

Evolution Artificielle

pour l'apprentissage supervisé

propositionnel

Michèle Sebag, LRI, Orsay

Plan

- Approche Pittsburg
- Approche Michigan
- Systèmes de classeurs

Apprentissage supervisé

Input

- Exemples étiquetés $\mathcal{E} = \{(x_i, y_i)\}$

$x_i \in X$: espace des instances.

$y_i \in Y = \{0, 1\}$: problème de classification

$y_i \in Y = \mathbb{R}$: problème de régression

Apprentissage supervisé, 2

Output

- Un classifieur : fonction $h : X \mapsto Y$

h : classifieur symbolique, base de règles, disjonction de conjonctions

h : classifieur numérique, fonction linéaire, réseau de neurones, programme...

Du point de vue de l'optimisation

Espace de recherche

- Espace des bases de règles (Pittsburg)
- Espace des règles (Michigan)
- Espace des stratégies : état $\rightarrow$ action optimale
(Système de Classeurs)

Du point de vue de l'optimisation

Fonction performance

- Erreur empirique
- Espérance du coût des erreurs
- Pénaliser les fonctions complexes
- Minimum Description Length
- Multi-objectif : couvrir chaque exemple est un objectif
- Co-évolution : entre les bonnes règles et les bons exemples

Représentation Attribut-Valeur

Exemple

Attribute	Values
Head_Shape	Round, Square, Octagon
Body_Shape	Round, Square, Octagon
Is_Smiling	Yes, No
Holding	Sword, Balloon, Flag
Jacket_Color	Red, Yellow, Green, Blue
Has_Tie	Yes, No

Remarque : premiers ages de la littérature, pas de numériques.
discrétisation préalable

Approche Pittsburg

Un individu génétique == une base de règles

Représentation
sémantique

the head is round and the jacket is red,
or the head is square and it is holding a balloon

logique

$$(< S = R > \wedge < J = R >) \\ \vee (< S = S > \wedge < H = B >)$$

génétique

$$R * * * R * \quad S * * B * *$$

Approche Pittsburg, représentation, 2

Représentation, suite
génétique

$R * * * R * S * * B * *$

informatique

100|111|11|111|1000|11 010|111|11|010|1111|11

Test de couverture prémisses - exemple

- Les 0 et les 1 doivent coïncider
- Valeur *don't care* pour un attribut : 11111...11

Croisement Pittsburgh

Echange de règles

Parent 1 < 100|111|11|111|1000|11 $\vee$ 010|111|11|010|1111|11 >

Parent 2 < 111|001|01|111|1111|01 $\vee$ 110|100|10|111|0010|01 >

→

Enfant 1 < 100|111|11|111|1000|11 $\vee$ 111|001|01|111|1111|01 >

Enfant 2 < 010|111|11|010|1111|11 $\vee$ 110|100|10|111|0010|01 >

Croisement Pittsburg, 2

Mouvement de règles

Parent 1 < 100|111|11|111|1000|11 $\vee$ 010|111|11|010|1111|11 >

Parent 2 < 111|001|01|111|1111|01 $\vee$ 110|100|10|111|0010|01 >

→

Enfant 1 < 100|111|11|111|1000|11 $\vee$ 111|001|01|111|1111|01
 $\vee$ 110|100|10|111|0010|01 >

Enfant 2 < 010|111|11|010|1111|11 >

Mutation Pittsburgh, 2

Abandon de règles

Parent $< 100|111|11|111|1000|11 \vee 010|111|11|010|1111|11$
 $\vee 100|010|10|010|0010|01 >$

→

Enfant $< 100|111|11|111|1000|11 >$

Ajout d'un cas

Parent $< 100|111|11|111|1000|11 \vee 010|111|11|010|1111|11 >$

Ex non couvert $(100|010|10|010|0010|01)$

→

Enfant $< 100|111|11|111|1000|11 \vee 010|111|11|010|1111|11$
 $\vee 100|010|10|010|0010|01 >$

