Basic Probability Rules

$P(A,B) = P(A|B)\cdot P(B) = P(B|A)\cdot P(A)$

$P(X_1,\dots,X_n) = P(X_1)\cdot P(X_2|X_1)P(X_3|X_1,X_2)\cdot\dots P(X_n|X_{1:n-1})$

$P(X_{1:i-1}, X_{i+1:n}) = \sum_{x_i} P(X_{1:i-1}, X_i = x_i, X_{i+1:n})$

$P(X_i) = \sum_{x_1,\dots,x_{i-1},x_{i+1},\dots,x_n} P(x_1,\dots,x_{i-1},X_i,x_{i+1},\dots,x_n)$

$$P(X|Y) = \frac{P(X)\cdot P(Y|X)}{P(Y)} = \frac{P(X)\cdot P(Y|X)}{\sum_x P(X=x)P(Y|X=x)}$$

$P(X,Y|Z) = P(X|Y,Z)P(Y|Z)$

$P(X=x) = \sum_y P(X=x,Y=y) = \sum_y P(X=x|Y=y)P(Y=y)$

$X \perp Y|Z$ iff $P(X,Y|Z) = P(X|Z)\cdot P(Y|Z)$
or if $P(Y|Z)>0$ then $P(X|Z,Y) = P(X|Z)$

$X\perp Y|Z \Rightarrow Y\perp X|Z \quad X\perp(Y,W)|Z \Rightarrow X\perp Y|Z$
$(X\perp Y|Z)\wedge(X\perp W|Y,Z) \Rightarrow X\perp Y,W|Z$
$X\perp Y,W|Z \Rightarrow X\perp Y|Z,W$
$(X\perp Y|W,Z)\wedge(X\perp W|Y,Z) \Rightarrow X\perp Y,W|Z$ if $P(\cdot)>0$

Bayesian Network

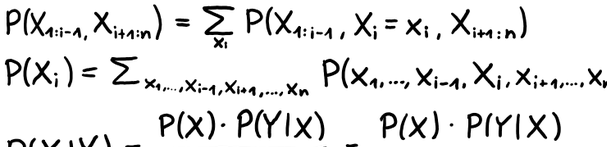
Consists of a structure: directed acyclic graph where each vertex is interpreted as a random variable.

Consists of a set of conditional probability distributions (CPTs) where all nodes S have $P(X_S|Pa_{X_S})$ ← parents of X_S

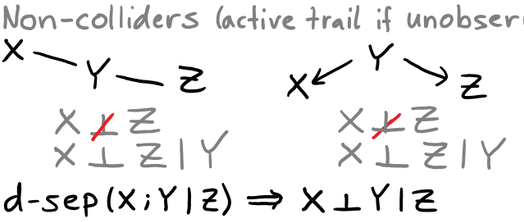
$P(X_1,\dots,X_n) = \prod P(X_i|Pa_{X_i})$

d-separation Rules

Colliders (only active trail if observed):



Non-colliders (active trail if unobserved):



d-sep(X;Y|Z) ⇒ $X\perp Y|Z$

Inference

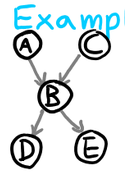
tree-structured BN: variable elimination or belief propagation (gives exact marginals)
loopy networks: loopy belief propagation (fast, may not converge), variational inference (fast, converges, may be incorrect), Gibbs sampling (slow, converges to correct marginals)

Variable Elimination Order

For a polytree Bayesian Network (a directed acyclic graph which is a tree when dropping edge directions)
1. Drop edge directions
2. Choose root (e.g. for P(A), pick A)
3. Orient all edges towards root
4. Eliminate from leaves

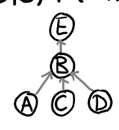
Always remains bounded: $O(n\cdot 2^{\max_i Pa_i})$

Example Variable Elimination



$P(A,B,C,D,E) = P(A)\cdot P(C)\cdot P(B|A,C)\cdot P(D|B)\cdot P(E|B)$

Want to get $P(E) \Rightarrow$



$$P(E) = \sum_b\sum_a\sum_c\sum_d P(a,b,c,d,E)$$
$$= \sum_b P(E|b)\sum_a P(a)\sum_c P(c)P(b|a,c)\underbrace{\sum_d P(d|b)}_{=1}$$
$$= \sum_b P(E|b)\sum_a P(a)\underbrace{\sum_c P(c)P(b|a,c)}_{:=g_1(a,b)}$$
$$= \sum_b P(E|b)\underbrace{\sum_a P(a)\,g_1(a,b)}_{:=g_2(b)}$$
$$= \sum_b P(E|b)\,g_2(b) = g_3(E)$$

