Rules for the Mean and Variance

$$
\begin{aligned}
E(cX) &= cE(X) \\
E(X+Y) &= E(X) + E(Y) \\
Var(cX) &= c^2 Var(X) \\
Var(X \pm Y) &= Var(X) + Var(Y) \\
&\pm 2Cov(X,Y) \\
Var(X) &= \mathbb{E}[(X - \mathbb{E}[X])^2]
\end{aligned}
$$

Bayesian Linear Regression

$$
\begin{aligned}
p(w|X,y) &= \mathcal{N}(w; \bar{\mu}, \bar{\Sigma}) \\
\bar{\mu} &= (X^T X + \frac{\sigma_n^2}{\sigma_p^2} I)^{-1} X^T y \\
\bar{\Sigma} &= (\frac{1}{\sigma_n^2} X^T X + \frac{1}{\sigma_p^2} I)^{-1} \\
y^* &= w^T x^* + \epsilon \\
p(y^*|X,y,x*) &= \mathcal{N}(\bar{\mu}^T x^*, x^{*T} \bar{\Sigma} x^* + \sigma_n^2)
\end{aligned}
$$

MLE and MAP regression

$$
\begin{aligned}
w_{MLE} &= (X^T X)^{-1} X^T y \\
w_{MAP} &= (I \frac{\sigma_n^2}{\sigma_p^2} + X^T X)^{-1} X^T y
\end{aligned}
$$

Gaussian Proccesess

$A \in \mathbb{R}^{m \times n}$, f_A is a collection of R.V s.t. $f_A \sim \mathcal{N}(\mu_A, K_{AA})$.

$$
K_{AA} = \begin{bmatrix} k(x_1,x_1) & \dots & k(x_1,x_m) \\ \vdots & \ddots & \vdots \\ k(x_m,x_1) & \dots & k_(x_m,x_m) \end{bmatrix}
$$

$$
\mu_A = \begin{bmatrix} \mu(x_1) \\ \vdots \\ \mu(x_m) \end{bmatrix}
$$

For more than one new point $k(x,x')$ is a matrix like K_{AA}.

$$
\begin{aligned}
\mu'(x) &= \mu(x) \\
&+ k_{x,A}(K_{AA} + \sigma_n^2 I)^{-1}(y_A - \mu_A) \\
k'(x,x') &= k(x,x') \\
&- k_{x,A}(K_{AA} + \sigma_n^2 I)^{-1} k_{x',A}^T \\
k_{x,A} &= \begin{bmatrix} k(x_1,x) \\ \vdots \\ k(x_m,x) \end{bmatrix}
\end{aligned}
$$

Online GP's

$K_{AA} = k(x_{t+1}, x_{t+1})$ then calculate the posterior for a new arbitrary data point $x*$.

Maximize the marginal likelihood of the data

$K(\theta)$ is the Kernel matrix.

$$
\begin{aligned}
&\underset{\theta}{\text{argmax}} \int p(y_{train} \mid f, x_{train}, \theta) p(f \mid \theta) df \\
=&\underset{\theta}{\text{argmax}} \int \mathcal{N}(f(x), \sigma_n^2) \mathcal{N}(0, K(\theta)) df \\
=&\underset{\theta}{\text{argmax}}\, \mathcal{N}(0, K(\theta) + I\sigma_n^2) \\
=&\underset{\theta}{\text{argmax}}\, p(y_{train} \mid x_{train}, \theta)
\end{aligned}
$$

Variational Inference

KL divergence

Reverse KL div: $KL(q||p)$. Forward KL: $KL(p||q)$ (gives more conservative variance estimates).

$$
KL(q||p) = \int q(\theta) \log \frac{q(\theta)}{p(\theta)} d\theta
$$

Minimizing KL divergence

$$
\begin{aligned}
&\underset{q \in Q}{\text{argmin}}\, KL(q||p(\theta|y)) \\
=& \underset{q \in Q}{\text{argmax}}\, \mathbb{E}_{\theta \sim q(\theta)}[\log p(\theta, y)] + H(q) \\
=& \underset{q \in Q}{\text{argmax}}\, \mathbb{E}_{\theta \sim q(\theta)}[\log p(y|\theta)] - KL(q||p(\theta))
\end{aligned}
$$