Mutation Pittsburg, 3

Généralisation interne

Parent $\langle 100|111|11|111|1000|11 \vee 010|111|11|010|1111|11$
 $\vee 100|010|10|010|0010|01 \rangle$

→

Enfant $\langle 100|111|11|111|1000|11 \vee 110|111|11|010|1111|11 \rangle$

Spécialisation interne

Parent $\langle 100|111|11|111|1000|11 \vee 010|111|11|010|1111|10$
 $\vee 111|010|10|010|1111|11 \rangle$

→

Enfant $\langle 100|111|11|111|1000|11 \vee 010|010|10|010|1111|10 \rangle$

Approche Michigan

Un individu génétique == une règle

Une règle :

$$< 110|0001|100 > \rightarrow 3$$

(it can walk, it can jump but it cannot fly AND it barks AND it is 90 cm long
→ it is a dog)

Un exemple d'apprentissage = un message

Un message → { règles déclenchables }

les prémisses s'apparient

Approche Michigan, 2

Evaluation

- Déclenchement : mécanisme d'enchères (bid).
- La règle avec enchère max gagne
- Si la conclusion de cette règle est exacte, récompense positive.
- Sinon récompense négative

Approche Michigan, Croisement

Opérateurs classiques

- Un-point, deux points, ...

$$11001|011 + 11011|111 = 11001|111$$

- Uniforme

$$\mathbf{11001011} + 110\mathbf{11101} = 11011111$$

Approche Michigan, Croisement, 2

Opérateurs dédiés

- Généralisation (ou logique)

$$Offspring = Parent1 \text{ OR } Parent2$$

- Spécialisation (et logique)

$$Offspring = Parent1 \text{ AND } Parent2$$

Approche Michigan, Mutation

Opérateurs classiques

- Avec probabilité p (très faible), chaque bit est inversé
- Le nombre de bits à inverser est déterminé ; ces bits sont tirés uniformément et inversés

Systèmes de classeurs

Un individu = une règle

comme Michigan

Différences

- La conclusion (envoi de message)

if condition then action

- Une évaluation parallèle

plus rapide, plus performant

Applications : robotique, planification.

Systèmes de classeurs, exemple

Nom	Classifier
A	0 1 * * : 0 0 0 0
B	0 0 * 0 : 1 1 0 0
C	1 1 * * : 1 0 0 0
D	* * 0 0 : 0 0 0 1

1. L'environnement envoie un message : 0111
2. Le classeur *A* est déclenché : envoie le message 0000
3. Les classeurs *B* et *D* sont déclenchés, envoient les messages 1100 et 0001
4. Le message 1000 déclenche *D*. FIN.

Systèmes de classeurs, suite

Déclenchement : Un mécanisme d'enchères

- Les classeurs en compétition sont ceux qui sont déclenchés par les messages courants
- La plus grande enchère gagne (voir suite)
- Il peut y avoir plusieurs gagnants
- Chaque gagnant paie une récompense
à celui qui a envoyé le message
- L'environnement récompense le dernier classeur actif.

Systèmes de classeurs, Evaluation

Première époque

- fitness == accumulation des récompenses
== espérance de la récompense
- enchère $\propto$ fitness

Seconde époque

- - valeur = espérance de la récompense
 - fiabilité = variance de la valeur
 - précision des prémisses
- enchère = $\mathcal{F}$ (valeur, fiabilité, précision)

