Multivariate Gaussian

$\frac{1}{(2\pi)^{n/2}\sqrt{|\Sigma|}}\exp(-\frac{1}{2}(y-\mu)^T\Sigma^{-1}(y-\mu)$

$p(X_A|X_B=x_B)=\mathbb{N}(\mu_{A|B},\Sigma_{A|B})$

$\mu_{A|B}=\mu_A+\Sigma_{AB}\Sigma_{BB}^{-1}(X_B-y_B)$

$\Sigma_{A|B}=\Sigma_{AA}-\Sigma_{AB}\Sigma_{BB}^{-1}\Sigma_{BA}$

$f*=w^Tx^*, y=w^Tx+\varepsilon$

$P(f*|X,y,x^*)=\mathbb{N}(mu^Tx^*, x^{*T}\Sigma x^*)$

$P(y|X,y,x^*)=\mathbb{N}(mu^Tx^*, x^{*T}\Sigma x^*+\sigma_n^2)$

MAP,MLE

$w_{MLE}=(X^TX)^{-1}X^Ty$

$P(w|X,y)=P(w|\mu_w,\Sigma_w)$

$\Sigma_w=(\sigma_w^{-2}\mathbb{I}+\sigma_y^{-2}X^TX)^{-1}$

$\mu=\sigma_y^{-2}\Sigma_wX^Ty$

$w_{MAP}=\sigma_y^{-2}\Sigma_wX^Ty$

$\lambda=\sigma_y^2/\sigma_w^2$, $\sigma_y^2=\frac{1}{n}\sum_{i\le n}(y_i-w^Tx_i)^2$, $\sigma_w^2=\frac{\sigma_y^2}{\lambda}$

Bayesian Filtering

Conditioning

$P(X_t|y_{1:t})=\frac{1}{Z}P(X_t|y_{1:t+1})P(y_t|X_t)$

Prediction

$P(X_{t+1}|y_{1:t})=\int P(X_{t+1}|x_t)P(x_t|t_{1:t})$

$\mu_{t+1}=\frac{\sigma_y^2\mu_t+(\sigma_t^2+\sigma_x^2)y_{t+1}}{\sigma_t^2+\sigma_x^2+\sigma_y^2}$

$\sigma_{t+1}^2=\frac{(\sigma_t^2+\sigma_x^2)\sigma_y^2}{\sigma_t^2+\sigma_x^2+\sigma_y^2}$

Kalman FIlters

$X_{t+1}=FX_t+\varepsilon_t, Y_t=HX_t+\eta t$

TM: $P(x_{t+1}|x_t)=N(x_{t+1};Fx_t,\Sigma_x)$

SM: $P(y_t|x_t)=N(y_t;Hx_t,\Sigma_y)$

$\mu_{t+1}=F\mu_t=K_{t+1}(y_{t+1}-HF\mu_t)$

$\Sigma_{t+1}=(I-K_{t+1}H)(F\Sigma_tF^T+\Sigma_x)$

$K_{t+1}=(F\Sigma_tF^T+\Sigma_x)H^T(H(F\Sigma_tF^T+\Sigma_x)H^T+\Sigma_y)^{-1}$

Gaussian Processes

Optimizing Kernel Params: Max Marginal Likelihood: $log(p(y|X,\theta))=-\frac{1}{2}y^TK_y^{-1}y-\frac{1}{2}log|K_y|-\frac{n}{2}log(2\pi)$

Kernels

A valid kernel function must be:

-positive semi-definite

-symmetric

linear $k(x,x')=xx'^T$

Squared Exponential:

$k(x,x')=\exp(-||x-x'||_2^2/h^2)$

Exponential:

$k(x,x')=\exp(-||x-x'||_2/h^2)$

Matern:

$k(x,x')=\frac{2^{1-\nu}}{\Gamma(\nu)}(\frac{\sqrt{s\nu}||x-x'||_2}{\rho})^\nu K_\nu(\frac{\sqrt{2\nu}||x-x'||}{\rho})$

Stationary Kernel:

$k(x,x')=k(x-x')$

Isotropic Kernel

$k(x,x')=k(||x-x'||_2)$

Fast GP Methods

Shift Invariant K: $k(x,y)=k(x-y)$

A SI Kernel is positive semi-definite only if p(w) is nonegative

$k(x-y)=\int_{\mathbb{R}^d}p(w)e^{jw^T(x-y)}dw=\mathbb{E}_{w,b}[z_{w,b}(x)z_{w,b}(y)]=\frac{1}{m}\sum_{i\le n}z_{w^i,b^i}(x)z_{w^i,b^i}(y)=\phi(x)^T\phi(y)$

Inducing points: $p(f*,f)\sim q(f*,f)=\int q(f*|u)q(f|u)p(u)du$

$p(f,f)=N(K_{f,u}K_{u,u}^{-1}u,K_{f,f}-Q_{f,f})$

$p(f*,f)=N(K_{f*,u}K_{u,u}^{-1}u,K_{f*,f*}-Q_{f*,f*})$ where $Q_{A,B}=K_{A,u}K_{u,u}^{-1}K_{u,A}$

SoR Approx:

$p(f,f)=N(K_{f,u}K_{u,u}^{-1}u,0)$

FITC Approx:

$p(f,f)=N(K_{f,u}K_{u,u}^{-1}u,diag(K_{f,f}-Q_{f,f})$

Approximate Inference

Laplace Approximation

$q(\theta)=N(\theta;\hat{\theta},\Lambda^{-1})$

$\hat{\theta}=argmax_\theta p(\theta|y)$

GD=$w\leftarrow w(1-2\lambda\eta_t)+\eta_tyxP(Y=-y|w,x)$

$\Lambda=-\nabla\nabla log(p(\hat{\theta}|y))=\sum_{i\le n}x_ix_i^T\pi_i(1-\pi_i)$, $\pi=\sigma(\hat{w}^Tx_i)$

Var. Inf.=KL Divergence

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

$p=N(\mu_0,\Sigma_0,q=N(\mu_1,\Sigma_1)$

$KL(p||q)=\frac{1}{2}(tr(\Sigma_1^{-1}\Sigma_0)+(\mu_1-\mu_0)^T\Sigma^{-1}(\mu_1-\mu_0)-d+ln(\frac{|\Sigma_1|}{|\Sigma_0|}))$

Entropy

$H(q)=-\int q(\theta)log(q(\theta)d\theta$

Minimizing KL

$argmin_qKL(q||p)=argmax_q\mathbb{E}_{\theta\sim q(\theta)}[logp(y|\theta)]-KL(q||p(.))=argmax_q\mathbb{E}_{\theta\sim q(\theta)}[logp(y,\theta)]+H(q)$

Maximizing ELBO

$log(p(y))\ge\mathbb{E}[logp(y|\theta)]-KL(q||p(.))$

Gradient ELBO

$\nabla_{C,\mu}\mathbb{E}_{\varepsilon\sim N(0,I)}[logp(y|C\varepsilon+\mu)]-\nabla_{C,\mu}KL(q_{C,\mu}||p(.))$

Markov Chain Monte Carlo

Law of Large Numbers

$E_P[f(X)]=\lim_{N\to\infty}\frac{1}{N}\sum_{i\le N}f(x_i)$

Hoeffding's Inequality:

f bounded in [0,C]

$P(|\mathbb{E}\mathbb{P}[f(x)]-\frac{1}{N}\sum_{i\le N}f(x_i)|>\varepsilon)\le 2\exp(-2N\varepsilon^2/C)$

Stationary Distribution

$\lim_{N\to\infty}P(X_N=x)=\pi(x)$

Detailed Balance Equation

$Q(x)P(x'|x)=Q(x')P(x|x')$

Designing Markov Chain

Proposal Distribution=$R(X'|X)$

$X_t=x,x'\sim R(X'|X=x)$

$\alpha=min\frac{Q(x')R(x|x')}{Q(x)R(x'|x)}\to X_{t+1}=x'$

Metropolis, Hastings: The stationary distribution is $Z^{-1}Q(x)$

Gibbs Sampling

$x_0\leftarrow x0$

for $t=1\to\infty$

$i\sim(1..n)/B$

$x_i\sim P(X_{i+1}|x_{-i})$

Variance of Gibbs

$x_0\leftarrow x0$

for $t=1\to\infty$

$x^{(t)}=x^{(t-1)}$

$for i=1\to\infty$

$x_i\sim P(X_{i+1}|x_{-i})$

Ergodic Thm (Special Case): suppose $X_2,..X_n$ is an ergodic Markov chain over a finite state space D with stationary distribution π and let f be a function in D

