Simple generative model for assessing feature selection based on relevance, redundancy, and redundancy

James Theiler

Intelligence and Space Research Division
Los Alamos National Laboratory, Los Alamos, NM 87545

ABSTRACT

An experimental procedure is proposed for measuring the performance of feature selection algorithms in a way that is not directly tied either to particular machine learning algorithms or to particular applications. The main interest is in situations for which there are a large number of features to be sifted through. The approach is based on simulated training sets with adjustable parameters that characterize the "relevance" of individual features as well as the collective "redundancy" of sets of features. In some cases, these training sets can be virtualized; that is, having specified their properties, one does not actually have to explicitly generate them. As a specific illustration, the method is used to compare variants of the minimum redundancy maximum relevance (mRMR) algorithm, and to characterize the performance of these variants in different regimes.

Keywords: machine learning, feature selection

Experience has shown, and a true philosophy will always show,
that a vast, perhaps the larger, portion of truth arises
from the seemingly irrelevant. —Edgar Allan Poe

Say something once, why say it again?
—Talking Heads, *Psycho Killer*

1. INTRODUCTION

Feature selection serves multiple purposes in machine learning, though those purposes are sometimes at cross-purposes with each other. The first and most tangible purpose is to improve the performance of classification or regression algorithms by reducing the number of features used by those algorithms; this alleviates the so-called Hughes effect,[1] or as Bellman calls it, the curse of dimensionality.[2] A second and somewhat subjective purpose is to find the most "relevant" features for the problem at hand, and thereby to learn something qualitative about this problem. While less assessable, this second purpose is in some ways more appealing, because it promises to get at the underlying mechanism (the "physics") of whatever it is that is being measured. From columns of numbers, you get insight – is data science wonderful, or what?

A difficulty with this second enterprise is that "relevance" turns out to be an elusive and not easily quantified quality. And this complication is amplified when the features are not independent of each other. Are two redundant features *both* relevant? Are they as relevant as another pair of features that are independently relevant? Is the relevance of a set of features equal to the sum of the relevancies of each of the features individually? Is a mildly relevant feature still relevant if there are not enough training samples to learn how to exploit that feature?

Driven by both of these desires – greater interpretability and lower out-of-sample error – there has been considerable development of feature selection algorithms and approaches. It is not the intent of this paper to introduce yet another feature selection algorithm (though the author must sheepishly admit that one new algorithm – mRMRx – will be introduced here). The primary objective here is to address the problem of *evaluating* feature selection algorithms. To this end, a model will be proposed for generating artificial datasets that can be used for testing feature selection algorithms under conditions that can be carefully controlled.

To begin, however, this paper will take a brief detour in Section 2 to describe a specific feature selection algorithm, the minimum redundancy maximum relevance (mRMR) algorithm. The explicit treatment of relevance

To appear in: *Proc. SPIE* **11139** (2019)

and redundancy in mRMR motivates the model, developed in Section 3, that allows the generation of features with pre-specified amounts of relevance and redundancy. In Section 4, this model is used to evaluate several mRMR-related feature selection schemes in regimes of higher and lower redundancy, and with a higher and lower range of relevance among the features.

The emphasis in this paper is on problems for which there are many candidate features. It is assumed that these features are not all equally relevant and not all independent of each other. This assumption is motivated in part by practical experience. But the assumption is also made because the alternative would be a machine learning dystopia. If there are many independent and equally relevant features, then each individual feature can provide only a tiny contribution to the total solution, which means that characterizing that contribution will be difficult, and in particular will require a lot of training data. And this characterization of weak contributions will be necessary for all of the many features.

Another issue is that efforts to actually understand the system, to produce models of the system that are explainable*, favor a smaller number of features. (As an aside, other practical issues, such as limited computer memory or the need for fast *in situ* processing, also favor fewer features.) Getting the lowest error fit of a model to data, however, often leads to a larger number of features.

2. FEATURE SELECTION

Our ultimate goal is to predict y from its associated $\mathbf{x}$, and the goal for feature selection is to identify a subset of features of $\mathbf{x}$ from which to learn how to make this prediction. That is, if $\mathbf{x} = [x_1, \ldots, x_M]^\mathsf{T}$, then we write $\mathcal{S} = \{s_1, \ldots, s_m\} \subset \{1, \ldots, M\}$ as a subset of features, and $\mathbf{x}_\mathcal{S} = [x_{s_1}, \ldots, x_{s_m}]^\mathsf{T}$ as the feature-selected $\mathbf{x}$ from which we seek to make our predictions.

