the probabilistic network framework

Burak Galip ASLAN, PhD

probabilistic network framework

exact probabilistic inference;

- ***Judea Pearl's*** *(easiest to explain and comprehend)*

more efficient algoritms for exact inference;

- ***jointree propagation*** (Lauritzen et al.)
- ***variable elimination*** (Dechter & Cooper)

research on yet more efficient algorithms both for exact and approximate inference, and also for dealing with networks that include continuous variables;

- ***loop cutset conditioning*** (Becker & Geiger)

the probabilistic network formalism

concise representation of a joint probability distribution on a set of statistical variables;

- *Bayesian belief network*
- *belief network*
- *probabilistic network*

the probabilistic network formalism

concise representation of a joint probability distribution on a set of statistical variables;

- *Bayesian belief network*
- *belief network*
- *probabilistic network*

a probabilistic network comprises two parts:

1. a qualitative part
2. a quantitative part

the probabilistic network formalism

1. qualitative part

   graphical representation of the independence holding among variables in the probability distribution

   a (***minimal***) **directed I-map** for the independence relations of distribution

the probabilistic network formalism

2. quantitative part

   a set of functions representing numerical quantities from the distribution; with each vertex in the digraph being associated with a probability assesment function which is basically a set of (conditional) probabilities describing the values of the vertex' predecessors on the probabilities of this vertex itself

the probabilistic network formalism

a probabilistic network is a tuple $B=(G, \Gamma)$ where
$G=(V(G), A(G))$ is an acyclic digraph with vertices $V(G)=\{V_1,...,V_n\}$, $n\geq 1$
and arcs $A(G)$ as $\Gamma=\{\gamma v_i | V_i \in V(G)\}$ is a set of real-valued non-negative functions where $\gamma vi:\{c_{V_i}\}X\{c_{PG(V_i)}\}\rightarrow[0,1]$ called (conditional) probability assesment functions such that; for each configuration $c_{PG(V_i)}$ of the set $PG(V_i)$ of (immediate) predecessors of vertex V_i in G, we have that $\gamma vi(\neg V_i | c_{PG(V_i)})=1-\gamma v_i(V_i | c_{PG(V_i)})$, $i=1,...,n$

V_i is a vertex from the digraph and it is also a statistical variable at the same time

probabilistic inference

the probabilistic network is generally used for probabilistic inference, that is, for making **probabilistic statements** concerning the variables that are represented in the network

probabilistic inference

the probabilistic network is generally used for probabilistic inference, that is, for making **probabilistic statements** concerning the variables that are represented in the network

the digraph of a probabilistic network can be exploited for any interest on variables of the network using the basic rules of conditioning and marginalisation

probabilistic inference

the probabilistic network is generally used for probabilistic inference, that is, for making **probabilistic statements** concerning the variables that are represented in the network

the digraph of a probabilistic network can be exploited for any interest on variables of the network using the basic rules of conditioning and marginalisation

however it is **computationally infeasible** (not exploiting the independences of the qualitative part)

Pearl's algorithm

best explained by object-centered point of view

digraph -> computational architecture

vertices -> autonomous objects

arcs -> **bidirectional** communication channels

each vertex has a **local processor** capable of performing simple probabilistic calculations and a **local memory** to store its associated probability assessment function

vertices send each other **parameters** providing information about the joint probability distribution

Pearl's algorithm

parameters:

- information determined from the network's probability assessments *(quantitative part)*
- independences portrayed in the digraph *(qualitative part)*
- evidence that has been entered and processed so far *(belief updates w.r.t. observations)*

Pearl's algorithm

parameters:

- information determined from the network's probability assessments *(quantitative part)*
- independences portrayed in the digraph *(qualitative part)*
- evidence that has been entered and processed so far *(belief updates w.r.t. observations)*

so each each vertex is able to compute the probabilities of its values from its own probability assessment function and the information it receives from its neighbors.

