* Access Models

→ Full probabilistic Description.

Costs. P_{ss}^{a}

→ Can't store

→ Information theoretically hard

→ Deterministic Generative model

→ $f(X, u) \rightarrow X'$

Set the random seed. to recreate same scenario

→ Generative model.

Programmatic access

→ Reset Model.

Execute $f(x, u)$ sequences

and anytime reset to x_0.

→ Trace Model

Can't ever go back.

Approximate Dynamic Programming:-
(Neuro-Dynamic Programming)

(1) Too expensive to compute and store value iteration $V(x, t)$

(2) Need weaker access to model
↳ generative model.

Fitted Value Iteration :- (FVI)

Regression Algorithm as oracle.

$$D = \{x_i, y_i\}_{i=1 \ldots m}$$

x_i: $\mathbb{R}$ vector of features {representation of state space}

y_i: output, $\mathbb{R}$ (value)

$f : X \rightarrow Y$

Ex: $\frac{1}{N} \sum_{i=1}^{N} (f(x_i) - y_i)^2$ ~~is small~~

→ Squared error, sensitive to outliers

→ Overfitting

↓

How to fix?

by

→ ① minimize $\underset{P(x,y)}{E}\left[(f(x)-y)^2\right] \leq \varepsilon$

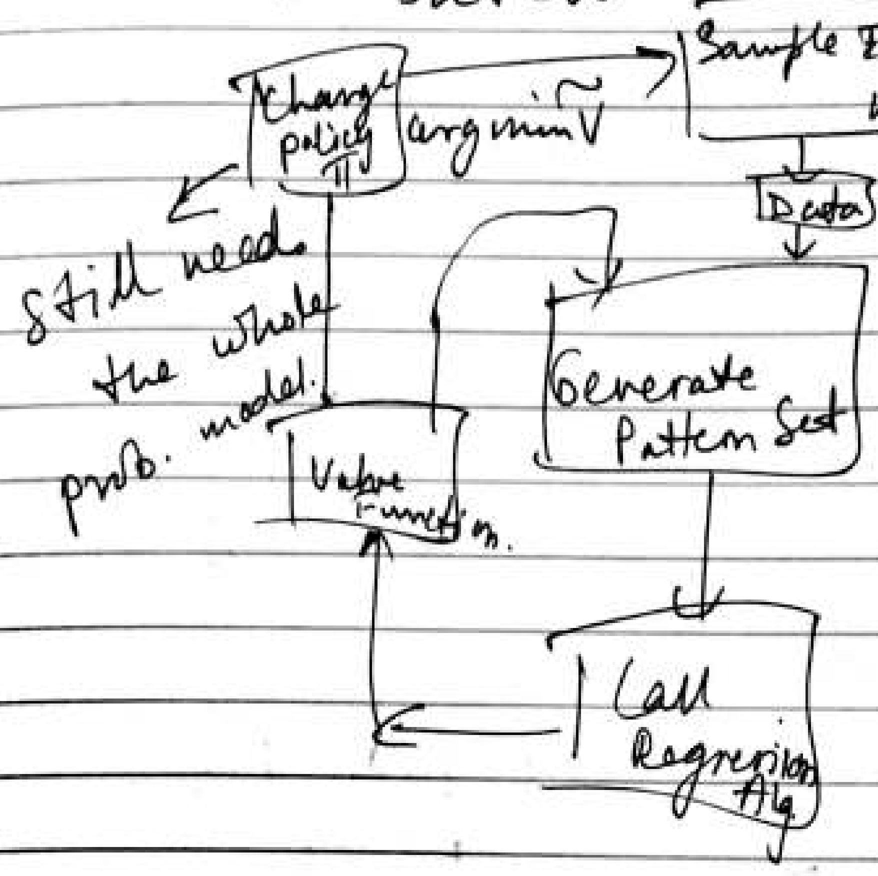
and ② Regularize

What Regression Algos?

Linear Regression.
Locally Weighted Regression.
Gaussian Processes. [or other kernel machines]
Regression Tree.
Random Forests.
Neural Networks.
Splines.

$$V_k(s) = \min_a \; cost(s, a) + \gamma \sum_{s'} p(s' | s, a) \, V_{k+1}(s)$$

* General Batch ADP Data Flow:-



→ Needs only the reset model

Data = $\{States\}_{i=1..m}$

④ Initially get by

① Watch human play

② Watch Random play

③ Randomly populate board.

