

Bayesia Engine API

Overview

Developers can access many of BayesiaLab's functions outside the graphical user interface by using Bayesia's Modeling and Inference Engines. You can thus leverage Bayesian networks in your own applications and deploy them for external access, e.g. by clients, without requiring them to install their own copy of BayesiaLab.

The Bayesia Engine API is implemented as a pure Java class library (jar file), which can be easily integrated into any software project.

With the Bayesia Modeling Engine you can create your own Bayesian networks from within your code and subsequently perform inference with the Bayesia Inference Engine.

The Bayesia Inference Engine allows you to perform inference on Bayesian networks from within your own application. Networks created with BayesiaLab, or with the Modeling Engine, can both be used for computing inference with the Bayesia Inference Engine.

A typical implementation scenario would be developing a Bayesian network offline with BayesiaLab and then deploying this network for real-time prediction on streaming data with the Bayesia Inference Engine.

Modeling API

```
/* BE - Bayesia Engine - 2012 - Bayesia S.A.S
 * www.bayesia.com
 * This example is provided to show how to use the BE Modeling api.
 * To use it, the BE.jar, xercesImpl.jar and xmlParserAPIs.jar must be in the classpath.
 */
import com.bayesia.api.*;
public class ExampleModeling {
    public static void main(String[] args) {
        if (args.length != 4) {
            System.err.println("You must provide the host and port of the BayesiaLicenseServer and
a user name and password corresponding to an access to the license specified on the BayesiaLicenseServer!");
            System.exit(0);
        }
        APIModeling api = new APIModeling(args[0], Integer.parseInt(args[1]), args[2], args[3]);
        api.setComment("Simple network for lumb cancer diagnosis.");
        // Creates new nodes
        System.out.println("Creating nodes...");
        // Creates node Visit Asia
        api.addChanceLabelNode("Visit Asia", new String[] {"No", "Yes"});
        api.setNodeComment("Visit Asia", "Have you been in Asia recently?");
        // Creates node Tuberculosis
        api.addChanceLabelNode("Tuberculosis", new String[] {"No", "Yes"});
        api.setNodeComment("Tuberculosis", "Do you have tuberculosis?");
        // Creates node Age
        float[][] intervals = new float[3][2];
        intervals[0][0] = 0;
        intervals[0][1] = intervals[1][0] = 35;
        intervals[1][1] = intervals[2][0] = 60;
        intervals[2][1] = 99;
        api.addChanceIntervalNode("Age", new String[] {"<=35", "<=60", ">60"}, intervals);
        api.setNodeComment("Age", "How old are you?");
        // Long names are added to each state
        api.setStateLongNames("Age", new String[] {"Young", "Adult", "Elderly"});
        // Creates node Smoking
        api.addChanceLabelNode("Smoking", new String[] {"No", "Yes"});
        api.setNodeComment("Smoking", "Do you smoke?");
        // Creates node Bronchitis
        api.addChanceLabelNode("Bronchitis", new String[] {"No", "Yes"});
        api.setNodeComment("Bronchitis", "Do you have a bronchitis?");
        // Creates node Cancer
        api.addChanceLabelNode("Cancer", new String[] {"No", "Yes"});
        api.setNodeComment("Cancer", "Do you have a cancer?");
        // Creates node Dyspnea
        api.addChanceLabelNode("Dyspnea", new String[] {"No", "Yes"});
        api.setNodeComment("Dyspnea", "Do you have dyspnea?");
        // Creates node TbOrCa
        api.addChanceLabelNode("TbOrCa", new String[] {"False", "True"});
        api.setNodeComment("TbOrCa", "Do you have a cancer or tuberculosis?");
        // Creates node X-Ray
        api.addChanceLabelNode("X-Ray", new String[] {"Abnormal", "Normal"});
        api.setNodeComment("X-Ray", "How is your X-Ray?");
    }
}
```

```

// Adds relationships between nodes
System.out.println("Creating arcs...");
api.addArc("Visit Asia", "Tuberculosis");
api.addArc("Age", "Cancer");
api.addArc("Age", "Smoking");
api.addArc("Smoking", "Cancer");
api.addArc("Smoking", "Bronchitis");
api.addArc("Bronchitis", "Dyspnea");
api.addArc("Tuberculosis", "TbOrCa");
api.addArc("Cancer", "TbOrCa");
api.addArc("TbOrCa", "Dyspnea");
api.addArc("TbOrCa", "X-Ray");
// Creates the conditional probability tables
System.out.println("Creating conditional probability tables...");
double[] visitAsiaTable = new double[] {0.99, 0.01};
api.setConditionalProbabilityTable("Visit Asia", visitAsiaTable);
double[] ageTable = new double[] {0.35, 0.30, 0.35};
api.setConditionalProbabilityTable("Age", ageTable);
double[] tuberculosisTable = new double[] {0.99, 0.95, 0.01, 0.05};
api.setConditionalProbabilityTable("Tuberculosis", tuberculosisTable);
double[] smokingTable = new double[] {0.40, 0.50, 0.75, 0.60, 0.50, 0.25};
api.setConditionalProbabilityTable("Smoking", smokingTable);
double[] bronchitisTable = new double[] {0.70, 0.40, 0.30, 0.60};
api.setConditionalProbabilityTable("Bronchitis", bronchitisTable);
double[] cancerTable = new double[] {0.995, 0.99, 0.95, 0.985, 0.94, 0.85, 0.005, 0.01, 0.05,
0.015, 0.06, 0.15};
api.setConditionalProbabilityTable("Cancer", cancerTable);
double[] dyspneaTable = new double[] {0.90, 0.20, 0.30, 0.10, 0.10, 0.80, 0.70, 0.90};
api.setConditionalProbabilityTable("Dyspnea", dyspneaTable);
double[] xRayTable = new double[] {0.05, 0.98, 0.95, 0.02};
api.setConditionalProbabilityTable("X-Ray", xRayTable);
// Creates the conditional probability table of TbOrCa with a formula
api.setFormula("TbOrCa", "?Tuberculosis? OR ?Cancer?", false, 1000, 0, false, 0, false);
// Set no observation cost for TbOrCa
api.setNodeCost("TbOrCa", null);
// Display the nodes on a rectangular grid
api.arrangeNodePositions();
// Saves the network
api.save("AsiaAge.xml");
System.out.println("Network saved as AsiaAge.xml");
api.close();
}
}

