



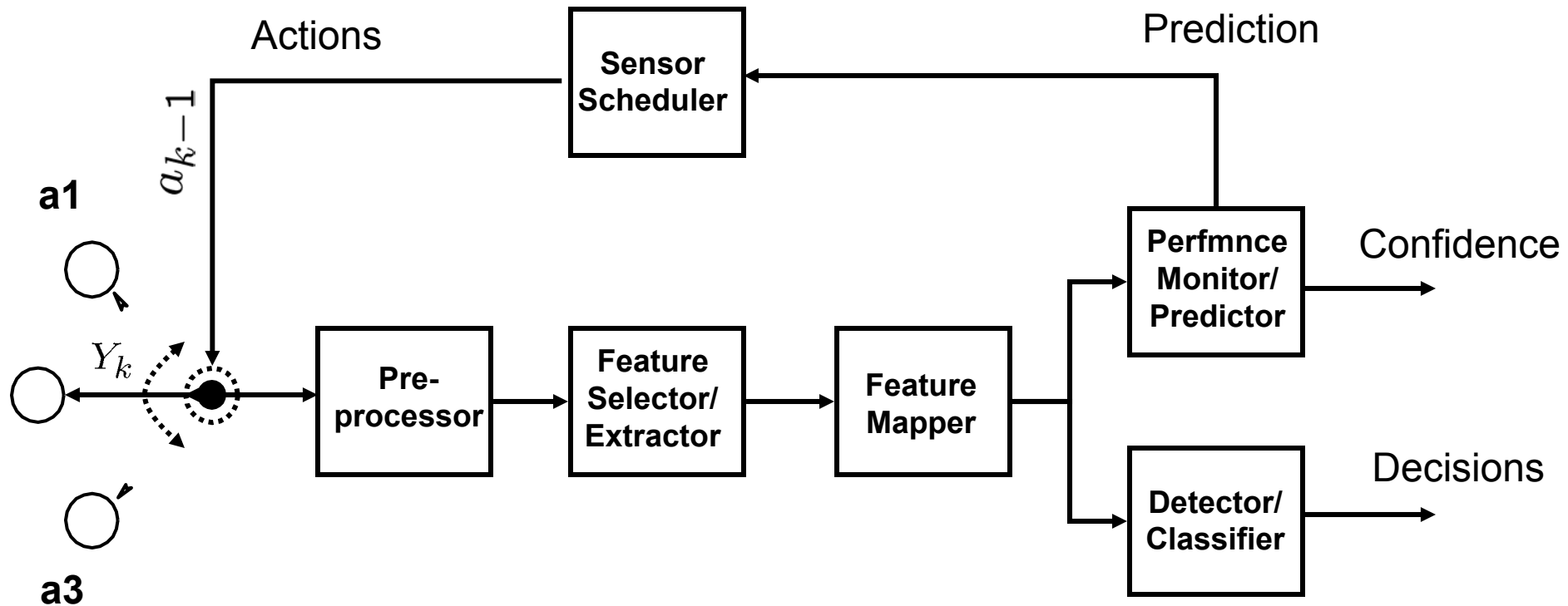
# Classification reduction for active sensing

Alfred Hero and Doron Blatt  
University of Michigan  
Ann Arbor

## Outline

1. Active sensing system
2. Classification Reduction
3. Illustrations and applications
4. Conclusions

# 1. Active Sensing System



# Multistage active sensor management

- Unknown states  $S_k, k = 1, 2, \dots$
- Sensor  $a_k$  acquires data  $y_{k+1}$  having density

$$p(y_{k+1}|S_{k+1}, a_k)$$

- Adaptive sensor scheduling

$$a_k = \varphi_k((y_k, a_k), \dots, (y_1, a_1)) = \varphi_k(Y^k, a^k)$$

- Active sensor management: design policy  $\{\varphi_k\}_k$  to
  - Minimize predicted MSE,  $P_e$ , ( $P_m$ ,  $P_f$ ), time-to-detect, etc.
  - Maximize true alarms under false alarm constraint

- Optimal policy generates action sequence satisfying

$$\max_{a_1, \dots, a_k} E\left[\sum_{j=1}^k \beta_j r(S_{j+1}, a_j)\right]$$

# Special Case: MDP/POMDP

- Information states:  $\pi_k = p(S_{k+1}|Y^k, a^k), k = 1, 2, \dots$
- Q-functions:  $Q_k(\pi, a)$
- Predicted reward:  $\bar{r}(S_{k+1}, a_k) = E[r(S_{k+1}, a_k)|\pi_k]$
- Optimal policy specified by dynamic programming

$$Q_k(\pi_k, a_k) = \bar{r}(S_{k+1}, a_k) + \beta \max_{a_{k+1}} E[Q_{k+1}(\pi_{k+1}, a_{k+1})|\pi_k, a_k]$$

$$a_k = \operatorname{amax}_{a_k} Q_k(\pi_k, a_k), \quad k = 1, 2, \dots;$$

- When “E” not computable: use Monte Carlo methods to learn Q function or optimal policy (Q-learning, stochastic policy search, neurodynamic programming)

# Optimal Classification

- Observations:  $Z_t$
- Space of class labels  $c \in \{0, 1, 2, \dots, L\}$
- True class label  $C$  is random variable with prior  $P(C=c)$
- Observation has conditional density (likelihood model)

$$f(Y_t|C = c)$$

- A classifier is a mapping

$$\hat{C} : Z_t \rightarrow \{0, 1, \dots, L\}$$

- Objective: choose mapping to maximize correct classification probability

$$P(\hat{C}(Z_t) = C) = E[I(\hat{C}(Z_t) = C)]$$

# Optimal classification (ctd)

- Optimal classifier:

$$\begin{aligned}\hat{C}^* &= \operatorname{amax}_{\hat{C}} E[I(\hat{C}(Z_t) = C)] \\ &= \operatorname{amin}_{\hat{C}} E[I(\hat{C}(Z_t) \neq C)]\end{aligned}$$

- Design of classifiers

- For known likelihood model MAP classification rule is optimal
- For unknown model can use empirical risk minimization:
  - Non-parametric density estimation (CART, kNN, MoGs)
  - Kernel methods (SVM, RVM)
  - Ensembles of base learners (Adaboost, Isle, MART)
  - Non-linear regression (RBM, NN)
- These classifiers achieve good performance by combining empirical risk approximation and constraining classifier class
- Main issue: tradeoff btwn fitting the (available) training sample and fitting the (unavailable) test sample
  - Controlling classifier generalization errors