↳ Some states are not reachable.

But keep in mind that regular Value Iteration also has that problem.

Pattern Set

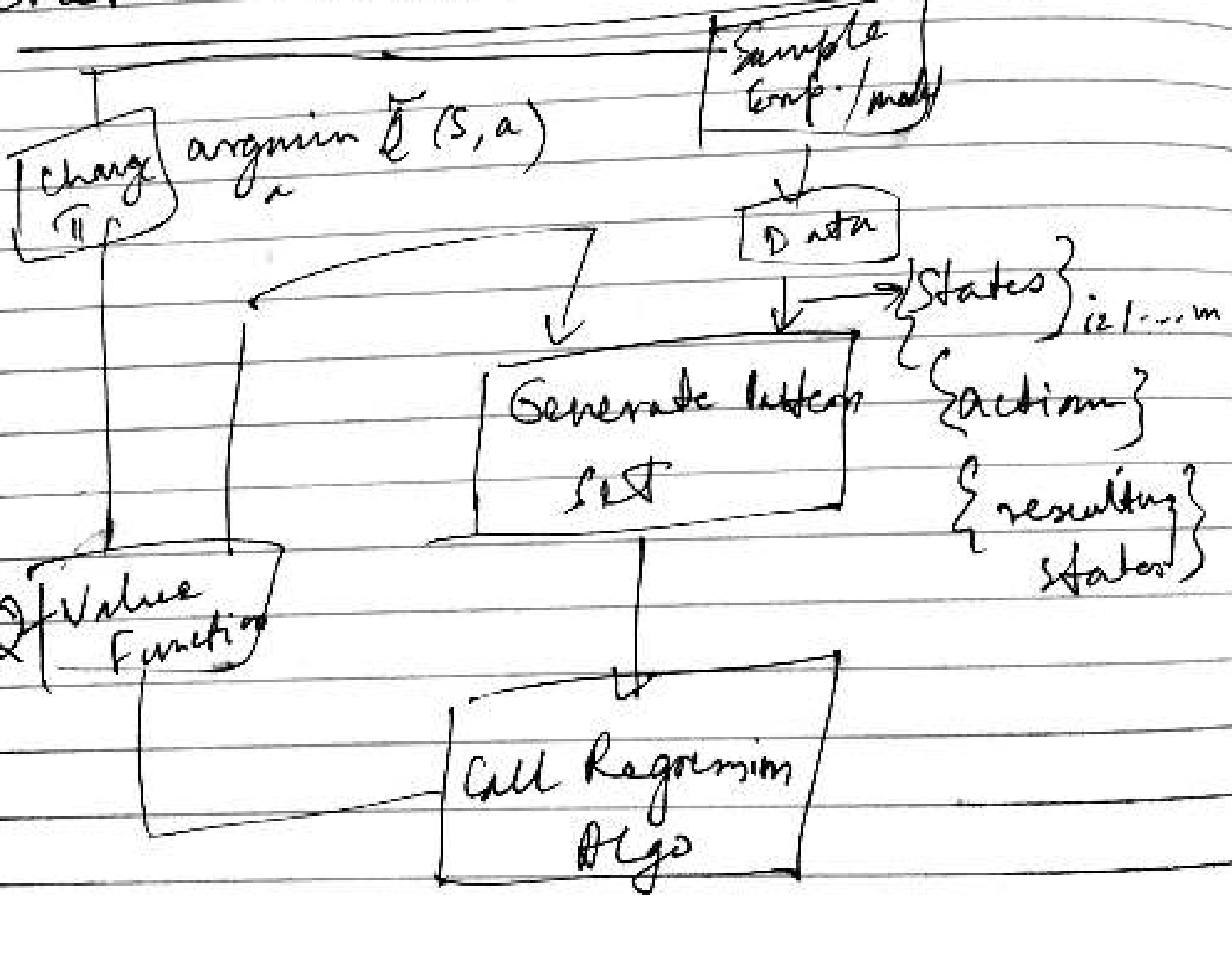
$$\left\{ \left(\underset{\text{States }(X)}{\cdot}, \quad \min_a \text{cost}(s,a) + \sum_{s'} p(s'|s,a) \cdot V_{k+1}(s') \right) \right\}$$

to get V / Value

This is set to 0 for the 1st iteration

(Generative model)

General Batch ADP Data Flow :-



Fitted Q-Iteration (FQI)

Quality Function.

