Artificial Intelligence
CC-721

Bayesian Network Prediction Algorithms

- The *prediction problem* concerns trying to predict the value of a *target variable* from a set of other variables called the *predictors*.

- In *supervised learning* we learn the prediction function from data.

- I will discuss supervised learning using Bayesian networks.

- *Unsupervised learning* concerns trying to find hidden structure in data.

- The *clustering problem* is as follows:

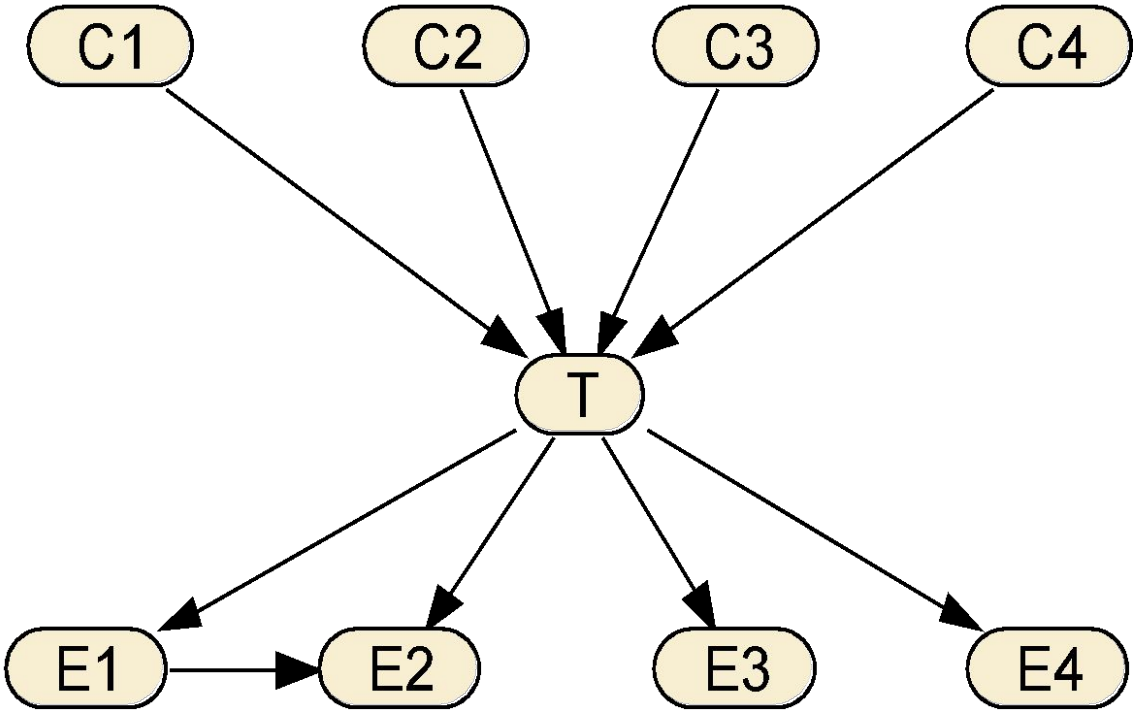
Given a collection of unclassified entities and features of those entities,

organize the entities into categories that in some sense maximize the similarity of features of entities in the same category.

Example: Cluster a bunch of animals in a pen when we know nothing about species.

Supervised Learning

When we cast supervised learning in terms of Bayesian networks, we identify causes and effects of the target.

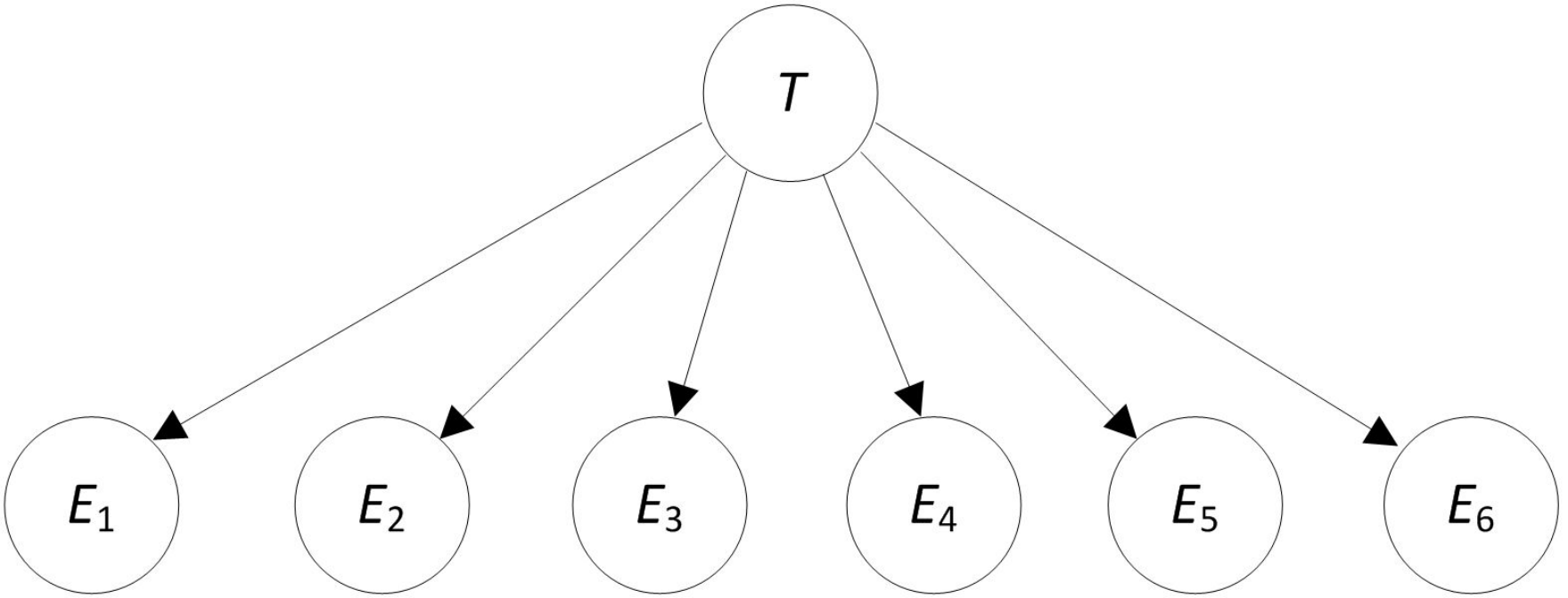


- The simple case is when all predictors are effects, and there are no arrows between the predictors.

