Health Care Manag Sci (2015) 18:35–57
DOI 10.1007/s10729-014-9295-x

Accept/decline decision module for the liver simulated allocation model

Sang-Phil Kim · Diwakar Gupta · Ajay K. Israni · Bertram L. Kasiske

Received: 26 September 2013 / Accepted: 6 August 2014 / Published online: 30 August 2014

Abstract Simulated allocation models (SAMs) are used to evaluate organ allocation policies. An important component of SAMs is a module that decides whether each potential recipient will accept an offered organ. The objective of this study was to develop and test accept-or-decline classifiers based on several machine-learning methods in an effort to improve the SAM for liver allocation. Feature selection and imbalance correction methods were tested and best approaches identified for application to organ transplant data. Then, we used 2011 liver match-run data to compare classifiers based on logistic regression, support vector machines, boosting, classification and regression trees, and Random Forests. Finally, because the accept-or-decline module will be embedded in a simulation model, we also developed an evaluation tool for comparing performance of predictors, which we call sample-path accuracy. The Random Forest method resulted in the smallest overall error rate, and boosting techniques had greater accuracy when both sensitivity and specificity were simultaneously considered important. Our comparisons show that no method dominates all others on all performance measures of interest. A logistic regression-based classifier is easy to implement and allows for pinpointing the contribution of each feature toward the probability of acceptance. Other methods we tested did not have a similar interpretation. The Scientific Registry of Transplant Recipients decided to use the logistic regression-based accept-decline decision module in the next generation of liver SAM.

Keywords Organ transplantation · Classification · Machine learning · Simulation

1 Introduction

The extreme shortage of transplantable human organs in the United States leads to long waiting times and wait-list deaths for patients with end-stage organ diseases, and necessitates prioritizing matched candidates for each available organ. Pursuant to the National Organ Transplant Act (NOTA), the difficult task of setting allocation priorities rests with the Organ Procurement and Transplantation Network (OPTN), whose contractor is United Network of Organ Sharing (UNOS). The Scientific Registry of Transplant Recipients (SRTR) supports organ transplant operations by performing policy evaluation [18]. The Health Resources and Services Administration, US Department of Health and Human Services, provides oversight of the activities of the OPTN and SRTR contractors.

SRTR is tasked with development, maintenance, and distribution of computer simulation programs, called simulated

S.-P. Kim
Krannert School of Management, Purdue University,
West Lafayette, IN, USA

D. Gupta (✉)
Industrial & Systems Engineering, University of Minnesota,
Minneapolis, MN, USA
e-mail: guptad@umn.edu

A.K. Israni
Scientific Registry of Transplant Recipients, Minneapolis Medical Research Foundation; Department of Medicine, Hennepin County Medical Center; Division of Epidemiology and Community Health, School of Public Health, University of Minnesota,
Minneapolis, MN, USA

B. L. Kasiske
Department of Medicine, Hennepin County Medical Center;
Scientific Registry of Transplant Recipients, Minneapolis Medical Research Foundation, Minneapolis, MN, USA

allocation models (SAMs), used to evaluate the impact of allocation policies on organ distribution, waitlist statistics, and posttransplant outcomes. There are three SAMs, one each for assessing allocation policies for livers (LSAM), kidney-pancreas (KPSAM), and thoracic organs (TSAM). Each SAM simulates candidate and donor arrivals, candidate health status updates, and candidate accept-decline decisions regarding organs offered in the priority sequence dictated by a current or proposed allocation policy.

This study concerns a module in LSAM that predicts candidate accept or decline decisions. We refer to the module as a classifier because it classifies each matched candidate-donor pair into decline (denoted by N) or accept (denoted by Y) categories. The classifier must produce realistic results based on donor, candidate, and policy attributes that drive such decisions in practice. The accept-or-decline classifier is one of the most important SAM modules because it directly affects performance metrics that OPTN committees consider before implementing policy changes. We used historical match-run data to train and compare five classification techniques, as part of a broader program to improve SAMs.

LSAM was first developed in 2001 for OPTN. Its predecessor, the UNOS liver allocation model (ULAM) [20], was also a simulation model. ULAM's candidate-choice model was relatively simple. Each candidate accepted the offered organ with a probability that equaled the historical acceptance rate within a stratum defined by the transplant center, candidate medical urgency status, and donor quality [12]. Other simulation-based studies used matching criteria to assign organs to candidates without considering their choices [16, 21, 25].

The current LSAM includes a logistic regression (LR)-based model, similar in some respects to the model in ULAM. A major difference is that the LSAM LR model provides a unique acceptance probability for each donor and candidate pair, whereas ULAM used common acceptance rates in each stratum. Also, for non-urgent candidates, the LSAM LR model uses 147 variables consisting of donor and candidate characteristics, whereas ULAM used less than 10 variables [29].

Modeling candidate choices has been the focus of several articles in the Operations Research/Management Science (ORMS) literature, summarized in Table 1. These models consider the accept-or-decline problem faced by rational decision makers (candidates and surgeons) when organ offers are received. If the decision maker accepts the offer, a reward equal to the expected posttransplant life-years is realized. If the offer is declined, a reward equal to the expected remaining life-years of the candidate is realized, upon accounting for optimal decisions with respect to future organ offers and candidate health status evolution. A key difference between these approaches and our efforts is that we focus on predicting accept or decline decision as closely as possible to the *actual* decisions made by candidates. In contrast, ORMS literature focuses on developing prescriptive models that predict how rational candidates *should* make such decisions.

The key insight from these studies is that an optimal policy for a candidate can be described by a series of thresholds. For each fixed health status, a donor quality threshold exists such that the candidate should accept the organ if quality is above this threshold. Similarly, for each donor quality, a threshold of health status exists such that if a candidate's health status is below the threshold (worse health), the candidate should accept the organ. Each threshold typically changes monotonically. For example, the quality threshold monotonically decreases as the candidate's health status worsens. Monotonicity may not hold when rank information is considered. In such cases, rank movements can induce a candidate to deviate from monotone thresholds because of changed expectation about future offers. Models summarized in Table 1 are useful for understanding the key drivers of a decision maker's choice, but they are not suitable for implementation within LSAM for reasons discussed below.

The first difficulty in implementing normative models relates to the problem of estimating candidates' utility functions. Data are insufficient to allow this for every candidate individually. For practical reasons, candidates are grouped into subsets with the assumption that members of the same set have the same utility function. Within each group, it is typically assumed that all candidates assign the same mean utility to the same organ but that a particular candidate's actual utility has a random component with mean zero. Such a model can be shown to be equivalent to the LR model when the random component of utility function in each class has a logistic distribution. That is, the LR method already provides a practical implementation of the normative approach under certain assumptions.

The second difficulty is that decisions under ideal conditions assumed in normative models may not match actual candidate choices. Candidates may have insufficient information about an available donor to evaluate the life-years to be gained from transplant, or their life expectancy if an offer is declined. These uncertainties, coupled with possibly time-varying degree of risk aversion, can lead to different decisions even for donors who are similar with respect to characteristics available in our dataset. For these reasons, we did not consider a normative approach.

Bertsimas et al.[3] compared different classifiers on several different data sets, though not on organ transplant data. The authors calculated overall classification accuracy and found that none of the methods they considered consistently dominated others across different data sets. In our setting, the data are highly imbalanced and there are multiple

Table 1 Summary of candidate-choice models in operations research literature

Author(s)	Organ	Health status update	Queue rank known?	Monotone policy structure?	Modeling methodology
David, Yechiali [9]	Kidney	No	No	Yes	MDP*
Ahn, Hornberger [1]	Kidney	Partial†	No	Yes	MDP*
Howard [13]	Liver	Yes	No	Yes	MDP*
Su, Zenios [27]	Kidney	Partial†	No	Yes	Queueing
Su, Zenios [26]	Kidney	No	Yes	Yes	MDP*
Alagoz et al. [2]	Liver	Yes	No	Yes	MDP*
Sandikci et al. [22]	Liver	Yes	Yes	No	MDP*
Sandikci et al. [23]	Liver	Yes	Yes	No	POMDP‡

*MDP, Markov decision process.

†For candidate functional status only (waiting, underwent transplant, died). In these studies, disease progression is not modeled.

‡POMDP, partially observable Markov decision process.

measures of accuracy. Comparing classifiers across different accuracy measures gives a similar outcome as comparing classifiers across different data sets. That is, in our setting also, none of the classifiers dominates all others on all performance measures. To our knowledge, no previous study has compared accuracy of different classification methods in the context of organ transplants or across different measures of accuracy.

Greater penetration of health information technology, in particular electronic medical record systems, has made it possible to collect and store large amounts of patient-level data in easy-to-access electronic format. Health care delivery organizations are increasingly interested in utilizing these data to predict adverse health outcomes and using such predictive tools to target resources toward high-risk patients. The problem of identifying high-risk patients falls into the broad category of classification, similar to the problem we studied in this paper. We believe our study provides some generalizable insights for healthcare practitioners and analysts who may want to consider machine learning approaches for classification. We expect that the two types of data limitations we identified in this paper will be common in other healthcare data. Specifically, our data were imbalanced with many more negative responses, and there were no labels attached to the vast majority of donor-candidate pairs. The missing responses were consistently for donor-candidate pairs with lower-ranked candidates. This made it difficult to learn/impute the missing responses in a reliable fashion.

In cases similar to ours, it is important to pay attention to which measures of classifier accuracy are important and tune the classifier appropriately. This study also highlights that biased predictors may produce better overall accuracy and that the costs associated with false positives or false negatives may drive the choice of classifier. Finally, our study highlights that there may not be a single classifier that dominates others on all metrics of interest. This makes it necessary to train classifiers differently, depending on application-specific priorities.

2 Background

NOTA divided the US into 11 regions, which are further divided into approximately 58 donation service areas (DSAs), each served typically by a single organ procurement organization (OPO). Candidates registered at a transplant center within the DSA served by an OPO are considered local candidates for all donors in that DSA. Regional candidates are registered at transplant centers outside the DSA but within the DSA's region. All other candidates are classified as national candidates. According to current policy, livers are typically placed locally first, then regionally, and last nationally. Within this geographical priority system, candidates with more severe liver disease, as measured by model for end-stage liver disease (MELD) or pediatric end-stage liver disease (PELD) score, have higher priority. The highest-priority candidates are those with fulminant liver failure. They are classified as 1A (age older than 18 years) or 1B (age up to 18 years). After a donor is identified, UNOS at the request of the OPO runs a match algorithm that rank orders waitlisted candidates according to current allocation priority. The OPO then makes offers to transplant centers in the match-run order. Typically, offers are made in batches with 2 or 3 transplant centers receiving offers for a few of their highest-ranked candidates in each batch. Within each batch of offers, a successful placement occurs if one candidate accepts the organ. Otherwise, the OPO may make additional offers, exercise an expedited placement protocol, or terminate the offer process.

When a transplant center receives a tentative offer, it need not make a firm decision until its turn arrives, i.e., until all preceding candidates turn down the offer. In some centers, a staff person makes tentative initial decisions according to rules provided by transplant surgeons. These fall into two categories: N (no) or Z (provisional yes). We did not have access to rules used by any transplant center to initially classify offers into N or Z. Therefore, we are unable to comment on which rules might be used by which transplant centers. However, anecdotally, we think that such rules may relate to donor-candidate age, gender, race, height, weight, and blood-type match. When the center's turn arrives, a surgeon may reconsider a Z-classified offer and make the final decision in consultation with the candidate. Thus, the tentative Z is converted into a firm N or Y (yes). Tentative Z responses later changed to Y or N are not recorded. In the match-run data, which was used to perform the analysis reported in this paper, only the final decisions were recorded. Therefore, we did not know how many donor-candidate pairs initially received Z responses and were later converted to N. We did know that exactly one such match was converted to Y, if any. Moreover, once a Y was identified, candidates that had lower ranks and initially responded as Z typically left their responses as Z.

