The LinkStrength Package for BNT (Version 0.1)–

A Package for the Calculation and Visualization of Entropy, Link Strengths and Connection Strengths in Discrete Bayesian Networks

by Imme Ebert-Uphoff (ebert@me.gatech.edu)

November 29, 2006

The newest version of this document can always be found at www.DataOnStage.com.

Abstract

The LinkStrength package provides functions to calculate and visualize entropy, connection strengths and link strengths for discrete Bayesian Networks. The package is implemented for MATLAB's Bayes Net Toolbox (BNT).

The visualization component relies on the Graphviz package to provide the actual picture of the graph. Within the graph varying gray scales of the links indicate link strengths and varying shades of the nodes indicate connection strengths relative to a specific node, while the actual numbers are provided as the labels of the links or nodes.

The following measures are implemented:

- Entropy is used to measure the uncertainty in a single node.
- Mutual information is used to measure connection strength.
- Two measures derived from mutual information are available to measure link strength, namely *True Average Link Strength* and *Blind Average Link Strength*.
- In addition, mutual information *percentage* and link strength *percentage* are provided to measure the *percentage* of the existing uncertainty that has been removed.

This document discusses installation and use of the package, including complete mathematical expressions for all the measures.

Derivation and interpretation of the measures, however, are *not* included here. Those are discussed in great detail in [3].

Contents

1 The Difference Between Link Strength and Connection Strength **3**

2 Component Overview **4**

3 Installation Instructions **5**

4 How to Use the Package **6**
4.1 How to Make the Names of the Nodes Appear 6
4.2 How to Generate the Rendering of the Graphs 6
4.3 Interpretation of Results 6

5 Command Reference **7**
5.1 Calculating Entropy, Connection Strength and Link Strength 7
5.2 Plain Text Output Functions 9
5.3 Graph Output 10
5.3.1 Available Functions for Graph Output 10
5.3.2 Input Variables for Graph Output 11

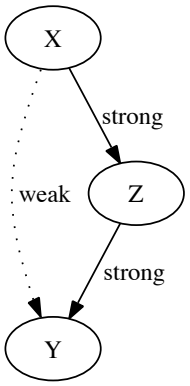
6 Closing Comments **12**

7 References **12**

1 The Difference Between Link Strength and Connection Strength

The concepts of link strength and connection strength for discrete Bayesian Networks were introduced formally by Boerlage in 1992 [1]. In [1] *connection strength* is defined to apply to any pair of nodes (adjacent or not) and measures the strength between the nodes taking any possible path between them into account. In contrast *link strength* applies to a specific edge between two adjacent nodes and measures the strength of connection only along that single edge.

To demonstrate the difference between link strength and connection strength consider the network shown in Figure 1. Each of the three nodes only has two states, $True$ and $False$. The values for $X = False$, etc., are omitted in Figure 1, since they follow immediately from the values provided.



$$
\begin{array}{ll}
\mathbf{X}: & P(X = True) = 0.5 \\
\mathbf{Z}: & P(Z = True|X = True) = 0.9 \\
 & P(Z = True|X = False) = 0.1 \\
\mathbf{Y}: & P(Y = True|X = True, Z = True) = 0.9 \\
 & P(Y = True|X = False, Z = True) = 0.89 \\
 & P(Y = True|X = True, Z = False) = 0.1 \\
 & P(Y = True|X = False, Z = False) = 0.11
\end{array}
$$

Figure 1: Sample BN with weak link from X to Y, but strong links from X to Z and from Z to Y.

Let us focus on the connection between nodes X and Y. For this sample network the *direct* link from X to Y is weak[1] , while the indirect link from X to Y through Z is very strong. According to the above (vague) concept definitions, the connection strength between X and Y is strong, but the link strength of the edge $X \rightarrow Y$ is weak:

$$
\begin{array}{lcl}
\mathrm{CS}(X, Y) & = & \text{strong}, \\
\mathrm{LS}(X \rightarrow Y) & = & \text{weak}.
\end{array}
$$

Any pair of measures for link strength and connection strength should yield this result for the considered example.

