Bayes Optimal Feature Selection for Supervised Learning with General Performance Measures

Saneem Ahmed C.G.[†]
IBM Research
Bangalore 560045, India

Harikrishna Narasimhan
Indian Institute of Science
Bangalore 560012, India

Shivani Agarwal
Indian Institute of Science
Bangalore 560012, India

Abstract

The problem of feature selection is critical in several areas of machine learning and data analysis. Here we consider feature selection for supervised learning problems, where one wishes to select a small set of features that facilitate learning a good prediction model in the reduced feature space. Our interest is primarily in filter methods that select features independently of the learning algorithm to be used and are generally faster to implement than wrapper methods. Many common filter methods for feature selection make use of mutual information based criteria to guide their search process. However, even in simple binary classification problems, mutual information based methods do not always select the best set of features in terms of the Bayes error. In this paper, we develop a filter method that directly aims to select the optimal set of features for a general performance measure of interest. Our approach uses the Bayes error with respect to the given performance measure as the criterion for feature selection and applies a greedy algorithm to optimize this criterion. We demonstrate application of this method to a variety of learning problems involving different performance measures. Experiments suggest the proposed approach is competitive with several state-of-the-art methods.

1 INTRODUCTION

The problem of feature selection is critical in several areas of machine learning and data analysis, particularly for learning prediction models with good generalization ability and for reducing the running time of learning algorithms [1–3]. In this paper, we consider feature selection for supervised learning problems, where one wishes to select a small set of features that facilitate learning a good prediction model in the reduced feature space. Our focus is primarily on filter methods that select features independently of the learning algorithm to be used, typically by greedily maximizing some suitable feature selection criterion. These methods are generally faster in practice and easier to implement than other approaches to feature selection such as wrapper or embedded methods.

Over the years, there has been much work on designing filter methods for feature selection, many of which make use of mutual information based criteria to guide their search process [4–7]. However, these methods do not explicitly consider the performance measure used to evaluate a model in the learning problem. In fact, even in the case of simple binary classification, one can construct settings where the popular mutual information criterion does not yield the best set of features in terms of the Bayes 0-1 classification error (as we shall shortly see with an example) [8]. Clearly, there is a need for filter methods that are tailored to directly optimize a given performance measure of interest.

In this paper, we develop a *Bayes optimal* filter method for a general performance measure. Our approach directly aims to find the optimal set of features in terms of the Bayes error for the given loss or performance measure, thus allowing for the possibility of learning a good model in the reduced feature space. We show that the mutual information criterion mentioned above is a special case of our setting when the loss function of interest is the logarithmic loss for class probability estimation. We use a greedy forward selection algorithm for approximately optimizing the Bayes criterion for the given performance measure, and demonstrate application of this method to various learning problems involving different performance measures. Experiments on several learning tasks suggest that the proposed approach is competitive with the state-of-the-art methods.

Indeed in the simpler setting of classification with the 0-1 error, there have been some works that have suggested the use of Bayes error as a criterion for feature selection/transformation [8–13]. Of these, only Yang and Hu (2012) provide an experimental evaluation of a filter method for optimizing the Bayes 0-1 error [13]; however,

[†]Work done while at Indian Institute of Science, Bangalore.

even here, the objective eventually optimized is different from the Bayes optimal criterion for the 0-1 error (we elaborate on this in Section 3.1). On the other hand, we provide in this paper the first systematic study of Bayes optimal filter methods for general performance measures, going well beyond the simple setting of 0-1 classification, and handling a variety of learning settings, including those with complex performance measures such as the F-measure.

1.1 RELATED WORK

Filter methods have received much attention from the machine learning/data mining/artificial intelligence communities, resulting in various hand-crafted filter criteria and heuristic techniques for optimizing the proposed objectives [4–7, 14–21]. Predominant among these are methods that use the mutual information (MI) between a given feature subset and the output label as a measure of *relevance* of the feature subset to the given learning task, often with additional information theoretic terms to account for *redundancy* among the features in the given subset [4–7, 16–21]. While there have been arguments made to justify the use of MI as a criterion for binary classification by establishing lower/upper bounds on MI in terms of the 0-1 Bayes error [21–24], these bounds are tight only for certain settings; in general, the optimal feature subset for the MI criterion need not be the same as that for the 0-1 Bayes error.

There has also been some work on designing filter methods for specific learning tasks, such as text retrieval [25], class imbalanced classification [26], and ranking [27]. However, the feature selection criteria proposed therein are either based on heuristics and do not explicitly promote feature subsets that are Bayes optimal for the given problem, or as in the case of [25], require a certain (binary) representation of the features and do not apply to general settings.

Apart from filter methods, other popular families of feature selection techniques include *wrapper methods*, where the quality of a subset of features is determined by explicitly learning a model on the feature subset and evaluating its accuracy on a held-out sample [2, 28, 29]; and *embedded methods*, which combine model learning and feature selection into a single step, such as using sparse regularization in the learning problem [30]. While both these approaches allow us to incorporate different loss functions during feature selection, filter methods are computationally cheaper as they decouple feature selection from model learning, and are typically simpler to implement in practice.

Organization. Section 2 gives preliminaries, together with an example illustrating that the MI feature selection criterion can be suboptimal for binary classification. Section 3 describes the proposed Bayes feature selection method, followed by examples of how it can be applied to different learning problems and performance measures. Section 4 gives results of experiments on several learning tasks.

2 PRELIMINARIES AND BACKGROUND

Notation. For $n \in \mathbb{Z}_+$, we denote $[n] = \{1, \ldots, n\}$. For a vector $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$ and set $\mathcal{J} = \{j_1, \ldots, j_k\} \subseteq [n]$ with $j_1 < \ldots < j_k$, we denote $\mathbf{x}_\mathcal{J} = (x_{j_1}, \ldots, x_{j_k}) \in \mathbb{R}^k$. For random variables X and Y, we denote by $H(X)$ the entropy of X, by $H(Y|X)$ the conditional entropy of Y given X, and by $I(X;Y)$ the mutual information between X and Y. For a predicate ϕ, we denote by $\mathbf{1}(\phi)$ the indicator of ϕ, which takes the value 1 if ϕ is true and 0 otherwise. For any $z \in \mathbb{R}$, $\text{sign}(z) = 1$ if $z > 0$ and -1 otherwise.