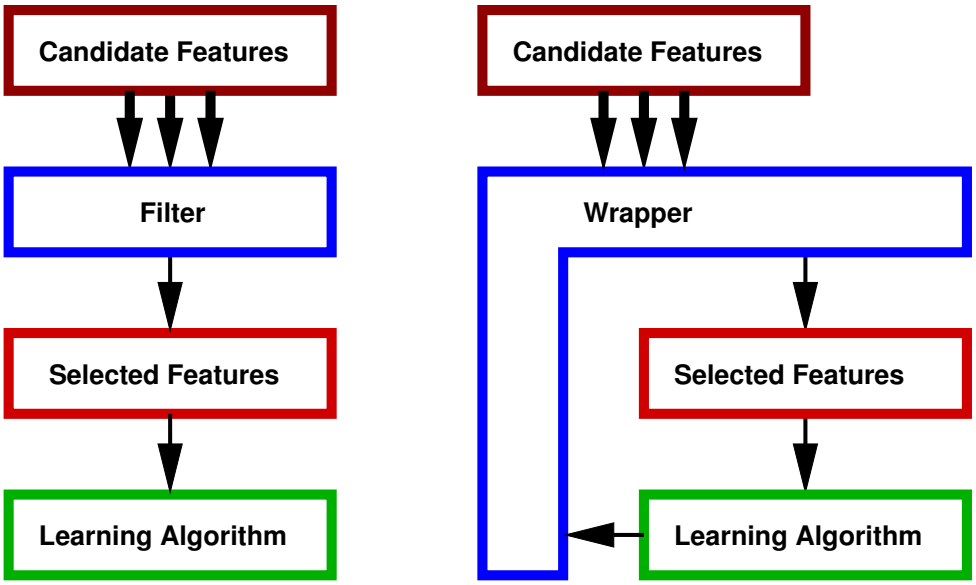


Figure 1. Two classes of feature selection algorithms. The filter (left) is fast, agnostic to the choice of learning algorithm, and able to deal with a very large number of features. The wrapper (right) is iterative and therefore slower, but because it is adaptive to the learning algorithm, it promises potentially better performance.

2.1. Filter vs Wrapper

Feature selection algorithms have traditionally been partitioned into two classes: filters and wrappers.[4,5] Although modern machine learning algorithms (such as LASSO,[6] LARS,[7] Grafting,[8] and Random Forests[9])

*Of course, the term *explainable*, in the increasingly popular context of explainable machine learning, is itself somewhat resistant to clear explanation.[3]

often combine aspects of both, the distinction – illustrated in Fig. 1 – is still useful. When computation is at a premium and/or the number of candidate features is especially large, filter-based feature selectors are more attractive. The filter sifts through the candidate features, selects a subset, and passes this subset off to the learning algorithm. Although wrapper-based feature selectors often achieve better performance, they are computationally more demanding, particularly when the number of candidate features is very large, and the choice of features ends up being influenced by the choice of learning algorithm.

2.2. Dependency, Relevancy, Redundancy

From an information theoretic point of view, we want to know how much information a particular subset of features of $\mathbf{x}$ provides about y. We begin with an information-theoretic measure of uncertainty in a single random variable X whose distribution is given by $p(x)$; this is the entropy:

$$H(X) = -\int p(x) \log p(x)\, dx. \tag{1}$$

For arbitrary random variables X,Y (in terms of entropy H) we define the mutual information:

$$I(X,Y) = H(X) + H(Y) - H(X,Y) = \iint p(x,y) \log \left[\frac{p(x,y)}{p(x)p(y)} \right] dx\, dy \tag{2}$$

Observe that $I(X,Y) = H(Y) - H(Y|X)$, where $H(Y|X)$ is the *conditional entropy* of Y conditioned on X: it is the uncertainty in Y if X is known. We'd like to choose the features in X so as to minimize this uncertainty. This is equivalent to maximizing $I(X,Y)$.

To find the m best features, what we ultimately want is to maximize $I(X_{\mathcal{S}}, Y)$ over the constraint that $|\mathcal{S}| \leq m$. One difficulty with this formulation is that as m grows, there are combinatorially many subsets that satisfy this constraint. The more immediate difficulty, however, lies in the expression for mutual information $I(X_{\mathcal{S}}, Y)$ for even just one subset. As m grows, computation of mutual information becomes more expensive, approximations become less accurate, and estimates from limited training data become increasingly problematic.

The minimum redundancy maximum relevancy (mRMR) approach of Peng *et al.*[10] recognizes this difficulty and replaces the optimization of this mutual information (called "dependency" in the mRMR paper)

$$\begin{aligned} I(\mathcal{S}, y) &= I(\{x_{s_1}, x_{s_2}, \ldots, x_{s_m}\}, y) \\ &= \iint \cdots \int p(x_{s_1}, x_{s_2}, \ldots, x_{s_m}, y) \log \left[\frac{p(x_{s_1}, x_{s_2}, \ldots, x_{s_m}, y)}{p(x_{s_1}, x_{s_2}, \ldots, x_{s_m})p(y)} \right] dx_{s_1} \cdots dx_{s_m}\, dy \end{aligned} \tag{3}$$

with a simultaneous optimization of two other quantities, which the mRMR paper calls "relevancy" and "redundancy." A key aspect of mRMR is these new quantities depend only on *pairwise* mutual information computations. In particular,

$$\text{Relevancy:} \quad D(\mathcal{S}, y) = \frac{1}{|\mathcal{S}|} \sum_{j \in \mathcal{S}} I(x_j, y) \tag{4}$$

$$\text{Redundancy:} \quad R(\mathcal{S}) = \frac{1}{|\mathcal{S}|^2} \sum_{j,k \in \mathcal{S}} I(x_j, x_k) \tag{5}$$

We seek a subset $\mathcal{S}$ with high relevancy (so that the elements of $\mathcal{S}$ are highly correlated with – *i.e.*, exhibit high mutual information with – the output value y) but low redundancy (so the elements of $\mathcal{S}$ are relatively independent of each other). In particular, the mRMR solution is chosen to maximize $D(\mathcal{S}, y) - R(\mathcal{S})$.

