Summer 2009

Medical decision support systems based on machine learning

Chih-Lin Chi
University of Iowa

MEDICAL DECISION SUPPORT SYSTEMS BASED ON MACHINE LEARNING METHODS

by

Chih-Lin Chi

An Abstract

Of a thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy degree
in Informatics (Health Informatics)
in the Graduate College of
The University of Iowa

July 2009

Thesis Supervisor: Associate Professor W. Nick Street

ABSTRACT

This dissertation discusses three problems from different areas of medical research and their machine learning solutions. Each solution is a distinct type of decision support system. They show three common properties: personalized health care decision support, reduction of the use of medical resources, and improvement of outcomes.

The first decision support system assists individual hospital selection. This system can help a user make the best decision in terms of the combination of mortality, complication, and travel distance. Both machine learning and optimization techniques are utilized in this type of decision support system. Machine learning methods, such as Support Vector Machines, learn a decision function. Next, the function is transformed into an objective function and then optimization methods are used to find the values of decision variables to reach the desired outcome with the most confidence.

The second decision support system assists diagnostic decisions in a sequential decision-making setting by finding the most promising tests and suggesting a diagnosis. The system can speed up the diagnostic process, reduce overuse of medical tests, save costs, and improve the accuracy of diagnosis. In this study, the system finds the test most likely to confirm a diagnosis based on the pre-test probability computed from the patient's information including symptoms and the results of previous tests. If the patient's disease post-test probability is higher than the treatment threshold, a diagnostic decision will be made, and vice versa. Otherwise, the patient needs more tests to help make a decision. The system will then recommend the next optimal test and repeat the same process.

The third decision support system recommends the best lifestyle changes for an individual to lower the risk of cardiovascular disease (CVD). As in the hospital

recommendation system, machine learning and optimization are combined to capture the relationship between lifestyle and CVD, and then generate recommendations based on individual factors including preference and physical condition. The results demonstrate several recommendation strategies: a whole plan of lifestyle changes, a package of n lifestyle changes, and the compensatory plan (the plan that compensates for unwanted lifestyle changes or real-world limitations).

Abstract Approved: _____
Thesis Supervisor

_____
Title and Department

_____
Date

MEDICAL DECISION SUPPORT SYSTEMS BASED ON MACHINE LEARNING METHODS

by

Chih-Lin Chi

A thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy degree
in Informatics (Health Informatics)
in the Graduate College of
The University of Iowa

July 2009

Thesis Supervisor: Associate Professor W. Nick Street

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

PH.D. THESIS

This is to certify that the Ph.D. thesis of

Chih-Lin Chi

has been approved by the Examining Committee
for the thesis requirement for the Doctor of Philosophy
degree in Informatics (Health Informatics)
at the July 2009 graduation.

Thesis Committee: W. Nick Street, Thesis Supervisor

Jennifer G. Robinson

Connie W. Delaney

Marcia M. Ward

Padmini Srinivasan

Samuel A. Burer

Ying Zhang

To mom and dad

ACKNOWLEDGMENTS

I would like to thank Professor Nick Street for the great mentorship. I also want to thank the input from Professors Connie Delaney, Jennifer Robinson, David Katz, Marcia Ward, Padmini Srinivasan, Samuel Burer, and Ying Zhang.

I appreciate the support from my family. I especially like to thank my wife Tzu-Ying. I don't believe I can finish this dissertation without her support.

ABSTRACT

This dissertation discusses three problems from different areas of medical research and their machine learning solutions. Each solution is a distinct type of decision support system. They show three common properties: personalized health care decision support, reduction of the use of medical resources, and improvement of outcomes.

The first decision support system assists individual hospital selection. This system can help a user make the best decision in terms of the combination of mortality, complication, and travel distance. Both machine learning and optimization techniques are utilized in this type of decision support system. Machine learning methods, such as Support Vector Machines, learn a decision function. Next, the function is transformed into an objective function and then optimization methods are used to find the values of decision variables to reach the desired outcome with the most confidence.

The second decision support system assists diagnostic decisions in a sequential decision-making setting by finding the most promising tests and suggesting a diagnosis. The system can speed up the diagnostic process, reduce overuse of medical tests, save costs, and improve the accuracy of diagnosis. In this study, the system finds the test most likely to confirm a diagnosis based on the pre-test probability computed from the patient's information including symptoms and the results of previous tests. If the patient's disease post-test probability is higher than the treatment threshold, a diagnostic decision will be made, and vice versa. Otherwise, the patient needs more tests to help make a decision. The system will then recommend the next optimal test and repeat the same process.

The third decision support system recommends the best lifestyle changes for an individual to lower the risk of cardiovascular disease (CVD). As in the hospital

recommendation system, machine learning and optimization are combined to capture the relationship between lifestyle and CVD, and then generate recommendations based on individual factors including preference and physical condition. The results demonstrate several recommendation strategies: a whole plan of lifestyle changes, a package of n lifestyle changes, and the compensatory plan (the plan that compensates for unwanted lifestyle changes or real-world limitations).

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF FIGURES

Figure

CHAPTER I
INTRODUCTION

Tens of thousands of Americans die each year due to errors in the health care system, and tens of thousands suffer from nonfatal injuries due to the same cause [62]. Health IT Framework [93] proposes several strategies, such as IT adoption, collaboration, and informed consumer choice of clinicians and institutions. Electronic health record (EHR) adoption is one of the most important steps because it facilitates the development of new tools for error prevention, cost reduction, and health promotion.

More and more decision support systems (DSSs) have been designed to help clinicians. These DSSs store and use knowledge when a query comes in. The knowledge of these DSS tools usually comes from the direct input from human experts, for example, rules. The proposed DSSs gather knowledge automatically, and use optimization methods to return appropriate answers to queries.

This dissertation explores the use of electronic health record (EHR) with new data mining techniques and discusses solutions for three distinct types of health care problems. They are DSSs constructed by machine learning (ML) methods. They show the potential of machine learning to facilitate personalized health care, to reduce the use of health resources, and to improve outcomes.

The first project is individualized hospital referral [22]. Appropriate hospital selection can reduce the risk of suffering (e.g., death, complications). A hospital-referral decision based on the integration of individual and institutional variables not only estimates individual risk in an institution but also addresses the trade-off problem when deciding the most appropriate hospital, e.g., trade-off among several desired targets, such as survival, complication, travel distance, length of stay, cost, and other factors that relate to hospital referral decision. For example, the system shows a patient's survival and complication probabilities of each hospital and their

travel distances for a user who wants to find the best local hospital. The system will be discussed in Chapter III.

The second project is the cost-effective diagnosis problem [20]. In order to achieve the goal of cost-effective diagnosis, a good balance of diagnostic accuracy, testing cost, and the duration of diagnosis is very important. For this problem, a new method is proposed to optimize the diagnosis based on individual pre-test probability of disease and each test's contribution and cost. Diagnosis can be made when a patient's probability of the disease of interest crosses the treatment/non-treatment threshold. Testing cost and the number of tests can be significantly reduced while the diagnostic performance remains high, and sometimes improves. The system will be discussed in Chapter IV.

The third project is individualized lifestyle recommendations. One may lower the risk of CVD through appropriate selection of nutrition and lifestyle. For example, Lichtenstein et al. [66] provided recommendations of diet and lifestyle for the whole population. Compared to population recommendations, individualized recommendations have several advantages. First, the recommendation is given based on the current status of an individual. For example, the recommendation for an athlete, a vegetarian, and a meat or a fish lover should vary because they may have different needs of nutrition. Second, the recommendation can be constructed based on one's preference. For someone who doesn't have time for exercise, the recommendation may add certain nutrition or other lifestyle to compensate for exercise. Third, the system can measure personal risk reduction. The resulting system has the potential to provide a flexible and a customized lifestyle recommendation. The system will be discussed in Chapter V.

In general, this dissertation shows that machine learning can solve many health care problems at the individual level and, thus, provide flexible recommendations. This dissertation discusses the relevant literature in Chapter II. The methods and

results for each problem are discussed in Chapters III to V. A conclusion and future work are discussed in Chapter VI.

CHAPTER II
LITERATURE REVIEW

This chapter discusses expert systems (Section 2.1), relevant machine learning (ML) topics (Section 2.2), and the use of ML in expert system design (Section 2.3). Finally, we discuss current decision support systems in health care (Section 2.4).

2.1 Expert Systems

An expert system is defined as "a computer program that represents and reasons with knowledge of some specialist subject with a view to solving problems or giving advice" [58]. It usually consists of a knowledge source and a mechanism for problem solving that returns a response based on the information provided by the query. The knowledge source of most expert systems (e.g., knowledge-based systems (KBS), fuzzy expert systems) is based on direct input from domain experts and evidence from the literature. As an early example, MYCIN [87] provides diagnostic and therapeutic recommendations. The knowledge in MYCIN is stored in the form of rules, which were elicited from infectious disease experts. The process of transforming human knowledge to machine-usable form is called knowledge acquisition and is considered a bottleneck [41] because it is time- and labor-intensive. In addition, maintaining the knowledge base is very labor-intensive [100, 25].

Other systems use techniques such as case-based reasoning and machine-learning methods for inference, and are thus based exclusively on data. They can avoid the knowledge acquisition problem, e.g., case-based reasoning (CBR) as described in [99]. In CBR, the knowledge consists of previous cases, including the problem, the solution, and the outcome, stored in a central location, called the case library. To obtain the solution for a new case, one simply identifies the case that is most similar to the problem in the case library, and the proposed solution can be adapted from the

retrieved case.

Similar to case-based systems, ML-based expert systems can avoid the bottleneck of knowledge acquisition because knowledge is directly obtained from data. In addition, ML-based expert systems are able to give recommendations that are generated by non-linear forms of knowledge, and are easily updated by simply adding new cases. This dissertation shows an application of an ML-based expert system that uses a non-linear form of knowledge and optimization techniques to guide selection of hospitals, diagnostic testing sequences, and lifestyles.

2.2 Machine Learning

Machine learning is an area of artificial intelligence that uses algorithms to, for example, improve performance over time, or find patterns in data [37]. This dissertation applies new algorithms to discover useful information for three different health care problems. Generally, machine learning methods can be classified as supervised and unsupervised methods [36]. Supervised methods are trained with labeled data; that is, cases that have known outcomes. Decision functions, which result from the training process, can transfer variable values into predicted scores or labels. We examine prediction performance by comparing with true labels, which is a process called validation. In order to implement this idea, one needs to divide a dataset into two subsets, one for training and the other one for testing (or comparing). A better validation method with low variance, low bias, and easy computation properties is called n-fold cross-validation [37], which uses $n-1$ partitions of the data for training and one fold for testing and repeats the process for n times. Unsupervised methods learn from unlabeled data, and group data based on similarity. The machine learning methods discussed in this dissertation focus on the first type.

The decision functions of supervised learning methods constitute the knowledge that is mined from data. With an appropriate design, we can apply these functions to

many problems, for example, prediction, word recognition, movie recommendation, etc. This dissertation discusses and explores new usage of decision functions in medical decision making and introduces several topics of machine learning that relate to the three DSSs in this section.

2.2.1 Support Vector Machines

Support Vector Machines (SVMs) [97] are well-known for solving high-dimensional prediction problems and providing good generalization. The training process of SVMs can be expressed as the quadratic optimization problem.

$$\begin{aligned} \underset{W,\epsilon}{\text{minimize}} \quad & \langle W \cdot W \rangle + C(\sum_{i=1}^{l} \epsilon_i) \\ \text{subject to} \quad & y_i(\langle W \cdot x_i \rangle + b) \geq 1 - \epsilon_i, \ i = 1, \ldots, l, \\ & \epsilon_i \geq 0, \ i = 1, \ldots, l. \end{aligned} \tag{2.1}$$

where W is weight vector, ϵ_i is the error of training case i, y is the class label, x is the data for training, and b is the bias term for the decision function.

The objective function enables the training process to generate a predictive model with minimized training error ϵ and maximized margin r, where $r = \frac{|f(x)|}{\|W\|}$ and $f(x)$ is a decision function. Maximizing r is mathematically equivalent to minimizing $\|W\|$.

In the ideal situation, the label y (1 or -1) times the decision function $\langle W \cdot x_i \rangle + b$ of a training case should always be greater than one. However, for most data, this is infeasible. Therefore, the error term ϵ is introduced to tolerate the above infeasibility.

Optimizing both error and margin enables the predictive model to avoid the over-fitting problem so that the model can generalize well to unseen data. The parameter C can balance training error and margin.

To solve the optimization problem, the formulation is transformed to dual form

(Equation 2.2). α is a Lagrange multiplier, and $K(x_i, x_j)$ is a kernel function. The kernel function is very important in solving problems with high dimensions. For example, consider a problem in which the input space has three dimensions x_1, x_2, and x_3. We transfer them into a high-dimensional feature space x_1, x_2, x_3, x_1^2, x_1x_2, x_2x_3, x_1x_3, x_2^2, x_3^2... to produce a non-linear model. When the number of input features is large, the computation will be very slow. The computation of space transformation can be ignored by using the kernel function so that the computation of non-linear decision functions in SVM is very efficient.

$$\begin{aligned} \underset{\alpha}{\text{maximize}} \quad & \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j K(x_i, x_j) \\ \text{subject to} \quad & \textstyle\sum_{i=1}^{l} y_i \alpha_i = 0 \\ & 0 \leq \alpha_i \leq C, i = 1, \ldots, l. \end{aligned} \tag{2.2}$$

SVMs have been successfully used in several areas such as face detection and authentication, object detection and recognition, handwritten character recognition, speaker recognition and speech recognition, information and image retrieval, and data condensation [17].

2.2.2 Predicting Probabilities (Calibration)

Although SVMs can classify well, the output scores are not probabilities. Scores can not be interpreted as the chances of membership in the class. For example, a score can predict the degree of survival of a patient, but the score cannot show the probability of survival among all patients. The range of SVM output scores is $[-a, b]$, where a and b depend on the specific problem. In addition, to rank predicted scores from different SVM classifiers is inappropriate.

A forecaster is well-calibrated if the predicted probability is close to the empirical class membership probability [28, 105]. The quality of calibration can be analyzed

by a reliability diagram [28]. In this diagram, the prediction space is separated into 10 bins. Cases with values between 0 and 0.1 are located in the first bin, cases with the values 0.1 to 0.2 are located in the second bin, etc. Then we calculate and plot the point of the proportion of positive cases against the average predicted probability for each bin. If the forecaster is well calibrated, all points should be close to diagonal line.

The output scores of an SVM tend to be away from the extreme ends. The predicted points would form a sigmoid shape on the reliability diagram and a good calibration method can adjust the sigmoid shape to a near-diagonal line on the diagram [77]. Platt [82] used a sigmoid function to calibrate SVM scores. The raw SVM scores are transfered into posterior probabilities by

$$P(y = +1|d) = \frac{1}{1 + exp(Ad(x) + B)}. \tag{2.3}$$

The parameters A and B are trained from the negative log likelihood of the data, which is a cross-entropy error function (2.4). Platt suggests using an independent calibration set to train this function instead of the dataset that has been trained with the SVM classifiers. If we use the same dataset, most raw SVM scores are either 1 or -1, and bias will be introduced. The objective function is

$$\underset{A,B}{\text{minimize}} \qquad \sum_i t_i log(p_i) + (1 - t_i) log(1 - p_i) \tag{2.4}$$

where $p_i = \frac{1}{1+exp(Ad(x_i)+B)}$ and

$$t_i = \begin{cases} \frac{N_+ + 1}{N_+ + 2} & \text{, if } y_i = +1 \\ \frac{1}{N_- + 2} & \text{, if } y_i = -1 \end{cases}.$$

where N_+ is the frequency of the data point with $y_i = +1$ and N_- is the frequency

of the data point with $y_i = -1$.

Validation is difficult in recommendation systems [58] (e.g., hospital-referral and lifestyle decision support systems). This is because we cannot know the gold standard after recommendations. Patients in a dataset never received the system's recommendations. The dataset provides only patients' original actions, and we would like to know whether or not performing the recommended actions from the system can improve outcomes, e.g., if switching from the originally-chosen hospital to the recommended hospital can improve the chance of survival. We compare predictive probabilities (by Equation 2.3) of both hospitals.

The calibration method also enables comparison across classifiers that are trained with different sets of variables. In Chapter IV, we need to compare tests with the same known information one at a time. Each classifier is trained with a test and the known information. Ranking scores from classifiers trained with different subsets are inappropriate because the scale of each classifier varies. The calibration method enables the comparison because the range of the scale is standardized.

2.2.3 Learning with Costs of Misclassification and Attributes

Most supervised learning methods focus on minimizing misclassification errors. In certain situations, the misclassification error is not the only concern. For example, the cost of misclassification may be very small and the cost of introducing an attribute to improve the classification very high. Several criteria for machine learning can be expressed as costs [95], such as misclassification, attributes, instances, etc. Among them, the costs of interest are costs of diagnostic tests and misclassification.

In diagnoses, we want to minimize the costs of medical tests and maximize the diagnostic power (or minimize the misclassification cost). This is the important goal of triage because we want to sort patients accurately based on risks. Finally, we can reduce time and medical resources and diagnose correctly.

The misclassification cost of different classes usually varies in real life. For example, the misclassification cost of diagnosing a patient with a serious disease as healthy is far more than diagnosing a healthy patient as ill. A cost-sensitive classifier [35] was originally formulated to treat misclassification of different classes unequally and uses different costs to construct the model. When learning, a cost-sensitive classifier pays more attention to the class with a higher cost in order to reduce the chance of misclassifying this class. Studies such as [94, 69, 18, 68, 107] not only considered different misclassification costs but also test cost for the medical diagnostic problems. They explicitly minimize both type of costs in specialized classifiers, such as decision trees. Minimizing both costs results in trade-off choices.

A very similar research area considering cost, misclassification, and attributes is active learning. These learning approaches (e.g., [85, 106, 72, 60]) automatically acquire feature values, instances, and/or class labels to reduce the cost of data collection and obtain maximum information.

This dissertation proposes a novel machine learning approach to maximize diagnosistic accuracy by selecting an appropriate individual sequence of tests. In other words, one can be diagnosed with a minimum number of tests (and minimum cost) when following the individual testing order. This approach significantly reduces the cost of tests (and number of tests) without sacrificing (while sometimes improving) diagnostic performance. This approach is an instance-based (the model specifically constructed for an individual) learning algorithm that selects a testing sequence based on an individual's pre-test (prior) probability and given treatment (or non-treatment) thresholds.

2.2.4 Feature Selection

The purpose of feature selection is to reduce the number of predictive features used in the model while improving or without degrading performance. A large number of features causes several problems. The first problem is the cost of computation. When the number of features is high, the computation time and space will rise dramatically. The problem becomes intractable for some simple induction algorithms. Another problem is the generalization of predictive performance. Complexity increases with the number of features, and high complexity may result in over-fitting because too many features may be redundant or misleading. In addition, a large number of features requires a lot of storage space and may increase the cost of data collection.

An example of feature selection is in the area of gene selection. We want to know what genes relate to different health states. Gene expressions are variables whose size may range from 6000 to 60000, for both healthy and diseased patients [47]. Feature selection can reduce the number of variables to a few thousand.

John et al. [59] classified feature selection techniques as either filter or wrapper models. The filter model is a preprocessing step to induction methods. Feature ranking is an example of filtering. In feature ranking, we use a function independent of the induction method to rank features based on scores. For example, features can be ranked by the Pearson correlation coefficients, $R(i) = \frac{cov(X_i,Y)}{\sqrt{var(X_i)var(Y)}}$, where X is the variable set, Y is the class label, and i indicates the variable of interest. We can also build several single-variable classifiers, and rank classifiers (features) based on error rates.

In wrappers, the induction method is used in the feature selection procedure. Examples of wrappers are forward selection and backward elimination [59]. Both are greedy search methods in the space of possible feature subsets. In forward selection, we start with an empty set for the model and add the most promising feature to the

model in each iteration. In backward elimination, we start with a model full of all features, and eliminate the least promising feature in each iteration. Other examples of wrappers include using heuristic search methods, such as genetic algorithms, to find the most promising feature subset [96].

Another group of feature selection techniques use embedded methods [47]. Similar to wrappers, the prediction model is involved; however, we do not need to retrain with every subset of features, as the learning objective is changed to explicitly include a cost for including features. An example is sensitivity analysis. We use the magnitude of change of the cost function caused by removing a feature (setting the weight to zero) to rank features.

Direct objective optimization is in the family of embedded methods. The optimization of the model construction directly includes feature selection. For example, the model construction of [14, 81] includes optimization of a loss function and a regularizer. The regularizer is l_1 norm, which can force a subset of weight to zero. The objective function is

$$CF = \frac{1}{m}\sum_{i=1}^{m} L(y_i, f_i) + \lambda \sum_{j=1}^{n} |w_j|^p \,, \tag{2.5}$$

where L is the loss function, w_j represents the weight of a feature, and $p = 1$ in the regularizer.

In this objective function, we want to have the minimum CF. Although introducing a nonzero weight can decrease the loss function, the penalty of the regularizer will increase CF. Therefore, we only keep the weights whose reduction of the mean loss outweighs the penalty of regularizer. Because of the efficiency of obtaining feature subsets, direct objective optimization was applied to online feature selection (OFS) [81]. In this scenario, we assume that features come at different time points. Without examining every subset of features when a new feature comes in, the OFS model

directly returns the best subset model seen so far.

Many researchers [1, 102] have used unequal feature weights to conduct feature selection for lazy learning algorithms. Lazy learning methods, such as k-nearest neighbor, use distance functions to decide the predicted label for a query case. When feature weights are equal, irrelevant features can destroy predictions. Hence, assigning weights to features based on relevance to the class labels is an alternative form of feature selection.

These feature selection methods select one set of features for all data points, which is a global strategy. Local variants create multiple sets of features for different data points. This local strategy examines local regions of the case space because the relevance of features may vary across different clusters of cases [2]. The most extremely local strategy uses one set of features for each instance. For example, Domingos [34] uses a clustering-like approach to select sets of locally relevant features. The feature-weighted methods also use different weighted sets for different instances or classes. For example, Howe and Cardie [52] use class distribution weighting to compute a different weight vector for each class, and Park et al. [79] use Artificial Neural Networks (ANNs) to compute the feature weights for an instance.

These local feature-selection techniques can select relevant features for an instance, in which the order of including features is not a concern. For the sequential diagnosis problem, we assume that a query is a new patient, such that each medical test can be seen as a feature whose value is initially unknown. Feature-selection techniques may identify at the outset all important features that one needs to characterize a patient. In the sequential diagnosis scenario, one needs to select tests sequentially and dynamically because one needs to determine whether more evidence is needed or a diagnosis can be made, especially when the test or the cost of outcome is very expensive. In addition, various feature subsets may result in different optimal selection of a feature, i.e., the optimum selection of a feature may be influenced by features at

hand [48]. Thus, recursively including one test at a time may be better than including all relevant tests simultaneously.

In this dissertation, we propose a new feature-selection technique called Locally Temporal Feature Selection (LTFS). LTFS is a local feature-selection technique for one instance (query patient) that recursively determines the most relevant feature based on the current subset of features and their values. In other words, the next test is determined based on the known information at hand, e.g., symptoms and previous test results. Once the treatment threshold (the probability threshold that determines whether a patient has a disease) is reached, the LTFS algorithm stops. The criteria used to select a test include the relevance of a feature and the associated cost; these criteria are implemented in speed-based and cost-based objective functions, which will be discussed in Section 4.1.3.

2.2.5 Lazy Learning

Lazy learning – also called instance-based, case-based, or memory-based learning – builds a prediction model specifically for the query case. For example, a k-NN classifier finds the k closest training cases to decide the label for the query case. Lazy learning algorithms show three types of properties [3]. First, the classifiers defer processing of the output until a query case appears. Second, their responses combine the training with the query information. Third, they discard the constructed answer and any intermediate results.

Eager learning algorithms, such as SVMs and decision trees, have different properties from lazy learning. Eager learning algorithms compile the data in advance and use it to construct a predictive model. They use this global model to give responses to all queries. Thus, the compilation process is called training, which is unnecessary in lazy learning methods. Instead of training, lazy learning algorithms need to store all the training cases.

We can also apply the idea of lazy learning to decision tree algorithms. For example, the case-specific decision tree assigns training cases with the same feature value as the query case recursively during the training process [43]. There is also lazy version of cost-sensitive decision tree. For instance, [68] assigns the attribute costs of the known attributes of the query to be zero because known attributes are patient-specific information. As a result, tests with the known values are very likely to appear at the top of the trees.