Problem Setup. Let $\mathcal{X} \subseteq \mathbb{R}^n$ be an n-dimensional instance space. We will be interested in feature selection for *supervised learning problems*, where there is some label space $\mathcal{Y}$ and prediction space $\widehat{\mathcal{Y}}$; one receives a training sample $S = ((\mathbf{x}^1, y^1), \cdots, (\mathbf{x}^m, y^m)) \in (\mathcal{X} \times \mathcal{Y})^m$, and the goal is to learn a prediction model $h : \mathcal{X} \to \widehat{\mathcal{Y}}$.[1] Typically, one assumes all examples (both training examples and future test examples) are drawn i.i.d. from some probability distribution D on $\mathcal{X} \times \mathcal{Y}$, and the goal is to learn a prediction model with good prediction performance (according to a suitable performance measure) on future examples from D. We will denote by (X, Y) a random variable drawn from D. Often, performance is measured via a loss function $\ell : \mathcal{Y} \times \widehat{\mathcal{Y}} \to \mathbb{R}_+$; the goal then is to learn a model h minimizing the expected loss on a new example from D, which we refer to as the *ℓ-error* of h w.r.t. D: $\text{er}_D^\ell[h] = \mathbf{E}_{(X,Y)\sim D}[\ell(Y, h(X))]$. The smallest achievable ℓ-error over all possible prediction models is called the *Bayes ℓ-error* for D: $\text{er}_D^{\ell,*} = \inf_{h:\mathcal{X}\to\widehat{\mathcal{Y}}} \text{er}_D^\ell[h]$. For example, in binary classification, one has $\mathcal{Y} = \widehat{\mathcal{Y}} = \{\pm 1\}$, and the loss function of interest is often the 0-1 loss $\ell_{0\text{-}1} : \{\pm 1\} \times \{\pm 1\} \to \mathbb{R}_+$ defined as $\ell_{0\text{-}1}(y, \widehat{y}) = \mathbf{1}(\widehat{y} \neq y)$. For problems with binary labels $\mathcal{Y} = \{\pm 1\}$, we will denote by $p = \mathbf{P}(Y = 1)$ the overall probability of label $+1$ under D, and by $\eta : \mathcal{X} \to [0, 1]$ the associated class probability function: $\eta(\mathbf{x}) = \mathbf{P}(Y = 1 \,|\, X = \mathbf{x})$. Here the Bayes 0-1 error has the form $\text{er}_D^{0\text{-}1,*} = \mathbf{E}_X[\min(\eta(X), 1 - \eta(X))]$.

The *feature selection problem* we are interested in is to select a subset of features $\mathcal{J} \subseteq [n]$ of some specified size $k \in [n]$ (usually $k \ll n$), such that one can then learn a good prediction model in the reduced k-dimensional feature space $\mathcal{X}_\mathcal{J} = \{\mathbf{x}_\mathcal{J} \,|\, \mathbf{x} \in \mathcal{X}\} \subseteq \mathbb{R}^k$.[2] Specifically, given a training sample $S \in (\mathcal{X} \times \mathcal{Y})^m$ as above, one works with the reduced training sample $S_\mathcal{J} = ((\mathbf{x}^1_\mathcal{J}, y^1), \cdots, (\mathbf{x}^m_\mathcal{J}, y^m)) \in (\mathcal{X}_\mathcal{J} \times \mathcal{Y})^m$, and learns a prediction model $h_\mathcal{J} : \mathcal{X}_\mathcal{J} \to \widehat{\mathcal{Y}}$ in the reduced space $\mathcal{X}_\mathcal{J}$. We will denote by $D_\mathcal{J}$ the marginal distribution of D on $\mathcal{X}_\mathcal{J} \times \mathcal{Y}$; for problems with binary la-

[1] Often $\widehat{\mathcal{Y}} = \mathcal{Y}$, but this is not always the case.

[2] In this paper, we assume for simplicity that the target feature subset size k is given as part of the problem. However, the methods developed easily extend to settings where k is unknown.

bels $\mathcal{Y} = \{\pm 1\}$, we will also denote by $\eta_{\mathcal{J}} : \mathcal{X}_{\mathcal{J}} \to [0,1]$ the class probability function on the reduced feature space $\mathcal{X}_{\mathcal{J}}$: $\eta_{\mathcal{J}}(\mathbf{z}) = \mathbf{P}(Y = 1 \,|\, X_{\mathcal{J}} = \mathbf{z})$, where $X_{\mathcal{J}}$ contains components of the random vector X corresponding to indices in $\mathcal{J}$. Clearly, if the examples in S are drawn i.i.d. from D, then the examples in $S_{\mathcal{J}}$ can be viewed as being drawn i.i.d. from $D_{\mathcal{J}}$. In the loss function setting, the performance of a model $h_{\mathcal{J}}$ learned in the reduced feature space is measured via its ℓ-error w.r.t. $D_{\mathcal{J}}$: $\mathrm{er}^{\ell}_{D_{\mathcal{J}}}[h_{\mathcal{J}}] = \mathbf{E}_{(Z,Y)\sim D_{\mathcal{J}}}[\ell(Y, h_{\mathcal{J}}(Z))] = \mathbf{E}_{(X,Y)\sim D}[\ell(Y, h_{\mathcal{J}}(X_{\mathcal{J}}))]$.

Feature Selection as (Approximate) Optimization. We will view feature selection methods as (approximately) optimizing some objective or criterion $C_D : 2^{[n]} \to \mathbb{R}$, which typically depends on distribution D. Given such a criterion C_D and a target feature subset size k, one aims to select

$$\mathcal{J}^* \in \operatorname*{argmax}_{\substack{\mathcal{J} \subseteq [n] \\ |\mathcal{J}| = k}} C_D(\mathcal{J})\,. \tag{1}$$

Of course, in practice, one does not know the distribution D, and so instead uses an approximate version of the criterion C_D based on the training sample S, which we shall denote as $\widehat{C}_S : 2^{[n]} \to \mathbb{R}$. Moreover, the combinatorial optimization problem (over $\binom{n}{k}$ subsets) is generally computationally hard, and so one settles for an approximate search strategy, such as a greedy approach. We shall elaborate further on both these approximations below.

Filter Methods and Mutual Information (MI) Criterion. In a filter method for feature selection, the choice of the feature subset does not depend on the particular learning algorithm to be used in the reduced feature space, i.e. the criterion C_D is independent of the particular learning algorithm to be used. A popular filter criterion that is widely used in feature selection for supervised learning is the *mutual information* (MI) criterion, defined as the mutual information between the selected features and the labels:

$$C^{\mathrm{MI}}_D(\mathcal{J}) = I(X_{\mathcal{J}}; Y)\,, \tag{2}$$

where (X, Y) denotes a random variable distributed according to D.The motivation for using the MI criterion is that it is expected to preserve the information necessary for learning a good prediction model. Indeed, in the case of binary classification, monotonic functions of the mutual information $I(X;Y)$ are known to both upper and lower bound the Bayes error $\mathrm{er}^{0\text{-}1,*}_D$ [21–24]. However, as seen below, even in the case of binary classification, there are situations where the MI criterion does *not* select an optimal set of features:

Example 1 (Suboptimality of MI criterion for binary classification with 0-1 error)**.** *Consider a binary classification problem on a 2-dimensional instance space with binary features: $\mathcal{X} = \{0,1\}^2$, $\mathcal{Y} = \{\pm 1\}$. Let D be a probability distribution on $(\mathcal{X} \times \mathcal{Y})$ under which $\mathbf{P}(Y = 1) = 0.3$, the random variables X_1, X_2 (components of the random vector X) are conditionally independent given the label Y, and the class-conditional distributions are given by*

$$\mathbf{P}(X_1 = 1 \,|\, Y = 1) = 0.4\,; \quad \mathbf{P}(X_1 = 1 \,|\, Y = -1) = 0.1\,;$$
$$\mathbf{P}(X_2 = 1 \,|\, Y = 1) = 0.9\,; \quad \mathbf{P}(X_2 = 1 \,|\, Y = -1) = 0.4.$$

Clearly, $\mathbf{P}(X_1 = 1) = 0.19$, $\mathbf{P}(X_2 = 1) = 0.55$, $\eta_{\{1\}}(0) = 0.22$, $\eta_{\{1\}}(1) = 0.63$, $\eta_{\{2\}}(0) = 0.07$ and $\eta_{\{2\}}(1) = 0.49$. Now consider selecting a single feature for use in learning a binary classifier (thus here $n = 2$, $k = 1$). It can be verified that under the above distribution,

$$C^{\mathrm{MI}}_D(\{1\}) = I(X_1; Y) = 0.08$$
$$C^{\mathrm{MI}}_D(\{2\}) = I(X_2; Y) = 0.17.$$

Therefore the MI criterion would select feature 2 and learn a classifier in the feature space $\mathcal{X}_{\{2\}}$. One can also compute the Bayes 0-1 errors in $\mathcal{X}_{\{1\}}$ and $\mathcal{X}_{\{2\}}$; these can be verified to be

$$\mathrm{er}^{0\text{-}1,*}_{D_{\{1\}}} = 0.25\,; \quad \mathrm{er}^{0\text{-}1,*}_{D_{\{2\}}} = 0.30\,.$$

Thus even if one uses the best possible learning algorithm in the feature space $\mathcal{X}_{\{2\}}$ selected by the MI criterion, the best classifier one can learn will have 0-1 error 0.30. On the other hand, if we had selected feature 1, we could potentially have learned a classifier with 0-1 error 0.25!