- That is, the predictors are independent conditional on the target.

- The network representing this situation is called a
  - *naïve Bayesian network*
  - *naïve Bayesian classifier*.

A Naïve Bayesian Network



Learning a Naïve Bayesian Network

To Learn a Naïve Bayesian network from $Data$ on set of individuals do the following:

1. Learn the conditional probability distribution of each effect E_i given T from the $Data$.

2. Ascertain the prior probability of T in the population in which the classifier will be used.

Inference with a Naïve Bayesian Network

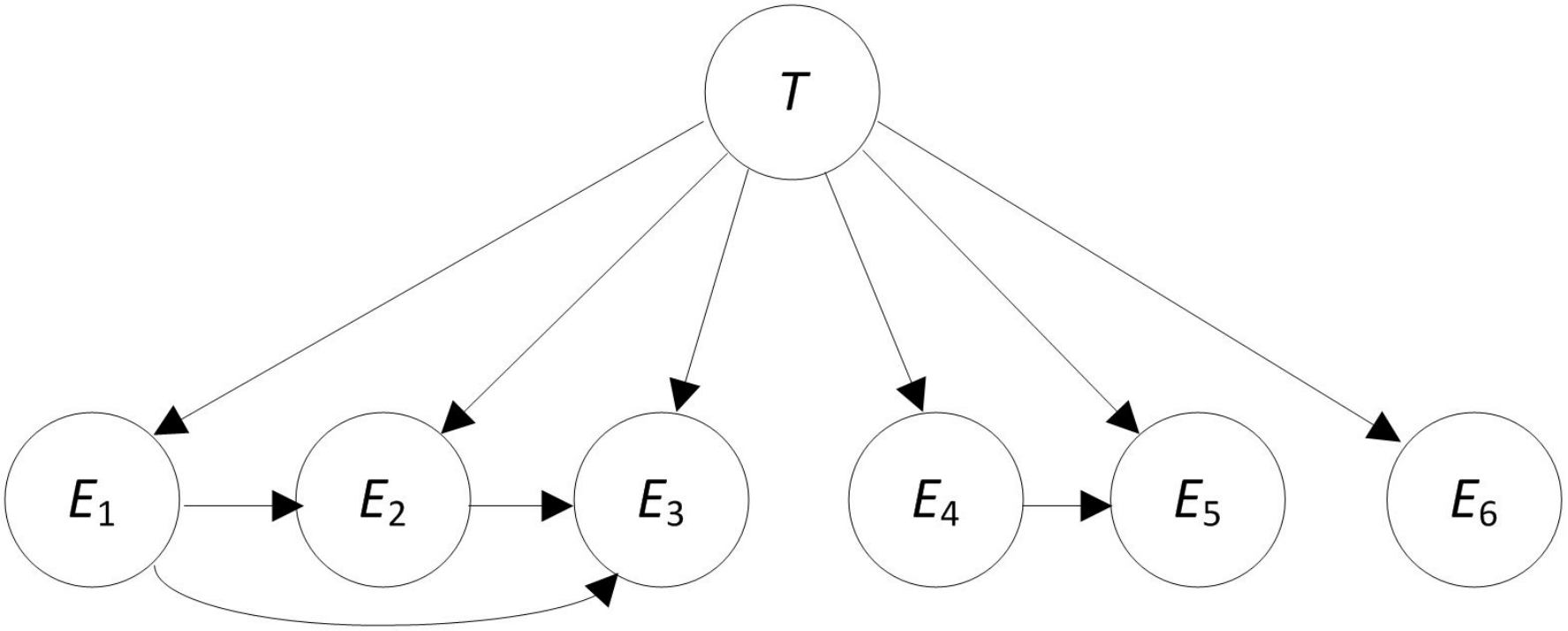
After the Bayesian network is learned, we do inference using $DataX$ on a specific individual X as follows:

$$
\begin{aligned}
P(T \mid DataX) &= KP(DataX \mid T)P(T) \\
&= KP(E_1, E_2, E_3, E_4, E_5, E_6 \mid T)P(T) \\
&= K[\prod_{i=1}^{6} P(E_i \mid T)]P(T).
\end{aligned}
$$

