Graphical probabilistic models – broadening (time, continuous variables, undirected graphs)

Jiří Kléma

Department of Cybernetics,
FEE, CTU at Prague

http://ida.felk.cvut.cz

Outline

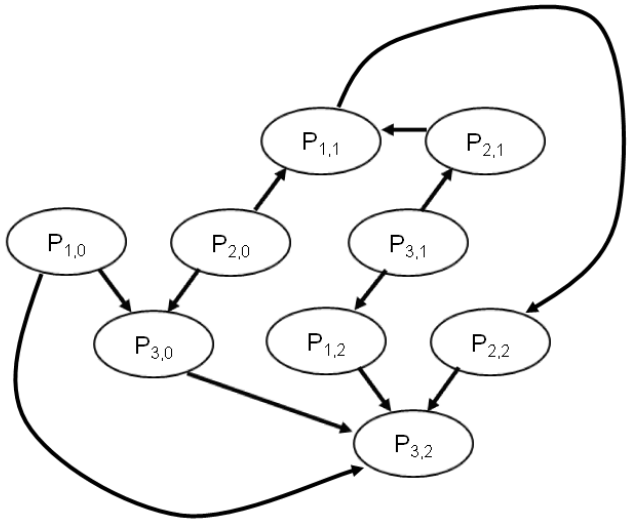
- So far we dealt with the "classic" (most simple) BN representation
  - static model,
  - binary variables.
- We will generalize towards
  - dynamic Bayesian networks
    - capable to capture system development in time,
  - other types of variables
    - categorical variables, continuous variables and their combinations.

- We will also discuss another type of graphs
  - directed $\longrightarrow$ undirected graphs,
  - Markov Random Fields
    - different properties and application.

Dynamic Bayesian Networks (DBNs)

- nodes = variables corresponding to the values of real variables in concrete time,
- real variables re-appear (see $P_{1,t}$, $P_{2,t}$, $P_{3,t}$ in figure),
- edges stand for causal relationship,
- additional constraints must be introduced

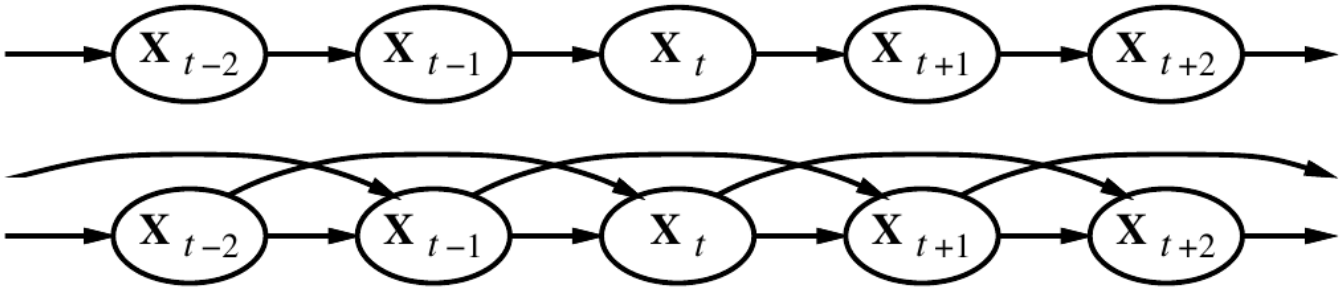
  (otherwise too many possible relationships),



- variable notation (**implicit constraints**)
  - t – **discrete time** (the choice of step size is problem dependent),
  - X_t – the set of unobservable **state** variables at time t,
    - $Position_t$, $Velocity_t$, $Acceleration_t$, $Battery_t$, ...
  - E_t – the set of observable **evidence** variables at time t,
    - $Odometry_t$, ...
  - $X_{a:b}$ – X_a, X_{a+1}, ..., X_{b-1}, X_b.

Dynamic Bayesian Networks (DBNs)

- Constraint #1: nth order **Markov assumption**
  - transition model: $Pr(X_t|X_{0:t-1}) = Pr(X_t|X_{t-n:t-1})$



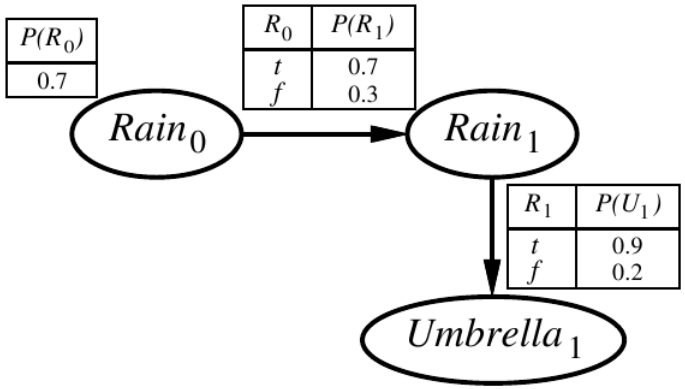
  - sensor model: $Pr(E_t|X_{0:t}, E_{0:t-1}) = Pr(E_t|X_t)$
- Constraint #2: **stationary** process
  - neither transition nor sensor model changes in time

    $\forall t : Pr(X_t|X_{t-n:t-1}) = Pr(X_{t+1}|X_{t-n+1:t})$

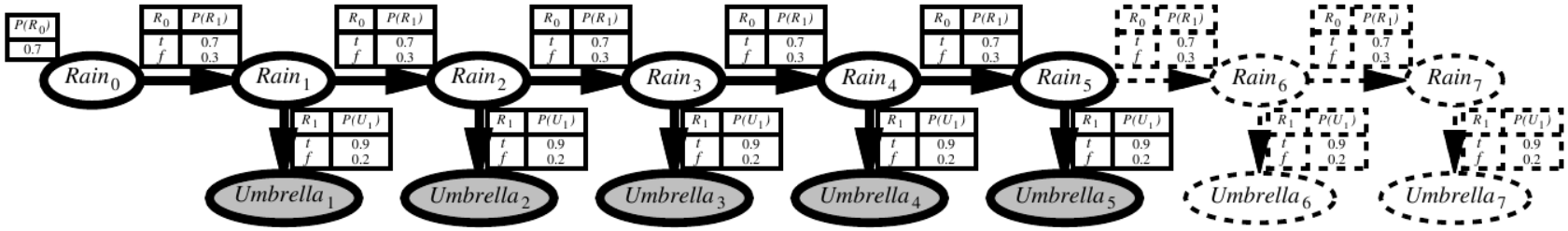
    $\forall t : Pr(E_t|X_t) = Pr(E_{t+1}|X_{t+1})$
- Under the first-order Markov assumption, DBN can be defined by
  - initial conditions: $Pr(X_0)$,
  - transition and sensor model (for single t only).

