

Projet Foot 2I013

Nicolas Baskiotis

`nicolas.baskiotis@lip6.fr`

`http://webia.lip6.fr/~baskiotisn`

`http://github.com/baskiotisn/SoccerSimulator`

Université Pierre et Marie Curie (UPMC)
Laboratoire d'Informatique de Paris 6 (LIP6)

S2 (2016-2017)

Plan

Résultats de la semaine

Machine Learning

Optimisation et contrôle

Classification supervisée : Arbres de décision




Tournoi 1v1

Equipe (login)	Points	(G,N,P)	(GA,GP)
ChingChong (JimTitor)	80	(25,5,2)	(195,40)
Liban (hassanharb)	70	(23,1,8)	(180,69)
UPMC (mouloudETjeffrey)	67	(22,1,9)	(139,79)
EDF (SoccerJess)	66	(21,3,8)	(242,59)
TeamFUT (SinghKevin)	65	(21,2,9)	(203,75)
Toho (MoroMane)	62	(20,2,10)	(155,62)
team1 (fabien25)	60	(20,0,12)	(245,78)
MFC (jeandeb)	58	(17,7,8)	(70,46)
Ariouati (Ariouati)	48	(12,12,8)	(58,61)
DZ (manelfil)	44	(12,8,12)	(93,66)
team (yxd117)	38	(12,2,18)	(112,88)
FUT team (KimmengLy)	28	(8,4,20)	(43,187)
Eq1 (exrivalis)	27	(7,6,19)	(34,171)
EGY (EgyAlg)	22	(2,16,14)	(4,49)
CityHunter (YasmineAitMimoun)	18	(4,6,22)	(31,221)
leoniro1 (leoniethiriat)	14	(3,5,24)	(20,246)
serpentard (GeekyCeline)	6	(0,6,26)	(2,229)

Tournoi 2v2

Equipe (login)	Points	(G,N,P)	(GA,GP)
ChingChong (JimTitor)	80	(25,5,2)	(88,6)
MFC (jeandeb)	75	(24,3,5)	(111,26)
team2 (fabien25)	74	(24,2,6)	(217,29)
team2 (MoroMane)	72	(23,3,6)	(110,26)
UPMC (mouloudETjeffrey)	63	(19,6,7)	(91,23)
Ariouati (Ariouati)	62	(18,8,6)	(79,25)
France (hassanharb)	57	(16,9,7)	(83,29)
TeamFUT (SinghKevin)	50	(15,5,12)	(111,55)
team (yxd117)	45	(14,3,15)	(143,88)
FUT team (KimmengLy)	39	(10,9,13)	(39,95)
CityHunter (YasmineAitMimoun)	37	(10,7,15)	(42,77)
EDF (SoccerJess)	31	(10,1,21)	(51,148)
DZZ (manelfil)	30	(8,6,18)	(46,88)
Eq2 (exrivalis)	22	(7,1,24)	(29,175)
leoniro1 (leoniethiriat)	17	(3,8,21)	(5,117)
EGY (EgyAlg)	10	(0,10,22)	(0,72)
serpentard (GeekyCeline)	7	(1,4,27)	(3,169)

Résumé hebdomadaire

Semaine # joueurs	4		5		6		7		8		9	
	1	2	1	2	1	2	1	2	1	2	1	2
Ariouati	3	1		9	6							
EgyAlg				14	16							
exrivalis	5	6		13	14							
fabien25	8	5		7	3							
GeekyCeline	10	7		17	17							
hassanharb	4	8		2	7							
jeandeb	2	4		8	2							
JimTitor				1	1							
KimmengLy	6	2		12	10							
leoniethiriat				16	15							
manelfil				10	13							
MoroMane	7	3		6	4							
mouloudETjeffrey				3	5							
SinghKevin	9	9		5	8							
SoccerJess				4	12							
YasmineAitMimoun				15	11							
yxd117	1	10		11	9							

Plan

Résultats de la semaine

Machine Learning

Optimisation et contrôle

Classification supervisée : Arbres de décision

Les grandes familles

Apprentissage supervisé

- Classification
- Regression
- Forecasting
- Complétion de données
- Ranking
- Recommendation