[1]This can easily be seen in the probabilities in Figure 1, because the state of X has little effect on the value of $P(Y = True|X, Z)$.

2 Component Overview

This package contains the following components:

1. **Calculation Routines** provide functions for the calculation of
   - Entropy,
   - Mutual information to measure connection strength between two nodes,
   - Conditional mutual information and derivatives thereof (True Average Link Strength, Blind Average Link Strength and their Percentages) to measure link strength of any directed edge.
2. **Printing Routines**
   - The printing routines provide a simple way to print the entropy, connection strengths and link strengths for an entire graph on the screen.
   - These functions are very simple and are meant as samples that can easily be customized by the user.
3. **Visualization Routines** generate graphs visualizing link and connection strength.
   - Just like the existing 'graph_to_dot' routine of BNT, this component generates a graph description of a discrete Bayesian Network and writes it to a file that can be read by Graphviz. The Graphviz package can then be invoked to generate the actual graph rendering.
   - In contrast to graph_to_dot, these functions include information on link strengths and connection strengths in the graph. Those are indicated by numbers (below arrows and node names) and by the gray scales used for the arrows and nodes.

3 Installation Instructions

Step 1: Download and Unpack the LinkStrength Package.
(All available at www.DataOnStage.com.)

Step 2: Place folder LinkStrength somewhere (for example FullBNT-1.0.2/BNT/LinkStrength).

Step 3: Move file 'get_key.m' into directory FullBNT-1.0.2/BNT/@assocarray

Step 4: Add the LinkStrength folder to the Matlab path *or* set the Matlab working directory to the LinkStrength folder

Step 5: If you *only* want to use the text output of the measures of the LinkStrength package, you're done. Otherwise, if not already installed, get the Graphviz package (available at http://www.graphviz.org for most platforms) which is *very* easy to install and use. (You will probably find many other uses for Graphviz, too.)

4 How to Use the Package

The easiest way to get started is to try out the sample programs ('SAMPLE_USE_SCREEN_OUTPUT' and 'SAMPLE_USE_GRAPHS') and to play around by changing commands in those files. 'SAMPLE_USE_SCREEN_OUTPUT' prints results for link strength, etc., of a sample network on the screen. 'SAMPLE_USE_GRAPHS' generates graph files. Modify the files to call your own networks. (If the node indices show up, instead of node names, see Section 4.1 below.)

For example, feel free to use your own BNets.

4.1 How to Make the Names of the Nodes Appear

In many cases you may want to use the actual names of a node, rather than their index. This is particularly true when visualizing large graphs with many nodes.

To use the node names with this package you only need to make sure that when creating your BNet in BNT using 'mk_bnet' you include the node names using the 'names' option.

For example, to make the existing 'mk_asia_bnet' function work with node names, you simply need to change the call

```
bnet = mk_bnet(dag, ns, 'discrete', dnodes);
```

to

```
bnet = mk_bnet(dag, ns, 'discrete', dnodes, 'names', ...
   {'Smoking', 'Bronchitis', 'LungCancer', 'VisitToAsia', 'Tuberculosis', ...
    'CancerOrTuberculosis', 'Dyspnoea', 'XrayPositive'});
```

4.2 How to Generate the Rendering of the Graphs

Generating the rendering of the graph is in two steps:

1. Call the desired routine in Matlab to generate a graph file (dot-file).

2. Outside of Matlab:
   Open the DOT-file from Graphviz to generate automatic layout and rendering. A single command is generally sufficient. For example, to generate a postscript image on a Unix system from the file "graph.dot" generated above one simply types in the command line:

   ```
   dot -Tps graph.dot -o graph.ps
   ```

   Using the DOT command on Windows and other platforms is just as easy. Please see the documentation at http://www.graphviz.org for more information.

For more information, see [2].

4.3 Interpretation of Results

To learn how to choose and to interpret the different measures for connection strength and link strength provided by this package, see the extensive discussion in [3].

5 Command Reference

This section provides a reference for the *syntax* of all available commands along with the formulas used for connection strength, link strength, etc. For complete information on the *derivation* and *interpretation* of the formulas, please see [3].

