

## 1 Useful concepts and Introduction

### 1.1 Usefull math

$\varphi$  is convex  $\Rightarrow \varphi(\mathbb{E}[X]) \leq \mathbb{E}[\varphi(X)]$

**Hoeffding:**  $Z_1, \dots, iid, Z_i \in [0, C], \mathbb{E}[Z_i] = \mu$   
 $\Rightarrow P\left(\left|\mu - \frac{1}{n} \sum_{i=1}^n Z_i\right| > \epsilon\right) \leq 2 \exp(-2n \frac{\epsilon^2}{C}) \leq \delta$   
 $\Rightarrow n \geq \frac{C}{2\epsilon^2} \log \frac{2}{\delta}$

**Robbins Monro**  $\alpha_t \xrightarrow{RM} 0: \sum \alpha_t = \infty, \sum \alpha_t^2 < \infty$

### 1.2 Multivariate Gaussian

$$f(x) = \frac{1}{2\pi\sqrt{|\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}$$

Suppose we have a Gaussian random vector

$$\begin{bmatrix} X_A \\ X_B \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_A \\ \mu_B \end{bmatrix}, \begin{bmatrix} \Sigma_{AA} & \Sigma_{AB} \\ \Sigma_{BA} & \Sigma_{BB} \end{bmatrix}\right) \Rightarrow X_A|X_B = x_B \sim \mathcal{N}(\mu_A + \Sigma_{AB}\Sigma_{BB}^{-1}(x_B - \mu_B), \Sigma_{AA} - \Sigma_{AB}\Sigma_{BB}^{-1}\Sigma_{BA})$$

### 1.3 Information Theory elements:

**Entropy:**  $H(X) \doteq -\mathbb{E}_{x \sim p_X} [\log p_X(x)]$

$$H(X|Y) \doteq -\mathbb{E}_{(x,y) \sim p_{(X,Y)}} [\log p_{Y|X}(y|x)]$$

$$\text{if } X \sim \mathcal{N}(\mu, \Sigma) \Rightarrow H(X) = \frac{1}{2} \log[(2\pi e)^d \det(\Sigma)]$$

**Chain Rule:**  $H(X, Y) = H(Y|X) + H(X)$

**Mutual Info:**  $I(X, Y) \doteq KL(p_{(X,Y)} \| p_X p_Y)$

$$I(X, Y) = H(X) - H(X|Y)$$

if  $X \sim \mathcal{N}(\mu, \Sigma)$ ,  $Y = X + \epsilon$ ,  $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$ :

$$\text{then } I(X, Y) = \frac{1}{2} \log \left[ \det \left( I + \frac{1}{\sigma^2} \Sigma \right) \right]$$

### 1.4 Kullback-Leiber divergence

$$KL(p||q) = \mathbb{E}_p \left[ \log \frac{p(x)}{q(x)} \right]$$

if  $p_0 \sim \mathcal{N}(\mu_0, \Sigma_0)$ ,  $p_1 \sim \mathcal{N}(\mu_1, \Sigma_1) \Rightarrow KL(p_0||p_1)$

$$= \frac{1}{2} \left( \text{tr}(\Sigma_1^{-1} \Sigma_0) + (\mu_1 - \mu_0)^T \Sigma_1^{-1} (\mu_1 - \mu_0) - k + \log \frac{|\Sigma_1|}{|\Sigma_0|} \right)$$

$$\hat{q} = \arg \min_q KL(p||q) \Rightarrow \text{overconservative}$$

$$\hat{q} = \arg \min_q KL(q||p) \Rightarrow \text{overconfident}$$

## 2 Bayesian Regression

$$w \sim N(0, \sigma_p^2 I), \epsilon \sim N(0, \sigma_n^2 I), y = Xw + \epsilon$$

$$y|w \sim N(Xw, \sigma_n^2 I)$$

$$w|y \sim N((X^T X + \lambda I)^{-1} X^T y, (X^T X + \lambda I)^{-1} \sigma_n^2)$$