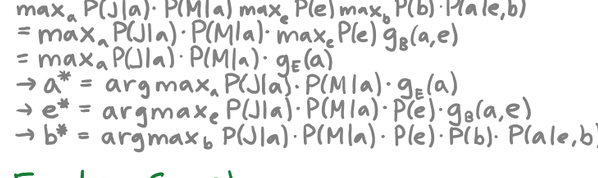
Example Variable Elimination for MPE

Idea: Given some evidence J, M we want to find the most probable explanation for the other variables:

$\text{argmax}_{e,b,a} P(e,b,a|J,M) = \text{argmax}_{e,b,a} \frac{1}{Z} P(e,b,a,J,M)$
$= \text{argmax}_{e,b,a} P(e)\cdot P(b)\cdot P(a|b,e)\cdot P(J|a)\cdot P(M|a)$
$\max_a P(J|a)\cdot P(M|a)\max_e P(e)\max_b P(b)\cdot P(a|e,b)$
$= \max_a P(J|a)\cdot P(M|a)\cdot \max_e P(e)\, g_B(a,e)$
$= \max_a P(J|a)\cdot P(M|a)\cdot g_E(a)$
$\rightarrow a^* = \text{argmax}_a P(J|a)\cdot P(M|a)\cdot g_E(a)$
$\rightarrow e^* = \text{argmax}_e P(J|a)\cdot P(M|a)\cdot P(e)\cdot g_B(a,e)$
$\rightarrow b^* = \text{argmax}_b P(J|a)\cdot P(M|a)\cdot P(e)\cdot P(b)\cdot P(a|e,b)$

Factor Graph

A bipartite graph consisting of variables (the random variables of the Bayesian Network) and factors (each factor is associated with a subset of variables s.t. all conditional probab. dist. of the BN are assigned to one of the factor nodes.



Sum-Product Message Passing

Initialize all messages as uniform distribution (=1)
Until converged do
  Pick ordering on the factor graph edges
  Update messages according to the ordering
  Break if messages change at most ε

Message from node v to factor u:
$\mu_{v\to u}(x_v) = \prod_{u'\in N(v)\setminus\{u\}} \mu_{u'\to v}(x_v)$

Message from factor u to node v:
$\mu_{u\to v}(x_v) = \sum_{\text{(marginalize } x_v)} f_u(x_u) \prod_{v'\in N(u)\setminus\{v\}} \mu_{v'\to u}(x_{v'})$ (messages that agree with x_u)

$P(X_v = x_v) \propto \prod_{u\in N(v)} \mu_{u\to v}(x_v)$

$P(X_u = x_u) \propto f_u(x_u) \prod_{v\in N(u)} \mu_{v\to u}(x_v)$

Belief Propagation of Polytree BN

- Factor graph is a tree as well
- Choose one node as root, send messages from leaves to root, and from root to leaves
- Converges to correct marginals after two rounds (one such pass)

Max-Product Message Passing

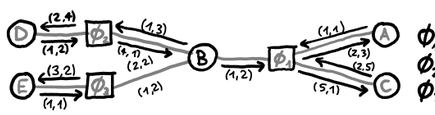
Only difference: $\mu_{u\to v}(x_v) = \max_{x_u\sim x_v} f_u(x_u)\prod_{v\in N(u)\setminus\{v\}}\mu_{v'\to u}(x_{v'})$

$P_{max}(X_v = x_v) = \max_{x\sim x_v} P(x)$

For tree BN: $P_{max}(X_v = x_v) \propto \prod_{u\in N(v)} \mu_{u\to v}(x_v)$

Example Belief Propagation

$P(A,B,C,D,E) \propto \phi_1(A,B,C)\,\phi_2(B,D)\,\phi_3(B,E)$



$\phi_1(a,b,c) = \begin{cases} 0 & \text{if } c=0 \\ a+b+c & \text{otw.} \end{cases}$
$\phi_2(b,d) = b+d$
$\phi_3(b,e) = b+e$

From these messages, compute the approx. marginals of B.
From belief propagation we have
$P(B) \propto \prod_{\phi_i\in N(B)} \mu^{(t)}_{\phi_i\to B}(B) = \mu^{(t)}_{\phi_1\to B}(B)\,\mu^{(t)}_{\phi_2\to B}(B)\,\mu^{(t)}_{\phi_3\to B}(B)$

↳ for B=0: $\hat P(B=0) = 3\cdot 4\cdot 1 = 12$
↳ for B=1: $\hat P(B=1) = 2\cdot 1\cdot 2 = 4$
$\Big\} P(B) = \frac{(12,4)}{Z} = (\frac{3}{4},\frac{1}{4})$

