Probabilistic Artificial Intelligence

Lectures

1 - Introduction and Probability

- Modeling: agents, environments, percepts, actions, states, performance evaluation (rationality)
- Environment types: fully vs partially observable, discrete vs continuous, known vs unknown, single vs multi-agent, deterministic vs stochastic vs nondeterministic
- Probability space, probability axioms (normalization/unitarity, non-negativity, sigma-additivity), independent events (pairwise vs mutual independence)
- **MIT:** In most experiments, the prior probabilities on hypotheses are not known. In this case, our recourse is the art of statistical inference: we either make up a prior (Bayesian) or do our best using only the likelihood (frequentist).
- Random variables, probability distributions (Bernoulli and Binomial, categorical and multinomial), continuous distributions (e.g. Gaussian), joint distributions (e.g. multivariate Gaussian)
    - Propositional symbols = binary RVs
- Conditional probability, conditional distributions, joint distribution rules: sum rule (marginalization) and product rule (chain rule)

2 - Probability and Bayesian Networks

- Bayes' rule: posterior inference, prior probability, likelihood
    - for n binary RVs we need 2^n - 1 parameters, marginalization sum has 2^n - 1 terms
    - independent RVs need n << 2^n - 1 parameters, computing marginals is O(1), independence too strong of an assumption
- Conditional independence properties: symmetry, decomposition, contraction, weak union, intersection
- Naïve Bayes models: multiple effects with a single cause, effects are conditionally independent given cause
- Bayesian networks: compact representation, causal parametrization (immediate causes/effects), directed acyclic graph
    - joint distribution defined by a BN structure and a set of conditional probability tables
    - every probability distribution can be described by a BN
    - ordering matters a lot for compactness of representation
- **Wikipedia**: A directed acyclic graph (DAG), is a finite directed graph with no directed cycles
- The Markov blanket for a node A in a Bayesian network is the set of nodes ∂A composed of A's parents, its children, and its children's other parents. Every set of nodes in the network is conditionally independent of A on the Markov blanket of the node A

3 - Bayesian Networks and d-separation

- BNs with 3 nodes (indirect causal effect, indirect evidential effect, common cause, common effect), v-structures, explaining-away

- Information flow, active trails, d-separation
    - d-separation implies conditional independence, converse does not hold in general
    - algorithm: mark observations and ancestors, do BFS and stop if path is blocked, runs in linear time
- Maximization queries: MPE (most probable explanation), MAP (maximum a posteriori)
- Inference in general BNs
    - Exact solution: #P-complete, any nontrivial approximation is NP-hard
    - MPE: NP-complete, MAP-NP^PP complete

4 - Bayesian Networks: Exact Inference

- Variable elimination: create factors by summing out variables
- Multiplying and marginalizing factors
- Variable elimination for polytrees:
    - A DAG is a polytree iff dropping edge directions results in a tree
    - Pick a root, orient edges towards root, eliminate variables according to topological ordering
- Variable elimination is correct even if there are loops, however the factors may blow up, finding a good ordering is hard
- Factor graphs, message passing
    - A factor graph for a BN is a bipartite graph (variables and factors), each factor is associated with a subset of variables, all CPDs of the BN have to be assigned to one of the factor nodes
    - Conditioning on observations: multiply with indicator function, then renormalize marginals
- Sum-product message passing (Belief Propagation Algorithm)
    - Initialize all messages as uniform distribution
    - Pick some ordering and update messages until convergence

5 - Bayesian Networks: Approximate Inference

- Belief Propagation is exact for polytree BNs, factor graph of a polytree is a tree
- Loopy belief propagation: does not generally converge (can oscillate), even if it does it may converge to incorrect marginals, still practically useful
    - Loopy BP multiplies same factors multiple times, often overconfident
- Variable elimination for MPE: max-product message passing
    - For tree factor graphs, max-product computes max-marginals
    - Can retrieve MAP solution from these max-marginals (must be careful when ties need to be broken)
- Deterministic inference techniques vs stochastic approximations
    - Deterministic: variable elimination, (loopy) belief propagation, variational inference for computing marginals, combinatorial optimization (e.g. graph cuts for MPE)
    - Stochastic: Algorithms that randomize to compute marginals as expectations, in contrast to deterministic approaches guaranteed to converge to right answer (if waited long enough), more exact, slower than deterministic variants, also work for continuous distributions
- Marginals as expectations: express marginal distribution as expectation of indicator function and approximate expectation by sampling

- Sample approximations of expectations
    - (Strong) Law of large numbers, almost sure convergence (with probability 1), suggests approximation using finite samples
    - **Wikipedia:** Strong law of large numbers: almost sure convergence (with probability 1), weak law of large numbers: converges in probability (for any nonzero margin)
- Hoeffding's inequality: provides an upper bound on the probability that the sum of bounded independent random variables deviates from its expected value by more than a certain amount
    - Probability of error decreases exponentially in N
- Sampling from a Bernoulli distribution, sampling from a multinomial, forward sampling from a BN, Monte Carlo sampling from a BN, computing probabilities through sampling, rejection sampling
    - **Oregon state:** Rejection sampling: do forward sampling but throw out samples where E does not equal e, rare evidence is the norm, as the number of evidence variables |E| grows the probability of evidence decreases exponentially
    - **Wikipedia**: Monte Carlo methods (or Monte Carlo experiments) are a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results. Their essential idea is using randomness to solve problems that might be deterministic in principle.