### 3 Kalman Filter

$$\begin{cases} X_{t+1} = F X_t + \epsilon_t & \epsilon_t \sim N(0, \Sigma_x) \\ Y_t = H X_t + \eta_t & \eta_t \sim N(0, \Sigma_y) \end{cases} \quad X_1 \sim N(\mu_p, \Sigma_p)$$

Then if  $X_0$  is Gaussian then  $X_t|Y_{1:t} \sim N(\mu_t, \Sigma_t)$ :

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

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

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

### 4 Gaussian Processes

$$f \sim GP(\mu, k) \Rightarrow \forall \{x_1, \dots, x_n\} \forall n < \infty$$

$$[f(x_1) \dots f(x_n)] \sim N([\mu(x_1) \dots \mu(x_n)], K)$$

where  $K_{ij} = k(x_i, x_j)$

## 2.1 Gaussian Process Regression

$f \sim GP(\mu, k)$  then:  $f|y_{1:n}, x_{1:n} \sim GP(\tilde{\mu}, \tilde{k})$

$$\tilde{\mu}(x) = \mu(x) + K_{A,x}^T (K_{AA} + \epsilon I_n)^{-1} (y_A - \mu_A)$$

$$\tilde{k}(x, x') = k(x, x') - K_{A,x}^T (K_{AA} + \epsilon I_n)^{-1} K_{A,x'}$$

Where:  $K_{A,x} = [k(x_1, x) \dots k(x_n, x)]^T$

$$[K_{AA}]_{ij} = k(x_i, x_j) \text{ and } \mu_A = [\mu(x_1) \dots \mu(x_n)]^T$$

### 4.2 Kernels

$k(x, y)$  is a kernel if it's symmetric semidefinite positive:

$\forall \{x_1, \dots, x_n\}$  then for the Gram Matrix

$$[K]_{ij} = k(x_i, x_j) \text{ holds } c^T K c \geq 0 \forall c$$

**Some Kernels:** (h is the bandwidth hyperp.)

$$\text{Gaussian (rbf): } k(x, y) = \exp\left(-\frac{\|x-y\|^2}{h^2}\right)$$

$$\text{Exponential: } k(x, y) = \exp\left(-\frac{\|x-y\|}{h}\right)$$

$$\text{Linear kernel: } k(x, y) = x^T y \text{ (here } K_{AA} = X X^T)$$

### 4.3 Optimization of Kernel Parameters

Given a dataset  $A$ , a kernel function  $k(x, y; \theta)$ .

$y \sim N(0, K_y(\theta))$  where  $K_y(\theta) = K_{AA}(\theta) + \sigma_n^2 I$

$$\hat{\theta} = \arg \max_{\theta} \log p(y|X; \theta)$$

$$\text{In GP: } \hat{\theta} = \arg \min_{\theta} y^T K_y^{-1}(\theta) y + \log |K_y(\theta)|$$

We can from here  $\nabla \downarrow$ :

$$\nabla_{\theta} \log p(y|X; \theta) = \frac{1}{2} \text{tr} \left( (\alpha \alpha^T - K^{-1}) \frac{\partial K}{\partial \theta} \right), \alpha = K^{-1} y$$

Or we could also be bayesian about  $\theta$

### 4.4 Aproximation Techniques

**Local method:**  $k(x_1, x_2) = 0$  if  $\|x_1 - x_2\| \gg 1$

**Random Fourier Features:** if  $k(x, y) = \kappa(x - y)$   
 $p(w) = \mathcal{F} \{ \kappa(\cdot), w \}$ . Then  $p(w)$  can be normalized to be a density.

$$\kappa(x - y) = \mathbb{E}_{p(w)} \left[ \exp \{ i w^T (x - y) \} \right] \text{ antitransform}$$

$$\kappa(x - y) = \mathbb{E}_{b \sim \mathcal{U}([0, 2\pi]), w \sim p(w)} [z_{w,b}(x) z_{w,b}(y)]$$

where  $z_{w,b}(x) = \sqrt{2} \cos(w^T x + b)$ . I can MC extract features  $z$ . If # features is  $\ll n$  then this is faster ( $X^T X$  vs  $XX^T$ )