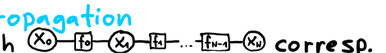
Compute the message $\mu^{(t+1)}_{B\to\phi_2}$
$\mu^{(t+1)}_{B\to\phi_2}(B) = \prod_{\phi_i\in N(B)\setminus\{\phi_2\}} \mu^{(t)}_{\phi_i\to B}(B) = \mu^{(t)}_{\phi_1\to B}(B)\cdot\mu^{(t)}_{\phi_3\to B}(B)$

↳ for B=0: $\mu^{(t+1)}_{B\to\phi_2}(0) = 3\cdot 1 = 3$
↳ for B=1: $\mu^{(t+1)}_{B\to\phi_2}(1) = 2\cdot 2 = 4$
$\Big\} \mu^{(t+1)}_{B\to\phi_2}(B) = (3,4)$

Compute the message $\mu^{(t+1)}_{\phi_1\to B}$
$\mu^{(t+1)}_{\phi_1\to B}(B) = \sum_{v^*\in V_{\phi_1}\setminus\{B\}} \phi_1(v_{\phi_1}) \prod_{v^*\in N(\phi_1)\setminus\{B\}} \mu^{(t)}_{v^*\to\phi_1}(v^*)$
$= \sum_a\sum_c \phi_1(a,B,c)\,\mu^{(t)}_{A\to\phi_1}(a)\,\mu^{(t)}_{C\to\phi_1}(c)$

for B=0: $\mu^{(t+1)}_{\phi_1\to B}(0) = 1\cdot 1\cdot 5 + 2\cdot 1\cdot 5 = 15$
for B=1: $\mu^{(t+1)}_{\phi_1\to B}(1) = 2\cdot 1\cdot 5 + 3\cdot 1\cdot 5 = 25$
$\Big\} \mu^{(t+1)}_{\phi_1\to B}(B) = (15,25)$

Example Belief Propagation

Given a factor graph  corresp. to a chain BN. For $0\le i\le N$ we assume $X_i\in\{0,\dots,M\}$. Belief propagation computes $P(X_i=x)\ \forall i\in\{0,\dots,N\}\ \forall x\in\{0,\dots,M\}$.
Suppose we want to compute $P(X_i = x | X_0\le X_4\le\dots X_N)$. How can this be done? $P(X_0\le\dots\le X_N) = \prod_{i=0}^{N-1} P(X_i\le X_{i+1})$. We propose the factor $\hat f_i(x_i,x_{i+1}) = \text{if } x_i\le x_{i+1} \text{ then } f_i(x_i,x_{i+1}) \text{ else } 0$. From belief propagation we know $\hat P(\hat X_i = x_i) = \frac{1}{Z}\sum_{x_{1:N}} \prod_k \hat f_k(x_k,x_{k+1})$.
Further $\hat Z = \sum_{x_{1:N}}\prod_k \hat f_k(x_k,x_{k+1})$. We evaluate $P(X_0\le\dots\le X_N)$ in the original graph: $P(X_0\le\dots\le X_N) = \frac{1}{Z}\sum_{x_0\le\dots\le x_N}\prod_k f_k(x_k,x_{k+1}) = Z/Z$
$P(X_i=x, X_0\le\dots\le X_N) = \frac{1}{Z}\cdot\sum_{x_0\le\dots\le x_i\le\dots\le x_N}\prod_k f_k(x_k,x_{k+1}) = \frac{1}{Z}\sum_{x_{1:N}}\prod_k \hat f_k(x_k,x_{k+1})$
It follows that $P(X_i=x|X_0\le\dots\le X_N) = P(X_i=x, X_0\le\dots\le X_N)/P(X_0\le\dots\le X_N)$
$= Z/Z\cdot\frac{1}{Z}\sum_{x_{1:N}}\prod_k \hat f_k(x_k,x_{k+1}) = \frac{1}{Z}\sum_{x_{1:N}}\prod_k f_k(x_k,x_{k+1}) = \hat P(X_i = x_i)$

Monte Carlo Sampling (Forward Sampling)

1. Sort variables in topological order
2. In this order, sample $x_i \sim P(X_i | X_1 = x_1,\dots,X_{i-1} = x_{i-1})$
3. Marginals: $P(X=x) \approx \frac{1}{N}\sum_{i=1}^N [x^{(i)}=1][x^{(i)}=x] = \frac{1}{N}\text{count}(x=+)$
   Conditionals: $P(X=x|Y=y) \approx \frac{\text{count}(X=x,Y=y)}{\text{count}(Y=y)}$ (rejection sampling. If y rare → large waste of samples