An alternative form of lazy learning is locally-weighted learning [6]. Instances are weighted based on the distance to a query point. The lazy learning property can be demonstrated in two forms, weighting the data directly and weighting the error criterion. Weighting the data directly can be seen as voting with unequal weights, which is

$$\hat{y}(q) = \frac{\sum_i y_i K(d(x_i, q))}{\sum_i K(d(x_i, q))}, \tag{2.6}$$

where $\hat{y}(q)$ is the predicted label of the query point q, $i \in$ all training data points. x_i is the ith training vector with the label y_i. K is the weighting function based on the distance function d. The labels of closer points can be weighed more highly than distant points. This is very similar to k-NN, but we do not need to decide the number of k.

Instead of weighting instances, the second format of locally-weighted learning weighs the error function based on instances. The general format is

$$P = \sum_i [L(f(x_i, w), y_i) K(d(x_i, q))], \tag{2.7}$$

where L is the loss function for the predicting function $f(x_i, w)$, and w is the parameter vector for the predicting function. The objective of training is to minimize P.

After training, the closer points will have less training error and vice versa.

Chapter IV discusses the second format, lazy SVM. The advantage of lazy SVM is accuracy improvement because the predictive model is more patient-specfic as more test results become known. Chapter V discusses more detail about k-NN classifiers as they are used as prediction models in the lifestyle recommendation project.

2.3 Combination of Expert Systems and Machine Learning

The ways to use machine learning in expert systems are very limited [8, 58]. Most methods that can provide help are rule induction approaches. These approaches are limited to deriving rules, evaluating rules, and optimizing the performance of rules [58] for expert systems.

However, with an appropriate design, one can directly turn machine learning algorithms into expert systems. Machine learning approaches are renowned for knowledge discovery. They can model real world problems using decision functions that exist in the form of mathematical equations, rules, or decision trees. These methods can predict well and provide useful information. For example, SEAS [101] can provide prediction for business actions whose classifiers are trained with low cost information.

Using a mathematical form of knowledge has the advantages of stability, observability, and controllability [56, 73]. However, capturing a complex real-world model using a mathematical equation is difficult. When the real-world problems can be modeled by mathematical equations, we can use several optimization approaches to decide actions. For example, Liau et al. [65] devised an expert system for a crude oil distillation unit to help control the parameters to maximize the oil production rate. In another example, Song et al. [90] used an expert system to find the optimal control settings to maximize the boiler's performance.

Generally, the types of solutions that can be structured by expert systems can

be divided into selection and construction [23]. For selection, several action sets have been pre-determined, and the solution is the most promising action set. On the other hand, construction needs to construct a set of actions from scratch. In order to avoid infeasible solutions, constraints can regulate the solution construction. In the hospital-selection problem, each hospital has a unique set of characteristics (pre-determined action set). Deciding the most promising action set is equivalent to selecting the best hospital, so it is a selection problem. The problem in the healthy life-style project is construction. There are no pre-determined lifestyles options, so we have to generate and evaluate possible options of lifestyle (e.g. the combination of cholesterol, total fat, weight, and exercise in different amount) and determine the one that maximally reduce one's cardiovascular disease risk.

2.4 Decision Support Systems in Health Care

The idea of decision support has been widely accepted in health care, starting from a simple database query to complex treatment recommendation. The development of clinical decision support systems (CDSSs) have drawn much attention. The scientific literature provides the major source of knowledge accompanied by local and practiced-based evidence [88]. The knowledge of CDSSs exists in the form of guidelines. There are four areas in the process of developing a guideline-based decision support system [27]. The first area is guideline modeling and representation. This area focuses on the representation of a guideline, e.g., the form of expression, knowledge type, maintenance, local adaptation, etc. The second area is guideline acquisition, which is a process that facilitates the knowledge acquisition process directly from a domain expert. The third area is guideline verification and testing. This process aims to ensure the machine-interpretable guideline is unambiguous and syntactically as well as semantically correct. In addition, we need to test guidelines using existing patient data. The final process is guideline execution, which focuses

on the execution time and ensures the guideline engine can run in multiple clinical domains and in various modes. Examples of the guideline-based decision support systems are Arden Syntax [24], GuideLine Interchange Format [78], and PRO*forma* [42].

An alternative approach to guideline-based approach is machine learning, e.g., artificial neural networks or support vector machines. Lisboa et al. [70] provides a systematic review of these approaches that provides decision support in cancer. Instead of the required process from guideline modeling to execution, these approaches gain knowledge automatically from clinical data and then use the knowledge to provide decision support.

CDSSs provide several functions to assist medical decisions. Medication-related CDSS [63] can provides basic and advanced decision support. Basic decision support includes drug-allergy checking, basic dosing guidance, formulary decision support, duplicate therapy checking, and drug-drug interaction checking. Advanced decision supports includes dosing support for renal insufficiency and geriatric patients, guidance for medication-related laboratory testing, drug-pregnancy checking, and drug-disease contraindication checking. Thus, CDSSs can help to reduce medication error rate, prescribing behaviors, corollary orders, etc. [61]

CDSSs seem very helpful in supporting medical decisions, but it is difficult to get health providers to actually use them, so Bates et al. [9] summarizes ten rules for sucessful implementation of CDSSs. These rules are: 1. speed of providing recommendation; 2. anticipate needs and deliver in real time; 3. fit into the user's workflow; 4. little things can make a big difference (usabilty matters); 5. recognize that physicians will strongly resist stopping (suggestions) (need to provide alternative options to avoid resistance); 6. changing direction is easier than stopping; 7. simple interventions work best (fit a guideline on a single screen); 8. ask for additional information only when you really need it; 9. monitor impact, get feedback and

respond; 10. manage and maintain your knowledge-based systems.

Decision support systems are also used in other domains of health care. For example, Bravata et al. [15] summarizes surveillance systems for early detetion of bioterrorism-related diseases. Integration of Geographic Information Systems (GIS) and health care allows describing and understanding the changing spatial organization of health care, examining the relationship between health outcomes and access, and exploring how the delivery of health care can be improved [71].

CHAPTER III
HOSPITAL-REFERRAL EXPERT SYSTEM

Hospital referral criteria usually come from research studies and personal experience. Many researchers have examined the relationship between outcomes of hospitals and various institutional characteristics. In particular, a large number of studies have related the volume of hospital surgical procedures to decreased in-hospital mortality [49, 51, 44, 11].

Likewise, teaching hospitals have been shown in several studies to have lower in-hospital mortality [64, 7]. Chen et al. [19] concluded that hospitals participating in the JCAHO survey process reported superior quality and outcomes. Elixhauser et al. [38] and others have reported that staffing affects quality. Among these institutional characteristics, the volume of patients or procedures is the most consistent predictor of in-hospital mortality and is broadly used as a hospital selection criteria. Although the volume-outcome relationship holds for a number of complex surgeries, the magnitude of association varies across procedures [11, 49]. Both "practice makes perfect" and "selective referral" appear to play a role in the volume-outcome relationship [54]. Usually, large institutions have favorable characteristics, such as technical sophistication and more staffing, and they are usually preferred for referral.

Although surgical volume is a strong predictor of outcomes, the usage of this indicator is sometimes criticized. Nallamothu et al [76]. explained three reasons that the quality of high-volume hospitals looks better than low-volume ones. First, low-volume hospitals may be less inclined to turn down high-risk cases. Second, large-volume hospitals attract more cases through health provider referral or self-referral. Third, patients with opportunity and desire to be referred may be healthier because of several factors. These reasons can help to explain variations in the volume-outcome relationship. Many low-volume centers have very good performance, while

some high-volume hospitals have poor performance because volume is an imperfect proxy measure of quality [33, 54, 11, 49].

While many of these studies examine only one or two predictive variables, for practical usage, a good hospital referral decision should consider numerous factors, specifically including geography. Some medical situations are time-critical, and transportation time plays a very important role in outcomes. For patients living in rural and underserved areas, distance is often the most important concern when selecting a hospital. Even for non-emergency conditions, proximity is highly desirable. Therefore, several studies [12, 32, 33, 76] have shown that patients often prefer local higher-risk hospitals over traveling to lower-risk hospitals. Geographic factors may influence the effect of institutional predictors. Ward et al. [98] indicated that the volume threshold suggested by The Leapfrog Group [46] does not perform well in a largely rural state. Other factors, such as a patient's physical condition, should also be considered. Glance et al. [45] stated that the risk reductions of high- and low-risk patients in different volume hospitals vary. If we considered the distance to an institution, the hospital-referral recommendation for a healthier patient and a sicker patient can be dramatically different. A good hospital referral recommendation considers not only institutional but also patient factors, including the travel distance a patient can tolerate, and the patient's risk factors. Not surprisingly, it is challenging to give hospital-selection advice that considers these multiple complex and interdependent issues.

Some practical problems may arise if we consider only institutional factors. For example, should an acute myocardial infarction (AMI) patient go to a mid-size teaching hospital with JCAHO accreditation 20 miles away or a large-volume non-teaching hospital 40 miles away? The Leapfrog Group [46] suggested that a good hospital would have a procedure volume greater than 450 for coronary artery bypass graft (CABG) surgery. Should an 70-year-old AMI patient with congestive heart

failure and diabetes who needs an emergency CABG choose a hospital with CABG procedure volume of 300, thirty miles away or another hospital with CABG procedure size of 450, forty miles away? How about a younger and healthier patient who is not in an emergency situation but needs surgery? Obviously, the answers would be different for different people. It is hard to tell which hospital is better when we consider only institutional characteristics. The hospital referral problem is even more complex if we add other practical concerns, such as insurance coverage and estimated charge during a hospital stay.

The purpose of this project is to apply Prediction and Optimization-Based Decision Support System (PODSS) to minimize these trade-off problems by customized hospital-selection decision support. The system can find the best match between a patient and a hospital based on characteristics of both and the patient's preference or consideration (e.g., the distance that doesn't incur a risk).

3.1 Method

There are a series of stages in the Prediction and Optimization-Based Decision Support System (PODSS) algorithm. As illustrated in Figure 3.1, the algorithm relies on classifiers to capture knowledge. This step is the same as training a prediction model. In this project, the model is a Support Vector Machine (SVM). Independent and dependent variables are required to train the model. The output score of the prediction model, which we convert to a probability (described in Section 2.2.2), can be interpreted as the confidence level of the desired class prediction. The purpose of optimization is to maximize the confidence level of the desired class label (e.g., the class value is either survival or decease and the desired value is survival) by selecting the best hospital.

3.1.1 Prediction and Optimization-Based Decision Support System (PODSS)

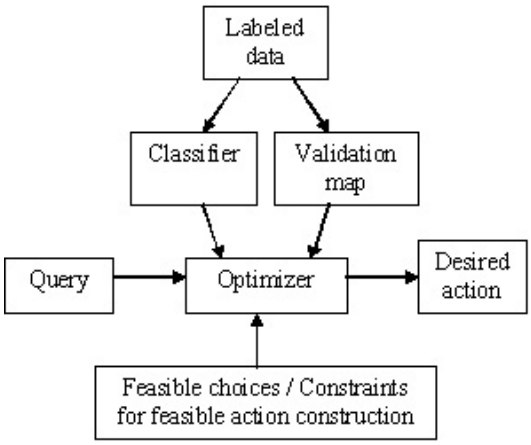


Figure 3.1: The PODSS process to capture and apply knowledge

This expert system is constructed by the PODSS algorithm, (illustrated in Figure 3.1). This algorithm consists of three major parts: the prediction model, the optimizer, and the validation map. As most prediction problems, a classifier is trained with labeled data. The binary labels are assumed to be desired and undesired outcomes, e.g., survival or not. The predicted scores can be seen as confidence. When we change the values of a subset of variables, the confidence will be different. Thus, we can use an optimizer to optimize the confidence of the desired label and decide what actions should take.

In many situations, we prefer probabilities over scores. A calibration method can transfer scores into probabilities. The calibration method can also provide indirect validation; therefore, it is called the validation map. Through this map, we can observe whether or not the change of actionable variables can influence the outcome.

Independent variables that can change and move a point are called controllable (changeable) variables, and variables that can not change are called uncontrollable

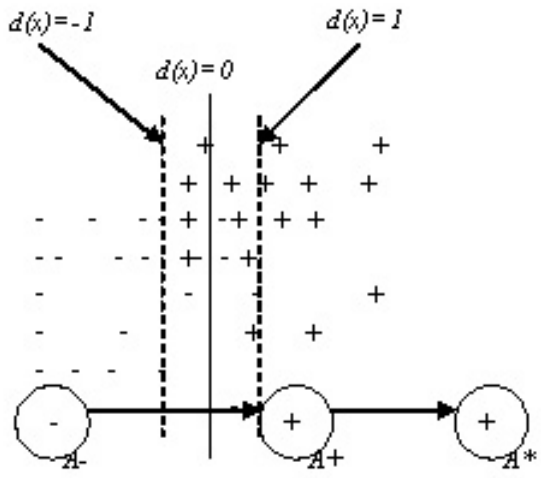


Figure 3.2: Separating hyperplane with maximum margin created by a support vector machine. + and - represent the class of each data point. We assume + is the desired result (survival). The decision function, $d(x) = \sum_{i=1}^{n} y_i \alpha_i K(x, x_i) + b$, can decide the location of a point as the prediction result. We can improve the probability of a point being positive by improving the decision value, $d(x)$, of this point , for example, moving the point A- to A+ or A*.

(unchangeable) valuables. For example, consider a model using age, sex, and life style to predict health. We can only change lifestyle to improve health. To be meaningful, the selected controllable variables should have causal relationship with outcomes. The recommendation is made based on these controllable variables.

In the point-movement scenario (Figure 3.2), a recommendation problem becomes an optimization problem, and the decision function turns into the objective function. In the optimization formulation, the controllable variables are the decision variables. When maximizing the decision function, a point can be moved toward the desired outcome maximally and the confidence of the desired outcome is maximum. Decision variables are the recommended action(s). The decision function is the knowledge source, and the optimization method is the mechanism for problem solving, which returns a customized recommendation based on the query's individual

information.

The optimization should provide a feasible recommendation. If the way to generate answers is selection, we simply provide feasible choices of solutions, and the optimizer will select the best one. In the selection problem, we can use exhaustive search or heuristic search (depending on the size of the solution space) to find the solution. When the way to obtain the answer is construction, the optimizer needs to construct an answer. In this case, mathematical programming can construct the answer. Without any constraint, the optimum recommendation may be infeasible in the real world. In order to solve this problem, the optimum answer should be subjected to certain constraints provided by human knowledge. Thus, expert knowledge can be incorporated into the optimization process.

Constraints can come from users or designers. A user may provide personal preference as a constraint through communication with the system. For example, after a user gives the maximum travel distance, all returned hospitals are constrained to be shorter than or equal to this maximum distance. A designer can prevent infeasible solutions through constraints. For example, if one uses PODSS to recommend drugs, the constraints of preventing drug interactions are necessary.

3.1.2 Dataset and Variables Design

The 2004 State Inpatient Dataset (SID) for Iowa from the Agency for Healthcare Research and Quality (AHRQ) Healthcare Cost and Utilization Project (HCUP) [50] was used in our study. There are almost 360,000 discharge records in the SID. For this project we chose to build a hospital referral algorithm for patients with a principal diagnosis of acute myocardial infarction (AMI) with ICD-9-CM codes of 410.01 to 410.91.

We selected AMI for several reasons. First, it is relatively common and easy to identify in the datasets. Second, the outcomes of interest, including mortality,

are also relatively common which facilitated model building. Third, AMI in-hospital mortality is being introduced by the Centers for Medicare and Medicaid Services as a required publicly-reported performance indicator for all hospitals, thus the algorithm described here could find application in the near future. While we use AMI for this first demonstration, the algorithm can be easily modified to work with nearly any disease of interest where referral is an issue.

The SID can be linked to hospital descriptive data from the American Hospital Association (AHA) [4] by a hospital identification number. There are 116 non-federal acute-care hospitals in Iowa. The SID contains zip codes for each patient and hospital, which permit the Euclidean distance between a patient and any hospital to be computed. The location (longitudinal and latitudinal) data was retrieved from http://www.brainyzip.com/. The distance estimation from MapQuest or Google Maps could be used to compute road distance estimation. Road distance estimation more accurately represents travel distance and is longer than Euclidean distance [57]. We choose the Euclidean distance estimation in our study because it was readily available for each patient/hospital pairing, whereby road distance estimation is not, and the difference between the two methods is relatively consistent in Midwestern states.

Four datasets were designed in this study. The labels of the first three datasets are patients' in-hospital survival status, and the labels of the fourth dataset are hospital-acquired complication status. The complication labels are identified using ICD-9-CM codes of complication defined by Elixhauser and colleagues [39]. The first dataset includes all AMI patients whether surgery is performed or not. The second dataset is designed for AMI patients who do not have any surgery. In this dataset, patients with any surgical Diagnosis Related Groups (DRG) were excluded. The third dataset includes only AMI patients who have coronary artery bypass graft (CABG) surgeries (ICD-9 36.10 to 36.19). The last dataset contains the same patients as the

third one but with a different label type. The data show only 12 hospitals in Iowa perform CABG surgeries. Thus, hospital selection is limited to these 12 hospitals in the third and fourth dataset. The size and percentage of the desired outcome label in each dataset are shown in Table 3.2.

Our approach depends on the problem having two distinct types of independent variables. The first type is uncontrollable (unchangeable) variables. The values of these variables are given and cannot be changed. For example, patient variables such as demographic data, medical test results, diagnostic results, admission type, surgery status, comorbidity scores [30] (indicates severity), and payment type are uncontrollable variables in this study. The second type is controllable (changeable) variables, whose values can be changed. The recommendation can be made based on these variables. In our application, each set of values of these variables describes a hospital. These hospital descriptive variables are owner type, hospital location, JCAHO accreditation, total number of surgical operations, AMI patient discharge volume, and the volume of CABG surgeries. Table 3.1 summarizes the variables used for classifier training. Variables 1 to 6 relate to the patient's characteristics and are uncontrollable; variables 7 to 13 relate to the hospital's characteristics and are controllable.

In the knowledge application stage, each type of variable plays a different role in the optimization process. The first set is constant and is provided by a user when querying. The solution variables comprise the second set. The optimal solution is the hospital with the most favorable descriptive variables that results in the highest optimum value (desired outcome with the highest probability). In a nonlinear model, the optimal solution may depend on the given uncontrollable variables due to variable interaction.

Dependent variables, such as survival status and complication status, are usually used to find the predictors of hospital quality. This study shows two types of expert

Table 3.1: Variables description

		Data type
1	patient age	numeric
2	patient sex	male, female
3	patient race	white, other
4	patient admission type	emergency, urgent, elective
5	patient comorbidity severity	numeric
6	patient payment type	Medicare, Blue Cross, Commercial, other
7	hospital ownership type	government owned or not
8	hospital bed size	numeric
9	hospital metropolitan status	numeric, from 0 to 6 based on population size
10	hospital JCAHO status	JCAHO accreditation or not
11	hospital surgery volume	numeric, total surgical operations
12	hospital discharge volume	numeric
13	hospital CABG volume	numeric

system based on the number of desired targets. The first is called a single-objective optimization experiment, in which we use a classifier to find the relationship between independent variables and patients' survival status. Datasets 1, 2 and 3 are used in this study. The second is called a multiple-objective optimization study. In addition to the survival status classifier, we add a second classifier for the complication status. The hospital-acquired complication is a very important concern to surgical patients, and the third and fourth datasets are used in this study.

3.1.3 Model Design

There are two steps in the system construction. The first step is knowledge capturing. This step is the same as training a predictive model. Then we use an optimization method to apply the knowledge, i.e., providing a recommendation.

Two applications can be represented using single- and multi-objective optimization formulations. The first problem selects the hospital with the highest survival probability. The formulation is Equation 3.1. The second problem selects the hospital with the highest survival and freedom from complication (FFC) probabilities. The formulation is Equation 3.2.

$$\begin{aligned} \underset{x_{2j}}{\text{maximize}} \quad & d(x_1 \cup x_{2j}) \\ \text{subject to} \quad & dist(j,x) \leq DL \\ & x_{2j} \in X_2 \end{aligned} \tag{3.1}$$

Here, x_1 represents the characteristics of the query patient, and DL is the maximum traveling distance. x_{2j} is the set of descriptive variables for hospital j. $dist(j,x)$ is the Euclidean distance between the patient and the hospital j.

In the first optimization problem, the objective function is the decision function subject to the maximum traveling distance. In other words, the recommended hospital

must be in the range of the maximum distance. Due to the limitation of our database, this research only considers hospitals in Iowa. Hence, no boarder health care systems of other states is considered. For an emergency case, the distance limit parameter, DL, is very small. We can simply use exhaustive search to solve this problem because the solution space is only 116 hospitals.

For the second problem, we also take complication into account. The desired targets are the highest survival probability, the highest FFC probability, and the lowest distance. In the problem formulation (Equation 3.2), the distance constraint has become a part of the objective function. This is a trade-off decision among the three targets. To allow a customized recommendation, the patient is the one that decides the balance of these targets. Thus, the best strategy to solve Equation 3.2 is to show the solution space instead of a single optimal hospital. The user can decide the best hospital according to the three-dimensional information, $d_1(x)$, $d_2(x)$, and $d_3(x)$. The optimization formulation can be expressed as

$$\begin{array}{ll} \underset{x_{2j}}{\text{maximize}} & D(x) = (d_1(x), d_2(x), -d_3(x)) \\ \text{subject to} & x_{2j} \in X_2 \end{array}, \tag{3.2}$$

where $x = x_1 \cup x_{2j}$.

Section 3.1.4 summarizes knowledge capturing and application.

3.1.4 The PODSS Algorithm

Building the recommendation system starts with knowledge capturing, which is summarized as Figure 3.3. We need to train parameters A and B for the validation map in ten-fold cross-validation (training steps 1 and 2) and a classifier with the whole dataset (step 3).

The recommendation method (Figure 3.4) begins when a query user enters the patient variables x_1 and the maximum distance DL. The recommendation system

Knowledge capturing

Input

Training data D, $D_i = x_{1i} \cup x_{2\bar{j}}$, where x_{1i} is the uncontrollable variable set for the patient i, $i = 1, \ldots, n$. $x_{2\bar{j}}$ is variable set of the hospital chosen by the patient

Outputs

1 $h(d(x))$: the sigmoid function

2 $d(x)$: the decision function

Training steps

1 Perform ten-fold cross-validation using the SVM classifier with D

2 Train Equation (2.4) with predicted values from the testing data and true classes to obtain $h(d(x))$

3 Re-train with whole dataset D, obtaining $d(x)$

4 Return $h(d(x))$ and $d(x)$

Figure 3.3: Training process in the PODSS algorithm

Captured knowledge application: Single-objective optimization

Inputs

1 Hospital characteristic data X_2

2 Query patient variables x_1

3 Maximum distance DL

4 Trained sigmoid function, $h(d(x))$, and the decision function, $d(x)$

Outputs

The hospital $\hat{j}$ with the highest survival probability, Pr, for the query patient.

Recommending steps

1 Find the hospital $\hat{j}$ for the query patient by equation 3.1.

2 Survival probability for query patient in this recommended hospital $\hat{j}$, $Pr = h(d(x_1 + x_{2\hat{j}}))$

3 Return hospital $\hat{j}$, $x_{2\hat{j}}$, and Pr

Figure 3.4: Knowledge application process for single-objective optimization.

that considers only one objective can find the best hospital using Equation 3.1 (Recommending step 1). Next, the score of the query patient with the hospital is converted to a survival probability (step 2).

We do not find the best hospital for the multi-objective recommendation. Instead, we show all hospitals within the distance limit to a query user. The steps are summarized in Figure 3.5. First, scores of survival and FFC of feasible hospitals

Captured knowledge application: Multi-objective optimization

Inputs

1 Hospital characteristic data X_2

2 Query patient variables x_1

3 Trained sigmoid function, $h_k(d)$, and decision function, $d_k(x)$, of survival ($k = 1$) and freedom from complication ($k = 2$)

Outputs

1 $SurvProb_j$: survival probabilities for all hospitals

2 $FFCProb_j$: FFC probabilities for all hospitals

3 $dist(x, j)$: the distances between each hospital to the query patient

Recommending steps

1 Compute $d_1(x_1 + x_{2j})$, $d_2(x_1 + x_{2j})$, and the distance, $dist(j, x_1)$, between the query patient and the hospital j. $j = 1, ...m$

2 Compute the survival probability, $SurvProb_j$, by $h_1(d)$ and the FFC probability, $FFCProb_j$, by $h_2(d)$ for hospital j. $j = 1, ...m$.

