Probabilistic AI

Probability & Statistics

Hoeffdings inequality:
$P(|E_p[f(X) - \frac{1}{N}\sum_{i=1}^N f(X_i)| > \varepsilon)$
$\leq 2\exp\left(\frac{-2N\varepsilon}{C^2}\right) =: \delta \Rightarrow N \geq \frac{C^2}{2\varepsilon^2}\log\left(\frac{2}{\delta}\right)$.
Chain Rule:
$P(X_{1:n}) = P(X_1)P(X_2|X_1)...P(X_n|X_{n-1})$
Bayes Rule:
$P(X|Y) = \frac{P(X)P(Y|X)}{\sum_x P(X=x)P(Y|X=x)}$
with Background:
$P(X|Y,B) = \frac{P(X|B)P(Y|X,B)}{P(Y|B)}$

Multivariate Gaussian
$p(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = ((2\pi)^d \cdot |\boldsymbol{\Sigma}|)^{-1/2}.$
$\cdot \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$
$P(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$

Conditional Independence:
If $P(Y = h|Z = z) > 0$, then $X \perp Y|Z \Leftrightarrow P(X = x|Z = z, Y = y) = P(X = x \mid Z = z)$.
Properties:
• Contraction: $(X \perp Y|Z) \wedge (X \perp W|Y,Z)$
$\Rightarrow X \perp X, W|Z$.
• Weak union:
$X \perp Y, W|Z \Rightarrow X \perp Y|W,Z$
• Intersection:
$(X \perp Y|W,Z) \wedge (X \perp W|Y,Z)$
$\Rightarrow X \perp Y, W|Z$

ICL Cond. Indep. (d-sep):
A path is **blocked** if it includes a node s.t.
• The arrows on the path meet either head-to-tail or tail-to-tail at the node, and the node is in the observed set, or
• The arrows meet head-to-head at the node, and neither the node nor any of its descendants is in the observed set.

Bayesian Linear Regression
Model: $Y = \mathbf{X}\beta + \epsilon,\ \epsilon \sim \mathcal{N}(0,\sigma^2)$
Likelihood: $P(Y|\mathbf{X},\beta,\sigma^2) = \mathcal{N}(\mathbf{X}\beta,\sigma^2)$
Prior: $P(\beta|\boldsymbol{\Lambda}) = \mathcal{N}(0,\boldsymbol{\Lambda}^{-1})$
Posterior: $P(\beta|\mathbf{X},\mathbf{y},\boldsymbol{\Lambda}) = \mathcal{N}(\mu_\beta,\Sigma_\beta)$
$\mu_\beta = (\mathbf{X}^T\mathbf{X} + \sigma^2\boldsymbol{\Lambda})^{-1}\mathbf{X}^T\mathbf{y}$
$\Sigma_\beta = \sigma^2(\mathbf{X}^T\mathbf{X} + \sigma^2\boldsymbol{\Lambda})^{-1}$
$(\beta-\mu_\beta)^\top\Sigma_\beta^{-1}(\beta-\mu_\beta)$
$= \beta^\top\Sigma_\beta^{-1}\beta - 2\beta^\top\Sigma_\beta^{-1}\mu_\beta + \mu_\beta^\top\Sigma_\beta^{-1}\mu_\beta$

Conditioning a Gaussian:
$p(\mathbf{x}_a|\mathbf{x}_b) = \mathcal{N}(\mathbf{x}_a|\mu_{a|b},\Sigma_{a|b})$
$\mu_{a|b} = \mu_a + \Sigma_{ab}\Sigma_{bb}^{-1}(\mathbf{x}_b - \mu_b), \Sigma_{a|b}$
$= \Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba}$
$\Lambda_{aa} = (\Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba})^{-1},$
$\Lambda_{ab} = -\Lambda_{aa}\Sigma_{ab}\Sigma_{bb}^{-1}$

Gaussian Process
Distribution over functions.
Prior: $P(f)$.
Likelihood: $P(data|f)$.
Posterior: $P(f|data)$.