How many samples? Error decreases exponentially in N

Example Rejection Sampling

d-dimensional target $p(x) = N(x;\mu,\sigma_p^{2/d})$ and the proposal $q(x) = N(x;\mu,\sigma_q^{2/d})$. Optimal acceptance rate can be accomplished with $k = \sigma_q/\sigma_p$. With $d=1000$ and $\sigma_q = 1.01\,\sigma_p \rightarrow k = 1/20000$ resulting in a huge waste in samples

Gibbs Sampling: Random Order

Start with initial assignment to all variables
Fix observed variables X_B to their observed value x_B
For $t = 1$ to ∞ do
  1. Pick a variable i at random (non-observed)
  2. Set v_i = values of all x except x_i
  3. Update x_i by sampling from $P(X_i | v_i)$

Satisfies detailed balance equation

Gibbs Sampling: Practical Variant

Start with initial assignment $x^{(0)}$ for all var.
Fix the fixed values to be conditioned on
For $t = 1$ to ∞ do
  For each variable X_i (that is not fixed)
    Sample value from $\sim P(X_i | v_i)$ ← other vars from the last iteration

No detailed balance but also has correct stationary distribution.

Computing Expectations via Gibbs Sampling

With GS get samples $X^{(1)},\dots,X^{(T)}$. Ignore the first t_0 samples (burn-in) and approx.: $\mathbb{E}[f(x)|x_B] \approx \frac{1}{T-t_0}\sum_{\tau=t_0+1}^{T} f(X^{(\tau)})$.

Example Gibbs Sampling

Assume you are running Gibbs sampler on the same factor graph and the last sample we drew is (A=0,B=0,C=1,D=1,E=1). Compute the distribution from which we should draw a new sample of A.

$P(A^{(t+1)}|B^{(t)},C^{(t)},D^{(t)},E^{(t)}) = \frac{P(A^{(t+1)},B^{(t)},C^{(t)},D^{(t)},E^{(t)})}{P(B^{(t)},C^{(t)},D^{(t)},E^{(t)})}$ ← constant

$P(A^{(t+1)}|B^{(t)},C^{(t)},D^{(t)},E^{(t)}) \propto P(A^{(t+1)},B^{(t)},C^{(t)},D^{(t)},E^{(t)})$
$= \phi_1(A^{(t+1)},B^{(t)},C^{(t)})\underbrace{\phi_2(B^{(t)},D^{(t)})\phi_3(B^{(t)},E^{(t)})}_{\text{constant for the value of A that we need}}$
$\propto \phi_1(A^{(t+1)},B^{(t)},C^{(t)})$

for A=0: $\phi_1(0,0,1) = 1$
for A=1: $\phi_1(1,0,1) = 2$
$\Big\}$ after normalizing: $P(A) = (\frac{1}{3},\frac{2}{3})$

Example Gibbs Sampling

P(C) = 0.5

C	P(S)
0	0.1
1	0.5

C	P(R)
0	0.8
1	0.2

S	R	P(W)
0	0	0.99
0	1	0.9
1	0	0.9
1	1	0.0

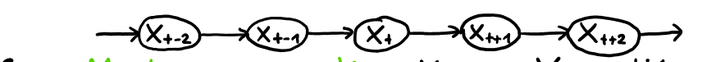
Query: P(R|S=1,W=1)

How many states does the Markov chain have? 4, because there are 2 unconstraint vars.
Calculate the transition matrix $T(x,y)$ containing $P(X_{t+1}=y|X_t=x)$ for all x, y.
Compute sampling distribution for each variable:

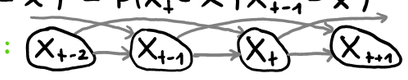
$P(C|r,s) = \frac{1}{Z}P(r,s|C)\cdot P(C) = \frac{1}{Z}P(r|C)P(s|C)P(C)$
$= \frac{1}{Z}\langle 0.8, 0.2\rangle\langle 0.1, 0.5\rangle\langle 0.5, 0.5\rangle$
$= \frac{1}{Z}\langle 0.04, 0.05\rangle = \langle 4/9, 5/9\rangle$

(same for $P(C|\neg r,s)$, $P(R|c,s,w)$, $P(R|\neg c,s,w)$)
There are 3 cases in the matrix (uniform sampling)
1) self-transition: here either one has to get sampled and stay the same e.g. c,r → c,r = 0.5 · P(c|r,s) + 0.5 · P(r|c,w,s)
2) Entries where one variable changes e.g. c,r → c,¬r = 0.5 · P(¬r|c,s,w)
3) Two variables change → not possible ⇒ 0

Markov Chains (1st order)



Markov assumption: $X_{1:t-1} \perp X_{t+1:T} | X_t$
Stationary assumption: $P(X_{t+1} = x | X_t = x') = P(X_t = x | X_{t-1} = x')$

Higher order: 

Convert to first order: Redefine states: $z_1 = X_1$, $z_2 = (X_1,X_2)$, $z_3 = (X_2,X_3)$...

