1 Review of useful concepts and Introduction

1.1 Usefull math

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

1.3 Information Theory elements:

Entropy: $H(X) \doteq -\mathbb{E}_{x\sim p_X}[\log p_X(x)]$
$H(X|Y) \doteq -\mathbb{E}_{(x,y)\sim p_{(X,Y)}}\left[\log p_{Y|X}(y|x)\right]$
if $X \sim \mathcal{N}(\mu, \Sigma) \Rightarrow H(X) = \frac{1}{2}\log\left[(2\pi e)^d \det(\Sigma)\right]$
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 $X \sim \mathcal{N}(\mu, \Sigma)$, $Y = X + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$:
then $I(X,Y) = \frac{1}{2}\log\left[\det(I + \frac{1}{\sigma^2}\Sigma)\right]$

1.4 Kullback-Leiber divergence

$KL(p\|q) = \mathbb{E}_p\left[\log\frac{p(x)}{q(x)}\right]$
if $p_0 \sim \mathcal{N}(\mu_0, \Sigma_0)$, $p_1 \sim \mathcal{N}(\mu_1, \Sigma_1) \Rightarrow KL(p_0\|p_1)$
$= \frac{1}{2}\left(tr\left(\Sigma_1^{-1}\Sigma_0\right) + (\mu_1 - \mu_0)^T\Sigma_1^{-1}(\mu_1 - \mu_0) - k + \log\frac{|\Sigma_1|}{|\Sigma_0|}\right)$
$\hat{q} = \arg\min_q KL(p\|q) \Rightarrow$ overconservative
$\hat{q} = \arg\min_q KL(q\|p) \Rightarrow$ overconfident

2 Bayesian Regression

$w \sim N(0, \sigma_p^2 I)$, $\epsilon \sim N(0, \sigma_n^2 I)$, $y = Xw + \epsilon$
$y|w \sim N(Xw, \sigma_n^2 I)$
$w|y \sim N((X^TX + \lambda I)^{-1}X^Ty, (X^TX + \lambda I)^{-1}\sigma_n^2)$

3 Kalman Filter

$\begin{cases} X_{t+1} = FX_t + \epsilon_t & \epsilon_t \sim N(0, \Sigma_x) \\ Y_t = HX_t + \eta_t & \eta_t \sim N(0, \Sigma_y) \end{cases} X_1 \sim N(\mu_p, \Sigma_p)$
Then if X_0 is Gaussian then $X_t|Y_{1:t} \sim N(\mu_t, \sigma_t)$:
$\mu_{t+1} = F\mu_t + K_{t+1}(y_{t+1} - HF\mu_t)$
$\Sigma_{t+1} = (I - K_{t+1}H)(F\Sigma_t F^T + \Sigma_x)$
$K_{t+1} = (F\Sigma_t F^T + \Sigma_x)H^T(H(F\Sigma_t F^T + \Sigma_x)H^T + \Sigma_y)^{-1}$

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

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

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 $[K]_{ij} = k(x_i, x_j)$ holds $c^T K c \geq 0 \forall c$
Some Kernels: (h is the bandwidth hyperp.)
Gaussian (rbf): $k(x,y) = \exp(-\frac{\|x-y\|^2}{h^2})$
Exponential: $k(x,y) = \exp(-\frac{\|x-y\|}{h})$
Linear kernel: $k(x,y) = x^T y$ (here $K_{AA} = XX^T$)

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 $K_y(\theta) = K_{AA}(\theta) + \sigma_n^2 I$
$\hat{\theta} = \arg\max_\theta \log p(y|X;\theta)$
In GP: $\hat{\theta} = \arg\min_\theta y^T K_y^{-1}(\theta) y + \log|K_y(\theta)|$
We can from here $\nabla\downarrow$:
$\nabla_\theta \log p(y|X;\theta) = \frac{1}{2}tr\left((\alpha\alpha^T - K^{-1})\frac{\partial K}{\partial\theta}\right)$, $\alpha = K^{-1}y$
Or we could also be baysian about θ

