Journal of Machine Learning Research 11 (2010) 3333-3369 Submitted 3/09; Revised 7/10; Published 12/10

Learning Instance-Specific Predictive Models

Shyam Visweswaran SHV3@PITT.EDU
Gregory F. Cooper GFC@PITT.EDU
Department of Biomedical Informatics
University of Pittsburgh
Pittsburgh, PA 15260, USA

Editor: Max Chickering

Abstract

This paper introduces a Bayesian algorithm for constructing predictive models from data that are optimized to predict a target variable well for a particular instance. This algorithm learns Markov blanket models, carries out Bayesian model averaging over a set of models to predict a target variable of the instance at hand, and employs an instance-specific heuristic to locate a set of suitable models to average over. We call this method the instance-specific Markov blanket (ISMB) algorithm. The ISMB algorithm was evaluated on 21 UCI data sets using five different performance measures and its performance was compared to that of several commonly used predictive algorithms, including nave Bayes, C4.5 decision tree, logistic regression, neural networks, *k*-Nearest Neighbor, Lazy Bayesian Rules, and AdaBoost. Over all the data sets, the ISMB algorithm performed better on average on all performance measures against all the comparison algorithms.

Keywords: instance-specific, Bayesian network, Markov blanket, Bayesian model averaging

1. Introduction

Prediction is a central problem in machine learning that involves inducing a model from a set of training instances that is then applied to future instances to predict a target variable of interest. Several commonly used predictive algorithms, such as logistic regression, neural networks, decision trees, and Bayesian networks, typically induce a single model from a training set of instances, with the intent of applying it to all future instances. We call such a model a *population-wide model* because it is intended to be applied to an entire population of future instances. A population-wide model is optimized to predict well on average when applied to expected future instances.

Recent research in machine learning has shown that inducing models that are specific to the particular features of a given instance can improve predictive performances (Gottrup et al., 2005). We call such a model an *instance-specific model* since it is constructed specifically for a particular instance (case). The structure and parameters of an instance-specific model are specialized to the particular features of an instance, so that it is optimized to predict especially well for that instance. The goal of inducing an instance-specific model is to obtain optimal prediction for the instance at hand. This is in contrast to the induction of a population-wide model where the goal is to obtain optimal predictive performance on average on all future instances.

There are several possible approaches for learning predictive models that are relevant to a single instance. One approach is to learn a model from a subset of the training data set that consists of instances that are similar in some way to the instance at hand. Another approach is to learn a

model from a subset of variables that are pertinent in some fashion to the instance at hand. A third approach, applicable to model averaging where a set of models is collectively used for prediction, is to identify a set of models that are most relevant to prediction for the instance at hand.

In this paper, we describe a new instance-specific method for learning predictive models that (1) uses Bayesian network models, (2) carries out Bayesian model averaging over a set of models to predict the target variable for the instance at hand, and (3) employs an instance-specific heuristic to identify a set of suitable models to average over. The remainder of this section gives a brief description of each of these characteristics.

Bayesian network (BN) models are probabilistic graphical models that provide a powerful formalism for representation, reasoning and learning under uncertainty (Pearl, 1988; Neapolitan, 2003). These graphical models are also referred to as probabilistic networks, belief networks or Bayesian belief networks. A BN model combines a graphical representation with numerical information to represent a probability distribution over a set of random variables in a domain. The graphical representation constitutes the BN structure, and it explicitly highlights the probabilistic independencies among the domain variables. The complementary numerical information constitutes the BN parameters, which quantify the probabilistic relationships among the variables. The instance-specific method that we describe in this paper uses Markov blanket models, which are a special type of BN models.

Typically, methods that learn predictive models from data, including those that learn BN models, perform model selection. In model selection a single model is selected that summarizes the data well; it is then used to make future predictions. However, given finite data, there is uncertainty in choosing one model to the exclusion of all others, and this can be especially problematic when the selected model is one of several distinct models that all summarize the data more or less equally well. A coherent approach to dealing with the uncertainty in model selection is Bayesian model averaging (BMA) (Hoeting et al., 1999). BMA is the standard Bayesian approach wherein the prediction is obtained from a weighted average of the predictions of a set of models, with more probable models influencing the prediction more than less probable ones. In practical situations, the number of models to be considered is enormous and averaging the predictions over all of them is infeasible. A pragmatic approach is to average over a few good models, termed *selective Bayesian model averaging*, which serves to approximate the prediction obtained from averaging over all models. The instance-specific method that we describe in this paper performs selective BMA over a set of models that have been selected in an instance-specific fashion.

The instance-specific method described here learns both the structure and parameters of BNs automatically from data. The instance-specific characteristic of the method is motivated by the intuition that in constructing predictive models, all the available information should be used including available knowledge of the features of the current instance. Specifically, the instance-specific method uses the features of the current instance to inform the BN learning algorithm to selectively average over models that differ considerably in their predictions for the target variable of the instance at hand. The differing predictions of the selected models are then combined to predict the target variable.

2. Characterization of Instance-Specific Models

Figure 1 illustrates the key difference between population-wide and instance-specific models: the instance-specific model is constructed from data in the training set, as well as, from the features

about the particular instance to which it will be applied. In contrast, the population-wide model is constructed only from data in the training set. Thus, intuitively, the additional information available to the instance-specific method can facilitate the induction of a model that provides better prediction for the instance at hand. In instance-specific modeling, different instances will potentially result in different models, because the instances contain potentially different values for the features.[1] The instance-specific models may differ in the variables included in the model (variable selection), in the interaction among the included variables (encoded in the structure of the model), and in the strength of the interaction (encoded in the parameters of the model). Another approach is to select a subset of the training data that are similar in their feature values to those of the instance at hand and learn the model from the subset. A generalization of this is to weight the instances in the training data set such that those that are more similar to the instance at hand are assigned greater weights than others, and then learn the model from the weighted data set. The following are two illustrative examples where instance-specific methods may perform better than population-wide methods.

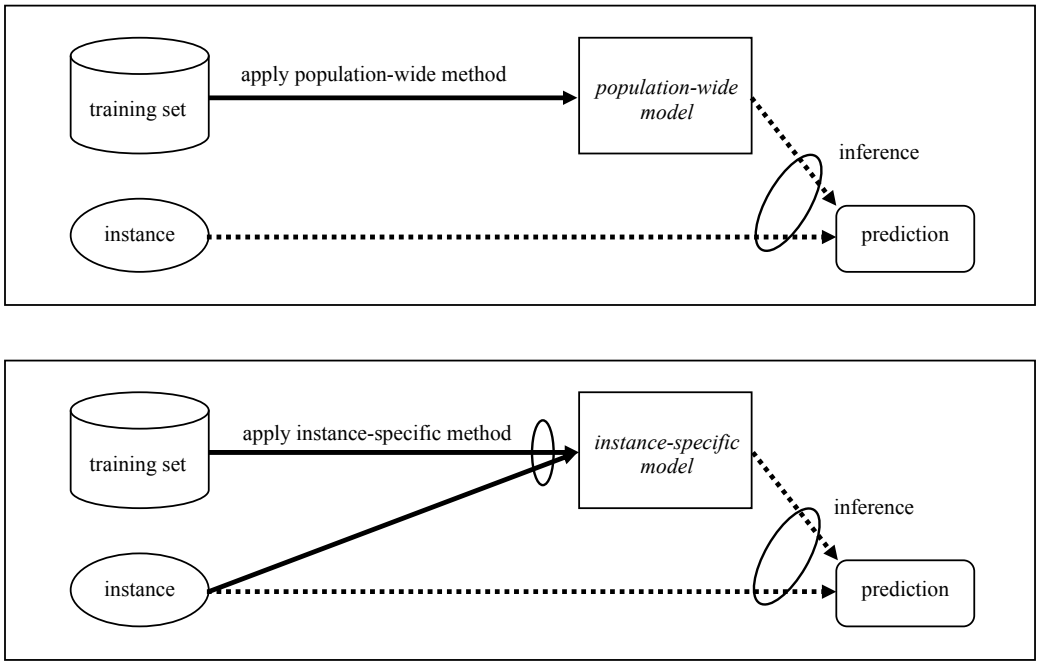


Figure 1: A general characterization of the induction of and inference in population-wide (top panel) and instance-specific (bottom panel) models. In the bottom panel, there is an extra arc from *instance* to *model*, because the structure and parameters of the model are influenced by the features of the instance at hand.

2.1 Variable Selection

Many model induction methods implicitly or explicitly perform variable selection, a process by which a subset of the domain variables is selected for inclusion in the model. For example, logistic

1. A feature is a variable-value pair, that is, a variable that has been assigned a value.

regression is often used with a stepwise variable selection process. An instance-specific version of logistic regression could, for example, select different variables for different instances being predicted, compared to the standard population-wide version that selects a single subset of variables. Consider a simple example where a gene G that has several alleles. Suppose that allele a1 is rare, and it is the only allele that predicts the development of disease D; indeed, it predicts D with high probability. For future instances, the aim is to predict $P(D|G)$. In a population-wide logistic regression model, G may not be included as a predictor (variable) of D, because in the vast majority of instances in the data set $G \neq a1$ and D is absent, and having G as a predictor would just increase the overall noise in predicting D. In contrast, if there is an instance at hand in which $G = a1$, then the training data may contain enough instances to indicate that D is highly likely. In this situation, G would be added as a predictor in an instance-specific model. Thus, for an instance in which $G = a1$, the typical population-wide logistic regression model would predict poorly, but an instance-specific model would predict well.

This idea can be extended to examples with more than one predictor, in which some predictors are characterized by having particular values that are relatively rare but strongly predictive for the outcome. A population-wide model tends to include only those predictors that on average provide the best predictive performance. In contrast, an instance-specific model will potentially include predictors that are highly predictive for the particular instance at hand; such predictors may be different from those included in the population-wide model.

2.2 Decision Theoretic Comparison of Population-Wide and Instance-Specific Models

We first introduce some notation and definitions and then compare population-wide with instance-specific models in decision theoretic terms. Capital letters like X, Z, denote random variables and corresponding lower case letters, x, z, denote specific values assigned to them. A feature is a specification of a variable and its value. Thus, $X = x$ is a feature that specifies that variable X is assigned the value x. Bold upper case letters, such as $\mathbf{X}$, $\mathbf{Z}$, represent sets of variables or random vectors, and their realization is denoted by the corresponding bold lower case letters, $\mathbf{x}$, $\mathbf{z}$. A feature vector is a list of features. Thus, $\mathbf{X} = \mathbf{x}$ is a feature vector that specifies that the variables in $\mathbf{X}$ have the values given by $\mathbf{x}$. In addition, Z denotes the target variable (class variable) being predicted, $\mathbf{X}$ denotes the set of predictor variables, M denotes a model (including both its structure and parameters), D denotes the training data set, $C^i =< \mathbf{X}^i, Z^i >$ denotes a generic training instance in D and $C^t =< \mathbf{X}^t, Z^t >$ denotes a generic test instance that is not in D. A test instance t is one in which the unknown value of the target variable Z^t is to be predicted from the known values of the predictors $\mathbf{X}^t$ and the known values of $< \mathbf{X}^i, Z^i >$ of a set of training instances.

A probabilistic *model* is a family of probability distributions indexed by a set of parameters. *Model selection* refers to the problem of using data to select one model from a set of models under consideration (Wasserman, 2000). The process of selecting a model typically involves model class selection (e.g., logistic regression, BN), variable selection, and parameter estimation. *Model averaging* refers to the process of estimating some quantity (e.g., prediction of the value of a target variable) under each of the models under consideration and obtaining a weighted average of their estimates (Wasserman, 2000).

Model selection can be done using either non-Bayesian or Bayesian approaches. Non-Bayesian methods of model selection include choosing among competing models by maximizing the likelihood, by maximizing a penalized version of the likelihood or by maximizing some measure of

interest (e.g., accuracy) using cross-validation. Use of multiple models to improve performance can also be done using either non-Bayesian or Bayesian approaches. Ensemble techniques such as bagging and boosting are non-Bayesian approaches that combine multiple models to create a new better performing model. In both bagging and boosting, the data are resampled several times, a model is constructed from each sample, and the predictions of the individual models are combined to obtain the final prediction. In the non-Bayesian approach, the heuristics used in model selection and model combination are typically different. In contrast, the Bayesian approach to model selection and model combination both involve computing the posterior probability of each model under consideration. In Bayesian model selection the single model found that has the highest posterior probability is chosen. The Bayesian model combination technique is called model averaging where the combined prediction is the weighted average of the individual predictions of the models with the model posterior probabilities comprising the weights.

When the goal is prediction of future data or future values of the target variable, BMA is preferred, since it suitably incorporates the uncertainty about the identity of the true model. However, sometimes interest is focused on a single model. For example, a single model may be useful for providing insight into the relationships among the domain variables or can be used as a computationally less expensive method for prediction. In such cases, Bayesian model selection maybe preferred to BMA. However, the optimal Bayesian approach is to perform model averaging, and thus, model selection is at best an approximation to model averaging.

Population-wide model selection and instance-specific model selection are characterized in decision theoretic terms as follows. In this paper, all conditional probabilities have a conditioning event K, which represents background knowledge and which we will leave implicit for notational simplicity. Given training data D and a generic test instance $< \mathbf{X}^t, Z^t >$, the *optimal population-wide model* is:

$$\arg\max_M \left\{ \sum_{\mathbf{X}^t} U\left[P(Z^t|\mathbf{X}^t,D), P(Z^t|\mathbf{X}^t,M)\right] P(\mathbf{X}^t|D) \right\} \tag{1}$$

where the utility function U gives the utility of approximating the *Bayes optimal estimate* $P(Z^t|\mathbf{X}^t,D)$ with the estimate $P(Z^t|\mathbf{X}^t,M)$ obtained from model M. For a model M, Expression 1 considers all possible instantiations of $\mathbf{X}^t$ and for each instantiation computes the utility of estimating $P(Z^t|\mathbf{X}^t,D)$ with the specific model estimate $P(Z^t|\mathbf{X}^t,M)$, and weights that utility by the posterior probability of that instantiation. The maximization is over the models M in a given model space.