3 Return $SurvProb_j$, $FFCProb_j$, and $dist(x_1, j)$, $j = 1, ...m$.

Figure 3.5: Knowledge application process for the multi-objective optimization.

are computed in the recommendation step 1. Then, these scores are transferred into probabilities in step 2. Finally, the query user can find the best hospital based on distances, survival probabilities, and FFC probabilities.

3.2 Results

Table 3.2 shows class distributions of the four datasets. The desired class (positive) is the majority in all datasets. Directly modeling these sets results in highly accurate but useless classifiers simply predicting all (or nearly all) points to be positive. We used over-sampling of the minority class [67] to balance each dataset according to the proportion of positive to negative classes. For example, the survival (positive class) probability is 95% in the CABG survival experiment, so we used the ratio 1/19 to balance positive and negative classes.

We compare the mean square error of the probabilities generated by calibrated SVM with those created using logistic regression in Table 3.3. This table shows that the mean square errors of the calibrated SVM are significantly lower than logistic regression in all experiments.

In the following sections, we present the results using single- and multi- optimization. In actual application of the single-optimization, the maximum tolerated distance should be decided by a user, and the returned optimal solution is customized. We varied this parameter in order to present results. In the application of multi-optimization, a user does not need to give a parameter. Instead, the user needs to choose the optimal solution in the solution space considering three desired targets. Similar to the single-optimization, we varied the distance target and discuss the user's decision considering the other two desired targets.

3.2.1 Single-Objective Optimization

In this problem, we want to find the hospital with the highest probability of survival during a hospital stay. SVM outputs are not probabilities, so we need to transfer scores into probabilities by a calibration function. Figure 3.6 shows validation maps, which are generated by the calibration functions (Equation 2.3) from three datasets.

Table 3.2: Description of four datasets.

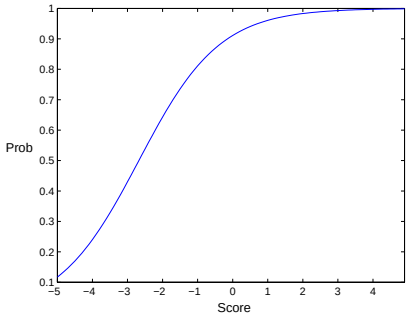
	desired outcome %	**data size**
all AMI	93.0%	6599
non-surgery	93.4%	5846
CABG	95.0%	466
CABG-FFC	81.8%	466

Note: Desired outcome for the first three databases is survival and the last database is freedom from complications (FFC).

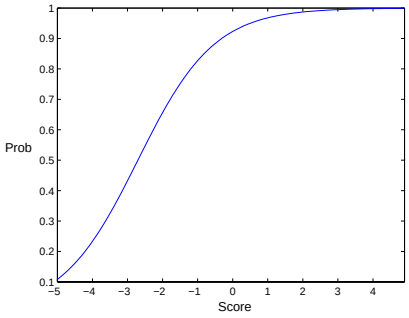
Table 3.3: Mean square error of predicted probability of survival for SVM and logistic regression

	Regression	**Calib. SVM**	**P-value**
all AMI	0.2137	0.0626	< 0.0001
non-surgery	0.2136	0.0593	< 0.0001
CABG	0.4461	0.0462	< 0.0001
CABG-FFC	0.2616	0.1481	< 0.0001

(a) All AMI patients experiment



(b) non-surgery AMI patients experiment



(c) CABG AMI patients experiment

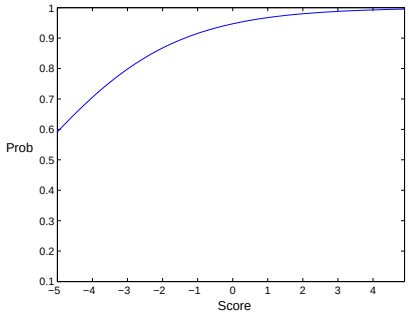


Figure 3.6: The validation map of all AMI patients and CABG-AMI patients.

Table 3.4 summarizes the results of recommendation for all patients. The original choice is the hospital originally chosen by patients. For the recommendation parts, we assume each patient uniformly gives 30, 50 100, and 200 mile distance limit parameters to the system. After entering the patient's variables (uncontrollable variables), the system will return the hospital recommendation (controllable variable set), the distance to this hospital, and the estimated score and probability of survival. There are average distance, average probability, and average score for each dataset. Average distance is the average distance between patients and their chosen hospitals. Average score is the average SVM score, and average probability is the average of probabilities transformed from scores. The average probability is very close to the true survival probability in Table 3.2. This indicates the predictive probability is closed to the truth.

Patients' expected survival probability can be improved after recommendation. For example, for all AMI patients (dataset 1), the predicted survival probability can be improved from 92.8% to 94.2% when the given distance limit is 50 miles. The average distance only increases from 20 to 28 miles. The CABG experiment (dataset 3) shows an interesting result. The improvement of average scores is the largest, but the average survival probability is not. The map for CABG in Figure 3.6 can explain the difficulty of improving survival probability for this dataset. For a surgery patient, not only survival but also FFC is very important. In multi-objective experiments, we combine these two targets for the hospital recommendation problem.

3.2.2 Multi-Objective Optimization

The best choice is the closest hospital with the highest survival and FFC probabilities. However, this is not the case for all people. Most people have to decide the trade-off among the three desired targets, and only the patient and his/her health

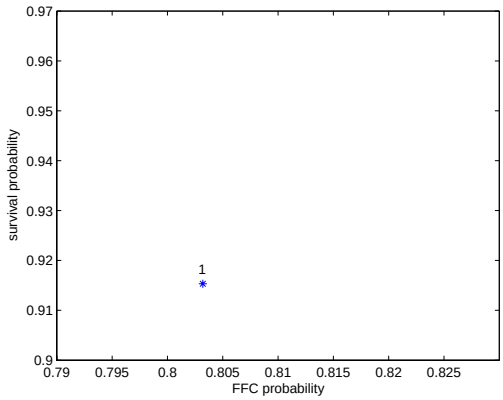
Table 3.4: Average predicted survival probabilities, average predicted scores, and distance comparison between originally chosen hospitals and recommended hospitals

dataset		Original Choice	Maximum distance (miles)			
			30	50	100	200
1	Avg dist	20	17	28	58	104
	Avg prob	92.8%	93.6%	94.2%	94.9%	95.2%
	Avg scr	0.575	0.714	0.830	0.973	1.047
2	Avg dist	19	17	27	58	104
	Avg prob	93.6%	94.3%	94.8%	95.4%	95.7%
	Avg scr	0.545	0.660	0.764	0.900	0.968
3	Avg dist	29	26	32	52	85
	Avg prob	95%	95.4%	95.7%	96.6%	97.2%
	Avg scr	0.3897	0.534	0.674	1.206	1.507

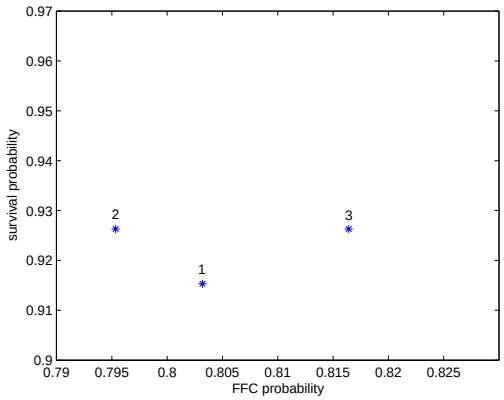
provider can decide the balance of these targets. In order to provide assistance efficiently, the system provides a list of hospitals with estimated information instead of recommending the single best choice. The result is presented by discussing a case study. In order to present results easily, we also use the distance limits 30, 50, 100, and 200 miles. Figure 3.7 shows each hospital in a location (x, y)=(FFC, survival) in each distance limit.

The patient P299 is an emergency case who is 70 years old. The original chosen hospital (hospital 1) is located at $(0.803, 0.915)$ in Figure 3.7a, and the distance to this hospital is 7 miles. Although hospital 1 has a large discharge and CABG surgery volume in a highly metropolitan area, both values of x and y are below average. The age and emergency admission can partially explain the low x and y. After entering 30 miles as the distance limit, the system shows three hospitals. Hospital 3 is a good

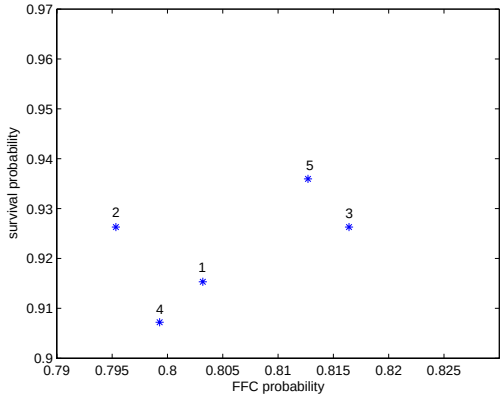
(a) The hospital of original choice



(b) Hospitals within 30 and 50 miles



(c) Hospitals within 100 miles



(d) Hospitals within 200 miles

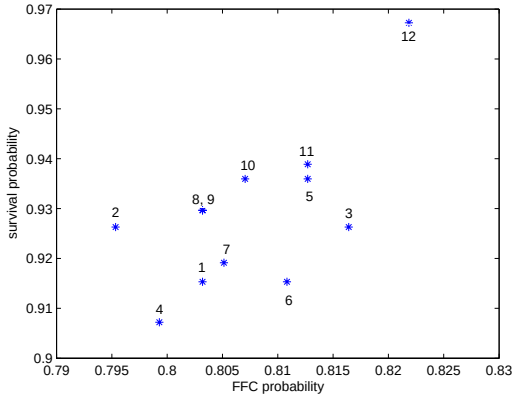


Figure 3.7: The visualization of hospitals: The solution space (hospital) can be demonstrated in location (x, y)=(freedom from complication probability, survival probability).

choice not only because of higher x and y but also the traveling distance is 8 miles.

Thirty (30) and fifty (50) mile limits result in the same search results, but 100 miles will result in 5 choices. There is a trade-off between survival and FFC in deciding between hospital 3 and 5. The emergency admission case of P299 may not choose the hospital 5 because of the distance of 75 miles. Hospital 5 shows a very interesting fact. It is located in rural area, and the volume is less than both hospitals 1 and 3. However, the 2004 SID data shows that this hospital did not have in-hospital death case for AMI patients who have the CABG surgery. Besides, the average age of patients (72.7 yrs) in this hospital is higher than hospitals 1 (67.2 yrs) and 3 (65.6 yrs). The average comorbidity score of patients (0.83) in this hospital is higher than hospitals 1 (0.47) and 3 (0.61). The frequency of emergency admissions (11/13) is also relatively higher than hospitals 1 (32/57) and 3 (50/64). These results show that although hospital 5 is a relatively smaller hospital in a rural area, sicker patients do well.

A searching range of 200 miles shows all hospitals that can perform CABG surgery in Iowa. Both x and y are the highest for hospital 12 which has the largest bed size and the largest total number of surgeries in Iowa. However, it is 110 miles away. The system can present complex information in a very efficient way, i.e., compile a patient's and institutional factors into a few numbers (personal survival, FFC probabilities, and travel distance). A user (can be patients, their family, health providers, or other people) can decide the best fit accordingly. For example, for patients in emergency, travel distance is the most important consideration. If two hospitals are close, we still can choose the best hospital between them. For some patients, they may want the best care using the helicopter emergency medical service transport, and they can certainly have more choices. The user can decide the best fit of hospital based on personal balance of importance among survival, complication, and distance.

CHAPTER IV
OPTIMAL DECISION PATH FINDER AND TRIAGE DIAGNOSIS PROBLEM

This chapter discusses another expert system that helps make cost-effective test selection based on an individual's information. To achieve this goal, Optimal Decision Path Finder (ODPF) algorithm is devised to construct the expert system that minimize necessary tests (and test costs) to cross treatment (or no-treatment) threshold.

Diagnostic decision making is based on experience and hypothetico-deductive reasoning [40]. In the face of diagnostic uncertainty, a health provider can either gather more evidence (test) or treat a patient [91, 80]. According to Bayesian theory, the clinician adjusts the likelihood of the disease in question with each new diagnostic test. The desired level of certainty depends largely on the consequences of inaccurate diagnosis; one generally needs to perform diagnostic tests until one has attained a treatment (or no treatment) threshold probability, i.e., the threshold of sufficient evidence [91, 40].

Each test result can revise the probability of disease in relation to the treatment (or no-treatment) threshold, but informativeness of the same test result may vary based on the pre-test (prior) probability, which in turn varies as a function of patient characteristics and other test results. In addition, test parameters, such as sensitivity and specificity may vary across different patient populations [75, 86, 55, 103].

For a given individual or patient subgroup, it would be highly desirable to identify the test sequence among all candidate sequences that optimizes the confidence of decision making while minimizing cost. In advance of knowing the possible test results for a given individual, one would like to identify the test whose result is most likely to approach the treatment (or no-treatment) threshold.

This project proposes a machine learning (ML)-based expert system approach, called Optimal Decision Path Finder (ODPF), to dynamically determine the test that is most likely to be informative in terms of diagnostic accuracy while minimizing the time and money spent on diagnostic testing. Two types of tests are considered, immediate and delayed tests. The first type of test such as blood pressure is routine and inexpensive, and the results can be known immediately. The second type of test is more costly, and the test results are not immediately available. This research focuses on the second type of test. This algorithm takes pre-test probability, interaction of variables, and the cost of each test into account and uses a greedy search to choose the test and generate an individualized test sequence.

4.1 Method

4.1.1 Model Design

Figure 4.1 shows a visual representation of the ODPF algorithm. Examining the confidence level of the diagnosis and selecting the next test are repeated in each step. Initially (step 1), we only have patients' symptoms and/or some immediately available test results. A lazy classifier is trained with these data and predicts whether or not a given patient has the disease of interest. The system then examines the prediction in relation to the treatment and no-treatment thresholds. If the prediction is sufficiently confident (see Figure 4.2), the system makes a diagnosis, and the process terminates. Otherwise, the system looks for a test that can facilitate the next prediction (in Step 2). Selection of the treatment and no-treatment thresholds can be determined by analysis of relative costs and benefits or by the health provider's intuitive estimation [91].

After the result of the selected test has been obtained, we train a new classifier with symptoms data and values of the selected test ($test(i)$) from the training cases.

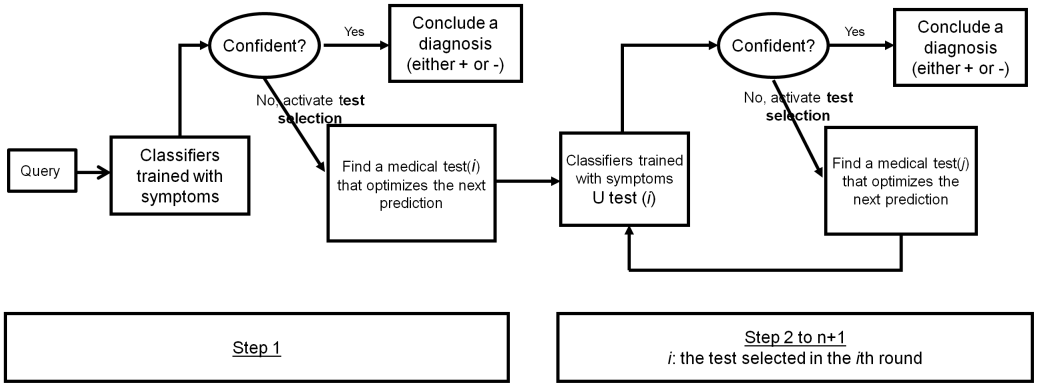


Figure 4.1: Summary of ODPF application

Then we examine the confidence of the prediction and, if necessary, select another new test. These two processes are repeated until a confident diagnosis occurs or until all options for testing have been exhausted, at which point a diagnosis is made.

We detail lazy learning, Locally Temporal Feature Selection(LTFS), inheritance strategies, missing values, multi-class prediction, unbalanced data, and description of datasets in the following sections.

4.1.2 Lazy Support Vector Machines (SVMs)

Support vector machines (SVMs) [97] are a popular predictive model that can avoid overfitting problems. Equation 4.1 shows the primal optimization model for training an SVM classifier. W is a weight vector, ϵ_i is the error of training case i, C is a given constant that controls the balance between error and model sparsity, y is the class label, x is the vector of predictive features, and b is the bias term for the decision function.

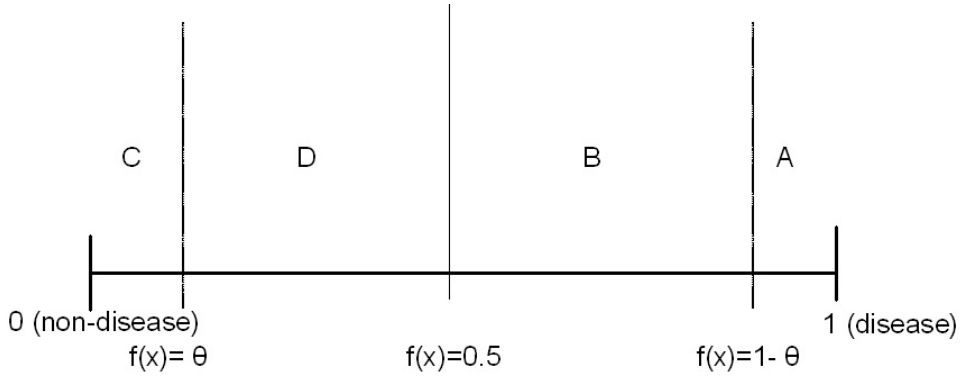


Figure 4.2: Confident diagnosis (prediction). $f(x) = 0.5$ represents a separation surface, which can distinguish presence from absence of the disease. A prediction with $f(x) > 0.5$ indicates a positive diagnosis (disease is present). There are two confidence thresholds $[1-\theta, \theta]$ for a decision of treatment or no-treatment. A diagnosis can be made only when the prediction of a patient's disease status crosses the threshold (area A or C). For example, if a patient's disease probability is located within area A, the patient should have treatment (or expected to have the disease of interest). If located within area C, the patient does not need to receive treatment. Otherwise, the patient needs more testing to support the diagnosis (area B or D).

$$\begin{aligned} \underset{W,\epsilon}{\text{minimize}} \quad & \langle W \cdot W \rangle + C(\sum_{i=1}^{l} \epsilon_i) \\ \text{subject to} \quad & y_i(\langle W \cdot x_i \rangle + b) \geq 1 - \epsilon_i, \ i = 1, \ldots, l \\ & \epsilon_i \geq 0, \ i = 1, \ldots, l \end{aligned} \tag{4.1}$$

A typical implementation of SVMs solves the dual of this problem and induces non-linearity in the separating surface using a kernel function. See [97] for details.

$$\begin{aligned} \underset{W,\epsilon}{\text{minimize}} \quad & \langle W \cdot W \rangle + (\sum_{i=1}^{l} C_i \epsilon_i) \\ \text{subject to} \quad & y_i(\langle W \cdot x_i \rangle + b) \geq 1 - \epsilon_i, \ i = 1, \ldots, l \\ & \epsilon_i \geq 0, \ i = 1, \ldots, l \end{aligned} \tag{4.2}$$

In this project, we use lazy SVMs, which learn a decision function specifically for the query case, as the base learners. Equation 4.2 shows the modified optimization

formulation when training a lazy SVM. The only difference between Equations 4.2 and 4.1 is C. In Equation 4.2, the value of C is different for each training case i. When C_i is large for the training case i, case i gets more attention when optimizing, reducing the chance of error for case i. In both equations, $(\langle W \cdot x_i \rangle + b)$ is the decision function $d(x)$ that classifies whether or not a patient has the disease of interest. In $d(x)$, W represents the degree of importance of variables and b represents the bias term.

The value of C_i is determined by the similarity to the query, which is solely based on the results of the medical tests. An extreme example is when all test values of a training case are exactly the same as the query, this training case will have a very large C_i. On the other hand, when a case has no test value matching the query, the case will have a small C_i.

We use a simple algorithm to update instance weight (C_i) as illustrated in Figure 4.3, which shows a query and three training cases. γ (≥ 1) is the multiplier for updating instance weights. The initial weight for all training cases is 1 because we do not have any prior knowledge of these training cases. In step 1, the query does not pass the confidence threshold and we select Test A. After the test result has been determined, only training case 3 has the same value as the query. Thus, its instance weight (C_3) is multiplied by a factor of γ (in this example, we use $\gamma = 2$). In step 2, the values of training cases 2 and 3 have the same value as the query, and their instance weights (C_2 and C_3) are multiplied by 2. In step 3, only C_2 and C_3 are multiplied again by 2.

After the selected test result has been obtained in each step, we can update the instance weights for all training cases. More similar cases have higher weight, making the training case population more specific to the query after more tests results are known. In our experiments, the multiplier γ is decided empirically based on predictive performance and costs.

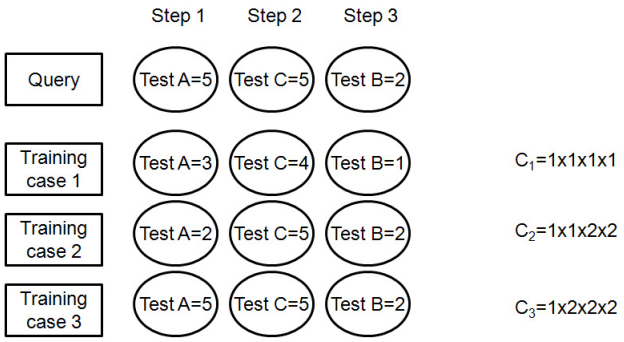


Figure 4.3: Instance weight update example

4.1.3 Speed-Based and Cost-Based Evaluation Functions

As discussed in 2.2.4, most feature selection methods focus on global features and do not take the order of including features into account. For sequential diagnosis problems, Locally Temporal Feature Selection (LTFS) is devised to specifically select features (medical tests) for a query based on features at hand (known information, e.g., symptoms and results of selected tests). We use LTFS to guess the next test to speed up diagnosis, so that only tests whose order needs to be determined are considered as features in LTFS (e.g. delayed tests). The degree of importance of a feature may change with different known patient information, demographic data, history, physical findings, and previously selected tests results.

One important property of LTFS is the sequence of features (tests). The symptoms and demographic data of each patient varies, so the timing of the same test for two patients may differ. For example, when the predicted probability of disease is very close to either 0 or 1 but has not crossed either threshold, most tests can help to cross the treatment (or no-treatment) threshold easily. However, if the prediction is close to 0.5, a very strong predictor may be necessary. The test chosen next in these two cases can be very different. As a result, each case will have an unique test sequence because the known information of each patient varies.

There are two selection strategies used with LTFS: speed-based and cost-based. The test selection strategy is represented by the evaluation function f. The selected feature can provide the most information for the query. For speed-based strategies, information means the speed of the next prediction moving toward a predicted probability of either 0 or 1. For cost-based strategies, information means the most cost-efficient choice, considering not only speed but also the cost of a test.

We consider four evaluation functions in each category of selection strategy. For the speed-based category, these evaluation functions are probability contribution, minimum uncertainty, expected uncertainty, and instance weight-expected uncertainty. Each function has a corresponding cost-based evaluation function.

We begin by describing the speed-based approaches. When determining a test, we have to consider which test is most likely to lead to diagnostic certainty (with a post-test probability approaching either 0 or 1). We cannot know the actual value of a test that has yet to be performed, but we can compute the predicted probability of diseases associated with all possible values of that test. We can then find the most promising test that can approach either end the fastest.

The probability contribution (PC) function finds the test that produces the greatest changes in predicted probability of disease. For all possible values of test i, f is the maximum difference of probabilities. This function is defined as follows:

$$f_{PC}(\overline{v} \cup u_i) = \max(h(d(\overline{v} \cup u_i^k)) - h(d(\overline{v} \cup u_i^{k'}))), \forall k, k' \in \{1, \ldots, |u_i|\}, k \neq k',$$

where $|u_i|$ is the number of possible values for the feature u_i, $\overline{v}$ is the known information of the query including symptoms, demographic data, and previously selected test results. u_i^k is the value k of test i (unknown information), k is the index of values of the test u_i. Once the result of the selected test is known, it becomes known information. d is the predictive function that returns SVM scores, and h transform SVM scores into probabilities.