The above example suggests looking directly for a feature subset that yields low Bayes error with respect to a given performance measure of interest.

3 BAYES OPTIMAL FEATURE SELECTION

Motivated by the above discussion, we now develop a filter method for feature selection that is tailored to optimize a general performance measure of interest. In particular, rather than selecting a feature subset by maximizing the mutual information with the labels, our approach optimizes *the information most relevant to the supervised learning task at hand*, with the aim of learning as good a prediction model in the reduced feature space as possible in terms of the given loss or performance measure. More formally, for a supervised learning problem with label space $\mathcal{Y}$, prediction space $\widehat{\mathcal{Y}}$, and with loss function $\ell : \mathcal{Y} \times \widehat{\mathcal{Y}} \to \mathbb{R}_+$, we will be interested in selecting a feature subset that minimizes the Bayes ℓ-error in the reduced feature space, or equivalently, maximizes the following criterion:

$$C^{\mathrm{Bayes},\ell}_D(\mathcal{J}) = -\mathrm{er}^{\ell,*}_{D_{\mathcal{J}}}\,. \tag{3}$$

Note that this is different from a wrapper method, which looks for a feature subset that maximizes prediction performance of a model learned by a particular algorithm; here, we are instead interested in finding the best feature subset for a given performance measure of interest, *without being tied to any particular learning algorithm*.

3.1 EXAMPLES OF BAYES CRITERION FOR VARIOUS LEARNING PROBLEMS AND PERFORMANCE MEASURES

Here we give several examples of the above Bayes criterion for specific learning problems/performance measures. We shall see that for the case of binary class probability estimation with the logarithmic loss, the Bayes criterion effectively reduces to the MI criterion (Example 5); thus the MI criterion can be viewed as finding a good feature space for class probability estimation. Similarly, for regression with squared error, the Bayes criterion is exactly the criterion optimized in the forward regression feature selection algorithm for sparse linear regression (Example 6). We begin with the simple case of binary classification with 0-1 error.

Example 2 (Bayes criterion for binary classification with 0-1 error)**.** *Let $\mathcal{Y} = \widehat{\mathcal{Y}} = \{\pm 1\}$, with $\ell_{0\text{-}1} : \{\pm 1\} \times \{\pm 1\} \to \mathbb{R}_+$ defined as $\ell_{0\text{-}1}(y, \widehat{y}) = \mathbf{1}(\widehat{y} \neq y)$. Then*

$$C_D^{\text{Bayes},0\text{-}1}(\mathcal{J}) = -\mathbf{E}_X\big[\min\big(\eta_{\mathcal{J}}(X_{\mathcal{J}}), 1 - \eta_{\mathcal{J}}(X_{\mathcal{J}})\big)\big]\,.$$

As noted earlier, the filter method provided by Yang and Hu (2012) [13] for optimizing the Bayes 0-1 error eventually optimizes an objective different from the above one; while the authors initially discuss a feature selection criterion of the above form, they end up prescribing and analyzing a variant $-\mathbf{E}_{X,Y}\big[(1-Y)\eta_{\mathcal{J}}(X_{\mathcal{J}}) + Y\big(1 - \eta_{\mathcal{J}}(X_{\mathcal{J}})\big)\big] = -\mathbf{E}_X\big[2\eta_{\mathcal{J}}(X_{\mathcal{J}})\big(1 - \eta_{\mathcal{J}}(X_{\mathcal{J}})\big)\big]$, which is not necessarily optimal for the 0-1 error (see Eq. (7) in their paper). In this work, we go well beyond the simple setting of 0-1 classification, and present a systematic study of Bayes optimal criteria for general performance measures, as seen below.

Example 3 (Bayes criterion for binary classification with cost-sensitive error)**.** *Let $\mathcal{Y} = \widehat{\mathcal{Y}} = \{\pm 1\}$. Let $c \in (0,1)$ denote the cost of a false positive and $(1-c)$ the cost of a false negative; the corresponding cost-sensitive loss $\ell_c : \{\pm 1\} \times \{\pm 1\} \to \mathbb{R}_+$ is defined as*

$$\ell_c(y, \widehat{y}) = \begin{cases} c & \text{if } y = -1, \widehat{y} = 1 \\ 1 - c & \text{if } y = 1, \widehat{y} = -1 \\ 0 & \text{otherwise.} \end{cases}$$

Then

$$\begin{aligned} & C_D^{\text{Bayes},c}(\mathcal{J}) \\ &= -\mathbf{E}_X\big[\min\big((1-c)\eta_{\mathcal{J}}(X_{\mathcal{J}}), c(1 - \eta_{\mathcal{J}}(X_{\mathcal{J}}))\big)\big]. \end{aligned}$$

Example 4 (Bayes criterion for binary classification with balanced 0-1 error)**.** *Let $\mathcal{Y} = \widehat{\mathcal{Y}} = \{\pm 1\}$. The balanced loss $\ell_{\text{bal}} : \{\pm 1\} \times \{\pm 1\} \to \mathbb{R}_+$ seeks to balance prediction errors on positive and negative examples by weighting them according to their inverse class probabilities, and is frequently used to measure classification performance in class imbalance settings [31]; it depends on the underlying distribution D via the probability $p = \mathbf{P}(Y = 1)$, and is defined as*

$$\ell_{\text{bal}}(y, \widehat{y}) = \begin{cases} \frac{1}{1-p} & \text{if } y = -1, \widehat{y} = 1 \\ \frac{1}{p} & \text{if } y = 1, \widehat{y} = -1 \\ 0 & \text{otherwise.} \end{cases}$$

Here the Bayes criterion becomes

$$C_D^{\text{Bayes,bal}}(\mathcal{J}) = -\mathbf{E}_X\left[\min\left(\frac{\eta_{\mathcal{J}}(X_{\mathcal{J}})}{p}, \frac{1 - \eta_{\mathcal{J}}(X_{\mathcal{J}})}{1-p}\right)\right].$$

Example 5 (Bayes criterion for binary class probability estimation with logarithmic loss)**.** *Let $\mathcal{Y} = \{\pm 1\}$ and $\widehat{\mathcal{Y}} = [0,1]$, with logarithmic loss $\ell_{\log} : \{\pm 1\} \times [0,1] \to \mathbb{R}_+$ defined as*

$$\ell_{\log}(y, \widehat{y}) = -\mathbf{1}(y = 1)\ln(\widehat{y}) - \mathbf{1}(y = -1)\ln(1 - \widehat{y})\,.$$

