An Empirical Evaluation of Techniques for Feature Selection with Cost

Stephen Adams, Ryan Meekins, and Peter A. Beling
Department of Systems and Information Engineering
University of Virginia
Charlottesville, VA
{sca2c,rmm6ey,beling}@virginia.edu

***Abstract*—Feature selection is the process of selecting a subset of relevant features from the larger set of collected features. As the amount of available data grows with technology, feature selection becomes a more important part of the system-design process. In real-world applications, there are several costs associated with the collection, processing, and storage of data. Given that these costs can vary between data streams, it is important to consider the cost of features when performing feature selection. A majority of the feature selection algorithms select a relevant feature subset solely based on the merit and do not consider cost. In this study, we evaluate a previously proposed cost-based feature selection framework. We expand on the previously conducted experiments by testing a wider range of feature selection methods paired with the cost-based framework, testing a variety of classifiers, and sequentially adding features to the relevant subset based on the results of the cost-based framework. We find that the selection of the weight parameter that balances the effect of feature merit versus cost is tied to the choice of feature selection technique. The weight must be appropriately scaled with the value of the merit. Further, we confirm the previously tested results and offer insight into future research directions on the topic of feature selection and cost.**

***Index Terms*—feature selection, cost, classification**

I. Introduction

In many applications, there are a number of different data sources that could be collected. However, it is unlikely that all of the available data sources and extracted features are useful for prediction. It is quite possible that extracted features that are nothing more than noise can degrade the predictive ability of a machine learning algorithm. Large feature sets can suffer from other problems such as increased computation time, increased storage requirements, and increased complexity. These properties are collectively known as the *curse of dimensionality* [10]. In order to build a parsimonious model, features that contribute little information or degrade the performance of a model should be selectively removed from the feature set.

In real-world applications, data sources are not free and have some associated collection cost. Generally, cost in a data mining context refers to misclassification cost, where there are different penalties for incorrectly labeling observations. However, there are many notions of cost beyond misclassification cost. For example, cost could refer to the upfront cost of purchasing the sensor for collecting the data or it could refer to the cost of storing the data on hardware while waiting to be processed. For another example, ordering a test for medical diagnosis might provide useful information but with an associated cost. As the number of data sources and extracted features grow, these costs can add up, i.e. larger feature sets can be more costly to utilize with a machine learning algorithm. In some applications, the cost of each data source or feature could vary significantly. For example, force sensors can cost up to tens of thousands of dollars while vibration sensors are on the order of thousands of dollars.

Feature selection is the process of selecting a subset of relevant features from the larger set of collected features [3], [7], [8]. One possible method is to exhaustively test every possible combination of feature subsets, but this approach quickly becomes impractical as the number of features grows. Numerous general feature selection methods have been explored, however existing models that take cost into consideration when selecting features are limited.

A general cost-based feature selection framework was recently developed [4]. This framework establishes a tradeoff between the merit of a feature with the cost of that feature. A weighting parameter controls the relative impact of these two competing measures and can be set by the practioner. In this paper, we evaluate this general framework for feature selection with cost and expand on the experiments performed in [4] in three facets:

1) We expand on the number of feature selection algorithms used to produce the merit of each feature subset.
2) We use multiple types of classifiers in order to assess the interaction between the feature selection technique and the predictive model.
3) We sequentially add features to the tested feature subset based on the results of the cost-based feature selection algorithm and evaluate the cost and performance of each of the subsets.

The objective of this study is to confirm the previously published results for the general cost-based feature methodology and expand on the experiments by adding more degrees of freedom in terms of type of classifier and dimensionality of the feature subset.

This paper is organized in the following fashion. Section II gives background on feature selection literature and literature that combines feature selection and cost. Section III describes the general cost-based feature selection framework and gives

2375-9259/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDMW.2017.153

details on the feature selection methods tested in this study. Section IV describes the numerical experiments performed on the data sets and their results. In section V, we discuss the results of the numerical experiments and offer our conclusions.

II. Background

Feature selection techniques can be roughly divided into three categories: filters, wrappers, and embedded techniques. Filters [2], [22] are independent of the model used for prediction and are considered a pre-processing step. Wrappers [12], on the other hand, evaluate a feature based on its predictive ability given a model. When using filters, a feature set is selected before the model is trained while wrappers require an iterative process of selecting a candidate feature set, training a model, and then evaluating the predictive ability of the candidate feature set. The third feature selection technique, referred to as embedded methods [6], simultaneously select a feature set and train a predictive model. The cost-based framework evaluated in this paper was designed for filtering feature selection techniques.