# Weighted Classification

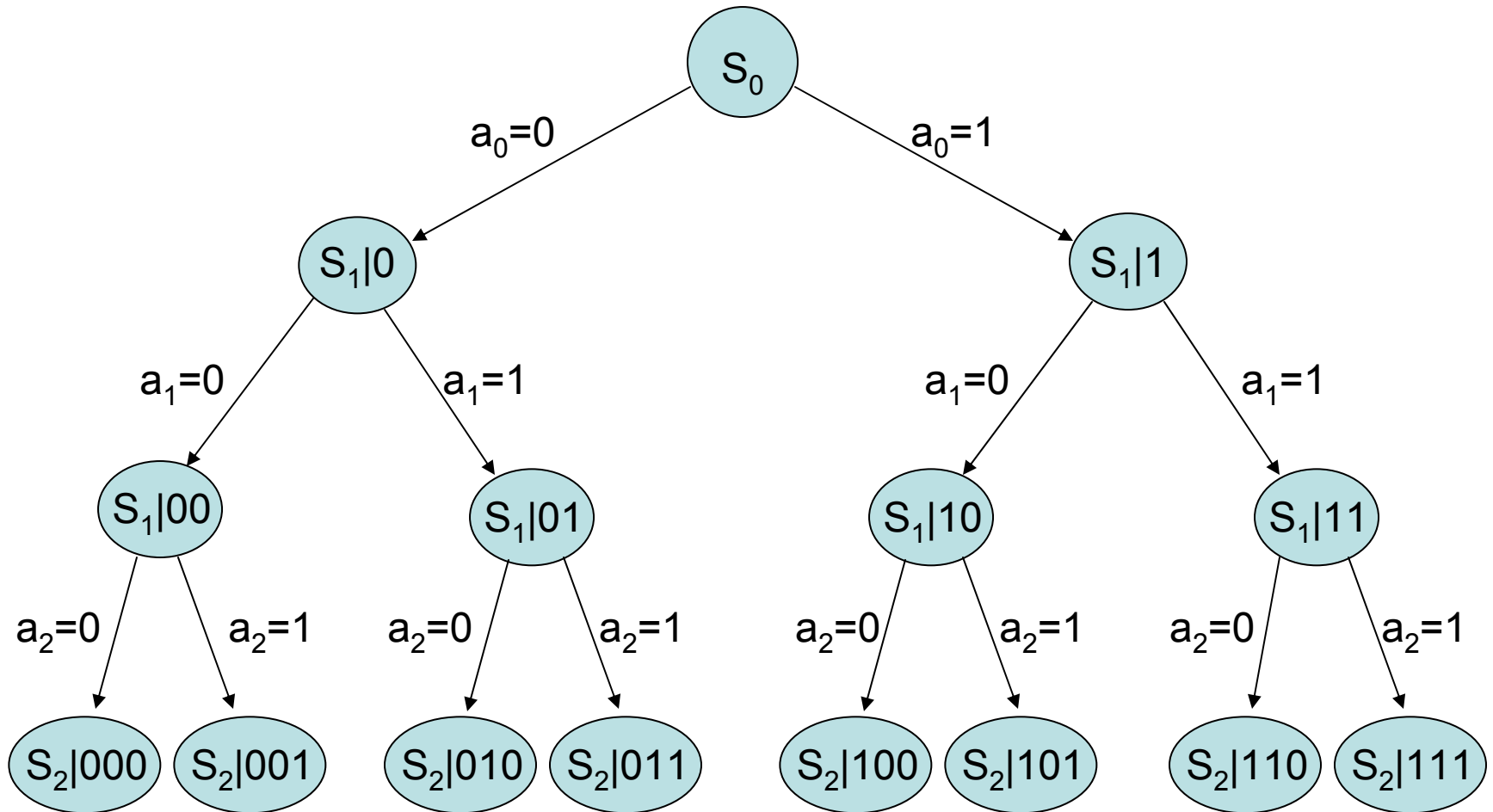
- In many cases different observations are assigned different weight in the design of optimal classifier
- Optimal weighted classifier minimizes risk

$$R(C, \hat{C}) = E[w(Z_t)I(\hat{C}(Z_t) \neq C)]$$

- Examples
  - SVM (Vapnik): support vectors have weight 1 while all others have weight 0
  - Boosting (Freund&Schapire): different measurements are weighted according to their impact on decision boundary
  - ISLE (Freidman): weights determined by importance sampling
  - Adaptive Bayes (Tipping, Carin): weights reflect confidence in each of the training samples

# Binary multistage policy search

Binary action tree of the decision process



Nodes are information states:  $s_k = f(s_k | Y_{k-1}, a_{k-1})$

## 2. Classification reduction of MDPs

- Recall: central driver of MDP dynamic programming soln

$$\bar{r}(S_{k+1}, a_k) = E[r(S_{k+1}, a_k) | Y^k]$$

- Key idea: MDP rewards for sensing actions might be representable as weighted rewards for a related classification problem
  - Define clairvoyant action

$$a_k^o = \operatorname{amax}_a r(S_{k+1}, a)$$

- Clairvoyant actions are unknowable since depend on future state
  - Optimal actions are approximations of clairvoyant actions
- This suggests treating clairvoyant actions as unknown class labels in a related classification problem
  - MDP rewards = correct classification probability
  - Optimal policy = optimal classifier

# Result 1: Classification Reduction

**Theorem** (ref: Blatt&Hero:NIPS05):

Let  $\pi_t$  be the information state,  $S_t$  be a state vector,  $\varphi$  be a policy, and  $a_t = \varphi(\pi_t) \in \mathcal{A}$  be the action resulting from applying the policy to  $\pi_t$  at time  $t$ . Then for any reward function  $r(S_{t+1}, a_t)$ :

$$\text{amax}_{\varphi} E[r(S_{t+1}, \varphi(\pi_t)) | \pi_t] = \text{amin}_{\varphi} E[w(S_{t+1}, C) I(\varphi(\pi_t) \neq C) | \pi_t],$$

where  $C$  is a class label

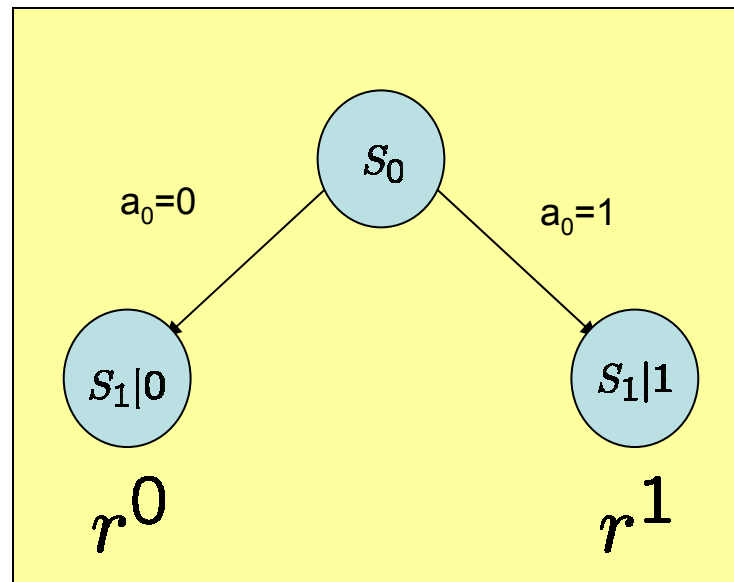
$$C = \text{amax}_{a_t} r(S_{t+1}, a_t),$$

and  $w(\pi_t, c)$  is the regret

$$w(\pi_t, c) = \frac{E[\max_a r(s_{t+1}, a) - r(s_{t+1}, c) | \pi_t]}{\sum_c E[\max_a r(s_{t+1}, a) - r(s_{t+1}, c) | \pi_t]}.$$

## Proof of Theorem:

- For simplicity, specialize to perfect state information, single stage policy, binary states



- Optimal policy is  $\varphi^* = \text{amax}_{\varphi} E[r(s_1, \varphi(s_0)) | s_0]$
- Define rewards

$$r^0 = r(s_1, \varphi(s_0) = 0)$$

$$r^1 = r(s_1, \varphi(s_0) = 1)$$

Proof of Theorem:

- Then have simple representation

$$r(s_1, \varphi(s_0)) = b - |r^1 - r^0| I(\varphi(s_0) \neq \underbrace{\text{amax}_{i=0,1}\{r^i\}}_C)$$

- and hence

$$\text{amax}_{\varphi} E[r(s_1, \varphi(s_0)) | s_0] = \text{amin}_{\varphi} E[|r^1 - r^0| I(\varphi(s_0) \neq C) | s_0]$$

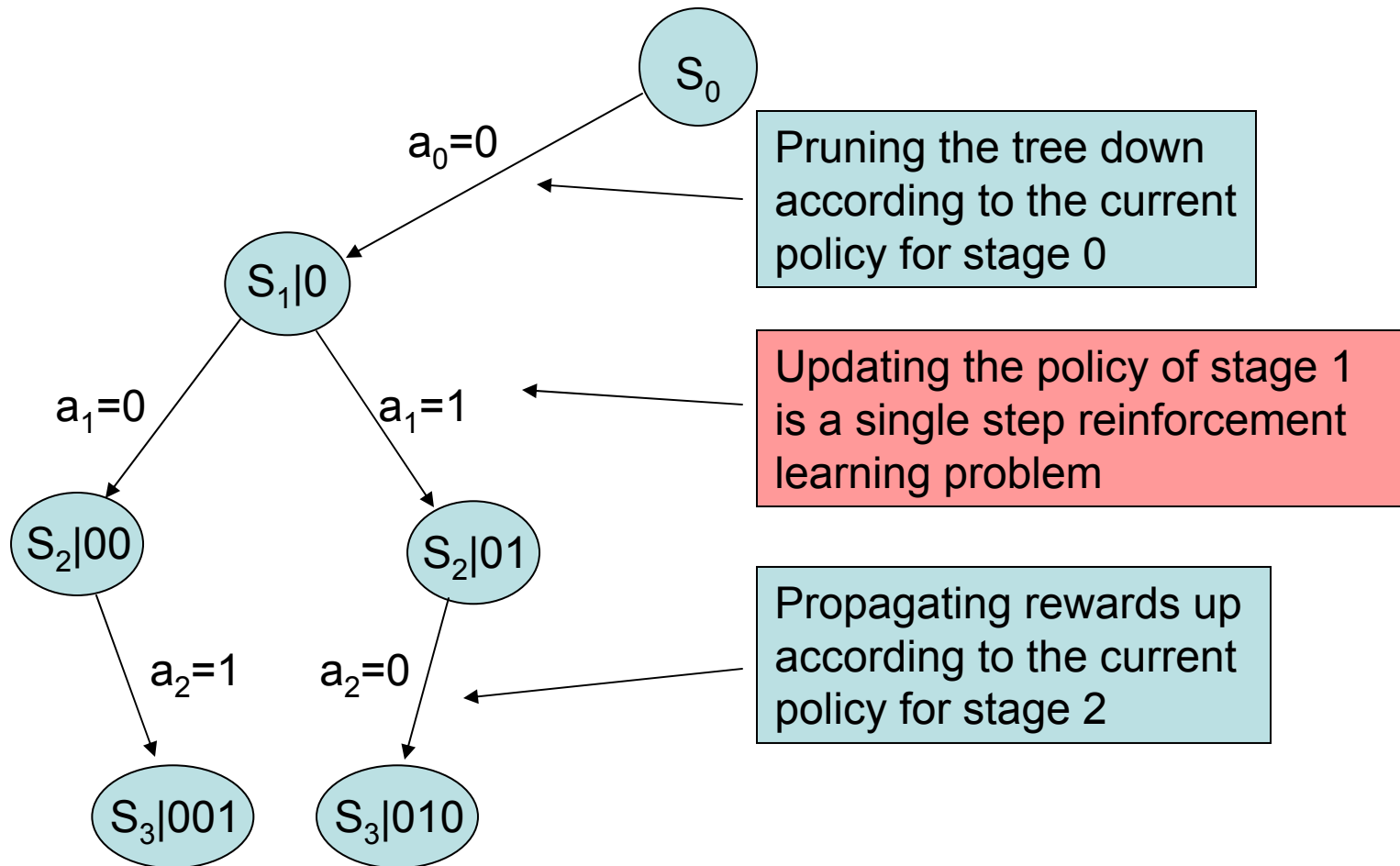
- Therefore optimal policy is equivalent to an optimal weighted classifier with class probabilities

$$P(C = c | s_0) = P(\text{amax}_{i=0,1}\{r^i\} = c | s_0)$$

- and weights

$$w^i(s_0) = \frac{E[\max_j r^j - r^i | s_0]}{\sum_{k=0,1} E[\max_j r^j - r^k | s_0]}$$

# Multistage Gauss-Siedel Implementation



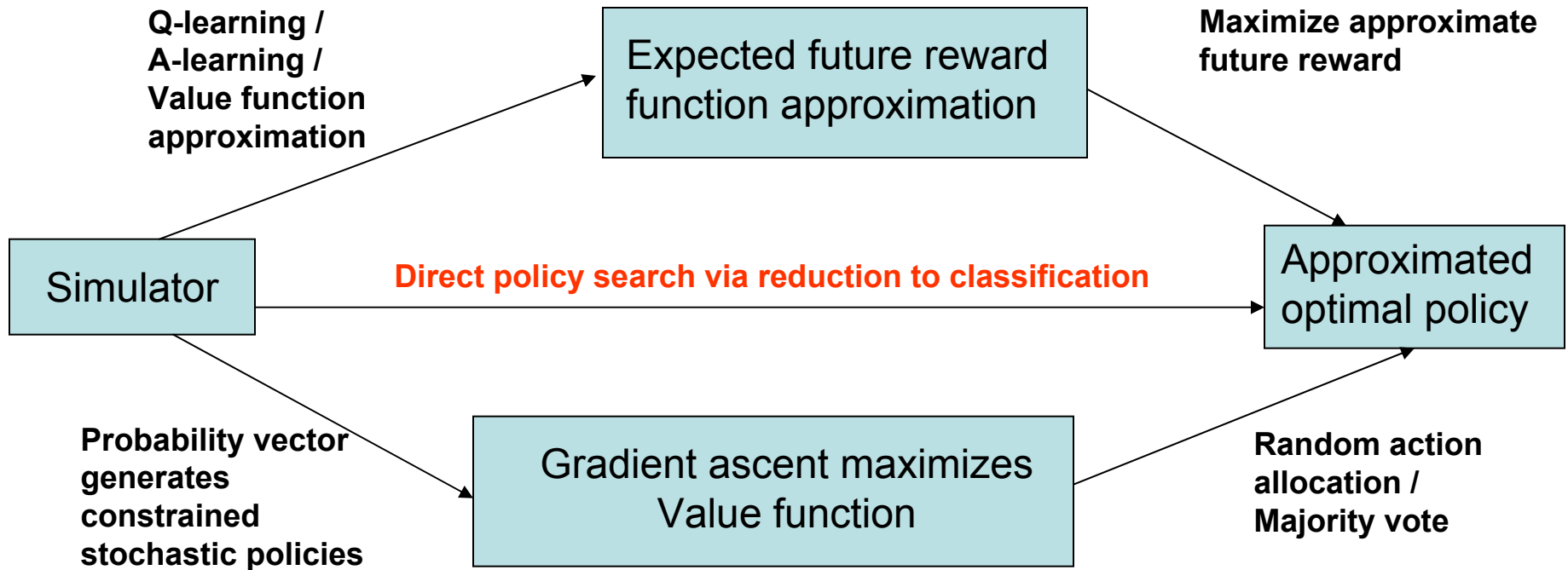
Updating the policy at stage 1 while holding the policies for stages 0 and 2 constant

