A Multiple Objective Classifier Selection Methodology for Real World Problems

---

A Thesis

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

---

in partial fulfillment
of the requirements for the degree

Master of Science

by

Ryan Matthew Meekins

May 2018

APPROVAL SHEET

This Thesis
is submitted in partial fulfillment of the requirements
for the degree of
Master of Science

Author Signature: 

This Thesis has been read and approved by the examining committee:

Advisor: Dr. Peter A. Beling

Committee Member: Dr. Cody Fleming

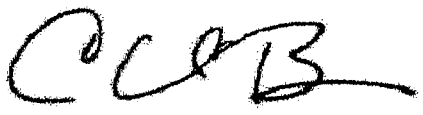
Committee Member: Dr. William Scherer

Committee Member: Dr. Stephen Adams

Committee Member: Dr. Kevin Farinholt

Committee Member: _____

Accepted for the School of Engineering and Applied Science:



Craig H. Benson, School of Engineering and Applied Science

May 2018

Abstract

Real world engineering problems often involve selecting between alternatives. As cyber-physical systems (CPS) become more prevalent in society, including smart cities, intelligent transportation networks, smart homes, etc., these alternatives will increasingly not only include variations in hardware but also in software. Stakeholders will have to decide between competing designs consisting of varying sensor sets and varying machine learning models. Ensuring that these selections are well guided and include multiple objectives such as performance, cost, and sensitivity is paramount. Many of these CPS will include machine learning classification models. The receiver operating characteristic (ROC) analysis is useful for selecting between alternative classification models based on performance and sensitivity, however, this analysis fails to treat classification models as a system of hardware and software, a CPS. Therefore, the ROC analysis ignores additional stakeholder objectives such cost and reliability. This thesis presents methods to extend the ROC analysis to include an additional objective of system cost. The methodology is demonstrated on three real world data sets that include additional cost information. The presented methodology is shown to drastically reduce the stakeholder decision space and guide stakeholder decisions using performance, cost, and sensitivity objectives.

Acknowledgements

I would like to thank everyone who supported me throughout graduate school at the University of Virginia. I am extremely grateful to my advisor, Dr. Peter Beling, for his support and guidance that truly helped shape my thesis. I would like to thank my committee members, Dr. Cody Fleming, Dr. William Scherer, and Dr. Stephen Adams, for their additional help and guidance. I would like to thank Luna Innovations Inc., my committee member, Dr. Kevin Farinholt, and additional members of the Performance Systems and Analytics Group at Luna Innovations for providing continued support, outstanding technical expertise, guidance, and motivation for this thesis research. I would like to thank my classmates, especially my good friend Jack Corbett, for their friendship and fresh perspectives on my research. I am also extremely grateful to all of my friends and family for supporting me and guiding me throughout my journey as a graduate student.

Contents

List of Figures

List of Tables

1 Introduction

Many engineering problems are characterized by selecting between alternatives. The goal is to select the alternative that's expected to be the best, where "best" is usually based on a domain specific objective. Real world engineering problems often involve selecting between alternatives using multiple objectives, such as selecting a system with a high performance, low cost, and a low sensitivity to uncertainty.

Cyber-physical systems (CPS) are becoming more prevalent in society and will be used for managing smart cities, intelligent transportation networks, smart grids, smart homes, autonomous vehicles etc. Each of these systems will require stakeholders to select between alternative designs, which will include variations in hardware and software, such as the decision model, sensors, processors, etc. Ensuring that these CPS selections are well guided and include multiple objectives, such as safety, reliability, cost, and performance, is paramount [1].

CPS rely heavily on machine learning classification models to make decisions based on sensor readings. Classifier selection is the process of selecting a classification model, which may include a system of hardware and software, such as the machine learning algorithm, features, sensors, etc., from a set of alternatives. Most classifiers are selected using classifier performance as the sole objective, where stakeholders select the classifier that returns the highest classification accuracy. However, real world classifier selection may involve additional objectives, such as cost, reliability, and sensitivity. Current classifier selection methods that include objectives other than performance are in the field of cost-sensitive learning.

Cost-sensitive learning is a type of machine learning that takes the costs of misclassifications and other types of cost into account [2]. The goal of cost-sensitive learning is to minimize the total cost, which consists of the misclassification cost, test or feature costs, computation costs and all other types of cost [3].

Cost-sensitive classifier selection seeks to select the optimal classifier that minimizes

the total cost. Cost-sensitive feature selection is a type of cost-sensitive classifier selection that aims to select a classifier that minimizes the feature set cost, an operating cost objective, while still maintaining a high performance. Generally, cost-sensitive classifier selection methods have focused on the objectives of maximizing performance and minimizing sensitivity to real world uncertainty, while cost-sensitive feature selection methods have focused on the objectives of minimizing operating costs and maximizing performance. However, a method that brings these ideas together to form one method that selects classifiers using multiple objectives including performance, operating cost, and sensitivity for real world problems has not been developed.

The presented methodology accomplishes this by building upon the receiver operating characteristics (ROC) analysis [4]. The ROC analysis is preferred because classification accuracy has been deemed an inadequate metric for real world problems because the costs of misclassification and class proportions are often unknown [5]. This new methodology expands upon the ROC Convex Hull (ROCCH) method, which selects a set of potentially optimal classifiers for any real world costs of misclassification and class distribution [6, 5]. The ROCCH method includes the objectives of performance and sensitivity, however, a common stakeholder objective of reducing operating cost is ignored. Therefore, the ROCCH method is expanded to include this cost objective by adding a new dimension of operating cost to the ROC graph. This new ROC Convex Hull with Cost (ROCCHC) method selects a set of classifiers using the objectives of performance, reduced operating cost, and low sensitivity and is based on the principle of Pareto optimality.

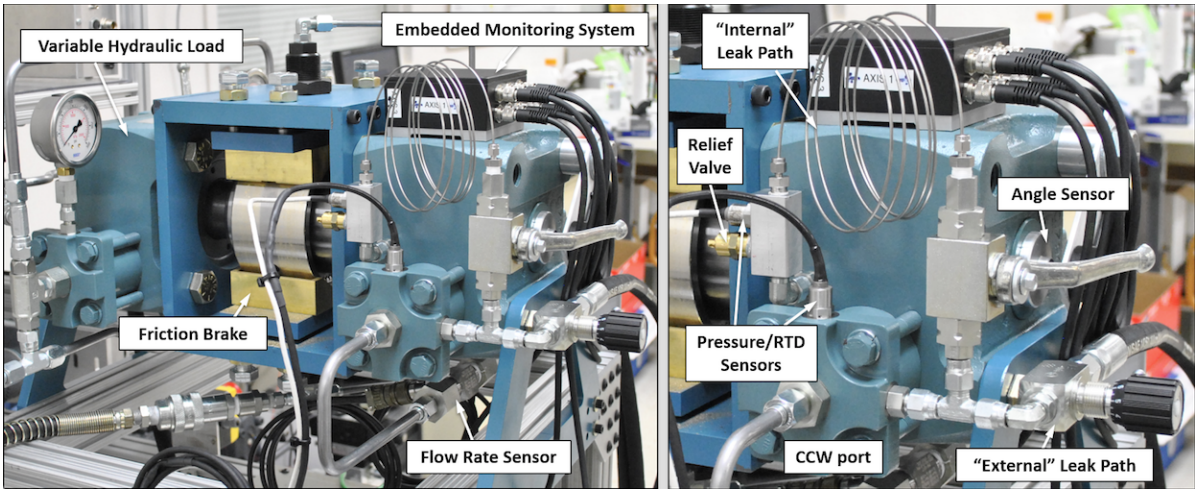


Figure 1: Hydraulic actuator setup at Luna Innovations Inc.

1.1 Motivation

We are currently working on developing a condition-based maintenance (CBM) system for hydraulic rotary actuators for the U.S. Navy with local engineering company, Luna Innovations Inc. The idea is for the Navy to move towards implementing these CPS on actuators using embedded hardware, where data-driven machine learning models will predict the health state of an actuator using sensor streams. Currently, these actuators are maintained according to a time-based maintenance policy on a five to six year interval. This cyclic repair amounts to huge costs in new parts and labor. The CBM systems could reduce these costs by providing decision makers with a "Do not repair" recommendation at a certain confidence level resulting from a classification model.

We are in the research and development phase of these CPS. Therefore, we don't have actual actuator sensor streams and health information from Navy vessels. We are working with data collected from an actuator experimental setup at Luna Innovations Inc. This experimental setup consists of an actuator rig with an embedded monitoring system, shown in Figure 1. The current sensors include accelerometers, pressure transducers, temperature sensors, flow-rate sensors, and an angular position sensor. The health state of the actuator can be physically manipulated by artificially inducing several types of common actuator faults, such as leaks or excessive loadings, with varying levels of severity.

A method for collecting labeled data streams from each sensor for each health state has been established. An automated method for importing this large amount of data and building classification models using various feature sets has been created. Classifiers with over 99% accuracy have been obtained using all sensors with several machine learning algorithms, including k-nearest neighbors, classification tree, and random forest [7]. This demonstrates that these sensors can be used to predict actuator health, however, accuracy isn't the only stakeholder objective and is known to be inadequate for real world problems [5]. Other objectives in this problem include system cost, energy consumption, and reliability. Using the full sensor and feature set would not be optimal in terms of these additional objectives because of the *curse of dimensionality* [8]. Therefore, methods are required to build and compare competing classifier systems using multiple objectives. These methods would be used to provide insights for the iterative design process of updating the research CPS and would ultimately be used for stakeholders to decide on the final CPS.

1.2 Purpose and Scope

The purpose of this thesis is to present a methodology for multiple objective classifier selection that can be applied to real world problems that are characterized by uncertainties. This methodology is different from past efforts in classifier selection because it includes performance, operating cost, and sensitivity objectives in one method. This methodology would be useful for stakeholders that have multiple objectives in mind when deciding on a classification system, such as performance, cost, and sensitivity.

This new methodology can be used for any binary classifier selection problem where an additional cost metric or cost ranking is known for each alternative. This method has been tested using three binary data sets, each of which has real world C_A information available. The Actuator data set stems from the motivating actuator problem in Chapter 1.1, where C_A is the classifier's sensor set cost. The other two data sets are from the UCI Machine Learning Repository [9] and include a data set for detecting diabetes in

Pima Native Americans and a data set for detecting hepatitis in hospital patients. Both of these data sets include real world test cost information, which is used as an additional objective.

The analysis of the Actuator data set will be presented to stakeholders at Luna Innovations Inc. This analysis will demonstrate how multiple objectives can be used for the selection of competing CPS. The results of this analysis will lead to more informed research and development decisions. There are additional objectives for the actuator problem that aren't analyzed in this work, such as reliability. Reliability and other information isn't currently known for competing systems. Further work for the actuator problem will include determining this information and expanding the presented methodology to include additional objectives.

2 Literature Review and Background

2.1 Chapter Overview

This chapter reviews current classifier selection methods in the literature and provides useful background for the thesis. Current methods in cost-sensitive feature selection are detailed and their shortcomings are analyzed. A background in binary classification and the ROC analysis is provided. This chapter exposes holes in current classifier selection methods and closes by suggesting an extension to the ROC analysis to include an additional cost objective.

2.2 Cost-Sensitive Feature Selection Methods

Cost-sensitive feature selection is a type of cost-sensitive classifier selection where the goal is to find an optimal feature set that has a high performance and a low feature set cost. There are three main ways to perform cost-sensitive feature selection, all of which expand on the traditional feature selection methods.

Traditional feature selection methods attempt to choose a subset of features that increase classifier performance. These methods include a filter, wrapper, and embedded method [10]. Filter methods use statistical merit metrics such as p-value to reduce or "filter" the feature set before choosing a classification algorithm. Wrapper methods use an iterative search process of changing a model's feature set and then evaluating the new model's performance. Embedded methods select features while building the machine learning model. A common example of an embedded method is the decision tree algorithm, which selects a new splitting feature from the full set by acting greedily with respect to an evaluation metric such as information gain. These three feature selection methods have been expanded to include feature costs for cost-sensitive feature selection.

Filter methods were expanded to include feature costs using a general cost-sensitive feature selection framework [11]. In this simple framework, feature costs are subtracted from a statistical merit metric to create a new evaluation metric that is cost-sensitive.

This subtraction includes a weight parameter placed on the feature cost, requiring stakeholders to determine a tradeoff between the statistical merit metric and the feature cost. This framework has also been expanded to include more filtering techniques and machine learning algorithms [12]. The problem with this general cost-sensitive feature selection framework is that it requires a tradeoff parameter between a statistical merit metric (i.e. p-value) and the feature set cost. How do stakeholders determine this tradeoff value for their specific problem? How sensitive are the results to this value? These current analyses are really performing a single objective analysis by collapsing objectives into one metric instead of a multiple objective analysis.

Wrapper methods have been recently expanded to include feature costs. Feature costs have been added to a backward selection wrapper by using two evaluation metrics, classification accuracy and feature set cost [13]. The algorithm searches for a minimal feature cost classifier that still achieves a certain accuracy. A backward selection wrapper has also been developed to find a high accuracy classifier that satisfies a maximum feature set cost constraint [14]. Generally, cost-sensitive wrapper methods use a heuristic approach of evaluating accuracy and feature set cost at each iteration in order to decide how to alter the feature set. The specific problem of these wrapper methods is that they use accuracy as the performance metric, which is an inadequate metric for real world problems.

