Feature Selection in MLPs and SVMs based on Maximum Output Information

Vikas Sindhwani, Subrata Rakshit, Dipti Deodhare, Deniz Erdogmus, Jose Principe, and Partha Niyogi

***Abstract*— We present feature selection algorithms for multi-layer Perceptrons (MLPs) and multi-class Support Vector Machines (SVMs), using mutual information between class labels and classifier outputs, as an objective function. This objective function involves inexpensive computation of information measures only on discrete variables; provides immunity to prior class probabilities; and brackets the probability of error of the classifier. The Maximum Output Information (MOI) algorithms employ this function for feature subset selection by greedy elimination and directed search. The output of the MOI algorithms is a feature subset of *user-defined* size and an associated *trained* classifier(MLP/SVM). These algorithms compare favorably with a number of other methods in terms of performance on various artificial and real-world data sets.**

***Index Terms*— Feature Selection, Support Vector Machines, Multi-Layer Perceptrons, Mutual Information**

I. Introduction

A supervised learning algorithm attempts to induce a decision rule from which to categorize examples of different concepts by generalizing from a set of training examples. A critical ingredient of a successful attempt is to provide the learning algorithm with an optimal description of the concepts. Since one does not *a-priori* know what attributes constitute this optimal description, a number of irrelevant and redundant features are recorded. Many learning algorithms suffer from the *curse of dimensionality*, *i.e.* the time and data requirements for successful induction may grow very fast as the number of features increases [5], [12]. Unnecessary features, in such a case, serve only to increase the learning period. They add undesirable complexity to the underlying probability distribution of the concept label which the learning algorithm tries to capture.

John, Kohavi & Pfleger [12] discuss notions of relevance and irrelevance that partition the set of features into useful degrees of dispensability. According to their definitions, *Irrelevant features* do not participate in defining the unknown concepts; *weakly relevant features* possess redundant information and can be eliminated if other features subsuming this information are included; and *strongly relevant features* are indispensable. Given the task of selecting K out of N features, as K is decreased one expects an ideal selection algorithm to first discard irrelevant features, then redundant features and finally start eliminating the strongly relevant features according to the strength of their relevance. While this is desired, it usually cannot be directly implemented as these properties of features are hard to determine *a-priori*. Thus the model selection problem (how many features) is usually driven by external constraints like building compact classifiers, data availability constraints or need for visualization in lower dimensions. In this paper we address the problem of which features to select, given a model selection (number of features).

In this paper, we are concerned with developing information-theoretic methods to address the optimal feature subset selection problem. Guyon & Elisseeff [9] review several approaches advocated in machine learning literature. In the *filter* approach, feature selection is independent of the learning algorithm. Many filters detect irrelevant features by estimating the importance of each feature independent of other features [13], [15]. Other filters perform a more complex search over multiple features in order to additionally identify and eliminate redundancy [1]. In the *wrapper* approach, the objective function for selection is a measure of classifier performance. Wrappers typically involve expensive search routines and are considered superior because they incorporate the inductive bias of the classifier [12].

Several information-theoretic solutions to this problem have been proposed and may also be categorized as described above. Filters like *Information Gain*, routinely used on very high dimensional problems like text classification [28], use mutual information $I(X_i, Y)$ between *a single feature* X_i and the class variable Y, to estimate the relevance of feature X_i. Yang & Moody [27] select *the two features* that maximize the joint mutual information $I(Y; X_i, X_j)$ over all possible subsets of two features and class labels. For optimization over more than two variables, search heuristics are used. Battiti [1] proposes an algorithm called *Mutual Information Feature Selection* (MIFS) that greedily constructs the set of features with high mutual information with the class labels while trying to minimize the mutual information among chosen features. Thus, the i^{th} feature X_i included in the set, maximizes $I(Y; X) - \beta \sum_{j=i}^{i-1} I(X; X_j)$ over all remaining features X for some parameter β. The *Maximum Mutual Information Projection* (MMIP) feature extractor, developed by Bollacker [2], aims to find a linear transform by maximizing, at each step, the mutual information between the class variable and a single direction in the feature subspace orthogonal to previously found directions. The *Separated Mutual Information Feature Extractor* (SMIFE) is a heuristic where a matrix of joint mutual information between class variables and each pair of features is constructed. Following an analogy with Principal Component Analysis (PCA), the eigenvectors of this matrix are found and the principal components are then used for feature transformation.

We observe two shortcomings of these methods: Firstly, any mutual information computation involving continuous features demands large amounts of data and high computational

complexity. Not only are features typically *continuous*, they will be highly numerous in problems of interest in feature selection. Secondly, all these methods are in the vein of the filter approach. Their objective functions disregard the classifier with which the selected features are to be used. As pointed out in [1] "... there is no guarantee that the optimal subset of features will be processed in the optimal way by the learning algorithm and by the operating classifier."

In this paper, we address both these shortcomings simultaneously. We formulate an information theoretic objective function that involves computation of mutual information *only between discrete random variables*. This objective function is the mutual information between the class labels and the discrete labels output by the classifier. Since, in a typical classification task, *the number of classes is much smaller than the number of features*, this suggests substantial gains in efficiency. We discuss theoretical justifications for using such an objective function in terms of upper and lower bounds on the error probability of the classifier, as well as justifications in terms of its merits as a performance evaluation criterion.

This objective function is used to design wrappers to select K features out of N for learning multi-layer Perceptrons (MLPs) [18] and multi-class Support Vector Machines (SVMs) [26]. Since the objective function is the mutual information between class labels and the output of the classifier, the class of algorithms we present are called *Maximum Output Information* (MOI) algorithms. Indirect feature crediting is achieved through an output side entropy evaluation. The MOI wrapper algorithm, implemented for both SVMs (called MOI-SVM) and MLPs (called MOI-MLP), then conducts a directed search by iteratively refining the feature subset. It aims to discover a subset with which the classifier delivers maximum information via its output. The maximization of output information may be seen as an extension of Linker's Infomax principle [14]: *Each layer of a multi-layered perceptual network should strive to transmit maximum information about its input to the next layer*. The principle is now being applied to the classifier as a whole, since we are interested in evaluating a trained classifier. A key difference is that the information measured here is information in the classifier output *specific to a desired task*. It may be noted that Linsker's approach in [14] was meant for *unsupervised* learning, where there could be no task specific measures.

Additionally, we utilize the applicability of SVMs on very high dimensional classification problems, to design two variants of MOI based on greedy elimination. The sparsity of the SVM solution is exploited in all these schemes.

The results presented in this paper illustrate the performance of these algorithms on artificial and real-world data sets. Experimental comparisons are made with several other methods for feature selection.

II. An Information Theoretic Objective Function

We consider the standard setting of the problem of pattern classification: A pattern drawn from a set $\mathcal{X}(= \mathcal{X}_1 \times .... \times \mathcal{X}_N)$, constructed from the N features $\{\mathcal{X}_i, i = 1...N\}$, is associated with a category whose label belongs to the set $\mathcal{Y} = \{1, 2, ...k\}$. When given a training sample consisting of a finite number of pairs of patterns and corresponding class labels (drawn according to the underlying unknown joint probability distribution $P_{\mathcal{X}\times\mathcal{Y}}$), the supervised machine learning framework aims to discover a function $f : \mathcal{X} \rightarrow \mathcal{Y}$, from a hypothesis class $\mathcal{H}$ of functions, that exhibits good generalization on unseen patterns. Let Y , $Y_f(= f(x), x \in \mathcal{X})$ be the discrete random variables over $\mathcal{Y}$ describing the unknown true label and the label predicted by the classifier respectively. (Note that discrete labels are often obtained from real-valued outputs. We include this operation as part of the classifier.)