Apprentissage non supervisé

- Clustering
- Apprentissage de représentation, de dictionnaire
- Analyse de séquences
- Représentation hiérarchique
- Détection d'anomalies

Apprentissage par renforcement

- Apprendre à jouer
- Apprendre à interagir avec l'environnement

Apprentissage supervisé

Données du problème

- Une représentation X des objets de l'étude
- Une sortie d'intérêt y qui peut être numérique, catégorielle, structurée, complexe (label, réponse, étiquette, ...)
- Un ensemble d'exemples, d'échantillons, sous leur représentation X et avec leur sortie connue $\{(x_1, y_1), \dots, (x_n, y_n)\}$

Objectifs

- Prédire de manière précise la sortie y pour un nouvel exemple x non vu
- Comprendre quels facteurs influencent la sortie
- Évaluer la qualité de nos prédictions

Apprentissage non supervisé

Données du problème

- Une représentation X des objets de l'étude
- Un ensemble d'exemples, d'échantillons, sous leur représentation X , $\{x_1, \dots, x_n\}$
- Pas de variable de sortie !

Objectifs

- Trouver des groupes d'objets “semblables”
 - Organiser les données d'une manière “logique”
 - Trouver les “similarités” des objets
 - Trouver des “représentations” des objets
- ⇒ on ne sait pas bien ce que l'on cherche
- ⇒ tout un art !

Apprentissage par renforcement

Apprentissage continu en fonction du retour d'expérience

Données du problème

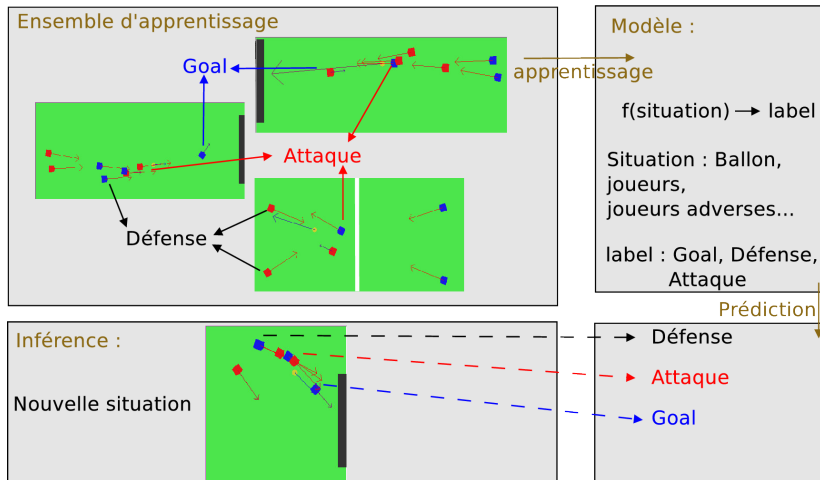
- Un état décrit l'environnement courant
- Un ensemble d'actions sont possibles
- Une politique permet de choisir en fonction de l'état l'action à effectuer
- A l'issue de chaque action, une récompense est observée

Objectifs

- S'améliorer ! (améliorer la politique de choix de l'action)
- Éviter les situations d'échecs
- Comprendre la dynamique du problème

Et l'utilité dans le cadre du projet ?

Apprentissage supervisé



Apprentissage par renforcement et algorithmes génétiques plus tard ...