Embedded methods for cost-sensitive feature selection have been developed recently. Feature costs and the costs of misclassification were added to the C4.5 decision tree algorithm by splitting on features that minimize the summation of the feature cost and the misclassification cost [15]. This algorithm is highly efficient, however, the resulting solution may not be globally optimal. The random forest algorithm was modified by setting the probability that a given feature will be selected as a potential split to the inverse of its cost [16]. This performed cost-sensitive feature selection because high cost features were unlikely to be selected. The specific problems for these embedded methods with cost is that [15] requires the costs of misclassification and class distribution, both of which are unknown for many real world problems. Also, [16] selects features using a

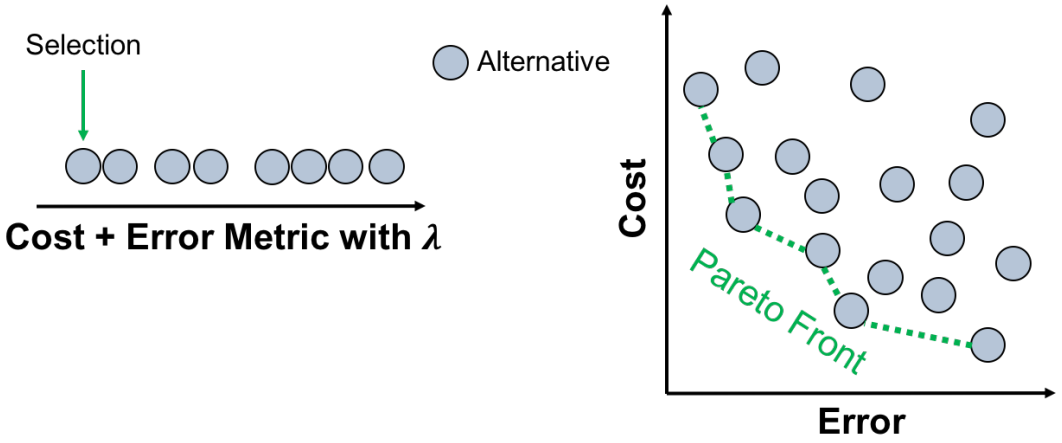


Figure 2: Collapsed analysis vs. a multiple objective analysis showing a Pareto Front of dominating alternatives

cost-sensitive probability, however, this isn't a robust methodology because it leaves cost-sensitive feature selection to chance. This methodology also doesn't allow for a sensitivity analysis and fails to allow stakeholders to make tradeoffs between objectives.

A general problem with these works in feature selection is that a collapsed objective analysis is performed, instead of a multiple objective analysis. These current methods collapse the performance and cost objectives into one metric that's optimized instead of optimizing over each objective. The difference between these two analyses is shown in Figure 2. This figure shows a collapsed vs. a multiple objective analysis with two objectives, cost and error, shown for various alternatives. The left side shows a collapsed objective analysis, where cost and error are collapsed using addition and a tradeoff parameter λ. This results in a ranking of the alternatives. The right side shows a multiple objective analysis where the two objectives are plotted for each alternative and a Pareto Front of dominating or Pareto optimal solutions can then be obtained by knowing that you want alternatives that have the lowest error and lowest cost. Each of the alternatives on the Pareto Front could then be selected as a final decision by using tradeoffs between objectives and constraints. This multiple objective analysis shows the sensitivity of any tradeoffs between the objectives and allows constraints in the objectives to be shown visually. Stakeholders could be presented with all of the Pareto optimal alternatives in

Table 1: Binary Classification Outcomes

	Assign Positive	**Assign Negative**
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

order to decide on tradeoffs or additional objectives and analyses in order to make a final decision.

There are an insufficient number of feature selection papers that perform a multiple objective optimization using Pareto optimality [17]. Two recent examples include a Particle Swarm Optimization [18] wrapper and an Artificial Bee Colony optimization [17] wrapper, both of which use objectives of error rate and the number of features used in the classifier. However, error rate or accuracy has been deemed inadequate for real world problems, where the costs of misclassification and class distribution are unknown. The presented methodology uses the ROC curve as the performance metric, which also allows for a sensitivity evaluation of competing classifiers. ROC curves are traditionally used for binary classification problems, however, they been expanded to multi-class problems [19]. The methodology in this paper focuses on binary classification, however, future work could be in expanding the presented multiple objective selection methodology to multi-class problems. The next sections discuss binary classification tasks and the ROC analysis.

2.3 Binary Classification

The goal of machine learning classification models is to assign new observations to their actual class. For binary classification models, where new observations can be assigned to either a "Positive" or "Negative" class, Table 1 shows the four possible outcomes. These are a true positive (TP: a "Positive" is correctly classified), a true negative (TN: a "Negative" is correctly classified), a false negative (FN: a "Positive" is incorrectly classified), and a false positive (FP: a "Negative" is incorrectly classified).

Binary classification models are often evaluated using false positive rate (FPR) and true positive rate (TPR) metrics. These are calculated as

$$\text{FPR} = \frac{\#\text{FP}}{\#\text{N}} = \frac{\#\text{FP}}{\#\text{FP} + \#\text{TN}}, \tag{1}$$

and

$$\text{TPR} = \frac{\#\text{TP}}{\#\text{P}} = \frac{\#\text{TP}}{\#\text{TP} + \#\text{FN}}, \tag{2}$$

where #N is the number of "Negatives" and #P is the number of "Positives" in the test data set. The other terms correspond to the number of a type of outcome from test observations.

Machine learning classification models assign new observations class membership probabilities based on what is learned from the training observations. For binary classification, an observation's class membership probabilities can be represented in terms of only its "Positive" class membership probability, denoted p_p, (where its "Negative" class membership probability is $1 - p_p$).

A cutoff threshold, $p_{cut} \in [0, 1]$, is used to assign new observations a class membership hypothesis based on the observation's p_p. All observations with $p_p \geq p_{cut}$ are assigned to the "Positive" class and the rest are assigned the "Negative" class. A perfect classifier has a cutoff threshold that corresponds to a FPR = 0.0 and TPR = 1.0, or no observations are incorrectly classified.

You can imagine that the value of p_{cut} greatly influences the FPR and TPR of a given model. Too low of a p_{cut} would result in a high FPR and too high a p_{cut} would result in a low TPR. The special cases of $p_{cut} = 0$ and $p_{cut} = 1$ assign all new observations to either the "Positive" or "Negative" class, respectively. The ROC graph is used to show all FPRs and TPRs for a given classification model as you alter p_{cut} from 0 to 1, showing the sensitivity of the model to p_{cut}.

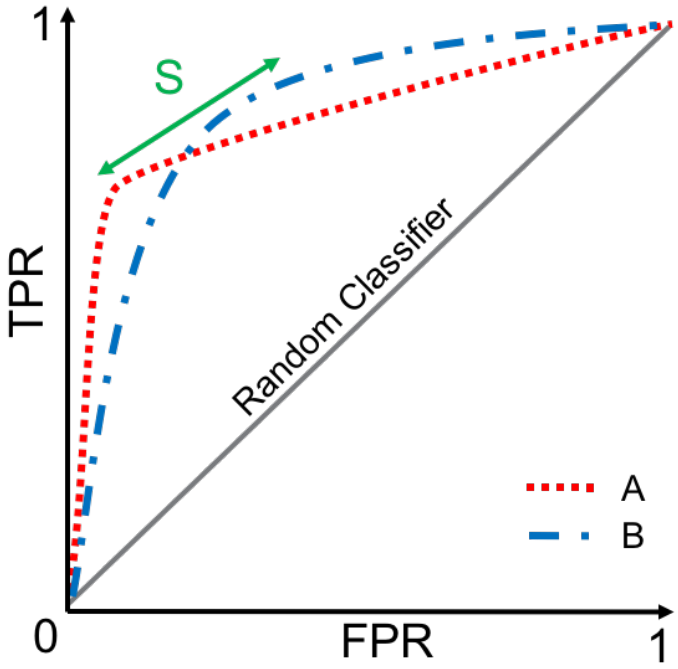


Figure 3: Receiver operating characteristics (ROC) graph.

2.4 Receiver Operating Characteristics

The ROC graph is a useful tool for visualizing, organizing, and selecting competing binary classifiers based on their performance. ROC curves have been favored in the machine learning community due to the realization that classification accuracy is an inadequate metric for real world problems, where the costs of misclassification and class distribution are often unknown [5].

The ROC curve shows a classifier's operation, in terms of its TPR and FPR, for all p_{cut} from 0 to 1. Figure 3 shows a ROC graph with classifiers, A and B, and a random classifier. The random classifier demonstrates a special case where TPR = FPR for all p_{cut}.

The ideal p_{cut} for a classifier can be determined in ROC space using a line with slope,

$$S = \frac{n \cdot c_{\mathrm{FP}}}{p \cdot c_{\mathrm{FN}}}, \tag{3}$$

where n is the expected number of "Negatives" in the real world, p is the expected number of "Positives" in the real world, c_{FP} is the real world cost of a FP, and c_{FN} is the real world cost of a FN. The ideal p_{cut} for a classifier is found by moving a line with slope S, also shown in Figure 3, from the upper left corner of the ROC graph down and to the right, until it intersects the classifier's ROC curve [4]. The point where this line intersects the classifier's ROC curve corresponds to the p_{cut} that would minimize the classifier's total misclassification cost, C_M, which is calculated as

$$C_M = n \cdot \text{FPR} \cdot c_{\text{FP}} + p \cdot (1 - \text{TPR}) \cdot c_{\text{FN}}. \tag{4}$$

This analysis ensures that C_M is minimized, however, the slope, S, can be hard to determine for real world problems because the costs of misclassification (c_{FP} and c_{FN}) and class distribution (p and n) are often unknown (Note that p and n may be different than the collected data's set class distribution, which is usually a real world sample and not the entire population). The ROC Convex Hull (ROCCH) method has been developed to deal with problems where these real world variables are unknown [5].

The ROC graph has been expanded by other researchers, including adding a third dimension of an algorithm's ability to detect difficult targets [20] and diagnostic latency for prognostic and health management applications [21]. These methods were useful for deciding on alternatives using a 3-D ROC, however, classifiers were not selected using additional cost information or Pareto optimality.

2.4.1 ROC Convex Hull Method

The ROC Convex Hull (ROCCH) method is used to identify a subset of classifiers that are potentially optimal without requiring the real world costs of misclassification or class distribution (slope S). This is accomplished using a convex hull in ROC space, where potentially optimal classifiers will have operating points (FPR, TPR) on the convex hull

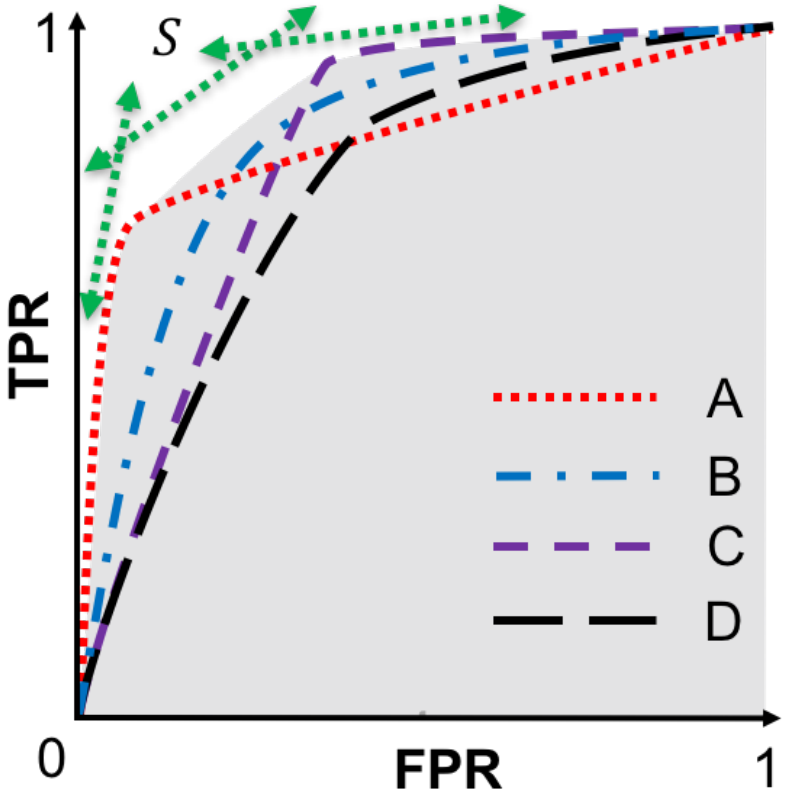


Figure 4: The ROC Convex Hull (ROCCH) method.

[5, 6]. The ROCCH method selects a set of classifiers using ROC performance as the objective.

The ROCCH analysis uses the concept of an "iso-performance" line, where all operating points in ROC space along a line with slope S will have the same misclassification cost, C_M. The "iso-performance" lines closer to the upper-left of the ROC space are more optimal because they correspond to a lower FPR and a higher TPR, resulting in a lower C_M. The ROCCH method selects the classifiers that correspond to a lower C_M by using a convex hull in ROC space. Classifiers that have operating points on the convex hull are potentially optimal for any real world slope S.

Classifier selection using the ROCCH method is shown in Figure 4. This figure shows a ROC graph consisting of four classifiers, A, B, C, and D. The convex hull of these classifiers is shown using the checkered region. Notice that classifiers A and C have

operating points on the ROC convex hull and classifiers B and D do not. This means that only classifiers A and C are potentially optimal or would be selected. Classifiers B and D would not be chosen for any real world class distribution and costs of misclassification. Further, all operating points of classifiers A and C that are not on the ROC convex hull would never be chosen and could be removed from consideration.

The ROCCH method selects classifiers using ROC performance, however, this method ignores additional cost objectives. For example, stakeholders could decide to implement a worse performing classifier if it is safer, more reliable, or less expensive. A sensitivity analysis can also be conducted on alternatives selected using the ROCCH method.