K is a normalizing constant

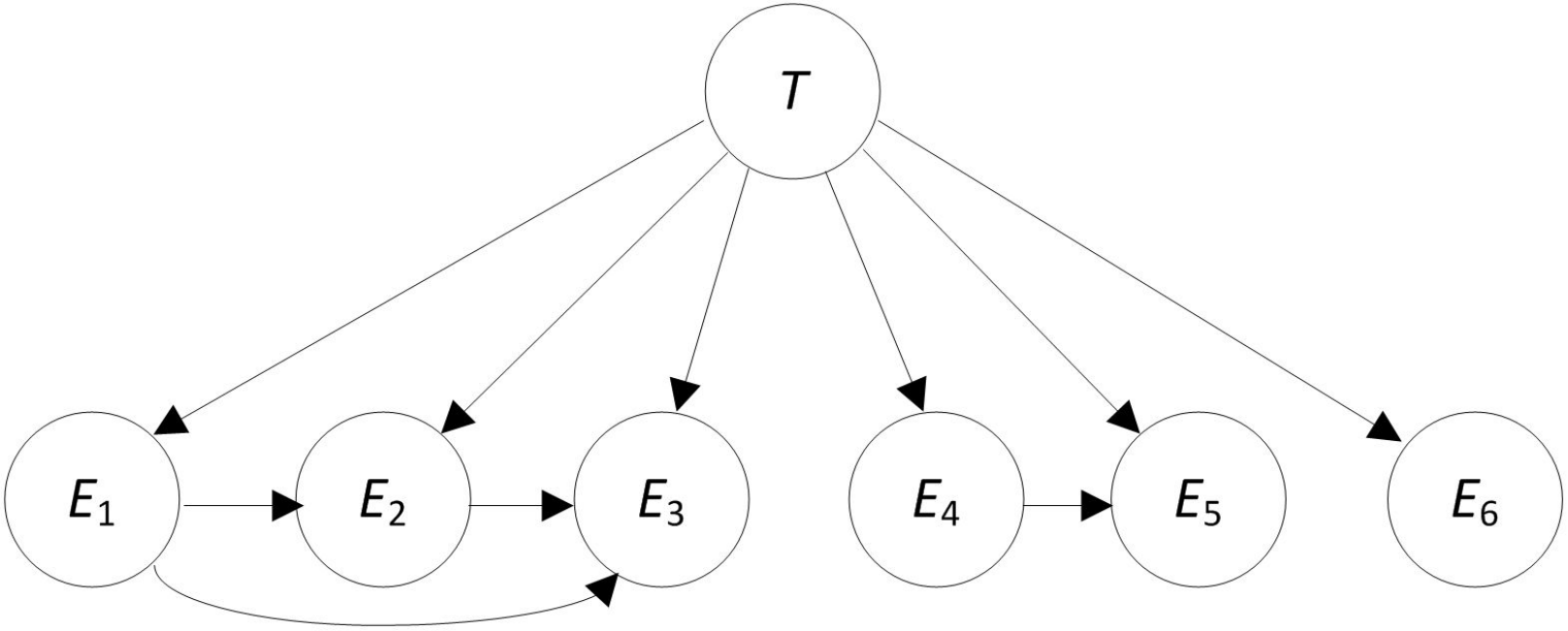
Augmented Naïve Bayesian Network

An augmented naïve Bayesian network can have arrows between children that are not independent given the target.

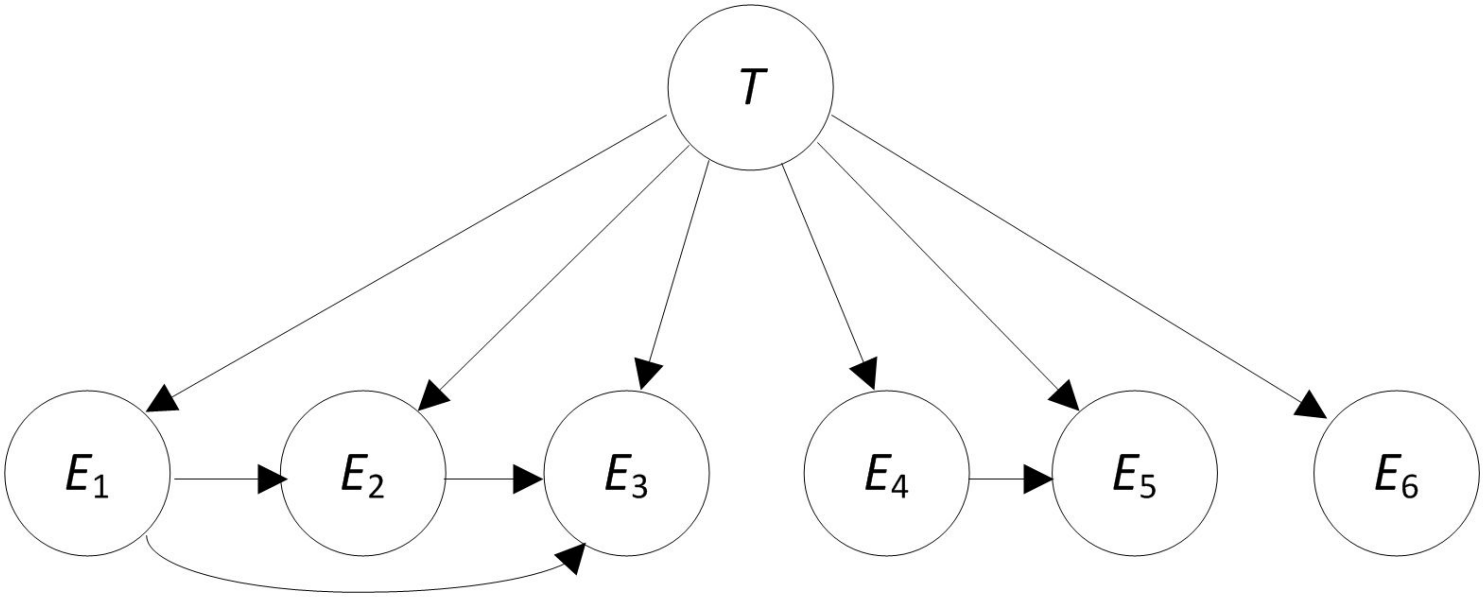


Learning an Augmented Naïve Bayesian Network

If we know which variables have edges between them, we simply learn the necessary conditional distributions from $Data$.