Then

$$\begin{aligned} & C_D^{\text{Bayes,log}}(\mathcal{J}) \\ &= -\mathbf{E}_X\big[-\eta_{\mathcal{J}}(X_{\mathcal{J}})\ln(\eta_{\mathcal{J}}(X_{\mathcal{J}})) \\ &\qquad -(1 - \eta_{\mathcal{J}}(X_{\mathcal{J}}))\ln(1 - \eta_{\mathcal{J}}(X_{\mathcal{J}}))\big] \\ &= -H(Y|X_{\mathcal{J}}) \;=\; I(X_{\mathcal{J}}; Y) - H(Y) \\ &= C_D^{\text{MI}}(\mathcal{J}) - H(Y)\,. \end{aligned}$$

This is equivalent to using the MI criterion! Thus, in the binary setting, the MI criterion effectively selects a feature set that minimizes Bayes log-error, i.e. that allows for a good class probability estimator (in terms of logarithmic loss) in the resulting feature space! (Note that this is not *the same as selecting good features for binary classification with 0-1 error or other performance measures; e.g. see Example 1. This is also demonstrated in our experiments in Section 4.)*

Example 6 (Bayes criterion for regression with squared error)**.** *Let $\mathcal{Y} = \widehat{\mathcal{Y}} = \mathbb{R}$, with squared loss $\ell_{\text{sq}} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$ defined as $\ell_{\text{sq}}(y, \widehat{y}) = (\widehat{y} - y)^2$. Then*

$$C_D^{\text{Bayes,sq}}(\mathcal{J}) = -\mathbf{E}_X\big[\mathbf{Var}(Y \mid X_{\mathcal{J}})\big]\,.$$

This is exactly the criterion used in the well-known forward regression *feature selection algorithm for sparse linear regression (where one assumes $\mathbf{E}[Y|X = \mathbf{x}] = \boldsymbol{\beta}^\top \mathbf{x}$ for some sparse $\boldsymbol{\beta} \in \mathbb{R}^n$) [32].*

While all examples seen so far have involved performance measures that can be expressed as an expected value of a loss function, we shall next consider examples of learning problems where the performance measure of interest is complex and non-additive.

Example 7 (Bayes criterion for binary classification/retrieval with F_β-measure)**.** *Let $\mathcal{Y} = \widehat{\mathcal{Y}} = \{\pm 1\}$, and consider a classification or retrieval problem where the goal is to learn a classifier $h : \mathcal{X} \to \{\pm 1\}$ with performance measured by the F_β-measure (higher values are better):*

$$\text{F}_{\beta,D}[h] = \frac{1 + \beta^2}{\frac{\beta^2}{\text{Prec}_D[h]} + \frac{1}{\text{Rec}_D[h]}},$$

Algorithm 1 ℓ-BayesGreedy

Inputs: $S = (\mathbf{x}^1, y^1), \ldots, (\mathbf{x}^m, y^m)) \in (\mathbb{R}^n \times \mathcal{Y})^m$
$k \in [n]$
Initialize: $\mathcal{J} \leftarrow \emptyset$
for $t = 1 \ldots k$ **do**
  $j_t \leftarrow \operatorname{argmax}_{j \in [n] \setminus \mathcal{J}} \widehat{C}_S^{\text{Bayes},\ell}(\mathcal{J} \cup \{j\})$
  $\mathcal{J} \leftarrow \mathcal{J} \cup \{j_t\}$
end for
Output: $\mathcal{J}$

where $\text{Prec}_D[h] = \mathbf{P}\big(Y = 1 \,|\, h(X) = 1\big)$ *and* $\text{Rec}_D[h] = \mathbf{P}\big(h(X) = 1 \,|\, Y = 1\big)$ *are the precision and recall of* h*, respectively, and* $\beta > 0$ *trades off the relative importance of these two quantities. In this case, the performance measure cannot be expressed as the expected value of a loss function over individual data points. Nevertheless, it is known that the Bayes optimal classifier for the* F_β*-measure is obtained by thresholding the class probability function* η *for the given distribution at an optimal point [24, 33]. One can therefore compute the Bayes optimal value of the* F_β*-measure for a given distribution, and use this as the criterion to be optimized in feature selection:*

$$\begin{aligned} C_D^{\text{Bayes},\text{F}_\beta}(\mathcal{J}) &= \sup_{h_\mathcal{J}:\mathcal{X}_\mathcal{J}\to\{\pm 1\}} \text{F}_{\beta,D_\mathcal{J}}[h_\mathcal{J}] \\ &= \sup_{t\in[0,1]} \text{F}_{\beta,D_\mathcal{J}}[\text{sign} \circ (\eta_\mathcal{J} - t)]. \end{aligned}$$

Example 8 (Bayes criterion for bipartite ranking with AUC)**.** *Let* $\mathcal{Y} = \{\pm 1\}$*, and consider a bipartite ranking problem where the goal is to learn a scoring function* $f : \mathcal{X} \to \mathbb{R}$*, with performance measured by the area under the ROC curve (AUC) (higher values are better):*

$$\begin{aligned} \text{AUC}_D[f] = \mathbf{E}\big[\mathbf{1}\big((Y - Y')(f(X) - f(X')) > 0\big) \\ + \tfrac{1}{2}\,\mathbf{1}\big(f(X) = f(X')\big) \,\big|\, Y \neq Y'\big], \end{aligned}$$

where $(X, Y), (X', Y')$ *are drawn i.i.d. from* D*. While here again, the performance measure cannot be expressed as an expectation of loss function, one can indeed compute the Bayes optimal value of the performance measure for a given distribution (e.g. see [34]); we use this as the criterion to be optimized in feature selection:*

$$\begin{aligned} &C_D^{\text{Bayes,AUC}}(\mathcal{J}) \\ &= \sup_{f_\mathcal{J}:\mathcal{X}_\mathcal{J}\to\mathbb{R}} \text{AUC}_{D_\mathcal{J}}[f_\mathcal{J}] \\ &= 1 - \frac{\mathbf{E}\big[\min\big(\alpha_\mathcal{J}(X_\mathcal{J}, X'_\mathcal{J}),\, \alpha_\mathcal{J}(X'_\mathcal{J}, X_\mathcal{J})\big)\big]}{2p(1-p)}, \end{aligned}$$

where $\alpha_\mathcal{J}(Z, Z') = \eta_\mathcal{J}(Z)(1 - \eta_\mathcal{J}(Z'))$.

3.2 GREEDY ALGORITHM FOR OPTIMIZING BAYES CRITERION

As noted earlier, in practice, one does not have access to the true distribution D, and therefore must optimize an approximate version of the Bayes criterion based on the training sample $S = ((\mathbf{x}^1, y^1), \ldots, (\mathbf{x}^m, y^m))$. In particular, for a label space $\mathcal{Y}$, prediction space $\widehat{\mathcal{Y}}$, and loss $\ell : \mathcal{Y} \times \widehat{\mathcal{Y}} \to \mathbb{R}_+$, note that the Bayes ℓ-error w.r.t. D can be written as

$$\text{er}_D^{\ell,*} = \mathbf{E}_X\Big[\inf_{\widehat{y}\in\widehat{\mathcal{Y}}} \mathbf{E}_{Y|X}\big[\ell(Y, \widehat{y})\big]\Big].$$