## Result 2: Convergence of iterative Algorithm

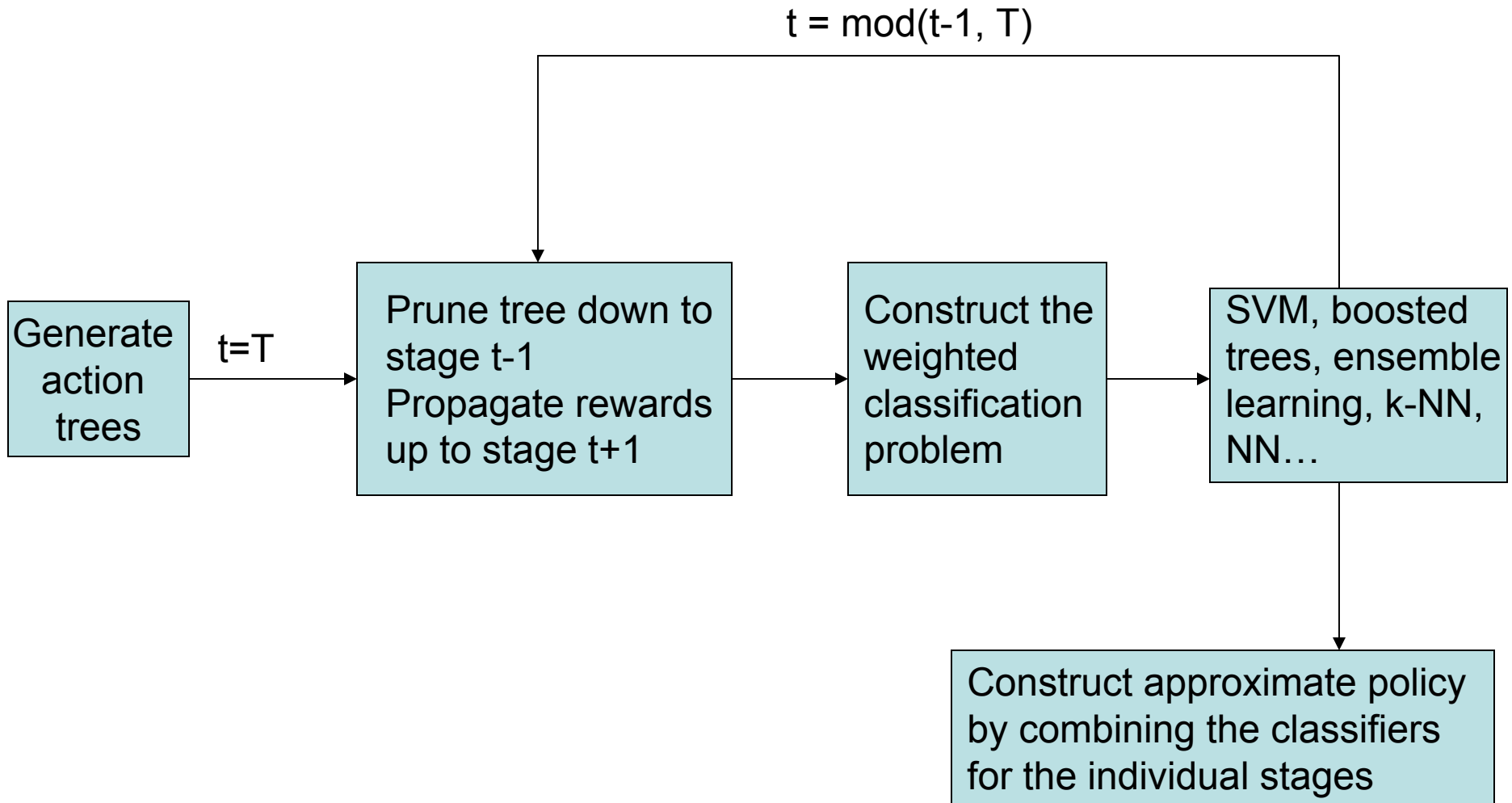
Theorem (Blatt&Hero NIPS2005) :The Gauss Siedel policy search algorithm converges monotonically after a finite number of iterations to a policy that cannot be further improved by changing one of the actions while holding the rest fixed.

# Monte Carlo Implementation

- Generative model assumption:
  - The exact model is unknown
  - It is possible to generate trajectories of the stochastic process while controlling the actions.



