

第七章 球队高级决策

机器人&智能技术实验室 A505/C501



本章的主要内容

- 概述
- UVA_Base_2003的决策
- 球队决策的一般设计模式
- SBSP跑位方法
- 传球对象的选择
- 局部战术之2Vs. 1示例
- 基于换位思考的多智能体协作
- 基于Advisor-Evaluator进攻模块

7.1 RoboCup高级决策概述

- 从足球比赛的角度来看，体现球队整体策略的就是球队的整体进攻和防守体系，当所有球员都按照相同的决策框架来进行决策时，球员的决策依据都相同，对其他队员的预测就比较准确，整个球队就能够体现出一定的协调性了。当Agent间的意愿一致的时候他们就会进行协作，当他们的意愿不一致的时候，就要进行协商，处理矛盾。



7.1 RoboCup高级决策概述

- (1). 整体策略指定了队伍的阵型以及其中队员的位置。更进一步，它可以决定什么情况下应该采用什么样的阵型。
- (2). 整体策略定义了阵型中不同的角色，并把角色分配给不同位置的球员，同时确定哪种异构球员适合哪种角色。
- (3). 对于每种类型的球员（后卫，中场，前锋等）球队策略应该根据它的角色确定分配给他哪些相应的行为。如对一个防守队员来说，一般它应该比进攻队员保守，它选择的动作也应该是偏向防守的。

7.1 RoboCup球队策略综述

- (4). 整体策略一般还包括球员应该如何根据场上形势来调整行为的信息。比如在特定的策略下，一个球员在对自己的动作进行选择的时候应该考虑到位置和位置所在的区域，同时还要考虑队友和对手的位置。
- (5). 整体策略应该指定如何协调各个球员的行为。因为不同的球员由于感知的不同，在策略的执行上可能出现不协调。如何减小这种潜在不协调的危险是策略的一部分。
- (6). 整体策略还应该能够在比赛中合理的管理球员的体力。如球员在比赛中如果自己的体力特别低，除非万不得已一般尽量减少跑动以恢复体力。

7.1 RoboCup球队策略综述

- 球队的整体策略还应该考虑其他一些因素：
 - 对手的强弱。对强弱不同的对手的动作是不一样的，如对手的截球能力强在传球的时候对传球的细节考虑就应该更细一点，对手弱，考虑的就少一点，可以为追求更好的结果而采用更加冒险的传球。
 - 对方球队的整体倾向。如对手是倾向于进攻还是倾向于防守。
 - 比赛的场上情况。比如说场上的比分，是落后还是领先。
 - 场上是否处于关键状态。如我们已经攻入对方禁区或对手攻入我方的禁区采取紧急动作
 - 其他一些因素。是否考虑换人、是否考虑球员类型的转换等等

7.2 UVA_Base_2003的决策

- 按开球时的位置进行站位。
- 是否能够看到球，若不可以，则寻找球的位置。
- 在可以看到球的情况下，队员就要判断是否可以踢球。若可以，就踢，否则就要找出最近的球员去踢球。
- 对基本的决策作出优化，主要是结合体力状况，预测球的可能位置，进行截球。以下是程序中不同体力值时对带球指令的解释。



7.3 球队决策的一般设计模式

- 进攻决策模块
- 防守决策模块



7.3.1 进攻体系

- 进攻是从本方队员控球开始，直到本方失去控球权为止。
- 进攻体系考虑的问题持球队员进行备选动作的选择，其他队员或进行策应，或进行合理的跑位。
- 可以分为战术配合（局部的几个球员）和战略配合。

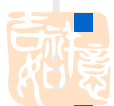


(1) 战术配合

- 战术触发条件
- 战术的实施过程
- 战术的终止条件
-

战术的设计主要涉及到2个方面：

- 战术本身的设计
- 通讯协议



(2) 持球队员的动作选择

- 条件行为树（if else规则）：这实际上是MAS中的策略学选取中的比较匹配函数、兴趣或爱好函数的方法。
- 基于效用对策论进行动作选择。



基于效用的对策论

- 各种可能行为（包括传球、射门、带球等）的成功率 p
- 也给出执行该行为的收益 U
- 最大化 $\max \sum p_{ij} \cdot \mu_{ij}$



(3) 接球队员的响应设计

- 传球队员一旦准备传球，就向需要响应的接球队员喊话(say)，可以出球的目标状态通过一定的通讯协议传递给接球队员，指导接球队员的跑位。



(4) 一般队员的跑位设计

- 一般队员按照**SBSP**跑位思想进行跑位。
- 注：清华采用把队员的跑位点通过神经网络做成关于球的位置和本身球员号码的一个函数 $P(B,i)$



7.3.2 防守体系

- 防守是从本方失球以后就开始的，一直到本方球员重新获得控球权。防守体系主要考虑的问题就是一般球员组成的防守体系和守门员的防守。更进一步就是考虑加入了守门员的防守体系。



防守动作

- Mark（盯人，主要是指盯无球队员）
- Block（卡位,堵截带球队员，防止其推进）
- Press(贴身回追带球队员)
- Formation（跑本位点）。
- Intercept(截球)

- 其它

- [注]：盯人的目的是防止被盯的对手接到传球；**Block**的目的是阻碍对方控球队员带球前进；跑本位点是指球员站在本位点处防守，即进行区域防守。

防守考虑要素

- 该体系用防守动作的目标点（对于**Mark**来说，防守动作的目标点在被盯队员附近；对于**Intercept**来说，防守动作的目标点在阻截点上）离本位点的远近
- 目标点离球员当前位置的远近
- 防守目标的对防守的威胁程度来作为一个防守动作的特征。
- 每个防守队员一般一个防守动作

防守规划



- 首先对所有的进攻-防守队员配对以及防守队员-本位点配对生成防守动作。
- 接着根据前面提到的防守动作的特征得出防守动作的优先级；
- 然后用分支定界的方法，求出这个最优的防守动作集；
- 最后防守队员从方案中查询应该有自己的执行的动作。



守门员的防守



- 守门员的防单刀球
- 守门员的站位
- 守门员的开球





7.4 基于场上形势的战术跑位(SBSP)

- SBSP首先是由FC Portugal提出的，它是结合了Peter Stone在CMU球队里面的阵型和位置的概念，并在此基础上引入了战术、场上情况、队员类型等概念。



SBSP的一些定义



定义 1: 阵型是通过站位来进行确定的。

$$Formations_i = \{ Formations_{i,1}, Formations_{i,2}, \dots, Formations_{i,n_{formations_i}} \}$$

$$Formation_{i,j} = (AgentPosi_{i,j,1}, AgentPosi_{i,j,2}, \dots, AgentPosi_{i,j,n_{Agent}})$$

$$\forall i=1..ntactics, \forall j=1..n_{formations_i}$$

定义 2: 阵型中 Agent 的站位是由基本参考位置、角色、位置的重要性组成。

$$Posi_{i,j,p} = (RefePosi_{i,j,p}, PosiRole_{i,j,p}, PosiImport_{i,j,p})$$

$$\forall i=1..ntactics, \forall j=1..n_{formations_i}, \forall p=1..n_{players_{i,j}}$$

定义 3: 预规划包括阵型转变规划以及随着时间变化的位置评估规划、角色评估规划、动作选择评估规划。

$$PresetPlans_i = \{ PrePlans_1, PrePlans_2, \dots, PrePlans_{n_{plans_i}} \}$$

$$PresetPlan_{i,k} = (PlanActInfo_{i,k}, PlanPosiEvo_{i,k}, PlanRolesEvo_{i,k}, PlanActEvo_{i,k})$$

$$\forall i=1..ntactics, \forall k=1..n_{plans_i}$$



SBSP的一些定义 (续)



定义 4: 战术是关于阵型、阵型触发规则、预规划的组合。

$$Tactic_i = (Formations_i, FormationActivRules_i, PresetPlans_i) \quad \forall i=1..ntactics$$