The additive nature of the expressions in Eq. (4) and Eq. (5) make it straightforward to build up an m-element set $\mathcal{S}$ one element at a time. The first element is the one that is most relevant, the one that maximizes $D(\{x_j\}, y) = I(x_j, y)$. Subsequent elements are given by

$$\text{argmax}_{j \notin \mathcal{S}} \ [I(x_j, y) - \text{mean}_{k \in \mathcal{S}} I(x_j, x_k)] \tag{6}$$

where

$$\text{mean}_{k\in\mathcal{S}} = \frac{1}{|\mathcal{S}|}\sum_{k\in\mathcal{S}}. \tag{7}$$

Thus, as we choose our next feature, we seek one that has high relevancy with respect to y but at the same time has low *average* redundancy with respect to the existing features.

Given this formulation, two alternatives naturally present themselves. A strawman formulation suggests that we should optimize relevance and ignore redundancy. This maximum relevance (MR) selector will maximize

$$\text{argmax}_{j\notin\mathcal{S}} \;\; I(x_j, y) \tag{8}$$

and the relative performance of MR and mRMR tells us how important it is to account for redundancy in selecting features.

To motivate the second alternative, consider a feature j that is *identical* to a feature k that has already been chosen. If this feature is not particularly correlated with other features in the existing set $\mathcal{S}$, then the penalty for exact redundancy with k will be attenuated by a factor of $1/|\mathcal{S}|$, and may end up chosen. But an exactly redundant feature cannot be helpful, so a larger penalty is appropriate for this case. This can be achieved by replacing the *mean* operator in Eq. (6) with a *max* operator. We call this the minimum redundancy maximum relevance with max (mRMRx) selector:

$$\text{argmax}_{j\notin\mathcal{S}} \;\; [I(x_j, y) - \max_{k\in\mathcal{S}} I(x_j, x_k)] \tag{9}$$

2.2.1. Computation

One of the main advantages of mRMR is that the estimate of pairwise mutual information (*e.g.*, as expressed in Eq. (2)) is much more reliable than estimates of higher dimensional mutual information (*e.g.*, as expressed in Eq. (3)), and is computationally much less expensive.

Further, by reducing the feature optimization problem to one involving only pairwise mutual information, the number of mutual information computations that need to be performed is bounded by $O(M^2)$, where M is the number of available features. Already this is much less than the potentially exponential number of subsets of features that might be considered by a more exhaustive scheme. But the greedy nature of the algorithms that construct $\mathcal{S}$ one feature at a time means that many of these pairwise mutual informations will never need to be computed. In the first step, there are the M computations of $I(x_j, y)$ but subsequent steps only compute $I(x_j, x_k)$ for $k \in \mathcal{S}$; this leads to $O(Mm)$ operations to compute m features.

3. GENERATING MODEL

The assessment of a feature selection algorithm can very much depend on the problem to which it is being applied. Although there is no shortage of test problems, from the UCI database[11] to specific feature selection challenges,[12] each problem has its own unique aspects. In addition to issues of how many features there are, how redundant they tend to be, and how many of them are relevant, there are further issues – such as the noise level in the data, or how nonlinear (and on the nature of that nonlinearity) the relationship is between the $\mathbf{x}$'s and y's – to further confound the comparisons that are being made.

What is proposed here is a model for creating artificial data sets to order – with adjustable "amounts" of relevance and redundancy in their features. In particular, we will have two parameters, α and β, that characterize how correlated (*i.e.*, redundant) features are with each other and how important (*i.e.*, relevant) features are to the quantity they are being used to predict.

This flexibility will enable us to investigate different aspects of the feature selection problem, but we also want to keep things simple and avoid confounding effects. For this reason, we make our model linear, and we take our features to have unit variance, to be normally distributed, and to be linearly correlated.

In particular, let the M components of a vector $\mathbf{x}$ be drawn from a multivariate normal distribution with covariance matrix R. That is: $x_1, \ldots, x_M$ are Gaussian, with $\langle\, x_j x_k \,\rangle = R_{jk}$. The matrix R will encode how redundant the features are, and in particular we will let R be Toeplitz, with $R_{jk} = \alpha^{|j-k|}$ for $0 \le \alpha < 1$.

As $\alpha \to 0$, the features become uncorrelated, and as $\alpha \to 1$, they become nearly identical. Although there are nominally M features, the number of "effectively independent" features is $O(M \log(1/\alpha))$. For instance, if $\alpha = 0.99$, then $\log(1/\alpha) \approx 0.01$, and a typical feature will be strongly correlated with $\sim$100 other features. So if $M = 1000$, say, then $M \log(1/\alpha) \approx 10$ suggests that there are (informally speaking) about ten distinct features, and the rest are correlated with them. But while those other features are correlated with these ten, they are not identical to them, so the eleventh feature will still be useful. In fact, as we will see, the fiftieth feature will still be useful, though there are diminishing returns after those first ten.

The model itself is linear and deterministic[†], so the y associated with a given $\mathbf{x}$ is defined by a vector of coefficients $\mathbf{a} = [a_1, \ldots, a_M]$:

$$y = \mathbf{a}^\mathsf{T}\mathbf{x} = a_1 x_1 + \cdots + a_M x_M \tag{10}$$

To make some features more relevant than others, we make their coefficients larger. So in particular, we take $a_k = z_k \times \exp(-\beta k/M)$ with z_k drawn from a unit-variance Gaussian. (Similar results are observed if z_k is drawn randomly from the interval $[-1, +1]$, or from the set $\{-1, +1\}$.) Thus larger values of the feature index k tend to have smaller magnitudes and the most relevant features will tend be those with low k values. As a technical point, we go ahead and normalize the coefficients: $a_k \leftarrow a_k/\sqrt{\mathbf{a}^T R \mathbf{a}}$ so that on average y will have unit variance. Informally, we can say that the effective number of relevant features is $O(M/\beta)$, and (even more informally) that the effective number of distinctly relevant features is $O(M \log(1/\alpha)/\beta)$.

Having defined covariance matrix R and coefficient vector $\mathbf{a}$, we can simulate as many $(\mathbf{x}, y)$ pairs as we need. Each pair is generated from a random vector $\mathbf{u}$, with each component independently generated from a zero-mean unit-variance Gaussian; from this $\mathbf{u}$, we then compute $\mathbf{x} = R^{1/2}\mathbf{u}$ and $y = \mathbf{a}^\mathsf{T}\mathbf{x}$. From a sufficiently large training set, we apply our feature selection algorithms to identify a suitable subset of features, use machine learning to estimate a model from these features (a linear model would be both simple and appropriate, but it is not required) and then we draw some more samples to estimate the out-of-sample RMS error of the learned model.

In the limit as the sample size becomes infinite, we can compute the root-mean square error for linear fits analytically. Let $\mathbf{c}$ be the vector of coefficients that are obtained by whichever feature selection and linear learning algorithms are employed. The RMS error is given by

$$\left\langle\, (y - \hat{y})^2 \,\right\rangle = \left\langle\, \left(\mathbf{a}^\mathsf{T}\mathbf{x} - \mathbf{c}^\mathsf{T}\mathbf{x}\right)^2 \,\right\rangle = \left\langle\, \left((\mathbf{a} - \mathbf{c})^\mathsf{T}\mathbf{x}\right)^2 \,\right\rangle = (\mathbf{a} - \mathbf{c})^\mathsf{T} \left\langle\, \mathbf{x}\mathbf{x}^\mathsf{T} \,\right\rangle (\mathbf{a} - \mathbf{c}) = (\mathbf{a} - \mathbf{c})^\mathsf{T} R (\mathbf{a} - \mathbf{c}). \tag{11}$$

The RMS error is minimized as $\mathbf{c}$ approaches $\mathbf{a}$, but although all of the elements of $\mathbf{a}$ are nominally nonzero, the feature-selected $\mathbf{c}$ will have only m nonzero elements.

From this, we can find, for a specified set of features, the optimal coefficients $\mathbf{c}$ in the $N \to \infty$ limit. Consider the $m \times M$ projection matrix P that maps the M-dimensional vector $\mathbf{x}$ of all candidate features onto the m-dimensional vector of selected features. The (j, s_j)'th element of P will be 1, for $j = 1, \ldots, m$, and all other elements of P will be zero. Let $\mathbf{b} = P\mathbf{c}$ be the m-dimensional vector of nonzero values in c. Note that $\mathbf{c} = P^\mathsf{T}\mathbf{b}$ projects this m-dimensional vector back into the M-dimensional space of all features, but with only m of the elements nonzero. We can extend Eq. (11) for RMS error to:

$$\begin{aligned} \left\langle\, (y - \hat{y})^2 \,\right\rangle =& (\mathbf{a} - \mathbf{c})^T R (\mathbf{a} - \mathbf{c}) \\ =& (\mathbf{a} - P^\mathsf{T}\mathbf{b})^\mathsf{T} R (\mathbf{a} - P^\mathsf{T}\mathbf{b}) = \mathbf{a}^\mathsf{T} R \mathbf{a} - 2\mathbf{b}^\mathsf{T} P R \mathbf{a} + \mathbf{b}^\mathsf{T} P R P^\mathsf{T} \mathbf{b} \end{aligned} \tag{12}$$

To find the optimal coefficients, take the derivative with respect to $\mathbf{b}$, and set to zero: $0 = -2PR\mathbf{a} + 2PRP^\mathsf{T}\mathbf{b}$, which leads to $\mathbf{b} = (PRP^\mathsf{T})^{-1} PR\mathbf{a}$, and since $\mathbf{c} = P^\mathsf{T}\mathbf{b}$, we have

$$\mathbf{c} = P^\mathsf{T}(PRP^\mathsf{T})^{-1} PR\mathbf{a} \tag{13}$$

as the optimal coefficients, given a projection matrix P.

[†]In this formulation, Eq. (10) does not have a "noise" term. This framework would readily permit such a term, but it is not clear that such a term is necessary. As long as we are in a regime with $m < M$, there will always be de-selected features, and the effect of these features will be the same as having additive noise.

Another advantage of this model is that it permits rapid calculation of mutual information. For two unit-variance Gaussian variables with cross-correlation ρ, the mutual information is given by the simple formula:

$$I = -\frac{1}{2}\log(1-\rho^2) \tag{14}$$

If we again consider the $N \to \infty$ limit, we can determine what these correlation coefficients are for the pairwise features; in particular,

$$\rho(x_j, x_k) = R_{jk}/\sqrt{R_{jj}R_{kk}} = R_{jk} \tag{15}$$

$$\rho(x_j, y) = (R\mathbf{a})_j/\sqrt{R_{jj}\mathrm{var}(y)} = (R\mathbf{a})_j \tag{16}$$

where the simplified form of these equations arise from the specification that y and each of the $\mathbf{x}$ components have unit variance. These formulas collectively permit us to work in the large N limit without ever actually simulating $(\mathbf{x}, y)$ samples. An obvious advantage here is computational, but another advantage is that we can investigate the behavior of algorithms in this asymptotic limit, and have one less confound (namely, finite N effects) to worry about. Of course, finite N effects are often important, so we will want to investigate those as well. But we can do this in a controlled way. For instance, we can use finite N to select features and fit coefficients and then use the $N \to \infty$ limit in Eq. (11) to get a precise measure of out-of-sample performance.

