(1)

- Supervised Learning
- Unsupervised Learning
- Semi-supervised learning

⇒ In supervised learning, we have loss functions such as l_{0-1} and corresponding risks

$$R_{0-1} = \mathbb{E}\left[l_{0-1}(f(x), y)\right]$$

---

* **Bayes Classifier** (Ideal classifier): → Pointwise minimizer (For 2 class classification ±1)

$$\eta(x) = P(y=1|x)$$

$$f^*(x) = \text{sign}(\eta(x) - 1/2) \rightarrow \text{Bayes classifier}$$

Show,

$$\Rightarrow R_{0-1}(g) \geq R_{0-1}(f^*) \quad \forall g \neq f^*$$

$$R_{0-1}(g) = \mathbb{E}\left[\mathbb{I}_{\{y \cdot g(x) < 0\}}\right]$$

$$= P(y \cdot g(x) < 0) \rightarrow \text{Probability of misclassification}$$

- Show, $R_{0-1}(g) - R_{0-1}(f^*) \geq 0$

$$\mathbb{E}\left[\mathbb{I}_{\{y \neq g(x)\}} - \mathbb{I}_{\{y \neq f^*(x)\}}\right] \geq 0$$

$$\Rightarrow \mathbb{E}_x\left[\mathbb{E}_{y|x}\left[\mathbb{I}_{\{y \neq g(x)\}}|x\right] - \mathbb{E}_{y|x}\left[\mathbb{I}_{\{y \neq f^*(x)\}}|x\right]\right] \geq 0$$

- $\mathbb{E}_{y|x}\left[\mathbb{I}_{\{y \neq g(x)\}}|x\right] = P\left[y \neq g(x)|x\right]$

$$= 1 - P\left[y = g(x)|x\right]$$

$$= 1 - P\left[y=1, g(x)=1|x\right]$$

$$= 1 - \mathbb{E}\left[\mathbb{I}_{\{y=1\}}\mathbb{I}_{\{g(x)=1\}}\,|\,x\right]$$
$$- \mathbb{E}\left[\mathbb{I}_{\{y=-1\}}\mathbb{I}_{\{g(x)=-1\}}|x\right]$$
$$= 1 - \mathbb{I}_{\{g(x)=1\}}\mathbb{E}\left[\mathbb{I}_{\{y=1\}}|x\right]$$
$$- \mathbb{I}_{\{g(x)=-1\}}\mathbb{E}\left[\mathbb{I}_{\{y=-1\}}|x\right]$$

Now,

① $\leftarrow \mathbb{E}_{y|x}\left[\mathbb{I}_{\{y\neq g(x)\}}|x\right] = 1 - \mathbb{I}_{\{g(x)=1\}}\cdot\eta(x)$
$- \mathbb{I}_{\{g(x)=-1\}}(1-\eta(x))$

② $\leftarrow \mathbb{E}_{y|x}\left[\mathbb{I}_{\{y\neq g^*(x)\}}|x\right] = 1 - \mathbb{I}_{\{g^*(x)=1\}}\cdot\eta(x)$
$- \mathbb{I}_{\{g^*(x)=-1\}}(1-\eta(x))$

$$① - ② = \eta(x)\left[\mathbb{I}_{\{g^*(x)=1\}} - \mathbb{I}_{\{g(x)=1\}}\right]$$
$$+ (1-\eta(x))\left[\mathbb{I}_{\{g^*(x)=-1\}} - \mathbb{I}_{\{g(x)=-1\}}\right]$$

$$① - ② = \eta(x)\left[\mathbb{I}_{\{g^*(x)=1\}} - \mathbb{I}_{\{g(x)=1\}}\right]$$
$$+ (\eta(x)-1)\left[\mathbb{I}_{\{g^*(x)=1\}} - \mathbb{I}_{\{g(x)=1\}}\right]$$

$$① - ② = (2\eta(x)-1)\left[\mathbb{I}_{\{g^*(x)=1\}} - \mathbb{I}_{\{g(x)=1\}}\right]$$

Case ①: $\eta(x) > 1/2$, $2\eta(x) - 1 > 0$

↙ Probability that $y = 1$ given x.

$$\mathbb{I}_{\{g^*(x)=1\}} = 1 \quad (\because \eta(x) > 1/2)$$

$\therefore$ ① - ② ≥ 0

$$\therefore \mathbb{E}_x \left[(2\eta(x) - 1) \left[\mathbb{I}_{\{g^*(x)=1\}} - \mathbb{I}_{\{g(x)=1\}} \right] \right] \geq 0$$

Case ②: $\eta(x) < 1/2$, $2\eta(x) - 1 < 0$

$$\mathbb{I}_{\{g^*(x)=1\}} = 0$$

$$\therefore \mathbb{I}_{\{g^*(x)=1\}} - \mathbb{I}_{\{g(x)=1\}} \leq 0$$

$$\therefore \mathbb{E}_x \left[(2\eta(x) - 1) \left[\mathbb{I}_{\{g^*(x)=1\}} - \mathbb{I}_{\{g(x)=1\}} \right] \right] \geq 0$$

$$\therefore R_{0-1}(g) - R_{0-1}(g^*) \geq 0$$

- All standard loss functions like log, exponential etc are classification calibrated (Achieve bayes like classifiers)
- Empirical vs Expected risk → Large number of samples required to reduce gap between the two.

(2)

Note: Underlying distribution is fixed but unknown in any ML related data source.

(We use Empirical risk (average) since we don't kno the true expected risk)

Online Learning: (One sample at a time)

- Stochastic Gradient descent (to update the hypoth

$$w_{t+1} = w_t - \eta \nabla_w L(f(x_t), y_t)$$

(For each datapoint)

*1) **Perceptron**: (Single sample algorithm)

- $g = \text{sign}(f(x)) = \text{sign}\left(\sum_{i=1}^{d} w_i x_i + b\right)$

$x = [x_1 \; x_2 \; \dots \; x_d]^T$

- Loss : $L_{\text{perceptron}}(f(x), y) = \max(0, -y \cdot f(x))$

(Not a 0-1 loss function)

- $w_{t+1} = \begin{cases} w_t + \eta y_t x_t & , y_t(w^T x_t + b) \le 0 \\ w_t & , y_t(w^T x_t + b) > 0 \end{cases}$

$b_{t+1} = \begin{cases} b_t + \eta y_t & , y_t(w^T x_t + b) \le 0 \\ b_t & , y_t(w^T x_t + b) > 0 \end{cases}$

Show,

$\Rightarrow y_t(w^T x_t + b_t) < y_t(w_{t+1} x_t + b_{t+1})$

$= y_t(w_t^T x_t + \eta y_t \|x_t\|^2 + b_t + \eta y_t)$

$= y_t(w_t^T x_t + b_t) + \eta(\|x_t\|^2 + 1) \cdot y_t^2$ → +ve value.

(After an update, the point is classified better)

Theorem 1: Finite convergence in linearly separable case.

** . Let $S = \{(x_1, y_1), \ldots, (x_T, y_T)\}$ be the examples in T iterations

(Assume bias = 0, can be done by $\tilde{w} = \begin{bmatrix} w \\ b \end{bmatrix}$, $\tilde{x} = \begin{bmatrix} x \\ 1 \end{bmatrix}$)

Ideal classifier (Assume)

- Let $\exists\ \hat{u} \in \mathbb{R}^d$ s.t $\|u\|_2 = 1$ and $y_t \cdot u \cdot x_t \geq \gamma$ $\forall\ t = 1 \ldots T$ & $\gamma > 0$ → Margin (Ideal)
- Let $R_2 = \max \|x_t\|$ ← Radius from which examples are taken.

$\Rightarrow$ Perceptron converges in $\left(\frac{R_2^2}{\gamma^2}\right)$ mistakes.

Proof: Let x_t gets misclassified, (u = Ideal classifier)

$$y_t(w_t x_t) < 0$$

$$w_{t+1} \cdot u = (w_t + y_t x_t) \cdot u$$

$$= w_t \cdot u + \underbrace{y_t x_t \cdot u}_{\geq \gamma}$$

$$\geq w_t u + \gamma$$

$$w_{t+1} \cdot u - w_t u \geq \gamma$$

- Suppose in T trials, k datapoints are misclassified

$$w_{T+1} \cdot u \geq k\gamma \quad (w_1 = 0)$$

($a^T b \leq \|a\| \|b\|$ } Cauchy Schwartz inequality)

$$\Rightarrow w_{T+1} \cdot u \leq \|u\| \cdot \|w_{T+1}\| = \|w_{T+1}\|$$

$$\Rightarrow \|w_{t+1}\|^2 = \|w_t + y_t x_t\|^2$$

$$= \|w_t\|^2 + \|x_t\|^2 + 2 y_t w_t x_t$$

$$\leq \|w_t\|^2 + R_2^2$$

($y_t w_t x_t < 0$)

$$\therefore \|w_{T+1}\|^2 \leq \|w_1\|^2 + k \cdot R_2^2$$

$$\|w_{T+1}\|^2 \leq k \cdot R_2^2$$ (k mistakes in T iterations)

Now, $\|w_{T+1}\| \geq w_{T+1} \cdot u \geq k\gamma$

$$\Rightarrow k\gamma \leq \|w_{T+1}\|$$

$$k^2\gamma^2 \leq \|w_{T+1}\|^2 \leq kR_2^2$$

$$\Rightarrow k \leq \frac{R_2^2}{\gamma^2}$$

} At most k misclassifications for convergence.

Note: Perceptron extension proofs (Multiclass etc))

---

* Note: $k = \frac{R_2^2}{\gamma^2}$ for examples like $x_i = (0, 0, \ldots \underset{\text{ith dim}}{1}, 0)$ and u_i here would be $\frac{y_i}{\|y_i\|}$ (i^{th} data point)

(Here $k = n$ iterations), $R_2^2 = 1$, $\gamma^2 = \frac{1}{\|u\|^2}$

Theorem 2: $S = \{(x_1, y_1), \ldots, (x_T, y_T)\}$ (Finite convergence in ~~[illegible]~~ non linearly seperable case)

** $(x_i, y_i) \in \mathbb{R}^d \times \{-1, 1\}$ $\forall i$

- $R_2 = \max_t \|x_t\|_2$, Let $\gamma > 0$

Then for any $u \in \mathbb{R}^d$ $(\|u\|_2 = 1)$

the number of mistakes ← Allowed error

$$k \leq \frac{(R_2 + D)^2}{\gamma^2}$$

$$\Rightarrow \left(D^2 = \sum_{t=1}^{T} d_t^2 \text{ and } d_t = \max\left[0, \gamma - y_t u x_t\right] \right.$$

- Classifier wont be able to classify perfectly in current space, so we project data.

$x_t \rightarrow x_t'$ where,

$$x_t' = \underbrace{[x_{t,1}\ x_{t,2}\ \ldots\ x_{t,d}\ 0\ \ldots\ 0}_{d+T \text{ dimension}}\ \underbrace{\Delta\ 0\ \ldots\ 0]}_{(d+t)^{th} \text{ component}}$$

- $u' = \left[\frac{u_1}{z}\ \frac{u_2}{z}\ \cdots\ \frac{u_d}{z}\ \frac{y_1 d_1}{\Delta \cdot z}\ \ldots\ \frac{y_T d_T}{\Delta z}\right]$

Set, $z = \sqrt{1 + \frac{D^2}{\Delta^2}}$ } when we equate $\|u'\|_2 = 1$.

- $y_t u' x_t' = y_t \left[\frac{u x_t}{z} + \frac{\Delta \cdot y_t d_t}{\Delta z}\right]$

$$= \frac{y_t \cdot u \cdot x_t}{z} + \frac{d_t}{z}$$

$$\geq \frac{y_t \cdot u \cdot x_t}{z} + \frac{\gamma - y_t \cdot u \cdot x_t}{z}$$

$y_t u' x_t' \geq \gamma/z$ ... ①

$\Rightarrow \|x_{t+1}\|^2 = \|x_t\|^2 + \Delta^2 \leq R_2^2 + \Delta^2$ ... (2)

(Using (1) & (2), similar to Theorem 1, we can get the two inequalities comparing which we get k)

$k \leq \frac{(R_2^2 + \Delta^2) \cdot z^2}{\gamma^2}$, Substituting z

$k \leq \frac{(R_2^2 + \Delta^2)(1 + D^2/\Delta^2)}{\gamma^2}$

$(\Delta = \sqrt{R_2 D}) \rightarrow$ (To minimize k, differentiate wrt Δ)

$(\because w_{t+1}' u' - w_t' u' \geq \gamma |z|)$

$(\because \|w_{t+1}'\|^2 = \|w_t'\|^2 + R_2^2 + \Delta^2)$

(3)

*SVM Recap:

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i$$

$$f(x) = w^T x + b$$

$$= \sum_{i=1}^{N} \alpha_i y_i (x_i^T x) + b$$

Primal: (with slack)

$$\min_{w,b,\xi} \frac{1}{2}\|w\|_2^2 + C \cdot \sum_{i=1}^{N} \xi_i$$

$$s.t \quad y_i(w^T x_i + b) \geq 1 - \xi_i \quad \forall i$$

$$\xi_i \geq 0 \quad \forall i$$

$\Downarrow$

Dual: (with slack)

$$\max_{\alpha_1 \ldots \alpha_N} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} y_i y_j \alpha_i \alpha_j (x_i^T x_j)$$

$$s.t \quad 0 \leq \alpha_i \leq C \quad \forall i$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0$$

$\alpha^T \cdot \mathbb{1} - \frac{1}{2}(z \odot y)^T G (z \odot y)$

- Replace $x_i^T x_j$ with $\phi(x_i, x_j)$, some kernel function to project non linearly seperable data onto higher dimensions.

⇒ Perceptron:

(details later)

$w^1 = 0$

$w = \sum_{j \in M} y_i x_i$, $f(x) = w^T x + b$

where $M = \{t \in \{1, \ldots, T\} \mid y_t w_t x_t \leq 0\}$

- This is similar to the SVM and we can use the kernel trick here as well.

Kernel function:

$K(x_1, x_2) = \phi(x_1)^T \cdot \phi(x_2)$

---

*Kernels:

- Linear kernel ⇒ $K(x_1, x_2) = x_1^T x_2$
- Polynomial kernel ⇒ $K(x_1, x_2) = (x_1^T x_2 + 1)^d$

ex: $K(x_1, x_2) = (x_{11} x_{21} + x_{12} x_{22} + 1)^2$

$= \phi(x_1)^T \cdot \phi(x_2)$

where $\phi(x_i) = [x_{i1}^2, x_{i2}^2, \sqrt{2} x_{i1}, \sqrt{2} x_{i2}, \sqrt{2} x_{i1} x_{i2}, 1]$

- Gaussian Kernel ⇒ $K(x_1, x_2) = \exp\left(\frac{-\|x_1 - x_2\|_2^2}{2\sigma^2}\right)$

↓ Projects to ∞ dimensions

(These kernels are all symmetric)

Gram Matrix: (A similarity matrix for given data)

- $G_{ij} = K(x_i, x_j)$ for a given set of vectors $\{x_1, \ldots x_n\}$

→ Positive Semidefinite Kernels:

- A symmetric kernel function is finite p.s.d if gram matrices formed by any finite subset of X is positive semi definite.