5.1 Calculating Entropy, Connection Strength and Link Strength

1. **Calculating Entropy**

   ```
   function entropy = calc_entropy(bnet,node,engine)
   ```

   Input variables:

   - bnet: Discrete Bayesian Network to which node belongs
   - node: Index of considered node (X)
   - engine: Optional parameter. Defines inference engine to be used for calculations.

   Output Variable: Returns the entropy of the node:

   $$U(X) = \sum_x P(x) \log_2 \left(\frac{1}{P(x)} \right),$$

   where the summation is over all discrete state of variable X.

2. **Calculating Connection Strength**

   ```
   function [MI,MI_perc] = calc_mutual_information(bnet, node1, node2, user_engine)
   ```

   Input variables:

   - bnet: Discrete Bayesian Network to which nodes belong
   - node1, node2: Indices of considered node pair.
     node1 is index of X, node2 is index of Y.
   - user_engine: Optional parameter. Defines inference engine to be used for calculations.

   Output Variables: Returns the mutual information and mutual information percentage of the node pair:

   $$\begin{aligned} MI(X,Y) &= U(Y) - U(Y|X) = U(X) - U(X|Y) \\ MI\%(X,Y) &= 100 \cdot \frac{U(Y) - U(Y|X)}{U(Y)} \end{aligned}$$

   where the definition of $U(X|Y)$ is included in the link strength section below.

3. **Calculating Link Strength**

```
function [LS,LS_perc] = calc_link_strength (bnet, arc_parent, arc_child, formula)
```

Input variables:

- bnet: Discrete Bayesian Network to which nodes belong
- arc_parent, arc_child: Indices of parent (X) and of child (Y) of directed arc.
- formula: possible values are 'TrueAverage' or 'BlindAverage'

Output Variables: Returns the link strength and link strength percentage of the directed arc according to formula TrueAverage or BlindAverage.

True Average Link Strength:

$$\begin{aligned} LS^{true}(X \to Y) &= U(Y|Z) - U(Y|X,Z), \\ LS\%^{true}(X \to Y) &= 100 \cdot \frac{U(Y|Z) - U(Y|X,Z)}{U(Y|Z)} \end{aligned}$$

where

$$\begin{aligned} U(Y|Z) &= \sum_z P(z) \sum_y P(y|z) \log_2 \frac{1}{P(y|z)} \\ U(Y|X,Z) &= \sum_{x,z} P(x,z) \sum_y P(y|x,z) \log_2 \frac{1}{P(y|x,z)} \end{aligned}$$

Blind Average Link Strength

$$\begin{aligned} LS^{blind}(X \to Y) &= \hat{U}(Y|Z) - \hat{U}(Y|X,Z), \\ LS\%^{blind}(X \to Y) &= 100 \cdot \frac{\hat{U}(Y|Z) - \hat{U}(Y|X,Z)}{\hat{U}(Y|Z)} \end{aligned}$$

where

$$\begin{aligned} \hat{U}(Y|Z) &= \frac{1}{\#(X)\#(Z)} \sum_{x,y,z} P(y|x,z) \log_2 \frac{\#(X)}{\sum_x P(y|x,z)}, \\ \hat{U}(Y|X,Z) &= \frac{1}{\#(X)\#(Z)} \sum_{x,y,z} P(y|x,z) \log_2 \frac{1}{P(y|x,z)} \end{aligned}$$

where $\#(X)$ is the number of states of variable X, etc.

5.2 Plain Text Output Functions

There are four simple routines to print graph structure, entropy, mutual information and link strength on the screen. All of them apply to an entire graph. These functions are very simple - feel free to customize them for your own use.

1. `function print_DAG_structure(bnet)`

   Prints number of nodes, node names (if available) and list of all edges on screen.

2. `function print_entropy(bnet)`

   Prints entropy values for all nodes on screen.

3. `function print_all_mutual_information(bnet)`

   Prints two values for *every node pair* of the graph: mutual information and mutual information percentage.

4. `function print_all_link_strength(bnet)`

   Prints four values for every directed edge of the graph: link strength and link strength percentage according to TrueAverage and BlindAverage formulas.

5.3 Graph Output

This section first lists the four different graph generation routines, then defines the input variables used.