定义 5: 球队策略中的角色应包含积极特性、战略特性、场上重要形势规则。

$$Role_i = (Act_Chars_i, Stra_Chars_i, Cri_Situ_Rule_i)$$

定义 6: 基于位置的角色 (PosiRole) 定义了 Agent 的行为特性。

$$PosiRole_{i,j,p} \in \{1..nroles\}$$

$$\forall i=1..ntactics, \forall j=1..nfomations_i, \forall p=1..nplayers_{i,j}$$

定义 7: 球队策略是关于战术、战术触发规则、Agent 角色、对手模型策略、对友模型策略以及通讯协议的集合。

$$TeamStrategy = (Tactics, TactActivRules, Roles, Opp_ModStras, Team_ModStras, Comm_Prots)$$



SBSP的一些定义 (续)

$$Tactics = \{Tactics_1, Tactics_2, \dots, Tactics_{nTactics}\}$$

该集合表示所有的战术

$$TactActivRules = \{TactActivRules_1, TactActivRules_2, \dots, TactActivRules_s\}$$

该集合表示所有的战术触发规则

$$Roles = \{Role_1, Role_2, \dots, Role_{nRoles}\}$$

该集合表示 Agent 的角色，它决定了 Agent 的动作倾向。

$$Opp_ModStras = \{OppModStrat_1, OppModStrat_2, \dots, OppModStrat_n\}$$

该集合表示所有的对手模型策略

$$Team_ModStras = \{TeamModStrat_1, TeamModStrat_2, \dots, TeamModStrat_m\}$$

该集合表示所有的队友模型策略

$$Comm_Prots = \{CommProt_1, CommProt_2, \dots, CommProt_t\}$$

该集合表示所有的通讯协议。

SBSP的一些定义（续）



定义 8: 积极形势是指 Agent 确认此时场上形势是重要形势时的场上形势。它的确定原则是场上重要形势规则为真。

定义 9: 战略形势是指 Agent 确认此时场上形势不是重要形势时的场上形势。它的确定原则是场上重要形势规则为假。

定义 10: Agent 的 SBSP 是关于当前的战术、场上形势（在阵型中定义）、球员类型（定义了 Agent 的战略特征）的函数。





SBSP的使用

在使用的时候首先定义了:

全场信息 *Global_Infor*=(*Time*, *Result*, *StaticticsInfo*, *OppModInfo*);

场上形势信息 *Situ_Info*=(*BallPossInfo*, *BallInfo*, *TeamInfo*, *OppInfo*); 其中:

StaticsInfo=(*BallPossInfo*, *ShootInfo*, *PassInfo*, *DribbleInfo*, *CornersInfo*, *OffsideInfo*,
FreekickInfo, *KickInInfo*, *GoalieKickInfo*);

PositionalInfo=(*BallLowlevel*, *OffsideLines*, *TeammateLowlevelInfo*, *OpponentLowLevelInfo*,
InterceptionInfo, *GongestionInfo*, *DangerInfo*);

ActionInfo=(*PassInfor* ,*ForwardInfo*, *ShootInf*, ...)。



SBSP的使用（续）

然后定义了场上重要形势信息：

$Cri_situInfo_i = \{ Cri_SituRule_{i,1}, Cri_SituRule_{i,2}, \dots, Cri_SituRule_{i,ncriticalrules} \}$ 以及 Agent 的战

略特性：

$Stra_Char_i = (BallPositionalAttraction_i, PositionalRectangle_i, Behindball_i, ballAttraction_i, BallAttractionRegion_i)$ 。

根据球员当前所处的位置确定球员在相应阵型下的位置角色：

$AlocRole_p = PlayerRole (AlocPositioning_p)$



SBSP的使用（续）



- 最后得出了球员(Agent)在场上的战略跑位：

$$\begin{aligned} \text{PlayerStrategicPosition}_p = & \\ & \text{AdjustedPosition}(\text{ReferencePosition}(\text{AlocPositioning}_p) + \\ & \text{BallPosAttraction}(\text{AlocRole}_p, \text{BallPos}, \text{BallPosAttrac}, \text{PosRectangle}) + \\ & \text{BallAttraction}(\text{AlcoRole}_p, \text{BallPos}, \text{BallAttraction}, \text{BallAttractionRegion})), \\ & \text{BehindBall}(\text{AlocRole}_p), \\ & \text{OffsideConversion}(\text{AlocRole}_p)) \end{aligned}$$

注：(1)首先根据球员的位置角色，得出球员应该跑的基本位置，再考虑球所在位置区域对球员的吸引力、球本身对球员的吸引力，简单修正得到球员应跑的位置。

(2)考虑相应球员是否应该在球的后面*，一般后卫，守门员应在球的后面。

(3)考虑是否越位。

考虑到(1)、(2)、(3)最后得出球员的应该跑的战略位置。



SBSP小结

- SBSP是一个比较成功的球队整体策略的应用。它的基本思想是把场上状态分为积极的和战略的。球员的策略也分积极的和战略的。当场上状态是积极的时候，一般这也是也就是关键的时候，如已经攻入了对方禁区，或进行战术配合的时候，这时Agent就不在考虑在进行高层的决策直接采用反应式的方式，如进入禁区面对守门员形成单刀，这是只有一个动作选择，那就是射门。当场上状态进入战略状态的时候，如进攻时，控球队员是处于积极状态，它进行动作的选择，其他队员都进入战略状态，这时它们的选择就是跑位，这就是使用SBSP，跑到一个战略位置。防守时在考虑盯人、阻截球的情况下，其他队员就要选择合适的位置进行跑位。这时使用的也是SBSP。

SBSP的一般算法



- *IF* 场上是战略状态
 - *IF* 本方进攻
 - 进攻球员进行动作选择、其他队员按SBSP跑位
 - *ELSE*
 - 防守队员截球、盯人、按SBSP跑位
- *ELSE*
 - 采用积极状态下的即时动作