2.4.2 ROC Sensitivity

ROC sensitivity corresponds to the uncertainty in slope S for real world problems due to unknown costs of misclassification and class distribution. While the ROCCH method selects classifiers for the entire range of slope S, each selected classifier would only be optimal for a certain range in S. This facilitates a sensitivity analysis that characterizes classifiers based on the range of S where the classifier would be optimal.

Classifiers that would be optimal over a wider range of slope S are less sensitive to real world uncertainty, while classifiers that would only be optimal over a narrower range of slope S are more sensitive to uncertainty. For example, Figure 5 shows two ROC graphs with classifiers A and B. The left ROC graph shows the range in slope S where classifier A would be chosen while the right ROC graph shows the range in S where classifier B would be chosen. These ranges show us that classifier A is less sensitive to real world uncertainty (the range in slope S is wide) than classifier B (the range in slope S is narrow).

This ROCCH sensitivity analysis allows you to further compare ROCCH selected classifiers. Generally, selecting a classifier that is less sensitive to real world uncertainty would be better than selecting a more sensitive classifier. However, the specific range in this sensitivity can be used for selection for a specific problem. If the costs of misclas-

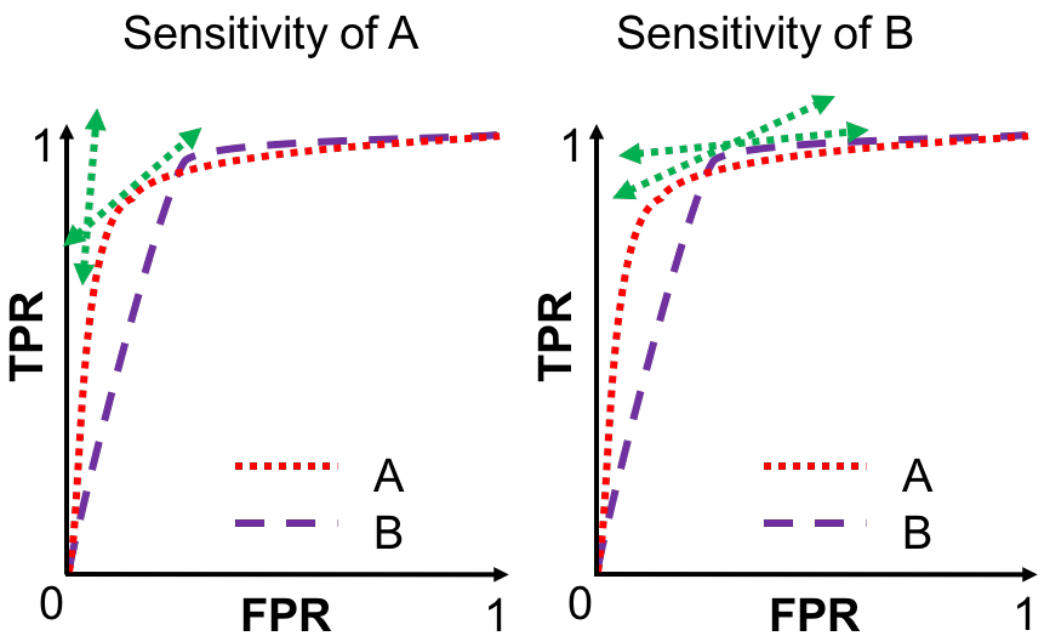


Figure 5: ROC sensitivity analysis.

sification and the real world class distribution can be estimated to be within a certain range, this information could be used during classifier selection. For example, classifier B in Figure 5 may be selected over classifier A if it's believed that the real S falls into the range in which classifier B would be optimal.

2.4.3 ROC Confidence Intervals

The literature mentions that ROC curves can really only be compared when there is a measure of variance [4]. Obtaining a measure of ROC variance requires generating multiple ROC curves for each classifier, which can be accomplished by generating the classifier with various training and testing sets or by using bootstrap methods. There are two methods for determining ROC variance using multiple ROC curves, vertical averaging and threshold averaging. Both of these methods determine an average ROC curve and pointwise confidence bounds at a specified confidence level. Vertical averaging samples the multiple ROC curves at fixed FPRs and averages the corresponding TPRs. This method generates pointwise confidence bounds in the TPR direction. Threshold

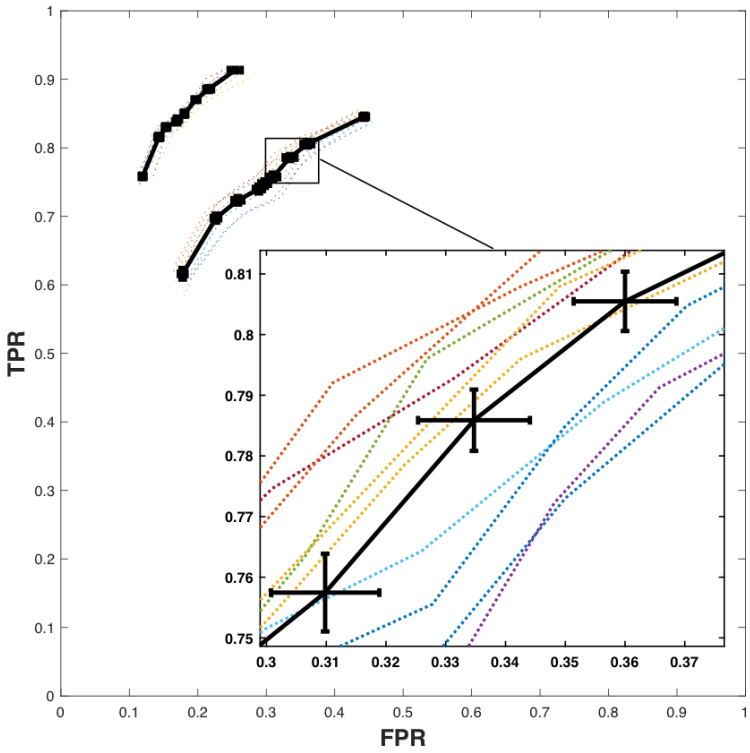


Figure 6: ROC confidence intervals using the threshold averaging method.

averaging samples the multiple ROC curves at fixed cutoff thresholds and averages the corresponding FPRs and TPRs [22]. This method generates pointwise confidence bounds in the FPR and TPR directions. The threshold averaging method has been preferred in the literature because the cutoff threshold, unlike FPR which is sampled in vertical averaging, is under direct control by a stakeholder.

Threshold averaging is shown in Figure 6. This figure shows two classifiers, each with multiple ROC curves, and threshold averaging with a 95% confidence level computed in MATLAB. The figure includes a zoom window to more easily see the multiple ROC curves, the average ROC, and pointwise confidence bounds for one of the classifiers. The multiple ROC curves are shown as dashed lines while the average ROC and pointwise confidence bounds are shown as solid lines. The pointwise confidence bounds show the TPR and FPR standard error at each discrete cutoff threshold.

Threshold averaging is used to generate ROC bands for a given classifier. An ROC

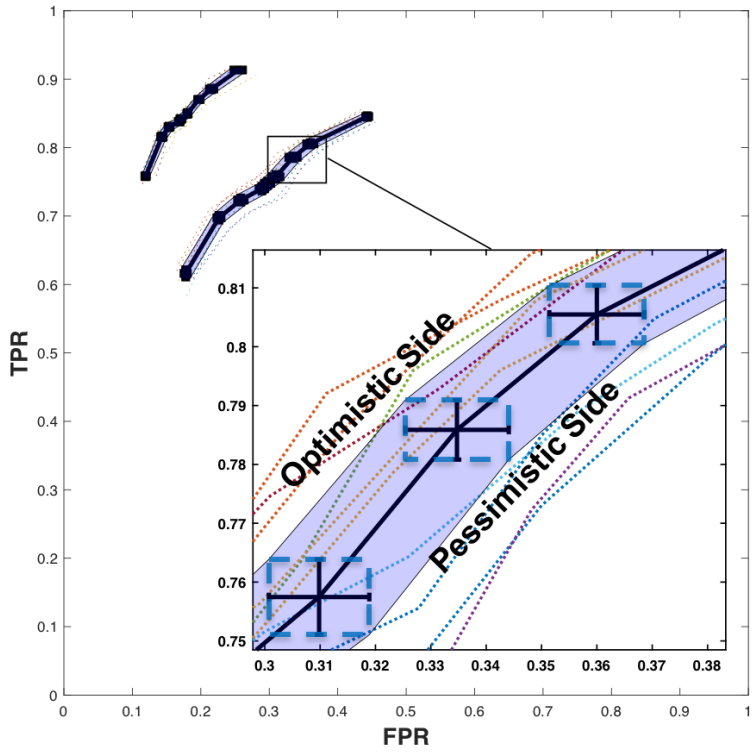


Figure 7: ROC bands using the Simultaneous Joint Confidence Regions (SJR) method.

band shows the range in expected performance of a classifier in ROC space for a certain confidence level. ROC bands are created by sweeping along the pointwise confidence bounds. One method for generating an ROC band is the Simultaneous Joint Confidence Regions (SJR) method [22]. This method sweeps along the upper left corners of the pointwise confidence bounds to create the optimistic side of the band and the lower right corners of the pointwise confidence bounds to create the pessimistic side of the band. The SJR method is shown in Figure 7.

This figure shows the same two classifiers from Figure 6, except now includes the classifier's ROC bands calculated using the SJR method as a shaded region. The figure also shows the optimistic and pessimistic sides of the ROC band, with the optimistic side having a better ROC performance.

2.5 Chapter Summary

This chapter reviewed literature, background, and methods in cost-sensitive feature selection, binary classification, and the ROC analysis. This section detailed current holes in the cost-sensitive learning literature and detailed the ROC analysis. This information will be useful for understanding the presented methodology, which expands on these current methods to fill holes in the literature. These current methods will be useful information for when they are applied to the actuator problem and other data sets in Chapter 4.

3 Methodological Approach

3.1 Chapter Overview

This chapter presents the classifier selection methodology that is used to select classifiers using multiple objectives. This methodology expands on the ROC analysis by adding a low operating cost objective. The ROC analysis was used because it's required for real world problems, which are characterized by unknown costs of misclassification and class distribution. The presented selection methodology includes the stakeholder objectives of high performance, low cost, and low sensitivity.

This chapter also presents a methodology for classifier generation. Generating classifiers is important because these are the competing alternatives. The "best" alternative can only be selected if it is known or has been generated. The two types of classifier generation are an exhaustive generation, where all classifiers are generated, and a search method, where classifiers are generated based on previously generated classifiers. This section details methods to determine which generation method should be implemented based on a given problem.

3.2 Methodology for Multiple Objective Classifier Selection

The presented multiple objective classifier selection methodology selects classifiers using the ROC analysis methods with Pareto optimality. Pareto optimality uses the idea of objective dominance to select a Pareto set of non-dominated alternatives. The current objectives in this methodology are high performance, low cost, and low sensitivity. The low cost objective is added to the ROC analysis in the next section.

3.2.1 ROC Convex Hull with Cost Method

The ROC Convex Hull with Cost (ROCCHC) method extends the ROC analysis to aid in real world classifier selection when additional cost information is known. The ROCCHC method adds an additional objective of low cost to the ROCCH method by adding this

additional cost dimension to the ROC graph. This additional cost may be any type of cost other than misclassification cost, such as the expected capital and operating expenses, including the hardware, software, personnel, electricity, etc., of the competing classifiers.

In the following, this additional cost is denoted C_{A}, where C_{A_i} corresponds to the additional cost of the i^{th} classifier. This analysis assumes that C_{A_i} is known for all competing classifiers, where C_{A} could be ordinal (i.e. 1^{st}, 2^{nd}, 3^{rd},...) or numerical.

This method builds upon the ROCCH method by using the convex hull in ROC space, however, this method requires computing a convex hull for each unique C_{A} in a set of competing classifiers. The ROCCHC method is shown visually in Figure 8, which shows classifiers A, B, C, and D each with C_{A} shown in parenthesis. Since there are two unique C_{A}'s of \$1 and \$2, two convex hulls are computed and shown.

The convex hull calculated at C_{A} of \$1 is shown the left ROC graph. This convex hull was computed while including classifiers B and D because their C_{A} is less than or equal to \$1. This convex hull selects classifier B. Classifier D is not selected because it has a worse performance and an equal cost to classifier B, or is dominated by classifier B.

The convex hull calculated at C_{A} of \$2 is shown in the right ROC graph. This convex hull was computed using all classifiers (A, B, C, and D) because all have a C_{A} of less than or equal to \$2. This convex hull selects classifiers A and C because neither are dominated because they have the highest ROC performance.

It is important to note that the ROCCHC method will always select at least one classifier at the minimum C_{A} because they can't be dominated in terms of cost. Also, a classifier i can only be selected when computing the convex hull for C_{A_i}, which will include all classifiers with a C_{A} equal to or less than C_{A_i}. This ensures that the convex hull method selects classifiers that are Pareto optimal, in terms of ROC performance and cost.

