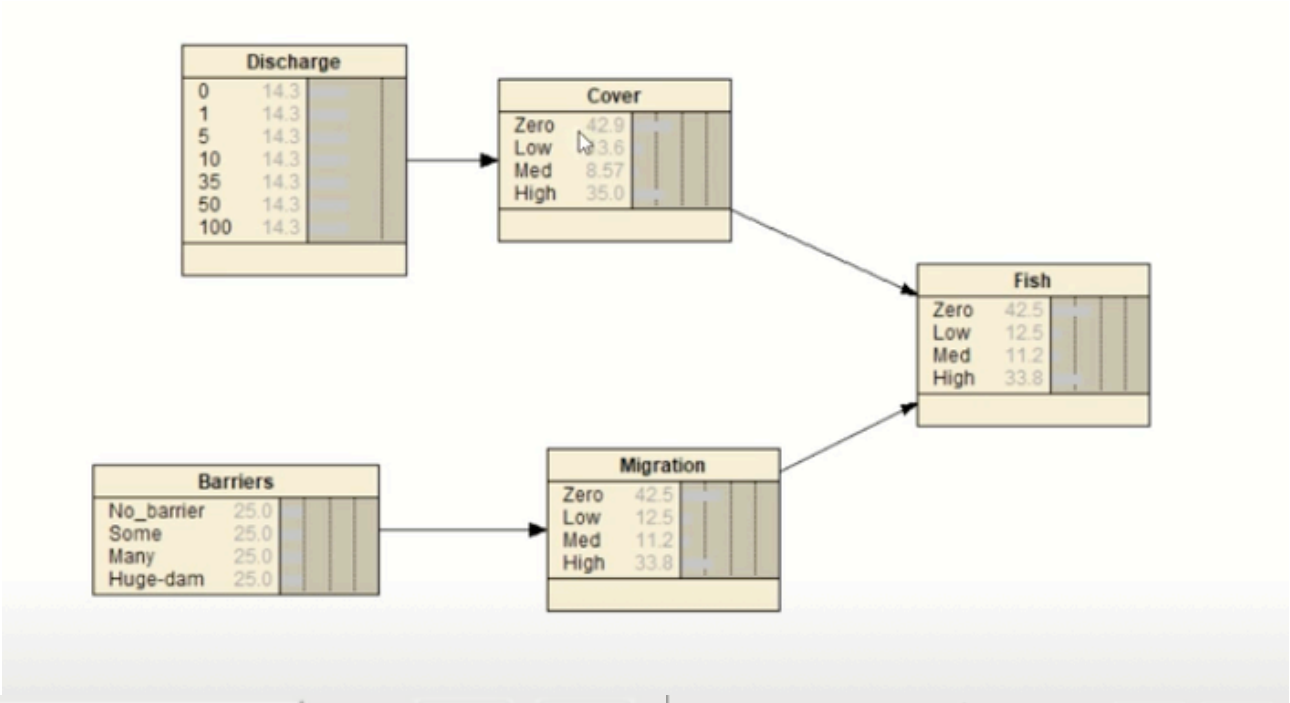
Report 14/06/23

Implementation of the Bayesian Network by Prof. Gordon Example.



Node: Migration | Apply | OK | Chance | % Probability | Reset | Close

Barriers	Zero	Low	Med	High
No_barrier	95	5	0	0
Some	50	20	20	10
Many	25	25	25	25
Huge-dam	0	0	0	100

Node: Fish | Apply | OK | Chance | % Probability | Reset | Close

Cover	Migration	Zero	Low	Med	High
Zero	Zero	100	0	0	0
Zero	Low	80	20	0	0
Zero	Med	60	20	15	5
Zero	High	0	0	50	50
Low	Zero	80	20	0	0
Low	Low	25	50	25	0
Low	Med	0	50	50	0
Low	High	0	0	25	75
Med	Zero	25	50	25	0
Med	Low	0	50	50	0
Med	Med	0	25	50	25
Med	High	0	0	5	95
High	Zero	0	0	50	50
High	Low	0	0	25	75
High	Med	0	0	5	95
High	High	0	0	0	100

Node: Cover | Apply | OK | Chance | % Probability | Reset | Close

Discharge	Zero	Low	Med	High
0	0	0	0	100
1	0	1	4	95
5	5	20	25	50
10	25	50	25	0
35	75	20	5	0
50	95	4	1	0
100	100	0	0	0