Systèmes de classeurs

Apprentissage par renforcement

Ressemblances

- SC permet aussi de déterminer une politique

chaque etat $\rightarrow$ action optimale

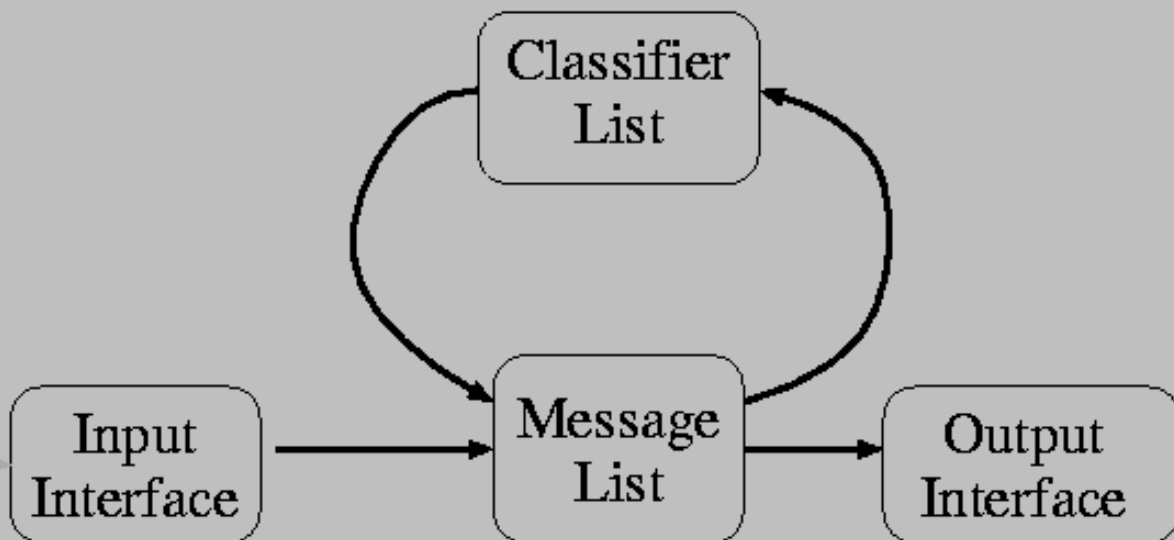
- Passe par l'apprentissage d'une fonction de valeur

Etat $\times$ Action $\rightarrow$ (valeur, fiabilite)

Différences

- Formalisme
- Exploration
- Evaluation

Basic CFS



Environment

Composants d'un CS

- Liste de messages
- Liste de classeurs
- Interface d'entrée
- Interface de sortie

Liste de messages

Rôle

- Base de données, base de faits
- Tableau noir

Particularités

- Un nombre fini de messages
- Un format fixe
- Une sémantique, claire s'ils viennent de l'interface d'entrée
obscure s'ils viennent des classeurs

Liste de classeurs

Formalisme

IF cond₁ AND cond₂... AND cond_N THEN action

- nb de conditions fixes
- opérateurs logiques : AND et NOT

Implémentation

(¬)cond₁, (¬)cond₂, ... (¬)cond_N action

(souvent N=1 ou 2)

Condition $\equiv$ Message

Représentation

- Chaines de longueur fixe
- Alphabet $\{0, 1, \#\}$: $\#$ = don't care

Déclenchement

Message List		Condition	Satisfied	Matched by
Label	Message			
		0101	Yes	a
		1101	No	
a	0101	#101	Yes	a
b	1010	1###	Yes	b, c
c	1111	##00	No	
		####	Yes	a, b, c