The steps to select classifiers from a list of m classifiers using the ROCCHC method is provided in Algorithm 1. The inputs to the algorithm are C_{A_i} and the ROC curve, r_i, of each competing classifier. The number of ROC curve points for each classifier is the

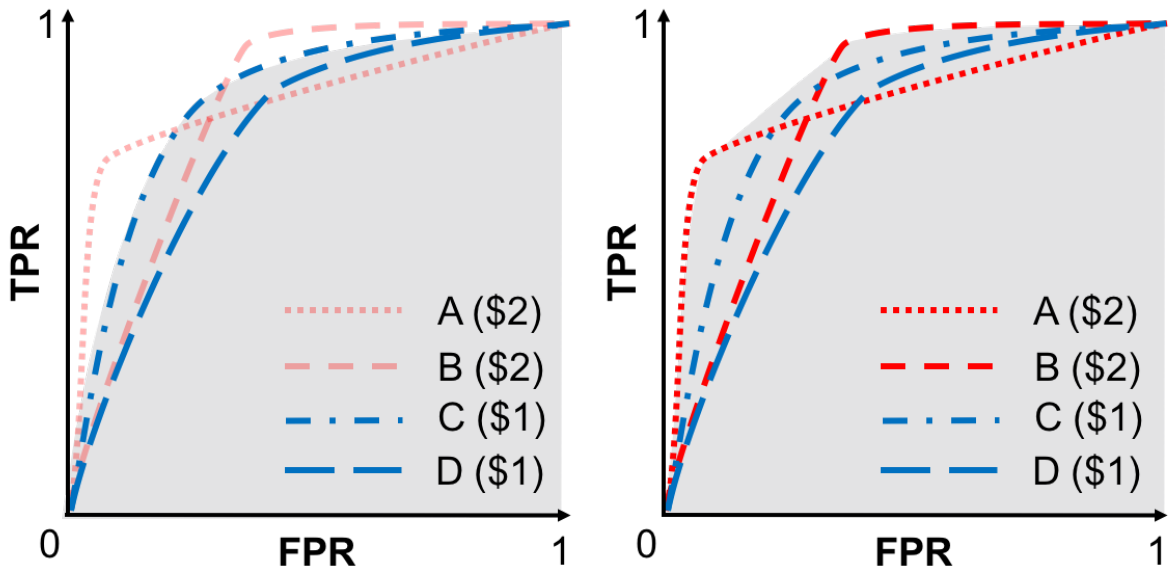


Figure 8: The ROC Convex Hull with Cost (ROCCHC) method.

number of cutoff thresholds, k. The algorithm outputs the index, i, of each classifier in the Pareto set. The algorithm goes to each unique C_{A} and calculates a ROC convex hull of all classifiers with C_{A_i} less than or equal to the current unique C_{A}. A classifier i is Pareto optimal if its C_{A_i} equals the current unique C_{A} and its r_i is on the convex hull, or it's not dominated in terms of performance and cost.

3.2.2 Sensitivity Analysis

Classifier selection using the ROCCH sensitivity analysis was discussed in Chapter 2.4.2, however, this sensitivity analysis doesn't include the additional low cost objective. To expand this sensitivity analysis to include the low cost objective, the ROCCHC sensitivity analysis is presented.

The former ROCCH sensitivity analysis is based upon the range of uncertainty in slope S for which a classifier would be selected. More sensitive classifiers would only be selected for a narrow range in slope S while less sensitive classifiers would be selected for a wider range in slope S.

This sensitivity analysis is expanded by using classifiers selected from the ROCCHC

Input: additional costs: $C_{\mathrm{A}_1}, \ldots, C_{\mathrm{A}_m} \in \mathbb{R}$, ROC curves with k cutoff thresholds: $r_1, \ldots, r_m$, where each $r_i \in \mathbb{R}^{2\times k}$ consists of all FPRs $\in \mathbb{R}^{1\times k}$ and TPRs $\in \mathbb{R}^{1\times k}$ of the i^{th} classifier
Output: α: indices of selected classifiers
$\alpha \leftarrow$ empty list
$\Gamma \leftarrow$ unique C_{A} sorted in ascending order
for u = *1* **to** *length(*Γ*)* **do**
  $R \leftarrow$ All r_i corresponding to $C_{\mathrm{A}_i} \leq C_{\mathrm{A}_u}$
  $H \leftarrow \mathrm{Conv}(R)$; where $\mathrm{Conv}(\bullet)$ returns i for each classifier r_i in the ROC convex hull
  for i = *1* **to** m **do**
    if $C_{A_i} = \Gamma_u$ ***and*** $i \in H$ **then**
      Append i to α
  end
end

Algorithm 1: ROC Convex Hull with Cost

method. Since the ROCCHC method computes several convex hulls, a sensitivity analysis in slope S can be conducted at each convex hull. This is shown in Figure 9 for two classifiers, A, and B (with $\mathrm{C_A}$ in parentheses). Both of these classifiers are selected using the ROCCHC method or both are Pareto optimal in terms of performance and cost. The left ROC graph shows the convex hull computed at $\mathrm{C_A}$ = \$1 as the shaded region, showing the selection of classifier A. The middle ROC graph shows this convex hull and the convex hull computed at $\mathrm{C_A}$ = \$2, showing the selection of classifier B. The arrows on left and middle graphs show the sensitivity of selecting classifier A based on each convex hull. The left shows a wide range in slope S or a low sensitivity, while the right shows a narrow range or a high sensitivity. The question is how to combine these sensitivities or ranges in slope S into one measure of sensitivity? Since classifier A is selected using each convex hull, this combination of sensitivities should reward classifier A. Classifiers that are on multiple convex hulls are less sensitive to slope S as compared to selected classifiers of differing costs. The right ROC graph shows the sensitivity of classifier B.

Ideally, you could obtain the angle between the range in slope S that selects a classifier to determine sensitivity, however, a method for accomplishing this hasn't been developed. To numerically determine sensitivity, a metric was created as the sum of a classifier's operating points (FPR,TPR) on all convex hulls. The more operating points on the

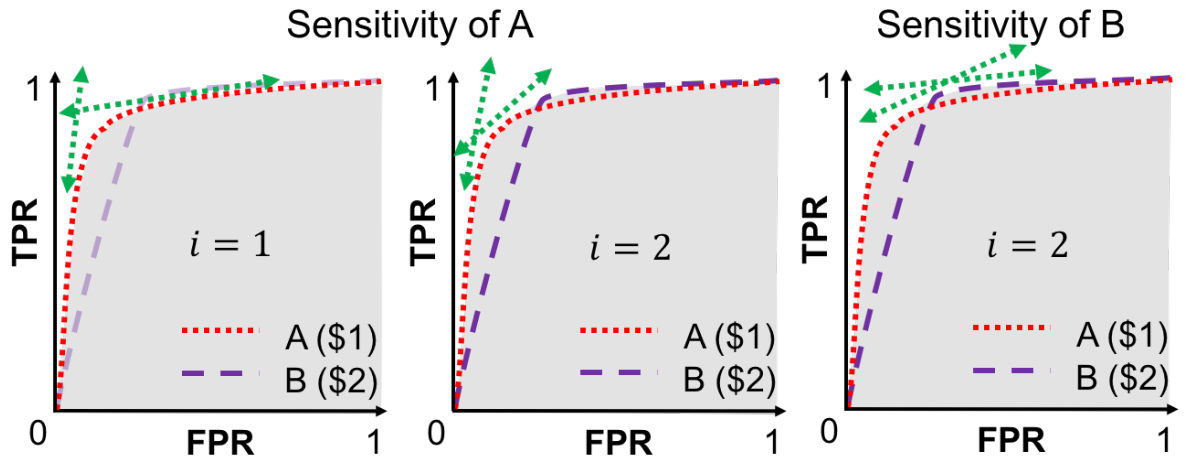


Figure 9: Sensitivity analysis with the ROCCHC method.

convex hulls, the larger the range in S for which the classifier is optimal, or the less sensitive the classifier is to real world uncertainty. This sensitivity metric, β, is calculated as

$$\beta = \sum_{i=1}^{\gamma} p_i, \tag{5}$$

where γ is the number of convex hulls or the number of unique C_A and p_i is the number of points on the i_{th} convex hull.

Higher β's mean less sensitive classifiers, while lower β's mean more sensitive classifiers. This metric adds an additional objective of low sensitivity to real world uncertainty for classifier selection. A set of Pareto optimal classifiers would be selected using the ROCCHC method to account for the high performance and low cost objectives and then compared using β to aid in selecting classifiers using a low sensitivity objective.

3.2.3 Dealing with Confidence Intervals

ROC confidence intervals are used to show ROC variance and are discussed in detail in Chapter 2.4.3. The idea is that there's an area of ROC space, instead of a line or ROC curve, where a classifier could operate at a certain confidence level. Since classifiers can only be compared when there's a measure of variance, we need to include these confidence

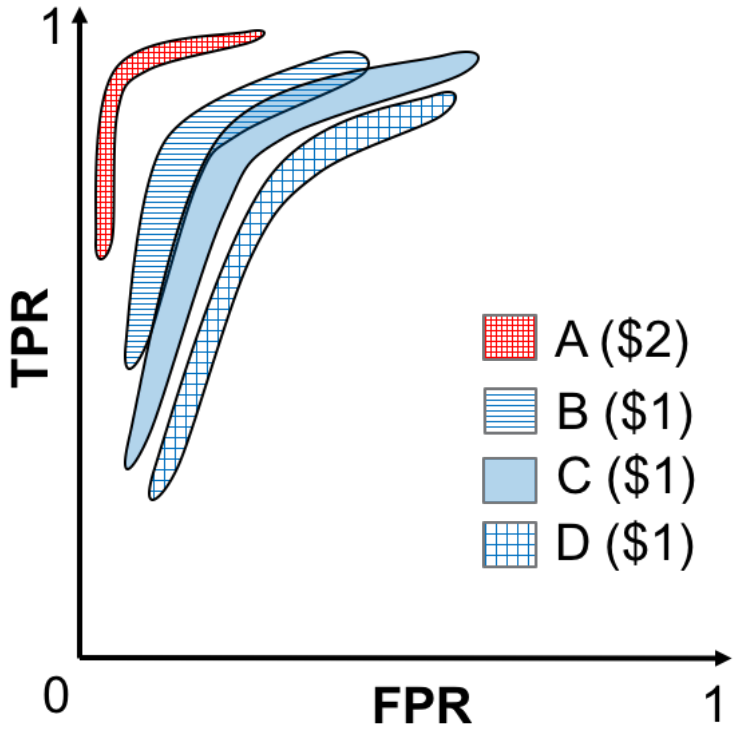


Figure 10: The ROCCHC method with a confidence level.

intervals in this analysis.

The ROCCHC selection method can include these ROC confidence intervals. This would allow for classifier selection at a certain level of confidence, such as there's a 99% confidence that a set of classifiers is the Pareto set. The selected classifiers would need to clearly dominate the others in terms of ROC performance or the ROC bands wouldn't overlap classifiers not in the Pareto set. It's expected that as you raise the confidence level for any problem, the set of Pareto optimal classifiers would grow. This is because as the confidence level is increased, the ROC bands would grow and eventually overlap classifier ROC bands not in the former Pareto set. Therefore, these overlapped classifiers would no longer be dominated in terms of performance and may be added to form a new Pareto set.

The ROCCHC method with a confidence level is demonstrated in Figure 10. This figure shows ROC bands for classifier A, B, C, and D with their C_A shown in parenthesis. The ROCCHC method would select classifiers A, B, and C and not classifier D. Classifier

D is dominated by classifiers B and C because they're at an equal C_A, yet B and C have a better performance band than classifier D. Classifier C is selected because part of its ROC band overlaps with classifier B's band. These are example ROC bands, however, real bands would be generated for a certain confidence level. Therefore, the Pareto set of classifiers could be presented to stakeholders with a certain level of confidence.

Table 2: Comparison of Exhaustive vs Search Methods for Classifier Generation

Method	**Implementation**	**Computation**	**Confidence**
Exhaustive	easier	more	higher
Search	harder	less	lower

3.3 Methodology for Generating Competing Classifiers

Real world classifier selection methods involve generating and selecting from various competing classifiers, where the competing classifiers would include different algorithms, feature sets, parameters, hardware, etc. Deciding on which classifiers to generate is a problem because we can only compare and select classifiers that we've generated. If the "best" classifier isn't generated, it will be missed and not selected. Two potential solutions to this generation problem are an exhaustive type method, where all classifiers are generated and compared, and a search type method, where classifiers are generated based on what has been learned from the previously generated classifiers. The exhaustive type method is the easiest to implement, however, it would require the most computational resources. The search type method would likely require less computational resources, however, you're less confident that the "best" classifier was found or generated. A comparison between these two generation methods is shown in Table 2.

These classifier generation methods may also require problem specific implementation based on the problem's objectives and constraints. For example, the actuator problem involves selecting a sensor set with a constraint that models must be interpretable. Therefore, classification tree models, which are interpreted easily, could be generated for each combination of sensors using the exhaustive method, while the search method would generate these models for specific sensor combinations based on what has been learned. Other problems may involve selecting between classification algorithms, algorithm specific hyperparameters, or feature sets.

In general, the exhaustive method should be implemented if the computational resources are available because its implementation is easier than developing a search method and there's a higher confidence in generating the "best" classifier. The search method

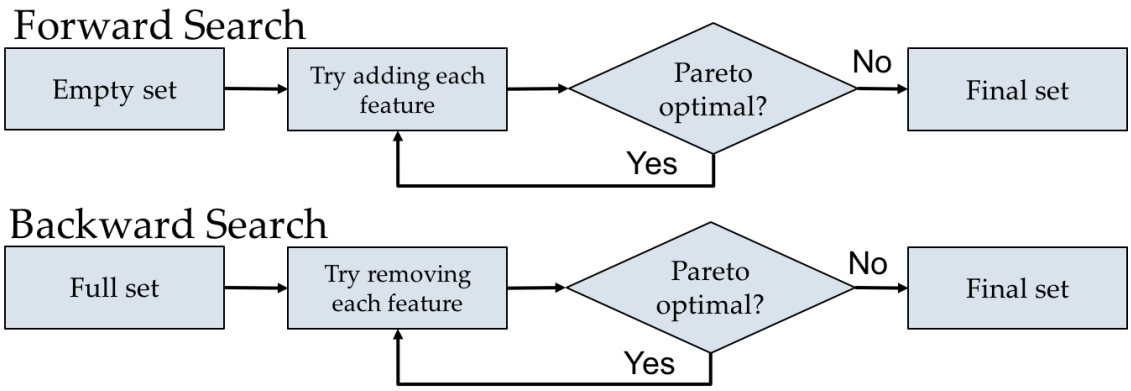


Figure 11: Sequential forward and backward search methods.

should be implemented if an exhaustive search is infeasible, such as if there are too many competing classifiers to be able to generate all of them within a reasonable amount of time.