DBN example (Russell, Norvig: AIMA)

- the network is defined by the initial conditions and single **slice** (transition and sensor model)

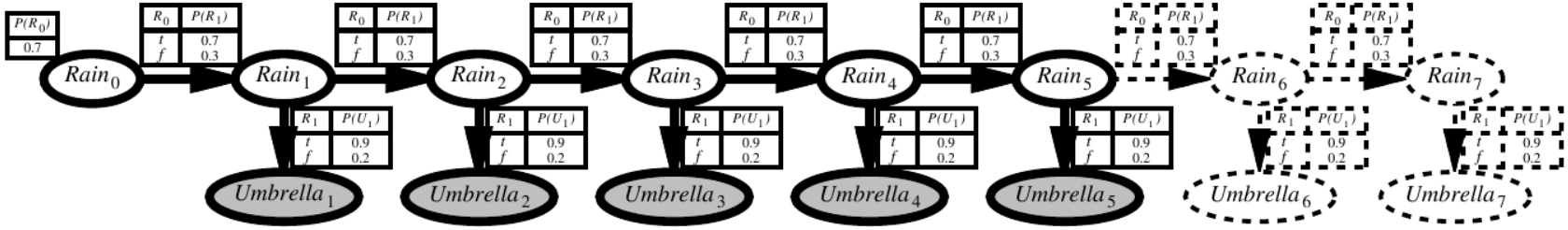


- the network can be unrolled up to an arbitrary time step.



DBN example (Russell, Norvig: AIMA) – continuation

- classical BN inference enumerates prob of arbitrary event given arbitrary evidence,
- naïve approach may ignore the split of variables (state vs evidence), or query the past states,
- what is the difference in prob of rain at day 1 knowing it did not rain at day 4 and umbrellas appeared at day 2?
  - $Pr(r_1) = Pr(r_0, r_1) + Pr(\neg r_0, r_1) = Pr(r_0)Pr(r_1|r_0) + Pr(\neg r_0)Pr(r_1|\neg r_0) = 0.58$,
  - Matlab BNT, jtree_inf_engine: $Pr(r_1|\neg r_4, u_2) = 0.68$,
  - rain prob at day 1 increased with the evidence,
- the number of time steps can be large/unlimited
  - only specialized tasks can be solved due to computational costs.



DBN inference tasks

- **filtering** – $Pr(X_t|e_{1:t})$
  - get the distribution of current state knowing (all) past evidence variables,
  - the fundamental task in decision making of a rational agent,
- **prediction** – $Pr(X_{t+k}|e_{1:t}), k > 0$
  - unlike filtering, a future state is of interest (or filtering without late evidence),
- **smoothing** – $Pr(X_k|e_{1:t})$, $0 \leq k < t$
  - improved estimate of past states, essential for learning,
- **most likely explanation** – $\arg\max_{x_{1:t}} Pr(x_{1:t}|e_{1:t})$
  - note it is not repeated smoothing – states mutually interact,
  - applied e.g. in speech recognition
    (speech signal $\rightarrow$ sentence and not the sequence of the most likely characters).

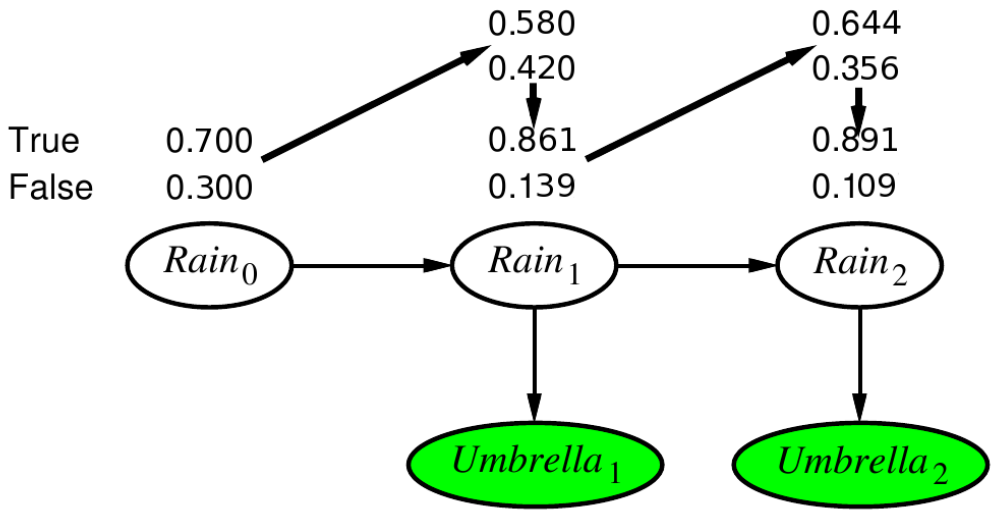
Filtering

- the aim is to devise a recursive state estimation algorithm
  - $Pr(X_{t+1}|e_{1:t+1}) = f(e_{t+1}, Pr(X_t|e_{1:t}))$,
  - filtering at time t and evidence at time $t+1 \rightarrow$ filtering at time $t+1$,

$$\begin{aligned} Pr(X_{t+1}|e_{1:t+1}) &= Pr(X_{t+1}|e_{1:t}, e_{t+1}) = \ldots \text{split evidence only} \\ &= \alpha Pr(e_{t+1}|X_{t+1}, e_{1:t}) Pr(X_{t+1}|e_{1:t}) = \ldots \text{apply Bayes theorem} \\ &= \alpha Pr(e_{t+1}|X_{t+1}) Pr(X_{t+1}|e_{1:t}) \ldots \text{follows from Markov property of evidence} \end{aligned}$$

