Simple Bayesian Networks on Netezza Box

By Mieczyslaw Klopotek and Przemyslaw Biecek and Justin Lindsey
Netezza Corporation, Polish Academy of Sciences

The NZ Analytics cartridge contains a bundle of implementations of useful statistical and data mining algorithms. When installed, they can be accessed either as SQL functionality or as R functions, depending on the interface you use. Diverse implementation technologies (stored procedures, user-defined functions, user-defined aggregates) were used depending on the nature of the solved problem. The algorithms implemented or wrapped into stored procedures are accessible also via an NZ-R interface. This presentation concerns the Bayesian Network implementation.

A **Bayesian Network** is a method of representing a joint probability distribution in many variables in a compact way. The representation consists of a directed acyclic graph structure DAG with conditional probabilities of a node given its parents attached to each node, $P(X_i \mid \pi(X_i))$. We talk about a simple Bayesian Network if each node has only one parent. Though this assumption is a significant simplification, it has been found useful for problems in a large number of variables. In spite of the simplicity of the case, the efficient approach of Chow/Liu [1] is of prohibitive memory complexity (quadratic in the number of variables, so 5,000 variables is a practical limitation for 1GB memory), hence ways to overcome the memory limitations need to be sought. Though various space- and time saving improvements have been proposed [2,4], they prove to be not useful under massively parallel database systems in which data is stored record-wise, because they restrict the number of dependency computations and not the number of passes through the database which most time-consuming.

To be able to compute BNs from data restricting the number of passes through the database, a new approach, based on insights from [3], is being proposed in this paper, with the following steps:

- Step 1: Take the first N variables for which we can fit their sufficient statistics into the memory
- Step 2: Build a Chow/Liu tree form them
- Step 3: Forget the sufficient statistics except those related to edges in the obtained tree (their amount will be linear in the number of edges)
- Step 4: Take the next portion of say M variables so that the sufficient statistics of the tree edges plus the sufficient statistics for the matrix N x the number of nodes in the tree will fit into the memory
- Step 5: Apply the iterations of the algorithm IT (starting with step 3) for the M new variables.
- Step 6 If any variables are left, go back to step 3, otherwise terminate (that is apply the rest of the Chow/Liu procedure of orienting edges and computing the conditionals from original data).

A correctness proof will be provided in the paper.

References

[1] Chow, C. K., Liu, C. N.: Approximating discrete probability distributions with dependence trees, IEEE Transactions on Information Theory, IT-14, No.3, 1968, pp. 462-467

[2] M.A.Kłopotek: A New Bayesian Tree Learning Method with Reduced Time and Space Complexity. *Fundamenta Informaticae*, 49(no 4)2002, IOS Press, pp. 349-367. –

[3] M.A.Kłopotek: A New Space-Saving Bayesian Tree Construction Method for High Dimensional Data *Demonstratio Mathematica*, Vol. 35, No. 3 (2002)pp. 671-684

[4] Meila, M.: An accelerated Chow and Liu algorithm: fitting tree distributions to high-dimensional sparse data. http://citeseer.nj.nec.com/363584.html

Kod pola został zmieniony