Gradient of the ELBO

$$
\begin{aligned}
&\nabla_\lambda \mathbb{E}_{\theta \sim q_\lambda}[f(\theta)] \\
=& \mathbb{E}_{\epsilon \sim \phi}[\nabla_\lambda f(g(\epsilon; \lambda))] \\
=& \nabla_{C,\mu} \mathbb{E}_{\epsilon \sim \mathcal{N}(0,I)}[\log p(y|C\epsilon + \mu)] \\
=& n \mathbb{E}_{\epsilon \sim \mathcal{N}(0,I)} \\
& \mathbb{E}_{i \sim \mathcal{U}(1,..,m)}[\nabla_{C,\mu} \log p(y_i|C\epsilon + \mu x_i)] \\
=& \frac{n}{m} \sum_{j=i}^{m} \nabla_{C,\mu} \log p(y_i|C\epsilon + \mu x_i)
\end{aligned}
$$

MCMC methods

Hoeffding's inequality

Given f is bounded between $[0, C]$:

$$
P(|\mathbb{E}_P[f(X)] - \frac{1}{N} \sum_{i=1}^{N} f(x_i)| > \epsilon) \leq
$$

$$
2 \exp^{\frac{-2N\epsilon^2}{C^2}}
$$

Error less than ϵ with probability $1 - \delta$:

$$
2 \exp^{\frac{-2N\epsilon^2}{C^2}} \leq \delta
$$

MH-MCMC

DBE: $Q(x)P(x'|x) = Q(x')P(x|x')$.

$$
\begin{aligned}
& R(X'|X = x) \\
& X_{t+1} = x', P(X_{t+1} = x') = \alpha \\
& \alpha = min \left\{1, \frac{Q(x')R(x|x')}{Q(x)R(x'|x)}\right\} \\
& \text{o.t.w } X_{t+1} = x
\end{aligned}
$$

Continuous RV

$$
\begin{aligned}
& p(x) = \frac{1}{Z} e^{-f(x)} \\
& \alpha = min \left\{1, \frac{R(x|x')}{R(x'|x)} e^{f(x) - f(x')}\right\}
\end{aligned}
$$

If $R(x'|x) = \mathcal{N}(x, \tau I)$ then $\alpha = min \left\{1, e^{f(x) - f(x')}\right\}$. Guaranteed efficient convergence for log-concave densities (f convex).

Improved Proposals

Metropolis adjusted Langevin (gradient for proposals), Stochastic Gradient Langevin Dynamics, Hamiltonian Monte Carlo (momentum).

Bayesian Neural Networks

MAP estimation with BNN's

$$
\begin{aligned}
\hat{\theta} &= \underset{\theta}{\text{argmin}}\ - \log p(\theta) - \sum_{i=1}^{n} \log p(y_i|x_i, \theta) \\
&= \underset{\theta}{\text{argmin}}\, \lambda ||\theta||_2^2 \\
&+ \frac{1}{2} \sum_{i=1}^{n} [\frac{1}{\sigma(x_i,\theta)^2} ||y_i - \mu(x_i,\theta)||_2^2 \\
&+ \log \sigma(x_i,\theta)^2]
\end{aligned}
$$

Variational Inference in BNN's

$$
\begin{aligned}
& p(y^* \mid x^*, X, y) \\
=& \int p(y^* \mid x^*, \theta) p(\theta \mid X, y) d\theta \\
=& \mathbb{E}_{\theta \sim p(\theta|X,y)} \left[p(y^* \mid x^*, \theta)\right] \\
\approx& \mathbb{E}_{\theta \sim q_\lambda} \left[p(y^* \mid x^*, \theta)\right] \\
\approx& \frac{1}{m} \sum_{j=1}^{m} p(y^* \mid x^*, \theta^{(j)}) \\
=& \frac{1}{m} \sum_{j=1}^{m} \mathcal{N}(\mu(x^*,\theta), \sigma^2(x^*,\theta))
\end{aligned}
$$

Uncertainty for Gaussians

$$
\begin{aligned}
& Var[y^\star|X,y,x^\star] = \mathbb{E}[\underset{aleat}{Var}[y^\star|x^\star,\theta]] \\
& + \underset{epis}{Var}[\mathbb{E}[y^\star|x^\star,\theta]] \\
& \approx \frac{1}{m} \sum_{j=1}^{m} \sigma^2(x^\star, \theta^{(j)}) \\
& + \frac{1}{m} \sum_{j=1}^{m} \left(\mu(x^\star, \theta^{(j)}) - \bar{\mu}(x^\star)\right)^2
\end{aligned}
$$

MC Dropout and Probabilistic Ensembles

$$p(y^* \mid x^*, X, y) \approx \frac{1}{m}\sum_{j=1}^{m} p(y^* \mid x^*, \theta^{(j)})$$

Active Learning

Given $Y = X + \epsilon$ and $\epsilon \sim \mathcal{N}(0, \sigma_n^2 I)$.

$$\begin{aligned} I(Y;X) &= H(Y) - H(Y|X) \\ &= H(Y) - H(\epsilon) \\ &= \frac{1}{2}\ln|I + \sigma_n^{-2}\Sigma| \end{aligned}$$

Uncertainty Sampling

S is the optimal set of observations, S_t the greedy set. Following the same regression scheme as before.

$$I(f(x_T), y_T) \geq \left(1 - \frac{1}{e}\right) \max_{|S| \leq T} I(f(x_S), y_S)$$

$$x_{t+1} = \operatorname*{argmax}_x \mathbb{I}(f; y_x | y_{S_t})$$

$$= \operatorname*{argmax}_x \frac{1}{2}\log\left(1 + \frac{\sigma_t^2(x)}{\sigma_n^2}\right)$$

Active Learning for Classification

Uncertainty sampling: $x_{t+1} = \operatorname{argmax}_x H(Y|x, X_t, Y_t)$. Better to use approximate inference to estimate MI:

$$\begin{aligned} x_{t+1} &= \operatorname*{argmax}_{x \in D} \mathbb{I}(\theta; y_{t+1} | Y_t, X_t, x_{t+1}) \\ &= H(y_{t+1}|Y_t, X_t, x_{t+1}) \\ &- \mathbb{E}_{\theta \sim p(|X_t, Y_t)}[H(y_{t+1}|\cdot, \theta)] \\ &\approx H(y_{t+1}|Y_t, X_t, x_{t+1}) \\ &- \frac{1}{m}\sum_{j=1}^{m} H(y_{t+1}|\cdot, \theta^{(j)}) \end{aligned}$$

Bayesian Optimization

Cumulative Regret

$$\frac{1}{T}\sum_{t=1}^{T}[f(x^*) - f(x_t)] \to 0$$

Upper confidence sampling

Convergence of the cumulative regret as a function of $\gamma_T = \max_{|S| \leq T} I(f; y_S)$

Thomson Sampling

Sample $\tilde{f} \sim \mathcal{P}(f|X_t, Y_t)$, and then $x_{t+1} \in \operatorname*{argmax}_{x \in D} \tilde{f}(x)$.

Markov Decision Processes

Expected Value of a Policy

For a deterministic reward, some π and state x:

$$\begin{aligned} &J(\pi|X_0 = x) = V^\pi(x) \\ &= \mathbb{E}[\sum_{t=0}^{\infty}\gamma^t r(X_t, \pi(X_t))|X_0 = x] \\ &= r(x, \pi(x)) + \gamma\sum_{x'} P(x'|x, \pi(x))V^\pi(x') \\ &V^\pi = r^\pi + \gamma T^\pi V^\pi \\ &V^\pi = (I - \gamma T^\pi)^{-1} r^\pi \end{aligned}$$

Fixed Point Iteration

Loop T times s.t. $V_t^\pi = r^\pi + \gamma T^\pi V_{t-1}^\pi$. Computational advantages for sparse solutions.

Policy Iteration

Init. arbitrary (e.g., random) policy π. Compute V^π. Compute greedy policy $\pi_V(x) = \operatorname*{argmax}_a r(x, \pi(x)) + \gamma\sum_{x'} P(x'|x, \pi(x))V(x')$ w.r.t. the previously computed V^π. Set $\pi \leftarrow \pi_V$.

Value Iteration

Init $V_0(x) = \max_a r(x, a)$. For $t = 1$ to ∞: For each x, a, $Q_t(x, a) = r(x, a) + \gamma\sum_{x'} P(x'|x, \pi(x))V_{t-1}(x')$. For each x, $V_t(x) = \max_a Q_t(x, a)$ Break if $\max_x |V_t(x) - V_{t-1}(x)| \leq \epsilon$, otw repeat.

Reinforcement Learning

Model Based RL

MLE

$$\hat{P}(X_{t+1}|X_t, A) = \frac{Count(X_{t+1}, X_t, A)}{Count(X_{t+1}, A)}$$

$$\hat{r} = \frac{1}{N_{x,a}}\sum R_t$$

Rmax Algorithm

Add fairy tale state $x^\star$. Init $r(x, a) = Rmax$, $P(x^\star|x, a) = 1$. Init $\pi|r, P$. Loop: Deploy π. If observed "enough" P / r, recompute π.

