1 Probability Calculus

$P(X|Y)=\frac{P(X)P(Y|X)}{\sum_x P(X)P(Y|X)}=\frac{Prior\times Likelihood}{Evidence}$ $P(X)=\sum_Z P(X,Z)$
$P(X,Y,Z)=P(X)P(Y|X)P(Z|X,Y)$ $P(X,Y|Z)=P(X|Z)P(Y|Z)$
$var(Y)=E_x[var(Y|X)]+var_x(E(Y|X))$ $P(X|Y,Z)=P(X|Z)$
$var(X)=cov(X,X)$ $var(f\text{-}g)=var(f)+var(g)\text{-}2cov(f,g)$
$cov(X,Y)=E[(X\text{-}E[X])(Y\text{-}E[Y])]$ $\mathcal{N}(\mu,\sigma)=\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$
$\mathcal{N}(\mu,\Sigma)=(2\pi)^{-\frac{n}{2}}|(\Sigma)|^{-\frac{1}{2}}e^{-\frac{1}{2}(\mathbf{x}-\mu)^\top\Sigma^{-1}(\mathbf{x}-\mu)}$ (mult+sums)
$p(X_A|X_B)=\mathcal{N}(\mu_A+\Sigma_{AB}\Sigma_{BB}^{-1}(x_B\text{-}\mu_B),\ \Sigma_{A|B}=\Sigma_{AA}\text{-}\Sigma_{AB}\Sigma_{BB}^{-1}\Sigma_{BA})$

2 Linear Conditioning (e.g $y=Ax+b+\varepsilon\sim\mathcal{N}(0,\Sigma_y)$)

$p(x)=\mathcal{N}(\mu,\Sigma_x)$, $p(y|x)=\mathcal{N}(Ax+b,\Sigma_{y|x})$
$p(y)=\mathcal{N}(y;A\mu+b,\Sigma_{y|x}+A\Sigma_x A^T)$

3 Bayesian Learning (Get posterior of θ and predict)

BLR and RR: $y=\mathbf{w}^T\mathbf{x}+\varepsilon$, $\varepsilon\sim\mathcal{N}(0,\sigma_n^2 I)$
$\min_\mathbf{w}\sum_i^n\left(y_i-\mathbf{w}^T\mathbf{x}_i\right)^2+\lambda\|\mathbf{w}\|_2^2$, $\hat{\mathbf{w}}=\left(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{y}$

3.1 Bayesian Regression

3.1.1 MAP $\equiv$ RR

$\arg\max_\mathbf{w} P(\mathbf{w})\prod_i P(y_i\mid\mathbf{x}_i,\mathbf{w})$, $\lambda=\sigma_n^2/\sigma_p^2$
σ_p^2 stands for prior (acts as Regulizer)

3.1.2 Posterior $p(\mathbf{w}\mid\mathbf{X},\mathbf{y})=\mathcal{N}(\mathbf{w};\bar{\mu},\bar{\Sigma})$

$p(\mathbf{w})=\mathcal{N}(0,\sigma_p^2\mathbf{I})$, $p(y\mid\mathbf{x},\mathbf{w},\sigma_n)=\mathcal{N}\left(y;\mathbf{w}^T\mathbf{x},\sigma_n^2\right)$
$\bar{\mu}=\left(\mathbf{X}^T\mathbf{X}+(\sigma_n^2/\sigma_p^2)\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{y}$, $\bar{\Sigma}=\left(\sigma_n^{-2}\mathbf{X}^T\mathbf{X}+\sigma_p^{-2}\mathbf{I}\right)^{-1}$

3.1.3 Predictions $p(y^*\mid\mathbf{X},\mathbf{y},\mathbf{x}^*)$

$=\int p(y^*|x^*,w)p(w|\mathbf{X},\mathbf{y})dw=\mathcal{N}\left(\bar{\mu}^T\mathbf{x}^*,\mathbf{x}^{*T}\bar{\Sigma}\mathbf{x}^*+\sigma_n^2\right)$
$\mathbf{x}^{*T}\bar{\Sigma}\mathbf{x}^*$: Uncertainty about f^*(epistemic)
σ_n^2: Noise / uncertainty about y^* given f^*(aleatoric)