**Inducing points:** We a vector of inducing variables  $u$

$$f_A|u \sim N(K_{Au} K_{uu}^{-1} u, K_{AA} - K_{Au} K_{uu}^{-1} K_{uA})$$

$$f_*|u \sim N(K_{*u} K_{uu}^{-1} u, K_{**} - K_{*u} K_{uu}^{-1} K_{u*})$$

**Subset of Regressors (SoR):**  $\blacksquare \rightarrow 0$

**FITC:**  $\blacksquare \rightarrow$  its diagonal

## 5 Approximate inference

### 5.1 Laplace Approximation

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

$$\Lambda = -\nabla_{\theta} \nabla_{\theta} \log p(\theta|y)|_{\theta=\hat{\theta}}$$

$$p(\theta|y) \simeq q(\theta) = N(\hat{\theta}, \Lambda^{-1})$$

## 5.2 Variational Inference

$$\hat{q} = \arg \min_{q \in \mathcal{Q}} KL(q||p(\cdot|y))$$

$$\hat{q} = \arg \max_{q \in \mathcal{Q}} ELBO \text{ Evidence Lower Bound}$$

$$ELBO \doteq \mathbb{E}_{\theta \sim q} [\log p(y|\theta)] - KL(q||p(\cdot)) \leq \log p(y)$$

### 5.3 Markov Chain Monte Carlo

**Idea:** All we need is sampling from posterior

**Ergodic Markov Chain:**

$$\exists t \text{ s.t. } \mathbb{P}(i \rightarrow j \text{ in } t \text{ steps}) > 0 \quad \forall i, j \Rightarrow$$

$$\exists ! \pi = \lim_{N \rightarrow \infty} \mathbb{P}(X_N = x) \text{ Limit distribution}$$

**Ergodic Theorem:** if  $(X_i)_{i \in \mathbb{N}}$  is ergodic:

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N f(X_i) = \mathbb{E}_{x \sim \pi} [f(x)]$$

**Detailed Blanced Equation:**

$P(x|x')$  is the transition model of a MC:

if  $R(x)P(x'|x) = R(x')P(x|x')$  then  $R$  is the limit distribution of the MC

**Metropolis Hastings Algo:** Sample from a MC

which has  $P(x) = \frac{Q(x)}{Z}$  as limit dist.

**Result:**  $\{X_i\}_{i \in \mathbb{N}}$  sampled from the MC

**init:**  $R(x|x')$

*/\* Good R choice  $\rightarrow$  fast convergence \*/*

**init:**  $X_0 = x_0$

**for**  $t \leftarrow 1, 2, \dots$  **do**

$$x' \sim R(\cdot, x_{t-1})$$

$$\alpha = \min \left\{ 1, \frac{Q(x')R(x_{t-1}|x')}{Q(x_{t-1})R(x'|x_{t-1})} \right\}$$

**with probability**  $\alpha$  **do**

$$\lfloor X_t = x';$$

**otherwise**  $X_t = x_{t-1};$

**Metropolis Adj. Langevin Algo (MALA):**

$$\text{Energy function: } P(x) = \frac{Q(x)}{Z} = \frac{1}{Z} \exp(-f(x))$$

We chose:  $R(x|x') = \mathcal{N}(x' - \tau \nabla f(x), 2\tau I)$

**Stoch. Grad. Langevin Dynamics (SGLD):**

We use SGD to Approximate  $\nabla f$ . Converges also without acceptance step

**Hamilton MC:** SGD performance improved by adding momentum (consider last step  $\nabla f$ )

**Gibbs sampling:** Practical when  $X \in \mathbb{R}^n$

Used when  $P(X_{1:n})$  is hard but  $P(X_i|X_{-i})$  is easy.

**init:**  $x_0 \in \mathbb{R}^n$ ;  $(x_0^{(B)} = x^{(B)})$  B is our data

**for**  $t = 1, 2, \dots$  **do**

**with**  $i \sim \mathcal{U}(\{1 : n\} \setminus B)$  **do**

$$\lfloor x_{t-1}^{(i)} \sim P(x^{(i)}|x^{(-i)})$$

\* if we do it  $\forall i \notin B$  no DBE but more practical

### 5.4 Variable elimination for MPE (most probable explanation):

With loopy graphs, BP is often **overconfident/oscillates**.