* Mercer's Theorem:

- If kernel function is psd there exists a map $\phi: X \to H$ → Hilbert space s.t $K(x_1, x_2) = \phi(x_1)^T \cdot \phi(x_2)$

---

* Kernel based Perceptrons:

- $w = \sum_{i \in M} y_i x_i$ where $M = \{t \in [1, \ldots, T] \mid$ $y_t w_t x_t \leq 0$

(Assume no bias, $w \leftarrow \begin{bmatrix} w \\ b \end{bmatrix}, x \leftarrow \begin{bmatrix} x \\ 1 \end{bmatrix}$)

$$\Rightarrow w = \sum_{k=1}^{N} \alpha_k \cdot y_k \cdot \phi(x_k)$$

↑ weight for training sample (Initially all α_k's are zero)

Update function:

i th sample misclassified $x_i \to x_{i+1}$

- If $\text{sign}(w^T \phi(x_i)) \neq y_i$,

$$w = w + y_i \phi(x_i)$$

$$\therefore w = \sum_{k \neq i} \alpha_k y_k \phi(x_k) + (\alpha_i + 1) y_i \phi(x_i)$$

$\Rightarrow$ Essentially, if $\text{sign}\left(\sum_{k=1}^{N} \alpha_k y_k \phi(x_k)^T \cdot \phi(x_i)\right) \neq y_i$

then $\alpha_i = \alpha_i + 1$

⇒ For $(t=1$ to $T)$ → Choose sample (x_t, y_t) randomly.

If $\{y_t\left(\sum_{k=1}^{t-1} \alpha_k y_k - K(x_k, x_t)\right) \leq 0$

$\alpha'_t \rightarrow \alpha_t + 1$

else

$\alpha'_t \rightarrow \alpha_t$

⇒ Find classifier, $f(x) = \sum_{t=1}^{T} \alpha_t y_t K(x_t, x)$

• Convergence of Kernel Perceptron:

* i) Separable case:

$y_t K(u, x_t) \geq \gamma \quad \forall t$

$R_2^2 = \max_{t=1 \to T} K(x_t, x_t), \quad \|\phi(u)\|_2 = 1$

• At most $\frac{R_2^2}{\gamma^2}$ mistakes

(Proof similar to non kernel version)

Note: Perceptron vs SVM, Perceptron just finds any classifier which separates train data well but SVM tries to maximize the margin. (we can start including margins in perceptron as well)

Note: Passive aggressive perceptron algorithms

(2006 JMLR Online Passive Aggressive)

—x—

(4) Learning from experts:

* $R_T = \sum_{t=1}^{T} L(\hat{y}_t, y_t) - \min_{i=1,\dots,n} \sum_{t=1}^{T} L(\hat{y}_t^i, y_t)$

* ↓ Minimize regret

Loss made by the best expert

$\hat{y}_t$ = Combined experts prediction

* Halving algorithm: (Assuming one of the experts is always right)

1) • All experts have weight of 1
   • Use majority vote as combined prediction $\hat{y}_t$
   • For each expert where prediction is incorrect ($\hat{y}_t$) incur loss with label y_t and remove experts if prediction doesn't match y_t, make weight zero.

$\Rightarrow C_1 = n$ (Number of experts)

if $\hat{y}_t \neq y_t$

then $C_{t+1} \leq \frac{1}{2} C_t$

• $\frac{C_{T+1}}{C_1} \leq \left(\frac{1}{2}\right)^k$ → Number of mistakes in T trials

$2^k \leq n$

$k \leq \log_2 n$ } upper bound of number of mistakes.

$\Rightarrow R_T = \sum_{t=1}^{T} L_{0-1}(\text{Halving}(x_t), y_t)$

$\therefore R_T = k$ — 0

$\Rightarrow R_T \leq \log_2 n$

↑ One expert is always right.

(Algo in slides)

2) Weighted majority algorithm:

(No expert needs to be always right)

* Experts making mistakes have their weights decreased. Weights are all 1's initially.

$$w_i^{t+1} = w_i^t \cdot \exp\left(-\eta L_{0\text{-}1}(\varepsilon_i^t, y_t)\right)$$

↑ Perform updates if $\hat{y}_t \neq y_t$

($\eta > 0$ is a parameter.)

(This is to ensure small errors by experts are ignored since no expert is always right)

Theorem: $m_T^* =$ Number of mistakes by best expert in T trials.

Then $m_T \leq \dfrac{\eta m_T^* + \ln(N)}{\ln\left(\frac{2}{1+e^{-\eta}}\right)}$

(N → Number of experts)

(m_T → Mistakes made by weighted majority in T trials.)

Proof:

⇒ Let $\hat{y}_t \neq y_t$

then $W^{t+1} = \sum_{j=1}^{n} w_j^{t+1}$

$$= \sum_{j=1}^{n} w_j^t \cdot e^{-\eta L_{0\text{-}1}(\varepsilon_j^t, y_t)}$$

$$= \sum_{j:\varepsilon_j^t = y_t} w_j^t + \sum_{j:\varepsilon_j^t \neq y_t} w_j^t \cdot e^{-\eta}$$

($\sum w_j^t$ is the majority, $\varepsilon_j \neq y_t$)

$\because \hat{y}_t \neq y_t$

$$= W_{\text{minority}} \cdot e^{-\eta} \cdot W_{\text{majority}}$$

We can upper bound it. ← $W^{t+1} \leq W_{\text{minority}} + e^{-\eta} W_{\text{majority}}$

$$+ (W_{\text{majority}} - W_{\text{minority}}) \frac{(1-e^{-\eta})}{2}$$

the term.

$$\Rightarrow W^{t+1} \leq \frac{1+e^{-\eta}}{2} \cdot W^t$$

$$\Rightarrow \frac{W^{t+1}}{W^t} \leq \frac{1+e^{-\eta}}{2}$$

$$\Rightarrow \frac{W^{T+1}}{W^1} = \prod_{t=1}^{T} \frac{W^{t+1}}{W^t}$$, Let m_T be mista in T trials thus,

$$\frac{W^{T+1}}{W^1} \leq \left(\frac{1+e^{-\eta}}{2}\right)^{m_T}$$

η ↙

$$\therefore W^{T+1} \leq K \cdot \left(\frac{1+e^{-\eta}}{2}\right)^{m_T} \rightarrow (1)$$

$$\Rightarrow \text{Now, } W^{T+1} = \sum_{i=1}^{n} w_i^{T+1} \geq \max_{i \in \{1 \ldots n\}} w_i$$

$$\Rightarrow W^{T+1} \geq \max_{i \in \{1 \ldots n\}} \left(w_i^1 \cdot e^{-\eta \sum_{t=1}^{T} L_{0-1}(\hat{\epsilon}_i^t, y}\right.$$

$$W^{T+1} \geq e^{-\eta \left(\min_{i \in \{1 \ldots n\}} \sum_{t=1}^{T} L_{0-1}(\hat{\epsilon}_i^t, y\right.}$$

m_T^* (Number of mistak made by bes experts

↙ This is a correct bound even with only k updates

$$W^{T+1} \geq e^{-\eta \cdot m_T^*} \rightarrow (2)$$

- From (1), (2)

$$\Rightarrow e^{-\eta \cdot m_T^*} \leq n \left(\frac{1+e^{-\eta}}{2}\right)^{m_T}$$

$$\Rightarrow -\eta m_T^* \leq \ln(n) + m_T \ln\left(\frac{1+e^{-\eta}}{2}\right)$$

$$\Rightarrow m_T \ln\left(\frac{2}{1+e^{-\eta}}\right) \leq \ln(n) + \eta m_T^*$$

$$\Rightarrow m_T \leq \frac{\ln(n) + \eta m_T^*}{\ln\left(\frac{2}{1+e^{-\eta}}\right)}$$

---

(5) ~~Winnow Algorithm:~~

3) Exponential Weighted Average Algorithm:

**
- Assume prediction $\hat{y}$ is from a convex set.
- Use a generalized loss: $\frac{|\hat{y}_t - y_t|}{2}$ → Bounded b/w 0 to 1
- Prediction using, $\hat{y}_t = \sum_{i=1}^{n} p_i^t \cdot \varepsilon_i^t$

  where $p_i^t = \frac{w_i^t}{\sum_{i=1}^{n} w_i^t}$

- Weight update (Update every iteration) $w_i^{t+1} = w_i^t \cdot \exp(-\eta L(\varepsilon_i^t, y_t))$

$$\Rightarrow R_T = \sum_{t=1}^{T} L(\hat{y}_t, y_t) - \min_{i \in \{1 \ldots n\}} \sum_{t=1}^{T} L(y_t, \varepsilon_i^t)$$

Theorem: $\sum_{t=1}^{T} L(\hat{y}_t, y_t) \leq \min_{i \in \{1 \ldots n\}} \sum_{t=1}^{T} L(\varepsilon_i^t, y_t) + \frac{\ln n}{\eta} + \frac{T\eta}{8}$

$$R_T \leq T\eta/8 + \frac{\log n}{\eta}$$

(Proof on the next page) →

Convex set: Let $X \subseteq \mathbb{R}^d$, X is convex if $\forall x_1, x_2 \in X$, $\lambda x_1 + (1-\lambda) x_2 \in X$ s.t $0 \leq \lambda \leq 1$

Proof: $W_{t+1} = \sum_{i=1}^{n} w_i^{t+1}$

$$* \Rightarrow \frac{W_{t+1}}{W_t} = \sum_{i=1}^{n} \frac{w_i^t}{W_t} \cdot e^{-\eta L(\xi_i^t, y_t)}$$

$$= \sum_{i=1}^{n} p_i^t \cdot e^{-\eta L(\xi_i^t, y_t)}$$

$$= \mathbb{E}_I \left[e^{-\eta L(\xi_I, y_t)} \right]$$

- We know $\mathbb{E}[e^{\alpha x}] \leq e^{\alpha \mu} \cdot e^{\alpha^2/8}$

?(Hoffding Lemma) $\mu = \mathbb{E}[X]$

$0 \leq x \leq 1$

$\Rightarrow$ Our $L(\xi_I, y_t)$ is bounded between 0, 1 so we can apply Hoffding

$$\therefore \frac{W_{t+1}}{W_t} \leq e^{-\eta \mathbb{E}_I(L(\xi_I, y_t))} \cdot e^{\eta^2/8}$$

- ~~Now, $\mathbb{E}_I[L(\xi_I, y_t)] = \sum_{i=1}^{n}$~~

$$\prod_{t=1}^{T} \frac{W_{t+1}}{W_t} \leq e^{-\eta \cdot \sum_{t=1}^{T} \mathbb{E}_{I_t}[L(\xi_{I_t}, y_t)]} \cdot e^{\frac{T\eta^2}{8}}$$

- Log on both sides,

$$\log W_{T+1} - \log W_1 \leq \frac{T\eta^2}{8} - \eta \sum_{t=1}^{T} \mathbb{E}_{I_t}$$

$\because W_1 = n$, $\log W_{T+1} - \log n \leq \frac{T\eta^2}{8} - \eta \sum_{t=1}^{T} \mathbb{E}_{I_t}[L$

- From Jensenn's inequality, (for convex functions)

$$f(\mathbb{E}[z]) \leq \mathbb{E}[f(z)]$$

$\therefore f(\sum p_i z_i) \leq \sum p_i f(z_i)$

$\Rightarrow$ Going back,

$$\log W_{T+1} - \log n \leq \frac{T\eta^2}{8} - \eta \sum_{t=1}^{T} L\left(\mathbb{E}_{I_t}[\varepsilon_{I_t}], y_t\right)$$

$$\rightarrow \mathbb{E}_{I_t}[\varepsilon_{I_t}] = \sum_{i=1}^{n} p_i^t \cdot \varepsilon_i^t = \hat{y}_t$$

$$\Rightarrow \log W_{T+1} - \log n \leq \frac{T\eta^2}{8} - \eta \sum_{t=1}^{T} L(\hat{y}_t, y_t) \rightarrow (1)$$

Now, $\Rightarrow \log W_{T+1} = \log \sum_{i=1}^{n} w_i^{T+1} \geq \max_{i \in \{1 \ldots n\}} \log w_i^{T+1}$

$$\log W_{T+1} \geq \max_{i \in \{1 \ldots n\}} \log w_i^T \cdot e^{-\eta L(\varepsilon_i^t, y_t)}$$

$$= \max_{i \in \{1 \ldots n\}} \log \left(w_i' \cdot e^{-\eta \sum_{t=1}^{T} L(\varepsilon_i^t, y_t)} \right)$$

$$= \log e^{-\eta \min_i \sum_{t=1}^{T} L(\varepsilon_i^t, y_t)}$$

$$= -\eta \min_{i \in \{1 \ldots n\}} \sum_{t=1}^{T} L(\varepsilon_i^t, y_t) \rightarrow (2)$$

- From (1), (2),

$$-\eta \min_{i \in \{1 \ldots n\}} \sum_{t=1}^{T} L(\varepsilon_i^t, y_t) - \log n \leq \frac{T\eta^2}{8} - \eta \sum_{t=1}^{T} L(\hat{y}_t, y_t)$$

$$\Rightarrow R_T \leq \frac{T\eta}{8} + \frac{\log n}{\eta}$$

Note: Doubling trick to find optimal η in online learning.
(To minimize regret bound)

Note: Winnow Algorithm:

- Based on weighted majority
- Each feature can be thought of as an expert
- Positive weight for each feature
- Useful when there are a few distinguishing features but we don't know which.

$\Rightarrow$ Initial weights are $1/N$ per expert.

$$\hat{y}_t = \text{sign}\left[(w^t)^T \cdot x_t\right]$$

$$y_t \in \{+1, -1\}$$

$$\Rightarrow \text{Loss} = L_{0-1}(\hat{y}_t, y_t)$$

- If $\hat{y}_t \neq y_t$, update,

$$w_i^{t+1} = \frac{w_i^t \cdot \exp(+\eta y_t x_i^t)}{Z_t}$$

If signs of x_i is same as y_t its weight increases

where $Z_t = \sum_{j=1}^{n} w_j^t \cdot \exp(\eta y_t x_j^t)$

(6) **Theorem:**

$y_i \in \{+1, -1\}$

$x_i \in \mathbb{R}^N$

$R_\infty = \max_{t \in \{1 \ldots T\}} \|x^t\|_\infty$ → Sup Norm

$$\|x^t\|_\infty = \max_{i \in \{1 \ldots N\}} |x_i^t|$$

- Assume γ such that $y_t \cdot u^T x^t \geq \gamma$ → Ideal weight
- $u \in R^N$, $u_i \geq 0 \ \forall i \in \{1 \ldots N\}$

$\|u\|_1 = \sum_{i=1}^{N} u_i$ (since all $u_i \geq 0$)

$$\Rightarrow k \leq \frac{\|u\|_1 \ln N}{\eta\gamma - \|u\|_1 \ln\left(\frac{\exp(\eta R_\infty) + \exp(-\eta R_\infty)}{2}\right)}$$

k ← Mistake bound

(optimizing wrt η)

$$k \leq \frac{2 R_\infty^2 \|u\|_1^2}{\gamma^2} \ln N$$

Proof: $\Rightarrow \bar{p} = \frac{u}{\|u\|_1}$ } ← The weight we want to match (Ideal weight)

* We want to minimize KL divergence,

Consider, $KL(\bar{p}, \bar{w}^t) - KL(\bar{p}, \bar{w}^{t+1})$

$$= \sum_{i=1}^{N} p_i \ln \frac{p_i}{w_i^t} - \sum_{i=1}^{N} p_i \ln \frac{p_i}{w_i^{t+1}}$$

$$= \sum_{i=1}^{N} p_i \ln \frac{w_i^{t+1}}{w_i^t} = \sum_{i=1}^{N} p_i \ln \frac{e^{\eta y_t x_i^t}}{Z_t}$$

$$= \sum_{i=1}^{N} p_i \left[\eta y_t x_i^t - \ln Z_t\right]$$

$$= \eta y_t \sum_{i=1}^{N} p_i x_i^t - \ln Z_t$$

$$= \eta y_t \bar{p} \cdot \mathbf{x}_t - \ln Z_t$$

$$= \eta\, y_t \cdot \frac{u}{\|u\|_1} \cdot x_t - \ln Z_t$$

From our assumption, $y_t \cdot u^T \cdot x^t \geq \gamma$

$$\therefore KL(\bar{p}, \bar{w}^t) - KL(\bar{p}, \bar{w}^{t+1}) \geq \frac{\eta\gamma}{\|u\|_1} - \ln Z_t \quad \rightarrow (1)$$