As a side remark: in this limit of infinite sample size, the subset selection problem becomes "monotone submodular" – for two subsets A and B, with $A \subset B$, the performance with A will be inferior to the performance with B; put another way: more features are always better, as long as there are enough training samples to properly take advantage of them.

4. RESULTS

4.1. Experiment

Choose a relatively small subset of m features from a large pool of M candidates ($1 \leq m \ll M$). In general, if $m \ll M$ features support a good fit to the data, that implies that there must either be a lot of irrelevant features or a lot of highly redundant features.

For the experiment in Fig. 2, datasets are generated with $M = 1000$ features and $N = 100$ data samples. With so many more features than samples, the potential for overfitting is acute. Good feature selection counters this problem (as does good regularization,[13] but to keep the experiment simple, simple linear fitting without regularization is performed), and for this experiment, the various mRMR-based feature selectors are compared. We can see in Fig. 2 that performance generally improves as the number of features m is increased, but especially for the more naive selectors, the performance turns around and gets worse as m is increased further.

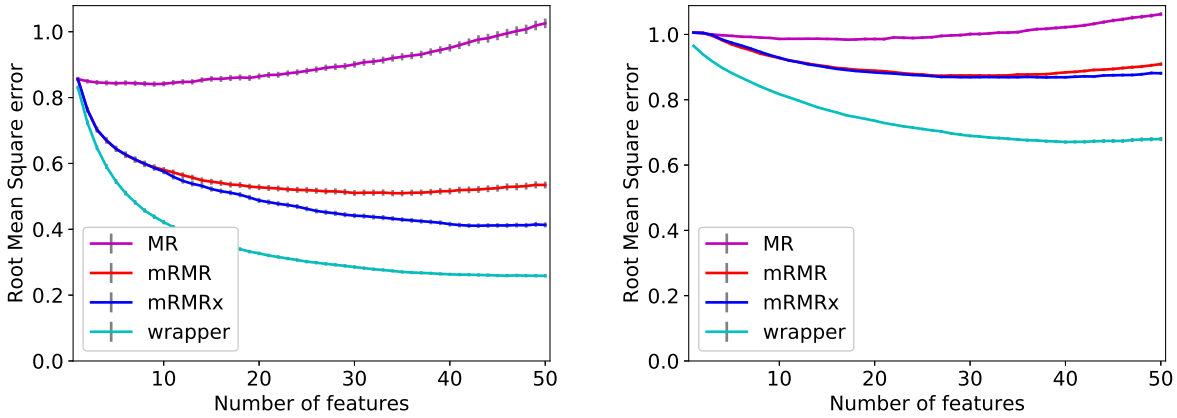
The experiment in Fig. 3 follows the same format as for Fig. 2, except that the $N \to \infty$ limit is considered. From a computational point of view, this is actually easier, because the appropriate quantities can be computed analytically without actually generating any data. We also observe, in this case, that increasing m always improves performance.

4.1.1. Wrapper

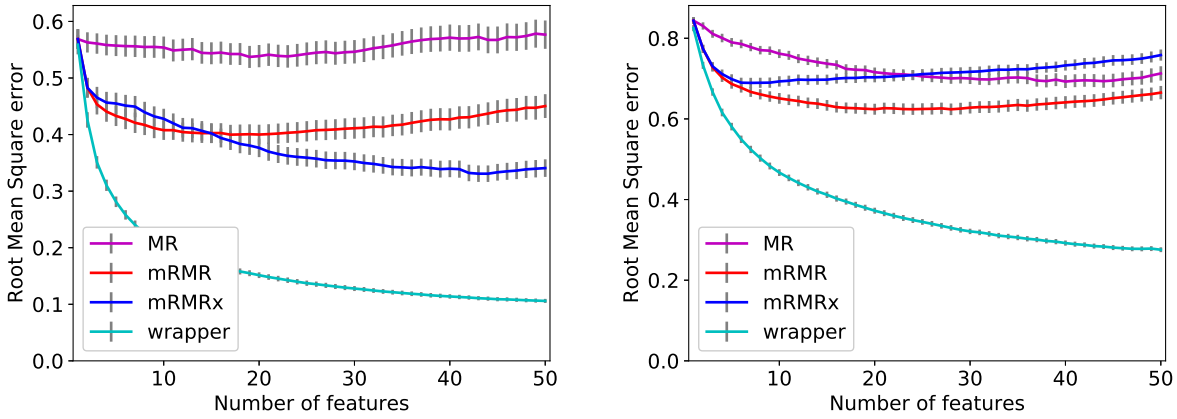
For the results in Fig. 2 and Fig. 3, the *wrapper* is stepwise forward selection. The first feature is the single best predictor (which is the same for the mRMR selectors; $s_1 = \mathrm{argmax}_s\ I(x_s, y) = \mathrm{argmax}_s\ \rho(x_s, y) = \mathrm{argmax}_s\ (R\mathbf{a})_s$), and subsequent features are added sequentially to minimize prediction error.

$$s_k = \mathrm{argmin}_j \mathrm{min}_f \left\langle\, (y - f(\mathbf{x}_{\mathcal{S}\cup j}))^2 \,\right\rangle. \tag{17}$$

In the $N \to \infty$ limit, we can again compute things analytically, without having to explicitly draw samples. Given an initial set of features $\mathcal{S}$, we consider for each $j \notin \mathcal{S}$ the projection matrix $P(j)$ based on the feature set $\mathcal{S} \cup j$. We compute $\mathbf{c}(j)$ from Eq. (13) using this $P(j)$, and then we compute RMS from Eq. (11) using this $c(j)$. The added feature j that produces the lowest RMS is the next feature that is selected.