The *Bayes optimal estimate* $P(Z^t|\mathbf{X}^t,D)$ in Expression 1 is obtained by combining the estimates of all models (in a given model space) weighted by their posterior probabilities:

$$P(Z^t|\mathbf{X}^t,D) = \int_M P(Z^t|\mathbf{X}^t,M)P(M|D)dM. \tag{2}$$

The term $P(\mathbf{X}^t|D)$ in Expression 1 is given by:

$$P(\mathbf{X}^t|D) = \int_M P(\mathbf{X}^t|M)P(M|D)dM. \tag{3}$$

The *optimal instance-specific model* for estimating Z^t is the one that maximizes the following:

$$\arg\max_M \left\{ U\left[P(Z^t|\mathbf{x}^t,D), P(Z^t|\mathbf{x}^t,M)\right] \right\}, \tag{4}$$

where $\mathbf{x}^t$ are the values of the predictors of the test instance $\mathbf{X}^t$ for which the target variable Z^t is to be predicted. The *Bayes optimal instance-specific prediction* $P(Z^t|\mathbf{X}^t, D)$ is derived using Equation 2, for the special case in which $\mathbf{X}^t = \mathbf{x}^t$, as follows:

$$P(Z^t|\mathbf{x}^t, D) = \int_M P(Z^t|\mathbf{x}^t, M)P(M|D)dM.$$

The difference between the population-wide and the instance-specific model selection can be noted by comparing Expressions 1 and 4. Expression 1 for the population-wide model selects the model that on average will have the greatest utility. Expression 4 for the instance-specific model, however, selects the model that will have the greatest utility for the specific instance $\mathbf{X}^t = \mathbf{x}^t$. For predicting Z^t given instance $\mathbf{X}^t = \mathbf{x}^t$, application of the model selected using Expression 1 can never have an expected utility greater than the application of the model selected using Expression 4. This observation provides support for developing instance-specific models.

Equations 2 and 3 carry out BMA over all models in some specified model space. Expressions 1 and 4 include Equation 2; thus, these expressions for population-wide and instance-specific model selection, respectively, are theoretical ideals. Moreover, Equation 2 is the Bayes optimal prediction of Z^t. Thus, in order to do optimal model selection, the optimal prediction obtained from BMA must already be known.

Model selection, even if performed optimally, ignores the uncertainty inherent in choosing a single model based on limited data. BMA is a normative approach for dealing with the uncertainty in model selection. Such averaging is primarily useful when no single model in the model space under consideration has a dominant posterior probability. However, since the number of models in practically useful model spaces is enormous, *exact BMA*, where the averaging is done over the entire model space, is usually not feasible. That is, it is usually not computationally feasible to solve for the exact solution given by Equation 2. In such cases, *selective BMA* is typically performed, where the averaging is done over a selected subset of models.

BMA has been shown to improve predictive performance, and several examples of significant decrease in prediction errors with the use of BMA are described by Hoeting et al. (1999). However, in other cases BMA has not proved to be better than ensemble techniques. For example, uniform averaging was shown by Cerquides and Mantaras (2005) to have better classification performance than BMA for one dependence estimators. This may be because, as Minka (2002) points out, BMA is better described as a method for 'soft model selection' rather than a technique for model combination.

3. Related Work

There exists a vast literature in machine learning, data mining and pattern recognition that is concerned with the problem of predictive modeling and supervised learning. We briefly describe some of the aspects of the similarity-based methods and instance-specific methods because these methods are most closely relevant to the present paper. Similarity-based methods are characterized by the use of a similarity (or distance) measure necessary for measuring the similarity between instances. Instance-specific methods, on the other hand, learn an explicit model or models from the training instances that are then applied to the test instance. The induction of a model or set of models are influenced by the values of the features of the test instance, and a similarity measure is not used.

3.1 Similarity-Based Methods

These methods are also known as memory-based, case-based, instance-based, or exemplar-based learners. They (1) use a similarity or a distance measure, (2) defer most of the processing until a test instance is encountered, (3) combine the training instances in some fashion to predict the target variable in the test instance, and (4) discard the answer and any intermediate results after the prediction. Typically, no explicit model is induced from the training instances at the time of prediction (Aha, 1998). The similarity measure evaluates the similarity between the test instance and the training instances and selects the appropriate training instances and their relative weights in response to the test instance (Zhang et al., 1997). The selected training instances can be equally weighted or weighted according to their similarity to the test instance. To predict the target variable in the test instance, the values of the target variable in the selected training instances are combined in some simple fashion such as majority vote, simple numerical average or fitted with a polynomial.

The *nearest-neighbor technique* is the canonical similarity-based method. When a test instance is encountered, the training instance that is most similar to the test instance is located and its target value is returned as the prediction (Cover and Hart, 1967). A straight-forward extension to the nearest-neighbor technique is the *k-Nearest Neighbor*(*k*NN) method. For a test instance, this method selects the k most similar training instances and either averages or takes a majority vote of their target values. Another extension is the distance-weighted k-Nearest Neighbor method. This weights the contribution of each of the k most similar training instances according to its similarity to the test instance, assigning greater weights to more similar instances (Dasarathy, 1991). A further extension is locally weighted regression that selects instances similar to the test instance, weights them according to their similarity, and performs regression to predict the target (Atkeson et al., 1997).

One drawback of the similarity-based methods is that they may perform poorly when predictors are redundant, irrelevant or noisy. To make the similarity metrics more robust, variable selection and variable weighting have been employed.

3.2 Instance-Specific Methods

Instance-specific methods are model-based methods that take advantage of the features in the test instance while inducing a model. Such methods are not as reliant on a similarity measure, if they use one at all, as the similarity-based methods.

Friedman et al. (1996) describe one such algorithm called LazyDT that searches for the best CART-like decision tree for a test instance. As implemented by the authors, LazyDT did not perform pruning and processed only nominal variables. The algorithm was compared to ID3 and C4.5 (standard population-wide methods for inducing decision trees), each with and without pruning. When evaluated on 28 data sets from the UCI Machine Learning repository, LazyDT generally out-performed both ID3 and C4.5 without pruning and performed slightly better than C4.5 with pruning.

Ting et al. (1999) developed a framework for inducing rules in a lazy fashion that are tailored to the features of the test instance. Zheng and Webb (2000) describe an implementation of this framework called the Lazy Bayesian Rules (LBR) learner that induces a rule tailored to the features of the test instance that is then used to classify it. A LBR rule consists of (1) a conjunction of the features (variable-value pairs) present in the test instance as the antecedent, and (2) a local naive Bayes classifier as the consequent. The structure of the local naive Bayes classifier consists of

the target variable as the parent of all other variables that do not appear in the antecedent, and the parameters of the classifier are estimated from those training instances that satisfy the antecedent. A greedy step-forward search selects the optimal LBR rule for a test instance to be classified. In particular, each predictor is added to the antecedent of the current best rule and evaluated for whether it reduces the overall error rate on the training set that is estimated by cross-validation. The predictor that most reduces the overall error rate is added to the antecedent and removed from the consequent, and the search continues; if no single predictor move can decrease the current error rate, then the search halts and the current rule is applied to predict the outcome for the test instance. LBR is an example of an instance-specific method that uses feature information available in the test instance to direct the search for a suitable model in the model space.

The performance of LBR was evaluated by Zheng and Webb (2000) on 29 data sets from the UCI Machine Learning repository and compared to that of seven algorithms: a nave Bayes classifier (NB), a decision tree algorithm (C4.5), a Bayesian tree learning algorithm (NBTree) (Kohavi, 1996), a constructive Bayesian classifier that replaces single variables with new variables constructed from Cartesian products of existing nominal variables (BSEJ) (Pazzani, 1998), a selective naive Bayes classifier that deletes irrelevant variables using Backward Sequential Elimination (BSE) (Pazzani, 1995), and LazyDT, which is described above. Based on ten three-fold cross validation trials (for a total of 30 trials), LBR achieved the lowest average error rate across the 29 data sets. The average relative error reduction of LBR over NB, C4.5, NBTree, BSEJ, BSE and LazyDT were 9%, 10%, 2%, 3%, 5% and 16% respectively. LBR performed significantly better than all other algorithms except BSE; compared to BSE its performance was better but not statistically significantly so.

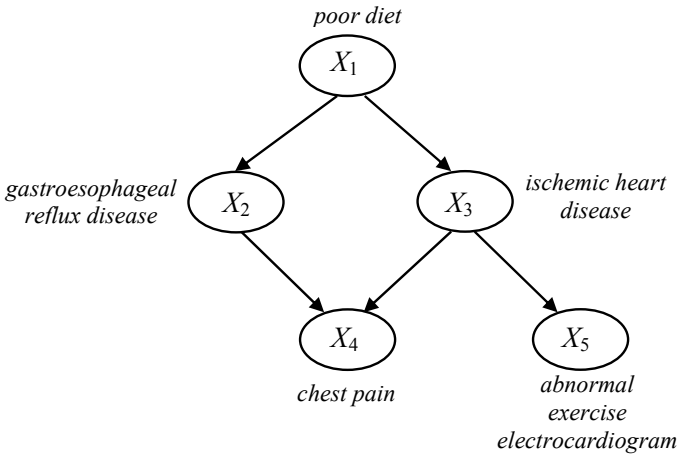
The instance-specific algorithms like LazyDT and LBR have limitations in that they can process only discrete variables, and continuous variables have to be discretized. Also, they are computationally more intensive than many other learning algorithms. However, they have been shown to have better accuracy than several of the population-wide methods.

4. Bayesian Networks

A Bayesian network (BN) is a probabilistic model that combines a graphical representation (the BN structure) with quantitative information (the BN parameterization) to represent a joint probability distribution over a set of random variables (Pearl, 1988; Neapolitan, 2003). More specifically, a BN model M representing the set of random variables $\mathbf{X}$ for some domain consists of a pair G, θ_G. The first component G is a directed acyclic graph (DAG) that contains a node for every variable in $\mathbf{X}$ and an arc between a pair of nodes if the corresponding variables are directly probabilistically dependent. Conversely, the absence of an arc between a pair of nodes denotes probabilistic independence between the corresponding variables. In this paper, the terms variable and node are used interchangeably in the context of random variables being modeled by nodes in a BN. Thus, a variable X_i in the domain of interest is represented by a node labeled X_i in the BN graph. Note that the phrase *BN structure* refers only to the graphical structure G, while the term BN (model) refers to both the structure G and a corresponding set of parameters θ_G.

The terminology of kinship is used to denote various relationships among nodes in a graph. These kinship relations are defined along the direction of the arcs. Predecessors of a node X_i in G, both immediate and remote, are called the *ancestors* of X_i. In particular, the immediate predecessors of X_i are called the *parents* of X_i. The set of parents of X_i in G is denoted by $\mathbf{Pa}(X_i, G)$ or more simply as $\mathbf{Pa}_i$ when the BN structure is obvious from the context. In a similar fashion, successors of

X_i in G, both immediate and remote, are called the *descendants* of X_i, and the immediate successors are called the *children* of X_i. A node X_j is termed a *spouse* of X_i if X_j is a parent of a child of X_i. The set of nodes consisting of a node X_i and its parents is called the *family* of X_i. Figure 2 gives an illustrative example of a simple hypothetical BN, where the top panel shows the graphical component G of the BN. In the figure, the variable *poor diet* is a parent of the variable *ischemic heart disease* as well as a parent of the variable *gastroesophageal reflux disease*. The variable *chest pain* is a child of the variable *lung cancer* as well as a child of the variable *gastroesophageal reflux disease*, and the variables *ischemic heart disease* and *abnormal electrocardiogram* are descendants of the variable *poor diet*.



Node X_1	$P(X_1 = \text{F}) = 0.70$	$P(X_1 = \text{T}) = 0.30$
Node X_2	$P(X_2 = \text{F} \mid X_1 = \text{F}) = 0.97$	$P(X_2 = \text{T} \mid X_1 = \text{F}) = 0.03$
	$P(X_2 = \text{F} \mid X_1 = \text{T}) = 0.96$	$P(X_2 = \text{T} \mid X_1 = \text{T}) = 0.04$
Node X_3	$P(X_3 = \text{F} \mid X_1 = \text{F}) = 0.94$	$P(X_3 = \text{T} \mid X_1 = \text{F}) = 0.06$
	$P(X_3 = \text{F} \mid X_1 = \text{T}) = 0.96$	$P(X_3 = \text{T} \mid X_1 = \text{T}) = 0.08$
Node X_4	$P(X_4 = \text{F} \mid X_2 = \text{F}, X_3 = \text{F}) = 0.90$	$P(X_4 = \text{T} \mid X_2 = \text{F}, X_3 = \text{F}) = 0.10$
	$P(X_4 = \text{F} \mid X_2 = \text{F}, X_3 = \text{T}) = 0.40$	$P(X_4 = \text{T} \mid X_2 = \text{F}, X_3 = \text{T}) = 0.60$
	$P(X_4 = \text{F} \mid X_2 = \text{T}, X_3 = \text{F}) = 0.50$	$P(X_4 = \text{T} \mid X_2 = \text{T}, X_3 = \text{F}) = 0.50$
	$P(X_4 = \text{F} \mid X_2 = \text{T}, X_3 = \text{T}) = 0.25$	$P(X_4 = \text{T} \mid X_2 = \text{T}, X_3 = \text{T}) = 0.75$
Node X_5	$P(X_5 = \text{F} \mid X_3 = \text{F}) = 0.80$	$P(X_5 = \text{T} \mid X_3 = \text{F}) = 0.20$
	$P(X_5 = \text{F} \mid X_3 = \text{T}) = 0.25$	$P(X_5 = \text{T} \mid X_3 = \text{T}) = 0.75$

Figure 2: A simple hypothetical Bayesian network for a medical domain. All the nodes represent binary variables, taking values in the domain T, F where T stands for True and F for False. The graph at the top represents the Bayesian network structure. Associated with each variable (node) is a conditional probability table representing the probability of each variable's value conditioned on its parent set. (Note: these probabilities are for illustration only; they are not intended to reflect frequency of events in any actual patient population.)