Pearl's algorithm

initially a probabilistic network is in an **equilibrium** state i.e. recomputation of various parameters that the vertices send one another does not result in a change in any value

Pearl's algorithm

initially a probabilistic network is in an **equilibrium** state i.e. recomputation of various parameters that the vertices send one another does not result in a change in any value

when a piece of evidence is entered into the network for some vertex, the equilibrium gets **perturbed**, variables are subject to change

Pearl's algorithm

initially a probabilistic network is in an **equilibrium** state i.e. recomputation of various parameters that the vertices send one another does not result in a change in any value

when a piece of evidence is entered into the network for some vertex, the equilibrium gets **perturbed**, variables are subject to change

the distribution should be **conditioned** on the (additional) evidence obtained

Pearl's algorithm

once the value of a variable is known with certainty, the probability distribution of that variable **degenerates and can no longer change**

Pearl's algorithm

once the value of a variable is known with certainty, the probability distribution of that variable **degenerates and can no longer change**

entering evidence for a variable into a probabilistic network thus basically amounts to **adding its corresponding vertex to the (initially empty) blocking-set W** for the digraph, thereby changing the independences that have been taken into account by the inference algorithm

Pearl's algorithm

to process evidence entered for some vertex, the parameters of this vertex sends to its neigbors are updated

Pearl's algorithm

to process evidence entered for some vertex, the parameters of this vertex sends to its neigbors are updated

after receiving updates its neigbors also compute new parameter values to send for their own neighbors and so on...

Pearl's algorithm

to process evidence entered for some vertex, the parameters of this vertex sends to its neigbors are updated

after receiving updates its neigbors also compute new parameter values to send for their own neighbors and so on...

the impact of evidence thus spreads throught the network by **message-passing between neighboring vertices**

directed trees

directed tree for qualitative part i.e. a vertex may have *several successors* but at most *one predecessor*

directed trees

directed tree for qualitative part i.e. a vertex may have *several successors* but at most *one predecessor*

let G be a digraph of a probabilistic network

let V be its set of vertices

a vertex $V_i \in V$ is called «instantiated» if its value is known with certainty

otherwise it is called «uninstantiated»

directed trees

let $X \subseteq V$ be the set of instantiated vertices from V

the configuration C_x of X that is known with certainty is called a «partial configuration» of V and will be denoted as $\tilde{C}_v$

There is no explicit specification of vertices of this subset $\tilde{C}_v$ i.e. it is a dynamic set

directed trees

at any time during probabilistic inference with a probabilistic network, the probabilities of the values of a vertex of interest are **dependent upon all evidence entered** so far into the network

directed trees

let B=(G,Γ) be a probabilistic network where

G=**(**V(G),A(G)**)** is a directed tree and

Let Pr be a joint probability distribution defined by B

Let $V_i \in V(G)$ be a vertex in G and

$V_i^- = \sigma^*(V_i)$ and $V_i^+ = V(G) \setminus V_i^-$ *(like all others)*

directed trees

let $B=(G,\Gamma)$ be a probabilistic network where

$G=(V(G),A(G))$ is a directed tree and

Let Pr be a joint probability distribution defined by B

Let $V_i \in V(G)$ be a vertex in G and

$V_i^- = \sigma^*(V_i)$ and $V_i^+ = V(G) \setminus V_i^-$ *(like all others)*

$$Pr(V_i|C_{V(G)}) = \alpha \cdot Pr(C_{v_i-}|V_i) \cdot Pr(V_i|C_{v_i+})$$

where $C_{V(G)} = C_{v_i-} \wedge C_{v_i+}$ and

α is a normalization constant

directed trees

Using d-separation criterion;