Action-Value Function.

$Q(S, a) \rightarrow$ Value you get if you choose a at 'S' and then follow the optimal policy

$V(S)$

* $Q(S, a) \geq V(S)$
* $V(S) = \min_{a} Q(S, a)$
* $\pi^* = \arg\min_{a} Q(S, a)$

$$\underline{Q(S, a, t)} = \text{cost}(S, a) + \underbrace{\min_{a'} Q(S', a', t+1)}$$

Storing this requires more space

$$= \text{cost}(S, a) + \mathbb{E}_{p(S' | S, a)} \min_{a'} Q(S', a', t+1)$$

Pattern Set

resulting state

$$\left\{ \left(s, a \right),\ cost(s,a) + \gamma \min_{a'} Q^k(s', a') \right\}$$

Just the samples input.
No model.

Fitted Policy Iteration (FPI)

Recap of Policy Iteration

Policy Iteration

(a) Evaluate Policy

Calculate $V^{\pi}(s)$

$V^{\pi}(s) = cost(s, \pi(s)) + \gamma E_{p(s'|s,\pi(s))}[V^{\pi}(s')]$

$Q^{\pi}(s) = cost(s,a) + \gamma E_{p(s'|s,a)}[Q^{\pi}(s', \pi(s'))]$

(b) Improve Policy

$\pi' = \arg\min_a Q^{\pi}(s, a)$

(Note: Don't need to use model)

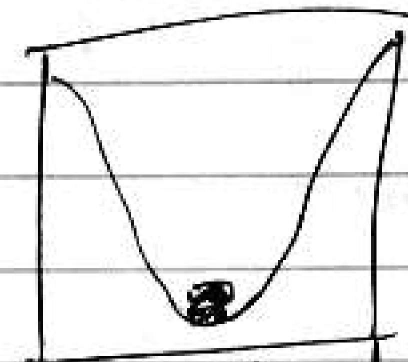
Pattern Set

$$\left\{ (s,a),\ cost(s,a) + \gamma Q^{\pi}(s', \pi(s')) \right\}$$

(Data Flow Diagram is same as FQI)

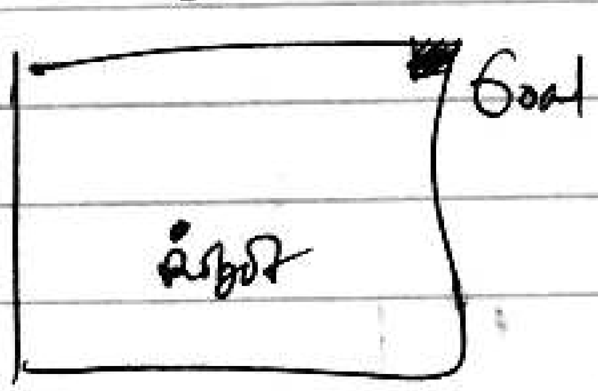
Advantages (wrt FVI)

→ wrt error blow up.
* Stabler than FVI. (as it is linear) in the Evaluate Policy equation.
*
*



* Toy Examples
(Boyan and moore)
→ Car-on-the-Hill problem: → under actuated
→ Non-minimum phase problem
→ You have to go the wrong way to go the right way.

Empty grid with goal at one corner.



Convergence Results?

— Relatively weak.

“The Averagers”

(Gordon et al.)

