

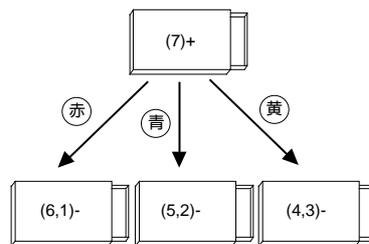
## マッチ箱とピースによるゲームの学習

- ・ 暗記学習
- ・ ゲームの戦略を学習
- ・ コンピュータを使わなくても、ゲームの結果によってピースを出し入れするという単純な操作で学習はできる
- ・ マッチ箱と人間で対戦、またはマッチ箱同士で対戦

1

## マッチ箱とピースの使い方

- ・ ゲームのすべての局面に対応するマッチ箱を用意する
- ・ 各マッチ箱には、その局面で可能な手に対応する色違いのピースを入れる
- ・ マッチ箱の手番の時には、その局面のマッチ箱をよく振って中からピースを1個取り出し、その色が示す手を指す



2

## ピースの出し入れ(1)

- ・ 勝負がついたときに、ピースの出し入れを行う
- ・ マッチ箱が勝ったときは、最初から最後までその局面のマッチ箱に、指した手と同じ色のピースを1個ずつ追加する
  - ・ その手を指す確率が増える
- ・ マッチ箱が負けたときは、最後の局面のマッチ箱から、最後に指した手と同じ色のピースをすべて取り去る
  - ・ その手は二度と指さなくなる

3

## ビーズの出し入れ(2)

---

- ・ 何回か負けると、空のマッチ箱ができることがある
- ・ 空のマッチ箱の局面に達したら、そこで負けを認める
- ・ その一つ手前のマッチ箱から、最後に指した手と同じ色のビーズをすべて取り去る
- ・ 十分な回数繰り返すと、負けにつながる手は指さなくなる

4

## 特徴

---

マッチ箱同士で対戦すると、勝手に強くなっていく

- 最終的には、どちらかの必勝になるか、または必ず引き分けになる

簡単なゲームでしかできない

- 完全なゲームの木を作る必要がある
- 三目並べくらいなら可能

5

## Samuel のチェッカープログラム

---

暗記、例示学習

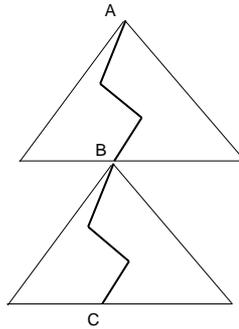
ゲームの局面の評価値、評価関数のパラメータを学習  
かなり強いプレイヤーになる（世界チャンピオンよりは弱い）



6

## 評価値の学習

- ある局面に対する評価値をミニマックス法を用いて計算したら、その局面と評価値の対を記憶しておく
- 先読みの葉のところに、記憶している局面が現われたら、静的評価関数の代わりに記憶している評価値を用いる
- 実質的に先読みの手数を長くする効果がある



7

## 記憶の組織化

- 局面の特徴によってインデックスをつけ、たぶん必要とされるであろう順序でテープに書き込む
  - 当時は外部記憶装置は磁気テープであった！
- 計算対記憶のトレードオフ
  - 記憶している局面には年齢がつけられる
  - メモリテープが書き換えられるとき、すべての局面は1歳だけ年をとる
  - 検索された局面の年齢は2で割る
  - 非常に古い局面はテープに書き戻されない

8

## 評価関数の学習

- 静的評価関数  $S = w_1 s_1 + \dots + w_n s_n$  ( $w_i$  重み,  $s_i$  特徴の値)
- 関数中にどのような特徴を用いるべきか？
- 特徴を結合して評価値を作り上げるためにどのような重みをつければよいか？

9

### 重みの決定(1)

- 完全なゲームの木ができれば評価関数は要らない
  - 評価値は真の値(勝ち、引き分け、負け)の近似値
- 正しい判断を下すのに役立つ特徴の重みは増加、他は減少
  - 1回のゲームで1度だけ調整
    - 遅すぎる
    - どの手が勝因または敗因か？ (credit assignment の問題)
  - 1手ごとに調整
    - それが正しい判断かどうか、どうやってわかるか？
    - 教師により模範手が与えられれば、その局面の特徴の値を用いて調整できる

10

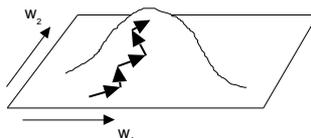
### 重みの決定(2)

- 静的評価関数の値が、ゲームの木の下からミニマックス法で上がってくる評価値にどれだけ対応するか
  - 先読みした値のほうが正確であると考えられる
- = ミニマックス法による評価値 - 静的評価関数による評価値
- が正なら過小評価だから正の項の重みを増加, 負の項の重みを減少させるべき
- 1手ごとに各項の符号と の符号が一致するかしないかを記録し、その相関係数によって各重みを変化させる

11

### 重みの決定(3)

- の値が0に近づくように重みを変化させる
- 重みづけ空間の中で山登り法による探索を行う
- 局所的な極大値の問題
  - 自分のコピーと対戦
  - コピーは学習しない
  - 強くなったら重みをコピーする
  - 3回負けたら重みを大きく動かす



12

## 特徴の決定

---

- ・ 評価多項式には38の特徴のうち16だけ含まれる
- ・ 残りの特徴は待ち行列に入れる
- ・ 多項式の中の各特徴について、重みが最も0に近かった回数を数える
- ・ 回数が32を越えたら、その特徴を多項式から取り除き、待ち行列の最後に入れる
- ・ 待ち行列の先頭の特徴を多項式に入れる

13

## 特徴表の学習

---

- ・ 評価値を多次元の配列に格納する（各次元が特徴に対応）
  - ・ 多項式より情報量が多い
- ・ 熟練者の対戦の棋譜を読むことにより学習
- ・ 棋譜に現われる局面には高い評価値、現われない局面には低い評価値を割り当てるようにする
- ・ 多項式の評価関数より強い

14