$\lim_{N\to\infty}\frac{1}{N}f(x_i)=\sum_{x\in D}\pi(x)f(x)$

Markov Processes continuous case

$p(x)=\frac{1}{Z}\exp(-f(x))$

$\alpha=min\frac{R(x'|x)}{R(x|x')}\exp(f(x)-f(x'))\to X_{t+1}=x'$

Metropolis Adjusted Langevin Algorithm

$R(x'|x)=\mathbb{N}(x';x+\tau\nabla f(x);2\tau I)$

Stochastic Gradient Langevin Dynamics

$\theta_0=\theta0$

$For t=0,1,..$

$\varepsilon_t\sim\mathbb{N}(0,2\eta_tI)$

$\theta_{t+1}=\theta_t-\eta_t(\nabla\log p(\theta_t)+\frac{n}{m}\sum_{j\le m}\nabla\log p(y_{i,j}|\theta_t,x_{i,j}))+\varepsilon_t$

Bayesian Neural Networks

Heteroscedastic noise

$p(y|x,\theta)=\mathbb{N}(y;f_1(x,\theta),exp(f_2(x,\theta))$

Log Likelihood

$argmin_\theta\lambda||\theta||_2^2+\sum_{i\le n}||y_i-\mu(x_i,\theta)||^2+\frac{1}{2}\log\sigma(x_i;\theta)^2$

Gradient

$\nabla_\lambda L(\lambda)=\nabla_{C,\mu}\frac{n}{m}\sum_{j\le m}\log p(y_{i,j}|C\varepsilon^{(j)}+\mu,x_{i,j})-\nabla_{C,\mu}KL(q_{C,\mu}||p(.))$

Prediction Mean and Variance

$P(y*|X,y,x*)\simeq\frac{1}{m}\sum_{j\le m}\mathbb{N}(y*;\mu(x*,\theta^{(j)}),\sigma^2(x*,\theta^{(j)}))$

Mean=$\frac{1}{m}\sum_{j\le m}\mu(x*,\theta^{(j)})$

Var=$\mathbb{E}[Var[y*|x*,\theta]]+Var\mathbb{E}[y*|x*,\theta]]\simeq\frac{1}{m}\sum_{j\le m}\sigma^2(x*,\theta^{(j)})+\frac{1}{m}\sum_{j\le m}(\mu(x*,\theta^{(j)}))-\overline{\mu}(x*))^2$

Drop Out as VI

$q(\theta|\lambda)=\prod_j(q_j(\theta_j|\lambda_j))$

$q_j(\theta_j|\lambda_j)=p\delta_0(\theta_j)+(1-p)\delta_{\lambda_j}(\theta_j)$

Probabilistic Ensambles

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

Active Learning

Mutual Information: Given random samples X and y I(X;Y) quantifies how much observing Y reduces the uncertainity of X

$I(X;Y)=\sum_{y\in Y}\sum_{X\in X}P(x,y)\log(\frac{P(x,y)}{P(x)P(y)})$

$I(X;Y)=H(Y)-H(\varepsilon)=\frac{1}{2}\ln((2\pi e)^d|\Sigma-\sigma^2\mathbb{I}|)-\frac{1}{2}\ln((2\pi e)^d|\sigma^2\mathbb{I}|)=\frac{1}{2}\ln(\sigma^{-2}\Sigma-\mathbb{I})$

$F(S)=H(f)-H(f|y_s)=I(f;y_s)=\frac{1}{2}\ln|\mathbb{I}+\sigma^{-2}\Sigma|$

Optimizing MI

$x_{t+1}=argmax_xF(S_tUx)=argmax_{x\in D}\sigma_{x|St}^2$

Submodularity of MI

$\forall x\in D\forall A\subseteq B\subseteq D:F(A\bigcup x)-F(A)\ge F(B\bigcup x)-F(B)$