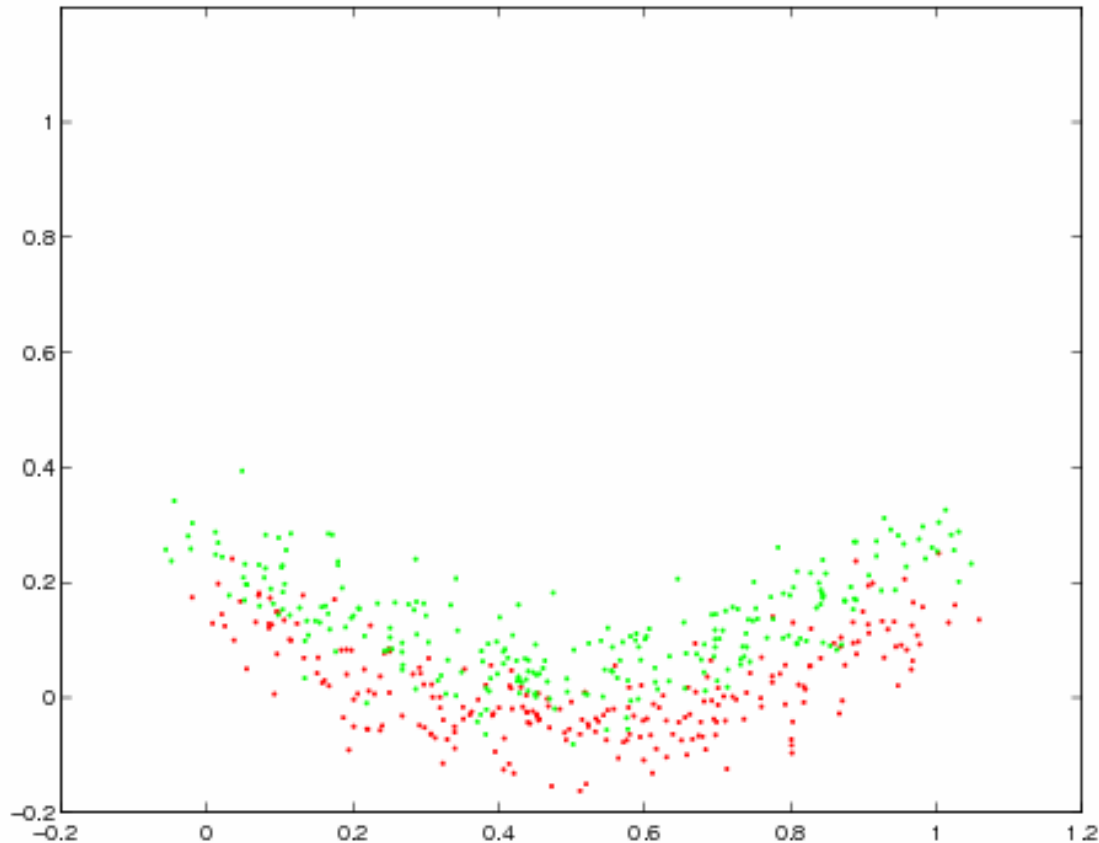
# Block diagram of the method



# Weighted Classification

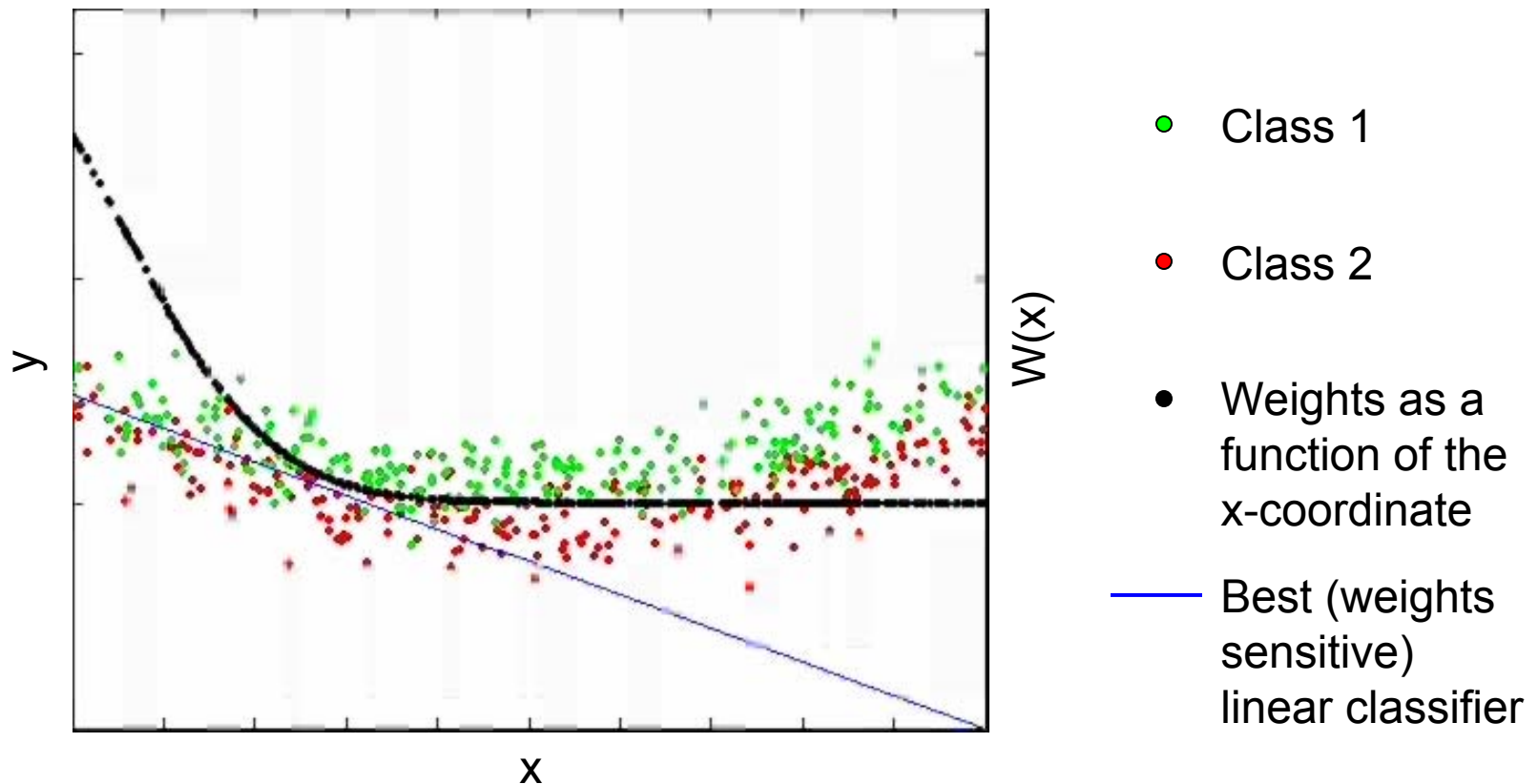
Weighted classification can be performed by:

- Sub-sampling methods.
- Modifying existing classifiers.



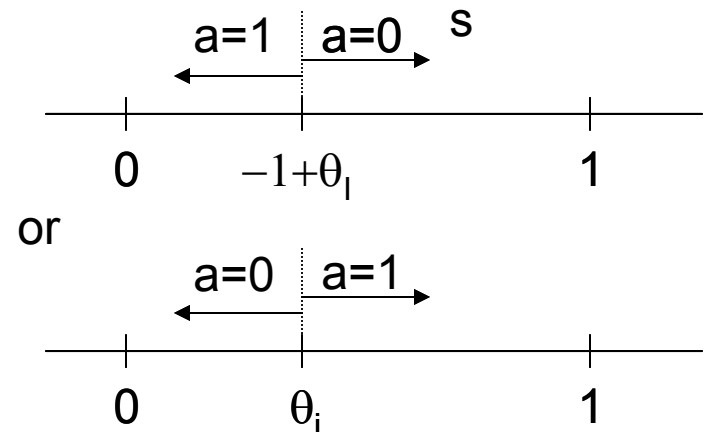
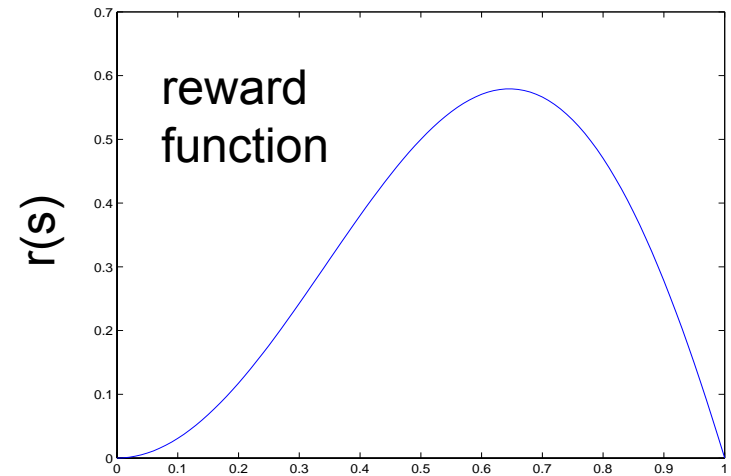
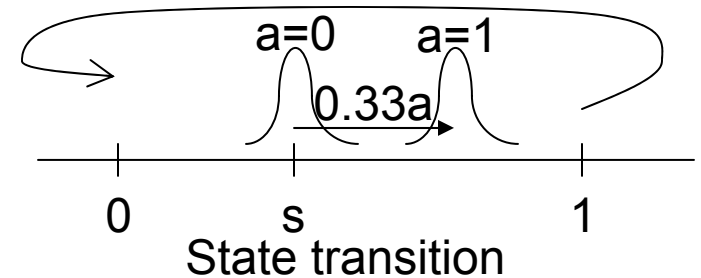
# Weighted Neural Net Classifier