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.
$\kappa(x - y) = \mathbb{E}_{p(w)}\left[\exp\left\{iw^T(x-y)\right\}\right]$ 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^TX vs XX^T)

Inducing points: We a vector of inducing variables u
$f_A|_u \sim N(K_{Au}K_u u^{-1}u, K_{AA} - K_{Au}K_u u^{-1}K_{uA})$
$f_*|_u \sim N(K_{*u}K_u u^{-1}u, K_{**} - K_{*u}K_u u^{-1}K_{u*})$

Subset of Regressors (SoR): $\blacksquare \to 0$
FITC: $\blacksquare \to$ 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 Variationa Inverence

$\hat{q} = \arg\min_{q\in Q} KL(q\|p(\cdot|y))$
$\hat{q} = \arg\max_{q\in Q} ELBO$ 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 postirior
Ergodic Markov Chain:
$\exists t$ s.t. $\mathbb{P}(i \to j$ in t steps$) > 0\ \forall i, j \Rightarrow$
$\exists!\pi = \lim_{N\to\infty}\mathbb{P}(X_n = x)$ Limit distribution

Ergodic Theorem: if $(X_i)_{i\in\mathbb{N}}$ is ergodic:
$\lim_{N\to\infty}\frac{1}{n}\sum_{i=1}^{N} f(X_i) = \mathbb{E}_{x\sim\pi}[f(x)]$

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 → 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* α **do**
    $X_t = x'$;
  otherwise $X_t = x_{t-1}$;

Metropolis Adj. Langevin Algo (MALA):
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 ∇f. Converges also without acceptance step
Hamilton MC: SGD performance improoved by adding momentum (consider last step ∇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**
  $x_t = x_{t-1}$
  with $i \sim \mathcal{U}(\{1:n\} \setminus B)*$ **do**
    $x_{t-1}^{(i)} \sim P(x^{(i)}|x^{(-i)})$

$*$ 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: $p(\theta) = \mathcal{N}(0, \sigma_p^2)$
$\theta_{MAP} = \arg\max\log(p(y,\theta))$

6.1 Variation inference:

Usually we use $Q =$ Set of Gaussians
$\hat{q} = \arg\max ELBO$ Reparameterization trick
q approx. the posterior but how to predict?
$p(y^*|x^*, \mathcal{D}) \simeq \frac{1}{m}\sum_{j=1}^{m} p(y^*|x^*, \theta^{(i)})$, $\theta \sim \hat{q}(\theta)$
Gaussian Mixture distribution: $\mathbb{V}(y^*|x^*, \mathcal{D}) \simeq$
$\simeq \frac{1}{m}\sum_{j=1}^{m}\sigma^2(x^*, \theta^{(i)}) + \frac{1}{m}\sum_{j=1}^{m}\left(\mu(x^*, \theta^{(j)} - \overline{\mu}(x^*))\right)$
$\blacksquare$ →Aletoric, $\blacksquare$ →Epistemic

Dropouts Regularization: Random ignore nodes in SGD iteration: Equavalent to VI with $Q = \left\{q(\cdot|\lambda) = \prod_j q_j(\theta_j|\lambda),\ \lambda \in \mathbb{R}^d\right\}$
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:
$\mu_i = \frac{1}{T}\sum_{j=1}^{T}\theta_i^{(j)}$ 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 θ_i^{MAP} on $\mathcal{D}_i$ with $i = 1, \dots, m$
$\mathcal{D}_i$ is a Bootstrap of $\mathcal{D}$ of same size
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 $\doteq \log p(y'_{1:m}|x'_{1:m}, \mathcal{D}_{train})$
$\geq \mathbb{E}_{\theta\sim\hat{q}}\left[\sum_{i=1}^{m}\log p(y'_i|x'_i, \theta)\right]$ (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: $F(\mathcal{S}) = \frac{1}{2}\log\left|I + \frac{1}{\sigma^2}K_{\mathcal{SS}}\right|$
This is NP Hard, $\Rightarrow$ Greedy Algo:

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

F is **Submodular** if: $\forall x \in \mathcal{D}$, $\forall A \subseteq B \subseteq D$ holds that: $F(A \cup \{x\}) - F(A) \geq F(B \cup \{x\}) - F(B)$
F is Submodular $\Rightarrow F(\mathcal{S}^*) \geq \left(1 - \frac{1}{e}\right) F(\hat{\mathcal{S}})$

8 Bayesian Optimization

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

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

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

8.1 Upper Confidence Sampling

With GP $x_t = \arg\max_{x \in \mathcal{D}} \mu_{t-1}(x) + \beta_t \sigma_{t-1}(x)$
Chosing the correct β_t we get: $\frac{R_T}{T} = \mathcal{O}\left(\sqrt{\frac{\gamma_T}{T}}\right)$.
Where $\gamma_t = \max_{|\mathcal{S}|<T} I(f; y_{\mathcal{S}})$. 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

$\mathcal{X} = \{1, \ldots, n\}$ states; $\mathcal{A} = \{1, \ldots, m\}$ actions;
$p(x'|x,a)$ transition probability;
$r(x,a)$ reward (can be random); $\pi : \mathcal{X} \to \mathcal{A}$ policy;
$T^\pi \in \mathbb{R}^{n \times n}$, $T^\pi_{ij} = p(j|i, \pi(i))$ Transition Matrix:
$J(\pi) = \mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^i r(X_i, \pi(X_i))\right]$ Expected value:
$V^\pi : \mathcal{X} \to \mathbb{R}$, $x \mapsto J(\pi | X_0 = x)$ 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) π^*, 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 Partialy Observable MDP (POMDP)

POMDP can be seen as MDP where:
1) $\mathcal{X}_{POMDP}$ 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:
$p(x'|x,a) \simeq \hat{p}_{x'|x,a} = \frac{Count(X_{t+1}=x', X_t=x, A_t=a)}{Count(X_t=x, A_t=a)}$
$r(x,a) \simeq \hat{r}_{x,a} = \frac{1}{Count(X_t=x, A_t=a)} \sum_{t|X_t=x, A_t=a} R_t$
How to chose a_t?

10.1.1 ϵ-greedy (On-Policy)

With probability ϵ, 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} algorithm (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: π = optimal policy w.r.t. p, r
repeat
Execute π 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 obeserved 'enough' in (x,a) recompute π 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 ϵ-optimal policy in a number of steps that is polynomial in $|X|, |A|, T, 1/\epsilon$ and $log(1/\delta)$.
Memory $O(|X|^2|A|)$.

10.2 Model-free RL

Learn π^* only via V^* or Q^{V^*}

10.2.1 TD-learning (On-Policy)

Given a policy π we want to learn V^π
$V^\pi(x) = \mathbb{E}_{R \sim r(x, \pi(x)), X' \sim p(\cdot|x, \pi(x))}[R + \gamma V^\pi(X')]$
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 α_t is a regulizer term (only 1 samlple)
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 $Q^* = Q^{V^*}$
$Q^*(x,a) = \mathbb{E}_{\substack{R \sim r(x,\pi(x)) \\ X' \sim p(\cdot|x,\pi(x))}}[R + \gamma \max_{a'} Q^*(X', a')]$
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 $O(|X||A|)$. Comp: $O(|A|)$.

11 Parametric RL

11.1 Parametric TD-learning

11.1.1 TD-learinging as SGD

TD-learing = 1 sample $(x', r | x, \pi(x))$ SGD on:
$\bar{l}_2(V; x, r) = \frac{1}{2}\left(V - r - \gamma \mathbb{E}_{x' \sim p(\cdot|x,\pi(x))}\left[\hat{V}^\pi(x')\right]\right)^2$
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-leaining

$\delta(\theta, \theta_{old}) = (Q(x,a;\theta) - r - \gamma \max_{a'} Q(x', a'; \theta_{old})$
We don't differiantiate with regard to θ_{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