The second component θ_G represents the parameterization of the probability distribution over the space of possible instantiations of **X** and is a set of local probabilistic models that encode quantitatively the nature of dependence of each variable on its parents. For each node X_i there is a probability distribution (that may be discrete or continuous) defined on that node for each state of its parents. The set of all the probability distributions associated with all the nodes comprises the complete parameterization of the BN. The bottom panel in Figure 2 gives an example of a set of parameters for G. Taken together, the top and bottom panels in Figure 2 provide a fully specified structural and quantitative representation for the BN.

4.1 Markov Blanket

The *Markov blanket* of a variable X_i, denoted by $\mathrm{MB}(X_i)$, defines a set of variables such that conditioned on $\mathrm{MB}(X_i)$ is conditionally independent of all variables given $\mathrm{MB}(X_i)$ for joint probability distributions consistent with BN in which $\mathrm{MB}(X_i)$ appears (Pearl, 1988). The minimal Markov blanket of a node X_i, which is sometimes called its Markov boundary, consists of the parents, children, and children's parents of X_i. In this paper, we refer to the minimal Markov blanket as the Markov blanket (MB). This entails that the variables in $\mathrm{MB}(X_i)$ are sufficient to determine the probability distribution of X_i. Since d-separation is applied to the graphical structure of a BN to identify all conditional independence relations, it can also be applied to identify the MB of a node in a BN. The MB of a node X_i consists of its parents, its children, and its children's parents and is illustrated in Figure 3. The parents and children of X_i are directly connected to it. In addition, the spouses are also included in the MB, because of the phenomenon of explaining away which refers to the observation that when a child node is instantiated its parents in general are statistically dependent. Analogous to BNs, the *MB structure* refers only to the graphical structure while the MB (model) refers to both the structure and a corresponding set of parameters.

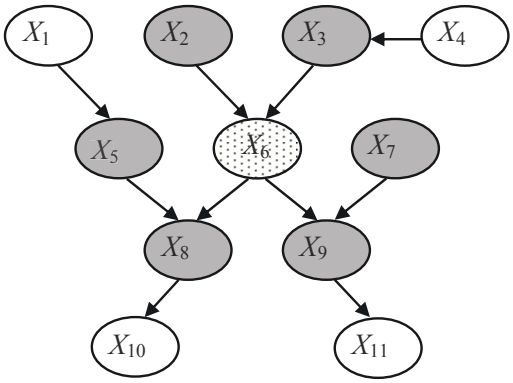


Figure 3: Example of a Markov blanket. The Markov blanket of the node X_6 (shown stippled) comprises the set of parents, children and spouses of the node and is indicated by the shaded nodes. The nodes in the Markov blanket include X_2 and X_3 as parents, X_8 and X_9 as children, and X_5 and X_7 as spouses of X_6. X_1, X_4, X_{10} and X_{11} are not in the Markov blanket of X_6.

The MB of a node is noteworthy because it identifies all the variables that shield the node from the rest of the network. In particular, when interest centers on the distribution of a specific target

node, as is the case in classification, the structure and parameters of only the MB of the target node need be learned.

4.2 Markov Blanket Algorithms

Many approaches for learning general BNs as well as for learning MBs from data have been described in the literature. Here we briefly review algorithms that learn MB classifiers. One of the earliest described MB learning algorithms is the Grow-Shrink (GS) Markov blanket algorithm that orders the variables according to the strength of association with the target and uses conditional independence tests to find a reduced set of variables estimated to be the MB (Margaritis and Thrun, 1999). Madden (2002a,b) described the Markov Blanket Bayesian Classifier (MBBC) algorithm that constructs an approximate MB classifier using a Bayesian score for evaluating the network. The algorithm consists of three steps: the first step identifies a set of direct parents and children of the target, the second step identifies a set of parents of the children, and the third step identifies dependencies among the children. The MBBC was competitive in terms of speed and accuracy relative to Nave Bayes, Tree-Augmented Nave Bayes and general Bayesian networks, when evaluated on a large set of UCI data sets.

Several MB algorithms have been developed in the context of variable selection and learning local causal structures around target variables of interest. Koller and Sahami (1996) showed that the optimal set of variables to predict a target is its MB. They proposed a heuristic entropy-based procedure (commonly referred to as the KS algorithm) that assumes that the target influences the predictor variables and that the variables most strongly associated with the target are in its MB. The KS algorithm was not guaranteed to succeed. Tsamardinos and Aliferis (2003) showed that for faithful distributions, the MB of a target variable is exactly the set of strongly relevant features, and developed the Incremental Association Markov Blanket (IAMB) to identify it. This algorithm has two stages: a growing phase that adds potential predictor variables to the MB and a shrinking phase that removes the false positives that were added in the first phase. Based on the faithfulness assumption, Tsamardinos et al. (2006) later developed the Min-Max Markov Blanket algorithm (MMMB) that first identifies the direct parents and children of the target and then parents of the children using conditional independence tests. A comparison of the efficiency of several MB learning algorithms are provided by Fu and Desmarais (2008). A recent comprehensive overview of MB methods of classification and the local structure learning is provided by Aliferis et al. (2010a,b).

Several methods for averaging over BNs for prediction or classification have been described in the literature, including Dash and Cooper (2002), Dash and Cooper (2004) and Hwang and Zhang (2005). In prior work, we developed a lazy instance-specific algorithm that performs BMA over LBR models (Visweswaran and Cooper, 2004) and showed that it had better classification performance than did model selection. However, to our knowledge, averaging over MBs has not been described in the literature.

5. The Instance-Specific Markov Blanket (ISMB) Algorithm

The goal of the instance-specific Markov blanket (ISMB) algorithm is to predict well a discrete target variable of interest. Relative to some model space, BMA is the optimal method for making predictions in the sense that it achieves the lowest expected error rate in predicting the outcomes of future instances. Such Bayes optimal predictions involve averaging over all models in the model space which is usually computationally intractable. One approach, termed *selective model averag-*

ing, has been to approximate the Bayes optimal prediction by averaging over a subset of the possible models and has been shown to improve predictive performance (Hoeting et al., 1999; Raftery et al., 1997; Madigan and Raftery, 1994). The ISMB algorithm performs selective model averaging and uses a novel heuristic search method to select the models over which averaging is done. The instance-specific characteristic of the algorithm arises from the observation that the search heuristic is sensitive to the features of the particular instance at hand.

The model space employed by the ISMB algorithm is the space of BNs over the domain variables. In particular, the algorithm considers only MBs of the target node, since a MB is sufficient for predicting the target variable. The remainder of this section describes the ISMB algorithm in terms of the (1) model space, (2) scoring functions including parameter and structure priors, and (3) the search procedure for exploring the space of models. The current version of the algorithm handles discrete variables.

5.1 Model Space

As mentioned above, the ISMB algorithm learns MBs of the target variable rather than entire BNs over all the variables. Typically, BN structure learning algorithms that learn from data induce a BN structure over all the variables in the domain. The MB of the target variable can be extracted from the learned BN structure by ignoring those nodes and their relations that are not members of the MB. The ISMB algorithm modifies the typical BN structure learning algorithm to learn only MBs of the target node of interest, by using a set of operators that generate only the MB structures of the target variable.

The ISMB algorithm is a search-and-score method that searches in the space of possible MB structures. Both, the BN structure learning algorithms and the MB structure learning algorithm used by ISMB, search in a space of structures that is exponential in the number of domain variables. Though the number of MB structures grows more slowly than the number of BN structures with the number of domain variables, the number of MB structures is still exponential in the number of variables (Visweswaran and Cooper, 2009). Thus, exhaustive search in this space is infeasible for domains containing more than a few variables and heuristic search is appropriate.

5.2 Instance-Specific Bayesian Model Averaging

The objective of the ISMB algorithm is to derive the posterior distribution $P(Z^t|,\mathbf{x}^t,D)$ for the target variable Z^t in the instance at hand, given the values of the other variables $\mathbf{X}^t = \mathbf{x}^t$ and the training data D. The ideal computation of the posterior distribution $P(Z^t|,\mathbf{x}^t,D)$ by BMA is as follows:

$$P(Z^t|\mathbf{x}^t,D) = \sum_{G \in M} P(Z^t|\mathbf{x}^t,G,D)P(G|D), \tag{5}$$

where the sum is taken over *all* MB structures G in the model space M. The first term on the right hand side, $P(Z^t|\mathbf{x}^t,G,D)$, is the probability $P(Z^t|\mathbf{x}^t)$ computed with a MB that has structure G and parameters $\hat{\theta}_G$ that are given by Equation 6 below. This parameterization of G produces predictions equivalent to those obtained by integrating over all the possible parameterizations for G. The second term, $P(G|D)$, is the posterior probability of the MB structure G given the training data D. In essence, Equation 5 states that a conditional probability of interest $P(Z^t|\mathbf{x}^t)$ is derived by taking a weighted average of that probability over all MB structures, where the weight associated

with a MB structure is the probability of that MB structure given the data. In general, $P(Z^t|\mathbf{x}^t)$ will have different values for the different sets of models over which the averaging is carried out.

5.3 Inference in Markov Blankets

Computing $P(Z^t|\mathbf{x}^t, G, D)$ in Equation 5 involves doing inference in the MB with a specified structure G. First, the parameters of the MB structure G are estimated using Bayesian parameters as given by the following expression (Cooper and Herskovits, 1992; Heckerman, 1999):

$$P(X_i = k|\mathbf{Pa}_i = j) \equiv \hat{\theta}_{ijk} = \frac{\alpha_{ijk} + N_{ijk}}{\alpha_{ij} + N_{ij}} \tag{6}$$

where (1) N_{ijk} is the number of instances in data set D in which $X_i = k$ and the parents of X_i have the state denoted by j, (2) $N_{ij} = \sum_k N_{ijk}$, (3) α_{ijk} is a parameter prior that can be interpreted as belief equivalent to having previously (prior to obtaining D seen α_{ijk} instances in which $X_i = k$ and the parents of X_i have the state denoted by j, and (4) $\alpha_{ij} = \sum_k \alpha_{ijk}$. The $\hat{\theta}_{ijk}$ in Equation 6 represent the expected value of the probabilities that are derived by integrating over all possible parameter values. For the ISMB algorithm we set α_{ijk} to 1 for all i, j, and k, as a simple non-informative parameter prior (Cooper and Herskovits, 1992). Next, the parameterized MB is used to compute the distribution over the target variable Z^t of the instance at hand given the values $\mathbf{x}^t$ of the remaining variables in the MB by applying standard BN inference (Neapolitan, 2003).

5.4 Bayesian Scoring of Markov Blankets

In the Bayesian approach, the scoring function is based on the posterior probability $P(G|D)$ of the BN structure G given data D. This is the second term on the right hand side in Equation 3. The Bayesian approach treats the structure and parameters as uncertain quantities and incorporates prior distributions for both. The specification of the structure prior $P(G)$ assigns prior probabilities for the different MB structures. Application of Bayes rule gives:

$$P(G|D) = \frac{P(D|G)P(G)}{P(D)}. \tag{7}$$

Since the denominator $P(D)$ does not vary with the structure, it simply acts as a normalizing factor that does not distinguish between different structures. Dropping the denominator yields the following Bayesian score:

$$score(G;D) = P(D|G)P(G). \tag{8}$$

The second term on the right in Equation 8 is the prior over structures, while the first term is the marginal likelihood (also know as the integrated likelihood or evidence) which measures the goodness of fit of the given structure to the data. The marginal likelihood is computed as follows:

$$P(D|G) = \int_{\theta_G} P(D|\theta_G, G)P(\theta_G|G)d\theta_G, \tag{9}$$

where $P(D|\theta_G, G)$ is the likelihood of the data given the BN (G, θ_G) and $P(\theta_G|G)$ is the specified prior distribution over the possible parameter values for the network structure G. Intuitively, the marginal likelihood measures the goodness of fit of the structure over all possible values of its parameters. Note that the marginal likelihood is distinct from the maximum likelihood, though both

are computed from the same function: the likelihood of the data given the structure. The maximum likelihood is the maximum value of this function while the marginal likelihood is the integrated (or the average) value of this function with the integration being carried out with respect to the prior $P(\theta_G|G)$.

Equation 9 can be evaluated analytically when the following assumptions hold: (1) the variables are discrete and the data D is a multinomial random sample with no missing values; (2) global parameter independence, that is, the parameters associated with each variable are independent; (3) local parameter independence, that is, the parameters associated with each parent state of a variable are independent; and (4) the parameters' prior distribution is Dirichlet. Under the above assumptions, the closed form for $P(D|G)$ is given by (Cooper and Herskovits, 1992; Heckerman, 1999):

$$P(D|G) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}, \tag{10}$$

where Γ denotes the Gamma function,n is the number of variables in G, q_i is the number of joint states of the parents of variable X_i that occur in D, r_i is the number of states of X_i that occur in D, and $\alpha_{ij} = \sum_k \alpha_{ijk}$. Also, as previously described, N_{ijk} is the number of instances in the data where node i has value k and the parents of i have the state denoted by j, and $N_{ij} = \sum_k N_{ijk}$.

The Bayesian score in Equation 7 incorporates both structure and parameter priors. The term $P(G)$ represents the structure prior and is the prior probability assigned to the BN structure G. For the ISMB algorithm, a uniform prior belief over all G is assumed which makes the term $P(G)$ a constant. Thus, $P(G|D)$ is equal to $P(D|G)$ up to a proportionality constant and the Bayesian score for $P(G)$ is defined simply as the marginal likelihood as follows:

$$score(G;D) = P(D|G) \propto P(G|D). \tag{11}$$

The parameter priors are incorporated in the marginal likelihood $P(D|G)$ as is obvious from the presence of the alpha terms in Equation 10. For the ISMB algorithm we set α_{ijk} to 1 for all i, j, and k in Equation 10, as a simple non-informative parameter prior, as mentioned in the previous section.