- filtering was decomposed into
  - prediction $Pr(X_{t+1}|e_{1:t})$ and integration of the last evidence $Pr(e_{t+1}|X_{t+1})$,
- prediction will be solved by summing out X_t
  - $Pr(X_{t+1}|e_{1:t}) = \sum_{x_t} Pr(X_{t+1}|x_t, e_{1:t}) Pr(x_t|e_{1:t}) = \sum_{x_t} Pr(X_{t+1}|x_t) Pr(x_t|e_{1:t})$,
- the result is a time and space constant (independent of t) recursive function f
  - $f_{1,t+1} = \text{FORWARD}(f_{1:t}, e_{t+1})$, where $f_{1:t} = Pr(X_t|e_{1:t})$

Filtering – example



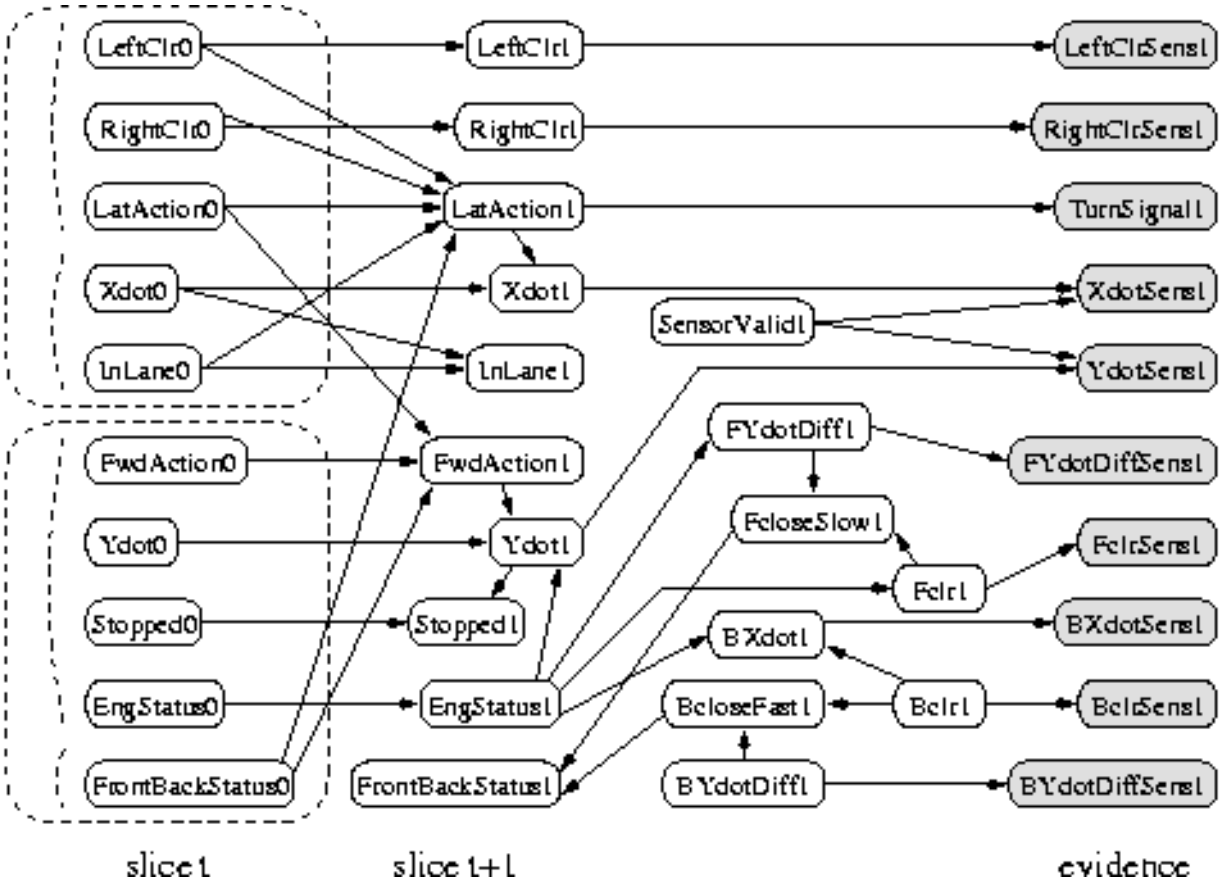
- we observe u_1 and u_2 – what is rain prob at day 1 (disregarding u_2) and rain prob at day 2?
  - $Pr(R_1|u_1) = \alpha_1 Pr(u_1|R_1) \sum_{R_0 \in \{r_0, \neg r_0\}} Pr(R_1|R_0) Pr(R_0)$,
  - $Pr(r_1|u_1) = \alpha_1 \times 0.9 \times (0.7 \times 0.7 + 0.3 \times 0.3) = \alpha_1 \times 0.9 \times 0.58$,
  - $Pr(\neg r_1|u_1) = \alpha_1 \times 0.2 \times (0.3 \times 0.7 + 0.7 \times 0.3) = \alpha_1 \times 0.2 \times 0.42$,
  - $\alpha_1 = 1.65 \rightarrow Pr(r_1|u_1) = 0.861,\ Pr(\neg r_1|u_1) = 0.139$
  - $Pr(R_2|u_1, u_2) = \alpha_2 Pr(u_2|R_2) \sum_{R_1 \in \{r_1, \neg r_1\}} Pr(R_2|R_1) Pr(R_1|u_1)$,
  - $Pr(r_2|u_1, u_2) = \alpha_2 \times 0.9 \times (0.7 \times 0.861 + 0.3 \times 0.139) = \alpha_2 \times 0.9 \times 0.644$,
  - $Pr(\neg r_2|u_1, u_2) = \alpha_2 \times 0.2 \times (0.3 \times 0.861 + 0.7 \times 0.139) = \alpha_2 \times 0.2 \times 0.356$,
  - $\alpha_2 = 1.54 \rightarrow Pr(r_2|u_1, u_2) = 0.891,\ Pr(\neg r_2|u_1, u_2) = 0.109$