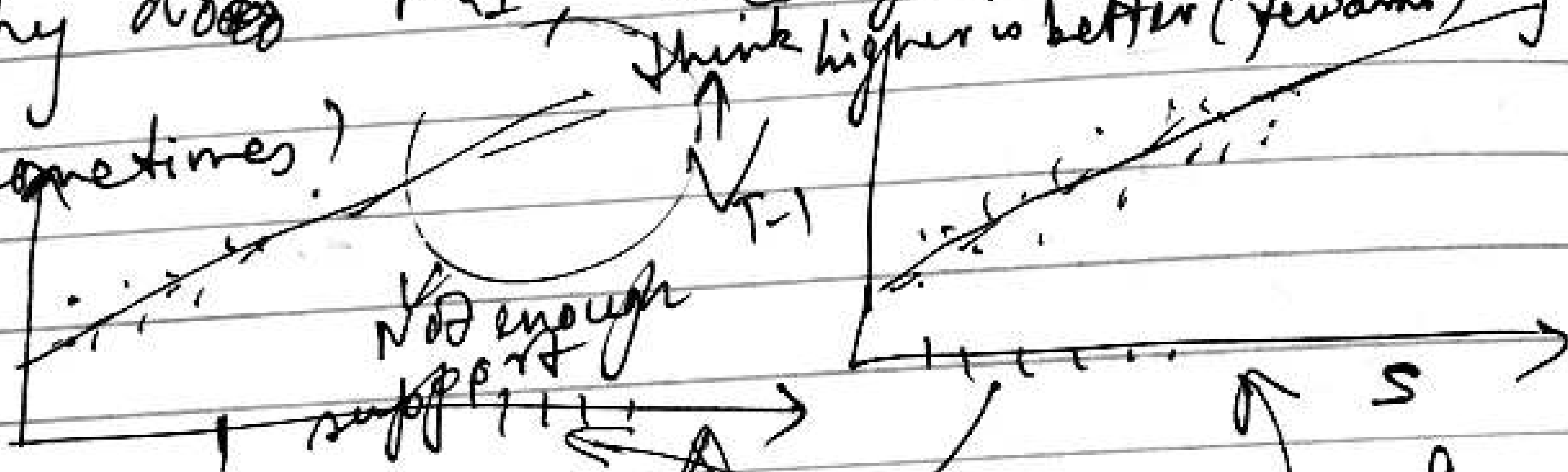
— “On policy” temporal difference ones will have stronger results.

→ Hang on, still coming.

— Error Analysis?

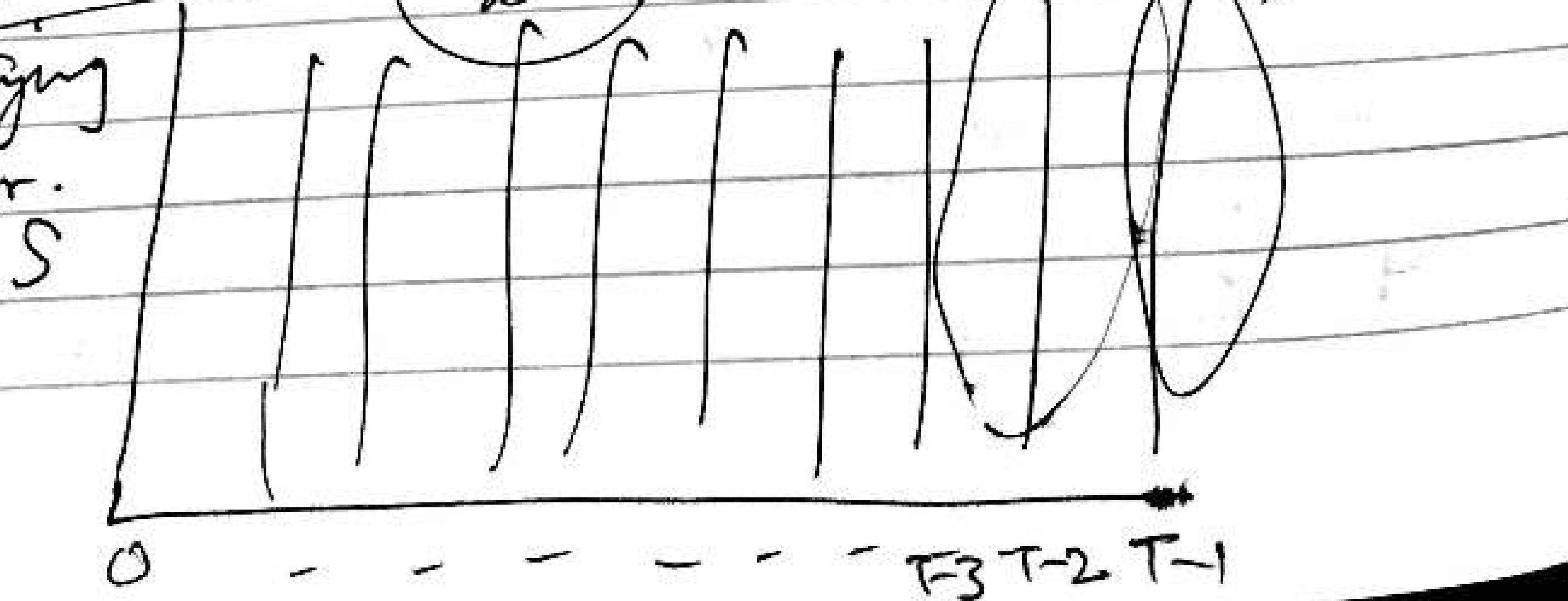
Horrendous.