If a liver is to be split, which usually occurs when part of the liver is offered to a child candidate, the transplant center whose candidate first accepted the organ decides who will receive the split organ. The recipient is usually a child candidate registered at the center in question. Because of how initial responses are obtained, decisions recorded in the match-run data that come after the first Y decision are not considered reliable because such responses include both initial Z decisions and N decisions from centers that accept offers for higher-ranked candidates. Also, most matched candidates do not receive offers because the offer process terminates once a Y is identified and OPOs make offers in small batches. Candidates know their allocation priority when their center receives an offer. This priority is also called the Potential Transplant Recipient (PTR) sequence number.

The PTR sequence number is determined by the match-run algorithm, which implements the current organ allocation rules. UNOS provides the PTR sequence numbers to OPOs when a donor is identified. For each donor and each organ, the PTR sequence numbers determine each matched candidate's allocation priority. The highest priority candidate is ranked 1 and subsequent candidates' ranks increment by 1.

Table 3 Offers up to the first yes

Candidate type	Number of accepts (Y)	Number of declines (N)
All	5,722 (5.2 %)	104,019 (94.8 %)
1A	250 (15.3 %)	1,385 (84 %)
Not-1A	54,729 (5.1 %)	102,634 (94.9 %)

In this study, we focused only on the first Y decision. That is, we did not consider splitting of livers or instances in which the OPO terminated the offer process without realizing organ placement. When candidates are bypassed, they do not receive offers, which is how we model such situations. This simplifies the classification problem into a binary classification, N or Y for each offer. Because our goal was to study the relative merits of using different approaches, the restricted problem setting serves as an appropriate test case.

After certain data manipulation steps, described in detail in the Appendix, 376 possible features per matched candidate-donor are selected for study. Statistics from the 2011 liver match-run data are summarized in Tables 2 and 3.

3 Methods

With only two labels (Y or N), the problem of predicting candidate responses within LSAM is the classic binary classification problem. Our approach involves three key steps: feature selection, imbalance correction, and classification. We briefly describe methods considered for each step in this section and include additional details in the Appendix. Figure 1 shows the sequence of tasks performed.

We chose the methods described below because they performed well in previous studies. Each method was implemented in MATLAB using publicly available libraries (http://www.mathworks.com/products/matlab/).

3.1 Feature selection methods

The three categories of feature selection methods are filters, wrappers, and embedded methods [35]. Filters are easy

Table 2 Transplantable livers and transplants

Total transplanted	To 1A candidates	To 1B candidates	To Not-1A/1B candidates	Discarded livers
5,675 (89.5 %)	247 (3.9 %)	57 (0.9 %)	5371 (84.7 %)	664 (10.5 %)

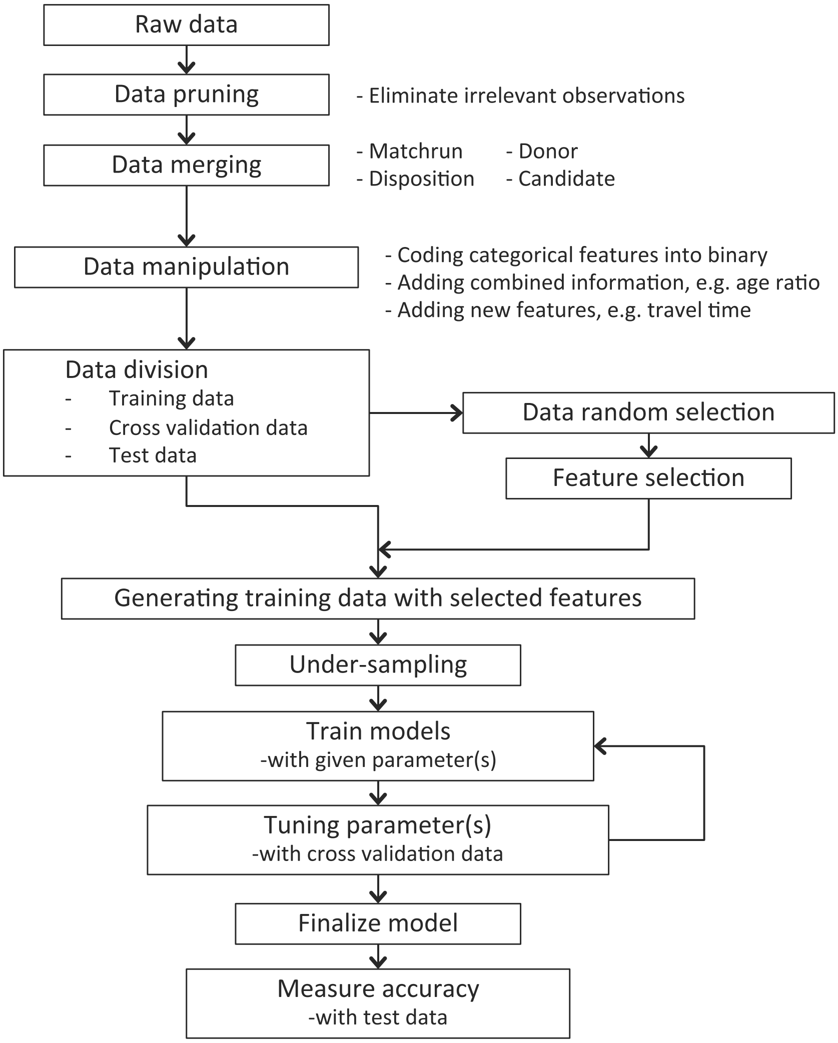

Fig. 1 Task flow chart

to use; they find the most important features or the importance score of each feature independently of the classifiers used. Wrappers find the best subset of features by performing evaluation of features during classifier training, but feature evaluation and training occur independently of each other. In embedded methods, feature selection is an integral part of training, i.e., feature selection and training occur simultaneously. Therefore, feature selection is tailored to the classification method. Wrappers and embedded methods are computationally demanding and run a high risk of over-fitting.

We compared four different filters: Fisher score [10], RELIEF [15], correlation based filter [34], and information gain [8]. Details are presented in the Appendix (see Appendix A). We chose filters because our dataset was large, with the result that wrapper and embedded approaches were not practical in our setting.

3.2 Imbalance correction methods

Because of the nature of the organ placement process, training data are highly imbalanced with many negative labels and only a few positive labels. As a result, classifiers tend to ignore classification errors of positive labels. To appreciate this problem, consider a classifier that provides all negative labels. In our setting, this classifier would be about 95 % accurate because nearly 95 % of the observations have the N label. Therefore, remedial measures are necessary.

Three widely used techniques deal with imbalanced data: under-sampling, over-sampling, and different error costs (DEC). Under-sampling discards some observations belonging to the majority class. The resulting data set has an equal number of observations for both classes. Discarded observations are selected based on redundancy and closeness to the boundary that separates the two labels [17]. Over-sampling can be considered the dual of under-sampling. It creates artificial observations for the minority class by interpolation. Sufficient numbers of artificial observations are generated to realize equal numbers of observations across the two classes [7]. The DEC technique assigns different costs to errors resulting in false negatives and false positives. Because our data include many more negatives than positives, we assign a higher cost to false negatives when using this method [31].

Upon testing, we found that the under-sampling technique worked best in our setting because it is computationally more efficient than the other methods and at least as good as other approaches with respect to classifier accuracy. It also avoids the difficulties of creating artificial observations for over-sampling or choosing error costs for the DEC approach. The specific under-sampling approach we used is described in detail in the Appendix (see Appendix B).

3.3 Classifiers

We present brief descriptions of classifiers that were tested. Additional details can be found in the cited references.

Logistic Regression (LR) [19] LR is a maximum-likelihood approach that fits the following model to the data to estimate coefficients β_i:

$$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n + \epsilon, \tag{1}$$

where x_i are the features of each matched donor-candidate pair, p is the probability of positive response, and ϵ denotes the error term. We tested two versions of LR, LR unweighted (LR[u]) and LR weighted (LR[w]). LR[w] applied a weight of 1 to negative labels and a weight equal to the ratio of the number of negative labels to the number of positive labels to the positive labels in an attempt to correct data imbalance.

Support Vector Machine (SVM) [6] SVM finds a hyper plane separating observations by label class that produces the widest gap between the closest observations of each class and the hyper plane. It has two generalizations: the soft margins approach allows misclassification with penalty; the kernel mapping method transforms the separating hyper plane into a nonlinear boundary. Gaussian kernel is the most commonly used kernel method.

Boosting (Adaptive or Gentle) [24] Boosting is an ensemble method. It generates a large number of computationally cheap and weak classifiers. Each successive weak classifier is designed to improve accuracy of prediction on misclassified observations by an earlier classifier. The predicted label is the one that receives the highest weighted sum of votes among all label classes. The difference between Adaptive boosting (AdaBoost) and Gentle boosting (GentleBoost) lies in the specification of the loss function. The loss function in Gentle boosting dampens the effect of outliers, which helps prevent over-fitting.

Classification and Regression Trees (CART) [4] The classification tree method constructs a decision tree by splitting the training data set into subsets according to the values of certain features. Splitting is repeated until only one class label remains at each node. Misclassifications are penalized. CART is the base predictor in Random Forest.

Random Forest (RF) [5] RF is also an ensemble method. At each step, it uses a random subset of observations and applies the CART methodology to this random sample. The process of bootstrapping and developing trees is continued until a user-specified number of trees is obtained. Predictions are based on votes received from each classification tree. We compared two versions of RF. In one instance, we first corrected for imbalance using under-sampling and then selected random subsets of observations. In the second (called RF-RUS), data were treated by random under-sampling and we controlled the ratio of positive and negative responses to increase both sensitivity and specificity.

4 Training objective

Many machine-learning techniques require users to choose optimal parameters by tuning the algorithm over a training sample. This step is also referred to as cross-validation. The parameters of the algorithm are tuned to realize best performance according to a user-specified objective. Because it is difficult to rank order parameter choices when users have multiple objectives, it is often necessary to specify a single aggregate training objective. For reasons explained below, the choice of an appropriate training objective is in itself a challenging task in our setting.

In order to meet OPTN objectives, the classifier embedded in LSAM needs to simultaneously perform well on several objectives. Examples of such objectives include minimize overall error rate, minimize the maximum of false positive and false negative rates, and minimize the amount by which the classifier over- or under-predicts the PTR sequence number of the first candidate who accepts each organ. The latter is important because outcome statistics such as life-years from transplant are calculated for the candidate who is projected to accept the organ in the simulated model, and such statistics are sensitive to a whole host of candidate characteristics. We also point out that it is typically the case that with a fixed dataset, higher accuracy in terms of reduced false positives can be achieved only at the expense of increased false negatives.

In our setting, the dataset available to train classifiers has several limitations, which makes it difficult to simultaneously achieve good performance on the above-mentioned metrics. The first limitation, which we mentioned earlier, is that the vast majority of valid responses are negative responses. This makes it much easier to achieve higher accuracy in predicting negative outcomes. Concomitantly, correctly predicting positive responses is much harder. In addition, the data contain valid responses only up to the first candidate who accepts the organ. That is, we do not know how lower-ranked candidates would have responded if they had received offers. A significant fraction of organs are accepted by the first few PTR sequence number candidates resulting in a highly skewed distribution of PTR sequence numbers at acceptance (see Fig. 2). This makes it difficult to learn what candidate and donor features result in the organ being rejected by high-PTR-sequence-number candidates and accepted by a particular lower-ranked candidate. In fact, if we target higher accuracy of PTR sequence numbers, then this invariably requires sacrificing accurate prediction for candidates with high PTR sequence numbers.

From many possible alternatives, we used the G-metric as the training objective in our analysis. G-metric is the square root of the product of sensitivity and specificity [28]. Sensitivity measures the proportion of positives that are correctly classified and specificity measures the proportion of negatives that are correctly classified. A high G-metric means high accuracy with respect to both positive and negative classifications simultaneously. We wish to emphasize that the classifiers we studied can be trained with respect to other objectives, depending on user preference.

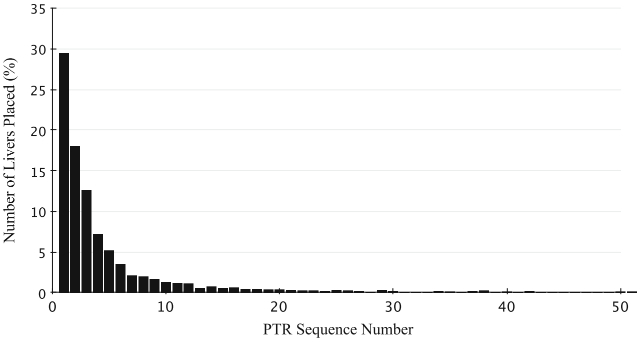

Fig. 2 Frequency of PTR sequence number at first yes

5 Performance measure

The ultimate test of a classifier in our setting is how well it performs in a simulation of current allocation policy, i.e., how well it predicts the outcome metrics of interest to OPTN relative to realized outcomes. Therefore, we focused on the attributes of the first candidate to accept an offer in reality and compared those to the attributes of the candidate who was projected to accept the same offer under simulation. We call such accuracy the sample-path accuracy because it refers to the realized features of each candidate who accepts the offer.

We introduced a measure of prediction accuracy under simulation, which we call sample-path square root of mean square errors (SMSE). To calculate the sample-path SMSE, we ran a simulation and tracked accept or decline labels in the test data for each value of the performance measure of interest. As an example, consider the subset of candidates with Y labels who have MELD score m_i. Let a_i denote the number of such donor-candidate pairs in the test data. Next, let y_{ij} denote the MELD score of the j-th first-yes candidate, so labeled by a particular classifier, among match runs to which the a_i pairs belong. Note that m_i, a_i, and y_{ij} are defined for each, $i = 1, \cdots, k$, where k is the number of categories and $j = 1, \cdots, a_i$. The sample-path SMSE is then calculated as follows.

$$SME = \sqrt{\sum_{i=1}^{k} \left(\frac{\sum_{j=1}^{a_i} (y_{ij} - m_i)^2}{a_i} \right)} \tag{2}$$

The purpose of this study is not to change clinical practice, but rather to improve a computer simulation program that predicts how current and future changes in allocation policy will affect the efficiency and fairness of the organ allocation system. As such, we are interested in predicting outcomes such as life-years from transplant, graft failures, waitlist statistics by age, gender, race, and blood type, and a whole host of other similar measures, as a function of the allocation policy. All of these measures depend on the characteristics of the candidate to whom the simulation model assigns the organ for each simulated donor arrival. For this reason, we felt compelled to develop a performance measure that captures the differences between the characteristics of the candidate who is predicted to accept an organ and the actual candidate who accepted the organ. The reason we used square root of mean square errors is that such measures have a long history in statistics. By squaring the differences, we are able to penalize large differences more than small differences.

6 Results

We first obtained rank-ordered feature sets according to each of the four feature selection methods and then tested the accuracy of each classifier trained on feature sets obtained by each feature selection method. Upon testing 4x6 combinations, we found that the information gain method resulted in the highest G-metric for all classifiers. Therefore, we used only those features that were predicted by the information gain method in all subsequent analysis. A list of the top 50 features appears in the Appendix (see Tables 13 and 15). Following each table, we also provide a correlation matrix among the top 10 features for each candidate type. The correlation structure does not affect the theoretical basis of machine learning methods, although the selection of highly correlated features does not improve prediction. Correlations among features do affect the LR model. A variety of techniques, such as Ridge regression [19], Principle Component Regression [14], and Partial Least Square Regression [33] may be employed to account for such correlations. In this study, we used the filter approach (see Section 3.1) to select features because our data were too large for other methods to be practical in conjunction with machine learning algorithms. Therefore to be consistent when comparing classifiers, we did not employ techniques to account for correlations among features just for the LR model. We recognize that all classifiers could be improved by further refining feature selection.

Note that correlations are generally low. In Table 14, moderately high correlations are observed between F1 and F2, which are PTR Sequence Number and PTR Waiting Time in Category, and between F5 and F6, which are Age Ratio and Weight Ratio. Both these are intuitively understandable. Candidates who have waited longer tend to be relatively well and therefore have a larger sequence number. Sicker patients are removed from the queue because they receive an organ, or die, or become too sick for transplant. Similarly, both donors and candidates tend to weigh more when they are older, which explains the correlation between F5 and F6. A similar pattern is observed in Table 16 as well.

Next, we performed a test in which for each of the five classification methods we selected 50, 30, and 10 top-ranked features and trained each classifier. We then used this tuned classifier on the test data to determine accuracy according to the G-metric. Results showed that the accuracy of classifiers did not increase substantially upon selecting more than 30 features for 1A and more than 10 features for not-1A candidates. For sample path comparisons, we chose the smallest subset of features (10 for not-1A and 50 for 1A) from thosesubsets that either resulted in the highest G-metric or

Table 4 Classification accuracy comparisons (best outcomes are shown in bold font)

Candidate type	Model	Error rate	Sensitivity	Specificity	G-metric
1A	LR[u]	22.35 %	28.57 %	84.56 %	49.15 %
	LR[w]	32.94 %	61.90 %	67.79 %	64.78 %
	SVM	40.59 %	80.95 %	56.38 %	67.56 %
	AdaBoost	42.35 %	95.24 %	52.35 %	70.61 %
	GentleBoost	38.24 %	80.95 %	59.06 %	69.15 %
	Random Forest	37.06 %	80.95 %	60.40 %	69.93 %
	CART	47.06 %	66.67 %	51.01 %	58.31 %
	RF-RUS	21.18 %	47.62 %	83.22 %	62.95 %
Not-1A	LR[u]	9.06 %	17.63 %	95.14 %	40.95 %
	LR[w]	30.78 %	70.68 %	69.14 %	69.90 %
	SVM	24.27 %	87.81 %	75.08 %	81.20 %
	AdaBoost	15.23 %	86.00 %	84.70 %	85.34 %
	GentleBoost	14.50 %	86.16 %	85.46 %	85.81 %
	Random Forest	14.89 %	85.67 %	85.08 %	85.37 %
	CART	19.37 %	74.79 %	80.96 %	77.82 %
	RF-RUS	6.98 %	47.45 %	95.63 %	67.36 %

CART = classification and regression trees; LR[u] = logistic regression un-weighted

LR[w] = logistic regression weighted

RF-RUS = Random Forest method with data not first treated for imbalance

SVM, support vector machines

Number of features: 10 for not-1A, 50 for 1A were within 0.5 % of the highest G-metric. Parsimonious models are preferred because they avoid over-fitting.

The top 10 features for not-1A candidates are: Donor/Candidate Weight Ratio, Donor/Candidate Height Ratio, Urgency Status Date (number of days since achieving urgent status), Waiting Time Category (waiting time points awarded), Match MELD, Total score for the candidate on the match (refers to points related to ABO match and wait time), Donor/Candidate Age ratio PTR Sequence Number, Candidate's Weight, and Minimum acceptable Donor Weight. Note that the numbers of transplant centers within 100, 200 and 300 miles of the donor hospital, which we believed would be correlated with cold ischemic time, were also relevant but not in the top-10 set of features. The fact that these attributes are most informative in determining a Y or N classification is highly relevant. Whereas it does not demonstrate a causal relationship, it does inform that in the aggregate these features are correlated with candidates' decisions. In ORMS literature, donor characteristics and candidate health status are modeled as separate scalar quantities. Our investigation reveals that there is interaction between donor and candidate characteristics for the purpose of understanding candidates' accept or decline decisions.

Accuracy of each classifier for not-1A and 1A candidates is shown in Table 4. The RF-RUS method had the smallest error rate, but boosting techniques yield the highest G-metric. Figures 3 and 4 show non-dominated classifiers when both sensitivity and specificity are considered together. RF-RUS, LR[w], and AdaBoost are not dominated for 1A candidates, and RF-RUS, GentleBoost, and SVM are not dominated for 1A candidates. Figure 5 shows sample path accuracy relative to the average performance of all classifiers for each performance metric of interest. Positive (negative) values indicate better (worse) than average performance. The best classifiers by performance metric are: GentleBoost for MELD and RF-RUS for PTR number, LRN, and Status. Actual values of SMSE are presented in the Appendix (see Table 12).

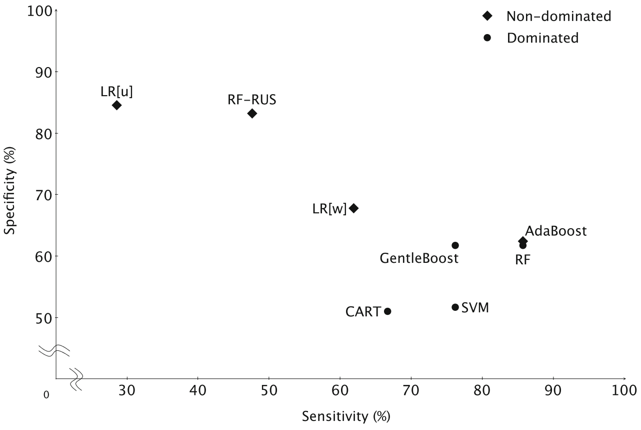

Fig. 3 Sensitivity-specificity plot for 1A candidates. CART, classification and regression trees; LR[u], logistic regression un-weighted; LR[w], logistic regression weighted; RF, Random Forest; RF-RUS, Random Forest method with data not first treated for imbalance; SVM, support vector machines

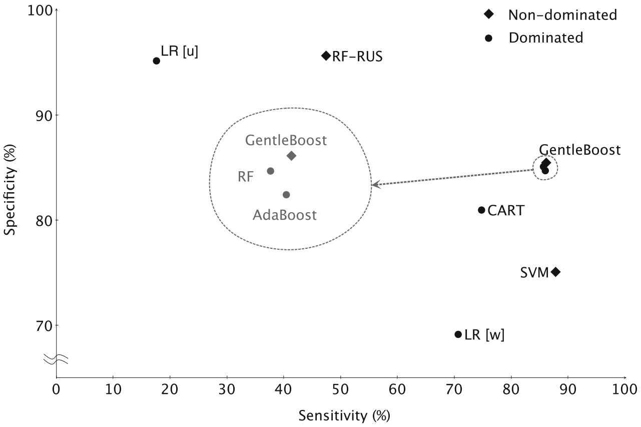

Fig. 4 Sensitivity-specificity plot for not-1A candidates. CART, classification and regression trees; LR[u], logistic regression un-weighted; LR[w], logistic regression weighted; RF, Random Forest; RF-RUS, Random Forest method with data not first treated for imbalance; SVM, support vector machines

7 Discussion

The final choice of a classifier depends on the preferred accuracy criterion and the importance of other factors such as ease of implementation and interpretation. The LR method is easy to implement and allows pinpointing the contribution of each feature toward the probability of acceptance. Machine-learning methods do not have a similar interpretation. Although machine-learning algorithms can be optimized to run quite fast, they still consume more computational effort than LR implementation. SRTR decided to use the LR method in the next generation of LSAM, after optimizing it further.

In addition to improving LSAM, classifiers developed in this study can be used to support operational decisions. For example, OPOs can be given information regarding the chances of placing an organ within a prespecified number of offers. This can help guide OPOs in making expedited placements and bypassing placement decisions when such decisions could result in a greater chance of successful placement. Because our models include variables that were not considered in previous models, e.g., PTR sequence number; number of transplant centers within a 100, 200, and 300 mile radius; and estimated travel times [11], we believe our predictive models can be useful to OPOs in making offer decisions. Classification methods that do not depend on candidate geographic location information can also help illuminate the effect of different local, regional, and national priority schemes.

8 Conclusion

This study is the first to perform a systematic evaluation of different classification methods for possible application in organ transplant operations. Using real data from 2011, we selected best features, corrected for imbalance, trained classifiers, and evaluated classification and sample-path accuracy. We found that information gain was the best method for feature selection, and that under-sampling was the most suitable method for correcting imbalance. We did not find a classifier that dominated all others according to classification and sample-path measures of accuracy. Efforts to improve accuracy for each performance metric involve tradeoffs. We find that greater overall accuracy typically implies greater prediction bias, i.e., significantly higher accuracy in predicting negative as opposed to positive responses.

Acknowledgements This work was conducted under the auspices of the Minneapolis Medical Research Foundation, contractor for the Scientific Registry of Transplant Recipients (SRTR), as a deliverable under contract no. HHSH250201000018C (US Department of Health and Human Services, Health Resources and Services Administration, Healthcare Systems Bureau, Division of Transplantation). As a US Government-sponsored work, there are no restrictions on its use. The views expressed herein are those of the authors and not necessarily those of the US Government. The authors thank University of Minnesota colleague Zhi Zhang for assistance with data preparation and interpretation, and SRTR colleagues Jon J. Snyder for valuable contributions to the study; David Schladt, Sally Gustafson, Eugene Shteyn, and Xinyue Wang for assistance with data preparation and interpretation; Susan Leppke for project assistance; Delaney Berrini for manuscript preparation; and Nan Booth for manuscript editing.

Appendix

In this appendix, we explain methods used in our paper for binary classification. These techniques are also applicable to multi-class problems. In addition, we provide cross-validation results on training sample and performance comparisons on test data in Tables 6 7 8 9 10 11 in the Appendix. Table 12 shows sample path SMSE comparisons for different performance criteria. Tables 13 and 15 contain lists of top 50 features selected by the Information Gain feature selection method for 1A and not-1A candidates respectively. These tables are provided in support of the abridged results presented in the paper. We begin by describing data coding procedures we used in this study.

All potentially relevant attributes associated with each donor-candidate pair are called *features*. We coded categorical variables with several categories into binary variables, e.g., race with 8 basic categories and 3 additional options for missing, multi-racial, and unknown categories, was coded into 11 0-1 variables. This resulted in an original data set of 554 features from three sources: match run, candidate, and donor features. We eliminated those features that were either empty or were not known to the candidates at the time of receiving an offer. For example, only those lab values were retained that were received before the offer date. We also removed certain features that could lead to over-fitting to the existing allocation policy. For example, candidate designation as local, regional, or national, relative to the donor OPO was not considered. Similarly, TxC indices were dropped. We instead created several variables that could explain TxC-specific strategies. In particular, we calculated and included the number of TxCs within 100, 200, and 300 miles of the donor hospital to capture the effect of competition among TxCs. We also included travel times between donor hospital and TxC where each candidate was located [11].

We coded certain features differently from how they appeared in the original data. For example, surgeons indicated that the donor and candidate weight, height, and age ratios were important to them in making accept or decline decisions. Therefore, we created features that consisted of these ratios and removed redundant features.

After the above-mentioned data manipulation steps, our data consists of 3,140 livers transplanted from deceased donors with n = 59,510 valid observations (donor-patient pairs with Y/N responses) from 2011 match-run data, each comprising a vector $\mathbf{x} = \{x_1, x_2, ..., x_k\}$ of donors' and candidates' characteristics and a label y that is either 1 (for accept) or -1 (for decline). The parameter $k = 376$ denotes the number of relevant characteristics. Throughout this appendix, we also use $(\mathbf{x}_i, y_i)$ to denote the i-th observation. Similarly, the notation x_{ij} refers to the value of the jth feature in the ith observation. There are four sections in this appendix, which provide, respectively, the mathematical underpinnings of feature selection and imbalance correction methods, information about MATLAB implementation of different methods, and cross validation and test results.

Appendix A: Feature selection

As described in the paper, we focus on filter methods [35]. In what follows, we describe each of the four methods we compared in the paper.

A.1 Fisher score

Fisher score measures the dissimilarity of feature values across different classes. Since it treats features one at a

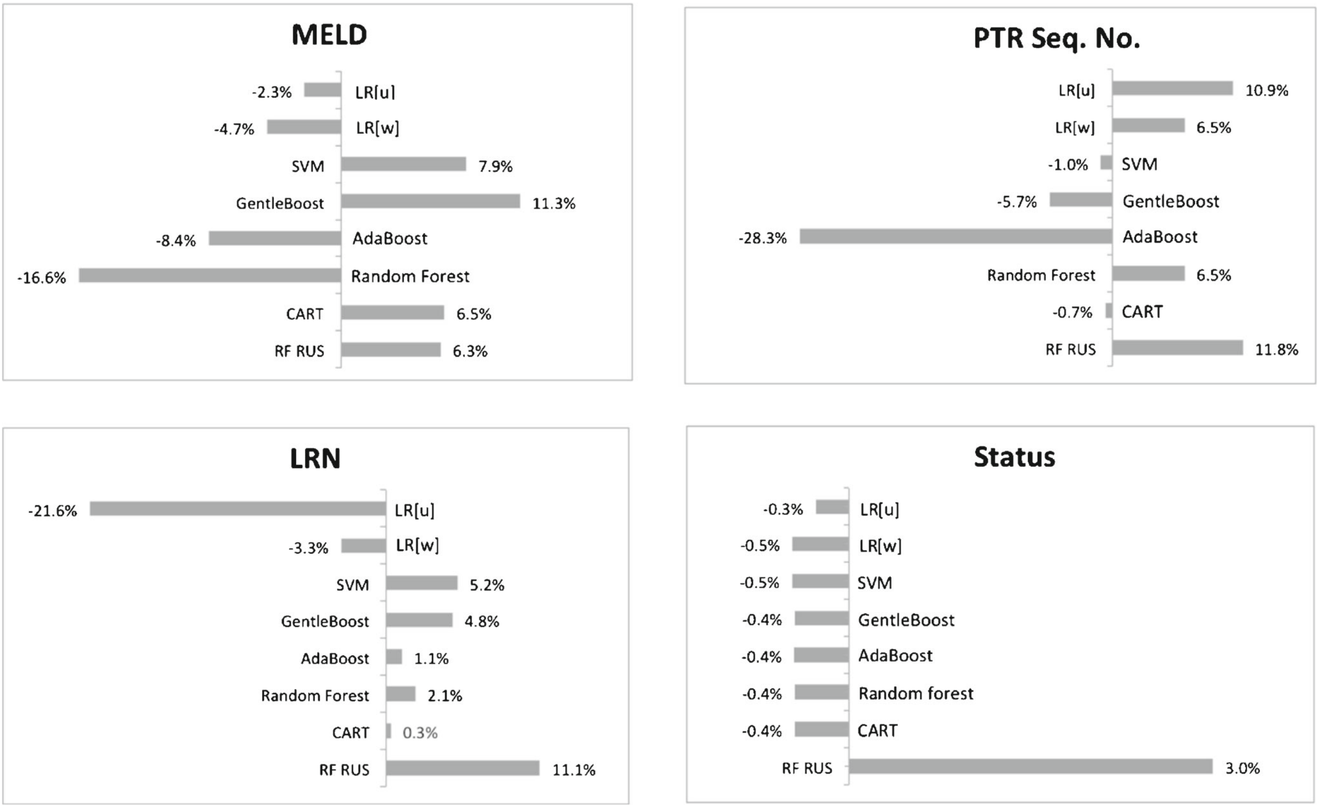

Fig. 5 Relative sample path performance. CART, classification and regression trees; LRN, local, regional, or national; LR[u], logistic regression un-weighted; LR[w], logistic regression weighted; MELD, model for end-stage renal disease; PTR, potential transplant recipient; RF, Random Forest; RF-RUS, Random Forest method with data not first treated for imbalance; Status, 1A, 1B, or MELD candidate; SVM, support vector machines

time,it cannot consider the redundancy of features [10]. It calculates a score for each feature. Features with higher scores are preferred because a higher score implies greater differences in feature values across different label classes. For a feature labeled f_j, its Fisher Score $FS(f_j)$ is calculated as follows:

$$FS(f_j) = \frac{\sum_{z=1}^{2} n_z(\mu_{j,z} - \mu_j)^2}{\sum_{j=1}^{2} n_z \sigma_{j,z}^2}, \qquad \text{where} \qquad (3)$$

μ_j: the mean value of the feature f_j for all classes,
n_z: the number of samples in the zth class,
$\mu_{j,z}$: the mean value of f_j within class z, and
$\sigma_{j,z}$: the variance of f_j values within class z.

A.2 Relief

Relief is a weighting method in which the weights are determined by measuring distances between the value of a feature for a particular observation and its value for the nearest same and different class observations [15]. The feature that provides the widest overall gap earns the highest weight. In particular, the score of a feature is calculated as follows:

$$RS(f_j) = \frac{1}{2}\sum_{i=1}^{n} \text{diff}(x_{ij} - f_{NM(x_{ij})}) - \text{diff}(x_{ij} - f_{NH(x_{ij})}), \qquad (4)$$

where x_{ij} is the value of jth feature of ith instance x_i, $f_{NH(x_{ij})}$ and $f_{NM(x_{ij})}$ are the values on the jth feature of nearest points to x_i with the same and different class label, respectively [35]. The function diff$(\cdot)$ is the distance measurement defined as following. When u and v are nominal variables,

$$\text{diff}(u, v) = \begin{cases} 0 \text{ if } u = v \\ 1 \text{ otherwise.} \end{cases} \qquad (5)$$

When u and v are numerical variables,

$$\text{diff}(u, v) = \frac{|u - v|}{Nu}, \qquad (6)$$

where Nu is an appropriately sized normalization unit needed to make the difference lie in the interval [0,1].

A.3 Information Gain (IG)

Information Gain measures the dependency between the feature and the class label [8]. It is widely used because it is simple to calculate and easy to interpret. In order to explain Information Gain, we first define the concept of entropy. Entropy $R(U)$ of random variable $U \in \mathcal{U}$ with probability mass function $p(u)$ is defined by

$$R(U) = -\sum_{u \in \mathcal{U}} p(u) \log_2 p(u). \tag{7}$$

Entropy $R(U)$ is a measure of uncertainty of the random variable U. Similarly, conditional entropy $R(U|V)$ is defined as

$$\begin{aligned} R(U|V) &= \sum_{u \in \mathcal{U}} p(u) R(V|U=u) && (8)\\ &= -\sum_{u \in \mathcal{U}} P(u) \sum_{y \in \mathcal{V}} p(v|u) \log p(v|u) && (9)\\ &= -\sum_{u \in \mathcal{U}} \sum_{v \in \mathcal{V}} p(u,v) \log p(v|u) && (10)\\ &= \mathbf{E} \log p(V|U) && (11) \end{aligned}$$

With these definitions in hand, the Information Gain (IG) of feature f_j for class label Y is calculated as follows:

$$IG(X_{f_j}, Y) = R(X_{f_j}) - R(X_{f_j}|Y) \tag{12}$$

For each label class, features are ranked in terms of Information Gain. Similar to Fisher score, IG considers each feature one at a time. Therefore, it cannot eliminate redundant features.

A.4 Correlation Based Filter (CBF)

This method uses the correlations between features and class labels and between features to eliminate redundant features [34]. There are several ways to measure correlation between features and labels, and between features. We present Fast CBF, which we used in the paper. In this method symmetrical uncertainty (SU) is employed to measure correlation. SU is defined as follows:

$$SU(\mathbf{u}, \mathbf{v}) = 2\left[\frac{IG(\mathbf{u}, \mathbf{v})}{R(\mathbf{u}) + R(\mathbf{v})}\right] \tag{13}$$

where $R(\mathbf{u})$ and $IG(\mathbf{u}, \mathbf{v})$ is defined (7) and (12), respectively.

Table 5 The CBF algorithm

```
1. for $j$ = 1 to k
        calculate $SU(x_{f_j}, y)$ for feature $f_j$
        if $SU(x_{f_j}, y) \geq \delta$ , append feature $j$ to $S_{list}$
2. end
3. sort $S_{list}$ in descending $SU(x_{f_j}, y)$
4. set base feature, $f_b$ = first element of $S_{list}$
5. Do
        $f_q = f_b$
        while ($f_q \neq$ NULL)
          $f_q$ = next element of $S_{list}$ after $f_q$
            if $SU(x_{f_q}, x_{f_b}) \geq SU(x_{f_q}, y)$ , remove $f_q$ from $S_{list}$
          end
        $f_b$ = next element of $S_{list}$ after $f_b$
6.end
7.$S_{best} = S_{list}$
```

The algorithm used by the CBF method is presented in Table 5. At the beginning, CBF chooses a set of features according to SU between features and class labels. If the SU between a feature and labels is higher than a predefined threshold (δ), then the feature is selected as a predominant feature, i.e., it then belongs to the set S_{list}. All features that are not selected as predominant features are discarded. Then, the feature with the highest SU with a class label becomes a base feature (f_b). If the SU between a feature(f_q) and the base feature(f_b) is higher than that between a feature(f_q) and class label, then that feature is dropped from the list. The next highest feature in the predominant list then becomes the base feature and the above process is repeated until no additional features can be dropped. All other methods described in this appendix provide a relative rank of each feature, leaving the task of selecting which features to use for tuning classifiers to the analyst. In contrast, CBF actually selects a certain number of features by itself.

Appendix B: Treating imbalanced data: under-sampling

Because we have many more observations with the N label, we use under-sampling to balance the number of observations within each label class. It is possible to use random selection to decide which N-labeled observations to keep in the training data. However, there exist better methods that select important observations. We describe one such approach [17]. In this approach, we remove two kinds of

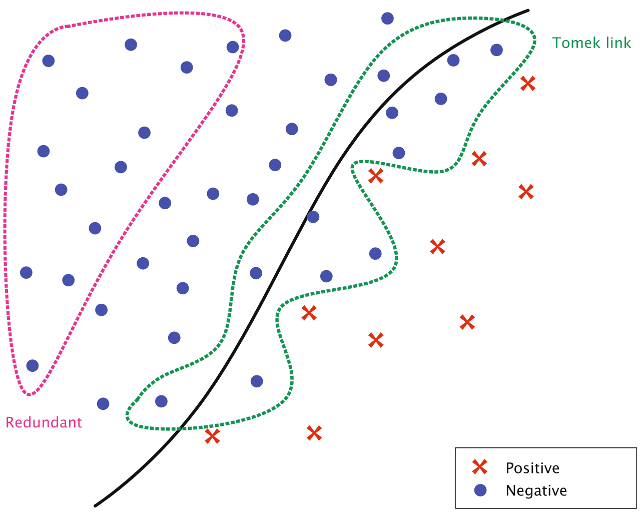

Fig. 6 Under-Sampling; Observations in the Two Envelopes are Removed

observations: redundant and Tomek links. Redundant observations do not hurt the classifier, but increase classification error cost of the N-class labels. In contrast, Tomek links are observations located near the boundary; see Fig. 6 for a graphical representation. These observations affect the classifier and their noisiness can lower the accuracy of the classifier if they are not removed.

Redundant observations are identified by using the One Nearest Neighbor (denoted 1-NN) method. 1-NN is one of the simplest classification methods that finds the nearest observation from each tagged observation and assigns to it the same label as that of its nearest neighbor. For implementing 1-NN, distance is calculated by the Value Difference Metric (VDM), which is shown below [32].

$$\begin{aligned}\mathrm{VDM}_{f_i}(u, v) &= \sum_{c=1}^{2}\left|\frac{N_{f_i,u,c}}{N_{f_i,u}} - \frac{N_{f_i,v,c}}{N_{f_i,v}}\right|^q \\ &= \sum_{c=1}^{2}\left|P_{f_i,u,c} - P_{f_i,v,c}\right|^q, \qquad (14)\end{aligned}$$

where

- $N_{f_i,u}$: the number of instances in the training set (T) that have value u for feature f_i,
- $N_{f_i,u,c}$: the number of instances in T that have value u for feature f_i and class c,
- c: output class. $c \in \{1, 2\}$, for binary classification.
- q: a constant, usually 1 or 2, and
- $P_{f_i,u,c}$: conditional probability that the output class is c given that the attribute f_i has value u, $P_{f_i,u,c} = P(c|u, f_i)$.

Finally, $P_{f_i,u,c}$ is defined as

$$P_{f_i,u,c} = \frac{N_{f_i,u,c}}{N_{f_i,u}}, \qquad (15)$$

and $N_{f_i,u}$ is the sum of $N_{f_i,u}$ over all classes. That is,

$$N_{f_i,u} = \sum_{c=1}^{C} N_{f_i,u,c}. \qquad (16)$$

Tomek links are defined as follows. Let $\delta(\mathbf{x}_i, \mathbf{x}_j)$ represent the distance between observations $\mathbf{x}_i$ and $\mathbf{x}_j$ with different class labels. The pair $(\mathbf{x}_i, \mathbf{x}_j)$ is called a Tomek link if there is no observation $\mathbf{x}_k$ such that $\delta(\mathbf{x}_i, \mathbf{x}_k) < \delta(\mathbf{x}_i, \mathbf{x}_j)$ and $\delta(\mathbf{x}_k, \mathbf{x}_j) < \delta(\mathbf{x}_i, \mathbf{x}_j)$ [30].

With these definitions in hand, the under-sampling algorithm has the following steps: (i) create a subset S with all the Y labeled observations, (ii) select one N-labeled observation at random and apply 1-NN, (iii) if the 1-NN observation is misclassified, then add the N-labeled observation to set S, else remove it from further consideration, (iv) repeat the procedure by selecting each N-labeled observation one by one until all such observations are considered, and finally (v) among data points in subset C, find Tomek-link observations and remove them. The resulting set S is the under-sampled training data set.

Appendix C: MATLAB Code

MATLAB codes we used for feature selection are available from Feature Selection Group's web site at the Arizona State University (featureselection.asu.edu).

- Fisher score: fsFisher.m
- Relief: fsReliefF.m
- CBF: fsFCBF.m
- Information Gain: fsInfoGain.m

We used MATLAB(R2012b) functions from statistics toolbox for classifiers (details can be found at the following URL: www.mathworks.com/products/statistics/description4.html).

- Logistic Regression : glmfit.m with the option, 'binomial'
- SVM : svmtrain.m
- Boosting : fitensemble.m with the options, 'AdaBoostM1' and 'GentleBoost'
- RF : treebagger.m
- CART : classregtree.m

Appendix D: Cross-validation and test results

This Appendix contains cross-validation and test results, as well as top features identified by the Information

Table 6 Cross-validation results for 1A

CM	FS	# of Features	Error rate	Accuracy	Sensitivity	Specificity	G-metric
LR[u]	Fisher score	50	29.26 %	70.74 %	26.47 %	80.52 %	46.17 %
		30	30.32 %	69.68 %	32.35 %	77.92 %	50.21 %
		20	18.09 %	81.91 %	0.00 %	100.00 %	0.00 %
		10	73.40 %	26.60 %	97.06 %	11.04 %	32.73 %
	RELIEF	50	31.91 %	68.09 %	32.35 %	75.97 %	49.58 %
		30	27.66 %	72.34 %	14.71 %	85.06 %	35.37 %
		20	25.00 %	75.00 %	41.18 %	82.47 %	58.27 %
		10	25.53 %	74.47 %	29.41 %	84.42 %	49.83 %
	CBF	12	27.66 %	72.34 %	17.65 %	84.42 %	38.60 %
	Information gain	50	27.13 %	72.87 %	38.24 %	80.52 %	55.49 %
		30	29.79 %	70.21 %	23.53 %	80.52 %	43.53 %
		20	27.13 %	72.87 %	23.53 %	83.77 %	44.40 %
		10	29.26 %	70.74 %	29.41 %	79.87 %	48.47 %
LR[w]	Fisher score	50	27.13 %	72.87 %	61.76 %	75.32 %	68.21 %
		30	40.43 %	59.57 %	67.65 %	57.79 %	62.53 %
		20	42.55 %	57.45 %	67.65 %	55.19 %	61.10 %
		10	46.28 %	53.72 %	55.88 %	53.25 %	54.55 %
	RELIEF	50	42.55 %	57.45 %	47.06 %	59.74 %	53.02 %
		30	44.15 %	55.85 %	32.35 %	61.04 %	44.44 %
		20	43.09 %	56.91 %	52.94 %	57.79 %	55.31 %
		10	55.85 %	44.15 %	52.94 %	42.21 %	47.27 %
	CBF	12	44.15 %	55.85 %	35.29 %	60.39 %	46.17 %
	Information gain	50	38.30 %	61.70 %	52.94 %	63.64 %	58.04 %
		30	37.23 %	62.77 %	52.94 %	64.94 %	58.63 %
		20	34.57 %	65.43 %	67.65 %	64.94 %	66.28 %
		10	48.40 %	51.60 %	38.24 %	54.55 %	45.67 %
SVM	Fisher score	50	30.32 %	69.68 %	73.53 %	68.83 %	71.14 %
		30	51.06 %	48.94 %	88.24 %	40.26 %	59.60 %
		20	76.06 %	23.94 %	97.06 %	7.79 %	27.50 %
		10	79.26 %	20.74 %	97.06 %	3.90 %	19.45 %
	RELIEF	50	35.11 %	64.89 %	64.71 %	64.94 %	64.82 %
		30	42.02 %	57.98 %	67.65 %	55.84 %	61.46 %
		20	38.30 %	61.70 %	64.71 %	61.04 %	62.85 %
		10	48.40 %	51.60 %	61.76 %	49.35 %	55.21 %
	CBF	12	46.28 %	53.72 %	79.41 %	48.05 %	61.77 %
	Information gain	50	43.09 %	56.91 %	82.35 %	51.30 %	65.00 %
		30	45.21 %	54.79 %	82.35 %	48.70 %	63.33 %
		20	47.87 %	52.13 %	82.35 %	45.45 %	61.18 %
		10	46.81 %	53.19 %	88.24 %	45.45 %	63.33 %
AdaBoost	Fisher score	50	42.55 %	57.45 %	73.53 %	53.90 %	62.95 %
		30	40.43 %	59.57 %	73.53 %	56.49 %	64.45 %
		20	81.38 %	18.62 %	100.00 %	0.65 %	8.06 %
		10	81.38 %	18.62 %	100.00 %	0.65 %	8.06 %
	RELIEF	50	37.77 %	62.23 %	61.76 %	62.34 %	62.05 %
		30	40.96 %	59.04 %	50.00 %	61.04 %	55.24 %
		20	36.70 %	63.30 %	55.88 %	64.94 %	60.24 %
		10	43.62 %	56.38 %	55.88 %	56.49 %	56.19 %
	CBF	12	38.83 %	61.17 %	58.82 %	61.69 %	60.24 %
	Information gain	50	44.15 %	55.85 %	85.29 %	49.35 %	64.88 %
		30	42.55 %	57.45 %	79.41 %	52.60 %	64.63 %
		20	41.49 %	58.51 %	76.47 %	54.55 %	64.58 %
		10	38.83 %	61.17 %	70.59 %	59.09 %	64.58 %

Table 7 Cross-validation results for 1A - continued

CM	FS	# of Features	Error rate	Accuracy	Sensitivity	Specificity	G-metric
GentleBoost	Fisher score	50	36.17 %	63.83 %	67.65 %	62.99 %	65.28 %
		30	47.34 %	52.66 %	73.53 %	48.05 %	59.44 %
		20	76.60 %	23.40 %	100.00 %	6.49 %	25.48 %
		10	79.79 %	20.21 %	100.00 %	2.60 %	16.12 %
	RELIEF	50	39.89 %	60.11 %	58.82 %	60.39 %	59.60 %
		30	40.96 %	59.04 %	55.88 %	59.74 %	57.78 %
		20	37.23 %	62.77 %	61.76 %	62.99 %	62.37 %
		10	43.09 %	56.91 %	58.82 %	56.49 %	57.65 %
	CBF	12	33.51 %	66.49 %	70.59 %	65.58 %	68.04 %
	Information gain	50	43.09 %	56.91 %	76.47 %	52.60 %	63.42 %
		30	43.62 %	56.38 %	61.76 %	55.19 %	58.39 %
		20	45.21 %	54.79 %	67.65 %	51.95 %	59.28 %
		10	44.68 %	55.32 %	58.82 %	54.55 %	56.64 %
Random forest	Fisher score	50	35.64 %	64.36 %	70.59 %	62.99 %	66.68 %
		30	43.62 %	56.38 %	73.53 %	52.60 %	62.19 %
		20	76.06 %	23.94 %	97.06 %	7.79 %	27.50 %
		10	79.26 %	20.74 %	97.06 %	3.90 %	19.45 %
	RELIEF	50	37.23 %	62.77 %	70.59 %	61.04 %	65.64 %
		30	38.83 %	61.17 %	70.59 %	59.09 %	64.58 %
		20	40.43 %	59.57 %	73.53 %	56.49 %	64.45 %
		10	43.62 %	56.38 %	61.76 %	55.19 %	58.39 %
	CBF	12	36.17 %	63.83 %	61.76 %	64.29 %	63.01 %
	Information gain	50	42.02 %	57.98 %	79.41 %	53.25 %	65.03 %
		30	42.55 %	57.45 %	76.47 %	53.25 %	63.81 %
		20	39.36 %	60.64 %	76.47 %	57.14 %	66.10 %
		10	40.43 %	59.57 %	58.82 %	59.74 %	59.28 %
CART	Fisher score	50	50.00 %	50.00 %	76.47 %	44.16 %	58.11 %
		30	48.94 %	51.06 %	58.82 %	49.35 %	53.88 %
		20	81.38 %	18.62 %	100.00 %	0.65 %	8.06 %
		10	81.38 %	18.62 %	100.00 %	0.65 %	8.06 %
	RELIEF	50	34.04 %	65.96 %	44.12 %	70.78 %	55.88 %
		30	33.51 %	66.49 %	58.82 %	68.18 %	63.33 %
		20	48.40 %	51.60 %	67.65 %	48.05 %	57.01 %
		10	45.21 %	54.79 %	52.94 %	55.19 %	54.06 %
	CBF	12	39.36 %	60.64 %	44.12 %	64.29 %	53.26 %
	Information Gain	50	43.62 %	56.38 %	61.76 %	55.19 %	58.39 %
		30	43.62 %	56.38 %	61.76 %	55.19 %	58.39 %
		20	42.55 %	57.45 %	61.76 %	56.49 %	59.07 %
		10	47.87 %	52.13 %	58.82 %	50.65 %	54.58 %

Gain feature selection method. Many machine learning models have one or more control parameters, e.g., σ in kernel for SVM. Before the final evaluation of our final model, we need to tune these control parameters. We divided the data into three groups; training (60 %), cross-validation(20 %), and test (20 %)

Table 8 Cross-validation results for Not-1A

CM	FS	# of Features	Error rate	Accuracy	Sensitivity	Specificity	G-metric
LR[u]	Fisher score	50	8.76 %	91.24 %	9.09 %	95.37 %	29.45 %
		30	9.38 %	90.62 %	6.73 %	94.84 %	25.27 %
		20	9.35 %	90.65 %	4.88 %	94.96 %	21.53 %
		10	9.29 %	90.71 %	4.55 %	95.04 %	20.78 %
	RELIEF	50	8.59 %	91.41 %	16.50 %	95.18 %	39.63 %
		30	8.98 %	91.02 %	9.26 %	95.13 %	29.68 %
		20	9.33 %	90.67 %	6.57 %	94.90 %	24.96 %
		10	9.07 %	90.93 %	6.73 %	95.16 %	25.31 %
	CBF	8	8.93 %	91.07 %	11.62 %	95.07 %	33.23 %
	Information Gain	50	8.91 %	91.09 %	20.88 %	94.62 %	44.44 %
		30	6.83 %	93.17 %	20.54 %	96.82 %	44.59 %
		20	8.81 %	91.19 %	19.87 %	94.78 %	43.39 %
		10	9.27 %	90.73 %	18.52 %	94.36 %	41.80 %
LR[w]	Fisher score	50	40.53 %	59.47 %	59.09 %	59.49 %	59.29 %
		30	49.83 %	50.17 %	52.69 %	50.04 %	51.35 %
		20	50.29 %	49.71 %	50.17 %	49.69 %	49.93 %
		10	49.26 %	50.74 %	52.53 %	50.65 %	51.58 %
	RELIEF	50	33.24 %	66.76 %	68.69 %	66.66 %	67.66 %
		30	41.79 %	58.21 %	56.90 %	58.27 %	57.58 %
		20	43.93 %	56.07 %	53.54 %	56.20 %	54.85 %
		10	44.55 %	55.45 %	55.56 %	55.45 %	55.50 %
	CBF	8	38.73 %	61.27 %	57.91 %	61.44 %	59.65 %
	Information gain	50	31.63 %	68.37 %	73.57 %	68.11 %	70.79 %
		30	31.83 %	68.17 %	68.18 %	68.16 %	68.17 %
		20	32.70 %	67.30 %	70.20 %	67.15 %	68.66 %
		10	35.07 %	64.93 %	69.36 %	64.71 %	67.00 %
SVM	Fisher score	50	33.20 %	66.80 %	74.07 %	66.47 %	70.17 %
		30	92.01 %	7.99 %	99.83 %	3.38 %	18.36 %
		20	92.34 %	7.66 %	99.83 %	3.03 %	17.39 %
		10	92.55 %	7.45 %	100.00 %	2.80 %	16.74 %
	RELIEF	50	23.83 %	76.17 %	75.55 %	76.23 %	75.89 %
		30	25.52 %	74.48 %	49.41 %	75.76 %	61.18 %
		20	32.33 %	67.67 %	56.66 %	68.24 %	62.18 %
		10	30.92 %	69.08 %	53.37 %	69.87 %	61.06 %
	CBF	8	16.86 %	83.14 %	45.45 %	85.04 %	62.17 %
	Information gain	50	22.54 %	77.46 %	83.64 %	77.21 %	80.36 %
		30	23.18 %	76.82 %	82.80 %	76.58 %	79.63 %
		20	26.10 %	73.90 %	85.50 %	73.38 %	79.21 %
		10	28.98 %	71.02 %	85.86 %	70.31 %	77.69 %
AdaBoost	Fisher score	50	31.06 %	68.94 %	71.04 %	68.83 %	69.93 %
		30	93.38 %	6.62 %	100.00 %	1.92 %	13.86 %
		20	93.38 %	6.62 %	100.00 %	1.92 %	13.86 %
		10	93.38 %	6.62 %	100.00 %	1.92 %	13.86 %
	RELIEF	50	18.60 %	81.40 %	71.89 %	81.88 %	76.72 %
		30	31.01 %	68.99 %	58.92 %	69.49 %	63.99 %
		20	31.60 %	68.40 %	60.27 %	68.81 %	64.40 %
		10	33.43 %	66.57 %	55.56 %	67.12 %	61.07 %
	CBF	8	29.04 %	70.96 %	66.50 %	71.19 %	68.80 %
	Information gain	50	14.40 %	85.60 %	82.66 %	85.75 %	84.19 %
		30	14.47 %	85.53 %	82.32 %	85.69 %	83.99 %
		20	15.12 %	84.88 %	81.14 %	85.07 %	83.09 %
		10	15.30 %	84.70 %	81.82 %	84.84 %	83.32 %

Table 9 Cross-validation results for Not-1A - continued

CM	FS	# of Features	Error rate	Accuracy	Sensitivity	Specificity	G-metric
GentleBoost	Fisher score	50	27.39 %	72.61 %	73.06 %	72.59 %	72.83 %
		30	92.49 %	7.51 %	99.83 %	2.87 %	16.92 %
		20	92.39 %	7.61 %	99.83 %	2.98 %	17.24 %
		10	92.60 %	7.40 %	100.00 %	2.75 %	16.58 %
	RELIEF	50	18.02 %	81.98 %	74.41 %	82.36 %	78.28 %
		30	30.59 %	69.41 %	59.43 %	69.92 %	64.46 %
		20	31.05 %	68.95 %	58.92 %	69.45 %	63.97 %
		10	32.52 %	67.48 %	55.05 %	68.11 %	61.23 %
	CBF	8	23.12 %	76.88 %	61.45 %	77.66 %	69.08 %
	Information gain	50	14.46 %	85.54 %	84.01 %	85.61 %	84.81 %
		30	14.49 %	85.51 %	83.67 %	85.61 %	84.63 %
		20	14.66 %	85.34 %	83.00 %	85.45 %	84.22 %
		10	14.87 %	85.13 %	81.82 %	85.30 %	83.54 %
Random forest	Fisher score	50	31.77 %	68.23 %	73.91 %	67.94 %	70.86 %
		30	92.06 %	7.94 %	99.83 %	3.33 %	18.22 %
		20	92.34 %	7.66 %	99.83 %	3.03 %	17.39 %
		10	92.55 %	7.45 %	100.00 %	2.80 %	16.74 %
	RELIEF	50	20.13 %	79.87 %	74.75 %	80.13 %	77.39 %
		30	31.92 %	68.08 %	60.61 %	68.45 %	64.41 %
		20	33.84 %	66.16 %	59.93 %	66.47 %	63.12 %
		10	35.45 %	64.55 %	51.18 %	65.22 %	57.77 %
	CBF	8	28.02 %	71.98 %	62.46 %	72.46 %	67.27 %
	Information gain	50	14.57 %	85.43 %	84.01 %	85.50 %	84.75 %
		30	14.62 %	85.38 %	84.01 %	85.45 %	84.73 %
		20	14.45 %	85.55 %	83.84 %	85.63 %	84.73 %
		10	14.89 %	85.11 %	82.15 %	85.26 %	83.69 %
CART	Fisher score	50	40.11 %	59.89 %	62.29 %	59.77 %	61.02 %
		30	92.53 %	7.47 %	99.83 %	2.83 %	16.80 %
		20	92.87 %	7.13 %	99.83 %	2.47 %	15.71 %
		10	92.60 %	7.40 %	100.00 %	2.75 %	16.58 %
	RELIEF	50	25.66 %	74.34 %	63.47 %	74.88 %	68.94 %
		30	35.50 %	64.50 %	55.05 %	64.97 %	59.81 %
		20	36.82 %	63.18 %	57.91 %	63.44 %	60.61 %
		10	34.89 %	65.11 %	52.36 %	65.75 %	58.67 %
	CBF	8	31.88 %	68.12 %	56.90 %	68.68 %	62.51 %
	Information gain	50	18.85 %	81.15 %	72.56 %	81.58 %	76.94 %
		30	18.74 %	81.26 %	75.08 %	81.57 %	78.26 %
		20	19.25 %	80.75 %	72.22 %	81.18 %	76.57 %
		10	19.21 %	80.79 %	74.07 %	81.13 %	77.52 %

data sets. We build a model with training data and tune the control parameters by intermediate evaluation with cross-validation, and the final model evaluation is done with test data set. The cross-validation helps us not only tune parameters but also prevent overfitting.

Tuning objective is maximizing G-metric except for RF-RUS (RF with Random Under Sampling).

Table 10 Performance: test data set with information gain for 1A

CM	# of Features	Error rate	Accuracy	Sensitivity	Specificity	G-metric
LR[u]	50	22.35 %	77.65 %	28.57 %	84.56 %	49.15 %
	30	22.94 %	77.06 %	33.33 %	83.22 %	52.67 %
	10	22.94 %	77.06 %	28.57 %	83.89 %	48.96 %
LR[w]	50	32.94 %	67.06 %	61.90 %	67.79 %	64.78 %
	30	40.00 %	60.00 %	61.90 %	59.73 %	60.81 %
	10	34.71 %	65.29 %	71.43 %	64.43 %	67.84 %
SVM	50	40.59 %	59.41 %	80.95 %	56.38 %	67.56 %
	30	45.29 %	54.71 %	76.19 %	51.68 %	62.75 %
	10	47.06 %	52.94 %	76.19 %	49.66 %	61.51 %
AdaBoost	50	42.35 %	57.65 %	95.24 %	52.35 %	70.61 %
	30	34.71 %	65.29 %	85.71 %	62.42 %	73.14 %
	10	38.82 %	61.18 %	76.19 %	59.06 %	67.08 %
GentleBoost	50	38.24 %	61.76 %	80.95 %	59.06 %	69.15 %
	30	36.47 %	63.53 %	76.19 %	61.74 %	68.59 %
	10	37.65 %	62.35 %	61.90 %	62.42 %	62.16 %
Random forest	50	37.06 %	62.94 %	80.95 %	60.40 %	69.93 %
	30	35.29 %	64.71 %	85.71 %	61.74 %	72.75 %
	10	37.06 %	62.94 %	61.90 %	63.09 %	62.49 %
CART	50	47.06 %	52.94 %	66.67 %	51.01 %	58.31 %
	30	47.06 %	52.94 %	66.67 %	51.01 %	58.31 %
	10	43.53 %	56.47 %	61.90 %	55.70 %	58.72 %
RF-RUS	50	21.18 %	78.82 %	47.62 %	83.22 %	62.95 %

In the cross-validation and test result tables,

- CM: Classification Method
- FS: Feature Selection Method
- # of Features: the number of features we selected from the top of the list
- Error rate: percent of observations that were incorrectly classified
- Accuracy: percente of observations that were correctly classified
- Sensitivity: Accuracy only within positive class
- Specificity: Accuracy only within negative class
- G-metric: the square root of the product of sensitivity and specificity

Because CBF determines the number of features by itself, it had only 12 and 8 features in the lists for 1A and Not-1A, respectively. In all other cases, we obtained a rank ordered list of all features. Information Gain resulted in the best G-metric for all methods in cross-validation results. Therefore, we used the feature set provided by the Information Gain method to finalize parameters of each classifier.

Table 11 Performance: test data set with information gain for Not-1A

CM	# of features	Error rate	Accuracy	Sensitivity	Specificity	G-metric
LR[u]	50	9.06 %	90.94 %	17.63 %	95.14 %	40.95 %
	30	7.68 %	92.32 %	19.28 %	96.49 %	43.13 %
	10	8.94 %	91.06 %	15.82 %	95.36 %	38.84 %
LR[w]	50	30.78 %	69.22 %	70.68 %	69.14 %	69.90 %
	30	31.41 %	68.59 %	71.17 %	68.44 %	69.79 %
	10	32.27 %	67.73 %	70.02 %	67.60 %	68.80 %
SVM	50	20.65 %	79.35 %	83.20 %	79.19 %	81.17 %
	30	20.57 %	79.43 %	83.53 %	79.25 %	81.36 %
	10	24.27 %	75.73 %	87.81 %	75.08 %	81.20 %
AdaBoost	50	14.53 %	85.47 %	86.49 %	85.41 %	85.95 %
	30	14.32 %	85.68 %	86.00 %	85.67 %	85.83 %
	10	15.23 %	84.77 %	86.00 %	84.70 %	85.34 %
GentleBoost	50	14.49 %	85.51 %	85.50 %	85.51 %	85.50 %
	30	14.92 %	85.08 %	85.83 %	85.03 %	85.43 %
	10	14.50 %	85.50 %	86.16 %	85.46 %	85.81 %
Random Forest	50	14.66 %	85.34 %	86.82 %	85.25 %	86.03 %
	30	15.07 %	84.93 %	85.67 %	84.88 %	85.27 %
	10	14.89 %	85.11 %	85.67 %	85.08 %	85.37 %
CART	50	19.62 %	80.38 %	76.77 %	80.59 %	78.66 %
	30	19.72 %	80.28 %	76.77 %	80.48 %	78.60 %
	10	19.37 %	80.63 %	74.79 %	80.96 %	77.82 %
RF-RUS	10	6.98 %	93.02 %	47.45 %	95.63 %	67.36 %

Table 12 Comparisons of Sample Path SMSE

	LRN*	MELD	PTR #	Status**
LR[u]	0.821	100.1	11032.3	0.207
LR[w]	0.698	102.5	11052.9	0.217
SVM	0.640	90.1	11053.3	0.235
GentleBoost	0.643	86.8	11051.0	0.246
AdaBoost	0.668	106.1	11051.6	0.298
Random Forest	0.661	114.1	11051.0	0.217
CART	0.673	91.4	11051.1	0.234
RF-RUS	0.600	91.7	10674.7	0.205

*LRN: Local, Regional or National

**Status: 1A, 1B candidate or MELD category candidate

Table 13 Top 50 features for 1A candidates by information gain

Rank	Features	Description
1	'ptr_sequence_num'	PTR Sequence Number (a ranking number for the PTR)
2	'ptr_waiting_tm_cat'	Waiting Time Category - waiting time points awarded
3	'can_acpt_abo_incomp_N'	Accept an incompatible blood type?
4	'travel_time'	Travel time
5	'age_ratio'	Donor/Candidate age ratio
6	'wgt_ratio'	Donor/Candidate weight ratio
7	'ABO_match'	Donor/Candidate Blood type Match or not
8	'don_wgt_kg'	Donor/s Weight in kilograms
9	'don_expand_don_ki_0'	Meet kidney expanded donor criteria for cadaveric
10	'ptr_tot_score'	Total score for the candidate on the match
11	'don_expand_don_flg_optn_0'	Does donor meet criteria to be an Expanded Donor? (Y)
12	'can_min_wgt'	Minimum acceptable Donor Weight
13	'don_hist_hyperten_1'	History of Hypertension 1: NO
14	'can_motor_develop_4'	Motor Development (Ped Only) 4: No Motor delay/impairment
15	'don_li_biopsy_N'	Liver Biopsy
16	'diff_MELD'	Difference btw Lab and match MELDS
17	'lab_MELD'	Lab MELD
18	'canhx_wgt_kg_m'	Candidate/s Weight in kilograms : missing
19	'canhx_albumin'	Albumin (used for MELD)
20	'canhx_hgt_cm_m'	Candidate/s Height (stored in cm) : missing
21	'don_li_biopsy_macro_fat_m'	% Macro vesicular fat: Missing
22	'don_abo_O'	Donor/s Blood Type O
23	'can_work_income_N'	Working for income//If No, Not Working Due To: Missing
24	'can_work_no_stat_996'	Working for income. If No, Not Working Due To- Hospitalized
25	'can_acpt_a2_don_A'	Accept A2 donor?
26	'can_abo_O'	Patient/s Blood Type O
27	'don_prerecov_steroids_Y'	Pre-Recov Meds given Donor: Steroids
28	'can_malig_ty_hepcarcinoma_0'	Previous Malignancy - Hepatocellular Carcinoma
29	'can_cognitive_develop_4'	Cognitive Development (Ped Only) 4: No Cognitive delay/impairment
30	'don_anti_hcv_P'	Anti-HCV P: positive
31	'don_hist_cancer_2'	History of Cancer 2: SKIN - SQUAMOUS, BASAL CELL
32	'don_hcv_stat_1'	HCV Antibody Status 1: Positive
33	'can_work_no_stat_2'	Working for income. Not Working Due To- Demands of Treatment
34	'can_acpt_a2_don_N'	Accept A2 donor?
35	'don_hbv_surf_antibody_N'	HBsAb (Hepatitis B Surface Antibody) N: Negative
36	'don_hist_hyperten_2'	History of Hypertension 2: YES, 0-5 YEARS
37	'can_acpt_hcv_pos_N'	Accept an HCV Antibody Positive Donor?
38	'don_hist_diab_1'	History of Diabetes 1: NO
39	'canhx_growth_fail_0'	Patient is experiencing growth failure
40	'don_ebna_P'	EBNA (Epstein-Barr nuclear antigen) P: Positive
41	'can_init_srtr_lab_meld_ty_M'	First SRTR MELD/PELD type given : M
42	'can_cognitive_develop_1'	Cognitive Development (Ped Only) 1: Definite Cognitive delay/impairment
43	'don_death_circum_6'	Cirumstances of Death 6: DEATH FROM NATURAL CAUSES
44	'can_dgn_4215'	Primary Diagnosis 4215: LI:ALCOHOLIC CIRRHOSIS
45	'can_dgn_4216'	Primary Diagnosis 4216: LI:ALCOHOLIC CIRRHOSIS WITH HEPATITIS C
46	'can_diab_ty_998'	Diabetes 998: Diabetes Status Unknown
47	'don_tx_ctr_ty_TX1'	Transplant Center Type: TX1
48	'can_race_128'	Patient/s race 128: Native Hawaiian or Other Pacific Islander
49	'can_dgn_4404'	Primary Diagnosis 4404: LI:PLM: HEPATOBLASTOMA (HBL)
50	'can_malig_ty_2'	Previous Malignancy Type(s) 2: Skin Non-Melanoma

Table 14 Correlation Coefficients among top 10 features for 1A Candidates by Information Gain

Features	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
F1	1	0.713	−0.076	0.206	−0.024	−0.036	0.011	−0.040	−0.115	−0.0953
F2	0.713	1	−0.136	0.350	−0.019	−0.021	−0.074	0.039	−0.106	−0.1595
F3	−0.076	−0.136	1	−0.051	−0.085	−0.093	0.143	−0.012	0.039	0.3296
F4	0.206	0.350	−0.051	1	0.191	0.114	−0.021	−0.414	0.141	−0.0944
F5	−0.024	−0.019	−0.085	0.191	1	0.698	−0.045	−0.018	0.029	−0.1326
F6	−0.036	−0.021	−0.093	0.114	0.698	1	−0.001	0.127	0.073	−0.0710
F7	0.011	−0.074	0.143	−0.021	−0.045	−0.001	1	−0.047	0.019	0.6145
F8	−0.040	0.039	−0.012	−0.414	−0.018	0.127	−0.047	1	−0.168	−0.0150
F9	−0.115	−0.106	0.039	0.141	0.029	0.073	0.019	−0.168	1	−0.0033
F10	−0.095	−0.160	0.330	−0.094	−0.133	−0.071	0.615	−0.015	−0.003	1

F1 = 'ptr_sequence_num'
F2 = 'ptr_waiting_tm_cat'
F3 = 'can_acpt_abo_incomp_N'
F4 = 'travel_time'
F5 = 'age_ratio'
F6 = 'wgt_ratio'
F7 = 'ABO_match'
F8 = 'don_wgt_kg'
F9 = 'don_expand_don_ki_0'
F10 = 'ptr_tot_score'

Table 15 Top 50 features for Not-1A candidates by information gain

Rank	Features	Description
1	'wgt_ratio'	Donor/CandidateWeight Ratio
2	'hgt_ratio'	Donor/Candidate Height Ratio
3	'ptr_stat_dt'	Urgency Status Date - relative date/time
4	'ptr_waiting_tm_cat'	Waiting Time Category - waiting time points awarded
5	'ptr_stat_cd'	Match MELD
6	'ptr_tot_score'	Total score for the candidate on the match
7	'age_ratio'	Donor/Candidate Age ratio
8	'ptr_sequence_num'	PTR Sequence Number (a ranking number for the PTR)
9	'canhx_wgt_kg'	Candidate/s Weight in kilograms
10	'can_min_wgt'	Minimum acceptable Donor Weight
11	'diff_MELD'	Difference btw Lab and match MELDS
12	'can_age_at_listing'	Calculated Candidate Age in Months at Listing
13	'canhx_hgt_cm'	Candidate/s Height (stored in cm)
14	'can_bmi'	Patient/s Bady mass index
15	'median wait'	Transplant center's median waiting time
16	'notx200-100'	Difference in No. Tx btw 200 and 100 miles radius
17	'notx300-200'	Difference in No. Tx btw 300 and 200 miles radius
18	'can_init_srtr_lab_meld_ty_M'	First SRTR MELD/PELD type given
19	'can_init_optn_lab_meld_ty_M'	First OPTN MELD/PELD type given
20	'canhx_inr'	International Normalized Ratio(used for MELD)
21	'can_activate_dt'	Activation Date - date/time waiting time clock started
22	'can_max_wgt'	Maximum acceptable Donor Weight
23	'canhx_serum_creat'	Serum creatinine (used for MELD)
24	'can_init_act_stat_dt'	Date of First Active Status
25	'can_listing_dt'	Listing Date - date/time candidate was added to the waiting list
26	'ptr_activate_dt'	PTR Waiting Time Date - relative date/time
27	'can_motor_develop_4'	Motor Development (Ped Only) 4: No Motor delay/impairment
28	'canhx_bili'	Bilirubin (used for MELD)

Table 15 (continued)

Rank	Features	Description
29	'can_init_optn_lab_meld'	First OPTN MELD/PELD given
30	'can_init_srtr_lab_meld'	First SRTR MELD/PELD given
31	'can_work_income_N'	Working for income:
32	'can_max_age'	Maximum acceptable Donor age
33	'canhx_serum_sodium'	Serum sodium (used for MELD)
34	'canhx_ascites_1'	Ascites (used for MELD)
35	'can_cognitive_develop_4'	Cognitive Development (Ped Only) 4: No Cognitive delay/impairment
36	'can_education_996'	Patient/s Educational Status 996: N/A (< 5 YRS OLD)
37	'lab_MELD'	Lab MELD
38	'can_dgn_4315'	Primary Diagnosis 4315: LI:METDIS: OTHER SPECIFY
39	'can_functn_stat_4080'	Patient/s Functional Status 4080: 80 % - Active
40	'can_dgn_4500'	Primary Diagnosis 4500: LI:TPN/HYPERALIMENTATION IND LIVER DISEASE
41	'canhx_growth_fail_0'	Patient is experiencing growth failure
42	'canhx_enceph_2'	Encephalopathy (used for MELD)
43	'can_prev_tx_0'	Previous Transplants
44	'can_functn_stat_996'	Patient/s Educational Status 996: N/A (< 5 YRS OLD)
45	'canhx_ascites_3'	Ascites (used for MELD)
46	'can_max_mile'	Maximum miles the implant team will travel
47	'travel_time'	Travel Time from donor hospital to candidate's transplant center
48	'status1b_6012'	Is candidate's status 1B?
49	'can_min_age'	Minimum acceptable Donor Age
50	'can_new_prev_pi_tx_N'	Previous Pancreas Islet Transplantation: No

Table 16 Correlation Coefficients among top 10 features for Not-1A Candidates by Information Gain (continued)

Features	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
F1	1	0.519	−0.035	−0.094	−0.190	0.157	0.708	−0.050	−0.519	−0.345
F2	0.519	1	−0.003	−0.058	−0.129	0.075	0.399	−0.029	−0.318	−0.223
F3	−0.035	−0.003	1	0.081	−0.113	−0.336	−0.059	0.159	0.052	−0.044
F4	−0.094	−0.058	0.081	1	0.010	−0.128	−0.040	0.554	0.109	0.000
F5	−0.190	−0.129	−0.113	0.010	1	0.504	−0.290	−0.092	0.133	0.122
F6	0.157	0.075	−0.336	−0.128	0.504	1	0.092	−0.263	−0.178	−0.070
F7	0.708	0.399	−0.059	−0.040	−0.290	0.092	1	−0.036	−0.341	−0.258
F8	−0.050	−0.029	0.159	0.554	−0.092	−0.263	−0.036	1	0.079	−0.020
F9	−0.519	−0.318	0.052	0.109	0.133	−0.178	−0.341	0.079	1	0.418
F10	−0.345	−0.223	−0.044	0.000	0.122	−0.070	−0.258	−0.020	0.418	1

F1 = 'wgt_ratio'	F6 = 'ptr_tot_score'
F2 = 'hgt_ratio'	F7 = 'age_ratio'
F3 = 'ptr_stat_dt'	F8 = 'ptr_sequence_num'
F4 = 'ptr_waiting_tm_cat'	F9 = 'canhx_wgt_kg'
F5 = 'ptr_stat_cd'	F10 = 'can_min_wgt'

References

1. Ahn JH, Hornberger JC (1996) Involving patients in the cadaveric kidney transplant allocation process: a decision-theoretic perspective. Manag Sci 42(5):629–641
2. Alagoz O, Maillart LM, Schaefer AJ, Roberts MS (2007) Determining the acceptance of cadaveric livers using an implicit model of the waiting list. Oper Res 55(1):24–36
3. Bertsimas D, Chang A (2012) ORC: ordered rules for classification a discrete optimization approach to associative classification. Research Working papers

4. Bishop CM (2006) Pattern recognition and machine learning. Springer
5. Breiman L (2001) Random forests. Mach Learn 45(1):5–32
6. Campbell C, Ying Y (2011) Learning with support vector machines. Morgan &; Claypool Publishers
7. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority over-sampling technique. J Artif Intell Res 16:321–357
8. Cover TM, Thomas JA (2012) Elements of information theory. Wiley-Interscience
9. David I, Yechiali U (1985) A time-dependent stopping problem with application to live organ transplants. Oper Res 33(3):491–504
10. Duda RO, Hart PE, Stork DG (2012) Pattern classification. Wiley-Interscience
11. Gentry S, Chow E, Wickliffe C, Massie A, Leighton T, Snyder JJ, Israni AK, Kasiske BL, Segev D (2013) Relationship of cold ischemia time to estimated transportation time for liver allografts. Research Working papers
12. Harper AM, Taranto SE, Edwards EB, Daily OP (2000) An update on a successful simulation project: the Unos Liver Allocation Model. In: Proceeding of Winter Simulation Conference, Winter Simulation Conference, pp 1955–1962
13. Howard DH (2002) Why do transplant surgeons turn down organs? A model of the accept/reject decision. J Heart Fail 21(6):957–969
14. Jolliffe I (1982) A note on the use of principal components in regression. J R Stat Soc Ser C (Appl Stat) 31(3):300–303
15. Kira K, Rendell LA (1992) A practical approach to feature selection. In: Proceedings of the ninth international workshop on machine learning. Morgan Kaufmann Publishers Inc., pp 249–256
16. Kreke J, Schaefer AJ, Angus DC, Bryce CL, Roberts MS (2002) Incorporating biology into discrete event simulation models of organ allocation. In: Proceeding of winter simulation conference, winter simulation conference
17. Kubat M, Matwin S (1997) Addressing the curse of imbalanced training sets: one-sided selection. In: Proceedings of the fourteenth international conference on machine learning
18. Leppke S, Leighton T, Zaun D, Chen SC, Skeans M, Israni AK, Snyder JJ, Kasiske BL (2013) Scientific registry of transplant recipients: Collecting, analyzing, and reporting data on transplantation in the United States. Transplant Rev 27:50–56
19. Neter J, Kutner MH, Nachtsheim CJ, Wasserman W (1996) Applied linear statistical models. McGraw-Hill/Irwin
20. Pritsker AAB, Daily OP, Pritsker KD (1996) Using simulation to craft a national organ transplantation policy. In: Proceeding of winter simulation conference. ACM Press, New York, pp 1163–1169
21. Ratcliffe J, Young T, Buxton M, Eldabi T, Paul R, Burroughs A, Papatheodoridis G, Rolles K (2001) A simulation modelling approach to evaluating alternative policies for the management of the waiting list for liver transplantation. Health Care Manag Sci 4(2):117–124
22. Sandikci B, Maillart LM, Schaefer AJ, Alagoz O, Roberts MS (2008) Estimating the patient's price of privacy in liver transplantation. Oper Res 56(6):1393–1410
23. Sandikci B, Maillart LM, Schaefer AJ, Roberts MS (2011) Alleviating the Patient's price of privacy through a partially observable waiting list. SSRN electronic journal
24. Schapire RE, Freund Y (2012) Boosting.Foundations and Algorithms, MIT Press
25. Shechter SM (2005) A clinically based discrete-event simulation of end-stage liver disease and the organ allocation process. Med Dec Making 25(2):199–209
26. Su X, Zenios SA (2005) Patient choice in kidney allocation: a sequential stochastic assignment model. Oper Res 53(3):443–455
27. Su XM, Zenios SA, Chertow GM (2004) Incorporating recipient choice in kidney transplantation. J Am Soc Nephrol 15(6):1656–1663
28. Tang Y, Zhang YQ, Chawla NV, Krasser S (2009) SVMs modeling for highly imbalanced classification. IEEE transactions on systems, man, and cybernetics. Part B (Cybern) 39(1):281–288
29. Thompson D, Waisanen L, Wolfe R, Merion RM, McCullough K, Rodgers A (2004) Simulating the allocation of organs for transplantation. Health Care Manag Sci 7(4):331–338
30. TOMEK I (1976) Two modifications of CNN. IEEE transactions on systems. Man Cybern 6:769–772
31. Veropoulos K, Campbell C, Cristianini N (1999) Controlling the sensitivity of support vector machines. In: Proceedings of the international joint conference on AI, pp 55–60
32. Wilson DR, Martinez TR (1997) Improved heterogeneous distance functions. J Artif Intell Res 6:1–34
33. Wold S, Sjöström M, Eriksson L (2001) PLS-regression: a basic tool of chemometrics. Chemometr Intell Lab Syst 58(2):109–130
34. Yu L, Liu H (2003) Feature selection for high-dimensional data: a fast correlation-based filter solution. Proc Twentieth Int Conf Mach Learn 20(2):856
35. