Subsumption Resolution: An Efficient and Effective Technique for Semi-Naive Bayesian Learning

Fei Zheng
Geoffrey I. Webb
Pramuditha Suraweera
Liguang Zhu
Faculty of Information Technology, Monash University, Vic. 3800, Australia

Abstract. Semi-naive Bayesian techniques seek to improve the accuracy of naive Bayes (NB) by relaxing the attribute independence assumption. We present a new type of semi-naive Bayesian operation, Subsumption Resolution (SR), which efficiently identifies occurrences of the specialization-generalization relationship and eliminates generalizations at classification time. We extend SR to Near-Subsumption Resolution (NSR) to delete near–generalizations in addition to generalizations. We develop two versions of SR: one that performs SR during training, called eager SR (ESR), and another that performs SR during testing, called lazy SR (LSR). We investigate the effect of ESR, LSR, NSR and conventional attribute elimination (BSE) on NB and Averaged One-Dependence Estimators (AODE), a powerful alternative to NB. BSE imposes very high training time overheads on NB and AODE accompanied by varying decreases in classification time overheads. ESR, LSR and NSR impose high training time and test time overheads on NB. However, LSR imposes no extra training time overheads and only modest test time overheads on AODE, while ESR and NSR impose modest training and test time overheads on AODE. Our extensive experimental comparison on sixty UCI data sets shows that applying BSE, LSR or NSR to NB significantly improves both zero-one loss and RMSE, while applying BSE, ESR or NSR to AODE significantly improves zero-one loss and RMSE and applying LSR to AODE significantly improves zero-one loss. The Friedman test and Nemenyi test show that AODE with ESR or NSR have a significant zero-one loss and RMSE advantage over Logistic Regression and a zero-one loss advantage over Weka's LibSVM implementation with a grid parameter search on categorical data. AODE with LSR has a zero-one loss advantage over Logistic Regression and comparable zero-one loss with LibSVM. Finally, we examine the circumstances under which the elimination of near-generalizations proves beneficial.

Keywords: Classification, Naive Bayes, Semi-naive Bayes, Feature Selection, AODE

Accepted for publication in *Machine Learning*.

1. Introduction

Naive Bayes (NB) is a simple, computationally efficient and effective probabilistic approach to classification learning (Domingos and Pazzani, 1996; Mitchell, 1997; Lewis, 1998; Hand and Yu, 2001). It has

Machine Learning **00:** 1–43, 2011.

many desirable features including the ability to directly handle missing values in a manner that minimizes information loss, learning in a single pass through the training data, support for incremental learning and a lack of parameters, avoiding the need parameter tuning. NB is built on the assumption of conditional independence between the attributes given the class. However, violations of this conditional independence assumption can render NB's classification sub-optimal.

We present Subsumption Resolution (SR), a new type of semi-naive Bayesian operation that identifies pairs of attribute-values such that one is a generalization of the other and deletes the generalization. SR can be applied at either training time or classification time. We show that this adjustment is theoretically correct and demonstrate experimentally that it can considerably improve both zero-one loss and RMSE.

This paper provides a substantially expanded presentation of the SR technique, which was first presented in (Zheng and Webb, 2006) under the potentially misleading name *Lazy Elimination*. The major extensions to the earlier paper include–

- two new subsumption resolution techniques, Eager Subsumption Resolution (ESR) which performs SR at training time and Near Subsumption Resolution (NSR) which extends the approach to near generalizations;
- an exploration of reasons for high percentages of generalizations on three data sets;
- an investigation of the situations under which the elimination of near-generalizations appears to be beneficial;
- a study of the effect of SR on RMSE in addition to zero-one loss; and
- an empirical comparison of SR applied to NB and AODE to Logistic Regression and Weka's LibSVM implementation.

Subsumption (De Raedt, 2010a) is a central concept in Inductive Logic Programming (De Raedt, 2010b), where it is used to identify generalization-specialization relationships between clauses and to support the process of unifying clauses. In this work we use it for an alternative purpose, the efficient identification and resolution of a specific form of extreme violation of the attribute-independence assumption.

The remainder of the paper is organized as follows. NB and AODE are introduced in the following sections. Section 4 introduces the BSE technique for feature selection with NB and AODE. The theoretical

justification of SR and NSR is given in section 5. NB and AODE with SR and NSR are detailed in section 6. The computational complexities of all the variants of NB and AODE are presented in section 7. Section 8 contains a detailed analysis of the effectiveness of all NB and AODE variants. The final section presents conclusions and future directions.

2. Naive Bayes (NB)

The task of supervised classification learning algorithms is to build a classifier from a labelled training sample such that the classifier can predict a discrete class label $y \in \{c_1, \ldots, c_k\}$ for a test instance $\mathbf{x} = \langle x_1, \ldots, x_n \rangle$, where x_i is the value of the i^{th} attribute X_i and c_i is the i^{th} value of the class variable Y. The Bayesian classifier (Duda and Hart, 1973) performs classification by assigning $\mathbf{x}$ to $\operatorname{argmax}_y \mathrm{P}(y \,|\, \mathbf{x})$. From the definition of conditional probability we have

$$\mathrm{P}(y \,|\, \mathbf{x}) = \mathrm{P}(y, \mathbf{x}) / \mathrm{P}(\mathbf{x}).$$

As $\mathrm{P}(\mathbf{x})$ is invariant across values of y, we have the following.

$$\operatorname*{argmax}_y \mathrm{P}(y \,|\, \mathbf{x}) = \operatorname*{argmax}_y \mathrm{P}(y, \mathbf{x}) \tag{1}$$

Where estimates of $\mathrm{P}(y \,|\, \mathbf{x})$ are required rather than a simple classification, these can be obtained by normalization,

$$\hat{\mathrm{P}}(y \,|\, \mathbf{x}) = \hat{\mathrm{P}}(y, \mathbf{x}) / \sum_{i=1}^{k} \hat{\mathrm{P}}(c_i, \mathbf{x}), \tag{2}$$

where $\hat{\mathrm{P}}(\cdot)$ represents an estimate of $\mathrm{P}(\cdot)$.

For ease of explication, we describe NB and its variants by the manner in which each calculates the estimate $\hat{\mathrm{P}}(y, \mathbf{x})$. This estimate is then utilized with (1) or (2) to perform respectively classification or conditional probability estimation.

Naive Bayes (NB) (Kononenko, 1990; Langley et al., 1992; Langley and Sage, 1994) makes an assumption that the attributes are independent given the class and estimates $\mathrm{P}(y, \mathbf{x})$ by

$$\hat{\mathrm{P}}(y, \mathbf{x}) = \hat{\mathrm{P}}(y) \prod_{i=1}^{n} \hat{\mathrm{P}}(x_i \mid y).$$

NB is simple and computationally efficient. At training time, it generates a one-dimensional table of prior class probability estimates,

indexed by class, and a two-dimensional table of conditional attribute-value probability estimates, indexed by class and attribute-value. If all attributes have discrete values this requires only a single scan of the training data. The time complexity of calculating the estimates is O(tn), where t is the number of training examples. The resulting space complexity is O(knv), where v is the mean number of values per attribute. At classification time, to classify a single example has time complexity O(kn) using the tables formed at training time with space complexity O(knv).

NB uses a fixed formula to perform classification, and hence there is no model selection. This may minimize the variance component of a classifier's error (Hastie et al., 2001). Since it only needs to update the probability estimates when a new training instance becomes available, it is suited to incremental learning. Although the attribute independence assumption is frequently unrealistic, NB has exhibited accuracy competitive with other learning algorithms for many tasks.

3. Averaged One-Dependence Estimators (AODE)

Numerous techniques have sought to enhance the accuracy of NB by relaxing the attribute independence assumption. We refer to these as semi-naive Bayesian methods. Previous semi-naive Bayesian methods can be roughly subdivided into five groups. The first group uses a *z-dependence classifier* (Sahami, 1996), in which each attribute depends upon the class and at most z other attributes. Within this framework, NB is a 0-dependence classifier. Examples include Tree Augmented Naive Bayes (TAN) (Friedman et al., 1997), Super Parent TAN (SP-TAN) (Keogh and Pazzani, 1999), NBTree (Kohavi, 1996), Lazy Bayesian Rules (LBR) (Zheng and Webb, 2000) and Averaged One-Dependence Estimators (AODE) (Webb et al., 2005). The second group remedies violations of the attribute independence assumption by deleting strongly related attributes (Kittler, 1986; Langley, 1993; Pazzani, 1996). Backwards Sequential Elimination (BSE) (Kittler, 1986) uses a simple heuristic wrapper approach that seeks a subset of the available attributes that minimizes zero-one loss on the training set. This has proved to be beneficial in domains with highly correlated attributes. However, it has high computational overheads, especially on learning algorithms with high classification time complexity, as it applies the algorithms repeatedly until there is no accuracy improvement. Forward Sequential Selection (FSS) (Langley and Sage, 1994) uses the reverse search direction to BSE. The third group applies NB to a subset of training instances (Langley, 1993; Frank et al., 2003). Note that

the second and third groups are not mutually exclusive. For example, NBTree and LBR classify instances by applying NB to a subset of training instances, and hence they can also be categorized to the third group. The fourth group performs adjustments to the output of NB without altering its direct operation (Hilden and Bjerregaard, 1976; Webb and Pazzani, 1998; Platt, 1999; Zadrozny and Elkan, 2001; Zadrozny and Elkan, 2002; Gama, 2003). The fifth group introduces hidden variables to NB (Kononenko, 1991; Pazzani, 1996; Zhang et al., 2004; Zhang et al., 2005; Langseth and Nielsen, 2006).

Domingos and Pazzani (1996) point out that interdependence between attributes will not affect NB's zero-one loss, so long as it can generate the correct ranks of conditional probabilities for the classes. However, the success of semi-naive Bayesian methods show that appropriate relaxation of the attribute independence assumption is effective. Further, in many applications it is desirable to obtain accurate estimates of the conditional class probability rather than a simple classification, and hence mere correct ranking will not suffice.

Of the z-dependence classifier approaches to relaxing the attribute conditional independence assumption, those such as TAN, SP-TAN and AODE that restrict themselves to one-dependence classifiers readily admit to efficient computation. To avoid model selection while attaining the efficiency and efficacy of one-dependence classifiers, Averaged One-Dependence Estimators (AODE) (Webb et al., 2005) utilizes a restricted class of one-dependence estimators (ODEs) and aggregates the predictions of all qualified estimators within this class. A single attribute, called a *super parent*, is selected as the parent of all the other attributes in each ODE.

In order to avoid unreliable base probability estimates, when classifying an instance $\mathbf{x}$ the original AODE excludes ODEs with parent x_i where the frequency of the value x_i is lower than limit m=30, a widely used minimum on sample size for statistical inference purposes. However, subsequent research (Cerquides and Mántaras, 2005) reveals that this constraint actually increases error and hence the current research uses m=1.

For any attribute value x_i,

$$\mathrm{P}(y, \mathbf{x}) = \mathrm{P}(y, x_i)\mathrm{P}(\mathbf{x} \mid y, x_i).$$

This equality holds for every x_i. Therefore, for any $I \subseteq \{1, \ldots, n\}$,

$$\mathrm{P}(y, \mathbf{x}) = \frac{\sum_{i \in I} \mathrm{P}(y, x_i)\mathrm{P}(\mathbf{x} \mid y, x_i)}{|I|},$$

where $|\cdot|$ denotes the cardinality of a set.

Thus,

$$\mathrm{P}(y, \mathbf{x}) = \frac{\sum_{i:1 \le i \le n \wedge \mathrm{F}(\mathrm{x_i}) \ge \mathrm{m}} \mathrm{P}(y, x_i)\mathrm{P}(\mathbf{x} \mid y, x_i)}{|\{i : 1 \le i \le n \wedge \mathrm{F}(\mathrm{x_i}) \ge \mathrm{m}\}|}, \tag{3}$$

where $\mathrm{F}(\mathrm{x_i})$ is the frequency of attribute-value x_i in the training sample.

AODE utilizes (3) and, for each ODE, an assumption that the attributes are independent given the class and the privileged attribute value x_i, estimating $\mathrm{P}(y, \mathbf{x})$ by

$$\hat{\mathrm{P}}(y, \mathbf{x}) = \frac{\sum_{i:1 \le i \le n \wedge \mathrm{F}(\mathrm{x_i}) \ge \mathrm{m}} \hat{\mathrm{P}}(y, x_i) \prod_{j=1}^{n} \hat{\mathrm{P}}(x_j \mid y, x_i)}{|\{i : 1 \le i \le n \wedge \mathrm{F}(\mathrm{x_i}) \ge \mathrm{m}\}|}.$$

At training time AODE generates a three-dimensional table of probability estimates for each attribute-value, indexed by each other attribute-value and each class. The resulting space complexity is $\mathrm{O}(k(nv)^2)$. The time complexity of forming this table is $\mathrm{O}(tn^2)$, as an entry must be updated for every training case and every combination of two attribute-values for that case. Classification requires the tables of probability estimates formed at training time, which have space complexity $\mathrm{O}(k(nv)^2)$. The time complexity of classifying a single example is $\mathrm{O}(kn^2)$, as we need to consider each pair of qualified parent and child attributes within each class.

As AODE makes a weaker attribute conditional independence assumption than NB while still avoiding model selection, it has substantially lower bias with a very small increase in variance. Previous studies have demonstrated that it has a considerably lower bias than NB with moderate increases in variance and time complexity (Webb et al., 2005) and that AODE has a significant advantage in average error over many other semi-naive Bayesian algorithms, with the exceptions of LBR (Zheng and Webb, 2000) and SP-TAN (Keogh and Pazzani, 1999). It shares a similar level of average error with these two algorithms without the prohibitive training time of SP-TAN or test time of LBR (Zheng and Webb, 2005). When a new instance is available, like NB, it only needs to update the probability estimates. Therefore, it is also suited to incremental learning.

Dash and Cooper (2002) present Exact Model Averaging with NB to efficiently average NB predictions over all possible attribute subsets. The difference between this method and AODE is that the former is a 0-dependence classifier that uses an attribute subset in each ensembled classifier and performs model averaging over all 2^n possible attribute subsets while the latter is a 1-dependence classifier that does not exclude any attributes in any ensembled classifier and performs model averaging over n possible super parents.

