SMU

- MODELY NAUČITELNOSTI PAC A ONLINE
- NAUČITELNOST KONJUNKCÍ A DISJUNKCÍ
- BAYESOVSKÉ SÍTĚ
- REINFORCEMENT LEARNING

- SEQUENTIAL

$$\mu(x y_{\leq n}) = \mu(x_1)\mu(y_1|x_1)\cdot\mu(x_2|x_1 y_1)\cdot\ldots\mu(x_n|x y_{<n})\mu(y_n|x_n x y_{<n})$$

- POLICY

$$y_n = \pi(x y_{<n} x_n)$$

- DETERMINISTIC

$$\mu(y_n|x y_{<n} x_n) = 1 \quad \text{FOR} \quad y_n = \pi(x y_{<n} x_n)$$
$$= 0 \quad \text{OTHERWISE}$$

- OBSERVATION, REWARD

$$x_n = (o_n, r_n)$$

$$\mu(x_n|x y_{<n}) = \mu(o_n, r_n|x y_{<n}) = \mu(o_n|r_n x y_{<n})\cdot\mu(r_n|x y_{<n}) =$$

$$= \mu(r_n|o_n x y_{<n})\cdot\mu(o_n|x y_{<n})$$

- NON-SEQUENTIAL

$$\mu_0(o_n|r_n, x y_{<n}) = \mu_0(o_n)$$

- REWARDS ~~DEPENDING~~ DEPEND ON PREVIOUS OBSERVATION AND ACTION ONLY

$$\mu_R(r_n|o_n, x y_{<n}) = \mu_R(r_n|o_{n-1} y_{n-1})$$

- BATCH LEARNING
  - NONSEQUENTIAL
  - LEARNING PHASE $k = 1, 2 \dots K$
  - ACTION PHASE $\lambda = K+1, K+2 \dots$
  - AGENT DOESN'T CHANGE ITS DECISION POLICY IN ACTION PHASE

    IF $\lambda, \lambda' > K$ AND $x_\lambda = x_{\lambda'}$ THEN $y_\lambda = y_{\lambda'}$

    AND IGNORES REWARDS
  - FOR $\lambda > K$

    $y_\lambda = \pi(x_{y_{\leq K}}, o_\lambda)$ - POLICY DEPENDS ON ACTUAL OBSERVATION AND STEPS OF LEARNING PHASE

- REWARDS
  - WE WANT TO MAXIMIZE SUM OF REWARDS UP TO FINITE HORIZONT $m \in N$

    $r_1 + r_2 + \dots r_m$
  - OR DISCOUNTED REWARD

    $\sum_{\lambda=1}^{\infty} r_\lambda \gamma_\lambda$ WHERE $\forall \lambda : \gamma_\lambda \geq 0$ AND $\sum_{i=1}^{\infty} \gamma_i < \infty$
  - BUT REWARDS ARE PROBABILISTIC
    - WE WANT TO MAXIMALIZE EXPECTED COMULATIVE REWARD

      $\sum_{r_{\leq m}} \gamma^{w}_{R}(r_{\leq m} \mid x_{y_{\leq m}}) \cdot (r_1 + r_2 + \dots r_m)$
    - OR DISCOUNTED REWARD

      $\lim_{m \to \infty} \sum_{r_{\leq m}} \gamma^{w}_{R} \cdot (r_{\leq m} \mid y_{\leq m}) \sum_{\lambda=1}^{m} r_\lambda \gamma_\lambda$
- IN BATCH LEARNING WE DONT CARE ABOUT REWARDS IN LEARNING PHASE

  $\sum_{r_\lambda \in R} \gamma^{w}_{R}(r_\lambda \mid x_{y_{\leq K}}) \cdot r_\lambda$

- ENVIRONMENT STATE
  - WE DON'T HAVE TO LOOK AT WHOLE SEQUENCE OF HISTORY
  - WE CAN LOOK ONLY AT STATE OF ENVIRONMENT

$$\mu(x_t | e_t)$$

  - INITIAL STATE IS "DUMMY"

$$\mu(x_1 | s^*) = \mu(x_1)$$

  - UPDATE DISTRIBUTION

$$\varepsilon(e_t | e_{t-1}, x_{t-1}, y_{t-1})$$

- $\mu(x_t | e_t) = \mu(x_t | x y_{<t})$
  - ASSUMPTION

$$|E| < E_{MAX}$$

    - STATES ARE BOUNDED BY E_{MAX}
    - THERE ARE LESS STATES THAN ALL POSSIBLE HISTORIES
  - PERCEPTS DEPEND ON STATE, BUT STATE DOESN'T DEPEND ON PERCEPTS

$$\varepsilon(e_t | e_{t-1}, x_{t-1}, y_{t-1}) = \varepsilon(e_t | e_{t-1}, y_{t-1})$$