Ergodic: If every state can be reached in finite steps
↳ have a unique positive stationary distribution.

Ergodic property: An ergodic Markov chain has a unique, positive stationary distribution $\pi(x)>0$ s.t. $\forall x\ \lim_{N\to\infty} P(X_N=x)=\pi(x)$ (π is independent of X_1)

Ergodic theorem: $\lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N} f(x_i)=\sum_{x\in D}\pi(x)f(x)$
where D is finite state space of X_i, f a function on D

Detailed balance: For $T(x,y)=P(y|x)$ a sufficient condition for ensuring π^∞ is stationary: $\pi(x)T(x,y)=\pi(y)T(y,x)$

Example Detailed Balance

Assume that given a Markov chain with state space Ω and transition matrix T, stationary distribution π.
Show that if for some t: X_t distributed accord. to π and the chain satisfies the detailed balance equation $\pi(x)T(x,y)=\pi(y)T(y,x)$ for all $x,y\in\Omega$, then $\forall k\geq 0$ and $x_0,\dots,x_k\in\Omega$ it holds: $P(X_t=x_0,\dots,X_{t+k}=x_k)=P(X_t=x_k,\dots,X_{t+k}=x_0)$

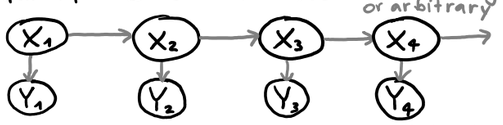
$P(X_t=x_0,\dots,X_{t+k}=x_k)=P(X_t=x_0)P(X_{t+1}=x_1\,|\,X_t=x_0)\dots$
$\dots P(X_{t+k}=x_k\,|\,X_{t+k-1}=x_{k-1})=\pi(x_0)T(x_0,x_1)\dots T(x_{k-1},x_k)$
$=T(x_1,x_0)\pi(x_1)\dots T(x_{k-1},x_k)=\dots=T(x_1,x_0)\dots T(x_k,x_{k-1})\pi(x_k)$
$=\pi(x_k)T(x_k,x_{k-1})\dots T(x_1,x_0)=P(X_t=x_k)\,P(X_{t+1}=x_{k-1}\,|\,X_t=x_k)$
$\dots P(X_{t+k}=x_0\,|\,X_{t+k-1}=x_1)=P(X_t=x_k,\dots X_{t+k}=x_0)$

Prediction in Markov Chains:

$P(X_i|X_1=x)=\sum_{x'}P(X_i,X_{i-1}=x'|X_1=x)$
$=\sum_{x'}P(X_i|X_{i-1}=x',X_1=x)\cdot P(X_{i-1}=x'|X_1=x)$
$=\sum_{x'}P(X_i|X_{i-1}=x')\cdot P(X_{i-1}=x'|X_1=x)$

Hidden Markov Model

$X_1\dots X_T$ unobserved (hidden) variables (categorical)
$Y_1\dots Y_T$ observations → can be categorical or arbitrary



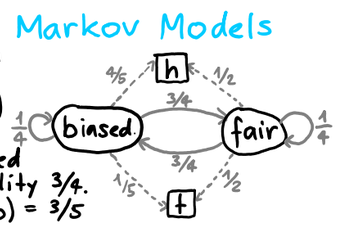
Transition probabilities: $P(X_t|X_{t-1})$

Emission probabilities: $P(Y_t|X_t)$

Tasks:
- Filtering $P(X_t|Y_{1:T})$
- Smoothing $P(X_t|Y_{1:T})$ $1\leq t<T$
- Prediction $P(X_{t+\tau}|Y_{1:t})$ $\tau\geq 1$
- Most probable explaination $\operatorname{argmax}_{X_{1:T}} P(X_{1:T}|Y_{1:T})$

Example Hidden Markov Models

In each time step a coin is flipped, resulting in heads (h) or tails (t). After each flip, the coin used changes with probability 3/4. The prior is $P(X_1=b)=3/5$ $P(X_1=f)=2/5$.