$\Rightarrow$ Now, $Z_t = \sum_{i=1}^{N} w_i^t e^{\eta y_t x_i^t}$

$$= \sum_{i=1}^{N} w_i^t \cdot e^{\left[\left(\frac{1+\eta y_t x_i^t}{2}\right) R_{00} + \left(\frac{1-\eta y_t x_i^t}{2}\right)\right]}$$

$$Z_t = \sum_{i=1}^{N} w_i^t \cdot e^{\eta\left[\left(\frac{1+(y_t x_i^t / R_{00})}{2}\right) R_{00} + \left(\frac{1-(y_t x_i^t / R_{00})}{2}\right)(-R_{00})\right]}$$

From, $f(\lambda z_1 + (1-\lambda) z_2) \leq \lambda f(z_1) + (1-\lambda) f(z_2)$ we can say

$$Z_t \leq \sum_{i=1}^{N} w_i^t \left[\frac{1 + \frac{y_t x_i^t}{R_\infty}}{2} \cdot e^{\eta R_\infty} + \frac{1 - \frac{y_t x_i^t}{R_\infty}}{2} \cdot e^{-\eta R_\infty} \right]$$

$$Z_t \leq \sum_{i=1}^{t} w_i^t \left(\frac{e^{\eta R_\infty} + e^{-\eta R_\infty}}{2} \right) + \left(\frac{e^{\eta R_\infty} - e^{-\eta R_\infty}}{2} \right) \cdot y_t \sum_{i=1}^{N} w_i^t$$

—ve when misclassified

$$\therefore Z_t \leq \sum_{i=1}^{t} w_i^t \left(\frac{e^{\eta R_\infty} + e^{-\eta R_\infty}}{2} \right)$$

$$Z_t \leq \frac{e^{\eta R_\infty} + e^{-\eta R_\infty}}{2} \rightarrow (2) \quad (\because \text{Weights sum up to 1})$$

From (1) & (2)

$$\therefore KL(\bar{P}, \bar{W}_x^t) - KL(\bar{P}, \bar{W}_x^{t+1}) \geq \frac{\eta\gamma}{\|u\|_1} - \ln\left(\frac{e^{\eta R_\infty} + e^{-\eta R_\infty}}{2}\right) \rightarrow (3)$$

- If the algo makes k mistakes in T trials

$$KL(\bar{P}, \bar{W}^1) - KL(\bar{P}, \bar{W}^{T+1}) \geq k\left(\frac{\eta\gamma}{\|u\|_1} - \ln\left(\frac{e^{\eta R_\infty} + e^{-\eta R_\infty}}{2}\right)\right)$$

Now, $KL(\bar{P}, \bar{W}^1) = \sum_{i=1}^{N} p_i \ln p_i - \sum p_i \ln \frac{1}{N}$

$$= \underbrace{\sum p_i \ln p_i}_{-ve} + \ln N \leq \ln N$$

$$\therefore \Rightarrow \ln N \geq k\left(\frac{\eta\gamma}{\|u\|_1} - \ln\left(\frac{e^{\eta R_\infty} + e^{-\eta R_\infty}}{2}\right)\right)$$

Now, differentiate wrt η to get η^*

$$\Rightarrow \eta^* = \frac{1}{2R_{\infty}} \ln\left(\frac{R_{\infty}\|u\|_1 + \gamma}{R_{\infty}\|u\|_1 - \gamma}\right)$$

$$\Rightarrow k \leq \frac{\ln N}{g\left(\frac{\gamma}{R_{\infty}\|u\|_1}\right)} \rightarrow (1)$$

where $g(\epsilon) = \frac{(1+\epsilon)}{2}\ln(1+\epsilon) + \frac{(1-\epsilon)}{2}\ln(1-\epsilon) \geq \frac{\epsilon^2}{2}$

ϵ ?

Should $\epsilon \leq 1$?

Note: Cauchy Schwartz: $a^T b \leq \|a\|_2 \cdot \|b\|_2$

extended $a^T b \leq \|a\|_1 \cdot \|b\|_\infty$

Now, $\gamma \leq y_t u^T x_t \leq \|u\|_1 \cdot \|y_t x_t\|_\infty$

$\leq \|u\|_1 \cdot R_\infty$

$\Rightarrow \frac{\gamma}{\|u\|_1 \cdot R_\infty} \leq 1$ } $\therefore$ This can be put in $g(\epsilon)$

$\therefore$ From (1), $k \leq \frac{2\ln N}{\frac{\gamma^2}{R_\infty^2 \cdot \|u\|_1^2}}$

$\Rightarrow k \leq \frac{2R_\infty^2 \|u\|_1^2}{\gamma^2} \cdot \ln N$

---x---

* Online Gradient Descent: (General)

* General gradient descent methods where we use a sample at a time for inputs.

$$\Rightarrow \tilde{x}^{t+1} = x_t - \eta \nabla C_t(x_t)$$

(C_t: cost function can change every epoch)

$$* \; x^{t+1} = P_\Omega(\tilde{x}^{t+1})$$

where $P_\Omega(\tilde{x}^{t+1}) = \min_{x \in \Omega} \|x - \tilde{x}^{t+1}\|_2^2$

(Ω is our search space, so if $\tilde{x}^{t+1}$ is outside search space, we adjust it accordingly)

Here, $R_T = \sum_{t=1}^{T} C_t(x_t) - \min_{x \in \Omega} \sum_{t=1}^{T} C_t(x)$

(Best possible x in entire search space)

$$\Rightarrow R_T \leq \frac{1}{2}\left(\frac{D^2}{\eta} + \eta G^2 T\right)$$

(Slides for details)

- Setting $\eta^* = \frac{D}{G\sqrt{T}}$

$$R_T \leq \frac{1}{2} DG\sqrt{T}$$

$\Rightarrow$ Since C_t is convex, $C_t(x) - C_t(x_t) \geq \nabla C_t(x_t)^T (x -$

$$\Rightarrow C_t(x_t) - C_t(x) \leq \nabla C_t(x_t)^T (x_t - x)$$

$$? \quad = \frac{x_t - \tilde{x}^{t+1}}{\eta}(x_t - x)$$

$$= \frac{1}{2\eta}\left[\|x - x^t\|^2 - \|x - \tilde{x}^{t+1}\|^2 + \|x^t - \tilde{x}^{t+1}\|^2\right.$$

Now, $\|x - \tilde{x}^{t+1}\|_2 \geq \|x - x^{t+1}\|_2$

$\forall x \in \Omega$

$$C_t(x_t) - C_t(x) \leq \frac{1}{2\eta}\left(\|x - x^t\|^2 + \|x - x^{t+1}\|^2 + \|x^t - \tilde{x}^{t+1}\|^2 \right)$$

$$= \frac{1}{2\eta}\left[\|x - x^t\|^2 - \|x - x^{t+1}\|^2 + \eta^2 \|\nabla C_t(x_t)\|^2 \right]$$

$$\therefore \sum_{t=1}^{T} C_t(x_t) - \sum_{t=1}^{T} C_t(x) \leq \frac{1}{2\eta}\left[\|x - x^1\|^2 - \|x - x^{T+1}\|^2 + \eta^2 \cdot G^2 \cdot T \right]$$

$$\left(G = \|\nabla C_t(x_t)\| \right) \qquad \leq \frac{1}{2\eta}\left[D^2 + \eta^2 G^2 T \right]$$

Note: $a^T b \leq \|a\|_p \|b\|_q$, $\frac{1}{p} + \frac{1}{q} = 1$

ex: $a^T b \leq \|a\|_1 \|b\|_\infty$

*Multi-Armed Bandits:

(7)

- n arms, t denotes trial

⇒ x_i^t is the reward obtained by pulling arm i on trial t

- $x_i^t \in [0,1]$ } Bounded , $x_i^t \sim q_i$ } Drawn from distribution q_i

⇒ Given $(i_1, x_{i_1}^1), (i_2, x_{i_2}^2), \ldots, (i_{t-1}, x_{i_{t-1}}^{t-1})$ we need to select arm $i_t \in \{1 \ldots n\}$ to play.

(Arm pulled at 1st play; Reward received)

* • Regret $R_T(A) = \left(\max_{i \in \{1 \ldots n\}} G_T(i)\right) - G_T(A)$

(MAB Algorithm; Arm selected by MAB)

Expected regret $= \mathbb{E}_{(x^1, \ldots x^T)}[R_T(A)]$ $G_T(i) = \sum_{t=1}^{T} x_i^t$

⇒ Let $\mu_i = \mathbb{E}_{x_i \sim q_i}[x_i]$ } Mean reward of arm i

$$\mu^* = \max_{i \in \{1..n\}} \mu_i, \quad \Delta_i = \mu^* - \mu_i$$

- Expected regret is hard to handle, so we use pseudo regret

* $\tilde{R}_T(A) = \max_{i \in \{1..n\}} \mathbb{E}_{(x^1, \ldots, x^T)}\left[\sum_{t=1}^{T} x_i^t - \sum_{t=1}^{T} x_{i_t}^t\right]$

Note: Show pseudo regret is upper bounded by expected regret

$\sum_{t=1}$

(i_t = Arm chosen by MAB algo at time t)

$$\tilde{R}_T(A) = T\mu^* - \sum_{t=1}^{T} \mathbb{E}_{(x^1, \ldots x^{T-1})}\left[\mathbb{E}_{(x^T | x^1 \ldots x^{T-1})}[x_{i_t}^t | x^1 \ldots x^{T-1}]\right]$$

$$\Rightarrow \tilde{R}_T(A) = T\mu^* - \sum_{t=1}^{T} \mathbb{E}_{(x^1,\ldots,x^{T-1})} \cdot \mathbb{E}_{x^T}\left[x_{i_t}^*\right]$$

(∵ Arms draw rewards independent of past here)

$$= T\mu^* - \sum_{t=1}^{T} \mathbb{E}_{(x^1,\ldots,x^{T-1})}\left[\mu_{i_t}\right]$$

$$\tilde{R}_T(A) = T\mu^* - \sum_{t=1}^{T} \mathbb{E}\left[\mu_{i_t}\right]$$

---

* **Action Values:** $\hat{\mu}_i^t$ = Estimated mean for arm i at epoch t (known as action values)

$$\left(\hat{\mu}_i^t = \frac{\sum_{s=1}^{t} x_i^s \cdot \mathbb{I}\{i_s = i\}}{\sum_{s=1}^{t} \mathbb{I}\{i_s = i\}}\right)$$

⇒ A simple greedy way of arm selection is select $i_t = \arg\max_{i \in \{1,\ldots,n\}} \hat{\mu}_i^{t-1}$

↳ This will never perform exploration though.

- This $\hat{\mu}_i^{t+1}$ can be derived just based on previous estimate (proof in slides)

? * $\hat{\mu}_i^{t+1} = \hat{\mu}_i^t$

* 1) **ϵ-Greedy Approach:**
  - In each trial, select best arm with prob $1-\epsilon$ & randomly select with prob ϵ.

(Tradeoff b/w exploration & exploitation)

* 2) **Upper confidence bound: (UCB)**
  - At each interval, choose arm with largest UCB
  - UCB of an arm can be high if its expected mean is high or it hasn't been sampled enough recently.

Note: $X \sim [0,1]$, $\mu = \mathbb{E}[X] = \left(\mathbb{E}_{x_i \sim \delta_i}[x_i] = \mu_i\right)$

* $\hat{\mu}_N = \frac{\sum_{i=1}^{N} x_i}{N}$ → From Hoffding Lemma!

$$\Rightarrow P\left[|\mu - \hat{\mu}_N| > t\right] \leq 2e^{-2Nt^2}$$

* (As number of trials increase, expected average goes closer to true average)

(Proof next page)

- Initially bounds → confidence interval are large, since we don't have a lot of info but as we perform more trials it gets better.

$$\Rightarrow \text{Select arm } i_t \in \arg\max_{i \in \{1,\dots,n\}} \left[\hat{\mu}_i^{t-1} + \sqrt{\frac{\alpha \ln t}{2N_i^{t-1}}}\right]$$

* ↑ We will prove why we use this later

(8) N_i^t = Number of times arm i is pulled in t trials.

~~$\Rightarrow P\left[\mu_i - \hat{\mu}_i^t > \epsilon\right]$~~

~~$\Rightarrow P\left[\mathbb{E}\left[\frac{1}{s}\sum_{k=1}^{s} x_i^k\right] - \frac{1}{s}\sum_{k=1}^{s} x_i^k\right]$~~

Note: (Assuming arm is pulled always else use indicators)

$$\mathbb{E}\left[\frac{1}{s}\sum_{k=1}^{s} x_i^k\right] = \frac{1}{s}\sum_{k=1}^{s}\mathbb{E}\left[x_i^k\right] = \frac{1}{s}\sum_{k=1}^{s}\mu_i = \mu_i$$

$$\Rightarrow \sum_{s=1}^{t-1} \mathbb{I}_{\{i_s = i\}} = N_i^{t-1}$$

Consider,

$$\therefore P\left[\mu_i - \hat{\mu}_i^t > \epsilon\right] = P\left[\frac{\mathbb{E}\left[\sum_{s=1}^{t-1} x_i^s \mathbb{I}_{\{i_s=i\}}\right]}{N_i^{t-1}} - \frac{\sum_{s=1}^{t-1} x_i^s \mathbb{I}_{\{i_s=i\}}}{N_i^{t-1}} > \epsilon\right]$$

$$= P\left[e^{\alpha\left\{\mathbb{E}\left[\sum_{s=1}^{t-1} x_i^s \mathbb{I}_{\{i_s=i\}}\right] - \sum_{s=1}^{t-1} x_i^s \mathbb{I}_{\{i_s=i\}}\right\}} > e^{\alpha N_i^{t-1}\epsilon}\right]$$

• We can do this because $P[Z > \epsilon] \Rightarrow P[e^{\alpha Z} > e^{\alpha \epsilon}]$

Since e^x is a monotonically increasing function.

$$= P\left[e^{\alpha \sum_{b=1}^{t-1} \{\mathbb{E}[x_i^b \mathbb{I}_{\{i_b = i\}}] - x_i^b \mathbb{I}_{\{i_b = i\}}\}} > e^{\alpha N_i^{t-1} \epsilon}\right]$$

$$\leq \frac{\mathbb{E}\left[e^{\alpha \sum_{b=1}^{t-1} \{\mathbb{E}[x_i^b \mathbb{I}_{\{i_b = i\}}] - x_i^b \mathbb{I}_{\{i_b = i\}}\}}\right]}{e^{\alpha N_i^{t-1} \epsilon}}$$

Applying Markov's inequality

$$* \left(P[X \geq a] \leq \frac{\mathbb{E}[X]}{a}\right)$$

Upon solving, we get $P[|\mu_i - \hat{\mu}_i^t| > \epsilon] \leq e^{-2N_i^{t-1}\epsilon^2}$

∴ As trials increase, expected average gets closer to true average.

Lemma: Let i^* denote optimal arm $\mu_{i^*} = \mu^*$ and fix any $i : \Delta_i > 0$. If UCB selects arm i at trial t ($i_t = i$) then atleast one of the following holds,

i) $\hat{\mu}_{i^*}^{t-1} \leq \mu^* - \sqrt{\frac{\alpha \ln t}{2N_{i^*}^{t-1}}}$

ii) $\hat{\mu}_i^{t-1} \geq \mu_i + \sqrt{\frac{\alpha \ln t}{2N_i^{t-1}}}$

iii) $N_i^{t-1} \leq \frac{2\alpha \ln T}{\Delta_i^2}$ $\quad (\Delta_i = \mu^* - \mu_i)$

Proof: (By contradiction) i) is false, $\mu^* < \hat{\mu}_{i^*}^{t-1} + \sqrt{\frac{\alpha \ln t}{2N_{i^*}^{t-1}}}$

$\Rightarrow N_i^{t-1} > \frac{2\alpha \ln T}{\Delta_i^2}$ $\}$ $\because$ iii) is false

$$\Delta_i > \sqrt{\frac{2\alpha \ln T}{N_i^{t-1}}}$$

$$\Rightarrow \hat{\mu}_{i^*}^{t-1} + \sqrt{\frac{\alpha \ln t}{2N_{i^*}^{t-1}}} > \underbrace{\mu_i + \Delta_i}_{\mu^*}$$

$$\Rightarrow \hat{\mu}_{i^*}^{t-1} + \sqrt{\frac{\alpha \ln t}{2N_{i^*}^{t-1}}} > \mu_i + \sqrt{\frac{2\alpha \ln T}{N_i^{t-1}}}$$

($\because$ ii) is false) $\quad \geq \mu_i + \sqrt{\frac{2\alpha \ln t}{N_i^{t-1}}}$

$$\geq \hat{\mu}_i^{t-1} + \sqrt{\frac{2\alpha \ln t}{N_i^{t-1}}} - \sqrt{\frac{\alpha \ln t}{2N_{i^*}^{t-1}}}$$

$$\geq \hat{\mu}_i^{t-1} + \sqrt{\frac{\alpha \ln t}{2N_{i^*}^{t-1}}}$$

- This implies that the algorithm pulls the optimal arm but our assumption was that arm i is pulled, hence its a contradiction.

