Toward Optimal Feature Selection

Daphne Koller
Gates Building 1A
Computer Science Department
Stanford University
Stanford, CA 94305-9010
koller@cs.stanford.edu

Mehran Sahami
Gates Building 1A
Computer Science Department
Stanford University
Stanford, CA 94305-9010
sahami@cs.stanford.edu

Abstract

In this paper, we examine a method for feature subset selection based on Information Theory. Initially, a framework for defining the theoretically optimal, but computationally intractable, method for feature subset selection is presented. We show that our goal should be to eliminate a feature if it gives us little or no additional information beyond that subsumed by the remaining features. In particular, this will be the case for both irrelevant and redundant features. We then give an efficient algorithm for feature selection which computes an approximation to the optimal feature selection criterion. The conditions under which the approximate algorithm is successful are examined. Empirical results are given on a number of data sets, showing that the algorithm effectively handles datasets with large numbers of features.

1 Introduction

In the classic *supervised learning* task, we are given a training set of labeled fixed-length feature vectors, or instances, from which to induce a classification model. This model is then used to predict the class label for a set of previously unseen instances. Thus, the information about the class that is inherent in the features determines the accuracy of the model. Theoretically, having more features should give us more discriminating power. However, the real-world provides us with many reasons why this is not generally the case.

First, the time requirements for an induction algorithm often grow dramatically with the number of features, rendering the algorithm impractical for problems with a large number of features. Furthermore, many learning algorithms can be viewed as performing (a biased form of) estimation of the probability of the class label given a set of features. In domains with a large number of features, this distribution is very complex and of high dimension. Unfortunately, in the real world, we are often faced with the problem of limited data from which to induce a model. This makes it very difficult to obtain good estimates of the many probabilistic parameters. In order to avoid over-fitting the model to the particular distribution seen in the training data, many algorithms employ the Occam's Razor (Blumer *et al.* 1987) bias to build as simple a model as possible that still achieves some acceptable level of performance on the training data. This bias often leads us to prefer a small number of relatively predictive features over a very large number of features that, taken in the proper, but complex, combination, are entirely predictive of the class label. Irrelevant and redundant features also cause problems in this context as they may confuse the learning algorithm by helping to obscure the distributions of the small set of truly relevant features for the task at hand.

If we reduce the set of features considered by the algorithm, we can therefore serve two purposes. We can considerably decrease the running time of the induction algorithm, and we can increase the accuracy of the resulting model. In light of this, a number of researchers have recently addressed the issue of feature subset selection in machine learning. As defined by (John, Kohavi, & Pfleger 1994), this work is often divided along two lines: filter and wrapper models.

In the filter model, feature selection is performed as a preprocessing step to induction. Thus the bias of the learning algorithm does not interact with the bias inherent in the feature selection algorithm. Two of the most well-known filter methods for feature selection are RELIEF (Kira & Rendell 1992) and FOCUS (Almuallim & Dietterich 1991). In RELIEF, a subset of features in not directly selected, but rather each feature is given a weighting indicating its level of relevance to the class label. RELIEF is therefore ineffective at removing redundant features as two predictive but highly correlated features are both likely to be highly weighted. The FOCUS algorithm conducts an exhaustive search of all feature subsets to determine the minimal set of features that can provide a consistent labeling of the training data. This consistency criterion makes FOCUS very sensitive to noise in the training data. Moreover, the exponential growth of the power set of the features makes this algorithm impractical for domains with more than 25-30 features.

Another feature selection methodolgy which has recently received much attention is the wrapper model (John, Kohavi, & Pfleger 1994) (Caruana & Freitag 1994) (Langley & Sage 1994). This model searches through the space of feature subsets using the estimated accuracy from an induction algorithm as the measure of goodness for a particular feature subset. Thus, the feature selection is being "wrapped around" an induction algorithm, so that the bias of the operators that define the search and that of the induction algorithm strongly interact. While these methods have encountered some success on induction tasks, they are often prohibitively expensive to run and can be intractable for a very large number of features. Furthermore, the methods leave something to be desired in terms of theoretical justification. While an important aspect of feature selection is how well a method helps an induction algorithm in terms of accuracy measures, it is also important to understand how the induction problem in general is affected by feature selection.

In this work, we address both theoretical and empirical aspects of feature selection. We describe a formal framework for understanding feature selection, based on ideas from Information Theory (Cover & Thomas 1991). We then present an efficient implemented algorithm based on these theoretical intuitions. The algorithm overcomes many of the problems with existing methods: it has a sound theoretical foundation; it is effective in eliminating both irrelevant and redundant features; it is tolerant to inconsistencies in the training data; and, most importantly, it is a filter algorithm which does not incur the high computational cost of conducting a search through the space of feature subsets as in the wrapper methods, and is therefore efficient for domains containing hundreds or even thousands of features.

2 Theoretical Framework

A data instance is typically described to the system as an assignment of values $\boldsymbol{f} = (f_1, \ldots, f_n)$ to a set of *features* $\boldsymbol{F} = (F_1, \ldots, F_n)$. As usual, we assume that each instance is drawn independently from some probability distribution over the space of feature vectors. Formally, for each assignment of values $\boldsymbol{f}$ to $\boldsymbol{F}$, we have a probability $\Pr(\boldsymbol{F} = \boldsymbol{f})$.