3.1.4 Recursive Bayesian Updates

$p^{j+1}(\theta)=p(\theta|y_{1:j+1})=\frac{1}{Z}p(\theta|y_{1:j})p(y_{j+1}|\theta,y_{1:j})$
$p(\theta|y_{1:j})=p^j(\theta)$, $p(y_{j+1}|\theta,y_{1:j})=p_{j+1}(y_{j+1}|\theta)$

3.2 Kalman Filters

X_i: Tracked object loc., Y_i: Obs., $P(X_1)$: Prior belief
$P(X_{t+1}|X_t)$ Motion (Trans): $\mathbf{X}_{t+1}=\mathbf{F}\mathbf{X}_t+\varepsilon_t$, $\varepsilon_t\in\mathcal{N}(0,\Sigma_x)$
$P(Y_t|X_t)$ Sensor (Obs): $\mathbf{Y}_t=\mathbf{H}\mathbf{X}_t+\eta_t$, $\eta_t\in\mathcal{N}(0,\Sigma_y)$
Conditioning: $P(X_t|y_{1:t})=\frac{1}{Z}P(X_t|y_{1:t-1})P(y_t|X_t)$

3.2.1 Parameter Estimation $y_t=x+\mu_t,\mu_t\sim\mathcal{N}(0,\sigma_y^2)$

$k_{t+1}=\sigma_t^2/(\sigma_t^2+\sigma_y^2)$, $\sigma_{t+1}^2=\sigma_y^2 k_{t+1}$, for $\sigma_{t=0}^2\to\infty$: $\mu_{t+1}=\frac{y_1+\ldots+y_{t+1}}{t+1}$
$\sigma_{t+1}^2=\frac{\sigma_{t=0}^2\sigma_y^2}{(t+1)\sigma_{t=0}^2+\sigma_y^2}$, $k_{t+1}=\frac{\sigma_{t=0}^2}{(t+1)\sigma_{t=0}^2+\sigma_y^2}$, for $t\to\infty$: $\mu_{t+1}\to\mu_t$

3.2.2 General Kalman Update (Gaussian)

Transition(Motion) model: $P(\mathbf{x}_{t+1}|\mathbf{x}_t)=\mathcal{N}(\mathbf{x}_{t+1};\mathbf{F}\mathbf{x}_t,\Sigma_x)$
Sensor model: $P(\mathbf{y}_t|\mathbf{x}_t)=\mathcal{N}(\mathbf{y}_t;\mathbf{H}\mathbf{x}_t,\Sigma_y)$
Update: $\mu_{t+1}=\mathbf{F}\mu_t+\mathbf{K}_{t+1}(\mathbf{y}_{t+1}-\mathbf{H}\mathbf{F}\mu_t)$
$\Sigma_{t+1}=(\mathbf{I}-\mathbf{K}_{k+1}\mathbf{H})\left(\mathbf{F}\Sigma_t\mathbf{F}^\top+\Sigma_x\right)$
Gain: $\mathbf{K}_{t+1}=\left(\mathbf{F}\Sigma_t\mathbf{F}^T+\Sigma_x\right)\mathbf{H}^T\left(\mathbf{H}\left(\mathbf{F}\Sigma_t\mathbf{F}^T+\Sigma_x\right)\mathbf{H}^T+\Sigma_y\right)^{-1}$
Can compute Σ_t and $\mathbf{K}_t$ offline (not dependant on var x_t & y_t)

3.3 Kernel - Lin. method (BLR) on nonlin. transf. data

Cost proportional to dim of feature space, $\mathbf{x}_i^T\mathbf{x}_j\Longrightarrow k(\mathbf{x}_i,\mathbf{x}_j)$
$p(\mathbf{w})=\mathcal{N}(0,\sigma_p^2\mathbf{I})$, $f=\mathbf{X}\mathbf{w}\Longrightarrow f\sim\mathcal{N}(0,\sigma_p^2\mathbf{X}\mathbf{X}^T)$
Kernelize: $f\sim\mathcal{N}(0,\sigma_p^2\mathbf{K})$, Symmetric & Positive Definite
$k(\mathbf{x},\mathbf{x}')=k_1(\mathbf{x},\mathbf{x}')+k_2(\mathbf{x},\mathbf{x}')$, $k(\mathbf{x},\mathbf{x}')=k_1(\mathbf{x},\mathbf{x}')k_2(\mathbf{x},\mathbf{x}')$
$k(\mathbf{x},\mathbf{x}')=ck_1(\mathbf{x},\mathbf{x}')$ for $c>0$, $k(\mathbf{x},\mathbf{x}')=f(k_1(\mathbf{x},\mathbf{x}'))$
Linear: x^Tx', $\phi(x)^T\phi(x')$, $\phi(x)\to$ poly, sine, ...
RBF, Gauss: $\exp\left(-\|x-x'\|_2^2/h^2\right)$, Exp.: $\exp\left(-\|x-x'\|_2/h\right)$
Matérn: $\sigma^2\frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu}\|\mathbf{x}-\mathbf{x}'\|_2}{\rho}\right)^\nu K_\nu\left(\frac{\sqrt{2\nu}\|\mathbf{x}-\mathbf{x}'\|_2}{\rho}\right)$