* Now, consider the pseudo regret $\tilde{R}_T(A)$ (From 2 pages earlier)

$$\tilde{R}_T = T\mu^* - \mathbb{E}\left[\sum_{t=1}^{T} \mu_{i_t}\right]$$

↳ Proved again 2 pages later.

$$= T\mu^* - \mathbb{E}\left[\sum_{i=1}^{n} N_i^T \mu_i\right]$$

(N_i^T = Number of times arm i is pulled in T trials)

$$= T\mu^* - \sum_{i=1}^{n} \mathbb{E}[N_i^T]$$

~~$\sum_{t=1}^{T}$~~

$$= \sum_{i=1}^{n} \mathbb{E}[N_i^T] \cdot \mu^* - \sum_{i=1}^{n} \mathbb{E}[\mu_i N_i^T]$$

$$= \sum_{i=1}^{n} \mathbb{E}[N_i^T(\mu^* - \mu_i)]$$

$$= \sum_{i=1}^{n} \mathbb{E}[N_i^T \Delta_i]$$ (Δ_i: Fixed value based on actual means)

$$= \sum_{i=1}^{n} \Delta_i \mathbb{E}[N_i^T]$$

---

(9) Theorem

$$P[|\mu - \hat{\mu}| \geq \epsilon] \leq 2e^{-2N\epsilon^2}$$

* $\left(P[|\mu_i - \hat{\mu}_i^*| \geq \epsilon] \leq e^{-2N_i^{t-1}\epsilon^2}\right)$ → Proof from Hoffding's Lemma

$\hat{\mu} = \frac{\sum x_i}{N}$, $x \in [0,1]$, $\mu = \mathbb{E}[x]$

$$\hat{\mu}_i^* = \frac{\sum_{s=1}^{t-1} x_i^s \mathbb{I}_{\{i_s = i\}}}{\sum_{s=1}^{t-1} \mathbb{I}_{\{i_s = i\}}} \rightarrow N_i^{t-1}$$

Proof: Complete proof in lecture notes (on Moodle)

---

* UCB Arm Selection:

*

- $P[\mu_i - \hat{\mu}_i^t \geq \epsilon] \leq e^{-2N_i^{t-1}\epsilon^2} \leq \delta_t$ ← Assume it is an upper bound

$$-2N_i^{t-1}\epsilon^2 \leq \ln \delta_t$$

$$\ln \frac{1}{\delta_t} \leq 2N_i^{t-1}\epsilon^2$$

$$\epsilon \geq \sqrt{\frac{\ln 1/\delta_t}{2N_i^{t-1}}} \rightarrow \epsilon^*$$

$\Rightarrow$ Now $P\left[\mu_i - \hat{\mu}_i^t \geq \sqrt{\frac{\ln(1/\delta_t)}{2N_i^{t-1}}}\right] \leq \delta_t$

$\Rightarrow$ Consider, $P[\mu_i - \hat{\mu}_i < \epsilon] > 1 - \delta_t$

that is, $\mu_i - \hat{\mu}_i < \epsilon^*$ with prob atleast $1 - \delta_t$

$\mu_i < \hat{\mu}_i + \epsilon^*$ with prob $1 - \delta_t$

$\mu_i < \hat{\mu}_i + \sqrt{\frac{\ln(1/\delta_t)}{2N_i^{t-1}}}$ with prob $1 - \delta_t$

- Choose $\delta_t = \frac{1}{t^\alpha}$ → Just has to be a function of t

* $\Rightarrow$ $\boxed{\mu_i < \hat{\mu}_i + \sqrt{\frac{\alpha \ln t}{2N_i^{t-1}}}}$ with prob $1-\delta_t$ (or which is high.)

$\therefore$ We use this as the criterion for arm selection in our UCB algo.

---

*UCB Regret Bound:

**

- $\tilde{R}_T = T\mu^* - \sum_{t=1}^{T} \mathbb{E}[\mu_{i_t}]$

$T = \mathbb{E}\left[\sum_{i=1}^{n} N_i^T\right]$

$\Rightarrow \tilde{R}_T = \mathbb{E}\left[\sum_{i=1}^{n} N_i^T\right]\mu^* - \sum_{t=1}^{T} \mathbb{E}[\mu_{i_t}]$

$\Rightarrow \tilde{R}_T = \mathbb{E}\left[\sum_{i=1}^{n} N_i^T\right]\mu^* - \mathbb{E}\left[\sum_{t=1}^{T} \mu_{i_t}\right]$

$= \mathbb{E}\left[\sum_{i=1}^{n} N_i^T\right]\mu^* - \mathbb{E}\left[\sum_{i=1}^{n} N_i^T \mu_i\right]$

$\left(\because \sum_{t=1}^{T} \mu_{i_t} = \sum_{t=1}^{T}\sum_{i=1}^{n} \mu_i \mathbb{I}_{\{i_t=i\}} = \sum_{i=1}^{n} \mu_i \left(\sum_{t=1}^{T} \mathbb{I}_{\{i_t=i\}}\right) \to N_i^T = \sum_{i=1}^{n} \mu_i N_i^T\right)$

$\Rightarrow \tilde{R}_T = \mathbb{E}\left[\sum_{i=1}^{n} N_i^T \cdot (\mu^* - \mu_i)\right]$

$= \mathbb{E}\left[\sum_{i=1}^{n} N_i^T \cdot \Delta_i\right]$

$\tilde{R}_T = \sum_{i=1}^{n} \mathbb{E}[N_i^T \Delta_i] \longrightarrow (1)$

- Now, $\mathbb{E}[N_i^T] = \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{I}_{\{i_t=i\}}\right]$

$= \mathbb{E}\left[\sum_{t=1}^{T} \left(\mathbb{I}_{\{i_t=i,\, N_i^{t-1} \le t_0\}} + \mathbb{I}_{\{i_t=i,\, N_i^{t-1} > t_0\}}\right)\right]$

$\mathbb{I}_{\{i_t=i,\, N_i^{t-1} \le t_0\}} \le t_0$ (upper bound)

$\because N_i^t < t$ always. $\Rightarrow \le t_0 + \mathbb{E}\left[\sum_{t=t_0+1}^{T} \mathbb{I}_{\{i_t=i,\, N_i^{t-1} > t_0\}}\right]$

$$\leq t_0 + \sum_{t=t_0+1}^{T} P\left[i_t = i, N_i^{t-1} > t_0\right]$$

Choose $t_0 = \left\lceil \frac{2\alpha \ln T}{\Delta_i^2} \right\rceil$, α = Just a parameter.

$$\therefore \mathbb{E}[N_i^T] \leq t_0 + \sum_{t=t_0+1}^{T} P[\text{event 1 or event 2}]$$

($i_t = i$, $N_i^{t-1} > t_0$)

$$\leq t_0 + \sum_{t=t_0+1}^{T} \{P[\text{event 1}] + P[\text{event 2}]\}$$

Now, $P[\text{event 1}] = P\left[\hat{\mu}_{i^*}^{t} \leq \mu^* - \sqrt{\frac{\alpha \ln t}{2N_{i^*}^{t-1}}}\right]$

(Since the 3 condition lemma must hold)

$$P[\text{event 1}] = P\left[\bigcup_{s=1}^{t-1} \left\{\hat{\mu}_{i^*}^{s} \leq \mu^* - \sqrt{\frac{\alpha \ln t}{2s}}\right\}\right]$$

↙ Assuming i^* was pulled any number of times from 1 to $t-1$.

$$P[\text{event 1}] \leq \sum_{s=1}^{t-1} P\left\{\hat{\mu}_{i^*}^{s} \leq \mu^* - \sqrt{\frac{\alpha \ln t}{2s}}\right\}$$

From Hoeffding's lemma, $P\{\hat{\mu}_i^s - \mu^* \leq -\sqrt{\frac{\alpha \ln t}{2s}}\}$

$$= 1/t^{\alpha}$$

$$\therefore P[\text{event 1}] \leq \sum_{s=1}^{t-1} 1/t^{\alpha} \leq 1/t^{\alpha-1}$$

$$\rightarrow \leq \sum_{s=1}^{t} 1/t^{\alpha}$$

$$\therefore \mathbb{E}[N_i^T] \leq t_0 + \frac{1}{\alpha-2} + \frac{1}{\alpha-2}$$

? $\left(\because \sum_{t=t_0+1}^{T} P[\text{event 1}] \leq \frac{1}{\alpha-2}\right)$

$\Rightarrow$ Similarly we can derive $\sum_{t=t_0+1}^{T} P[\text{event 2}] \text{ as } \leq \frac{1}{\alpha-2}$.

$$\therefore \mathbb{E}[N_i^T] \leq t_0 + \frac{2}{\alpha-2} = \frac{2\alpha \ln T}{\Delta_i^2} + 1 + \frac{2}{\alpha-2}$$

$$= \frac{2\alpha \ln T}{\Delta_i^2} + \frac{\alpha}{\alpha-2}$$

Now, going back to (1),

$$\tilde{R}_T = \sum_{i=1}^{n} \mathbb{E}[N_i^T \Delta_i]$$

$$= \sum_{i=1}^{n} \Delta_i \, \mathbb{E}[N_i^T]$$

$$\leq \sum_{i=1}^{n} \Delta_i \left(\frac{2\alpha \ln T}{\Delta_i^2} + \frac{\alpha}{\alpha-2} \right)$$

$$= \sum_{i:\Delta_i>0} \left(\frac{2\alpha \ln T}{\Delta_i} \right) + \sum_{i:\Delta_i>0} \frac{\alpha \Delta_i}{\alpha-2}$$

$$\therefore R_T[UCB(\alpha)] \leq \sum_{i:\Delta_i>0} \left(\frac{2\alpha}{\Delta_i} \right) \ln T + \sum_{i:\Delta_i>0} \frac{\alpha}{\alpha-2} \Delta_i$$

$*$

---

(10) Adversarial MAB: (EXP 3)

- Need to choose arms non-deterministically.
- Actions selected randomly based on weights of each arm (CDF)

↳ Let l_{it} be the loss corresponding to arm i at t.

$\Rightarrow \tilde{l}_{i_t}^t = \frac{l_{i_t}^t}{p_{i_t}^t} \mathbb{I}_{\{i_t = i\}}$ } Importance Sampling

EXP3 (Algo) – Adversarial MAB

$p^1 = \frac{1}{n}[1 \; 1 \; \cdots \; 1]$

for $t = 1$ to T:

Observe $l_{i_t}^t \in [0,1]$

for $i = 1$ to n:

$$\tilde{l}_i^t = \frac{l_i^t}{p_i^t} \mathbb{I}_{\{i_t = i\}}$$

$$\tilde{L}_i^t = \tilde{L}_i^{t-1} + \tilde{l}_i^t$$

for $i = 1$ to n:

$$p_i^{t+1} = \frac{e^{-\eta_t \tilde{L}_i^t}}{\sum_{j=1}^{n} e^{-\eta_t \tilde{L}_j^t}}$$

Note: $\mathbb{E}_{l_t \sim p^t}[\tilde{l}_i^t] = \mathbb{E}_{l_t \sim p^t}\left[\frac{l_i^t}{p_i^t} \mathbb{I}_{\{i_t = i\}}\right]$

$= \sum_{j=1}^{n} p_j^t \cdot \frac{l_i^t}{p_i^t} \mathbb{I}_{\{j=i\}}$

$= l_i^t$ //

- $\tilde{l}_i^t$ is an unbiased estimator of l_i^t //

Regret Analysis:

$$\phi_t = -\frac{1}{\eta} \log \sum_{i=1}^{n} e^{-\eta \tilde{L}_i^t}$$ } Potential function

$$\Rightarrow \phi_{t+1} - \phi_t = -\frac{1}{\eta} \log \left[\frac{\sum_{i=1}^{n} e^{-\eta \tilde{L}_i^{t-1}} \cdot e^{-\eta \tilde{l}_i^t}}{\sum_{j=1}^{n} e^{-\eta \tilde{L}_j^{t-1}}} \right]$$

$$\Rightarrow \phi_{t+1} - \phi_t = \frac{-1}{\eta} \log \left[\frac{\sum_{i=1}^{n} e^{-\eta \tilde{l}_i^t} \cdot e^{-\eta \tilde{l}_i^{t+1}}}{\sum_{j=1}^{n} e^{-\eta \tilde{l}_j^{t+1}}} \right]$$

$$= \frac{-1}{\eta} \log \left[\sum_{i=1}^{n} e^{-\eta \tilde{l}_i^t} \cdot p_i^t \right]$$

$$= \frac{-1}{\eta} \log \left[\mathbb{E}_{i \sim \bar{p}^t} \left[e^{-\eta \tilde{l}_i^t} \right] \right] \rightarrow (1)$$

Now, $e^{-x} \leq 1 - x + \frac{x^2}{2} \quad \forall x \geq 0$

(∵ The rest of the terms will end up summing to a -ve value)

- Putting this in (1),

$$\phi_{t+1} - \phi_t \geq -\frac{1}{\eta} \log \left[\mathbb{E}_{i \sim \bar{p}^t} \left[1 - \eta \tilde{l}_i^t + \frac{\eta^2 (\tilde{l}_i^t)^2}{2} \right] \right]$$

$$\phi_{t+1} - \phi_t \geq -\frac{1}{\eta} \log \left[1 - \mathbb{E} \left[\eta \tilde{l}_i^t - \frac{\eta^2}{2} (\tilde{l}_i^t)^2 \right] \right]$$