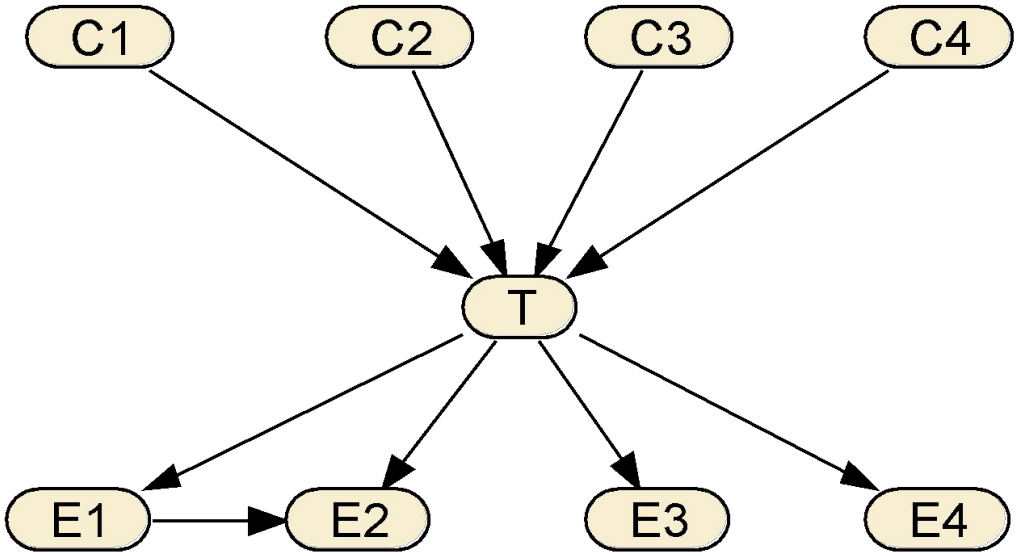


Inference with an Augmented Naïve Bayesian Network



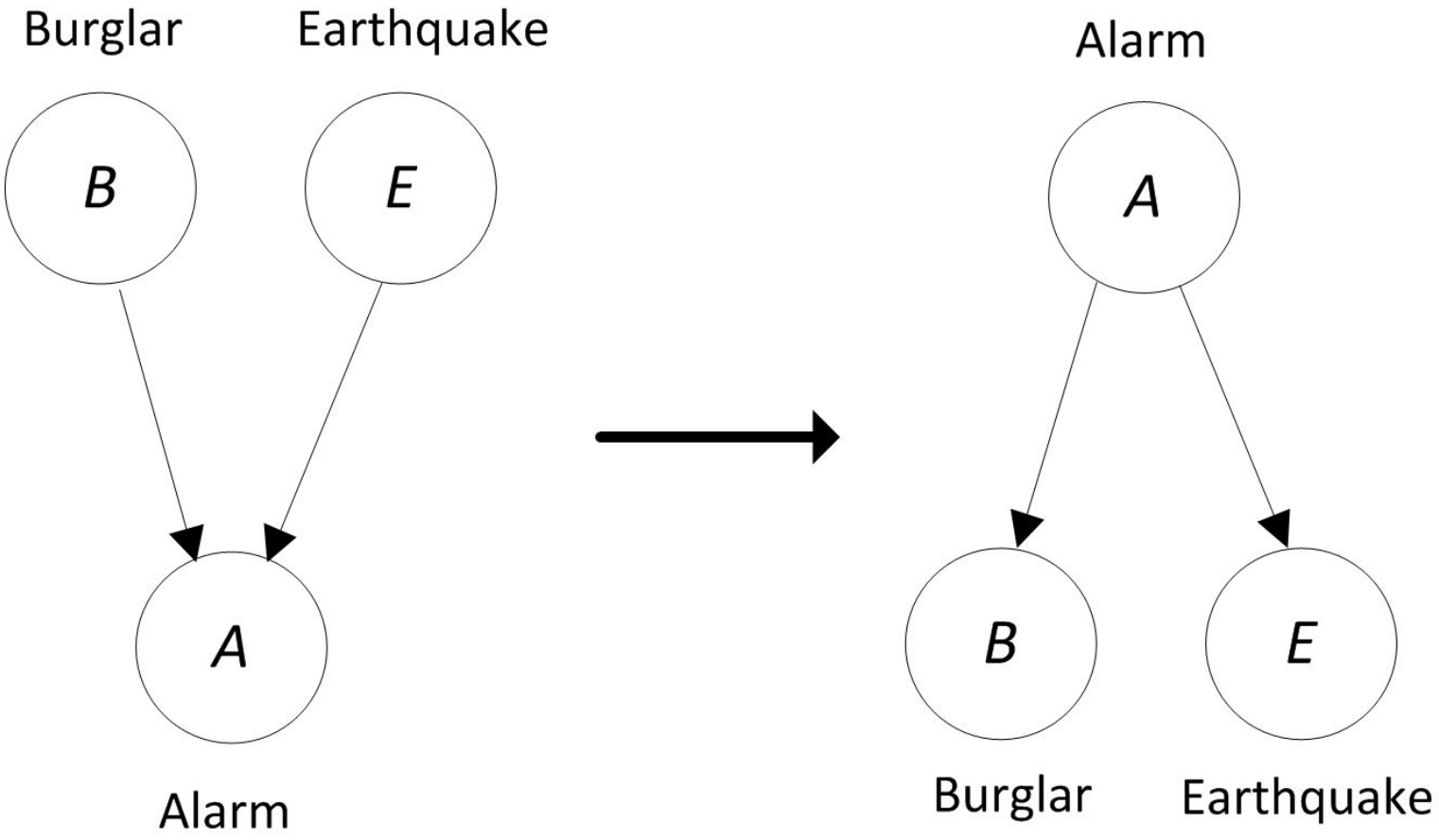
$$
\begin{aligned}
P(T \mid Data) &= KP(Data \mid T)P(T) \\
&= KP(E_1, E_2, E_3, E_4, E_5, E_6 \mid T)P(T) \\
&= KP(E_1, E_2, E_3 \mid T)P(E_4, E_5 \mid T)P(E_6 \mid T)P(T) \\
&= KP(E_3 \mid E_1, E_2, T)P(E_2 \mid E_1, T)P(E_1 \mid T)P(E_5 \mid E_4, T)P(E_4 \mid T)P(E_6 \mid T)P(T)
\end{aligned}
$$

Prediction Using Causes



- If we maintain the correct causal structure, we need a conditional probability distribution of T given every combination of values of the parents.
- If there are 20 binary parents, we need about 1 million probability distributions.