7.5 传球对象的选择

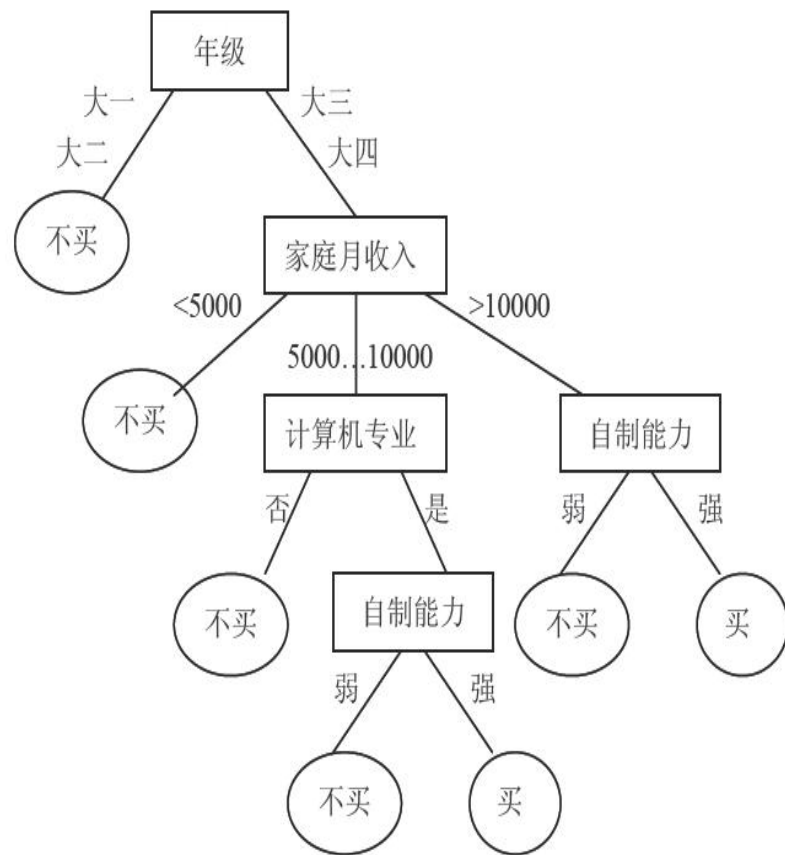


- 决策树
- 基于价值的学习



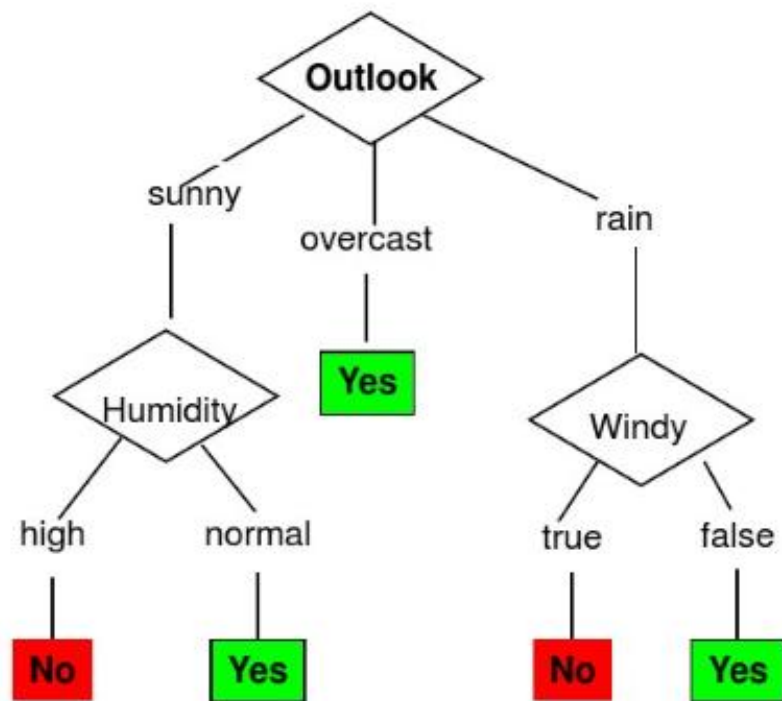
方法1-决策树方法

编号	是否买电脑	年级	家庭月收入	自制能力	计算机专业
1	不买	二	8000	强	是
2	买	三	6000	强	是
3	不买	一	7500	弱	否
4	不买	三	3000	强	是
5	买	四	12000	强	否
6	不买	二	7000	弱	是
....					



Whether go to play tennis

Day	Outlook	Temp	Humidity	Wind	Play
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No



Outlook: Sunny, Rain, Overcast

Humidity: High, Normal

Wind: Strong, Weak

Temp: Hot, Mild, Cool

典型决策树学习算法：ID3算法

- 基本的ID3算法通过自顶向下构造决策树来进行学习。构造过程是从“那一个属性将在树的根节点进行测试？”这个问题开始的。为了回答这个问题，使用 **(信息增益)** 来确定每一个实例属性单独分类训练样例的能力。分类能力最好的属性将被选作树的根节点。然后为根节点属性的每个可能值产生一个分枝，并把训练样例排列到该属性值对应的分枝之下。然后重复上述整个过程，用每个分支节点关联的训练样例来选取在该点被测试的最佳属性。直到终止条件 **(没有记录可分或没有属性)** 得到满足。

■ 用熵来度量训练样例的均一性即纯度

- S 表示训练集合
- 分类属性有 c 个不同的分类值

则 S 相对于 c 个状态的分类的熵定义为:

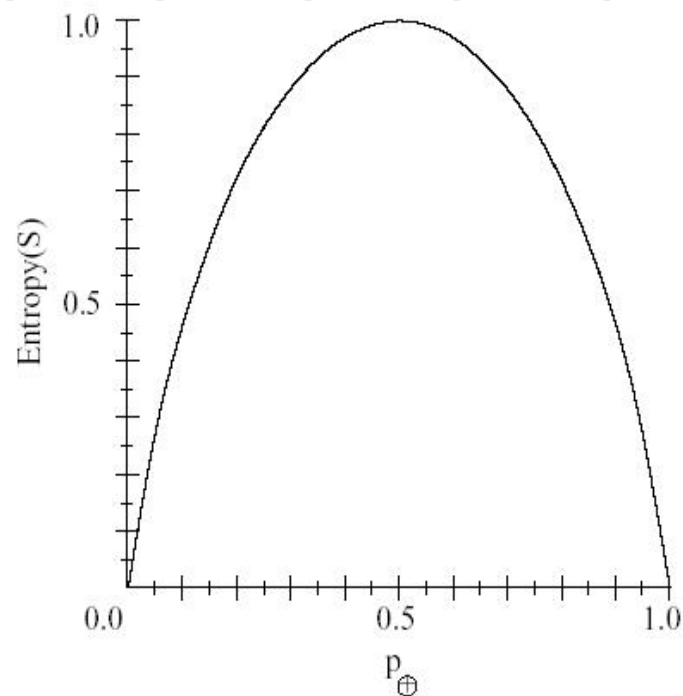
P_i 是 S 中属于类别 i 的比例

$$Entropy(s) = \sum_{i=1}^c -p_i \log_2 p_i$$

熵度量的是集合 S 的纯度，我们以分类值只有+，-两类的分类问题为例分析如图：

$$Entropy(s)$$

$$= -p_+ \log_2 p_+ - p_- \log_2 p_-$$



用信息增益来度量期望的熵降低

- 属性A
- 训练集合S
- $Values(A)$ 属性A的所有可能值的集合
- S_v 是S中属性A的值为v的子集

则属性A相对样例集合S的信息增益定义为:

$$Gain(S, A) \equiv Entropy(s) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

$Gain(S, A)$ 是由于知道属性A的值而导致的期望熵减少。换句话说来说, $Gain(S, A)$ 是由于给定属性A的值而得到的关于目标函数值的信息。

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

同样, 对于overcast而言, 有4个是类Yes, 0个是No, 即
 $p2=4, n2=0$; 计算 $I(p2, n2)$:

$$I(p2, n2)=0;$$

对于Rain也是这样: $p3=3, n3=2$,
 $I(p3, n3)=0.971$

考察了outlook这个属性以后, 期望信息要求

$$E(\text{outlook}) = (5/14) * I(p1, n1) + (4/14) * I(p2, n2) + (5/14) * I(p3, n3) = 0.694$$

属性Outlook的增益是

$$\text{Gain}(s, \text{outlook}) = I(p, n) - E(\text{outlook}) = 0.246$$

该示例中总共有14个对象, 9个类为Yes, 5个为No。分类要求的期望信息是:

$$I(p, n) = -(9/14) \log_2(9/14) - (5/14) \log_2(5/14) = 0.940$$

类似可以求出:
 $\text{Gain}(s, \text{temperature}) = 0.029$
 现在首先考虑属性outlook, 它的值是{sunny, overcast, rain}