- Neural nets can be easily modified to perform weighted classification:

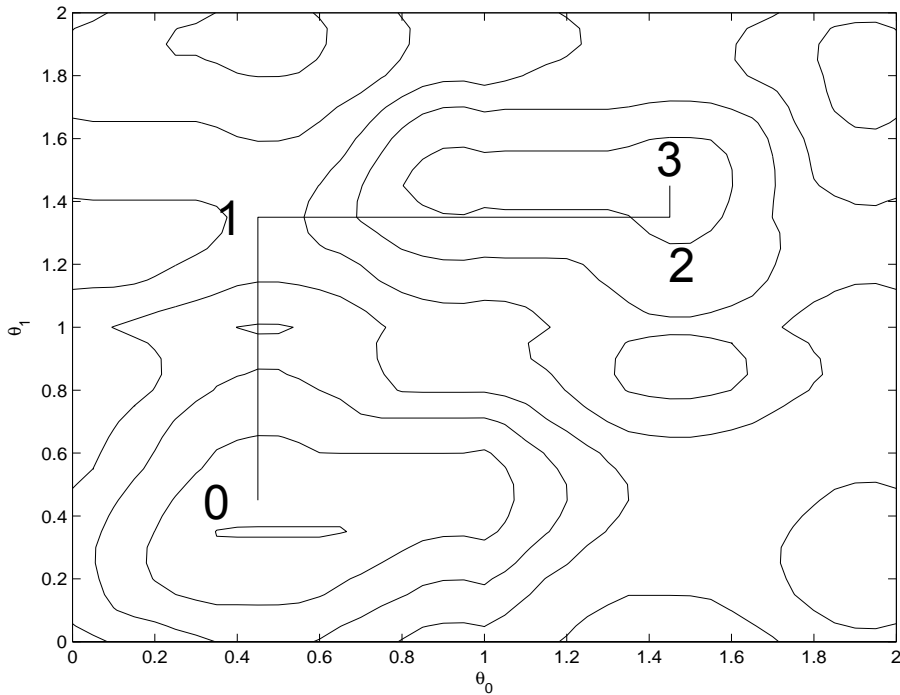


# Two stage MDP example

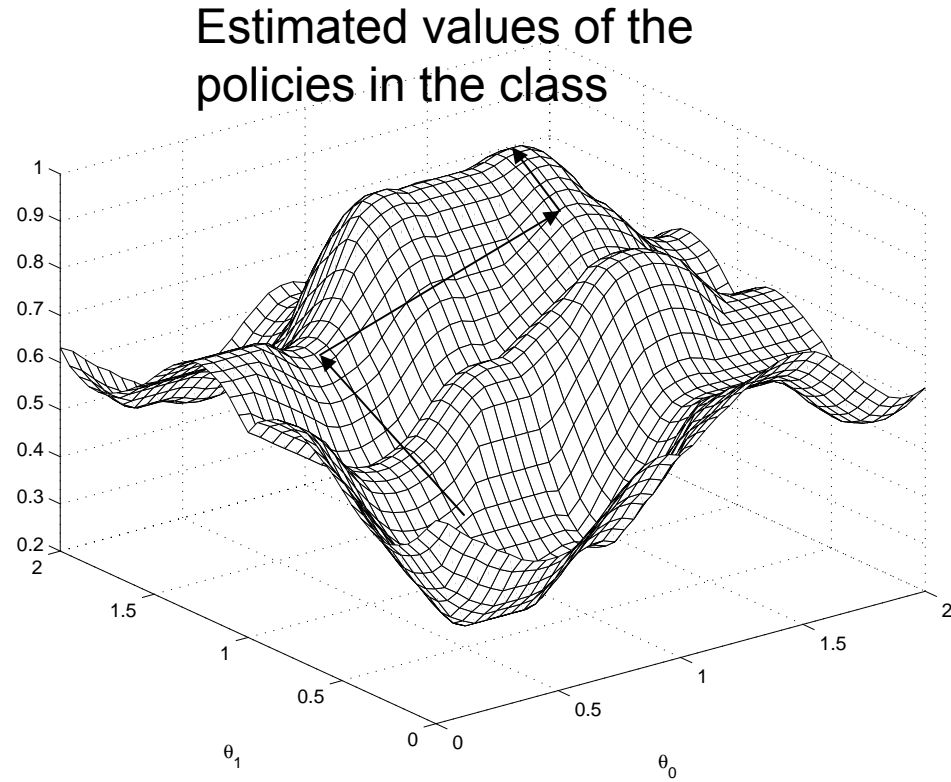
- Continuous state space  $S = [0, 1]$
- Binary action space  $A = \{0, 1\}$
- Two decision stages: Given state  $s$  and action  $a$  the next state  $s'$  is generated by  $s' = \text{mod}(s + 0.33a + 0.1N(0, 1), 1)$ .
- The reward function is given by  $r(s) = s \sin(\pi s)$ .
- We consider a class of policies parameterized by two continuous parameters:
  - $\Pi = \{ \pi(\cdot; \theta) \mid \theta = (\theta_0, \theta_1) \in [0, 2]^2 \}$ ,
  - where for  $i=0, 1$   $\pi_i(s; \theta_i) = 1$  when  $\theta_i \leq 1$  and  $s > \theta_i$  or when  $\theta_i > 1$  and  $s < \theta_i - 1$  and zero otherwise.



# Search for the optimal policy within the class



Path taken by the algorithm



Objective function

# 3. Landmine Detection Application

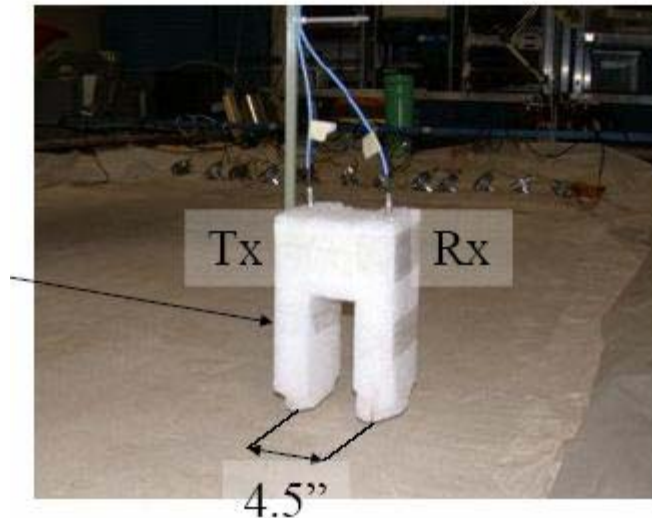
- Multimodality sensor suite can be implemented sequentially
- Data set used is the GATech “Three Sensor Dataset” (Feb.2004)
  - Includes metal detector, radar, and seismic vibrometer.
  - Collection performed on three scenarios of mine/clutter arrangements.