Derive $\lim_{i\to\infty} P(X_i=b)$: because i goes to infinity, prior distribution does not matter. Since probability b→f and f→b is the same, for $i\to\infty$ $P(X_i=b)=1/2$

Formal way: $V^i=\begin{bmatrix}\text{prob of fair coin}\\ \text{prob of biased coin}\end{bmatrix}=\begin{bmatrix}V_1\\ V_2\end{bmatrix}$
find transition matrix $T=\begin{bmatrix}P(f|f) & P(f|b)\\ P(b|f) & P(b|b)\end{bmatrix}$
(this is a stochastic matrix, as rows sum to 1 ⇒ has eig.value 1) $=\begin{bmatrix}1/4 & 3/4\\ 3/4 & 1/4\end{bmatrix}$

If we know the initial state V^1, we can calculate: $V^t=T\cdot V^{t-1}=T^t\cdot V^1$

We look for $V^\infty=T\cdot V^\infty\sim 1\cdot V^\infty=T\cdot V^\infty$ (eigenvalue ↑, ↕ eigenvector)
⇒ find eigenvector for eigenvalue 1:

$\begin{bmatrix}1/4-1 & 3/4\\ 3/4 & 1/4-1\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix}=\begin{bmatrix}0\\ 0\end{bmatrix}\sim\begin{bmatrix}x\\ y\end{bmatrix}=\begin{bmatrix}1/2\\ 1/2\end{bmatrix}$ ← result

Kalman Filter

Special case of HMM where X_t, Y_t are Gaussian and with Linear dynamics and observation model.
Motion model: $P(X_{t+1}|X_t)$ with $X_{t+1}=FX_t+\varepsilon_t$, $\varepsilon_t\sim N(0,\Sigma_x)$
Observation model: $P(Y_t|X_t)$ with $Y_t=HX_t+v_t$, $v_t\sim N(0,\Sigma_y)$
Filtering: $P(X_t|Y_t=y_{t:t})\propto P(X_t|y_{1:t-1})\cdot P(y_t|X_t)$
Prediction: $P(X_{t+1}|y_{1:t})=\int P(X_{t+1}|x_t)P(x_t|y_{1:t})\,dx_t$
↳ since Gaussian, can compute integral in closed form
Kalman Update: $\mu_{t+1}=F\mu_t+K_{t+1}(y_{t+1}-HF\mu_t)$ compute on the fly
$\Sigma_{t+1}=(I-K_{t+1})(F\Sigma_tF^T+\Sigma_x)$ compute offline
Kalman Gain: $K_{t+1}=(F\Sigma_tF^T+\Sigma_x)H^T(H(F\Sigma_tF^T+\Sigma_x)H^T+\Sigma_y)^{-1}$

Multivar. Gaussians: $N(y;\Sigma,\mu)=((2\pi)^{n/2}\sqrt{|\Sigma|})^{-1}\exp(-\frac{1}{2}(y-\mu)^T\Sigma^{-1}(y-\mu))$
$\mu_{A|B}=\mu_A+\Sigma_{AB}\Sigma_{BB}^{-1}(x_B-\mu_B)$ $\Sigma_{A|B}=\Sigma_{AA}-\Sigma_{AB}\Sigma_{BB}^{-1}\Sigma_{BA}$

Bayesian Filtering (for HMMs)

We have $P(X_t|y_{1:t-1})$ and want to compute $P(X_{t+1}|y_{1:t})$
1. Conditioning $P(X_t|y_{1:t})=P(X_t|y_t,y_{1:t-1})$
$=\frac{1}{Z}\cdot P(X_t|y_{1:t-1})P(y_t|X_t,y_{1:t-1})=\frac{1}{Z}\cdot P(X_t|y_{1:t-1})P(y_t|X_t)$ $O(D)$
2. Prediction $P(X_{t+1}|y_{1:t})=\sum_{x_t}P(X_{t+1},x_t|y_{1:t})$
$=\sum_{x_t}P(X_t|x_t,y_{1:t})\cdot P(x_t|y_{1:t})=\sum_{x_t}P(X_{t+1}|x_t)\cdot P(x_t|y_{1:t})$ $O(D^2)$
⇒ recursively compute starting at $P(X_1)$

Particle Filtering

Approximate posterior $P(X_t|y_{1:t})$ at each time with N i.i.d. particles (samples): $P(x_t|y_{1:t})\approx\frac{1}{N}\sum_{i=1}^{N}\delta_{x_{i,t}}(x_t)$
1. Prediction: $x_i'\sim P(X_{t+1}|x_{i,t})$ (sample according to before)
2. Conditioning: $w_i\propto P(y_{t+1}|x_i')$ (compute for each class)
Resample N particles: $X_{i,t+1}\sim\frac{1}{N}\sum_{i=1}^{N}w_i\,\delta_{x_i'}$
Why resampling? If we don't, then usually all weight concentrates on a single particle (mode collapse)