To obtain a sample-based estimate of $\text{er}_D^{\ell,*}$, one can replace the outer expectation over X by an average over the training instances $\mathbf{x}^i$ in S, and use an empirical estimate of the conditional distribution of Y given X in computing the inner expectation:

$$\widehat{\text{er}}_S^{\ell,*} \;=\; \frac{1}{m}\sum_{i=1}^m \inf_{\widehat{y}\in\widehat{\mathcal{Y}}} \widehat{\mathbf{E}}_{Y|X=\mathbf{x}^i}\big[\ell(Y, \widehat{y})\big],$$

where $\widehat{\mathbf{E}}_{Y|X}$ denotes expectation with respect to an approximate conditional distribution $\widehat{\mathbf{P}}(Y|X)$ estimated from the sample S. This gives the approximate Bayes criterion

$$\begin{aligned} \widehat{C}_S^{\text{Bayes},\ell}(\mathcal{J}) &= -\widehat{\text{er}}_{S_\mathcal{J}}^{\ell,*} \\ &= -\frac{1}{m}\sum_{i=1}^m \inf_{\widehat{y}\in\widehat{\mathcal{Y}}} \widehat{\mathbf{E}}_{Y|X_\mathcal{J}=\mathbf{x}^i_\mathcal{J}}\big[\ell(Y, \widehat{y})\big], \end{aligned}$$

where again $\widehat{\mathbf{E}}_{Y|X_\mathcal{J}}$ denotes expectation with respect to an approximate conditional distribution $\widehat{\mathbf{P}}(Y|X_\mathcal{J})$ estimated from the sample $S_\mathcal{J}$. For example, for binary classification with 0-1 error, one gets the approximate criterion:

$$\widehat{C}_S^{\text{Bayes,0-1}}(\mathcal{J}) \;=\; -\frac{1}{m}\sum_{i=1}^m \min\big(\widehat{\eta}_\mathcal{J}(\mathbf{x}^i_\mathcal{J}), 1 - \widehat{\eta}_\mathcal{J}(\mathbf{x}^i_\mathcal{J})\big),$$

where $\widehat{\eta}_\mathcal{J} : \mathcal{X}_J \to [0, 1]$ is a suitable estimate of $\eta_\mathcal{J}$ based on $S_\mathcal{J}$. An ideal algorithm would then select the best subset of k features according to the above approximate criterion:

$$\widehat{\mathcal{J}}_S \;\in\; \operatorname*{argmax}_{\substack{\mathcal{J}\subseteq[n] \\ |\mathcal{J}|=k}} \widehat{C}_S^{\text{Bayes},\ell}(\mathcal{J}).$$

However, this optimization problem (over $\binom{n}{k}$ subsets) is typically still hard due to its combinatorial nature. As is often done in other feature selection approaches, one possibility is to use an algorithm that selects features to maximize the above criterion in a greedy fashion. For example, one can use a forward selection algorithm which starts with an empty feature set, and at each iteration, adds the feature with the highest marginal value of the objective $\widehat{C}_S^{\text{Bayes},\ell}$ to the current set of features (see Algorithm 1).[3]

Conditional probability estimation for large k using s-variate approximations. Applying the above algorithm

[3]We note that the proposed greedy method easily extends to settings where the value of k is not available to us; for example, one can terminate this method based on an appropriate stopping criterion (such as when the difference in feature criterion across two successive iterations falls below a certain value) and use the features chosen up to that point to learn a suitable predictor.

Table 1: Data sets used in our experiments.

Data set	No. of features	No. of instances	Feature type	$p = \mathbf{P}(Y = 1)$
Mushroom	116	8124	Binary	0.482
Adult	123	48824	Binary	0.239
Splice	240	3190	Binary	0.519
Semeion	256	1593	Binary	0.102
KDDCup01	139351	1909	Binary	0.022
Pcmac	3289	1943	Integer	0.495
Basehock	4862	1993	Integer	0.501
Gisette	5000	6000	Integer	0.500
Waveform	40	5000	Real	0.331

as shown requires computing conditional probability estimates $\widehat{\mathbf{P}}(Y|X_{\mathcal{J}})$ for feature sets $\mathcal{J}$ of size up to k. For small k, this is easy to do; for example, for problems with binary labels, one computes:

$$\widehat{\eta}_{\mathcal{J}}(\mathbf{z}) = \widehat{\mathbf{P}}(Y = 1 \,|\, X_{\mathcal{J}} = \mathbf{z}) = \begin{cases} \frac{\sum_{i=1}^{m} \mathbf{1}(\mathbf{x}^i_{\mathcal{J}} = \mathbf{z}, y^i = 1)}{\sum_{i=1}^{m} \mathbf{1}(\mathbf{x}^i_{\mathcal{J}} = \mathbf{z})} & \text{if } \sum_{i=1}^{m} \mathbf{1}(\mathbf{x}^i_{\mathcal{J}} = \mathbf{z}) > 0 \\ \frac{1}{2} & \text{otherwise.} \end{cases}$$

When k is large, one runs into difficulties in later iterations of the algorithm. Specifically, consider the t-th iteration, when $(t-1) < k$ features $j_1, \ldots, j_{t-1}$ have been added to $\mathcal{J}$ and the t-th feature is to be selected. For large t, it is likely that most configurations of $\mathbf{x}^i_{\mathcal{J}}$ appear only once in the training sample, and therefore for all potential features $j_t \in [n] \setminus \mathcal{J}$, one gets (in a setting with binary labels) that $\widehat{\eta}_{\mathcal{J}\cup\{j_t\}}(\mathbf{x}^i_{\mathcal{J}\cup\{j_t\}})$ is either 0, 1 or $\frac{1}{2}$, thus giving many ties and no useful basis for selecting the next feature. This is an inherent difficulty that arises when estimating high-dimensional multivariate distributions from limited data. A common approach to overcome this problem, often used in the context of optimizing the MI criterion (e.g. see [5]), is to use approximate calculations that require estimating conditional distributions on only smaller subsets of the features; one such approach is a s-variate approximation (for some small $s < k$), where the given filter criterion on a set of k features $\mathcal{J}$ is approximated by the average value of the criterion on all subsets of $\mathcal{J}$ of size s [20]:

$$\widehat{C}_S^{\text{Bayes}}(\mathcal{J}) \approx \frac{1}{\binom{k}{s}} \sum_{A \subset \mathcal{J}, |A| = s} \widehat{C}_S^{\text{Bayes}}(A).$$

With such approximations, one can use algorithms based on both forward selection and backward elimination to greedily maximize the Bayes criterion. In our experiments with large feature subsets, we use the standard bivariate approximation with $s = 2$.

4 EXPERIMENTS

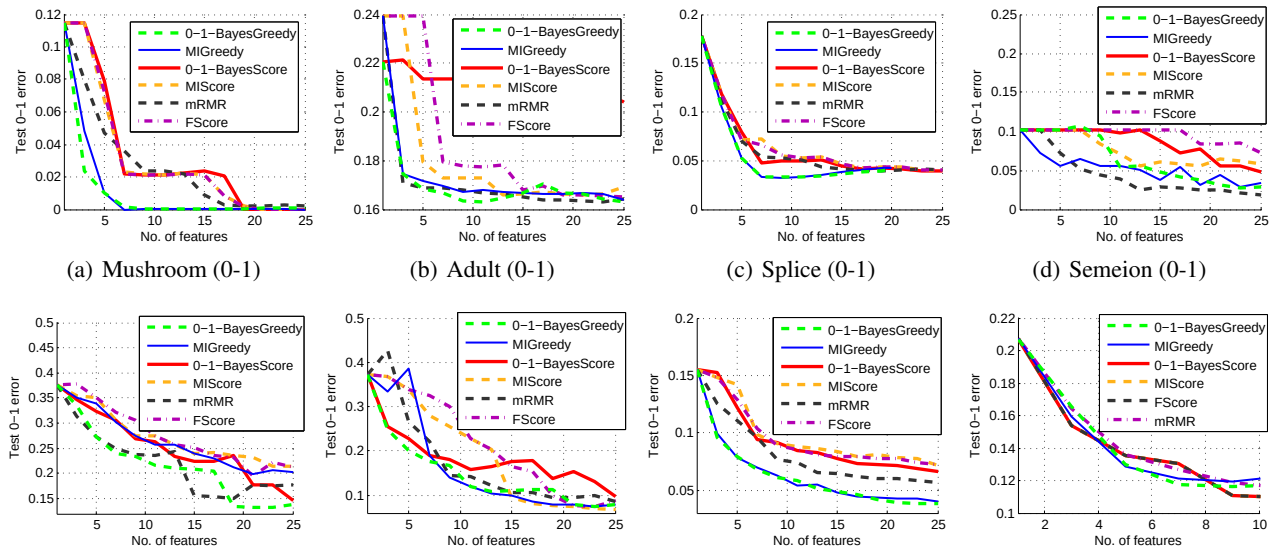
We now report results of experiments designed to evaluate the proposed Bayes optimal feature selection method in a variety of settings with different performance measures. These include binary classification with both the standard 0-1 and cost-sensitive losses, binary class probability estimation (CPE) with the logarithmic loss (under which our Bayes criterion reduces to MI criterion), and learning under class imbalance with the balanced 0-1 loss and F-measure. The data sets used in our experiments are shown in Table 1; these include varying numbers of features/examples, feature types, and class probabilities.[4] Each data set was split into train-test sets, with the feature selection methods and learning algorithms applied on the training set, and the learned model evaluated on the test set; the average performance over 5 random train-test splits is then reported. All tunable parameters in the learning algorithms used were chosen using a held-out portion of the training set.[5,6]

Baselines. Our main method, which optimizes the Bayes criterion corresponding to the loss or performance measure of interest in a greedy manner (possibly with some approximations in estimating high-dimensional conditional distributions), is termed BayesGreedy. We also include a score-based variant of our method (BayesScore) that scores each feature independently using the Bayes criterion evaluated on the corresponding one-dimensional feature space, and selects the top k features according to this score. As baselines, we consider a number of standard filter methods popular in practice. These include a method that optimizes the MI criterion in a greedy manner (again with some approximations in estimating high-dimensional conditional distributions), termed MIGreedy [5, 7]; a score-based vari-

[4] We obtained Pcmac and Basehock from the ASU repository (http://featureselection.asu.edu), KDDCup01 from the KDD Cup Challenge 2001 (http://pages.cs.wisc.edu/dpage/∼kddcup2001/) and the rest from the UCI ML repository (http://www.ics.uci.edu/∼mlearn/MLRepository.html). Of these, Semeion and Waveform are multi-class data sets, where one of the class was taken as positive, and the remaining were combined into the negative class.

[5] In the case of the larger Adult data set, 20% of the data was used for training and the remaining for testing. On all other data sets, 70% was used for training. In each case, a held-out 20% of the training set was used for parameter tuning.

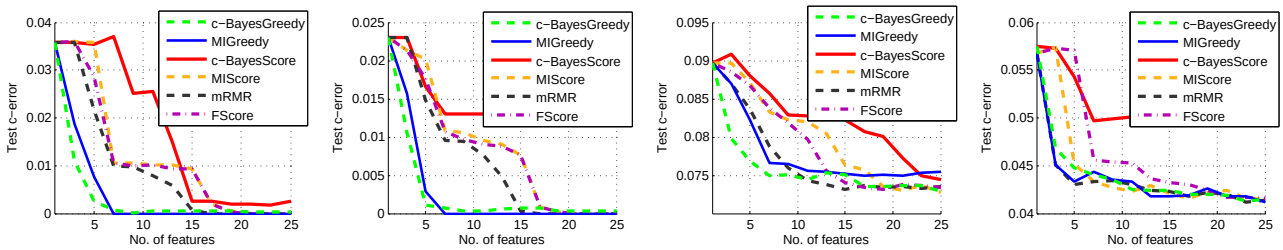
[6] For data sets with integer/real valued features, we discretized each feature into three categories based on intervals: $(-\infty, \mu - \sigma)$, $[\mu - \sigma, \mu + \sigma)$, and $[\mu + \sigma, \infty)$, where μ is the mean feature value and σ is the standard deviation.



(a) Mushroom (0-1) (b) Adult (0-1) (c) Splice (0-1) (d) Semeion (0-1)

(e) Pcmac (0-1) (f) Basehock (0-1) (g) Gisette (0-1) (h) Waveform (0-1)

Figure 1: Feature selection for binary 0-1 classification. Plots show test 0-1 error vs. number of features for different feature selection methods, with SVM (RBF kernel) as the classification algorithm.



(a) Mushroom ($c = 0.25$) (b) Mushroom ($c = 0.1$) (c) Adult ($c = 0.25$) (d) Adult ($c = 0.1$)

Figure 2: Feature selection for cost-sensitive binary classification with different costs c. Plots show test cost-sensitive error vs. number of features for various filter methods, with cost-sensitive SVM (RBF kernel) as the classification algorithm.

ant of this method for optimizing the MI criterion (MIScore) [21]; and a score-based method that optimizes another popular feature selection criterion, namely the Fischer score (F-score) [35]. Apart from the above methods, there are other filter methods based on MI that in addition to optimizing for relevant feature subsets, also seek to promote some form of 'diversity' among the chosen features. Popular among these is the minimal-redundancy-maximal-relevance (mRMR) method [6], which we include as a representative baseline from this category.

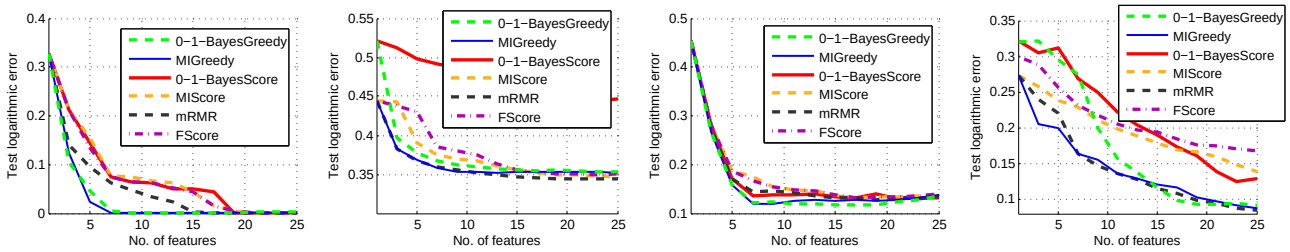
The above baselines are indeed representative of the various filter methods used in practice, with most other methods based on MI being variants of the MIGreedy or mRMR methods. Since the focus of this paper is entirely on filter methods, we do not compare our approach against wrapper or embedded methods, which unlike filter methods are closely tied to the learning algorithm used.

In experiments below, unless otherwise specified, the BayesGreedy and MIGreedy methods shall use exact estimates of class-conditional distributions.