Joint distribution of $[\mathbf{y}, y_{n+1}]$ is given by $\mathbf{y}|\mathbf{X},\sigma^2 \sim \mathcal{N}(0, \mathbf{X}^T\Lambda^{-1}\mathbf{X} + \sigma^2\mathbf{I})$, kernelised version:

$$p\left(\begin{bmatrix}\mathbf{y}\\ y_{n+1}\end{bmatrix}|x_{n+1},\mathbf{X},\sigma^2\right) = \mathcal{N}\left(\mathbf{0},\begin{bmatrix}\mathbf{C}_n & \mathbf{k}\\ \mathbf{k}^T & c\end{bmatrix}\right)$$

$\mathbf{C}_n = \mathbf{K} + \sigma^2\mathbf{I}_n; \quad c = k(x_{n+1},x_{n+1}) + \sigma^2$
$\mathbf{k} = k(x_{n+1},\mathbf{X}); \quad \mathbf{K} = k(\mathbf{X},\mathbf{X})$

$$p\left(\begin{bmatrix}\mathbf{a}_1\\ \mathbf{a}_2\end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix}\mathbf{a}_1\\ \mathbf{a}_2\end{bmatrix} \mid \begin{bmatrix}\mathbf{u}_1\\ \mathbf{u}_2\end{bmatrix}, \begin{bmatrix}\Sigma_{11} & \Sigma_{12}\\ \Sigma_{21} & \Sigma_{22}\end{bmatrix}\right)$$

$\mathbf{a}_1, \mathbf{u}_1 \in \mathbb{R}^e; \mathbf{a}_2, \mathbf{u}_2 \in \mathbb{R}^d;$
$\Sigma_{11} \in \mathbb{R}^{e\times e}$ PSD; $\Sigma_{12} \in \mathbb{R}^{e\times f}$ PSD
$\Sigma_{22} \in \mathbb{R}^{f\times f}$ PSD; $\Sigma_{2,1} \in \mathbb{R}^{f\times e}$ PSD

Predictive density:
$p(y_{n+1}|x_{n+1},\mathbf{X},\mathbf{y}) = \mathcal{N}(\mu_{n+1},\sigma_{n+1}^2)$
$\mu_{n+1} = \mathbf{k}^T\mathbf{C}_n^{-1}\mathbf{y}$; $\sigma_{n+1}^2 = c - \mathbf{k}^T\mathbf{C}_n^{-1}\mathbf{k}$

Bayesian Networks

Variable Elimination: (for MAP, MPE)
Given query $P(X|E=e)$
• Pick ordering $X_1,...,X_n$.
• Initialisation: $f_i = P(X_i|Pa_{X_i})$.
• For $i = 1,...,n,\ X_i \notin \{X,E\}$:
. • Multiply all factors incl. X_i.
. • $g_i := \sum_{x_i}\prod_j f_j$ (or $\max_{x_i}\prod_j f_j$).
• Renormalise.

Factor Graphs:
Probability measure factorises:
$P(X_1,...,X_n) = \frac{1}{Z}\prod_i \Psi_i(X_{A_i})$

Sum-Product Algorithm (for MPE)
• Initialise $\mu_{x\to f}(x) = 1$
• Initialise $\mu_{f\to x}(x) = f(x)$
• Until convergence, pass messages
. • from node to factor:
$\mu_{x_m\to f_s}(x_m) = \prod_{\ell\in ne(x_m)\setminus f_s}\mu_{f_\ell\to x_m}(x_m)$
. • from factor to node:
$\mu_{f_s\to x}(x) = \sum_{x_1,...,x_M} f_s(x,x_1,...,x_M)\cdot$
$\cdot\prod_{m\in ne(f_s)\setminus x}\mu_{x_m\to f_s}(x_m)$.
The *marginal* is given as the product of all incoming messages:
$p(x) \propto \prod_{f_i\in ne(x)}\mu_{f_i\to x}(x)$;
$p(\mathbf{X}_u = \mathbf{x}_u) \propto f_u(\mathbf{x}_u)\prod_{v\in ne(u)}\mu_{v\to u}(x_v)$.

