

**Instituto Politécnico Nacional**

*Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas*

Teoría de la información

**Tarea**

Información mutua

**Profesor**

Jorge Rojas Beltrán

**Grupo**

2TM4

**Alumno**

Alvarado Balbuena Jorge Anselmo

2019/03/19

**Código**

public static void Main()

{

double[,] matrizUno = {

{ 1.0/2, 1.0/3, 1.0/6 },

{ 1.0/6, 1.0/2, 1.0/3 },

{ 1.0/3, 1.0/6, 1.0/2 }

};

double[,] matrizDos = {

{ 1.0/3, 1.0/6, 1.0/2 },

{ 1.0/6, 1.0/2, 1.0/3 },

{ 1.0/2, 1.0/6, 1.0/6 }

};

double[,] matrizTres = {

{ 1.0/3, 1.0/3, 1.0/3 },

{ 1.0/2, 1.0/2, 0.0 },

{ 1.0/2, 1.0/3, 1.0/6 }

};

double[] probInicial = { 1.0 / 3, 1.0 / 3, 1.0 / 3 };

#region Informacion mutua

var infoMutuaUno = CalcularInformacionMutua(matrizUno, probInicial);

var infoMutuaDos = CalcularInformacionMutua(matrizDos, probInicial);

var infoMutuaTres = CalcularInformacionMutua(matrizTres, probInicial);

#endregion

#region Capacidad de canal

var matricesIniciales = MatrizProbIniciales();

var capacidadCanalUno = CalculaCapacidadCanal(matrizUno, matricesIniciales);

var capacidadCanalDos = CalculaCapacidadCanal(matrizDos, matricesIniciales);

var capacidadCanalTres = CalculaCapacidadCanal(matrizTres, matricesIniciales);

#endregion

}

private static CapacidadCanal CalculaCapacidadCanal(double[,] matriz,

IEnumerable<double[]> matricesInic)

{

var informaciones = new Dictionary<double[], double>();

foreach (var matrizInicial in matricesInic)

{

var informacionMutuaMatriz = CalcularInformacionMutua(matriz, matrizInicial);

informaciones.Add(matrizInicial, informacionMutuaMatriz);

}

var infoMutuaMax = informaciones.Max(i => i.Value);

var probInfoMutuaMax = informaciones

.FirstOrDefault(i => i.Value == infoMutuaMax)

.Key;

return new CapacidadCanal

{

MaxInformacionMutua = infoMutuaMax,

MaxMatrizInformacionMutua = probInfoMutuaMax,

ConjuntoMatrices = informaciones

};

}

private static List<double[]> MatrizProbIniciales()

{

var probA = 0.0000;

var probB = 0.0001;

var probC = 0.9999;

var probs = new List<double[]> { new[] { probA, probB, probC } };

for (var i = 0; i < 5001; i++)

{

probA += 0.0001;

probB += 0.0001;

probC -= 0.0002;

if (probC < 0) break;

probs.Add(new[]

{

probA, probB, probC

});

}

return probs;

}

private static double CalcularInformacionMutua(double[,] matriz, double[] probInicial)

{

double informacionMutua = 0;

for (var i = 0; i < matriz.GetLength(0); i++)

{

for (var j = 0; j < matriz.GetLength(1); j++)

{

var valorSumaInterior = probInicial.Select((t, k) => t \* matriz[k, j]).Sum();

if (matriz[i, j] == 0.0)

informacionMutua += 0.0;

else

informacionMutua += probInicial[i] \* matriz[i, j]

\* Math.Log(matriz[i, j] / valorSumaInterior, 2);

}

}

return informacionMutua;

}

}

internal class CapacidadCanal

{

public double MaxInformacionMutua { get; set; }

public double[] MaxMatrizInformacionMutua { get; set; }

public Dictionary<double[], double> ConjuntoMatrices { get; set; }

}

**Resultados**

*Matrices*

*Probabilidades iniciales*

*Información mutua*

|  |  |
| --- | --- |
|  | Información mutua [bits/símbolo] |
| Matriz uno | 0.12581458369391135 |
| Matriz dos | 0.14332414086996514 |
| Matriz tres | 0.13264530905025107 |

Tabla 1 Resultados información mutua.

*Capacidad de canal*

|  |  |
| --- | --- |
|  | Capacidad de canal [bits/símbolo] |
| Matriz uno | 0.12581458249162258 |
| Matriz dos | 0.14398203387492878 |
| Matriz tres | 0.19086505226137321 |

Tabla 2 Capacidad de canal.

*Vector de probabilidades cuando sucede el máximo de información mutua*

* Matriz uno
* Matriz dos
* Matriz tres

**Conclusiones**

Al inicio cuando se calcula la información mutua para las matrices de estudio, da un valor que únicamente proporciona información acerca de la calidad del canal. Pero cuando se trabaja en la tarea de realizar el cálculo con diferentes probabilidades para la obtención de la información mutua y posteriormente de entre estos valores elegir el máximo para conocer la capacidad de canal, se observa que cuando los símbolos de entrada son bastante aproximados a valores equiprobables, se obtiene el mayor valor de información mutua y por lo tanto la capacidad de canal.

Cabe resaltar que la observación puede ser no del todo correcta debido al caso de la tercera matriz. Esta matriz cuenta con una probabilidad condicional de cero, lo cual puede tener relevancia en el resultado de la capacidad de canal.