A *classifier* is a procedure that takes as input a data instance and classifies it as belonging to one of a number of possible classes $c_1, \ldots, c_\ell$. The classifier must make its decision based on the assignment $\boldsymbol{f}$ associated with an instance. Optimistically, the feature vector will fully determine the appropriate classification. However, this is rarely the case: we do not typically have access to enough features to make this a deterministic decision. Therefore, we use a probability distribution to model the classification function: For each assignment of values $\boldsymbol{f}$ to $\boldsymbol{F}$ we have a distribution $\Pr(C \mid \boldsymbol{F} = \boldsymbol{f})$ on the different possible classes, C. A learning algorithm implicitly uses the empirical frequencies observed in the training set — an approximation to the conditional distribution $\Pr(C \mid \boldsymbol{F})$ — to construct a classifier for the problem.

Let us consider the effect of feature space reduction on the distribution that characterizes the problem. Let $\boldsymbol{G}$ be some subset of $\boldsymbol{F}$. Given a feature vector $\boldsymbol{f}$, we use $\boldsymbol{f_G}$ to denote the projection of $\boldsymbol{f}$ onto the variables in $\boldsymbol{G}$. Consider a particular data instance characterized by $\boldsymbol{f}$. In the original distribution, this data instance induces the distribution $\Pr(C \mid \boldsymbol{F} = \boldsymbol{f})$. In the reduced feature space, the same instance induces the (possibly different distribution) $\Pr(C \mid \boldsymbol{G} = \boldsymbol{f_G})$. Our goal is to select $\boldsymbol{G}$ so that these two distributions are as close as possible. As our distance metric, we use the information-theoretic measure of cross-entropy (also known as KL-distance (Kullback & Leibler 1951)). Thus, we can view this as selecting a set of features $\boldsymbol{G}$ which causes us to lose the least amount of information in these distributions. While other measures of separability (notably *divergence*) have been suggested in the statistics community for feature selection (Fukunaga 1990), these measures are often aimed at selecting features to enhance the separability of the data and may have difficulty in very large dimensional spaces. Hence, they bring with them an inherent bias which may not be appropriate for particular induction algorithms. Our method seeks to eliminate non-informative features and thereby allow induction methods to employ their own bias in a much reduced feature space.

Formally, let μ and σ be two distributions over some probability space Ω. The *cross-entropy* of μ to σ is defined as $D(\mu, \sigma) = \sum_{x \in \Omega} \mu(x) \log \frac{\mu(x)}{\sigma(x)}$. Note that the roles of μ and σ are not symmetrical in this definition. Generally speaking, the idea is that μ is the "right" distribution, and σ is our approximation to it. Then, $D(\mu, \sigma)$ measures the extent of the "error" that we make by using σ as a substitute for μ. Thus, cross-entropy is particularly suitable for our application, with $\Pr(C \mid \boldsymbol{f})$ in the role of the "right" distribution μ, and $\Pr(C \mid \boldsymbol{f_G})$ in the role of σ. In this case, the probability space Ω is the set of possible classifications $\{c_1, \ldots, c_n\}$. Therefore, we define $\delta_{\boldsymbol{G}}(\boldsymbol{f}) = D(\Pr(C \mid \boldsymbol{f}), \Pr(C \mid \boldsymbol{f_G}))$.

Of course, in order to have a metric which allows us to compare one feature set $\boldsymbol{G}$ to another, we must integrate the values $\delta_{\boldsymbol{G}}(\boldsymbol{f})$ for different feature vectors $\boldsymbol{f}$ into a single quantity. Naively, we might think to simply sum the cross-entropy for the different feature vectors, or to consider the maximum cross-entropy over all feature vectors. Neither of these ideas take into consideration the fact that some feature vectors are far more likely to occur than others, and that we might not mind making a larger mistake in certain rare cases. Therefore, we want to find a feature set $\boldsymbol{G}$ for which $\Delta_{\boldsymbol{G}} = \sum_{\boldsymbol{f}} \Pr(\boldsymbol{f}) \delta_{\boldsymbol{G}}(\boldsymbol{f})$ is reasonably small.

Clearly, the feature set that minimizes this quantity is simply $\boldsymbol{F}$, since that maintains the exact distribution. This suggests that we use a backward elimination algorithm, where at each state we eliminate a feature F_i in a way that allows us to remain as close to this distribution as possible. Intuitively, we use a greedy algorithm where we eliminate the feature F_i which would cause us the smallest increase in Δ. That is, we have a current feature set $\boldsymbol{G}$, initially set to $\boldsymbol{F}$. At each stage, we want to eliminate the feature F_i such that $\Delta_{(\boldsymbol{G}-\{F_i\})}$ is as close as possible to $\Delta_{\boldsymbol{G}}$.

Unfortunately, it is impractical to simply implement this idea as described, since the computation of $\Delta_{\boldsymbol{G}}$ is exponential in the number of features in our domain. Furthermore, we cannot really compare our approximate distribution to the true conditional distribution $\Pr(C \mid \boldsymbol{F})$, since the precise distribution is not available to us. Rather, we have a training set which provides us only a rough approximation to it. In those cases where we have a large number of features, the number of data instances in our training set corresponding to any particular assignment $\boldsymbol{f}$ will be very small. Therefore, as the number of features grow, our ability to use the training set to approximate this conditional distribution decreases (exponentially).

As we now show, we can utilize ideas from probabilistic reasoning (Pearl 1988) to circumvent this problem (to some extent). Intuitively, features that cause a small increase in Δ are those that give us the least additional information beyond what we would obtain from the other features in $\boldsymbol{G}$. We can capture this intuition via the formal notion of *conditional independence*.

Definition 1 *Two sets of variables (e.g., F_i or C) are said to be* conditionally independent *given some set of variables $\boldsymbol{X}$ if, for any assignment of values $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{x}$ to the variables $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{X}$ respectively,* $\Pr(\boldsymbol{A} = \boldsymbol{a} \mid \boldsymbol{X} = \boldsymbol{x}, \boldsymbol{B} = \boldsymbol{b}) = \Pr(\boldsymbol{A} = \boldsymbol{a} \mid \boldsymbol{X} = \boldsymbol{x})$. *That is, $\boldsymbol{B}$ gives us no information about $\boldsymbol{A}$ beyond what is already in $\boldsymbol{X}$.*

Proposition 1 *Let $\boldsymbol{G}$ be a subset of features and F_i be a feature in $\boldsymbol{G}$. Then F_i is conditionally independent of C given $\boldsymbol{G}' = \boldsymbol{G} - \{F_i\}$ if and only if $\Delta_{\boldsymbol{G}'} = \Delta_{\boldsymbol{G}}$.*

Thus, we can eliminate a conditionally independent feature F_i from $\boldsymbol{G}$ without increasing our distance from the desired distribution. Intuitively, removing a feature which is "almost" conditionally independent will not make our distance grow too large.

While it is also impractical to test for conditional independence given $\boldsymbol{G}'$, this reformulation of the problem points the way to a solution. Intuitively, if all of the information in F_i is subsumed by the features in $\boldsymbol{G}'$, it is almost certainly subsumed by some subset of these features. After all, it is very unlikely that *all* of these (usually very many) features are actually required.

Definition 2 *Let $\boldsymbol{M}$ be some set of features which does not contain F_i. We say that $\boldsymbol{M}$ is a* Markov blanket *for F_i if F_i is conditionally independent of $(\boldsymbol{F} \cup C) - \boldsymbol{M} - \{F_i\}$ given $\boldsymbol{M}$. (Pearl 1988, p. 97).*

It is easy to see that if $\boldsymbol{M}$ is a Markov blanket of F_i, then it is also the case that the class C is conditionally independent of the feature F_i given $\boldsymbol{M}$. Therefore:

Corollary 2 *Let $\boldsymbol{G}$ be a subset of features and F_i be a feature in $\boldsymbol{G}$. Assume that some subset $\boldsymbol{M}$ of $\boldsymbol{G}$ is a Markov blanket of F_i. Then $\Delta_{\boldsymbol{G}'} = \Delta_{\boldsymbol{G}}$.*

However, the Markov blanket condition is stronger than conditional independence. It requires that $\boldsymbol{M}$ subsume not only the information that F_i has about C, but also about all of the other features. While it might be difficult to find such a set $\boldsymbol{M}$, use of Markov blankets as the basis for feature elimination has a number of very desirable properties.

Intuitively, we want to remove features for which we find a Markov blanket within the set of remaining features. We now show that features judged as unnecessary based on this criterion remain unnecessary during the rest of the process. Assume, for example, that we remove a feature F_i based on a Markov blanket $\boldsymbol{M}$. At some later phase, we might remove some other feature $F_j \in \boldsymbol{M}$. In general, the removal of F_j might now render F_i relevant again; that is, if we were to add F_i back in, we might not be able to remove it again. As we now show, this is not the case.

Theorem 3 *Let $\boldsymbol{G}$ be our current set of features, and assume that some (previously removed) feature $F_i \notin \boldsymbol{G}$ has a Markov blanket within $\boldsymbol{G}$. Let $F_j \in \boldsymbol{G}$ be some feature which we are about to remove based on some Markov blanket within $\boldsymbol{G}$. Then F_i also has a Markov blanket within $\boldsymbol{G} - \{F_j\}$.*

Proof: The proof is based on the basic independence properties of probability distributions, as described in (Pearl 1988, p. 84). We will use the notation $I(X, Y \mid Z)$ to denote the conditional independence of two variables or sets of variables X and Y given a set of variables Z. Let $\boldsymbol{M}_i \subseteq \boldsymbol{G}$ be the Markov blanket of F_i (note that this is not necessarily the same Markov blanket which we used in order to remove F_i in the first place); let $\boldsymbol{M}_j \subseteq \boldsymbol{G}$ be the Markov blanket which we are now using to remove F_j. It is straightforward to show that if $\boldsymbol{M}_i$ does not contain F_j, then it remains a Markov blanket for F_i even after the removal of F_j from $\boldsymbol{G}$. Therefore, consider the case where $F_j \in \boldsymbol{M}_i$ and define $\boldsymbol{M}_i = \boldsymbol{M}_i' \cup \{F_j\}$. We will show that $\boldsymbol{M}_i' \cup \boldsymbol{M}_j$ is a Markov blanket for F_i. Let X denote $\boldsymbol{G} - \{F_j\} - (\boldsymbol{M}_i' \cup \boldsymbol{M}_j)$. We need to show that $I(F_i, X \mid (\boldsymbol{M}_i' \cup \boldsymbol{M}_j))$. From the Markov blanket assumption for F_j and the Decomposition property, we have that $I(F_j, (X \cup \boldsymbol{M}_i') \mid \boldsymbol{M}_j)$. Using the Weak Union property, we obtain that $I(F_j, X \mid (\boldsymbol{M}_i' \cup \boldsymbol{M}_j))$. Similarly, we can derive that $I(F_i, (X \cup (\boldsymbol{M}_j - \boldsymbol{M}_i')) \mid \boldsymbol{M}_i' \cup \{F_j\})$, and therefore that $I(F_i, X \mid \boldsymbol{M}_i' \cup \boldsymbol{M}_j \cup \{F_j\})$. From these two facts, we can use the Contraction property to show the desired result. ∎