Parameter Learning:
ML Approach:
$\hat{\theta}_{X_i|Pa_{X_i}} = \frac{Count(X_i,Pa_{X_i})}{Count(Pa_{X_i})}$
(can use EM for incomplete observations)
Imposing a Prior: (Beta prior here)
$\hat{\theta}_{F=cherry} = \frac{Count(F=cherry)+\alpha_{cherry}}{N+\alpha_{cherry}+\alpha_{lime}}$

Structure Learning:
MLE Score:
$S_{ML}(G;D) = \max_\theta\{\log\big(P(D|\theta,G)\big)\}$
$\log P(D|\theta_G,G) = N\sum_{i=1}^n \hat{I}(X_i,Pa_{X_i}) + C,$
where θ_G is the maximiser.
G^*
$= \arg\max_G \underbrace{\sum_{i=1}^n \hat{I}(X_i,Pa_{X_i})}_{\text{ML Score}} - \underbrace{\frac{\log(N)}{2N}|G|}_{\text{BIC}},$
where $|G|$ is the number of parameters

Mutual Information: (Information gain)
$I(X_i,X_j) = \sum_{x_i,x_j} P(x_i,x_j)\log\left(\frac{P(x_i,x_j)}{P(x_i)P(x_j)}\right)$
Properties:
•$I(X_a,X_b) \geq 0$.
•$I(X_a,X_b) = 0 \Leftrightarrow X_a \perp X_b$.
•$I(X_a,X_b) = I(X_b,X_a)$ (Symmetry).
•$\forall B \subseteq C: \quad I(X_A,X_B) \leq I(X_A,X_C)$ (Monotonicity).
•$I(X_a,X_b) = H(X_a) - H(X_a|X_b)$,
where $H(x_i) = -\sum_{x_i} P(x_i)\log\big(P(x_i)\big)$ is *entropy*.

Chow-Liu Algorithm:
• For each (X_i,X_j), compute $\hat{P}(X_i,X_j) = \frac{Count(X_i,X_j)}{N}$ and $\hat{I}(X_i,X_j)$.
• Define complete graph with edges weighted by $\hat{I}(X_i,X_j)$.
• Find maximum spanning tree.
• Pick any variable as the root orient edges away.

Sampling-based Inference

Forward Sampling:
• Sort variables topologically $X_1,...,X_n$.
• for $i = 1,...,n$:
. • Sample:
$x_i \sim P(X_i|X_1 = x_1,...,X_{i-1} = x_{i-1})$.
$\hat{P}(X_a = x_a) = \frac{1}{N}Count(X_a)$.
$\hat{P}(X_a = x_a|X_b = x_b) = \frac{Count(x_a,x_b)}{Count(x_b)}$.
Absolute Error: (Hoeffdings ineq. $C = 1$)
$Prob(\hat{P}(x) \notin [P(x)-\varepsilon, P(x)+\varepsilon])$
$\leq 2\exp\big(-2N\varepsilon^2\big)$
Relative Error:
$Prob(\hat{P}(x) \notin P(x)\cdot(1\pm\varepsilon))$
$\leq 2\exp\big(-NP(x)\varepsilon^2/3\big)$

Detailed Balance:
$\forall x,x',\ Q(x)P(x'|x) = Q(x')P(x|x')$
$Q(x)T(x,x') = Q(x')T(x,x')$,
where $T(x,x') = P(x'|x)$

Gibbs Sampling:
• Start with assignment x to all variables.
• Fix observed variables X_b to x_b.
• for $t = 1,...,\infty$:

. • Pick i uniformly at random from $\{1,...,n\} \setminus b$.
. • Sample $x_i \sim P(X_i|X_{\{1,...,n\}\setminus\{i\}})$ (update x_i).

Designing Markov Chains (MCMC)
(1) Proposal distribution: $R(X'|X)$
(2) Acceptance distribution: . •$X_t = x$
. • With prob. $a = \min\left\{1, \frac{Q(x')R(x|x')}{Q(x)R(x'|x)}\right\}$, set $X_{t+1} = x'$
. • With prob. $(1-a)$ set $X_{t+1} = x$
Thm (Metropolis, Hastings): The stationary distribution is $\frac{1}{Z}Q(x)(= P(x))$.

Ergodicity:
Stationary Markov Chain is *ergodic* if $\exists t < \infty$ s.t. every state can be reached from every state in *exactly* t steps.

Temporal Models

Markov Assumption:
$X_{1:t-1} \perp X_{t+1:T}|X_t \quad (0 < t < T)$

Stationarity Assumption:
$\forall x, x', \;\; P(X_{t+1} = x|X_t = x')$ does not depend on t.