The idea of minimum uncertainty (MU) is to find the test that would push the

prediction closest to either 0 or 1. We define uncertainty as simply the minimum difference between the predicted probability and zero or one. This function is defined as

$$f_{MU}(\overline{v} \cup u_i) = \min\ (0.5 - |h(d(\overline{v} \cup u_i^k)) - 0.5|), \forall k, k' \in \{1, \ldots, |u_i|\}.$$

Compared to minimum uncertainty, the expected uncertainty (EU) is a more stable strategy. The uncertainty of each value of a test is weighted by its frequency in the training set. We find the test with minimum f. The expected uncertainty is defined as

$$f_{EU}(\overline{v} \cup u_i) = \sum_k \frac{(0.5 - |h(d(\overline{v} \cup u_i^k)) - 0.5|) \times freq_i^k}{\sum_k freq_i^k},$$

where $freq_i^k$ is the frequency of training data with the value u_i^k.

For the Instance Weight-expected uncertainty (IWEU), we use instance weights to replace frequency. The uncertainty of value u_i^k is weighted by the instance weights based on previous tests. Similar to expected uncertainty, we find the test with the minimum f. The formulation can be expressed as

$$f_{IWEU}(\overline{v} \cup u_i) = \sum_k \frac{(0.5 - |h(d(\overline{v} \cup u_i^k)) - 0.5|) \times C_i^k}{\sum_k C_i^k}.$$

The last three functions find the test with the smallest uncertainty based on a single (minimum uncertainty) or average (expected uncertainty and IW-expected uncertainty) test result(s). Locations of predictions provide information to guess which test is most likely to move the probability of disease upward or downward. The first function (f_{MU}) finds the test with some value that has the global minimum uncertainty, and the last two functions use expected uncertainty to select a test.

In the cost-based strategy, cost is defined as test cost while effectiveness is the degree of movement toward either end (as in the speed-based strategy). We use the ratio $\frac{effectiveness}{cost}$ instead of only $effectiveness$.

Each speed-based evaluation function has a corresponding cost-based function. The cost-based version of probability contribution becomes probability contribution per dollar. We want to select the test with maximum f. The cost-based objective

function is

$$f_{costPC}(\overline{v} \cup u_i) = \frac{\max(h(d(\overline{v} \cup u_i^k)) - h(d(\overline{v} \cup u_i^{k'})))}{cost_i}, \forall k, k' \in \{1, \ldots, |u_i|\}, k \neq k',$$

where $cost_i$ is the cost of using test i.

All cost-based functions in the uncertainty family become uncertainty reduction per dollar. For uncertainty reduction, we need to compute the reduction of uncertainty from known information $\overline{v}$. The uncertainty of known information is defined as $UC(\overline{v}) = (0.5 - |h(d(\overline{v})) - 0.5|)$. Cost-based minimum uncertainty reduction is

$$f_{costMU}(\overline{v} \cup u_i) = \frac{UC(\overline{v}) - (0.5 - |h(d(\overline{v} \cup u_i^k)) - 0.5|)}{cost_i}, \forall k, k' \in \{1, \ldots, |u_i|\}$$

We select the test with maximum f because we want to select the test with maximum uncertainty reduction per dollar. Cost-based expected uncertainty reduction is

$$f_{costEU}(\overline{v} \cup u_i) = \frac{UC(\overline{v}) - (\sum_k \ (0.5 - |h(d(\overline{v} \cup u_i^k)) - 0.5|) \times freq_i^k) / \sum_k freq_i^k}{cost_i},$$

and we want to find a test with the maximum f.

Similarly, cost-based IW expected uncertain reduction is

$$f_{costIWEU}(\overline{v} \cup u_i) = \frac{UC(\overline{v}) - (\sum_k \ (0.5 - |h(d(\overline{v} \cup u_i^k)) - 0.5|) \times C_i^k) / \sum_k C_i^k}{cost_i}.$$

4.1.4 The ODPF Algorithm

Algorithm 1 summarizes the whole process. The input dataset D consists of patients' known information V, such as symptoms and known results of tests, and delayed tests (or unknown information) U. Initially, this study assigns all immediate tests to V based on the definition of Turney [94]. When the result of U_j is known, this test becomes known information. θ is a confidence threshold in the range of 0 to 0.5. Thus, $1 - \theta$ represents a treatment threshold and θ represents a non-treatment threshold. The limitations of the $[\theta, 1 - \theta]$ threshold structure, along with a remedy, are discussed in Section 4.1.7.

The query with known information $\overline{v}$ activates the recommendation process. We want the algorithm to recommend a test at each step i after obtaining the previous

Algorithm 1 ODPF

Input :

1 Data D, $D = V \cup U_j$, $j \in \{tests\}$, $k \in$ data points

2 Threshold θ and weight update γ

3 Query $\overline{v}$

Output :

1 Test sequence SEQ before θ is met

2 Prediction

$C = \mathbf{1}$
$UT = \{\text{tests}\}$
$SEQ = \emptyset$
for $j = 1 \ldots |tests|$ **do**
  $[\hat{i}, h, d] = ChooseClassifier(D, \overline{v}, C, UT)$
  $UT = UT/\hat{i}$ and $U = U/U_{\hat{i}}$
  $\overline{v} = \overline{v} \cup \overline{u}_{\hat{i}}$ and $V = V \cup U_{\hat{i}}$
  $IW_k = IW_k * \gamma, \forall k : u_{k\hat{i}} = \overline{u}_{\hat{i}}$
  $SEQ = SEQ\&\hat{i}$
  if $h(d(\overline{v})) < \theta$ or $1 - h(d(\overline{v})) < \theta$ **then**
    return $[h(d(\overline{v})), SEQ]$
  end if
end for
return $[h(d(\overline{v})), SEQ]$

ChooseClassifier$(D, \overline{v}, C, UT)$:
$\hat{i} = 1$
for $i = 1 \ldots |UT|$ **do**
  $d = TrainPredictor(V \cup U_i, C)$
  $h = TrainCalibrationFunction(V \cup U_i)$
  if $f(\overline{v} \cup \overline{u}_i) > f(\overline{v} \cup \overline{u}_{\hat{i}})$ **then**
    $\hat{i} = i, h^* = h, d^* = d$
  end if
end for
return $\hat{i}, h^*, d^*$

test result $\overline{u}_{\hat{i}-1}$, and decide when to stop. At termination, the output is the diagnostic decision and the sequence of previously recommended tests.

In step 1, we set the instance weight (C) of each case to be the same. Step 2 initializes the pool of unused tests (UT) to the set of all tests, and Step 3 initializes the test sequence (SEQ).

Steps 4 to 13 are the repeating processes for confident diagnosis and test selection. Step 5 computes the best test $\hat{i}$, the corresponding calibration function h (Equation 2.3), and the decision function d (Equation 4.2). Step 6 removes $\hat{i}$ from UT and removes $U_{\hat{i}}$ from U. After the test value of $\hat{i}$ is revealed, Step 7 adds its value $\overline{u}_{\hat{i}}$ to $\overline{v}$ because the test result $\overline{u}_{\hat{i}}$ has become known information for the query case. Also, the column of training features $U_{\hat{i}}$ will be added to V. We do not add or remove data from the dataset D, but, in each iteration, we move a certain feature from U to V. Step 8 updates instance weights. When the result of the selected test for a training case k matches the query case, the instance weight of k is updated. Step 9 appends test $\hat{i}$ to SEQ.

Steps 10 to 12 decide whether the prediction is confident enough. If the answer is positive, the algorithm will return the diagnosis and the test sequence. Otherwise, the process repeats until all tests have been run.

For the test searching subroutine, Step 15 assigns the first test as the selected one. Then, Steps 16 to 22 update the selection. In each iteration, a trial dataset, which consists of training features (V, corresponding to the known information of the query) and an unused test feature (U_i), is used to train a decision function d (Step 17) and a calibration function h (Step 18).

Steps 19 to 21 use a function f to decide the best test. When test i is better than test $\hat{i}$, we record d^* and h^* of the new $\hat{i}$. Step 23 returns $\hat{i}$, d^*, and h^* of the best test.

4.1.5 Speed-Up Strategy: Inheritance

The ODPF algorithm uses a large number of classifiers to select the best test (or sequence of tests). Training these classifiers is computationally expensive. To reduce this problem, we allow a query to be able to inherit classifiers trained from a previous query. A new query case can always share the first group of classifiers to identify the first test because C is 1 for all cases. After selecting the first test, if the test result of the query is the same as a previous query, the query can inherit classifiers from that previous query.

Figure 4.4 illustrates the rule of inheritance. Unlike decision tree algorithms, a node represents a group of classifiers that determine the next test; one can also make a decision at this node. Instead of explicit rules, the choice of test is driven by classifiers. Query 1 has to train all classifiers for identifying tests. Test 5 is the first selected test. After performing this test, we obtain the value 1 for test 5. Next, test 6 is suggested by the second group of classifiers.

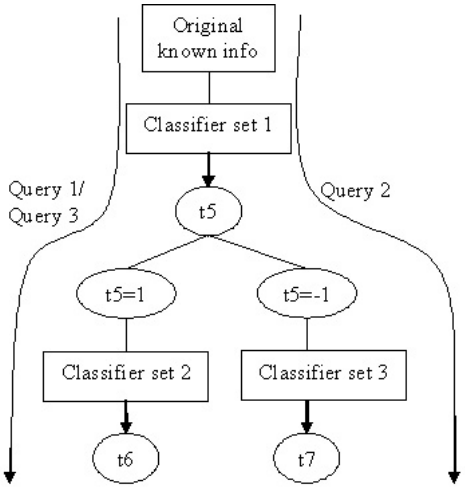


Figure 4.4: The rules of inheritance

Query 2 can inherit the first group of classifiers. Test 5 was again selected but

the value is -1. Query 2 is ineligible to inherit any more because lazy classifiers are trained specifically for a query. Thus, a query (e.g., query 2) can inherit a classifier set only when all previous tests and their values are exactly the same as a previous query (e.g., query 1). Therefore, we have to train a new group of classifiers (Classifier set 3) for query 2 in order to identify the next test. For query 3, not only is test 5 the first selected test but also the value is the same as query 1. Thus, query 3 can inherit both groups of classifiers from query 1.

4.1.6 Missing Values

Missing values are common in medical datasets, especially those recording diagnostic tests, since a health provider will not run tests considered to be unnecessary. Therefore, ODPF must adjust for missing values in both training and validation. First, for training, we impute a missing value using the class-conditional mean of the feature. Second, the C update depends on the proportion of the matching value in the training set. For example, consider a test with three possible values, 1, 2, and 3, whose proportions are 30%, 30%, and 40% in the training set. If the new test value of the query is 3, for example, each training instance with a missing value of this feature can be updated, but the instance-weight multiplier becomes $1 + (\gamma - 1) \times 40\%$.

When we apply ODPF in an actual clinical setting, all test values of a query (a new patient) are supposed to be missing, and we do not have any problem for the query with missing values. However, when validating ODPF, we need to slightly modify the query case with some missing values in order to validate reasonably. When some test values of a query are missing, we limit the selection of tests to those without missing values. When a diagnosis is highly uncertain, all available tests (without missing values) may not allow one to cross a probability threshold. In this case, the diagnosis must be made after receiving values of these available tests.

4.1.7 Unbalanced Data Adjustment

In some situations (e.g., a rare disease), the separation surface of the trained SVM will not correspond to a probability of 0.5. Therefore, unequal thresholds, such as $[\theta_1, \theta_2]$, may be better than equal thresholds $[\theta, 1-\theta]$.

The definition of uncertainty also needs to change in an unbalanced dataset. When a dataset is balanced, uncertainty is determined using either 0 or 1 as a reference. In an unbalanced dataset, uncertainty is decided using given thresholds instead of 0 or 1 as a reference. In an unbalanced dataset, we are still looking for a test with the smallest uncertainty in minimum uncertainty and (IW-)expected uncertainty functions.

In the previous definition, the smallest uncertainty is 0. In the new definition, when a test with some set of values crosses the threshold, its uncertainty is less than 0. In this case, one can still identify the test with the lowest value of uncertainty, and we still can use all speed-based and cost-based test selection functions as described in Section 4.1.3.

4.1.8 Multi-Class Strategies

Many diagnosis problems involve a choice among several candidate conditions instead of a single disease. We show that the ODPF algorithm can be adjusted for multi-class decision problems. Similar to binary problems, we wish to find a test that can approach one possible outcome (class) most quickly. The difference is that we are comparing several possible outcomes instead of two, but we can still use the test-selection functions described in Section 4.1.3. For example, in a binary problem, we are finding the test with the smallest uncertainty from either end (two possible outcomes). In a multi-class problem, we are finding the test with the smallest uncertainty from one of the outcomes.

There are several algorithms for solving multi-class problems [53] using SVMs.

We use the adapted one-against-all method as the base classifiers. Assuming n possible classes (candidate conditions), we need to train $n-1$ classifiers with $n-1$ sets of labels. Each set of labels indicates whether or not a record belongs to a specific class. Each class corresponds to a set of labels except for one class. [1] If the decision function of only one class is greater than zero, this class is the predicted class. If the decision function of all classes are less than zero, the majority class is the predicted class. When there is more than one class with decision function greater than zero, the predicted class is the class with the highest predicted probability.

4.1.9 Dataset and Variables Design

This project has applied ODPF to several datasets of two types: diagnosis and prediction of future risk. Each Dataset is described as follows.

4.1.9.1 Datasets for Diagnosis

We apply the ODPF algorithm to heart disease and thyroid datasets. Both datasets are obtained from the UCI machine learning repository [5]. The description of all datasets (Tables 4.1 to 4.4) is taken from [94]. In these tables, tests can be categorized as delayed or non-delayed. Non-delayed can be obtained immediately and less costly. For example, a patient is asked about age, sex, and chest pain type (Table 4.1), which cost a nominal charge $1. In this project, we are more interested in delayed tests, and all non-delayed tests are known information used to determine the selection of delayed tests.

Heart Disease dataset [29]: This analysis sample included 303 consecutive patients (mean age 54 years, 68% male) who were referred for coronary angiography at the Cleveland Clinic between May 1981 and September, 1984. No patient had a history of prior myocardial infarction or known valvular disease or cardiomyopathy.

[1] In this project, we choose the majority class.

All patients received a history and physical examination, resting electrocardiogram, and laboratory studies, as part of their routine evaluation. Study patients also underwent exercise stress testing, myocardial perfusion imaging, and cardiac fluoroscopy as part of a research protocol. The dataset included 4 clinical variables (age, sex, chest pain type, and systolic blood pressure), 2 laboratory variables (serum cholesterol and fasting glucose), and resting electrocardiographic variables (ST segment depression > 0.05 mV or T-wave inversions, probable or definite left ventricular hypertrophy based on Estes' criteria). The diagnosis of coronary artery disease was defined by the presence of $> 50\%$ narrowing of one or more coronary arteries on angiography. All coronary angiograms were interpreted by a cardiologist who was blinded to noninvasive test results.

The heart disease dataset came from Cleveland Clinic Foundation and was provided by the principal investigator Robert Detrano of the V.A. Medical Center, Long Beach and Cleveland Clinic Foundation. In this experiment, the dataset was downloaded from the Statlog project, in which 6 data points were discarded because of missing class value and 27 data points were retained in case of dispute [74], at the UCI machine learning repository. Originally, there were four different types of disease in this dataset. In this project, the classification task is simply to distinguish the presence of disease (all four types) from its absence. The dataset does not have missing values. Variables and costs are described in Table 4.1.

Thyroid Disease dataset [83]: This analysis sample consists of 3,772 cases that were referred to the Garvan Institute of St. Vincent's Hospital, Sydney, Australia for diagnostic evaluation starting in late 1984. The dataset includes 11 clinical attributes, which were abstracted from data provided by the referring health provider (age, sex, pregnancy, goiter, tumor, hypopituitarism, use of thyroid or antithyroid medication, history of thyroid surgery, complaint of malaise, psychological

Table 4.1: Description for heart disease dataset

	Test	Description	Group	Cost	Delayed
1	age	age in years		$1.00	no
2	sex	patient's gender		$1.00	no
3	cp	chest pain type		$1.00	no
4	trestbps	resting blood pressure		$1.00	no
5	chol	serum cholesterol	A	$7.27 if first test in group A; $5.17 otherwise	yes
6	fbs	fasting blood sugar	A	$5.20 if first test in group A; $3.10 otherwise	yes
7	restecg	resting electrocardiograph		$15.50	yes
8	thalach	maximum heart rate achieved	B	$102.90 if first test in group B; $1.00 otherwise	yes
9	exang	exercise induced angina	C	$87.30 if first test in group C; $1.00 otherwise	yes
10	oldpeak	ST depression induced by exercise relative to rest	C	$87.30 if first test in group C; $1.00 otherwise	yes
11	slope	slope of peak exercise ST segment	C	$87.30 if first test in group C; $1.00 otherwise	yes
12	ca	number of major vessels coloured by fluoroscopy		$100.90	yes
13	thal	3 = normal; 6 = fixed defect; 7 = reversible defect	B	$102.90 if first test in group B; $1.00 otherwise	yes
14	num	diagnosis of heart disease		diagnostic class	-

symptoms), and up to six test results (including TSH, T3, TT4, T4U, free thyroxine index, and TBG) requested by the referring health provider. Final diagnosis of hyperthyroidism, hypothyroidism, or euthyroid status was based on interpretation of all available clinical and laboratory data by a qualified endocrinologist (or in some cases, an expert system designed to diagnose thyroid disease). We follow Turney's paper [94] using four tests (TSH, T3, TT4, and T4U) because costs are provided for only those tests. The resulting dataset does not include missing values. Variables and costs are described in Table 4.2.

4.1.9.2 Datasets for prediction of future risk

We have also performed an analysis of two additional datasets in order to show the potential applicability of the ODPF method in predicting the future risk of disease or adverse events with fewer tests by determining an optimum patient-specific sequence. Both datasets are also obtained from the UCI machine learning repository [5]. These datasets are briefly described below:

Pima Indians diabetes dataset [89]: The dataset was collected by the National Institute of Diabetes and Digestive and Kidney Diseases. All subjects were females at least 21 years old of Pima Indian heritage. The dataset includes 6 clinical variables (age, diabetes pedigree function, body mass index, triceps skin fold thickness, diastolic blood pressure, and number of pregnancies) and two tests (glucose tolerance test and serum insulin test) to classify the risk of diabetes. Variables and costs are described in Table 4.3.

Hepatitis dataset [31]:. The dataset includes 14 clinical attributes (age, sex, patient on steroids, antiviral therapy, fatigue, malaise, anorexia, hepatomegaly, palpable firmness of liver, palpable spleen, presence of spider veins, ascites, presence of varices, and liver histology), laboratory tests (bilirubin, alkaline phosphotase, aspartate aminotransferase, albumin, and protime), and the prognostic outcome of disease

Table 4.2: Description for thyroid dataset

	Test	Description	Group	Cost	Delayed
1	age	age in years		$1.00	no
2	sex	gender		$1.00	no
3	on thyroxine	patient on thyroxine		$1.00	no
4	query thyroxine	maybe on thyroxine		$1.00	no
5	on antithyroid	on antithyroid medication		$1.00	no
6	sick	patient reports malaise		$1.00	no
7	pregnant	patient pregnant		$1.00	no
8	thyroid surgery	history of thyroid surgery		$1.00	no
9	I131 treatment	patient on I131 treatment		$1.00	no
10	query hypothyroid	maybe hypothyroid		$1.00	no
11	query hyperthyroid	maybe hyperthyroid		$1.00	no
12	lithium	patient on lithium		$1.00	no
13	goitre	patient has goitre		$1.00	no
14	tumour	patient has tumour		$1.00	no
15	hypopituitary	patient hypopituitary		$1.00	no
16	psych	psychological symptoms		$1.00	no
17	TSH	TSH value, if measured	A	$22.78 if first test in group A; $20.68 otherwise	yes
18	T3	T3 value, if measured	A	$11.41 if first test in group A; $9.31 otherwise	yes
19	TT4	TT4 value, if measured	A	$14.51 if first test in group A; $12.41 otherwise	yes
20	T4U	T4U value, if measured	A	$11.41 if first test in group A; $9.31 otherwise	yes
21	FTI	FTI–calculated from TT4 and T4U		not used	-
22	class	diagnostic class		diagnostic class	-

Table 4.3: Description for diabetes dataset

	Test	Description	Group	Cost	Delayed
1	times pregnant	number of times pregnant		$1.00	no
2	glucose tol	glucose tolerance test	A	$17.61 if first test in group A; $15.51 otherwise	yes
3	diastolic bp	diastolic blood pressure		$1.00	no
4	triceps	triceps skin fold thickness		$1.00	no
5	insulin	serum insulin test	A	$22.78 if first test in group A, $20.68 otherwise	yes
6	mass index	body mass index		$1.00	no
7	pedigree	diabetes pedigree function		$1.00	no
8	age	age in years		$1.00	no
9	class	diagnostic class		diagnostic class	-

(live or die). Variables and costs are described in Table 4.4.

4.2 Results

In this section, we show results for the heart disease, thyroid, diabetes, and hepatitis datasets. All results are generated using RBF-SVM and averaged over five 10-fold cross-validation (CV) runs. We use the 10-fold CV to replace leave-one-out [92] in order to benefit from inheritance as described in Section 4.1.5. In other words, each data point is a query which can share classifiers with another query in the same testing fold. The probability transformation function (Equation 2.3) is trained in 3-fold CV runs.

We used over-sampling of the minority class [67] to balance each dataset according to the proportion of positive to negative cases. For example, for a dataset with 95% positive cases, we used the ratio 1/19 to balance the positive and negative classes.

Table 4.4: Description for hepatitis dataset

	Test	Description	Group	Cost	Delayed
1	class	prognosis of hepatitis		prognostic class: live or die	-
2	age	age in years		$1.00	no
3	sex	gender		$1.00	no
4	steroid	patient on steroids		$1.00	no
5	antiviral	patient on antiviral		$1.00	no
6	fatigue	patient reports fatigue		$1.00	no
7	malaise	patient reports malaise		$1.00	no
8	anorexia	patient anorexic		$1.00	no
9	liver big	liver big on physical exam		$1.00	no
10	liver firm	liver firm on physical exam		$1.00	no
11	spleen palpable	spleen palpable on physical		$1.00	no
12	spiders	spider veins visible		$1.00	no
13	ascites	ascites visible		$1.00	no
14	varices	varices visible		$1.00	no
15	bilirubin	bilirubin–blood test	A	$7.27 if first test in group A; $5.17 otherwise	yes
16	alk phosphate	alkaline phosphotase	A	$7.27 if first test in group A; $5.17 otherwise	yes
17	sgot	aspartate aminotransferase	A	$7.27 if first test in group A; $5.17 otherwise	yes
18	albumin	albumin–blood test	A	$7.27 if first test in group A; $5.17 otherwise	yes
19	protime	protime–blood test	A	$8.30 if first test in group A; $6.20 otherwise	yes
20	histology	was histology performed?		$1.00	no

In our cost analyses, we apply a group discount when other tests belonging to the same group have already been requested. For example, blood-related tests have a group price because they share the cost of phlebotomy.

We use $\gamma = 1.4$ for constructing lazy SVM classifiers. The parameter was determined by predictive performance. For instance, the total accuracies of $\gamma = 1, 1.4, 1.8$ in probability contribution and minimum uncertainty functions are [0.837, 0.8407, 0.8185] and [0.8556, 0.8593, 0.8333], respectively. SVM classifiers turn into lazy SVM classifiers when γ is greater than 1. With an appropriate γ, the predictive performance of lazy learning can be improved.

We discuss our results in detail using the heart disease dataset. We demonstrate the multi-class result on the thyroid dataset with the minimum uncertainty function and report aggregated results of diabetes and hepatitis using all LTFS functions.

4.2.1 Heart Disease Dataset

Figure 4.5 summarizes the process to evaluate the model. We train and predict for each query patient (A). If a prediction does not cross either treatment or non-treatment threshold, more testing is needed (from B to C). We can utilize several different test selection functions in C, e.g., minimum uncertainty. One test will be appended to the sequence each time once goes through the A,B,C,A cycle. If the prediction crosses either threshold, a diagnosis is made (D) and validated.

Tables 4.5 and 4.6 summarize the diagnosis accuracy and cost saving of four evaluation functions. The notation ite1 to ite9 represents the number of delayed tests obtained by patients. Ite1 represents only 1 test performed, while ite4 represents 4 tests performed. Ite0 is the situation where a diagnosis (prediction) is made without obtaining any delayed tests. The sequence of tests of patients may vary, so we can not show actual test sequences (e.g., [test5, test3, test1, ...]) of all patients in these aggregated results. Results in these two tables are predictive results at different

stopping points in the test sequence.

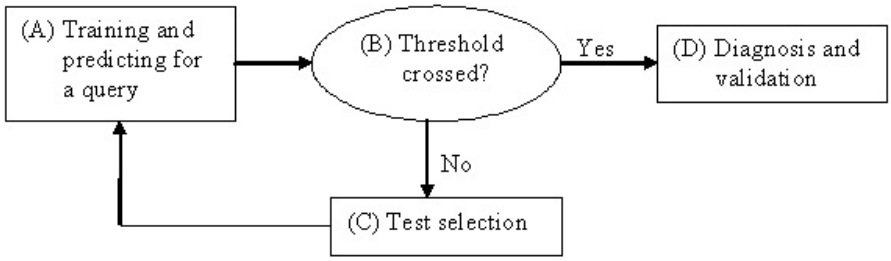


Figure 4.5: Evaluation Process

The stopping point for patients varies because the number of tests needed to diagnose is based on the individual patient. Some patients need few tests while others need many to support a diagnostic decision. Thus, patients may stop with any number of tests, and we summarized the aggregated results in ten strata (ite0 to ite9). When a testing stopped, we add one to the Total row, and compare the prediction (or diagnosis) with its class label (which shows whether the diagnosis was correct or not) at this stopping point. If the prediction is correct, one will be added to the Correct row. The percentage of correct predictions is shown in the % row. Total accuracy is the expected accuracy of all strata. "Average test" is the average number of tests used, and cost saving is the average cost saving of unused tests for all patients.

These two tables also show the accuracy of diagnosis in each iteration. Some patients do not need to receive any testing (ite0), but other patients need to receive all tests (ite9). The groups in ite0 to ite9 correspond to patients with increasing difficulty of diagnosis. In other words, when a case stops early (ite0 to ite8), its predictive probability is far away from the separation surface and satisfies the stopping criterion. On the other hand, the patients in ite9 remain close to the separation surface. Thus, the accuracies for early diagnosis patients tend to be high (ite0 to ite8).

Table 4.5: Speed-based ODPF on heart disease data

Functions	**Statistics**	**ite0**	**ite1**	**ite2**	**ite3**	**ite4**	**ite5**	**ite6**	**ite7**	**ite8**	**ite9**
ProbContr	Correct	238	146	77	109	127	71	45	50	44	219
	Total	265	180	83	120	152	82	50	62	51	305
	%	0.898	0.811	0.928	0.908	0.836	0.866	0.9	0.806	0.863	0.718
Total accuracy=0.834; average test= 4.16^{+}; cost saving= 53.13^{+}%											
minUC	Correct	238	374	146	129	36	34	16	17	13	147
	Total	265	416	157	153	47	38	26	23	13	212
	%	0.898	0.899	0.930	0.843	0.766	0.895	0.615	0.739	1	0.693
Total accuracy=0.852^{+}; average test= 2.89^{+}; cost saving= 49.59^{+}%											
expUC	Correct	238	186	218	97	77	55	37	26	20	179
	Total	265	220	232	111	97	62	50	32	23	258
	%	0.898	0.845	0.940	0.874	0.794	0.887	0.740	0.813	0.870	0.694
Total accuracy=0.8393; average test= 3.51^{+}; cost saving= 51.52^{+}%											
IWexpUC	Correct	238	188	210	100	79	53	37	30	19	178
	Total	265	220	223	114	100	62	51	37	21	257
	%	0.898	0.855	0.942	0.877	0.790	0.855	0.725	0.811	0.905	0.693
Total accuracy=0.839; average test= 3.53^{+}; cost saving= 51.23^{+}%											

Note: The treatment and non-treatment threshold are [0.85, 0.15]. The accuracy of the baseline trained with 9 tests is 0.8407. + or - indicates that ODPF is significantly better or worse than baseline, respectively ($\alpha = 5\%$).

For patients who remain in the ite9 group, one still needs to make a diagnosis for these patients once all available tests have been exhausted. As these patients are difficult to diagnose, the diagnostic accuracy for the ite9 group is low. Interestingly, the threshold plays the role of a filter by placing patients in the appropriate stratum.

Both tables use + and - to indicate whether a number is significantly better or worse than the baseline, in which all tests were performed ($\alpha = 5\%$). The accuracy of the baseline is 0.8407. Although total accuracies of most functions are lower than baseline, the differences except for the cost-based minimum uncertainty (minUC) function are not significant. However, the accuracy of minUC function (0.852) is significantly better than the baseline. Average number of tests and cost saving of all functions are significantly better than the baseline. The number of tests required ranged from 2.89 to 4.73 while most cost savings are more than 50%. In general, speed-based strategies diagnose with fewer tests than cost-based strategies, and cost-based strategies save more cost than speed-based strategies. This result may vary based on datasets.

Figure 4.6 compares accuracies of three thresholds that use the minimum uncertainty strategy. The x-axis represents the stopping points. Standard refers to the baseline classifier which is trained with all tests. A larger threshold has greater accuracy in all strata. However, total accuracies for thresholds 0.75, 0.85, and 0.95 are 0.813, 0.852, and 0.847 respectively. In other words, a higher threshold does not always result in higher total accuracy. This is due to the effect of early diagnosis.

Figure 4.7 summarizes the frequency of testing across the three thresholds. Many patients can be diagnosed early (i.e., before exhausting all tests). As expected, the barrier of a higher threshold results in fewer patients being diagnosed early. Therefore, more patients remained until ite9 when the threshold is higher. Total accuracy is an expected accuracy which results from the combination of accuracy and frequency in each stratum. The accuracies of early diagnostic strata (ite0 to ite8) are generally

Table 4.6: Cost-based ODPF on heart disease data

Functions	**Statistics**	**ite0**	**ite1**	**ite2**	**ite3**	**ite4**	**ite5**	**ite6**	**ite7**	**ite8**	**ite9**
ProbContr	Correct	238	89	70	104	152	68	67	64	44	226
	Total	264	111	79	113	181	77	74	80	55	316
	%	0.902	0.802	0.886	0.920	0.840	0.883	0.905	0.800	0.800	0.715
Total accuracy=0.831; average test= 4.45^{+}; cost saving= 53.38^{+}%											
minUC	Correct	238	35	82	71	203	73	53	83	21	258
	Total	265	43	93	77	223	90	62	118	24	355
	%	0.898	0.814	0.882	0.922	0.910	0.811	0.855	0.703	0.875	0.727
Total accuracy=0.827^{-}; average test= 4.7311^{+}; cost saving= 54.09^{+}%											
expUC	Correct	238	59	84	88	218	70	79	40	32	214
	Total	265	66	101	106	230	88	105	49	40	300
	%	0.898	0.894	0.832	0.830	0.948	0.795	0.752	0.816	0.800	0.713
Total accuracy=0.831; average test= 4.40^{+}; cost saving= 54.97^{+}%											
IWexpUC	Correct	238	57	81	92	214	68	78	42	35	216
	Total	265	67	93	110	227	86	106	49	43	304
	%	0.898	0.851	0.871	0.836	0.943	0.791	0.736	0.857	0.814	0.711
Total accuracy=0.830; average test= 4.43^{+}; cost saving= 54.64^{+}%											

Note: The treatment and non-treatment threshold are [0.85, 0.15]. The accuracy of the baseline that trained with 9 tests is 0.8407. + or - indicates that ODPF is significantly better or worse than baseline, respectively ($\alpha = 5\%$).

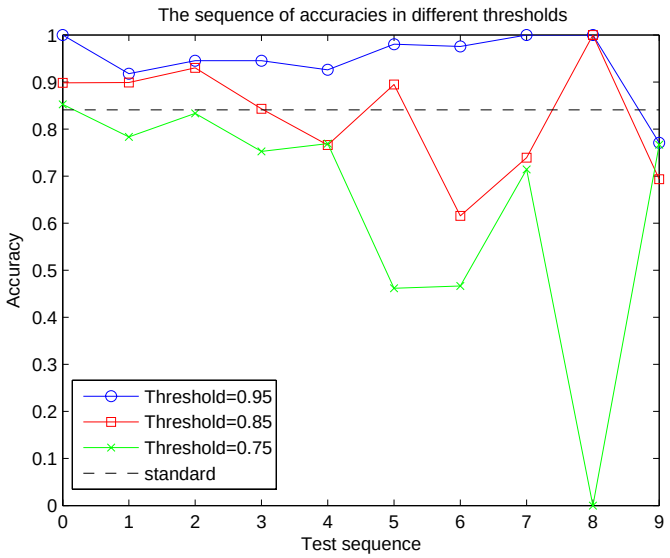


Figure 4.6: Accuracies of various thresholds

high, but the accuracy for ite9 is low. A very high threshold such as 0.95 forces many patients to stay until ite9 and greatly reduces the total accuracy. Although accuracies of most strata of an overly high threshold, such as 0.95, are higher, overall reduction (ite9) in the number of patients who are assigned a diagnosis outweighs this improvement (ite0 to ite8). Therefore, careful selection of an appropriate threshold helps to improve total accuracy. The fluctuation of accuracies in Figure 4.6 also results from early diagnosis, which results in small sample size from strata ite4 to ite8.

The relationship between effectiveness and costs is shown in Figure 4.8. We compare random, minimum uncertainty, and cost-based expected uncertainty. Random is the baseline strategy in which testing sequences are randomly created. This baseline allows the examination of the amount of contribution due to feature-selection strategies (i.e., minimum uncertainty and cost-based expected uncertainty). Random still benefits from the confident prediction structure because the threshold still determines the stopping point. Therefore, a lot of tests can still be saved. For example, when the threshold is 0.85, the system uses 3.68 tests with a total accuracy of 0.827.

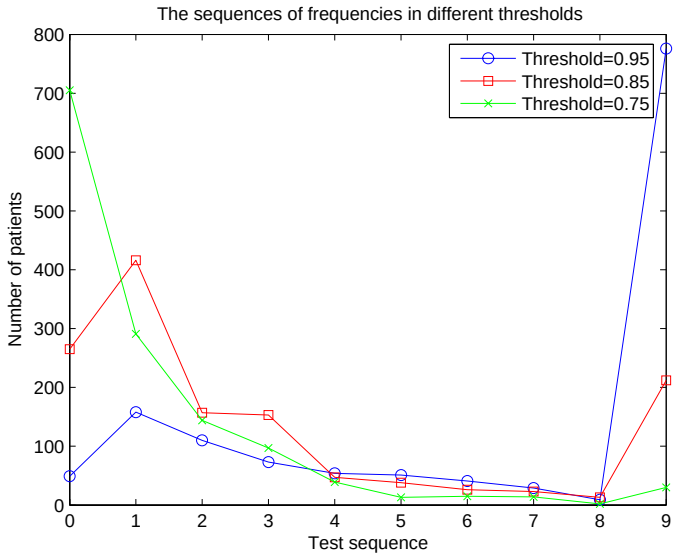


Figure 4.7: Frequency of each test

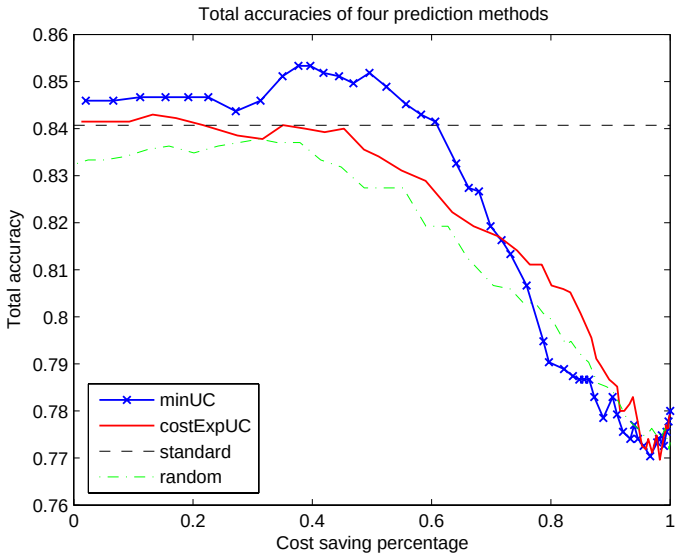


Figure 4.8: Accuracy v.s. cost saving

In Figure 4.8, we demonstrate that cost saving and probability threshold are negatively related. As expected, both strategies have more cost savings and higher total accuracies than random. For example, total accuracies for most points with same cost savings are higher in minUC than in random. Similarly, cost savings for most points with same total accuracies are more in minUC than in random.

In general, cost saving and total accuracy are a trade-off. Surprisingly, the minUC strategy provides more cost-savings, especially when the threshold is higher. For example, when the cost saving is 50%, the accuracy for minUC is 85.1% while the accuracy for costExpUC is 83.4%. The cost savings of the speed-based strategies comes from using a small number of tests that are usually expensive but strong predictors[2]. However, the cost savings of the cost-based strategies results from careful selection of tests. The selection prefers inexpensive tests with acceptable diagnostic power or powerful tests with reasonable cost. This result suggests that using expensive tests at the outset may be preferable in some situations. When the sequences are optimized, patients may be diagnosed more quickly and accurately, and more cost can be saved.

Confident prediction not only improves accuracy but also sensitivity and specificity. Figures 4.9 and 4.10 show sensitivity and specificity of each stratum for the minUC strategy. Both figures show that sensitivity and specificity for higher thresholds are better. Similar to accuracy, early diagnosis results in fluctuation of the curve. Standard represents sensitivity or specificity of the classifier trained with all tests. Similar to accuracy, confident prediction boosts both sensitivity and specificity for patients who are diagnosed early.

[2]In most cases, the selection starts from expensive and powerful tests. In some cases, an inexpensive test is preferred in the beginning depending on the known information before taking any tests.

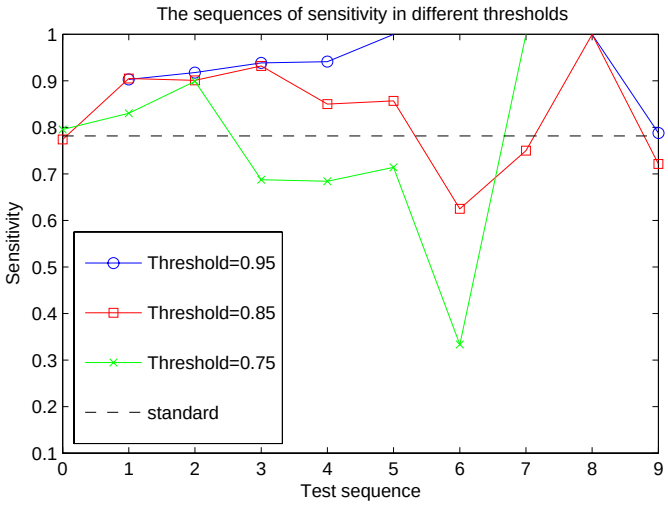


Figure 4.9: Sensitivity of each test

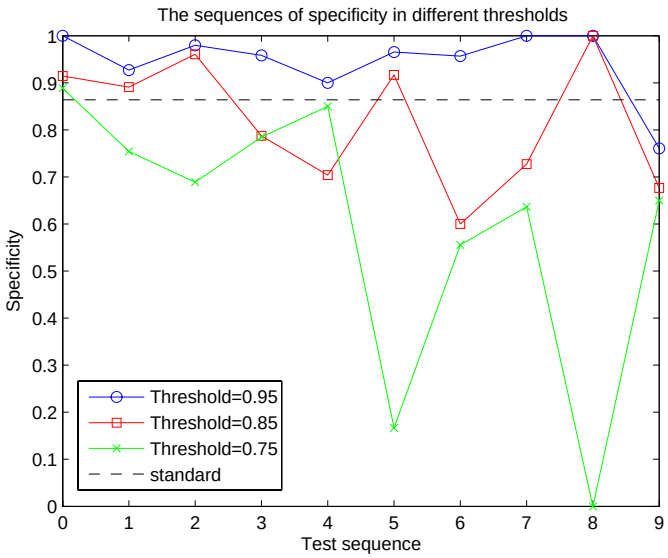


Figure 4.10: Specificity of each test

Table 4.7: Confusion matrix of the baseline for thyroid data

		Predicted Class			
		Hypothyroid	Hyperthyroid	Normal	Matched rate
True Class	Hypothyroid	250	200	15	0.538
	Hyperthyroid	54	681	220	0.713
	Normal	85	108	17247	0.989
	Predicted matched rate	0.643	0.689	0.987	
Total accuracy: 0.964; Cost: 53.81 (0% saving); Average number of tests: 4					

Table 4.8: Confusion matrix of ODPF (minimum uncertainty) for thyroid data

		Predicted Class			
		Hypothyroid	Hyperthyroid	Normal	Matched rate
True Class	Hypothyroid	352	99	14	0.7570^{+}
	Hyperthyroid	78	818	59	0.8565^{+}
	Normal	136	108	17196	0.9860^{+}
	Predicted matched rate	0.622	0.798^{+}	0.996^{+}	
Total accuracy: 0.974^{+}; Cost: 23.37 ($56.6\%^{+}$ saving); Average number of tests: 1.07^{+}					

Note: The treatment and non-treatment thresholds for both hyperthyroid and hypothyroid are [0.85, 0.02]. + indicates significantly better than baseline ($\alpha = 5\%$).

4.2.2 Thyroid Dataset

The thyroid dataset is highly unbalanced and has three classes, normal (92.47%), hyperthyroid (5.06%), and hypothyroid (2.47%). Tables 4.7 and 4.8 show confusion matrices of thyroid dataset from baseline (the classifier trained with all features) and ODPF using the function of minimum uncertainty.

For both tables, the third to fifth columns show predicted class frequencies of hypothyroid, hyperthyroid, and normal. The last column, matched rate, is the

proportion of correct prediction among all predictions in the same true class. For example, in Table 4.7, the matched rate of hypothyroid is $\frac{250}{465} = 0.538$. Similarly, the third to fifth rows show true class frequencies. The predicted-matched rate is the proportion of correctness of a predicted class among all true classes in the same prediction. For example, the predicted-matched rate of hypothyroid is $\frac{250}{389} = 0.643$. Total accuracy is the percentage of correct classification.

Comparing these two tables, most predicted performance indices are better than the baseline. The average number of tests and total cost of baseline (Table 4.7) are 4 and 53.81. In this dataset, we force the algorithm to give at least one test to all patients to improve prediction. In average, ODPF uses 1.07 tests and the total cost is 23.37 (cost saving 56.6%), both of which are significantly better than the baseline.

4.2.3 Application of The ODPF Method in Predicting Future Risk of Disease and Adverse Events

To determine whether ODPF can be extended to applications of diagnostic tests for prognosis and risk stratification, we performed an analysis of diabetes and hepatitis datasets.

Tables 4.9 and 4.10 show the aggregated results of all strata for the diabetes and hepatitis datasets. We use [0.75, 0.1] and [0.75, 0.08] as thresholds for diabetes and hepatitis datasets, respectively. They are determined based on reasonable combined results, e.g. reasonable performance, such as sensitivity or specificity, and cost reduction.

The first column of all tables shows all functions including speed-based methods and cost-based methods, and the following columns are accuracy, sensitivity, specificity, area under the ROC curve (AUC), cost, and average number of tests.

Standard is the baseline classifier that was trained with all features. Only the hepatitis dataset has missing values. In other words, health providers in the hepatitis dataset used various tests while they used all tests for all patients in other datasets.

Table 4.9: Diabetes summary

Functions	**Accuracy**	**Sensitivity**	**Specificity**	**AUC**	**Cost (save%)**	**Avg tests**
Standard	0.735	0.721	0.743	0.828	38.29 (0)	2
ProbContr	0.735	0.728^{+}	0.739^{-}	0.827^{-}	$27.80(27.4)^{+}$	1.474^{+}
minUC	0.735	0.727^{+}	0.739^{-}	0.827^{-}	$27.81(27.4)^{+}$	1.474^{+}
expUC	0.735	0.724	0.741^{-}	0.827	$30.04(21.5)^{+}$	1.578^{+}
expIWUC	0.735	0.722	0.742^{-}	0.827^{-}	$29.97(21.7)^{+}$	1.576^{+}
ProbContr_Cost	0.735	0.728^{+}	0.738^{-}	0.827^{-}	$27.82(27.3)^{+}$	1.475^{+}
minUC_Cost	0.735	0.728^{+}	0.738^{-}	0.827^{-}	$27.76(27.5)^{+}$	1.472^{+}
expUC_Cost	0.734^{-}	0.722	0.741^{-}	0.827	$29.98(21.7)^{+}$	1.576^{+}
expIWUC_Cost	0.734^{-}	0.722	0.741^{-}	0.828	$30.00(21.7)^{+}$	1.577^{+}

Note: The treatment and non-treatment threshold are [0.75, 0.1]. + or - indicates that ODPF is significantly better or worse than baseline, respectively ($\alpha = 5\%$).

Missing values are imputed using the class-conditional mean of a feature. Average tests and costs are computed based on tests and costs actually used and spent. Thus, the average number of tests in the baseline of Hepatitis is not an integer (4.213, see Table 4.10). We do not know the order of tests used in this database. When taking group discount into account, there may be more than one set of costs for a patient, and we take the minimum one for the baseline.

Although only two tests are involved in diagnosing diabetes, not both of them all required. All functions can reduce test cost from 21.5% to 27.5% while using an average number of tests ranging from 1.472 to 1.578. Most accuracies are close to baseline while cost and average tests are all significantly better than the baseline.

AUC can be improved because of the feature selection, but AUC can also degrade significantly due to early diagnosis. AUC is computed from predicted probabilities and corresponding labels of patients in all strata. However, treatment (or non-treatment) thresholds may stop us from improving probabilities. For example, consistent test results usually can further improve predicted probability, but ODPF

stops further testing when the current probability is confident enough. Therefore, the limitation of improving predicted probability can degrade AUC especially when the threshold is not high.

The Hepatitis dataset includes many missing values. Although both baseline and ODPF are trained and tested with the same dataset, the uses of the dataset are not exactly equal. There are two reasons.

First, missing values influence the test selection method. In order to train an SVM classifier, we impute missing data. The dataset for the baseline is fully imputed in the training and testing data. In ODPF, we still can impute missing values for training data and update C based on the rule as described in Section 4.1.6. However, for the query case, we do not impute missing values for the query. Instead, we avoid selecting tests with missing values. In other words, the selection of a test with strong potential to pass the threshold can be prohibited because the test feature value is missing in the query case.

Second, missing values influence the reporting of performance. A test with a missing value for a query case will be moved to the end of the test sequence, and it is marked as N/A because we do not use it. Such a query case will be either early diagnosis or in need of more tests than those available. Most cases are in the first situation, and a few of them are in the second. The performance indices of the second situation are reported based on the last test without a missing value. These query cases are forced to attain a diagnosis as a result of exhausting the tests with recorded values. In other words, their diagnoses are not ready but have to be made. The above two reasons may degrade the performance of all functions slightly.

Table 4.10 shows that specificity, number of tests, and cost saving of all functions of the Hepatitis dataset are significantly better than the baseline. In this experiment, both expIWUC and cost-based expIWUC have the highest diagnostic performance and the largest cost reduction among all functions.

