4)	(X_2, X_5)	(X_2, X_6)
$\circ\ \circ$	**81.0**	**94.9**	**97.1**	**89.2**	**96.9**	**95.4**	**97.6**	**96.7**	**92.8**	**90.1**	0
$\circ\!\rightarrow\!\circ$	12.4	3.8	1.9	6.9	2.8	4.3	2.3	2.7	4.8	9.1	**99.1**
$\circ\!\leftarrow\!\circ$	6.6	1.3	1.0	3.9	0.3	0.3	0.1	0.6	2.4	0.8	0.9
	(X_2, X_7)	(X_2, X_8)	(X_3, X_4)	(X_3, X_5)	(X_3, X_6)	(X_3, X_7)	(X_3, X_8)	(X_4, X_5)	(X_4, X_6)	(X_4, X_7)	(X_4, X_8)
$\circ\ \circ$	**78.4**	**94.1**	23.3	3.5	**85.4**	**99.1**	**94.4**	**93.2**	0	**76.1**	**80.6**
$\circ\!\rightarrow\!\circ$	19.2	5.7	4.7	26.2	0.1	0	2.5	6.0	**98.7**	19.5	18.5
$\circ\!\leftarrow\!\circ$	2.4	0.2	**72.0**	**70.3**	14.5	0.9	3.1	0.8	1.3	4.4	0.9
	(X_5, X_6)	(X_5, X_7)	(X_5, X_8)	(X_6, X_7)	(X_6, X_8)	(X_7, X_8)	For convenience, we denote here				
$\circ\ \circ$	**98.0**	**99.0**	0	20.3	30.2	**98.0**	X_1: ASIA, X_2: TUB., X_3: SMOKING,				
$\circ\!\rightarrow\!\circ$	0	0.4	**92.0**	**77.5**	**69.8**	2.0	X_4: LUNG, X_5: BRONCHITIS, X_6: TUB./LUNG,				
$\circ\!\leftarrow\!\circ$	2.0	0.6	8.0	2.2	0	0	X_7: X-RAY, X_8: DYSPNOEA.				

lead to quite good results in case of 2-bit models. In case of 3-bit models, the constraint-based algorithms seem to often perform better than most of the Bayesian algorithms considered. In case of 2DepPro and 3DepPro, KCL detected the connection between X_1 and X_2 (Table 3, row KCL), whereas PC erroneously deleted the edge in both cases (Table 3, row PC). Had PC detected the dependence between X_1 and X_2 correctly, it would not have been able to orient any edge, because no independence constraints are available. The result would be a fully connected skeleton. In contrast, although all dependences are correctly verified through kernel-based independence tests, the orientation phase of KCL provides useful hints about causal relations. Note that Both PC and KCL have left edges undirected in network 3DepPro. KCL performs better than PC in the sense that the former oriented as many edges as PC, but no edges are erroneously deleted.

In the second experiment, we focus our attention especially on our orientation procedure by means of "voting triples". The example is an expert-designed causal network with logical links, namely the Asia network. This network was first introduced by Lauritzen and Spiegelhalter (1988) who have specified reasonable transition properties for each variable given its parents. The underlying structure (Figure 1, left) expresses the following known qualitative medical knowledge. DYSPNOEA may be due to tuberculosis (TUB.), LUNG cancer (together TUB./LUNG) or BRONCHITIS, or none of them, or more than one of them. A recent visit to ASIA increases the chances of tuberculosis, while SMOKING is known to be a risk factor for both lung cancer and bronchitis. The results of a single chest X-RAY do not discriminate between lung cancer and tuberculosis, and neither does the presence or absence of DYSPNOEA.

Given the true corresponding skeleton (Figure 2, leftmost), we computed the dependence measure for the "voting triples" corresponding the 8 edges in skeleton. Extensive statistics of the orientation for the 8 edges by KCL can be found in Table 5. Taking the quotient values and λ of different level into account, Step 2.1 of KCL detected a v-structure TUB. $\rightarrow$ TUB./LUNG $\leftarrow$ LUNG; Step 2.2 detected the other v-structure TUB./LUNG $\rightarrow$ DYSPNOEA $\leftarrow$ BRONCHITIS. The unoriented edge TUB./LUNG $-$ X-RAY can be directed in Step 3. The three remaining unoriented edges (Figure 2, rightmost) is due to the limitation of methods which are based on v-structure identification. That is what such method can maximally achieve.

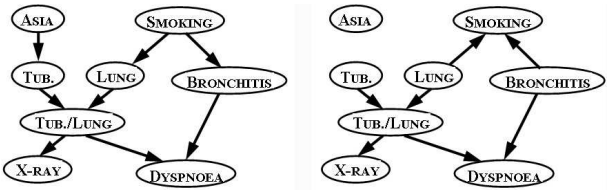

Figure 1. Graphical representation of medical knowledge by Asia network (left). Each node has two possible states representing responses "yes" and "no". The graph on the right side is the structure discoved by KCL+K2.

The performance of PC, see e.g. Fig. 2 of (Leray & Francois, 2004), is unsatisfactory in the sense that several edges are completely missing. Repeated experiments with a sample size from 500 to 5000 show that 3-5 from the total 8 edges are always missing. This result is traced back actually to the independence test of PC. In contrast, the kernel-based independence tests tend to induce redundant edges, particularly when we choose a very small ϵ_0 in Step 1. Therefore, it is important to check how robust KCL is to some addition of unnecessary edges. As an extreme case of $\epsilon_0 = 0$, we applied our orientation procedure to the fully connected skeleton, i.e. no conditional independence would have been verified[5]. As seen

[5]For Asia network, which partly includes very weak correla-

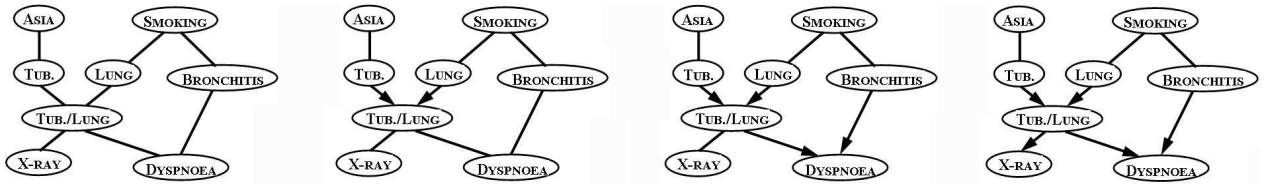

Figure 2. Stepwise results of KCL. The leftmost graph shows the true corresponding skeleton. The second and third graph from left illustrates the result after Step 2.1 and Step 2.2 of KCL, respectively. The rightmost graph is the final output of KCL after Step 3.

from Table 5. The resulting causal order makes often mistakes in orientation of ASIA → TUB., SMOKING → LUNG and SMOKING → BRONCHITIS. Based on the resulting causal order by KCL, we propose to make use of the well-known K2 algorithm by Cooper and Herskovits (1992) to delete edges. Regardless of redundant edges, the other 5 arrows can be discovered correctly. The result of the so-called "KCL+K2" (K2 with a initial causal order detected by KCL) contains no unoriented edges (see Figure 1, right). The missing arc from ASIA to TUB. is probably due to the too weak dependency between the two nodes in datasets of such small sample size. Although the edges from SMOKING to LUNG and BRONCHITIS are erroneously oriented[6], the result contains no unnecessary edges. Table 4 summarizes how often an arrow is detected by KCL+K2 after 1000 runs. An extensive comparison of well-known constraint-based and Bayesian algorithms with respect to Asia network is provided by Leray and Francois (2004). We can see that the KCL+K2 performs better than K2 with other initialization of causal orders, which indicates that our orientation procedure provides quite reliable causal ordering. Furthermore, KCL+K2 is also quite competitive with PC and other Bayesian methods[7]. Most notably, our result can be reliably achieved with datasets of moderate sample size.