Now, $\log(1-x) \leq -x \Rightarrow -\log(1-x) \geq x$

$$\therefore \phi_{t+1} - \phi_t \geq \frac{1}{\eta} \mathbb{E}_{i \sim \bar{p}^t} \left[\eta \tilde{l}_i^t - \frac{\eta^2}{2} (\tilde{l}_i^t)^2 \right]$$

$$= \mathbb{E}_{i \sim \bar{p}^t} \left[\tilde{l}_i^t \right] - \frac{\eta}{2} \mathbb{E}_{i \sim \bar{p}^t} \left[(\tilde{l}_i^t)^2 \right]$$

$$= \sum_{i=1}^{n} p_i^t \tilde{l}_i^t - \frac{\eta}{2} \sum_{i=1}^{n} p_i^t \cdot (\tilde{l}_i^t)^2$$

$$\phi_{t+1} - \phi_t \geq \sum_{i=1}^{n} l_i^t \mathbb{1}_{\{i_t = i\}} - \frac{\eta}{2} \sum_{i=1}^{n} \frac{(l_i^t)^2}{p_i^t} \mathbb{1}_{\{i_t$$

$$\Rightarrow \mathbb{E}\left[\phi_{t+1} - \phi_t \mid H_{t-1}\right] \geq \mathbb{E}\left[\sum_{i=1}^{n} l_i^t p_i^t$$

$$\because \mathbb{E}[\phi_{t+1} - \phi_t]$$

$$\mathbb{E}\left[\phi_{t+1} - \phi_t \mid H_{t-1}\right] \geq \mathbb{E}\left[\sum_{i=1}^{n} l_i^t \mathbb{1}_{\{i_t = i\}}\right.$$

$$\left. - \frac{\eta}{2} \sum_{i=1}^{n} \frac{(l_i^t)^2}{p_i^t} \mathbb{1}_{\{i_t = i\}}\right]$$

$$= \sum_{i=1}^{n} l_i^t \cdot \mathbb{E}\left[\mathbb{1}_{\{i_t = i\}}\right]$$

$$- \frac{\eta}{2} \sum_{i=1}^{n} \frac{(l_i^t)^2}{p_i^t} \mathbb{E}\left[\mathbb{1}_{\{i_t$$

$$\rightarrow (2)$$

$$=$$

Now, $\mathbb{E}_{i_t \sim p_t}\left[\mathbb{1}_{\{i_t = i\}}\right]$

$$= \sum_{i=1}^{n} p_{i_t}^t \mathbb{1}_{\{i_t = i\}} = p_i^t$$

Now, from (2),

$$\mathbb{E}\left[\phi_{t+1} - \phi_t \mid H_{t-1}\right] \geq \sum_{i=1}^{n} l_i^t p_i^t - \frac{\eta}{2} \sum_{i=1}^{n} (l_i$$

$$\because l_i^t \in [0,1] \qquad \geq \sum_{i=1}^{n} l_i^t p_i^t - \frac{\eta}{2} \cdot n$$

Now, $\mathbb{E}\left[\phi_{t+1} - \phi_t\right] \geq \mathbb{E}_{H_{t-1}}\left[\sum_{i=1}^{n} l_i^t p_i^t - \frac{\eta n}{2}\right]$

$$\mathbb{E}[\phi_{t+1} - \phi_t] \geq \mathbb{E}[l_{i_t}^t] - \frac{\eta n}{2}$$

Sum from 1 to T,

$$\mathbb{E}[\phi_{T+1} - \phi_1] \geq \mathbb{E}\left[\sum_{t=1}^{T} l_{i_t}^t\right] - \frac{\eta T n}{2} \quad \rightarrow (3)$$

Now, $\phi_{T+1} - \phi_1 = -\frac{1}{\eta} \log \frac{\sum_{i=1}^{n} e^{-\eta \tilde{L}_i^T}}{n}$

$$\leq -\frac{1}{\eta} \log\left(\frac{e^{-\eta \tilde{L}_{i^*}^T}}{n}\right)$$

where $i^* = \underset{i}{\operatorname{argmin}}\ \tilde{L}_i^T$

$$\Rightarrow \phi_{T+1} - \phi \leq \tilde{L}_{i^*}^T + \frac{1}{\eta} \log n$$

$$\mathbb{E}[\phi_{T+1} - \phi_1] \leq \tilde{L}_{i^*}^T + \frac{1}{\eta} \log n \quad \rightarrow (4)$$

Comparing LB & UB, (3) & (4)

$$\mathbb{E}\left[\sum_{t=1}^{T} l_{i_t}^t\right] - \frac{\eta T \cdot n}{2} \leq L_{i^*}^T + \frac{\log n}{\eta}$$

Note: $\mathbb{E}[\tilde{L}_i^T] = L_i^T$ (For any i)

$$\because \mathbb{E}[\tilde{L}_i^T] = \mathbb{E}_{\{i_1, \ldots, i_T\}}\left[\sum_{t=1}^{T} \frac{l_i^t}{p_i^t} \mathbb{1}_{\{i_t = i\}}\right]$$

$$= \sum_{t=1}^{T} \mathbb{E}_{\{i_t\}}\left[\frac{l_i^t}{p_i^t} \mathbb{1}_{\{i_t = i\}}\right]$$

$$= \sum_{t=1}^{T} l_i^t = L_i^t$$

//

$$\Rightarrow \underbrace{\mathbb{E}\left[\sum_{t=1}^{T} l_i^t\right] - L_{i^*}^T}_{\text{Regret}} \leq \frac{\log n}{\eta} + \frac{\eta \cdot Tn}{2}$$

Differentiating wrt η & maximizing,

$$\eta^* = \sqrt{\frac{2\log n}{Tn}}$$

$$\text{Regret with } \eta^* = \sqrt{2Tn\log n}$$

(In non adversarial, we get Regret(η^*) $= \sqrt{2T\log n}$)

$\Rightarrow$ This is a partial information setting so we have to pay the price of $\sqrt{n}$.

Note: EXT3 can be extended to contextual MAB setting. (Run EXT3 for each context available)

---

(11) Reinforcement Learning:

* S = Set of States
* R = Set of Rewards

$A(s)$ = Action set when state is s.

$\Rightarrow R_{t+1}$ = Reward at $t+1$ by action A_t being performed in state S_t.

- Return $G_t = R_{t+1} + \ldots + R_T \rightarrow$ Time horizon.
  ↳ Maximize this return.

⇒ Episodic tasks — Finite time horizon & reset and restart game every episode.

⇒ Continuous tasks — Infinite time horizon.

$$G_t = \sum_{k=0}^{\infty} \gamma^k \cdot R_{t+k-1}$$

↙ Discounted return

$\gamma \in [0, 1)$

* If $\gamma = 0$, we focus only on immediate reward (Myopic setting)

⇒ Markov ~~assumption~~ property. — Memoryless, only takes previous state & action into consideration

↓

$$P_R[S_{t+1} = S', R_{t+1} = r \mid S_0, A_0, R_1, S_1, \ldots R_t, S_t, A_t]$$

$$= P_R[S_{t+1} = S', R_{t+1} = r \mid S_t = S, A_t = a] \rightarrow (1)$$

---

* Markov Decision Process (MDP):

↳ Reinforcement learning tasks which satisfy the markov property (1) are called Markov decision processes.

Quantities of Interest → i) Expected Reward:

$$r(s, a) = \mathbb{E}[R_{t+1} \mid S_t = s, A_t = a]$$

$$= \sum_{r \in R} r \cdot P_R[R_{t+1} = r \mid S_t = s, A_t = a]$$

Marginalize ↓

$$* \Rightarrow r(s, a) = \sum_{r \in R} r \sum_{s' \in S} P_R[R_{t+1} = r, S_{t+1} = s' \mid S_t = s, A_t = a]$$

$$= \sum_{r \in R} r \sum_{s' \in S} p[r, s' \mid s, a]$$

ii) State transition probability:

$$P[S_{t+1}=s' \mid S_t=s, A_t=a]$$

$$= \sum_{r \in R} P[S_{t+1}=s', R_{t+1}=r \mid S_t=s, A_t=a]$$

$$= \sum_{r \in R} P[s', r \mid s, a]$$

~~* iii) Expected reward (for state-action-next state triplets):~~

- ~~$E[R_{t+1} \mid S_t=s, A_t=a, S$~~

ex: Recycling Robot

$S = \{high, low\}$

$A(high) = \{search, wait\}$

$A(low) = \{search, wait, recharge\}$

$S_t(s)$	$S_{t+1}(s')$	$A_t(a)$	$P(s' \mid s,a)$	$r(s,a)$
high	high	search	α	r_{search}
high	low	search	$1-\alpha$	r_{search}
high	high	wait	1	r_{wait}
high	low	wait	0	r_{wait}
low	high	search	$1-\beta$	-3
low	low	search	β	r_{search}
low	high	wait	0	r_{wait}
low	low	wait	1	r_{wait}
low	high	recharge	1	0
low	low	recharge	0	0

Assume ← it recharges somehow.

---

* **Policy:** A policy π is a mapping from a set of states and a set of actions to probabilities

$$\pi(a|s) = P[A_t = a | S_t = s]$$

i) **State-Value function:**

$$\cdot\ V_\pi(s) = \mathbb{E}_\pi[G_t | S_t = s]$$

↓ Discounted return

ii) **State-Action pair value function:**

$$* \cdot q_\pi(s,a) = \mathbb{E}_\pi[G_t | S_t = s, A_t = a]$$

$$V_\pi(s) = \mathbb{E}_\pi[G_t | S_t = s]$$

$$= \mathbb{E}\left[\mathbb{E}_\pi[G_t | S_t = s, A_t = a]\right]$$

$$= \sum_{a \in A(s)} \pi[a|s]\, q_\pi(s,a)$$

(12) $$V_\pi(s) = \mathbb{E}_\pi[G_t | S_t = s]$$

$$= \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,\middle|\, S_t = s\right]$$

$$= \mathbb{E}_\pi\left[R_{t+1} + \sum_{k=1}^{\infty} \gamma^k R_{t+k+1} \,\middle|\, S_t = s\right]$$

Also,

$$V_\pi(s) = \mathbb{E}_{A_t|S_t}\left[\mathbb{E}_\pi[G_t | S_t = s, A_t = a]\right]$$

$$= \mathbb{E}_{A_t|S_t}[q_\pi(s,a)]$$

$$= \sum_{a \in A(s)} q_\pi(s,a) \cdot P_\pi(A_t = a | S_t = s)$$

$$= \sum_{a \in A(s)} \pi(a|s) \cdot q_\pi(s,a)$$

$$\Rightarrow q_\pi(s,a) = \mathbb{E}_\pi\left[G_t \mid S_t=s, A_t=a\right]$$

$$= \mathbb{E}_\pi\left[R_{t+1} + \sum_{k=1}^{\infty} \gamma^k R_{t+k+1} \mid S_t=s, A_t=a\right]$$

$$= \mathbb{E}_{S_{t+1}=s' \mid S_t=s, A_t=a}\left[\mathbb{E}_\pi\left[R_{t+1} + \sum_{k=1}^{\infty} \gamma^k R_{t+k+1}\right]\right]$$

$$= \mathbb{E}_\pi\left[R_{t+1} \mid S_t=s, A_t=a\right]$$

$$+ \mathbb{E}_{S_{t+1}=s' \mid S_t=s, A_t=a}\left[\mathbb{E}_\pi\left[\gamma G_{t+1} \mid S_t=s, A_t=a, S_{t+1}=s'\right]\right]$$

$$= r(s,a) + \mathbb{E}_{S_{t+1}=s' \mid S_t=s, A_t=a}\left[\mathbb{E}_\pi\left[\gamma G_{t+1} \mid S_{t+1}=s'\right]\right]$$

$$= r(s,a) + \mathbb{E}_{s_{t+1}=s' \mid S_t=s, A_t=a}\left[\gamma \cdot V_\pi(s')\right]$$

$$= r(s,a) + \gamma \cdot \sum_{s' \in S} V_\pi(s') \cdot P_r\left[S_{t+1}=s' \mid S_t=s, A_t=a\right]$$

$$= r(s,a) + \gamma \sum_{s' \in S} V_\pi(s') \cdot p(s' \mid s,a)$$

$$= r(s,a) + \gamma \sum_{s' \in S} V_\pi(s') \cdot \sum_r p(r,s' \mid s,a)$$

∴ we have,

$$\boxed{V_\pi(s) = \sum_{a \in A(s)} \pi(a \mid s)\left[r(s,a) + \gamma \sum_{s' \in S} p(s' \mid s,a) \cdot V_\pi(s')\right]}$$

* Bellman equation for state value function.

* Optimal value function:

- The optimal state value function $v_*(s)$ is as follows,

$$v_*(s) = \max_\pi v_\pi(s) \quad \forall s$$

Optimal State-Action value function:

$$q_*(s,a) = \max_\pi q_\pi(s,a)$$

- Both of these maximums occur for the same policy π^*
- $v_*(s) = \max_\pi v_\pi(s) = \max_\pi \sum_{a \in A(s)} \pi(a|s) \cdot q_\pi(s,a)$

$= \max_{a \in A(s)} q_*(s,a)$

(Bellman optimality equation for state value)

(∵ If we know q_π then $\pi(a_i|s) = 1$ for some a_i & $\pi(a_j|s) = 0$ for the rest)

($q_\pi(s,a_i)$ is best; The best action to choose)

- $q_*(s,a) = \max_\pi q_\pi(s,a) = \max_\pi r(s,a) + \sum_{s' \in S} p(s'|s,a) v_\pi(s')$

$= \max_* \left(r(s,a) + \sum_{s' \in S} p(s'|s,a) \cdot \sum_{a' \in A(s')} \pi(a'|s') \cdot q_\pi(s',a') \right)$

$= \max_{a' \in A(s')} \left(r(s,a) + \sum_{s' \in S} p(s'|s,a) \cdot q_*(s',a') \right)$

Bellman optimality equation for state-action value.

---

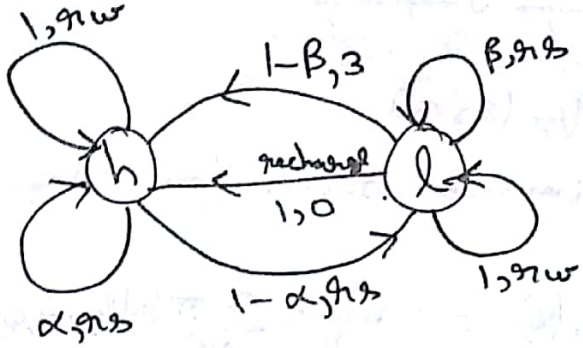
ex: Recycling robot (As we saw earlier)

$S = \{l, h\}$

$A(l) = \{s, w, r\}$

$A(h) = \{s, w\}$

S_t	S_{t+1}	A_t	$P(s'\|s,a)$	$r(s,a,s')$
h	h	s	α	r_s
h	l	s	$1-\alpha$	r_s
l	h	s	$1-\beta$	-3
l	l	s	β	r_s
h	h			
⋮	⋮	⋮	⋮	⋮



(13)

- $V_*(s) = \max_{a \in A(s)} q_{\pi_*}(s,a)$

$F^*(V) = \max_{a \in A(s)} r(s,a) + \sum_{s'} p(s'|s,a) \cdot V_*(s')$

↳ Bellman optimality operator

~~$\Rightarrow V_*(h) = \max \{ p(h|h,s) \}$~~

$V^\pi = r^\pi + \gamma P^\pi \cdot V^\pi$ } Considering actions at each step are known.

Bellman's expectation operator $\{ F^\pi(V) = r^\pi + P^\pi \cdot V$

↑ Set of all states

HW1:

Note: * Verify that Bellman Optimality operator satisfies monotonicity property.

HW2:

* Show that Bellman Optimality operator is γ-contraction mapping.

Use: $\max_a f(a) - \max_a g(a) \leq \max_a (f(a) - g(a))$

* Properties of Bellman ops: (expectation operator)

i) Monotonicity → $V \leq U \Rightarrow F(V) \leq F(U)$

$F(V) = r^\pi + \gamma P^\pi V$

$F(U) = r^\pi + \gamma P^\pi U$

$\therefore F(V) - F(U) = \gamma P^\pi (z) \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$

ii) Offset;

$$F(V + ce) = F(V) + \gamma ce$$

(e: vector of all 1's)

* iii) γ-contraction in L_∞ norm:

$$\Rightarrow \|F(V) - F(U)\|_\infty \leq \gamma \|V - U\|_\infty$$

$\because F^\pi(V) - F^\pi(U) = \gamma P^\pi (V - U)$

Note: $\|V\|_\infty = \max_{i \in \{1 \ldots N\}} |V_i|$

$$\Rightarrow \|F^\pi(V) - F^\pi(U)\|_\infty = \|\gamma P^\pi (V - U)\|_\infty$$

$$= \gamma \max_{j = \{1 \ldots N\}} \sum_i P_{i,i}^\pi \left(V(s_i) - U(s_i) \right)$$

$\because P^\pi (V - U)$

$$= \begin{bmatrix} \sum_i P_{i,1}^\pi (V(s_i) - U(s_i)) \\ \vdots \\ \sum_i P_{i,N}^\pi (V(s_i) - U(s_i)) \end{bmatrix}$$

$$\leq \gamma \max_{i \in \{1, \ldots N\}} |V(s_i) - U(s_i)| \cdot \left(\sum_i P_{i,\gamma}^\pi \right)$$

$$\leq \gamma \max_i |V(s_i) - U(s_i)|$$

$$\leq \gamma \|V - U\|_\infty$$

* Banarch's Fixed Point theorem:

- Ensures the existence of unique fixed point (converges)

↳ $\lim_{k \to \infty} (F^{\pi})^k (V) = V^{\pi}$

$\lim_{k \to \infty} (F)^k (V) = V^*$ → Solution to bellman's optimality equations.