Example Particle Filtering / HMM

HMM with $X\in\{N,V,R\}$ and $Y\in\{C,H\}$. Given probabilities: $P(H|V)=1$, $P(H|R)=1$, $P(C|N)=1$.
Also transition probabilities.
If CHHC is observed, find max likely states.
$\operatorname{argmax}_{x_{1:4}} P(X_{1:4}|CHHC)$ since we know that $P(C|N)=1$, it is the only possible scenario. We can simplify: $\operatorname{argmax}_{x_{2:3}} P(X_{2:3}|X_1=N,X_4=N,CHHC)$. There are four options to compute: RR, RV, VR, VV. → find max
You have 5 particles for N, 3 for V, 2 for R.
The new observation is H, compute the weights.
$w_i\propto P(H|X=x)$. For $x=N$: $w_{1:5}=\frac{1}{Z}P(H|N)=0$
For $x=V/R$: $w_{6:8}=\frac{1}{Z}P(H|V)=\frac{1}{Z}1=\frac{1}{Z}P(H|R)=w_{9:10}$
↳ compute $Z=\sum w_i=5\sim w_{1:5}=0$, $w_{6:10}=1/5$
What is the expected number of particles for N, V, R after resampling?
$\mathbb{E}[\sum_{i=1}^{10}w_i\cdot 1_{X_i,t-1=N}]=0$
$\mathbb{E}[\sum_{i=1}^{10}w_i\cdot 1_{X_{i,t-1}=R}]=\sum_{i=1}^{10}\mathbb{E}[w_i\cdot 1_{X_{i,t-1}=R}]=10\cdot\frac{1}{5}\cdot 2=4$
$\mathbb{E}[\sum_{i=1}^{10}w_i\cdot 1_{X_{i,t-1}=V}]=\sum_{i=1}^{10}\mathbb{E}[w_i\cdot 1_{X_{i,t-1}=V}]=10\cdot\frac{1}{5}\cdot 3=6$

Example Particle Filter

Given: -1D movement on states $x\in\mathbb{Z}$
- $x_{t+1}=x_t+\varepsilon_t$ where ε_t uniformly distributed with integer values in $[-3,3]$
- measurements $y_t=(x_t+\eta_t)^2$ where η_t is dist: $P(\eta_t)=\begin{cases}0.6 & \text{if } \eta_t=0\\ 0.4 & \text{if } \eta_t=1\\ 0 & \text{o.w.}\end{cases}$

You want: particle filter with 6 particles.
robot at t=0: x_0. Particles $x_i=0$ $\forall i\in\{0,\dots,5\}$

You get for ε_0: (-1, -1, 0, 1, 2, 3) (samples)
↳ update particles $x_1'=(-1,-1,0,1,2,3)$

You get measurement $y_1=1$. What are the particle weights?
From the measurement model we obtain:
$\eta_t=\pm\sqrt{y_t}-x_t$ ⇒ $P(y_{t+1}|x_t')=P(\eta_{t+1}=\pm\sqrt{y_{t+1}}-x_t')$

The particle weights are computed as:
$w_i=\frac{1}{2}P(y_{t+1}|x_i')$: $P(y_1=1|x_0'=-1)=P(\eta_1=0)+P(\eta_1=2)$
$=0.6+0.0=0.6$ ⇒ same for others
(0.6, 0.6, 0.4, 0.6, 0.2, 0.0)
Compute $Z=\sum_{i=0}^{5}P(y_1=1|x_i')=0.6+0.6+0.4+0.6+0.2=\frac{24}{10}$
⇒ compute weights: $w_0=5/24,\dots,w_5=0$

In reality need more samples to capture probability distributions accurately.

Partially Observable MDPs (POMDP)

Idea: interpret POMDP as an MDP with enlarged state space: $P(x_{t+1}|x_t,A_t)$, new states correspond to beliefs $P(x_t|y_{1:t})$ in the original MDP.
Can calculate optimal action using dynamic programming
Can approximate using policy gradients

Markov Decision Process (MDP)

States $X=\{1,\dots,n\}$, actions $A=\{1,\dots,m\}$, transition probabilities $P(x'|x,a)$, reward function $r(x,a,x')$.
Modes: Finite horizon (T steps) or discounted rewards $\gamma<1$

Value of policy: $V_\pi(x)=\mathbb{E}[\sum_{t=0}^{\infty}\gamma^t r(X_t,\pi(X_t))\,|\,X_0=x]$
$=\sum_{x'}P(x'|x,\pi(x))[r(x,\pi(x),x')+\gamma V_\pi(x')]=r(x,\pi(x))+\gamma\sum_{x'}P(x'|x,\pi(x))V_\pi(x')$

Greedy Policy: $\pi_g(x)=\operatorname{argmax}_a\sum_{x'}P(x'|x,a)(r(x,a,x')+\gamma V_\pi(x'))$

Bellman Equation: Policy is optimal ⇔ greedy w.r.t. its induced value function. $V^*(x)=\max_a[r(x,a)+\gamma\sum_{x'}P(x'|x,a)V^*(x')]$