The next experiment is a real-world dataset[8] (Fraumeni, 1970) containing the numbers of CIGARETTES (hundreds per capita) smoked (sold) in 43 states in the US and the District of Columbia in 1960 together with death rates per 100 thousand population from various forms of cancer, i.e., BLADDER cancer, LUNG cancer, KIDNEY cancer and LEUKEMIA. The fact that Nevada and the District of Columbia are outliers in the distribution of cigarette consumption contributes to the difficulty of the analysis. The ready explanation for the outliers is that cigarette sales are increased by tourism (Nevada) and commuting workers (District of Columbia). It is known that the consumption

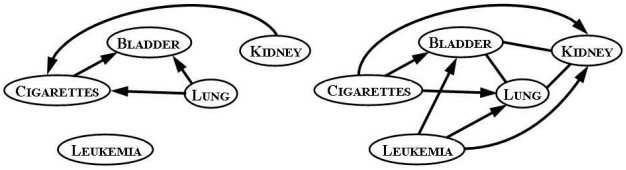

Figure 3. Graphical result of the PC (left) and KCL algorithm (right) for smoking and cancer data.

of cigarettes is a cause of various forms of cancer. As seen from Figure 3 (right), KCL discovers CIGARETTES as the common cause of BLADDER, LUNG and KIDNEY, which confirms the common-sense understanding of the causal influences. Due to our lack of medical understanding, we do not speculate on the plausibility of the absence of the influence from CIGARETTES to LEUKEMIA as well as the orientation from LEUKEMIA to other forms of cancer. Remark that the result of PC (Figure 3, left) contains significantly fewer edges and is less specific. In particular, the orientations from LUNG and KIDNEY to CIGARETTES are obviously erroneous.

5. Conclusion

We have proposed a kernel-based approach for automatically generating causal structures. The idea is to define unconditional and conditional cross-covariance operators in RKHSs and consider the Hilbert Schmidt norm of these operators as a measure for the unconditional and conditional dependence. We specify the independence test of IC with the kernel-based independence test and apply our dependence measure not only for the decision of independence but also for getting additional hints on the causal directions by additionally taking the strength of dependences into account. Our method is even applicable to data without any

tion, it is hard to find an appropriate ϵ_0 for 400 data points.

[6] "KCL+K2+KCL" (using KCL to orient the adjacency structure of the result of KCL+K2) would revise both erroneous orientations into unoriented.

[7] Actually, with regard to the sample size, the result by KCL+K2 is better than all 12 algorithms listed in Fig. 2 of (Leray & Francois, 2004) concerning the so-called "editing measures". Editing measure (Leray & Francois, 2004) is defined by the length of the minimal sequence of operators needed to transform the original graph into the resulting one. Operators are edge-insertion, edge-deletion and edge reversal. Our result has an editing measure of merely 3

[8] The data are collected in the Data and Story Library (DASL), available at http://lib.stat.cmu.edu/DASL/DataArchive.html, and listed also as an example for the causal learning software TETRAD http://www.phil.cmu.edu/projects/tetrad_examples/.

verified statistical independence, e.g. exploring the causal order for the K2 algorithm.

The kernel-based approach provides a unifying method which can treat continuous, discrete and even hybrid models. Because of the explosion of parameters and non-trivial specification of good priors, continuous variables, which have large value sets after discretization, are problematic for Bayesian approaches. Constraint-based PC requires Gaussian assumption for continuous domains and cannot deal with hybrid models at all. Although dependences can be captured by mutual information in theory, the estimation of conditional mutual information, to the best of our knowledge, is a non-trivial problem currently unsolved in its generality and involves the explicit estimation of the densities, which is hard for high dimensional data, unless suitable smoothness assumptions are made. We are of the opinion that the kernel method provides a convenient tool to use smoothness assumptions in an implicit way. The extension of KCL to vectorial variables is straightforward.

References

Aronszajn, N. (1950). Theory of reproducing kernels. *Trans. of the Am. Math. Soc., 68*, 337–404.

Bach, F., & Jordan, M. (2002). Kernel independent component analysis. *J. of Mach. Learning Res., 3*, 1–48.