Thus, the Markov blanket criterion only removes attributes that are really unnecessary. As interesting is the fact that the converse is also true. Two types of attributes are generally perceived as being unnecessary: attributes that are irrelevant to the target concept, and attributes that are redundant given other attributes. The Markov blanket criterion captures both of these. Attributes that are irrelevant will be unconditionally independent of everything, so they will be removed based on a Markov blanket consisting of the empty set of features. Even if we have a set of attributes that are correlated only with each other, but are completely independent of the class variable, the Markov blanket criterion will remove them one by one: at each stage, the remaining irrelevant features will be used as a Markov blanket for the one we are trying to remove. If, on the other hand, we have a feature whose value is fully determined (or even probabilistically determined) by some set S, we will be able to remove it by using S as its Markov blanket.

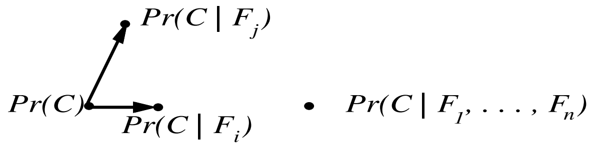

Figure 1: Forward vs. backward selection

It is interesting to compare our approach to another, seemingly very similar one, often used in the literature (Singh & Provan 1996). There, rather than starting with the full feature set and eliminating features, we begin with an empty set of features and add features one by one. Usually, the metric used to add features is *information gain*: we add to our current $\boldsymbol{G}$ the feature F_j that maximizes the expected cross-entropy between $\Pr(C \mid \boldsymbol{G})$ and $\Pr(C \mid \boldsymbol{G} \cup \{F_j\})$. It is easy to show that our idea of using a Markov blanket to estimate the cross-entropy can also be applied in the case of forward selection. Therefore, it might seem that the two approaches are essentially minor variants on the same theme. We claim that this is not the case: Our formal framework provides us with the tools to compare forward selection and backward elimination, and justifies our choice of backward elimination.

Recall that our goal was to remain as close as possible to the "correct" conditional distribution $\Pr(C \mid \boldsymbol{F})$. By removing features that only take "small steps" away from this distribution, we can remain close to it. By contrast, the forward selection scheme starts out with the prior distribution $\Pr(C)$ given no features. It then tries to take "large steps" away from that distribution. If the goal of this process is to get as close as possible to the "right" distribution, the problem becomes clear. There is no guarantee that taking a large step away from initial distribution actually gets us closer to the goal distribution. For example, as illustrated in Figure 1, adding F_j might let us take a much larger step than adding F_i, but the resulting distribution $\Pr(C \mid F_j)$ is actually further from the "right" distribution than $\Pr(C \mid F_i)$. As we show in Section 4, this behavior actually occurs on some of our data sets.

3 An Approximate Algorithm

Previously, we showed how we can eliminate a feature F_i from a candidate feature set $\boldsymbol{G}$ by finding a Markov blanket $\boldsymbol{M}$ for F_i. Unfortunately, there might not be a full Markov blanket for a feature, but rather only an approximate one that subsumes the information content of the feature. Furthermore, finding either a true or an approximate Markov blanket might be hard. We now present a simple algorithm which provides a heuristic approach to dealing with this problem. Broadly, our algorithm iteratively selects one candidate set $\boldsymbol{M}_i$ for each feature F_i, and uses a rough heuristic to estimate how close $\boldsymbol{M}_i$ is to being a Markov blanket for F_i; the feature F_i for which $\boldsymbol{M}_i$ is closest to being a Markov blanket is eliminated, and the algorithm repeats. Our intuition for constructing a candidate Markov blanket is as follows: Assume that F_i does, in fact, have a Markov blanket $\boldsymbol{M}_i$. We can think of F_i as *directly influencing* the features in $\boldsymbol{M}_i$. Therefore, these features will tend to be quite strongly correlated with F_i. Other features, on the other hand, are conditionally independent of F_i given $\boldsymbol{M}_i$. Thus, F_i influences them only indirectly, via $\boldsymbol{M}_i$. There is a well-known "folk-theorem" that probabilistic influence tends to attenuate over distance; that is, direct influence is typically stronger than indirect influence. Therefore, we heuristically choose, as an approximation to the Markov blanket, some set of K features which are strongly correlated with F_i. We tested a number of feature "correlation" metrics including statistical correlation, mutual information between features, class conditional mutual information and "pair-wise" cross-entropy (decribed below). The latter metric provided the best initial results and was used in the experiments reported here.