Let G be a subset of features, *i.e*, $G \subset \{X_1, X_2.....X_N\}$. Let $\mathcal{H}_G$ denote the restriction of $\mathcal{H}$ on G, *i.e*, the class of functions in $\mathcal{H}$ that map G to $\mathcal{Y}$. The optimization problem we would ideally like to solve, for selecting K features out of N, is the following:

$$f^*_{G^*} = \max_{G:|G|=K} \max_{f\in\mathcal{H}_G} I(Y; Y_f) \tag{1}$$

where $I(Y; Y_f)$ is the mutual information between Y and Y_f. Since this is the average rate of information delivered by the classifier via its output, we refer to this quantity as classifier *output information* and sometimes also denote it by I_f, in subsequent discussion.

The inner maximization constitutes the problem of training a classifier, for a given set of input features. This is usually done such as to minimize a training objective function related to the error rate of the classifier, while the criterion above calls for an information maximization. This section deals with the relation between these two measures (probability of error and output mutual information) and the rationale for substituting one for the other. The outer maximization deals with the feature selection problem, once again with an information theoretic approach. This will be addressed in the next section. Note that an optimization over K, the model selection problem, has been omitted as it is often dependent on external factors like data availability and resources available for implementing the classifier. We now discuss the estimation of $I(Y; Y_f)$ on a labeled data set and then argue in favor of such an information theoretic evaluation.

A. Estimation of Output Information

Given a classifier f and a labeled data set, we may estimate the information delivered by the classifier about the unknown class label as follows: Let $|\mathcal{Y}| = k$ be the number of classes; Let Q_f be the confusion matrix, where q_{ij} is the number of times over the labeled data set, an input pattern belonging to class i is classified by f as belonging to class j. Clearly, the diagonal elements q_{ii} represent all the correct classifications while the off-diagonal terms represent the errors. Note that in what follows a summation over i is a sum over values of Y, *i.e.,* over various rows of a given column of the confusion matrix. Similarly, a summation over j is a sum over values of Y_f, *i.e.,* across various columns of a given row of the confusion matrix. Both variables are summed from $1 \ldots k$. $S = \sum_{ij} q_{ij}$ is the total number of input samples. It is possible to estimate all relevant probabilities using this $k \times k$ matrix Q_f in order

to estimate $I(Y;Y_f)$ as follows:

$$\begin{aligned}\hat{P}(Y=i) &= \frac{\sum_j q_{ij}}{S}\\ \hat{P}(Y_f=j) &= \frac{\sum_i q_{ij}}{S}\\ \hat{P}(Y=i|Y_f=j) &= \frac{q_{ij}}{\sum_i q_{ij}}\end{aligned}$$

where $\hat{P}(Y=i)$ is the empirical prior probability of class i; $\hat{P}(Y_f=j)$ is the frequency with which the classifier outputs class j, and $\hat{P}(Y=i|Y_f=j)$ (more conveniently written as $\hat{P}_{ij}$) is the empirical probability of the true label being class i when the classifier outputs class j. The relevant empirical entropies are now given by:

$$\begin{aligned}\hat{H}(Y) &= \sum_i -\hat{P}(Y=i)\log(\hat{P}(Y=i))\\ \hat{H}(Y|Y_f=j) &= \sum_i -\hat{P}_{ij}\log\hat{P}_{ij}\\ \hat{H}(Y|Y_f) &= \sum_j \hat{P}(Y_f=j)H(Y|Y_f=j)\end{aligned}$$

and the estimated value of the mutual information between class labels and classifier output is given in terms of above entropies, simply by $\hat{I}(Y;Y_f)=\hat{H}(Y)-\hat{H}(Y|Y_f)$. Note that this mutual information computation involves only discrete variables that typically assume a small number of values.

B. Merits as an Evaluation Criterion

We now consider, briefly, the way output information $I(Y;Y_f)$ (or more conveniently I_f) differs from two widely used criteria, the root mean square error (RMSE) and classification accuracy (CA). The RMS error is very sensitive to the margin by which a misclassification occurs and is often dominated by outliers. It also penalizes correct classifications if the output values do not exactly match the labels. It is this graded nature of the RMSE that makes it a desirable objective function for training but a bad choice for classifier performance evaluation. On the other hand, both I_f and CA are sensitive to the number of misclassifications irrespective of the margins. This makes them insensitive to the outlier and the residual error problem, at the cost of making both non-differentiable functions of the classifier (*e.g*, MLPs or SVMs) parameters.

There are two other significant differences between I_f on one hand and both RMSE and CA on the other hand. I_f takes into account the input sampling bias, *i.e.*, the fact that all classes are not equally likely *a-priori*. In real applications, it is quite likely that there will be fewer data points for certain classes. For such problems, classifiers can often reduce their initial RMSE, or improve their initial CA, by learning to ignore the smaller classes. I_f, as an evaluation criterion, is immune to biased input samples (though it does not in any way solve the problem of training a classifier with such biased data).

The second major difference is the sensitivity of I_f to the *pattern* of errors. Neither RMSE nor CA take into account the distribution of errors across various classes. This is a significant issue for problems with a large number of classes. Consider the confusion matrices for three class problems given below.

$Y_f \rightarrow$ / $Y\downarrow$	(a) 1	2	3	(b) 1	2	3	(c) 1	2	3
1	15	0	5	16	2	2	1	0	4
2	0	15	5	2	16	2	0	1	4
3	0	0	20	1	1	18	1	1	48

The classification accuracy for all cases is $50/60 = 83\%$. $H(Y)$ for case (c) is 0.82 bits indicating little prior uncertainty as compared to $H(Y) = 1.58$ bits for cases (a) and (b) where the three classes have equal prior probability involving 20 examples in each. The observer can demonstrate good accuracy even without the classifier in (c) by labelling all instances as class 3. As the confusion matrix shows, even without constructing good decision boundaries for class 1 and class 2, the classifier achieves a high classification accuracy. However, $I_f(c) = 0.06$ bits indicating that the underlying classification task has not been solved by the classifier, whereas $I_f(a) = 0.96$ bits and $I_f(b) = 0.78$ bits. The classifier performs better on (a) because with the information that the classifier has output class 1 or 2, the observer can be confident about the true class of the input. In (b), when the classifier outputs class 1 or class 2 it maintains slightly greater uncertainty than (a) by sometimes also claiming for patterns of other classes. Notice that even though in (b), the classifier output 3 is more reliable than that in (a), the classifier overall performs better in (a) on account of greater reliability of its class 1 and class 2 outputs. Information measures tend to put a high premium on certainty. The third example, (c), shows how $I_f = H(Y) - H(Y|Y_f)$ takes into account, through the $H(Y)$ term, the input distribution in determining the performance of a classifier.

Consider a hypothetical binary classifier that classifies all instance of class 1 as class 2 and vice versa. Such a classifier has nil accuracy but delivers the very useful information that its output implies the other class with no uncertainty. This classifier can be well utilized by a *sentient observer* who does not take the output at face value, but rather uses the information it delivers. The formulated objective function I_f is capable of taking this into account and is unique in this respect to the best of our knowledge.

Finally, we compare I_f to another information theoretic objective function - the output cross entropy [20], [3]. This measure is actually closer in spirit to the RMSE than I_f. It measures the extent to which classifier outputs have converged to desired outputs, *i.e.*, the residual approximation error. It does not depend on classification accuracy or the actual pattern of misclassifications. Due to its dependence on the approximation error, the output cross entropy is a differentiable function of the classifier parameters. In a similar vein, one may look for a differentiable approximation of I_f. In this paper, a different approach is used. The non-differentiable I_f is used for evaluation, a differentiable approximation (like MSE for MLPs) to classifier error rate is used for training, and a link is established between the two to show that minimisation of error rate would lead to an approximate maximization of I_f.

C. Relationship to Error Probability

Information-theoretic inequalities have been derived, that establish a strong connection between the objective function described above and the performance of the classifier in terms of its error rate. Erdogmus and Principe [6] provide a family of upper and lower bounds on the misclassification probability $P_e(f)$ of a classifier by applying Jensen's inequalities in conjunction with Renyi's definitions of entropy and mutual information [16]. The tightest upper and lower bounds in this family involve only Shannon's definitions [21], and are as follows:

$$P_e(f) \geq \frac{H(Y) - I(Y;Y_f) - h(P_e(f))}{log(|\mathcal{Y}| - 1)} \quad (2)$$

$$P_e(f) \leq \frac{H(Y) - I(Y;Y_f) - h(P_e(f))}{\min_i H(Y|e, Y_f = i)} \quad (3)$$

where $h(P_e(f))$ is the binary Shannon entropy [1]; $H(Y|e, Y_f = i)$ is the (Shannon) entropy of the distribution over erroneous classes given that the classifier incorrectly outputs class i. Note that the lower bound is the familiar Fano bound [7] whose extension to an upper bound is made possible via Renyi's definitions. The bounds can be suitably modified to exclude terms involving $P_e(f)$ and to be applicable to 2-class problems [6]. Thus, the information transferred by the classifier $I(Y;Y_f)$, brackets its error rate from above and below.

Conversely, the error rate also brackets the quantity $H(Y) - I(Y,Y_f)$. In particular, (2) may be rearranged as a lower bound on I_f, other terms remaining constant. The denominator term prevents (3) from being used as an effective upper bound on I_f. One may consider the denominator to be highlighting the fact that it is possible to deliver useful information about the class labels without being accurate. Thus a low error rate guarantees a high information rate but a high error rate does not rule out a high information rate.

The extended Fano inequalities (2) , (3) theoretically confirm the intuition that a classifier, optimal in the sense of minimum error, maximizes the mutual information $I(Y;Y_f)$. Since typical training objective functions (like MSE in MLPs and margins in SVMs) attempt to achieve minimal error rates, it is justifiable (in that sense) to also use the minimization of training objective functions to maximize I_f.

III. Feature Selection by MOI

We now describe the components of the MOI algorithms that heuristically solve the optimization problem (1). First, note that the inner maximization in (1) requires training classifiers with $I(Y;Y_f)$ as the objective function. Since this function is not differentiable with respect to classifier parameters, this problem cannot be solved without resorting to non-differentiable optimization techniques like genetic algorithms. Differentiable substitutes such as cross-entropy [3], [20] may be used instead. However, in this paper, our approach is to approximate the optimization process rather than approximate the objective function. This is done by replacing (1) by the following problem:

$$f^*_{G^*} = \max_{G:|G|=K} I(Y, Y_{f^*_G}) \quad (4)$$

where f^*_G is a classifier that has been trained using any convenient training objective function with the feature set G. The *trained classifier* is evaluated according to its output information. In this paper, we are concerned with multi-layer Perceptrons (MLPs) and multi-class Support Vector Machines (SVMs). The training objective function for MLPs is RMSE and its minimization is performed using the popular error back-propagation algorithm [18]. For SVMs, a quadratic optimization problem is solved in order to maximize the margin of separation between examples of two classes either in the original input space or in an implicitly mapped higher dimensional space by the use of *kernel* functions [19], [26]. A common strategy to construct multi-class SVMs is the *one-against-rest* approach where a binary SVM is trained to separate each class from the other classes. A test example is labeled according to the maximum output among the binary SVMs [19].

Since the complexity (*e.g* as measured by VC-dimension [26]) of MLPs is proportional to the number of weights in the network [5], a large number of irrelevant features construct complex networks that may overfit the data and also make the training algorithm more prone to local minima. SVMs on the other hand, are claimed to be able to overcome the curse of dimensionality by maximizing the margin globally. Additionally, the SVM solution is sparse, involving a small subset of the training data [19], [26].

Such differences motivate different strategies for feature selection in MLPs and SVMs. Whereas training MLPs in lower dimensional feature spaces is preferable, intuitively, the capability to generalize well in very high dimensional spaces suggests that irrelevant features are implicitly identified in SVMs. This provides a motivation to *use* SVMs for feature selection, as well as to apply such selection procedures to improve their own performance. Also, *non-linear feature selection via different kernels* may be attempted. Feature selection may utilize *only the relevant examples* as discovered by the SVM training to construct a subset of relevant features. The MOI algorithms described below attempt to utilize these facts about the nature of the classifier.

A. Information Backpropagation

The problem of feature selection as formulated in (4) aims to select a feature subset of required size K from a full set of N features. Starting from a K-sized feature subset G, f^*_G in (4) is obtained by training the classifier using the features in G. We use error backpropagation in MLPs and margin maximization in SVMs, briefly mentioned above. These trained classifiers are evaluated by their output information. In order to perform the maximization of output information in (4), over all possible subsets of size K, we need to formulate a directed search algorithm that iteratively refines G. For this purpose, we need a reasonable heuristic to assign, to *each feature in* G, suitable

[1] $h(x) = -x \log x - (1-x)\log(1-x)$

credits for the information delivered by the classifier trained on G. This is done by the *information backpropagation* heuristic described in this section.

We visualize a trained classifier as transmitting information across multiple layers of components; starting from the input layer containing features to the output layer containing *class indicators*; one of which *fires* when the classifier is shown a pattern. The class indicators may be output neurons in an MLP or individual binary SVMs in a multi-class SVM. We measure the information transmitted by the trained classifier by evaluating it on a test set and computing I_f from the confusion matrix as described in Sec II(B). We then *back-propagate this information measure across the layers* of components of the classifier using a heuristic to measure how each component contributes to the information flow. At the end of this credit distribution, we obtain an estimate of the contribution made by each feature.

Credit Assignment for Class Indicators: It is desirable that the information credited to each class indicator be such that (a) It reflect the difference in *a-priori* uncertainty (prior to observing the output of the class indicator) and *a-posteriori* uncertainty (after observing the output of the class indicator). (b) It reflect the frequency of firing of the indicator. (c) The sum of credits across the indicators should add up to I_f. The first condition ensures that class indicators that fire *only for patterns of a particular class* are given more credit. Class indicators firing for patterns of multiple classes cause observer uncertainty. The second condition ensures that rarely firing class indicators get less credit. Such indicators might be either specializing in rare classes or failing to fire even when the input pattern is from their class. The third condition is a normalization constraint.

The usefulness of the knowledge of the firing of a particular indicator can be measured relative to a hypothetical worst case indicator whose firing only maintains maximum uncertainty about the actual class. The uncertainty associated with such an indicator is $log|\mathcal{Y}|$. The uncertainty about the class of a pattern given that class indicator j has fired is $H_S(Y|Y_f = j)$. Thus the usefulness of this indicator measured relative to the worst case indicator can be written as :

$$I_j = I_f \frac{P(Y_f = j)(\log|\mathcal{Y}| - H_S(Y|Y_f = j))}{\sum_i P(Y_f = i)(\log|\mathcal{Y}| - H_S(Y|Y_f = i))} \quad (5)$$

Each of the quantities involved can be easily estimated from the confusion matrix. Note that we have dealt with the issue of *a-priori* uncertainty of a class indicator, which is not well-defined, by taking it to mean *uncertainty prior to training*. Intuitively, this would imply the worst-case class indicator as used in the crediting above. This also ensures non-negative credits; and zero credits for worst-case performance. Thus, the individual binary SVMs in a multiclass SVM or the output neurons in an MLP are credited for their contribution to information flow according to (5).

Credit Assignment for Features: Backpropagating information further,we need to distribute credits across the next layer of components of the classifier. For SVMs we regard this layer to be the features themselves; for MLPs, this layer is the outer hidden layer. We are again guided by two considerations: (a) Credit assigned to a component must be proportional to the degree of influence it has on the components it connects to. (b) Credits must be normalized to add up to I_f.

This is implemented differently for SVMs and MLPs. Since the generalization performance of an SVM is deeply related to its margin [26], we compute the degree of influence of a feature on an SVM, by the sensitivity of the margin of the SVM to the feature. The squared reciprocal of the margin of an SVM is given by:

$$w^2 = \sum_{ij} \alpha_i \alpha_j y_i y_j K(\vec{x_i}, \vec{x_j})$$

where α_i is the Lagrange multiplier and y_i is the label corresponding to the i^{th} support vector $\vec{x}_i$; and K is the kernel function used [26], [19]. We compute the derivative based sensitivity to feature k in the following manner:

$$\mathcal{D}_k(w^2) = \sum_i |\sum_j \alpha_i \alpha_j y_i y_j \frac{\partial K(\vec{x_i}, \vec{x_j})}{\partial x_j^k}| \quad (6)$$

where x_j^k is the k^{th} feature in $\vec{x}_j$. We may now perform the normalization step. Feature k can be credited for the overall information flow according to:

$$I_k = \sum_r \left(I_r \frac{\mathcal{D}_k^r}{\sum_p \mathcal{D}_p^r} \right) \quad (7)$$

where r indexes the SVMs and p indexes the features; $\mathcal{D}_p^r$ is computed from (6) for feature p and SVM r; and the information credit I_r of the r^{th} SVM is calculated from (5). Recall that I_f in (5) is the output information of the multiclass SVM.

For multi-layer perceptrons, information backpropagation is performed across the multiple hidden layers. Consider the neuron J in the layer indexed by j and let the layer being fed by this layer be indexed by k. Denoting the output of a neuron in layer k as O_k and the weight of the interconnection between neurons k and j as w_{kj}, we define the credit for neuron J as:

$$I_J = \sum_k \left(I_k . \frac{|c_{kJ}|}{\sum_j |c_{kj}|} \right) \quad (8)$$

where we use the covariance $c_{kj} = Cov(O_k, w_{kj}.O_j)$ as the sensitivity measure. After using (5) to determine the credit for each output layer neuron, one can use (8) to recursively compute the credit for neurons in the non-output layers. In the end, the layer containing the features is credited.

Note that we have assumed that components in a layer are delivering mutually independent information so that feature crediting involves simple additive arithmetic. To capture dependence between connected components, it would be appropriate to compute their mutual information, given that we have committed to use information measures. Instead, we have used simple derivative and covariance based sensitivity so as to be consistent with the objective of avoiding discretization of continuous variables; and also to utilize the sparsity of the SVM solution (which makes Eqn (6) involve very few training patterns *i.e* only the support vectors). The error incurred in ignoring the correlation between intra-layer components is corrected to first order by the requirement of normalization in each layer.

B. Algorithms

The Information Backpropagation heuristic described above guides the directed search required to perform the maximization in (4) by projecting the output information $I(Y;Y_{f_G^*})$ onto the individual features in G. We now describe the algorithms that perform this directed search, in increasing order of their complexity. Let $\mathcal{X} = \{X_1, X_2, ..., X_N\}$ denote the full set of N features; $\mathcal{D}_{train}$ denote the training data set; $\mathcal{D}_{test}$ denote a test data set; $\mathcal{D}_{train}(G)$, and $\mathcal{D}_{test}(G)$ denote the training and test data sets restricted to the features in the subset G.

Algorithm 1 MOI Pseudo-Wrapper for SVMs

Require: : $0 < K \leq N$; $\mathcal{D}_{train}$; $\mathcal{D}_{test}$

1: Train multi-class SVM on $\mathcal{D}_{train}$. {Train using all N features. This identifies the support vectors and corresponding Lagrange multipliers for each binary SVM.}
2: Estimate its I_f on $\mathcal{D}_{test}$.
3: Assign Credits to each binary SVM by (5).
4: Assign Credits to each feature by (7).
5: Return best K features according to these credits.

Algorithm 2 MOI Backward Elimination Wrapper for SVMs

Require: $0 < K \leq N$; $\mathcal{D}_{train}$; $\mathcal{D}_{test}$

1: INITIALIZE: $G = \mathcal{X}$ {G is initialized as the set of all N features}
2: **while** $|G| > K$ **do**
3: Train multi-class SVM on $\mathcal{D}_{train}(G)$.
4: Estimate its I_f on $\mathcal{D}_{test}(G)$.
5: Assign Credits to each binary SVM by (5).
6: Assign Credits to each feature by (7).
7: Eliminate worst feature according to these credits, *i.e*, G=G-{X} where X receives least credits in step 6.
8: **end while**
9: Return the multi-class SVM trained on the current feature subset.

The MOI-Pseudowrapper (MOI-P) algorithm (Algorithm 1) relies on the ability of SVMs to generalize well in high dimensional spaces. The multi-class SVM is trained on the full set of features. The amount of information it delivers about the class labels using the full feature set, is estimated over a test set. Feature credits are obtained by information backpropagation. These credits are used as relevance estimates similar to a filter. Thus, no search routine is employed and the complexity of this algorithm is roughly the complexity of training a single SVM on the full feature set. In our experiments, we have used this scheme as a filter *for* SVMs themselves.

The MOI Backward elimination (MOI-BE) algorithm (Algorithm 2) implements the pseudowrapper approach recursively. At each step, the worst feature is eliminated and an SVM with one less feature is trained. Since, the algorithm uses better estimates at each step, we expect it to out-perform the filter approach. On the other hand, since features are greedily eliminated with no back-tracking, an incorrect elimination can be immensely harmful down the recursion. Neither of these two wrappers are feasible for MLPs since they rely on the performance on the full feature set. This algorithm involves training $(N - K + 1)$ multi-class SVMs with decreasing dimensionality. It may be desirable to eliminate more than one feature at each step of the recursion when dealing with very large data sets.

Algorithm 3 The MOI Wrapper for SVMs and MLPs

Require: $0 < K \leq N$; $\mathcal{D}_{train}$; $\mathcal{D}_{test}$

1: INITIALIZE: G {A randomly selected subset of size K. For SVMs we use the MOI-P for initialization.}
2: INITIALIZE: RESET=0
3: Train the classifier (MLP or SVM) on $\mathcal{D}_{train}(G)$.
4: Estimate its I_f on $\mathcal{D}_{test}(G)$.
5: **If** performance is satisfactory go to EXIT.
6: Obtain credits for each feature by information backpropagation.
7: **If** G gives the best performance so far, set $\hat{G} = G$ and RESET=0.
8: **If** there are untested features Replace the least informative current feature (according to the current credits) by the next untested feature.
9: **If** all features have been tried once - Determine (a) the best feature not currently being used and (b) the worst feature currently being used.
   1) **If** Credit(a) $>$ Credit(b) : replace feature (b) by (a) in G and go to Step 2.
   2) **If** Credit(a) $<$ Credit(b) and $\hat{G} = G$ go to EXIT.
   3) **If** Credit(a) $<$ Credit(b) and $\hat{G} \neq G$ - set $G = \hat{G}$ and RESET = RESET+1.
10: **If** RESET=2 go to EXIT **else** Go to Step 3.
11: EXIT: Return the multi-class SVM trained on the current subset.