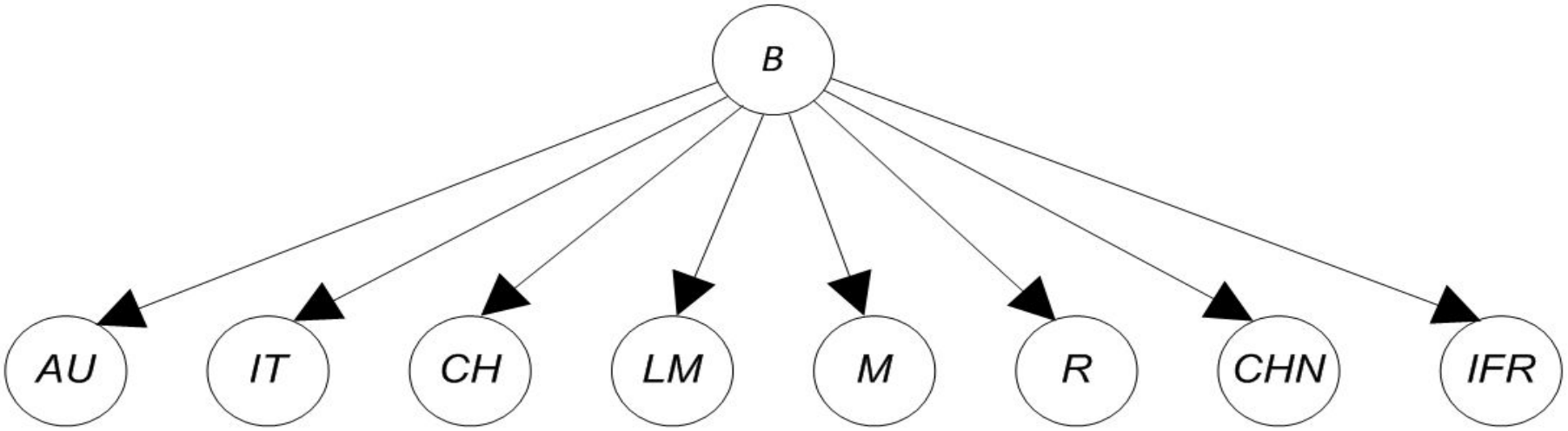
- A procedure often taken is simply to invert the causal structure.
- Make T parents of the causes.
- This violates the independence assumptions.



Even though independence assumptions are violated when we invert the causal structure,

we still often obtain good prediction performance.

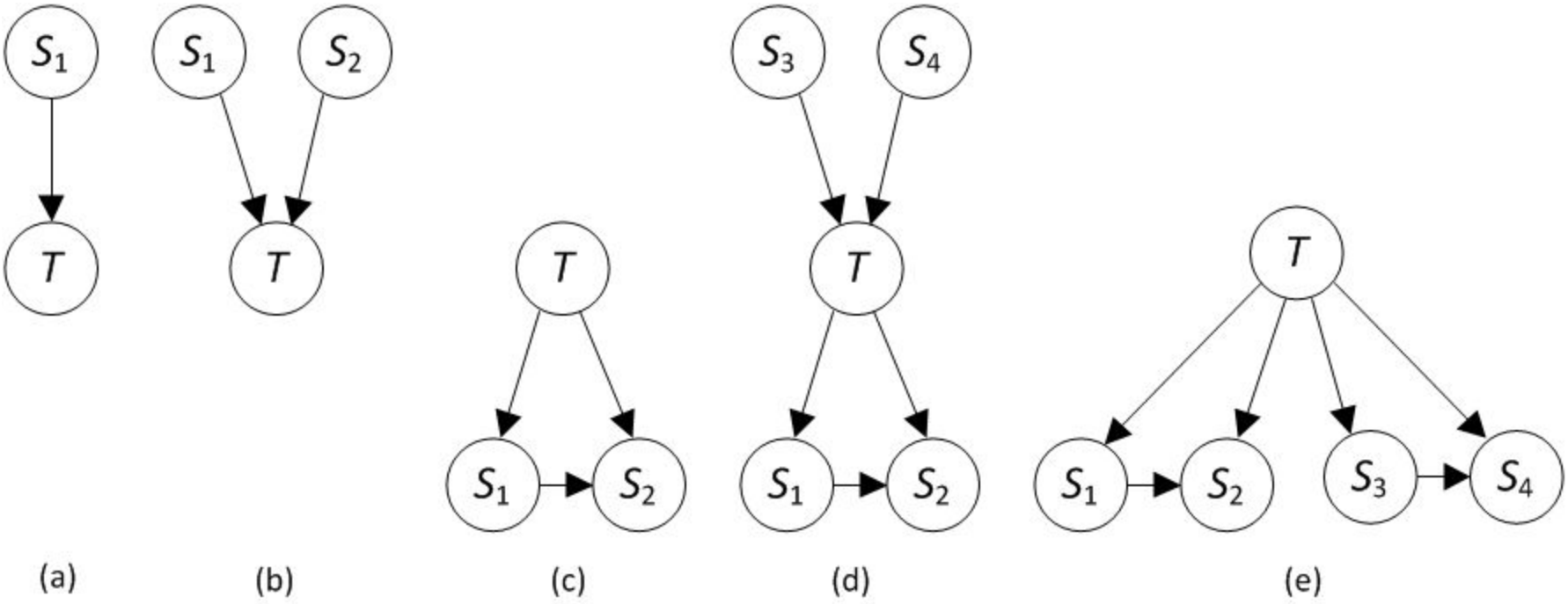
Bankruptcy Prediction [1,2]



- Variables are all causal in this problem.
- M: Firm's Size
- True positive rate: 81.12%
- True negative rate: 81.25%

- Sometimes this strategy has very poor results.
- In [3] the naïve Bayesian classifier performed the worst of all methods tested.

- *Efficient Bayesian multivariate classifier* (*EBMC*) [4] ameliorates this difficulty to some extent.
- The variables can be causes and effects.

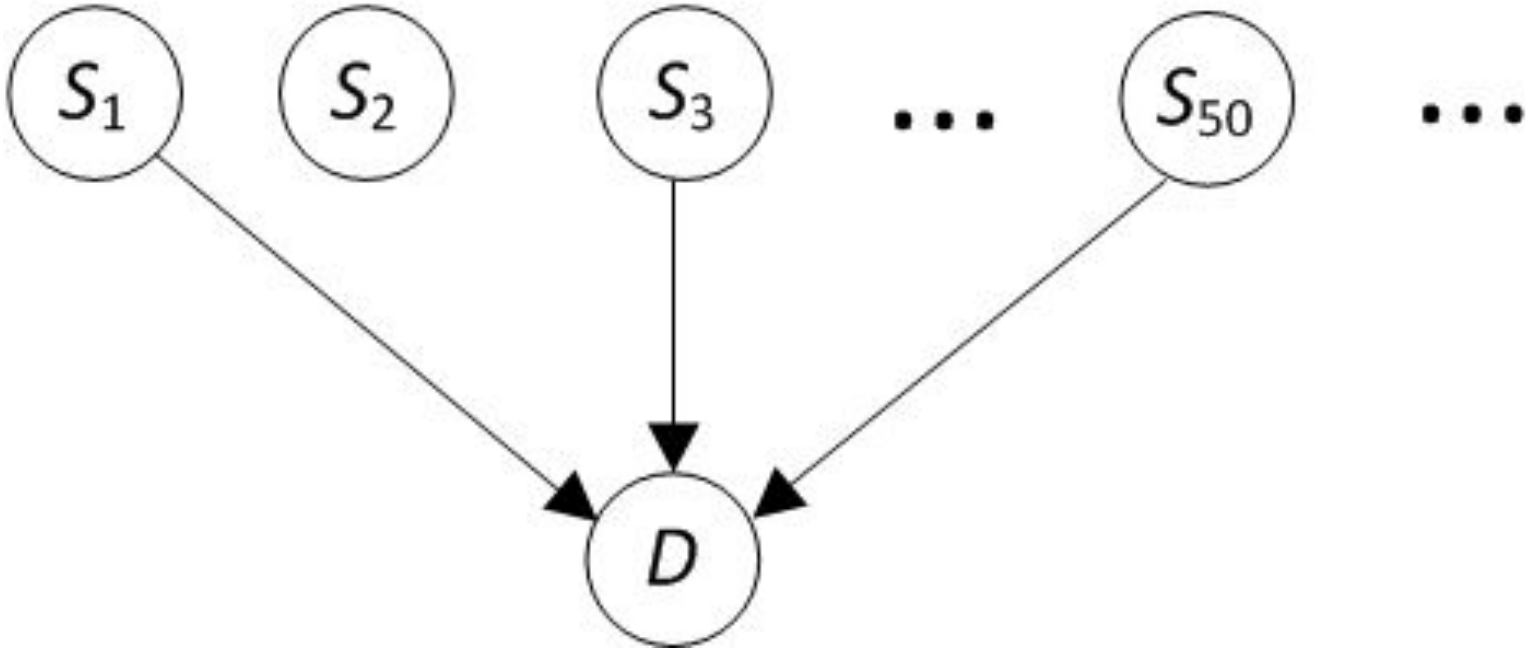


Evaluation of Methods

- Jiang et al. [5] compared 9 prediction methods using simulated and real GWAS datasets.
- First I describe GWAS datasets.
- Then I discuss the methods used in the evaluation.
- Finally, I provide the results.

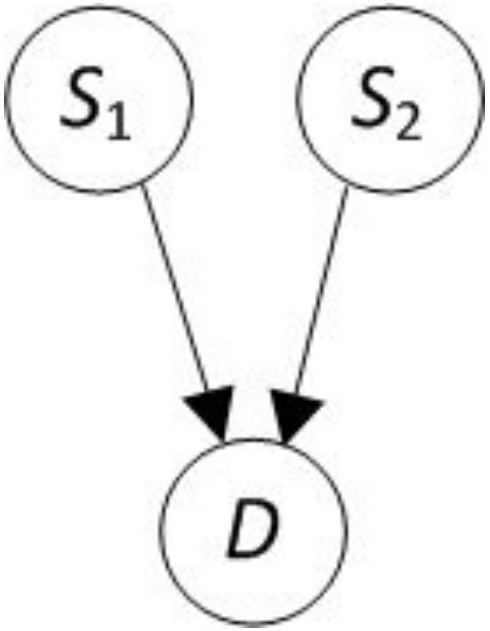
GWAS

- A *single nucleotide polymorphism* (*SNP*) results when a nucleotide that is typically present at a specific location on the genomic sequence is replaced by another nucleotide.
- There are hundreds of millions of SNPs.
- A *genome wide association study* (*GWAS*) involves sampling in a population of individuals up to millions of SNPs.
- Ordinarily, they are case-controls studies where cases have a disease and controls do not.
- Using a GWAS we try to determine the genetic basis of disease.



- We search for the causative SNPs.
- Then we use those SNPs to predict disease.

Epistasis



- Complicating the problem is epistasis.
- Epistasis is the interaction of two or more loci to affect phenotype with little or no marginal effect.

Datasets evaluated

- 100 1000 SNP datasets containing 15 causative SNPs based on 5 models of epistasis
  - 1000 cases
  - 1000 controls

- 10 10,000 SNP datasets containing those same 15 causative SNPs
  - 1000 cases
  - 1000 controls

- A real late onset Alzheimer's disease (LOAD) dataset containing 312,260 SNPs
  - 861 cases
  - 644 controls