5.5 Selective Bayesian Model Averaging

Since Equation 5 sums over a very large number of MB structures, it is not feasible to compute it exactly. Hence, complete model averaging given by Equation 5 is approximated with selective model averaging, and heuristic search (described in the next section) is used to sample the model space. For a set R of MB structures that have been chosen from the model space by heuristic search, selective model averaging estimates $P(Z^t|\mathbf{x}^t, G)$ as:

$$P(Z^t|\mathbf{x}^t, D) \cong \sum_{G \in R} P(Z^t|\mathbf{x}^t, G, D) \frac{P(G|D)}{\sum_{G' \in R} P(G'|D)}. \tag{12}$$

Substituting Equation 11 into Equation 12, we obtain:

$$P(Z^t|\mathbf{x}^t, D) \cong \sum_{G \in R} P(Z^t|\mathbf{x}^t, G, D) \frac{score(G;D)}{\sum_{G' \in R} score(G';D)}. \tag{13}$$

The ISMB algorithm performs selective model averaging and seeks to locate a good set of models over which averaging is carried out.

5.6 Instance-Specific Search

The ISMB algorithm uses a two-phase search to sample the space of MB structures. The first phase (phase 1) ignores the evidence $\mathbf{x}^t$ from the instance at hand, while searching for MB structures that best fit the training data. The second phase (phase 2) continues to add to the set of MB structures obtained from phase 1, but now searches for MB structures that have the greatest impact on the prediction of Z^t for the instance at hand. We now describe in greater detail the two phases of the search.

Phase 1 uses *greedy hill-climbing search* and accumulates the best model discovered at each iteration of the search into a set R. At each iteration of the search, successor models are generated from the current best model; the best of the successor models is added to R *only if* this model is better than current best model; and the remaining successor models are discarded. Since, no backtracking is performed, phase 1 search terminates in a local maximum.

Phase 2 uses *best-first search* and adds the best model discovered at each iteration of the search to the set R. Unlike greedy hill-climbing search, best-first search holds models that have not been expanded (i.e., whose successors have not be generated) in a *priority queue* Q. At each iteration of the search, successor models are generated from the current best model and added to Q; after an iteration the best model from Q is added to R *even if* this model is not better than the current best model in R. Phase 2 search terminates when a user set criterion is satisfied. Since, the number of successor models that are generated can be quite large, the priority queue Q is limited to a capacity of at most w models. Thus, if Q already contains w models, addition of a new model to it leads to removal of the worst model from it. The queue allows the algorithm to keep in memory up to the best w scoring models found so far, and it facilitates limited backtracking to escape local maxima.

5.7 Search Operators and Scores

The operators used by the ISMB algorithm to traverse the space of MB structures are the same as those used in standard BN structure learning with minor modifications. The standard BN structure learning operators are (1) add an arc between two nodes if one does not exist, (2) delete an existing arc, and (3) reverse an existing arc, with the constraint that an operation is allowed only if it generates a legal BN structure (Neapolitan, 2003). This constraint simply implies that the graph of the generated BN structure be a DAG. A similar constraint is applicable to the generation of MB structures, namely, that an operation is considered valid if it produces a legal MB structure of the target node. This constraint entails that some of the operations be deemed invalid, as illustrated in the following examples. With respect to a MB, the nodes can be categorized into five groups: (1) the target node, (2) parent nodes of the target, (3) child nodes of the target, (4) spousal nodes, which are parents of the children, and (5) other nodes, which are not part of the current MB. Incoming arcs into parents or spouses are not part of the MB structure and, hence operations that add such arcs are deemed invalid. Arcs between nodes not in the MB are not part of the MB structure and, hence operations that add such arcs are also deemed invalid. Figure 4 gives exhaustively the validity of the MB operators. Furthermore, the application of the delete arc or the reverse arc operators may lead to additional removal of arcs to produce a valid MB structure (see Figure 5 for an example).

As described in the previous section, the search for MB structures proceeds in two sequential phases. In phase 1 the candidate MB structures are scored with the Bayesian score (phase 1 score) shown in Equation 11. Since this phase selects the highest scoring MB structure at each iteration, it

X \ Y	T	P	C	S	O
T				✓	✓
P			✓		
C			✓*		
S	✓		✓		
O	✓		✓		

(a) Add arc X → Y

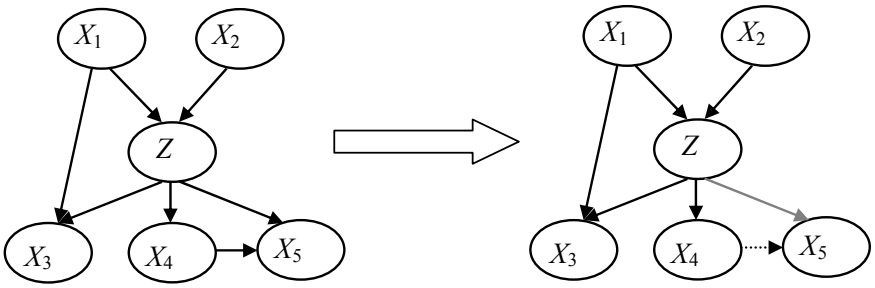
X \ Y	T	P	C	S	O
T			✓		
P	✓		✓		
C			✓		
S			✓		
O					

(b) Delete arc X → Y

X \ Y	T	P	C	S	O
T			✓		
P	✓				
C			✓*		
S					
O					

(c) Reverse arc X → Y

Figure 4: Constraints on the Markov blanket operators. The nodes are categorized into five groups: T = target, P = parent, C = child, S = spouse, and O = other (not in the Markov blanket of T). The cells with check marks indicate valid operations and are the only ones that need to be considered in generating candidate structures. The cells with an asterisk indicate that the operation is valid only if the resulting graph is acyclic.



(a) (b)

Figure 5: An example where the application of an operator leads to additional removal of arcs to produce a valid Markov blanket structure. Deletion of arc $Z \rightarrow X_5$ leads to removal of the arc $X_4 \rightarrow X_5$ since X_5 is no longer a part of the Markov blanket of Z. Reversal of the same arc also leads to removal of the arc $X_4 \rightarrow X_5$ since X_5 is now a parent and is precluded from having incoming arcs. Also, unless $X_4 \rightarrow X_5$ is removed there will be a cycle.

accumulates MB structures with high marginal likelihood. The purpose of this phase is to identify a set of MB structures that are highly probable, given data D.

Phase 2 searches for MB structures that change the current model-averaged estimate of $P(Z^t|\mathbf{x}^t, D)$ the most. The notion here is to find viable competing MB structures for making this posterior probability prediction. When no competitive MB structures can be found, the prediction is assumed to be stable. Phase 2 differs from the phase 1 in two aspects: it uses best-first search and it employs a different scoring function for evaluating candidate MB structures.

At the beginning of the phase 2, R contains MB structures that were generated in phase 1. Successors to the MB structures in R are generated, scored with the phase 2 score (described in detail below) and added to the priority queue Q. At each iteration of the search, the highest scoring MB structure in Q is removed from Q and added to R; all operations leading to legal MB structures are applied to it; the successor structures are scored with the phase 2 score; and the scored structures are added to Q. Phase 2 search terminates when no MB structure in Q has a score higher than some small value ε or when a period of time t has elapsed, where ε and t are user specified parameters.

In phase 2, the model score is computed as follows. Each successor MB structure G^* to be added to Q is scored based on how much it changes the current estimate of $P(Z^t|\mathbf{x}^t,D)$; this is obtained by model averaging over the MB structures in R. More change is better. Specifically, we use the Kullback-Leibler (KL) divergence between the two estimates of $P(Z^t|\mathbf{x}^t,D)$, one estimate computed with and another computed without G^* in the set of models over which the model averaging is carried out. The KL divergence, or relative entropy, is a quantity which measures the difference between two probability distributions (Cover and Joy, 2006). Thus, the phase 2 score for a candidate MB structure G^* is given by:

$$f(R,G^*) = \mathrm{KL}(p||q) \equiv \sum_x p(x) \log \frac{p(x)}{q(x)},$$

where

$$p(x) = \sum_{G \in R} P(Z^t|\mathbf{x}^t,G,D) \frac{P(G|D)}{\sum_{G' \in R} P(G'|D)}$$

and

$$q(x) = \sum_{G \in R \cup G^*} P(Z^t|\mathbf{x}^t,G,D) \frac{P(G|D)}{\sum_{G' \in R \cup G^*} P(G'|D)}.$$

Using Equation 11 the term $P(G|D)$ that appears in $p(x)$ and $q(x)$ can be substituted with the term $score(G;D)$. Using this substitution, the score for G^* is:

$$f(R,G^*) = \mathrm{KL}(p||q) \equiv \sum_x p(x) \log \frac{p(x)}{q(x)}, \tag{14}$$

where

$$p(x) = \sum_{G \in R} P(Z^t|\mathbf{x}^t,G,D) \frac{score(G;D)}{\sum_{G' \in R} score(G';D)}$$

and

$$q(x) = \sum_{G \in R \cup G^*} P(Z^t|\mathbf{x}^t,G,D) \frac{score(G;D)}{\sum_{G' \in R \cup G^*} score(G';D)}.$$

The pseudocode for the two-phase search procedure used by ISMB algorithm is given in Figure 6.

```
ProcedureSearchForISMB
    // phase 1: greedy hill-climbing search
    R ← empty set
    BestModel ← empty MB (graph containing only the target node)
    Score BestModel with phase 1 score
    BestScore ← phase 1 score of BestModel
    Add BestModel to R

    Do
        For every possible operator O that can be applied to BestModel
            Apply O to BestModel to derive Model
            Score Model with phase 1 score
            ModelScore ← phase 1 score of Model
            If ModelScore > BestScore
                BestModel ← Model
                BestScore ← ModelScore
                FoundBetterModel ← True
            End if
        End for
        If FoundBetterModel is True
            Add BestModel to R
        Else
            Terminate do
        End if
    End do

    // phase 2: best-first search
    Q ← empty priority queue with maximum capacity w
    Generate all successors for the MB structures in R and add them to Q
    Score all MB structures in Q with phase 2 score

    Do while elapsed time < t
        BestModel ← remove MB structure with highest phase 2 score from Q
        BestScore ← phase 2 score of BestModel
        For every possible operator O that can be applied to BestModel
            Apply O to BestModel to derive Model
            Score Model with phase 2 score
            Add Model to Q
        End for
        If BestScore > ε
            Add BestModel to R
        Else
            Terminate do
        End if
    End do

    Return R
```

Figure 6: Pseudocode for the two-phase search procedure used by the ISMB algorithm. Phase 1 uses greedy hill-climbing search while phase 2 uses best-first search.

5.8 Complexity of the ISMB Algorithm

For one instance, the ISMB algorithm runs in $O(bdmn)$ time and uses $O((w+d)mn)$ space, where m is the number of instances in the training data set D, n is the number of domain variables, d is the

total number of iterations of the search in the two phases 2, b (the branching factor) is the maximum number of successors generated from a MB structure, and w is the capacity of the priority queue Q.

5.8.1 TIME COMPLEXITY

At each iteration of the search, a maximum of b successor MB structures are generated. For d iterations of the search, the number of MB structures generated and scored with the phase 1 score is $O(bd)$. Note that both phases of the search require successor MB structures to be scored with the phase 1 score.

Since the phase 1 score decomposes over the MB nodes, to compute it for a newly generated MB structure only those MB nodes whose parent nodes have changed need be evaluated. The number of MB nodes that need to be evaluated is either one (when the *add* or *remove* operator is applied) or two (when the *reverse* operator is applied). Computing the phase 1 score for a MB node entails estimating the parameters for that node and calculating the marginal likelihood from those parameters. Estimating the parameters requires one pass over D and takes $O(mn)$ time which determines the time complexity of the phase 1 score.

The phase 2 score computes the effect of a candidate MB structure on the model averaged estimate of the distribution of the target variable. This requires doing inference for the target node in a MB that contains all measured variables which takes $O(n)$ since at most n nodes influence the target distribution and hence at most n sets of parameters need be retrieved. Computing both phase 1 and phase 2 scores for a MB structure therefore takes $O(mn)$ time. Thus, the total time required by the ISMB algorithm that runs for d iterations of the search and generates b MB structures at each iteration is $O(bdmn)$. However, the branching factor b is $O(n^2)$ and d is $O(n)$ and hence the overall complexity is $O(mn^4)$. This complexity limits that algorithm's applicability to data sets of small to medium dimensionality with up to several hundred variables.

5.8.2 SPACE COMPLEXITY

The ISMB algorithm searches in the space of MB structures using greedy hill-climbing search for phase 1 and best-first search with a priority queue of capacity w for phase 2. For d iterations of the search, the maximum number of MB structures that is stored is $O(w+d)$. The space required for each MB structure is linear in the number of its parameters.

For a given MB node, the number of parameters (using a conditional probability table) is exponential in the number of its parent nodes. However, the number of distinct parameters cannot be greater than the number of instances m in the training data D; the remaining parameters for a node have a single default value. Thus, the space required for the parameters of a MB node is $O(m)$. In a domain with n variables, a MB structure can have up to n nodes and thus requires space of $O(mn)$. In total, the space required by the ISMB algorithm that runs for d iterations of the search is $O((w+d)mn)$.

6. Evaluation of the ISMB Algorithm

This section describes the evaluation of the ISMB algorithm on a synthetic data set and several data sets from the UCI Machine Learning repository (UCI data sets). We first describe the preprocessing of variables, the evaluation measures and the comparison algorithms.

6.1 Preprocessing

Any instance that had one or more missing values was removed from the data set, as was done by Friedman et al. (1997). Sixteen of the 21 UCI data sets have no missing values and no instances were removed. In the remaining five data sets, removal of missing values resulted in a decrease in the size of the data set of less than 10%. After the removal of instances with missing values, the data sets were evaluated with two stratified applications of 10-fold cross-validation. Hence, each data set was split twice into 10 stratified training and test folds to create a total of 20 training and test folds. All experiments were carried out on the same set of 20 training and test folds. All target variables in all the data sets are discrete. However, some of the predictor variables are continuous. All continuous variables were discretized using the method described by Fayyad and Irani (1993). The discretization thresholds were determined only from the training sets and then applied to both the training and test sets.

6.2 Performance Measures

The performance of the ISMB algorithm was evaluated on two measures of discrimination (i.e., prediction under 0-1 loss) and three probability measures. The discrimination measures used are the misclassification error and the area under the ROC curve (AUC). For multiple classes, we used the method described by Hand and Till (2001) for computing the AUC. The discrimination measures evaluate how well an algorithm differentiates among the various classes (or values of the target variable). The probability measures considered are the logarithmic loss, squared error, and calibration. The closer the measure is to zero the better. For the multiclass case, we computed the logarithmic loss as described by Witten and Frank (2005) and the squared error as described by Yeung et al. (2005). For calibration, we used the CAL score that was developed by Caruana and Alexandru (2004) and is based on reliability diagrams. The probability measures indicate how well probability predictions correspond to reality. For example, consider a subset C of test instances in which target outcome is predicted to be positive with probability p. If a fraction p of C actually has a positive outcome, then such performance will contribute toward the probability measures being low. A brief description of the measures is given in Table 1.

Performance measure	Range	Best score
Misclassification error	[0, 1]	0
Area under the ROC curve (AUC)	[0, 1]	1
Logarithmic loss	$[0, \infty)$	0
Squared error	[0, 1]	0
Calibration score (CAL)	[0, 1]	0

Table 1: Brief description of the performance measures used in evaluation of the performance of the algorithms.

6.3 Comparison Algorithms

The performance of the instance-specific algorithms was compared to the following methods: nave Bayes (NB), C4.5 decision tree (DT), logistic regression (LR), neural networks (NN), *k*-Nearest Neighbor (*k*NN), Lazy Bayesian Rules (LBR), and AdaBoost (AB). The first four are representative

population-wide methods, the next two are examples of instance-specific methods, and AB is an ensemble method. *k*NN is a similarity-based method. The LBR algorithm induces a rule tailored to the features of the test instance that is then used to classify it, and is an example of a model-based instance-specific method that performs model selection. For all the seven comparison methods, we used the implementations in the Weka software package (version 3.4.3) (Witten and Frank, 2005). We used the default settings provided in Weka for NB, DT, and LR. For NN, we set the number of hidden nodes to $(n+c)/2$ where n is the number of predictor variables and c is the number of classes, the learning rate to 0.3 and the momentum to 0.2 (these are the default settings in Weka) and the number of iterations to 1000 since this setting resulted in slightly better performance than the default setting of 500. For *k*NN, we used the Weka setting that identifies the best value for k (i.e., the number of neighbors) by way of cross validation. For AB, we used Weka's AdaBoostM1 procedure with the decision tree J48 as the base classifier and the number of iterations set to $n/log(m)$, where n is the number of variables and m is the number of instances in the training data set. We did not perform variable selection as a pre-processing step before applying the above classification methods. However, DT, LBR and AB perform variable selection as part of the model learning procedure, while the other the methods do not.

Three versions of the ISMB algorithm were used in the experiments described later in this section, and they are listed in Table 2. The ISMB algorithm performs selective model averaging to estimate the distribution of the target variable of the instance at hand as described in Section 5. The ISMB-MS algorithm is a *model selection* version of the ISMB algorithm. It chooses the MB structure that has the highest posterior probability from those found by the ISMB algorithm in the two-phase search, and uses that single model to estimate the distribution of the target variable of the instance at hand. Comparing the ISMB algorithm to the ISMB-MS algorithm measures the effect of approximating selective model averaging by using model selection. When the training data set is large the performance of the ISMB algorithm and the ISMB-MS algorithm may be similar if a single model with a relatively large posterior probability overwhelms the contributions of the remaining models during model averaging.

Acronym	Algorithm	Phase 1	Phase 2	Prediction
ISMB	Instance-specific Markov blanket	Is non-instance-specific Uses greedy hill-climbing search Uses phase 1 score	Is instance-specific Uses best-first search Uses phase 2 score	By model averaging over models selected in phase 1 and phase 2
ISMB-MS	Instance-specific Markov blanket - Model Selection	Same as ISMB	Same as ISMB	Based on the highest scoring model from models found by ISMB
NISMB	Non-instance-specific Markov blanket	Same as ISMB	Is non-instance-specific Uses best-first search Uses phase 1 score	By model averaging; number of selected models is the same as in ISMB

Table 2: Three versions of the ISMB algorithm.

The NISMB algorithm is the *non-instance-specific* (i.e., population-wide) version of the ISMB algorithm. Phase 1 of the NISMB algorithm is identical to that of the ISMB algorithm. In phase 2, the NISMB algorithm accumulates the same number of MB models as the ISMB algorithm except that the models are identified on the basis of the non-instance-specific phase 1 score. Thus, the NISMB algorithm averages over the same number of models as the ISMB algorithm. Comparing the ISMB algorithm to the NISMB algorithm measures the effect of the instance-specific heuristic on the performance of model averaging.

6.4 Evaluation on a Synthetic Data Set

This section describes the evaluation of the ISMB algorithm on a small synthetic data set. The synthetic domain consists of five binary variables A, B, C, D, Z where Z is a deterministic function of the other variables:

$$Z = A \vee (B \wedge C \wedge D).$$

On such a small data set it is possible to perform model averaging over all models, and this establishes the best possible prediction performance that is attainable using MB models. The training and the test sets used in the experiments are shown in Figure 7. The training set simulates a low occurrence of $A = \mathrm{T}$ (only five out of 69 instances have $A = \mathrm{T}$), and the test set consists of three instances of $A = \mathrm{T}$ which are not present in the training set.

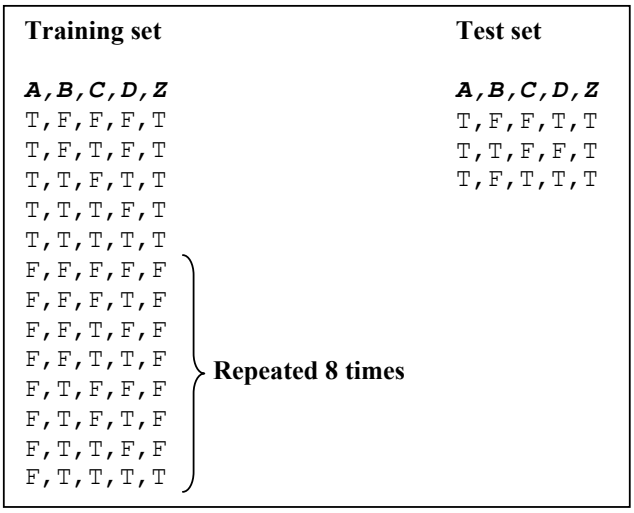


Figure 7: Training and test data sets derived from the deterministic function $Z = A \vee (B \wedge C \wedge D)$. The training set contains a total of 69 instances and the test set a total of three instances as shown; the test instances are not present in the training set. The training set simulates low prevalence of $A = \mathrm{T}$ since only five of the 69 instances have this variable-value combination.

The following algorithms were used in the experiments: (1) a complete model averaged version of the ISMB algorithm where model averaging is carried out over all 3567 possible MB structures, (2) the ISMB algorithm, (3) the ISMB-MS algorithm, and (4) the NISMB algorithm.

The settings used for the ISMB algorithm are as follows:

- Phase 1: As described in Section 5.
- Phase 2: The model score for phase 2 is computed using Equation 14 that is based on KL-divergence. Phase 2 uses best-first search with a priority queue Q whose maximum capacity w was set to 1000. Phase 2 search terminates when no MB structure in Q has a phase 2 score higher than $\varepsilon = 0.001$ for 10 consecutive iterations of the search. The maximum period of running time t for phase 2 was not specified since the algorithm terminated in a reasonable period of time with the specified value for ε.
- The predicted distribution for the target variable Z of the test instance is computed using Equation 13; for each MB structure the parameters are estimated using Equation 6.

The results are given in Table 3. All performance measures except the AUC were computed for the test set of three instances. The AUC could not be computed since all the instances in the test set are from the same class, $Z = \mathrm{T}$. The results from complete model averaging represent the best achievable expected performance that could be achieved by the ISMB algorithm. The ISMB and the NISMB algorithms that average over a subset of all models had poorer performance than complete model averaging but performed better than ISMB-MS. However, the ISMB algorithm improved over the performance of the NISMB algorithm. Though both methods average over the same number of models, the ISMB algorithm uses the instance-specific phase 2 score to choose phase 2 models while the ISMB algorithm uses the non-instance-specific phase 1 score to choose both phase 1 and phase 2 models. The phase 2 models chosen by the ISMB algorithm are potentially different for each test instance in contrast to the NISMB algorithm which selects the same models irrespective of the test instance. These results, while limited in scope, provide support that the instance-specific search for models may be able to choose models that better approximate the distribution of the target variable of the instance at hand.

Performance measure	ISMB complete model averaged	ISMB	ISMB-MS	NISMB
Misclassification error	0.0000	0.0000	0.3333	0.3333
AUC	-	-	-	-
Logarithmic loss	0.0406	0.0505	0.0596	0.0585
Squared error	0.0684	0.0783	0.0902	0.0862
CAL score	0.3720	0.4092	0.4534	0.4284

Table 3: Results obtained from the training and test sets that are given in Figure 7. The AUC could not be computed since the test set instances are all from a single class. Results in the first column are obtained by model averaging over all 3567 MBs.

Figure 8 plots the estimate of $P(Z^t = \mathrm{T}|\mathbf{x}^t, D)$ for each test instance t as it varies with each addition of a model to the set of models being averaged over. A second curve plots the model score as the logarithmic posterior probability of the model given the data; this score measures the relative contribution of the model to the final estimate of $P(Z^t = \mathrm{T}|\mathbf{x}^t, D)$. Each row in the figure contains a pair of plots for a single test instance, the plot on the left is obtained from the ISMB algorithm and

the corresponding plot on the right is obtained from the NISMB algorithm. The plot for the estimate of $P(Z^t = \mathrm{T}|\mathbf{x}^t, D)$ is shown in black while the plot for the model score is shown in gray. In each plot, on going from left to right, the estimate of $P(Z^t = \mathrm{T}|\mathbf{x}^t, D)$ initially fluctuates considerably and then settles to a stable estimate as the number of models providing the estimate increases. In the first two test instances the final estimates of $P(Z^t = \mathrm{T}|\mathbf{x}^t, D)$ obtained from the instance-specific and non-instance-specific model averaging respectively are very close; both the ISMB and the NISMB algorithms predicted the value of Z correctly as T. In the third test instance, the final estimates of $P(Z^t = \mathrm{T}|\mathbf{x}^t, D)$ are quite different; the ISMB algorithm predicted the value of Z correctly as T while the NISMB algorithm predicted the value of Z incorrectly as F.

6.5 Evaluation on UCI Data Sets

We now describe the evaluation of the ISMB algorithm on 21 data sets from the UCI Machine Learning repository (UCI data sets) (Frank and Asuncion, 2010). The selected UCI data sets have between four and 60 predictor variables and a single target variable that has between two and seven classes. The size of the data sets, the number and type of predictor variables, and the number of classes (states) taken by the target variable are given in Table 4 The performance of the ISMB algorithm is compared to that of the ISMB-MS and the NISMB algorithms, and also to that of the seven comparison machine learning methods described in Section 6.3.

6.5.1 EXPERIMENTAL DESIGN

The experimental design is as follows:

- For each data set, a total of 10 machine learning algorithms were run: ISMB, ISMB-MS, NISMB, NB, DT, LR, NN, *k*NN, LBR and AB.
- The data sets used in the experiments are the 21 UCI data sets listed in Table 4.
- Summary statistics were measured using 10-fold stratified cross-validation done twice for a total of 20 training-test pairs. The summary statistics were computed for misclassification error, the AUC, logarithmic loss, squared error and the CAL score.
- The statistical tests performed were (1) significance testing with the Wilcoxon paired-samples signed ranks test, and (2) effect size testing with paired-samples *t* test.

The settings for the ISMB algorithm are the same as those stated in Section 6.4 for the synthetic data evaluation.

6.5.2 RESULTS

Table 5 gives the average number of models selected by the ISMB and the NISMB algorithms in each of the phases for each data set. The average number of models varies from 17.99 for the iris data set (with four predictor variables) to 89.38 for the lymphography data set (with 18 predictor variables).