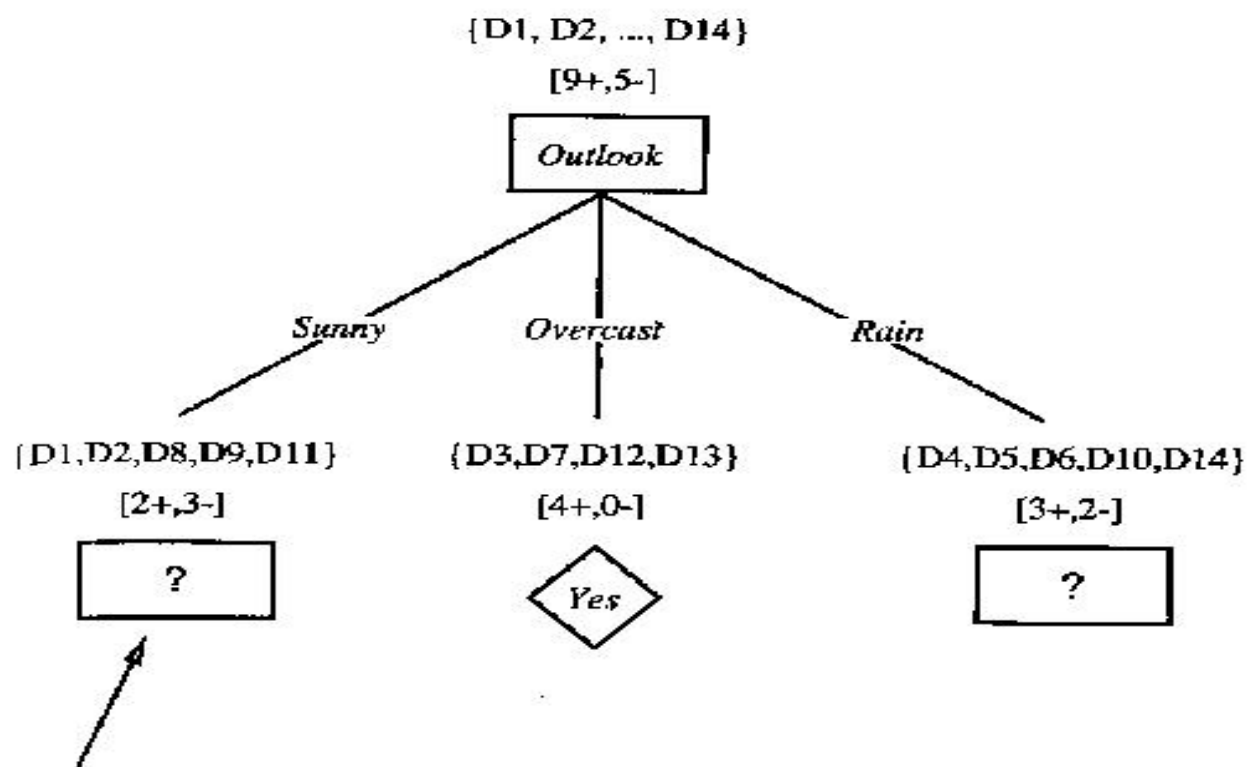
Gain(s, humidity)=0.151
 它的值是{normal, high}, 12个是类Yes, 3个是类No, 即

$$\text{Gain}(s, \text{windy}) = 0.048$$

$$I(p1, n1) = (2/5) \log_2(2/5) + (3/5) \log_2(3/5) = 0.971$$

由此可见使用outlook进行分类的信息增益最大, 因此选择outlook属性作为根节点属性。根据outlook属性的值把对象化为子树。





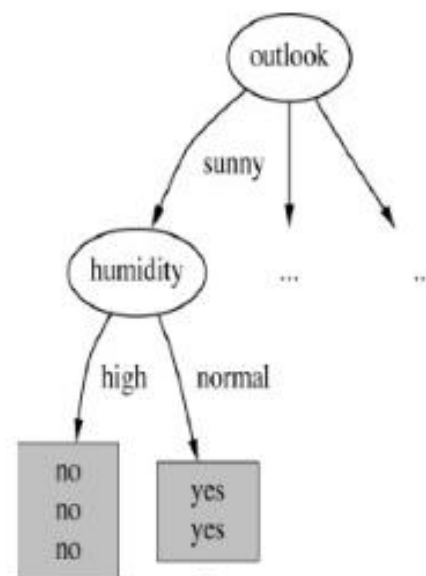
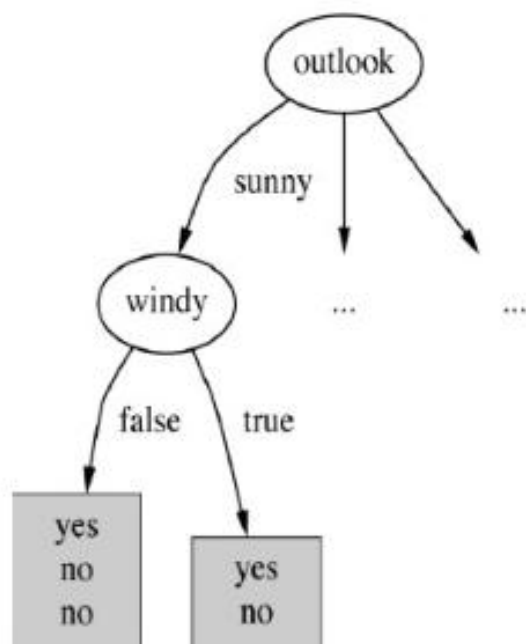
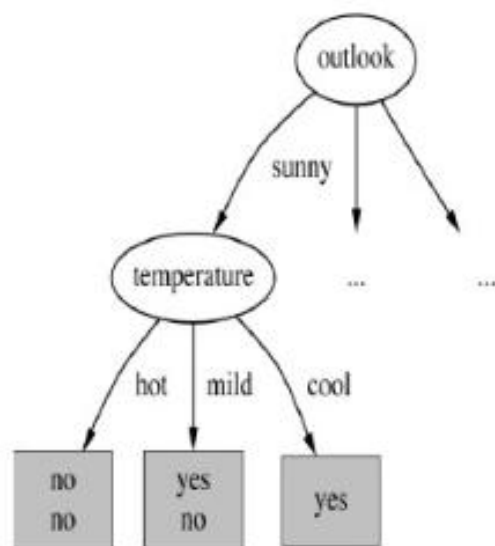
Which attribute should be tested here?

$$S_{\text{sunny}} = \{D1, D2, D8, D9, D11\}$$

$$\text{Gain}(S_{\text{sunny}}, \text{Humidity}) = .970 - (3/5) 0.0 - (2/5) 0.0 = .970$$

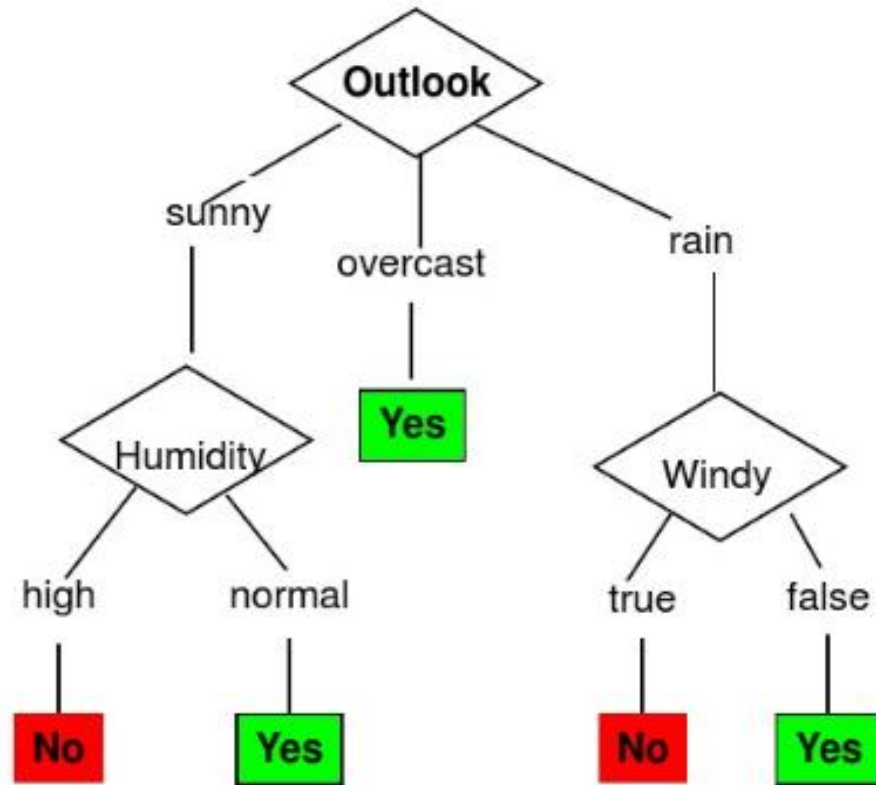
$$\text{Gain}(S_{\text{sunny}}, \text{Temperature}) = .970 - (2/5) 0.0 - (2/5) 1.0 - (1/5) 0.0 = .570$$

$$\text{Gain}(S_{\text{sunny}}, \text{Wind}) = .970 - (2/5) 1.0 - (3/5) .918 = .019$$



