Bayesian Linear Regression

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

Online Bayesian Linear Regression

$$
X^TX=\sum_{i=1}^{t}x_ix_i^T
$$

$$
X^Ty=\sum_{i=1}^{t}y_ix_i
$$

MLE and MAP regression

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

Gaussian Proccesess

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

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

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

Linear Kernel

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

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

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.

$$
\begin{aligned}
&\underset{\theta}{\text{argmax}}\int p(y_{train}\mid f,x_{train},\theta)p(f\mid\theta)df\\
=&\underset{\theta}{\text{argmax}}\int \mathcal{N}(f(x),\sigma_n^2)\mathcal{N}(0,K(\theta))df\\
=&\underset{\theta}{\text{argmax}}\,\mathcal{N}(0,K(\theta)+I\sigma_n^2)\\
=&\underset{\theta}{\text{argmax}}\,p(y_{train}\mid x_{train},\theta)\\
=&\underset{\theta}{\text{argmin}}\ -\log p(y_{train}\mid x_{train},\theta)\\
=&\underset{\theta}{\text{argmin}}\,\frac{1}{2}\big(y(K(\theta)+I\sigma_n^2)^{-1}y\\
+&\log(\det K_y)\big)
\end{aligned}
$$

Laplace Approximation

In the context of Log. Regression.

$$
\begin{aligned}
q(\theta) &= \mathcal{N}(\hat{w},\Lambda^{-1})\\
\hat{w} &= \underset{w}{\text{argmax}}\,p(w\mid y)\\
&= \underset{w}{\text{argmax}}\,\frac{1}{Z}p(w)p(y\mid w)\\
&= \underset{w}{\text{argmin}}\,\frac{1}{2\sigma_p^2}\|w\|_2^2\\
&+\sum_{i=1}^{n}\log(1+e^{-y_iw^Tx_i})\\
\Lambda &= -\nabla\nabla\log p(\hat{w}\mid x,y)\\
&= X\ diag([\pi_i(1-\pi_i)]_i)\ X\\
\pi_i &= \sigma(\hat{w}^Tx_i)
\end{aligned}
$$

Prediction

$$
\begin{aligned}
&p(y^*\mid x^*,X,y)\\
=&\int p(y^*\mid x^*,w)p(w\mid X,y)dw\\
=&\int p(y^*\mid x^*,w)q_\lambda(w)dw\\
=&\int p(y^*\mid f^*)p(f^*\mid w)q_\lambda(w)dwdf^*\\
&q_\lambda(w)\sim N(\mu,\Sigma)\\
&p(f^*\mid w)=x^*\\
&\int p(f^*\mid w)q_\lambda(w)dw\\
=&N(\mu^Tx^*,x^*\Sigma x^*)\\
&p(y^*\mid x^*,X,y)\\
=&\int p(y^*\mid f^*)N(\mu^Tx^*,x^*\Sigma x^*)df^*\\
&p(y^*\mid f^*)=\sigma(y^*f^*)
\end{aligned}
$$

Variational Inference

KL divergence

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

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

$$
\begin{aligned}
&KL(p||q)\\
=&\frac{1}{2}\big(tr(\Sigma_1^{-1}\Sigma_0)+(\mu_1-\mu_0)^T\Sigma_1^{-1}(\mu_1-\mu_0)\\
&-d+ln(\frac{|\Sigma_1|}{|\Sigma_0|})\big)\\
&p=\mathcal{N}(\mu_0,\Sigma_0)\\
&q=\mathcal{N}(\mu_1,\Sigma_1)
\end{aligned}
$$

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 (e.g. f is convex).

Improved Proposals

Metropolis adjusted Langevin Algo (gradient to prefer proposals into regions with higher density), Stochastic Gradient Langevin Dynamics (stochastic gradient), Hamiltonian Monte Carlo (momentum).