- AGENT STATE

$$|A| < A_{MAX}$$

  - ACTION IS DETERMINED BY AGENT'S STATE

$$y_t = \pi(a_t)$$

$$y_t = \pi(a_t, o_t)$$

  - STATE IS UPDATED DETERMINISTICALLY

$$a_t = A(a_{t-1}, x_t)$$

  - FOR BATCH LEARNING, THE AGENT DOESN'T UPDATE ITS STATE IN ACTION PHASE $a_t = a_k \quad \forall t \geq k$

- ON-LINE CONCEPT LEARNING

  - CONCEPT OF ENVIRONMENT

    - $C = \{o \in O \mid \mu^{w}_{0}(o|1) > 0\}$

    - SET OF OBSERVATIONS WHICH CAN BE GENERATED FROM STATE 1

    - OBSERVATIONS COMING FROM THIS CONCEPT ARE POSITIVE OBSERVATIONS

    - $O \setminus C$ ARE NEGATIVE OBSERVATIONS

    - AGENT PERFORMS CLASSIFICATION LEARNING

  - AGENT TRIES TO PREDICT THE STATE OF ENVIRONMENT

    $Y = E = \{0, 1\}$

  - REWARD

    $$r_{t+1} = \begin{cases} 0 & \text{IF } y_t = \gamma_t \\ -1 & \text{OTHERWISE} \end{cases}$$

  - AGENT'S (HYPOTHESIZED) CONCEPT

    - $C(a_t) = \{o \in O \mid \pi(a_t, o) = 1\}$

    - AGENT WANT TO IDENTIFY ITS CONCEPT WITH CONCEPT OF ENVIRONMENT

      $C(a_t) = C$

  - AGENT'S HYPOTHESIS

    $h_t = a_t$

    $\gamma_t = \pi(h_t, o_t)$

    $C(h_t) = \{o \in O \mid \pi(a_t, o) = 1\}$

    $C(h_t) = C$

- UPDATE RULE

$$a_t = (o_t, h_t)$$

$$A(a_{t-1}, x_t) = A((o_{t-1}, h_{t-1}), (o_t, r_t)) = (o_t, h_t)$$

- GENERALIZING AGENT

  - USES CONJUNCTIONS

  - $\gamma_t = \pi(h_t, o_t) = \begin{cases} 1 & \text{IF } o_t \models h_t \\ 0 & \text{OTHERWISE} \end{cases}$

  - STARTS WITH THE MOST SPECIFIC HYPOTHESIS

    - CONJUNCTION OF ALL PROPOSITIONAL VARIABLES AND THEIR NEGATION

    - INITIAL HYPOTHESIS IS GENERALIZED TOWARDS THE CORRECT ONE

    - $h_1 = P_1 \wedge \neg P_1 \wedge P_2 \wedge \neg P_2 \wedge P_3 \wedge \neg P_3$

  - UPDATE RULE

$$h_t = \begin{cases} h_{t-1} & \text{IF } r_t = 0 \\ \text{DELETE}(h_{t-1}, o_{t-1}) & \text{OTHERWISE} \end{cases}$$

$$\text{DELETE}\left(\bigwedge_{i \in I} P_i \bigwedge_{j \in J} \neg P_j, (o^1, o^2, \dots, o^m)\right) = \bigwedge_{\substack{i \in I \\ o^i = 1}} P_i \bigwedge_{\substack{i \in I \\ o^i = 0}} \neg P_i$$

  - DELETE KEEPS LITERALS FROM h_{t-1} WHICH ARE CONSISTENT WITH o_{t-1}

- WE NEED TO ASUME THAT THERE EXISTS "CORRECT" CONJUNCTION h^*

$$\pi(h^*, o_t) = r_t \quad \forall o_t \in O$$

- AGENT MAKES MISTAKES ONLY ON POSITIVE EXAMPLES

$$h_t \supseteq h^*$$

- MISSTAKES ARE CORRECTED BY REMOVING AT LEAST ONE INCONSISTENT LITERAL

- GENERALIZING AGENT MAKES AT MOST $2n$ MISTAKES
  - $2n$ LITERALS (POSITIVE AND NEGATIVE) (REMOVES AT LEST ONE FOR EACH SAMPLE)
- CUMULATIVE REWARD

$$\sum_{k=1}^{m} r_k \geq -2n$$

- CAN ALSO LEARN DISJUNCTIONS

$$\neg(P_1 \vee P_2 \vee \dots \vee P_n) = \neg P_1 \wedge \neg P_2 \wedge \dots \wedge \neg P_n$$