纯子节点

吉祥如意



$(\text{Outlook}=\text{Sunny} \wedge \text{Humidity}=\text{Normal}) \vee (\text{Outlook}=\text{Overcast}) \vee (\text{Outlook}=\text{Rain} \wedge \text{Wind}=\text{Weak})$

作业:

序号	属性						决策: 是否等待
	有否其他 选择	饿否	价格	餐馆类型	餐馆顾客人数	等待时间 (分钟)	
1	是	是	\$\$\$	法式	有人	0-10	是
2	是	是	\$	中餐	客满	30~60	否
3	否	否	\$	快餐	有人	0~10	是
4	是	是	\$	中餐	客满	10~30	是
5	是	否	\$\$\$	法式	客满	> 60	否
6	否	是	\$\$	意大利式	有人	0~10	是
7	否	否	\$	快餐	无人	0~10	否
8	否	是	\$\$	中餐	有人	0~10	是
9	否	否	\$	快餐	客满	> 60	否
10	是	是	\$\$\$	意大利式	客满	10~30	否
11	是	否	\$	中餐	无人	0~10	否
12	否	是	\$	快餐	客满	30~60	是

C4.5

- C4.5如何选择具有最高信息增益的属性:

使用“分裂信息（**split information**）”值将**gain**规范化

$$SplitInfo_A(D) = - \sum_{j=1}^v \frac{|D_j|}{|D|} \times \log_2 \frac{|D_j|}{|D|}$$

$$\frac{|D_j|}{|D|}$$

表示属性**A**第**j**个划分的权重

信息率

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo(A)}$$

C4.5算法概述



- ❑ C4.5算法是ID3算法的扩展
- ❑ 能够处理连续型的属性。首先将连续型属性离散化，把连续型属性的值分成不同的区间，依据是比较各个分裂点Gain值的大小。
- ❑ 缺失数据的考虑：在构建决策树时，可以简单地忽略缺失数据，即在计算增益时，仅考虑具有属性值的记录。



连续值的处理

☑ 选取(连续值的)哪个分界点？

■ 贪婪算法！

1. 排序

60 70 75 85 90 95 100 120 125 220

若进行“二分”，则可能有9个分界点。

例子：

60 70 75 85 90 95 100 120 125 220

分割成 $\text{TaxIn} \leq 80$ 和 $\text{TaxIn} > 80$

60 70 75 85 90 95 100 120 125 220

分割成 $\text{TaxIn} \leq 97.5$ 和 $\text{TaxIn} > 97.5$

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

✦ 实际上，这就是“离散化”过程

连续值的处理

2. 对**每个**候选的分界点，分别计算熵

例子: 测试以80分界的情形

(1). $\text{TaxIn} \leq 80$

$$\text{entropy}(\text{cheat}) = 0 \text{ bits}$$

(2). $\text{TaxIn} > 80$

$$\text{entropy}(\text{cheat}) = -\frac{3}{7} \log_2 \frac{3}{7} - \frac{4}{7} \log_2 \frac{4}{7} = 0.985 \text{ bits}$$

(3). 加权平均

$$\text{Info}(\text{TaxIn}|80) = \frac{3}{10} \times 0 + \frac{7}{10} \times 0.985 = 0.670 \text{ bits}$$

同理，测试以95分界的情形， $\text{Info}(\text{TaxIn}|95) = 0.600 \text{ bits}$

3. 比较取**每个**分界点的信息增益，选择最大的一个

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

传球决策树的应用

- 首先设定传球队员的位置，随机设置接球和其他队员的位置；
- 传球队员确认要传球；
- 其他队员进行跑位；
- 传球队员根据决策树确定接球队员。
- 接球队员和其他队员（指对手球员）都采用已经训练好的截球动作去获得控球权。
- 接球队员获得控球权就认为是一次成功的传球，否则，就认为失败。



1 Passer: ●

3 Teammates: ●

4 Defenders: ●

+2 other players per team





Sample Decision Tree Output

- **Teammate 2** : Success with confidence 0.8
- Teammate 3** : Failure with confidence 0.6
- Teammate 4** : Success with confidence 0.3



吉祥



吉祥

吉祥

吉祥

属性的选择 (174个)



- Distance and Angle to the receiver (2 features):
 - `passer-receiver-distance` = `dist(passer,receiver)`.
 - `passer-receiver-angle` = `ang(passer,receiver)`.
- Distance and Angle to other teammates sorted by angle from the receiver (18 features):
 - Sort the 9 other teammates such that $\forall i,j$, if $i < j$, then `rel-ang(passer,teammate-i,receiver) < rel-ang(passer,teammate-j,receiver)`.
 - For $i=1-9$, `passer-teammate(i)-distance` = `dist(passer,teammate-i)`.
 - For $i=1-9$, `passer-teammate(i)-angle` = `ang(passer,teammate-i)`.
- Distance and Angle to opponents sorted by angle from the receiver (22 features):
 - Sort the 11 opponents such that $\forall i,j$, if $i < j$, then `rel-ang(passer,opponent-i,receiver) < rel-ang(passer,opponent-j,receiver)`.
 - For $i=1-11$, `passer-opponent(i)-distance` = `dist(passer,opponent-i)`.
 - For $i=1-11$, `passer-opponent(i)-angle` = `ang(passer,opponent-i)`.
- Distance and Angle from the receiver to teammates sorted by distance (20 features):
 - Sort the 10 teammates (including the passer) such that $\forall i,j$, if $i < j$, then `dist(receiver,teammate-i) < dist(receiver,teammate-j)`.
 - For $i=1-10$, `receiver-teammate(i)-distance` = `dist(receiver,teammate-i)`.
 - For $i=1-10$, `receiver-teammate(i)-angle` = `ang(receiver,teammate-i)`.