## 6 Bayesian Neural Nets

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

Prior:  $p(\theta) = \mathcal{N}(0, \sigma_p^2)$

$$\theta_{MAP} = \arg \max_{\theta} \log(p(y, \theta))$$

### 6.1 Variation inference:

Usually we use  $\mathcal{Q}$  = Set of Gaussians

$\hat{q} = \arg \max_{\mathcal{Q}} ELBO$  Reparameterization trick

$q$  approx. the posterior but how to predict?

$$p(y^*|x^*, \mathcal{D}) \simeq \frac{1}{m} \sum_{j=1}^m p(y^*|x^*, \theta^{(j)}), \theta \sim \hat{q}(\theta)$$

Gaussian Mixture distribution:  $\mathbb{V}(y^*|x^*, \mathcal{D}) \simeq$

$$\simeq \frac{1}{m} \sum_{j=1}^m \sigma^2(x^*, \theta^{(j)}) + \frac{1}{m} \sum_{j=1}^m (\mu(x^*, \theta^{(j)}) - \bar{\mu}(x^*))^2$$

■  $\rightarrow$  Alestoric, ■  $\rightarrow$  Epistemic

**Dropouts Regularization:** Random ignore nodes in SGD iteration: Equavalent to VI with

$$Q = \{q(\cdot|\lambda) = \prod_j q_j(\theta_j|\lambda), \lambda \in \mathbb{R}^d\}$$

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

This allows to do Dropouts also in prediction

### 6.2 MCMC:

MCMC but cannot store all the  $\theta^{(i)}$ :

1) Subsampling: Only store a subset of the  $\theta^{(i)}$

2) Gaussian Aproximation: We only keep:

$$\mu_i = \frac{1}{T} \sum_{j=1}^T \theta_i^{(j)} \text{ and } \sigma_i = \frac{1}{T} \sum_{j=1}^T (\theta_i^{(j)} - \mu_i)^2$$

And update them online.

**Predictive Esnable NNs:**

Let  $\mathcal{D} = \{(x_i, y_i)\}_{i=1:n}$  be our dataset.

Train  $\theta_i^{MAP}$  on  $\mathcal{D}_i$  with  $i = 1, \dots, m$

$\mathcal{D}_i$  is a Bootstrap of  $\mathcal{D}$  of same size

$$\text{and } p(y^*|x^*, \mathcal{D}) \simeq \frac{1}{m} \sum_{j=1}^m p(y^*|x^*, \theta_i^{MAP})$$

### 6.3 Model calibration

Train  $\hat{q}$  on  $\mathcal{D}_{train}$

Evaluate  $\hat{q}$  on  $\mathcal{D}_{val} = \{(y', x')\}_{i=1:m}$

Held-Out-Likelihood  $\doteq \log p(y'_{1:m}|x'_{1:m}, \mathcal{D}_{train})$

$$\geq \mathbb{E}_{\theta \sim \hat{q}} \left[ \sum_{i=1}^m \log p(y'_i|x'_i, \theta) \right] \text{ (Jensen)}$$

$$\simeq \frac{1}{k} \sum_{j=1}^k \sum_{i=1}^m \log p(y'_i|x'_i, \theta^{(j)}), \theta^{(j)} \sim \hat{q}$$

**Evaluate predicted accuracy:** We divide  $\mathcal{D}_{val}$  into bins according to predicted confidence values. In each bin we compare accuracy with confidence

## 7 Active Learning

Let  $\mathcal{D}$  be the set of observable points.