Policy Iteration

Start with arbitrary (e.g. random) policy
Until converged do:
  Compute value function $V^\pi(x)$
  $V^\pi(x)=r(x,\pi(x))+\gamma\sum_{x'}P(x'|x,\pi(x))V^\pi(x')$
  Compute greedy policy π_G w.r.t. V^π
  $\pi_V(x)=\operatorname{argmax}_a r(x,a)+\gamma\sum_{x'}P(x'|x,a)V(x')$
  $\pi\leftarrow\pi_G$

Guaranteed to monotonically improve and converge in $O(n^2m/(1-\gamma))$ iterations
Cost per iteration: $O(|S|^3+|S|\cdot|A|\cdot D^T)$ (max paths a action can have ≤ |S|)

Value Iteration

Initialize $V_0(x)=\max_a r(x,a)$
For t=1 to ∞ do
  For each x, a let
    $Q_t(x,a)=r(x,a)+\gamma\sum_{x'}P(x'|x,a)V_{t-1}(x')$
  For each x let
    $V_t(x)=\max_a Q_t(x,a)$
  Break if $\|V_t-V_{t-1}\|_\infty=\max|V_t(x)-V_{t-1}(x)|\leq\varepsilon$
Then choose greedy policy w.r.t. V_t

Guaranteed to converge to ε-optimal policy
Cost per iteration: $O(|S|^2|A|\cdot D^T)$ (max paths a action can have ≤ |S|)

Exploration-Exploitation Dilemma

Too much exploration: get correct estimates, may do poorly
Too much exploitation: quickly get reward, stuck suboptimal

Learning BN: Parameter learning

Given structure & complete observations, can get MLE with $\hat\theta_{x_i|Pa_i}=\text{count}(x_i,Pa_i)/\text{count}(Pa_i)$ ~ can also use beta prior with pseudo counts

Mutual Information:

$I(X_A,X_B)=\mathbb{E}_{X_A,X_B}\left[\log\frac{P(X_A,X_B)}{P(X_A)\cdot P(X_B)}\right]$

$\hat I(X_A,X_B)=\sum_{x_i,x_j}\hat P(x_i,x_j)\log\left(\frac{\hat P(x_i,x_j)}{\hat P(x_i)\cdot\hat P(x_j)}\right)\frac{\text{count}(x_i)}{N}$

Finding optimal tree: Do maximum spanning tree over $\hat I(X_A,X_B)$. Pick root, orient edges away from the root. (Chow-Liu Algorithm)

Regularized: Score $=\sum_{i=1}^{n}\hat I(X_i,Pa_i)-\frac{\log N}{2N}|G|$

Model-Based RL

Idea: Learn MDP, optimize policy based on MDP
Estimate transitions and rewards based on counts
ε-greedy: With prob. ε_t pick random action, $(1-\varepsilon_t)$: best action
Rmax: If r(x,a) unknown, set $r(x,a)=R_{max}$, if P(x'|x,a) unknown, set $P(x^*|x,a)=1$ where $r(x^*,a)=R_{max}$ $\forall a$, $P(x^*|x^*,a)=1$
Execute π, $\forall(x,a)\in$ visited: update r(x,a), estimate P(x'|x,a) and after "enough" steps recompute π in current model.

Model-Free RL: Q-Learning

When observing a transition x, a, x' with r

$Q_{new}(x,a)=(1-\alpha_t)Q_{old}(x,a)+\alpha_t(r+\gamma\max_{a'}Q_{old}(x',a'))$

Learning rate condition: $\sum_t\alpha_t=\infty$, $\sum_t\alpha_t^2<\infty$
if they hold & actions chosen randomly → will converge

R_{max}: initialize with $Q(x,a)=\frac{R_{max}}{1-\gamma}\prod_{i=1}^{\text{Limit}}(1-\alpha_i)^{-1}$
↳ idea: give high reward for exploration

Deep Q Networks

Idea: use CNN to approximate Q function. Clone network to maintain constant "target" values accross episodes.
$L(\theta)=\sum_{(x,a,r,x')\in D}(r+\gamma\max_{a'}Q(x',a';\theta^{old})-Q(x,a,\theta))^2$
Double DQN: a' chosen as $\operatorname{argmax}_{a'}Q(x,a;\theta)$ instead $\operatorname{argmax}_{a'}Q(x,a;\theta^{old})$

Monte Carlo Tree Search

Heuristic Algorithm. Approximate the value of a state by repeated simulation (simulate to terminal state, according to default policy e.g. random, observe reward, propagate back result). Steer the search using these values. Iterate and select best action after sufficiently many runs or if time budget exceeded.

Policy Gradient Method: Directly learn parameterized policy. For episodic tasks compute expected reward

Gaussian Processes: normal distribution over functions.