Methods Evaluated

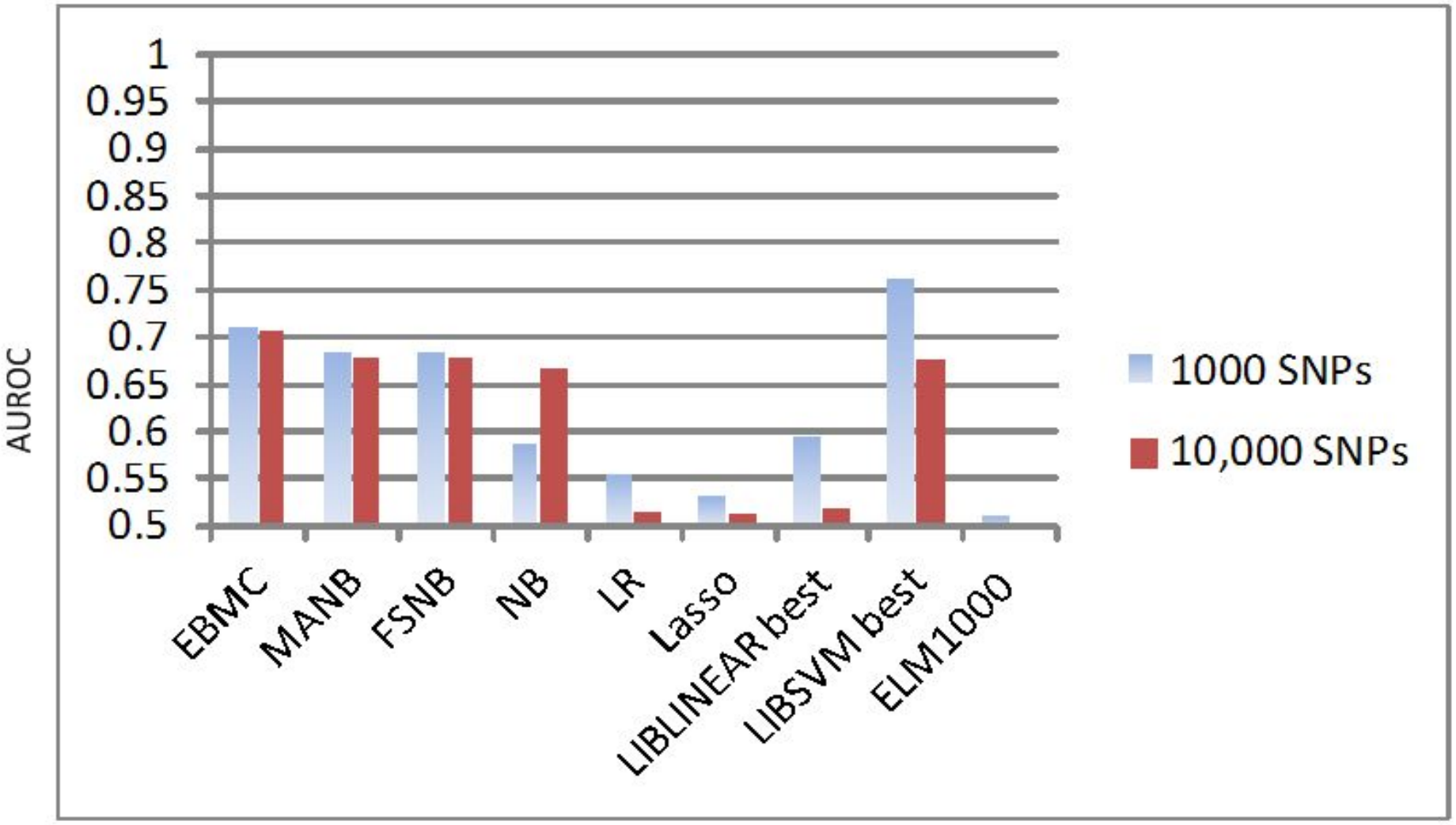
1. Naïve Bayes (NB)
2. Efficient Bayesian multivariate classifier (EBMC)
3. Feature Select Naïve Bayes (FSNB)
4. Model Averaging Naïve Bayes (MANB)
5. Logistic Regression (LR)
6. Lasso
7. Support Vector machines (SVM) with a linear kernel
8. Support Vector machines with the Radial Basis Function (RBF) kernel
9. Extreme Learning Machines (ELM) (neural networks)

Parameters

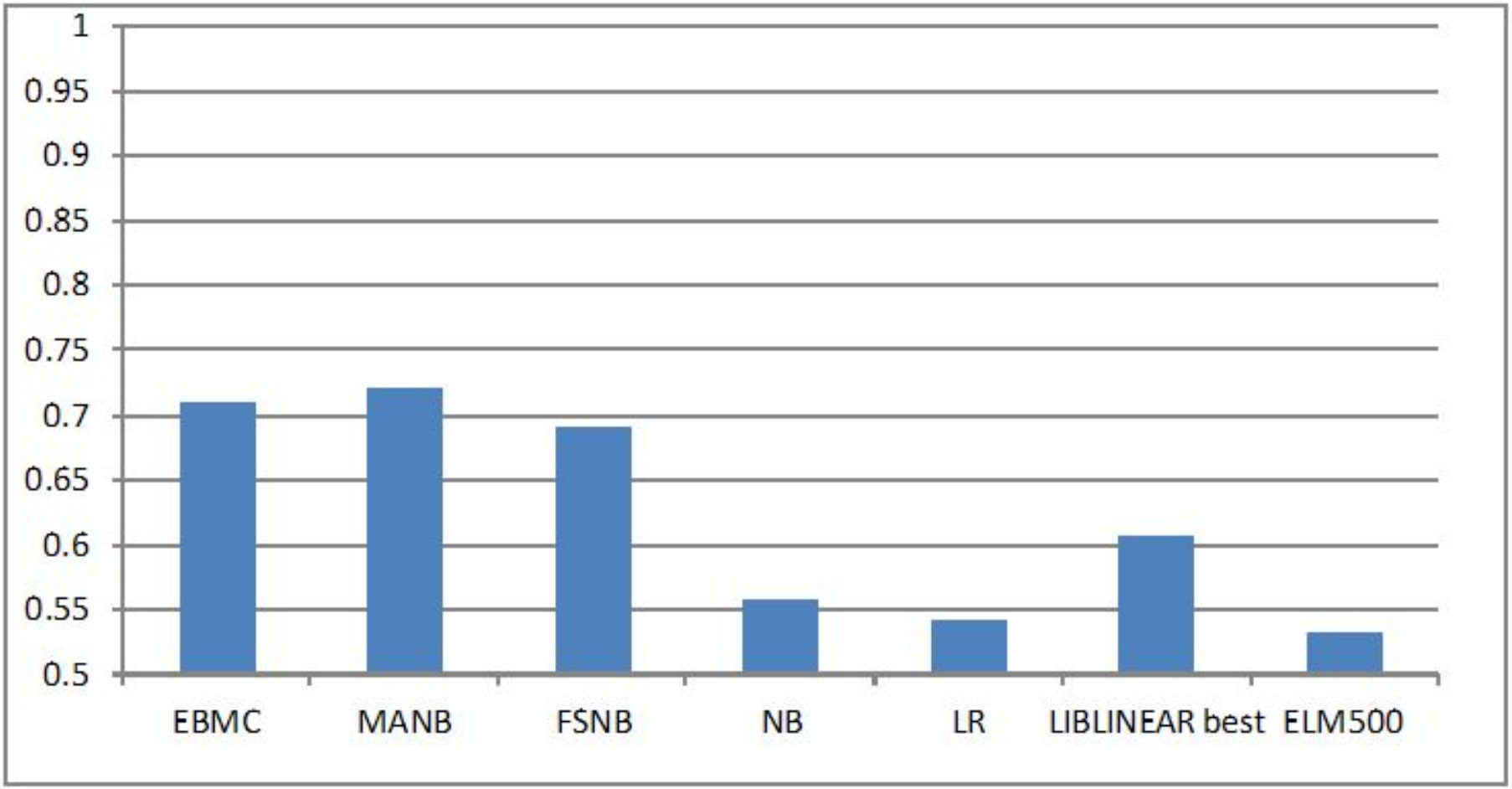
- SVM with a linear kernel has a penalty parameter C.
  - We used $C = 2^{-5}, 2^{-1}, 2^{3}, 2^{7}, 2^{11}$, and 2^{15}
- SVM with the RBF kernel has a penalty parameter C and a kernel parameter γ
  - We did a grid search with
    - $C = 2^{-5}, 2^{-1}, 2^{3}, 2^{7}, 2^{11}$, and 2^{15}
    - γ = 15, -11, -7, -3, 1, 5
- ELM has a parameter which is the number of hidden neurons.
  - We used 10, 500, and 1000 hidden neurons.
- I will show the best results obtained for the methods.