Baker, C. (1973). Joint measures and cross-covariance operators. *Trans. of the Am. Math. Soc., 186*, 273–289.

Cheng, J., Greiner, R., Kelly, J., Bell, D., & Liu, W. (2002). Learning Bayesian networks from data: An information-theory based approach. *Art. Intl. J., 137*, 43–90.

Chickering, D. (2003). Optimal structure identification with greedy search. *J. of Mach. Learning Res., 3*, 507–554.

Chickering, D., & Meek, C. (2006). On the incompatibility of faithfulness and monotone DAG faithfulness. *J. of Art. Intl., 170*, 653–666.

Chow, C., & Liu, C. (1968). Approximating discrete probability distributions with dependence trees. *IEEE Trans. on the Information Theory, 14*, 462–467.

Cooper, G., & Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning, 9*, 309–347.

Fine, S., & Scheinberg, K. (2001). Efficient SVM training using low-Rank kernel representations. *J. of Machine Learning Res., 2*, 243–264.

Fraumeni, J. (1970). Cigarette smoking and cancers of the urinary tract: Geographic variations in the United States. *J. of the National Cancer Institute, 41*, 1205–1211.

Fukumizu, K., Bach, F., & Gretton, A. (2007). Statistical consistency of kernel canonical correlation analysis. *J. of Machine Learning Res., 8*, 361–383.

Fukumizu, K., Bach, F., & Jordan, M. (2004). Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces. *J. of Machine Learning Res., 5*, 73–99.

Fukumizu, K., Bach, F., & Jordan, M. (2006). *Kernel dimension reduction in regression* (Technical Report 715). Dept. of Statistics, University of California, Berkeley.

Gretton, A., Bousquet, O., Smola, A., & Schölkopf, B. (2005). Measuring statistical dependence with Hilbert-Schmidt norms. *Proc. Algorithmic Learning Theory* (pp. 63–77). Berlin: Springer-Verlag.

Heckerman, D., Geiger, D., & Chickering, D. (1994). Learning Bayesian networks: The combination of knowledge and statistical Data. *Proc. 10th Conf. Uncertainty in Art. Intl.* (pp. 293–301). San Francisco, CA: Morgan Kaufmann Publishers.

Herskovits, E. (1991). *Computer-based probabilistic network construction.* Doctoral dissertation, Medical Information Sciences, Stanford University, Stanford, CA.

Lauritzen, S., & Spiegelhalter, D. (1988). Local computations with probabilities on graphical structures and their application to expert systems. *J. of the Royal Statist. Soc. Series B (Methodological), 50*, 157–224.

Leray, P., & Francois, O. (2004). *BNT structure learning package: Documentation and experiments* (Technical Report FRE CNRS 2645). Laboratoire PSI, Universitè et INSA de Rouen.

Pearl, J. (2000). *Causality: Models, reasoning, and inference*. Cambridge University Press.

Schölkopf, B., & Smola, A. (2002). *Learning with kernels*. Cambridge, MA: MIT Press.

Schölkopf, B., Smola, A., & Müller, K. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation, 10*, 1299–1319.

Spirtes, P., & Glymour, C. (1991). An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review, 9*, 67–72.

Spirtes, P., Glymour, C., & Scheines, R. (1993). *Causation, prediction, and search (lecture notes in statistics)*. New York, NY: Springer-Verlag.

Steinwart, I. (2001). On the influence of the kernel on the consistency of support vector machines. *J. of Machine Learning Res., 2*, 67–93.

Table 5. Statistics of arcs detected by KCL. 400 data points are sampled from Asia network 1000 times. The entries are percentages within columns. "TS" stands for the percentages achieved by given true corresponding skeleton; "FS" stands for the percentages achieved by given fully connected skeleton. For convenience, we denote here X_1: ASIA, X_2: TUB.,X_3: SMOKING, X_4: LUNG, X_5: BRONCHITIS,X_6: TUB./LUNG, X_7: X-RAY, X_8: DYSPNOEA.

	(X_1, X_2)		(X_2, X_6)		(X_3, X_4)		(X_3, X_5)		(X_4, X_6)		(X_5, X_8)		(X_6, X_7)		(X_6, X_8)	
	TS	FS	TS	FS	TS	FS	TS	FS	TS	FS	TS	FS	TS	FS	TS	FS
$\circ\!\!\rightarrow\!\!\circ$	0.2	16.4	**97.7**	**95.8**	15.7	6.6	0.4	20.2	**97.2**	**94.3**	**77.7**	**88.7**	**93.6**	**75.5**	**92.6**	**95.5**
$\circ\!\!\leftarrow\!\!\circ$	0.1	**65.0**	0.1	2.0	29.3	**72.2**	44.6	**56.4**	0.6	0.7	16.6	5.7	4.2	13.9	6.5	3.7
$\circ - \circ$	**99.7**	18.6	2.2	2.2	**55.0**	21.2	**55.0**	23.4	2.2	5.0	5.7	5.6	2.2	10.6	0.9	0.8

Table 6. The underlying true model: 3-bit OR gates (see Table 1) and the structures generated by different algorithms (see text).

	3IndDet				3IndPro				3DepPro			
True Direction	$\circ\ \circ$	$\circ\!\!\rightarrow\!\!\circ$	$\circ\!\!\leftarrow\!\!\circ$	$\circ - \circ$	$\circ\ \circ$	$\circ\!\!\rightarrow\!\!\circ$	$\circ\!\!\leftarrow\!\!\circ$	$\circ - \circ$	$\circ\ \circ$	$\circ\!\!\rightarrow\!\!\circ$	$\circ\!\!\leftarrow\!\!\circ$	$\circ - \circ$
(X_1, X_2)	**100**	0	0	0	**100**	0	0	0	0	**100**	0	0
(X_1, X_3)	**100**	0	0	0	**100**	0	0	0	0	**100**	0	0
(X_1, X_4)	0	**100**	0	0	0	**100**	0	0	0	**100**	0	0
(X_2, X_3)	**100**	0	0	0	**100**	0	0	0	0	**100**	0	0
(X_2, X_4)	0	**100**	0	0	0	**100**	0	0	0	**100**	0	0
(X_3, X_4)	0	**100**	0	0	0	**100**	0	0	0	**100**	0	0
KCL	$\circ\ \circ$	$\circ\!\!\rightarrow\!\!\circ$	$\circ\!\!\leftarrow\!\!\circ$	$\circ - \circ$	$\circ\ \circ$	$\circ\!\!\rightarrow\!\!\circ$	$\circ\!\!\leftarrow\!\!\circ$	$\circ - \circ$	$\circ\ \circ$	$\circ\!\!\rightarrow\!\!\circ$	$\circ\!\!\leftarrow\!\!\circ$	$\circ - \circ$
	74.0	8.0	2.8	15.2	**71.8**	5.0	9.7	13.5	11.1	2.2	2.7	**84.0**
	76.8	8.2	2.7	12.3	**74.3**	4.9	12.4	8.4	1.5	**51.3**	19.4	27.8
	0	**98.6**	0.5	0.9	0.1	**91.0**	7.1	1.8	1.1	**47.2**	25.4	26.3
	76.8	4.7	3.9	14.6	**71.9**	4.8	11.5	11.8	1.5	**67.0**	10.1	21.4
	0	**96.0**	3.0	1.0	0	**93.8**	4.4	1.8	0.2	**62.5**	12.6	24.7
	0	**94.8**	3.9	1.3	0	**96.4**	1.2	2.4	0.2	1.8	2.4	**95.6**
PC	$\circ\ \circ$	$\circ\!\!\rightarrow\!\!\circ$	$\circ\!\!\leftarrow\!\!\circ$	$\circ - \circ$	$\circ\ \circ$	$\circ\!\!\rightarrow\!\!\circ$	$\circ\!\!\leftarrow\!\!\circ$	$\circ - \circ$	$\circ\ \circ$	$\circ\!\!\rightarrow\!\!\circ$	$\circ\!\!\leftarrow\!\!\circ$	$\circ - \circ$