3.4 GP

$p(f(x'))=GP(f;\mu(x'),k(x',x'))=\mathcal{N}(f(x');\mu(x'),k(x',x'))$
$\mu'(\mathbf{x})=\mu(\mathbf{x})+\mathbf{k}_{x,A}\left(\mathbf{K}_{AA}+\sigma^2\mathbf{I}\right)^{-1}(\mathbf{y}_A-\mu_A)$
$k'(\mathbf{x},\mathbf{x}')=k(\mathbf{x},\mathbf{x}')-\mathbf{k}_{x,A}\left(\mathbf{K}_{AA}+\sigma^2\mathbf{I}\right)^{-1}\mathbf{k}_{x',A}^T$
$var_n(f(x_*))=k(x_*,x_*)\text{-}k_*^T(K_n+\sigma_2 I_n)^{-1}k_*$

3.4.1 Model Selection, $\hat{\theta}=\arg\max_\theta p(\mathbf{y}\mid X,\theta)$

$\log p(\mathbf{y}\mid X,\theta)=-\frac{1}{2}\mathbf{y}^T\mathbf{K}_y^{-1}\mathbf{y}-\frac{1}{2}\log|\mathbf{K}_y|-\frac{n}{2}\log 2\pi$
Or: $p(\mathbf{y}\mid X,\theta)=\int p(\mathbf{y}\mid X,f)p(f\mid\theta)df$

3.4.2 Fast GP Methods

Kernel function approximations, Inducing point methods
Summarize data via func. vals of f at a set u of m ind. points
$p(\mathbf{f}^*,\mathbf{f})\approx q(\mathbf{f}^*,\mathbf{f})=\int q(\mathbf{f}^*\mid\mathbf{u})q(\mathbf{f}\mid\mathbf{u})p(\mathbf{u})d\mathbf{u}$

4 Bayesian learning

Posterior & Pred. not in closed-form (intractable) $\Rightarrow$ Approx.

4.1 Approximate Inference

$p(\theta\mid y)=\frac{1}{Z}p(\theta,y)\approx q(\theta\mid\lambda)$ or $q(\theta)=\mathcal{N}\left(\theta;\hat{\theta},\Lambda^{-1}\right)$
$\hat{\theta}=\arg\max_\theta p(\theta\mid y)$, $\Lambda=-\nabla\nabla\log p(\hat{\theta}\mid y)$
$p(\mathbf{w}\mid\mathbf{x}_{1:n},y_{1:n})\approx q(\mathbf{w})=\mathcal{N}\left(\mathbf{w};\hat{\mathbf{w}};\Lambda^{-1}\right)$
$p(y^*\mid\mathbf{x}^*,\mathbf{x}_{1:n},y_{1:n})\approx\int\sigma\left(y^*\mathbf{w}^T\mathbf{x}\right)\mathcal{N}\left(\mathbf{w};\hat{\mathbf{w}},\Lambda^{-1}\right)d\mathbf{w}=$
$\int\sigma(y^*f)\mathcal{N}\left(f;\hat{\mathbf{w}}^T\mathbf{x}^*,\mathbf{x}^{*T}\Lambda^{-1}\mathbf{x}^*\right)df$

4.2 Variational Inference $q^*\in\arg\min_{q\in Q}KL(q\|p)$

4.3 KL-Divergence (non-negative)

$KL(q\|p)=\int q(\theta)\log\frac{q(\theta)}{p(\theta)}d\theta$
$\arg\min_q KL(q\|p)=\arg\min_q\int q(\theta)\log\frac{q(\theta)}{\frac{1}{Z}p(\theta,y)}d\theta=$
$\arg\max_q\{\mathbb{E}_{\theta\sim q(\theta)}[\log p(\theta,y)]+H(q)\}=$
$\arg\max_q\{\mathbb{E}_{\theta\sim q(\theta)}[\log p(y\mid\theta)]-KL(q\|p(\cdot))\}$
$D_{KL}(p\|q)=\sum_x\sum_y p(x,y)log p(x,y)/q(x)q(y)=D_{KLx}\text{-}D_{KLy}+c$