Table 4.10: Hepatitis summary

Functions	**Accuracy**	**Sensitivity**	**Specificity**	**AUC**	**Cost (save%)**	**Avg tests**
standard	0.819	0.744	0.839	0.863	24.78(0)	4.213
ProbContr	0.817	0.675^{-}	0.854^{+}	0.857	$16.37(33.9)^{+}$	2.688^{+}
minUC	0.817	0.675^{-}	0.854^{+}	0.856	$15.96(35.6)^{+}$	2.622^{+}
expUC	0.819	0.700^{-}	0.850^{+}	0.859	$15.94(35.7)^{+}$	2.635^{+}
expIWUC	0.830^{+}	0.744	0.852^{+}	0.859	$14.05(43.3)^{+}$	2.394^{+}
ProbContr_Cost	0.817	0.675^{-}	0.854^{+}	0.857	$16.35(34.0)^{+}$	2.689^{+}
minUC_Cost	0.817	0.675^{-}	0.854^{+}	0.855	$15.90(35.9)^{+}$	2.628^{+}
expUC_Cost	0.819	0.700^{-}	0.850^{+}	0.858	$15.98(35.5)^{+}$	2.645^{+}
expIWUC_Cost	0.830^{+}	0.744	0.852^{+}	0.859	$14.06(43.3)^{+}$	2.395^{+}

Note: The treatment and non-treatment threshold are [0.75, 0.08]. + or - indicates that ODPF is significantly better or worse than baseline, respectively ($\alpha = 5\%$).

Previous results suggest that either sensitivity or specificity can be higher in ODPF. The either-or relationship can be adjusted by tuning γ (see Algorithm 1) or the class balance of positive to negative classes based on diseases.

4.3 Population-Based Method

This section discusses a machine-learning approach to find the best testing sequence based on population. In the end of this section, we compare this approach with previous individualized method.

A diagnostic guideline for a disease of interest describes the best testing sequence for a population. This section describes a potential way to determine such a sequence automatically. We use optimization to find promising test sequences. This project aims to find the test sequence that can diagnose most patients with high accuracy no matter how many tests they need. The number of tests required for diagnosis varies based on patients (influenced by, e.g., pre-test probabilities). When patients receive tests following this sequence, the diagnostic accuracies are high. This project also

proposes another objective: finding the testing sequence that can diagnose patients with the best balance of accuracy and cost.

4.3.1 Methods

This experiment uses heuristic search methods to evaluate and find the best sequence [21] for the above problem. As with the interactive method, using all tests is not required as long as the sequence can yield more benefits, i.e., save more money and diagnose more accurately. To obtain this sequence, we need to consider not only the order but also the number of tests in this sequence. Therefore, the number of possible sequences is $\sum_{n=1}^{m} C_n^m \times n!$, where m is the number of all tests, and n is the number of tests in a sequence. For example, if we use nine tests, the number of possible sequences is close to a million. Since finding the optimal test sequence is computationally prohibitive, we find a good locally-optimal sequence using a heuristic search.

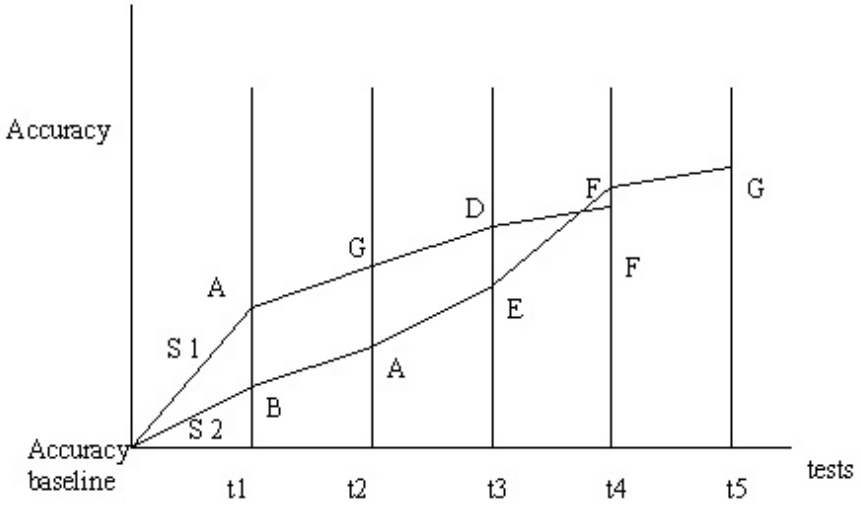


Figure 4.11: Accuracy gain and test selection for global test sequence

A good sequence should have high accuracies for most tests. A health provider

may make a diagnosis after a test, and a good sequence can enhance the diagnostic performance. Figure 4.11 illustrates this idea. Sequence $S1$ is better than $S2$ because most accuracies on $S1$ are better than $S2$. To compare sequences with a different number of tests, we use mean accuracy gain (MAG) to evaluate sequences. This accuracy-based evaluation function is defined as

$$MAG = mean(accuracy_i - accuracy_{base}), \forall i,$$

where $accuracy_i$ is the accuracy at test i and $accuracy_{base}$ is the accuracy of the classifier based on only patient-specific information.

The other evaluation function is cost-sensitive. The cost term is added to the above method. The cost-based function ($cMAG$) is

$$cMAG = mean(\frac{accuracy_i - accuracy_{base}}{cost_i}), \forall i,$$

where $cost_i$ is the cost of test i.

The next section will show the results of using both evaluation functions in random search, greedy search, simulated annealing, and genetic algorithm (see [16] for each method). Random search is used as a baseline to the other approaches. Greedy search may stop in a locally-optimal solution, but simulated annealing and genetic algorithms can avoid it.

The neighborhood function for greedy search and simulated annealing is an approach combining the 2-exchange neighbor and random key [10] approaches. In the beginning, all tests are assigned a random number ranging 0 to 1. After sorting these numbers, we can initiate a test sequence. If the assigned number is less than 0.5, the corresponding test is removed from the sequence (inactive). Active status for each test may change when creating new neighbors.

When creating a new neighbor (new sequence), we randomly choose a test whose assigned number will add a positive or a negative random number. The active status of this test may change depending on whether or not the new value is higher than 0.5 (becomes either active or inactive). For example, the assigned numbers for tests

1 to 9 are 0.98, 0.45, 0.7, 0.33, 0.68, 0.23, 0.15, 0.57, and 0.11. Only tests 1, 3, 5, and 8 are active because their values are 0.98, 0.7, 0.68, and 0.57, respectively. When test 2 is chosen and then a random number 0.1 is added, its active status will change. Therefore, the new active tests are 1, 2, 3, 5, and 8.

A newly active test will be added to the end of the current sequence, and a newly inactive test will disappear from the current sequence. In the previous example, the sequence with order [1,3,5,8] will be [1,3,5,8,2] when test 2 becomes active. If test 5 instead of test 2 is chosen and de-activated, the new sequence is [1,3,8].

Sometimes, the active status of a chosen test does not change. In order to ensure the creation of a different local solution, we also exchange the order of two active tests in this sequence.

For the neighborhood function of the genetic algorithm, we use another adapted random key approach. Similar to the above method, we assign random numbers between 1 and 10 to all tests in the beginning. After performing cross-over and mutation on these random numbers, we sort these numbers and decide the active status of tests based on the threshold 5. Finally, we can obtain a sequence with active tests. For example, the assigned random numbers from test 1 to test 9 are 1.2, 4.5, 6.7, 3.6, 7.7, 9.3, 2.5, 7.4, and 3.3. After sorting these numbers, we have a new test sequence [t6(9.3) t5(7.7) t8(7.4) t3(6.7) t2(4.5) t4(3.6) t9(3.3) t7(2.5) t1(1.2)]. Finally, the sequence of active tests is [t6, t5, t8, t3].

4.3.2 Results

In this study, we presented a potential method of finding the optimal sequence for a population by using several heuristic search methods. The sequence found by random search is the baseline. The dataset has been described in Table 4.1. Table 4.11 lists the results produced by accuracy- and cost-based evaluation functions. The delay tests are shown as T5 to T13. The row “Tests” shows the found sequence, and the row

Table 4.11: Random Search

Accu-	**Tests**	**T9**	**T8**	**T13**	**T10**	**T5**	
based	**Accuracy**	0.7296	0.7415	0.8000	0.8059	0.8074	
(MAG)	MAG=0.0036; cMAG=0.6321						
Cost-	**Tests**	**T8**	**T7**	**T10**	**T6**	**T9**	**T11**
based	**Accuracy**	0.7615	0.7615	0.7726	0.7748	0.7911	0.7911
(cMAG)	MAG=0.0021; cMAG=0.2919						

"Accuracy" shows the accuracy of the partial sequence. For example, in the accuracy-based evaluation function, the accuracy of the whole sequence [T9,T8,T13,T10,T5] is 0.8074, and the accuracy of the partial sequence [T9,T8,T13] is 0.8. Later, we simply call them the accuracy at T5 or T13. Each accuracy is a predicted accuracy for all patients from five 3-fold cross-validation runs. When tests are given based on this sequence, and a patient is diagnosed, the accuracy of the diagnosis can be found from this table. For example, when a patient is diagnosed after receiving test T9, T8, and T13, the expected accuracy is 0.8.

Next, we use hill climbing (greedy search) to find the optimal sequence. Table 4.12 shows the found sequence and the accuracy at each test. Hill climbing can find larger MAG and cMAG values. The accuracy-based method uses MAG as the evaluation function while cost-based method uses cMAG. It is interesting to note that when we use cMAG to guide the search, the value of its MAG improves, too. Using cMAG finds the most cost-effective sequence, so effectiveness (MAG) improves.

Greedy search finds a local optimum, so the solution is usually not good enough. Using a genetic algorithm (GA) or a simulated annealing (SA) avoids this problem. Tables 4.13 and 4.14 show the results of the two methods. Both MAG and cMAG improve, and accuracies are greater than 0.8 after receiving the second test (e.g., in Table 4.13, T10 in accuracy-based method or T8 in cost-based method). In addition,

Table 4.12: Greedy Search

Accu-based (MAG)	**Tests**	**T13**	**T10**	**T5**	**T11**	**T7**	**T8**	**T9**
	Accuracy	0.7756	0.7874	0.7911	0.7926	0.7948	0.7993	0.8119
	MAG=0.0199; cMAG=0.6273							
Cost-based (cMAG)	**Tests**	**T12**	**T11**	**T6**	**T9**	**T10**		
	Accuracy	0.7844	0.8126	0.8119	0.7978	0.8126		
	MAG=0.0306; cMAG=0.7174							

Table 4.13: Genetic Algorithm

Accu-based (MAG)	**Tests**	**T12**	**T10**	**T6**	**T8**	**T9**	**T11**	**T13**	**T5**	**T7**
	Accuracy	0.7807	0.8015	0.8015	0.8074	0.82	0.8252	0.8326	0.8267	0.8326
	MAG=0.0409; cMAG=0.9899; cost saving=0%									
Cost-based (cMAG)	**Tests**	**T12**	**T8**	**T10**	**T6**	**T9**	**T11**	**T13**		
	Accuracy	0.78	0.8126	0.8222	0.817	0.823	0.8304	0.8333		
	MAG=0.0436; cMAG=1.2585; cost saving=6.5%									

we continue to have high accuracies after the second test. Compared to the results of greedy search, we can have high accuracies very fast and continuously. Cost saving comes from unnecessary tests, e.g., T5 and T7 are not considered in the cost-based method in Table 4.13 because MAG or cMAG reduces when introducing these tests.

Compared with individualized methods discussed previously, there are several different points to be made. First, the target population of decision support is different, individual vs. population. The prior approach constructs the optimum sequence for each individual based on the pre-test probability and the information at hand. The latter approach can only construct the testing sequence based on population.

Second, the way of support is different (interactive vs. non-interactive). For the prior method, we need to continuously provide a patient's information, so the

Table 4.14: Simulated Annealing

Accu-	**Tests**	**T12**	**T10**	**T7**	**T9**	**T13**	**T6**	**T11**
based	**Accuracy**	0.7763	0.8089	0.8104	0.8119	0.8356	0.8341	0.8348
(MAG)	MAG=0.0427; cMAG=0.8228; cost saving=1.9%							
Cost-	**Tests**	**T8**	**T12**	**T10**	**T9**	**T11**	**T13**	
based	**Accuracy**	0.7726	0.8156	0.817	0.8207	0.8252	0.8319	
(cMAG)	MAG=0.0405; cMAG=1.3233; cost saving=8.1%							

system can interactively recommend the most promising test for the next time. In addition, the system recommends when a diagnosis can be made when the evidence is sufficient. On the other hand, a health provider needs to decide when the evidence is sufficient when following the test sequence constructed by the second approach.

Third, the methods of accuracy and cost-saving estimation are different. The first approach can recommend when to diagnose, and we can certainly examine the accuracy, cost and test saving of the recommendation by comparing with the data. On the other hand, accuracies in the second approach are from the whole population. In addition, cost and test savings come from eliminating unnecessary tests.

Fourth, the way to use is different. The first approach relies on interacting with the system, and a health provider needs to continuously communicate with a computer (or the system continuously needs the input of electronic medical records and then return recommendations). This is because a testing sequence is generated dynamically. On the other hand, the second approach can help make a diagnostic protocol, which can be used without a computer.

CHAPTER V
LIFESTYLE RECOMMENDATION

Lifestyle recommendation[1] traditionally focuses on the whole population. For example, the 2006 AHA Scientific Statement [66] suggests aims for healthy body weight, desired lipid profile, normal blood pressure, physical activity, etc. It also suggests specific thresholds for diet and lifestyle such as limitation of saturated fat intake $< 7\%$ of energy, cholesterol intake $< 300\ mg$, total fat intake between $25\%\ to\ 35\%$ of energy. These recommendations are good for the population, but they may not be the best for each individual. Intuitively, lifestyle recommendations should vary based on an individual's health and preferences, e.g., an athlete, a vegetarian, and a smoker should set different suggestions. Ideally, everyone should receive the recommendations of lifestyle changes that they would most benefit from, but this individualization problem is very complex because too many factors influence an individual simultaneously.

There are several advantages of individualized lifestyle recommendation. First, the recommendation of lifestyle changes can be constructed based on an individual's need. For example, an athlete has different needs of caloric and nutrition requirements from most non-athletes. If fact, each person has unique needs of nutrition and lifestyle [26] and individualized lifestyle recommendations have the potential to provide the maximum benefit for each individual. Second, when constructing an individualized plan, personal preferences can be taken into account. One may prefer activities to promote health while another may prefer nutrition changes. In addition, it can be difficult to fit a non-individualized plan for a healthy lifestyle into one's working and daily life without the individual's participation [104] in deciding the plan. Thus, it may be easier to comply with an individualized plan of lifestyle.

[1] In this chapter we use the term "lifestyle" to represent "nutrition and lifestyle."

This project proposes a machine learning algorithm to construct an expert system that can recommend an individualized plan of lifestyle changes for a query. Following this plan, the query patient is predicted to have the minimal 10-year risk of cardiovascular disease (CVD). The plan includes many lifestyle behaviors including exercise, smoking, and many kinds of nutrients. When constructing the plan, the algorithm takes personal preference and many individual factors, such as demographic data, medication in use, HDL, LDL, smoking history, etc., into account.

5.1 Method

We use a revised version of the PODSS algorithm (see Figure 3.1) to construct this expert system. In contrast to the hospital-referral project, the system uses k-NN (described in 5.1.2) instead of SVM as the predictive model. Unlike the hospital-referral project, lifestyle variables can be changed individually (in the hospital-referral project, the set of hospital-characteristics variables changes when switching hospitals). In addition, this PODSS has a built-in validation approach (described in 5.1.5) instead of using a validation map.

Section 5.1.1 describes the data used in this project, Section 5.1.2 discusses the k-nearest neighbor algorithm, Section 5.1.3 addresses our missing data handling approach, and Section 5.1.5 describes the validation method. The description of the optimization formulation of the healthiest plan is discussed in Section 5.1.4.

5.1.1 Data Preparation

Knowledge for this system is extracted from the data of the Atherosclerosis Risk in Communities (ARIC) study [84]. This study contains the Cohort Component and the Community Surveillance Component of four communities. The Cohort Component began in 1987 and subjects are examined every three years. ARIC recruited around 4000 individuals aged 45-64 from each of four communities. The total sample

size is 15,792. The baseline period is 1987-89, and the follow-up periods are 1990-92, 1993-95, and 1996-98. The Community Surveillance Component is the investigation of the community-wide occurrence of hospitalized myocardial infarction and coronary heart disease deaths in men and women aged 35-84 years. Patients without any CVD event before the baseline (1987-89) are selected in this research. Their 10-year CVD outcomes (including both CHD and stroke) are defined using the Community Surveillance Component.

All variables are from usual care, which can be obtained by asking a patient or doing a simple exam. We discretized all variables based on equal population in order to simplify the problem (i.e., similar population size in each discrete value). They include patients' characteristics and lifestyle. Similar to PODSS, a patients' characteristics describe the patient but one cannot change them. On the other hand, lifestyle variables are changable and one can change them to improve health. Table 5.1 summarizes both types of variables. The description of each variable is followed by several numbers. They are cutpoints for discretization. For example, there are five discrete values for body mass index, less than 23.347, between 23.347 and 25.746, between 25.746 and 28.058, between 28.058 and 31.378, more than 31.378.

Most of these variables (Table 5.1) are taken directly from the ARIC dataset except for total sport hours of lifestyle. The ARIC survey asked participants for their four most common activities and hours per week. Total sport hours is the sum of all listed activity time. The outcome of this model is binary, whether a patient has any CVD event (CHD event and stroke) in ten years. CHD is defined as any of the following diagnoses: probable MI, definite MI, suspect MI, missing pain (ECG and/or enzyme diagnosis), definite fatal CHD, definite MI, and possible fatal CHD. Stroke is defined as definite TIB (definite brain infarction, Thrombotic), probable TIB, possible stroke of undetermined type, undocumented fatal cases with stroke codes, out-of-hospital deaths with stroke codes.

Table 5.1: Variables from ARIC data used for lifestyle recommendation

	#	Types	Variables, values
Lifestyle	1	NUM	Body mass index, [23.347, 25.746, 28.058, 31.378]
	2	NUM	alcohol intake (g) per day, [0, 9.4286]
	3	CHAR	smoking status, [0,1]
	4	NUM	total activity hours per week, [3, 5, 7, 10]
	5	NUM	carbohydrate (g), [128.79, 166.19, 203.57, 258.52]
	6	NUM	dietary cholesterol (mg), [147.5, 200.23, 256.84, 337.56]
	7	NUM	dietary fiber (g), [10.52, 14.12, 17.82, 22.93]
	8	NUM	protein (%kcal), [14.524, 16.683, 18.668, 21.079]
	9	NUM	saturated fatty acid (%kcal), [9.545, 11.254, 12.676, 14.373]
	10	NUM	total fat (%kcal), [27.321, 31.409, 34.711, 38.418]
	11	NUM	cigarette years of smoking, [0, 280, 660]
	12	CHAR	cholesterol lowering medication use, [0, 1]
	13	CHAR	diabetes, [0, 1]
	14	CHAR	education level, [(1) Grade school or 0 years education (2) High school, but no degree (3) High school graduate (4) Vocational school (5) College (6) Graduate school or Professional school]
	15	CHAR	sex, [0, 1]
	16	NUM	HDL cholesterol in mg/dl, [37.557, 45, 52.965, 64.521]
	17	CHAR	hypertension, [0, 1]
	18	NUM	LDL cholesterol in mg/dl, [105, 126, 145, 168]

Table 5.1: continued

	19	CHAR	menopausal status, [(1) Primary Amenorrhea (2) Premenopause (3) Perimenopause (4) Post, Natural (5) Post, Surgical (6) Unknown Ovarian Status]
	20	CHAR	race, [B: black, N: non-black]
	21	NUM	total cholesterol in mmol/L, [4.6548, 5.2237, 5.7409, 6.3874]
	22	NUM	total triglycerides in mmol/L, [0.82417, 1.0838, 1.4112, 1.9419]
Characteristics	23	NUM	age, [48, 52, 56, 60]
	24	CHAR	high blood pressure medication in past 2 weeks, [Yes, No, Unknown]
	25	NUM	2nd and 3rd systolic blood pressure average, [106, 115, 123, 135]
	26	NUM	2nd and 3rd diastolic blood pressure blood pressure average, [65, 70, 76, 82]
	27	NUM	standing height to nearest CM, [160, 165, 171, 177]
	28	NUM	waist girth to nearest CM, [85, 93, 99, 107]
	29	NUM	hip girth to nearest CM, [97, 101, 105, 111]
	30	NUM	heart rate, [58, 63, 68, 75]
	31	NUM	white blood count, [4.6, 5.4, 6.3, 7.4]
	32	NUM	apolipoprotein AI (MG-DL), [107, 122, 137, 157]
	33	NUM	apolipoprotein B (MG-DL), [69, 83, 97, 116]
	34	NUM	APOLP(A) DATA (UG-ML), [19, 43, 86, 175]
	35	NUM	creatinine (MG-DL), [0.9, 1, 1.1, 1.2]

5.1.2 k-Nearest Neighbor

As opposed to SVM, k-Nearest Neighbor (k-NN) doesn't compile a universal predictive model in advance. It postpones induction until classification. In other words, it stores the whole training data and predicts by utilizing the distance-weighted classes of the query's k nearest neighbors [102]. The model is described as

$$p(c_j|q) = \frac{\sum_{X \in K_q} 1(x_c = c_j) \cdot K(d(x,q))}{\sum_{X \in K_q} K(d(x,q))}, \tag{5.1}$$

where $c_j \in classes\ J$. x_c is the class membership of query q. 1() is 1 iff the argument is true. Note that there are J possible classes and 1() defines the specific class to which a query q belongs. K is the distance weighted function, and K_q is the set of q's k nearest neighbors among the training data. The distance function between q and x is defined as

$$d(x,q) = (\sum_{f \in F} w(f) \cdot \delta(x_f, q_f)^r)^{\frac{1}{r}}, \tag{5.2}$$

where F is the feature set. In this project, we define $r = 2$ (i.e., Euclidean distance). $\delta()$ is defined in Equation 5.3. $w(f)$ is the feature weighting function which is defined in Equation 5.4.

$$\delta(x_f, q_f) = \begin{cases} |x_f - q_f|,\ f \text{ is numeric} \\ 0,\ f \text{ is categorical and } x_f = q_f \\ 1,\ f \text{ is categorical and } x_f \neq q_f \end{cases} \tag{5.3}$$

In Equation 5.4, mutual information (MI, $w(f)$) between values of a feature and the class is defined to be the weight of the feature f. v is a value of a feature and V_f is

the value set of f. The function of assigning MI to each variable is similar to feature selection. However, weights in feature selection are binary but weights in k-NN are continuous.

$$w(f) = \sum_{v \in V_f} \sum_{c_j \in J} p(c_j, x_f = v) \cdot log \frac{p(c_j, x_f = v)}{p(c_j) \cdot p(x_f = v)} \tag{5.4}$$

5.1.3 Handling Missing Values

Missing values are very common in medical data. They may result from an unwillingness to answer questions, the non-inclusion of unnecessary tests, or other reasons. The common approach is to impute missing values (e.g., compute the average). In this project, we impute distance measures instead of missing values. We use expected distance to impute the distance between a query and a training case.

There are two possible scenarios: either a query or a training case has a value missing, or both are missing values. For the first scenario, the value of either a query or a training case is known, and we compute the expected distance measure of matching this known value. For example, a feature has three categorical values [r,g,b] whose probabilities are [0.4,0.25,0.35], respectively. When the known value of either a query or a training case is b, and the other is missing, the expected distance is $0.4 \times (1) + 0.25 \times (1) + 0.35 \times (0) = 0.6$. On the other hand, if the three values are [-1,0,1] and the known value is 1, the expected distance is $0.4 \times (|1 - (-1)|) + 0.25 \times (|1 - 0|) + 0.35 \times (|1 - 1|) = 1.05$.

For the second scenario, both values are missing, The probability for values r to r, r to g, r to b, g to r, g to g, g to b, b to r, b to g, and b to b are 0.16, 0.1, 0.14, 0.1, 0.0625, 0.0875, 0.14, 0.0875, and 0.1225, respectively. The expected distance for a categorical variable is $0.16 \times (0) + 0.1 \times (1) + 0.14 \times (1) + 0.1 \times (1) + 0.0625 \times (0) + 0.0875 \times (1) + 0.14 \times (1) + 0.0875 \times (1) + 0.1225 \times (0) = 0.655$. The expected distance for a numeric variable is $0.16 \times (|-1-(-1)|) + 0.1 \times (|-1-0|) + 0.14 \times (|-1-$

$1|) + 0.1 \times (|0 - (-1)|) + 0.0625 \times (|0 - 0|) + 0.0875 \times (|0 - 1|) + 0.14 \times (|1 - (-1)|) + 0.0875 \times (|1 - 0) + 0.1225 \times (|1 - 1) = 0.935.$

5.1.4 Optimization: The Healthiest Plan

We aim at finding the best lifestyle plan for each individual. There are two scenarios. The first one finds the single lifestyle component with a new value that can minimize one's CVD risk. The formulation is described as follows.

$$\begin{aligned} \text{minimize} \quad & p(x_1 \cup x_{2ij}) \\ \text{subject to} \quad & i = 1, ..., |x_2| \\ & j = 1, ..., |S_i|, \end{aligned} \tag{5.5}$$

where x_{2ij} represents one lifestyle component i with the value j. S represents the set of possible values for i. The objective is to find the best value j of the single lifestyle choice i for a patient with characteristic vector x_1. p is the decision function as described in (5.1). We can simply use exhaustive search to try all possible values of each lifestyle variable since all variables have been discretized.

