**201533661 이승수’s Machine Learning homework#6**

**Q) Suppose 3 coin-toss experiment were done using 2 coins.(10 tosses with 2 coins)**

|  |  |
| --- | --- |
| **Case 1** | **H T T T H H T H T H** |
| **Case 2** | **H H H H T H H H H H** |
| **Case 3** | **H T H H H H H T H H** |

**Start with initial guess for unknown parameters, p=0.1,q=0.9, Show your work out until parameter converge.**

**[Iteration: 1]**

Start with (p=0.1, q=0.9).

We can get P(A|S) and P(B|S) with method at below.

P(S|A)=p^count(H) \* (1-p)^count(T)

P(S|B)=q^count(H) \* (1-q)^count(T)=1-P(S|A)

P(A|S)=P(S|A)\*0.5 / (P(S|A)\*0.5 + P(S|B)\*0.5)

P(B|S)=P(S|B)\*0.5 / (P(S|A)\*0.5 + P(S|B)\*0.5)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | P(A | S) | P(B | S) | Count(H) | Count(T) |
| Case 1 | 0.5 | 0.5 | 5 | 5 |
| Case 2 | 0.5 | 0.5 | 9 | 1 |
| Case 3 | 0.5 | 0.5 | 8 | 2 |

Then, we can get expected number of heads of each cases as P(A|S)\*count(H). Do same with tail of each cases.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | EN of H of coinA | EN of T of coinA | EN of H of coinB | EN of T of coinB |
| Case 1 | 2.5 | 2.5 | 2.5 | 2.5 |
| Case 2 | 4.5 | 0.5 | 4.5 | 0.5 |
| Case 3 | 4.0 | 1.0 | 4.0 | 1.0 |
| Sum of all Case | 11 | 4 | 11 | 4 |

We can update p and q as sum(EN\_H) / (sum(EN\_H)+sum(EN\_H)) of coin A and coin B.

If updated p and q didn’t changed, stop iteration.

|  |  |  |
| --- | --- | --- |
|  | Before | after |
| P | 0.1 | 0.73 |
| Q | 0.9 | 0.73 |

**P,Q is updated, so do iteration again.**

**[Iteration: 2]**

Start with (p=0.73, q=0.73).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | P(A | S) | P(B | S) | Count(H) | Count(T) |
| Case 1 | 0.5 | 0.5 | 5 | 5 |
| Case 2 | 1.0 | 0.0 | 9 | 1 |
| Case 3 | 1.0 | 0.0 | 8 | 2 |

Then, we can get expected number of heads of each cases as P(A|S)\*count(H). Do same with tail of each cases.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | EN of H of coinA | EN of T of coinA | EN of H of coinB | EN of T of coinB |
| Case 1 | 2.5 | 2.5 | 2.5 | 2.5 |
| Case 2 | 9.0 | 1.0 | 0.0 | 0.0 |
| Case 3 | 8.0 | 2.0 | 0.02 | 0.005 |
| Sum of all Case | 19.5 | 5.5 | 2.52 | 2.505 |

We can update p and q as sum(EN\_H) / (sum(EN\_H)+sum(EN\_H)) of coin A and coin B.

If updated p and q didn’t changed, stop iteration.

|  |  |  |
| --- | --- | --- |
|  | Before | after |
| P | 0.73 | 0.78 |
| Q | 0.73 | 0.50 |

**P,Q is updated, so do iteration again.**

**[Iteration: 3]**

Start with (p=0.78, q=0.5).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | P(A | S) | P(B | S) | Count(H) | Count(T) |
| Case 1 | 0.13 | 0.86 | 5 | 5 |
| Case 2 | 0.96 | 0.04 | 9 | 1 |
| Case 3 | 0.87 | 0.12 | 8 | 2 |

Then, we can get expected number of heads of each cases as P(A|S)\*count(H). Do same with tail of each cases.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | EN of H of coinA | EN of T of coinA | EN of H of coinB | EN of T of coinB |
| Case 1 | 0.66 | 0.66 | 4.33 | 4.33 |
| Case 2 | 8.64 | 0.96 | 0.35 | 0.04 |
| Case 3 | 6.97 | 1.74 | 1.02 | 0.25 |
| Sum of all Case | 16.27 | 3.36 | 5.72 | 4.63 |

We can update p and q as sum(EN\_H) / (sum(EN\_H)+sum(EN\_H)) of coin A and coin B.

If updated p and q didn’t changed, stop iteration.

|  |  |  |
| --- | --- | --- |
|  | Before | after |
| P | 0.78 | 0.83 |
| Q | 0.50 | 0.55 |

**P,Q is updated, so do iteration again.**

**[Iteration: 4]**

Start with (p=0.83, q=0.55).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | P(A | S) | P(B | S) | Count(H) | Count(T) |
| Case 1 | 0.05 | 0.95 | 5 | 5 |
| Case 2 | 0.98 | 0.02 | 9 | 1 |
| Case 3 | 0.93 | 0.07 | 8 | 2 |

Then, we can get expected number of heads of each cases as P(A|S)\*count(H). Do same with tail of each cases.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | EN of H of coinA | EN of T of coinA | EN of H of coinB | EN of T of coinB |
| Case 1 | 0.28 | 0.28 | 4.71 | 4.71 |
| Case 2 | 8.88 | 0.99 | 0.11 | 0.01 |
| Case 3 | 7.42 | 1.95 | 0.57 | 0.14 |
| Sum of all Case | 16.58 | 3.12 | 5.41 | 4.87 |

We can update p and q as sum(EN\_H) / (sum(EN\_H)+sum(EN\_H)) of coin A and coin B.

If updated p and q didn’t changed, stop iteration.

|  |  |  |
| --- | --- | --- |
|  | Before | after |
| P | 0.83 | 0.84 |
| Q | 0.55 | 0.52 |

**P,Q is updated, so do iteration again.**

**[Iteration: 5]**

Start with (p=0.84, q=0.52).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | P(A | S) | P(B | S) | Count(H) | Count(T) |
| Case 1 | 0.05 | 0.95 | 5 | 5 |
| Case 2 | 0.98 | 0.02 | 9 | 1 |
| Case 3 | 0.89 | 0.11 | 8 | 2 |

Then, we can get expected number of heads of each cases as P(A|S)\*count(H). Do same with tail of each cases.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | EN of H of coinA | EN of T of coinA | EN of H of coinB | EN of T of coinB |
| Case 1 | 0.21 | 0.21 | 4.78 | 4.78 |
| Case 2 | 8.81 | 0.98 | 0.18 | 0.02 |
| Case 3 | 7.14 | 1.78 | 0.85 | 0.21 |
| Sum of all Case | 16.17 | 2.98 | 5.82 | 5.02 |

We can update p and q as sum(EN\_H) / (sum(EN\_H)+sum(EN\_H)) of coin A and coin B.

If updated p and q didn’t changed, stop iteration.

|  |  |  |
| --- | --- | --- |
|  | Before | after |
| P | 0.84 | 0.84 |
| Q | 0.52 | 0.53 |

**P,Q is updated, so do iteration again.**

**[Iteration: 6]**

Start with (p=0.84, q=0.53).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | P(A | S) | P(B | S) | Count(H) | Count(T) |
| Case 1 | 0.05 | 0.95 | 5 | 5 |
| Case 2 | 0.98 | 0.02 | 9 | 1 |
| Case 3 | 0.90 | 0.01 | 8 | 2 |

Then, we can get expected number of heads of each cases as P(A|S)\*count(H). Do same with tail of each cases.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | EN of H of coinA | EN of T of coinA | EN of H of coinB | EN of T of coinB |
| Case 1 | 0.21 | 0.21 | 4.78 | 4.78 |
| Case 2 | 8.84 | 0.98 | 0.15 | 0.01 |
| Case 3 | 7.23 | 1.80 | 0.76 | 0.19 |
| Sum of all Case | 16.29 | 0.98 | 5.70 | 4.99 |

We can update p and q as sum(EN\_H) / (sum(EN\_H)+sum(EN\_H)) of coin A and coin B.

If updated p and q didn’t changed, stop iteration.

|  |  |  |
| --- | --- | --- |
|  | Before | after |
| P | 0.84 | 0.84 |
| Q | 0.53 | 0.53 |

**P,Q is converged, so end Iteration.(admit error range<0.01)**

**Final value is p=0.84 and q=0.53, with 6 iterations.**