4.3.1 Inference as Optimization (ELBO)

$\mathcal{L}(q)=\mathbb{E}_{\theta\sim q(\theta)}[\log p(\theta,y)]+H(q)$, $q(\theta\mid\lambda)=$
$\phi(\varepsilon)|\nabla_\varepsilon g(\varepsilon;\lambda)|^{-1}\Rightarrow\nabla_\lambda\mathbb{E}_{\theta\sim q_\lambda}[f(\theta)]=\mathbb{E}_{\varepsilon\sim\phi}[\nabla_\lambda f(g(\varepsilon;\lambda))]$

4.3.2 Hoeffding's Inequality

$P(|\mathbf{E}[f(x)]-\frac{1}{N}\sum_i=1^N f(x_i))|>\varepsilon)\leq 2\exp(-\frac{2N\varepsilon^2}{C^2})$

5 Markov-Chain Monte Carlo (MCMC)

$\mathbb{E}_{\theta\sim p(\cdot|x_{1:n},y_{1:n})}[f(\theta)]\approx\frac{1}{N}\sum_{i=1}^N f\left(\theta^i\right)$, $f(\theta)=p(y^*\mid x^*,\theta)$
Given Unnormalized Distribution: $P(x)=\frac{1}{Z}Q(x)=\pi(\mathbf{x})$
Ergodic: $\lim_{N\to\infty}P(X_N=x)=\pi(x)$, independent of $P(X_1)$
$\mathbb{E}[f(\mathbf{X})\mid\mathbf{x}_B]\approx\frac{1}{T-t_0}\sum_{\tau=t_0+1}^T f\left(\mathbf{X}^{(\tau)}\right)$

5.1 Metropolis Hastings (MH)

Detailed Balance Equation $Q(\mathbf{x})P(\mathbf{x}'\mid\mathbf{x})=Q(\mathbf{x}')P(\mathbf{x}\mid\mathbf{x}')$
Proposal dist. (Transition prob.): $x'\sim R(X'|X)$
$\alpha=\min\left\{1,\frac{Q(x')R(x|x')}{Q(x)R(x'|x)}\right\}\to X_{t+1}=x'$, $1-\alpha\to X_{t+1}=x$

5.2 Gibbs Sampling (Random Order, Practical Variant)

$x^{(0)}$ to all variables, fix observed varss X_B to observed val x_B
for($t=1$ to ∞)$\{x^{(t)}=x^{(t-1)}$;
foreach $(X_i\notin\mathbf{B})\{v_i=(x^{(t)}\neq x_i)$; sample $x_i^{(t)}$ from $P(X_i|v_i)\}\}$

5.3 MCMC for Continuous RVs: $p(\mathbf{x})=\frac{1}{Z}\exp(-f(\mathbf{x}))$

MH with Gaussian: $R(x'\mid x)=\mathcal{N}(x';x;\tau I)$
MALA: $R(x'\mid x)=\mathcal{N}(x';x-\tau\nabla f(x);2\tau I)$
SGLD: $\theta\sim\exp\left(\log p(\theta)+\sum_{i=1}^n\log p(y_i\mid x_i,\theta)\right)$
$L(\theta)=\log p(\theta)+\sum_{i=1}^n\log p(y_i\mid x_i,\theta)$

6 Bayesian Neural Networks

$p(\theta)=\mathcal{N}\left(\theta;0,\sigma_p^2 I\right)$, $p(y\mid\mathbf{x},\theta)=\mathcal{N}\left(y;f(\mathbf{x},\theta),\sigma^2\right)$
Noise: $p(y\mid\mathbf{x},\theta)=\mathcal{N}(y;f_1(\mathbf{x},\theta),\exp(f_2(\mathbf{x},\theta)))$
$\hat{\theta}=\arg\min_\theta\{-\log p(\theta)-\sum_{i=1}^n\log p(y_i\mid\mathbf{x}_i,\theta)\}=\arg\min_\theta$
$\{-\lambda\|\theta\|_2^2+\sum_{i=1}^n\frac{1}{2\sigma(\mathbf{x}_i;\theta)^2}\|y_i-\mu(\mathbf{x}_i;\theta)\|^2+\frac{1}{2}\log\sigma(\mathbf{x}_i;\theta)^2\}$
Integrals are Intractable $\Rightarrow$ Approximate Inference (inf):
Variational Inf (VI), MCMC, Dropout as VI, Prob. Ensembles