Greedy Constant Factor

$F(S_t)=(1-\frac{1}{e})max_{S\subseteq D,|S|\le T}F(S)$

Natural Generalization MI optimization

$x_{t+1}\in argmax_x\frac{\sigma_f^2}{\sigma_n^2}$

Learning to Optimize

$f\in Ff:D\to\mathbb{R},y_t=f(x_t)+\varepsilon_t$

$CR:R_T=\sum_{i\le T}(max_xf(x)-f(x_t))$

Sublinearity: $R_T/T\to 0$

implies $max_tf(x_t)\to f(x*)$

Upper Confidence Sampling

$x_t=argmax_{x\in D}\mu_{t-1}(x)+\beta_t\sigma t-1(x)$

Regret of GP-UCB: assuming β_t correct

$\frac{1}{T}\sum_{t\le T}[f(x*)-f(x_t)]=O^*(\sqrt{\frac{\gamma T}{T}})$ where $\gamma T=max_{|S|\le T}I(f;ys)$

Frequentist approach:

$\frac{1}{T}\sum_{t\le T}[f(x*)-f(x_t)]=O^*(\sqrt{\frac{\beta_T\gamma T}{T}})$

Expected Improvement

$x_t=argmax_{x\in D}EI(x)$

EI=$E[(t-y)_+]=\int_{-\infty}^{\infty}(t-y)_+p(y|x)dy$

$y_t=f(x_t)+\varepsilon_t$

Bayesian update for μ_y,σ_t Thompson Sampling

$\tilde{f}\sim P(f|x_{1:t},y_{1:t})$

$x_{t+1}=argmax_{x\in D}\tilde{f}(x)$

Markov Decision Processes

A set of States 1,..n

A set of actions 1,..,m

Transition Probabilities

$P(X_{t+1}=x'|X_t=x)=P(x'|x,\pi(x))$

$P(X_{t+1}|x'|X_t=x)=\sum_a\pi(a|x)P(x'|x,a)$

Reward Function

EV: $J(\pi)=E[r(X_0,\pi(X_0))+\gamma r(X_i,\pi(X_1))+\gamma^2r(X_2+\pi(X_2))+...]$

$V^\pi(x)=J(\pi|X_0=x)=\mathbb{E}_{x=1:\infty}[\sum_{t\le\infty}\gamma^tr(X_t,\pi(X_t))|X_0=x]=r(x,\pi(x))+\gamma\sum_{x'}P(x'|x,\pi(x))V^\pi(x')$

Solving for a policy

$V^\pi=(I-\gamma T^\pi)^{-1}r^\pi$

Fixed point iteration

$V_t^\pi=r^\pi+\gamma T^\pi V_{t-1}^\pi$

$t\ge\frac{ln\frac{||V_0^\pi-V^\pi||_\infty}{\varepsilon}}{ln\frac{1}{\gamma}}$

Greedy Policy

$\pi(x)_V=argmax_ar(x,a)+\gamma\sum_{x'}P(x'|x,a)V(x')$

Theorem(Bellman): A policy acts greedly w.r.t its induced value function

$V^*(x)=max_a[r(x,a)+\gamma\sum_{x'}P(x'|x,a)V^*(x')]$

MDP

observation: $P(Y_t|X_t)$

state update: $P(X_{t+1}|X_t,A_t)$

typically extremely intractable $\to$

PODPS=$\mathbb{B}=\nabla(1,...,N)=b:1,...,n\to[0,1],\sum_xb(x)=1$

observation: $P(Y_{t+1}=Y|b_t(x),A_t)=\sum_{x,x'}b_t(x)P(x'|x,a_t)P(y|x')$

state update: $b_{t+1}=\frac{1}{Z}\sum_xb_t(x)P(X_{t+1}=x'|X_t=x,a_t)P(y_{t+1}|x')$

reward: $r(b_t,a_t)=\sum_xb_t(x)r(x,a_t)$

Reinforcement Learning

Model Based

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

$r(x,a)\simeq\frac{1}{N_{x,a}}\sum_{t:X_t=x,A_t=a}R_t$

$\varepsilon greedy$

$\varepsilon_t\sim$ pick random action

$1-\varepsilon_t\sim$ pick best action

Converges if ε_t satisfies Monroe's Conditions: $\sum_t\varepsilon_t=\infty,\sum_t\varepsilon_t\le\infty$

The R_{max} Algorithm

$r(s,a)=R_{max}$ for unknown states

$P(x^*|x,a)=1$ add a fairy tale state

- For each visited action pair x,a update r(x,a)
- Estimate Transition Probabilities P(x'|x,a)
- If observed ënough" compute policy π

ENOUGH? Hoeffding's Bound

$P(|\mu-\frac{1}{n}\sum_{i\le n}Z_i|\ge\varepsilon)\le 2\exp(-2n\varepsilon^2/C^2)$, $n\in O(\frac{R_{max}^2}{\varepsilon^2}\log(\frac{1}{\delta}))$

- Every T time steps w.h.p R_{max} either obtains near optimal reward, or visits at least one unknown state-action pair
- Performance of R_{max}: With probability $1-\delta$, R_{max} will reach an ε-optimal policy in a number of steps that is polynomial in |X|,|A|,T, $\frac{1}{\varepsilon}$ and $\log\frac{1}{\delta}$

Issues with MB RL
1) Memory required: store all $P(x'|x,a)$ and $r(x,a)$
2) Computation Time: Solve Value Iteration or Policy iteration

Model Free

Temporal Difference Learning:
$V(x) \leftarrow (1-\alpha_t)V(x) + \alpha_t(r+\gamma V(x'))$
The learning rate has to satisfy Monroe conditions
Q function:
$V^*(x) = max_a Q^*(x,a)$,where
$Q^*(x,a) = r(x,a)+\sum_{x'} P(x'|x,a)V^*(x')$

Q-learning
$Q(x,a) \leftarrow (1-\alpha_t)Q(x,a)+\alpha_t(r+\gamma \max_{a'} Q(x',a'))$,where
$V(x') = \max_{a'} Q(x',a')$
The learning rate has to satisfy Monroe conditions
Optimistic Q-learning:
$Q(x,a) = \frac{R_{max}}{1-\gamma}\prod_{t\le T_{init}}(1-\alpha_t)^{-1}$
- With Probability $1-\delta$, optimistic Q learning obtains an ε-optimal policy after a number of time steps polynomial in |X|,|A|, $\frac{1}{\varepsilon}$ and $\log\frac{1}{\delta}$

Reinforcement Learning through Functional Approximation

$V^\pi(x) = r(x,\pi(x)) + \gamma\sum_{x'}P(x'|x,\pi(x))V^\pi(x')$
$Q^\pi(x,a) = r(x,a)+\gamma\sum_{x'}P(x'|x,a)V^\pi(x')$
$V^*(x) = max_a[r(x,a)+\gamma\sum_{x'}P(x'|x,a)V^*(x')]$
$Q^*(x) = r(x,a)+\gamma\sum_{x'}P(x'|x,a)V^*(x')$
TD-Learning as Stochastic Gradient Descent/Bootstapping
$l_2(\theta;x,x',r) = \frac{1}{2}(V(x;\theta)-r-\gamma V(x';\theta_{old}))^2$
$\nabla l_2 = (V(x;\theta)-r-\gamma V(x';\theta_{old})) := \delta$
$V \leftarrow V - \alpha_t\delta$
Parametric Function Approximation:
$Q(x,a;\theta) = \theta^T\phi(x,a)$
$\theta \leftarrow \theta - \alpha_t + \nabla_\theta l_2(\theta;x,a,r,x')$
$\theta \leftarrow \theta - \alpha_t\delta\phi(x,a)$
generally:
$\delta := Q(x,a;\theta) - r - \gamma max_{a'}Q(x',a;\theta)$
$\theta \leftarrow \theta - \alpha_t\delta\nabla_\theta Q(x,a;\theta)$

Neural Fitted Q-Iteration/DQN
$L(\theta) = \sum_{(x,a,r,x')\in D}(r+\gamma max_{a'}Q(x',a';\theta)-Q(x,a;\theta))^2$
Double DQN: solve maximization bias
$L(\theta) = \sum_{(x,a,r,x')\in D}(r+\gamma max_{a'}Q(x',a';\theta^{old})-Q(x,a;\theta))^2$