4. Backwards Sequential Elimination

One approach to repairing harmful interdependencies is to remove highly correlated attributes. Backwards Sequential Elimination (BSE) (Kittler, 1986) selects a subset of attributes using leave-one-out cross validation zero-one loss as a selection criterion. Starting from the full set of attributes, BSE operates by iteratively removing successive attributes, each time removing the attribute whose elimination best reduces training set zero-one loss. This process is terminated if there is no zero-one loss improvement. BSE does not support incremental learning as it has to reselect the subset of attributes when a new training instance becomes available.

4.1. NB with BSE

NB with BSE (NB^{BSE}) selects a subset of attributes using leave-one-out cross validation zero-one loss on NB as a selection criterion and applies NB to the new attribute set. The subset of selected attributes is denoted as L. Independence is assumed among the resulting attributes given the class. Hence, NB^{BSE} estimates $\text{P}(y, \mathbf{x})$ by

$$\hat{\text{P}}(y, \mathbf{x}) = \hat{\text{P}}(y) \prod_{x \in L} \hat{\text{P}}(x \mid y).$$

At training time NB^{BSE} generates a two-dimensional table of probability estimates as NB does. As it performs leave-one-out cross validation to select the subset of attributes, it must also store the training data, with additional space complexity $\text{O}(tn)$. Keogh and Pazzani (1999) speed up the process of evaluating the classifiers by using a two-dimensional table, indexed by instance and class, to store the probability estimates, with space complexity $\text{O}(tk)$. Since k is usually much less than n, the resulting space complexity is $\text{O}(tn + knv)$. The time complexity of a single leave-one-out cross validation is reduced from $\text{O}(tkn)$ to $\text{O}(tk)$ by using the speed up strategy, and the total time complexity of attribute selection is $\text{O}(tkn^2)$, as leave-one-out cross validation will be performed at most $\text{O}(n^2)$ times. NB^{BSE} has identical time and space complexity to NB at classification time.

4.2. AODE with BSE

In the context of AODE, BSE uses leave-one-out cross validation zero-one loss on AODE as the deletion criterion, and averages the predictions of all qualified classifiers using the resulting attribute set. Because attributes play multiple roles, either parent or child, in an AODE model,

there are four types of attribute elimination for AODE (Zheng and Webb, 2007). To formalize the various attribute elimination strategies we introduce into AODE the use of a *parent* (p) and a *child* (c) set, each of which contains the set of indices of attributes that can be employed in respectively a parent or child role in AODE.

All four types of attribute elimination start with p and c initialized to the full set. The first approach, called *parent elimination* (PE), deletes attribute indexes from p, effectively deleting a single ODE at each step. The second approach, called *child elimination* (CE), deletes attribute indexes from c, effectively deleting an attribute from every ODE at each step. *Parent and child elimination* (P$\wedge$CE) (Zheng and Webb, 2006) at each step deletes the same value from both p and c, thus eliminating it from use in any role in the classifier. *Parent or child elimination* (P$\vee$CE) performs any one of the other types of attribute eliminations in each iteration, selecting the option that best reduces zero-one loss.

These four types of attribute elimination for AODE estimate $\mathrm{P}(y, \mathbf{x})$ by

$$\hat{\mathrm{P}}(y, \mathbf{x}) = \frac{\sum_{i \in p: \mathrm{F(x_i)} \geq \mathrm{m}} \hat{\mathrm{P}}(y, x_i) \prod_{j \in c} \hat{\mathrm{P}}(x_j \mid y, x_i)}{|\{i : i \in p \wedge \mathrm{F(x_i)} \geq \mathrm{m}\}|}.$$

Zheng and Webb reported that the types of attribute elimination that remove child attributes from within the constituent ODEs can significantly reduce bias and error, but only if a statistical test is employed to provide variance management[1]. In this paper, of the strategies that use child elimination, we select P$\wedge$CE, as it leads to more efficient classification. We use AODE^{BSE} to indicate AODE with P$\wedge$CE.

At training time AODE^{BSE} generates a three-dimensional table of probability estimates, as AODE does. A three-dimensional table, indexed by instance, class and attribute, is introduced to speed up the process of evaluating the classifiers, with space complexity $\mathrm{O}(tkn)$. Therefore, the resulting space complexity is $\mathrm{O}(tkn + k(nv)^2)$. Deleting attributes has time complexity of $\mathrm{O}(tkn^3)$, as a single leave-one-out cross validation is order $\mathrm{O}(tkn)$ and it is performed at most $\mathrm{O}(n^2)$ times. AODE^{BSE} has identical time and space complexity to AODE at classification time.

[1] A standard binomial sign test is used to assess whether an improvement is significant. Treating the examples for which an attribute deletion corrects a misclassification as a *win* and one for which it misclassifies a previously correct example as a *loss*, a change is accepted if the number of wins exceeds the number of losses and the probability of obtaining the observed number of wins and losses if they were equiprobable was no more than 0.05.

5. Related Attribute-Values and Subsumption Resolution

This section introduces an extreme type of interdependence between attribute values and presents adjustments for such an interdependence relationship.

5.1. The Generalization, Substitution and Duplication Relationships

One extreme type of inter-dependence between attributes results in a value of one being a generalization of a value of the other. For example, let *Gender* and *Pregnant* be two attributes. *Gender* has two values: *female* and *male*, and *Pregnant* has two values: *yes* and *no*. If *Pregnant=yes*, it follows that *Gender=female*. Therefore, *Gender=female* is a generalization of *Pregnant=yes*. Likewise, *Pregnant=no* is a generalization of *Gender=male*. We formalize this relationship as:

Definition 1. (Generalization and Specialization) *For two attribute values x_i and x_j, if* $\mathrm{P}(x_j \mid x_i) = 1.0$ *then x_j is a* generalization *of x_i and x_i is a* specialization *of x_j.*

In a special case when x_i is a generalization and specialization of x_j, x_i is a substitution of x_j.

Definition 2. (Substitution) *For two attribute values x_i and x_j, if* $\mathrm{P}(x_j \mid x_i) = 1.0$ *and* $\mathrm{P}(x_i \mid x_j) = 1.0$*, x_i is a substitution of x_j and so is x_j of x_i. For two attributes X_i and X_j, we say that X_i is a substitution of X_j if the following condition holds:*

$\forall a \exists b \mathrm{P}(x_j^b \mid x_i^a) = \mathrm{P}(x_i^a \mid x_j^b) = 1.0$*, where* $a, b \in \{1, \ldots, |X_i|\}$*, x_i^a is the a^{th} value of X_i and x_j^b is the b^{th} value of X_j.*

Definition 3. (Duplication) *For two attribute values x_i and x_j, if x_i is a substitution of x_j and $x_i = x_j$ then x_i is a duplication of x_j. For two attributes X_i and X_j, we say that X_i is a duplication of X_j if $x_i = x_j$ for all instances.*

In Table I (a), because $\mathrm{P}(X_j{=}0 \mid X_i{=}0) = 1.0$ and $\mathrm{P}(X_i{=}1 \mid X_j{=}1) = 1.0$, $X_j{=}0$ is a generalization of $X_i{=}0$ and $X_i{=}1$ is a generalization of $X_j{=}1$.

Table I (b) illustrates an example of substitution. $\mathrm{P}(X_j{=}2 \mid X_i{=}0) = 1.0$ and $\mathrm{P}(X_i{=}0 \mid X_j{=}2) = 1.0$, hence $X_j{=}2$ is a substitution of $X_i{=}0$ and so is $X_i{=}0$ of $X_j{=}2$. Likewise, $X_j{=}0$ is a substitution

Table I. Examples for Generalization, Substitution and Duplication

X_i	X_j
0	0
0	0
0	0
1	0
1	0
1	0
1	1
1	1

(a) Generalization

X_i	X_j
0	2
0	2
0	2
0	2
1	0
1	0
1	0
1	0

(b) Substitution

X_i	X_j
0	0
0	0
0	0
0	0
1	1
1	1
1	1
1	1

(c) Duplication

of X_i=1 and so is X_i=1 of X_j=0. As both X_i=0 and X_i=1 have substitutions, X_i is a substitution of X_j.

As illustrated in Table I (c), X_i is a duplication of X_j. It is interesting that the specialization-generalization relationship can be defined in terms of the definitions of Generalization, Specialization, Substitution and Duplication. A duplication is a special form of substitution. A substitution is a generalization that is also a specialization. This relationship is illustrated in Figure 1.

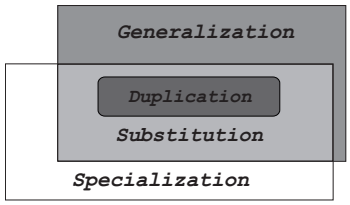


Figure 1. Relationship between Duplication, Substitution, Specialization and Generalization.

The generalization relationship is very common in the real world. For example, *City=Melbourne* is a specialization of *Country=Australia* and *CountryCode*=61 is a substitution of *Country=Australia*. Given an example with *City=Melbourne*, *Country=Australia* and *CountryCode*=61, NB will effectively give three times the weight to evidence relating to *Country=Australia* relative to the situation if only one of these attributes were considered. Ignoring such redundancy may reduce NB's zero-one loss and improve the accuracy of its probability estimates. The next section is devoted to resolving this problem.

5.2. Subsumption Resolution (SR) and Near-Subsumption Resolution (NSR)

Subsumption Resolution (SR) (Zheng and Webb, 2006) identifies pairs of attribute values such that one appears to subsume (be a generalization of) the other and deletes the generalization. Near-Subsumption Resolution (NSR) is a variant of SR. It extends SR by deleting not only generalizations but also near–generalizations.

5.2.1. *Subsumption Resolution (SR)*

Theorem. If x_j *is a generalization of* $x_i, 1 \leq i \leq n, 1 \leq j \leq n, i \neq j$ *then* $\mathrm{P}(y \mid x_1, \ldots, x_n) = \mathrm{P}(y \mid x_1, \ldots, x_{j-1}, x_{j+1}, \ldots x_n)$.

Proof. Note, $\forall Z$, given $\mathrm{P}(x_j \mid x_i) = 1.0$, it follows that $\mathrm{P}(Z \mid x_i, x_j) = \mathrm{P}(Z \mid x_i)$ and hence $\mathrm{P}(x_i, x_j, Z) = \mathrm{P}(x_i, Z)$. Therefore,

$$\begin{aligned}
&\mathrm{P}(y \mid x_1, \ldots, x_n) \\
&= \frac{\mathrm{P}(y, x_1, \ldots, x_n)}{\mathrm{P}(x_1, \ldots, x_n)} \\
&= \frac{\mathrm{P}(y, x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n)}{\mathrm{P}(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n)} \\
&= \mathrm{P}(y \mid x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n)
\end{aligned}$$

□

Given $\mathrm{P}(y \mid x_1, \ldots, x_n) = \mathrm{P}(y \mid x_1, \ldots, x_{j-1}, x_{j+1}, \ldots x_n)$ and $x_1, \ldots, x_{j-1}, x_{j+1}, \ldots x_n$ are observed, deleting the generalization x_j from a Bayesian classifier should not be harmful. Further, such deletion may improve a classifier's estimates if the classifier makes unwarranted assumptions about the relationship of x_j to the other attributes when estimating intermediate probability values, such as NB's independence assumption.

To illustrate this, consider the data presented in Table II for a hypothetical example with three attributes *Gender*, *Pregnant* and *MaleHormone* and class *Normal*. *Pregnant=yes* is a specialization of *Gender=female* and *Gender=male* is a specialization of *Pregnant=no*. As these two attributes are highly related, NB will misclassify the object ⟨*Gender=male*, *Pregnant=no*, *MaleHormone=3*⟩ as *Normal=no*, even though it occurs in the training data. In effect NB double counts the evidence from *Pregnant=no*, due to the presence of its specialization *Gender=male*. The new object can be correctly classified as *Normal=yes* by deleting attribute value *Pregnant=no*.

In contrast, if *Gender=female* we cannot make any definite conclusion about the value of *Pregnant*, nor about the value of

Table II. A Hypothetical Example

Gender	*Pregnant*	*MaleHormone*	*Normal*
male	no	3	yes
female	yes	3	yes
female	yes	2	yes
female	yes	2	yes
male	no	1	no
female	no	3	no
female	no	4	no
female	yes	4	no

Gender if *Pregnant=no*. If both of these values (*Gender=female* and *Pregnant=no*) are present, deleting either one will lose information. Therefore, if neither attribute-value is a generalization of the other, both should be used for classification. In the case when x_i is a substitution of x_j ($\mathrm{P}(x_j \mid x_i) = 1.0$ and $\mathrm{P}(x_i \mid x_j) = 1.0$), only one of the two attribute-values should be used for classification.

Note that simple attribute selection, such as BSE, cannot resolve such interdependencies, as for some test instances one attribute should be deleted, for other test instances a different attribute should be deleted, and for still further test instances no attribute should be deleted.

For a test instance $\mathbf{x} = \langle x_1, \ldots, x_n \rangle$, after SR the resulting attribute set consists of non-generalization attributes and substitution attributes. We denote the set of indices of attributes that are not a generalization of any other attributes as

$$\overline{G} = \{i \mid 1 \leq i \leq n, \neg\exists 1 \leq j \leq n, i \neq j, \mathrm{P}(x_i \mid x_j) = 1\}. \tag{4}$$

For substitutions, we keep the attribute with the smallest index and delete the other attributes. For instance, if x_1, x_3 and x_4 are substitutions of each other, we only use x_1 for classification. We denote the set of indices of resulting substitutions as

$$S = \bigcup_{i=1}^{n} min(S_i), \tag{5}$$

where

$$S_i = \{j \mid 1 \leq j \leq n, \mathrm{P}(x_i \mid x_j) = 1 \wedge \mathrm{P}(x_j \mid x_i) = 1\} \tag{6}$$

and $min(S_i)$ is $\emptyset$ if $S_i = \emptyset$ and the smallest index in S_i otherwise. The set of indices of the resulting attribute subset is $\overline{G} \bigcup S$.

SR requires a method for inferring from the training data whether one attribute value is a generalization of another. It uses the criterion

$$|T_{x_i}| = |T_{x_i,x_j}| \geq l$$

to infer that x_j is a generalization of x_i, where $|T_{x_i}|$ is the number of training cases with value x_i, $|T_{x_i,x_j}|$ is the number of training cases with both values, and l is a user-specified minimum frequency.

5.2.2. *Near-Subsumption Resolution (NSR)*
It is possible that noisy or erroneous data might prevent detection of a specialization–generalization relationship. Further, as we can only infer whether a specialization–generalization relationship exists, it is likely that in some cases we assume one does when in fact the relationship is actually a near specialization–generalization relationship. In consequence, we investigate deletions of near-generalizations as well.

Definition 4. (Near-Generalization, Near-Specialization and Near-Substitution) *For two attribute values x_i and x_j, if $\mathrm{P}(x_j \mid x_i) \geq \mathrm{P}(x_i \mid x_j)$ and $\mathrm{P}(x_j \mid x_i) \approx 1.0$, we say that x_j is a* near-generalization *of x_i and x_i is a* near-specialization *of x_j. If $\mathrm{P}(x_j \mid x_i) = \mathrm{P}(x_i \mid x_j)$ and $\mathrm{P}(x_j \mid x_i) \approx 1.0$, we say that x_j is a* near-substitution *of x_i and so is x_i of x_j.*

In this research, $\mathrm{P}(x_j \mid x_i)$ is used to estimate how approximately x_i and x_j have the specialization–generalization relationship. Let r be a user-specified lower bound, $0 \leq r \leq 1.0$. If $\hat{\mathrm{P}}(x_j \mid x_i) \geq \hat{\mathrm{P}}(x_i \mid x_j)$ and $\hat{\mathrm{P}}(x_j \mid x_i) \geq r \approx 1.0$, x_j is a near-generalization of x_i. As $\mathrm{P}(x_i, Z) - (1 - \mathrm{P}(x_j, x_i))/\mathrm{P}(x_i) \geq \mathrm{P}(x_i, x_j, Z) \leq \mathrm{P}(x_i, Z) + (1 - \mathrm{P}(x_j, x_i))/\mathrm{P}(x_i)$, when $\mathrm{P}(x_j \mid x_i) \approx 1.0$, we have $\mathrm{P}(x_i, x_j, Z) \approx \mathrm{P}(x_i, Z)$, and hence $\mathrm{P}(y \mid x_1, \ldots, x_n) \approx \mathrm{P}(y \mid x_1, \ldots, x_{j-1}, x_{j+1}, \ldots x_n)$. If an appropriate r is selected, removing x_j from a Bayesian classifier might positively affect a Bayesian classifier. However, in the absence of domain specific knowledge, there does not appear to be any satisfactory apriori method to select an appropriate value for r. Deleting weak near-generalizations might prove effective on some data sets, while only eliminating strong near-generalization may prove more desirable on other data sets. One practical approach to selecting r is to perform a parameter search, finding the value with the lowest leave-one-out cross validation zero-one loss. To provide variance management, a statistical test can be used to assess whether an zero-one loss reduction resulting from using an r value is significant.

SR can be simply extended to manipulate the near specialization–generalization relationship by using the criterion

$$|T_{x_j}| \geq |T_{x_i}| \wedge |T_{x_i,x_j}| \geq r|T_{x_i}| \wedge |T_{x_i,x_j}| \geq l$$

to infer that x_j is a near-generalization or perfect generalization (when $|T_{x_i,x_j}| = |T_{x_i}|$) of x_i.

$\overline{G}$ (Equation 4) can be extended to the set of indices of attributes $\overline{NG}$ that are not a near-generalization or perfect generalization of any other attributes by substituting $\mathrm{P}(x_i \mid x_j) \geq r$ for $\mathrm{P}(x_i \mid x_j) = 1$. S (Equation 5) can be extended to $\overline{NS}$, the set of indices of resulting near-substitutions or perfect substitutions, by substituting $\mathrm{P}(x_i \mid x_j) \geq r \wedge \mathrm{P}(x_j \mid x_i) \geq r$ for $\mathrm{P}(x_i \mid x_j) = 1 \wedge \mathrm{P}(x_j \mid x_i) = 1$ in Equation 6. This extension is called Near-Subsumption Resolution (NSR).

6. NB and AODE with SR and NSR

Attribute values that are subsumed by others can be either identified during training time or classification time. Eager learning, which identifies subsumed attribute-values during training time, transforms the data prior to training the classifier and is independent of the classification algorithm. On the other hand, lazy learning, deletes attributes at classification time based on the attribute-values that are instantiated in the instance being classified.

Although, this is suited to probabilistic techniques, such as NB and AODE, it is not suited for similarity techniques, such as k-nearest neighbours. This can be illustrated by a simple example in which there are two attributes *Pregnant* and *Gender*, the test instance is $\langle Gender{=}female,\ Pregnant{=}yes \rangle$ and the distance between two instances is defined as the number of attributes that have different values. The distance between the test instance and $\langle Gender{=}female,\ Pregnant{=}no \rangle$ is one and that of the test instance and $\langle Gender{=}male,\ Pregnant{=}no \rangle$ is two. In such a case, attribute *Gender* is important to measure the similarity and hence deleting attribute value $Gender{=}female$ of the test instance is clearly not correct, which process results in both distances equal one.

6.1. Lazy Subsumption Resolution

The lazy versions of SR (LSR) and NSR delay the computation of elimination until classification time. They delete different attributes depending upon which attribute values are instantiated in the object being classified, that is, different attributes may be used to classify different test instances. Consequently, LSR can only be applied to algorithms which can use different attributes for different test instances.

When LSR is applied to NB or AODE, the resulting classifier acts as NB or AODE except that it deletes generalization attribute-values if a specialization is detected. We denote NB and AODE with Lazy Subsumption Resolution as NB^{LSR} and AODE^{LSR} respectively. As

LSR eliminates highly dependent attribute values in a lazy manner, it does not interfere with NB and AODE's capacity for incremental learning.

Classification of instance $\mathbf{x} = \langle x_1, \ldots, x_n \rangle$ consists of two steps:

1. Set R to $\overline{G} \bigcup S$ (refer to Equation 4 and 5).
2. Estimate $\mathrm{P}(y, \mathbf{x})$ by

$$\hat{\mathrm{P}}(y, \mathbf{x}) = \begin{cases} \hat{\mathrm{P}}(y) \prod\limits_{i \in R} \hat{\mathrm{P}}(x_i \mid y) & \mathrm{NB}^{LSR} \\ \frac{\sum_{i:i \in R \wedge \mathrm{F(x_i)} \geq \mathrm{m}} \hat{\mathrm{P}}(y, x_i) \prod_{j \in R} \hat{\mathrm{P}}(x_j | y, x_i)}{|\{i:i \in R \wedge \mathrm{F(x_i)} \geq \mathrm{m}\}|} & \mathrm{AODE}^{LSR} \end{cases},$$

where $\mathrm{F(x_i)}$ is the frequency of x_i and m is the minimum frequency to accept x_i as a super parent.

NB^{LSR} generates at training time a two-dimensional table of probability estimates for each attribute-value, conditioned by each other attribute-value in addition to the two probability estimate tables generated by NB, resulting in a space complexity of $\mathrm{O}(knv + (nv)^2)$. The time complexity of forming the additional two-dimensional probability estimate table is $\mathrm{O}(tn^2)$. Classification of a single example requires considering each pair of attributes to detect dependencies and is of time complexity $\mathrm{O}(n^2 + kn)$. The space complexity is $\mathrm{O}(knv + (nv)^2)$.

AODE^{LSR} has identical time and space complexity to AODE. At training time it behaves identically to AODE. At classification time, it must check all attribute-value pairs for generalization relationships, an additional operation of time complexity $\mathrm{O}(n^2)$. However, the time complexity of AODE at classification time is $\mathrm{O}(kn^2)$ and so this additional computation does not increase the overall time complexity.

When NSR is applied to NB or AODE, R is extended to $\overline{NG} \bigcup NS$ (refer to Section 5.2.2). If r is pre-selected, this extension does not incur any additional computational complexity compared to the original LSR as it only changes the criterion to accept the relationship. However, if we perform a parameter search to select r by using leave-one-out cross validation, the training time complexity of NB^{NSR} and AODE^{NSR} will be $\mathrm{O}(tn^2 + tkn)$ and $\mathrm{O}(tkn^2)$ respectively. This also incurs an additional space complexity $\mathrm{O}(tn)$ to store the training data.

6.2. Eager Subsumption Resolution

Eager subsumption resolution (ESR) eliminates subsumed attribute-values at training time by transforming the training data. They are identified using a two-dimensional table of probability estimates for each attribute-value, conditioned by each other attribute-value. Each

attribute value that is x_j subsumed by another x_i (that is, for which $\hat{\mathrm{P}}(x_j \mid x_i) = 1$), is removed from the training data by replacing X_i and X_j with a single attribute X_iX_j with all combinations of values of X_i and X_j except for those for which $\mathrm{P}(x_i, x_j) = 0$.

The condition for merging two attributes can be relaxed further, by allowing two attributes (X_i and X_j) to be merged if they have any values x_i and x_j such that $\hat{\mathrm{P}}(x_i, x_j) = 0$. This condition is equivalent to the first, if the domain of X_j has two attribute values. On the other hand, it is more relaxed in the case where X_j has more than two values. We evaluated the effectiveness of both variants.

To illustrate the difference between these two conditions, consider the data presented in Table III, which contains three attributes *TopLeft, TopMiddle, TopRight* and class *Class*. Based on the data, *TopRight* and *TopLeft* satisfy both subsumption criteria as $\hat{\mathrm{P}}$(TopRight=x | TopLeft=x) = 1 and $\hat{\mathrm{P}}$(TopRight=o | TopLeft=x) = 0. However, *TopLeft* and *TopMiddle* only satisfy the latter criterion as $\hat{\mathrm{P}}$(TopMiddle=x | TopLeft=o) = 0, $\hat{\mathrm{P}}$(TopMiddle=o | TopLeft=o) $\neq$ 1 and $\hat{\mathrm{P}}$(TopMiddle=b | TopLeft=o) $\neq$ 1.

Table III. A Hypothetical Example

TopLeft	*TopMiddle*	*TopRight*	*Class*
x	x	x	positive
b	o	o	negative
b	x	b	negative
o	b	o	negative
o	o	b	negative
b	x	x	negative

The ESR algorithm repeatedly merges attributes until no further merges are possible. During each iteration, all attribute pairs X_i, X_j are identified that satisfy the subsumption criteria and the frequencies of all their attribute-value pairs are either 0 or greater than a predefined minimum frequency m. If multiple candidates are found, the X_i, X_j with the highest information gain ratio is merged. This process is repeated until no further X_i, X_j pairs are found.

The data transformation is implemented as a filter that is applied to the training data. The trained filter is applied to each test instance prior to classification. Thus the transformation is transparent to the classification algorithm. In the case of applying ESR to NB or AODE, the conditional probabilities are estimated based on the transformed data, and the posterior probabilities are also calculated based on the transformed data. Consequently, neither NB nor AODE requires any modifications.

At training time NB^{ESR} requires a two-dimensional table of probability estimates for each attribute-value conditioned by each other attribute-value in addition to the probability estimate tables of NB. This results in an overall space complexity of $\text{O}(knv + (nv)^2)$. The time complexity of forming this table is $\text{O}(tn^2)$. As the ESR algorithm repeatedly merges attributes, the worse case overall time complexity is $\text{O}(tn^2)$. Classification of a single example in NB^{ESR} does not have any effect on the time or space complexity of NB. In the case of AODE, the space complexity of AODE^{ESR} is identical to AODE. Its worst case training time complexity is $\text{O}(tn^2)$. The classification time and space complexities of AODE^{ESR} are identical to AODE.

7. Complexity Summary

Table IV. Computational Complexity

	Training		**Classification**	
Algorithm	**Time**	**Space**	**Time**	**Space**
NB	$\text{O}(tn)$	$\text{O}(knv)$	$\text{O}(kn)$	$\text{O}(knv)$
NB^{BSE}	$\text{O}(tkn^2)$	$\text{O}(tn + knv)$	$\text{O}(kn)$	$\text{O}(knv)$
NB^{LSR}	$\text{O}(tn^2)$	$\text{O}(knv + (nv)^2)$	$\text{O}(n^2 + kn)$	$\text{O}(knv + (nv)^2)$
NB^{ESR}	$\text{O}(tn^2)$	$\text{O}(knv + (nv)^2)$	$\text{O}(kn)$	$\text{O}(knv)$
NB^{NSR}	$\text{O}(tn^2 + tkn)$	$\text{O}(tn + knv + (nv)^2)$	$\text{O}(n^2 + kn)$	$\text{O}(knv + (nv)^2)$
AODE	$\text{O}(tn^2)$	$\text{O}(k(nv)^2)$	$\text{O}(kn^2)$	$\text{O}(k(nv)^2)$
AODE^{BSE}	$\text{O}(tkn^3)$	$\text{O}(tkn + k(nv)^2)$	$\text{O}(kn^2)$	$\text{O}(k(nv)^2)$
AODE^{LSR}	$\text{O}(tn^2)$	$\text{O}(k(nv)^2)$	$\text{O}(kn^2)$	$\text{O}(k(nv)^2)$
AODE^{ESR}	$\text{O}(tn^2)$	$\text{O}(k(nv)^2)$	$\text{O}(kn^2)$	$\text{O}(k(nv)^2)$
AODE^{NSR}	$\text{O}(tkn^2)$	$\text{O}(tn + k(nv)^2)$	$\text{O}(kn^2)$	$\text{O}(k(nv)^2)$

k is the number of classes

n is the number of attributes

t is the number of training examples

v is the mean number of values for an attribute

Table IV summarizes the complexity of each of the algorithms discussed. We display the time complexity and the space complexity of each algorithm for each of training time and classification time.

AODE^{BSE} has the highest training time complexity that is cubic in the number of attributes. However, it may in practice find parent and child sets with less computation due to the statistical test employed. NB^{BSE} has the second highest training time complexity and lowest

classification time complexity. It can efficiently classify test instances once the models are generated. The training time of all variants is linear with respect to number of training examples. When classification time is of major concern, NB^{BSE}, NB^{ESR} and NB may excel. NB^{LSR}, NB^{NSR}, AODE and all its variants have high classification time when the number of attributes is large, for example, in text classification. Nonetheless, for many classification tasks with moderate or small number of attributes, their classification time complexity is modest. AODE^{BSE} and AODE^{NSR} have relatively high training space complexity.