7 Bayesian Learning (uncertainty decides data)

Active Learning (Query points whose observation provides most useful information about the unknown function)

7.1 Optimizing Mutual Information

$F(S) := H(f) - H(f \mid y_S) = I(f; y_S) = \frac{1}{2}\log\left|I + \sigma^{-2}K_S\right|$

Greedy Algorithm: For $S_t = \{x_1, ..., x_t\}$

$x_{t+1} = \arg\max_{x \in D} F(S_t \cup \{x\}) = \arg\max_{x \in D} \sigma^2_{x|S_t}$

Heteroscedastic case: $x_{t+1} \in \arg\max_x \frac{\sigma_f^2(x)}{\sigma_n^2(x)}$

7.2 Active learning for classification

Max. entropy of pred. label $x_{t+1} \in \arg\max_x H(Y \mid x, x_{1:t}, y_{1:t})$

7.3 Bayesian Optimization

Cumulative regret: $R_T = \sum_{t=1}^T (\max_x f(x) - f(x_t))$

$R_T/T \to 0 \Rightarrow$ Sublinear $\Rightarrow \max_t f(x_t) \to f(x^*)$

7.3.1 Optimistic Bayesian Optimization with GPs

Acquisition Func.: $x_t = \arg\max_{x \in D} \mu_{t-1}(x) + \beta_t \sigma_{t-1}(x)$

EI: $(\mu(x)\text{-}f(x^+)\text{-}\xi)\,\Phi(Z) + \sigma(x)\phi(Z) \to \sigma_x > 0; 0 \to \sigma(x)=0$

where : $\mathbf{Z}=(\mu(x)\text{-}f(x^+)\text{-}\xi)/(\sigma(x)) \to \sigma_x > 0; 0 \to \sigma(x)=0$

Alternatives: Prob. of Improv.(PI), Infor. Directed Sampling

Thompson Sampling: $x_{t+1} = \arg\max \tilde{f} \quad \tilde{f} \sim P(f|\mathscr{D})$

Foreach t: $\tilde{f} \sim P(f \mid x_{1:t}, y_{1:t}) \to x_{t+1} \in \arg\max_{x \in D} \tilde{f}(x)$

8 Markov Decision Processes(MDP)

$J(\pi) = \mathbb{E}[r(X_0, \pi(X_0)) + \gamma r(X_1, \pi(X_1)) + \gamma^2 r(X_2, \pi(X_2)) + ...]$

$V^\pi(x) = J(\pi \mid X_0 = x) = \mathbb{E}[\sum_{t=0}^\infty \gamma^t r(X_t, \pi(X_t)) \mid X_0 = x]$

Recursion: $V^\pi(x) = r(x, \pi(x)) + \gamma \sum_{x'} P(x' \mid x, \pi(x)) V^\pi(x')$

Fixed point iteration $V^\pi = r^\pi + \gamma T^\pi V^\pi$

$V_i^\pi = V^\pi(x_i)$, $r_i^\pi = r^\pi(x_i, \pi(x_i))$, $T_{i,j}^\pi = P(x_j \mid x_i, \pi(x_i))$

8.1 Policy Iteration: π, $V^\pi(x) \to \pi_G$

8.2 Value Iteration $V_0(x) = max_a r(x,a)$

$Q_t(x,a)=r(x,a)+\gamma\sum_{x'} P(x' \mid x,a) V_{t\text{-}1}(x')$, $V_t(x)=\max_a Q_t(x,a)$

Stop when $\|V_t - V_{t-1}\|_\infty = \max_x |V_t(x) - V_{t-1}(x)| \le \varepsilon$

8.3 POMDP = Belief-state MDP

$P(Y_{t+1} = y \mid b_t, a_t) = \sum_{x,x'} b_t(x) P(x' \mid x, a_t) P(y \mid x')$

$b_{t+1}(x') = \frac{1}{Z}\sum_x b_t(x) P(X_{t+1} = x' \mid X_t = x, a_t) P(y_{t+1} \mid x')$