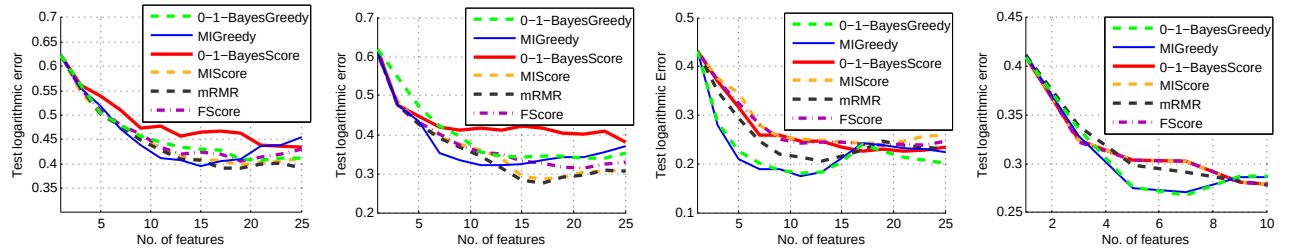
4.1 BINARY CLASSIFICATION (0-1 ERROR AND COST-SENSITIVE ERROR)

The first task that we consider is binary classification with the standard 0-1 error (see Example 2 for the Bayes criterion for this performance measure). We used kernel SVM (with RBF kernel) as the learning algorithm for this task. Figure 1 contains the test 0-1-error for the different feature selection methods as a function of the number of features chosen. As seen, on all data sets except Semeion and for most feature subset sizes, the features chosen by the proposed BayesGreedy method (that explicitly optimizes the 0-1 error) perform comparable to or better than the baseline methods. The poor performance of BayesGreedy on the Semeion data set was due to the inexact/greedy search technique used by the method (when the Bayes criterion was optimized exactly on this data set using an exhaustive search over feature subsets, we did obtain better performance than the MI criterion).

We also consider the task of binary classification with cost-sensitive error (see Example 3 for the Bayes criterion).

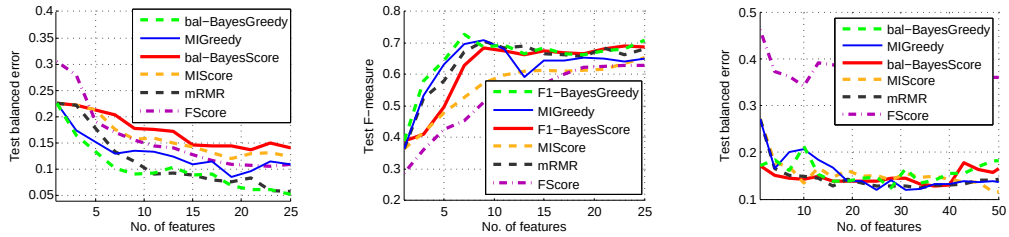


(a) Mushroom (CPE) (b) Adult (CPE) (c) Splice (CPE) (d) Semeion (CPE)



(e) Pcmac (CPE) (f) Basehock (CPE) (g) Gisette (CPE) (h) Waveform (CPE)

Figure 3: Feature selection for binary CPE. Plots show test logarithmic error vs. number of features for different feature selection methods, with logistic regression (RBF kernel) as the CPE algorithm. Here, MI is the Bayes optimal criterion for the logarithmic error; one can see that MIGreedy performs the best in most cases.



(a) Semeion (Balanced 0-1) (b) Semeion (F_1-measure) (c) KDDCup01 (Balanced 0-1)

Figure 4: Feature selection for learning under class imbalance. (a), (c) Test balanced 0-1 error vs. number of features, with balanced SVM (RBF kernel) used as the learning algorithm. (b) Test F_1-measure vs. number of features, with a plug-in method that uses logistic regression (RBF kernel) followed by empirical thresholding as the learning algorithm. Bivariate approximations were used in estimating class-conditionals for KDDCup01. Higher values are better for F_1-measure.

We used cost-sensitive kernel SVM as the learning algorithm here. Figure 2 contains results on the Adult and Mushroom data sets with different costs. In three of four cases, BayesGreedy yields lower cost-sensitive error than the baselines for smaller feature subset sizes and comparable values for larger feature subset sizes.

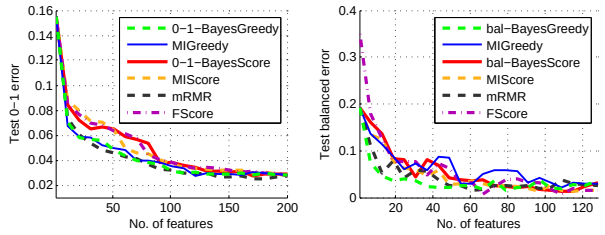
On most data sets, the score-based methods do not perform as well as the other methods; this is due to their naive search strategy where the features are scored independently.

4.2 BINARY CLASS PROBABILITY ESTIMATION (LOGARITHMIC ERROR)

The next task that we consider is class probability estimation with the logarithmic error. As mentioned earlier, the Bayes criterion here effectively reduces to the MI criterion (see Example 5). We used regularized kernel logistic regression (with RBF kernel) as the class probability estimation algorithm here. Figure 3 contains plots of the test logarithmic error vs. the number of features chosen for different feature selection methods; we also include for comparison methods that optimize the Bayes criterion for the 0-1 loss. MIGreedy, which optimizes the Bayes criterion for the logarithmic error, performs comparable to or better than the other methods for most feature subset sizes.

4.3 LEARNING UNDER CLASS IMBALANCE (BALANCED 0-1 ERROR AND F-MEASURE)

We now move to the task of binary classification under class imbalance. Commonly used performance measures in this setting include the balanced 0-1-error and F_1-measure, both of which aim to balance errors on either classes (see Examples 4 and 7 for the Bayes criterion for these measures). We used a balanced version of SVM (where the positive and negative points were weighted with costs $1/p$ and $1/(1-p)$ respectively) as the learning algorithm for



(a) Gisette (0-1) – large k (b) Semeion (Balanced 0-1) – large k

Figure 5: Selecting larger numbers of features. The settings here are similar to previous plots, except that bivariate approximations were used in estimating class-conditionals.

the balanced 0-1-error; and a plug-in method using logistic regression followed by thresholding of the resulting class probability estimate at a sample-based optimal point as the learning algorithm for the F_1-measure [33, 36, 37]. Figure 4 contains results on the class-imbalanced Semeion ($p = 0.102$) and KDDCup01 ($p = 0.022$) data sets. In the case of Semeion, the BayesGreedy methods perform the best over all. With KDDCup01, where we include results for the balanced 0-1 error (the performance measure used in the KDD Cup 2001 challenge), there is no clear winner; here, BayesScore performs the best for smaller feature subsets, and MIGreedy performs better for larger subsets.

4.4 SELECTING LARGER NUMBER OF FEATURES

We also evaluated the proposed filter methods on large feature subset sizes k. As noted earlier, an exact implementation of the prescribed greedy algorithm is difficult in this case as estimation of high-dimensional class-conditional distributions from limited data is prone to errors and is also computationally expensive. We therefore resorted to the bivariate approximation technique described in Section 3.2 for estimating the class-conditional distributions; the MIGreedy method also used the same estimation procedure, while the search technique in mRMR inherently used a similar approximation [6]. Figure 5 contains results on the Gisette (binary classification with 0-1 loss) and Semeion (binary classification with balanced 0-1 loss) data sets. In the case of Semeion, the BayesGreedy method consistently performs as well as (if not better than) the baselines; in the case of Gisette, BayesGreedy is the second best over all.