Bayesian Neural Networks

MAP estimation with BNN's

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

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

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

MCMC in BNN's

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

Dropout and Probabilistic Ensembles

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

Calibration

Reliability Diagrams

If well calibrated $freq(B_m) = conf(B_m)$ for all bins.

$$\begin{aligned}
&freq(B_m) = \frac{1}{|B_m|}\sum_{i\in B_m} 1(y_i = 1)\\
&conf(B_m) = \frac{1}{|B_m|}\sum_{i\in B_m} \hat{p}_i\\
&ECE = \sum_{m=1}^{M}\frac{|B_m|}{n}|freq(B_m) - conf(B_m)|\\
&MCE = \max_{m\in(1,\dots,M)} |freq(B_m) - conf(B_m)|
\end{aligned}$$

Calibration Methods

Histogram binning: Assign calibrated score to each bin $\hat{q}_i = freq(B_m)$. **Isotonic regression**: Find piecewise constant function f, $\hat{q}_i = f(\hat{p}_i)$ that minimizes the bin-wise squared loss, by adjusting the bins. **Platt scaling**: Learn $a, b \in \mathbb{R}$ that minimize the NLL loss over the validation set when applied to the logits z_i, $\hat{q}_i = \sigma(az_i + b)$. Temperature scaling for multiple classes uses single parameter T s.t. $\hat{q}_i = \max_k \sigma_{softmax}(z_i/T)^{(k)}$

Active Learning

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

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

Uncertainty Sampling

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

$$\begin{aligned}
&I(f(x_T), y_T) \geq \left(1 - \frac{1}{e}\right)\max_{|S|\leq T} I(f(x_S), y_S)\\
&x_{t+1} = \underset{x}{\operatorname{argmax}}\ \mathbb{I}(f; y_x|y_{S_t})\\
&= \underset{x}{\operatorname{argmax}}\ \frac{1}{2}\log\left(1 + \frac{\sigma_t^2(x)}{\sigma_n^2}\right)
\end{aligned}$$

Active Learning for Classification

Uncertainty sampling: $x_{t+1} = \operatorname{argmax}_x\ H(Y|x, X_t, Y_t)$. Even better, use approximate inference to estimate the MI and use it as selection criteria.

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

Bayesian Optimization

Cumulative Regret

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

Upper confidence sampling

$x_{t+1} = \underset{x\in D}{\operatorname{argmax}}\ \mu_t(x) + \beta_t\sigma_t(x)$. Convergence of the cumulative regret as a function of $\gamma_T = \max_{|S|<=T} I(f; y_S)$

Probability of Improvement

$$PI(x) = P(f(x) > f^*) = \Phi\left(\frac{\mu_t(x) - f^*}{\sigma_t(x)}\right)$$

Expected Improvement

$$\begin{aligned}
&x_{t+1} =\\
&\underset{x\in D}{\operatorname{argmax}}\ \mathbb{E}_{f_t(x)\sim\mathcal{N}(\mu_t,\sigma_t)}[max(0, f_t(x) - f^*)]
\end{aligned}$$

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 determinnistic policy π and a given state x.

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

$\forall x$:

$$\begin{aligned}
V^\pi &= r^\pi + \gamma T^\pi V^\pi \\
V^\pi &= (I - \gamma T^\pi)^{-1} r^\pi
\end{aligned}$$

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

Convergence $\pi^\star$ in $O * (n^2 m/(1-\gamma))$ iterations. Start with an 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

Initialize $V_0(x) = \max_a r(x, a)$. 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.

POMDP's

New state has probability $P(X_{t+1} = x'|x_t, a_t)$ and we observe $y_t \sim P(Y_t|X_t = x_t)$.

$$\begin{aligned}
&b_{t+1}(x) \\
&= P(X_{t+1} = x'|y_{t+1}) \\
&= \frac{1}{Z} \sum_{x'} b_t(x) P(X_{t+1} = x'|x', a_t) \\
&P(y_{t+1}|x) \\
r(b_t, a_t) &= \sum_x b_t(x) r(x, a_t)
\end{aligned}$$

Reinforcement Learning

Stationary Value Function: $\forall\ t$, $V_i^\star(x) = V_j^\star(x)$. Stationary Policy: For a given state the optimal action is the same regardless of the time step at which the state is visited).

Model Based RL

MLE

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

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

Rmax Algorithm

Initially: Add fairy tale state $x^\star$. Set $r(x, a) = Rmax$ for all states x and actions a. Set $P(x^\star|x, a) = 1$ for all (x, a). Choose optimal policy for r and P. Loop: Execute policy π. for each visited state action pair update $r(x, a)$. estimate transition probabilities $P(x'|x, a)$. if observed "enough" transitions / rewards. recompute policy π according to current model P and r.