Déclenchement, suite

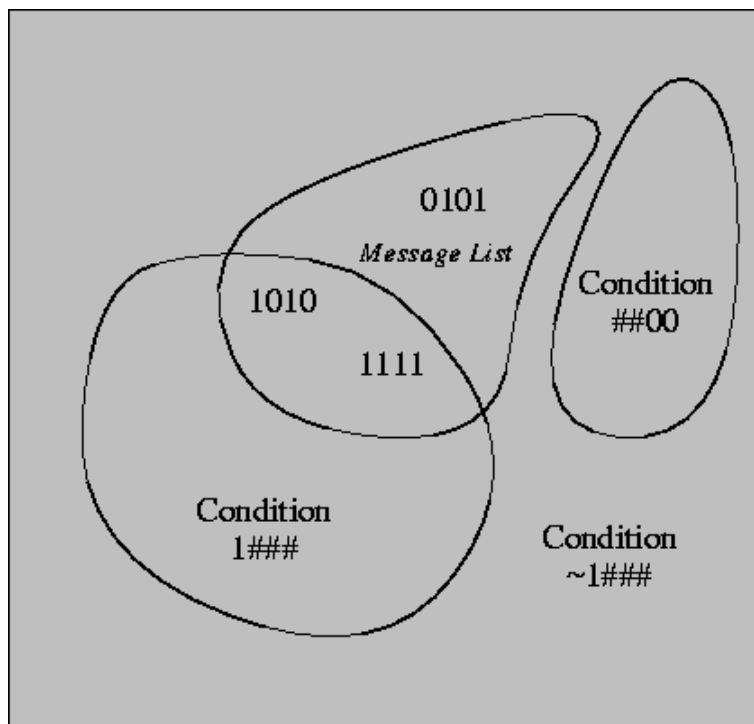
Conditions négatives

Message List		Condition	Satisfied	Matched by
Label	Message			
a	0101	$\neg 0101$	No	a
b	1010	$\neg 1101$	Yes	
c	1111	$\neg \#101$	No	a
		$\neg 1###$	No	b, c
		$\neg \##00$	Yes	
		$\neg ####$	No	a, b, c

Spécificité

- Condition positive : $\propto \frac{1}{nb\ de\ \#}$
- Condition négative : $\propto nb\ de\ \#$

Conditions, vision ensembliste



Actions

Représentation

- Chaines
- même taille, même alphabet que les messages

Rôle : Faits paramétrés

- Les 0 et 1 sont conservés
- Les # sont remplacés par les caractères correspondant de la condition (la première s'il y en a deux)

: pass-through operator

Exemple

Message List

Label	Message
a	0101
b	1010
c	1111

Classifier List

Label	Classifier
i	#11#, #110 / 00##
ii	###1, #110 / ###0
iii	##1#, 1110 / 0##0

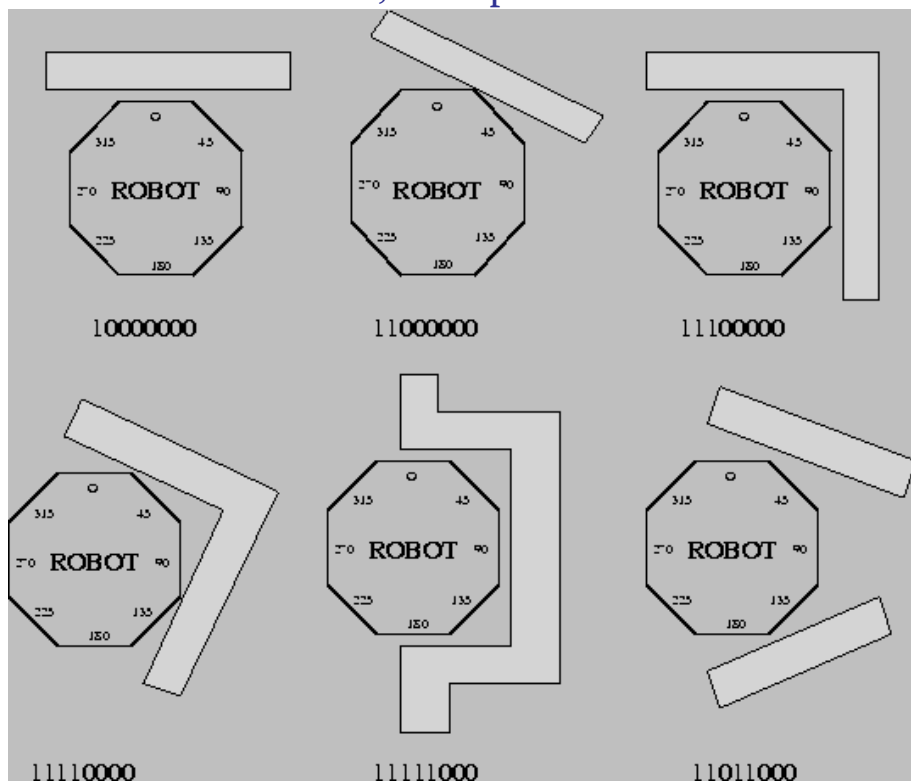
	Message	Origine
	1111	Posted by i, c matches cond-1
→	0100	Posted by ii, a matches cond-1
	1110	Posted by ii, c matches cond-1
	0010	Posted by iii, b matches cond-1
	0110	Posted by iii, c matches cond-1