The MOI algorithm (Algorithm 3) trains classifiers with K-features. Thus, it is suitable for feature selection in MLPs also when K is small enough. This algorithm has a back-tracking component and specializes in searching the space of subsets of size K , guided by information backpropagation. It is therefore expected to out-perform both the previous approaches. The key steps of this directed search are Steps 3, 4 and 9(1). The rest are initialization and exception handling activities. For SVMs, the initial K-feature subset is constructed by the pseudo-wrapper, whereas for MLPs we initiate randomly. Untested features are given preference initially so that all features can be credited. The directed search makes sense only after this has been achieved. The search may terminate prematurely only if a satisfactory classifier gets trained even before all features have been tried. At each iteration only K features have their credits updated. This means that the feature credits are updated asynchronously (with respect to each other). The credit associated with a feature depends on the choice of other $K-1$ features used when it was last selected. The choice of a new subset G, based on these current feature credits, is thus only a good guess. The performance of the classifier trained on G is not guaranteed to be better than the previous. Therefore the $\hat{G}$ is required to give MOI the ability to backtrack if stuck

in a G inferior to some previous subset G. The RESET counter ensures that the algorithm terminates with that best set if it is stuck in a limit cycle. The MOI algorithm needs to train atmost $4(N-K+1)$ K-feature classifiers in order to converge. In the worst case there will be $N-K+1$ iterations to test every feature once, another $N-K+1$ iterations after the first reset in which a new $\hat{G}$ is found and $2(N-K+1)$ iterations to exit after being reset twice to this new $\hat{G}$. In most cases, it requires far fewer iterations – usually less than $2(N-K+1)$. Each iteration requires the classifier to be trained on the current feature set. As stated before, the selection of the training objective function and training algorithm is made extraneous to MOI by the approximation (4) and outside the scope of this paper.

The computation of $I(Y;Y_f)$, being based on only $|\mathcal{Y}|^2$ numbers, is inexpensive. The Information backpropagation step needs to run once per iteration, on the trained classifier only. The computation of all the covariances for MLPs can be achieved in a single pass through the test data set; whereas computation of the derivative based sensitivity for SVMs requires a pass over the small set of support vectors. Thus the computation cost per iteration is dominated by the cost of training a K-feature classifier. In [10], the cost of training a K-feature MLP was (empirically) found to vary as K^3. This cubic dependence on input dimensionality makes MOI-MLP computationally attractive *vis-a-vis* wrappers that require training of networks with N features before pruning down to $K < N$ [11]. The cost of training an SVM is linear in dimensionality [19]. Thus, the complexity of MOI-SVM and MOI-BE are comparable.

IV. Experiments

The MOI algorithms were tested on several standard data sets. Detailed results on 2 artificial datasets are presented to illustrate the behavior, strengths, and weaknesses. Specific aspects are analyzed and comparisons are made against several other methods on real world data sets.

A. Artificial Data Sets

1) Corral: The Corral problem is an artificially constructed problem created to test decision trees [12] and other feature selectors. [4] discusses the relative merits of several feature selection algorithms on this insightful problem. There are six features (all binary) and one binary class variable. The class is defined in terms of the first 4 features as a boolean function. The fifth feature is irrelevant. The sixth feature is correlated to the target, matching it in 75% of the data points.

For $K = 6$ (all features), information back-propagation is performed once. The point of interest is to see the credit assigned to the six features. The average information credited to the six inputs by MOI-MLP over 4 random initializations are $0.217, 0.204, 0.222, 0.177, 0.083, 0.086$. The credits assigned by MOI-SVM (Gaussian Kernel, width parameter $\gamma = 1$) are $0.184, 0.175, 0.176, 0.190, 0.146, 0.118$ The four relevant features are clearly credited more than the last two. Thus given all the features, MOI-MLP and MOI-SVM are able to rank the highly correlated sixth feature and the irrelevant fifth feature as lower than the first four (relevant) features.

For $K = 5$ (reject one), the point of interest is to see which input is rejected. The relevant features are always selected for both MOI-MLP and MOI-SVM. Of the six possible initializations for MOI-MLP, feature 5 is rejected five times (leading to 95.3% performance on the validation set) and feature 6 once (100%). For MOI-SVM, the initialization by MOI-P causes it to reject feature 6 with 100% performance on the validation set.

For $K = 4$, we find MOI-MLP and MOI-SVM to always converge to the set of relevant features (irrespective of the initialization for MOI-MLP). The credits assigned to features 5 and 6 by information backpropagation are more distinctively lower than those assigned to the relevant features.

For $K = 2$, the optimal selection should be $\{1,2\}$, $\{3,4\}$ or $\{X,6\}$. It can be shown that in each case a 75% performance should be achievable. However, due to bias in the output, it is actually possible to achieve higher than 75% accuracy based on $\{1,2\}$ or $\{3,4\}$. The results for MOI-MLP with all possible random initializations of 2-feature sets is as follows. The sets $\{1,2\}$ or $\{3,4\}$ are picked 9 times, $\{X,6\}$ is picked in 4 cases and $\{1,4\}$ is picked in two cases. The accuracies achieved are 81.3%, 78.1% and 68.8% on the validation set. On the other hand, MOI-SVM as initialized by MOI-P, again converges to a best set $\{3,4\}$.

For $K = 1$, both MOI-MLP and MOI-SVM converge on the sixth input achieving a 75% accuracy. For the $K = 1$ case, this is in fact the (unique) optimal choice. In isolation, none of the four relevant features can predict the output better than chance. The information credits assigned to the six features, for both MOI-MLP and MOI-SVM, are $0.105, 0.105, 0.105, 0.105, 0.000, 0.180$. All four relevant features are equivalent, the irrelevant feature is useless and the correlated feature is the best.

The Corral problem illustrates the ability of MOI to pick good subsets for various $K \leq N$. The process of elimination begins with the irrelevant and redundant features, but it can continue beyond that if required. The second key point illustrated is that the context dependency of the credits associated with the features is *desirable*. The difference in credits for $K = 6, 4, 1$ reflects the utility of each feature in the context of obtaining an optimal classifier with $6, 4, 1$ features. The correlated feature (feature 6) is useless but benign for $K = 6$, its inclusion is harmful for $K = 4$ and it is the best input for $K = 1$. Another interesting observation is that MOI-SVM is always well initialized by MOI-P on this problem.

We now compare the behavior of MOI-P, MOI-BE and MOI-SVM. We find that for $K = 6, 4$ all these algorithms return 100% performance on training and validation sets, identifying the best feature subsets. For $K = 2$, MOI-P returns the sub-optimal set $\{1,4\}$ and the performance on the validation set is 68.75%, whereas MOI-BE and SVM-MOI are equal and optimal. For $K = 1$, MOI-BE, having discarded feature 6 early in the recursion, returns suboptimal performance relative to MOI-SVM. Thus, our intuition is confirmed: In general, the ability to back-track makes MOI-SVM superior to MOI-BE, which in turn improves upon MOI-

P.

2) Parity: The Parity problem studied here is a 15 feature version, with 5 irrelevant features, 5 relevant features and 5 redundant features. Each redundant feature is a duplicate of a relevant feature. This problem has been examined to show situations where MOI algorithms fail. We report only for MOI-MLP, but the conclusions generalize to other algorithms. When tested with $K = 10, 7, 5$, it is found that there is a very large variance in the final performance. For certain initial sets, MOI-MLP happens to pick up a set with 5 independent relevant features and the classifier performs close to 100%. For other initial sets, it fails to find such a set and the performance of the final classifier is close to 50%. The failure of MOI can be understood as follows. For the parity problem, the fall in classifier performance for incorrect features is total. At a performance close to random guesswork, $I(Y;Y_f) = 0$. Thus as soon as such a set is picked during the iterative process, all the current features receive 0.0 credits. In the absence of graded credits, the directed search cannot function. The algorithm iterates blindly and sometimes chances upon a good set. (Thus the 'success' of MOI for the parity problem is due to the density of good subsets among the total set of size K subsets rather than a successful directed search.) This situation arises because the lack of a single feature can totally degrade classifier performance for Parity. For real problems, the absence of a single feature seldom reduces the performance of a trained classifier to chance level.