$r(b_t, a_t) = \sum_x b_t(x) r(x, a_t)$

9 Reinforcement Learning (RL)

Exploration (rnd A) $\to$ poor in rewards

Exploitation (best A) $\to$ stuck in suboptimum

On-Policy: Agent $\to$ action, choose exploration/exploitation

Off-Policy: Agent $\not\to$ actions, only observational data

9.1 Model-based RL

Learn MDP and optimize policy based on estimated MDP

Estimate transitions $P(X_{t+1} \mid X_t, A) \approx \frac{\text{Count}(X_{t+1}, X_t, A)}{\text{Count}(X_t, A)}$

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

9.1.1 ε_t Greedy

With probability ε_t: Pick random action

With probability $(1 - \varepsilon_t)$: Pick best action

9.1.2 R_{max} Algorithm

Input: x_0, γ

Init: $\forall x, a$: $x^* \to$ MDP, $r(x,a) = R_{max}$, $P(x^*|x,a) = 1$, π:

exec π, $\forall x_{visited}, a_{visited}$: update $r(x,a)$, $P(x'|x,a)$, recomp. π

Every T timesteps, $R_{max} \to$ near-opt reward || visit unkn. (x,a)

9.1.3 Receding-Horizon/Model-Predictive control (MPC)

$\max_{a_{t:t+H-1}} \sum_{\tau=t:t+H-1} \gamma^{\tau-t} r_\tau(x_\tau, a_\tau)$ s.t. $x_{\tau+1} = f(x_\tau, a_\tau)$

$x_\tau := x_\tau(a_{t:\tau-1}) := f(f(\ldots(f(x_t, a_t), a_{t+1}), \ldots), a_{\tau-1})$

$J_H(a_{t:t+H-1}) := \sum_{\tau=t:t+H-1} \gamma^{\tau-t} r_\tau(x_\tau(a_{t:\tau-1}), a_\tau)$

Pick $a_{t:t+H}^{(i^*)}$ that optimizes $i^* = \arg\max_{i \in \{1..m\}} J_H\left(a_{t:t+H-1}^{(i)}\right)$

If V: $J_H(a_{t:t+H-1}) \leftarrow J_H(a_{t:t+H-1}) + \gamma^H V(x_{t+H})$

9.1.4 MPC for stochastic transition models

$\max_{a_{t:t+H\text{-}1}} \mathbb{E}_{x_{t+1:t+H}}\left[\sum_{\tau=t:t+H\text{-}1} \gamma^{\tau\text{-}t} r_\tau + \gamma^H V(x_{t+H}) \mid a_{t:t+H\text{-}1}\right]$

$J_H(a_{t:t+H\text{-}1}) := \mathbb{E}_{x_{t+1:t+H}}\left[\sum_{\tau=t:t+H\text{-}1} \gamma^{\tau\text{-}t} r_\tau + \gamma^H V(x_{t+H}) \mid a_{t:t+H\text{-}1}\right]$

9.1.5 Unknown Dynamics (f and r are unknown)

Due to the Markovian structure of the MDP, observed transitions and rewards are (conditionally) independent

9.2 Model-free RL

Estimate the value function directly

9.2.1 Temporal Difference (TD)-Learning

$V_{t+1}^\pi = (1\text{-}\alpha_t) V_t^\pi(x) + \alpha_t(r + \gamma V_t^\pi(x')) \qquad \delta = r + \gamma V_t^{\pi]}(x') \text{-} V^\pi(x)$

$\ell_2(\theta; x, x', r) = \frac{1}{2}(V(x;\theta) - r - \gamma V(x'; \theta_{\text{old}}))^2$

9.2.2 Q-Learning

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

$Q(x,a) \leftarrow (1 - \alpha_t) Q(x,a) + \alpha_t(r + \gamma \max_{a'} Q(x', a'))$

9.2.3 Policy Gradients Methods

$J(\theta) = \mathbb{E}_{x_{0:T}, a_{0:T} \sim \pi_\theta} \sum_{t=0}^T \gamma^t r(x_t, a_t) = \mathbb{E}_{\tau \sim \pi_\theta} r(\tau)$

$\nabla J(\theta) = \nabla \mathbb{E}_{\tau \sim \pi_\theta} r(\tau) = \mathbb{E}_{\tau \sim \pi_\theta}[r(\tau) \nabla \log \pi_\theta(\tau)]$

Exploiting the MDP structure $r(\tau) = \sum_{t=0}^T \gamma^t r(s_t, a_t)$:

$\pi_\theta(\tau) = P(x_0) \prod_{t=0}^T \pi(a_t \mid x_t; \theta) P(x_{t+1} \mid x_t, a_t)$

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

Reducing Variance(Baseline b):

$\mathbb{E}_{\tau \sim \pi_\theta}[r(\tau) \nabla \log \pi_\theta(\tau)] = \mathbb{E}_{\tau \sim \pi_\theta}[(r(\tau) - b) \nabla \log \pi_\theta(\tau)]$

State-dependent baselines: $\mathbb{E}_{\tau \sim \pi_\theta}\left[\sum_{t=0}^T r(\tau) \nabla \log \pi(a_t \mid x_t; \theta)\right]$

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

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

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

$G_t = \sum_{t'=t}^T \gamma^{t'-t} r_{t'}$ reward to go followed action α_t

On: RF, AC methods, TRPO, R_{max}, optimistic Q-Learning

Off: DDPG, TD3, normal SAC, Q-Learning, optimistic Q-Learning with noise

10 REINFORCE Algorithm

Input Init a parametrized policy distr. $\pi(a|x, \theta)$ then **Loop**

Generate an episode $\tau^{(i)}$ (rollout) sampling from π

For t = 0,...T in the recorded episode

Set $G_t = R_t$ to the return from step t

Update $\theta \leftarrow \theta + \eta \gamma^t G_t \nabla_\theta \log \pi(A_t | X_t; \theta) = \theta + \eta \nabla_\theta J(\theta)$

+ Baselines : rtg $G_t = \sum_{t'=t}^T \gamma^{t'-t} r_{t'}$

10.0.1 Actor-Critic (AC) Algorithm

Advantage $A^\pi(x,a) = Q^\pi(x,a) - V^\pi(x)$

$\nabla J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta}\left[\sum_{t=0}^\infty \gamma^t Q(x_t, a_t; \theta_Q) \nabla \log \pi(a_t \mid x_t; \theta)\right] =$

$\mathbb{E}_{(x,a) \sim \pi_\theta}[Q(x,a;\theta_Q) \nabla \log \pi(a \mid x; \theta)]$

$\rho(x) = \sum_{t=0}^\infty \gamma^t p(x_t = x)$

- Allows application in the online (non-episodic) setting

$\theta_\pi \leftarrow \theta_\pi - \eta_t Q(x,a;\theta_Q) \nabla \log \pi(a \mid x; \theta_\pi)$

$\theta_Q \leftarrow \theta_Q\text{-}\eta_t(Q(x,a;\theta_Q)\text{-}r\text{-}\gamma Q(x', \pi(x', \theta_\pi); \theta_Q)) \nabla Q(x,a;\theta_Q)$

10.0.2 A2C Algorithm: Variance reduction via baselines

$\theta_\pi \leftarrow \theta_\pi + \eta_t[Q(x,a;\theta_Q) - V(x;\theta_V)] \nabla \log \pi(a \mid x; \theta_\pi)$

Advantage Function Estimate: $[Q(x,a;\theta_Q) - V(x;\theta_V)]$

10.0.3 Replace exact maximum by parametrized policy

$L(\theta_Q) = \sum_{(x,a,r,x') \in D}\left(r + \gamma Q\left(x', \pi(x'; \theta_\pi); \theta_Q^{\text{old}}\right) \text{-} Q(x,a;\theta_Q)\right)^2$

10.0.4 Deep Deterministic Policy Gradients (DDPG)

Actor Critic Method

11 Reinforcement Learning via Function Approximation

Parametrization

11.1 Parametric Value Function Approximation

To scale to large state spaces, learn an approximation of (action) value function $V(x;\theta)$ or $Q(x,a;\theta)$

11.1.1 Examples

(Deep) Neural Networks $\to$ Deep RL; Gradients for Q-learning with Function Approximation; Neural Fitted Q-iteration / DQN; Double DQN

11.2 Policy Search Methods (Deal. w/ large action sets)

Learning a Parameterized Policy $\pi(x) = \pi(x;\theta)$

For episodic tasks (i.e., can "reset" "agent"), can compute expected reward $J(\theta)$ by "rollouts"

Find optimal parameters through global optimization $\theta^* = \arg\max_\theta J(\theta)$

12 Langevin

V is diffbar und convex: $(\nabla V(x)\text{-}\nabla V(y))(x\text{-}y) \ge 0$