We can observe  $\mathcal{S} \subseteq \mathcal{D}, |\mathcal{S}| \leq R$

Information Gain:  $\hat{\mathcal{S}} = \arg \max_{\mathcal{S}} F(\mathcal{S}) = I(f, y_{\mathcal{S}})$

For GPs:  $F(\mathcal{S}) = \frac{1}{2} \log |I + \frac{1}{\sigma^2} K_{\mathcal{SS}}|$

This is NP Hard,  $\Rightarrow$  Greedy Algo:

**init:**  $S^* = \emptyset$   
**for**  $t = 1 : R$  **do**  
     $x_t = \arg \max_{x \in \mathcal{D}} F(S^* \cup \{x\})$   
     $(x_t = \arg \max_{x \in \mathcal{D}} \sigma_x^2 |S \text{ for GPs})$   
     $(x_t = \arg \max_{x \in \mathcal{D}} \frac{\sigma_{f|S}^2(x)}{\sigma_n^2(x)} \text{ for heter. GPs})$   
     $S^* = S \cup \{x_t\}$

F is **Submodular** if:  $\forall x \in \mathcal{D}, \forall A \subseteq B \subseteq \mathcal{D}$  holds that:  $F(A \cup \{x\}) - F(A) \geq F(B \cup \{x\}) - F(B)$

F is Submodular  $\Rightarrow F(S^*) \geq (1 - \frac{1}{e})F(\mathcal{S})$

## 8 Bayesian Optimization

Like Active Learning but we only want to find the optima. We pick  $x_1, x_2, \dots$  from  $\mathcal{D}$  and observe  $y_i = f(x_i) + \epsilon_i$ .

**Cumulative regret:**  $R_T = \sum_{t=1}^T (\max_{x \in \mathcal{D}} f(x) - f(x_t))$

**Oss:**  $\frac{R_T}{T} \rightarrow 0 \Rightarrow \max_t f(x_t) \rightarrow \max_{x \in \mathcal{D}} f(x)$

### 8.1 Upper Confidence Sampling

With GP  $x_t = \arg \max_{x \in \mathcal{D}} \mu_{t-1}(x) + \beta_t \sigma_{t-1}(x)$

Choosing the correct  $\beta_t$  we get:  $\frac{R_T}{T} = \mathcal{O}\left(\sqrt{\frac{\gamma_T}{T}}\right)$ .

Where  $\gamma_t = \max_{|S| \leq T} I(f; \gamma_S)$ . On d dims:

Linear:  $\gamma_T = \mathcal{O}(d \log T)$  RBF:  $\gamma_T = \mathcal{O}((\log T)^{d+1})$

**Optimal**  $\beta_t = \mathcal{O}(\|f\|_K^2 + \gamma_t \log^3 T)$

**Oss:**  $\beta \uparrow$  = more exploration

### 8.2 Thompson Samling

$x_t = \arg \max_{x \in \mathcal{D}} \tilde{f}(x), \tilde{f} \sim p(f|x_{1:n}, y_{1:n})$

## 9 Markov Decision Process (MDP)

### 9.1 Definitions

$\mathcal{X} = \{1, \dots, n\}$  states;  $\mathcal{A} = \{1, \dots, m\}$  actions;

$p(x'|x, a)$  transition probability;

$r(x, a)$  reward (can be random);  $\pi: \mathcal{X} \rightarrow \mathcal{A}$  policy;

$T^\pi \in \mathbb{R}^{n \times n}, T_{ij}^\pi = p(j|i, \pi(i))$  Transition Matrix:

$J(\pi) = \mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^i r(X_i, \pi(X_i))\right]$  Expected value:

$V^\pi: \mathcal{X} \rightarrow \mathbb{R}, x \mapsto J(\pi|X_0 = x)$  Value function;

$Q^V(x, a) = r(x, a) + \gamma \sum_{x' \in \mathcal{X}} p(x'|x, a) V(x')$  Q func;

$\pi_G^V(x) = \arg \max_a Q^V(x, a)$  greedy policy w.r.t. V;

### 9.2 Value function Theorem

$V^\pi(x) = r(x, \pi(x)) + \gamma \sum_{x' \in \mathcal{X}} p(x'|x, \pi(x)) V^\pi(x')$

Matrix formulation:  $(I - \gamma T^\pi) V^\pi = r^\pi$

### 9.3 Bellman Theorem

1)  $\pi^*, V^*$  are optimal policy and it's value func.