- WE HAVE TO CHANGE

$$\bar{o}_k = (1-o_k^1, 1-o_k^2, \dots, 1-o_k^n) = \neg P_1 \wedge \neg P_2 \wedge \dots \wedge \neg P_n$$

$$\bar{x}_k = 1 - x_k$$

- SUBSUMPTION
  - SUBSUMPTION LATTICE
    - PARTIALLY ORDERED
    - EACH TWO ELEMENTS HAVE THEIR UNIQUE LEAST UPPER BOUND AND THE GREATEST LOWER BOUND

- $h_1 \subseteq h_2$ IMPLIES $h_2 \vdash h_1$ IF h_1 AND h_2 ARE CONJUNCTIONS

- h_1 ENTAILS h_2 IF ANY MODEL OF h_1 IS ALSO MODEL OF h_2

$$h_1 \vdash h_2$$

- FOR DISJUNCTION

$$h_1 \subseteq h_2 \text{ IMPLIES } h_1 \vdash h_2$$

- SEPARATING AGENT

  - h_t WILL BE REPRESENTED BY NON-LOGICAL MEANS

  - DEFINES HYPERPLANE IN $O = \{0,1\}^n$

  - $C(h_t)$ INCLUDE OBSERVATIONS LYING ABOVE HYPERPLANE

  - POLICY

$$y_t = \pi(h_t, o_t) = \begin{cases} 1 & \text{IF } h_t \cdot o_t > \frac{n}{2} \\ 0 & \text{OTHERWISE} \end{cases}$$

  - h_t IS n-TUPLE OF INTEGER VALUES BOUNDED BY CONSTANT $q \in N$

  - INITIAL HYPOTHESIS

$$h_1 = (1, 1, 1 \ldots, 1)$$

  - UPDATE RULE

$$h_t = \begin{cases} h_{t-1} \text{ IF } r_t = 0 \\ \text{UPDATE } (2, h_{t-1}, o_{t-1}) \text{ IF } h_{t-1} \cdot o_{t-1} \leq \frac{n}{2} \\ \text{UPDATE } (0, h_{t-1}, o_{t-1}) \text{ IF } h_{t-1} \cdot o_{t-1} > \frac{n}{2} \end{cases}$$

$$\text{UPDATE} : h_t^i = \begin{cases} \alpha \cdot h_{t-1}^i \text{ IF } o_{t-1}^i = 1 \\ h_{t-1}^i \text{ OTHERWISE} \end{cases}$$

  - INTEGER COUNTERPART OF PERCEPTRON ALGORITHM

  - MAKES AT MOST $2 + 2s \lg n$ MISTAKES

  - COMULATIVE REWARD IS

$$\sum_{t=1}^{n} r_t \geq -2 - 2s \lg n$$

  - PERFORMS BETTER IF NUMBER OF VARIABLES n IS LARGER THAN NUMBER OF RELEVANT VARIABLES

- HYPOTHESIS CLASS
  - SET OF ALL HYPOTHESES AN AGENT CAN EXPRESS
  - FINITE SET
  - GENERALIZING AGENT
    - ALL CONJUNCTIONS MADE OF AT MOST n VARIABLES
  - SEPARATING AGENT
    - SET OF q-BOUNDED n-TUPLES OF INTEGERS
  - EACH HYPOTHESIS CLASS INDUCES SET OF CONCEPTS

$$C(H) = \{c(h) \mid h \in H\}$$

    - CONCEPT CLASS
  - WE WOULD LIKE HUGE HYPOTHESIS SPACE FOR AGENT, WHICH INCREASES THE CHANCE THAT $C \in C(H)$, BUT IT INCREASES NUMBER OF MAXIMAL ERORS $|H|-1$

- VERSION SPACE AGENT
  - MISTAKE BOUND IMPROVED FROM $|H|-1$ TO $\lg|C|$
  - ON EACH OBSERVATION AGENT DISCARTS ALL HYPOTHESES FROM HYPOTHESIS CLASS WHICH ARE INCONSISTENT WITH THE OBSERVATION
  - VERSION SPACE AGENT'S STATE IS

$$a_t = (O_t, H_t)$$

    - O_t IS MEMORIZED OBSERVATION AND H_t IS SET OF HYPOTHESES
  - DECISIONS ARE DETERMINED BY VOTING AMONG ALL HYPOTHESES IN H_t

$$y_t = \pi(H_t, O_t) = \begin{cases} 1 \text{ IF } |\{h_i \in H_t \mid O_t \models h_i\}| > |H_t|/2 \\ 0 \text{ OTHERWISE} \end{cases}$$