Model Free RL

TD-Learning

Guarantees convergence conditional on α_t.

$$\hat{V}^\pi(x) = (1 - \alpha_t)\hat{V}^\pi(x) + \alpha_t(r + \gamma\hat{V}^\pi(x'))$$

SGD on the squared loss

Old value estimates are labels/targets $(r + \gamma V(x'; \theta_{old}) = y)$. Same insight applies for the $Q(x, a)$.

$$l_2(\theta; x, x', r) = \frac{1}{2}(V(x, \theta) - r - \gamma V(x'; \theta_{old}))^2$$

Q-learning

Optimistic initialization = guaranteed convergence. General convergence if $\forall\ (a, x)$ are visited ∞ many times. Otw trade off with epsilon greedy strategy.

$$Q^\star(x, a) = r(x, a) + \gamma\sum_{x'} P(x'|x, a)V^\star(x')$$

$$V^\star(x) = \max_a Q^\star(x, a)$$

$$\begin{aligned} &Q^\star(x, a) \leftarrow (1 - \alpha_t)Q^\star(x, a) \\ &+ \alpha_t(r + \gamma\max_{a'} Q^\star(x', a')) \end{aligned}$$

Unfeasible for continues state spaces because of memory requirement $\forall\ (a, x)$.

Approximating value functions

$\phi(x, a)$ is a set of hand designed features.

$$\begin{aligned} &\hat{Q}(x, a; \theta) = \theta^T\phi(x, a) \\ &l_2(\theta; x, a, x', r) = \frac{1}{2}(Q(x, a, \theta) - r \\ &-\gamma\max_{a'} Q(x', a'; \theta_{old}))^2 \\ &\delta = Q(x, a, \theta) - r - \gamma\max_{a'} Q(x', a'; \theta_{old}) \\ &\theta \leftarrow \theta - \alpha_t\delta\nabla_\theta Q(x, a; \theta) \\ &\theta \leftarrow \theta - \alpha_t\delta\phi(x, a) \\ &L(\theta) = \sum l_2(\theta; x, a, x', r) \end{aligned}$$

Policy search methods

$$\begin{aligned} \pi(x) &= \pi(x, \theta) \\ r(\tau^{(i)}) &= \sum_{t=0}^{T}\gamma^t r_t^{(i)} \\ J(\theta) &\approx \frac{1}{m}\sum_{i=1}^{m} r(\tau^{(i)}) \\ \theta^\star &= \operatorname*{argmax}_\theta J(\theta) \\ \nabla_\theta J(\theta) &= \nabla\mathbb{E}_{\tau\sim\pi_\theta} r(\tau) \\ &= \mathbb{E}_{\tau\sim\pi_\theta}[r(\tau)\nabla\log\pi_\theta(\tau)] \end{aligned}$$

REINFORCE Algorithm

$$\mathbb{E}_{\tau\sim\pi_\theta}[\sum_{t=0}^{T} r(\tau)\nabla\log\pi_\theta(a_t|x_t;\theta)]$$

$$\mathbb{E}_{\tau\sim\pi_\theta}[\sum_{t=0}^{T}(r(\tau) - b(\tau_{0:t-1}))$$
$$\nabla\log\pi_\theta(a_t|x_t;\theta)]$$

$$b(\tau_{0:t-1}) = \sum_{t'=0}^{t-1}\gamma^{t'} r_{t'}$$

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}[\sum_{t=0}^{T}\gamma^t G_t\nabla\log\pi_\theta(a_t|x_t;\theta)]$$

$$G_t = \sum_{t'=t}^{T}\gamma^{t'-t} r_{t'}$$

Initialize policy weights $\pi(a|x;\theta)$. Repeat: Generate an episode. For every t get G_t. Update $\theta \leftarrow \theta + \eta\gamma^t G_t\nabla_\theta\log\pi(A_t|X_t;\theta)$

Policy Gradient Theorem

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}[Q(x, a)\nabla\log\pi_\theta(a|x;\theta)]$$

Online Actor Critic

$$\begin{aligned} &\theta_\pi \leftarrow \theta_\pi + \eta_t Q(x, a; \theta_Q)\nabla\log\pi(a|x;\theta_\pi) \\ &\theta_Q \leftarrow \theta_Q - \eta_t(Q(x, a; \theta_Q) - r \\ &\quad - \gamma Q(x', \pi(x', \theta_\pi); \theta_Q))\nabla Q(a|x;\theta_\pi) \end{aligned}$$