6 - Bayesian Networks: Approximate Inference (Sampling)

- Sample complexity for probability estimates, absolute and relative errors
- Sampling from rare events: want to directly sample from posterior distribution, Markov Chain Monte Carlo (MCMC) methods (important example: Gibbs sampling)
- Sampling from intractable distribution:
    - Given unnormalized distribution, normalizer Z is intractable
    - **Stack exchange**: Problems are said to be tractable if they can be solved in terms of a closed-form expression
    - Idea: create a Markov chain that is efficient to simulate and has stationary distribution P(X)
- Markov chains, ergodicity
    - **Wikipedia**: A Markov chain is a stochastic model describing a sequence of possible events in which the probability of each event depends only on the state attained in the previous event
    - An ergodic Markov Chain has a unique and positive stationary distribution π and this stationary distribution is independent of prior $P(X_1)$
    - If a Markov chain is simulated sufficiently long, sample X_N is drawn from a distribution very close to stationary distribution π
    - Need to specify transition probabilities to ensure correct stationary distribution
- Detailed balance equation
- Designing Markov Chains, proposal distribution, acceptance distribution, Metropolis-Hastings Sampler
    - **Wikipedia**: A standard empirical method to assess convergence is to run several independent simulated Markov chains and check that the ratio of inter-chain to intra-chain variances for all the parameters sampled is close to 1
    - **Wikipedia**: In probability theory, the mixing time of a Markov chain is the time until the Markov chain is "close" to its steady state distribution
- Gibbs Sampling: sampling from X_i given an assignment to all other variables is efficient

    - Random order (uniform) satisfies detailed balance equation (we accept the proposal at every step)
    - Practical variant does not satisfy detailed balance, but also has correct stationary distribution
    - Re-sampling X_i only requires multiplying factors containing it and renormalizing
    - Works for any joint distribution that is specified as a (possibly unnormalized) product of factors
    - **Wikipedia:** Generally, samples from the beginning of the chain (the *burn-in period*) may not accurately represent the desired distribution and are usually discarded. It has been shown, however, that using a longer chain instead (e.g. a chain that is *n* times as long as the initially considered chain using a thinning factor of *n*) leads to better estimates of the true posterior.
    - Joint sample at time *t* depends on sample at time *t-1*, thus the law of large numbers (and sample complexity bounds such as Hoeffding's inequality) do not apply (not i.i.d.)
    - Ergodic theorem (special case) applies, limit almost surely holds
    - To let the Markov chain *'burn in'*, ignore the first t_0 samples, and approximate
    - **Wikipedia**: Gibbs sampling is popular partly because it does not require any 'tuning'
    - **Wikipedia**: In multivariate distributions, the classic Metropolis–Hastings algorithm as described above involves choosing a new multi-dimensional sample point. When the number of dimensions is high, finding the right jumping distribution to use can be difficult, as the different individual dimensions behave in very different ways, and the jumping width (see above) must be "just right" for all dimensions at once to avoid excessively slow mixing. An alternative approach that often works better in such situations, known as Gibbs sampling, involves choosing a new sample for each dimension separately from the others, rather than choosing a sample for all dimensions at once. This is especially applicable when the multivariate distribution is composed out of a set of individual random variables in which each variable is conditioned on only a small number of other variables, as is the case in most typical hierarchical models. The individual variables are then sampled one at a time, with each variable conditioned on the most recent values of all the others.

7 - Sequential Models

- Temporal models, basic idea: create copies of variables, one per time step, typically assume discrete, unit-length time steps
- Markov chains, Markov assumption, stationary assumption
    - **Wikipedia**: Roughly speaking, a process satisfies the Markov property if one can make predictions for the future of the process based solely on its present state just as well as one could knowing the process's full history, hence independently from such history; i.e., conditional on the present state of the system, its future and past states are independent
    - Can always reduce k-order MCs to 1st order
    - **Book**: The more uncertainty there is in the transition model, the shorter will be the mixing time and the more the future is obscured.
- Hidden Markov Models (HMMs), Kalman Filters, transition and emission probabilities
    - $X_1,...,X_T$: unobserved (hidden or latent) variables (states) — $Y_1,...,Y_T$: observations
    - HMMs X_i categorical, Y_i categorical (or arbitrary) — Kalman Filters: X_i, Y_i Gaussian distributions
- Inference tasks
    - Filtering: forward algorithm

    - Prediction
    - Smoothing: forward-backward (sum-product) algorithm
    - Most probable explanation: Viterbi (max-product) algorithm
- Bayesian filtering: conditioning, prediction
- Kalman filters (Gaussian HMMs), multivariate Gaussian distributions, Bayesian inference in Gaussian distributions, conditional distributions
    - Multiples of Gaussians are Gaussian
    - Sums of Gaussians are Gaussian
- Kalman filters: transition model, sensor model, Bayesian Filtering in KFs, Kalman update, Kalman gain

8 - Sequential Models and MDPs

- **Book**: Now, if every HMM is a DBN and every DBN can be translated into an HMM, what's the difference? The difference is that, by decomposing the state of a complex system into its constituent variables, the can take advantage of sparseness in the temporal probability model
- MDPs (states, actions, transition probabilities, reward function),
    - Stationarity, additive rewards vs discounted rewards
    - Computing the value of a policy, solving for the value of a policy
    - Bellman equation, Bellman Theorem: a policy is optimal iff it is greedy w.r.t. its induced value function
    - **Book**: We have seen that any fixed action sequence won't solve the problem, because the agent might end up in a state other than the goal. Therefore, a solution must specify what the agent should do for any state that the agent might reach. A solution of this kind is called a policy.
    - **Book**: An optimal policy is a policy that yields the highest expected utility

9 - Probabilistic Planning

- Value iteration: The basic idea is to calculate the utility of each state and then use the state utilities to select an optimal action in each state
    - Solve Bellman equation using dynamic programming
    - In practice, it often occurs that π_i becomes optimal long before U_i has converged
    - Convergence of value iteration: Bellman update is a contraction (existence of a unique fixed point & convergence to the fixed point)
    - Finds ε-optimal solution in polynomial # iterations
    - It will take at least the diameter of the MDP until we start to see a signal.
- Policy iteration: The algorithm alternates the policy evaluation and policy improvement, beginning from some initial policy π_0
    - Start with random policy π, compute exact value function V^{π} (matrix inversion), select greedy policy w.r.t. V^{π} and iterate
    - Finds exact solution in polynomial # iterations
    - Every iteration requires computing a value function
- Can combine the ideas of both algorithms.
- MDP is a controlled Markov chain. It becomes a Markov chain when the policy is fixed.
- POMDPs
    - Noisy observations of the hidden states. Very powerful but typically extremely intractable.

10 - Learning Bayesian Networks

- Learning, learning BN from data
- Parameter learning
  - Estimating CPDs (exactly like rejection sampling)
  - MLE for general Bayes nets, algorithm for Bayes net MLE
  - Regularizing Bayesian Networks, pseudo-counts, beta prior over parameters
  - Bayesian parameter estimation, learning parameters for dynamic models
- Structure learning
  - Score based structure learning
  - MLE score, mutual information, empirical mutual information
    - Mutual information is symmetric and monotonic.
    - Due to monotonicity, trivial optimal solution is a fully connected Bayes net.
  - Likelihood score, maximizing the MLE score
  - Finding the optimal MLE structure, regularizing a BN
    - Number of parameters |G| is the total number of entries in the CPTs.
    - BIC is consistent, but it is NP-hard on arbitrary graphs.
  - Finding the optimal tree, Chow-Liu algorithm
    - Maximum spanning tree problem, can be solved optimally in time O(|E|log|E|)

11 - Introduction to Reinforcement Learning

- MDP setting solves the credit assignment problem by the Markovity assumptions.

12 - Deep Reinforcement Learning

- Monte Carlo Tree Search (MCTS)
  - Selection -> Expansion -> Simulation -> Backpropagation
  - Reward of a state is (# of wins / # of visits to that state)
  - Tree policy: e.g. UCT, epsilon-greedy
  - Default policy: e.g. random (fast to evaluate)
  - Asymmetric growth of the search tree: Evaluates most promising parts first
  - Optimality: Converges to minimax solution under certain conditions

13 - RL: Policy Search and Bayesian Optimization

Piazza

- Soundness and faithfulness
  - Soundness means correctness, that you cannot deduce conditional independence relations (conditioned on anything or even nothing) by using the BN that you don't have in your model, every thing you deduce using your Bayes net is correct
  - Faithfulness means minimality, that every independence you have in your model can be deduced with the BN, factorization-correctness tradeoff
  - Fully connected BN: no independence, certainly sound, no factorization
  - Fully disconnected BN: everything is independent, always faithful, might be inconsistent with the model
  - Choosing the minimal subsets for every variable in the ordering ensures soundness of the BN

- Soundness (Markovness) states d-separation implies conditional independence whereas faithfulness states the opposite, only if both properties hold we can state the equivalence

Variable Elimination has exponential complexity.

Chow-Liu: Any measure of association $A(X_i, X_j)$, such that $min(A(X_i, X_j), A(X_j, X_k)) > A(X_i, X_k)$ can be used.