	98.5	0	0	1.5	**96.8**	0.5	0	2.7	**69.8**	12.2	5.9	12.1
	98.1	0	0	1.9	**98.5**	0.2	0	1.3	29.3	**47.2**	6.9	16.6
	0	**100**	0	0	3.7	**96.2**	0	0.1	18.7	**55.8**	6.0	19.5
	97.4	0	0	2.6	**97.1**	0.1	0	2.8	20.4	**54.9**	10.5	14.2
	0	**100**	0	0	0.9	**99.0**	0.1	0	7.9	**61.7**	10.5	19.9
	0	**99.8**	0.2	0	0.3	**99.5**	0.2	0	5.4	11.5	23.3	**59.8**
BN-PC	$\circ\ \circ$	$\circ\!\!\rightarrow\!\!\circ$	$\circ\!\!\leftarrow\!\!\circ$	$\circ - \circ$	$\circ\ \circ$	$\circ\!\!\rightarrow\!\!\circ$	$\circ\!\!\leftarrow\!\!\circ$	$\circ - \circ$	$\circ\ \circ$	$\circ\!\!\rightarrow\!\!\circ$	$\circ\!\!\leftarrow\!\!\circ$	$\circ - \circ$
	97.5	0.4	2.1	0	**96.9**	0.7	2.4	0	**71.6**	9.9	18.5	0
	97.8	0.6	1.6	0	**97.0**	0.3	2.7	0	28.5	23.8	**47.7**	0
	0	**74.2**	25.8	0	0.6	31.3	**68.1**	0	20.2	27.0	**52.8**	0
	96.8	0.7	2.5	0	**97.6**	0.2	2.2	0	19.3	29.4	**51.3**	0
	0	**65.9**	34.1	0	0.8	39.8	**59.4**	0	8.1	32.4	**59.5**	0
	0	48.7	**51.3**	0	0	47.2	**52.8**	0	4.9	12.6	**82.5**	0
ES	$\circ\ \circ$	$\circ\!\!\rightarrow\!\!\circ$	$\circ\!\!\leftarrow\!\!\circ$	$\circ - \circ$	$\circ\ \circ$	$\circ\!\!\rightarrow\!\!\circ$	$\circ\!\!\leftarrow\!\!\circ$	$\circ - \circ$	$\circ\ \circ$	$\circ\!\!\rightarrow\!\!\circ$	$\circ\!\!\leftarrow\!\!\circ$	$\circ - \circ$
	99.3	0.1	0.2	0.4	**98.8**	0.2	0.5	0.5	**74.6**	10.7	11.2	3.5
	99.2	0	0.2	0.6	**99.4**	0.5	0.1	0	35.6	**49.8**	10.3	4.3
	0	**100**	0	0	2.8	**49.5**	40.0	7.7	24.3	**62.2**	8.6	4.9
	98.8	0.5	0.4	0.3	**98.7**	0.5	0.7	0.1	34.3	**54.0**	8.5	3.2
	0	**100**	0	0	0.9	**60.5**	30.3	8.3	14.3	**69.3**	10.7	5.7
	0	**100**	0	0	0.4	**61.1**	30.6	7.9	11.8	30.2	**46.5**	11.5
GS	$\circ\ \circ$	$\circ\!\!\rightarrow\!\!\circ$	$\circ\!\!\leftarrow\!\!\circ$	$\circ - \circ$	$\circ\ \circ$	$\circ\!\!\rightarrow\!\!\circ$	$\circ\!\!\leftarrow\!\!\circ$	$\circ - \circ$	$\circ\ \circ$	$\circ\!\!\rightarrow\!\!\circ$	$\circ\!\!\leftarrow\!\!\circ$	$\circ - \circ$
	90.1	4.4	5.5	0	**97.5**	1.3	1.2	0	25.3	36.0	**38.7**	0
	93.1	2.7	4.2	0	**97.4**	1.8	0.8	0	**50.9**	17.2	31.9	0