$\langle X,\{V_i\},Y\rangle_{\mathbf{G}}^{\mathbf{d}}$

Since G is a directed I-map for the joint probability distribution Pr, we can conclude that $I_{\mathbf{Pr}}(X,\{V_i\},Y)$ for all sets $X \subseteq V_i^-$ and $Y \subseteq V_i^+$

directed trees

the probabilities of the values of a vertex of interest can be expressed in terms of two factors describing the influence of evidence entered for this vertex' descendants and for all other vertices separately

recall;

let B=(G,Γ) be a probabilistic network where

G=**(**V(G),A(G)**)** is a directed tree and

Let Pr be a joint probability distribution defined by B

Let $V_i \in V(G)$ be a vertex in G and

$V_i^- = \sigma^*(V_i)$ *(successors)* and $V_i^+ = V(G) \setminus V_i^-$ *(and all others)*

directed trees

the **compound causal parameter** for the vertex describes the combined influence of evidence entered for all other vertices in digraph

directed trees

the **compound causal parameter** for the vertex describes the combined influence of evidence entered for all other vertices in digraph

the **compound diagnostic parameter** for a vertex describes the combined influence on this vertex' probabilities of all evidence that has been entered for its descendants

directed trees

the **compound causal parameter** for the vertex describes the combined influence of evidence entered for all other vertices in digraph

the **compound diagnostic parameter** for a vertex describes the combined influence on this vertex' probabilities of all evidence that has been entered for its descendants

i.e. talking about influence of evidences on the vertex in question

directed trees

talking about: **uninstantiated** vertices having either no incoming arcs **or** no outgoing arcs

directed trees

talking about: **uninstantiated** vertices having either no incoming arcs **or** no outgoing arcs

a directed tree has one vertex W without any incoming arcs i.e. it is the **root** of the tree

directed trees

talking about: **uninstantiated** vertices having either no incoming arcs **or** no outgoing arcs

a directed tree has one vertex W without any incoming arcs i.e. it is the **root** of the tree

the directed tree may further include several vertices with no outgoing arcs i.e. they are the **leaves** of the tree

directed trees

talking about: **uninstantiated** vertices having either no incoming arcs **or** no outgoing arcs

a directed tree has one vertex W without any incoming arcs i.e. it is the **root** of the tree

the directed tree may further include several vertices with no outgoing arcs i.e. they are the **leaves** of the tree

and talking about: **instatiated** vertices

directed trees

data fusion lemma implies that the compound causal? and diagnostic parameters for a vertex provide it with enough information for computing the probabilities of its values, that is, no further knowledge of the joint probability distribution is needed

directed trees

data fusion lemma implies that the compound causal? and diagnostic parameters for a vertex provide it with enough information for computing the probabilities of its values, that is, no further knowledge of the joint probability distribution is needed

the compound *diagnostic* parameter for a vertex specifies probabilistic information from all its descendants combined; an analogous observation applies to the compound *causal* parameter for the vertex

directed trees

for exploiting the digraph of a probabilistic network as a computational architecture, the compound parameters for a vertex have to be computed from separate parameters originating from its various neighbors

directed trees

a **causal parameter** is a parameter that a vertex sends to a successor to provide this successor with information concerning its non-descendants

directed trees

a **causal parameter** is a parameter that a vertex sends to a successor to provide this successor with information concerning its non-descendants

a **diagnostic parameter** is a parameter that a vertex sends to its predecessor to provide this predecessor with information concerning the vertices in the subtree rooted at the vertex at hand

directed trees

a **causal parameter** is a parameter that a vertex sends to a successor to provide this successor with information concerning its non-descendants

a **diagnostic parameter** is a parameter that a vertex sends to its predecessor to provide this predecessor with information concerning the vertices in the subtree rooted at the vertex at hand

the separate causal and diagnostic are the messages the vertices send to each other through the communication channels of the computational architecture

directed trees

for the **root** of a directed tree **no diagnostic parameter** is defined because it does not have a predecessor

for the **leaves** of a tree **no causal parameters** are defined as these vertices do not have any successors

directed trees

a vertex can compute its **compound causal parameter** from the causal parameter it receives from its predecessor and its own probability assessment function