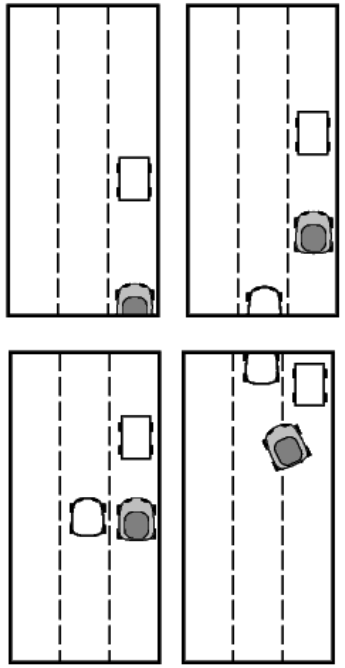
DBN vs HMM

- Hidden Markov Model (HMM)
  - frequent application in speech recognition or biomedicine,
- HMM is a special (and trivial) case of dynamic Bayesian network
  - state is described by the only variable (usually there is a single evidence variable as well),
  - i.e., Rain-Umbrella example represents both DBN and HMM,
- HMM is a statistical model doable by polynomial algorithms in the number of time steps
  - works in simple worlds = the state and sensor variable have a reasonable number of values
    - a set of discrete (evidence or state) variables can make a single variable (whose values correspond to tuples),
  - simple tasks prevail, HMMs appear more frequently than their DBN generalizations,
  - however, DBNs are more efficient than HMMs in complex tasks with sparse dependencies
    - DBN decomposes complex systems using independence,
    - exponentially fewer parameters than HMMs,
    - ex.: 10 binary state variables, 2 parents each
      - DBN has $10 \times 2^2 = 400$ parameters, HMM has $2^{10} \times 2^{10} \approx 10^6$ parameters.

Bayesian Automated Taxi (project, IJCAI paper, 1995)

- the project of autonomous vehicle in regular highway traffic
  - overtaking a slower vehicle and bypassing a stationary vehicle,
  - response to the maneuvers of other drivers or to the change of the number of lanes.





DBNs – overview

- so far, we applied BNs to model constant relationships among different variables
  - time was not concerned,
  - or the model captured a single moment.
- often we need to track time changes
  - the state of the world is described and its dynamics is modeled,
- for the sake of feasibility, constraints must be introduced
  - Markov assumption: the current state depends solely on a limited history of past states,
  - stationarity: transition and evidence probabilities do not change in time, slices stay identical,
  - discrete time: slices (world states at particular time steps) can be distinguished,
  - explicit prior split of variables: hidden/state and observable/evidence variables,
- for each time step a whole slice (state and sensor model) is copied
  - the network is unrolled then,
  - DBN can be understood as an equivalent of static BN,
  - however, this analogy is rarely used due to complexity reasons,
  - new standard tasks: filtering, prediction, smoothing, most likely explanation.

Other types of variables in BN

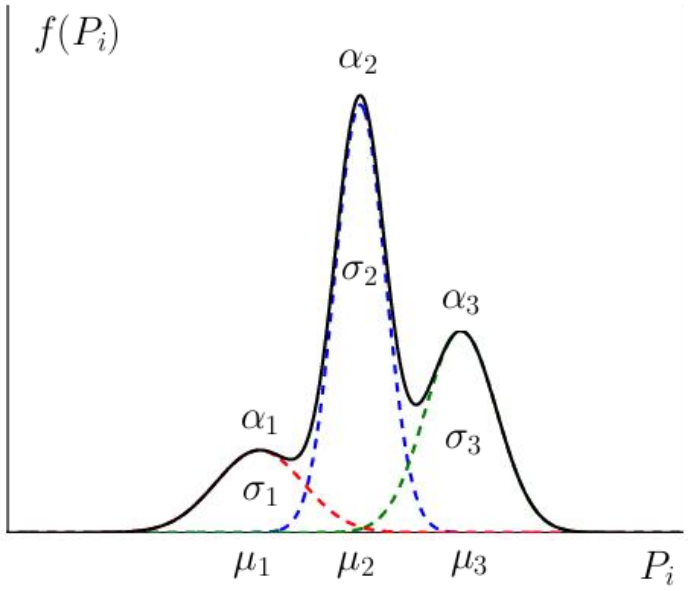
- BN must record $Pr(P_i|parents(P_i))$ by definition
  - CPT is a special case of more general **conditional probability distribution** (CPD),
- categorical variables
  - (can take on one of a limited, and usually fixed, number of possible values),
  - CPTs again, they just may grow more quickly with the number of parents,
    - q_i ... the number of unique instantiations of P_i parents,,
    - r_i ... the number of distinct P_i values,
    - r ... the maximum number of distinct values of one network variable ($r = \max_{i=1...n} r_i$),
    - the total number of independent BN parameters

      $$K = \sum_{i=1}^{n} q_i(r_i - 1),$$

    - for binary variables $r_i = 2$ a $q_i = 2^{|parents(P_i)|}$,
    - for categorical variables $2 \leq r_i \leq r$ a $q_i \leq r^{|parents(P_i)|}$.

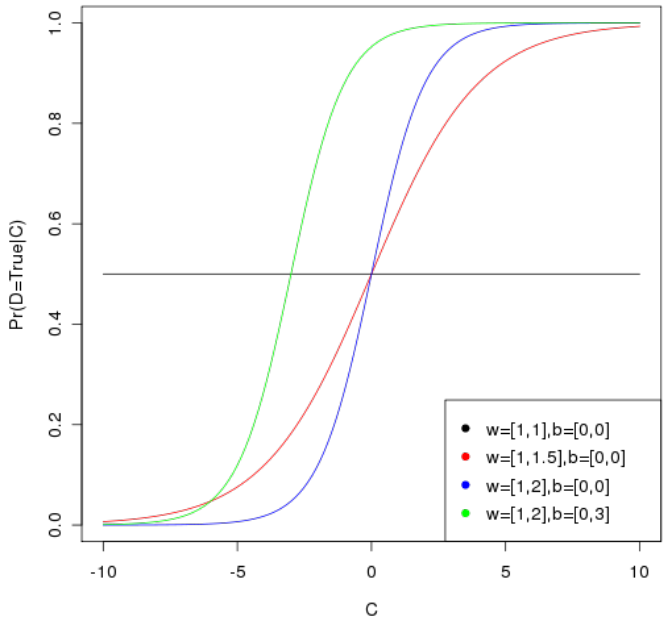
Other types of variables in BN

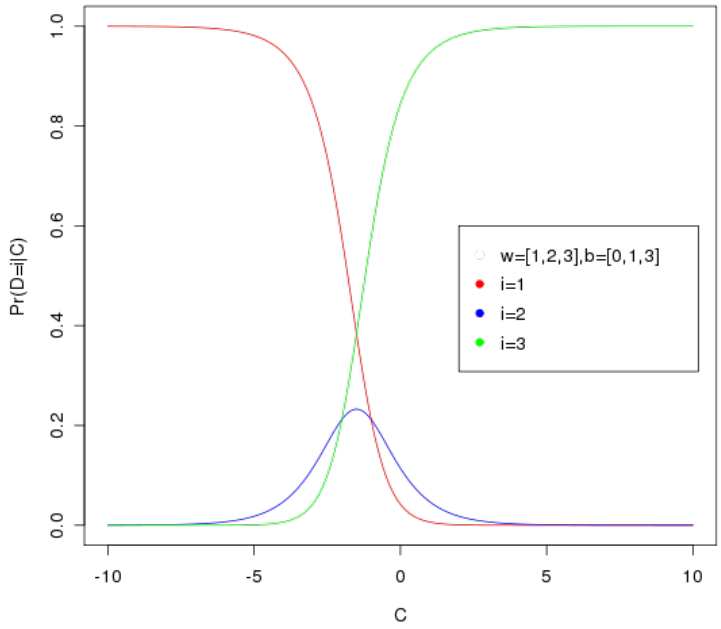
- continuous variables
  - CPT gets infinite, discretization $\longrightarrow$ categorical variable
    - dilemma: large CPTs vs information loss,
  - representation/approximation with canonical probability distributions
    - finite, often simple parametrization,
    - the most often option: normal (Gaussian) distribution and its mixtures,
    - figure: definition of a continuous node without any parent with 9 parameters.



Categorical variable with a continuous parent

- CPD for a categorical child D with a continuous parent C
  - soft thresholding using softmax function (multinomial logistic function)
    $$Pr(D = d_i | C = c) = \frac{e^{w_i \times c + b_i}}{\sum_{j=1}^{|D|} e^{w_j \times c + b_j}}$$
  - $\vec{w}$ – gives softness (slope) of the threshold, $\vec{b}$ – gives position of the threshold,
- applicable even for multiple continuous parents
  - $\mathbf{W}$ and $\mathbf{B}$ in a matrix format, contribution of the individual parents gets combined,
- binary $D \rightarrow$ softmax $\approx$ sigmoid function,
- ternary D – probabilities of target values,



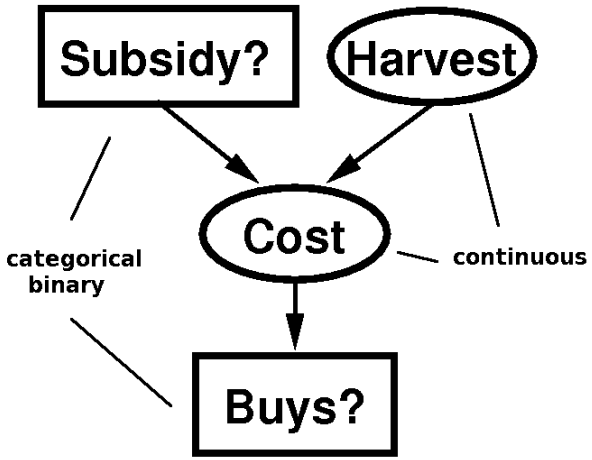


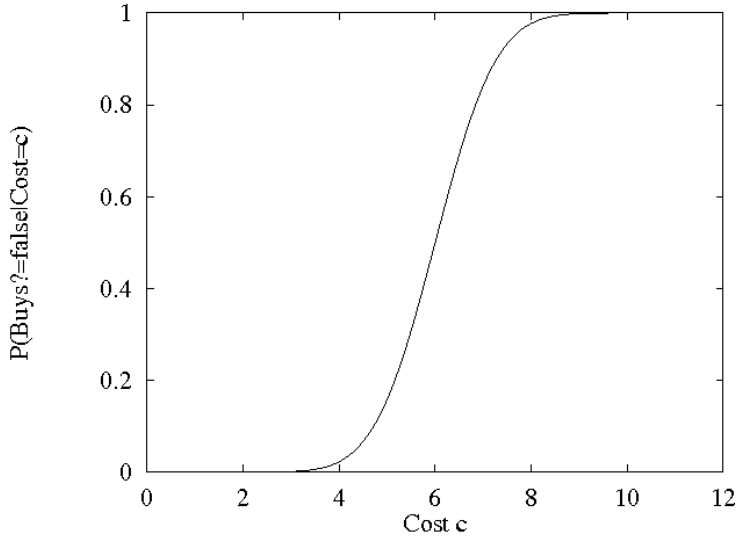
Continuous child node, hybrid parents

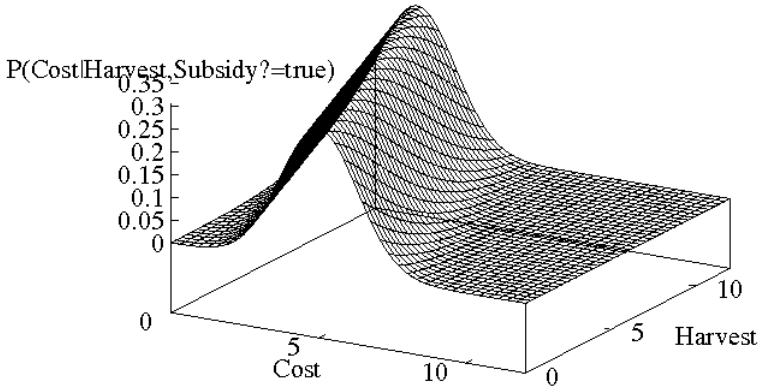
- a continuous child C_p having continuous parent C_r and categorical parent D
  - the most common CPD form is the linear Gaussian (LG) model
    - mean child value varies linearly with the continuous parent, variance is fixed,
      $Pr(C_p = c_p | C_r = c_r, D = d) = N(\mu_d = a_d c_r + b_d, \sigma_d)$
    - reasonable model provided that the range of continuous parent is likely to be narrow,
  - we define a LG child function for each value of categorical parent
    - for each value d a distinct set of parameters a_d, b_d and σ_d is defined,
- generalization for more parents
  - mean child value corresponds to a linear combination of values of continuous parents,
  - a new LG parametrization must be given for each possible assignment to categorical parents when having more of them.