The methodological demonstration includes both an exhaustive method and two search methods, sequential forward selection and sequential backward selection. Sequential forward selection and sequential backward selection are shown graphically in Figure 4.3. These search methods were used to generate classifiers with varying sensor or feature sets. The forward selection method starts from an empty set and adds each sensor or feature one at a time until the resulting classifiers are no longer part of the Pareto set. The backward selection method starts from a full set and removes each sensor or feature until the resulting classifiers are no longer part of the Pareto set.

To better illustrate how these selection methods were implemented with the ROCCHC method, Figure 12 shows a detailed example search web using the sequential forward selection method. This figure shows a search for a problem that has four possible features. Each circle corresponds to a different classifier consisting of a different feature set, where the feature set is represented using a bit string. The string shows a "0" for features not used in the classifier and a "1" for features used in the classifier. Since this example has four features, there are 15 possible feature sets or classifiers ($2^4 - 1$, the empty feature set isn't counted as a classifier). The forward search web starts with an empty feature set on

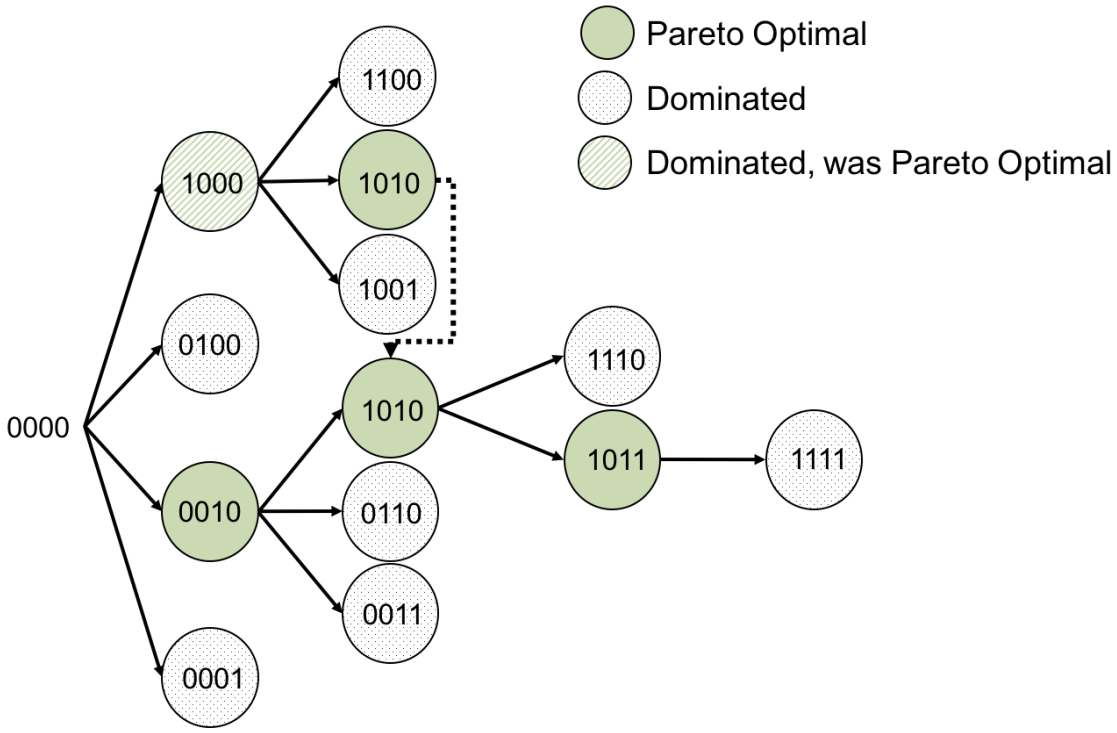


Figure 12: Sequential forward search with the ROCCHC method.

the left and generates four new classifiers, each with one feature. These new classifiers are then tested to determine the Pareto set using the ROCCHC methodology. Classifiers that are dominated are shown using a dotted pattern and the Pareto optimal classifiers are shaded. This process is continued for classifiers in the Pareto set until none of the new classifiers are in the Pareto set.

This figure also shows a problem unique to search methods. That classifiers once part of the Pareto set can later be dominated by newly generated classifiers. This idea is shown using the "1000" classifier, which has a diagonal line pattern. This classifier was part of the Pareto set, however, isn't part of the final Pareto set. Also, search methods could generate the same classifiers multiple times. The "1010" classifier in this example is shown twice, with a dashed arrow between the two. To avoid evaluating the same classifier more than once, the developed search algorithms check to see if each new classifier has already been generated to save computational resources. The dotted line shows that the "1010" classifier is only evaluated once.

The three classifier generation methods, exhaustive, sequential forward selection, and

sequential backward selection, are demonstrated in Chapter 4.3 using the three data sets. The search methods are also compared using the ground truth Pareto set resulting from the exhaustive method.

3.4 Chapter Summary

This chapter presented a multiple objective classifier selection methodology and a classifier generation methodology. The selection methodology expanded on the current ROC analysis by introducing an objective of low cost to the ROC Convex Hull method. This new selection methodology also incorporates a sensitivity analysis and ROC confidence intervals. The classifier generation methodology discusses exhaustive and search methods and provides stakeholders advice on which should be implemented for a given problem. These classifier generation methods are also expanded for use with the multiple objective classifier selection methodology.

4 Demonstration of Method

4.1 Chapter Overview

These chapter demonstrates the presented methodologies using three real world binary classification data sets, each of which include additional cost information. These data sets include a data set from the actuator problem mentioned in Chapter 1.1, as well as two data sets from the UCI Machine Learning Repository [9]. These are the Pima Indians Diabetes data set and the Hepatitis data set.

The classifier selection and generation of competing classifier methodologies were used to analyze these three data sets. An ROC analysis is justified for these problems because the real world costs of misclassification and the real world class distribution are unknown for each. Analysis with the presented methodologies is justified because these data sets include additional cost information, which the current ROC analysis ignores. This additional cost information is data set specific and includes costs for each feature or sensor.

This additional cost information also includes feature groupings with feature group discounts for each data set. These group discounts are applied when groups of features share a common cost. For the health care examples, the features from two separate blood tests may have a common cost if the same sample of blood can be used for both tests, resulting in a savings or discount in supplies (i.e. the needle and vial) and personnel (i.e. the nurse). The total additional cost of a classifier that includes these two features would be discounted by the common cost. These feature group discounts were accounted for while computing a classifier's additional cost, C_A, by using a developed cost discount function. This simple function takes in a classifier's used feature names and returns the corresponding discount.

Table 3 shows the three data sets including the number of features and observations. The additional costs for each feature, the feature groups, and feature group common costs are specified for the Pima Indians Diabetes and Hepatitis data sets on the UCI Machine Learning Repository. The feature costs for the Actuator data set are the actual purchase

Table 3: Binary data sets with real cost information

Data set	**# of features**	**# of observations**
Actuator	56 (8 sensors)	2,340
Pima	8	768
Hepatitis	19	155

price of the sensor from which each feature is derived. The feature groups correspond to each feature's required sensor and the common costs are each sensor's actual purchase price. For example, the C_A of a classifier that includes two features from one sensor will be the purchase price of the sensor and not twice this amount.

The methodology for generating competing classifiers mentions a comparison between exhaustive and search methods. Deciding on which method to use for a given problem depends on the available resources for implementation (i.e. the number of graduate students available), the available computational resources, and the desired confidence that all Pareto optimal classifiers are found. To demonstrate these concepts, both, an exhaustive method and two search methods, sequential forward and sequential backward selection, were implemented. To easily compare these three methods, the same classifiers were created, including the same classification algorithm, hyperparameters and training/testing sets. Since an exhaustive method was implemented in this analysis, its resulting Pareto set is the ground truth and the two search methods try and find all of these classifiers.

4.2 Exhaustive Method Demonstration

An exhaustive list of competing classifiers was created for each data set. These classifiers were built using each combination of features for the Diabetes and Hepatitis data sets and each combination of sensors for the Actuator data set. The resulting number of classifiers can be determined using $2^b - 1$, where b is the number of features or sensors in the data set. Each classifier was created from the feature set using the CART algorithm [23] in MATLAB with default hyperparameters and validated using 10-fold cross validation. The classification tree algorithm was used because of its low training time, as compared to

the random forest algorithm, and because the resulting models are easily interpretable, which is a stakeholder requirement for the actuator problem. Cross-validation was used to better judge the performance of each classifier [24].

The exhaustive generation process is shown in Algorithm 2. This algorithm shows how m classifiers were generated using k-fold cross validation. The ROC curves, r, were saved for each classifier, along with other classifier information like the classifier's sensor or feature set, the classifier's C_A, and a unique classifier identification number. The unique identification number was created by converting a bit string, which represents the classifier's used features or sensors (i.e. 011 corresponds to selecting the last two features when there are three available), into a number. These unique identification numbers were used to quickly store and access any classifier.

An exhaustive list of classifiers was generated for each data set and then analyzed using the multiple objective selection methodology. The ROCCHC method was used to obtain the Pareto set of all generated classifiers using Algorithm 1. These Pareto optimal classifiers were then compared to the ROCCH selected classifiers for each data set. The Pareto optimal classifiers aren't compared to other methods that use accuracy because these analyses assume that the real world costs of misclassification and real world class distribution are known. The classifier selection results for each data set are presented in the following sections.

4.2.1 Actuator data set

The presented classifier selection methods are first demonstrated on the Actuator data set using the exhaustive classifier generation method. The classification task for this data set is to detect faults in the hydraulic rotary actuator given sensor data streams. A goal for this problem is to determine the "best" sensor combination that performs this classification task. The presented multiple objective selection methodology applies to this data set because the problem includes objectives of high performance, low cost, and low sensitivity. The current additional cost information for this data set includes sensor

Obtain the number of classifiers, m, using $m = 2^b - 1$
Generate k training sets and k testing using k-fold cross validation
for *i = 1:m* **do**
  Get the classifier's cost, C_{A_i} based on the combination of features or sensors and feature group discounts
  Assign the classifier a unique identification number using the combination of features or sensors
  for *j = 1:k* **do**
    Build a classifier using the corresponding feature subset and the j_{th} training set
    Test the classifier using the j_{th} testing set and save each test observation's posterior probability
  end
  Generate the classifier's ROC curve, r_i, using the observation posterior probabilities
end

Algorithm 2: Exhaustive generation of competing classifiers

Table 4: Sensor prices for the Actuator data set

Accelerometer (Accel#)	**Angular Position (Angle)**	**Flow Rate (FlowOut#)**	**Pressure (PG#)**
\$35	\$40	\$3,245	\$429

purchase prices, however, future work will include other operating costs, such as sensor energy usage.

The eight sensors in this data set include an angular position sensor, two flow rate sensors, two pressure sensors, and three accelerometers. The data set includes seven features per sensor, which were calculated during each actuation cycle, and include mean, variance, standard deviation, skewness, kurtosis, minimum, and maximum. This results in 56 (7 feature functions $\times$ 8 sensors) features per observation, with no missing data. The current objective is sensor selection, therefore, instead of building classifiers for every combination of features (resulting in generating an infeasible 7.2×10^{16} ($2^{56}-1$) classifiers), classification models were built using every combination of the eight sensors, amounting to a manageable 255 ($2^8 - 1$) classifiers. The additional cost of each of these classifiers is the total purchase price of the classifier's sensor set, where the actual sensor purchase prices are shown in Table 4.

Figure 13 shows a ROC graph with the exhaustive set of 255 classifiers, with the color

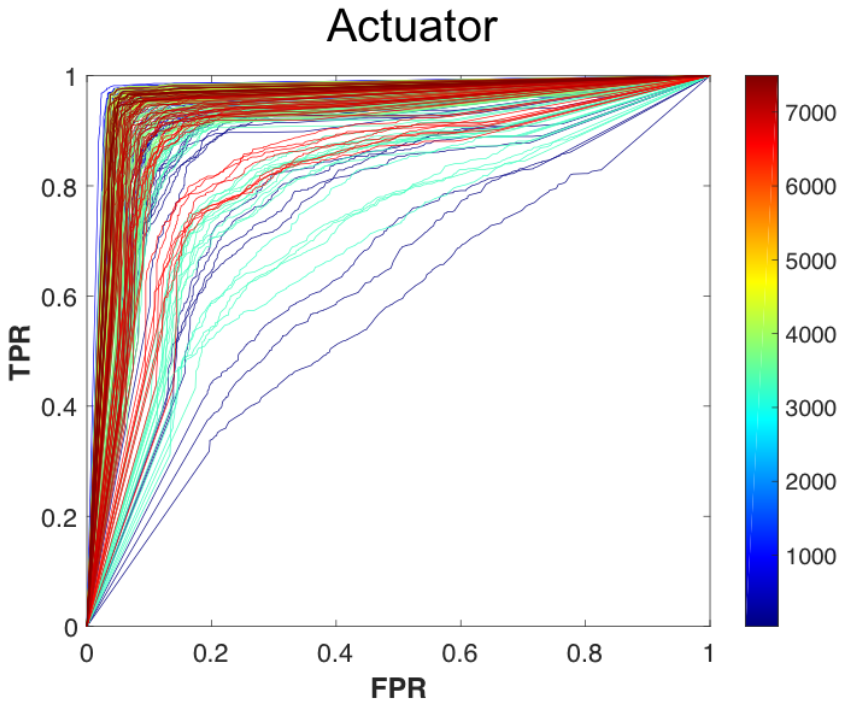


Figure 13: ROC graph of the exhaustive set for the Actuator data set.

corresponding to the sensor set cost based on the color bar. This figure shows the initial decision space before using the presented selection methodology. This figure shows that it would be very difficult for stakeholders to make decisions from this exhaustive set of alternatives and demonstrates why a selection method and algorithm is needed.