Parametrized Policy Gradients

Parametrized policy
$\pi(x)=\pi(x;\theta)$
Find policy approximation through Monte Carlo forward Sampling:
$J(\theta) = \frac{1}{m}\sum_{i\le m} r(\tau_i)$
$r(\tau_i) = \sum_{i\le T}\gamma_t r_t^{(i)}$
Policy Gradients:
$J(\theta) = \mathbb{E}_{x_{0:T},a_{0:T}\sim\pi\theta}\sum_{t\le T}\gamma^t(x_t,a_t)$
$= \mathbb{E}_{\tau\sim\pi_\theta} r(\tau)$
$\nabla J(\theta) = \nabla\mathbb{E}_{\sim\pi_\theta}[r(\tau)] = \mathbb{E}_{\tau\sim\pi_\theta}[r(\tau)\nabla\log\pi_\theta(\tau)]$
Exploiting MDP Structure:
$\pi_\theta(\tau) = p(x_0)\prod_{t\le T}\pi(a_t|x_t;\theta)p(x_{t+1}|x_t,a_t)$
$\mathbb{E}_{\tau\sim\pi_\theta}[r(\tau)\nabla_\theta\log\pi_\theta(\tau)] = \mathbb{E}_{\tau\sim\pi_\theta}[r(\tau)\sum_{t\le T}\nabla_\theta\log\pi(a_t|x_t;\theta)]$
Lower Variance by introducing baseline
$\mathbb{E}_{\tau\sim\pi_\theta}[r(\tau)\sum_{t\le T}\nabla_\theta\log\pi(a_t|x_t;\theta)] = \mathbb{E}_{\tau\sim\pi_\theta}[(r(\tau)-b)\sum_{t\le T}\nabla_\theta\log\pi(a_t|x_t;\theta)]$
$\mathbb{E}_{\tau\sim\pi_\theta}[r(\tau)\sum_{t\le T}\nabla_\theta\log\pi(a_t|x_t;\theta)] = \mathbb{E}_{\tau\sim\pi_\theta}[(r(\tau)-b(\tau_{0:t-1}))\sum_{t\le T}\nabla_\theta\log\pi(a_t|x_t;\theta)]$
$b(\tau_{0:t-1}) = \sum_{t'\le t-1}\gamma^{t'}r_{t'} \rightarrow$
$\nabla J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}[\sum_{t\le T}\gamma^t G_t\nabla\log\pi(a_t|x_t;\theta)]$
$G_t = \sum_{t\le T}\gamma^{t'-t}r_{t'}$
Reinforce
Input $\pi(a|x;\theta)$
Initialize weights θ Generate an episode: $(X_0,A_0,R_0,...)$
$for t = 0...T$
$\theta = \theta+\eta\gamma^t G_t\nabla_\theta\log\pi(A_t|X_t;\theta)$
Further Variance Reduction:
$\nabla J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}[\sum_{t\le T}\gamma^t G_t\nabla\log\pi(a_t|x_t;\theta)]$
$= \nabla J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}[\sum_{t\le T}\gamma^t(G_t - b_t(x_t))\nabla\log\pi(a_t|x_t;\theta)]$
$b_t := b_t = \frac{1}{T}\sum_{t\le T-1}G_t$

Deep RL and Actor Critic Methods

Advantage Function
$A^\pi(x,a) = Q^\pi(x,a)-V^\pi(x) = Q^\pi(x,a)-\mathbb{E}_{a'\sim\pi(x)}Q^\pi(x,a')$
$\forall\pi\forall x: max_a A^\pi(x,a)\ge 0$
$\pi*$ is optimal $\forall x,a, A^*(x,a)\le 0$
$\pi_G(x) = argmax_a Q^\pi(x,a) = argmax_a A^\pi(x,a)\rightarrow$ for Greedy Policy