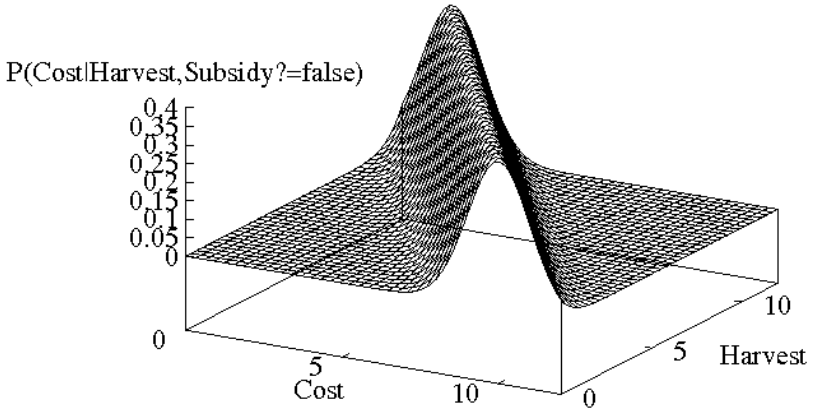
Continuous nodes – example (Russell, Norvig: AIMA)

- Fruit sales are influenced by price, the price is given by harvest yields and government subsidy.







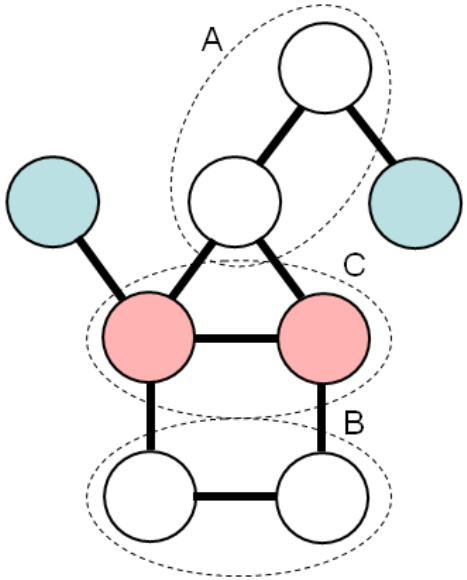


Continuous variables – summary

- parametrization of continuous variables CPDs
  - commonly with normal distribution and its mixtures, linearized by continuous parents,
  - in general not more difficult/demanding than for categorical variables,
  - several combinations for connections wrt node types exist (see BNT for Matlab),
- the global joint probability model
  - having categorical softmax nodes and continuous LG nodes ...
  - ...joint probability takes a form of multivariate Gaussian
  - (over all continuous variables for each combination of discrete variable values).
- inference in principle does not change
  - nevertheless, more difficult, all the algorithms cannot be applied,
  - they need to be generalized (dealing with atomic events etc.),
- new methods for learning from data appear
  - softmax is in BNT learned by the iteratively reweighted least squares (IRLS) algorithm,
  - for other node and connection models other dedicated algorithms.

Undirected graphs

- undirected graph model = Markov network = **Markov Random Field** (MRF),
- the same aim as the directed models
  - to factorize/decompose the joint probability using conditional independence,
- notion of conditional independence in undirected graphs
  - simple graph separation – in comparison with directed d-separation it is trivial,
  - $A \perp\!\!\!\perp B|C$: all the paths from A from B must disappear when all the nodes in C removed.



Undirected graphs

- joint probability factorization is based on the definition of conditional independence
  - for two nodes not connected by an edge it holds
    $Pr(P_i, P_j | P_{\setminus\{i,j\}}) = Pr(P_i | P_{\setminus\{i,j\}}) Pr(P_j | P_{\setminus\{i,j\}})$
  - Hammersley-Clifford theorem
    $Pr(P_1, \ldots, P_n) = \frac{1}{Z} \prod_{C \in mcl\{G\}} \psi_C(P_C)$
    Z ... normalization const, $mcl\{G\}$ ... set of maximal cliques G, ψ_C ... potential function
- possible difficulties
  - factors (usually) do not have interpretation (they are not probabilities),
  - product of factors must be normalized (to express the joint probability),
- potential functions
  - HC theorem assumes their strict positivity,
  - traditionally given as an exponential energy function E
    $\psi_C(P_C) = \exp^{-E(P_C)}$
  - joint probability corresponds to the sum of energies of all maximal cliques.

MRF: example (Bishop: Graphical models, chapter 8)

- noise filtering in a binary image (2 colors)

