Chapter 15: Associative Learning

Brandon Morgan

2/1/2021

E15.9

Please read the exercise in the book for a full explanation of the background of the problem. Only the code will be used here:

The outstar rule is given by $w_j(q) = w_{ij} + \alpha(a_i(q) - w_{ij}(q-1))p_j(q)$.

```
w0 = diag(2)
w = matrix(0, ncol=3, nrow=2)
bias = matrix(-0.5, nrow=2, ncol=1)
p00= matrix(0, nrow=2, ncol=1)
p01 = matrix(-1, nrow=2, ncol=1)
p02 = matrix(c(1, -1), ncol=1)
p03 = matrix(c(1, -1), ncol=1)
p04 = matrix(1, nrow=2, ncol=1)

p1 = matrix(c(1,0,0), ncol=1)
p2 = matrix(c(0,1,0), ncol=1)
p3 = matrix(c(0,0,1), ncol=1)
```

```
hardlim = function(x) {
      result = matrix(0, nrow=nrow(x), ncol=1)
      for(i in 1:nrow(x)) {
        if(x[i,1]<0) {
          result[i,1] = 1
        }
        else {
          result[i,1] = -1
        }
      }
      result
}

outstar = function(w0, w, input0, input, bias, learning_rate, maxIter, mask, tol) {
  size = length(input)
  index = 1
  iter = 0



  prevWeight = w
```

```
  while(TRUE) {

    if(index > size) {
      index = 1
    }

    p0 = input0[[index]]

    p = input[[index]]

    n = w0%*%p0+w%*%p+bias

    a = hardlim(n)
    for(i in 1:ncol(w)) {
      w[,i] = w[,i]+learning_rate*(a-w[,i])*(p[i,1])
    }

    index = index + 1
    if(iter > maxIter-1){
      print(sprintf("   MAX ITER TAKEN"))
      print("Final Weights: ")
      print(w)
      return(w)
    }
    if(!mask) {
      print(sprintf("   ITERATION: %d", iter+1))
      print("n: ")
      print(n)
      print("a: ")
      print(a)
      print(sprintf("New weights:"))
      print(w)
    }

    iter = iter + 1
   # print(norm(prevWeight-w, "F"))
    if(norm(abs(prevWeight)-abs(w),"F")<tol) {
      break
    }
    prevWeight=w
  }
  print(sprintf("   ALGORITHM CONVERGED"))
  print(sprintf("Iter taken: %d", iter))
  print("Final Weights: ")
  print(w)
  return(w)
}
```

```
w=outstar(w0, w, list(p04, p03, p01, p03, p02,p04),
          list(p1, p2, p3, p2, p3, p1), bias, 0.6, 10000, 1, 1e-7)
```

```
## [1] "   ALGORITHM CONVERGED"
## [1] "Iter taken: 57"
```

```
## [1] "Final Weights: "
##            [,1]       [,2]      [,3]
## [1,] -0.7948718 -0.7948718 0.4285714
## [2,] -0.7948718  1.0000000 1.0000000
```

Now if they do not push any buttons:

President

```
n=w0%*%p00+w%*%p1+bias
hardlim(n)
```

```
##      [,1]
## [1,]    1
## [2,]    1
```

We can see that el Presidente goes to floor 4

Vice-President

```
n=w0%*%p00+w%*%p2+bias
hardlim(n)
```

```
##      [,1]
## [1,]    1
## [2,]   -1
```

Vice-President goes to floor second floor

Chairman

```
n=w0%*%p00+w%*%p3+bias
hardlim(n)
```

```
##      [,1]
## [1,]    1
## [2,]   -1
```

Chairman goes to floor second floor

New input

President pushes 3, Vice-President pushes 2, Chairman pushes 4

```
w=outstar(w0, w, list(p03, p02, p04), list(p1, p2, p3), bias, 0.6,
          10000, 1, 1e-7)
```

```
## [1] "   MAX ITER TAKEN"
## [1] "Final Weights: "
##           [,1]      [,2]      [,3]
## [1,] 0.2820513 0.2820513 0.2820513
## [2,] 1.0000000 1.0000000 0.2820513
```

Now if they do not push any buttons:

President

```
n=w0%*%p00+w%*%p1+bias
hardlim(n)
```

```
##      [,1]
## [1,]    1
## [2,]   -1
```

We can see that el Presidente goes to floor 2

Vice-President

```
n=w0%*%p00+w%*%p2+bias
hardlim(n)
```

```
##      [,1]
## [1,]    1
## [2,]   -1
```

Vice-President goes to floor second floor

Chairman

```
n=w0%*%p00+w%*%p3+bias
hardlim(n)
```

```
##      [,1]
## [1,]    1
## [2,]    1
```

Chairman goes to floor 4th floor