2)  $\pi^* = \pi_G^{V^*}$

3)  $V^*(x) = \max_a [r(x, a) + \gamma \sum_{x' \in \mathcal{X}} p(x'|x, a) V^*(x)]$

1)  $\Leftrightarrow$  2)  $\Leftrightarrow$  3)

## 9.4 Algorithms

### 9.4.1 Policy iteration

**while** no more changes **do**

$\pi \leftarrow \pi_G^V$  (Update the Policy)

$V \leftarrow (I - \gamma T^\pi)^{-1} r^\pi$  (Update the value)

### 9.4.2 Value iteration

**while**  $\|V_t - V_{t-1}\| \leq \epsilon$  **do**

**foreach**  $x \in \mathcal{X}, a \in \mathcal{A}$  **do**

$Q_t(x, a) \leftarrow r(x, a) +$

$\gamma \sum_{x' \in \mathcal{X}} p(x'|x, a) V_{t-1}(x)$

**foreach**  $x \in \mathcal{X}$  **do**

$V_t(x) \leftarrow \max_a Q_t(x, a)$

$\hat{\pi} = \pi_G^{V_T}$ ; where  $V_T$  last found Value

## 9.5 Partially Observable MDP (POMDP)

POMDP can be seen as MDP where:

1)  $\lambda_{POMDP}$  are prob. distribution over  $\mathcal{X}_{MDP}$

2) the actions are the same

3)  $r_{POMDP}(b, a) = \mathbb{E}_{x \sim b} [r_{MDP}(x, a)]$

4) Trans. model:  $b_{t+1}(x) = \mathbb{P}(X_{t+1} = x | y_{1:t+1}, a_t)$

$b_{t+1}(x) = \frac{1}{Z} p(y_{t+1} | X_{t+1} = x) \sum_{x' \in \mathcal{X}_{MDP}} p(x|x', a_t) b_t(x')$

**How to solve?** Discretize  $\mathcal{X}_{POMDP}$  and treat it as a MDP or Policy gradient techniques

## 10 Non Parametric RL

It is an MDP with unknown  $p(x'|x, a)$  and  $r(x, a)$

### 10.1 Model-based RL

From all steps  $X_{t+1}, R_t | X_t, A_t$  we can learn:

$p(x'|x, a) \approx \hat{p}_{x'|x, a} = \frac{\text{Count}(X_{t+1}=x', X_t=x, A_t=a)}{\text{Count}(X_t=x, A_t=a)}$

$r(x, a) \approx \hat{r}_{x, a} = \frac{1}{\text{Count}(X_t=x, A_t=a)} \sum_t |X_t=x, A_t=a R_t$

How to chose  $a_t$ ?

#### 10.1.1 $\epsilon$ -greedy (On-Policy)

With probability  $\epsilon$ , pick random action.

With probability  $1 - \epsilon$ , pick  $a = \arg \max Q(x, a)$ .

**Oss:** Q is caclulated from  $(\hat{p}, \hat{r})$

**Th:** If  $\epsilon_t \xrightarrow{RM} 0$  then  $(\hat{r}, \hat{p}) \xrightarrow{a.s.} (r, p)$

#### 10.1.2 Softmax (On-Policy)

Draw  $a \sim q(a|x) = \text{softmax} \frac{Q(x, a)}{\tau}$

If  $\tau \uparrow$  it means I trust less Q

### 10.1.3 $R_{max}$ Reinforcement (On-Policy)

We add a fairy state  $x^*$

**init:**  $r(x, a) = R_{max} \forall x \in \mathcal{X} \cup \{x^*\}, a \in \mathcal{A}$

**init:**  $p(x^*|x, a) = 1 \forall x \in \mathcal{X}, a \in \mathcal{A}$

**init:**  $\pi$  = optimal policy w.r.t.  $p, r$

**repeat**

    Execute  $\pi$  and get  $x_{t+1}$  and  $r_t$

    Update belief of  $r(x_t, \pi(x_t))$  and

$p(x_{t+1}|x_t, \pi(x_t))$

    If oberserved 'enough' in  $(x, a)$

        recompute  $\pi$  using the updated

        belief only in  $(x, a)$

**until;**

'Enough'? See Hoeffding's inequality

( $\hat{p} \in [0, 1], \hat{r} \in [0, R_{max}]$ ).

