CptS 580: Advanced Topics in Machine Learning

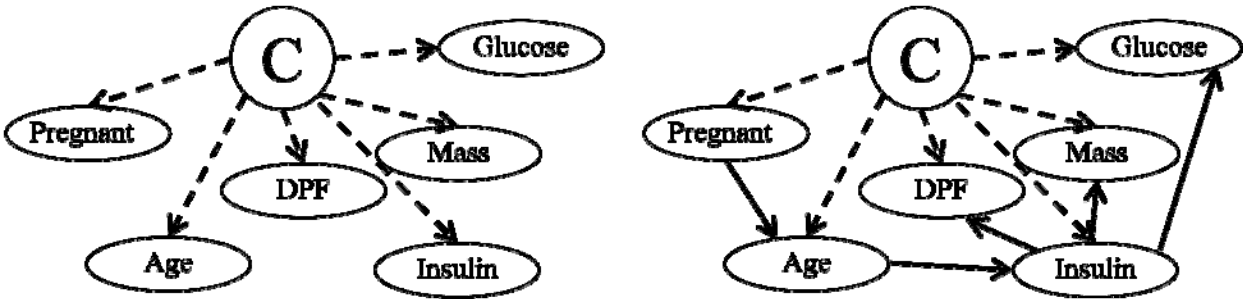
Homework #3
Due: Tuesday, November 8, 2011

Background

In many classification tasks the naïve Bayes classifier (NBC) is competitive, even though it ignores possible dependences between features. The other extreme is complete structure learning, which models the full joint distribution and does not always correspond to a good estimate of the class-conditional distribution.

A tree-augmented naïve Bayes classifier (TAN) augments the standard NBC by modeling correlations between features. The TAN has been shown to outperform naïve Bayes, but at the same time it maintains the computational simplicity of NBC because no exponential search is involved. Like the NBC, the TAN is a simple network in which each feature node is pointed to by the parent class node. Feature nodes in a TAN can have an additional parent which is one other feature node in the network. Because such dependencies are explicitly modeled, the TAN avoids "double counting" that can occur in NBC when multiple features are used that are very similar (have a high correlation value).

In an augmented structure, an edge from feature i to feature j implies that the influence of feature i on the class label also depends on the value of j. For example, the figure on the right below shows that the influence of attribute "Glucose" on class C depends on the value of "Insulin", while in the NBC on the left the influence of each attribute on C is independent of other attributes. The subset of nodes and edges from the structure which do not include the class node induce a directed tree.



Part One

The goal of part one is to implement an algorithm for learning the structure of a TAN model. You can implement TAN in Python and add it to Orange or implement it in a language of your choice as a standalone program.

Using the voting dataset, draw the structure (directed acyclic graph) that is produced using this algorithm. The algorithm is described in detail as the Chow-Liu algorithm in Friedman et al., Bayesian Network Classifiers, Machine Learning, 29:2-3:131-163, 1997. In this algorithm we let C represent the class variable and $\{X_1,..,X_n\}$ represents the features.

1. Compute CMI, the conditional mutual information between each pair of variables i≠j.

$$CMI(X,Y \mid C) = \sum_{x,y,c} p(x,y,c) \log \frac{p(x,y \mid c)}{p(x \mid c)\, p(y \mid c)}$$

The probabilities are estimated based on the number of times the values of the feature occur / co-occur in the training data, as would be estimated for a NBC or decision tree classifier.

2. Build a complete undirected graph in which nodes represent the features. Label edges by the MI value between the corresponding features.

3. Find a maximum weighted spanning tree on the graph. This algorithm should have been learned in Data Structures or Algorithmics and is available in most Algorithms text books.

4. Transform the resulting undirected tree to a directed tree T by picking an arbitrary node as the root and setting the direction of all edges to be outward from the root. I recommend using depth-first search for this step.

5. The structure of the TAN model is a naïve Bayes model augmented by the edges in T.

Part Two

In the second part of this assignment you will use your TAN structure as a classifier. To do this, you will need to compute and store conditional probability tables for each of the dependencies in your tree, the values of which are estimated from your training data.

In order to classify the data, you will need to compute $p(C|X_1,..,X_n)$. You can use Bayes rule to generate this for NBC and apply Bayes rule to TAN once the conditional probabilities are estimated. Test NBC and TAN on voting data set that was used for Homework Assignment #1 using leave one out testing repeated for each data point.

Report the accuracy results of the NBC and TAN classifiers. Comment on the results for this experiment. Do you expect a TAN will always outperform (or underperform) a NBC?

Turning in the Assignment

The assignment should be mailed as a TXT or PDF file to cook@eecs.wsu.edu by 9:00am on the due date. Include your source code, sample output, and summarized results with observations. No late assignments will be accepted.