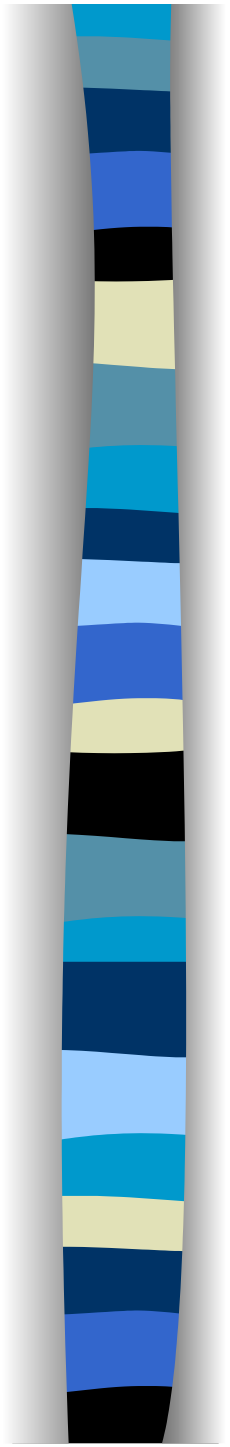
Generating decision rules by reinforcement learning for a class of crop management problems

F. Garcia, R. Martin-Clouaire
Biométrie et Intelligence Artificielle
INRA-Toulouse
{fgarcia, rmc}@toulouse.inra.fr

G. Nguyen
Information Tecnology
Hanoi Univ. of Technology
giangnl@it-hut.edu.vn



PLAN

Crop management problem

Basic Q-learning vs. learning fuzzy rules

Learning fuzzy rules by finite horizon Q-learning

Conclusion

Crop management problem

decision making about technical operations in the crop production process (e.g. seeding, fertilization, irrigation, pest control, harvest)

problem of sequential decision under uncertainty in finite horizon

- a few stages, continuous or discrete domains of state and decision
- given objective, find robust management strategy (forward looking decision commitment)
- apply management strategy to current state

Winter wheat

	Sowing	*1st N fertilization*	*2nd N fertilization*
state	sowing time	tillering stage nb of plants	residual N start of stem elong.
decision	seed rate wheat cultivar	date quantity	date quantity

Strategy as a set of fuzzy rules

Management strategy :

- classically a function from state to decision for every stage

- for every stage, a set of fuzzy rules mapping set of states to decision

If *dT is around Dec 1st and Np is around 180* ***then***
dN1 should be Dec15 and qN1 should be 15

rules better for intelligibility and implementation (though some loss)

Reinforcement learning

Basic Q-learning =

- discrete state space

s = current state, d = taken decision
s ’= resulting state,
r = immediate reward after d

- given (s, d, s’, r) using simulator, learn the value Q(s,d) of (s,d) by

$$Q(s, d) \leftarrow Q(s,d) + \varepsilon \cdot (r + \max_{d'} Q(s',d') - Q(s,d))$$

learning rate

Temporal difference error

- optimal policy: for any state s, optimal decision = $\arg_d \max Q(s,d)$

How to deal with continuous state space:

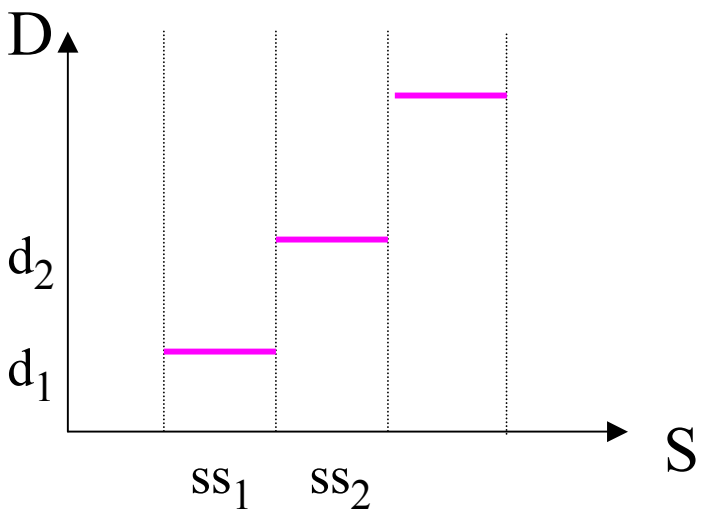
- state aggregation
- CMAC
- neural nets