- AGENT STARTS WITH FULL HYPOTHESIS STATE
- WE REMOVE ALL HYPOTHESIS IN CONSISTENT WITH o_{n-1}
- AGENT MAKES AT MOST $\lg |H|$ MISTAKES
- COMULATIVE REWARD IS

$$\sum_{n=1}^{m} r_n \geq -\lg |H|$$

- IF MISSTAKE IS MADE, AT LEAST HALF OF HYPOTHESIS ARE REMOVED. IN WORST CASE, THE LAST HYPOTHESIS IS CORRECT
- IT NEEDS A LOT OF MEMORY

- AGENT LEARNS CLASS H ON-LINE IF IT MAKES AT MOST POLY (n) IN THE ONLINE SCENARIO IF $C \in C(H)$. n IS SIZE OF OBSERVATIONS.
- AGENT THAT LEARNS HYPOTHESIS CLASS H ON-LINE IS SAID TO LEARN IT EFFICIENTLY IF IT SPENDS AT MOST POLYNOMIAL TIME (IN n) BETWEEN THE RECEIPT OF A PERCEPT AND GENERATION OF THE NEXT ACTION
- IF $|H|$ IS AT MOST EXPONENTIAL IN n, THAN VERSION SPACE AGENT LEARNS CLASS H ON-LINE
- VERSION SPACE AGENT DOESN'T LEARN H EFICIENTLY
- VAPNIK-CHERVONENKIS DIMENSION
  - CARDINALITY OF LARGEST SET $O' \subseteq O$ THAT IS SHATTERED BY H
  - NO UPPER BOUND ON THE NUMBER OF MISTAKES MADE BY AGENT IS SMALLER BY $VC(H)$

- BATCH LEARNING WITH GENERAL ON-LINE AGENTS

  - STANDARD ON-LINE AGENT

    - USES SINGLE HYPOTHESIS

    - CHANGES IT IF AND ONLY IF A MISSTAKE HAS BEEN MADE

  - IF STADARD ON-LINE AGENT RETAINS A HYPOTHESIS h_i FOR q STEPS $(h_i = h_{i+1} = \dots = h_{i+q})$ THEN $ERR(h_i) \leq \varepsilon$ WITH PROBABILITY AT LEAST $1 - e^{-q\varepsilon}$

  - SO WE CAN WAIT FOR $h_i = h_{i+1} = \dots = h_{i+q}$ TO HAPPEN AND THEN KEEP h_i WITH PROBABILISTIC ERROR BOUND

- CONSISTENT AGENT

  - EQUIVALENT OF VERSION SPACE AGENT, BUT FOR BATCH LEARNING

  - COLLECT ALL OBSERVATIONS

  - AT THE END OF TRAINING PHASE SELECTS ~~OR PICKS~~ RANDOM HYPOTHESIS CONSISTENT WITH ALL OBSERVATIONS

- PAC LEARNING MODEL

  - AGENT PROBABLY APPROXIMATELY LEARNS CLASS H (IN BATCH SETTING) IF AT THE END OF THE TRAINING PHASE IT PRODUCES R_K SUCH THAT $ERR(R_K) \leq \epsilon$ WITH PROBABILITY AT LEAST $1-\delta$ AND $K-1 = m \leq POLY(n, \frac{1}{\delta}, \frac{1}{\epsilon})$

  - IT LEARNS THE CLASS EFFICIENTLY IF IT SPENDS AT MOST POLYNOMIAL (IN THE SAME VARIABLES) TIME BETWEEN THE RECEIPT OF PERCEPT AND THE GENERATION OF THE NEXT ACTION IN THE TRAINING PHASE

- GENERALIZING AGENT EFFICIENTLY PAC-LEARNS CONJUNCTIONS

- ANY STANDARD AGENT LEARNING (EFFICIENTLY) A CONCEPT CLASS C ON-LINE, HAS A COUNTERPART WHICH (EFFICIENTLY) PAC-LEARNS C.

- IF $C \in CL(H)$ AND $|H|$ IS AT MOST EXPONENTIAL IN m THEN THE CONSISTENT AGENT PAC-LEARNS H.