- Distance and Angle from the receiver to opponents sorted by distance (22 features):
 - Sort the 11 opponents such that $\forall i,j$, if $i < j$, then $\text{dist}(\text{receiver}, \text{opponent-}i) < \text{dist}(\text{receiver}, \text{opponent-}j)$.
 - For $i=1-11$, $\text{receiver-opponent}(i)\text{-distance} = \text{dist}(\text{receiver}, \text{opponent-}i)$.
 - For $i=1-11$, $\text{receiver-opponent}(i)\text{-angle} = \text{ang}(\text{receiver}, \text{opponent-}i)$.
- Player distribution statistics from the passer's (45 features) and receiver's (45 features) perspectives:
 - For $i=1-6$, $\text{passer-teammates-ang}(i) = |\{k \in \text{teammates} \mid \text{rel-ang}(\text{passer}, k, \text{receiver}) \leq i \wedge \text{dist}(\text{passer}, k) \leq \text{dist}(\text{passer}, \text{receiver})\}|$.
 - For $i=1-6$, $\text{passer-opponents-ang}(i) = |\{k \in \text{opponents} \mid \text{rel-ang}(\text{passer}, k, \text{receiver}) \leq i \wedge \text{dist}(\text{passer}, k) \leq \text{dist}(\text{passer}, \text{receiver})\}|$.
 - For $i=1-6$, $\text{passer-players-ang}(i) = \text{passer-teammates-ang}(i) + \text{passer-opponents-ang}(i)$.
 - For $i=1-3, j=1-3$, $\text{passer-teammates-dist}(4i)\text{-ang}(4j) = |\{k \in \text{teammates} \mid \text{rel-ang}(\text{passer}, k, \text{receiver}) \leq 4i \wedge \text{rel-dist}(\text{passer}, k, \text{receiver}) \leq 4j\}|$.
 - For $i=1-3, j=1-3$, $\text{passer-opponents-dist}(4i)\text{-ang}(4j) = |\{k \in \text{opponents} \mid \text{rel-ang}(\text{passer}, k, \text{receiver}) \leq 4i \wedge \text{rel-dist}(\text{passer}, k, \text{receiver}) \leq 4j\}|$.
 - For $i=1-3, j=1-3$, $\text{passer-players-dist}(4i)\text{-ang}(4j) = \text{passer-teammates-dist}(4i)\text{-ang}(4j) + \text{passer-opponents-dist}(4i)\text{-ang}(4j)$.
 - For $i=1-6$, $\text{receiver-teammates-ang}(i) = |\{k \in \text{teammates} \mid \text{rel-ang}(\text{receiver}, k, \text{passer}) \leq i \wedge \text{dist}(\text{receiver}, k) \leq \text{dist}(\text{passer}, \text{receiver})\}|$.
 - For $i=1-6$, $\text{receiver-opponents-ang}(i) = |\{k \in \text{opponents} \mid \text{rel-ang}(\text{receiver}, k, \text{passer}) \leq i \wedge \text{dist}(\text{receiver}, k) \leq \text{dist}(\text{passer}, \text{receiver})\}|$.
 - For $i=1-6$, $\text{receiver-players-ang}(i) = \text{receiver-teammates-ang}(i) + \text{receiver-opponents-ang}(i)$.
 - For $i=1-3, j=1-3$, $\text{receiver-teammates-dist}(4i)\text{-ang}(4j) = |\{k \in \text{teammates} \mid \text{rel-ang}(\text{receiver}, k, \text{passer}) \leq 4i \wedge \text{rel-dist}(\text{receiver}, k, \text{passer}) \leq 4j\}|$.
 - For $i=1-3, j=1-3$, $\text{receiver-opponents-dist}(4i)\text{-ang}(4j) = |\{k \in \text{opponents} \mid \text{rel-ang}(\text{receiver}, k, \text{passer}) \leq 4i \wedge \text{rel-dist}(\text{receiver}, k, \text{passer}) \leq 4j\}|$.
 - For $i=1-3, j=1-3$, $\text{receiver-players-dist}(4i)\text{-ang}(4j) = \text{receiver-teammates-dist}(4i)\text{-ang}(4j) + \text{receiver-opponents-dist}(4i)\text{-ang}(4j)$.

传球决策树的生成:

```
passer-opponents-ang6 > 0 : F (1266.0/251.3)
passer-opponents-ang6 <= 0 :
  passer-opponent1-angle > 13 : S (1054.6/290.5)
  passer-opponent1-angle <= 13 :
    passer-teammates-ang6 <= 0 :
      passer-receiver-distance <= 22 :
        passer-opponent1-distance <= 20.9 :
          passer-players-dist8-ang12 <= 3 : S (162.0/72.9)
          passer-players-dist8-ang12 > 3 : F (15.0/5.8)
        passer-opponent1-distance > 20.9 :
          passer-opponent2-distance <= 21.1 :
            passer-opponents-dist12-ang8 <= 1 :
              passer-teammates-dist8-ang8 <= 1 :
                passer-opponents-dist12-ang4 <= 0 :
                  passer-receiver-distance <= 20.3 : S (52.2/8.6)
                  passer-opponents-dist12-ang4 > 0 :
                    passer-opponents-dist4-ang12 <= 1 : S (60.5/20.0)
                passer-opponent2-distance > 21.1 :
                  receiver-teammates-dist8-ang12 <= 1 : S (704.3/139.6)
            passer-receiver-distance > 22 :
              passer-opponent1-distance <= 23.1 :
                passer-opponents-dist12-ang12 <= 0 :
                  receiver-players-dist8-ang12 <= 1 : S (87.5/44.7)
                passer-opponents-dist12-ang12 > 0 :
                  passer-opponents-dist12-ang8 > 0 : F (191.0/63.0)
                  passer-opponents-dist12-ang8 <= 0 :
                    passer-opponents-dist4-ang12 > 1 : S (14.0/6.8)
                    passer-opponents-dist4-ang12 <= 1 :
                      passer-teammate1-distance <= 19.5 : S (15.0/6.8)
                      passer-teammate1-distance > 19.5 : F (234.0/91.7)
              passer-opponent1-distance > 23.1 :
                passer-opponents-dist12-ang12 <= 1 : S (665.9/259.2)
                passer-opponents-dist12-ang12 > 1 :
                  passer-players-dist4-ang12 <= 1 : F (11.0/5.6)
                  passer-players-dist4-ang12 > 1 :
                    passer-opponents-dist4-ang8 <= 0 : S (49.0/19.9)
                    passer-opponents-dist4-ang8 > 0 :
                      passer-opponent2-distance <= 23.1 : F (85.0/26.5)
                      passer-opponent2-distance > 23.1 :
                        passer-opponents-dist12-ang12 <= 2 :
                          receiver-opponent3-angle <= 48 : F (9.6/2.2)
                          receiver-opponent3-angle > 48 : S (108.4/43.0)
      passer-teammates-ang6 > 0 :
        passer-teammates-ang5 > 0 : F (21.0/7.0)
```

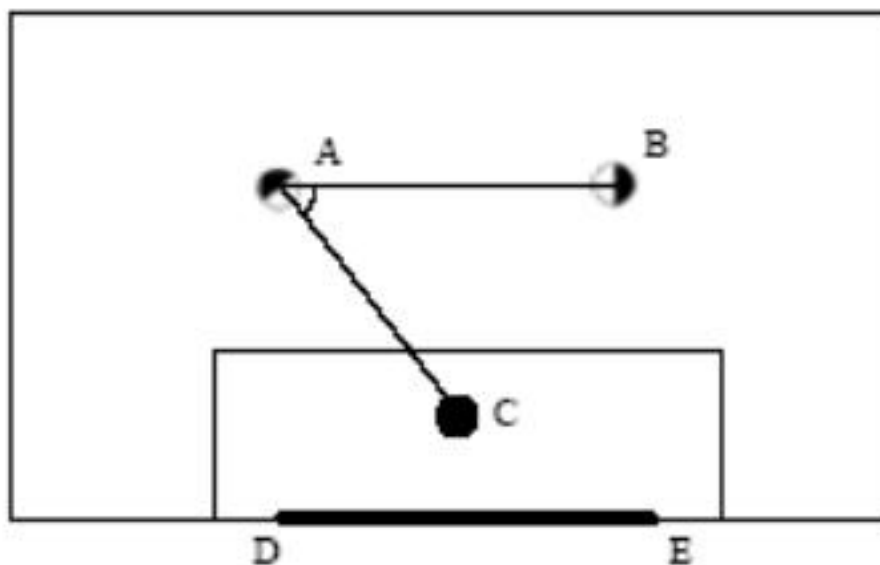
传球-基于价值的学习



- 传球的考虑因素:
- 位置价值、是否可传、越位程度、可信度、得到球时的射门角度、通讯得到的此球员的进攻或防守欲望、是否是守门员、是否是前锋、球员类型、传球距离、是否为一个突破位置、该队员体力是否充沛、该队员被人的盯防程度、传球失败的可能性与危险系数的乘积、传球成功的可能性与危险系数的乘积、对此队员的传球成功率记录、接应队员是否在一个对我方比较有威胁的区域、接应队员是否在一个对对方有威胁的区域、接应队员的进攻方向是否有人阻挡、对此队员的信任度、是否有可能进入到某一战术系统中、距离对方球门的距离、距离我方球门的距离.
- 哪些属性有意义,比较重要?