The second scenario finds the combination of several lifestyle components that minimize one's CVD risk. The formulation is described as follows:

$$\begin{aligned} \text{minimize} \quad & p(x_1 \cup x_2) \\ \text{subject to} \quad & x_{2i} \in \text{S}, \ \text{i= } 1, ..., |x_2|, \end{aligned} \tag{5.6}$$

where x_2 represents one's lifestyle vector, and the returned x_2 is the best lifestyle vector for an individual. In order to return x_2 immediately, we use forward selection [59] to solve the problem. We start with an empty lifestyle vector, and then include a lifestyle component in each iteration given x_1 and the previously included lifestyle components. Finally, we can construct the entire x_2 vector.

5.1.5 Validation Method

Validation is difficult because patients never received any recommendation from the system. We can certainly validate a recommendation system by a clinical trial, but it is out of the scope of machine learning. This dissertation discusses a possible validation approach using machine learning.

Figure 5.1 illustrates this validation method. Assume Q is a query patient with individual characteristics P and originally-chosen lifestyle L_0. Q' is the same query patient with characteristics P but receives the recommended lifestyle L^*. In other words, Q represents the real patient-lifestyle pair that we observe in the dataset, and Q' represents the non-existent patient-ideal lifestyle pair.

We then estimate the risk of both Q and Q' by a holdout dataset (cases independent of the data used for recommendation). Finally, one can compare whether Q' shows lower risk than Q. Unlike prediction, validation cannot be done through comparing between predicted and true labels. In this problem, we compare predicted CVD risks between original and recommended lifestyles for the same subject P.

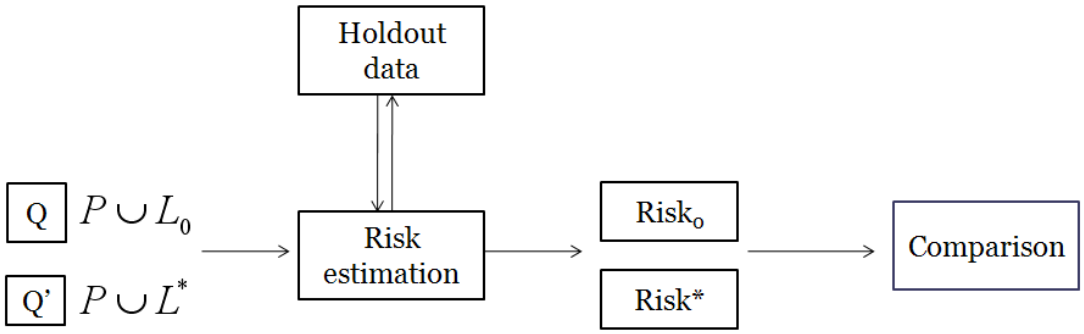


Figure 5.1: The validation method

In order to implement this idea, we stratify and assign 50% of data for training and yielding recommendation and 50% for validation (holdout data). We obtain the best L^* for the query P by using k-NN and optimization with the first dataset, and

then estimate risks of both Q and Q' by using k-NN with the second dataset. After producing lifestyle recomemndations and risk estimations for all query patients, we can compare whether patients in group Q' show lower risk than group Q. Thus, we can estimate if CVD risk can be reduced when following lifestyle recommendations.

5.2 Results

Table 5.2 shows the comparison between original and recommended lifestyle. The "Original" column shows the average CVD risk of patients with their original lifestyles, and "Recommend" shows the average CVD risk of patients with recommended lifestyle changes. Each subject receives two type of recommendations, single lifestyle changes ("Predict-Single" column) and multiple lifestyle changes ("Predict-Multiple" column). In the first situation, the system gives only one lifestyle change recommendation to each user. In this situation, we select the best lifestyle change (the change with the maximum CVD-risk reduction) for each user. On the other hand, each user receives multiple lifestyle changes recommendation (more than one for most cases) in the second situation. The number of lifestyle changes recommendation varies based on patients. In general, query patients with many bad behaviors (e.g., smoking) need more changes than the ones with fewer bad behaviors.

The row "p-value-improvement" examines whether each recommendation is better than "Predicted-Ori". p values in both cases are very small. There are two reasons for the small p-value. First, the data size is big (13006 data points). Second, most data points show consistent improvement.

The table suggests that lifestyle recommendation can successfully reduce risk, but the reduction of predictive probabilities is very small. There is one important reason. Many variables are the possible cause of change of the other variables. Independent variables are divided into patient characteristics and lifestyle, and we can only change lifestyle to reduce CVD risk. In the real world, patient characteristics

Table 5.2: Comparison between true and predictive outcomes

	Original	**Recommend**	
	Pedict-Ori	**Predict-Single**	**Predict-Multiple**
Avg prob	0.0803	0.0798	0.079
p-value-improvement	–	<<0.0005	<<0.0005

would change with lifestyle. For example, when a patient is recommended to reduce cholesterol intake, the CVD-risk reduction is computed only based on reduction of cholesterol intake. However, true CVD-risk reduction should also take changes of patient characteristics such as HDL and LDL levels into account. When computing predictive CVD-risk reduction, this model considers the change of cholesterol intake without possible changes of HDL and LDL levels. Therefore, this model underestimates the true CVD-risk reduction. General speaking, the recommendation of lifestyle changes for a query is generated by observing the lifestyle of similar subjects with low-CVD risks. Thus, the recommendation of lifestyle is still the best for the query. In other words, although actual risk reduction estimation is a limitation of this model, we still can use the relative risk reduction to identify a healthy lifestyle.

Table 5.3 shows the proportions of current smoking status, obesity, over-intake of cholesterol, over-intake of saturated fat, over-intake of total fat, and less activity for hypertension patients, diabetes patients, and smokers. 31.9% of subjects have hypertension, 8.2% have diabetes, and 43.6% are smokers. Obesity is determined by whether a subject's BMI is greater than 30 [13]. Too much cholesterol, saturated fat, and total fat are determined by cutpoints 300 mg, 7% energy, and 35% energy as suggested by [66]. There is no sugested cutpoint to decide less activity. In this project, the cutpoint is 5 hours, which is the median activity value.

Table 5.4 summarizes three most frequent single lifestyle recommendations for smokers. All smokers receive only one lifestyle change recommendation. In other

words, this lifestyle change is the most important. Although the single lifestyle recommendation is not practical, we use it to examine how the system reasons. As expected, the majority (54.4%) of smokers should quit smoking first, 37.2% should control their cholesterol, and 4.8% should control their weight first. Compared with Table 5.3, we can see the first group of subjects are less obese (11.8%), and take less cholesterol (5.5%), saturated fat (93.4%), and total fat (28.1%), so quit smoking is certainly the first recommendation. The second group of subjects are slightly more obese (20.8%), takes more cholesterol (77.2%), saturated fat (98.9%), and total fat (56.3%), especially cholesterol intake. That's the reason they are recommended to control cholesterol first. The third group of subjects are very obese (94.2%), but they have less cholesterol (8.4%), saturated fat (94.8%), and total fat (35.7%) intake. Thus, they are recommended to control weight.

Through the comparison, we can see that the recommendation is given based on one's characteristics. A related question is why some people are recommended to control cholesterol or weight before quitting smoking. This is because CVD risk reduction of cholesterol or weight control for the second or third groups of subjects are higher than quitting smoking. Thus, the recommendation is also given based on the best match that results in the maximum CVD reduction.

To be realistic, the system needs to provide more than one lifestyle change recommendation. Every subject can receive a whole plan of lifestyle change recommendations. However, one may prefer to start from a couple most effective lifestyle changes instead of recommending all lifestyle changes at once. Thus, we use a simple optimization method to find the most effective combination (from two to five) of lifestyle changes (called a package of n). Tables 5.5 to 5.8 shows the packages of two to five lifestyle changes recommendations. Quitting smoking is always included in each of these packages.

For example, in the package of three lifestyle changes (Table 5.6), the three

most frequent recommendations are [cholesterol control, total fat control, quit smoking] (35.5%), [weight control, cholesterol control, quit smoking] (23.9%), and [weight control, total fat control, quit smoking] (9.5%).

Tables 5.9 to 5.13 show results of single to a package of five lifestyle changes for subjects with diabetes. Similarly, Tables 5.14 to 5.18 show results of single to a package of five lifestyle changes for subjects with hypertension.

Most smokers with diabetes or hypertension are recommended to quit smoking (e.g., 100% in the third large group in Table 5.10). For the first and the second large groups, other recommendations resulting in higher CVD risk reduction than smoking cessation. Thus, there are a few smokers in the first and the second large groups in Table 5.10. Clinically, quitting smoking is a very important recommendation because smoking influences not only CVD but also many other problems. However, in this project, we only consider the influence on CVD. If the purpose of recommendation aims to reduce the risk of combination of many diseases, we expect that the recommendation of smoking cessation will be much more frequent.

Clinically, weight control is usually recommended for diabetes and hypertension patients, and the systems correctly identify it for obese patients. In the single recommendation for both diseases (Tables 5.9 and 5.14), weight control is the recommendation for either the largest or the second largest groups. In addition, lowering cholesterol has the most benefit for reducing CVD risks, and the system did show that lowering cholesterol is one of the most frequent recommendations.

It is very interesting to note that although physical activity appears in the five-lifesyle-in-a-package recommendation for each category (Tables 5.8, 5.13, 5.18), in general, physical activity is not a frequent recommendation. The reason may be due to the limitation of the dataset. Many responses (36.5%) for activity hours are missing. If not missing, the minimum value is 1 hour per week. In other words, no response has 0 activity hour. According to AHA guideline [66], 61% of US adults

Table 5.3: Distribution of smoking, cholesterol intake, obese, saturated fat, total fat, and activity in hypertension patients, diabetes patients, and smokers

	Hypertension (31.9%)	**Diabetes (8.2%)**	**Smokers (43.6%)**
smoking	22.6%	22.4%	100%
obese	39.6%	51.9%	19.4%
too much cholesterol	28.5%	33.3%	32.2%
too much saturated fat	94.8%	95%	95.8%
too much total fat	36.3%	43.5%	41.5%
less activity	28.1%	28.2%	25.2%

do not engage in any regular physical activity. In our dataset, we expect that there should be many missing responses whose true values are 0. Thus, the data limit the system's learning, and result in under-estimating the benefit of activity. Specifically, it learns the benefit from little activity to more activity but not from no activity to some activity. As a result, the under-estimation causes limited recommendation in activity.

Finally, we use a case study to shows another flexible application of this system in Table 5.19. The column "Original" shows subject P48's original lifestyle. There are three recommendations. "The whole plan" shows P48 should have 9 lifestyle changes. A small threshold (1e-12) was set to determine whether a lifestyle change should be included. In other words, because of small estimated CVD-risk reduction, the system doesn't consider carbohydrate in the whole plan. This may indicate carbohydrate intake is adequate for P48, and there is no need to change.

P48 may only want a couple lifestyle changes instead of the whole plan. The best package of three is controlling weight, total fat, and quit smoking. If P48 follows these recommendations, his CVD risk reduction would be 0.00125, i.e., 72.8% of the total

Table 5.4: Single lifestyle recommendation for smokers

	Quit smoking (54.4%)	**Cholesterol control (37.2%)**	**Weight control (4.8%)**
Smoking	1	1	1
Obese	0.118	0.208	0.942
Over cholesterol	0.055	0.772	0.084
Over saturated fat	0.934	0.989	0.948
Over total fat	0.281	0.563	0.357
Less activity	0.259	0.237	0.240

Table 5.5: A package of two lifestyle recommendations for smokers

	Cholesterol control and quit smoking (44.3%)	**Weight control and quit smoking (18.6%)**	**Total fat control and quit smoking (17.3%)**
Smoking	1	1	1
Obese	0.082	0.474	0.068
Over cholesterol	0.504	0.020	0.031
Over saturated fat	0.976	0.880	1
Over total fat	0.330	0.129	0.790
Less activity	0.238	0.254	0.280

Table 5.6: A package of three lifestyle recommendations for smokers

	CHOL and TOT fat control and quit smoking (35.5%)	**Weight and CHOL control and quit smoking (23.9%)**	**Weight and total fat control and quit smoking (9.5%)**
Smoking	1	1	1
Obese	0.052	0.442	0.485
Over cholesterol	0.550	0.384	0
Over saturated fat	1	0.956	1
Over total fat	0.716	0.143	0.528
Less activity	0.246	0.235	0.282

Table 5.7: A package of four lifestyle recommendations for smokers

	CHOL, SAT fat, and TOT fat and quit smoking (29.9%)	**Weight, CHOL, and TOT fat and quit smoking (28.1%)**	**Weight, SAT fat, and TOT fat and quit smoking (6.9%)**
Smoking	1	1	1
Obese	0.002	0.408	0.389
Over cholesterol	0.487	0.455	0
Over saturated fat	1	1	1
Over total fat	0.710	0.492	0.597
Less activity	0.245	0.241	0.312

Table 5.8: A package of five lifestyle recommendations for smokers

	Weight, CHOL, SAT fat, and TOT fat control and quit smoking (43.1%)	**CHOL, SAT fat, and TOT fat control, quit smoking, and activity (9.1%)**	**CHOL, SAT fat, and TOT fat control, fiber and quit smoking (5.2%)**
Smoking	1	1	1
Obese	0.296	0	0
Over cholesterol	0.455	0.469	0.317
Over saturated fat	1	1	1
Over total fat	0.598	0.623	0.611
Less activity	0.248	0.342	0.174

Table 5.9: Single lifestyle recommendation for diabetes

	Weight control (42.8%)	**Cholesterol control (39.7%)**	**Total fat control (9.8%)**
Smoking	0.294	0.428	0.340
Obese	0.787	0.405	0.115
Over cholesterol	0.105	0.709	0.048
Over saturated fat	0.914	0.991	1
Over total fat	0.321	0.5	0.865
Less activity	0.323	0.249	0.269

Table 5.10: A package of two lifestyle recommendations for diabetes

	Weight and CHOL control (37.9%)	**Weight and TOT Fat control (15.5%)**	**Weight control and quit smoking (14.7%)**
Smoking	0.173	0.132	1
Obese	0.740	0.703	0.583
Over cholesterol	0.526	0.030	0.013
Over saturated fat	0.963	1	0.833
Over total fat	0.333	0.733	0.077
Less activity	0.300	0.315	0.263

Table 5.11: A package of three lifestyle recommendations for diabetes

	Weight, CHOL, and TOT fat control (38.2%)	**Weight and CHOL control and quit smoking (18.5%)**	**CHOL and TOT fat control and quit smoking (9.4%)**
Smoking	0.049	1	1
Obese	0.677	0.569	0.05
Over cholesterol	0.502	0.320	0.5
Over saturated fat	1	0.924	1
Over total fat	0.643	0.036	0.66
Less activity	0.308	0.269	0.26

Table 5.12: A package of four lifestyle recommendations for diabetes

	Weight, CHOL, SAT fat, and TOT fat control (31.2%)	**Weight, CHOL, and TOT fat control and quit smoking (27.4%)**	**Weight, SAT fat, and TOT fat control and quit smoking (7.1%)**
Smoking	0	1	1
Obese	0.605	0.598	0.613
Over cholesterol	0.485	0.395	0
Over saturated fat	1	1	1
Over total fat	0.705	0.385	0.587
Less activity	0.316	0.282	0.24

Table 5.13: A package of five lifestyle recommendations for diabetes

	Weight, CHOL, SAT fat, and TOT fat control and quit smoking (46.2%)	**Weight, CHOL, SAT, fat, and TOT fat control and activity (13.5%)**	**Weight, CHOL, SAT fat, and TOT fat control and fiber (5.0%)**
Smoking	1	0	0
Obese	0.578	0.573	0.491
Over cholesterol	0.424	0.455	0.340
Over saturated fat	1	1	1
Over total fat	0.567	0.552	0.736
Less activity	0.275	0.462	0.132

Table 5.14: Single lifestyle recommendation for hypertension

	Cholesterol control (44.1%)	**Weight control (33.2%)**	**Quit smoking (13.4%)**
Smoking	0.389	0.188	1
Obese	0.306	0.716	0.080
Over cholesterol	0.587	0.066	0.025
Over saturated fat	0.984	0.921	0.866
Over total fat	0.434	0.241	0.156
Less activity	0.263	0.305	0.248

Table 5.15: A package of two lifestyle recommendations for hypertension

	CHOL and weight control (31.6%)	**CHOL and total fat control (17.4%)**	**weight control and quit smoking (15.2%)**
Smoking	0.149	0.196	1
Obese	0.653	0.160	0.494
Over cholesterol	0.428	0.522	0.003
Over saturated fat	0.976	0.999	0.843
Over total fat	0.243	0.820	0.046
Less activity	0.281	0.253	0.294

Table 5.16: A package of three lifestyle recommendations for hypertension

	Weight, CHOL, and TOT fat control (29.4%)	**Weight and CHOL control and quit smoking (20.8%)**	**CHOL and TOT fat control and quit smoking (10.0%)**
Smoking	0.044	1	1
Obese	0.581	0.490	0.022
Over cholesterol	0.439	0.317	0.452
Over saturated fat	0.998	0.951	1
Over total fat	0.586	0.064	0.597
Less activity	0.283	0.288	0.234

Table 5.17: A package of four lifestyle recommendations for hypertension

	Weight, CHOL, SAT fat, and TOT fat control (25.5%)	**Weight, CHOL, and TOT fat control and quit smoking (24.7%)**	**CHOL, SAT fat, and TOT fat control and quit smoking (9.1%)**
Smoking	0	1	1
Obese	0.493	0.521	0
Over cholesterol	0.416	0.382	0.443
Over saturated fat	1	0.999	1
Over total fat	0.629	0.361	0.642
Less activity	0.261	0.288	0.244

Table 5.18: A package of five lifestyle recommendations for hypertension

	Weight, CHOL, SAT fat, and TOT fat and quit smoking (36.7%)	**Weight, CHOL, SAT fat, and TOT fat and activity (12.5%)**	**Weight, CHOL, SAT fat, and TOT fat and fiber (5.5%)**
Smoking	1	0	0
Obese	0.485	0.460	0.460
Over cholesterol	0.410	0.404	0.274
Over saturated fat	1	1	1
Over total fat	0.528	0.524	0.602
Less activity	0.272	0.366	0.106

possible reduction. If P48 doesn't want to quit smoking, he needs to add cholesterol control, saturated fat control, total fat control, and more fiber to compensate for not quit smoking. Thus, P48 needs to consider if he wants to take the simple three changes including quit smoking or five changes without quit smoking.

Originally, we include monounsaturated (MUFA) and polyunsaturated (PUFA) fats in the system, but they are removed due to two reasons. First, usually they are not the recommendation targets. The only recommendation about unsaturated fat is the consumption of all kind of fat (total fat) should be below 35% of energy threshold. Second, it is recommended to replace saturated fat with MUFA and PUFA, but the recommendation from the previous version of system usually considers MUFA as bad fat. Thus, many recommendations involved controlling MUFA, but such recommendations conflict with current understanding about MUFA.

As a result, we conducted a simple investigation about MUFA. Table 5.20 shows correlation coefficients of CVD and various nutrition elements in five levels of total

Table 5.19: A case study of P48

		Original	**The whole plan**	**A package of three**	**Compensation**
Lifestyles	BMI	> 31.38	< 23.3	< 23.3	< 23.3
	alcohol (g)/day	0	> 9.4		
	smoking	yes	no	no	
	sport hours/wk	< 3	> 10		
	carbohydrate (g)	< 128.8			
	cholesterol (mg)	147.5 to 200.2	< 147.5		< 147.5
	fiber (g)	< 10.5	> 22.93		> 22.93
	protein (%kcal)	> 21.1	< 14.5		
	saturated fat (%kcal)	> 14.4	< 9.5		< 9.5
	total fat (%kcal)	> 38.4	< 27.3	< 27.3	< 27.3
	CVD risk reduction		0.00172	0.00125 (72.8% of total possible reduction)	0.00128 (74.4% of total possible reduction)

Table 5.20: Correlation coefficients of CVD and nutrition intake

	Carbohydrate	**Cholesterol**	**Fiber**	**MUFA**	**PUFA**	**Protein**	**SAT fat**	**TOT fat**
1 st	0.023	0.044	-0.009	0.025	-0.039	-0.027	-0.008	–
2 nd	0.033	0.048	-0.012	0.047	-0.038	-0.028	-0.012	–
3 rd	0.013	0.083	0.002	0.019	-0.032	0.006	-0.007	–
4 th	0.006	0.037	-0.017	0.012	-0.053	0.003	0.015	–
5 th	0.011	0.064	-0.006	0.027	-0.013	-0.012	0.038	–
Total	0.010	0.067	-0.013	0.048	-0.009	-0.005	0.038	0.042

fat (based on four cutpoints 27.3%, 31.4%, 34.7%, and 38.4%) and in the whole population. Each column represents one nutrition element. As expected, fiber and PUFA show negative correlation with CVD, and cholesterol shows positive correlation. Surprisingly, MUFA positively relate with CVD. In addition, saturated fat is positively related to CVD, but it is negatively related with CVD when the consumption of total fat is not much (first three levels of total fat). Saturated fat positively relates with CVD in the whole population. This surprising finding will be investigated further in future work.

CHAPTER VI
CONCLUSION AND FUTURE WORK

This central idea of this dissertation is aimed at facilitating personal health care, reducing costs of health care, and improving outcomes. This dissertation proposes new machine-learning algorithms for three disjointed health care problems: hospital referral, cost-effective diagnosis, and lifestyle recommendation. These problems have been well studied by single-fits-all methods[1]. This dissertation uses novel single-fits-single approaches to examine these problems.

Each individual is defferent. In order to have the best outcome, everyone should receive individualized care, i.e., a care solution specific to each individual given the unique properties. Furthermore, an individualized health care solution should be the best for a specific patient and not required to be so for the whole population. In addition, preference and real-world limitations of each individual vary. A human expert such as a health care provider can certainly provide individualized care based on experience and such an event can be recorded as data. This dissertation shows that machine learning can extract medical knowledge from data and then optimization methods obtain the individualized optimal solution for decision support based on one's properties and preference.

In the hospital referral and lifestyle recommendation projects, individualized recommendation is generated based on the input of personal characteristics and preferences. The systems can then return the best individual solution (hospital selection or the plan of lifestyle changes) that fits one's preference and personal considerations. In the cost-effective diagnosis project, the recommendation of a test is provided based on individual information (including symptoms and previous test results). The recommended test has the highest potential to cross (or get close to) the treatment

[1]The best solution for the population, e.g., choose a large volume hospital.

(or non-nontreatment) threshold. In other words, we optimize diagnosis in terms of the number of tests and the amount of cost without sacrificing accuracy (sometimes improving accuracy).

There are several advantages of machine learning-based decision support systems. First, the knowledge of the systems is automatically extracted from data. Thus, we do not need to spend a lot of time and labor in constructing and maintaining the knowledge base. When constructing or maintaining the knowledge source, we simply train the model with data. In these projects, humans set parameters (e.g., deciding the treatment threshold) and communicate with the system (e.g., providing personal preferences or considering the trade-off between travel distance and survival probability).

Second, the recommendation is the best solution for an individual. Knowledge is extracted by predictive models and recommendations are generated by optimization techniques. Optimization techniques can find the best solution (e.g., the most efficient way to allocate medical staff) with a given function of patterns. For many problems, the function is not easy to formulate. Fortunately, one can use a predictive model (e.g. SVM) to capture patterns of the real world as a decision function. Thus, the integration of a predictive model and an optimization method can automatically generate the best solution observed so far.

Third, the extracted knowledge fits the real world better. In the real world, outcomes, such as 10-year CVD, are influenced by several factors and their interaction. A non-linear predictive model can flexibly capture the relationship between variables and outcomes, and hence, usually predict better. Similarly, the recommendation may be better when using the non-linear form of knowledge, compared to standard recommendation based on correlation with outcome (risk).