8. Empirical Study

To evaluate the efficacy of ESR, LSR and NSR, we compare NB and AODE with and without ESR, LSR or NSR using the bias and variance definitions of Kohavi and Wolpert (1996) together with the repeated cross-validation bias-variance estimation method proposed by Webb (2000) on sixty natural domains from the UCI Repository (Newman et al., 1998). In order to maximize the variation in the training data from trial to trial we use two-fold cross validation. We also compare these methods to NB and AODE with BSE, logistic regression (LR) and LibSVM. As we cannot obtain results of LibSVM on the two largest data sets (Covertype and Census-Income (KDD)), the comparison in Section 8.5 only includes 58 data sets.

Table V summarizes the characteristics of each data set, including the number of instances, attributes and classes. Algorithms are implemented in the Weka workbench (Witten and Frank, 2005). Experiments on algorithms, except LR and LibSVM, were executed on a 2.33GHz Intel(R) Xeon(R) E5410 Linux computer with 4Gb RAM, and those on LR and LibSVM were executed on a Linux Cluster based on Xeon 2.8GHz CPUs.

The base probabilities were estimated using m-estimation (m=0.1) (Cestnik, 1990)[2]. When we use MDL discretization (Fayyad and Irani, 1993) to discretize quantitative attributes within each cross-validation fold, many quantitative attributes have only one value. Attributes with only one value do not provide information for classification, and hence we discretize quantitative attributes using 3-bin equal frequency discretization. In order to allow the techniques to be

[2] As m-estimation often appears to lead to more accurate probabilities than Laplace estimation, this paper uses m-estimation to estimate the base probabilities. Therefore, the results presented here may differ from that of Zheng and Webb (2006, 2007) which uses Laplace estimation.

Table V. Data sets

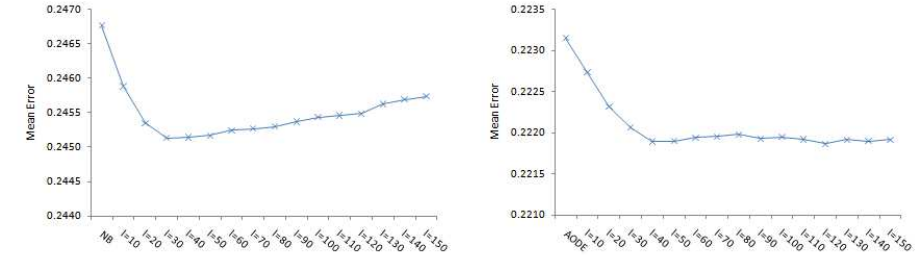
No.	Domain	Case	Att	Class	No.	Domain	Case	Att	Class
1	Abalone	4177	9	3	31	Liver Disorders (Bupa)	345	7	2
2	Adult	48842	15	2	32	Lung Cancer	32	57	3
3	Annealing	898	39	6	33	Lymphography	148	19	4
4	Audiology	226	70	24	34	MAGIC Gamma Telescope	19020	11	2
5	Auto Imports	205	26	7	35	Mushrooms	8124	23	2
6	Balance Scale	625	5	3	36	Nettalk(Phoneme)	5438	8	52
7	Breast Cancer (Wisconsin)	699	10	2	37	New-Thyroid	215	6	3
8	Car Evaluation	1728	8	4	38	Nursery	12960	9	5
9	Census-Income (KDD)	299285	40	2	39	Optical Digits	5620	49	10
10	Connect-4 Opening	67557	43	3	40	Page Blocks Classification	5473	11	5
11	Contact-lenses	24	5	3	41	Pen Digits	10992	17	10
12	Contraceptive Method Choice	1473	10	3	42	Pima Indians Diabetes	768	9	2
13	Covertype	581012	55	7	43	Postoperative Patient	90	9	3
14	Credit Screening	690	16	2	44	Primary Tumor	339	18	22
15	Echocardiogram	131	7	2	45	Promoter Gene Sequences	106	58	2
16	German	1000	21	2	46	Segment	2310	20	7
17	Glass Identification	214	10	3	47	Sick-euthyroid	3772	30	2
18	Haberman's Survival	306	4	2	48	Sign	12546	9	3
19	Heart Disease (Cleveland)	303	14	2	49	Sonar Classification	208	61	2
20	Hepatitis	155	20	2	50	Splice-junction Gene Sequences	3190	62	3
21	Horse Colic	368	22	2	51	Statlog (Shuttle)	58000	10	7
22	House Votes 84	435	17	2	52	Syncon	600	61	6
23	Hungarian	294	14	2	53	Teaching Assistant Evaluation	151	6	3
24	Hypothyroid(Garavan)	3772	30	4	54	Tic-Tac-Toe Endgame	958	10	2
25	Ionosphere	351	35	2	55	Vehicle	846	19	4
26	Iris Classification	150	5	3	56	Volcanoes	1520	4	4
27	King-rook-vs-king-pawn	3196	37	2	57	Vowel	990	14	11
28	Labor Negotiations	57	17	2	58	Waveform-5000	5000	41	3
29	LED	1000	8	10	59	Wine Recognition	178	14	3
30	Letter Recognition	20000	17	26	60	Zoo	101	17	7

compared with Weka's LR and LibSVM, missing values for qualitative attributes are replaced with modes and those for quantitative attributes are replaced with means from the training data.

8.1. Minimum Frequency for Identifying Generalizations for LSR

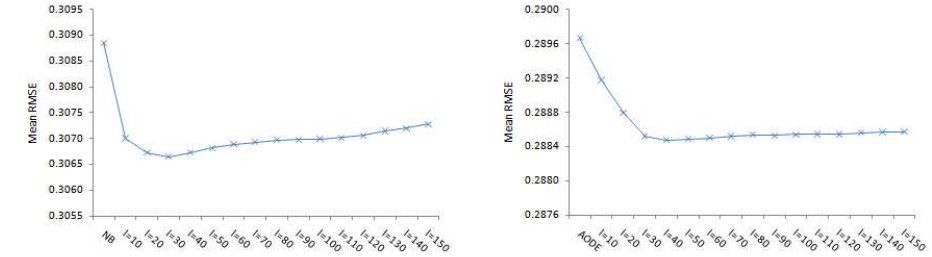
As there does not appear to be any formal method to select an appropriate value for l, we perform an empirical study to select it. We present the zero-one loss and RMSE results in the range of $l = 10$ to $l = 150$ with an increment of 10.

Mean Zero-one Loss and RMSE. Averaged results across all data sets provides a simplistic overall measure of relative performance. We present the averaged zero-one loss and RMSE of NB^{LSR} and AODE^{LSR} across 60 data sets as a function of l in Figure 2 and 3. In order to provide comparison with NB and AODE, we also include NB and AODE's zero-one loss and RMSE in each graph. For all settings of



(a) NB and NB^{LSR} with l (b) AODE and AODE^{LSR} with l

Figure 2. Averaged zero-one loss across 60 data sets, as function of l.



(a) NB and NB^{LSR} with l (b) AODE and AODE^{LSR} with l

Figure 3. Averaged RMSE across 60 data sets, as function of l.

l, NB^{LSR} and AODE^{LSR} enjoy lower mean zero-one loss and RMSE compared to NB and AODE respectively.

Table VI. Win/Draw/Loss comparison of **zero-one loss**

				NB^{LSR}				
W/D/L	l=10	l=20	l=30	l=40	l=50	l=60	l=70	l=80
NB	21/15/24	17/18/25	**13/21/26**	**8/25/27**	**7/27/26**	**6/27/27**	**6/29/25**	**5/31/24**
				NB^{LSR}				
W/D/L	l=90	l=100	l=110	l=120	l=130	l=140	l=150	
NB	**5/32/23**	**4/35/21**	**4/35/21**	**3/36/21**	**3/38/19**	**3/38/19**	**3/40/17**	
				AODE^{LSR}				
W/D/L	l=10	l=20	l=30	l=40	l=50	l=60	l=70	l=80
AODE	24/12/24	21/18/21	17/21/22	11/29/20	11/30/19	11/29/20	**9/32/19**	**8/32/20**
				AODE^{LSR}				
W/D/L	l=90	l=100	l=110	l=120	l=130	l=140	l=150	
AODE	**9/32/19**	**8/35/17**	**7/36/17**	**5/38/17**	**5/38/17**	**5/38/17**	**4/40/16**	

Table VII. Win/Draw/Loss comparison of **RMSE**

	$\mathbf{NB}^{LSR}$							
W/D/L	l=10	l=20	l=30	l=40	l=50	l=60	l=70	l=80
NB	**17/12/31**	**12/20/28**	**9/23/28**	**8/26/26**	**8/27/25**	**6/28/26**	**6/31/23**	**4/35/21**
	$\mathbf{NB}^{LSR}$							
W/D/L	l=90	l=100	l=110	l=120	l=130	l=140	l=150	
NB	**4/35/21**	**3/39/18**	**3/40/17**	**3/41/16**	**3/41/16**	**3/42/15**	**3/42/15**	
	$\mathbf{AODE}^{LSR}$							
W/D/L	l=10	l=20	l=30	l=40	l=50	l=60	l=70	l=80
AODE	22/15/23	21/18/21	13/25/22	**6/30/24**	**6/31/23**	**8/33/19**	**8/34/18**	9/33/18
	$\mathbf{AODE}^{LSR}$							
W/D/L	l=90	l=100	l=110	l=120	l=130	l=140	l=150	
AODE	9/33/18	8/36/16	8/36/16	8/37/15	**6/39/15**	**6/39/15**	**5/40/15**	

Zero-one Loss. Table VI presents the win/draw/loss records of zero-one loss for NB against NB^{LSR} and AODE against AODE^{LSR}. We assess a difference as significant if the outcome of a one-tailed binomial sign test is less than 0.05. Boldface numbers indicate that wins against losses are statistically significant. NB^{LSR} enjoys a significant advantage in zero-one loss over NB when $30 \leq l \leq 150$. The advantage of AODE^{LSR} over AODE is statistically significant when $70 \leq l \leq 150$.

RMSE. The win/draw/loss records of RMSE for NB against NB^{LSR} and AODE against AODE^{LSR} are presented in Table VII. An advantage to NB^{LSR} over NB is evident for all evaluated settings of l (ie $10 \leq l \leq 150$). AODE^{LSR} has significant RMSE advantage over AODE for $40 \leq l \leq 70$ and $130 \leq l \leq 150$. AODE^{LSR} also enjoys nearly significant zero-one loss advantage ($p < 0.1$) for $80 \leq l \leq 110$.

Minimum Frequency Selection. A larger value of l can reduce the risk of incorrectly inferring that one value subsumes another, but at the same time reduces the number of true generalizations that are detected. The setting $l = 100$ for NB^{LSR} has a significant zero-one loss and RMSE advantage over NB. It also has a significant zero-one loss advantage for AODE^{LSR}. The RMSE advantage of AODE^{LSR} with setting $l = 100$ is nearly significant ($p = 0.08$). Consequently, the setting $l = 100$ is selected in our current work.

In the earlier paper (Zheng and Webb, 2006) where Laplace estimation is employed, 30 is used as the minimum frequency because it is a widely used heuristic for the minimum number of examples from

which an inductive inference should be drawn. In fact, in other unreported experiments we have performed, when using Laplace estimation, NB^{LSR} and AODE^{LSR} have a significant zero-one loss advantage over NB and AODE respectively at all settings of l except 10 and 20. NB and AODE's RMSE can be significantly reduced by the addition of LSR at all settings of l.

8.2. Attribute Merging Criterion for ESR

As explained in Section 6.2, ESR repeatedly merges pairs of attributes that satisfy the subsumption criteria of having attribute-values x_i and x_j that either satisfy $\hat{\text{P}}(x_j \mid x_i) = 1$ or $\hat{\text{P}}(x_j, x_i) = 0$. All pairs of attributes that satisfy $\hat{\text{P}}(x_j \mid x_i) = 1$ also have an attribute-value x_k that satisfies $\hat{\text{P}}(x_k, x_i) = 0$. However, the reverse may not be true if one of the two attributes has more than two values.

We refer to the version of ESR that merges attributes if two attribute values satisfy $\hat{\text{P}}(x_j \mid x_i) = 1$ as strict ESR (ESR_s), and the other as relaxed ESR (ESR_r). The win/draw/loss records for NB_s^{ESR} and NB_r^{ESR} are given in Table VIII. The p value is the outcome of a one-tailed binomial sign test. NB_s^{ESR} significantly reduces bias and RMSE of NB at the expense of significantly higher variance. It has a nearly significant zero-one loss advantage over NB ($p = 0.09$). NB_r^{ESR} also significantly reduces the bias of NB at the expense of significantly increased variance. It performs nearly significantly better than NB in terms of zero one loss ($p = 0.06$) and RMSE ($p = 0.11$). Based on the win/draw/loss records, NB_s^{ESR} does not have any significant differences in comparison to NB_r^{ESR}.

Table VIII. Win/Draw/Loss: NB_s^{ESR}, NB_r^{ESR} vs. NB and NB_s^{ESR} vs NB_r^{ESR}

	$\textbf{NB}_s^{ESR}$ **vs. NB**		$\textbf{NB}_r^{ESR}$ **vs. NB**		$\textbf{NB}_s^{ESR}$ **vs.** $\textbf{NB}_r^{ESR}$	
	W/D/L	p	W/D/L	p	W/D/L	p
0-1 loss	14/39/7	0.09	11/45/4	0.06	9/40/11	0.41
Bias	17/39/4	**<0.001**	14/44/2	**<0.001**	12/41/7	0.18
Var	6/39/15	**0.04**	1/45/14	**<0.001**	12/41/7	0.18
RMSE	15/40/5	**0.02**	11/44/5	0.11	11/40/9	0.41

Table IX presents the win/draw/loss records of comparisons between AODE_s^{ESR} and AODE_r^{ESR}. Both AODE_s^{ESR} and AODE_r^{ESR} perform significantly better than AODE in terms of zero-one loss, bias and RMSE. The variance of AODE_r^{ESR} is nearly significantly worse than AODE. AODE_s^{ESR} has a significant RMSE advantage over AODE_r^{ESR}. Although AODE_s^{ESR} has lower zero-one loss and bias more often than AODE_r^{ESR} this is not statistically significant.