Code Implementation

```
import norsys.netica.*;
import norsys.neticaEx.aliases.Node;

public class Demo {

    public static void main(String[] args) {
        try {


            System.out.println("\nWelcome to Netica-J!\n");

            Node.setConstructorClass("norsys.neticaEx.aliases.Node");
            new Environ(null);

            Net net = new Net();

            //Sets the name of the node.         states of the node
            Node discharge = new Node("discharge", "zero,low,medium,high", net);
            Node cover = new Node("cover", "zero,low,medium,high", net);
            Node fish = new Node("fish", "zero,low,medium,high", net);
            Node migration = new Node("migration", "zero,low,medium,high", net);
            Node barriers = new Node("barriers", "none,some,many,huge_dam", net);

            //add links to the nodes
            cover.addLink(discharge);
            migration.addLink(barriers);
            fish.addLink(migration);
            fish.addLink(cover);

            //sets the CP table for the node cover
            cover.setCPTable("zero", new float[]{1.0f, 0.0f, 0.0f, 0.0f});
            cover.setCPTable("low", new float[]{0.25f, 0.50f, 0.25f, 0.0f});
            cover.setCPTable("medium", new float[]{0.0f, 0.04f, 0.01f, 0.95f});
            cover.setCPTable("high", new float[]{0.0f, 0.0f, 0.0f, 1.0f});

            //sets the CP table for the node migration
            migration.setCPTable("none", new float[]{0.95f, 0.05f, 0.0f, 0.0f});
            migration.setCPTable("some", new float[]{0.50f, 0.20f, 0.20f, 0.10f});
            migration.setCPTable("many", new float[]{0.25f, 0.25f, 0.25f, 0.25f});
            migration.setCPTable("huge_dam", new float[]{0.0f, 0.0f, 0.0f, 1.0f});
```

```
      // Sets the CP table for the node fish
      fish.setCPTable("zero,zero", new float[]{1.0f, 0.0f, 0.0f, 0.0f});
      fish.setCPTable("zero,low", new float[]{0.80f, 0.20f, 0.0f, 0.0f});
      fish.setCPTable("zero,medium", new float[]{0.60f, 0.20f, 0.15f, 0.05f});
      fish.setCPTable("zero,high", new float[]{0.0f, 0.0f, 0.50f, 0.50f});

      fish.setCPTable("low,zero", new float[]{0.80f, 0.20f, 0.0f, 0.0f});
      fish.setCPTable("low,low", new float[]{0.25f, 0.50f, 0.25f, 0.0f});
      fish.setCPTable("low,medium", new float[]{0.0f, 0.50f, 0.50f, 0.0f});
      fish.setCPTable("low,high", new float[]{0.0f, 0.0f, 0.25f, 0.75f});

      fish.setCPTable("medium,zero", new float[]{0.25f, 0.50f, 0.25f, 0.0f});
      fish.setCPTable("medium,low", new float[]{0.0f, 0.50f, 0.50f, 0.0f});
      fish.setCPTable("medium,medium", new float[]{0.0f, 0.25f, 0.50f, 0.25f});
      fish.setCPTable("medium,high", new float[]{0.0f, 0.0f, 0.05f, 0.95f});

      fish.setCPTable("high,zero", new float[]{0.0f, 0.0f, 0.50f, 0.50f});
      fish.setCPTable("high,low", new float[]{0.0f, 0.0f, 0.25f, 0.75f});
      fish.setCPTable("high,medium", new float[]{0.0f, 0.0f, 0.05f, 0.95f});
      fish.setCPTable("high,high", new float[]{0.0f, 0.0f, 0.0f, 1.0f});


      net.compile();



      //calculates the belief (probability) of the state "Zero" for the migration
      double migrationZero = migration.getBelief("zero");
      System.out.println("The probability of fish migration is when there is no barriers is " + migrationZero);
      //calculates the belief (probability) of the state "Zero" for the cover
      double coverZero = cover.getBelief("zero");
      System.out.println("The probability of fish being covered is when the discharge is zero " + coverZero);
      //calculates the belief (probability) of the state "Zerp" for the fish wellbeing
      double beliefZero = fish.getBelief("zero");
      System.out.println("Fish wellbeing when the state is zero: " + beliefZero);


      //calculates the belief (probability) of the state "low" for the migration
      double migrationLow = migration.getBelief("low");
      System.out.println("\nThe probability of fish migration is when there is some barriers is " + migrationLow);
      //calculates the belief (probability) of the state "low" for the cover
      double coverLow = cover.getBelief("low");
      System.out.println("The probability of fish being covered is when the discharge is low " + coverLow);
      //calculates the belief (probability) of the state "low" for the fish wellbeing
      double beliefLow = fish.getBelief("low");
      System.out.println("Fish wellbeing when the state is low: " + beliefLow);


      //calculates the belief (probability) of the state "medium" for the migration
      double migrationMed = migration.getBelief("medium");
      System.out.println("\nThe probability of fish migration is when there is many barriers is " + migrationMed);
      //calculates the belief (probability) of the state "medium" for the cover
      double coverMed = cover.getBelief("medium");
      System.out.println("The probability of fish being covered is when the discharge is low " + coverMed);
      //calculates the belief (probability) of the state "medium" for the fish wellbeing
      double beliefMed = fish.getBelief("medium");
      System.out.println("Fish wellbeing when the state is medium: " + beliefMed);

      //calculates the belief (probability) of the state "high" for the migration
      double migrationHigh = migration.getBelief("high");
      System.out.println("\nThe probability of fish migration is when there is huge barriers is " + migrationHigh);
      //calculates the belief (probability) of the state "high" for the cover
      double coverHigh = cover.getBelief("high");
      System.out.println("The probability of fish being covered is when the discharge is low " + coverHigh);
      //calculates the belief (probability) of the state "high" for the fish wellbeing
      double beliefHigh = fish.getBelief("high");
      System.out.println("Fish wellbeing when the state is high: " + beliefHigh);



        net.finalize();
      } catch (Exception e) {
        e.printStackTrace();
      }
    }
}
```

Results/Output

```
Welcome to Netica-J!

The probability of fish migration is when there is no barriers is 0.42500001192092896
The probability of fish being covered is when the discharge is zero 0.3125
Fish wellbeing when the state is zero: 0.2395453155040741

The probability of fish migration is when there is some barriers is 0.125
The probability of fish being covered is when the discharge is low 0.13500000536441803
Fish wellbeing when the state is low: 0.06431250274181366

The probability of fish migration is when there is many barriers is 0.11249999701976776
The probability of fish being covered is when the discharge is low 0.06499999761581421
Fish wellbeing when the state is medium: 0.21925625205039978

The probability of fish migration is when there is huge barriers is 0.3375000059604645
The probability of fish being covered is when the discharge is low 0.48750001192092896
Fish wellbeing when the state is high: 0.47688594460487366
```