Stationary distribution:
π does not depend on the initial state.
$\pi(x) = \lim_{N\to\infty} P(X_t|Y_{1:t})$.
(2016P7 HMM):
Solve for π_b, π_f for the stationary distrib.
$\pi_b = \frac{1}{4}\pi_b + \frac{3}{4}\pi_f$
$\pi_f = \frac{3}{4}\pi_b + \frac{1}{4}\pi_f$

Inference Tasks in HMMs:
• *Filtering*: $P(X_t|Y_{1:t})$.
• *Prediction*: $P(X_{t+k}|Y_{1:t}),\ k \in \mathbb{N}$.
• *Smoothing*: $P(X_t|Y_{1:T}),\ t < T$.
• *MPE*: $\text{argmax}_{x_{1:T}} P(x_{1:T}|Y_{1:T})$.

Bayesian Filtering:
• Start with $P(X_1)$.
• At time t:
. • Assume we have $P(X_t|y_{1:t-1})$.
. • *Conditioning*:
$P(X_t|y_{1:t}) = \frac{P(X_t|y_{1:t-1})P(y_t|X_t)}{\sum_x P(x,y_t|y_{1:t})}$.
. • *Prediction*:
$P(X_{t+1}|y_{1:t}) = \sum_x P(X_{t+1}|x)P(x|y_{1:t})$.

Particle Filtering:
• Suppose $P(X_t|y_{1:t}) \approx \frac{1}{N}\sum_{i=1}^N \delta_{x_i,t}$ (delta-dirac)
• *Prediction*:
Propagate each particle i:
$x_i' \sim P(X_{t+1}|x_{i,t})$.
• *Conditioning*:
Weigh particles:
$w_i = \frac{1}{Z}P(y_{t+1}|x_i')$.
Resample N particles:
$x_{i,t+1} \sim \sum_{i=1}^N w_i \delta_{x_i'}$.

Kalman Filters:
$P(X_{t+1}|X_t)$: (motion model)
$\mathbf{X}_{t+1} = \mathbf{F}\mathbf{X}_t + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, \Sigma_x)$.
$P(Y_t|X_t)$: (sensor model)
$\mathbf{Y}_t = \mathbf{H}\mathbf{X}_t + \eta_t, \quad \eta_t \sim \mathcal{N}(0, \Sigma_y)$.

General Kalman Update:
• Transition model:
$P(\mathbf{x}_{t+1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{t+1}; \mathbf{F}\mathbf{x}_t, \Sigma_x)$
• Sensor model:
$P(\mathbf{y}_t|\mathbf{x}_t) = \mathcal{N}(\mathbf{y}_t; \mathbf{H}\mathbf{x}_t, \Sigma_y)$
• *Kalman update*:
$\mu_{t+1} = \mathbf{F}\mu_t + \mathbf{K}_{t+1}(\mathbf{y}_{t+1} - \mathbf{H}\mathbf{F}\mu_t)$.
$\Sigma_{t+1} = (\mathbb{I} - \mathbf{K}_{t+1})(\mathbf{F}\Sigma_t\mathbf{F}^\top + \Sigma_x)$.
• *Kalman gain*:
$\mathbf{K}_{t+1} = (\mathbf{F}\Sigma_t\mathbf{F}^\top + \Sigma_x)\mathbf{H}^\top(\mathbf{H}(\mathbf{F}\Sigma_t\mathbf{F}^\top + \Sigma_x)\mathbf{H}^\top + \Sigma_y)^{-1}$.

Markov Decision Process

Expected value for policy:
$J(\pi) = E[r(X_0, \pi(X_0)) + \gamma r(X_0, \pi(X_0)) + \gamma^2 r(X_0, \pi(X_0)) + ...]$
Value Function:
$V^\pi(x) = J(\pi|X_0 = x)$
$= E\left[\sum_{t=0}^\infty \gamma^t r(X_t, \pi(X_t))|X_0 = x\right]$

Bellman Theorem:
Policy optimal ⇔ greedy w.r.t. values fct.
$V^*(x)$
$= \max_a\{r(x,a) + \gamma\sum_{x'} P(x'|x,a)V^*(x')\}$

Policy Iteration:
• Start w/ an arbitrary (educated guess) π.
• Until convergence:
. • Compute $V^\pi(x),\ \forall x$.
. • Compute greedy policy π_g w.r.t. V^π.
. • $\pi = \pi_g$; $[O^*(n^2m/(1-\gamma)]$

Value Iteration:
• Initialise $V_0(x) = \max_a\{r(x,a)\}$.
• for $t = 1, ..., \infty$:
. • $\forall x, a, \quad Q_t(x,a) := r(x,a) + \gamma\sum_{x'} P(x'|x,a)V_{t-1}(x')$.
. • $\forall x, \;\; V_t(x) := \max_a\{Q(x,a)\}$.
. • Break if $\|V_t - V_{t-1}\|_\infty \le \varepsilon$.
• Choose greedy policy w.r.t. V_t.