When the financial cost of data streams varies significantly, this information should be included in the feature selection process. Cost can refer to many aspects of the data including financial cost, storage cost, and collection cost. These types of costs are generally referred to as "test cost" [14], [16], [21]. Learning algorithms that incorporate cost are referred to as "cost-sensitive learning algorithms". However, these algorithms take a different approach to the problem than that taken by classical feature selection techniques. Cost-sensitive learning algorithms assume that each measurement of an observation is associated with a cost. The classifier must decide whether the measurement or feature is needed in each instance given its cost. The classic example is medical diagnosis. When a patient presents with symptoms, which test should the doctor order to achieve the best diagnosis, given that the tests have varying cost? Can the doctor make a diagnosis once the information from the first test is received or are more tests required?

There are several methods for incorporating feature cost into the feature selection process. Min and Zhu [17] propose a wrapper that backtracks through the feature space but the acquisition cost for each feature is delayed. Min, Hu, and Zhu [15] developed a backtracking algorithm which puts a constraint on the total cost of the selected feature set. Grouped features with cost are considered in [18] where it is assumed that if a single feature from a group is included in the feature set the remaining features in that group can be acquired for free. These methods suffer from the same limitation that all wrappers face, namely that they do not scale well to large feature sets.

Embedded techniques do not suffer from scaling issues but the choice of a predictive model is restricted. Cost can be incorporated into decision tree construction by adding cost to the splitting criterion [14]. Cost can be added to the random forest algorithm by making the probability that a feature is randomly chosen for a tree in the forest inversely proportional to the cost [23]. Cost can be incorporated into the selection of features for latent variable models, such as hidden Markov models, through the use of prior distributions [1].

In general, filters offer the ability to scale to large feature sets and can also be paired with any type of predictive model. The general cost-based framework proposed in [4] was originally tested with two types of filtering techniques: correlation-based feature selection and minimum-redundancy-maximal-relevance (mRMR). Because this method is independent of the predictive model, it can be paired with numerous types of classification and regression techniques. Support vector machines were originally used as the classification model. In this paper, we evaluate the cost-based feature selection framework with three new filtering techniques and three new types of classifiers.

Iswandy and Koenig [9] propose a cost-based feature selection method that balances the evaluation of the feature set with acquisition cost of the features. This method differs from the general framework proposed in [4] by utilizing a specific type of filter and combining the filter results with the cost for use as the evaluation function of a genetic algorithm.

III. Framework

In this section, we first outline the general cost-based feature selection framework we implement. Then, we describe the four feature selection algorithms we pair with the framework. These methods are feature selection via concave minimization (fsvFS), infinite feature selection (infFS), mRMR, and ReleifF. The infFS method has a supervised and unsupervised formulation. The supervised version, referred to as SinfFS, is restricted to binary classification problems, while the unsupervised version, referred to as infFS, can be applied to a problem with any number of classes because the class labels are not required.

A. General Cost-Based Feature Selection

The general cost-based feature selection framework proposed in [4] balances the merit of a feature set with the cost of that feature set. Let M_S represent the merit of a candidate feature set S with dimensionality k. The merit of S is calculated using an evaluation function $f(\cdot)$ so that $M_S = f(S)$. Let C_S be the average cost of the features in S. The evaluation function for the cost-based framework is

$$MC_S = M_S - \lambda * C_S, \tag{1}$$

where λ is the parameter that controls the influence of the cost on the feature selection process. When the feature selection method ranks each feature, the dimensionality of the proposed feature set is reduced to 1, and the cost of feature set S becomes the cost of the proposed feature set instead of the average values of the features in S.

B. fsvFS

Feature selection via concave minimization [5] is a wrapper for feature selection in a binary classification problem. In this method, a separating hyperplane is found that maximizes the

separation between the two classes. The optimization problem is constrained to minimize the dimensions in the hyperplane and thus performs feature selection. This method is considered a wrapper because it iteratively solves the optimization problem. Let $\mathcal{A}$ and $\mathcal{B}$ represent two classes where the matrices $A \in R^{m \times n}$ and $B \in R^{k \times n}$ contain the data for these two classes. Let $P = \{x | x \in R^n, x^T w = \gamma\}$ be a separating hyperplane between $\mathcal{A}$ and $\mathcal{B}$. The objective is to find w and γ so that the plane adequately separates the two classes. This leads to the constraints $Aw \geq e\gamma + e$ and $Bw \leq e\gamma - e$ after normalization where e is vector of ones with an arbitrary length. This yields the following linear programming problem

$$\begin{aligned} \underset{w,\gamma,y,z}{\text{minimize}} \quad & \frac{e^T y}{m} + \frac{e^T z}{k} \\ \text{subject to} \quad & -Aw + e\gamma + e \leq y \\ & Bw - e\gamma - e \leq z \\ & y \geq 0, z \geq 0. \end{aligned} \tag{2}$$