Q-learning with state aggregation

- homogenous discretization of the state space in small regions ss in which states are undistinguished,

- learn the value Q(ss,d) of (ss,d)

- optimal policy: for any state s in ss, optimal decision = $\arg_d \max Q(ss,d)$

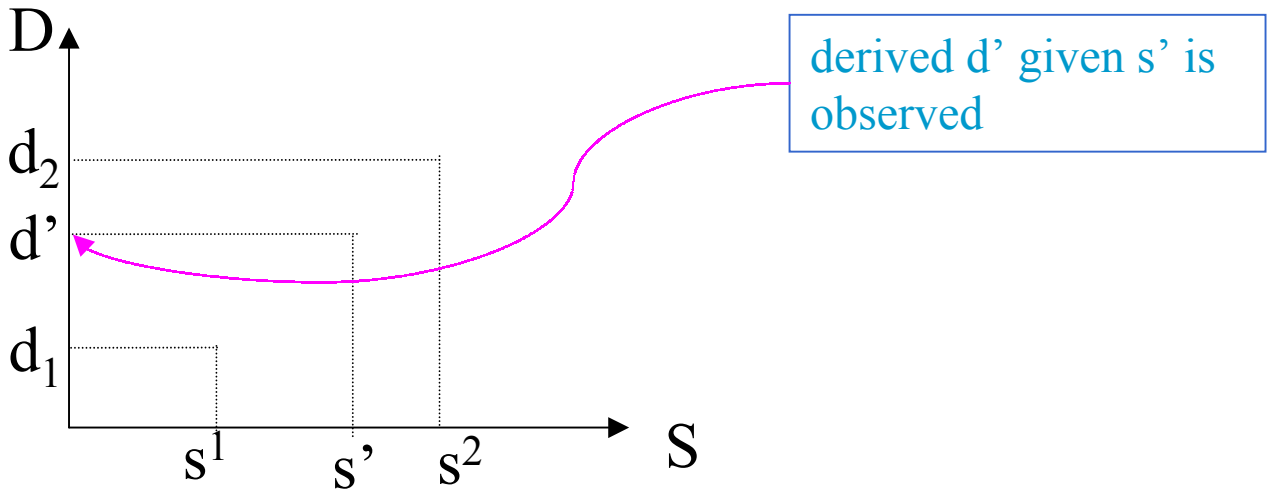


Learning fuzzy rules

Basic idea at work in the use of fuzzy rules (Jouffe, 1998)

- learn the value of well-chosen points (s,d)

  for any state s' between s_1 and s_2, compute the decision d' by interpolation between d_1 and d_2