- REINFORCEMENT LEARNING
  - AGENT INTERACT WITH ENVIRONMENT
  - LEARNS HOW TO ACT OPTIMALLY
  - WORD IS MARKOV DECISION PROCESS
  - TRANSITION AND REWARD IS NOT KNOWN
  - GOAL IS TO LEARN OPTIMAL POLICY
  - DIFFICULTIES
    - ACTION EFFECTS ARE NON-DETERMINISTIC
    - REWARDS ARE SPARSE AND DELAYED
    - GREEDY DOESN'T WORK
    - LARGE AND COMPLEX WORD

- PASSIVE LEARNING
  - FIXED POLICY π
  - IT LEARNS HOW GOOD THE POLICY IS BY INTARACTING WITH THE ENVIRONMENT (LEARNS $U^{\pi}(s)$)

- ACTIVE LEARNING
  - AGENT SEARCHES FOR OPTIMAL POLICY
  - IT EXPLORES MANY DIFFERENT ACTIONS IN MANY DIFFERENT STATES

- MODEL BASED LEARNING
  - LEARNS MDP MODEL
    - S - TRANSITION FUNCTION
    - R - REWARD FUNCTION
- MODE FREE LEARNING
  - DOESN'T LEARN MODEL DIRECTLY
  - ESTIMATES Q FUNCTION

- PASSIVE REINFORCEMENT LEARNING
  - DIRECT UTILITY ESTIMATION (DUE)
    - MODEL FREE
    - LEARN EXPECTED REWARD-TO-GO FROM OBSERVED REWARDS-TO-GO
    - REWARD-TO-GO OF STATE S IS THE SUM OF THE DISCOUNTED REWARDS FROM STATE AND UNTIL A TERMINAL STATE IS REACHED
    - EXPECTED REWARD-TO-GO MATCHES THE TRUE STATE UTICITY GIVEN THE POLICY
    - ESTIMATE $U^{\pi}(s)$ AS AVERAGE TOTAL REWARD OF EPOCHS CONTAINING S (IF SAMPLE GOES THROUGH S, WE TAKE IT FOR CALCULATION OF $U^{\pi}(s)$)
  - REDUCES RL TO SUPERVISED LEARNING
  - GUARANTEED CONVERGENCE, BUT SLOW
    - DOES NOT EMPLOY DEPENDENCE AMONG STATE UTILITIES
- ADAPTIVE DYNAMIC PROGRAMMING (ADP)
  - COMPLEX MODEL BASED METHOD
  - STEP 1
    - LEARN TRANSITION MODEL S AND REWARD FUNCTION R
    - STORE EACH R(S)
    - $S(s'|s,a)$ - KEEP TRACK HOW OFTEN WE ENTER s' FROM S BY ACTION a
  - STEP 2
    - PERFORM POLICY EVALUATION BASED ON UNDERLYING MDP
    - SOLVES n BELLMAN EQUATIONS $U^{\pi}(s) = R(s) + \gamma \sum_{s'} S(s'|s, \pi(s)) U^{\pi}(s')$

- TEMPORAL DIFFERENCE LEARNING (TD)

  - $U^{\pi}(s) = R(s) + \gamma \sum_{s'} S(s' | s, \pi(s)) U^{\pi}(s')$

  - INSTEAD OF SUMMING OVER ALL SUCCESSORS, ONLY ADJUST THE UTILITY OF THE STATE BASED ON THE SUCCESSOR OBSERVED IN THE TRIAL

    $U^{\pi}(s) \sim R(s) + \gamma U^{\pi}(s')$

  - TRANSITION MODEL IS NOT NEEDED

  - $U^{\pi}(s) \leftarrow U^{\pi}(s) + \alpha \cdot (R(s) + \gamma U^{\pi}(s') - U^{\pi}(s))$