Feature selection is incorporated by suppressing the components of w that do not help separate $\mathcal{A}$ and $\mathcal{B}$. This is incorporated into (2) by adding $e^T |w|_*$ to the objective where the components of $|w|_*$ are equal to 1 if the corresponding component of w is non-zero and 0 otherwise. A weight parameter θ is added to the objective function to control the tradeoff between $e^T |w|_*$ and the original objective function. This procedure is further refined by mapping to the training of a support vector machine, which strives to maximize the distance between the two parallel planes that separate the classes. The ∞-norm replaces $e^T |w|_*$ and the optimization problem becomes

$$\begin{aligned} \underset{w,\gamma,y,z,s}{\text{minimize}} \quad & (1-\theta)\left(\frac{e^T y}{m} + \frac{e^T z}{k}\right) + \frac{\theta}{2} e^T s \\ \text{subject to} \quad & -Aw + e\gamma + e \leq y \\ & Bw - e\gamma - e \leq z \\ & -s \leq w \leq s \\ & y \geq 0, z \geq 0. \end{aligned} \tag{3}$$

In practice, this procedure forces the components of w to 0 for irrelevant features. All features can be ranked using the absolute value of the estimated w. When paired with the general cost-based framework, the merit/cost metric for a single feature is $MC_S = |w| - \lambda C_S$.

C. infFS

Infinite feature selection [20] is a filtering feature selection technique that evaluates the relevance of each feature while considering all possible feature sets. The score produced by this algorithm considers the interaction with other features. This method maps the feature set to a graph and then models a feature subset as a particular path through that graph. Let $G = (V, E)$ be a graph, where V represents the set vertices in the graph and E represents the set of edges. Each vertex corresponds to a feature. The graph can be compactly represented as an adjacency graph A, where the elements of A represent a pairwise energy term. The energies are a linear combination of pairwise measures linking feature i to feature j

$$a_{ij} = \alpha \sigma_{ij} + (1 - \alpha) c_{ij}, \tag{4}$$

where α is a weight in $[0, 1]$, $\sigma_{ij} = max(\sigma_i, \sigma_j)$, σ_i is the standard deviation of the i^{th} feature, and c_{ij} is 1 minus the absolute value of the Spearman's rank correlation coefficient. The elements of A are calculated for $i, j = 1...N$ where N is the number of features. Let $\gamma = \{v_0 = i, v_1, ..., v_l = j\}$ denote a particular path of length l through G. The energy of path γ is calculated by

$$\xi_\gamma = \prod_{k=0}^{l-1} a_{v_k, v_{k+1}}. \tag{5}$$

The energy of all paths of length l between i and j can be calculated by using the adjacency matrix and the matrix power law

$$R_l(i, j) = A^l(i, j). \tag{6}$$

Using this, a single feature's energy can be calculated by

$$s_l(i) = \sum_{j \in V} A^l(i, j). \tag{7}$$

If the path length is extended to infinity, the energy for a single feature is calculated by

$$\begin{aligned} s(i) &= \sum_{l=1}^{\infty} \left(\sum_{j \in V} R_l(i, j) \right) \\ &= \left[\left(\sum_{l=1}^{\infty} A^l \right) \mathbf{e} \right] \\ &= [S\mathbf{e}]_i, \end{aligned} \tag{8}$$

where $\mathbf{e}$ is an array of ones. The convergence property of the geometric powers series allows for the computation of $\check{S} = (\mathbf{I} - \mathbf{r}A)^{-1} - \mathbf{I}$, which encodes all the information about the features. The score for each feature is calculated by $\check{s}(i) = [\check{S}\mathbf{e}]_i$. Features are ranked in descending order of the score. This method is unsupervised but a supervised method for binary classification can be implemented. The binary labels are used in calculating the correlation between features.

For ranking individual features using the cost-based framework, let $S = \check{s}(i)$. Then, the merit/cost metric becomes $MC_S = \check{s}(i) - \lambda C_S$.

D. mRMR

Minimal-redundancy-maximum-relevance feature selection [19] is a filter that utilizes information theory to select a relevant feature set with only a small number of redundant features. This method combines two measures, each representing one of these concepts. The relevance of the feature set S