**PAC bound:** With probability  $1 - \delta$ ,  $R_{max}$  will reach an  $\epsilon$ -optimal policy in a number of steps that is polynomial in  $|\mathcal{X}|, |\mathcal{A}|, T, 1/\epsilon$  and  $\log(1/\delta)$ . Memory  $\mathcal{O}(|\mathcal{X}|^2 |\mathcal{A}|)$ .

### 10.2 Model-free RL

Learn  $\pi^*$  only via  $V^*$  or  $Q^V$

#### 10.2.1 TD-learning (On-Policy)

Given a policy  $\pi$  we want to learn  $V^\pi$

$V^\pi(x) = \mathbb{E}_{R \sim r(x, \pi(x)), X' \sim p(\cdot | x, \pi(x))} [R + \gamma V^\pi(X')]$

After seeing  $(x_{t+1}, r_t | x_t, \pi(x_t))$  we update:

$V_{t+1}(x_t) \leftarrow (1 - \alpha_t) V_t(x_t) + \alpha_t (r_t + \gamma V_t^\pi(x_{t+1}))$

Where  $\alpha_t$  is a regularizer term (only 1 sample)

**Th:** If  $\alpha_t \xrightarrow{RM} 0$  then  $V \xrightarrow{a.s.} V^\pi$

#### 10.2.2 Q-learning (Off Policy)

Given experience we want to learn  $Q^* = Q^{V^*}$

$Q^*(x, a) = \mathbb{E}_{R \sim r(x, \pi(x)), X' \sim p(\cdot | x, \pi(x))} [R + \gamma \max_{a'} Q^*(X', a')]$

After seeing  $(x_{t+1}, r_t | x_t, a_t)$  we update:

$Q(x_t, a_t) \leftarrow (1 - \alpha_t) Q(x_t, a_t) + \alpha_t (r_t + \gamma \max_{a'} Q(x_{t+1}, a'))$

**Th:** If  $\alpha_t \xrightarrow{RM} 0$  then  $Q \xrightarrow{a.s.} Q^*$

**Optimistic Q learning:**

Initialize:  $Q(x, a) = \frac{R_{max}}{1 - \gamma} \prod_{t=1}^{T_{init}} (1 - \alpha_t)^{-1}$

Same convergence time as with  $R_{max}$ . Memory  $\mathcal{O}(|\mathcal{X}| |\mathcal{A}|)$ . Comp:  $\mathcal{O}(|\mathcal{A}|)$ .

## 11 Parametric RL

### 11.1 Parametric TD-learning

#### 11.1.1 TD-learning as SGD

TD-learning = 1 sample  $(x', r | x, \pi(x))$  SGD on:

$\bar{l}_2(V; x, r) = \frac{1}{2} (V - r - \gamma \mathbb{E}_{x' \sim p(\cdot | x, \pi(x))} [\hat{V}^\pi(x')])^2$

1 sample estimate of  $\nabla_V \bar{l}_2 = \delta = V - r - \gamma \hat{V}^\pi(x')$

$\Rightarrow V \leftarrow V - \alpha_t \delta$  where  $V = \hat{V}^\pi(x)$

#### 11.1.2 TD-parametric

If  $\hat{V}^\pi(x) = V(x, \theta)$  then:

$\delta = [V(x; \theta) - r - \gamma V(x'; \theta_{old})] \nabla_\theta V(x, \theta)$

### 11.2 Parametric Q-learning

$\delta(\theta, \theta_{old}) = (Q(x, a; \theta) - r - \gamma \max_{a'} Q(x', a'; \theta_{old}))$

We don't differantiate with regard to  $\theta_{old}$

The SGD step is:  $\theta \leftarrow \theta - \alpha_t \delta(\theta, \theta) \nabla_\theta Q(x, a; \theta)$