Why does FQI, go horribly wrong sometimes?



and your $\max_a r(s,a) + \gamma \mathbb{E}_{p(s'|s,a)} \left[V_{T-1}(s') \right.$

Amplifying error.



Problem

① where should sample set be?

② max in value iteration amplifyies errors.

Can Fix?

In policy iteration, the inner loop has no max or min term and hence is more stable.

Recap

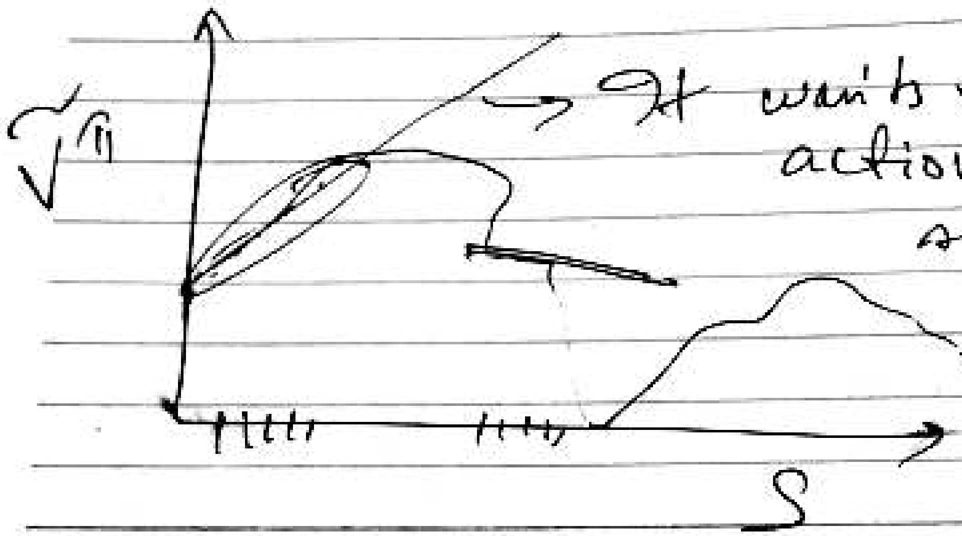
Fitted Dynamic Programming FPI FQI

① Bootstrapping increases error (feedback) we can fix this

② where should the sample set be?

For problem 1

the problem in FPI case
happens in the policy improvement
box. (in the outer loop)



→ It wants to change the actions for every state such that the next state you land in has a higher value.

But you don't have values in that area.

Fix

High level Idea:-

Cache policies not value functions. ← We do this in both FPI and FSA

* An optimal policy has the property that no matter what you do now, hereafter

you must act optimally.