On Policy Actor Critic

$\nabla J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}[\sum_t\gamma^t Q(x_t,a_t;\theta_Q)\nabla\log\pi(a_t|x_t;\theta)]$
$= \int_x p^\theta(x)\mathbb{E}_{a\sim\pi_\theta(x)}[Q(x,a,\theta_Q)\nabla\log\pi(a_t|x_t;\theta)]dx$
where $p^\theta(x) = \sum_t\gamma^t p_\theta(X_t=x)$
$:= \mathbb{E}_{(x,a)\sim\pi_\theta(x)}[Q(x,a;\theta_Q)\nabla\log\pi(a_t|x_t;\theta)]$
Actor Critic update:
$\theta_\pi \leftarrow \theta_\pi+\eta_t Q(x,a;\theta_Q)\nabla\log(a_t|x_t;\theta_\pi)$
$\theta_Q \leftarrow \theta_Q - \eta_t(Q(x,a;\theta_Q)-r-\gamma Q(x',\pi(x,\theta_\pi);\theta_Q))\nabla Q(x,a;\theta_Q)$
Variance reduction via Baselines: Advantage Function Estimate → A2C Algorithm
$\theta_\pi \leftarrow \theta_\pi+\eta_t(Q(x,a;\theta_Q)-V(x;\theta_V))\nabla\log(a_t|x_t;\theta_\pi)$
TRPO
$\theta_{k+1} = agmx_\theta\hat{J}(\theta_k,\theta) s.t. KL(p||q)\le\delta$
$\hat{J}(\theta_k,\theta) = \mathbb{E}_{x,a\sim\pi_\theta}[\frac{\pi(a|x;\theta)}{\pi(a|x;\theta_k)}A^{\pi_{\theta_k}}(x,a)]$
PPO
Heuristic Variant of TRPO (uses a certain clipped surrogate), effective and wildly used in practice

Off-Policy Actor Critic

$L(\theta_Q) = \sum_{(x,a,r,x')}(r+\gamma Q(x',\pi(x';\theta_\pi);\theta_Q^{old})-Q(x,a,\theta_Q))^2$
Allowing rich enough
$\theta_\pi^*\in argmax_\theta\mathbb{E}_{x\,\sin\mu}[Q(x,\pi(x;\theta);\theta_Q)]$
$\mu(x)>0$ means ëxplore all states"
Unbiased gradient estimate by sampling from $x\sim\mu$
$\nabla_\theta Q(x,\pi(x;\theta);\theta_Q) = \nabla_a Q(x,a)|_{a=\pi(x;\theta)}\nabla_\theta\pi(x;\theta)$
DDPG: Deep Deterministic Policy Gradient Init θ_{Q,θ_π}, replay buffer D=; $\theta_Q^{old}=\theta_Q,\theta_\pi^{old}=\theta_\pi$
repeat:observe x→action $a(x,\theta_\pi)+\varepsilon$
Execute a, observe reward and next state store $(x,a,r,x')\in D$"
if update:
$\theta_Q \leftarrow \theta_Q - \eta\nabla\frac{1}{|B|}\sum_{(x,a,r,x')\in B}(Q(x,a,\theta_Q)-r+\gamma Q(x',\pi(x;\theta_\pi^{old},\theta_Q^{old}))$
$\theta_\pi = \theta_\pi+\eta\nabla\frac{1}{|B|}\sum_{(x,a,r,x')\in B}Q(x,\pi(x;\theta_\pi);\theta_Q)$
$\theta_Q^{old}\leftarrow(1-\rho)\theta_Q^{old}+\rho\theta_Q$
$\theta_\pi^{old}\leftarrow(1-\rho)\theta_\pi^{old}+\rho\theta_\pi$
Twind Delayed DDPG→(TD3)
Extends DDPG by using two critics networks, and evaluating the advantage with the smaller one. (addresses maximization bias)

Randomized Policies:
$\nabla_{\theta_\pi}\mathbb{E}_{\alpha\sim\pi_{\theta_\pi}}Q(x,a;\theta_Q) = \mathbb{E}_\varepsilon\nabla_{\theta_\pi}Q(x,\psi(x;\theta_\pi\varepsilon);\theta_Q) = \mathbb{E}_\varepsilon[\nabla_a Q(x,a;\theta_Q)|_{a=\psi(x;\theta_\pi,\varepsilon)}\nabla_{\theta_\pi}\psi(x;\theta_\pi,\varepsilon]$
Entropy Regularization:
$J_\lambda = J(\theta)+\lambda H(\pi_\theta) = \mathbb{E}_{(x,a)\sim\pi_\theta}[r(s,a)+\lambda H(\pi(.|x))]$
By using reparametrization gradients → SAC (Soft Actor Critic)