B. Real World Data Sets

We have performed experiments on real world data sets drawn from the UCI machine learning repository and subsets of the Reuters-21578 text collection [17]. 12 data sets - 6 each for MLPs and SVMs were selected and their particulars (Number of Classes, Features, Training/Test/Validation splits, and the classifier parameters used) are listed in Table I and Table II. The choice was made so as to facilitate comparisons with results reported elsewhere. Separate sets were used for training the classifier and for computing $I(Y;Y_f)$. The performance of the final classifier was tested on an unseen validation data set. Separate validation sets were not used if three splittings caused under-representation of any class. In some cases, results were averaged over multiple random splits so as to match experimental protocols used elsewhere. Due to the difference between software packages used for training and unreported parameters, the performance of the classifiers on the full set of features for a data set often did not match across published results and our experiments. In such cases, in order to focus on feature selection, we report relative performance in terms of the ratio of accuracies obtained with the selected features and the full feature set.

C. Feature Selection in MLPs

1) Breast-Cancer and Vote: Table III and IV compare the performance of MOI-MLP with the NNFS [20] and the ANNIGMA [11] wrappers. These wrappers are reported to be the best performing methods on these data sets, among a variety of feature selection methods for MLPs explored in [11]. NNFS uses a training objective function consisting of two terms - the output cross-entropy and a penalty on the number of weights in the network. The basis of feature selection in MLPs using NNFS is therefore, network pruning. ANNIGMA implements a directed search strategy based on crediting features according to the weights associated with them. NNFS and ANNIGMA are designed to also find an optimal K; we make a comparison with the best two values of K for MOI-MLP. The relative improvement in MLP performance with feature selection is best using MOI-MLP, on the breast cancer data set.

TABLE I

DATA SETS USED FOR MLPS. (*xN denotes N random splits; K-X-Y denotes an architecture with K features, X hidden layer neurons and Y output neurons*)

Data Set	$\|\mathcal{Y}\|$	N	Train,Test,Validation	Architecture
Breast-Cancer	2	9	174,176,349 (x30)	K-12-2
DNA	3	180	2000,1186,0	K-5-3
Landsat	6	36	4435,1000,1000	K-60-6
Sonar	2	60	104,104,0	K-3-2
Vehicle	4	18	423,423,0	K-30-4
Vote	2	16	197,21,217 (x30)	K-3-2

TABLE II

DATA SETS USED FOR SVMS. (*xN denotes N random splits*)

Data Set	$\|\mathcal{Y}\|$	N	Train,Test,Validation	SVM Kernel
SAT	6	36	4435,1000,1000	RBF($\gamma = 0.001$)
Vehicle	4	18	282,282,282	RBF($\gamma = 0.001$)
Yeast	5	79	121,87,0 (x8)	RBF($\gamma = 0.01$)
Reuters-1	3	2225	199,113,0	Linear
Reuters-2	3	2344	193,162,0	Linear
Reuters-3	5	8167	3257,2912,0	Linear

2) DNA: The DNA data set was used to compare the performance of MOI-MLP with popular filters like Information Gain (IG) [15] and χ^2-statistic (Chi) [25]. Another well known filter *Relieff* assigns a relevance weight to each feature based on its capability to distinguish between nearest examples of the same class and opposite classes [13]. Table V reports the performance of an MLP trained with the filtered features using these methods versus MOI-MLP. We find that MOI-MLP outperforms the filters for K=80,30,11 and 3.

TABLE III

COMPARATIVE PERFORMANCE OF MOI-MLP ON BREAST-CANCER. (*K:N denotes ratio of MLP accuracy with the selected K and all N features.*)

Methods	ANNIGMA	NNFS	MOI	MOI
K	2.8	2.7	3	4
K:N	0.9958	1.0022	1.0082	1.0118

TABLE IV

COMPARATIVE PERFORMANCE OF MOI-MLP ON VOTE. (*K:N denotes ratio of accuracies with K and N features.*)

Methods	ANNIGMA	NNFS	MOI	MOI
K	2	2	2	1
K:N	1.0	1.03	0.999	1.01

TABLE V
COMPARATIVE PERFORMANCE OF MOI-MLP ON DNA

K	MOI	IG	Chi	ReliefF
180	94.2	94.2	94.2	94.2
80	95.3	94.9	91.9	94.6
30	95.5	95.1	95.1	94.4
11	94.0	89.5	89.5	91.7
6	88.5	87.0	89.0	89.0
3	80.7	75.4	75.4	75.4

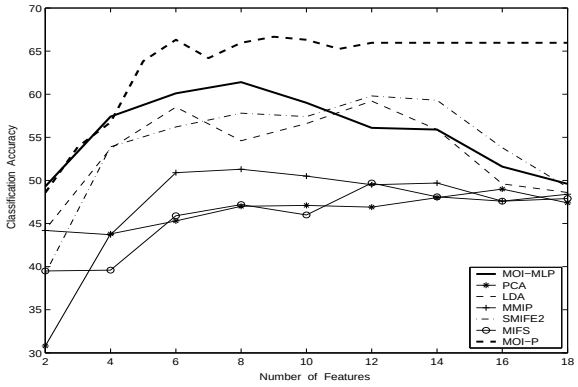


Fig. 1. Comparative performance of MOI-MLP on Vehicle

3) Vehicle: Fig 1 plots the performance of MOI-MLP for the entire range of feature selections and compares it with the performance of an MLP trained on features provided by MIFS (described in Secion I) and a variety of feature transformation methods - PCA, LDA [5], MMIP and SMIFE (described in Secion I) . MOI-MLP out-performs all other methods especially for smaller values of K. For comparison, we have also plotted the performance of MOI-P for an SVM. It can be seen that the SVM performance is relatively unaffected by the presence of unnecessary features (K=8 to K=16). A feature selection performed by MOI-P indeed improves its performance (e.g, for K=6).

4) Landsat: Table VI compares the performance of MOI-MLP with a number of other methods. Most methods are able to improve upon the performance of the MLP with respect to its performance on the full feature set for moderate values of K. For K=5, we find only SMIFE and MOI-MLP to maintain improvements in performance.

5) Sonar: The results of MOI-MLP for this data set are presented in Table VII. For comparison, results reported in

TABLE VI
COMPARATIVE PERFORMANCE OF MOI-MLP ON LANDSAT. (*First row reports performance on N features. Other rows report the ratio of accuracies with K and with N features.*)

K	MOI	SMIFE2	MMIP	MIFS	IG	PCA	LDA
36	71.5%	77.6%	79.0%	79.3%	85.1%	78.9%	78.7%
30	1.05	1.03	1.01	1.00	1.02	1.00	1.00
18	1.04	1.04	1.01	1.00	1.02	1.00	1.00
10	1.02	1.04	1.00	0.98	1.01	1.00	0.98
5	1.03	1.03	1.01	0.87	0.97	1.00	0.89

TABLE VII
COMPARATIVE PERFORMANCE OF MOI-MLP ON SONAR

K	Random	PCA	MIFS	MOI	Iter.
60	82.9 (2.9)	82.9 (2.9)	82.9 (2.9)	82.9 (2.9)	1.0
18	78.5 (5.1)	72.7 (3.7)	79.2 (1.3)	84.5 (1.1)	60.5
12	76.9 (4.1)	63.7 (3.2)	78.9 (3.0)	83.7 (0.7)	59.7
6	68.1 (4.1)	63.2 (4.0)	75.1 (4.1)	80.6 (1.5)	63.0

TABLE VIII
COMPARATIVE PERFORMANCE ON YEAST. (*First row reports performance on N features. Other rows report the ratio of accuracies with K and with N features.*)

K	AROM	RFE	MOI-P	MOI-BE	MOI-SVM
79	95.7(1.1)	95.7(1.1)	94.52(1.31)	94.52(1.31)	94.52(1.31)
10	1.0199	0.9801	0.9746	0.9822	1.0068
20	0.9906	0.9801	0.9890	0.9999	1.0215
40	1.0094	1.0094	0.9930	1.0053	1.0188

Battiti [1] are also listed for MIFS, PCA and random selection. The last column confirms that the average number of iterations required for convergence scales linearly with $N - K$.