4.5 RUN-TIME COMPARISONS

We now present run-time comparisons of the various filter methods for different values of k. Table 2 contains the run-times (in seconds) for cost-sensitive classification on Adult data, and 0-1 binary classification with large k on Gisette data (with approximations used to estimate conditional distributions). All methods here were implemented in MATLAB. As expected, the score-based methods, requiring only a single sort operation, offer the least run-times; however, as seen earlier, these methods often perform poorly in terms of accuracy. Among the other methods, BayesGreedy is significantly faster than MIGreedy, despite both methods using the same search procedure (this is because the Bayes criteria for the 0-1 and cost-sensitive losses involve simple 'max' operations that can be implemented efficiently). On the Adult data, where BayesGreedy computes exact estimates of conditional distributions, it is slower than mRMR; however, when BayesGreedy uses computationally cheaper bivariate approximations to estimate probabilities, it yields lower run-times than mRMR even for larger values of k, as seen with the Gisette data.

Adult ($c = 0.25$)			
	$k = 5$	$k = 10$	$k = 15$
c-BayesGreedy	1.75	12.78	71.14
MIGreedy	1.95	15.63	94.06
c-BayesScore	0.10	0.10	0.10
MIScore	0.17	0.19	0.17
mRMR	1.32	5.27	11.71
FScore	0.06	0.08	0.08
Gisette (0-1) – large k			
	$k = 25$	$k = 50$	$k = 75$
0-1-BayesGreedy	954	3784	8475
MIGreedy	2123	8330	18475
0-1-BayesScore	2.62	2.65	2.71
MIScore	2.67	2.73	2.76
mRMR	1242	5046	11384
FScore	1.01	1.08	1.04

Table 2: Run-time comparison of various filter methods for different values of k. All values are in *seconds*. The settings here are same as before. For Gisette, bivariate approximations were used to estimate conditional distributions.

5 CONCLUSION

We have developed a Bayes optimal filter method for feature selection with supervised learning considering general performance measures, and provided instantiations of our method for a variety of learning problems and performance measures. Experiments demonstrate that our approach is competitive with many state-of-the-art methods. While our focus has been on problems with binary labels, our approach easily generalizes to multiclass settings.

A possible direction of work in the future is to investigate approximation guarantees for the greedy algorithm used to optimize a given Bayes optimal criteria. Indeed (under specific assumptions) such guarantees have been established for the MI criterion and the criterion for regression with squared loss, by leveraging tools from submodular optimization [38, 39]. It would be interesting to explore similar results for the other filter criteria developed in this work.

Acknowledgements. HN acknowledges support from a Google India PhD fellowship. SA thanks DST for support under a Ramanujan Fellowship.

References

[1] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[2] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422, 2002.

[3] Y. Yang and J.O. Pedersen. A comparative study on feature selection in text categorization. In *ICML*, 1997.

[4] R. Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5(4):537–550, 1994.

[5] D. Koller and M. Sahami. Toward optimal feature selection. In *ICML*, 1996.

[6] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1226–1238, 2005.

[7] G. Brown, A. Pocock, M-J. Zhao, and M. Luján. Conditional likelihood maximisation: A unifying framework for information theoretic feature selection. *Journal of Machine Learning Research*, 13:27–66, 2012.

[8] W. Duch. Filter methods. In Isabelle Guyon, Steve Gunn, Masoud Nikravesh, and Lofti Zadeh, editors, *Feature Extraction: Foundations and Applications*. Springer, 2006.

[9] G. Saon and M. Padmanabhan. Minimum Bayes error feature selection for continuous speech recognition. In *NIPS*, 2000.

[10] L.C. Molina, L. Belanche, and À. Nebot. Feature selection algorithms: A survey and experimental evaluation. In *ICDM*, 2002.

[11] G. Carneiro and N. Vasconcelos. Minimum Bayes error features for visual recognition by sequential feature selection and extraction. In *CRV*, 2005.

[12] S-H. Yang, H. Zha, S.K. Zhou, and B-G. Hu. Variational graph embedding for globally and locally consistent feature extraction. In *ECML PKDD*. 2009.

[13] S-H. Yang and B-G. Hu. Discriminative feature selection by nonparametric Bayes error minimization. *IEEE Transactions on Knowledge and Data Engineering*, 24(8):1422–1434, 2012.

[14] M. Robnik-Šikonja and I. Kononenko. Theoretical and empirical analysis of relieff and rrelieff. *Machine learning*, 53(1-2):23–69, 2003.

[15] Q. Gu, Z. Li, and J. Han. Generalized fisher score for feature selection. In *UAI*, 2011.

[16] D.A. Bell and H. Wang. A formalism for relevance and its application in feature subset selection. *Machine learning*, 41(2):175–195, 2000.

[17] N. Kwak and C-H. Choi. Input feature selection for classification problems. *IEEE Transactions on Neural Networks*, 13(1):143–159, 2002.

[18] F. Fleuret. Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research*, 5:1531–1555, 2004.

[19] L. Yu and H. Liu. Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research*, 5:1205–1224, 2004.

[20] P.E. Meyer, C. Schretter, and G. Bontempi. Information-theoretic feature selection in microarray data using variable complementarity. *Journal of Selected Topics in Signal Processing*, 2(3):261–274, 2008.

[21] N. Vasconcelos. Feature selection by maximum marginal diversity. In *NIPS*, 2002.

[22] R.M. Fano. *Transmission of information: A statistical theory of communications*. M.I.T. Press, 1961.

[23] L. Devroye, L. Gyorfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer, 1996.

[24] M-J. Zhao, N. Edakunni, A. Pocock, and G. Brown. Beyond Fano's inequality: Bounds on the optimal F-score, BER, and cost-sensitive risk and their implications. *Journal of Machine Learning Research*, 14(1):1033–1090, 2013.

[25] G. Forman. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3:1289–1305, 2003.

[26] Z. Zheng, X. Wu, and R. Srihari. Feature selection for text categorization on imbalanced data. *ACM SIGKDD Explorations Newsletter*, 6(1):80–89, 2004.

[27] X. Geng, T-Y. Liu, T. Qin, and H. Li. Feature selection for ranking. In *SIGIR*, 2007.

[28] R. Kohavi and G.H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1):273—324, 1997.

[29] I. Tsamardinos and C.F. Aliferis. Towards principled feature selection: Relevancy, filters and wrappers. In *AISTATS*, 2003.

[30] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[31] A.K. Menon, H. Narasimhan, S. Agarwal, and S. Chawla. On the Statistical Consistency of Algorithms for Binary Classification under Class Imbalance. In *ICML*, 2013.

[32] A. Miller. *Subset Selection in Regression*. Chapman and Hall, 2002.

[33] N. Ye, K.M.A. Chai, W.S. Lee, and H.L. Chieu. Optimizing F-measures: A tale of two approaches. In *ICML*, 2012.

[34] S. Clémençon, G. Lugosi, and N. Vayatis. Ranking and empirical minimization of U-statistics. *Annals of Statistics*, 36:844–874, 2008.

[35] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley-Interscience, 2000.

[36] H. Narasimhan, R. Vaish, and S. Agarwal. On the statistical consistency of plug-in classifiers for non-decomposable performance measures. In *NIPS*, 2014.

[37] O. Koyejo, N. Natarajan, P. Ravikumar, and I.S. Dhillon. Consistent binary classification with generalized performance metrics. In *NIPS*, 2014.

[38] A. Krause and C.E. Guestrin. Near-optimal nonmyopic value of information in graphical models. 2005.

[39] A. Das and D. Kempe. Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. In *ICML*, 2011.