Tables 6 to 10 report the means of the misclassification error, the AUC, logarithmic loss, squared error and the CAL score respectively for the ISMB algorithm, its variants and the comparison algorithms. In each table, a row represents a data set and a column represents an algorithm. The last

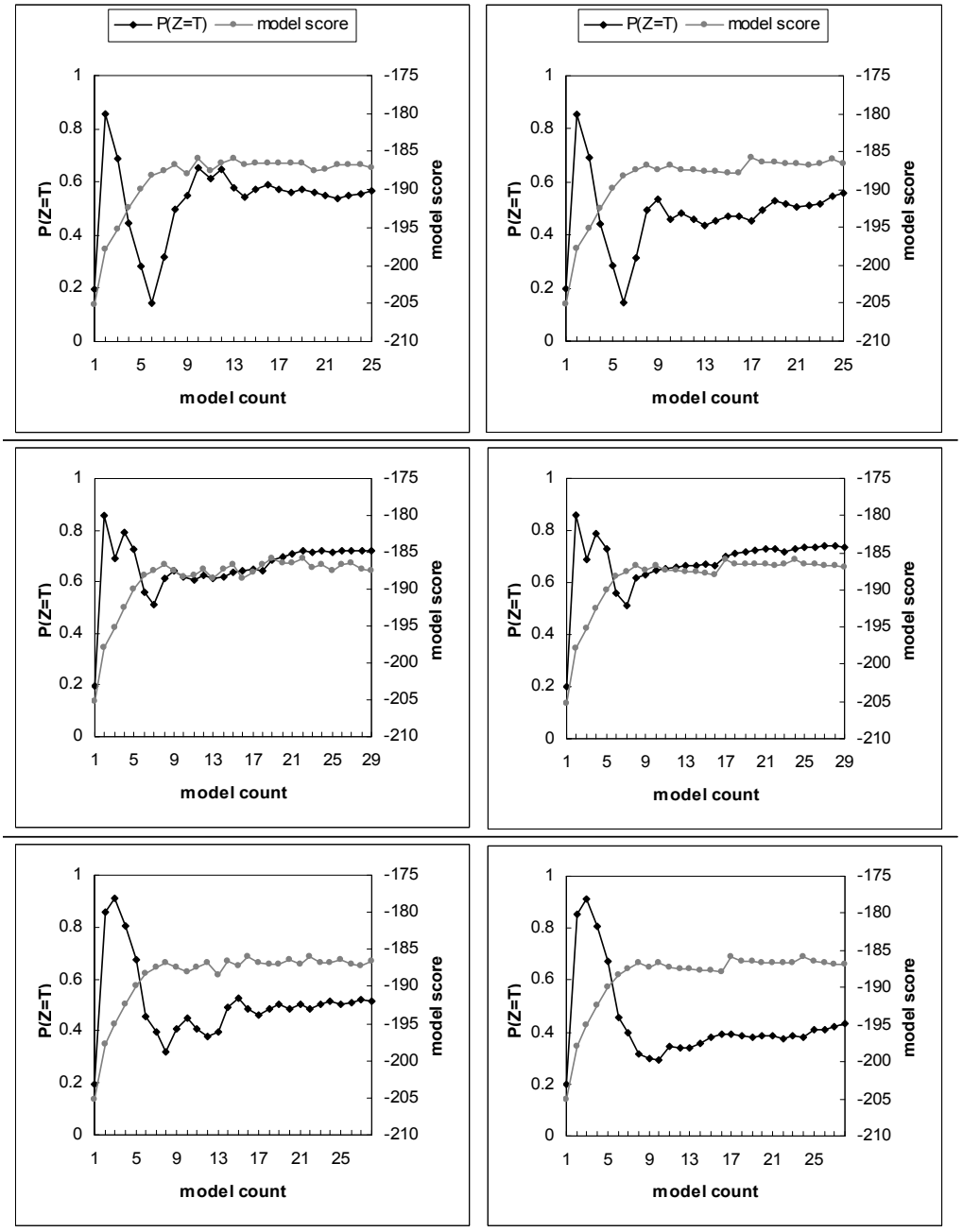


Figure 8: Plots of model averaged estimate of $P(Z^t = \mathrm{T}|\mathbf{x}^t, D)$ that is abbreviated as $P(Z = \mathrm{T})$ and model score obtained by ISMB and NISMB algorithms on the three test cases given in Figure 7. Each row represents a single test case with the plot on the left obtained from the ISMB algorithm and the plot on the right obtained from the NISMB algorithm. The value of the final averaged estimate of $P(Z^t = \mathrm{T}|\mathbf{x}^t, D)$ is the point where the darker curve meets the Y-axis on the right.

Data Set	# Predictors (cnt + dsc = total)	# Classes	# Cases
australian	6 + 8 = 14	2	690
breast-cancer	9 + 0 = 9	2	683
cleveland	6 + 9 = 13	2	296
corral	0 + 6 = 6	2	128
crx	6 + 9 = 15	2	653
diabetes	8 + 0 = 8	2	768
flare	0 + 10 = 10	2	1066
german	7 + 13 = 20	2	1000
glass2	9 + 0 = 9	2	163
glass	9 + 0 = 9	7	214
heart	13 + 0 = 13	2	270
hepatitis	6 + 13 = 19	2	80
iris	4 + 0 = 4	3	150
lymphography	0 + 18 = 18	4	148
pima	8 + 0 = 8	2	768
postoperative	1 + 7 = 8	3	87
sonar	60 + 0 = 60	2	208
vehicle	18 + 0 = 18	4	846
vote	0 + 16 = 16	2	435
wine	13 + 0 = 13	3	178
zoo	0 + 16 = 16	7	101

Table 4: Description of the 21 UCI data sets used in the experiments. In the column on predictors, the number of continuous (cnt) and discrete (dsc) predictors as well as the total number of predictor variables (excluding the target variable) are given. In the column on instances, the numbers of instances used in the experiments are given; this may be less than the total number of instances in the original UCI data set since instances with missing values were removed.

row in each table gives for each algorithm the overall mean of the specified performance measure across all the data sets. From the tables, it is seen that on all five performance measures, the ISMB algorithm achieved a better overall average score than each of the other algorithms.

Tables 11 and 12 report results from pair-wise comparisons of the performance of the algorithms on all the data sets that are aimed at assessing the statistical significance and the magnitude of the observed differences in the measures. Table 11 reports results from a two-sided Wilcoxon paired-samples signed ranks test, and Table 12 reports results from a two-sided paired-samples t test.

Table 13 reports the running times of the ISMB and the comparison algorithms. The experiments were performed on a server with 8 GB of RAM and two dual core Pentium processors of 3 GHz each that were running the Windows XP operating system. The algorithms were restricted to a single core in all the experiments. Averaged over all the data sets, the ISMB took approximately 2 minutes for a test instance.

We ran additional experiments on the first seven data sets (see Table 4) to analyze the sensitivity of the ISMB algorithm to the parameters w (queue capacity) and ε (change in Phase 2 score). For w, we evaluated values of 100, 200, 400, 800, 1600, 3200 and 6400. The performance on all the evaluation measures peaked at values of 800 or 1600 and beyond 1600 no further improvement was

Data Set	# models phase 1	# models phase 2	# models phases 1 and 2
australian	28.55	11.00	39.55
breast-cancer	18.85	10.15	29.00
cleveland	20.45	11.99	32.44
corral	10.65	15.03	25.68
crx	32.10	13.42	45.52
diabetes	11.65	10.03	21.68
flare	20.75	11.44	32.19
german	22.45	19.23	41.68
glass2	12.05	13.26	25.31
glass	15.80	10.73	26.53
heart	18.50	11.32	29.82
hepatitis	27.45	26.63	54.08
iris	7.25	10.74	17.99
lymphography	51.55	37.83	89.38
pima	40.40	16.97	57.37
postoperative	12.00	10.02	22.02
sonar	11.65	10.09	21.74
vehicle	1.15	21.09	22.24
vote	59.80	18.44	78.24
wine	39.30	10.73	50.03
zoo	45.55	13.53	59.08

Table 5: Average number of models in phases 1 and 2 over which averaging is carried out by the ISMB and NISMB algorithms. Both algorithms average over the same number of models in each phase. Both algorithms select the same models in phase 1 but potentially different models in phase 2. The number of models in phases 1 and 2 is the sum of the models selected in the two phases.

seen. For ϵ, we evaluated values of 1.0, 0.1, 0.01, 0.001, 0.0001 and 0.00001. The performance improved as ϵ decreased until 0.001 or 0.0001, but did not improve further for smaller values of ϵ.

The results are encouraging in that they show that the ISMB algorithm never underperformed on any performance measure when compared to the other learning methods including the variants of the ISMB algorithm that do model selection and non-instance-specific model averaging. For misclassification error, logarithmic loss, squared error and the CAL score, the mean difference is always negative which denotes that the ISMB algorithm always has a lower score on these measures. For the AUC, the difference is always positive which means that the ISMB algorithm always has a higher AUC. However, all mean differences are not statistically significant at the 0.05 level as can be seen by the p-values in Tables 11 and 12. The best performance is seen in logarithmic loss where the ISMB algorithm significantly outperforms all other methods, followed by squared error and CAL score where the ISMB algorithm significantly outperforms many of the methods. On misclassification error and the AUC, the ISMB algorithm has smaller performance gains.

Overall, the ISMB algorithm significantly improved on the probabilities of the predictions while maintaining or slightly improving on discrimination over all other algorithms used in the experiments. The non-instance-specific NISMB algorithm had inferior performance on logarithmic loss and squared error but similar performance on the other measures when compared to the ISMB algorithm. Both the ISMB and the NISMB algorithms average over the same number of models and

Data Set	ISMB	ISMB-MS	NISMB	NB	DT	LR	NN	*k*NN	LBR	AB
australian	0.1457	0.1457	0.1435	0.1449	0.1333	0.1486	0.1848	0.1457	0.1471	0.1453
breast-cancer	0.0256	0.0271	0.0256	0.0256	0.0403	0.0337	0.0373	0.0286	0.0256	0.0264
cleveland	0.1740	0.1791	0.1740	0.1655	0.2095	0.1655	0.1993	0.1791	0.1655	0.1985
corral	0.0000	0.0156	0.0000	0.1328	0.0508	0.1289	0.0000	0.0977	0.1250	0.0000
crx	0.1547	0.1577	0.1485	0.1348	0.1317	0.1424	0.1692	0.1485	0.1340	0.1308
diabetes	0.2116	0.2129	0.2142	0.2201	0.2194	0.2135	0.2272	0.2201	0.2207	0.2244
flare	0.1806	0.1834	0.1825	0.2012	0.1735	0.1721	0.2054	0.1806	0.1750	0.1730
german	0.2580	0.2585	0.2580	0.2445	0.2845	0.2425	0.2980	0.2695	0.2475	0.2818
glass2	0.1503	0.1564	0.1472	0.1595	0.1933	0.1442	0.1442	0.1411	0.1503	0.1503
glass	0.2150	0.2220	0.2196	0.2687	0.2500	0.2547	0.2220	0.2173	0.2500	0.2420
heart	0.1778	0.1778	0.1778	0.1630	0.1870	0.1630	0.1963	0.1741	0.1630	0.1724
hepatitis	0.0938	0.1000	0.1000	0.1375	0.1250	0.1375	0.1688	0.0688	0.1375	0.1040
iris	0.0567	0.0600	0.0633	0.0533	0.0600	0.0567	0.0633	0.0633	0.0533	0.0600
lymphography	0.1622	0.1486	0.1622	0.1486	0.2365	0.2365	0.1622	0.1622	0.1520	0.1622
pima	0.2155	0.2135	0.2142	0.2214	0.2259	0.2148	0.2389	0.2246	0.2227	0.2224
postoperative	0.3391	0.3851	0.3391	0.3103	0.2989	0.3736	0.4138	0.3333	0.3103	0.2111
sonar	0.1635	0.1659	0.1731	0.1490	0.1659	0.1442	0.1611	0.1707	0.1490	0.1742
vehicle	0.2600	0.2577	0.2612	0.3712	0.2843	0.2914	0.2825	0.2766	0.2784	0.2923
vote	0.0453	0.0582	0.0453	0.0927	0.0388	0.0733	0.0711	0.0819	0.0927	0.0438
wine	0.0084	0.0084	0.0056	0.0112	0.0702	0.0253	0.0169	0.0281	0.0112	0.0617
zoo	0.0347	0.0396	0.0347	0.0644	0.0792	0.0594	0.0495	0.0347	0.0644	0.0658
average	0.1463	0.1511	0.1471	0.1629	0.1647	0.1629	0.1672	0.1546	0.1560	0.1496

Table 6: Mean misclassification errors of different algorithms based on 10-fold cross-validation done twice. The bottom row gives the average misclassification errors. Best results are underlined.

both select the same models in phase 1 of the search. In phase 2 of the search, while the number of selected models is the same, the two methods identify potentially different models. This provides evidence that the models selected in phase 2 by the ISMB algorithm, using instance-specific search, are able to improve the performance of the ISMB algorithm over the already good performance obtained by the NISMB algorithm. Of note, LBR, which is an instance-specific approach that performs model selection, is tied with ISMB on mean error and comes second after ISMB on AUC, but it performs more poorly on the probabilistic measures.

7. Discussion

This paper described the development and evaluation of a new approach for learning predictive models that are relevant to a single instance. The instance-specific method we developed uses MB models, carries out selective BMA to predict the outcome of interest for the instance at hand, and employs an instance-specific heuristic to locate a set of suitable models to average over. The essence of the instance-specific method lies in the model score used in phase 2 of the search. This score is sensitive to both the posterior probability of the model and the predicted distribution for the outcome variable of the instance at hand. Typically, methods that evaluate models with a score employ a score that is sensitive only to the fit of the model to the training data and not to the prediction of the outcome variable.

Data Set	ISMB	ISMB-MS	NISMB	NB	DT	LR	NN	*k*NN	LBR	AB
australian	0.9315	0.9303	0.9313	0.9200	0.9032	0.9187	0.8937	0.9092	0.9186	0.9172
breast-cancer	0.9926	0.9922	0.9925	0.9933	0.9613	0.9879	0.9818	0.9930	0.9933	0.9802
cleveland	0.9098	0.9079	0.9084	0.9141	0.7952	0.9089	0.8781	0.8995	0.9141	0.8350
corral	1.0000	0.9997	1.0000	0.9252	0.9916	0.9459	1.0000	0.9827	0.9373	0.9932
crx	0.9303	0.9280	0.9302	0.9301	0.9087	0.9138	0.9002	0.9057	0.9302	0.9140
diabetes	0.8468	0.8468	0.8466	0.8438	0.7991	0.8439	0.8311	0.8148	0.8423	0.8004
flare	0.7289	0.7288	0.7261	0.7557	0.4916	0.7451	0.6445	0.6797	0.7520	0.7034
german	0.7662	0.7633	0.7641	0.7903	0.6736	0.7839	0.7340	0.7442	0.7891	0.7911
glass2	0.8703	0.8653	0.8700	0.8769	0.7982	0.8845	0.8483	0.8384	0.8826	0.8744
glass	0.9364	0.9361	0.9361	0.9408	0.8834	0.9101	0.9241	0.9112	0.9434	0.9488
heart	0.9055	0.9049	0.9073	0.9106	0.8239	0.9032	0.8649	0.8791	0.9106	0.8332
hepatitis	0.9225	0.9262	0.9237	0.9013	0.8203	0.7784	0.8436	0.8792	0.8970	0.9004
iris	0.9890	0.9900	0.9905	0.9938	0.9629	0.9846	0.9785	0.9886	0.9938	0.9808
lymphography	0.9139	0.9156	0.9173	0.9193	0.7741	0.8571	0.9192	0.9087	0.9175	0.8830
pima	0.8431	0.8424	0.8424	0.8450	0.7977	0.8456	0.8237	0.8134	0.8449	0.8284
postoperative	0.5026	0.4943	0.4538	0.5035	0.4228	0.4515	0.4113	0.3665	0.5035	0.4975
sonar	0.9203	0.9204	0.9217	0.9343	0.8521	0.9275	0.9331	0.9132	0.9345	0.9142
vehicle	0.9234	0.9228	0.9235	0.8655	0.8761	0.9016	0.8931	0.9032	0.9109	0.8965
vote	0.9875	0.9850	0.9854	0.9684	0.9578	0.9582	0.9871	0.9735	0.9660	0.9498
wine	0.9994	0.9994	0.9994	1.0000	0.9660	0.9967	0.9994	0.9981	1.0000	1.0000
zoo	0.9994	0.9992	0.9992	0.9989	0.9565	0.9967	0.9916	0.9995	0.9989	0.9622
average	0.8962	0.8952	0.8938	0.8919	0.8293	0.8783	0.8705	0.8715	0.8943	0.8764

Table 7: Mean AUCs of different algorithms based on 10-fold cross-validation done twice. The bottom row gives the average AUCs. Best results are underlined.

The experimental results demonstrate that the ISMB algorithm improves prediction of the target variable on a variety of performance measures when compared to several population-wide predictive algorithms. The greatest improvements occur in logarithmic loss and squared error, followed by good improvement in calibration and smaller improvements in misclassification error and the AUC. BMA had better performance than Bayesian model selection, and within model averaging, instance-specific BMA had better performance than non-instance-specific BMA though the improvement is not as large as that of model averaging over model selection. The improved performance by ISMB may arise from not only the model averaging but also from the variable selection that is performed implicitly by the Markov blanket models. Both these components likely explain the better performance of ISMB over comparison methods such as NB, LR and *k*NN that do not perform variable selection. However, the superiority of ISMB over ISMB-MS suggests that model averaging is an important component in the improved performance of the former. We have also evaluated ISMB on several medical data sets and obtained good results (Visweswaran et al., 2010).

Several situations are possible where the instance-specific method has no advantage over a population-wide method. As one example, in a domain where complete BMA is tractable and model averaging is carried out over all models in the model space, a search heuristic that selects a subset of models such as the one used by the instance-specific method is superfluous. Typically, in real life domains, complete BMA over all models is not tractable due to the enormous number of models in the model space. Thus, the ISMB algorithm is useful for selective model averaging where it identifies a potentially relevant set of models that is predictive of the instance at hand. As another

Data Set	ISMB	ISMB-MS	NISMB	NB	DT	LR	NN	*k*NN	LBR	AB
australian	0.3390	0.3456	0.3417	0.4476	0.4091	0.7136	0.4263	0.8627	0.4482	0.3850
breast-cancer	0.1068	0.1138	0.1083	0.2497	0.2955	0.1485	0.1205	0.2138	0.2497	0.2772
cleveland	0.3925	0.4067	0.4021	0.4491	1.3001	0.6500	0.4584	0.8625	0.4491	0.7044
corral	0.1018	0.1101	0.0989	0.3326	0.1475	0.2753	0.1542	0.0175	0.3130	0.1280
crx	0.3451	0.3564	0.3525	0.4113	0.3783	0.9377	0.4678	0.8747	0.4018	0.3845
diabetes	0.4601	0.4606	0.4604	0.4809	0.5497	0.4588	0.6039	0.5028	0.4826	0.4478
flare	0.4282	0.4294	0.4314	0.5904	0.4879	0.4042	0.5333	0.5858	0.5182	0.4032
german	0.5331	0.5413	0.5377	0.5213	1.4604	0.5229	0.5801	1.5415	0.5221	0.7981
glass2	0.4238	0.4302	0.4246	0.4532	0.8498	0.4154	0.8853	0.4562	0.4447	0.4530
glass	0.7112	0.7239	0.7113	0.7697	2.3005	4.0749	1.3612	0.8685	0.7264	0.9452
heart	0.3996	0.4069	0.3973	0.4560	0.6920	0.3907	0.6109	0.8483	0.4560	0.3788
hepatitis	0.2396	0.2517	0.2583	0.4247	0.6122	17.7871	0.3562	0.6253	0.4272	0.2548
iris	0.1560	0.1909	0.1620	0.1621	0.5287	0.7579	0.5770	0.2240	0.1621	0.1712
lymphography	0.4100	0.4289	0.4430	0.4282	2.9112	21.6371	0.5765	0.7272	0.4409	0.7422
pima	0.4647	0.4657	0.4657	0.4793	0.5268	0.4572	0.5873	0.5114	0.4774	0.4880
postoperative	0.7381	0.7776	0.7287	0.7953	1.1395	2.8236	1.3339	1.9418	0.7953	0.9453
sonar	0.3573	0.3726	0.3743	0.4573	1.2814	0.5762	0.4170	0.5728	0.4554	0.4344
vehicle	0.5863	0.5900	0.5866	1.8645	2.3842	3.9997	1.0134	1.2590	0.7815	1.0042
vote	0.1393	0.1635	0.1588	0.6804	0.3028	5.5427	0.3171	0.2782	0.5629	0.1562
wine	0.0418	0.0402	0.0367	0.0303	0.8270	0.9593	0.1032	0.0409	0.0303	0.0531
zoo	0.1297	0.1202	0.1268	0.1474	1.1102	0.5325	0.0596	0.1595	0.1474	0.1130
average	0.3573	0.3679	0.3622	0.5063	0.9759	3.0507	0.5497	0.6654	0.4425	0.4604

Table 8: Mean logarithmic losses of different algorithms based on 10-fold cross-validation done twice. The bottom row gives the average logarithmic losses. Best results are underlined.

example, in a domain where features that are relevant are commonly present, selection of relevant variables may not be a problem. In such a situation, the variables selected by a population-wide method are likely to be relevant for predicting any future instance and the instance-specific method that performs model selection will likely select the same set of variables for each new instance.

Improvements in the phase 1 search may make the phase 2 search relatively less contributory to the overall performance. We believe that the greedy hill climbing approach used in phase 1 of ISMB serves as a useful starting point for investigating this algorithm. Nonetheless, such an approach may become trapped in local maxima, leading it to miss finding highly probable MB structures. To explore this issue a number of search strategies that augment local greedy search that have been successfully applied to learning BN structures can be tried, such as best-first search (Neapolitan, 2003), simulated annealing (Heckerman et al., 1995), tabu lists (Friedman et al., 1999), random restarts to escape the numerous local optima (Heckerman et al., 1995), and optimal reinsertion (Moore and Wong, 2003). Algorithms that have been developed specifically for learning MBs such as the Markov Blanket Bayesian Classifier (MBBC) (Madden, 2002a), HITON (Aliferis et al., 2003), the Incremental Association Markov Blanket (IAMB) (Tsamardinos and Aliferis, 2003), and the Min-Max Markov Blanket algorithm (MMMB) (Tsamardinos et al., 2006) are additional candidates for consideration. Investigating the use of such alternative search methods in phase 1 is an interesting open problem.

There are several open questions regarding the behavior of the instance-specific method. Characterizing theoretically the bias of the selective model averaged prediction of the instance-specific

Data Set	ISMB	ISMB-MS	NISMB	NB	DT	LR	NN	*k*NN	LBR	AB
australian	0.2054	0.2082	0.2060	0.2234	0.2066	0.2116	0.3062	0.2287	0.2287	0.2075
breast-cancer	0.0440	0.0449	0.0441	0.0474	0.0731	0.0542	0.0689	0.0484	0.0474	0.0622
cleveland	0.2433	0.2499	0.2462	0.2553	0.3516	0.2339	0.3364	0.2526	0.2553	0.3314
corral	0.0352	0.0463	0.0354	0.2056	0.0887	0.1836	0.0038	0.1051	0.1951	0.0787
crx	0.2081	0.2146	0.2087	0.2092	0.1965	0.2121	0.2948	0.2363	0.2078	0.1987
diabetes	0.2978	0.2981	0.2979	0.3073	0.3219	0.2978	0.3156	0.3315	0.3086	0.2978
flare	0.2619	0.2626	0.2652	0.3145	0.2846	0.2513	0.3203	0.2843	0.2700	0.2498
german	0.3526	0.3570	0.3555	0.3419	0.4196	0.3368	0.5104	0.3591	0.3433	0.4050
glass2	0.2469	0.2513	0.2468	0.2450	0.3116	0.2409	0.2572	0.2603	0.2393	0.2572
glass	0.3609	0.3635	0.3605	0.3823	0.4186	0.4363	0.4075	0.3880	0.3673	0.3857
heart	0.2444	0.2486	0.2420	0.2570	0.3113	0.2394	0.3273	0.2611	0.2570	0.2565
hepatitis	0.1410	0.1495	0.1534	0.2079	0.2170	0.2750	0.2579	0.1481	0.2090	0.1483
iris	0.0727	0.0828	0.0753	0.0751	0.1122	0.0942	0.1032	0.1086	0.0751	0.0834
lymphography	0.2391	0.2353	0.2433	0.2344	0.4162	0.4545	0.2687	0.2650	0.2406	0.2388
pima	0.3009	0.3011	0.3011	0.3065	0.3264	0.2968	0.3248	0.3332	0.3060	0.3130
postoperative	0.4772	0.5044	0.4748	0.4894	0.4525	0.6011	0.7221	0.6168	0.4894	0.4512
sonar	0.2349	0.2391	0.2369	0.2411	0.2887	0.2228	0.2764	0.2402	0.2405	0.2614
vehicle	0.3471	0.3481	0.3470	0.5805	0.4171	0.4109	0.4672	0.3934	0.4059	0.3815
vote	0.0788	0.0903	0.0810	0.1681	0.0703	0.1461	0.1172	0.1293	0.1529	0.0911
wine	0.0183	0.0158	0.0142	0.0191	0.1268	0.0503	0.0213	0.0407	0.0191	0.0255
zoo	0.0612	0.0652	0.0630	0.0860	0.1415	0.0991	0.0568	0.0406	0.0860	0.0877
average	0.2129	0.2179	0.2142	0.2475	0.2644	0.2547	0.2745	0.2415	0.2354	0.2292

Table 9: Mean squared errors of different algorithms based on 10-fold cross-validation done twice. The bottom row gives the average squared errors. Best results are underlined.

method is an open problem. In contrast, the bias of selective BMA over models that are chosen randomly is low. However, the variance of selective BMA over models that are chosen randomly is likely to be much larger than the variance of selective BMA over models chosen by the instance-specific method which is constrained to prefer models that are good fit to the training data. The results here support that as a practical matter ISMB is attaining a good balance between bias and variance.

The experimental work presented in this paper is a first step in exploring the utility of the instance-specific framework, and several directions of future work are possible. The computation of the phase 2 score (see Equation 14) requires a dissimilarity metric to compare the predictive distributions of the target variable in candidate MB structures. The current implementation of the ISMB algorithm uses KL divergence as the dissimilarity metric. The experimental results indicate that KL divergence optimizes most logarithmic loss and the largest improvement in performance is observed on this measure. Alternative dissimilarity metrics that may optimize other performance measures are worth exploring.

Data Set	ISMB	ISMB-MS	NISMB	NB	DT	LR	NN	*k*NN	LBR	AB
australian	0.0470	0.0459	0.0454	0.0775	0.0463	0.0440	0.0526	0.1423	0.0817	0.0454
breast-cancer	0.0146	0.0146	0.0144	0.0200	0.0261	0.0155	0.0114	0.0299	0.0200	0.0210
cleveland	0.0497	0.0630	0.0569	0.0930	0.0690	0.0295	0.0432	0.1543	0.0930	0.0632
corral	0.0583	0.0656	0.0561	0.0470	0.0505	0.0473	0.0162	0.0115	0.0368	0.0516
crx	0.0452	0.0518	0.0503	0.0711	0.0440	0.0394	0.0722	0.1354	0.0689	0.0430
diabetes	0.0403	0.0401	0.0411	0.0618	0.0633	0.0433	0.0813	0.0662	0.0590	0.0400
flare	0.0551	0.0546	0.0562	0.1260	0.0467	0.0414	0.0762	0.1000	0.0707	0.0404
german	0.0684	0.0696	0.0699	0.0625	0.1038	0.0504	0.0547	0.2363	0.0645	0.0942
glass2	0.0359	0.0395	0.0373	0.0644	0.0386	0.0322	0.0482	0.0561	0.0569	0.0349
glass	0.0188	0.0189	0.0186	0.0282	0.0223	0.0262	0.0258	0.0246	0.0241	0.0232
heart	0.0498	0.0585	0.0513	0.0913	0.0641	0.0321	0.0624	0.1385	0.0913	0.0524
hepatitis	0.0422	0.0294	0.0381	0.0488	0.0306	0.0462	0.0197	0.0492	0.0466	0.0288
iris	0.0110	0.0115	0.0114	0.0132	0.0188	0.0142	0.0219	0.0205	0.0132	0.0144
lymphography	0.0226	0.0259	0.0256	0.0326	0.0279	0.0863	0.0272	0.0512	0.0359	0.0269
pima	0.0532	0.0539	0.0539	0.0596	0.0660	0.0444	0.0960	0.0805	0.0586	0.0588
postoperative	0.0404	0.0358	0.0438	0.0436	0.0450	0.0707	0.0844	0.1175	0.0436	0.0430
sonar	0.0437	0.0656	0.0643	0.1042	0.0591	0.0814	0.0503	0.1336	0.1045	0.0535
vehicle	0.0479	0.0481	0.0480	0.1272	0.0654	0.0632	0.0567	0.0984	0.0690	0.0543
vote	0.0247	0.0285	0.0306	0.0722	0.0227	0.0520	0.0603	0.0346	0.0658	0.0235
wine	0.0062	0.0043	0.0054	0.0083	0.0247	0.0154	0.0256	0.0133	0.0083	0.0103
zoo	0.0065	0.0067	0.0069	0.0078	0.0094	0.0055	0.0029	0.0075	0.0078	0.0067
average	0.0372	0.0396	0.0393	0.0600	0.0450	0.0419	0.0471	0.0810	0.0533	0.0395

Table 10: Mean CAL scores of different algorithms based on 10-fold cross-validation done twice. The bottom row gives the average CAL scores. Best results are underlined.