---

1) Policy Iteration: (Details in slides)

* For a given policy, we try to find the values of each state (estimated rewards)

⇒ This gives us the best reachable reward from each state, we then update the policy

↳ The policy is updated as such, (policy improvement)

if $\exists a \in A(s)$ s.t $q_{\pi}(s,a) \geq V_{\pi}(s)$

then $\pi'(s) = a$ //

$$\Rightarrow \pi_{k+1}(s) = \arg\max_{a \in A(s)} \left[r(s,a) + \gamma \sum_{s' \in S} p(s'|s,a) \cdot V_{\pi_k}(s') \right]$$

(Steps are policy evaluation & policy improvement and then repeat)

⇒ We stop at k^{th} iteration when $V_{\pi_{k+1}}(s) = V_{\pi_k}(s)$ $\forall s$ //

- No of possible policies = $|A|^{|S|}$ → No. of states.
  - ↓ No. of all possible actions
  - ↓ Finite number and hence policy iteration won't repeat policies.

Lemma: Let T be a γ-contraction mapping, $T: R^N \to R^N$

** then $V_{n+1} = TV_n$

* then $\lim_{n\to\infty} V_n = V_\gamma^*$, $TV_\gamma^* = V_\gamma^*$ ← Fixed point. (Unique)

Show its cauchy,

$\Rightarrow \|V_{n+m} - V_n\| \leq \epsilon$ } For every $\epsilon > 0$, there exists n_0 such that $n, m \geq n_0$

LHS: $\|V_{n+m} - V_n\| = \left\| \sum_{k=0}^{m-1} (V_{n+k+1} - V_{n+k}) \right\|$

$$\leq \sum_{k=0}^{m-1} \|V_{n+k+1} - V_{n+k}\|$$

$$= \sum_{k=0}^{m-1} \|T^{k} V_{n+k} - T^{k} V_{n+k-1}\|$$

(Property of γ contraction) $$\leq \sum_{k=0}^{m-1} \gamma \|V_{n+k} - V_{n+k-1}\|$$

$$\leq \sum \gamma^{n+k} \|V_1 - V_0\|$$

$$= \sum_{k=0}^{m-1} \gamma^{n+k} \|V_1 - V_0\|$$

G.P ($\gamma < 1$) $$= \gamma^n \frac{(1-\gamma^m)}{1-\gamma} \cdot \|V_1 - V_0\|$$

$\leq \epsilon$

$\therefore$ Its a cauchy sequence { since all the terms are controllable and they don't diverge

$\Rightarrow \lim_{n\to\infty} \gamma^n = 0$, thus $\forall \epsilon > 0 \; \exists n_0(\epsilon)$ s.t $\forall n \geq n_0$, $\|V_{n+m} - V_n\| \leq \epsilon$

$\Rightarrow$ Let V^* be the limit,

Thus, $\lim_{n\to\infty} T^n V_0 = V^*$

Now, we show that V^* is a fixed point which means $TV^* = V^*$

a) $\|TV^* - V^*\| \geq 0$

b) $\|TV^* - V^*\| \leq \|TV^* - V_n\| + \|V_n - V^*\|$

$= \|TV^* - TV_{n-1}\| + \|V_n - V^*\|$

$\leq \gamma \|V^* - V_{n-1}\| + \|V_n - V^*\|$

But $\|V_{n-1} - V^*\| \leq \epsilon \;\forall\; n \geq n_0(\epsilon)$

$\therefore \|TV^* - V^*\| \leq \gamma\epsilon + \epsilon = \epsilon(\gamma+1)$

$\|TV^* - V^*\| \leq \epsilon(\gamma+1)$

$\Rightarrow TV^* = V^*$ ($\because$ It holds for any ϵ)! ($\because$ Norm is getting close to zero)

Theorem: The policy iteration algo generates a sequence of policies with non-decreasing performance.

$$V^{\pi_{k+1}} \geq V^{\pi_k} \;\forall k$$

(Slides) optimality

Proof: $V^{\pi_k} = F^{\pi_k} V^{\pi_k}$ } F is the bellman operator

V^{π_k} is the fixed point of F^{π_k}

$\Rightarrow V^{\pi_k} = F^{\pi_k} V^{\pi_k}$

$\leq F V^{\pi_k}$

$= F^{\pi_{k+1}} V^{\pi_k}$

$\Rightarrow F^{\pi_{k+1}} V^{\pi_k} \leq \left(F^{\pi_{k+1}}\right)^2 \cdot V^{\pi_k}$ $\rightarrow$ Monotonicity

$$\left(F^{\pi_{k+1}}\right)^{n-1} V^{\pi_k} \leq \left(F^{\pi_{k+1}}\right)^{n} V^{\pi_k}$$

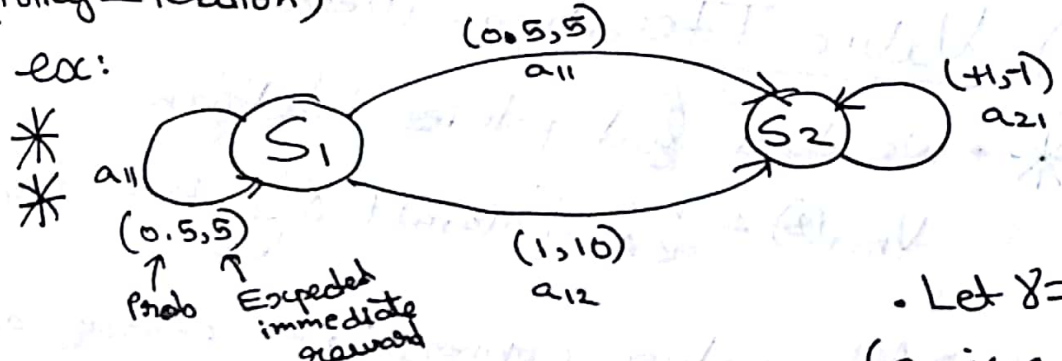
This will have a fixed point since it's a gamma contraction mapping

$$\Rightarrow V^{\pi_k} \leq F^{\pi_{k+1}} \cdot V^{\pi_k} \leq \ldots\ldots \left(F^{\pi_{k+1}}\right)^{n} V^{\pi_k} \to V^{\pi_{k+1}}$$

$V^{\pi_{k+1}}$: Fixed point of $F^{\pi_{k+1}}$

(Policy Iteration)

ex:



- Let $\gamma = 0.95$

(a_{11} is a probabilistic action)

Step 1: $\pi^0(S_1) = a_{12}$, $\pi^0(S_2) = a_{21}$ } Let this be our initial policy

- Policy evaluation: $V_{\pi_0}(S_1) = 10 + 0.95\left[V_{\pi_0}(S_2)\right]$

$V_{\pi_0}(S_2) = -1 + 0.95\left[V_{\pi_0}(S_2)\right]$

$\because P(S_2 | S_2, a_2) = 1$

$\therefore V_{\pi_0}(S_1) = -9$

$V_{\pi_0}(S_2) = -20$

- Policy improvement: $\pi_1(S_1) = \underset{a \in A(S_1)}{\arg\max}\, r(S_1, a) + \gamma \sum_{S' \in A(S_1)} P(S'|S_1) \cdot V_{\pi_0}(S')$

$$= \max \begin{bmatrix} 5 + 0.95 \times \frac{1}{2} \times V_{\pi_0}(S_1) + 0.95 \times \frac{1}{2} \times V_{\pi_0}(S_2) \\ 10 + 0.95 \times V_{\pi_0}(S_2) \end{bmatrix}$$

$$= \max \begin{bmatrix} -0.8775 \\ -9 \end{bmatrix}$$

$\therefore \pi_1(s_1) = a_{11}$
$\pi_1(s_2) = a_{21}$
} After change of policy (s_2 has only 1 action)

Step 2:

- Policy evaluation: $V_{\pi_1}(s_1) = -8.571$
  $V_{\pi_1}(s_2) = -20$
- We keep repeating until $V_{\pi_k}(s)$ converges.

---

2) Value Iteration: (Details in slides)

** - We don't find policies inbetween

$$V_{n+1}(s) = \max_{a \in A(s)} \left[r(s,a) + \gamma \sum_{s' \in S} p(s'|s,a) \cdot V_n(s') \right]$$

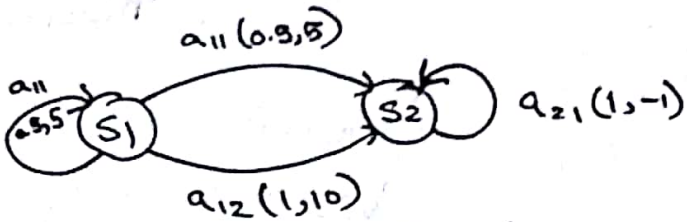
→ After values converge, we choose an output policy as,

$$\pi_\epsilon(s) \in \arg\max_{a \in A(s)} \left[r(s,a) + \sum_{s' \in S} p(s'|s,a) \cdot V_{n+1}(s') \right]$$

(15) - If $\|V_{n+1} - V_n\| < \epsilon \cdot \frac{1-\gamma}{2\gamma}$, then we stop.

(Value Iteration)

ex:



$$r(s,a,s') = \mathbb{E}[R_{t+1} \mid S_t = s, A_t = a, S_{t+1} = s']$$

$\epsilon = 0.01$, $\gamma = 0.95$

$V_0(s_1) = V_0(s_2) = 0$

$$V_{n+1}(s_1) = \max \begin{cases} 5 + 0.5 \times 0.95 \times V_n(s_1) + 0.5 \times 0.95 \times V_n(s_2) \\ 10 + 0.95 \cdot V_n(s_2) \end{cases}$$

$$V_{n+1}(s_2) = -1 + 0.95 \times V_n(s_2)$$

$V_1(s_1) = \max(5, 10) = 10$

$V_1(s_2) = -1$

At $n = 162$, $\|V^{163} - V^{162}\| < \frac{0.01 \times 0.05}{2 \times 0.95} = 0.00026$

will converge to some values

$V_{162}(s_1) =$ ~~8.3565~~ ....

$V_{162}(s_2) =$ ~~10.3565~~ ....

$\pi_\epsilon(s_1) = a_{11}$

$\pi_\epsilon(s_2) = a_{21}$

Now, consider optimality,

$$\Rightarrow V_*(s_1) = \max \begin{cases} 5 + 0.5 \times \gamma \times V_*(s_1) + 0.5 \times \gamma \times V_*(s_2) \\ 10 + \gamma \times V_*(s_2) \end{cases}$$

$$V_*(s_2) = -1 + \gamma \cdot V_*(s_2)$$

- Case 1: $\gamma = 0$

$V_*(s_1) = 10$, $\pi^*(s_1) = a_{12}$

$V_*(s_2) = -1$, $\pi^*(s_2) = a_{21}$

- Case 2: $\gamma = 0.5$

$$V_*(s_2) = -1 + 0.5\, V_*(s_2) \Rightarrow V_*(s_2) = -2$$

$$V_*(s_1) = \max \begin{cases} 5 + 0.25 \times V_*(s_1) - 0.5 \\ 10 - 1 \end{cases}$$

$$= \max \begin{cases} 4.5 + 0.25 \times V_*(s_1) \\ 9 \end{cases}$$

$$V_*(s_1) \geq 4.5 + 0.25 \times V_*(s_1)$$

$$\Rightarrow 0.75\, V_*(s_1) \geq 4.5$$

$V_*(s_1) \geq 6$ } 9 is better

$\therefore V_*(s_1) = 9$

$\pi_*(s_1) = a_{12}$, $\pi_*(s_2) = a_{21}$

- Case 3: $\gamma = 0.95$

$$V_*(s_2) = -1 + 0.95\, V_*(s_2)$$

$$V_*(s_2) = -20$$

$$V_*(s_1) = \max \begin{cases} 5 + 0.475\, V_*(s_1) - 20 \times 0.475 \\ 10 - 20 \end{cases}$$

$$= \max \begin{cases} -4.5 + 0.475 \times V_*(s_1) \\ -10 \end{cases}$$

$$V_*(s_1) \geq -4.5 + 0.475 \times V_*(s_1)$$

$$V_*(s_1) \geq \frac{-4.5}{0.525} \quad , \quad \therefore V_*(s_1) \neq -10$$

$$\therefore V_*(s_1) = \frac{-4.5}{0.525}$$

$$\pi_*(s_1) = a_{11} \quad , \quad \pi_*(s_2) = a_{21}$$

---

* Correctness of Value Iteration Algo:

Theorem: For the series V_n and policy π_ϵ computed by VI, following will hold.

i) $\lim_{n \to \infty} V_n = V_\gamma^*$

ii) $\exists\, n_0$ s.t. $\forall\, n \geq n_0$, $\|V_{n+1} - V_n\| \leq \epsilon \cdot \frac{1-\gamma}{2\gamma}$

* iii) The policy π_ϵ is ϵ-optimal

$$\|V_\gamma^* - V_\gamma^{\pi_\epsilon}\| \leq \epsilon$$

iv) If $\|V_{n+1} - V_n\| < \epsilon \cdot \frac{1-\gamma}{2\gamma}$ then $\|V_{n+1} - V_\gamma^*\| < \frac{\epsilon}{2}$ (Assuming n is after convergence)

Proof:

** iii) We have to show, $\|V_\gamma^* - V_\gamma^{\pi_\epsilon}\| \leq \epsilon$

LHS: $\|V_\gamma^* - V_\gamma^{\pi_\epsilon}\| = \|V_\gamma^* - V_{n+1} + V_{n+1} - V_\gamma^{\pi_\epsilon}\|$

$\leq \|V_\gamma^* - V_{n+1}\| + \|V_{n+1} - V_\gamma^{\pi_\epsilon}\|$ ← Δ'le inequality

Part 1: $\|V_\gamma^* - V_{n+1}\|$ → Bellman optimality operator

$= \|V_\gamma^* - FV_{n+1} + FV_{n+1} - V_{n+1}\|$

$\leq \|V_\gamma^* - FV_{n+1}\| + \|FV_{n+1} - V_{n+1}\|$

$\Rightarrow F V_\gamma^* = V_\gamma^*$

$\therefore \|V_\gamma^* - V_{n+1}\| \leq \|FV_\gamma^* - FV_{n+1}\| + \|FV_{n+1} - FV_n\|$

$\leq \gamma \|V_\gamma^* - V_{n+1}\| + \gamma \|V_{n+1} - V_n\|$

$(1-\gamma)\|V_\gamma^* - V_{n+1}\| \leq \gamma \|V_{n+1} - V_n\|$

$\|V_\gamma^* - V_{n+1}\| \leq \frac{\gamma}{1-\gamma} \times \frac{\epsilon \times 1-\gamma}{2\gamma} = \frac{\epsilon}{2}$

↳ stopping condition

Part 2: $\|V_\gamma^{\pi\epsilon} - V_{n+1}\| = \|V_\gamma^{\pi\epsilon} - FV_{n+1} + FV_{n+1} - V_{n+1}\|$

$\Rightarrow \|V_\gamma^{\pi\epsilon} - V_{n+1}\| \leq \|V_\gamma^{\pi\epsilon} - FV_{n+1}\| + \|FV_{n+1} - V_{n+1}\|$

(Note: $FV_{n+1} = F^{\pi\epsilon} V_{n+1}$)

↓ Since At n, we converge. (Verify yourself)

$\leq \|F^{\pi\epsilon} V_\gamma^{\pi\epsilon} - F^{\pi\epsilon} V_{n+1}\| + \|FV_{n+1} - FV_n\|$

Similar to part 1,

$\|V_\gamma^{\pi\epsilon} - V_{n+1}\| \leq \frac{\epsilon}{2}$

$\therefore \|V_\gamma^* - V_\gamma^{\pi\epsilon}\| \leq \epsilon$ //

---

* Convergence of Value Iteration:

Claim: Monotonicity of convergence

if $FV^n \geq V^n$ then $\forall\, m \geq 0$

$V_{n+m-1} \geq V_{n+m}$

$(F^{m+1} V^n \geq F^m V^n)$

Theorem: Let V_n be the sequence of state values found by VI, then

i) $\|V_n - V_\gamma^*\| \leq \frac{\gamma^n}{1-\gamma} \|V_1 - V_0\|$

ii) $\|V_\gamma^{\pi_\epsilon} - V_\gamma^*\| \leq \frac{2\gamma^n}{1-\gamma} \|V_1 - V_0\|$

Proof approach

$\|V_\gamma^* - V_{n+1}\| \leq \frac{\gamma}{1-\gamma} \|V_{n+1} - V_n\|$ } We have shown this

$\leq \frac{\gamma}{1-\gamma} \|LV_{n-1} - LV_{n-2}\|$

$\leq \frac{\gamma^2}{1-\gamma} \|V_{n-1} - V_{n-2}\|$

$\vdots$

$\leq \frac{\gamma^n}{1-\gamma} \|V_1 - V_0\|$

- Similarly, $\|V_\gamma^{\pi_\epsilon} - V_{n+1}\| \leq \frac{\gamma^n}{1-\gamma} \|V_1 - V_0\|$

$\therefore \|V_\gamma^{\pi_\epsilon} - V_\gamma^*\| \leq \|V_\gamma^{\pi_\epsilon} - V_n + V_n - V_\gamma^*\|$

$\leq \frac{2\gamma^n}{1-\gamma} \|V_1 - V_0\|$

(Convergence slower when γ is close to 1) → Some improvements have been suggested.

(16) Monte Carlo Methods: (for MDPs)

- Monte carlo methods learn from experience
- On-line experience
→ Episodic tasks with reward at the end of each episode.

* • Each occurrence of state s in an episode is called a visit.

↳ Naive approach:

$$\hat{V}_{\pi}(s) = \frac{1}{m}\sum_{i=1}^{m} r_i$$ → Return of T_i (episode)

⇓

For each episode, run π from s for m times. (Actions may be different at each episode)

* MC Approaches:

* i) ↳ First visit monte carlo:

- Average returns for first time s is visited in an episode → $V_{\pi}(s) = \frac{1}{m}\sum_{i=1}^{m} r(s, T_i)$

ii) ↳ Every visit monte carlo:

- Average returns for every time s is visited in an episode.
(Assuming we get rewards between episodes as well)
- $V_{\pi}(s) = \frac{1}{\sum_{i=1}^{m} N_i(s)} \sum_{i=1}^{m} \sum_{j=1}^{N_i(s)} r(s, T_i, j)$

$\sum_{i=1}^{m} N_i(s)$ ⇓ Number of occurrences of s in an episode.

Note: example in slides — Blackjack — First visit MC

⇒ MC does not require full knowledge of the environment unlike DP.

- State values estimates for each state are independent unlike DP.

*MC Action value estimation:

- Useful when underlying model is not available
- For every state-action pair, perform mean reward calculation like earlier. (follow first visit MC)
  - ↳ we need to maintain exploration starts so that we actually visit different state with different trials even in deterministic policies.

*MC Control: (Slides)

- Use estimation & improvement cycles as in policy iteration.
- ? • Policy improvement: $\pi(s) = \max_a V_\pi(s,a)$
- * • We use exploration starts to visit all actions from the start state s to get a better approximation of $V_\pi(s,a)$ ∀a.

⇒ Ideally we want ∞ episodes, but we can allow a tolerance in errors for finite episodes.

Without this we would only update $V_\pi(s,a_i)$ where $\pi(s)=a_i$, so we won't be able to update π properly.

} To fix this issue we follow method similar to ∈-Greedy. (Best arm with ∈, random with 1−∈ etc.)

END – Mid2

---

(17) * • Issues with exploration starts is optimal policy not stable.

* ↳ So we use ∈-Greedy like approach.

↳ Or we could use "Off Policy" where one policy is used as the behaviour policy and the policy being learnt is called the target policy. (exc: behaviour is probabilistic and target is deterministic)

(On Policy is evaluate and improve on the same policy)

Note: Example of first visit monte carlo in slides.

- In "On Policy" methods we need non deterministic policies since we want to have exploration. (ex: ϵ-Greedy)

⇒ "On Policy" & "Off Policy" are two categories.

*1) On Policy First Visit: (ϵ Greedy like)

* • After evaluation of $Q(s,a)$ get $A^* = \underset{Q(s,a)}{\text{argmax}}$ and then set,

$$\pi(a|s) = 1-\epsilon+\epsilon/|A(s)| \quad \text{if } a \doteq A^*$$

else

$$\epsilon/|A(s)| \quad \text{if } a \neq A^*$$

- Evaluation is just average reward over all the runs.

- $$q_\pi(s,\pi_1(s)) = q_\pi(s,A^*)(1-\epsilon) + \sum_a \frac{\epsilon}{|A(s)|} q_\pi(s,a)$$

$$\geq (1-\epsilon)\sum_a \left(\frac{\pi(a|s) - \frac{\epsilon}{|A(s)|}}{1-\epsilon}\right) q_\pi(s,a)$$

$$+ \frac{\epsilon}{|A(s)|}\sum_a q_\pi(s,a)$$

$$= \sum \pi(a|s)\, q_\pi(s,a) = v_\pi(s)$$

$\therefore q_\pi(s,\pi_1(s)) \geq v_\pi(s)$

$\therefore$ State values are $\geq$ older ones.

---

*2) Off Policy Control:

** • If policies are π & μ. Then if action has non-zero probability in π $\pi(a|s) > 0 \Rightarrow \mu(a|s) > 0$

→ Assumption of coverage

- Typically π would be a greedy policy.

$\Rightarrow \pi$ = Target policy
μ = Behaviour policy

---

* Importance Sampling:

$$\mathbb{E}_{x\sim p}[\hat{x}] = \sum_x \left[p(x)\cdot \frac{x\,q(x)}{p(x)}\right] = \mathbb{E}_{x\sim q}[x]$$

$$\left(\hat{x} = \frac{x\,q(x)}{p(x)}\right)$$

- We use importance sampling among π & μ.
- Probability of state–action trajectory under policy π is,

$$P_\pi(\{\{S_t, A_t\}, \ldots, \{S_T\}\})$$

$$= \prod_{k=1}^{T-1} \pi(A_k|S_k)\cdot p(S_{k+1}|S_k, A_k)$$

→ Since its markov

Considering, $\rho_t^T = \dfrac{\prod_{k=1}^{T-1} \pi(A_k|S_k)\cdot p(S_{k+1}|S_k,A_k)}{\prod_{k=1}^{T-1} \mu(A_k|S_k)\cdot p(S_{k+1}|S_k,A_k)}$

$$= \frac{\prod_{k=1}^{T-1} \pi(A_k|S_k)}{\prod_{k=1}^{T-1} \mu(A_k|S_k)}$$

- $\mathcal{T}(s)$ = Set of all time steps where s is visited. For first visit $\mathcal{T}(s)$ is just a value of first time stamp.

- $T(t)$ is the first time of termination following time t.
  $G(t)$ is return after t till $T(t)$

$\Rightarrow \{G_t\}_{t \in T(s)}$ are returns of corresponding to state s. $\{\rho_t^{T(t)}\}_{t \in T(s)}$ are importance sampling ratios.

---

eoc: $s_1, a_1, s_2, a_2, s_1, a_2, s_3, a_3, s_2, a_2, s_1$

First visit: $T(s_1) = 1$, $T(1) = 6$

Every visit : $T(s_1) = \{1, 3, 6\}$, $T(1) = 3$
$T(3) = 6$

*

* • $V(s) = \dfrac{\sum_{t \in T(s)} \rho_t^{T(t)} \cdot G_t}{|T(s)|}$ ← Importance sampling; → Return of policy μ

- This is ordinary importance sampling.

---

(18)

* $\Rightarrow$ It is an unbiased estimate of $V_\pi(s)$

$\downarrow$

$E_\mu[\hat{V}_\pi(s)]$

Proof: $E_\mu[V_\pi(s)] = E_\mu\left[\dfrac{\sum_{t \in T(s)} \rho_t^{T(t)} G_t}{|T(s)|}\right]$

$$= \frac{1}{|T(s)|} \sum_{t \in T(s)} E_\mu\left[\rho_t^{T(t)} \cdot G_t\right]$$

- $E_\mu\left[\rho_t^{T(t)} \cdot G_t\right] = E_\mu\left[\prod_{k=t}^{T(t)-1} \frac{\pi(A_k|S_k)}{\mu(A_k|S_k)} G_t\right]$

$$= E_\mu\left[\prod_{k=t}^{T(t)-1} \frac{\pi(A_k|S_k)}{\mu(A_k|S_k)} \cdot \sum_{R=0}^{T(t)-t-1} \gamma^R R_{t+R+1}\right]$$

$$= \sum_{R=0}^{T(t)-t-1} \gamma^R \cdot E_\mu\left[\prod_{k=t}^{T(t)-1} \frac{\pi(A_k|S_k)}{\mu(A_k|S_k)} \cdot R_{t+t-1}\right]$$

(The Π term is just 1) $\longrightarrow$

$$= \sum_{l=0}^{T(t)-t-1} \gamma^l \left[E_\mu\left(\frac{\pi(A_l|S_l)}{\mu(A_l|S_l)} \cdot R_{t+l-1}\right) \prod_{\substack{k=t \\ R \neq l}}^{T(t)-1} E_\mu\left[\frac{\pi(A_k|S_k)}{\mu(A_k|S_k)}\right]\right.$$

$$= \sum_{l=0}^{T(s)-t-1} \gamma^l \mathbb{E}_\pi [R_{t+l+1}]$$

$$= \mathbb{E}_\pi [G_t | S_t = s]$$

$$\therefore \mathbb{E}_\mu [\hat{V}_\pi(s)] = \mathbb{E}_\pi \left[\sum_{l=0}^{T(s)-t-1} \gamma^l R_{t+l+1} \,\middle|\, S_t = s \right] = V_\pi(s)$$

Note:

$$\text{Var}(X) = \mathbb{E}[X^2] - [\mathbb{E}[X]]^2$$

, $\mathbb{E}[X^2]$ can make this unbounded since $\mathbb{E}[X]$ is bounded

---

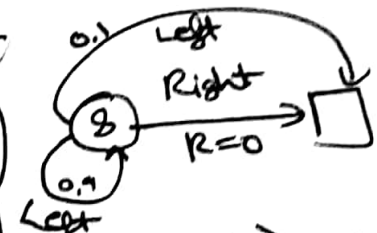
ex:

$$\hat{V}_\pi(s) = \prod_{t=0}^{T-1} \frac{\pi(A_t|S_t)}{\mu(A_t|S_t)} \cdot G_0$$

$$\mathbb{E}_\mu \left[(\hat{V}_\pi(s))^2 \right] = \mathbb{E}_\mu \left[\left(\prod_{t=0}^{T-1} \frac{\pi(A_t|S_t)}{\mu(A_t|S_t)} G_0 \right)^2 \right]$$

Consider trajectory: ← Consider this example

S, left, 0, S, left, 0,
S, left, 0, S, left, +1



$\pi(\text{left}|s) = 1$

$\mu(\text{left}|s) = 1/2$

$$\Rightarrow \mathbb{E}_\mu [(\hat{V}_\pi(s))^2] = \underbrace{\frac{1}{2} \times 0.1 \times \left(\frac{1}{0.5}\right)^2}_{\text{Length 1 trajectory}} + \underbrace{\frac{1}{2} \times 0.9 \times \frac{1}{2} \times 0.1 \times \left(\frac{1}{0.9}\right)^2}_{\text{Length 2 trajectory}}$$

$$+ \dots$$

$$= 0.1 \sum_{k=0}^{\infty} (0.9)^k \cdot 2^k \cdot 2 = \infty$$

(Variance is not bounded using ordinary importance sampling)

---

* Weighted Importance Sampling:

* • $\mathbb{E}_{\mu}[\hat{v}_{\pi}(s)] = \mathbb{E}_{\mu}\left[\frac{\sum_{t \in \mathcal{T}(s)} \rho_t^{T(t)} \cdot G_t}{\sum_{t \in \mathcal{T}(s)} \rho_t^{T(t)}}\right] = v_{\mu}(s)$

∴ This is not an unbiased estimator like earlier.
(Variance however converges to zero here)

Note: Off Policy every visit MC evaluation algo in slides.
** (Both with π given & unknown π) ⇒ Initialized randomly & improved.

* Discounting Aware Importance Sampling:

* • Why should G_0 be weighted by $\prod_{k=0}^{T-1} \frac{\pi(A_k|S_k)}{\mu(A_k|S_k)}$,
instead use $\frac{\pi(A_0|S_0)}{\mu(A_0|S_0)}$? → Could have high variance

⇒ Flat Partial Returns:

$$\bar{G}_t^h = \sum_{k=1}^{h} R_{t+k}, \quad 0 \le t < h \le T$$

$$G_t = (1-\gamma)R_{t+1} + \gamma(1-\gamma)(R_{t+1}+R_{t+2}) + \gamma^2(1-\gamma)(R_{t+1}+R_{t+2}+R_{t+3}) + \cdots + (1-\gamma)\gamma^{T-t-1}(R_{t+1}+\cdots+R_T)$$

γ = Prob of termination at a given timestep.

$$G_t = (1-\gamma)\sum_{h=t+1}^{T-1} \gamma^{h-t-1} \bar{G}_t^h + \gamma^{T-t-1}\bar{G}_t^h$$

• Ordinary importance sampling:

$$\hat{V}(s) = \frac{\sum_{t=\mathcal{T}(s)} \left\{(1-\gamma)\sum_{h=t+1}^{T(t)-1} \gamma^{h-t-1}\rho_t^h \bar{G}_t^h + \gamma^{T(t)-t-1}\rho_t^{T(t)-1} G_t\right\}}{|\mathcal{T}(s)|}$$

$\bar{\rho}_t^h = \prod_{k=t}^{h} \frac{\pi(A_k|S_k)}{\mu(A_k|S_k)}$; similarly we can define weighted sampling.