The ROCCHC selection method reduces the set of 255 competing classifiers to a set of only 11 Pareto optimal classifiers, a 96% reduction (1-11/255). These 11 classifiers are shown in Figure 14, which includes a 3-D ROC graph with a third axis of sensor set cost, on the left and a corresponding ROC graph on the right, which includes a legend with each classifier's sensor set cost and the classifier's sensor set. As expected, the higher cost classifiers in the Pareto set outperform the lower cost classifiers.

The set of classifiers resulting from the ROCCH method were also computed for a comparison. The ROCCH method selected only one classifier, the "$868: PG1, PG2" classifier. This classifier is shown in Figure 14 as the classifier with the best ROC performance, however, this classifier also has the highest sensor set cost. The ROCCH method selects one of the 11 Pareto optimal classifiers, therefore, ignoring the additional cost

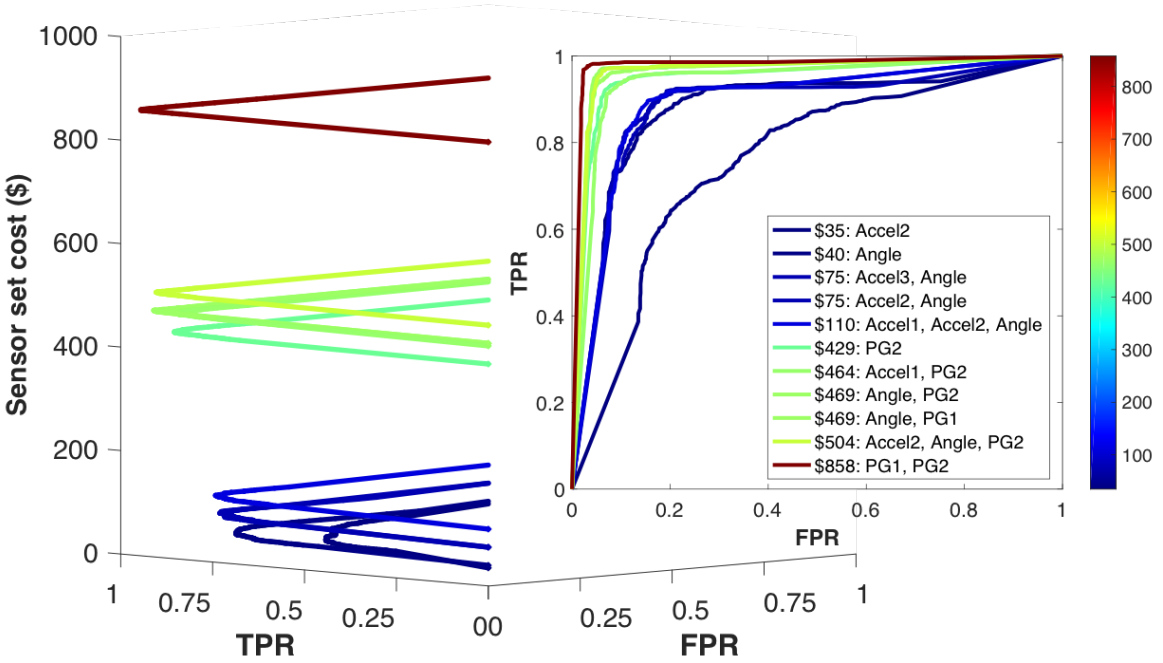


Figure 14: Pareto set for the Actuator data set.

(sensor set cost in this case) by using the ROCCH method would miss 10 Pareto optimal classifiers, or 91% (10/11).

The ROCCHC method drastically reduces the decision space, however, doesn't select one classifier as the "best" but a set of Pareto optimal classifiers. To select a final classifier, you would need a tradeoff between the performance and cost objectives or additional objectives.

The sensitivity analysis can be used to further reduce the decision space. This analysis tries to quantify the range in slope S for which a classifier would be selected. The wider this range, the less sensitive the classifier to uncertainty. The metric for this sensitivity analysis, β, is the summation of all of the classifier's operating points on each convex hull from Equation 5. Figure 15 shows an ROC graph of the Pareto set, with the metric β in parenthesis in the legend. The ROC graph shows operating points for each classifier that were on a convex hull. Four of the classifiers, "Accel3, Angle", "Accel1, Accel2, Angle", "Accel1, PG2", and "Accel2, Angle, PG2" have a $\beta \leq 4$. These classifier are highlighted in the figure using the arrows. These classifiers are very sensitive to slope

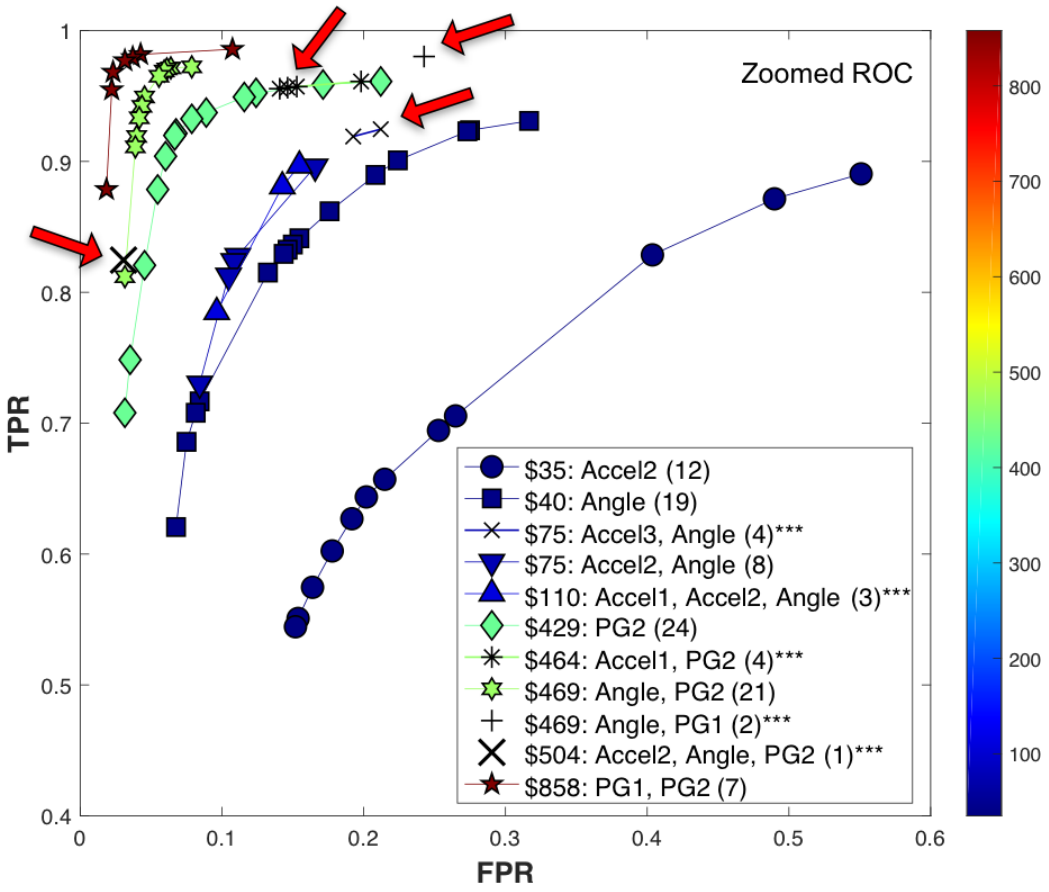


Figure 15: Actuator data set sensitivity results.

S and could be removed from the selection set presented to stakeholders, resulting in only seven classifiers. To further reduce these seven classifiers, additional objectives such as low energy consumption could be analyzed, a tradeoff between performance and cost could be formulated, or a maximum cost constraint could be decided.

4.2.2 Pima Indians Diabetes Data Set

The classifier selection methods are demonstrated on the Pima Indians Diabetes data set using the exhaustive classifier generation method. The classification task for this data set is to detect diabetes in a patient using various health observations, measures, and tests. This data set has eight features, which consist of health measures, observations, and bodily tests, each of which includes real cost information.

The exhaustive method was used to generate classifiers, with each classifier having a different feature set. This results in a set of 255 ($2^8 - 1$) competing classifiers. Figure 16 shows a ROC graph with this exhaustive set of classifiers, with the color corresponding to the feature set cost based on the color bar. This figure shows the initial decision space for this problem and demonstrates that it would be very difficult for stakeholders to make decisions from this exhaustive set.

Analyzing this set with the ROCCHC selection method results in a set of 23 Pareto optimal classifiers, reducing the decision space by 91% (1-23/255). An ROC graph with these classifiers is shown in Figure 17. This figure shows a 3-D ROC graph, with a third axis of feature set cost along with a corresponding ROC graph on the right. Again, as expected, there is a trend that the higher cost classifiers have a better ROC performance than lower cost classifiers.

The set of Pareto optimal classifiers is compared to the set resulting from the ROCCH method. The ROCCH method selects seven classifiers, therefore, ignoring the test cost by using the ROCCH method would miss 16 Pareto optimal classifiers, which is 70%.

The sensitivity analysis was conducted on the Pima data set in less detail than the actuator problem. The sensitivity results are shown in Figure 18, which shows a histogram

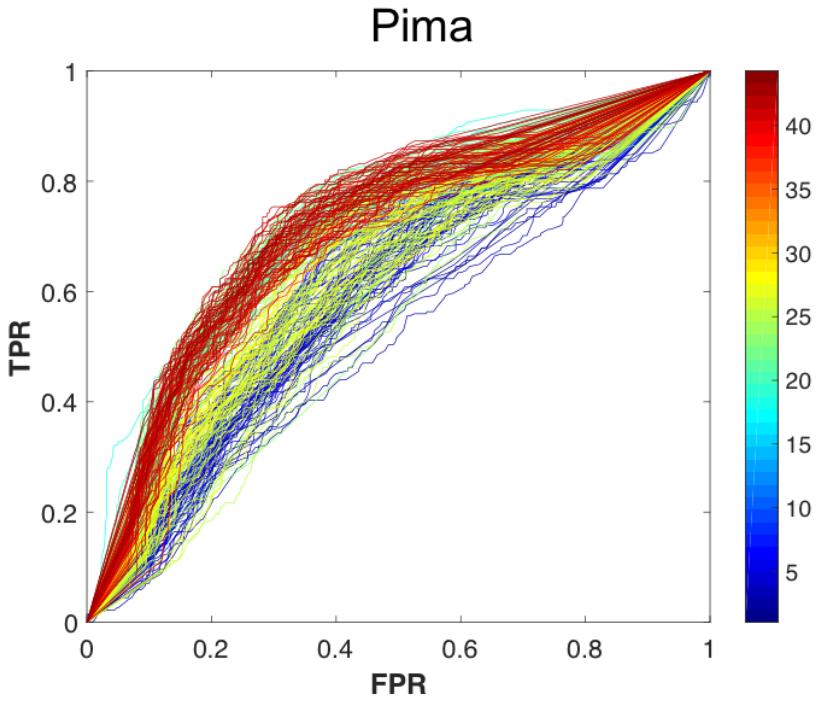


Figure 16: ROC graph of the exhaustive set for the Pima data set.

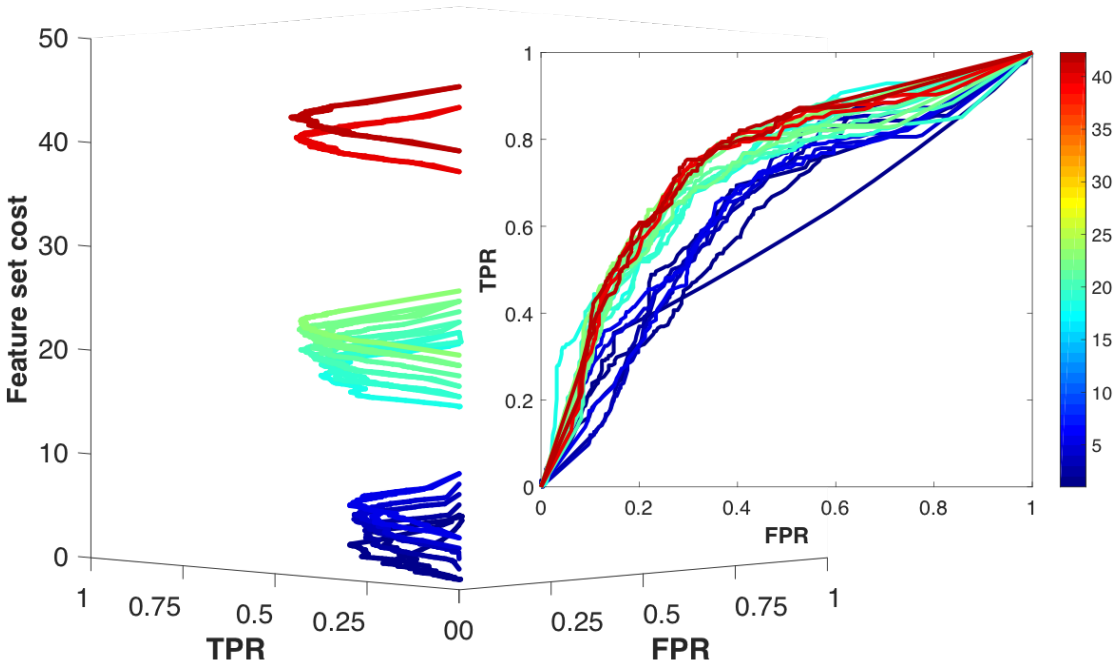


Figure 17: Pareto set for the Pima data set.