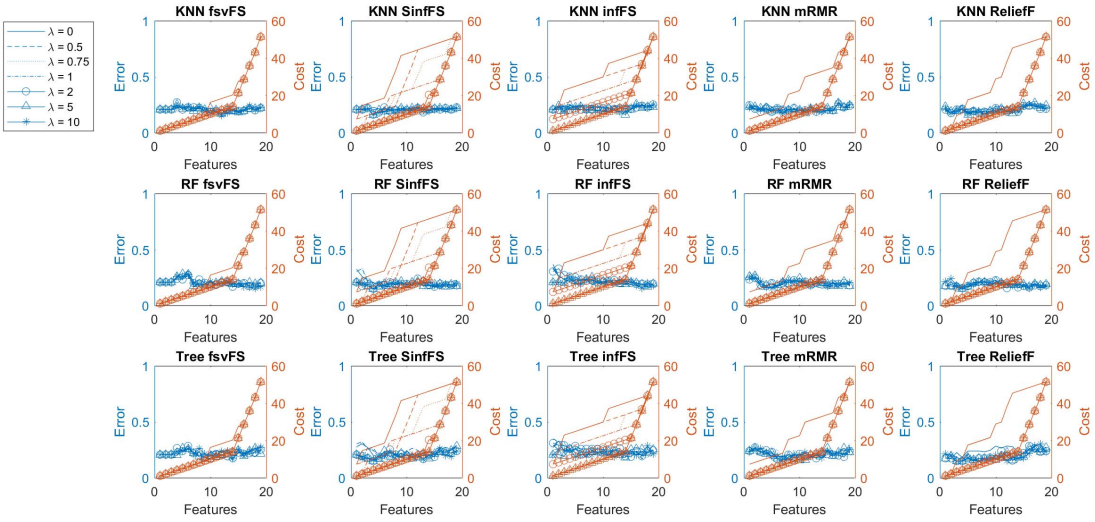

Fig. 1. Results from hepatitis data set experiments.

is assessed by calculating the average mutual information of all features in S with the class label c

$$D(S,c) = \frac{1}{|S|}\sum_{x_i \in S} I(x_i;c), \tag{9}$$

where x_i is the i^{th} feature, and $I(x_i;c)$ is the mutual information. The mutual information can be calculated using

$$I(x;y) = \int\int p(x,y)\log\frac{p(x,y)}{p(x)p(y)}dxdy. \tag{10}$$

The redundancy is assessed by calculating the mutual information between each feature in S

$$R(S) = \frac{1}{|S|^2}\sum_{x_i,x_j \in S} I(x_i;x_j). \tag{11}$$

The relevance and redundancy are combined to form the feature selection metric

$$\Phi(D,R) = D(S,c) - R(S). \tag{12}$$

This principle can be transformed into an iterative feature selection method that adds the feature to the relevant feature set that maximizes the following condition:

$$\max_{x_j \in X - S_{m-1}} \left[I(x_j;c) - \frac{1}{m-1}\sum_{x_i \in S_{m-1}} I(x_j;x_i) \right], \tag{13}$$

where S_{m-1} is the current set of relevant features.

When using the cost-based framework, the cost metric can be directly incorporated into the ranking process. Equation 13 becomes

$$\max_{x_j \in X - S_{m-1}} \left[I(x_j;c) - \frac{1}{m-1}\sum_{x_i \in S_{m-1}} I(x_j;x_i) - \lambda C_j \right]. \tag{14}$$

E. Relief and ReliefF

Relief was first introduced in [11] and was later generalized to ReliefF [13]. The Relief feature selection algorithm attempts to find features that are statistically relevant to the class. The original algorithm was restricted to a binary classification problem. At each iteration of the algorithm, an observation is chosen at random from the data set. This observation is compared to its nearest neighbor in the other class. The nearest neighbor is determined based on some type of distance measure. The nearest neighbor from the same class as the drawn observation is called a *near hit*, and the nearest neighbor from the other class is called a *near miss*. The weight w_l for the l^{th} feature is updated by

$$w_l = w_l - (x_l - x_l^h) + (x_l - x_l^m), \tag{15}$$

where x_l is the l^{th} feature value from the randomly chosen sample, x_l^h represents the feature value from the near hit, and x_l^m represents the feature value from the near miss. ReliefF [13] is an extension of the original Relief algorithm that can be used on multi-class problems.

The extension of ReliefF to the cost-based framework is achieved by directly substituting w_l for M_S when S is composed solely of the l^{th} feature. The merit/cost metric under these circumstances becomes $MC_S = w_S - \lambda C_S$.

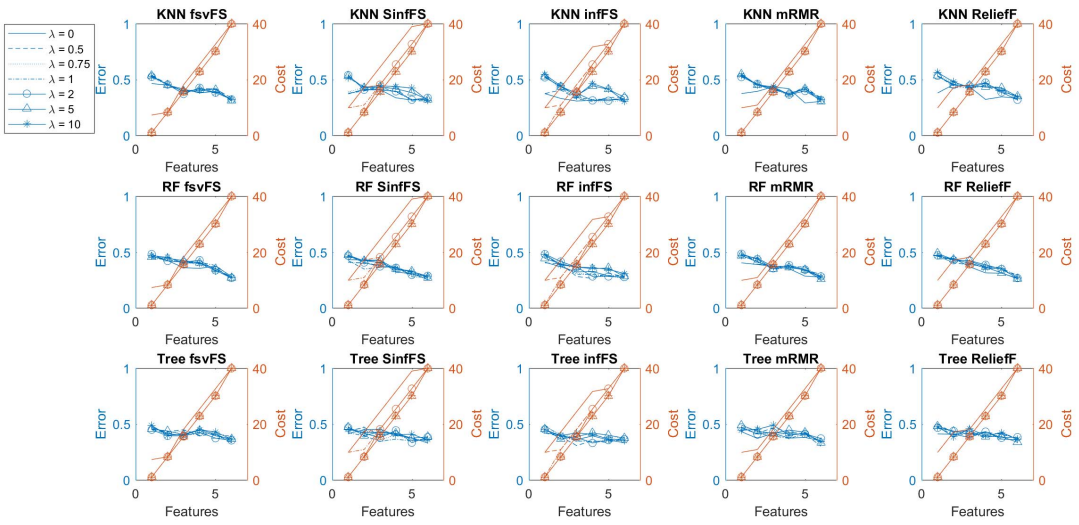

Fig. 2. Results from liver data set experiments.

Algorithm 1 Feature Evaluation Algorithm

1: Select a feature selection algorithm and classifier.
2: Select a cost weight parameter λ.
3: Rank the L features using the cost-based feature selection method and the selected feature selection algorithm.
4: **for** $m = 1 : L$ **do**
5: Put the top m features into the relevant feature set S.
6: Calculate the cost of the feature set C_S
7: Perform 10-fold cross validation using the selected classifier.
8: Calculate the average error E_S over the 10-folds.
9: **end for**

IV. Numerical Experiments

In this section, we describe the numerical experiments performed on the data sets using the various feature selection methods. We divided the experiments into those performed on a binary classification problem and those performed on a multiclass problem. We separate these experiments into two groups because some of the feature selection methods are specific to a binary problem. Specifically, the fsvFS method and the supervised form of the infFS method are limited to a binary problem.

The experiments are performed on 10 publicly available data sets from the UCI machine learning repository. Table I contains the binary classification data sets and includes the number of features and the number of samples in each data set. Table II contains the data sets for the multiclass problem and includes the number of features, number of samples, and number of classes. Only four of the data sets have costs associated with the features. For the remaining six data sets, we randomly generated feature costs between 0 and 1. The data sets with synthetically generated costs are starred in Tables I and II. This selection of data sets gives a wide range in the number of features, samples, and classes.

TABLE I
Binary Data Sets. Starred data set has synthetic costs.

Data set	# Features	# Samples
Hepatitis	19	155
Liver	6	345
Magic*	10	19,020
Pima	8	768

TABLE II
Multiclass Data Sets. Starred data sets have synthetic costs.

Data set	# Features	# Samples	# Classes
Letter*	16	20,000	26
Optdigits*	64	3823	10
Pendigits*	16	7494	10
Segmentation*	19	2310	7
Thyroid	21	3772	3
Yeast*	8	1484	10

On each data set, the general cost-based feature selection method is combined with one of the feature selection methods and used to rank all features. In order to see the effect of the weight parameter, we use $\lambda = [0, 0.5, 0.75, 1, 2, 5, 10]$. Note that $\lambda = 0$ represents a feature selection method that does not consider cost. Features are sequentially added to the relevant feature set based on the feature selection ranking. For each feature subset, we then perform a 10-fold cross validation to assess the error. Three standard classifiers are tested: K-nearest neighbor, random forest, and classification trees. Algorithm 1 displays the steps used to evaluate each feature selection method.

A. Binary Classification

In this section, we perform the numerical experiments on the binary classification data sets in Table I. We combine the general cost-based feature selection method with five feature selection methods: fsvFS, infFS, SinfFS, mRMR, and ReliefF. The cost and error calculated using Algorithm 1 for each data set are displayed in Figures 1 to 4.

The results from the numerical experiments demonstrate that the cost-based framework successfully selects feature subsets with less cost than a general feature selection algorithm ($\lambda = 0$). In a few of the experiments, the general feature selection algorithm selects feature subsets that outperform the cost-based methods in terms of error. However, this is only a small proportion of the test scenarios and generally all values of λ yield a similar error rate. This indicates that selecting cost effective features does not necessarily degrade predictive performance.

We found that error rate is independent of the classifier, i.e. all classifiers generally produced the same error rate given the same feature subset. Further, error rate decreased as the feature set grew in size, which is to be expected because classifiers generally perform better with more features.

On the binary problem, when there was a lot of weight on penalizing costly features, all the feature selection algorithms selected feature sets that minimize cost. These experiments indicate that the choice of λ is greatly influenced by the feature selection algorithm. The infinite feature selection technique produces weights with larger values than the other feature selection algorithms, e.g. the merit of the feature set M_S for infinite feature selection for any arbitrary feature set is larger than the M_S for the other tested feature selection algorithms. This means that larger values of λ must be chosen to minimize cost. Other than this fact, there was little difference between the feature selection algorithms. However, it is interesting to note that the unsupervised version of the infinite feature selection algorithm usually outperformed the supervised version in terms of minimizing cost.

B. Multiclass Classification

In this section, we perform the numerical experiments on the multiclass data sets in Table II. We combine the general cost-based feature selection method with three feature selection methods that can handle multiclass problems: infFS, mRMR, and ReliefF. The cost and error calculated using Algorithm 1 for each data set are displayed in Figures 5 to 10.

The results for the multiclass problem tend to reflect those from the binary problem. Generally, the type of classifier did not influence the overall error rate and error rate tended to decrease with larger feature sets. The cost-based feature selection method chose relevant feature subsets with less total cost than the methods that did not consider the cost of features. These experiments confirm that more care must be taken when selecting a value for λ when using the infinite feature selection technique because of the larger value of M_S.

V. Discussion and Conclusions

In this study, we confirm the results reported in [4] for the general cost-based feature selection framework. This framework can incorporate the cost of individual features into the feature selection process. We expand on the experiments performed in [4] by testing more feature selection methods to be used in the cost-based framework, test multiple types of classifiers, and perform experiments that sequentially add features to the relevant feature subset and then evaluate their performance. Specifically, we find that the three classifiers tested in this study general result in the same error rate given the same feature subset. This is to be expected because all feature selection methods used in this study are filters and therefore independent of the classifier.

Further, we find that the type of feature selection technique used in the cost-based feature selection framework does not affect the cost of the relevant subset or the performance in terms of accuracy as long as an appropriate value for λ is chosen. We tested seven values for λ. When $\lambda = 0$, the cost-based framework reduces to the normal feature selection problem which only tries to optimize for feature relevance. The selected value for the weight parameter is important to the type of feature selection algorithm calculating the merit of the feature set M_S because different algorithms have different scales for M_S. This is evident when comparing the results of fsvFS and infFS in Figure 1. Therefore, as long as the value for λ is scaled accordingly with M_S there does not appear to be much difference in the cost of the selected feature subset. When λ is set to a large value, the cost is minimized and all feature selection algorithms select the same feature subset.

This study is limited to a filtering feature selection approach. In future work, we would like to extend the analysis of incorporating cost into the feature selection process to wrappers and embedded feature selection techniques. While the numerical experiments in this study build on those presented in [4], they are far from exhaustive. Future work could continue to expand on the types of classifiers, types of feature selection methods paired with the cost-based framework, and the domain. The data sets tested in this study were all taken from the UCI machine learning repository and have been thoroughly investigated. Real-world applications of the cost-based framework are needed.

References

[1] Stephen Adams, Peter A Beling, and Randy Cogill. Feature selection for hidden Markov models and hidden semi-Markov models. *IEEE Access*, 4:1642–1657, 2016.

[2] Hussein Almuallim and Thomas G Dietterich. Learning with many irrelevant features. In *AAAI*, volume 91, pages 547–552, 1991.

[3] Avrim L Blum and Pat Langley. Selection of relevant features and examples in machine learning. *Artificial intelligence*, 97(1):245–271, 1997.

[4] Verónica Bolón-Canedo, Iago Porto-Díaz, Noelia Sánchez-Maroño, and Amparo Alonso-Betanzos. A framework for cost-based feature selection. *Pattern Recognition*, 47(7):2481–2489, 2014.

[5] Paul S Bradley and Olvi L Mangasarian. Feature selection via concave minimization and support vector machines. In *ICML*, volume 98, pages 82–90, 1998.