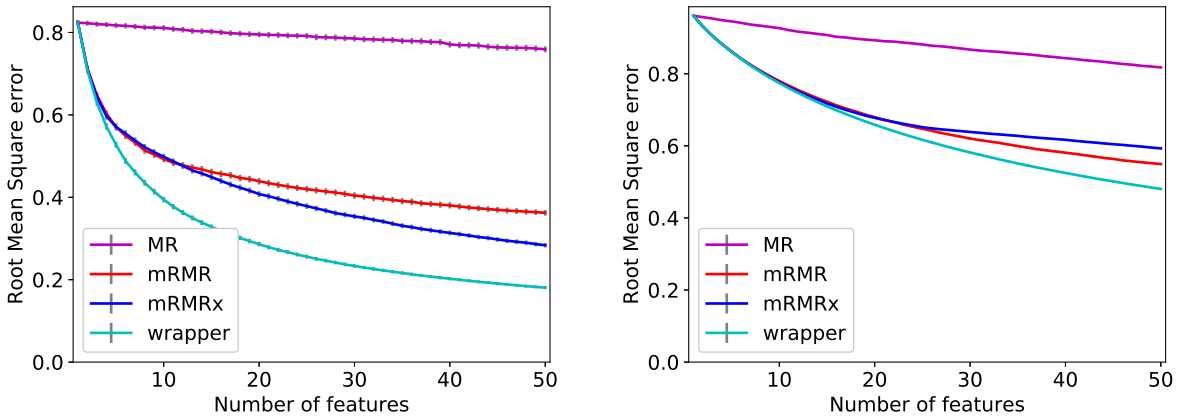
(a) $\alpha = 0.99$, $\beta = 0$ **(b)** $\alpha = 0.9$, $\beta = 0$



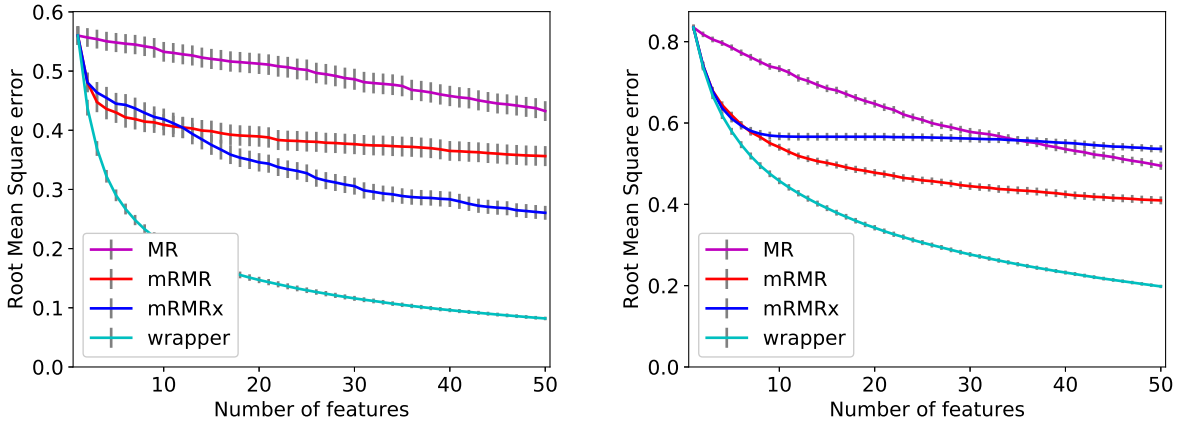
(c) $\alpha = 0.99$, $\beta = 10$ **(d)** $\alpha = 0.9$, $\beta = 10$

Figure 2. Performance of feature selection on simulated data with $N = 100$ samples and $M = 1000$ candidate features, shown for increasing number of selected features m. Performance is measured in terms of the root mean square error (so lower values are better) of the out-of-sample predictions, averaged over 100 trials. Vertical gray bars indicate standard error (standard deviation divided by square root of number of trials). The various algorithms sequentially choose the first fifty features according to their various criteria. In all of the above examples, the wrapper algorithm outperforms the filter-based algorithms, but also provides a kind of lower bound on what kind of performance is possible. The MR algorithm corresponds to a greedy "maximum relevance" filter that doesn't consider redundancy at all; it just selects features in order of relevance. (a,c) In the regime of high redundancy ($\alpha = 0.99$), the modified mRMRx algorithm performs better than the default mRMR. (b,d) In the regime of moderate redundancy ($\alpha = 0.9$), the standard mRMR outperforms the modified mRMRx. Larger values of β corresponded to diminished availability of features, and appears to magnify the discrepancy between mRMR and mRMRx.

To appear in: *Proc. SPIE* **11139** (2019)



(a) $\alpha = 0.99$, $\beta = 0$ **(b)** $\alpha = 0.9$, $\beta = 0$



(c) $\alpha = 0.99$, $\beta = 10$ **(d)** $\alpha = 0.9$, $\beta = 10$

Figure 3. Same as Fig. 2, but for the $N \to \infty$ limit. Here, we see that performance is monotonic with number of features.

5. CONCLUSIONS AND FUTURE WORK

The primary contribution of this work is a simple linear model that can provide a testbed for feature selection. Features in this testbed can be adjusted for greater or less redundancy with each other, and greater or less relevance to the quantity being predicted. The testbed can be used in a purely numerical manner, providing multiple trials with training and testing datasets of specified size for use in evaluating the performance of machine learning algorithms with different feature selection modules. Because the model is so simple, it can also be used to investigate some properties (such as the $N \to \infty$ limit) analytically, without ever generating actual data.

This testbed was employed to address a very specific feature selection question, regarding different variants of the popular minimum redundancy maximum relevance (mRMR) approach.

Redundancy matters. It was demonstrated that a feature selector based purely on maximum relevance (MR) performed poorly, compared to selectors that also tried to minimize redundancy. In comparing two variants – standard mRMR and modified mRMRx – it was found that mRMRx was better in situations where features exhibited a high degree of redundancy, but that mRMR did better with moderately redundant features.

The experiments also confirmed a long-held view of filters vs wrappers, in that the wrapper-based feature selector substantially outperformed all of the variants of the filter-based mRMR feature selectors.

For the work reported here, the generated data was used to investigate variants of the mRMR feature selection approach. An obvious extension of this work would be to consider other variants of mRMR, such as

$$\text{argmax}_{j \notin \mathcal{S}} \ [I(x_j, y) - \lambda \, \text{mean}_{k \in \mathcal{S}} I(x_j, x_k)] \tag{18}$$

which puts an adjustable weight λ on the redundancy penalty. There is an extensive literature on feature selection algorithms. Many of these algorithms will be amenable to analysis of the kind performed here.

In this paper, we restricted ourselves to regression problems. Replacing real-valued y with a binary value would enable the extension to classification problems. Thresholding at zero is the most natural choice, in this case, but other threshold values could be used to create imbalanced training sets, which would be more appropriate for analysis of feature selection in applications involving rare target detection. Using multiple coefficient vectors $\mathbf{a}$, multi-class training sets could be generated. Finally, replacing $\mathbf{x}$ with $\text{sign}(\mathbf{x})$ would enable the investigation of feature selection algorithms that employ binary features; the conditional mutual information maximization (CMIM) algorithm of Fleuret[14] is of particular interest: it is similar to mRMR, though it requires mutual information of *triplets* of variables.

The current algorithm requires that a set of M candidate features be specified at the beginning. In the online feature selection problem,[15,16] these features are not known *a priori*, but are made available in a sequential fashion. For instance, a genetic algorithm might be employed to create image processing pipelines that generate image planes from the raw imagery, with these image plane features then being fed into a back-end classifier[17] or regressor.[18] Due to memory constraints, only a bounded number of features can be kept at any given time, so the online feature selection algorithm needs criteria for accepting new features and discarding old features so as to continually improve classification performance. It will be of interest to modify the proposed synthetic data generation scheme to enable online feature generation.

Feature selection is a pervasive issue for a wide variety of data analysis problems that arise in many fields of science and engineering. A very practical application arises in spectral imaging, in which an effective subset of the available spectral bands is sought. The band selection problem[19,20] arises in multispectral[21,22] and especially in hyperspectral[23–36] imagery; these data-driven approaches can be informed by physics-based band selection criteria.[37–39]

6. ACKNOWLEDGMENTS

This work was supported by the United States Department of Energy (DOE) through the Laboratory Directed Research and Development (LDRD) program at Los Alamos National Laboratory.