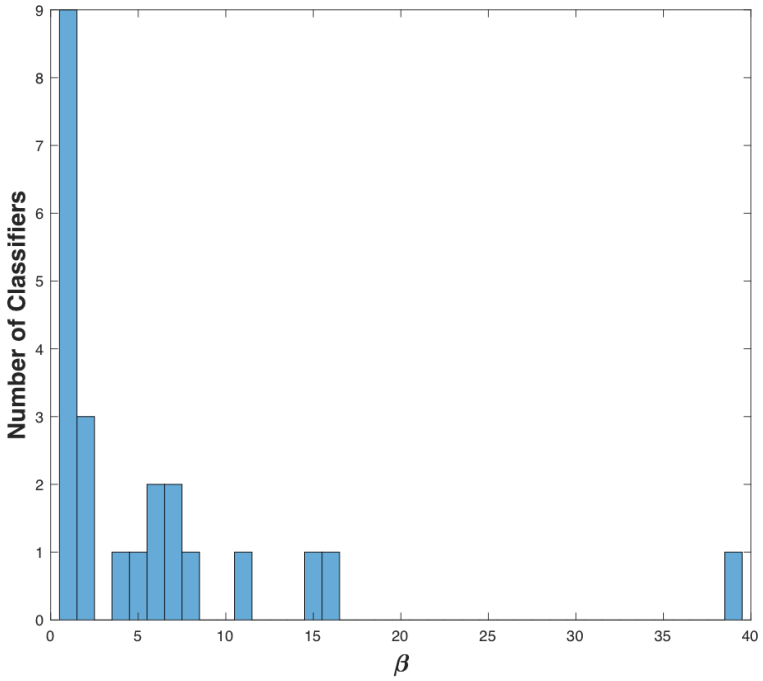


Figure 18: Pima data set sensitivity results.

of the Pareto set's β values. This histogram shows that nine Pareto optimal classifiers are very sensitive with only one operating point on the ROC convex hulls. These nine classifiers could be removed from the set presented to stakeholders, resulting in a set of 14 classifiers.

4.2.3 Hepatitis Dataset

The classifier selection methods are finally demonstrated on the Hepatitis data set. The classification task for this problem is to correctly classify a patient as having hepatitis or as not having hepatitis. This data set has 19 features, which consist of health measures, observations, and bodily tests, each of which includes real cost information. The exhaustive method was used to generate competing classifiers using each possible combination of features, resulting in 524,287 ($2^{19} - 1$) classifiers.

The ROCCHC method results in a set of 57 Pareto optimal classifiers, reducing the decision space by 99.99% (1-57/524,287). This demonstrates that this classifier selection

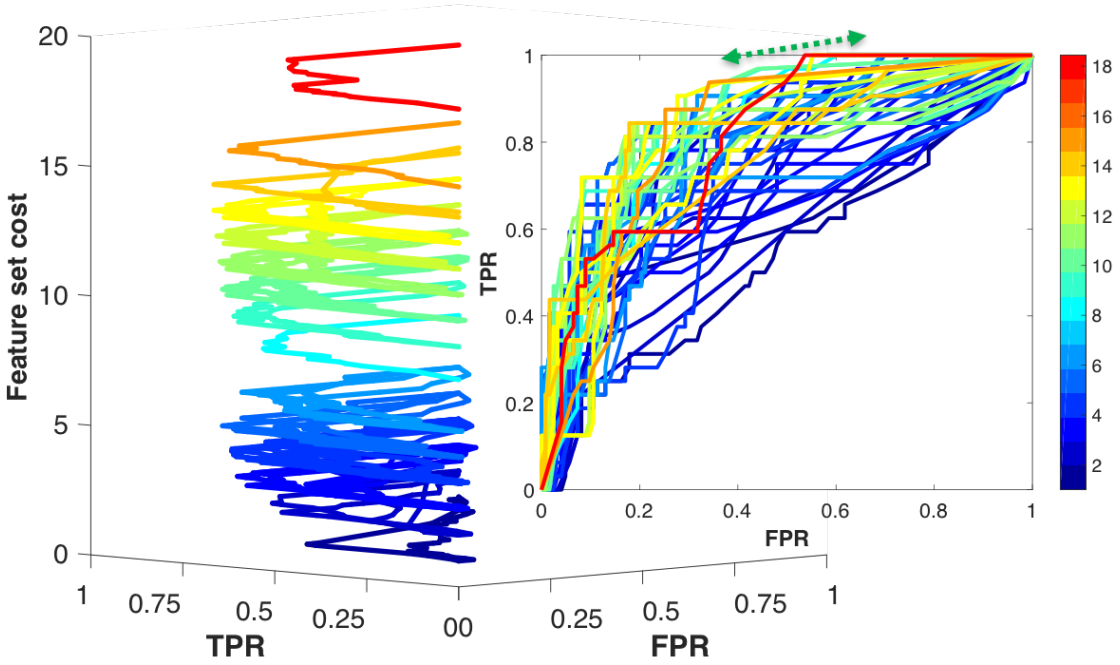


Figure 19: Pareto set for the Hepatitis data set.

method can drastically reduce the decision space even when there's a very large number of competing classifiers. An ROC graph consisting of this Pareto set of classifiers is shown in Figure 19, which shows a 3-D ROC graph, with a third axis of feature set cost along with the corresponding ROC graph on the right.

The ROCCH method selects only eight classifiers, which are shown in Figure 20. Therefore, ignoring feature set cost by using the ROCCH method would miss 49 Pareto optimal classifiers, which is 84% of the optimal set.

To illustrate the need for a sensitivity analysis, the ROC graph in Figure 19 includes a line at a slope S. This shows the S where the most expensive classifier (the red one), would be selected as ideal over a less costly classifier. This means that the most expensive classifier is also very sensitive to uncertainty.

The Hepatitis data sets sensitivity results are shown in Figure 21, which shows a histogram of the Pareto set's β values. This histogram shows that 37 Pareto optimal classifiers are very sensitive with only one operating point on the ROC convex hulls. These 37 classifiers could be removed from the set presented to stakeholders, resulting in

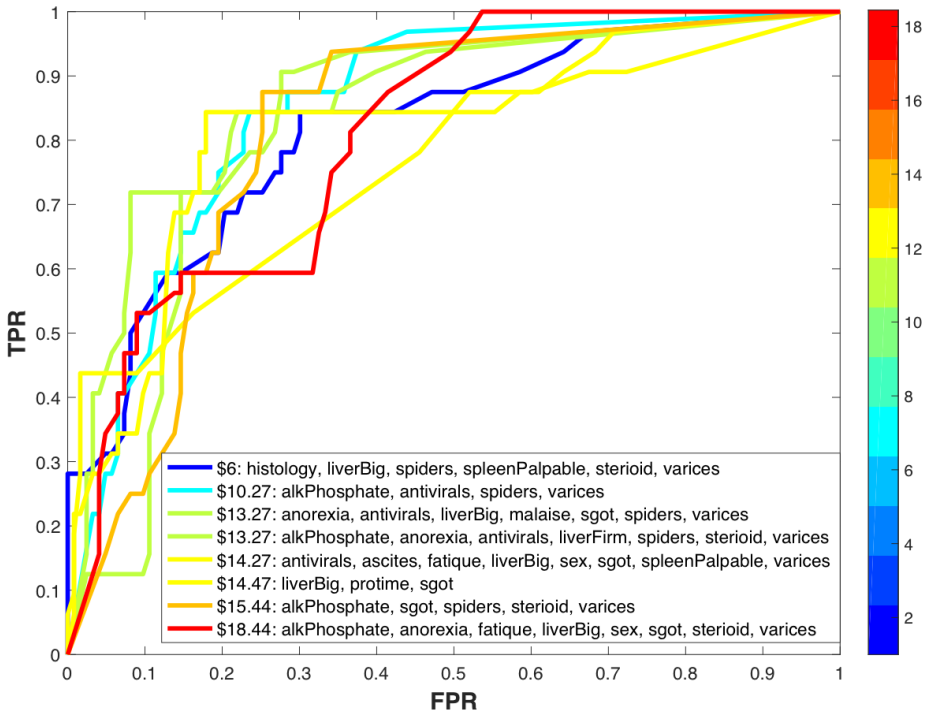


Figure 20: ROCCH set for the Hepatitis data set.

a set of 20 classifiers. This is a very drastic reduction from 524,287 classifiers.

4.3 Search Method Demonstration

There may be real world problems where an exhaustive classifier generation couldn't be completed due to limited computational resources and a high number of features or problems where a selection decision is urgent. For these problems, it may be best to implement a search strategy. A search strategy learns from previously generated classifiers in an attempt to find better classifiers and is quicker and less computationally expensive than a exhaustive search.

To demonstrate that the presented multiple objective classifier selection methodology can be used with a search method, sequential forward and backward selection were implemented with the ROCCHC methodology. These sequential searches were implemented in MATLAB. These two searches are compared by using the exhaustive method's set of

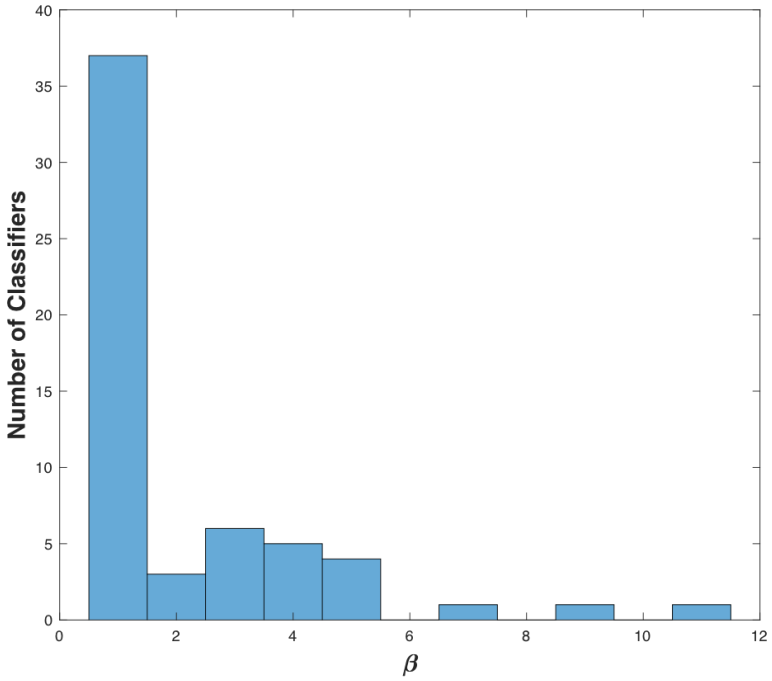


Figure 21: Hepatitis data set sensitivity results.

Pareto optimal classifiers, which is the ground truth set because it includes all competing classifiers. The search results are shown in Figure 22, which shows the x-axis as the proportion of classifiers evaluated based on the number of exhaustive classifiers, and the y-axis as the proportion of exhaustive method's Pareto set that the search has found. Ideally, a search method would have a steep slope that starts on the left side of the figure and reaches the very top, indicating that the search method found all of the Pareto optimal classifiers very quickly.

This figure shows the search results for the three data sets using both sequential searches, forward and backward. For the Actuator data set, both searches find all of Pareto optimal classifiers, with the forward search evaluating less classifiers. The forward search evaluates 29% while the backward search evaluates 67% of the 255 competing classifiers. The forward search was able to find all of the Pareto optimal classifiers faster than the backward search because the set of Pareto optimal classifiers includes three or fewer sensors and the backward search starts with all eight sensors.