	0	**75.2**	24.8	0	2.5	32.4	**65.1**	0	33.7	26.2	**40.1**	0
	93.1	3.1	3.8	0	**95.6**	2.0	2.4	0	**44.9**	19.5	35.6	0
	0	**69.0**	31.0	0	1.0	42.4	**56.6**	0	25.8	25.8	**48.4**	0
	0	**64.5**	35.5	0	0.5	47.2	**52.3**	0	1.3	41.2	**57.5**	0
MWST+GS	$\circ\ \circ$	$\circ\!\!\rightarrow\!\!\circ$	$\circ\!\!\leftarrow\!\!\circ$	$\circ - \circ$	$\circ\ \circ$	$\circ\!\!\rightarrow\!\!\circ$	$\circ\!\!\leftarrow\!\!\circ$	$\circ - \circ$	$\circ\ \circ$	$\circ\!\!\rightarrow\!\!\circ$	$\circ\!\!\leftarrow\!\!\circ$	$\circ - \circ$
	97.4	2.5	0.1	0	**98.7**	0.9	0.4	0	43.8	**46.4**	9.8	0
	97.7	2.0	0.3	0	**99.3**	0.5	0.2	0	**50.5**	37.5	12.0	0
	0	**94.4**	5.6	0	2.5	**68.5**	29.0	0	33.9	**50.1**	16.0	0
	94.8	2.5	2.7	0	**91.7**	4.0	4.3	0	**44.8**	24.2	31.0	0
	0	**82.9**	17.1	0	0.9	24.5	**74.6**	0	25.6	**39.8**	34.6	0
	0	**78.1**	21.9	0	0.5	32.2	**67.3**	0	1.1	41.8	**57.1**	0
MWST+K2	$\circ\ \circ$	$\circ\!\!\rightarrow\!\!\circ$	$\circ\!\!\leftarrow\!\!\circ$	$\circ - \circ$	$\circ\ \circ$	$\circ\!\!\rightarrow\!\!\circ$	$\circ\!\!\leftarrow\!\!\circ$	$\circ - \circ$	$\circ\ \circ$	$\circ\!\!\rightarrow\!\!\circ$	$\circ\!\!\leftarrow\!\!\circ$	$\circ - \circ$
	34.1	**65.9**	0	0	**96.4**	3.6	0	0	9.8	**90.2**	0	0
	66.3	33.7	0	0	**95.9**	4.1	0	0	44.6	**55.4**	0	0
	0	**100**	0	0	3.0	**97.0**	16.0	0	26.1	**73.9**	16.0	0
	34.9	0.1	**65.0**	0	**92.4**	0.1	7.5	0	38.0	0.1	**61.9**	0
	0	0	**100**	0	0.9	0.7	**98.4**	0	17.4	0.2	**82.4**	0
	0	0	**100**	0	0.5	0.2	**99.3**	0	0	41.6	**58.4**	0
MCMC	$\circ\ \circ$	$\circ\!\!\rightarrow\!\!\circ$	$\circ\!\!\leftarrow\!\!\circ$	$\circ - \circ$	$\circ\ \circ$	$\circ\!\!\rightarrow\!\!\circ$	$\circ\!\!\leftarrow\!\!\circ$	$\circ - \circ$	$\circ\ \circ$	$\circ\!\!\rightarrow\!\!\circ$	$\circ\!\!\leftarrow\!\!\circ$	$\circ - \circ$
	86.8	6.2	7.0	0	**91.7**	3.4	4.9	0	**37.4**	29.8	32.8	0
	86.0	6.7	7.3	0	**90.8**	3.5	5.7	0	31.5	**42.7**	25.8	0
	0	**99.6**	0.4	0	5.8	43.8	**50.4**	0	22.9	**46.6**	30.5	0
	86.7	6.0	7.3	0	**89.5**	5.4	5.1	0	31.8	**42.6**	25.6	0
	0	**99.3**	0.7	0	1.9	48.8	**49.3**	0	13.7	**52.1**	34.2	0
	0	**99.4**	0.6	0	0.7	**51.3**	48.0	0	9.8	38.6	**51.6**	0