  - α - LEARNING RATE

  - IF LEARNING RATE DECREASES, IT WILL CONVERGE TO TRUE UTILITY VALUE

- ACTIVE RL

  - FIND OPTIMAL POLICY

  - ACTIVE ADP

    - DO NOT KEEP POLICY FIXED

    - ALWAYS TAKE ACTION THAT MAXIMIZES EXPECTED REWARD

      - GIVEN EXISTING ENVIRONMENT MODEL AND UTILITY ESTIMATES

        $U(s) = R(s) + \gamma \max_{a} \sum_{s'} S(s' | s, a) U(s')$

    - BUT CHOOSING OPTIMAL ACTION LEAD TO SUBOPTIMAL RESULTS

    - WE NEED MORE EXPLORATION

    - ε -GREEDY STRATEGY

      - ε - RANDOM CHOICE

      - REST GREEDY

- SOFTMAX STRATEGY

  - ACTION a PROBABILITY AT TIME t IS A FUNCTION TO ITS RELATIVE VALUE

$$P(a) = \frac{e^{Q_t(a)/\tau}}{\sum_{b=1}^{n} e^{Q_t(b)/\tau}}$$

  - τ IS TEMPERATURE

    - $\tau \to 0$ CONVERGES TO GREEDY STRATEGY

    - SHOULD DECREASE WITH TIME

- OPTIMISTIC UTILITIES

  - ACTIONS NOT TRIED OFTEN GETS HIGHER WEIGHTS

$$U^+(s) = R(s) + \gamma \max_a f\left(\sum_{s'} S(s'|s,a)\, U^+(s'), N(s,a)\right)$$

↖ NUMBER OF TIMES a TAKEN IN s

- GLIE

  - MUST TRY EACH ACTION IN EACH STATE AN UNBOUNDED NUMBER OF TIMES SO THAT IT DOES NOT MISS ANY OPTIMAL ACTION

  - MUST EVENTUALLY BECOME GREEDY

  - EXAMPLE : ϵ-GREEDY, OPTIMISTIC UTILITY

Q - LEARNING

- WE WILL LEARN $Q(s,a)$ INSTEAD OF $U(s)$

$$U(s) = \max_a Q(s,a)$$

$$a^{\text{GREEDY NEXT}} = \arg\max_a Q(s,a)$$

$$U(s) = R(s) + \gamma \max_a \sum_{s'} s(s'|s,a)\, U(s')$$

$$\downarrow$$

$$Q(s,a) = R(s) + \gamma \sum_{s'} s(s'|s,a)\, U(s')$$

$$\downarrow$$

$$Q(s,a) = R(s) + \gamma \sum_{s'} s(s'|s,a) \max_{a'} Q(s',a')$$

- UPDATE STEP FROM TD

$$U^{\pi}(s) = U^{\pi}(s) + \alpha\left(R(s) + \gamma U^{\pi}(s') - U^{\pi}(s)\right)$$

$$\downarrow$$

$$Q^{\pi}(s,a) = Q^{\pi}(s,a) + \alpha\left(R(s) + \gamma \max_{a'} Q^{\pi}(s',a') - Q^{\pi}(s,a)\right)$$

- OFF-POLICY UPDATE

- SARSA

  - SLIGHTLY DIFFERENT UPDATE

$$Q^{\pi}(s,a) = Q^{\pi}(s,a) + \alpha\left(R(s) + \gamma Q(s',a') - Q^{\pi}(s,a)\right)$$

  - ON-POLICY UPDATE

- GUARANTEED CONVERGENCE

- SLOWER THAN ADP IN TERMS OF EPOCHS, BECAUSE IT DOESN'T ENFORCE CONSISTENCY AMONG VALUES THROUGH THE MODEL

- PROBABILISTIC GRAPHICAL MODELS
  - FULL JOINT MODEL
    - MARGINALIZATION

$$Pr(X) = \sum_{y \in Y} Pr(X, y)$$

    - NORMALIZATION

$$Pr(X|Y) = \frac{Pr(X, Y)}{Pr(Y)}$$

    - OR WITH NORMALIZATION CONSTANT α

$$Pr(X|Y) = \alpha \cdot Pr(X, Y), \quad \alpha \text{ IS SET SO THAT } \sum_{x \in X} Pr(x, Y) = 1$$

  - UNIVERSAL
  - FOR SUFICIENT SAMPLE SIZE ITS LEARNING CONVERGES
  - INTRACTABLE FOR REAL PROBLEMS
    - $2^n - 1$ PROBABILITIES FOR n PROPOSITIONS
    - COURSE OF DIMENSIONALITY
    - IMPEMETRABLE FOR REAL TASKS
      - MODEL GIVES NO EXPLICIT KNOWLEDGE ABOUT THE DOMAIN