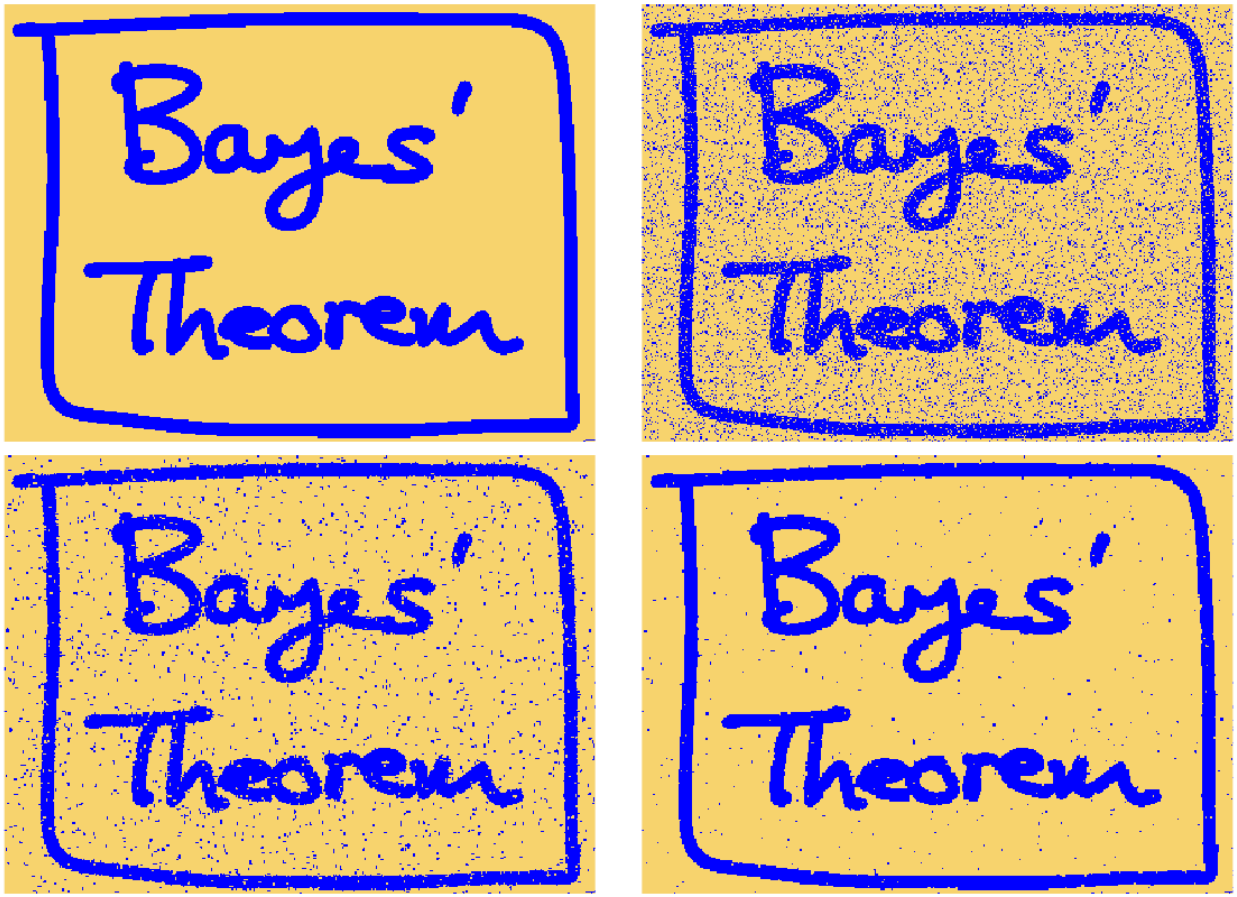
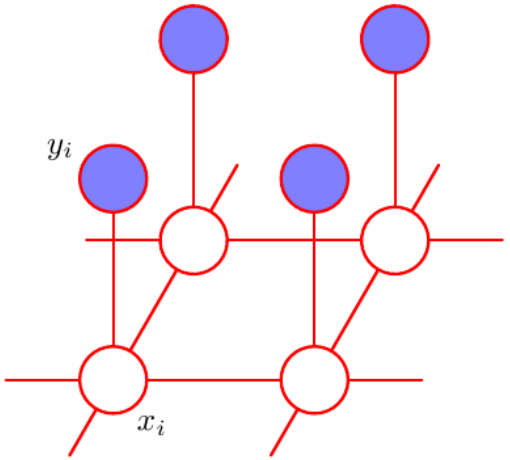


Figure 8.30 Illustration of image de-noising using a Markov random field. The top row shows the original binary image on the left and the corrupted image after randomly changing 10% of the pixels on the right. The bottom row shows the restored images obtained using iterated conditional models (ICM) on the left and using the graph-cut algorithm on the right. ICM produces an image where 96% of the pixels agree with the original image, whereas the corresponding number for graph-cut is 99%.

MRF: example (Bishop: Graphical models, chapter 8)

- structure of the model is proposed a priori
  - color of the adjacent pixels strongly correlated,
  - distant pixels can be ignored (distant=not an immediate neighbor $\rightarrow$ first-order model)
    - a regular grid of nodes X_i, $X_i \in \{-1, 1\}$
  - noise is modeled by another "layer" of nodes Y_i, $Y_i \in \{-1, 1\}$
    - low-level noise ($Pr(noise) \ll 0.5$) $\rightarrow$ strong correlation with X_i,
  - two types of maximal cliques with 2 nodes
    - adjacent X_i and X_j, connected Y_i and X_i,
- corresponds to the Ising model.

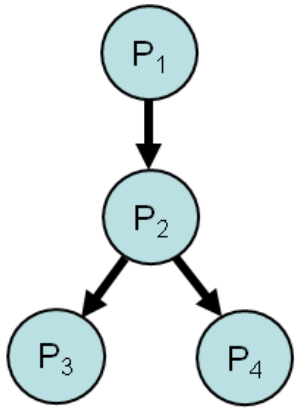


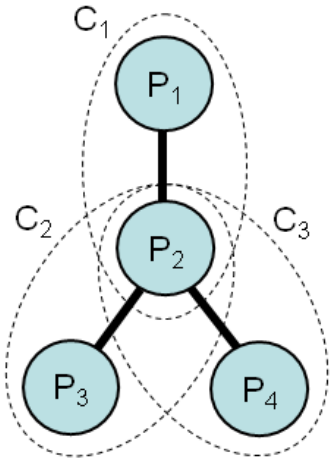
MRF: example (Bishop: Graphical models, chapter 8)

- learning of joint probability $Pr(\mathbb{X}, \mathbb{Y})$
  - clique energy: $-\eta X_i Y_i$ and $-\beta X_i X_j$,
    - η and β positive parametrization constants,
    - in case of agreement of node signs the energy is lower and the probability is higher,
  - the energy function
    $E(\mathbb{X}, \mathbb{Y}) = h \sum_i X_i - \beta \sum_{\{i,j\}} X_i X_j - \eta \sum_i X_i Y_i$,
    - first term prefers one of the colors (background color is more frequent a priori),
  - the target probability
    $Pr(\mathbb{X}, \mathbb{Y}) = \frac{1}{Z} \prod_C \psi_C(P_C) = \frac{1}{Z} \exp -E(\mathbb{X}, \mathbb{Y})$
- de-noising = learning of the model
  - Y_i is observed (noisy image), we search for X_i such that $Pr(\mathbb{X}, \mathbb{Y})$ is maximized,
  - can be done by a gradient-descent algorithm (which concerns pixel coordinates)
    1. initialization $\mathbb{X}$ ($\forall i \; X_i = Y_i$),
    2. $\forall i$ gradually and repeatedly (in a random order or in a systematic order)
       (a) determine the the total energy for $X_i = +1$ and $X_i = -1$,
       (b) take the setting with the lower energy,
    3. stop if energy did not decrease (after all the cycle $\forall i$).

Transformation from directed to undirected model

- an important part of the junction tree inference algorithm,
- clique potentials must concern the conditional probabilities in directed graph,
- this is easy to meet in directed graphs where nodes have at most one parent.