Policy Gradient Algorithms on v.s off policy

ON-POLICY

Reinforce
Actor Critic Methods (A2C,A3C)
TRPO,PPO

OFF-POLICY

DDPG
TD3
SAC

Model-Based Deep RL

Knowing a model can help reduce the sample complexity
Planning with known deterministic model
$max_{a_{0:\infty}}\sum_t\gamma^t r(x_t,a_t)$ s.t.$x_{t+1}=f(x_t,a_t)$
Plan Over a finite horizon:
$max_{a_{t:t+H-1}}\sum_{\tau=t:t+H-1}\gamma^{\tau-t}r_\tau(x_\tau,a_\tau)$,
s.t.$x_{\tau+1}=f(x_\tau,a_\tau)$
$x_\tau := x_\tau(a_{t:\tau-1}) := f(f(...(f(x_t,a_t),a_{t+1}),...),a_{\tau-1})$
$J_H(a_{t:\tau-1}) = \sum_{\tau=t:t+H-1}\gamma^{\tau-t}r_\tau(x_\tau(a_{t:\tau-1}),a_\tau)$
Optimization: Random Shooting Samples
Randomly generate m sets of samples $a_{t:\tau-1}^{(i)}$ that optimizes
$i^* = argmax_{i\in 1...m}J_H(a_{t:\tau-1}^{(i)})$
Using Value Estimate:
$J_H(a_{t:\tau-1}) = \sum_{\tau=t:t+H-1}\gamma^{\tau-t}r_\tau(x_\tau(a_{t:\tau-1}),a_\tau)+\gamma^H V(x_{t+H})$
Supposing H=1
$J(a_t) = r(x_t,a_t)+\gamma V(x_{t+1}(a_t)) = Q(x_t,a_t)$
$agmx_a J_\pi(a) = agmx_a Q(x_t,a) = \pi_G^V(x_t)$
Planning for randomized policies
$max_{a_{t:t+H-1}}\mathbb{E}_{x_{t+1:t+H}}[\sum_{\tau=t:t+H-1}\gamma^{\tau-t}r_t+\gamma^H V(x_{t+H})|a_{t:t+H-1}]$
Optimizing Expected Performance
$x_{t+1} = f(x_t-a_t+\varepsilon_t)$
Obtain Unbiased Estimates of J_H
$\hat{J}(a_{t:t+H-1}) = \frac{1}{m}\sum_{i\le m}\sum_{\tau=t:t+H-1}\gamma^{\tau-t}r_\tau(x_\tau(a_{t:\tau-1},\varepsilon_{t:\tau-1}^{(i)}),a_\tau)+\gamma^H V(x_{t+H})$
Using Parametrized Policies:
$J(\theta) = \mathbb{E}_{x_0\sim\mu}[\sum_{\tau=0:H-1}\gamma^\tau r_\tau+\gamma^H Q(x_H,\pi(x_H,\theta))|\theta]$
For H=0 it's identical to the DDPG objective

$J(\theta) = \mathbb{E}_{x_0\sim\mu}[Q(x_0,\pi(x_0,\theta))]$
Unknown Dynamics
- Start with policy π
- Iterate for several episodes
a) Roll out policy to collect data
b) Learn model for f,r and Q
d) Plan new policy based on estimated models

PETS Algorithm:
- ensamble of NNs
- Trajectory Sampling to evaluate performance
- MPC used for planning

Thompson Sampling
- Start with empty D=, Prior P(f)=P(f|)
iterate
$max_\pi J(\pi,f)$
- Roll-out Policy
- Update posterior

Optimistic exploration:
- Start with empty D=, Prior P(f)=P(f|)
iterate
$max_\pi max_{f\in M(D)}J(\pi,f)$
- Roll-out Policy
- Update posterior