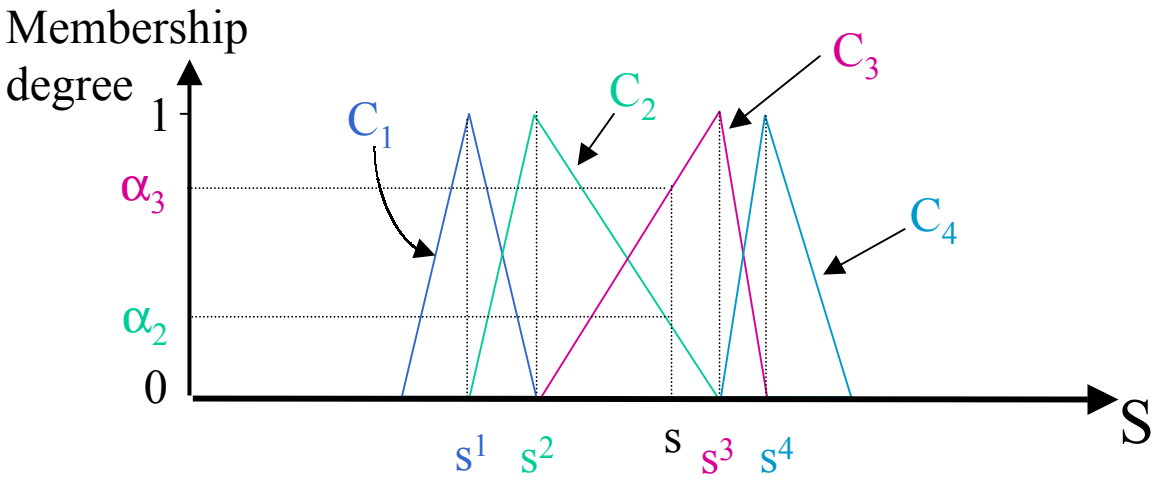
- learn the location of the s^i's

Semantics of fuzzy rules

General form of a fuzzy rule

$Rule_r$: If s is in C_a then decision should be d_a

a fuzzy set meaning «around s^a»



If s is observed, optimal decision = $(\alpha_2 . d_2 + \alpha_3 . d_3) / (\alpha_2 + \alpha_3)$

Finite horizon Q-learning Ndiaye (1998)

Use of iterative mechanism of Q-learning extended to finite horizon case

- a value function Q_i for each stage i

- updating factor (temporal difference error) ΔQ_i given (s_i, d_i, s_{i+1}, r_i)

$$\Delta Q_i = r_i + \max_{d'} Q_{i+1}(s_{i+1},d') - Q_i(s_i,d_i)$$

Immediate reward in stage i

Sowing	*N1*	*N2*	*Harvest*
sowing time	tillering stage nb of plants	residual N start of stem elong.	N in soil yield
seed rate wheat cultivar	date quantity	date quantity	
r_{sowing}	r_{N1}	r_{N2}	$r_{harvest}$

Learning the value of decision associated to state s^a

Assume:
- there are m fuzzy sets C_a covering domain of s in current stage
- domains of decision variables are discretized in p values

Learn a degree of pertinence q_a^j of each association (C_a, d_a^j)
where C_a means «around s^a», a = 1,m and j = 1,p

Q-learning used to compute $\Delta Q_i = r_i + \max_{d'} Q_{i+1}(s_{i+1},d') - Q_i(s_i,d_i)$
that is used in the updating of the q_a^j 's
- what is d_i ?
- how are $Q_i(s_i,d_i)$ and $\max_{d'} Q_{i+1}(s_{i+1},d')$ computed ?
- how are q_a^j 's updated ?

Learning the value of decision associated to state s^a

what is d_i ?

choosing d_i = picking one decision value (among p) for each rule set

C_1: $\{d^1,...,d^j,....,d^p\}$ $\quad d_1°$

.

C_a: $\{d^1,d^2,....,d^p\}$ $\quad d_a°$

.

C_m: $\{d^1,d^2,....,d^p\}$ $\quad d_m°$

$d_i = \Sigma_{k=1,m}\ \alpha_k(s_i)\ .\ d_k°$

interpolation between chosen $d_k°$ weighted by compatibility (relevance) degrees with s_i

compatibility of s_i with C_k

decision applied in stage i

Learning the value of decision associated to state s^a

how are $Q_i(s_i,d_i)$ and $\max_{d'} Q_{i+1}(s_{i+1},d')$ computed ?