7.6 局部战术之2 vs 1问题



Q学习!

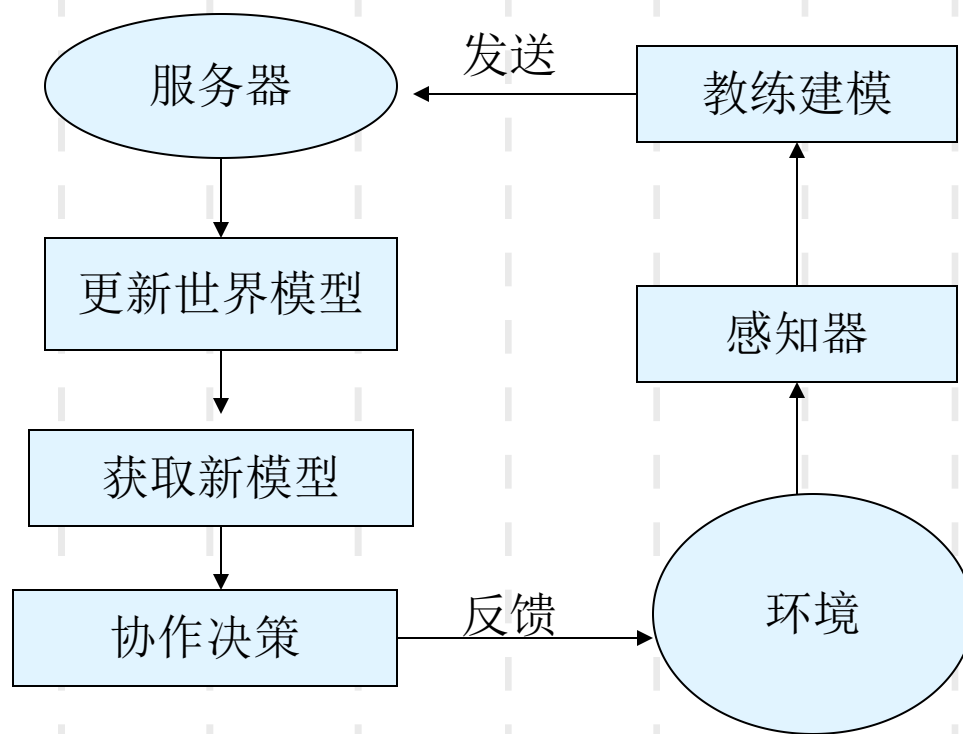
CMAC (小脑关节模型, Cerebellar Model Articulation Controller, 简称CMAC)网络对连续状态空间进行泛化表示

控球球员智能体可选的动作集为{Shoot(), Dribble(), Pass()},
无球队员可选的动作集为{Position()}。

7.7 基于换位思考的多智能体局部协作

- 对**Agent**行为进行预测，来实现对智能体的世界状态模型更新的深入研究。
- 目的让智能体**A**了解智能体**B**的意图;
- 框架
 - 首先，教练通过感知器感知到所有队友的当前信息，包括位置，速度，身体角度等;
 - 其次，教练利用当前信息，根据高斯分布抽象出智能体行为模式，比如带球，传球，射门等模式;
 - 然后，将新的模式反馈给环境;再者，队友智能体感知到新的行为模式，利用改进**Q** 学习得到最好的协作动作;
 - 最后，执行决策。

换位思考模型框架图



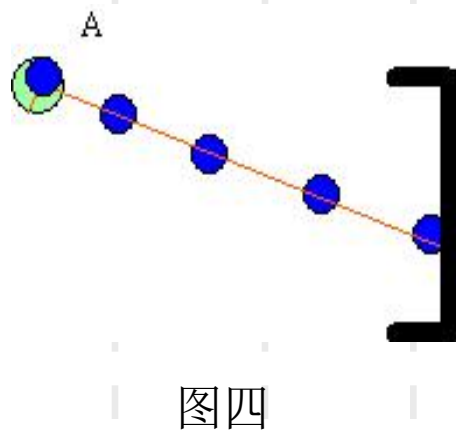
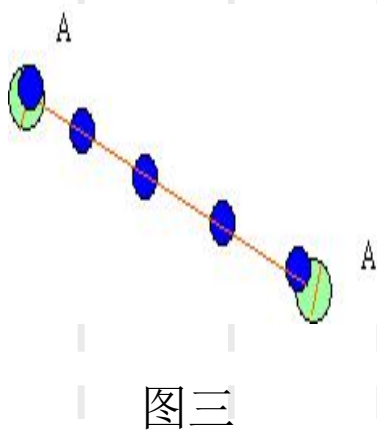
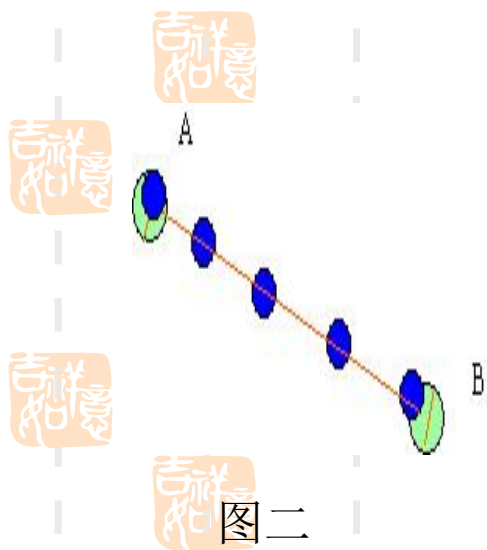
换位思考模型解决的问题

换位思考模型是为了减少智能体有限视觉范围获取的信息做出的决策误差而提出的新模型。该模型的核心思想是利用**教练建模**预测智能体行为模式，并更新世界状态模型，这样每个智能体可以预测其他智能体的行为模式，强化学习出自己最佳的协作策略。



教练建模

教练建模在获取其他智能体精确行为的过程中起到关键作用。模式即是球员在当前环境下的行为意识。比如常见的抽象模式有带球，传球，持球，射门等等。



- 由于仿真比赛环境是动态和随机的，可以假定每个周期智能体的行为是正态分布的(智能体是独立做出决策的)，独立正态分布和服从 χ^2 分布。因此我们可以利用 χ^2 分布预测智能体的行为模式。

$$\chi^2 = \sum_{i=1}^k \frac{(O_i - E_i)^2}{E_i}$$

利用上述公式分布计算传球，带球等模式匹配的概率，并判断其所属模式。若计算结果小于分布表中的给定值，则判断当前智能体行为模式为该模式。

基于换位思考模型的多智能体协作学习

- 如何利用换位思考得到的精确智能体模式，做出最好的决策动作是本文的另一个要讨论的关键性问题。
- 传统的学习算法只是学习得到单个智能体的最有效动作。为了得到整个球队的最有效的协作动作，本文在传统Q学习算法的基础上考虑了其他智能体的行为模式，即如果B感知到A的准确行为模式是传球，则B在协作中执行接应球的动作回报为1，其他动作为0。加入了其他智能体行为模式的考虑可以更精确的实现多智能体之间的协作的准确度。学习算法迅速收敛对于实时性要求很高的RoboCup具有很强的实际应用价值。

协作学习算法研究:

将仿真环境信息抽象为四元组 如下:

$(n; S_1 \dots S_n; A_1, \dots, A_n; R_1, \dots, R_n; CR_1, \dots, CR_n)$ 其中 n 代表智能体个数, S_i 代表智能体的状态, A_i 代表智能体 i 采取的动作, R_i 代表智能体 i 在状态 S_i 下执行动作 A_i 的回报, CR_i 代表智能体 i 在状态 S_i 下根据其他所有智能体的行为模式执行动作 A_i 的平均回报。每执行一个特定动作, 得到相应的环境回报 R_i 和模式平均回报 CR_i

$$CR_i = \frac{1}{n-1} \sum_{j=1, j \neq i}^{j=n} CR_{ij}$$

算法流程:

- 1.对每个状态 S_i , 动作 A_i , 初始化 $Q(S_i, A_i)$ 为0;
- 2.选择任一动作 A_i , 执行该动作;
- 3.记录当前的回报 R_i 和平均模式回报 CR_i ;
- 4.观察新的状态, 用以下公式更新Q值表;

$$Q(S_i, A_i) = R_i + \gamma \max_{a'} Q(S_i', A_i') + \eta CR_i$$

■ 其中, γ, η 为学习比率, 本文中取的经验值分别为0.8, 0.7;

■ 5.更新 $S_i = S_i'$, 若训练没有终止, 循环步骤2至5, 否则退出。

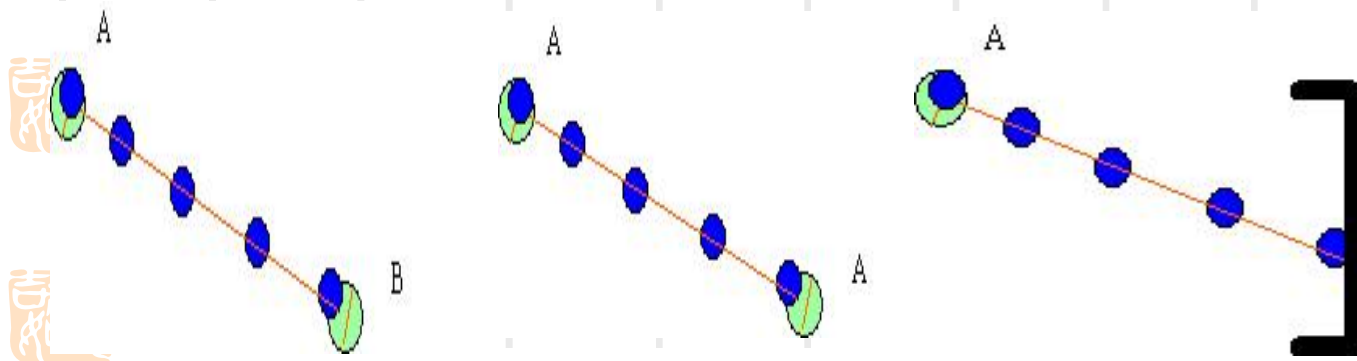
该模型在传球中的应用

吉祥如意

- 换位思考模型在传球中的应用方案

根据图2，图3，图4判定当前队友的行为模式。利用公式(1)计算当前的行为模式是否为传球模式。其中 O_i 表示实际的传球次数， E_i 代表期望的传球次数。

- 采用523阵型，即5个后卫，2个中场，3个前锋的阵型模式，统计2号分别传球给6号，7号，8号，10号的部分结果如下表所示。



吉祥如意

吉祥如意

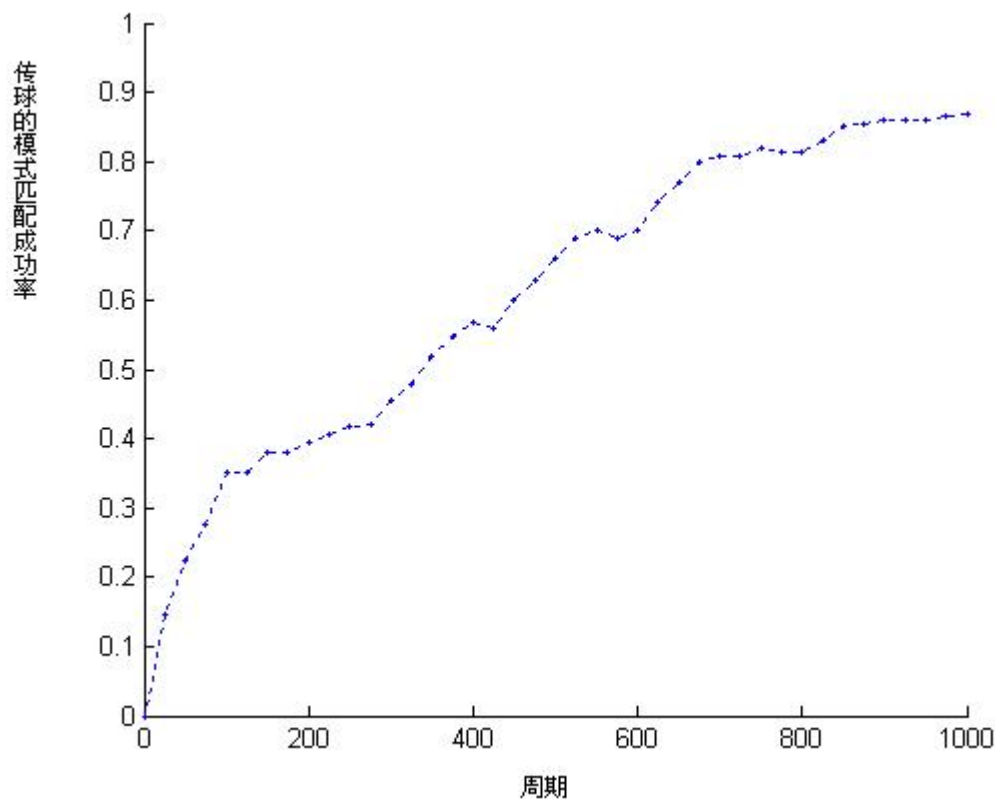
吉祥如意

吉祥如意

	6号球员	7号球员	8号球员	10号球员	总计
实际传球次数	93	454	272	181	1000
期望传球次数	80	470	280	170	1000
χ^2	2.11	0.545	0.23	0.711	3.596

从上表看到，2号给6号，7号，8号，10号的实际传球次数和期望传球次数，并利用公式计算出最终结果为3.596，这里取 $\alpha=0.1$ ，即要求模式匹配概率达到90%。从分布表中得到的值为6.251，由于 $3.596 < 6.251$ ，我们可以判定球员2号的当前行为模式为传球。

传球的模式匹配与比赛周期关系图



图五

- 当前6号球员控球，并且其他球员站在6号的角度思考6号球员的行为模式为传球，表7.6分别表示在6号为传球的条件下，其他队友的所采取的行为模式的Q值。比如9号最好的决策是接应，而10号最好的决策是跑位。

Q 值	盯防	跑位	接应	截球
2号	63	20	5	5
3号	57	15	10	6
4号	35	70	5	5
5号	15	68	7	1
7号	2	80	34	31
8号	59	33	4	2
9号	1	6	80	4
10号	2	87	3	5
11号	1	77	36	1

7.7 基于换位思考的多智能体局部协作

平均传球率	Apollo	WE2008	AmonnsisNQ
传统方法	0.815	0.753	0.8
本文方法	0.895	0.820	0.867

7.8基于Advisor-Evaluator进攻设计

- Evaluator从Advisor提交的所有Method中选取价值最高Method，然后提交给动作执行层执行。
 - Shoot-Advisor, Pass-Advisor和Dribble-Advisor;
 - 这3个 Advisor提交的策略具有以下三个参数：
 - 类型参数（method.type）用来说明策略类型是Shoot, Pass还是Dribble。
 - 可能性参数（method.probability）是一个0-1之间的实数，是Advisor模块对该策略被成功执行的可能性评价。
 - 位置参数（method.position）用来描述球在该策略实施后的场上位置。

$$\text{Score}(\text{method}) = x \times \text{utility}(\text{type}) * \text{method.probability} + y \times t(\text{position})$$