We now want to figure out how close $\boldsymbol{M}_i$ is to being a Markov blanket for F_i. Unfortunately, evaluating the conditional independence expression in Definition 2 is typically very expensive. We try to approximate this notion by observing that, if $\boldsymbol{M}_i$ is really a Markov blanket for F_i, then $D(\Pr(C \mid \boldsymbol{M} = \boldsymbol{f_M}, F_i = f_i), \Pr(C \mid \boldsymbol{M} = \boldsymbol{f_M})) = 0$ for any assignment of feature values $\boldsymbol{f_M}$ and f_i to $\boldsymbol{M}$ and F_i respectively. We therefore define the expected cross-entropy:

$$\delta_{\boldsymbol{G}}(F_i \mid \boldsymbol{M}_i) = \sum_{\boldsymbol{f}_{\boldsymbol{M}_i}, f_i} \Pr(\boldsymbol{M}_i = \boldsymbol{f}_{\boldsymbol{M}_i}, F_i = f_i) \cdot D(\Pr(C \mid \boldsymbol{M} = \boldsymbol{f_M}, F_i = f_i), \Pr(C \mid \boldsymbol{M} = \boldsymbol{f_M})).$$

If $\boldsymbol{M}_i$ is, in fact, a Markov blanket for F_i, then $\delta_{\boldsymbol{G}}(F_i \mid \boldsymbol{M}_i) = 0$. (This follows from the same techniques used in Corollary 2.) Hopefully, if it is an approximate Markov blanket, then this value will be low.

These approximations result in the following algorithm: We begin by computing the cross-entropy of the class distribution given pairs of features, $\gamma_{ij} = D(\Pr(C \mid F_i = f_i, F_j = f_j), \Pr(C \mid F_j = f_j))$ of every pair of features F_i and F_j. We then instantiate $\boldsymbol{G}$ to $\boldsymbol{F}$, and iterate the following steps until some prespecified number of features have been eliminated: (1) For each feature $F_i \in \boldsymbol{G}$, let $\boldsymbol{M}_i$ be the set of K features F_j in $\boldsymbol{G} - \{F_i\}$ for which γ_{ij} is smallest. (2) Compute $\delta_{\boldsymbol{G}}(F_i \mid \boldsymbol{M}_i)$ for each i. (3) Choose the i for which this quantity is minimal, and define $\boldsymbol{G} = \boldsymbol{G} - \{F_i\}$.

This algorithm is simple and fairly easy to implement. However, it is clearly suboptimal in many ways, particularly due to the very naive approximations that it uses. We now discuss the consequences of this and some ways in which the algorithm can be improved. First, the current algorithm eliminates a prespecified number of features, and constructs $\boldsymbol{M}_i$ sets of a fixed prespecified size K. It is easy to have the algorithm stop automatically when the expected cross-entropy estimate for dropping any remaining feature gets too large. It is also fairly straightforward to extend the algorithm to pick a different size $\boldsymbol{M}_i$ based on the number of features which were highly correlated with F_i. There is, however, an important tradeoff that must be kept in mind. Theoretically, the larger the conditioning set, the likelier it is to subsume all of the information in the feature, thereby forming a Markov blanket. On the other hand, larger conditioning sets fragment our training set into small chunks (corresponding to the different assignment of values to the features in $\boldsymbol{M}_i$), reducing the accuracy of our probability and hence cross-entropy estimates. Therefore, it is crucial, when doing this modification, to have a penalty term associated with adding additional features to $\boldsymbol{M}_i$.

More importantly, we would like to improve our techniques for choosing the candidate Markov blankets $\boldsymbol{M}_i$ and for evaluating how close each one is to fulfulling the Markov blanket assumption. In particular, the expected cross-entropy does not really test for the Markov blanket property: The expected cross-entropy will also have value 0 if F_i is conditionally independent of C given $\boldsymbol{M}_i$. But we have already pointed out that conditional independence is a weaker property than the Markov blanket assumption. In fact, using conditional independence as a selection criterion can lead to counterintuitive behavior. For example, as we can see in our results, increasing the size K of the conditioning set can actually cause the results to degrade. While some of this degradation is due to fragmentation of the training set (see below), some of it is caused by the fact that conditional independence is not a monotonic property. That is, it is possible for a certain feature to be conditionally independent of C given some conditioning set $\boldsymbol{M}$, but strongly correlated with C given a strict superset of $\boldsymbol{M}$. In a way, this fact is not surprising. It is well-known that additional information can cause correlations that were not present before to appear (Pearl 1988).

To illustrate this phenomenon in our context, consider a text classification problem, where the data instances are documents, the features are the presence or absence of a word, and the classes are document topics. The word *mining* is not significantly correlated with the topic *machine-learning*. Therefore, if we were to run our algorithm with $K = 0$, we would probably eliminate *mining* fairly early. However, this word is strongly correlated with the word *data*; moreover, if we condition on the presence of the word *data*, there is a strong correlation between the word *mining* and the topic *machine learning*. Thus, by putting the word *data* into our conditioning set $\boldsymbol{M}$, we have caused a seemingly irrelevant word to become relevant. The converse can also occur, so that we can get the estimated "relevance" of a feature fluctuating multiple times as we change K. We believe that the performance of our algorithm will be significantly improved by the use of more refined techniques (e.g., Bayesian methods) to choose a candidate (or even several candidates) Markov blanket, and by the use of a more precise formula for evaluating how close the different candidates are to fulfilling the requirement.