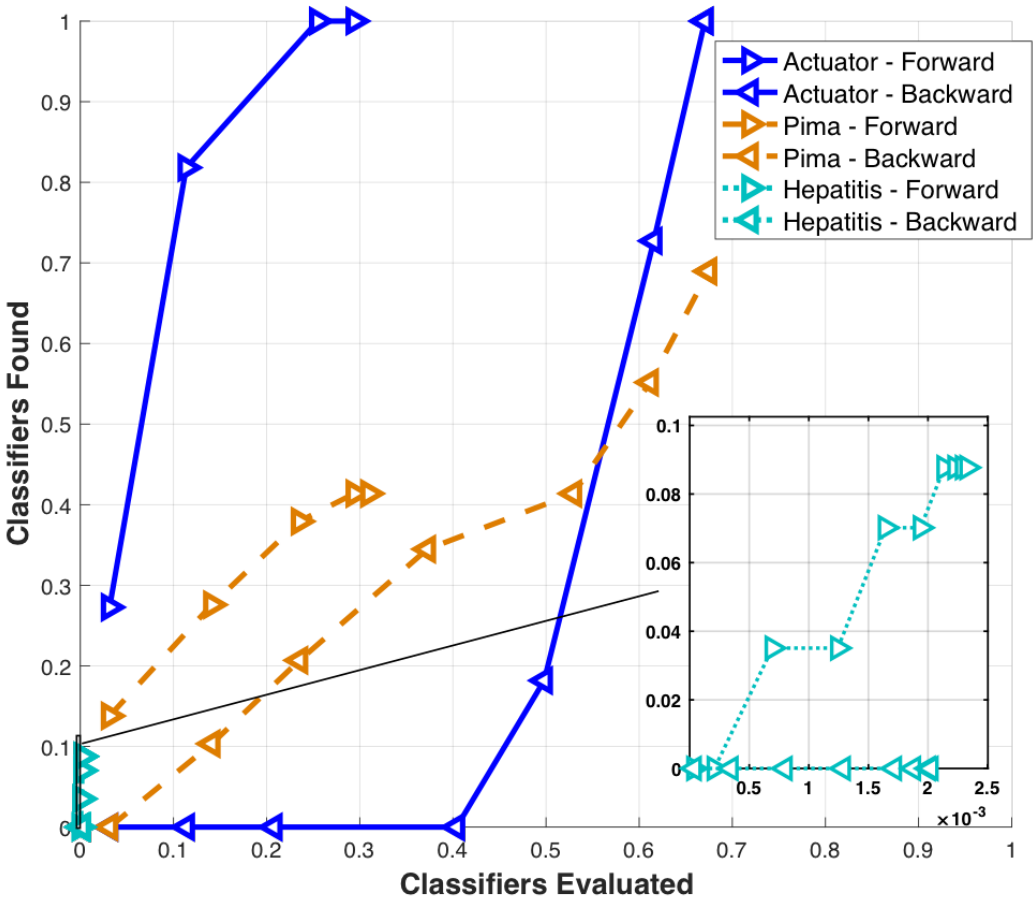


Figure 22: Sequential forward and backward search results.

For the Pima data set, the forward search finds 41% while the backward search finds 68% of the Pareto optimal set. The backward search finds more, however, it evaluates more than twice as many classifiers as the forward search. It's hard to determine which search method is better in this case as there's likely a cost to evaluating more classifiers in an attempt to find more Pareto optimal classifiers.

The forward and backward search results aren't as good for the Hepatitis data set. Both searches stopped after only evaluating about 0.2% of the total competing classifiers and only find 9% for the forward search and 0% for the backward search. The search space of this data set is very large (524,287 classifiers) and these sequential search methods aren't able to cover a large enough area of this space. These search results would likely be improved for this large data set if the searches started from many different randomized starting points. This may improve the performance by covering more of the search space, however, there is still a real world cost to evaluating more classifiers using this type of multi-start search process.

These search method results were presented to demonstrate that the ROCCHC method can be used with a search method. This is very important for real world problems where an exhaustive search wouldn't be possible. The forward and backward searches were conducted because they're easy to understand and implement, however, more advanced search methods may return better results and are left for future work.

4.4 Confidence Intervals

ROC curves can really only be compared when there's a measure of variance. This variance can be obtained by creating multiple ROC curves for a classifier and then computing ROC confidence intervals using methods discussed in section 2.4.3. The ROCCHC optimal classifiers can then be obtained at a certain confidence level.

To demonstrate that the ROCCHC classifier selection methodology can be used with ROC confidence intervals, ROC bands were generated for all competing classifiers using various levels of confidence for the Actuator and Pima data sets. These ROC bands

were then used when determining the set of Pareto optimal classifiers using methods in Chapter 3.2.3. These confidence interval results are shown in Figure 23, which shows the proportion of Pareto optimal classifiers to the total number of competing classifiers versus the level of confidence for the Pima and Actuator data sets. This figure shows a general trend that as the confidence level increases, the proportion of selected classifiers also increases, as expected.

This figure shows a big difference in the proportion of Pareto optimal classifiers at the higher confidence levels between the data sets. The Pima data set results in over half of its classifiers as selected at a 99% confidence, while the Actuator data set results in only 13%. This may be due to a large variance in feature or sensor predictability for the Actuator data set versus the Pima data set. This figure also shows a slight drop in the proportion after the 0% confidence level. This may be because the 0% confidence level doesn't include ROC curve averaging while the 10% and higher confidence levels include averaging. This averaging may change the position of the ROC curves enough to reduce the number of selected classifiers.

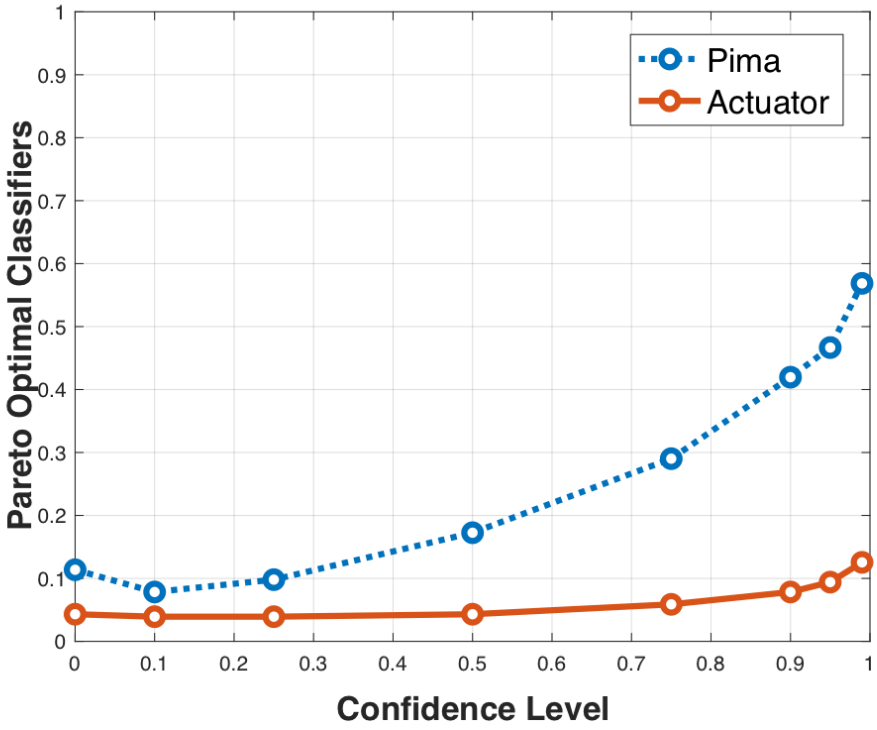


Figure 23: Pareto set versus the confidence level results.

5 Conclusions

5.1 Findings and Contributions

Through development of this multiple objective classifier selection methodology, it was shown that multiple objectives can be used to guide stakeholders through real world problems involving binary classifier selection. Current advances in machine learning are allowing engineering alternatives to be evaluated quickly and efficiently by relying on huge amounts of data. It will be interesting to see a futuristic society where machine learning methods quickly and automatically evaluate all alternatives for a given engineering problem and present stakeholders with the best solutions.

A major contribution of this thesis is that a method for performing a multiple objective analysis of competing classification models was presented that is robust to real world uncertainties. The presented methods expand on the current ROC analysis for situations where stakeholders have additional cost information about competing classifiers. This

analysis brings feature selection methods and the ROC analysis together for the first time, making feature selection methods more useful for real world problems.

Another contribution of this thesis is that the forward and backward search algorithm's performance evaluation is updated to include Pareto optimality using the ROC analysis. This is the first time the ROC curve has been used as an evaluation for a search method. This makes these search methods more useful for real world problems.

The demonstration of this method on three data sets concludes that a stakeholder's decision space can be quickly and efficiently reduced even with the multiple objectives of high performance, low cost, and low sensitivity. The results show the ability of this method to reduce the decision space by over 91% for the three data sets. The search results demonstrate that this new multiple objective selection method can be used with common search algorithms, allowing stakeholders to avoid an exhaustive search of alternatives. The confidence level results demonstrate this selection methodology can be applied using a certain level of confidence and even at a 99% confidence, the decision space can be reduced by over 40%.

5.2 Future Work

The methodology focuses on three objectives, however, real world problems may include additional objectives such as reliability and safety. Further work in classifier selection could involve adding these additional objectives. These methodologies would require additional information relating to the objective for each competing classifier. In this world of data, this additional information is likely out there, however, finding it presents an additional challenge. Finding additional information about each alternative and determining how to use it would be interesting future work.

The presented methodology is for binary classification problems, however, many real world problems involve multiple class classification problems. The goals and objectives of these problems would likely be the same, however, the methodology would have to be expanded. ROC analysis has been extended to multiple class problems in the literature.

The ROCCH method has been extended to multiple classes using multiple dimension convex hulls [25]. This means that the ROCCH method can be used to select classifiers in multiple class problems. Future work could be in expanding the ROCCHC method to multiple class problems in a similar manner.

This methodology includes two search methods, however, there are many more that remain to be tested with these selection methods. Randomized search algorithms such as genetic algorithms and particle swarm optimization algorithms have been favored recently in the literature [26, 18]. These algorithms are harder to implement than the sequential search algorithms but may provide better performance. Further work could be in developing a strategy on when to use certain search methods and implementing these search methods with the ROCCHC method.

References

[1] Edward A. Lee. Cyber Physical Systems: Design Challenges. In *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, pages 363–369. IEEE, 5 2008.

[2] Charles X. Ling and Victor S. Sheng. Cost-Sensitive Learning. In *Encyclopedia of Machine Learning*, pages 231–235. Springer US, Boston, MA, 2011.

[3] Peter D. Turney. Types of Cost in Inductive Concept Learning. In *Workshop on Cost-Sensitive Learning at the Seventeenth International Conference on Machine Learning*, 2000.

[4] Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, (27):861–874, 2006.

[5] Foster Provost and Tom Fawcett. Analysis and Visualization of Classifier Performance: Comparison under Imprecise Class and Cost Distributions. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, pages 43–48. AAAI, 1997.

[6] P. Beling, Z. Covaliu, and R. M. Oliver. Optimal scoring cutoff policies and efficient frontiers. *Journal of the Operational Research Society*, 56(9):1016–1029, 2005.

[7] Stephen Adams, Ryan Meekins, Peter A Beling, Kevin Farinholt, Nathan Brown, Sherwood Polter, and Qing Dong. A Comparison of Feature Selection and Feature Extraction Techniques for Condition Monitoring of a Hydraulic Actuator. In *Annual Conference of the Prognostics and Health Management Society*, St. Petersburg, FL., 2017.

[8] 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.

[9] M. Lichman. UCI Machine Learning Repository, 2013.

[10] Isabelle Guyon and André Elisseeff. An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, 3(Mar):1157–1182, 2003.

[11] V. Bolón-Canedo, I. Porto-Díaz, N. Sánchez-Maroño, and A. Alonso-Betanzos. A framework for cost-based feature selection. *Pattern Recognition*, 47(7):2481–2489, 2014.

[12] Stephen Adams, Ryan Meekins, and Peter A Beling. An Empirical Evaluation of Techniques for Feature Selection with Cost. In *International Conference on Data Mining (ICDM)*, New Orleans, LA, 2017.

[13] R Ganggang Kong, Liangxiao Jiang, and Chaoqun Li. Beyond accuracy: Learning selective Bayesian classifiers with minimal test cost. *Pattern Recognition Letters*, 80:165–171, 2016.

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

[15] Charles X. Ling, Qiang Yang, Jianning Wang, and Shichao Zhang. Decision trees with minimal costs. In *Twenty-first international conference on Machine learning - ICML '04*, page 69, New York, New York, USA, 2004. ACM Press.

[16] 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(C):1–11, 3 2016.

[17] Emrah Hancer, Bing Xue, Mengjie Zhang, Dervis Karaboga, and Bahriye Akay. Pareto front feature selection based on artificial bee colony optimization. *Information Sciences*, 422:462–479, 1 2018.

[18] Bing Xue, Mengjie Zhang, and Will N. Browne. Particle Swarm Optimization for Feature Selection in Classification: A Multi-Objective Approach. *IEEE Transactions on Cybernetics*, 43(6):1656–1671, 12 2013.

[19] Richard M. Everson and Jonathan E. Fieldsend. Multi-class ROC analysis from a multi-objective optimisation perspective. *Pattern Recognition Letters*, 27(8):918–927, 6 2006.

[20] Stephen G. Alsing, Erik P. Blasch, and Kenneth W. Bauer, Jr. Three-dimensional receiver operating characteristic (ROC) trajectory concepts for the evaluation of target recognition algorithms faced with the unknown target detection problem. volume 3718, pages 449–458. International Society for Optics and Photonics, 8 1999.

[21] Donald L Simon. A Three-Dimensional Receiver Operator Characteristic Surface Diagnostic Metric. In *Annual Conference of the Prognostics and Health Management Society*, 2010.

[22] Sofus A Macskassy and Foster Provost. Confidence Bands for ROC Curves: Methods and an Empirical Study. In *Proceedings of the 22nd international conference on Machine learning - ICML '05*, pages 537–544. Proceedings of the First Workshop on ROC Analysis in AI. August 2004., 2005.

[23] Leo. Breiman, Jerome Friedman, Richard Olshen, and Charles Stone. *Classification and regression trees*. Wadsworth, Belmont, CA, 1984.

[24] Trevor Hastie, Robert Tibsharani, and Jerome Friedman. The Elements of Statistical Learning. *The Mathematical Intelligencer*, 27(2):83–85, 2009.

[25] Ashwin Srinivasan. Note on the location of optimal classifiers in n-dimensional ROC space. Technical report, Oxford University Computing Library, Oxford, England, 1999.

[26] Chandra Sekhara Rao Annavarapu, Suresh Dara, and Haider Banka. Cancer microarray data feature selection using multi-objective binary particle swarm optimization algorithm. *EXCLI journal*, 15:460–473, 2016.