Alg. II to fix

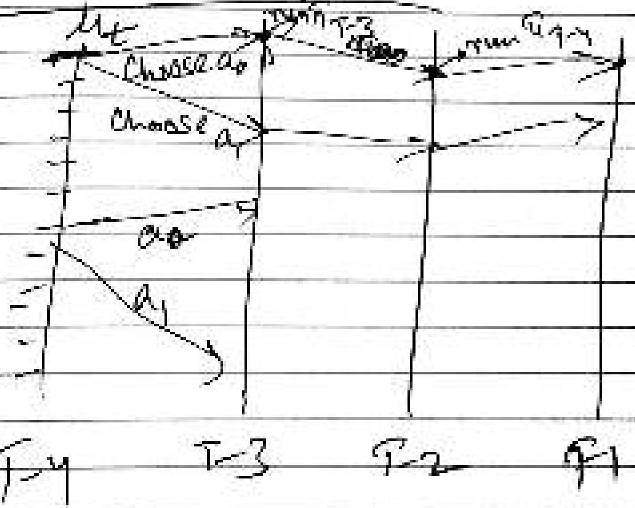
Policy Search by Dynamic Prog. — may not be practical as tied to finite horizon.

for each $t \in T-1, \ldots 0$:

$d_t = \text{GenerateSamples}(\pi_{\{t+1, \ldots T-1\}}, \mu_t)$

$\pi_t = \text{Learn}(d_t)$

Generate Samples



$\tilde{Q}^{\pi}(a_0, S, T-4)$

$\tilde{Q}(a_1, S, T-4)$

Chose the as the higher reward

Learn (d_t)

(1) ~~We can say~~ that this is a classification problem.

Find which state at T-4 maps to which action.

Weight the classification problem as.

$$Q^{\pi}(a_1, s, T-4) - Q^{\pi}(a_0, s, T-4)$$

For multiple weights make this the best action.

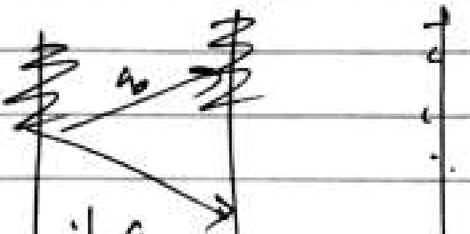
(2) Regression on Q's

$$\pi(s,t) = \arg\max_a \tilde{Q}(s,a,t)$$

* This is different than FVI, FQI,

* This is more expensive as we have to do T times more work [$O(T^2)$] but no bootstrapping - stable.

* Errors in bootstrapping are [illegible] propagated but rollouts are not.



if a_0 goes here then the Q-value it will see will be small and hence won't be selected at that state.

* Performance guarantee

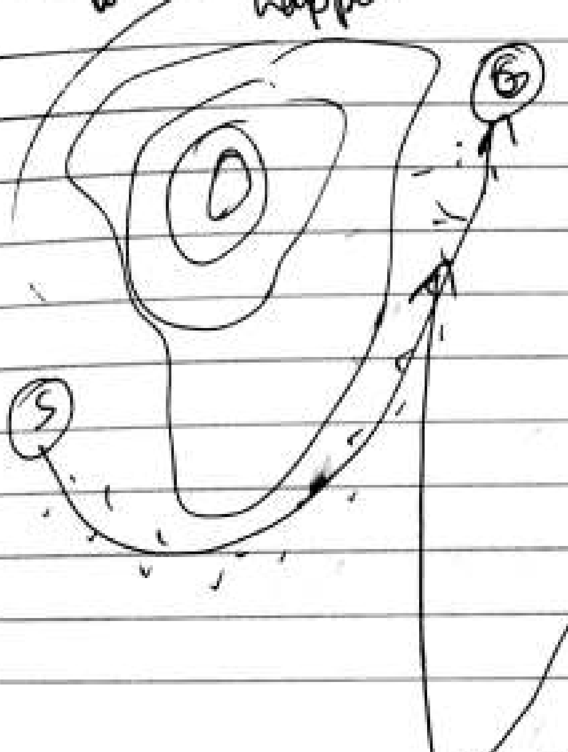
distribution of states induced when you ran π_{ref}