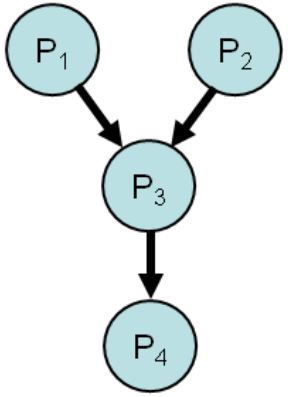
$$Pr(P_1,\ldots,P_4) = \frac{1}{Z}\prod_i \psi_{C_i}(P_{C_i})$$

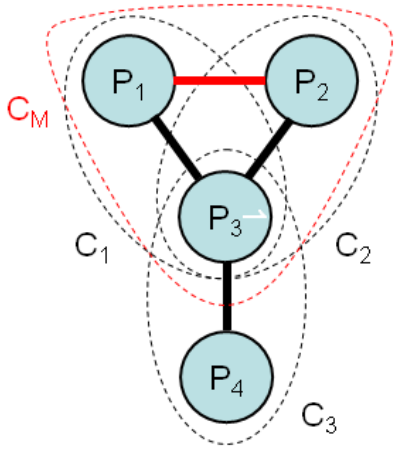
$$Pr(P_1,\ldots,P_4) = \\ = Pr(P_1)Pr(P_2|P_1)Pr(P_3|P_2)Pr(P_4|P_2)$$

$$\psi_{C_1}(P_{C_1}) = \psi_{1,2}(P_1,P_2) = Pr(P_1)Pr(P_2|P_1)$$
$$\psi_{C_2}(P_{C_2}) = \psi_{2,3}(P_2,P_3) = Pr(P_3|P_2)$$
$$\psi_{C_3}(P_{C_3}) = \psi_{2,4}(P_2,P_4) = Pr(P_4|P_2)$$
$$Z = 1$$

Transformation from directed to undirected model

- the networks with converging nodes must be moralized (see junction tree algorithm),
  - otherwise, there are some conditional probs that cannot be applied within any of the cliques!
  - undirected graph would encode a different set of (conditional) independence relationships.





$$Pr(P_1,\ldots,P_4) = \frac{1}{Z}\prod_i \psi_{C_i}(P_{C_i}) \quad (Z=1)$$

$$Pr(P_1,\ldots,P_4) = \\ = Pr(P_1)Pr(P_2)Pr(P_3|P_1,P_2)Pr(P_4|P_3)$$

$$\psi_{C_M}(P_{C_M}) = \psi_{1,2,3}(P_1,P_2,P_3) = \\ = Pr(P_1)Pr(P_2)Pr(P_3|P_1,P_2)$$

$$\psi_{C_3}(P_{C_3}) = \psi_{3,4}(P_3,P_4) = Pr(P_4|P_3)$$

- transformation from undirected to directed model more difficult and less frequent.

Summary

- BNs belong to a large class of graphical stochastic models,
- BN frequently used in
  - knowledge-based systems, causal diagrams,
- changes in the formalism or additional constraints lead to models applicable in other domains
  - BN vs DBN/HMM
    - speech recognition, signal processing, time series,
  - BN vs MRF
    - gases and fluids, society, images,
- variables can be both discrete and continuous
  - for simplicity and educational reasons this course focused on binary variables.

Applications of Bayesian networks

- PATHFINDER (Stanford, 1980')
  - diagnosis of the lymph nodes diseases,
  - tested on 53 patients, diagnostics outperformed the best pathologists,
  - followed by a series of medical diagnostic systems,
- other diagnostic systems/applications
  - interpretation of electromyography (EMG),
  - gas turbines,
  - corn yields,
- gene expression modeling
  - mutual interactions among genes (gene regulatory networks),
  - G1 encodes an enzyme that stimulates expression of another gene G2,
  - G1 "causes" G2.
- interactive problem solving in MS Windows systems
  - troubleshoots both HW and SW problems,
  - similarly car diagnostics, jet engines etc,
- modeling in geology, hydrology, sociology,
- among others BN Weka classifier.

BN – summary

- probability is a rigorous tool for uncertainty modeling,
- each **atomic** event has its joint probability,
- queries answered by aggregation (adding with subsequent division) of atomic events,
- needs to be simplified in non-trivial domains,
- the tool is independence and conditional independence
  - **Bayesian network** = graph (ind. relationships) + conditional probability distributions (quantities),
  - inference is still NP-hard wrt to the number of variables,
  - solution can be in special network types (singly connected, sparse networks), ex.: trees – at most one parent,
  - or approximate inference,
- BNs can be learned from data, two tasks exist
  - parameter learning, less difficult, can be seen as a subtask of ...
  - structure learning.

Advantages of Bayesian networks

- the problem structure given as a graph is clear and its semantics understandable for user
  - it can naturally be interpreted in the cause-effect way, one can also manually generate it,
  - it implements both causal and diagnostic inference,
  - evidence can be propagated in an arbitrary way – a universal model,
  - for comparison, a neural network as a " black box",
- ability to merge prior knowledge with data
  - existing network (typically outlined by an expert) can be modified to explain the data,
  - learning purely from data is a special case (empty/random init model),
- theoretically sound, robust, widely applicable
  - small modifications in model result in small perturbations in results, does not overfit,
  - application in exploration analysis, classification or regression,
- ability to deal with missing values and hidden variables
  - model can estimate a missing value given its relations to other values/variables,
  - hidden variable may explain "suspicious" dependency between two observed variables,
- what stands against?
  - learning is difficult,
  - there is no general and widely-applicable algorithm applicable without its understanding.

Recommended reading, lecture resources

- Russel, Norvig: **AI: A Modern Approach**, Uncertain Knowledge and Reasoning (Part IV)
  - representation of CPDs with continuous variables,
  - temporal probability models (chap. 15, not in the first edition),
  - Google books (isbn:0136042597),
- Murphy: **A Brief Introduction to Graphical Models and Bayesian Networks.**
  - BN Toolbox and continuous variables?
  - http://www.cs.ubc.ca/~murphyk/Bayes/bayes.html,
- Bishop: **Pattern Recognition & Machine Learning**, Graphical Models (Chapter 8)
  - a general chapter, in this lecture it inspired the part on undirected graphs,
  - http://research.microsoft.com/en-us/um/people/cmbishop/prml/Bishop-PRML-sample.pdf.
- Forbes et al.: **The BATmobile: Towards a Bayesian Automated Taxi**, IJCAI 1995
  - TAXI case study,
  - http://www.cs.berkeley.edu/~russell/papers/ijcai95.