```

Inference API

```

/* BE - Bayesia Engine - 2012 - Bayesia S.A.S
 * www.bayesia.com
 * This example is provided to show how to use the BE Inference api.
 * To use it, the BE.jar, xercesImpl.jar and xmlParserAPIs.jar must be in the classpath.
 */
import com.bayesia.api.*;
public class ExampleInference {
    public static void main(String[] args) {
        if (args.length != 4) {
            System.err.println("You must provide the host and port of the BayesiaLicenseServer and
a user name and password corresponding to an access to the license specified on the BayesiaLicenseServer!");
            System.exit(0);
        }
        APIInference api = new APIInference("Asia.xml", args[0], Integer.parseInt(args[1]), args[2],
args[3], APIInference.EXACT_INFERENCE);
        // Displays the name of the network
        System.out.println("Network: " + api.getName());
        // Displays the node names
        String[] names = api.getNodeNames();
        System.out.println("Node names:");
        printArray(names);
        System.out.println();
        // Displays the probabilities of each node
        for (int i = 0; i < names.length; i++) {

```

```

        System.out.println("Probabilities of " + names[i] + ": ");
        printArray(api.getProbabilities(names[i]));
        System.out.println();
    }
    // Observes the node Smoking to Yes
    System.out.println("The node Smoking is observed to Yes.");
    SQL
    ///Read data from ENI
    For Each Customer
        For each variable of the network
            api.observe("Variable", "Value");
            api.getProbabilities("TargetNode");

        System.out.println();
        // Tries to fix the probabilities of the node X-Ray to the given distribution whatever the
other evidence is
        System.out.println("The distrtribution of probabilities of the node X-Ray is fixed to {0.05,
0.95}.");
        api.setObservedProbabilities("X-Ray", new double[] {0.05, 0.95});
        // Displays the probabilities of the node X-Ray
        System.out.println("Probabilities of X-Ray : ");
        printArray(api.getProbabilities("X-Ray"));
        System.out.println();
        // Displays the probabilities of the node Cancer
        System.out.println("Probabilities of Cancer : ");
        printArray(api.getProbabilities("Cancer"));
        System.out.println();
        // Tries to fix the mean of the node Age to the given mean whatever the other evidence is
        System.out.println("The mean of the node Age is fixed to 60.");
        api.setObservedMean("Age", 60);
        System.out.println();
        // Observes the node Smoking to No
        System.out.println("The node Smoking is observed to No.");
        api.observe("Smoking", "No");
        System.out.println();
        // Displays the probabilities of the node X-Ray
        System.out.println("The probabilities of X-Ray remain the same: ");
        printArray(api.getProbabilities("X-Ray"));
        System.out.println();
        // Displays the mean of the node Age
        System.out.println("The mean of Age remain the same: ");
        System.out.println(api.getMean("Age"));
        System.out.println();
        // Displays the probabilities of the node Cancer
        System.out.println("And the probabilities of Cancer change: ");
        printArray(api.getProbabilities("Cancer"));
        api.close();
    }
    private static void printArray(double[] a) {
        for (int i = 0, n = a.length; i < n; i++) {
            System.out.print(a[i] + " ");
        }
        System.out.println();
    }
    private static void printArray(String[] s) {
        for (int i = 0, n = s.length; i < n; i++) {
            System.out.print(s[i] + " ");
        }
        System.out.println();
    }
}

```