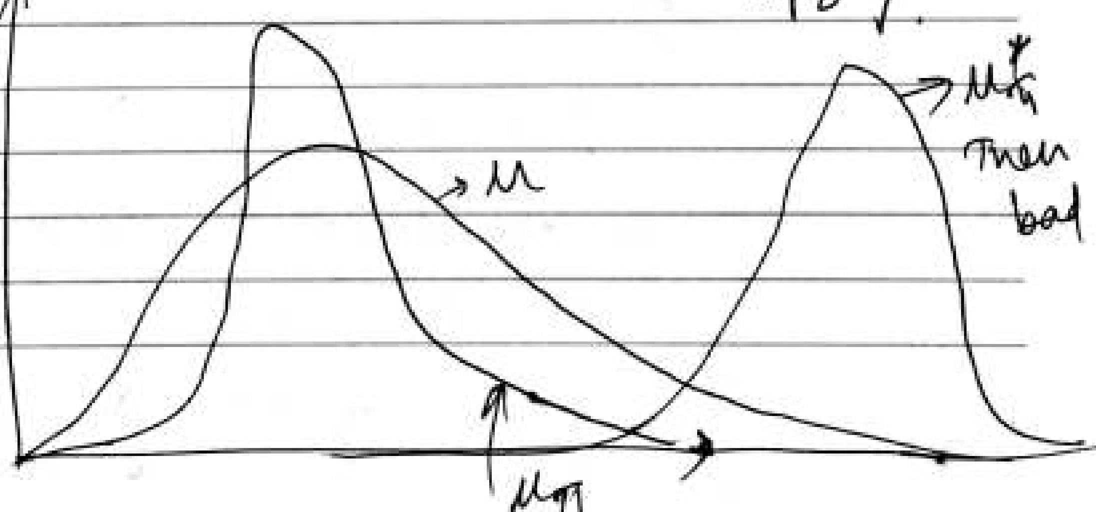
$$V_{\pi_{learned}} \geq V_{\pi_{ref}} - \sum_t \epsilon \left\| \frac{\mu_{\pi_{ref}}}{\mu} \right\|_\infty$$

any policy

How well you are classifying?

will never happen





Algo 2 to fix

Conservative Policy Iteration

Evaluate Policy

Improve Policy

$$\pi' = \arg\max_a \tilde{Q}^\pi(s,a)$$

Conservative

$$\pi_{new} = \alpha \pi' + (1-\alpha)\pi$$

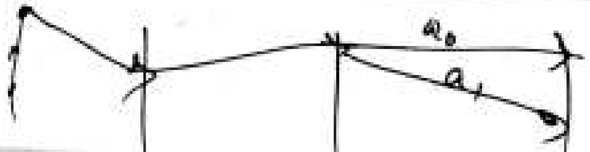
mixing parameter.

where $\alpha \simeq O\left(\frac{1}{T}\right)$

$\tilde{Q}^\pi = ?$

Sample from μ

Run the current policy π_{new} for some # of steps.



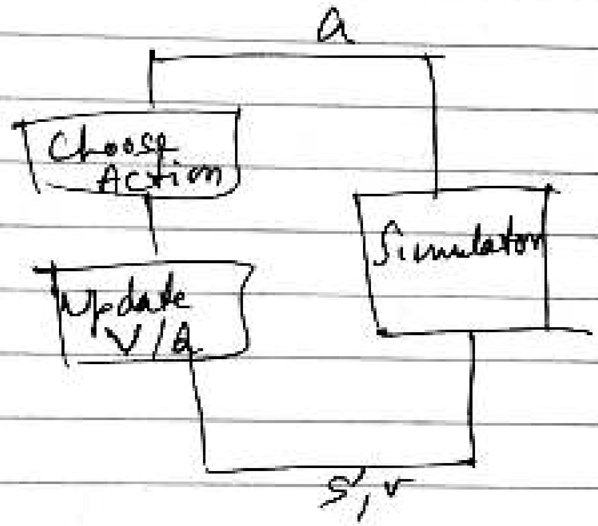
with $(1-\gamma)$ probability you stop and try out actions. γ is the discount factor

Temporal Difference Methods:-

Upto now we have seen batch methods

$$\begin{pmatrix} \text{FVI} \\ \text{FQI} \\ \text{FPI} \end{pmatrix} \begin{matrix} \longrightarrow \\ \longrightarrow \left\{ \begin{matrix} \text{Sarsa} \\ \text{Q-Learning} \end{matrix} \right. \\ \longrightarrow \{ \text{TD} \end{matrix}$$

Overview



* TD: policy evaluation.

given π, estimate $V_\pi(s), \forall s$

Goal Find V s.t. $V_\pi(S) = R(s, \pi(S)) + \gamma \underset{s'}{E}[V_\pi(s')]$