- CONDITIONAL INDEPENDENCE
  - A AND B ARE CONDITIONALLY INDEPENDENT GIVEN C IF

$$Pr(A, B|C) = Pr(A|C) \cdot Pr(B|C) \quad \forall A, B, C \; Pr(C) \neq 0$$

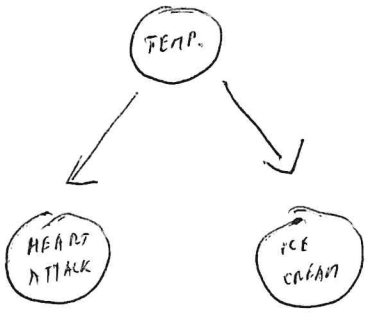
  - DENOTES

$$A \perp\!\!\!\perp B|C$$

  - SOME OBSERVATIONS BECOMES REDUNDANT

$$Pr(B|C) = Pr(B|A, C) \quad Pr(A|C) = Pr(A|B, C)$$

- DIVERGING CONNECTION



- HEART ATTACK AND ICE CREAM INDEPENDENT GIVEN TEMPERATURE

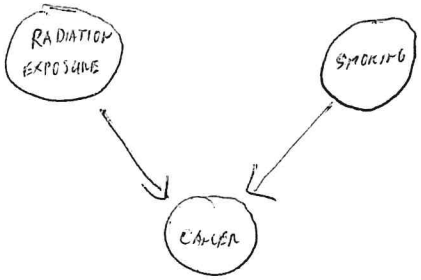
$$Pr(H|I,T) = Pr(H|T)$$

- LINEAR CONNECTION

- PHD_g AND PHD INDEPENDENT GIVEN PHD_p

$$Pr(PHD|PHD_p, PHD_g) = Pr(PHD|PHD_p)$$

- CONVERGING CONNECTION

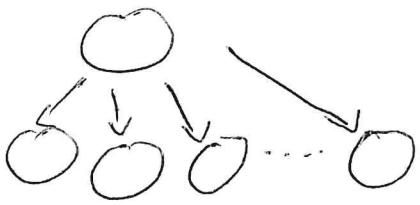


- SMOKING AND RADIATION EXPOSURE BECAME DEPENDENT GIVEN CANCER
- $Pr(\text{RADIATION EXPOSURE}|\text{CANCER, SMOKING}) \neq Pr(\text{RADIATION EXPOSURE}|\text{CANCER})$

- D-SEPARATION
  - EQUIVALENT TO CONDITIONAL INDEPENDENCE BETWEEN X AND Y GIVEN Z

- BAYESIAN NETWORK
  - DIRECTED ACYCLIC GRAPH
  - NODES REPRESENTS RANDOM VARIABLES
  - EDGES REPRESENTS DIRECT DEPENDENCE
  - NODES ANNOTATED BY PROBABILITIES
    - NODE CONDITIONED BY CONJUNCTION OF ALL ITS PARENTS
    - ROOT NODES ANNOTATED BY PRIOR
  - REDUCE COMPLEXITY BY FOCUSING ONLY ON FUNDAMENTAL RELATIONSHIPS
  - NAIVE
    - NO EDGES
    - COMPLETELY INDEPENDENT
    - $Pr(O_1, O_2, \ldots O_n) = Pr(O_1) \cdot Pr(O_2) \cdot \ldots Pr(O_n)$

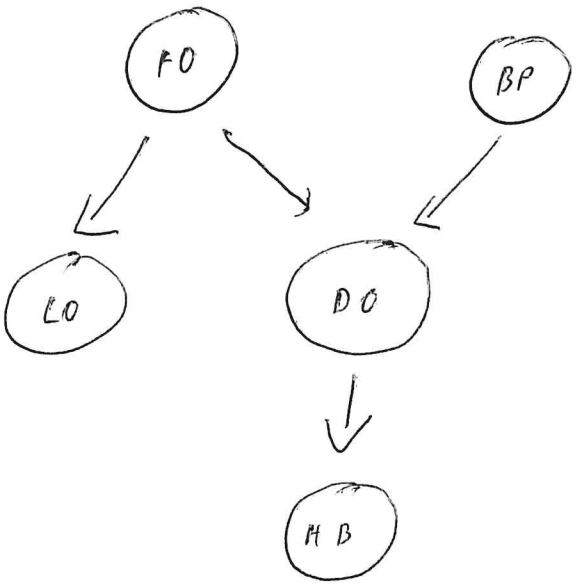
      OR
    - CI OF VARIABLES GIVEN DIAGNOSIS



  - FULLY CONNECTED GRAPH
    - DEPENDENCE OF ALL VARIABLES
    - COMPLEXITY AS JOINT MODEL

- REASONABLE MODEL LIES IN BETWEEN

- FUNDAMENTAL TASK
  - INFERENCE
    - FROM OBSERVED EVENTS ASSUME PROBABILITY OF OTHER EVENTS
  - LEARNING MODEL PARAMETERS FROM DATA
  - LEARNING NETWORK STRUCTURE
    - WHICH EDGES TO USE