**Deep Q Networks (DQN):** Version of Q-learning where we update Q only each batch:

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

**Double DQN (better):**

$L(\theta) = \sum_{(x, a, r, x') \in \mathcal{D}} (r + \gamma Q(x', a^*(\theta); \theta_{old}) - Q(x, a; \theta))^2$

where:  $a^*(\theta) \doteq \arg \max_{a'} Q(x', a'; \theta)$

### 11.3 Policy-Search method

$\pi(x) = \pi(x; \theta); r(\tau) = \sum_{t=0}^T \gamma^t r(x_t, a_t)$

$J(\theta) \doteq J(\pi(\cdot; \theta)) = \mathbb{E}_{\tau \sim \pi_\theta} [r(\tau)] \simeq \frac{1}{m} \sum_{i=1}^m r(\tau^{(i)})$

with  $\pi_\theta(\tau) := p(x_0) \prod_{t=1}^T \pi(a_t | x_t; \theta) p(x_{t+1} | x_t, a_t)$

$\hat{\theta} = \arg \max_\theta J(\theta) \Rightarrow$  SGD

#### 11.3.1 Theory results:

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

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

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

#### 11.3.2 REINFORCE

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

Let  $G_t = \sum_{t'=t}^T \gamma^{t'-t} r_{t'} = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \dots$

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

SGD:  $\theta \leftarrow \theta + \eta_t \gamma^t G_t \nabla_\theta \log \pi(a_t | x_t; \theta) \quad \forall t = 1 : T$

**Oss:**  $G_t \leftarrow G_t - \frac{1}{T} \sum_{t'=0}^{T-1} G_{t'} \Rightarrow$  Less Variance

### 11.4 Actor Critic

After seeing  $(x, a, r, x')$ :

$\theta_\pi \leftarrow \theta_\pi + \eta_t Q(x, a, \theta_Q) \nabla_{\theta_\pi} \log \pi(a | x; \theta_\pi)$

$\theta_Q \leftarrow \theta_Q - \eta_t \delta \nabla_{\theta_Q} Q(x, a; \theta_Q)$

$\delta = Q(x, a; \theta_Q) - r - \gamma Q(x', \pi(x', \theta_\pi), \theta_Q)$

#### 11.4.1 Theory justification:

$\mathbb{E}_{\tau_{t+1:\infty} \sim \pi_\theta} [G_t | a_t, x_t] = Q^\pi(a_t, x_t)$

$\nabla_{\theta_\pi} J(\theta_\pi) = \mathbb{E}_{\tau \sim \pi_{\theta_\pi}} \left[ \sum_{t=0}^T \gamma^t \nabla_{\theta_\pi} \log \pi(a_t | x_t; \theta_\pi) Q(x_t, a_t) \right]$

$\nabla_{\theta_\pi} J(\theta_\pi) = \mathbb{E}_{\tau \sim \pi_{\theta_\pi}, x \sim \rho(\cdot)} [\nabla_{\theta_\pi} \log \pi(a_t | x_t; \theta_\pi) Q(x_t, a_t)]$

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

#### 11.4.2 Advantage Function Baseline

We can us  $V(x; \theta_V)$  as baseline for  $Q(x, a; \theta_Q)$

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

**Oss:**  $\forall \pi \forall x \max_a A^\pi(x, a) \geq 0$

**Oss:**  $\pi$  optimal  $\forall x A^\pi(x, \pi(x)) \leq 0$ .

#### 11.4.3 Off policy variation

$\hat{\theta}_\pi = \arg \max J(\theta); J(\theta) = \mathbb{E}_{x \sim \mu} [Q(x, \pi(x; \theta); \theta_Q)]$

where  $\mu$  visits all states. SGD can now be done

in Batches and Off-policy also for  $\theta_\pi$ :

$\theta_\pi \leftarrow \theta_\pi + \eta \nabla_{\theta_\pi} \frac{1}{|B|} \sum_{(x, a, r, x') \in B} Q(x, \pi(x, \theta_\pi); \theta_Q)$

If  $\pi$  is randomized we can use the reparametrization trick.

$J_\lambda(\theta) = J(\theta) + \lambda H(\pi_\theta)$  to encourage exploration