D. Feature Selection in SVMs

1) Yeast: Table VIII lists the performance of MOI-P, MOI-BE and MOI-SVM and compares them with results using the *Approximation of zero norm minimization*(AROM) [24] (AROM) and the recursive feature elimination (RFE) methods reported in [24]. AROM sets up linear programs to approximately minimize the number of non-zero components of the SVM weight vector under suitable constraints. RFE is very similar to MOI-BE and performs margin based backward elimination. MOI-SVM is the only algorithm that demonstrates relative improvement for all selections attempted and outperforms MOI-P and MOI-BE as expected. The comparison is not rigorous since we train the multi-class SVM on lesser data using gaussian kernels, and report the results averaged over 8 random data splits. In [24], 8-fold cross validation results are reported on experiments using a linear kernel.

2) Vehicle: Fig 2 compares the performance of the SVM wrappers with classical feature transformation methods like PCA and LDA and several popular filters. The Shared Variance (SV) and Gain Ratio (GR) are normalized versions of the χ^2-statistic and Information Gain (IG) respectively [25]. We find the multiclass SVM with LDA features to excel for K=3, but LDA cannot produce more than 3 features for this 4-class problem. For moderate feature selection (K=9 to K=18) MOI-P, MOI-BE and MOI-SVM return identical performance. In this domain, MOI-P is the best strategy since it is inexpensive and performs just as well. For smaller values of K, MOI-SVM is able to select better features and the performance of MOI-BE is bracketed between MOI-P and MOI-SVM. Observe that for K=2 and K=3, MOI-BE is actually worse than MOI-P. This reiterates the harmful effect of relying on the performance of SVMs successively trained with features that are greedily eliminated.

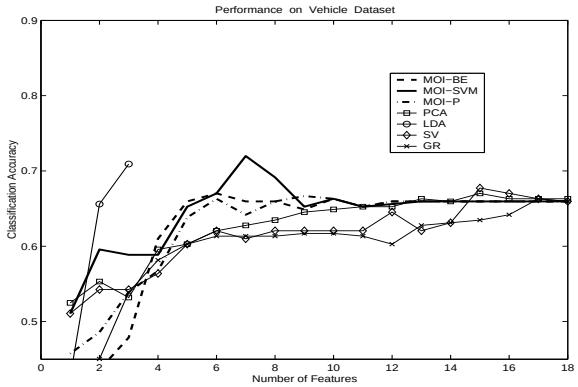


Fig. 2. Comparative performance of SVM on VEHICLE

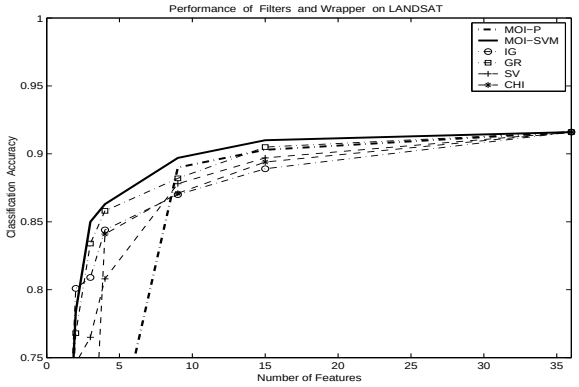


Fig. 3. Comparative performance of SVM on LANDSAT

3) Landsat: The Landsat data set is considered difficult for greedy feature selection algorithms [23]. The objective here is to compare MOI-P with statistical and information theoretic filters. We find, as shown in Fig 3, MOI-P to compare favorably for low to moderate feature selection (K=36 to K=7), but is outperformed for smaller values of K. In this domain, MOI-SVM is able to improve upon all the filters.

4) Text Classification: We constructed three subsets of the Reuters collection involving a very large number of features. The first subset Reuters-1 comprises of articles on the topics *coffee, iron-steel* and *livestock*. These topics are not likely to have many meaningful overlapping words. The second subset Reuters-2 contains articles on *reserves, gold* and *gross national product*, likely to have similar words used in different contexts across these topics. Reuters-3 was constructed to examine the performance of MOI-P on even larger dimensionality. It contains articles on the five most frequent Reuters categories : *earn, acq, money-fx, grain* and *crude*. Each article was binary-encoded where each feature denoted whether a particular word occurred in the article or not. As a preprocessing step, all articles with more than 30% content numeric were excluded from the dataset and words occurring less than 3 times in each dataset were eliminated to remove extremely rare words. We compare MOI-P against Information gain for drastic feature selection selecting top 20% features [22].

As Table IX shows, with both IG and MOI-P, the classifier maintains acceptable levels of performance on drastic feature reduction. Recall that IG involves direct computation of mutual information (MI) between the *continuous* inputs and desired outputs. MOI-P involves (i) an approximate optimization using SVM training and (ii) MI computation only at the output and indirect labeling of inputs. The results in Table IX validates the appropriateness of both these approximations. As for resource consumption, the major component of MOI-P runtime is the training time of the binary SVMs with all the features. For the largest dataset, Reuters-3, the training time had an average of 11 min per SVM. The feature crediting module is very quick since it processes only the support vectors, which in Reuters-3, average 475 per class. It is reasonable to hope to improve upon these results using MOI-BE, eliminating multiple features per step or by using MOI-SVM for more exhaustive selection.

TABLE IX

PERFORMANCE ON THE REUTERS DATASET (*Classification Accuracy (CA) and Relative Output Information (R.O.I)* $I(Y;Y_f)/H(Y)$)

Dataset Train, Test	N K	IG CA, ROI	MOI-P CA, ROI
Reuters-1	2225	96.46%, 87.42%	96.46%, 87.42%
199,113	431	99.12%, 96.18%	98.23%, 93.45%
Reuters-2	2344	94.44%, 77.95%	94.44%, 77.95%
193,162	458	95.67%, 83.72%	93.83%, 76.52%
Reuters-3	8167	93.00%, 78.80%	93.00%, 78.80%
3257,2912	1167	92.00%, 76.77%	93.10%, 79.00%

V. CONCLUSIONS

We have proposed output information $I(Y;Y_f)$ as a new information theoretic objective function for evaluating classifiers; and demonstrated its utility for the task of feature selection in MLPs and SVMs. This objective function is computationally inexpensive and scalable, immune to bias in input distribution and theoretically well founded. The MOI algorithms attempt to optimize it by greedy feature elimination and directed search in the feature subset space. These algorithms incorporate useful properties of the classifiers and compare favorably with a number of statistical and information-theoretic methods on several artificial and real world problems.

REFERENCES

[1] Battiti, R., "Using Mutual Information for Selecting Features in Supervised Neural Net Training," *IEEE Trans Neural Networks*, vol 5, no 4, pp 537-550. July 1994.

[2] Bollacker, K.D., Ghosh, J., "Linear Feature Extractors Based on Mutual Information," *Proc ICNN 1996*, vol. 3, pp 1528-1533. Washington D.C., June 1996.

[3] Bridle, J.S., "Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters," *Advances in Neural Information Processing Systems*, vol. 2, pp. 211-217. D.S. Touretzky, ed. San Mateo, C.A.: Morgan Kauffman, 1990.