Actions, suite

- Pas de destruction des messages : écrasement (taille limitée)
- Activation en parallèle
- Un classeur peut poster autant de messages qu'il est déclenché
- Résolution de conflit nécessaire si plus de messages que la taille autorisée

Interface d'entrée

Description de l'environnement, exemple



Interface de sortie

Sélection des messages pertinents : meta-bits (tags)

Tag	Interpretation
00	Output Message
01	Internal Message
10	Input Message
11	Internal Message

Exemple, contrôle des actionneurs du robot

```
1##### / 00 0100 00
##1##### / 00 0001 00
####1### / 00 1000 00
#####1# / 00 0010 00
```

bits 1-2 : tags

bits 3-6 : contrôle

bits 7-8 : remplissage

Système de classeurs, boucle

Indéfiniment

- Activate the input interface and post the input messages it generates to the message list
- Perform the matching of all the conditions of all classifiers against the message list.
- Activate the fireable classifiers (those whose conditions are satisfied) and add the messages they generate to the message list.
- Activate the output interface, i.e. remove the output messages from the message list and perform the actions they describe.
- Repeat the previous steps.

Système de classeurs - Apprentissage

Adaptation des crédits

- On garde les mêmes règles
- On les déclenche à meilleur escient

Découverte de règles

- A la main ? ne passe pas à l'échelle
- Par évolution artificielle

Fitness

- Force : utilité appris en cours d'exécution
- Spécificité : $L - nb \#$ constant
- Enchère = produit des critères (multi-critères...)

Remarques

- Plus d'un gagnant $\Leftrightarrow$ parallélisme
- Enchères stochastiques pour assurer l'exploration
- Seule la force peut être contrôlée

Allocation de crédit

Bucket brigade

$\text{Bid} = 0.1 \times \text{strength} \times \text{specificité}$

Time=t-2

100#1# / XX0001 / S=100 B=10 Active

XX#0#1 / YY0000 / S=120 B=0

YY#00# / 000000 / S=160 B=0

Time=t-1

100#1# / XX0001 / S=90 B=0

XX#0#1 / YY0000 / S=120 B=12 Active

YY#00# / 000000 / S=160 B=0

Time=t

100#1# / XX0001 / S=102 B=0

XX#0#1 / YY0000 / S=108 B=0

YY#00# / 000000 / S=160 B=16 R=20 Active

Time=t+1

100#1# / XX0001 / S=102 B=0

XX#0#1 / YY0000 / S=124 B=0

YY#00# / 000000 / S=164 B=0

Découverte de Règles

Learning Classifiers Systems : LCS

1. Read input messages from sensors
2. Find the classifiers that can fire and select those with a highest importance (fitness)
3. Reduce the fitness of the fireable classifiers
4. Clear the message list
5. Fire the fireable classifiers
6. Evaluate the resulting behaviour
7. Punish or reward the classifiers that fired (reinforcement learning)
8. Generate a new population of classifiers with EC
9. Iterate

Système de classeurs : Conclusions

Remarques

- Un style de programmation
- Arbitrairement complexe (si au moins deux conditions)
- Difficile à programmer : $\approx$ assembleur
- Avec des hiérarchies de comportement par défaut
programmation économique
- $\Rightarrow$ Apprentissage faisable par EC