(19) Partially observable MDPs:

* • Not completely sure about which state it is in.
  $\langle S, A, T, R, \Omega, O, \gamma \rangle$ (Ω: set of observations; O: observation probabilities)
  • We move from belief state b to b' on taking action a and getting an observation o.

$$\therefore b'(s_j) = P(s_j | o, a, b) = \frac{P(o|s_j, a) \sum P(s_j|s_i, a) \cdot b(s_i)}{\sum_{s_j} P(o|s_j, a) \sum P(s_j|s_i, a) \cdot b(s_i)}$$

$b'(s_j)$: state estimator

⇒ Converting POMDPs to belief-MDPs:

** $\langle S, A, T, R, \Omega, O, \gamma \rangle$

↓

$\langle B, A, P, \rho, \gamma \rangle$ } Belief MDP

- $\rho(b, a) = \sum_{s \in S} b(s) \cdot R(s, a)$
- $P(b'|b, a) = \sum_{o \in O} P(b'|b, a, o) \cdot P(o|a, b)$

↳ Discretizing states:

- Belief state needs to be discretized, so that Belief MDP has finite number of states.
- This would however lead to an exponential number of states compared to POMDPs. → Practically infeasible generally

(we can use bellman equation to solve this)

* Policy Tree:

* $V_\pi(b) = \sum_{s_j} b(s_j) \cdot V_\pi(s_j) = \underbrace{b}_{\text{Vector product}} \cdot \underbrace{V_\pi}_{\alpha_\pi \text{ (Representation)}}$

$\rightarrow V_*(b) = \max_\pi V_\pi(b)$

} Policy trees

- $V_\pi(s) = R(s, a(\pi)) + \gamma \sum_{s' \in S} T(s'|s, a(\pi)) \cdot \sum_{o \in O} T(s', a(\pi), o) \cdot V_{o(\pi)}(s')$
- This is similar to DP in MDPs, it is simpler to compute than $V_\pi(b)$ and hence can be used in practice.
- $V_\pi(b)$ is linear in b, however taken $V_*(b)$ will be a combination of policies (This shows that $V_*(b)$ is piecewise linear function & will be convex) → (Not proved completely though)

Note: Witness algorithm

---

(20) TD Learning: (Temporal Difference)

* • Apply updates at every epoch and not have to wait for an entire episode.
  - $V(S_t) = V(S_t) + \alpha\left(r_{t+1} + \gamma V(S_{t+1}) - V(S_t)\right)$

A naive TD approach. → [TD(0)]

Note: Sampling & Bootstrapping in DPs Monte Carlo & TD. (Slides)

Note: TD(0) estimation of state values (in slides)
↳ Generates sequence of states in a run & apply state value updates considering state s & next state s'.

$$\ast \rightarrow \underbrace{G_t - V(S_t)}_{\text{Monte Carlo error}} = R_{t+1} + \gamma G_{t+1} - V(S_t)$$

$$= R_{t+1} + \gamma G_{t+1} + \gamma V(S_{t+1}) - \gamma V(S_{t+1}) - V(S_t)$$

$$= \delta_t + \gamma (G_{t+1} - V(S_{t+1}))$$

δ_t: TD Error

$$= \delta_t + \gamma [\delta_{t+1} + \gamma (G_{t+2} - V(S_{t+2}))]$$

$$= \delta_t + \gamma \delta_{t+1} + \gamma^2 (G_{t+2} - V(S_{t+2}))$$

$\therefore G_T - V(S_T) = 0$ //

$$= \sum_{k=1}^{T-t-1} \gamma^k \delta_{t+k}$$

- TD converges with probability 1 if we use decreasing alphas (with certain properties)
- Batch updating in TD & MC: Average updates over a batch of episodes & update actual state values with these averages

we can use incremental batch sizes.

$$\Rightarrow Q(S_t, A_t) = Q(S_t, A_t) + \alpha (R_{t+1} + \gamma \{Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)\}$$

$\ast$ ↑ Updating Q values using TD.

---

* Sarsa: (On-Policy TD Control)

- Apply Q value updates to greedily find best actions to update policy iteratively.
- ϵ-greedy to choose best action or random action.
- $Q(S_t, A_t) = Q(S_t, A_t) + \alpha(R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t))$

21

Note: GLIE (Greedy in the Limit with Infinite Exploration)

↳ ϵ-Greedy falls in this

↳ All state action pairs visited ∞ times

↳ Converges to a solution.

- SARSA — State Action Reward State Action
- Algo in slides — SARSA is GLIE sequence of policies
- SARSA converges if $\sum_{t=1}^{\infty} \alpha_t = \infty$ & $\sum_{t=1}^{\infty} \alpha_t^2 < \infty$

---

* Q-Learning: (Off policy TD control)

*

- $Q_t(S_t, A_t) = (1-\alpha) Q_{t-1}(S_t, A_t) + \alpha [R_{t+1} + \gamma V_{t-1}(S_{t+1})]$

where $V_{t-1}(S_{t+1}) = \max_a Q_{t-1}(S_{t+1}, a)$

↑ Since we do this it is different from SARSA where we use observed action a' instead.

- This is off policy since updates use older policy and then the policy changes as we move on. (Updates use $t-1^{th}$ policy, whereas selection uses t^{th} policy)

* ↑ Due to this it is off policy

- Q-Learning is more aggressive than SARSA since we use the max term.

⇒ In examples we keep proceeding in episodes until we reach terminal states & then apply all the updates.

---

*Convergence of Q-Learning:

Define: Contraction operator H as follows,

$$HQ(s,a) = \sum_{s' \in S} p(s'|s,a)\left[r(s,a,s') + \gamma \max_{a' \in A(s')} Q(s',a')\right]$$

$H: \mathbb{R}^d \to \mathbb{R}^d$, $d = |S| \times |A|$

Lemma: H is a contraction operator in the ∞ norm

$$\|HQ_1 - HQ_2\|_\infty \leq \gamma \|Q_1 - Q_2\|$$

$Q_1, Q_2 \in \mathbb{R}^{|A| \times |S|}$

Proof:

$$\|HQ_1 - HQ_2\|_\infty = \max_{s,a} \left| \sum_{s' \in S} p(s'|s,a) \left[r(s,a,s') + \gamma \max_{a'} Q_1(s',a') - r(s,a,s') - \gamma \max_{a'} Q_2(s',a') \right] \right| \rightarrow (1)$$

Note: $\|v_1 - v_2\|_\infty = \max_{i \in [d]} |v_1^i - v_2^i|$

$$\therefore \|HQ_1 - HQ_2\|_\infty = \max_{s,a} |HQ_1(s,a) - HQ_2(s,a)|$$

From (1), $\|HQ_1 - HQ_2\|_\infty = \max_{s,a} \left| \gamma \sum_{s'} p(s'|s,a) \left[\max_{a'} Q_1(s',a') - \max_{a'} Q_2(s',a') \right] \right|$

$$\leq \max_{s,a} \gamma \sum_{s'} \left| p(s'|s,a) \left(\max_{a'} Q_1(s',a') - \max_{a'} Q_2(s',a') \right) \right|$$

$$\leq \max_{s,a} \gamma \sum_{s'} p(s'|s,a) \max_{a'} |Q_1(s',a') - Q_2(s',a')|$$

$$\leq \max_{s,a} \gamma \sum_{s'} p(s'|s,a) \max_{s',a'} |Q_1(s',a') - Q_2(s',a')|$$

$$\leq \max_{s,a} \gamma \sum_{s'} p(s'|s,a) \|Q_1 - Q_2\|_\infty$$

$$= \gamma \|Q_1 - Q_2\|_\infty \quad //$$

Theorem: The random process $\{\Delta_t\}$ taking values in $\mathbb{R}^n$ and defined as,

$$\Delta_{t+1}(x) = (1-\alpha_t(x))\Delta_t(x) + \alpha_t(x) F_t(x) - \Delta_t$$

converges to 0 with prob 1 under the following assumptions,

i) $0 \le \alpha_t \le 1$, $\sum_{t=1}^{\infty} \alpha_t = \infty$, $\sum_{t=1}^{\infty} \alpha_t^2 < \infty$

ii) $\| \mathbb{E}[F_t(x) | \mathcal{F}_t] \|_w \le \gamma \| \Delta_t \|_w$ with $\gamma < 1$

iii) $\text{Var}[F_t(x) | \mathcal{F}_t] \le C[1 + \|\Delta_t\|_w^2]$ for $C > 0$

($\mathcal{F}_t$: History till time t)

* Proof not required (Slides has a slightly different version)
* Step sizes & how they satisfy these.
* General theorem for stochastic approximation

(22) "Almost surely" — Law of large numbers ↓ Converges with probability 1.

---

* Convergence proof of Q-Learning:

* Let $\{Q_t(s,a)\}_{t \ge 0}$ denote the sequence of state-action value functions generated by Q-learning algo.

- $Q_{t+1}(s,a) = Q_t(s,a) + \alpha_t(s,a)\left[r(s,a) + \gamma \max_{a'} Q_t(s',a') - Q_t(s,a)\right]$

$$= Q_t(s,a) + \alpha_t(s,a)\left[r(s,a) + \gamma \mathbb{E}_{s' \sim P_t[s'|s,a]} \left(\max_{a'} Q_t(s',a')\right) - Q_t(s,a)\right]$$

$$+ \alpha_t(s,a)\left[\gamma \max_{a'} Q_t(s',a') - \gamma \mathbb{E}_{s' \sim P_t[s'|s,a]} \left(\max_{a'} Q_t(s',a')\right)\right]$$

$(Q_t \in \mathbb{R}^d)$

- $Q_t(s,a)$ defines a component corresponding to $S_t = s, A_t = a$

$$W_t(s') = \gamma \max_{a'} Q_t(s',a') - \gamma \mathbb{E}_{s' \sim P_t[s'|s,a]} \max_{a'} Q_t(s',a')$$

Now, $\mathbb{E}[W_t | \mathcal{F}_t] = 0$, So this assumption of general theorem for stochastic approximation is satisfied.

$$\Rightarrow Q_{t+1}(s,a) = Q_t(s,a) + \alpha_t(s,a)\left[HQ_t(s,a) - Q_t(s,a) + W_t(s)\right] \longrightarrow (1)$$

$$\left(\because HQ(s,a) = r(s,a) + \gamma \mathbb{E}_{s' \sim P_t(s'|s,a)} \max_{a'} Q_t(s',a')\right)$$

- We can see that (1) represents the formulation as in general theorem for stochastic approx.

$$\Rightarrow |W_t(s')| \leq \gamma \max_{a'} |Q_t(s',a')| + \gamma \left| \mathbb{E}_{s' \sim P_t[s'|s,a]} \max_{a'} Q_t(s',a') \right|$$

$$\leq \gamma \max_{a',s'} |Q_t(s',a')| + \gamma \cdot \max_{a',s'} Q_t(s',a')$$

$$= 2\gamma \max_{a',s'} |Q_t(s,a)| = 2\gamma \|Q_t\|_\infty$$

$\therefore$ All conditions are met.

$\therefore$ It converges.

* Expected SARSA:

Sarsa: $Q_t(s,a) = Q_{t-1}(s,a) + \alpha_t(s,a)\left[R_{t+1}(s,a) + \gamma Q_{t-1}(s',a') - Q_{t-1}(s,a)\right]$

Expected Sarsa:

$$Q_t(s,a) = (1-\alpha) Q_{t-1}(s,a) + \alpha\left[R_{t+1} + \gamma \sum_{a \in A} \pi(a'|s') \cdot Q_{t-1}(s',a')\right]$$

$\Downarrow$

We use probabilities of all actions, unlike sarsa where mostly the best action only contributed.

- We can prove convergence of expected SARSA in a manner similar to convergence of Q-Learning.
- Performs better than SARSA but is computationally more expensive.

---

* Maximization Bias:

- In Q-Learning & Sarsa, max over estimated values is used implicitly as an estimate of the maximum value which leads to a positive bias.
  (Since observed values → max over them is not the mean, but higher)

  ↳ In Q-Learning, early on we choose a lot of wrong actions but later on it's better. Those wrong actions are due to maximization bias.

---

* Double Q-Learning: (Algo in slides)

*. Use 2 action-value functions Q_1 & Q_2

- Do Q learning on both independently, randomly updating Q_1 or Q_2 at each step.
- On updating Q_1, use Q_2 for the value of the next state. Max action, over Q_1 only however.
- Action selections are ϵ greedy wrt sum of Q_1 & Q_2.

---

* Multistep Bootstrapping:

- Updates are not performed at every timestep, but uses a few timesteps together.
- Inbetween TD(0) & Monte-Carlo.

(23) Multistep TD learning:

*. Infinite step TD is monte carlo
↳ 1 update per episode

⟹ n step return:

$$G_t^{(n)} = R_{t+1} + \gamma R_{t+2} + \dots + \gamma^n V_{t+n-1}(S_{t+n})$$

? ⟹ $V_{t+n}(S_t) = V_{t+n-1}(S_t) + \alpha\left[G_t^{(n)} - V_{t+n-1}(S_t)\right]$
(Updates start after n steps)

→ To make up for those n-1 updates we perform n-1 additional updates at the end of the episode.

- In practice from n=1 (TD(0)), as we increase n, MSE drops but after a while it starts to rise again.
- Step size α also has an optima inbetween where it increases as we move away from it.

*⟹ Error reduction property: (Used to show convergence of multi step TD)

$$\max_s \left| E_\pi\left[G_t^{(n)} \mid S_t = s\right] - V_\pi(s) \right| \leq \gamma^n \max_s \left| V_{t+n-1}(s) - V_\pi(s) \right|$$

Proof:

$$G_t^{(n)} = R_{t+1} + \gamma R_{t+2} + \dots + \gamma^n V_{t+n-1}(S_{t+n})$$

$$= G_t - \sum_{k=n}^{\infty} \gamma^k R_{t+k-1} + \gamma^n V_{t+n-1}(S_{t+n})$$

$$\Rightarrow E_\pi\left[G_t^{(n)} \mid S_t = s\right] - E_\pi\left[G_t \mid S_t = s\right]$$

$$= E_\pi\left[\gamma^n V_{t+n-1}(S_{t+n}) - \sum_{k=n}^{\infty} \gamma^k R_{t+k-1}\right]$$

$$= \mathbb{E}_\pi \left[\gamma^n V_{t+n-1}(S_{t+n}) - \gamma^n G_{t+n} \right]$$

$$\because \gamma^n \sum_{k=n}^{\infty} \gamma^{k-n} R_{t+k+1} = \gamma^n \sum_{l=0}^{\infty} R_{(t+n)+l+1}$$

$$= \gamma^n G_{t+n}$$

$$= \gamma^n \mathbb{E}_\pi \left[V_{t+n-1}(S_{t+n}) - G_{t+n} \right]$$

$$\mathbb{E}_\pi \left[G_t^{(n)} \mid S_t = s \right] - V_\pi(s) \quad \because \mathbb{E}_\pi[G_t | S_t = s] = V_\pi(s)$$

$$\therefore \mathbb{E}_\pi \left[G_t^{(n)} \mid S_t = s \right] = \mathbb{E}_\pi \left[\gamma^n V_{t+n-1}(S_{t+n}) - \gamma^n G_{t+n} \right]$$

$$= \mathbb{E}_{A_{t+n} | S_{t+n}} \left[\mathbb{E}_\pi \left[\left(\gamma^n V_{t+n-1}(S_{t+n}) - \gamma^n G_{t+n} \right) \middle| S_{t+n} \right] \right]$$

$$= \mathbb{E}_{s'} \left[\gamma^n V_{t+n-1}(s') - \gamma^n V_\pi(s') \right]$$

$$\therefore \left| \mathbb{E}_\pi \left[G_t^{(n)} \mid S_t = s \right] - V_\pi(s) \right|$$

$$= \left| \mathbb{E}_{s'} \left[\gamma^n V_{t+n-1}(s') - \gamma^n V_\pi(s') \right] \right|$$

$$\leq \mathbb{E}_{s'} \left| \gamma^n V_{t+n-1}(s') - \gamma^n V_\pi(s') \right|$$

$$\leq \max_{s'} \left| \gamma^n V_{t+n-1}(s') - \gamma^n V_\pi(s') \right|$$

Since this holds for all s,

$$\max_s \left| \mathbb{E}_\pi \left[G_t^{(n)} \mid S_t = s \right] - V_\pi(s) \right|$$

$$\leq \max_{s'} \gamma^n \left| V_{t+n-1}(s) - V_\pi(s) \right|$$

Note: n-step SARSA & n-step expected SARSA are done in a similar manner (skip)

⇒ n-Step Off Policy by Importance sampling

$$\rho_{t}^{t+n} = \prod_{k=t}^{\min(t+n-1,\,T-1)} \frac{\pi(A_k|S_k)}{\mu(A_k|S_k)}$$

↓

Off policy n-step TD → $V_{t+n}(S_t) = V_{t+n-1}(S_t) + \alpha \rho_{t}^{t+n} (G_{t:t+n} - V_{t+n-1}(S_t))$

Off policy n-step SARSA → $Q_{t+n}(S_t, A_t) = Q_{t+n-1}(S_t, A_t) + \alpha \rho_{t+1}^{t+n} (G_t^{(n)} - Q_{t+n-1}(S_t, A_t))$

(G_t would vary for expected SARSA)

Note: n-Step tree backup algorithm