Fourth, one can apply different predictive models for all three systems. For example, the hospital-referral and lifestyle recommendation projects use the same

PODSS framework. However, the predictive models in the former one are SVMs and in the latter are k-NNs. There is no globally best predictive model, as different models are more appropriate for different datasets. When applying these algorithms, one is allowed to choose the best predictive model that fits data the best, allows visualization of knowledge, or some other reasons. For example, our reasons for using k-NN in the lifestyle recommendation is simple implementation.

There are several limitations of the data in these projects. For the hospital-referral project, the area of data is limited to Iowa. Thus, hospitals in border states are not considered. The results of this application are limited to the specific type of disease, area, and time period. The best hospital for CABG surgery is not necessarily the best hospital for complex cancer surgeries. If the data for training are outdated, such as ten years old, the recommendation may not reflect current practices. We must update knowledge by training the system with new data because the outcomes for a hospital are likely to change over time. For example, a hospital may adopt new technologies, promote quality improvement, or experience surgeon turnover.

For the cost-effective diagnosis project, limitations of this study are described below. First, all patients in the heart disease dataset received three non-invasive cardiac tests and complete clinical and ECG data were collected for these patients. As not all patients in clinical practice would be expected to receive a comprehensive battery of non-invasive tests, the projected reduction in number of diagnostic tests ordered (and the associated costs) attributable to use of the ODPF algorithm may be optimistic. In fact, this limitation applies to most datasets in our analysis (except for the hepatitis dataset). Second, we did not evaluate the recommended sequences of diagnostic tests selected by the ODPF algorithm to determine whether these test sequences would be clinically acceptable to practicing health providers. To address this issue, one may set appropriate constraints (identified by clinician experts) to create viable test sequences in clinical practice. Third, the source dataset includes

patients who were referred to a single center for diagnostic evaluation, and it is unclear how the ODPF algorithm would perform in an unselected, contemporaneous samples of patients with suspected CAD or thyroid disease.

For the lifestyle project, the dataset is the description of a specific population. First, the recommended lifestyle is thus the best lifestyle observed from that population. Some known healthy behavior may not be recommended when most people don't have the behavior. For example, when most people consume saturated fat more than 7% energy (this cutpoint is suggested by American Heart Association), the system may not recognize the benefit of satisfying this criterion. In addition, due to bias, the recommended lifestyle may not be the best for patients not in that population. For example, one population smokes a lot and the other population rarely smokes. If the model is constructed by training with the second population and then recommend for the first population, the recommendation would under-estimate the influence of smoking. Second, for some patients, 10 years is not long enough to track their CVD events.

We describe future work as follows. For the hospital referral project, we would select hospitals from the nation instead of a single state. We can also broaden disease options. In addition, we can use real-road distance instead of Euclidean distance between two zip codes. In this project, survival, complication, and travel distance are three targets. To be realistic, incorporating more targets is necessary, e.g., insurance coverage, health providers, cost, etc. The system can provide decision support for reimbursement policy making if the cost of treatment is considered because one can find out the most cost-effective institution for a specific type of patients.

For cost-effective diagnosis, most datasets in this study have binary outcomes. Clinical practice, however, is often more complex. For example, a patient who presents with chest pain may have one of several possible diseases (e.g, myocardial infarction, pulmonary embolism, chest wall pain, etc.). Our future work aims to apply ODPF

in more complex clinical settings, especially in the emergency department in which the speed of diagnosis is of critical importance. Sometimes, a health provider may order several tests simultaneously instead of one test at a time. In this case, we may want to find the most promising group of tests instead of one most promising test. In future work, we will also evaluate different search methods to find the most promising group of diagnostic tests.

For the lifestyle recommendation project, we can find the best lifestyle for patients belonging to a specific group, e.g., the patient who is taking cholesterol lowering drugs, the patient with diabetes, etc. We can also change the desired target to make other healthy recommendations, for example, recommending lifestyle to lower LDL level or to lose weight most efficiently. Another direction is to compute when a patient can start cholesterol lowering drug in order to maximally lower the risk of CVD. We can apply PODSS to comparative effectiveness research, finding the best match between a patient and a treatment option (or the combination of several). In addition, PODSS also has the potential to find or rate nursing interventions that can result in good nursing outcomes.

REFERENCES

[1] D. Aha. Feature weighting for lazy learning algorithms. In: H. Liu and H. Motoda (Eds.) Feature Extraction, Construction and Selection: A Data Mining Perspective. Springer, 1998.

[2] D. Aha and R. Goldstone. Concept learning and flexible weighting. In *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society*, pages 534–539, Bloomington, IN, 1992. Lawrence Erlbaum Associates.

[3] D.W. Aha. *Lazy Learning*. Kluwer Academic Publishers, 1997.

[4] The American Hospital Association. Data source. Accessed in Apr. 2007: http://www.thirdwaveresearch.com/aha_wizard/default.aspx.

[5] A. Asuncion and D.J. Newman. UCI machine learning repository, 2007. Accessed at http://www.ics.uci.edu/∼mlearn/MLRepository.html.

[6] C.G. Atkeson, A.W. Moore, and S. Schaal. Locally weighted learning. *Artificial Intelligence Review*, 11:11–73, 1997.

[7] J.Z. Ayanian and J.S. Weissman. Teaching hospitals and quality of care: A review of the literature. *The Milbank Quarterly*, 80(3):569–593, 2002.

[8] R.K. Bali, D.D. Feng, F. Burstein, and A.N. Dwivedi. Introduction to the special issue on advances in clinical and health-care knowledge management. *IEEE Transactions on Information Technology in Biomedicine*, 9(2):157–161, 2005.

[9] D.W. Bates, G.J. Kuperman, S. Wang, T. Gandhi, A. Kittler, L. Volk, C. Spurr, R. Khorasani, M. Tanasijevic, and B. Middleton. Ten commandments for effective clinical decision support: Making the practice of evidence-based medicine a reality. *Journal of the American Medical Informatics Association*, 10(6):523–530, 2003.

[10] J.C. Bean. Genetics and random keys for sequencing and optimization, 1993. Technical Reports 92-43. Department of Industrial and Operations Engineering, University of Michigan.

[11] J.D. Birkmeyer, A.E. Siewers, E.V.A. Finlayson, T.A. Stukel, F.L. Lucas, I. Batista, H.G. Welch, and D.E. Wennberg. Hospital volume and surgical mortality in the United States. *The New England Journal of Medicine*, 346(15):1128–1137, 2002.

[12] J.D. Birkmeyer, A.E. Siewers, N.J. Marth, and D.C. Goodman. Regionalization of high-risk surgery and implications for patient travel times. *The Journal of the American Medical Association*, 290(20):2703–2708, 2003.

[13] Body Mass Index (BMI). The description of BMI. Accessed in Apr. 2009: http://www.cdc.gov/healthyweight/assessing/bmi/adult_bmi/index.html.

[14] P.S. Bradley, O.L. Mangasarian, and W.N. Street. Feature selection via mathematical programming. *INFORMS Journal on Computing*, 10(2):209–217, 1998.

[15] D.M. Bravata, K.M. McDonald, W.M. Smith, C. Rydzak, H. Szeto, D.L. Buckeridge, C. Haberland, and D.K. Owens. Systematic review: surveillance systems for early detection of bioterrorism-related diseases. *Annals of Internal Medicine*, 140(11):910–922, 2004.

[16] E.K. Burke and G. Kendall (Editors). *Search Methodologies*. Springer, 2006.

[17] H. Byun and S.-W. Lee. A survey on pattern recognition applications of support vector machines. *International Journal of Pattern Recognition and Artificial Intelligence*, 17(3):459–486, 2003.

[18] X. Chai, L. Deng, Q. Yang, and C.X. Ling. Test-cost sensitive naive Bayes classification. In *Proceedings of the Fourth IEEE International Conference on Data Mining*, pages 51 – 58. IEEE Computer Society, 2004.

[19] J. Chen, S.S. Rathore, M.J. Radford, and H.M. Krumholz. JCAHO accreditation and quality of care for acute myocardial infarction. *Health Affairs*, 22(2):243–254, 2003.

[20] C.-L. Chi and W.N. Street. A data mining technique for risk-stratification diagnosis. In *Proceedings of the 2007 AMIA Annual Symposium*, page 909, Chicago, IL, 2007. American Medical Informatics Association.

[21] C.-L. Chi and W.N. Street. The optimal diagnostic decision sequence. In *Proceedings of the 2008 AMIA Annual Symposium*, page 902, Washington, DC, 2008. American Medical Informatics Association.

[22] C.-L. Chi, W.N. Street, and M.M Ward. Building a hospital referral expert system with a prediction and optimization-based decision support system algorithm. *Journal of Biomedical Informatics*, 41(2):371–386, 2008.

[23] W.J. Clancey. Heuristic classification. *Artificial Intelligence*, 27(3):289–350, 1985.

[24] P.D. Clayton, T.A. Pryor, O.B. Wigertz, and G. Hripcsak. Issues and structures for sharing medical knowledge among decision-making systems: The 1989 arden homestead retreat. In *Proceedings of the Annual Symposium on Computer Application in Medical Care*.

[25] F. Coenen and T.J.M. Bench-Capon. Maintenance and maintainability in regulation based systems. *ICL Technical Journal*, pages 76–84, 1992.

[26] A. Coulston, M. Feeney, and L. Hoolihan. The challenge to customize. *Journal of the American Dietetic Association*, 103(4):443–444, 2003.

[27] P.A. de Clercq, J.A. Blom, H.H. Korsten, and A. Hasman. Approaches for creating computer-interpretable guidelines that facilitate decision support. *Artificial Intelligence in Medicine*, 31(1):1–27, 2004.

[28] M.H. DeGroot and S.E. Fienberg. The comparison and evaluation of forecasters. *The Statistician*, 32:12–22, 1983.

[29] R. Detrano, A. Janosi, W. Steinbrunn, M. Pfisterer, J. Schmid, S. Sandhu, K. Guppy, S. Lee, and V. Froelicher. International application of a new probability algorithm for the diagnosis of coronary artery disease. *American Journal of Cardiology*, 64:304–310, 1989.

[30] R.A. Deyo, D.C. Cherkin, and M.A. Ciol. Adapting a clinical comorbidity index for use with ICD-9-CM administrative databases. *Journal of Clinical Epidemiology*, 45(6):613–619, 1992.

[31] P. Diaconis and B. Efron. Computer-intensive methods in statistics. *Scientific American*, 248:116–130, 1983.

[32] J.B. Dimick and G. Ailawadi. The volume-outcome effect for abdominal aortic surgery. *Archives of Surgery*, 137(7):828–832, 2002.

[33] J.B. Dimick, S.R.G. Finlayson, and J.D. Birkmeyer. Regional availability of high-volume hospitals for major surgery. *Health Affairs*, Web Exclusive:VAR45–VAR53, 2004.

[34] P. Domingos. Control-sensitive feature selection for lazy learners. *Artificial Intelligence Review*, 11:227—253, 1997.

[35] P. Domingos. Metacost: A general method for making classifiers cost-sensitive. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 155–164, San Diego, California, 1999. Association for Uncertainty in Artificial Intelligence Press.

[36] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification, Second Edition.* Wiley-Interscience, 2001.

[37] M.H. Dunham. *Data Mining: Introductory and Advanced Topics.* Prentice Hall, 2003.

[38] A. Elixhauser, C. Steiner, and I. Fraser. Volume thresholds and hospital characteristics in the United States. *Health Affairs*, 22(2):167–177, 2003.

[39] A. Elixhauser, C. Steiner, D.R. Harris, and R.M. Coffey. Comorbidity measures for use with administrative data. *Medical Care*, 36(1):8–27, 1998.

[40] A.S. Elstein and A. Schwartz. Clinical problem solving and diagnostic decision making: Selective review of the cognitive literature. *British Medical Journal*, 324:729–732, 2002.

[41] E.A. Feigenbaum. The art of artificial intelligence: I. Themes and case studies of knowledge engineering. Technical report, Stanford, CA, USA, 1977.

[42] J. Fox, N. Johns, and A. Rahmanzadeh. Disseminating medical knowledge: the proforma approach. *Artificial Intelligence in Medicine*, 14(1), 1998.

[43] J.H. Friedman, R. Kohavi, and Y. Yun. Lazy decision trees. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 717–724, Portland, OR, 1996. AAAI Press.

[44] A. Gandjour, A. Bannenberg, and K.W. Lauterbach. Threshold volumes associated with higher survival in health care: A systematic review. *Medical Care*, 41(10):1129–1141, 2003.

[45] L.G. Glance, A.W. Dick, D.B. Mukamel, and T.M. Osler. Is the hospital volume-mortality relationship in coronary artery bypass surgery the same for low-risk versus high-risk patients? *The Annals of Thoracic Surgery*, 76:1155–1162, 2003.

[46] The Leapfrog Group, 2004. Evidence-Based Hospital Referral. Accessed in Mar. 2006: http://www.leapfroggroup.org/media/file/Leapfrog-Evidence-based_ Hospital_Referral_Fact_Sheet.pdf.

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

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

[49] E.A. Halm, C. Lee, and M.R. Chassin. Is volume related to outcome in health care? A systematic review and methodologic critique of the literature. *Annals of Internal Medicine*, 137(6):511–520, 2002.

[50] Healthcare Cost & Utilization Project (HCUP). The project description and data. Accessed in Apr. 2007: http://www.ahrq.gov/data/hcup/.

[51] B.E. Hillner, T.J. Smith, and C.E. Desch. Hospital and physician volume or specialization and outcomes in cancer treatment: Importance in quality of cancer care. *Journal of Clinical Oncology*, 18(11):2327–2340, 2000.

[52] N. Howe and C. Cardie. Examining locally varying weights for nearest neighbor algorithms. In *Proceedings of the Second International Conference on Case-Based Reasoning Research and Development*, pages 455–466, Providence, RI, 1997. Springer-Verlag.

[53] C.-W. Hsu and C.-J. Lin. A comparison of methods for multi-class support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, 2002.

[54] I. Ihse. The volume-outcome relationship in cancer surgery: A hard sell. *Annals of Surgery*, 238(6):777–781, 2003.

[55] L. Irwig, P. Bossuyt, P. Glasziou, C. Gatsonis, and J. Lijmer. Evidence base of clinical diagnosis: Designing studies to ensure that estimates of test accuracy are transferable. *British Medical Journal*, 324:669–671, 2002.

[56] Y. Ishida. Using global properties for qualitative reasoning: A qualitative system theory. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 1174–1179. Morgan Kaufmann, 1989.

[57] M. Jaana, D. Wakefield, and G. Rosenthal. Access to healthcare services within the VA system: Comparison of two methods for computing travel distances for CABG patients. Poster presented at the COM/CPH Research Week, 2003.

[58] P. Jackson. *Introduction to Expert Systems, Third Edition*. Addison Wesley, Reading, MA, 1999.

[59] G.H. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In *Proceedings of the 11th International Conference on Machine Learning*, pages 121–129, San Francisco, CA, 1994. Association for Computing Machinery.

[60] A. Kapoor and R. Greiner. Learning and classifying under hard budgets. In *Proceedings of the 16th European Conference on Machine Learning*, pages 170–181, Porto, Portugal, 2005. Springer.

[61] R. Kaushal, K.G. Shojania, and D.W. Bates. Effects of computerized physician order entry and clinical decision support systems on medication safety: a systematic review. *Archives of internal medicine*, 163(12):1409–1416, 2003.

[62] L.T. Kohn, J.M. Corrigan, and M.S. Donaldson (eds). *To Err Is Human: Building a Safer Health System*. Washington, address National Academy Press, 2000.

[63] G.J. Kuperman, A. Bobb, T.H. Payne, A.J. Avery, T.K. Gandhi, G. Burns, D.C. Classen, and D.W. Bates. Medication-related clinical decision support in computerized provider order entry systems: A review. *Journal of the American Medical Informatics Association*, 14(1):29–40, 2007.

[64] J. Kupersmith. Quality of care in teaching hospitals: A literature review. *Academic Medicine*, 80(5):458–466, 2005.

[65] L. C.-K. Liau, T. C.-K. Yang, and M.-T. Tsai. Expert system of a crude oil distillation unit for process optimization using neural networks. *Expert Systems with Applications*, 26(2):247–255, 2004.

[66] A.H. Lichtenstein, L.J. Appel, M. Brands, M. Carnethon, S. Daniels, et al. Diet and lifestyle recommendations revision 2006: A scientific statement from the American Heart Association Nutrition Committee . *Circulation*, 114(1):82–96, 2006.

[67] C.X. Ling and C. Li. Data mining for direct marketing: Problems and solutions. In *Proceedings of The Fourth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 73–79. New York, NY, AAAI Press, 1998.

[68] C.X. Ling, V.S. Sheng, and Q. Yang. Test strategies for cost-sensitive decision trees. In *IEEE Transactions on Knowledge and Data Engineering*, pages 1055 – 1067. IEEE, 2006.

[69] C.X. Ling, Q. Yang, J. Wang, and S. Zhang. Decision trees with minimal costs. In *Proceedings of the Twenty-first International Conference on Machine Learning*, page 69, Banff, Canada, 2004. Association for Computing Machinery.

[70] P.J. Lisboa and A.F.G. Taktak. The use of artificial neural networks in decision support in cancer: a systematic review. *Neural Networks*, 19(4):408–415, 2006.

[71] S.L. McLafferty. Gis and health care. *Annual Review of Public Health*, 24:25–42, 2003.

[72] P. Melville, M. Saar-Tsechansky, F. Provost, and R. Mooney. Active feature-value acquisition for classifier induction. In *Proceedings of the Fourth IEEE International Conference on Data Mining*, pages 483–486, Brighton, UK, 2004. IEEE Computer Society.

[73] T.J. Menzies. Knowledge elicitation: The state of the art. In *Handbook of Software Engineering and Knowledge Engineering*. World Scientific Pub. Co., 2002.

[74] D. Michie, D.J. Spiegelhalter, and C.C. Taylor (editors). *Machine Learning, Neural and Statistical Classification*. Prentice Hall, 2004.

[75] K.G.M. Moons, G.-A. van Es, J.W. Deckers, J.D.F. Habbema, and D.E. Grobbee. Limitations of sensitivity, specificity, likelihood ratio, and Bayes' theorem in assessing diagnostic probabilities: A clinical example. *Epidemiology*, 8(1):12–17, 1997.

[76] B.K. Nallamothu, S. Saint, and T.P. Hofer. Impact of patient risk on the hospital volume-outcome relationship in coronary artery bypass grafting. *Archives of Internal Medicine*, 165(3):333–337, 2005.

[77] A. Niculescu-Mizil and R. Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22th International Conference on Machine Learning*, pages 625–632, Bonn, Germany, 2005. ACM Press.

[78] L. Ohno-Machado, J.H. Gennari, S.N. Murphy, N.L. Jain, S.W. Tu, D.E. Oliver, E. Pattison-Gordon, R.A. Greenes, E.H. Shortliffe, and G.O. Barnett. The guideline interchange format: a model for representing guidelines. *Journal of the American Medical Informatics Association*, 5(4):357–72, 1998.

[79] J.H. Park, K.H. Im, C.-K. Shin, and S.C. Park. MBNR: Case-based reasoning with local feature weighting by neural network. *Applied Intelligence*, 21(3):265—276, 2004.

[80] S.G. Pauker and J.P. Kassirer. The threshold approach to clinical decision making. *The New England Journal of Medicine*, 302:1109–1117, 1980.

[81] S. Perkins and J. Theiler. Online feature selection using grafting. In *Proceedings of the 20th International Conference on Machine Learning*, pages 592–599, Washington DC, 2003. Association for Computing Machinery.

[82] J. Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 1999.

[83] J.R. Quinlan, P.J. Compton, K.A. Horn, and L. Lazurus. Inductive knowledge acquisition: A case study. In *Proceedings of the Second Australian Conference on Applications of Expert Systems*, pages 137–156, Sydney, Australia, 1986. Addison-Wesley Longman Publishing Co., Inc.

[84] Atherosclerosis risk in communities study (ARIC). The project description and data. Accessed in Apr. 2008: http://www.cscc.unc.edu/aric/.

[85] M. Saar-Tsechansky, P. Melville, and F. Provost. Active feature-value acquisition. *Management Science. Forthcoming*, 2009.

[86] D.L. Sackett and R.B. Haynes. Evidence base of clinical diagnosis: The architecture of diagnostic research. *British Medical Journal*, 324:539–541, 2002.

[87] E.H. Shortliffe, R. Davis, S.G. Axline, B.G. Buchanan, C.C. Green, and S.N. Cohen. Computer-based consultations in clinical therapeutics: Explanation and rule acquition capabilities of the MYCIN system. *Computers and Biomedical Research*, 8(4):303–320, 1975.

[88] I. Sim, P. Gorman, R.A. Greenes, R.B. Haynes, B. Kaplan, H. Lehmann, and P.C. Tang. Clinical decision support systems for the practice of evidence-based medicine. *Journal of the American Medical Informatics Association*, 8(6):527–534, 2001.

[89] J. Smith, J. Everhart, W. Dickson, W. Knowler, and R. Johannes. Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In *Proceedings of the Symposium on Computer Applications and Medical Care*, pages 261–265, Washington, DC, 1988. IEEE Computer Society.

[90] Z. Song and A. Kusiak. Optimization of temporal processes: A model predictive control approach. *IEEE Transactions on Evolutionary Computation*, 13(1):169–179, 2009.

[91] H.C. Sox, M.A. Blatt, M.C. Higgins, and K.I. Marton. *Medical Decision Making*. Butterworth, 1988.

[92] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison Wesley, 2005.

[93] T.G. Thompson and D.J. Brailer, 2004. The decade of health information technology: Delivering consumer-centric and information-rich health care: Framework for strategic action. Washington (DC): Department of Health and Human Services, National Coordinator for Health Information Technology. Accessed in March 2007: http://www.hhs.gov/healthit/documents/hitframework.pdf.

[94] P.D. Turney. Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. *Journal of Artificial Intelligence Research*, 2:369–409, 1995.

[95] P.D. Turney. Types of cost in inductive concept learning. In *Proceedings of the International Conference on Machine Learning Conference*, pages 15–21, Stanford, CA, 2000. Morgan Kaufmann Publishers.

[96] H. Vafaie and K. DeJong. Robust feature selection algorithms. In *Proceedings of the Fifth Conference on Tools for Artificial Intelligence*.

[97] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, N.Y, 1995.

[98] M.W. Ward, M. Jaana, D.S. Wakefield, and R.L. Ohsfeldt. What would be the effect of referral to high-volume hospitals in a largely rural state? *The Journal of Rural Health*, 20(4):344–354, 2004.

[99] I. Watson and F. Marir. Case-based reasoning: A review. *The Knowledge Engineering Review*, 9(4):355–381, 1994.

[100] I.D. Watson, A. Basden, and P.S. Brandon. The client centered approach: Expert system maintenance. *Expert Systems*, 9(4):189–196, 1992.

[101] S.M. Weiss, S.J. Buckley, S. Kapoor, and S. Damgaard. Knowledge-based data mining. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 456–461, Washington, DC, 2003. Association for Computing Machinery.

[102] D. Wettschereck, D.W. Aha, and T. Mohri. A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. *Artificial Intelligence Review*, 11:273–314, 1997.

[103] P. Whiting, A.W. Rutjes, J.B. Reitsma, A.S. Glas, P.M. Bossuyt, and J. Kleijnen. Sources of variation and bias in studies of diagnostic accuracy: A systematic review. *Annals of Internal Medicine*, 140(3):189–202, 2004.

[104] P. Wright, S. Belt, and C. John. Helping people assess the health risks from lifestyle choices: Comparing a computer decision aid with customized printed alternative. *Communication and Medicine*, 1:183–192, 2004.

[105] B. Zadrozny and C. Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 694–699, Edmonton, Alberta, Canada, 2002. Association for Computing Machinery.

[106] Z. Zheng and B. Padmanabhan. On active learning for data acquisition. In *Proceedings of the 2002 IEEE International Conference on Data Mining*, pages 562–569, Maebashi City, Japan, 2002. IEEE Computer Society.

[107] V.B. Zubek and T.G. Dietterich. Pruning improves heuristic search for cost-sensitive learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 19–26, Sydney, Australia, 2002. Morgan Kaufmann Publishers.