Table IX. Win/Draw/Loss: AODE_s^{ESR}, AODE_r^{ESR} vs. AODE and AODE_s^{ESR} vs AODE_r^{ESR}

	AODE_s^{ESR} vs. AODE		**AODE_r^{ESR} vs. AODE**		**AODE_s^{ESR} vs. AODE_r^{ESR}**	
	W/D/L	p	W/D/L	p	W/D/L	p
0-1 loss	15/40/5	**0.02**	15/44/1	**<0.001**	11/42/7	0.24
Bias	15/39/6	**0.04**	16/42/2	**<0.001**	11/41/8	0.32
Var	9/38/13	0.26	4/45/11	0.06	13/41/6	0.08
RMSE	17/38/5	**0.01**	14/44/2	**<0.001**	15/40/5	**0.02**

Considering that AODE_s^{ESR} has significantly lower RMSE in comparison to AODE_r^{ESR} and it has lower zero-one loss more often than AODE_r^{ESR}, we chose ESR_s for further analysis. For ease of exposition, we refer to ESR_s as ESR from here on.

8.3. Effects of BSE, LSR, NSR, ESR

In this section, we evaluate the effect of BSE, NSR, LSR and ESR on NB and AODE. As NB^{BSE} without a binomial sign test has a significant zero-one loss advantage relative to NB^{BSE} with a binomial sign test (win/draw/loss being 38/2/20), we only present the results of NB^{BSE} without a binomial sign test. In the context of AODE, the zero-one loss advantage of P∧CE with a binomial sign test relative to P∧CE without a binomial sign test is significant (win/draw/loss 37/0/23). P∧CE with a binomial sign test frequently obtains lower zero-one loss than CE, P∧CE and P∨CE with a binomial sign test. Therefore, we present the result of P∧CE with a binomial sign test, which is indicated as AODE^{BSE}.

The minimum frequency for LSR, NSR and ESR is set to $l = 100$. We select r value for NSR in the range of 0.75 to 0.99 with an increment of 0.01 by using leave-one-out cross validation. The value with the lowest cross-validation zero-one loss is selected. A binomial sign test is used to assess whether an zero-one loss reduction resulting from using a r value is significant. If the best leave-one-out cross validation zero-one loss is not significantly higher than the zero-one loss of its base learner, NSR defaults to LSR.

Table X presents the win/draw/loss records for NB^{BSE}, NB^{NSR}, NB^{LSR} and NB^{ESR} against NB on sixty data sets. The p value is the outcome of a one-tailed binomial sign test. All four improvements to NB have significant zero-one loss, bias and RMSE advantages over NB. NB^{BSE}, NB^{LSR} and NB^{NSR} have significant variance disadvantage relative to NB. The variance disadvantage of NB^{LSR} is not significant.

Table X. Win/Draw/Loss: NB^{BSE}, NB^{LSR} NB^{NSR} and NB^{ESR} vs. NB

	NB^{BSE} vs. NB		**NB^{NSR} vs. NB**		**NB^{LSR} vs. NB**		**NB^{ESR} vs. NB**	
	W/D/L	p	W/D/L	p	W/D/L	p	W/D/L	p
0-1 loss	22/34/4	**<0.001**	23/32/5	**<0.001**	21/35/4	**<0.001**	14/39/7	0.09
Bias	26/32/2	**<0.001**	26/33/1	**<0.001**	20/36/4	**<0.001**	17/39/4	**<0.001**
Var	8/33/19	**0.03**	6/33/21	**<0.001**	11/35/14	0.35	6/39/15	**0.04**
RMSE	22/32/6	**<0.001**	20/35/5	**<0.001**	18/39/3	**<0.001**	15/40/5	**0.02**

Table XI. Win/Draw/Loss: $AODE^{BSE}$, $AODE^{LSR}$, $AODE^{NSR}$ and $AODE^{ESR}$vs. AODE

	$AODE^{BSE}$ vs. AODE		**$AODE^{NSR}$ vs. AODE**		**$AODE^{LSR}$ vs. AODE**		**$AODE^{ESR}$ vs. AODE**	
	W/D/L	p	W/D/L	p	W/D/L	p	W/D/L	p
0-1 loss	16/38/6	**0.03**	21/30/9	**0.02**	17/35/8	**0.05**	15/40/5	**0.02**
Bias	20/38/2	**<0.001**	23/32/5	**<0.001**	15/37/8	0.11	15/39/6	**0.04**
Var	7/39/14	0.09	10/30/20	**0.05**	14/35/11	0.35	9/38/13	0.26
RMSE	14/39/7	0.09	20/30/10	**0.05**	16/36/8	0.08	17/38/5	**0.01**

The win/draw/loss records for $AODE^{BSE}$, $AODE^{NSR}$, $AODE^{LSR}$ and $AODE^{ESR}$ are shown in Table XI. All four improvements to AODE significantly reduce AODE's zero-one loss. $AODE^{NSR}$ and $AODE^{ESR}$ have significant RMSE advantages over AODE. $AODE^{BSE}$ and $AODE^{LSR}$ have lower RMSE more often than AODE and this result is almost significant ($p < 0.1$). $AODE^{BSE}$, $AODE^{NSR}$ and $AODE^{ESR}$ have significant bias advantage over AODE. $AODE^{NSR}$ has a significant variance disadvantage relative to AODE. $AODE^{BSE}$, $AODE^{LSR}$ and $AODE^{ESR}$ also have variance disadvantages relative to AODE, however, these results are not significant.

8.4. Handling Missing Values

Both NB and AODE have the ability to directly handle missing values. We evaluate the effectiveness of the discussed improvements to NB and AODE in handling missing values, by comparing their performance when doing so against their performance on the 20 datasets with missing values replaced by either modes or means from the training data.

The win/draw/loss records of NB and its improvements on datasets with and without missing values are given in Table XII. NB and its improvements have zero-one loss and RMSE advantages when missing values are handled directly. The zero-one loss advantage of NB^{ESR} when missing values are directly handled is nearly significant

($p = 0.07$). NB^{NSR}, NB^{LSR} and NB^{ESR} directly handling missing values have a nearly significant ($p = 0.07$) variance advantage over the corresponding algorithms that use missing value imputation.

Table XII. Win/Draw/Loss: NB and its improvements handling missing values directly compared with missing value imputation for the 20 datasets that contain missing values.

	NB		**NB**^{BSE}		**NB**^{NSR}		**NB**^{LSR}		**NB**^{ESR}	
	W/D/L	p	W/D/L	p	W/D/L	p	W/D/L	p	W/D/L	p
0-1 loss	11/3/6	0.17	11/3/6	0.17	11/3/6	0.17	11/3/6	0.17	12/3/5	0.07
Bias	9/3/8	0.50	11/3/6	0.17	8/4/8	0.60	8/4/8	0.60	10/3/7	0.31
Var	10/3/7	0.31	10/2/8	0.41	12/3/5	0.07	12/3/5	0.07	12/3/5	0.07
RMSE	11/1/8	0.32	12/1/7	0.18	12/1/7	0.18	11/1/8	0.32	12/1/7	0.18

AODE, AODE^{BSE}, AODE^{NSR} and AODE^{LSR} directly handling missing values have lower RMSE significantly more often than the corresponding algorithms with missing value imputation. The zero-one loss of all the AODE variants is also reduced more often when missing values are directly handled, but this result is not significant. The variance also is reduced marginally more often when missing values are directly handled.

Table XIII. Win/Draw/Loss: AODE and its improvements handling missing values directly compared with missing value imputation for the 20 datasets that contain missing values.

	AODE		**AODE**^{BSE}		**AODE**^{NSR}		**AODE**^{LSR}		**AODE**^{ESR}	
	W/D/L	p	W/D/L	p	W/D/L	p	W/D/L	p	W/D/L	p
0-1 loss	12/1/7	0.18	11/2/7	0.24	11/2/7	0.24	12/2/6	0.12	10/3/7	0.31
Bias	10/3/7	0.31	8/3/9	0.50	7/3/10	0.31	9/3/8	0.50	9/3/8	0.50
Var	9/4/7	0.40	10/4/6	0.23	12/2/6	0.12	11/3/6	0.17	11/2/7	0.24
RMSE	13/3/4	**0.02**	13/3/4	**0.02**	13/3/4	**0.02**	14/3/3	**0.01**	11/3/6	0.17

8.5. Comparison of Ten Algorithms

In this section, we compare the ten algorithms discussed with LR and LibSVM. We use Weka's implementations and default settings of LR, which builds a multinomial logistic regression model with a ridge estimator whose default value is e^{-8}. We use Weka's implementations and default settings of LibSVM with the exceptions of turning on normalization of data and performing a "grid-search" on C and γ for the RBF kernel using 5-fold cross-validation. Each pair of (C, γ) is tried ($C = 2^{-5}, 2^{-3}, \ldots, 2^{15}, \gamma = 2^{-15}, 2^{-13}, \ldots, 2^{3}$) and the one with the lowest cross-validation zero-one loss is selected. Due to the high time

complexity of this process, the results of LibSVM on Covertype and Census-Income (KDD) have not been obtained and those on Connect-4 Opening, Shuttle, Adult, Letter Recognition and MAGIC Gamma Telescope are obtained from five runs of two-fold cross-validation. LibSVM in Weka uses a logistic function to calibrate the probability output. however, it is substantially slower than LibSVM without calibration. To avoid even slower training, we do not calibrate its output to produce probability estimates. It uses the one-against-one approach to generalizing from two-class classification to multi-class classification.

The NB and AODE variants discussed can only handle categorical data. On the other hand, LibSVM and LR cannot handle missing values. In order to compare the discussed algorithms with LibSVM and LR, we evaluated all the algorithms using 3-bin equal frequency discretization of quantitative attributes and missing values replaced by either their modes or means. While LibSVM's performance is superior on numeric data, it was evaluated on discretized data to provide a comparison of performance on categorical data. Although AODE has been extended to handle numeric data, the research is still at an early stage (Flores et al., 2009). Thus, comparison of AODE with LR and LibSVM on numerical datasets is left for future work.

Demšar (2006) recommends the Friedman test (Friedman, 1937; Friedman, 1940) for comparisons of multiple algorithms over multiple data sets. It first calculates the ranks of algorithms for each data set separately (average ranks are assigned if there are tied values), and then compares the average ranks of algorithms over data sets. The null-hypothesis is that there is no difference in average ranks. We reject the null-hypothesis if the Friedman statistic derived by Iman and Davenport (1980) is larger than the critical value of the F distribution with $a-1$ and $(a-1)(D-1)$ degrees of freedom for $\alpha = 0.05$, where a is the number of algorithms and D is the number of data sets. If the null-hypothesis is rejected then it is probable that there is a true difference in the average ranks of at least two algorithms. Post-hoc tests, such as the Nemenyi test, are used to determine which pairs of algorithms have significant differences.

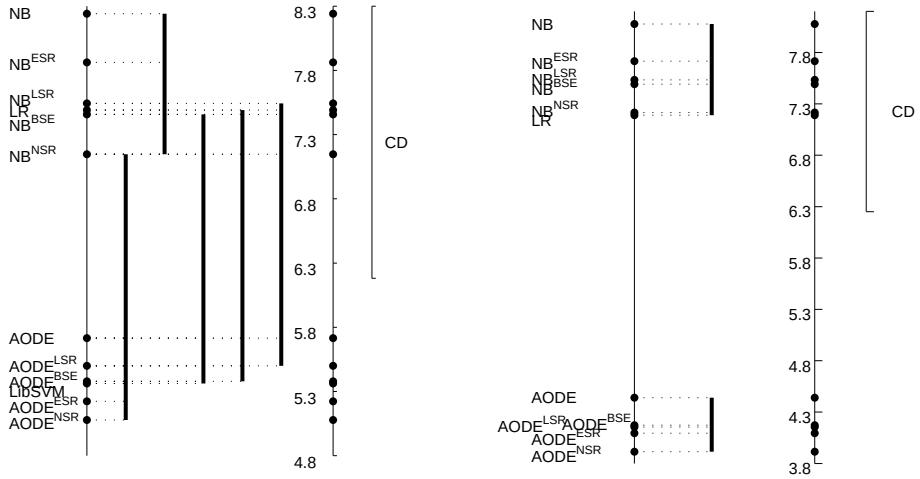
With 12 algorithms and 58 data sets, the Friedman statistic is distributed according to the F distribution with $12 - 1 = 11$ and $(12 - 1) \times (58 - 1) = 627$ degrees of freedom, The critical value of $\mathrm{F}(11, 627)$ for $\alpha = 0.05$ is 1.8039. The Friedman statistic for zero-one loss, bias and variance in our experiments are 7.8281, 17.7483 and 9.0832 respectively, and hence we reject all the null-hypotheses.

The Nemenyi test is used to further analyze which pairs of algorithms are significantly different. Let d_i^j be the difference between i^{th} algorithm and j^{th} algorithm. We assess a difference between i^{th}

algorithm and j^{th} algorithm as significant if $d_i^j >$ Critical Difference (CD). With 12 algorithms and 58 data sets, the Critical Difference for $\alpha = 0.05$ is CD $= 3.164 \times \sqrt{a \times (a+1)/(6 \times D)} = 3.164 \times \sqrt{12 \times (12+1)/(6 \times 58)} = 2.1184$.

As we do not obtain LibSVM's probability estimates, we present RMSE results for all the algorithms except LibSVM. The critical value of $F(10, 570)$ for $\alpha = 0.05$ is 1.8473, and the Friedman statistic for RMSE is 19.5832. Therefore, the null-hypothesis that there is no difference in average RMSE ranks is rejected. The Critical Difference for $\alpha = 0.05$ with 11 algorithms and 58 data sets is CD = 1.9486.

Following the graphical presentation proposed by Demšar, we show the comparison of these algorithms against each other with the Nemenyi test on zero-one loss, bias, variance and RMSE in Figures 4 and 5. We plot the algorithms on the left line according to their average ranks, which are indicated on the parallel right line. Critical Difference (CD) is also presented in the graphs. The lower the position of algorithms, the lower the ranks will be, and hence the better the performance. The algorithms are connected by a line if their differences are not significant. Since the comparison involves 12 algorithms, the power of the Nemenyi test is low and so only large effects are likely to be apparent.