$\Sigma_{k=1,m}\ \alpha_k(s_i)\ .\ q_k^\circ$

interpolation between q_k°
weighted by compatibility degrees,
q_k° = current value of d_k°

$\Sigma_{k=1,m'}\ \alpha_k(s_{i+1})\ .\ \max_{k=1,m'}\ q_k^{i+1}$

choosing best decision known in resulting stage s_{i+1}, weighted by compatibility of s_{i+1} with rule conditions in stage i+1

Updating of q_a° in stage i

$\Delta q_a^\circ = \Delta Q_i\ .\ \alpha_a(s_i)$

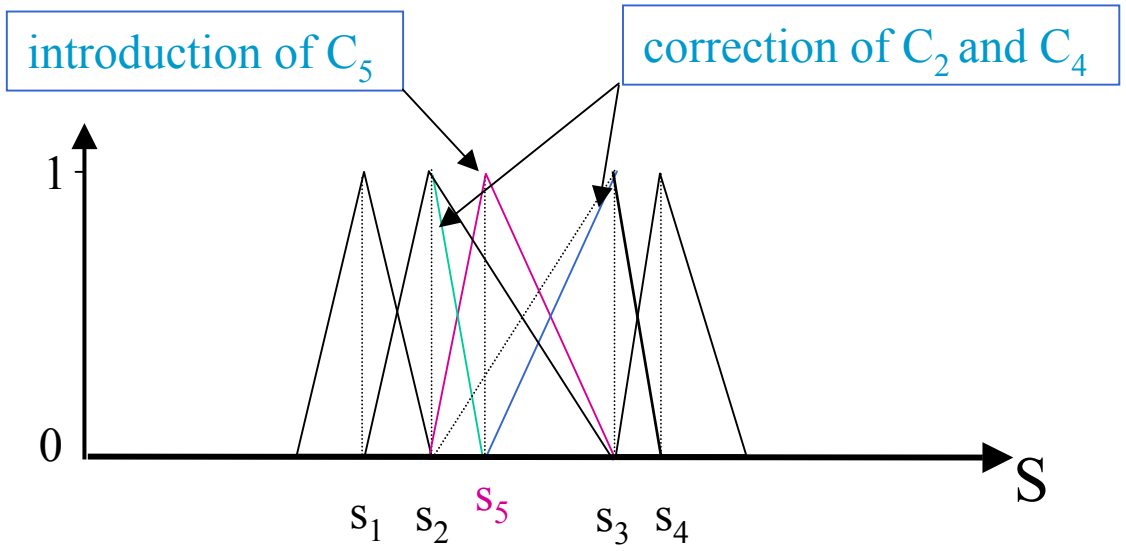
Ultimately, keep only most pertinent (C_a, d_a^j) for each C_a

Learning the location of pertinent points in S

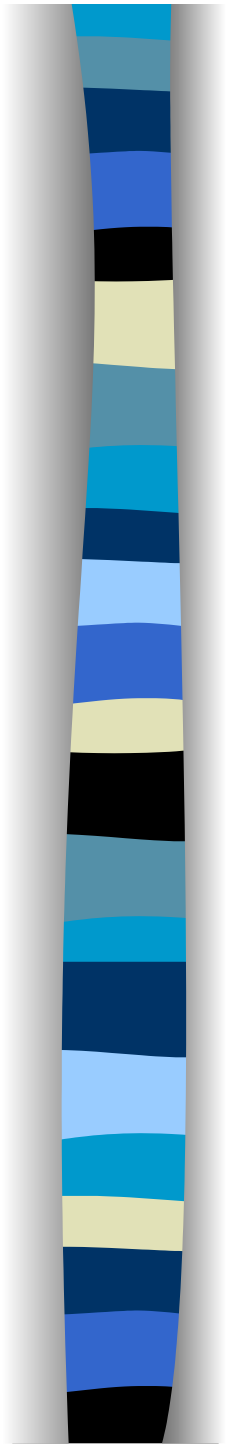
Refinement of the fuzzy partition on a variable domain

- identify the region between two s_i's in which the recent ΔQ values are most scattered

- introduce new Cj and modify neighbors



- initialize or modify degrees of pertinence



Conclusion

So far, only preliminary results obtained on winter wheat problem

Need to be compared with another method for generating non-fuzzy rules (CMAC Q-function + binary tree clustering)

Limitations

- interpolation has to make sense
- strategies expressed as a flat set of rules