5.3.1 Available Functions for Graph Output

1. **Plain Graph with Node Names**

   ```
   function graph_plain_to_dot (bnet, filename, varargin)
   ```

   Functionality: Graph showing nodes with node names, connected by arrows. This does essentially the same for Bayesian Nets as the existing routine 'graph_to_dot', but includes the names of the nodes if available.

2. **Entropy Graph for Discrete BN**

   ```
   function graph_with_entropy_to_dot (bnet, filename, varargin)
   ```

   Functionality: Creates graph including entropy for each node. The entropy is shown in the graph as a number below the node name.

3. **Graph with Connection Strengths for Discrete BN**

   ```
   function graph_with_mutual_info_to_dot (bnet, filename, target_node,
                                           is_percentage, varargin )
   ```

   Functionality: Writes the graph of a discrete bnet to a file in dot-format (Graphviz format) including Mutual Information.

   - For each node (other than target_node): Calculate mutual information relative to target_node. and write its value underneath the node name
   - For target node: Use different shape (shape2) for target node, calculate entropy value and write its value underneath node name.
   - Gray scale: Use gray scale for nodes to emphasize results (dark = strong connection to target node). Cap darkness to a maximal value if node name would otherwise be impossible to read.

4. **Graph with Link Strengths for Discrete BN**

   ```
   function graph_with_link_strength_to_dot (bnet, filename, formula,
                                             is_percentage, varargin )
   ```

   Functionality:
   Writes the graph of a discrete BNet to a file in dot-format (Graphviz format) including Link Strength.

- For each arrow in the graph: Calculate link strength according to 'formula' and 'is_percentage' and write its value underneath the arrow.
- Gray scale: Use gray scale for arrows to emphasize results (dark = strong connection). If arrow would be too light to see, use dashed line (and light gray) for arrow instead.

5.3.2 Input Variables for Graph Output

- bnet
  Meaning: Discrete Bayesian Network.
  Exception: may also be continuous or mixed for function graph_plain_to_dot.
- filename
- target_node
  Meaning: index of node relative to which to calculate mutual information
  Comment: Only required for graph with mutual information.
- is_percentage:
  if 'false' - Use Standard Mutual Information Value
  if 'true' - Use Percentage Value
  Comment: Only required for graphs with mutual information or link strength.
- formula:
  if 'TrueAverage' - Use True Average Link Strength
  if 'BlindAverage' - Use Blind Average Link Strength
  Comment: Only required for graph with link strength.

Optional Input Variables:
The following optional variables are supported by all functions.

- use_node_names
  Default: true
  Meaning: Do you want to show node NAMES or node INDICES?
- shape1
  Default: 'ellipse'
  Meaning: Name of standard node shape to be used in graph. Possible shapes and their names can be found in [2].
- shape2
  Default: 'doubleoctagon'
  Meaning: Currently only used as shape for target node of Mutual Information graph.

To provide a value for optional variables, use the format

```
'variable_name', variable_value
```

For example:

```
graph_plain_to_dot (bnet, 'plain.dot', 'use_node_name', false, 'shape1', 'octagon')
```

forces the function to use node indices in the graph (rather than node names) and to use an octagon as standard node shape (instead of an ellipse).

6 Closing Comments

If you find any bugs or have any comments or suggestions, *please* drop me an e-mail. If you have any other measures you would like to see implemented or are interested in a collaboration on a related topic, feel free to contact me. Of course, it would make my day to hear from anyone who found this package useful.

7 References

[1] Boerlage, B., "Link Strengths in Bayesian Networks", M.S. Thesis, Dept. of Computer Science, The University of British Columbia, October 1992.

[2] Gansner, E., Koutsofios, E. and North, S., "Drawing graphs with *dot*", Feb. 2002. Available at http://graphviz.org/ by clicking on "Documentation" and selecting "User's Guide: dot".

[3] Imme Ebert-Uphoff, "On Measuring Connection Strengths and Link Strengths in Discrete Bayesian Networks", Research Report EBE-BN-CSLS-Mar-06, March 24, 2006. Available at www.DataOnStage.com.