REFERENCES

1. G. F. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Trans. Information Theory* **IT14**, pp. 55–63, 1968.
2. R. E. Bellman, *Adaptive Control Processes*, Princeton University Press, Princeton, NJ, 1961.
3. V. Vanhoucke, "Interpretability and post-rationalization: What neuroscience teaches us about making machines accountable." [Online] `https://becominghuman.ai/interpretability-and-post-rationalization-b812eda13783`.
4. R. Kohavi and G. John, "Wrappers for feature subset selection," *Artificial Intelligence* , pp. 273–324, 1997.
5. I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Machine Learning Research* **3**, pp. 1157–1182, 2003.
6. R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Statist. Soc. B* **58**, pp. 267–288, 1996.
7. B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Ann. Stat.* **32**, pp. 407–499, 2004.
8. S. Perkins, K. Lacker, and J. Theiler, "Grafting: Fast, incremental feature selection by gradient descent in function space," *J. Machine Learning Research* **3**, pp. 1333–1356, 2003.
9. L. Breiman, "Random forests," *Machine Learning* **45**, pp. 5–32, 2001.
10. H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Analysis and Machine Intelligence* **27**, pp. 1226–1238, 2005.
11. D. Dua and C. Graff, "UCI machine learning repository," 2019.
12. I. Guyon, S. Gunn, S. B. Hur, and G. Dror, "Result analysis of the NIPS2003 feature selection challenge," *Proc. NIPS* **17**, pp. 545–552, 2004.
13. J. H. Friedman, "Regularized discriminant analysis," *J. Am. Statistical Assoc.* **84**, pp. 165–175, 1989.
14. F. Fleuret, "Fast binary feature selection with conditional mutual information," *J. Machine Learning Research* **5**, pp. 1531–1555, 2004.
15. S. Perkins and J. Theiler, "Online feature selection using grafting," *Proc. ICML* **20**, pp. 592–599, 2003.
16. K. Glocer, D. Eads, and J. Theiler, "Online feature selection for pixel classification," *Proc. ICML* **22**, pp. 249–256, 2005.
17. N. R. Harvey, J. Theiler, S. P. Brumby, S. Perkins, J. J. Szymanski, J. J. Bloch, R. B. Porter, M. Galassi, and A. C. Young, "Comparison of GENIE and conventional supervised classifiers for multispectral image feature extraction," *IEEE Trans. Geoscience and Remote Sensing* **40**, pp. 393–404, 2002.
18. J. Theiler, N. R. Harvey, S. P. Brumby, J. J. Szymanski, S. Alferink, S. J. Perkins, R. B. Porter, and J. J. Bloch, "Evolving retrieval algorithms with a genetic programming scheme," *Proc. SPIE* **3753**, pp. 416–425, 1999.
19. S. De Backer, P. Kempeneers, W. Debruyn, and P. Scheunders, "A band selection technique for spectral classification," *IEEE Geoscience and Remote Sensing Letters* **2**, pp. 319–323, 2005.
20. C.-Y. Kuan and G. Healey, "Band selection for recognition using moment invariants," *Proc. SPIE* **5806**, pp. 122–130, 2005.
21. A. Murni, S. Mulyono, and D. Chahyati, "Evaluation of five feature selection methods for remote sensing data," *Proc. SPIE* **4553**, 2001.
22. J. Karlholm and I. Renhorn, "Wavelength band selection method for multispectral target detection," *Applied Optics* **41**, pp. 6786–6795, 2002.
23. S. D. Stearns, B. E. Wilson, and J. R. Peterson, "Dimensionality reduction by optimal band selection for pixel classification of hyperspectral imagery," *Proc. SPIE* **2028**, pp. 118–127, 1993.
24. S. Kumar, J. Ghosh, and M. M. Crawford, "Best-bases feature extraction algorithms for classification of hyperspectral data," *IEEE Trans. Geoscience and Remote Sensing* **39**, pp. 1368–1379, 2001.
25. S. B. Serpico and L. Bruzzone, "A new search algorithm for feature selection in hyperspectral remote sensing images," *IEEE Trans. Geoscience and Remote Sensing* **39**, pp. 1360–1367, 2001.

To appear in: *Proc. SPIE* **11139** (2019)

26. S. B. Serpico, M. D'Inca, F. Me1gani, and G. Moser, "A comparison of feature reduction techniques for classification of hyperspectral remote-sensing data," *Proc. SPIE* **4885**, pp. 347–358, 2003.
27. N. Keshava, "Angle-based band selection for material identification in hyperspectral processing," *Proc. SPIE* **5093**, pp. 440–451, 2003.
28. N. Keshava, "Distance metrics and band selection in hyperspectral processing with applications to material identification and spectral libraries," *IEEE Trans. Geoscience and Remote Sensing* **42**, pp. 1552–1565, 2004.
29. D. Korycinski, M. M. Crawford, and J. W. Barnes, "Adaptive feature selection for hyperspectral data analysis," *Proc. SPIE* **5238**, pp. 213–225, 2004.
30. M. Kumar, C. J. Duffy, and P. M. Reed, "Enhancing the performance of feature selection algorithms for classifying hyperspectral imagery," *Proc. International Geoscience and Remote Sensing Symposium (IGARSS)* **5**, pp. 3264–3267, 2004.
31. Y.-L. Chang and H. Ren, "A greedy modular eigenspace-based band selection approach for hyperspectral imagery," *Proc. SPIE* **5546**, pp. 406–415, 2004.
32. X. Cheng, T. Yang, Y.-R. Chen, and X. Chen, "Feature extraction and band selection methods for hyperspectral imagery applied for identifying defects," *Proc. SPIE* **5996**, 2005.
33. M. S. Alam and S. Ochilov, "Adaptive hyperspectral band selection," *Proc. SPIE* **5908**, 2005.
34. B. Paskaleva and M. M. Hayat, "Optimized algorithm for spectral band selection for rock-type classification," *Proc. SPIE* **5806**, pp. 131–138, 2005.
35. J. Theiler and K. Glocer, "Sparse linear filters for detection and classification in hyperspectral imagery," *Proc. SPIE* **6233**, p. 62330H, 2006.
36. I. Delibasoglu and M. Cetin, "Hyperspectral band selection using structural information via hierarchical clustering," *J. Applied Remote Sensing* **13**, p. 014526, 2019.
37. E. M. Bassett and S. S. Shen, "Information theory-based band selection for multispectral systems," *Proc. SPIE* **3118**, pp. 28–35, 1997.
38. W. B. Clodius, P. G. Weber, C. C. Borel, and B. W. Smith, "Multi-spectral band selection for satellite-based systems," *Proc. SPIE* **3377**, pp. 11–21, 1998.
39. S. S. Shen and E. M. Bassett, "Information theory based band selection and utility evaluation for reflective spectral systems," *Proc. SPIE* **4725**, pp. 18–29, 2002.

To appear in: *Proc. SPIE* **11139** (2019)