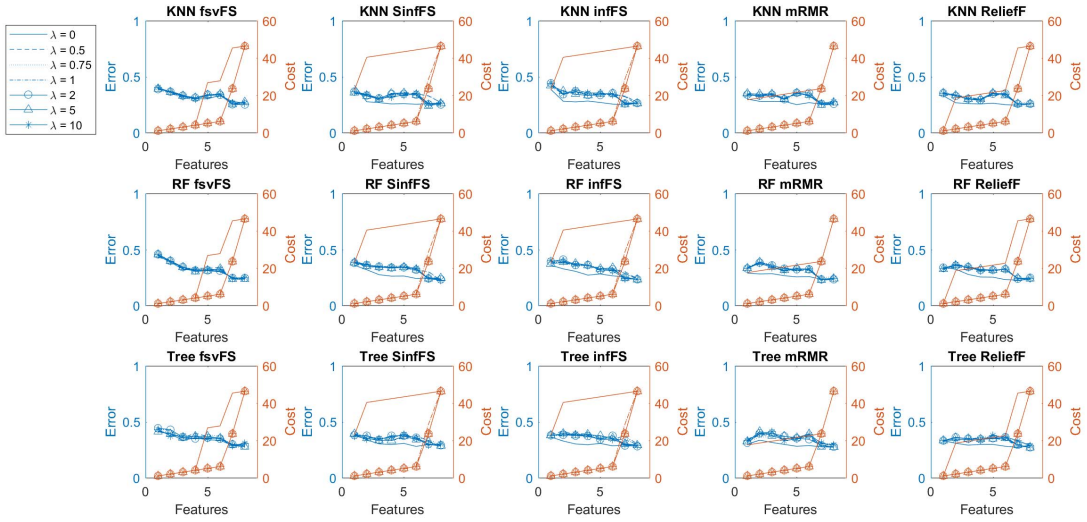

Fig. 3. Results from pima data set experiments.

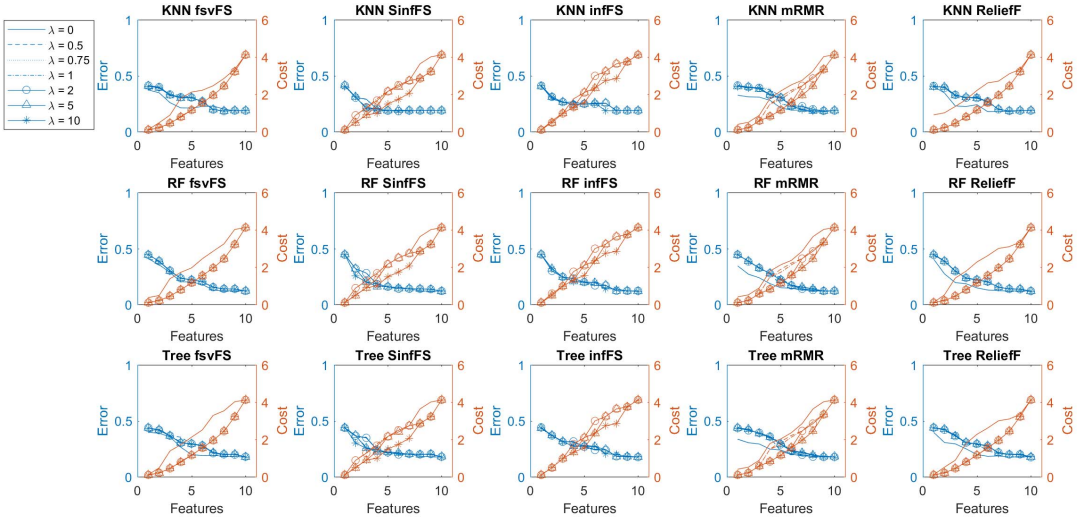

Fig. 4. Results from magic data set experiments.

[6] Walter Daelemans, Véronique Hoste, Fien De Meulder, and Bart Naudts. Combined optimization of feature selection and algorithm parameters in machine learning of language. In *ECML*, pages 84–95. Springer, 2003.

[7] Manoranjan Dash and Huan Liu. Feature selection for classification. *Intelligent data analysis*, 1(1-4):131–156, 1997.

[8] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[9] K Iswandy and A Koenig. Feature selection with acquisition cost for optimizing sensor system design. *Advances in Radio Science: ARS*, 4:135, 2006.

[10] Anil Jain and Douglas Zongker. Feature selection: Evaluation, application, and small sample performance. *IEEE transactions on pattern analysis and machine intelligence*, 19(2):153–158, 1997.

[11] Kenji Kira and Larry A Rendell. The feature selection problem: Traditional methods and a new algorithm. In *AAAI*, volume 2, pages 129–134, 1992.

[12] Ron Kohavi and George H John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2):273–324, 1997.

[13] Igor Kononenko. Estimating attributes: analysis and extensions of RELIEF. In *European conference on machine learning*, pages 171–182. Springer, 1994.

[14] Charles X Ling, Qiang Yang, Jianning Wang, and Shichao Zhang. Decision trees with minimal costs. In *Proceedings of the twenty-first international conference on Machine learning*, page 69. ACM, 2004.

[15] Fan Min, Qinghua Hu, and William Zhu. Feature selection with test cost constraint. *International Journal of Approximate Reasoning*, 55(1):167–179, 2014.