[4] Dash,M. and Liu,H. "Feature selection for classification. Intelligent Data Analysis", vol. 1, pp 131–156, 1997.

[5] Devroye L., Gyorfi L., and Lugosi G., *A Probabilistic Theory of Pattern Recognition.* Springer-Verlag New York, Inc., 1996.

[6] Erdogmus, D., & Principe, J., (2003). Lower and Upper Bounds For Misclassification Probability Based on Renyi's Information. *Journal of VLSI Signal Processing Systems (Special Issue on Wireless Communications and Blind Signal Processing)*

[7] Fano, R.M., (1961). *Transmission of Information: A Statistical Theory of Communications*. New York: MIT Press & John Wiley Sons Inc.

[8] Guyon,I., Weston,J., Barnhill,S., and Vapnik,V., "Gene selection for cancer classification using support vector machines," *Machine Learning*, 2000.

[9] Guyon,I., and Elisseeff, A., "An Introduction to Variable and Feature Selection," *Journal of Machine Learning Research (Special Issue on Variable and Feature Selection)*, 2003
[10] Hinton, G.E., "Connectionist Learning Procedures," *Artificial Intelligence*, vol. 40, no. 1, pp 143-150. 1989.
[11] Hsu,C.N., Huang,H.J., and Schuschel,D.,"The ANNIGMA-wrapper approach to Fast Feature Selection for Neural Nets," *IEEE Transactions on Systems, Man, and Cybernetics Part B*, 32(2):207-212, April 2002.
[12] John,G.H., Kohavi,R., & Pfleger,K., "Irrelevant features and the subset selection problem". *In Proceedings of the 11th International Conference on Machine Learning*, pp 121-129, San Mateo, CA, Morgan Kaufmann, 1994.
[13] Kira, K. and Rendell,L., "The feature selection problem: Traditional methods and a new algorithm," *In Proceedings of the Tenth National Conference on Artificial Intelligence* (AAAI-92), pages 129–134, Menlo Park, CA, USA, 1992. AAAI Press.
[14] Linsker, R., "Self-organization in a perceptual network," *Computer Magazine*, vol. 21, pp.105-117. 1988.
[15] Quinlan, R.,*C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1993.
[16] Renyi, A. (1970). *Probability Theory*, NY: American Elsevier Publishing Company Inc.
[17] Reuters-21578 Text Collection: http://www.daviddlewis.com/resources/testcollections/reuters21578
[18] Rumelhart, D. E., & McClelland, J. L. (Eds.). *Parallel distributed processing: Explorations in the microstructure of cognition (Vol. 1)*. Cambridge, MA: MIT Press, 1986
[19] Schőlkopf, B. & Smola A.J, (2002) *Learning with Kernels*. Cambridge, MA: MIT Press
[20] Setiono,R., and Liu,H., "Neural network feature selector", *IEEE Transactions on Neural Networks*, vol.8, no. 3, pp.645-662, 1997.
[21] Shannon,C.E. & Weaver,W., *The mathematical theory of communications*, Urbana, IL: The University of Illinois Press, 1949.
[22] Sindhwani, V., Bhattacharyya, P., & Rakshit, S. (2001), Information Theoretic Feature Crediting in Multiclass Support Vector Machines. *Proceedings of First SIAM International Conference on Data Mining*.
[23] Torkkola, K. and Campbell W.M., (2000) Mutual information in learning feature transformations, *Proceedings of the Seventeenth International Conference on Machine Learning*.(pp 1015-1022). San Francisco:Morgan Kaufmann.
[24] Weston,J., Elisseeff, A., Tipping,M., and Schőlkopf,B., "Use of the zero norm with linear models and kernel methods" *JMLR special Issue on Variable and Feature selection*, 2002.
[25] White, A. and W.Z. Liu 1994. Bias in Information-Based Measures in Decision Tree Induction. Machine Learning 15, 321-329. 3
[26] Vapnik, V.(1998). *Statistical Learning Theory*. New York: Wiley
[27] Yang,H. and Moody,J., " Data Visualization and Feature Selection: New Algorithms for nongaussian data," *In Advances in Neural Information Processing Systems*, 12, pp.687-693. MIT Press, 2000.
[28] Yang, Y. and Pedersen, J. O., "A comparative study on feature selection in text categorization," *Proc. of ICML'97*, pages 412-420, 1997. 19

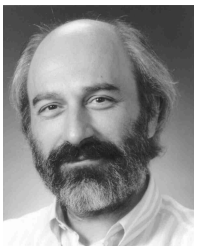
Vikas Sindhwani received the Bachelor's degreee in Engineering Physics from the Indian Institute of Technology, Bombay, India in 2001.

He is a doctoral student in the Department of Computer Science at the University of Chicago. He has been a visitor at the Center for Artificial Intelligence and Robotics, Bangalore, India, and the Department of Empirical Inference and Machine Learning, Max Planck Insitute of Biological Cybernetics, Tuebingen, Germany. His current research interests are in the areas of Machine Learning, Information Theory and Signal Processing.

Subrata Rakshit (M '96) was born in Jamshedpur, India. He received his B.Tech in Engg Physics from IIT Bombay in '88 and MS, PhD in EE from Caltech, USA, in '89 and '94 respectively. After a brief post-doctoral fellowship at the Washington Univ School of Medicine, St Louis, MO, he has been working at the Centre for AI and Robotics (CAIR) in Bangalore, India, since Dec 1994. He currently heads the Computer Vision Group. His research interests include ANNs, Information Theory, Image Processing and Multi-Sensor Data Fusion.

Dipti Deodhare received her MSc(CS) from Pune University, MS(Engg) and PhD in Computer Science and Automation from the Indian Institute of Science, Bangalore. She has been working at the Centre for Artificial Intelligence and Robotics (CAIR), Bangalore, since 1991. She currently heads the AI and NN Group at CAIR. Her research interests include Neural Networks, Pattern Recognition and Decision Support Systems.

Deniz Erdogmus received the B.S. degree in electrical and electronics engineering and mathematics in 1997 and the M.S. degree in electrical and electronics engineering, with emphasis on systems and control, in 1999, both from the Middle East Technical University, Ankara, Turkey. He received the Ph.D. degree in electrical and computer engineering from the University of Florida, Gainesville, in 2002. He was a Research Engineer at the Defense Industries Research and Development Institute (SAGE), Ankara, from 1997 to 1999. Since 1999, he has been with the Computational NeuroEngineering Laboratory, University of Florida, working under the supervision of Dr. J. C. Principe. His current research interests include information theory and its applications to adaptive systems and adaptive systems for signal processing, communications, and control. Dr. Erdogmus is a member of IEEE, Tau Beta Pi, and Eta Kappa Nu.



Jose Principe Jose C. Principe is a Distinguished Professor of Electrical and Computer Engineering and Biomedical Engineering at the University of Florida where he teaches advanced signal processing, machine learning and artificial neural networks (ANNs) modeling. He is BellSouth Professor and the Founder and Director of the University of Florida Computational NeuroEngineering Laboratory (CNEL). His primary area of interest is processing of time varying signals with adaptive neural models. The CNEL Lab has been studying signal and pattern recognition principles based on information theoretic criteria. Dr. Principe is an IEEE Fellow. He is a member of the ADCOM of the IEEE Signal Processing Society, Member of the Board of Governors of the

Partha Niyogi received the Bachelor's degree in Electrical Engineering at the Indian Institute of Technology, New Delhi, India, and MS and Ph.D. degrees in machine learning theory at the Department of Electrical Engineering and Computer Science at the Massachussetts Institute of Technology.

He is an associate professor in the Computer Science and Statistics departments at the University of Chicago. His research interests are generally in the field of artificial intelligence and specifically in pattern recognition and machine learning problems that arise in the computational study of human speech and language.