directed trees

- a vertex can compute its **compound causal parameter** from the causal parameter it receives from its predecessor and its own probability assessment function
- a vertex can compute its **compound diagnostic parameter** from the separate diagnostic parameters it receives from its successors in the digraph

directed trees

a vertex can compute the probabilities of its values from **its own** probability assessment function and causal and diagnostic parameters it receives from its neighbors

directed trees

- a vertex can compute the probabilities of its values from **its own** probability assessment function and causal and diagnostic parameters it receives from its neighbors
- a vertex can also compute the diagnostic parameter **to send** to its predecessor and causal parameter **to send** to a successor

directed trees

data fusion lemma and 4 computation rules provided by lemmas 4.2.4, 4.2.5, 4.2.6 and 4.2.7 constitute Pearl's algorithm for probabilistic inference with a probabilistic network comprising a directed tree for its qualitative part

singly connected digraphs

removal of any arc splits the graph into two separate components

singly connected digraphs

removal of any arc splits the graph into two separate components

a vertex has m neighbors and correspondingly m subgraphs each containing a neighbor of that vertex where after the removal of that vertex and all its incoming and outgoing arcs, there does not exist a path from one subgraph to another

singly connected digraphs

removal of any arc splits the graph into two separate components

a vertex has m neighbors and correspondingly m subgraphs each containing a neighbor of that vertex where after the removal of that vertex and all its incoming and outgoing arcs, there does not exist a path from one subgraph to another

only redefining compound parameters of directed trees (upper and lower graphs)

multiply connected digraphs

singly connected digraph exploits independences that are read from the network's graph by local inspection of a vertex' incoming and outgoing arcs

multiply connected digraphs

singly connected digraph exploits independences that are read from the network's graph by local inspection of a vertex' incoming and outgoing arcs

P.S. there must be **at most one chain** between between vertices

multiply connected digraphs

singly connected digraph exploits independences that are read from the network's graph by local inspection of a vertex' incoming and outgoing arcs

P.S. there must be **at most one chain** between between vertices

above assumptions does **not** hold in a multiply connected digraph

multiply connected digraphs

vertices may indefinitely send newly updated messages, originating **from the same evidence**, to their neighbors, causing the network:

multiply connected digraphs

vertices may indefinitely send newly updated messages, originating **from the same evidence**, to their neighbors, causing the network:

- to **never** reach an equilibrium or

multiply connected digraphs

vertices may indefinitely send newly updated messages, originating **from the same evidence**, to their neighbors, causing the network:

- to **never** reach an equilibrium or
- even if it reaches an equilibrium, it is **not** guaranteed to reflect the correct updated joint probability distribution

multiply connected digraphs

vertices may indefinitely send newly updated messages, originating **from the same evidence**, to their neighbors, causing the network:

- to **never** reach an equilibrium or
- even if it reaches an equilibrium, it is **not** guaranteed to reflect the correct updated joint probability distribution

Pearl's algorithm has to be supplemented with **an additional method** for coping with the loops in such a digraph

multiply connected digraphs

loop cutset conditioning [Pearl, 1988]

reasoning by assumption

some vertices **cut all loops** in digraph so that it behaves like a singly connected digraph

the selected vertices are said to constitute a **loop cutset** for the network's digraph

multiply connected digraphs

loop cutset conditioning [Pearl, 1988]

reasoning by assumption

some vertices **cut all loops** in digraph so that it behaves like a singly connected digraph

the selected vertices are said to constitute a **loop cutset** for the network's digraph

finding an optimal loop cutset is a NP-hard problem

[Suermondt & Cooper, 1990] heuristics for finding a good loop cutset

references

Probabilistic Reasoning

(Probabilistisch Redeneren)

authors: Linda van der Gaag
Silja Renooij

Fall 2017

Universiteit Utrecht

WIKIPEDIA

Google™