- $Pr(FO, BP, DO, LO, HB) = Pr(FO) \cdot Pr(BP) \cdot Pr(LO|FO) \cdot Pr(DO|FO, BP) \cdot Pr(HD|DO)$

- INFERENCE IS NP-HARD
- COMPLEXITY GROWS WITH NUMBER OF EDGES
- APPROXIMATE INFERENCE BY STOCHASTIC SAMPLING
  - MONTE CARLO
    - SAMPLES FROM JOINT PROBABILITY DISTRIBUTION
    - ESTIMATE TARGET CONDITIONAL PROBABILITY FROM SAMPLE SET

- FORWARD SAMPLING
  - TOPOLOGICALLY SORT THE NETWORK NODES
    - PARENT COMES BEFORE CHILDREN
  - INSTANTIATE VARIABLES ALONG TOPOLOGICAL ORDERING
    - TAKE $P_r(O_j \mid \text{PARENTS}(O_j))$ AND SAMPLE O_j
  - $P_r(q \mid e) \approx \frac{N(q, e)}{N(e)}$
  - SAMPLES CONTRADICTING EVIDENCE ARE NOT USED
- REJECTION SAMPLING
  - REJECT PARTIALY GENERATED SAMPLES AS SOON AS THEY VIOLATES EVIDENCE e
- LIKELIHOOD WEIGHTING
  - FIX VALUES OF e, WE SAMPLE THE REST
  - BUT SAMPLES MUST BE WEIGHTED
  - WEIGHT EQUALS TO LIKELIHOOD OF EVENT GIVEN THE EVIDENCE
  - INITIALIZE SAMPLE WEIGHT
    $w^m = 1$
  - INSTANTIATE VARIABLES
  - IF $O_j \in E$ THEN TAKE o_j^m FROM e AND $w^m \leftarrow w^m \cdot P_r(o_j \mid \text{PARENTS}(o_j$
  - $P_r(o_i \mid e) \approx \frac{\sum_{m=1}^{n} w^m \delta(o_i^m, o_i)}{\sum_{m=1}^{n} w^m}$ $\quad \delta(i, j) = \begin{cases} 1 & \text{FOR } i = j \\ 0 & \text{FOR } i \neq j \end{cases}$
- GIBBS SAMPLING

- LEARNING BAYESIAN NETWORK FROM DATA

  - FREQUENCY TABLE

    - GIVES NUMBER OF SAMPLES WITH PARTICULAR CONFIGURATION

    - 2^N ENTRIES

  - WE KNOW THE STRUCTURE, WE SEARCH FOR CONDITIONAL PROBABILITY TABLES IN EACH NODE

  - MAXIMUM LIKELIHOOD ESTIMATE

$$L(\theta : D) = \prod_{m=1}^{n} Pr(d_m : \theta) = \prod_{m=1}^{n} Pr(O_{1m}\, O_{2m} \dots O_{rm} : \theta) =$$

$$= \prod_{i=1}^{\tilde{n}} \prod_{m=1}^{n} Pr(O_i \mid PARENTS(O_i) : \theta_i) = \prod_{i=1}^{\tilde{n}} L_i(\theta_i : D)$$

  - CONTRIBUTION OF EACH NETWORK NODE $L_i(\theta_i : D)$ IS DETERMINED (MAXIMALIZED) INDEPENDENTLY

$$\hat{\theta}_i = \arg\max_{\theta} L_i(\theta_i : D)$$

  - GENERALIZED FORMULA FOR ML IS

$$\hat{\theta}_{O_i \mid PARENTS(O_i)} = \frac{N(O_i, PARENTS(O_i))}{N(PARENTS(\theta_i))} \approx Pr(O_i \mid PARENTS(P_i))$$

  - PROBLEM OF INCOMPLETE DATA

    - EM ALGORITHM

      - INITIALIZE NETWORK

      - E-STEP

        - TAKE EXISTING NETWORK AND CALCULATE MISSING VALUES (INFERENCE)

      - M-STEP

        - MODIFY THE NETWORK ACCORDING TO THE CURRENT COMPLETE OBSERVATION

      - REPEAT UNTIL CHANGE

- STRUCTURE LEARNING
  - CAN'T SEARCH WHOLE SPACE OF POSSIBILITIES
  - NAIVE SEARCH CAN'T BE USED
  - K2, MCMC ALGORITHMS
  - SEVERAL SCORES HOW TO EVALUATE HOW WELL A STRUCTURE MATCHES THE DATA
    - LOG LIKELIHOOT DOESN'T WORK, IT OVERFITS
    - BAYESIAN SCORE
    - LOCAL CI TESTS