Lab 5

Name: Yuvraj Singh
Roll No: 23072021
M.Tech CSE

Problem: Suppose you have a sample space representing the number of heads obtained when tossing n fair coins and biased coins. For the biased coin, the probability of getting a head p(head)) is 0.3. Compute the probability distributions for both the fair and biased coins regarding the number of heads, and then calculate the statistical distance between these two distributions.

Solution:
Let X represent the number of heads obtained when tossing n coins. The probability mass function (PMF) for X can be represented by the binomial distribution:

$$P(X = k) = \binom{n}{k} \cdot p^k \cdot (1 - p)^{n-k}$$

Where:

- n is the number of coin tosses,
- k is the number of heads obtained,
- p is the probability of getting a head.

For the fair coin:

$$p_{\text{fair}} = 0.5$$

For the biased coin:

$$p_{\text{biased}} = 0.3$$

Now, the probability distributions for the fair coin:

$$P_{\text{fair}}(X = k) = \binom{n}{k} \cdot 0.5^k \cdot 0.5^{n-k}$$

The probability distributions for the biased coin:

$$P_{\text{biased}}(X = k) = \binom{n}{k} \cdot 0.3^k \cdot 0.7^{n-k}$$

Now, the statistical distance between two probability distributions P and Q is defined as:

$$\Delta(P, Q) = \frac{1}{2} \sum_{x} |P(x) - Q(x)|$$

Where x ranges over all possible values.

Let's calculate the statistical distance between the fair and biased coin distributions.

$$\Delta(P_{\text{fair}}, P_{\text{biased}}) = \frac{1}{2} \sum_{k=0}^{n} |P_{\text{fair}}(X = k) - P_{\text{biased}}(X = k)|$$

$$\Delta(P_{\text{fair}}, P_{\text{biased}}) = \frac{1}{2}\sum_{k=0}^{n}\left|\binom{n}{k}\cdot 0.5^k \cdot 0.5^{n-k} - \binom{n}{k}\cdot 0.3^k \cdot 0.7^{n-k}\right|$$

For $\mathbf{n = 5}$

Fair Coin Probability Distribution:

$$P_{\text{fair}}(0 \text{ heads}) = 0.03125$$
$$P_{\text{fair}}(1 \text{ heads}) = 0.15625$$
$$P_{\text{fair}}(2 \text{ heads}) = 0.3125$$
$$P_{\text{fair}}(3 \text{ heads}) = 0.3125$$
$$P_{\text{fair}}(4 \text{ heads}) = 0.15625$$
$$P_{\text{fair}}(5 \text{ heads}) = 0.03125$$

Biased Coin Probability Distribution:

$$P_{\text{biased}}(0 \text{ heads}) = 0.16807$$
$$P_{\text{biased}}(1 \text{ heads}) = 0.36015$$
$$P_{\text{biased}}(2 \text{ heads}) = 0.3087$$
$$P_{\text{biased}}(3 \text{ heads}) = 0.1323$$
$$P_{\text{biased}}(4 \text{ heads}) = 0.02835$$
$$P_{\text{biased}}(5 \text{ heads}) = 0.00243$$

Statistical Distance between Distributions: 0.34072

For $\mathbf{n = 10}$

Fair Coin Probability Distribution:

$$P(0 \text{ heads}) = 0.0078125$$
$$P(1 \text{ heads}) = 0.0546875$$
$$P(2 \text{ heads}) = 0.164062$$
$$P(3 \text{ heads}) = 0.273438$$
$$P(4 \text{ heads}) = 0.273438$$
$$P(5 \text{ heads}) = 0.164062$$
$$P(6 \text{ heads}) = 0.0546875$$
$$P(7 \text{ heads}) = 0.0078125$$

Biased Coin Probability Distribution:

$$P(0 \text{ heads}) = 0.0823543$$
$$P(1 \text{ heads}) = 0.247063$$
$$P(2 \text{ heads}) = 0.317652$$
$$P(3 \text{ heads}) = 0.226894$$
$$P(4 \text{ heads}) = 0.0972405$$
$$P(5 \text{ heads}) = 0.0250047$$
$$P(6 \text{ heads}) = 0.0035721$$
$$P(7 \text{ heads}) = 0.0002187$$

Statistical Distance between Distributions: 0.420507

Implementation in C++

```cpp
#include <bits/stdc++.h>
using namespace std;

double pow_(double base, long long exponent) {
    double result = 1;

    while (exponent > 0) {
        if (exponent & 1)
            result = result * base;

        exponent >>= 1;
        base = base * base;
    }

    return result;
}

// Function to calculate binomial coefficient (n choose k)
long long binCoeff(long long n, long long k) {
   long long C[k+1];
   memset(C, 0, sizeof(C));
   C[0] = 1;
   for (long long i = 1; i <= n; i++) {
      for (int j = min(i, k); j > 0; j--)
         C[j] = C[j] + C[j-1];
   }
   return C[k];
}


double fairCoinProbability(long long n, long long k) {
    double p_head = 0.5;
    return binCoeff(n, k) * pow_(p_head, k) * pow_(1 - p_head, n - k);
}

double biasedCoinProbability(long long n, long long k) {
    double p_head_bias = 0.3;
    return binCoeff(n, k) * pow_(p_head_bias, k) * pow_(1-p_head_bias, n-k);
}

double statisticalDistance(vector<double>& dist1, vector<double>& dist2) {
    double distance = 0.0;
    for (size_t i = 0; i < dist1.size(); ++i) {
        distance += 0.5 * fabs(dist1[i] - dist2[i]);
    }
    return distance;
}
```

```cpp
int main() {
    long long n = 7; // Number of coin tosses

    // Calculate probability distributions for fair and biased coins
    vector<double> fairDistrib(n + 1, 0.0);
    vector<double> biasedDistrib(n + 1, 0.0);

    for (long long k = 0; k <= n; ++k) {
        fairDistrib[k] = fairCoinProbability(n, k);
        biasedDistrib[k] = biasedCoinProbability(n, k);
    }

    // Output probability distributions
    cout << "Fair Coin Probability Distribution:\n";
    for (long long k = 0; k <= n; ++k)
        cout << "P(" << k << " heads) = " << fairDistrib[k] << "\n";

    cout << "\nBiased Coin Probability Distribution:\n";
    for (long long k = 0; k <= n; ++k)
        cout << "P(" << k << " heads) = " << biasedDistrib[k] << "\n";

    // Calculate and output the statistical distance
    double distance = statisticalDistance(fairDistrib, biasedDistrib);
    cout << "\nStatistical Distance between Distributions: " << distance;

    return 0;
}
```