Plan

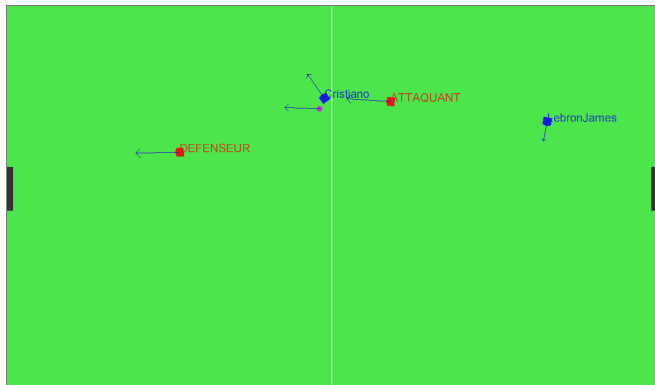
Résultats de la semaine

Machine Learning

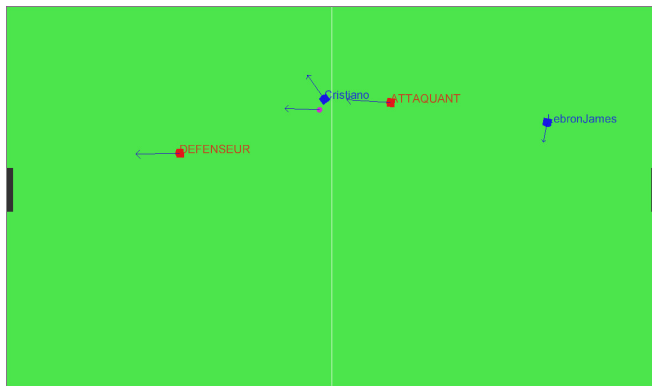
Optimisation et contrôle

Classification supervisée : Arbres de décision

Comment bien courir (ou tirer, dribbler,...) ?



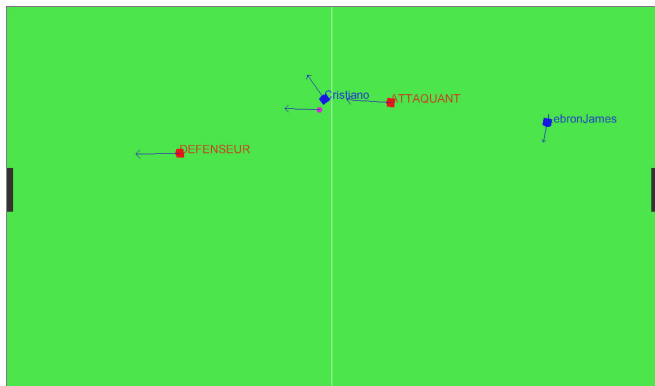
Comment bien courir (ou tirer, dribbler,...) ?



Que cherche-t-on ?

- A réagir \Rightarrow contrôler
- en fonction de l'environnement : distance à la balle, vitesse, ...
- dans un but précis !

Comment bien courir (ou tirer, dribbler,...) ?



Pour l'instant, le contrôle c'est surtout

- des conditions
- des paramètres x_1, x_2, \dots, x_d

⇒ l'efficacité est fonction de ces paramètres !

Comment optimiser cette efficacité ?

Recherche exhaustive

- Parcourir toutes les plages de paramètres possibles
- Evaluer l'efficacité de chacun
- Sauvegarder les meilleurs et les utiliser

Nécessite :

- Un objectif simple bien défini et facile à évaluer !
- L'utilisation intensive de simulations pour estimer l'efficacité
- beaucoup de temps ... (ou une recherche plus intelligente)

Alternative :

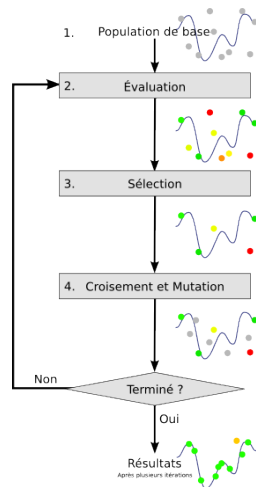
- Paramétrer le comportement par une fonction (avec peu de paramètres)
- Trouver le meilleur paramètre pour cette fonction

⇒ Exemple : $\text{vitesse}(d) = 1 - e^{-d}$

Algorithme génétique

Objectif

- Optimiser une fonction $f(x_1, x_2, \dots, x_d)$
- ⇒ trouver son minimum
- Mais trop grand nombre de paramètres
- ⇒ impossible de tout explorer
- Inspiration biologique darwinienne :
- ⇒ évolution d'une population (ensemble) de solutions par mutation et croisements aléatoires



Algorithme génétique

Algorithme

- Choisir un codage des paramètres : point crucial !
- Générer au hasard une population initiale
- Itérer :
 - Evaluer chaque individu (chaque solution potentielle)
 - Garder les meilleurs (25% par exemple)
 - Croiser les individus au hasard
 - Muter les individus avec une faible probabilité.