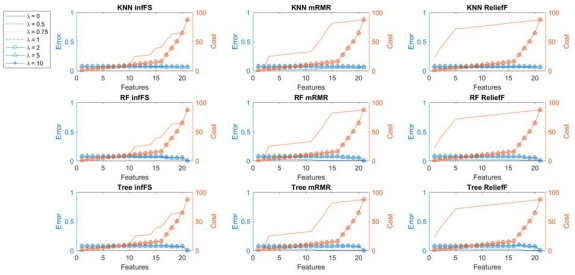

Fig. 5. Results from thyroid data set experiments.

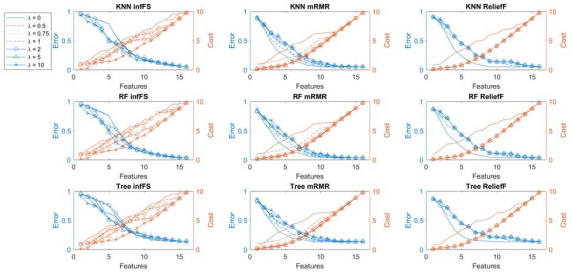

Fig. 6. Results from letter data set experiments.

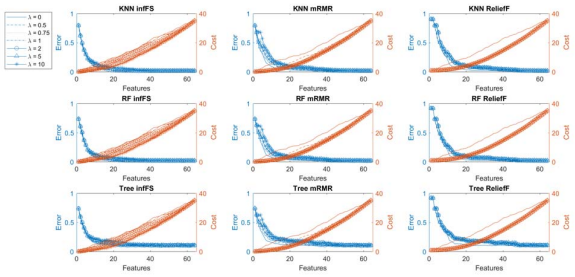

Fig. 7. Results from Optdigits data set experiments.

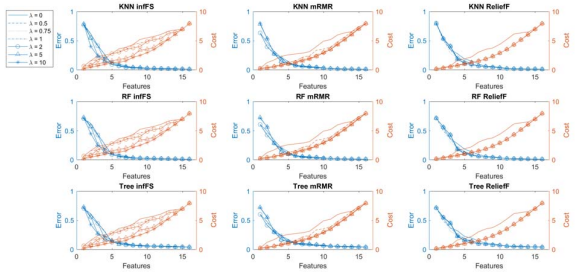

Fig. 8. Results from Pendigits data set experiments.

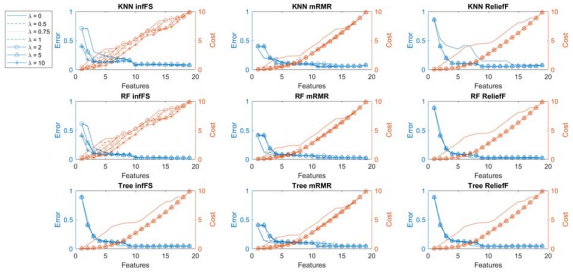

Fig. 9. Results from segmentation data set experiments.

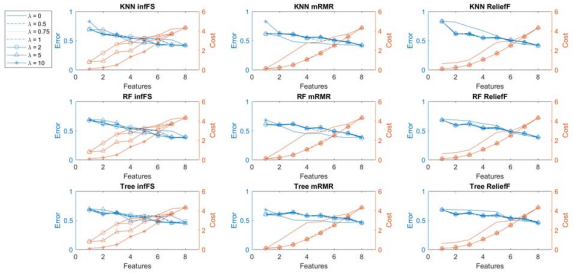

Fig. 10. Results from yeast data set experiments.

[16] Fan Min and Qihe Liu. A hierarchical model for test-cost-sensitive decision systems. *Information Sciences*, 179(14):2442–2452, 2009.

[17] Fan Min and William Zhu. Minimal cost attribute reduction through backtracking. *Database Theory and Application, Bio-Science and Bio-Technology*, pages 100–107, 2011.

[18] Pavel Paclík, Robert PW Duin, Geert MP van Kempen, and Reinhard Kohlus. On feature selection with measurement cost and grouped features. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 461–469. Springer, 2002.

[19] Hanchuan Peng, Fuhui Long, and Chris 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(8):1226–1238, 2005.

[20] Giorgio Roffo, Simone Melzi, and Marco Cristani. Infinite feature selection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4202–4210, 2015.

[21] Qiang Yang, Charles Ling, Xiaoyong Chai, and Rong Pan. Test-cost sensitive classification on data with missing values. *IEEE Transactions on Knowledge and Data Engineering*, 18(5):626–638, 2006.

[22] Lei Yu and Huan Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pages 856–863, 2003.

[23] Qifeng Zhou, Hao Zhou, and Tao Li. Cost-sensitive feature selection using random forest: Selecting low-cost subsets of informative features. *Knowledge-Based Systems*, 95:1–11, 2016.