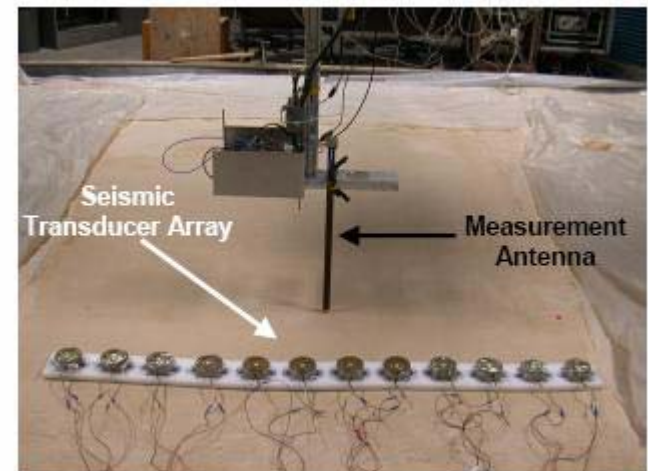
Sensors from the Three Sensor Dataset



EMI



GPR



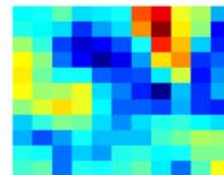
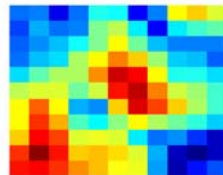
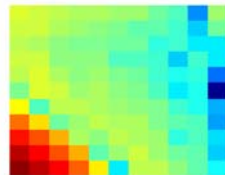
Seismic

# Landmine Signatures

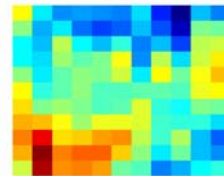
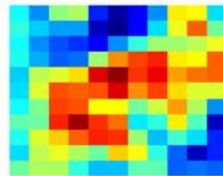
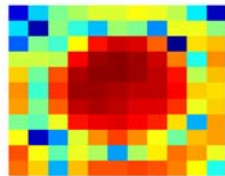
EMI

GPR

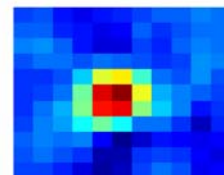
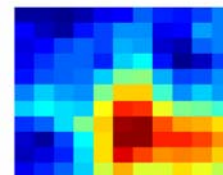
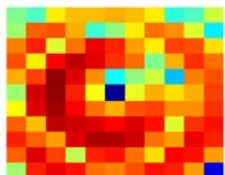
Seismic



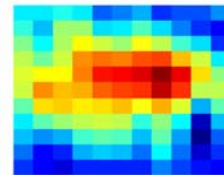
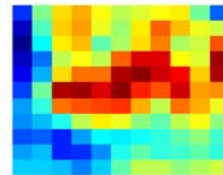
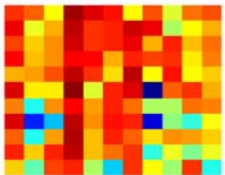
Rock



Nail



Plastic Anti-personnel  
Mine



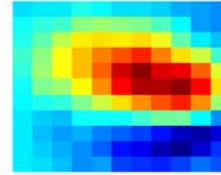
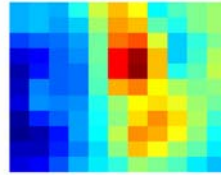
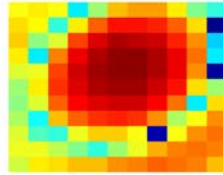
Plastic Anti-tank Mine

# Clutter Signatures

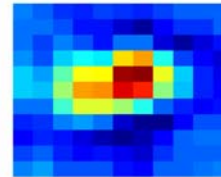
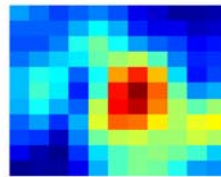
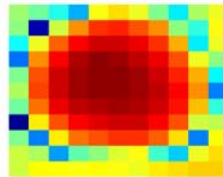
EMI

GPR

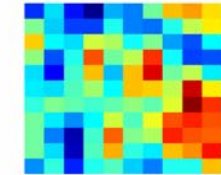
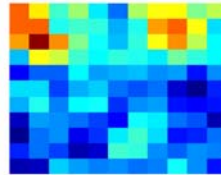
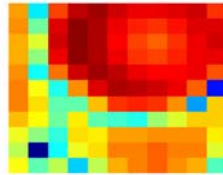
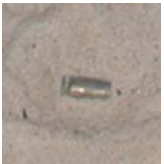
Seismic



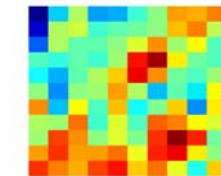
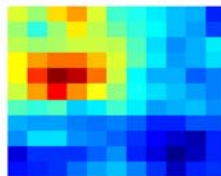
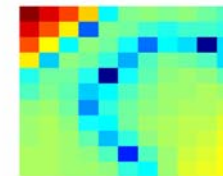
Popcan (Crushed)



Popcan (Uncrushed)