Model Free RL

Estimate the value function directly.

TD-Learning

On-policy method. Estimate $V^\pi(x)$. 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

Off-policy. Estimate the optimal policy via some behavioral policy. Optimistic initialization possible (guaranteed convergence). General convergence guaranteed if $\forall\ (a, x)$ are visited infinitely many times. Otw trade off with epsilon greedy strategy.

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

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

Approximating value functions

Linear function approximation, where $\phi(x, a)$ is a set of hand designed features.

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

To reduce variance keep the target values constant across episodes (e.g. replay buffer or twin network).

$$\begin{aligned}
L(\theta) = \sum_{(x,a,r,x') \in D} &(Q(x, a, \theta) - r \\
&- \gamma \max_{a'} Q(x', a'; \theta_{old}))^2
\end{aligned}$$

Double DQN avoids maximization bias (overconfidence about certain actions given the noise in the observations) by maximizing w.r.t. the current network instead of the old one. Nevertheless the maximization remains intractable for continues action spaces.

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

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

Can use approximations for Q. Parametrized policy (actor) and value function approx (critic). Vanilla policy search methods are slow, actor-critic improves it.

Online Actor Critic

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

Advantage Active Critique

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

Off-Policy Actor Critic

Maximization → training param. policy. Gradients possible for both deterministic and stochastic parametrized policies.

$$\begin{aligned}&\max_a Q(x',a',\theta^{old}) \approx Q(x',\pi(x';\theta_\pi);\theta_Q^{old})\\ &\nabla_\theta\hat{J}_\mu(\theta) = \mathbb{E}_{x\sim\mu}[\nabla_\theta Q(x,\pi(x;\theta);\theta_Q)]\\ &\nabla_{\theta_\pi} Q(x,\pi(x;\theta_\pi);\theta_Q)\\ &= \nabla_a Q(x,a)|_{a=\pi(x;\theta_\pi)}\nabla_{\theta_\pi}\pi(x;\theta_\pi)\end{aligned}$$

Model-based Deep RL

Planning in the known model

$$\begin{aligned}J(a_{t:t+H-1}) \triangleq \sum_{\tau=t}^{t+H-1}&\gamma^{\tau-t} r_\tau(x_\tau(a_{t:\tau-1}),a_\tau)\\ &+ \gamma^H V(x_{t+H})\end{aligned}$$

Stochastic transition setting

Choose the sequence of actions that maximizes the expectation over the randomness in the model, but also re plan after each action. Expectation estimated via MC sampling.

Offline training

Optimizing a policy (deterministic or stochastic) that is fast to evaluate online. Look-ahead helps policies improve more rapidly, by anticipating consequence down the road.

$$\begin{aligned}J(\theta) = \mathbb{E}_{x\sim\mu}[&\sum_{\tau=0:H-1}\gamma^\tau r_\tau\\ &+ \gamma^H Q(x_H,\pi(x_H;\theta);\theta_Q)|\theta]\end{aligned}$$

Unknown Dynamics

$$\begin{aligned}&\hat{J}_H(a_{t:t+H-1})\\ &= \frac{1}{m}\sum_{i=1}^{m}\sum_{\tau=t}^{t+H-1}\gamma^{\tau-t} r_\tau\\ &(x_\tau(a_{t:\tau-1},\epsilon_{t:\tau-1}^{(i)},f^{(i)}),a_\tau) + \gamma^H V(x_{t+H})\end{aligned}$$

Greedy, Thompson and Optimistic Exploration

Greedy

$D = []$; prior $P(f) = P(f|[])$, then iterate the following: Plan new policy $\max_\pi \mathbb{E}_{f\sim P(\cdot|D)} J(\pi,f)$. Roll out π and add collected data to D. Update posterior $P(f|D)$.

Thompson Sampling

Only difference is we sample the model $f \sim P(\cdot,D)$.

Optimistic Exploration

Main diff. is we plan new policy $\max_\pi \max_{f\in M(D)} \mathbb{E}_{f\sim P(\cdot|D)} J(\pi,f)$.