We analyzed the systems using 5-fold cross validation.

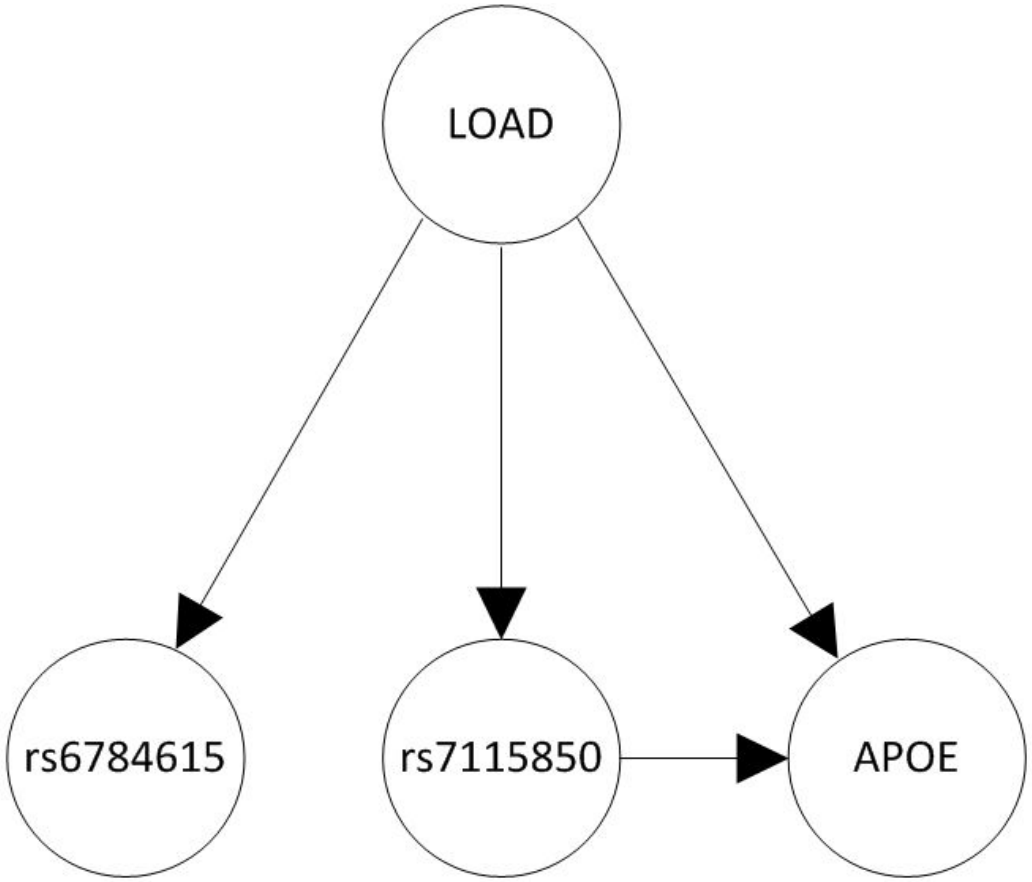
Average AUROCs for the 100 1000 and 10 10,000 SNP datasets



Average AUROCs for the LOAD Dataset



Model Learned by EBMC from the Entire LOAD Dataset



Future Research

- SVM with the RBF kernel could not handle the high-dimensional GWAS dataset.

- The ReliefF algorithm [6] ranks a set of possible predictor variables in terms of how well they predict the target variable.
  - By using a stratified nearest-neighbor-based search, it avoids assuming that the predictors are conditionally independent of each other.

- Two-stage prediction
  - ReliefF used in the first stage to identify good predictors.
  - SVM used in the second stage.

References

1. Sun L, Shenoy P. Using Bayesian networks for bankruptcy prediction: some methodological issues. European Journal of Operational Research; 2007; 180(2):738-753.
2. Neapolitan RE, Jiang X. Probabilistic Methods for Financial and Marketing Informatics, Burlington, MA; Morgan Kaufmann; 2007.
3. Mandel B, Culotta A, Boulahanis J, Stark D, Lewis, B, Rodrigue J. A demographic analysis of online sentiment during hurricane Irene. Proceedings of the Second Workshop on Language in Social Media; 2012; 27-36.
4. Cooper GF, et al. An efficient Bayesian method for predicting clinical outcomes from genome-wide data; 2010; Proceedings of AMIA 2010. Washington, D.C.
5. Jiang X, Cai B, Xue D, Lu X, Neapolitan RE, Cooper GF, A comparative analysis of methods for predicting clinical outcomes using high-dimensional genomic datasets, under revision for JAMIA
6. Marko RS, Igor K. Theoretical and empirical analysis of Relief and ReliefF. Machine Learning Journal 2003, 53:23-69.