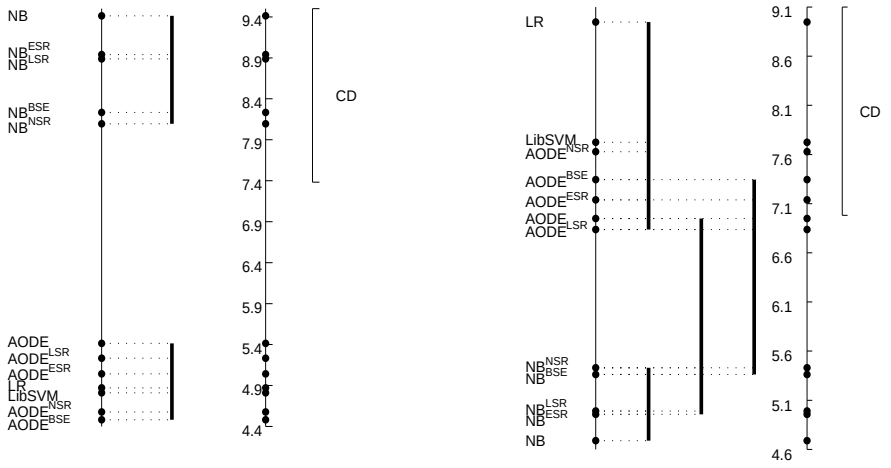
(a) *Zero-one Loss* (b) *RMSE*

Figure 4. Zero-one Loss and RMSE comparison with the Nemenyi test on 58 data sets. CD = 2.1184 for zero-one loss and CD = 1.9486 for RMSE.

8.5.1. *Zero-one Loss and RMSE*

AODE^{NSR} achieves the lowest mean zero-one loss rank (5.078), followed by AODE^{LSR} (5.224). They enjoy a significant zero-one loss advantage relative to NB^{BSE}, LR, NB^{LSR}, NB^{ESR} and NB. LibSVM is ranked third overall (5.362). The Nemenyi test differentiates LibSVM from LR, NB^{LSR}, NB^{ESR} and NB. AODE^{BSE} has a significantly lower mean zero-one loss rank than NB^{LSR}, NB^{ESR} and NB. AODE^{NSR}, AODE^{ESR}, AODE^{BSE} and AODE^{LSR} have lower mean zero-one loss ranks than AODE, but not significantly so. Due to the low power of the Nemenyi test when a large number of algorithms are compared, these results differ from those of Section 8.3, in which BSE, NSR and LSR provide significant zero-one loss reductions in NB and all four improvements to AODE (BSE, NSR, LSR and ESR) significantly improve upon the zero-one loss of AODE.

When RMSE is compared, there are two clear groups. AODE^{NSR}, AODE^{ESR}, AODE^{LSR}, AODE^{BSE} and AODE deliver significantly lower mean zero-one loss ranks than all the other algorithms. AODE^{NSR} and AODE^{ESR} achieve the lowest and second lowest mean RMSE ranks (3.914 and 4.095 respectively). The differences RMSE ranks between NB^{NSR}, NB^{LSR}, NB^{BSE} and NB^{ESR} are small, ranging from 8.095 to 8.888.



(a) *Bias* (b) *Variance*

Figure 5. Bias and variance comparison with the Nemenyi test on 58 data sets. CD = 2.1184.

8.5.2. *Bias and Variance*
AODE^{BSE} obtains the lowest mean bias rank (4.4828), followed closely by AODE^{NSR} (4.5776). LibSVM and LR come next (4.8103 and 4.8707 respectively). The bias disadvantages of NB^{NSR}, NB^{BSE}, NB^{LSR}, NB^{ESR} and NB relative to all remaining algorithms are clear. BSE has the largest effect on reducing the bias of AODE and NSR has the largest effect on reducing the bias of NB. However, due to the statistical test employed, the effect is not significant.

NB has the lowest mean variance rank. NB^{ESR} and NB^{LSR} have the second and the third lowest mean variance ranks. These three algorithms achieve significantly lower mean variance ranks than AODE^{ESR}, AODE^{BSE}, AODE^{NSR}, LibSVM and LR. The NB variant algorithms achieve significantly lower mean variance ranks than AODE^{NSR}, LibSVM and LR.

8.6. Average Elimination Ratio

To observe the percentage of generalizations or near-generalizations, we calculate average attribute elimination ratios for LSR and NSR on each data set, obtained by dividing the number of attributes deleted by the number of attributes across all the test examples and iterations:

$$e^{LSR} = \frac{\sum_{i=1}^{u} \sum_{o=1}^{t} e_o^i}{tnu}, \tag{7}$$

where u is the number of iterations (it is 50 in our experiment) and e_o^i is the number of attributes deleted for the o^{th} instance in the i^{th} iteration. NSR also uses (7) to calculate the average elimination ratio e^{NSR}.

8.6.1. *Average Elimination Ratio of LSR*
Figures 6 shows average elimination ratios of LSR. An average elimination ratio of zero represents no deletions. The larger the elimination ratio, the more attributes that are deleted. The data sets in Figures 6 are in the number sequence of Table V. Since the attributes deleted do not change from classification algorithm to algorithm, NB^{LSR} and AODE^{LSR} have an identical elimination ratio on the same data set. As illustrated in Figure 6, elimination occurs on 22 out of 60 data sets. For more than 5% data sets, over 50% of attribute values are eliminated. For more than 15% of data sets, over 10% of attribute values are eliminated. As a larger value of l can reduce the number of true generalizations that are detected, higher percentages of attribute values are deleted when smaller values of l are used.

The average elimination ratios on four data sets are greater than 0.5. The zero-one loss ratios of NB^{LSR} to NB and AODE^{LSR} to AODE

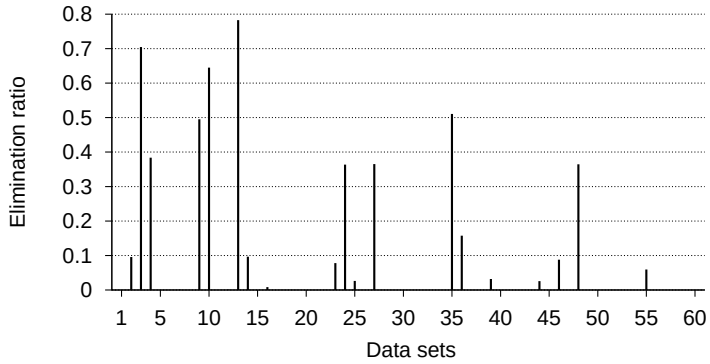
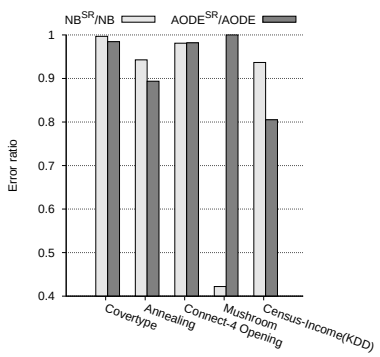


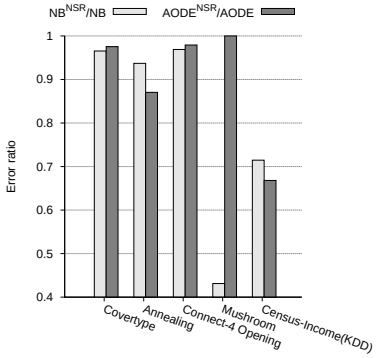
Figure 6. Average attribute elimination ratio of LSR. The data sets are in the number sequence of Table V.

on these four data sets are shown in Figure 7 (a) and those of NB^{NSR} to NB and AODE^{NSR} to AODE are shown in Figure 7 (b). RMSE results are shown in Figure 8 (a) and (b). Ratios less than one indicate improvement. On Covertype, $e^{LSR} = 0.7826$, indicating that more than 78% of attribute values are eliminated. The reason for this high ratio is because this data sets has 44 binary attributes, each having a value of 0 if a type of wilderness area or soil is absent and a value of 1 otherwise. The 11^{th} to 14^{th} attributes describe four types of wilderness areas respectively. If one of these attributes has a value of 1, then the other attributes will have a value of 0. For any pair of these four attributes, there are three possible values of $\{0, 0\}$, $\{0, 1\}$ and $\{1, 0\}$, two of them have the substitution relationship. For all these four attributes, there are 18 possible pairs of attribute values, two-thirds of them having the substitution relationship. The same rule applies to the 15^{th} to 54^{th} attributes. There are 2340 possible pairs of attribute values with two-thirds of them having the substitution relationship. In addition, the generalization relationship is frequently detected between wilderness areas and soil type. An example for this relation is that the 15^{th} with value of 1 is a specialization of the 11^{th} with value of 0. For this data, the test time of AODE is substantially reduced from 4987.96 seconds to 928.60 seconds, despite LSR has an additional step to detect generalizations. The zero-one loss ratios of NB^{LSR} to NB and AODE^{LSR} to AODE are 0.9967 and 0.9845 and the RMSE ratios are 0.9953 and 0.9897 respectively. When we apply LSR to this data without considering the relationships between these binary attributes, the average elimination ratio is 0.0994.

The average elimination ratio on Annealing is 0.7046. One factor that contributes to this high ratio is that this data has many miss-

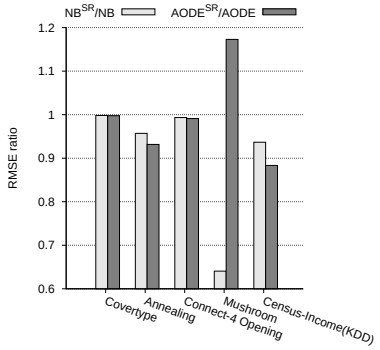


(a) NB^{LSR}/NB and AODE^{LSR}/AODE

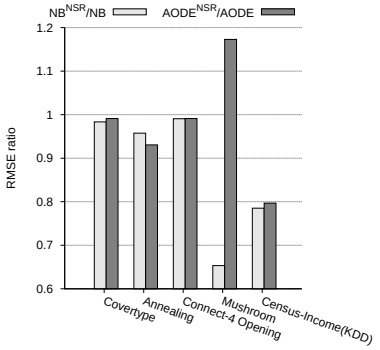


(b) NB^{NSR}/NB and AODE^{NSR}/AODE

Figure 7. Zero-one loss ratio



(a) NB^{LSR}/NB and AODE^{LSR}/AODE



(b) NB^{NSR}/NB and AODE^{NSR}/AODE

Figure 8. RMSE ratio

ing values. More than 76% attributes have missing values and 70% attributes have more than 50% missing values. Many attributes only have one value in addition to missing values. Quantitative attributes in this data do not have missing values. When those missing values for qualitative attributes are replaced with modes, these attributes only have one value, which results in a large number of generalizations. As NB and AODE can deal with missing values, we apply LSR to NB and AODE without replacing missing values. The resulting average elimination ratio is 0.1126. Several exemplar generalizations for this data are: Len<0.5 is a generalization of $Shape$=$COIL$, $Bore$=0000 is of $Width$<609.95, $Shape$=$SHEET$ and Len>821, $Shape$=$SHEET$ is of Len>0.5 and $Strength$<150 is of $Thick$<0.6995. AODE's test time is reduced from 2.99 seconds to 1.16 seconds. The zero-one loss ratios of NB^{LSR} to NB and AODE^{LSR} to AODE are 0.9427 and 0.8936 and the RMSE ratios are 0.9569 and 0.9316 respectively. The test time of AODE is reduced from 4.11 seconds to 3.10 seconds.

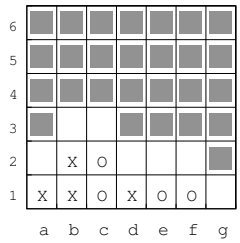


Figure 9. An example from Connect-4 Opening. A grey colored square is a generalization of the lowest unoccupied square in the column.

The third largest average elimination ratio is on Connect-4 Opening. It deletes 64.51% of attribute values on average. Connect 4 is a game in which two players take turns in placing pieces on a 7-column, 6-row vertically-suspended grid and will try to get four connected singly-colored pieces, either horizontally, vertically or diagonally. The 6 rows are numbered 1 through 6 and the 7 columns are labeled 'a' through 'g'. There are 42 attributes, each having 3 values. An attribute has a value of `x` if the corresponding square is occupied by the first player and a value of `o` if the square is occupied by the second player. Otherwise, this attribute has a value of `b`. Figure 9 shows an example from this data set. When a piece is placed in one of the columns, it will fall down to the lowest unoccupied square in the column. Therefore, all squares higher than the lowest unoccupied square are empty. From this rule, we have $\forall z \in \{a, b, c, d, e, f, g\}$ and $1 \leq i < j \leq 6$ if $z_i = \texttt{b}$ then $z_j = \texttt{b}$. In other words, $z_j = \texttt{b}$ is a generalization of $z_i = \texttt{b}$. In Figure 9, 27 empty squares (those grey colored squares) are generalizations. Due to a large number of attribute values deleted, LSR substantially reduces the test time of AODE from 222.91 seconds to 59.97 seconds. The zero-one loss ratios of NB^{LSR} to NB and AODE^{LSR} to AODE are 0.9810 and 0.9819 and the RMSE ratios are 0.9933 and 0.9911 respectively.

On average, 51.07% attribute values are deleted on Mushroom. The number of attributes is 22 and the mean number of values per attribute is 6.7. In total, there are $22 \times 6.7 \times (22 \times 6.7 - 6.7)/2 = 10340.11$ combinations of attribute values. Among them, 227 pairs of attribute values are detected to have the generalization relationship. For example, *Cap-shape=convex* is a generalization of *Odor=creosote* and *Gill-attachment=free* is a generalization of *Gill-spacing=crowed*. LSR reduces NB's zero-one loss and RMSE from 0.0109 and 0.0946 to 0.0046 and 0.0606 respectively. The zero-one loss of AODE is unchanged by the addition of LSR. However, it is not immediately clear why the application of LSR increases the RMSE of AODE from 0.0162 to 0.0190.

Almost half of attribute values are deleted on Census-Income (KDD) (e^{LSR} being 0.4953). On the whole training data, 332 attribute values are identified as a generalization of another attribute value. The generalization relationship between attribute values in this data is often obvious. For instance, if *State-of-previous-residence=Florida*, then *Region-of-previous-residence=south*. The discretized values for the first attribute *Age* are $\leq$21.5, $\leq$43.5 $\wedge$ $>$21.5 and $>$43.5. The 23^{th} attribute is *Detailed-household-and-family-stat*. If the value of this attribute is *Child<18-never-marr-not-in-subfamily*, then *Age*$\leq$21.5. The 32^{th} attribute is *Family-members-under-18*. All values except *not-in-universe* are specializations of *Age*$\leq$21.5. The 27^{th} attribute is *Migration-code-change-in-reg* with 8 values. The 28^{th} attribute is *Migration-code-move-within-reg* with 9 values. Five values of these two attributes are identical. These values are *not-in-universe*, *nonmover*, *same-county*, *different-county-same-state* and *abroad*. The 27^{th} attribute with any of these five values is a substitution (and so generalization) of the 28^{th} attribute with a corresponding value. The zero-one loss ratios of NB^{LSR} to NB and AODE^{LSR} to AODE are 0.9367 and 0.8052 and the RMSE ratios are 0.9367 and 0.8834 respectively (shown in Figure 7 and 8). The test time of AODE is reduced from 478.93 seconds to 291.20 seconds.

8.6.2. *Average Elimination Ratio of NSR*

Figures 10 presents average elimination ratios for NSR. As NSR consid-

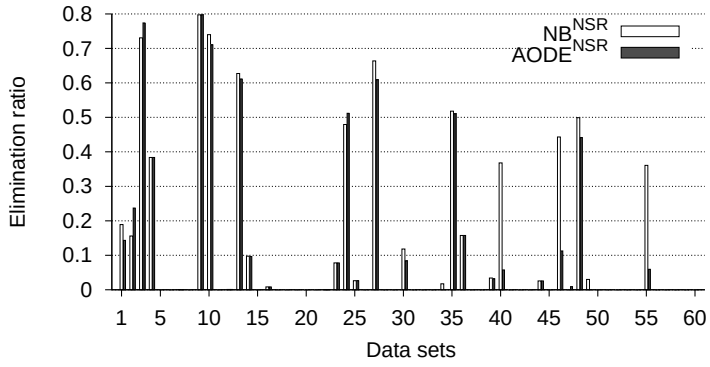


Figure 10. Average attribute elimination ratio of NSR. The data sets are in the number sequence of Table V.

ers specific classification algorithms in the process of selecting r value, NB^{NSR} and AODE^{NSR} can have different average elimination ratios. When NSR is applied to NB and AODE, elimination occurs on more than 40% data sets. For 10% data sets, over 50% of attribute values are

eliminated. For more than 20% of data sets, more than 10% of attribute values are eliminated.

The average elimination ratio on Covertype is 0.8145 for NB^{NSR} and 0.8087 for AODE^{NSR}. The zero-one loss ratios of NB^{NSR} to NB and AODE^{NSR} to AODE are 0.9604 and 0.9689 and the RMSE ratios are 0.9834 and 0.9935 respectively (see Figure 7 (b) and 8 (b)). To observe the number of pairs of attribute values that have the near-generalization relationship, we apply NSR to the whole training data. As r value changes from one cross-validation run to another, we use the most frequently selected value of r=0.75 in 50-run 2-fold cross validation for NB^{NSR}. 3078 pairs of attribute values are identified to have near-generalization relationships. An example for the relationships is that if aspect in degrees azimuth is between 78.5 and 205.5 then we can roughly infer that hill shade index at 9am is greater than 226.5.

Census-Income (KDD) has the second largest $e^{NSR} = 0.7976$ for both NB^{NSR} and AODE^{NSR}. The zero-one loss and RMSE of NB and AODE are substantially reduced by the addition of NSR. The zero-one loss ratios of NB^{NSR} to NB and AODE^{NSR} to AODE are 0.7146 and 0.6680 and the RMSE ratios are 0.7851 and 0.7967 respectively. When NSR is applied to the full training data with r=0.75, a value selected by most folds for NB^{NSR}, 10235 attribute values are detected as near-generalizations. For example, if *Class-of-worker=never-worked*, we can infer that most of them are younger than 21.5. If *Wage-per-hour*>800.5, we can approximately infer that *Class-of-worker=private*. Most people that work in the construction industry are male. Over 90% people whose wage per hour is greater than 800.5 dollars were born in the United States.

8.6.3. *Advantages of Deleting Near-Generalizations*

Figure 4 does not reveal the zero-one loss and RMSE differences between LSR and NSR as significant since the number of algorithms compared is large and consequently the power of the Nemenyi test is low. When NSR is compared with LSR, the former has a significant zero-one loss, bias and RMSE advantages relative to the latter on NB and AODE. Table XIV presents the win/draw/loss records for NSR against LSR and BSE on NB and AODE.

The zero-one loss and RMSE differences between NSR and BSE are small when they are applied to NB, while NSR has a marginal zero-one loss advantage and significant RMSE advantage relative to BSE when they are applied to AODE. In this section, we investigate the circumstances under which deleting near-generalizations proves advantageous based on two exemplar data sets (Adult and Abalone), both deleting more than 10% of attribute values.

Table XIV. Win/Draw/Loss: NB^{NSR} vs. NB^{LSR} and NB^{BSE} and $AODE^{NSR}$ vs. $AODE^{LSR}$ and $AODE^{BSE}$

	NB^{NSR} vs. NB^{LSR}		**$AODE^{NSR}$ vs. $AODE^{LSR}$**	
	W/D/L	p	W/D/L	p
0-1 loss	13/44/3	**0.011**	11/48/1	**0.003**
Bias	17/43/0	**<0.001**	13/47/0	**<0.001**
Variance	3/43/14	**0.006**	1/47/12	**0.002**
RMSE	14/44/2	**0.002**	11/49/0	**<0.001**
	NB^{NSR} vs. NB^{BSE}		**$AODE^{NSR}$ vs. $AODE^{BSE}$**	
	W/D/L	p	W/D/L	p
0-1 loss	27/1/32	0.301	20/29/11	0.075
Bias	7/4/49	**<0.001**	23/28/9	**0.010**
Variance	48/2/10	**<0.001**	16/29/15	0.500
RMSE	30/1/29	0.500	22/29/9	**0.015**

8.6.4. *Adult*

The classification task of Adult is to predict whether income exceeds fifty thousand US dollars a year. It has 14 attributes (6 continuous and 8 discrete) besides the class label. The 5^{th} attribute *Education-num* recodes the 4^{th} attribute *Education* from a descriptive to a numeric format and hence *Education-num* without discretization is a substitution of *Education* and *Education-num* with discretization is a generalization of *Education*. Given *Education*, *Education-num* is redundant. This redundancy can be detected by LSR. The zero-one losses of AODE with all attributes and all attributes except *Education-num* are 0.1598 and 0.1588 respectively. However, NB with all attributes has lower zero-one loss (0.1727) than NB with all attributes except *Education-num* (0.1851). The zero-one loss of NB^{LSR} and $AODE^{LSR}$ are 0.1802 and 0.1575 respectively. As NB^{LSR} and $AODE^{LSR}$ delete attributes depending upon which attribute values are instantiated in the object being classified, they may have different results from that of NB and AODE when deleting complete attributes. The generalization relationship is detected between another 6 pairs of attribute values.

NSR substantially improves upon NB and AODE on Adult. The zero-one loss of NB and AODE are reduced from 0.1727 and 0.1598 to 0.1550 and 0.1484 respectively and the RMSE of NB and AODE are reduced from 0.3550 and 0.3383 to 0.3309 and 0.3213 respectively. We investigate the attributes that are deleted by NSR employing $r = 0.98$, the most frequently selected value in 50-run 2-fold cross validation for NB^{NSR}. Our experiment reveals that both LSR and NSR delete generalizations discussed above for most test instances, and NSR also

deletes two other types of attributes. They are near-generalizations and attributes with noise.

8.6.4.1. *Near-Generalizations.* The 6^{th} attribute *Marital-status* and the 8^{th} attribute *Relationship* are closely associated. If a person is classified as a wife (or husband), she (or he) must be a married person. However, we could not make the further judgement whether a married person is either a *Married-civ-spouse* or *Married-AF-spouse* due to two types of marriage being listed in the data set. There are 22379 instances of *Married-civ-spouse* and 37 instances of *Married-AF-spouse*. It is obvious that civilian marriages account for the majority of marriages. Therefore, we can approximately infer that a married person belongs to a civilian marriage. That is, *Marital-status=Married-civ-spouse* is a near-generalization of *Relationship=Husband* and *Relationship=wife*. To evaluate the effect of deleting *Marital-status=Married-civ-spouse* using $r = 0.98$, we apply NSR to NB and AODE but restrict its application to deleting only values of *Marital-status*. The zero-one loss and RMSE of NB^{NSR} are 0.1572 (< 0.1727) and 0.3402 (< 0.3550) respectively, and those of AODE^{NSR} are 0.1511 (< 0.1598) and 0.3280 (< 0.3383) respectively. These results suggest that the elimination of a near-generalization accounting for a large part of population to which near-specializations belong might be positive. There are 10 pairs of attribute values are identified to have the near-generalization relationship.

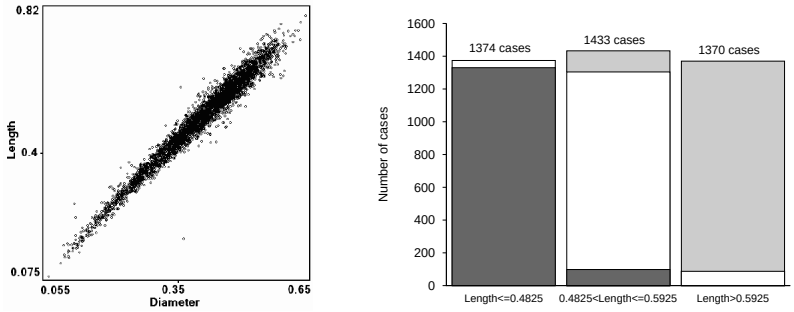
8.6.4.2. *Attributes with Noise.* The 10^{th} attribute *Sex* has two values of *female* and *male*. These values have a clear generalization relationship with the two values (*husband* and *wife*) of attribute *Relationship*. That is, *Sex=female* is a generalization of *Relationship=wife* and *Sex=male* is a generalization of *Relationship=husband*. However, due to noise in the data, the relation can not be detected by LSR. The values of *Relationship* and *Sex* of the 7110^{th} instance in Adult are *husband* and *female* respectively. Another two cases are the 576^{th} and 27142^{th} instances in which *Relationship=wife* and *Sex=male*. When we apply NSR to NB and AODE but restrict its application to deleting only values of *Sex*, NB^{NSR} and AODE^{NSR} have zero-one losses of 0.1659 (< 0.1727) and 0.1574 (< 0.1598) respectively and RMSEs of 0.3476 (< 0.3550) and 0.3349 (< 0.3383) respectively. These results indicate that the near-generalization technique can be useful in at least some cases of noise.

8.6.5. *Abalone*

In Abalone, the classification task is to predict the age of an abalone from its physical measurements, many of which are closely correlated to one another. Since NB and AODE cannot handle numeric classes, we select the only attribute (Sex) that has categorial values (M, F and I) as the class.

Figure 11 (a) presents the scatter graph that plots the values of *Length* versus the corresponding values of *Diameter* without discretization. The relationship between these two attributes is linear. Similar relationships are observed for other attributes (scatter graphs are not presented), specifically *Shucked-weight* and *Viscera-weight* are linearly related, *Whole-weight* and *Length*, and *Diameter* and *Shell-weight* are roughly linearly related. As *Diameter* and *Length* are positively linearly related, it is logical to infer that an abalone with small diameter is shorter. However, there are some exceptional cases where abalones with small diameter measure longer than average.

Figure 11 (b) presents the relationship between *Length* and *Diameter* with discretized values. Dark grey blocks are cases with *Diameter*$\leq$0.3775, light grey blocks are cases with *Diameter*$>$0.5925 and white blocks are cases with other values. The *Diameter* of 1330 cases out of 1374 cases with *Length*$\leq$0.4825 is less than 0.3775, that of 1206 cases out of 1433 cases with 0.4825$<$*Length*$\leq$0.5925 is between 0.3775 and 0.4625, and 1283 cases out of 1370 cases with *Length*$>$0.5925 is larger than 0.4625. The near-generalization relationship between values of these two attributes is clear, for example, *Diameter*$\leq$0.3775 is a near-generalization of *Length*$\leq$0.4825.



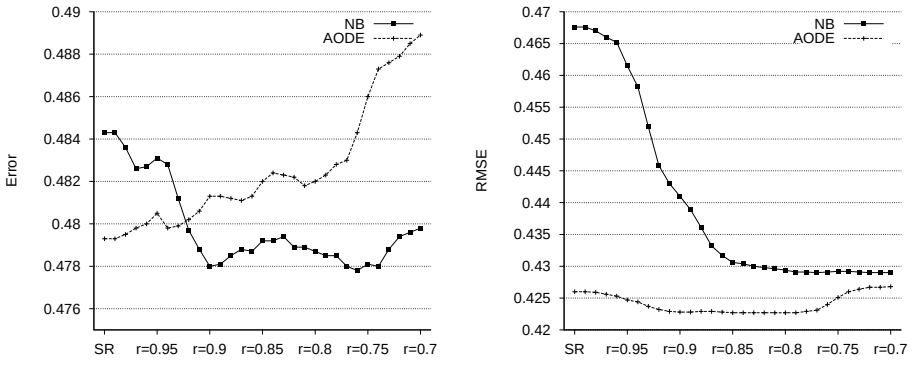
(a) Without discretization (b) Discretized into 3 bins

Figure 11. Close Correlation of *Length* and *Diameter*

LSR cannot find these near-relations most of the time. Note that when the relationship is detectable, a deletion will only occur for test cases that are not themselves outliers as, for example, a long

abalone with small diameter will not be an instance of the detected near-generalization relationship.

To observe the effect of elimination of near-generalizations, we apply NSR to NB and AODE using values of r in the range of 0.99 to 0.70 with an decrement of 0.1. Figure 12 (a) shows the learning curves on



(a) Zero-one loss (b) RMSE

Figure 12. Learning curves on Abalone.

zero-one loss in which each point represents the zero-one loss of NB^{NSR} and AODE^{NSR} corresponding to each r on the x-axis. Figure 12 (b) presents the learning curves on RMSE for NB^{NSR} and AODE^{NSR}. The zero-one loss and RMSE of LSR are also included in the graphs. The two decimal numbers on the x-axis are the lower bounds of the near-generalization relationship.

NB^{NSR} has a largely downward trend in zero-one loss over the range of $r = 0.99$ to $r = 0.76$. The RMSE of NB starts with a steady decline and stabilizes to 0.429 from $r = 0.72$. The RMSE of AODE decreases slightly from $r = 0.99$ to $r = 0.85$, stabilizes to 0.4227 from $r = 0.85$ to $r = 0.79$, and then increases slightly. These results suggest that selecting an appropriate r value for NSR can have a positive effect on RMSE. There is a largely upward trend for the zero-one loss of AODE^{NSR} with decreasing values of r. The bias of NB^{NSR} and AODE^{NSR} have a clear downward trend and the variance of these two methods have a clear upward trend (graphs are not presented). One possible reason for discrepant trends in zero-one loss of NB and AODE is the greater complexity of an AODE model compared to an NB model, resulting in greater variance. The increase in variance provided by NSR outweighs the reduction in bias and results in overall increase in zero-one loss for AODE, while NSR provides an appropriate bias-variance trade-off and results in overall reduction in zero-one loss for NB.

9. Conclusions and Future Work

We have proposed novel techniques, LSR and NSR, to efficiently detect the generalization and near-generalization relationships, special forms of inter-dependency, and delete generalizations and near-generalizations at classification time. We have also proposed ESR, which is a filter that transforms the training data to remove these relationships at training time. We investigate the effect of LSR, NSR and ESR on zero-one loss and RMSE by applying them to NB and AODE. Extensive experimental results (win/draw/loss records) show that LSR and NSR significantly improve upon NB's zero-one loss and RMSE. ESR also significantly improves upon NB's probability estimates, but its zero-one loss improvements are marginal. The zero-one loss and RMSE of AODE can be significantly enhanced by the addition of NSR and ESR. Whilst LSR improved the zero-one loss and RMSE of AODE more often than not, only the zero-one loss was improved significantly more often. LSR, NSR and ESR are suited to probabilistic techniques, such as NB and AODE, but not to similarity techniques.

SR is related to attribute elimination, although it only eliminates specific values and only in the context of other specific values. For this reason we compared SR to BSE. BSE has considerably higher training time overheads than LSR. In the context of AODE, NSR has marginal classification and probabilistic prediction advantage relative to BSE. LSR inherits NB and AODE's capacity for incremental learning, while ESR, NSR and BSE do not support incremental learning. We believe that the appropriate conclusion to draw from our results is that LSR, NSR and ESR are effective at reducing error, rather than that they are necessarily superior to the BSE strategy in this respect in the AODE context. It is also possible that SR may be complementary to attribute elimination, with attribute elimination in the context of SR removing attributes that are problematic for reasons other than generalization-specialization relationships.

We explore reasons for high percentages of generalizations on three data sets. We also investigate the circumstances that NSR proves beneficial based on two exemplar data sets. When a near-generalization accounts for the majority of the population to which the corresponding near-specialization belongs, elimination of the near-generalization may excel. It might have an advantage when attributes are closely rather than perfectly associated. Furthermore, it may provide tolerance for noise to some extent.

LSR and ESR provide computationally efficient techniques of reducing the dimensionality of the data. There are number of avenues for extending these techniques. The near generalization parameter r for

NSR is currently chosen by performing a parameter search using cross-validation. A theoretical analysis to identify a more effective method of choosing r is an area of future work. Applying SR techniques to higher order average n dependence estimation algorithms such as A2DE and A3DE (see Webb *et. al.*, in-press, for details) is another area of future research. The order in which attributes are chosen for merging in ESR has a direct effect on the final outcome and the optimum order is likely to be different for NB and AODE. Exploration of effective methods of choosing the attribute merge order is a further area for future work.

We use the Friedman and Nemenyi tests to compare NB, AODE and their variants with LR and LibSVM with a grid parameter search on categorical data. The results reveal the outstanding performance of AODE^{NSR} and AODE^{ESR} on our datasets. They enjoy considerable advantage in zero-one loss and RMSE over NB, NB^{BSE}, NB^{NSR} and NB^{LSR} and LR. They also have a better mean zero-one loss rank in comparison to LibSVM. AODE^{LSR} also achieves high zero-one loss and RMSE with low training time and modest test time overheads.

It is notable that all of the SR variants of AODE obtain zero-one loss comparable to SVM with a grid parameter search. This comparable performance is obtained with far less computation. It is not possible to provide meaningful compute time comparisons because the computational requirements of LibSVM on the large data sets required that it be run in a heterogeneous grid computing environment from which it is inherently not possible to obtain useful timing comparisons. Notably, the AODE variants are linear on the quantity of data and are capable of directly handling missing data. In addition, NSR is the only variant that has been tested here using a parameter search. The only parameters used by the other variants are l, which has been fixed to 100, and the value of m which is used in m-estimation, which is fixed at 0.1. Finally, LSR supports incremental learning and learns in a single pass through the training data, making it possible to learn from data that are too large to reside in RAM.

Acknowledgements

This research has been supported by the Australian Research Council under grant DP0772238. The authors are grateful to Janez Demšar for his kind help with the Nemenyi test.

References

Cerquides, J. and R. L. D. Mántaras: 2005, 'Robust Bayesian Linear Classifier Ensembles'. In: *Proceedings of the Sixteenth European Conference on Machine Learning*. pp. 70–81.

Cestnik, B.: 1990, 'Estimating Probabilities: A Crucial Task in Machine Learning'. In: *Proceedings of the Ninth European Conference on Artificial Intelligence*. pp. 147–149, London: Pitman.

Dash, D. and G. F. Cooper: 2002, 'Exact Model Averaging with Naive Bayesian Classifiers'. In: *Proceedings of the Nineteenth International Conference on Machine Learning*. pp. 91–98, Morgan Kaufmann.

De Raedt, L.: 2010a, 'Logic of Generality'. In: C. Sammut and G. I. Webb (eds.): *Encyclopedia of Machine Learning*. Springer US, pp. 624–631.

De Raedt, L. D.: 2010b, 'Inductive Logic Programming'. In: C. Sammut and G. I. Webb (eds.): *Encyclopedia of Machine Learning*. Springer US, pp. 529–537.

Demšar, J.: 2006, 'Statistical Comparisons of Classifiers over Multiple Data Sets'. *Journal of Machine Learning Research* **7**, 1–30.

Domingos, P. and M. J. Pazzani: 1996, 'Beyond Independence: Conditions for the Optimality of the Simple Bayesian Classifier'. In: *Proceedings of the Thirteenth International Conference on Machine Learning*. pp. 105–112, Morgan Kaufmann.

Duda, R. O. and P. E. Hart: 1973, *Pattern Classification and Scene Analysis*. New York: John Wiley and Sons.

Fayyad, U. M. and K. B. Irani: 1993, 'Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning'. In: *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*. pp. 1022–1029, Morgan Kaufmann.

Flores, M., J. Gámez, A. Martínez, and J. Puerta: 2009, 'GAODE and HAODE: two proposals based on AODE to deal with continuous variables'. In: *Proceedings of the 26th Annual International Conference on Machine Learning*. pp. 313–320.

Frank, E., M. Hall, and B. Pfahringer: 2003, 'Locally Weighted Naive Bayes'. In: *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence*. pp. 249–256, Morgan Kaufmann.

Friedman, M.: 1937, 'The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance'. *the American Statistical Association* **32**(200), 675–701.

Friedman, M.: 1940, 'A Comparison of Alternative Tests of Significance for the Problem of m Rankings'. *the American Statistical Association* **11**(1), 86–92.

Friedman, N., D. Geiger, and M. Goldszmidt: 1997, 'Bayesian Network Classifiers'. *Machine Learning* **29**(2), 131–163.

Gama, J.: 2003, 'Iterative Bayes'. *Theoretical Computer Science* **292**(2), 417–430.

Hand, D. J. and K. Yu: 2001, 'Idiot's Bayes: Not So Stupid after All?'. *International Statistical Review* **69**(3), 385–398.

Hastie, T., R. Tibshirani, and J. Friedman: 2001, *Elements of Statistical Learning: Data Mining, Inference and Prediction*. New York: Springer.

Hilden, J. and B. Bjerregaard: 1976, 'Computer-aided diagnosis and the atypical case'. In: F. T. de Dombal and F. Gremy (eds.): *Decision Making and Medical Care: Can Information Science Help*. Amsterdam: North-Holland, pp. 365–378.

Iman, R. L. and J. M. Davenport: 1980, 'Approximations of the Critical Region of the Friedman Statistic'. *Communications in Statistics* pp. 571–595.

Keogh, E. J. and M. J. Pazzani: 1999, 'Learning Augmented Bayesian Classifiers: A Comparison of Distribution-based and Classification-based Approaches'. In: *Pro-*

ceedings of the International Workshop on Artificial Intelligence and Statistics. pp. 225–230.

Kittler, J.: 1986, 'Feature Selection and Extraction'. In: T. Y. Young and K.-S. Fu (eds.): *Handbook of Pattern Recognition and Image Processing*. New York: Academic Press.

Kohavi, R.: 1996, 'Scaling Up the Accuracy of Naive-Bayes Classifiers: a Decision-Tree Hybrid'. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. pp. 202–207.

Kohavi, R. and D. Wolpert: 1996, 'Bias Plus Variance Decomposition for Zero-One Loss Functions'. In: *Proceedings of the Thirteenth International Conference on Machine Learning*. pp. 275–283, San Francisco: Morgan Kaufmann.

Kononenko, I.: 1990, 'Comparison of Inductive and Naive Bayesian Learning Approaches to Automatic Knowledge Acquisition'. In: B. Wielinga, J. Boose, B.Gaines, G. Schreiber, and M. van Someren (eds.): *Current Trends in Knowledge Acquisition*. Amsterdam: IOS Press.

Kononenko, I.: 1991, 'Semi-naive Bayesian classifier'. In: *Proceedings of the Sixth European Working Session on Machine learning*. pp. 206–219, Berlin: Springer-Verlag.

Langley, P.: 1993, 'Induction of Recursive Bayesian Classifiers'. In: *Proceedings of the 1993 European Conference on Machine Learning*. pp. 153–164, Berlin: Springer-Verlag.

Langley, P., W. Iba, and K. Thompson: 1992, 'An Analysis of Bayesian Classifiers'. In: *Proceedings of the Tenth National Conference on Artificial Intelligence*. pp. 223–228, AAAI Press and MIT Press.

Langley, P. and S. Sage: 1994, 'Induction of Selective Bayesian Classifiers'. In: *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*. pp. 399–406, Morgan Kaufmann.

Langseth, H. and T. D. Nielsen: 2006, 'Classification using Hierarchical Naive Bayes models'. *Machine Learning* **63**(2), 135 – 159.

Lewis, D. D.: 1998, 'Naive Bayes at Forty: The Independence Assumption in Information Retrieval'. In: *Proceedings of the Tenth European Conference on Machine Learning*. Berlin, pp. 4–15, Springer.

Mitchell, T.: 1997, *Machine Learning*. McGraw Hill.

Newman, D., S. Hettich, C. Blake, and C. Merz: 1998, 'UCI Repository of machine learning databases. Irvine, CA: University of California, Department of Information and Computer Science'.

Pazzani, M. J.: 1996, 'Constructive Induction of Cartesian Product Attributes'. *ISIS: Information, Statistics and Induction in Science* pp. 66–77.

Platt, J. C.: 1999, 'Probabilistic outputs for support vector machines and comparison to regularized likelihood methods'. In: *Advances in Large Margin Classifiers*. MIT Press.

Sahami, M.: 1996, 'Learning Limited Dependence Bayesian Classifiers'. In: *Proceedings of the Second International Conference on Knowledge Discovery in Databases*. pp. 334–338, Menlo Park, CA: AAAI Press.

Webb, G. I.: 2000, 'MultiBoosting: A Technique for Combining Boosting and Wagging'. *Machine Learning* **40**(2), 159–196.

Webb, G. I., J. Boughton, and Z. Wang: 2005, 'Not So Naive Bayes: Aggregating One-Dependence Estimators'. *Machine Learning* **58**(1), 5–24.

Webb, G. I., J. Boughton, F. Zheng, K. M. Ting, and H. Salem: in-press, 'Learning by extrapolation from marginal to full-multivariate probability distributions: Decreasingly naive Bayesian classification'. *Machine Learning*.

Webb, G. I. and M. J. Pazzani: 1998, 'Adjusted Probability Naive Bayesian Induction'. In: *Proceedings of the Eleventh Australian Joint Conference on Artificial Intelligence*. pp. 285–295, Berlin:Springer.

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

Zadrozny, B. and C. Elkan: 2001, 'Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers'. In: *Proceedings of the Eighteenth International Conference on Machine Learning*. pp. 609–616, Morgan Kaufmann, San Francisco, CA.

Zadrozny, B. and C. Elkan: 2002, 'Transforming classifier scores into accurate multiclass probability estimates'. In: *Proceedings of the Eighth International Conference on Knowledge Discovery and Data Mining*. pp. 694–699, ACM Press.

Zhang, H., L. Jiang, and J. Su: 2005, 'Hidden Naive Bayes'. In: *Proceedings of the Twentieth National Conference on Artificial Intelligence*. pp. 919–924, AAAI Press.

Zhang, N. L., T. D. Nielsen, and F. V. Jensen: 2004, 'Latent Variable Discovery in Classification Models'. *Artificial Intelligence in Medicine* **30**(3), 283–299.

Zheng, F. and G. I. Webb: 2005, 'A Comparative Study of Semi-naive Bayes Methods in Classification Learning'. In: *Proceedings of the Fourth Australasian Data Mining Conference*. pp. 141–156.

Zheng, F. and G. I. Webb: 2006, 'Efficient Lazy Elimination for Averaged-One Dependence Estimators'. In: *Proceedings of the Twenty-third International Conference on Machine Learning*. pp. 1113–1120, ACM Press.

Zheng, F. and G. I. Webb: 2007, 'Finding the Right Family: Parent and Child Selection for Averaged One-Dependence Estimators'. In: *Proceedings of the Eighteenth European Conference on Machine Learning*. pp. 490–501, Springer Berlin / Heidelberg.

Zheng, Z. and G. I. Webb: 2000, 'Lazy Learning of Bayesian Rules'. *Machine Learning* **41**(1), 53–84.