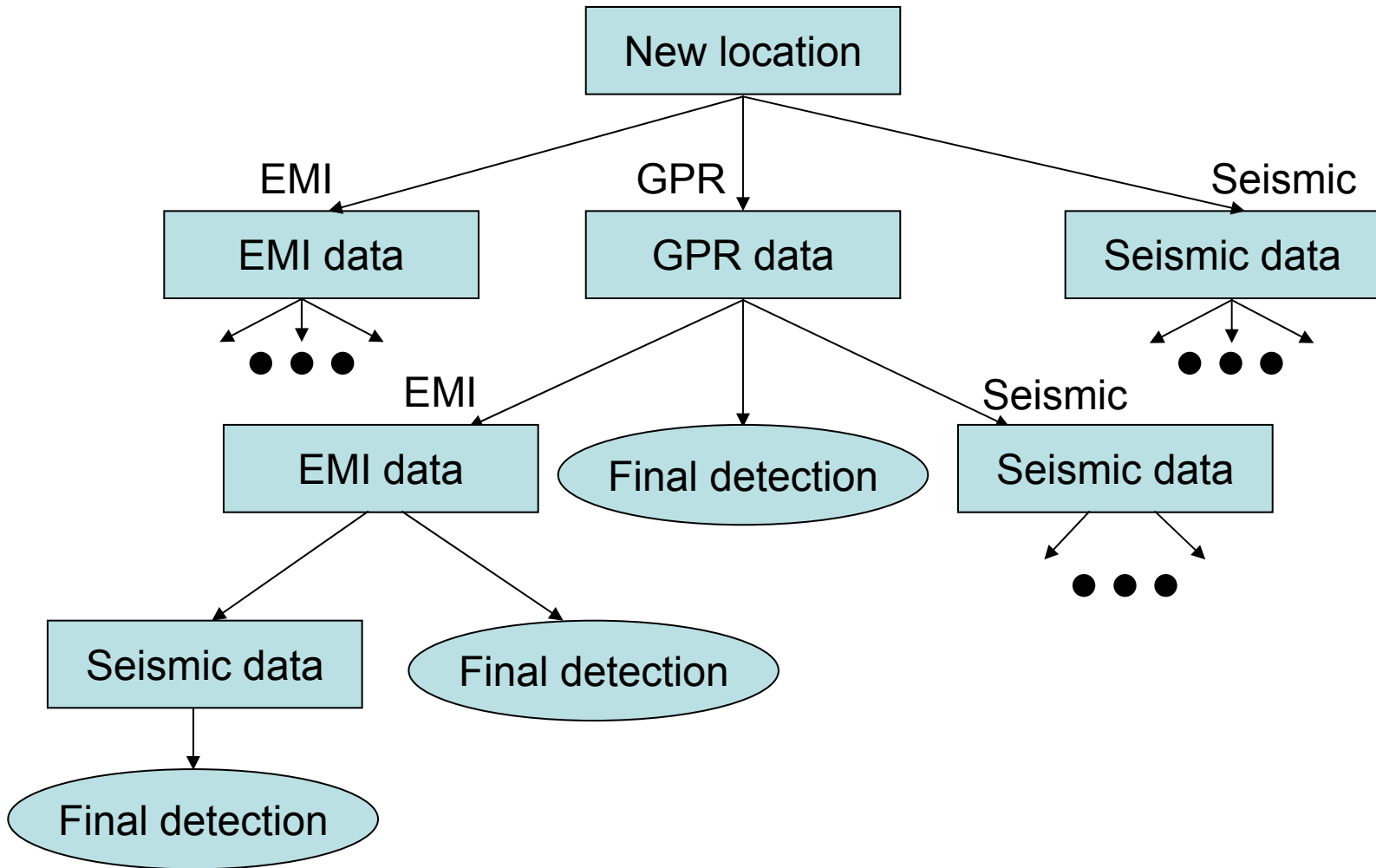


Shell



Penny

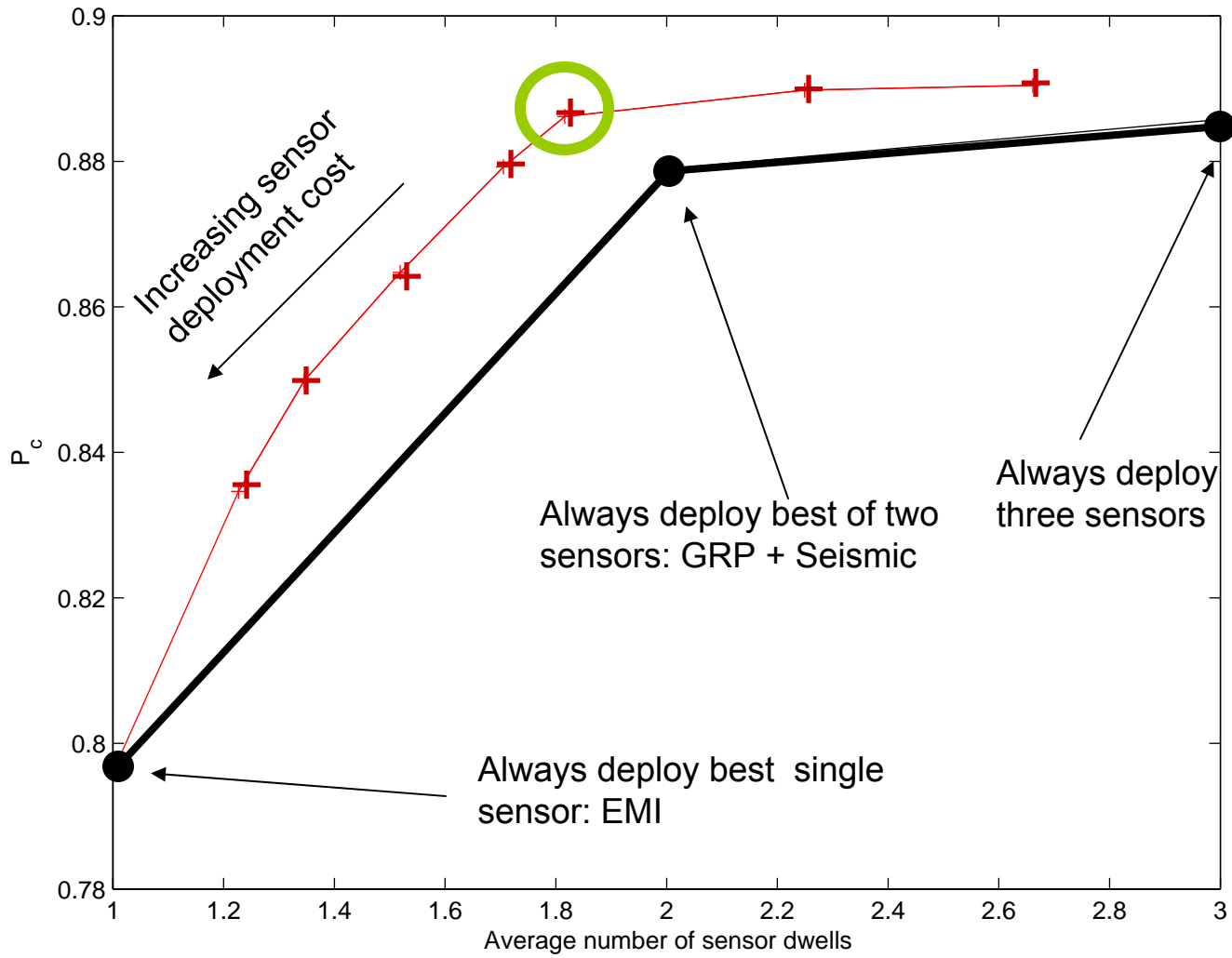
# The Sensor Scheduling and Decision tree



# Simulation Details

- State at time  $t$ : all the available information.
- Expected reward:  
  
(Prob. of correct decision) – (cost of sensor deployment) (number of dwells)
- Scheduler:
  - Weighted classification building block:
    - Weights sensitive combination of [7,2] and [7,3] [tansig, logsig] NN.
- Mine Detector:
  - Unweighted classification building block:
    - [7,2] [tansig, logsig] feed forward NN.
- Training used 1000 trajectories
  - Equiprobable mine/clutter scenarios
  - Adaptive-length gradient learning with momentum term
  - Reseeding applied to avoid local minima
- Performance evaluation using 10,000 trajectories.

# Performance Comparison



- Performance obtained by randomized sensor allocation
- + Performance obtained by optimal sensor scheduling

Optimal sensor scheduling improves detection performance while reducing average dwell time.

# Optimal Policy for Mean States

Policy for specific scenarios:

		Object Type								Feature Description
		1	2	3	4	5	6	7	8	
		M-AT	M-AP	P-AT	P-AP	Ctr-1	Ctr-2	Ctr-3	Bkg	
Sensor	EMI (1)	High	High	Medium	High	High	Low	Low	Low	Conductivity
		High	High	High	Medium	Medium	Low	Low	Low	Size
	GPR (2)	High	Medium	High	Medium	Low	Low	Low	Low	Depth
		High	Medium	High	Medium	High	High	High	Low	RCS
	Seismic (3)	High	Medium	High	Medium	Medium	Medium	Low	Low	Resonance

Optimal	2	2	2	2	2	2	2	2
sequence	3	1	3	1	3	3	3	3
for mean	D	D	D	3	D	D	D	D
state				D				

# 4. Conclusions

- Classification reduction+GaussSiedel - new paradigm for optimal policy approximation
- Relation to classification allows RL to be formulated as learning optimal classifiers
- Can be expected to improve over Q-learning since it approximates best policy not best reward
- Can be implemented in continuous state spaces via standard methods of function approximation
- Has been applied to multi-stage mine detection with demonstrated improvements over fixed non-adaptive sensor scheduling strategy.