4 Results

In order to empirically test our theoretical model for feature selection as implemented by our approximate algorithm, we ran a number of experiments on both artificial and real-world data. These datasets include: the Corral data which was artificially constructed by John *et al* (1994) specifically for research in feature selection; the LED24, Vote, and DNA datasets from the UCI repository (Murphy & Aha 1995); and two datasets which are a subset of the Reuters document collection (Reuters 1995). These datasets are detailed in Table 1. We selected these datasets as they are either well understood in terms of feature relevance or they contain many features and are thus good candidates for feature selection.

Dataset	No. Classes	No. Features	Training Set Size	Testing Set Size
Corral	2	6	32	128
LED24	10	24	3200	5-fold CV
Vote	2	48*	435	5-fold CV
DNA	3	180*	3186	5-fold CV
Reuters1	3	1675	337	5-fold CV
Reuters2	3	1646	379	5-fold CV

Table 1: Datasets. *Denotes Boolean encoding.

We first analyze the artificial domains. The Corral dataset has been noted by previous researchers (John, Kohavi, & Pfleger 1994) as particularly difficult for filter methods since, of the 6 features in this domain, the target concept is a Boolean function of only four of the features: $(A \wedge B) \vee (C \wedge D)$. The fifth feature is entirely irrelevant and the sixth feature is "correlated" with the target concept in that it matches the class label 75% of the time. Thus, many filter approaches which use forward selection are likely to always select the correlated feature. This poses a problem for certain induction methods; C4.5, for example, is likely to initially split on the correlated feature, thus fragmenting the data enough that the true target concept cannot be recovered in the subtrees. One should note, however, that, due to the disjunctive nature of the target function, the Naive Bayesian classifier is actually better off with the correlated feature than without it. This seems to be more a shortcoming of the simplicity of this induction method than a flaw with feature selection methods that eliminate the correlated feature.

We verified this analysis experimentally, finding that forward selection (even with conditioning) always selects the correlated feature (thereby taking a "large" step in a suboptimal direction, as in Figure 1). Running backward elimination with conditioning, on the other hand, avoids this problem: we eliminate the correlated feature, since it has no effect on the class distribution for the function given the features that determine the target concept (or some large subset thereof). When we conditioned on 2 or more features and set the algorithm to drop 2 features, it consistently eliminated both the correlated and irrelevant features.

In the LED24 domain, we find a situation (albeit artificial) where conditioning on correlated features actually makes it more difficult to determine an appropriate subset of features. This domain contains 7 relevant and 17 irrelevant features. Moreover, the class label in the LED24 domain entirely determines the value of each relevant feature (modulo a noise term), whereas the irrelevant features are random. Thus, there is no dependence between features given the class label. As a result, we would expect that conditioning on correlated features would only confuse our algorithm by forcing it to unnecessarily estimate a larger number of probability values with the same amount of data, thus leading to poorer estimates. Again, this conjecture was verified experimentally, as our method in fact only consistently selected the 7 relevant features when we conditioned on no variables.

To test how our method of feature subset selection affected classification, we employed both a Naive Bayesian classifier (Duda & Hart 1973) and C4.5 (Quinlan 1993) as induction algorithms; these were applied both to the original datasets and to the datasets filtered through our feature selection algorithm (using both forward selection and backward elimination). Accuracy results for the UCI data are given in Table 2.

As seen in the accuracy results for Corral (using C4.5), Vote (using Naive Bayes with aggressive feature elimination), and DNA, selection of the appropriate feature set can have a large impact on classification accuracy. In the DNA domain we see some of the most dramatic results: accuracy improvements after eliminating 100, or even 150, of the 180 features with our method! A two-tailed paired T-test over the cross-validation folds reveals statistically significant improvements ($P < 0.10$) in accuracy for the Vote domain using Naive Bayes with aggressive feature selection and in the DNA domain for backward elimination used in conjunction with C4.5. Moveover, feature selection never significantly degraded accuracy in any of the real-world datasets tested. More importantly, however, is the fact that, in many domains, our feature selection algorithm can make dramatic reductions in the feature space and consequently improve running-time performance as well.

As far as computational expense, our filter approach also shows promise for scaling to larger domains. Both theoretical and empirical results show that the time complexity of our algorithm is quite low. Theoretically, it requires $O(n^2(m + \log n))$ operations for computing the pairwise cross-entropy matrix and sorting it, where n is the initial number of features and m is the number of instances. The subsequent feature selection process requires $O(rnmkc2^k)$ time, where r is the number of features to eliminate, k is the small, fixed number of conditioning features and c is the number of classes. Using caching schemes, it is possible to reduce the second term by close to a factor of n, due to the fact that an eliminated feature is likely to be in the $\boldsymbol{M}_i$ of only a few of the remaining features. Thus, we need only recompute a new $\boldsymbol{M}_i$ and its expected cross-entropy for this small number of features.

Empirically, this low running time allows us to deal with very large domains in a reasonable amount of time. By way of comparison, Kohavi (1995) obtains similar accuracy results on the DNA dataset for Naive-Bayes and C4.5 using the wrapper approach, but notes that doing so takes 15 hours on a Sun SPARC 10. In our experiments, an inefficient implementation of our algorithm (one that did not utilize clever data structures to reduce the running time) reduced the DNA dataset by 100 features using between 10 and 20 minutes on the same machine (depending on the number of conditioning variables). This is a time savings of nearly two orders of magnitude! Moreover, since our approach is a filter method, we do not need to re-run the algorithm for every induction algorithm we choose to run on a reduced-feature dataset.