Reinforcement Learning

Learning the MDP:
ML approach:
(estimate transitions):
$\hat{P}(X_{t+1}|X_t, A) = \frac{Count(X_{t+1}, X_t, A)}{Count(X_t, A)}$.
(estimate rewards):
$\hat{r}(x,a) = \frac{1}{N_{x,a}}\sum_{t:\ X_t=x, A_t=a} R_t$.
Can be combined w/ ε-greedy exploration.

Robbins-Monro (RM) Conditions:
$\varepsilon_t \xrightarrow{t\to\infty} 0, \;\; \sum_{t=1}^\infty \varepsilon_t \to \infty, \;\; \sum_{t=1}^\infty \varepsilon^2 < \infty$

Rmax Exploration:
• Initialise $r(x,a) = R_{\max}$
• Initialise $P(x^*|x,a) = 1$, where x^* is a 'fairy tale' state.
Converges to ε-optimal policy with probability $1-\delta$ in time polynomial in $|X|, |A|, T, \frac{1}{\varepsilon}$ and $\log\left(\frac{1}{\delta}\right)$.
$\forall T$ time steps w/ high prob., $R_{\max}$ either
• Obtains near-optimal reward; or
• Visits at least one unknown (x,a).

Q-Learning:
• Initialise $Q(x,a)$ arbitrarily.
• $x = x_0$.
• for $t = 1, ..., \infty$:
. • Perform some action a
. • Observe r and x'.
. • $Q(x,a) = (1-\alpha_t)Q(x,a) + \alpha_t[r(x,a,x') + \gamma\max_{a'}\{Q(x',a')\}]$
If $\{\alpha\}_t$ satisfies RM conditions Q-Learning converges to Q^* with probability 1.

Optimistic Q-Learning: ($\approx$ to $R_{\max}$)
Initialise $Q(x,a) = \frac{R_{\max}}{1-\gamma}\prod_{t=1}^{T_{\text{init}}}(1-\alpha)^{-1}$.
With prob. $1-\delta$ obtains ε-opt. policy after time polynomial in $|X|, |A|, \frac{1}{\varepsilon}, \log\left(\frac{1}{\delta}\right)$.

Parametric Q-Learning
$Q(x,a;\theta) = \theta^\top\phi(x,a)$,
where ϕ =features of (x,a).
Optimise
$L(\theta) = \sum_{(x,a,r,x')\in D}$
$(r + \gamma\max_{a'}\{Q(x',a';\theta^{\text{old}}) - Q(x,a;\theta))^2$.

Policy Search & Bayes. Opt.

Upper Confidence Sampling:
$x_t = \arg\max_{x\in D}\{\mu_{t-1}(x) + \beta_t\sigma_{t-1}(x)$

Choosing β_t:
For learnt $\|f\|_k$, we choose
$\beta_t = \mathcal{O}(\gamma_t\log^3(t))$
Bounds on γ_t:
$\gamma_t = \mathcal{O}((\log(T)^{d+1})$ (squared-exponential)
$\gamma_t = \mathcal{O}(d\log(T))$ (linear kernel)
$\gamma_t = \mathcal{O}(T^{\frac{d(d+1)}{2\nu+d(d+1)}}\log(T))$ (Matérn $\nu > 2$)

Safe Bayesian Optimisation:
$\max_\theta\{f(\theta)$ s.t. $g(\theta) \ge 0$.
Idea: keep track of the lower bound.
Safe opt. algo.: Solves this problem under conditions on f and g, $\exists T(\varepsilon,\delta)$, s.t. $\forall\varepsilon > 0, \delta > 0$, w/ prob. $1-\delta$
(1) No unsafe decision
(2) ε-optimal in $\mathcal{O}(T(\varepsilon,\delta))$.