# Fusion, Propagation, and Structuring in Belief Networks

**Judea Pearl**

Cognitive Systems Laboratory, Computer Science Department,

University of California. Los Angeles, CA 90024, U.S.A

Recommended by Patrick Hayes

### ABSTRACT

Belief networks are directed acyclic graphs in which the nodes represent propositions (or variables), the arcs signify direct dependences between the linked propositions, and the strengths of these dependencies are quantified by conditional probabilities. A network of this sort can be used to represent the generic knowledge of a domain expert, and it turns into a computational architecture if the links are used not merely for storing factual knowledge but also for directing and activating the data flow in the computations which manipulate this knowledge.

The first part of the paper deals with the task of fusing and propagating the impacts of new information through the networks in such a way that, when equilibrium is reached, each proposition will be assigned a measure of belief consistent with the axioms of probability theory. It is shown that if the network is singly connected (e.g. tree-structured), then probabilities can be updated by local propagation in an isomorphic network of parallel and autonomous processors and that the impact of new information can be imparted to all propositions in time proportional to be longest path in the network.

The second part of the paper deals with the problem of finding a tree-structured representation for a collection of probabilistically coupled proposition using auxiliary (dummy) variables, colloquially called “hidden causes.” It is shown that if such a tree-structured representation exists, then it is possible to uniquely uncover the topology of the tree by observing pairwise dependencies among the available propositions (i.e. the leaves of the tree). The entire tree structure, including the strengths of all internal relationships, can be reconstructed in time proportional to n log n, where n is the number of leaves.

### 1.Introduction

This s