The ability to deal effectively with very high-dimensional datasets allows us to apply our work to the domain of information retrieval. (In fact, this was part of the original motivation for our work.) In these applications, we have a feature for every word in the corpus, which leads to an overwhelming number of features. Therefore, such datasets present an exceptional challenge for many feature selection algorithms. In particular, feature selection using a wrapper method is simply intractable due to the prohibitive cost of running an induction algorithm thousands of times on very high-dimensional data. Hence, an efficient filter method, akin to the one method described here, is the only suitable approach.

To test this empirically, we constructed two high-dimensional datasets from the Reuters collection, each of which contains articles on three topics (classes). The first subset, Reuters1, is comprised of articles on the topics *coffee, iron-steel,* and *livestock*. These topics are not likely to have many meaningful overlapping words. Reuters2, on the other hand, contains articles on *reserves, gold*, and the *gross national product*, which are likely to have many similar words used in different contexts across topics. Each article was encoded into a binary vector, where each feature denoted whether a particular word occured in the article or not. As a simple pre-processing step, all words which occurred less than 3 times in each dataset were eliminated simply as a means for removing extremely rare words such as unique names.

We ran our feature selection algorithm on the Reuters

Dataset	# Features Orig./Final	K	Naive Bayes Accuracy Orig.	Fwd.	Bckwd.	C4.5 Accuracy Orig.	Fwd.	Bckwd.
Corral	6 / 4	0	90.6	84.4	84.4	81.2	81.2	75.0
		1		81.3	84.4		75.0	87.5
		2		81.3	87.5		75.0	100.0
		3		81.3	87.5		81.2	100.0
		4		81.3	87.5		81.2	100.0
LED-24	24 / 14	0	72.1 ± 2.1	72.1 ± 1.0	72.1 ± 1.0	71.1 ± 1.2	71.3 ± 1.2	71.3 ± 1.3
		1		71.9 ± 0.9	72.1 ± 0.7		71.0 ± 1.0	70.9 ± 1.2
		2		72.2 ± 1.4	72.4 ± 1.4		71.9 ± 1.0	71.3 ± 0.9
LED-24	24 / 7	0	72.1 ± 2.1	72.8 ± 1.5	72.8 ± 1.5	71.1 ± 1.2	72.1 ± 0.9	72.1 ± 0.9
		1		72.2 ± 1.8	72.2 ± 1.8		71.3 ± 1.9	71.3 ± 1.9
		2		72.1 ± 1.5	72.1 ± 0.8		71.6 ± 1.3	71.5 ± 1.1
Vote	48 / 28	0	90.1 ± 1.8	90.1 ± 1.8	90.1 ± 1.8	95.2 ± 1.5	95.7 ± 1.5	95.7 ± 1.5
		1		90.1 ± 2.7	90.1 ± 2.7		95.0 ± 2.8	95.2 ± 2.8
		2		90.3 ± 3.4	90.3 ± 3.4		94.5 ± 1.3	94.7 ± 1.3
Vote	48 / 8	0	90.1 ± 1.8	92.0 ± 2.7	92.0 ± 2.7	95.2 ± 1.5	95.4 ± 2.9	95.4 ± 2.9
		1		93.6 ± 1.8	92.7 ± 2.5		95.9 ± 1.5	95.7 ± 1.5
		2		95.2 ± 2.6	93.1 ± 5.4		95.0 ± 2.5	96.0 ± 2.2
DNA	180 / 80	0	94.0 ± 0.6	95.0 ± 0.5	95.0 ± 0.5	92.3 ± 0.7	93.6 ± 0.7	93.5 ± 0.7
		1		95.4 ± 0.7	95.5 ± 0.9		93.4 ± 0.8	93.3 ± 0.6
		2		94.9 ± 0.9	94.8 ± 0.6		93.6 ± 1.4	93.5 ± 1.1
DNA	180 / 30	0	94.0 ± 0.6	94.1 ± 1.0	94.1 ± 1.0	92.3 ± 0.7	93.6 ± 0.2	93.6 ± 0.3
		1		94.2 ± 1.1	94.3 ± 1.1		93.6 ± 0.8	93.4 ± 0.7
		2		92.3 ± 1.8	93.8 ± 1.0		91.7 ± 1.5	93.3 ± 1.0

Table 2: Accuracy percentages for Naive-Bayes and C4.5 using feature selection.

Dataset	# Features Orig./Final	K	Naive Bayes Accuracy Orig.	Fwd.	Bckwd.	C4.5 Accuracy Orig.	Fwd.	Bckwd.
Reuters1	1675 / 675	0	90.5 ± 5.2	90.7 ± 5.1	90.7 ± 5.1	95.2 ± 2.5	95.5 ± 2.4	94.9 ± 2.7
		2		91.9 ± 3.1	96.1 ± 3.3		95.8 ± 2.5	96.4 ± 2.0
Reuters2	1646 / 646	0	89.9 ± 2.2	92.3 ± 1.7	92.3 ± 1.7	91.5 ± 1.2	93.0 ± 0.6	93.0 ± 0.6
		2		90.6 ± 2.5	94.1 ± 2.8		91.5 ± 2.9	94.9 ± 2.0

Table 3: Accuracy percentages for Reuters text datasets using feature selection.