Performance measure	ISMB-MS	NISMB	NB	DT	LR	NN	*k*NN	LBR	AB
Misclassification	-2.338	-0.776	-2.121	-2.070	-1.181	-3.861	-0.825	-0.368	-1.720
error	0.019	0.438	0.034	0.038	0.238	<0.001	0.409	0.713	0.085
AUC	-2.085	-1.257	-1.511	-4.457	-2.197	-4.029	-4.203	-0.927	-3.198
	0.037	0.209	0.131	0.001	0.028	0.001	0.001	0.354	0.001
Logarithmic	-3.595	-2.426	-4.280	-4.457	-3.340	-4.254	-4.026	-4.051	-3.215
loss	0.001	0.015	0.001	0.001	0.001	0.001	0.001	0.001	0.001
Squared error	-3.608	-2.313	-3.975	-3.24	-2.121	-4.127	-3.518	-3.213	-2.839
	0.001	0.021	0.001	0.001	0.034	0.001	0.001	0.001	0.005
CAL score	-2.032	-1.867	-4.026	-2.806	-0.063	-4.076	-1.892	-3.543	-1.443
	0.042	0.062	0.001	0.005	0.949	0.001	0.058	0.001	0.149

Table 11: Two-sided Wilcoxon paired-samples signed ranks test comparing the performance of ISMB with other algorithms. For each performance measure the number on top is the Z statistic and the number at the bottom is the corresponding p-value. The Z statistic is negative when ISMB has a lower score on a performance measure than the competing algorithm. On all measures, a negative Z statistic indicates better performance by ISMB. Underlined results indicate p-values of 0.05 or smaller.

Performance measure	ISMB-MS	NISMB	NB	DT	LR	NN	*k*NN	LBR	AB
Misclassification error	-0.004 0.077	-0.001 0.312	-0.021 0.014	-0.013 0.021	-0.012 0.065	-0.019 <0.001	-0.005 0.334	-0.007 0.289	-0.003 0.258
AUC	-0.001 0.077	-0.002 0.242	-0.001 0.975	-0.104 <0.001	-0.017 0.022	-0.032 <0.001	-0.023 <0.001	-0.001 0.932	-0.020 0.001
Logarithmic loss	-0.009 0.001	-0.004 0.026	-0.163 0.005	-0.211 <0.001	-0.215 0.006	-0.306 <0.001	-0.140 <0.001	-0.071 <0.001	-0.103 0.002
Squared error	-0.004 0.003	-0.001 0.054	-0.044 0.002	-0.044 <0.001	-0.034 0.009	-0.062 0.002	-0.023 <0.001	-0.019 0.017	-0.016 0.007
CAL score	-0.003 0.044	-0.002 0.058	-0.033 <0.001	-0.011 0.018	-0.003 0.441	-0.047 <0.001	-0.008 0.079	-0.016 0.001	-0.002 0.237

Table 12: Two-sided paired-samples t test comparing the performance of ISMB with other algorithms. For each performance measure the number on top is the mean difference between ISMB and the indicated algorithm and the number at the bottom is the corresponding p-value. The mean difference is negative when ISMB has a lower score on a performance measure than the competing algorithm. On all measures, a negative mean difference indicates better performance by ISMB. Underlined results indicate p-values of 0.05 or smaller.

Algorithm	Average running time
NB	$<$ 1 second
DT	$<$ 1 second
LR	$<$ 1 second
NN	$<$ 1 second
kNN	$<$ 1 second
LBR	$\approx$ 1 second
AB	$<$ 1 second
ISMB	$\approx$ 2 minutes

Table 13: Approximate running times of the various algorithms. For each algorithm, the time shown is the average running time over all the UCI data sets. For the instance-specific algorithms LBR and ISMB the reported running time is for a single test instance, while for the other algorithms the reported running time is over all test instances.

Acknowledgments

This work was supported by a grant from the National Library of Medicine (NLM R01-LM008374) and a training grant from the National Library of Medicine to the University of Pittsburgh's Biomedical Informatics Training Program (T15-LM007059).

References

D. W. Aha. Feature weighting for lazy learning algorithms. In L. Huan and M. Hiroshi, editors, *Feature Extraction, Construction and Selection: A Data Mining Perspective*, pages 13–32. Kluwer Academic Publisher, Norwell, MA, 1998.

C. F. Aliferis, I. Tsamardinos, and A. Statnikov. Hiton: A novel markov blanket algorithm for optimal variable selection. In *Proceedings of the American Medical Informatics Association (AMIA) Annual Symposium*, pages 21–5, 2003.

C. F. Aliferis, A. Statnikov, I. Tsamardinos, S Mani, and X. D. Koutsoukos. Local causal and markov blanket induction for causal discovery and feature selection for classification part i: Algorithms and empirical evaluation. *Journal of Machine Learning Research*, 11(Jan):171–234, 2010a.

C. F. Aliferis, A. Statnikov, I. Tsamardinos, S Mani, and X. D. Koutsoukos. Local causal and markov blanket induction for causal discovery and feature selection for classification part ii: Analysis and extensions. *Journal of Machine Learning Research*, 11(Jan):235–284, 2010b.

C. G. Atkeson, A. W. Moore, and S. Schaal. Locally weighted learning. *Artificial Intelligence Review*, 11(1-5):11–73, 1997.

R. Caruana and N-M. Alexandru. Data mining in metric space: An empirical analysis of supervised learning performance criteria. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 69–78, Seattle, WA, 2004. ACM Press.

J. Cerquides and R. Mantaras. Robust bayesian linear classifier ensembles. In *Machine Learning: ECML 2005*, volume 3720 of *Lecture Notes in Computer Science*, pages 72–83. Springer Berlin / Heidelberg, 2005.

G. F. Cooper and E. Herskovits. A bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4):309–347, 1992.

T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

T. M. Cover and A. T. Joy. *Elements of Information Theory*. Wiley-Interscience, 2nd edition, 2006.

B. Dasarathy. *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. IEEE Computer Society Press, Los Alamitos, California, 1991.

D. Dash and G. F. Cooper. Exact model averaging with naive bayesian classifiers. In C. Sammut and A. Hoffmann, editors, *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 91–98, Sydney, Australia, 2002. Morgan Kaufmann.

D. Dash and G. F. Cooper. Model averaging for prediction with discrete bayesian networks. *Journal of Machine Learning Research*, 5(Sep):1177–1203, 2004.

U. M. Fayyad and K. B. Irani. Multi-interval discretization of continuous-valued attributes for classification. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1022–1027, Chambry, France, 1993. Morgan Kaufmann.

A. Frank and A. Asuncion. Uci machine learning repository, 2010. URL `http://archive.ics.uci.edu/ml`.

J. H. Friedman, R. Kohavi, and Y. Yun. Lazy decision trees. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 717–724, Portland, Oregon, 1996. AAAI Press.

N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29 (2-3):131–163, 1997.

N. Friedman, I. Nachman, and D. Pe'er. Learning bayesian network structure from massive datasets: The 'sparse-candidate' algorithm. In K. B. Laskey and H. Prade, editors, *Proceedings of the Fifteenth Annual Conference in Uncertainty in Artificial Intelligence*, pages 206–215, Stockholm, Sweden, 1999. Morgan Kaufmann.

S. Fu and M. Desmarais. Tradeoff analysis of different markov blanket local learning approaches. In *PAKDD'08: Proceedings of the 12th Pacific-Asia conference on Advances in knowledge discovery and data mining*, pages 562–571, Berlin, Heidelberg, 2008. Springer-Verlag. ISBN 3-540-68124-8, 978-3-540-68124-3.

C. Gottrup, K. Thomsen, P. Locht, O. Wu, A. G. Sorensen, W. J. Koroshetz, and L. Ostergaard. Applying instance-based techniques to prediction of final outcome in acute stroke. *Artificial Intelligence in Medicine*, 33(3):223–236, 2005.

D. J. Hand and R. J. Till. A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine Learning*, 45(2):171–186, 2001.

D. Heckerman. A tutorial on learning with bayesian networks. In M. Jordan, editor, *Learning in Graphical Models*. MIT Press, Cambridge, MA, 1999.

D. Heckerman, D. Geiger, and D. M. Chickering. Learning bayesian networks - the combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243, 1995.

J. A. Hoeting, D. Madigan, A. E. Raftery, and C. T. Volinsky. Bayesian model averaging: A tutorial. *Statistical Science*, 14(4):382–401, 1999.

K. B. Hwang and B. T. Zhang. Bayesian model averaging of bayesian network classifiers over multiple node-orders: application to sparse datasets. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 35(6):1302–10, 2005.

R. Kohavi. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In E. Simoudis, J. Han, and U. Fayyad, editors, *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 202–207, Portland, Oregon, 1996. AAAI Press.

D. Koller and M. Sahami. Toward optimal feature selection. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pages 284–292, 1996.

M. G. Madden. A new bayesian network structure for classification tasks. In *AICS '02: Proceedings of the 13th Irish International Conference on Artificial Intelligence and Cognitive Science*, pages 203–208, London, UK, 2002a. Springer-Verlag.

M. G. Madden. Evaluation of the performance of the markov blanket bayesian classifier algorithm. *CoRR*, cs.LG/0211003, 2002b.

D. Madigan and A. E. Raftery. Model selection and accounting for model uncertainty in graphical models using occam's window. *Journal of the American Statistical Association*, 89:1335–1346, 1994.

D. Margaritis and S. Thrun. Bayesian network induction via local neighborhoods. In S. A. Solla, T. K. Leen, and K.-R. Mller, editors, *Proceedings of the 1999 Conference on Advances in Neural Information Processing Systems*, Denver, CO, 1999. MIT Press.

T. P. Minka. Bayesian model averaging is not model combination. Technical report, MIT Media Lab, 2002.

A. Moore and W. K. Wong. Optimal reinsertion: A new search operator for accelerated and more accurate bayesian network structure learning. In T. Fawcett and N. Mishra, editors, *Proceedings of the 20th International Conference on Machine Learning*, pages 552–559. AAAI Press, 2003.

R. E. Neapolitan. *Learning Bayesian Networks*. Prentice Hall, Upper Saddle River, New Jersey, 1st edition, 2003.

M. J. Pazzani. Searching for dependencies in bayesian classifiers. In D. Fisher and H. J. Lenz, editors, *Proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics*, pages 239–248, Fort Lauderdale, Florida, 1995. Springer-Verlag.

M. J. Pazzani. Constructive induction of cartesian product attributes. In L. Huan and M. Hiroshi, editors, *Feature Extraction, Construction and Selection: A Data Mining Perspective*. Kluwer Academic Publisher, Norwell, MA, 1998.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Mateo, California, 1988.

A. E. Raftery, D. Madigan, and J. A. Hoeting. Model selection and accounting for model uncertainty in linear regression models. *Journal of the American Statistical Association*, 92:179–191, 1997.

K. M. Ting, Z. Zheng, and G. I. Webb. Learning lazy rules to improve the performance of classifiers. In *Proceedings of the Nineteenth SGES International Conference on Knowledge Based Systems and Applied Artificial Intelligence*, pages 122–131, Cambridge, UK, 1999. Springer-Verlag.

I. Tsamardinos and C. Aliferis. Towards principled feature selection: Relevancy, filters and wrappers. In Christopher M. Bishop and Brendan J. Frey, editors, *Ninth International Workshop on Artificial Intelligence and Statistics*, Key West, FL, USA, 2003.

I. Tsamardinos, L. Brown, and C. Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78, 2006.

S. Visweswaran and G. F. Cooper. Instance-specific bayesian model averaging for classification. In *Proceedings of the Eighteenth Annual Conference on Neural Information Processing Systems*, Vancouver, Canada, 2004.

S. Visweswaran and G. F. Cooper. Counting markov blanket structures. Technical Report DBMI-09-12, University of Pittsburgh, 2009.

S. Visweswaran, D. C. Angus, M. Hsieh, L. Weissfeld, D. Yealy, and G. F. Cooper. Learning patient-specific predictive models from clinical data. *Journal of Biomedical Informatics*, 43(5):669–85, 2010.

L. Wasserman. Bayesian model selection and model averaging. *Journal of Mathematical Psychology*, 44(1):92–107, 2000.

I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2nd edition, 2005.

K. Y. Yeung, R. E. Bumgarner, and A. E. Raftery. Bayesian model averaging: Development of an improved multi-class, gene selection and classification tool for microarray data. *Bioinformatics*, 21(10):2394–402, 2005.

J. P. Zhang, Y. S. Yim, and J. M. Yang. Intelligent selection of instances for prediction functions in lazy learning algorithms. *Artificial Intelligence Review*, 11(1-5):175–191, 1997.

Z. J. Zheng and G. I. Webb. Lazy learning of bayesian rules. *Machine Learning*, 41(1):53–84, 2000.