datasets in order to reduce the feature space by 1000 features — down to nearly 1/3 its original size! The results of these experiments are shown in Table 3, which also includes results for forward selection. In the Reuters1 domain, where we expect more distinct terms between topics (and hence less feature interaction) we see that both feature selection methods have a tendency to work comparably well without conditioning information, producing good accruacy results. When conditioning is introduced, however, the results using the backward elimination method clearly dominate those obtained using forward selection. Employing forward selection is simply inadequate for finding good features with conditioning information. In the second Reuters domain, however, we see that employing backward elimination allows the algorithm to effectively make use of conditioning information to increase the accuracies of both induction methods in a drastically reduced feature space. The results for Reuters2 using our backward elimination method with $K = 2$ are statistically significant ($P < 0.10$) improvements over the accuracy on the original dataset.

As for resource comsumption, eliminating 1000 features from the Reuters datasets took about 2.5 hours on a Sun SPARC 10. By way of comparison, a rough estimate of the time required by a wrapper approach such as that of Caruana & Freitag (1994) or John *et al* (1994) to eliminate this many features is on the order of thousands of hours, assuming the method does not get caught in a local minima first and prematurely stops eliminating attributes as a result.

5 Conclusions

We have presented a theoretically justified model for optimal feature selection based on using cross-entropy to minimize the amount of predictive information lost during feature elimination. Within this theoretical framework, we prove several desirable properties of using such a method for feature selection. Moreover, we present an algorithm that approximates our theoretical model and provide extensive empirical testing. We show that this algorithm is effective at drastically reducing the feature space in many learning tasks while also helping to improve accuracy in many cases.

Our method attempts to eliminate features in a way that keeps the conditional probability of the class given the features as close to the original distribution as possible. This is not the same as attempting to maintain the same *classification* for each instance. While this too is a desirable goal, it is necessarily specific to a particular induction algorithm. Rather, we focus on an algorithm-independent paradigm for feature subset selection, viewing an induction algorithm as a biased method for approximating the probability distribution of class labels given features and transforming this distribution into a classification. We stay free of the bias of a particular induction algorithm by simply maintaining as much as possible the underlying conditional distribution of class labels that the induction algorithm attempts to approximate.

Due in large part to its induction-bias-free nature, our approach sometimes provides only modest gains in accuracy. However, it can be used as a tool for obtaining much better accuracy for very high-dimensional datasets. Currently, when faced with such a domain, we are essentially forced to use simple, less computationally intensive induction algorithms such as Naive Bayes. The use of our feature selection algorithm as a pre-processing step will enable the use of more powerful, but computationally expensive, induction algorithms (such as full Bayesian classifiers). In this way, we hope to make such induction methods much more applicable to problems with many features. Furthermore, although we argue that wrapper methods are, a priori, too computationally expensive for such datasets, we can use them on the feature-reduced datasets resulting from our algorithm. This will allow us to produce a classifier which is optimized for accuracy with respect to a specific induction algorithm, by searching in the *filtered* feature space. We hope that these techniques, taken together, will allow us to effectively tackle induction problems in very large feature spaces.

Acknowledgements

This work was supported by NSF/ARPA/NASA under Stanford's Digital Libraries grant.

References

Almuallim, H., and Dietterich, T. G. 1991. Learning with many irrelevant features. In *Proc. AAAI-91*, 547–552. MIT Press.

Blumer, A.; Ehrenfeucht, A.; Haussler, D.; and Warmuth, M. K. 1987. Occam's razor. *Information Processing Letters* 24:377–380.

Caruana, R., and Freitag, D. 1994. Greedy attribute selection. In *Proc. ML-94*. Morgan Kaufmann.

Cover, T. M., and Thomas, J. A. 1991. *Elements of Information Theory*. Wiley.

Duda, R., and Hart, P. 1973. *Pattern Classification and Scene Analysis*. Wiley.

Fukunaga, K. 1990. *Introduction to Statistical Pattern Recognition*. Academic Press.

John, G.; Kohavi, R.; and Pfleger, K. 1994. Irrelevant features and the subset selection problem. In *Proc. ML-94*, 121–129. Morgan Kaufmann.

Kira, K., and Rendell, L. A. 1992. The feature selection problem: Traditional methods and a new algorithm. In *Proc. AAAI-92*, 129–134. MIT Press.

Kohavi, R. 1995. *Wrappers for Performance Enhancement and Oblivious Decision Graphs*. Ph.D. Dissertation, Stanford, Computer Science Dept.

Kullback, S., and Leibler, R. A. 1951. On information and sufficiency. *Annals of Mathematical Statistics* 22:76–86.

Langley, P., and Sage, S. 1994. Induction of selective bayesian classifiers. In *Proc. UAI-94*, 399–406. Seattle, WA: Morgan Kaufmann.

Murphy, P. M., and Aha, D. W. 1995. UCI repository of machine learning databases. http://www.ics.uci.edu/~mlearn/MLRepository.html.

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

Quinlan, J. R. 1993. *C4.5: Programs for Machine Learning*. Los Altos, California: Morgan Kaufmann.

Reuters. 1995. Reuters collection available via anonymous ftp. ftp://ciir-ftp.cs.umass.edu/pub/reuters1.

Singh, M., and Provan, G. M. 1996. Efficient learning of selective bayesian network classifiers. Submitted for publication.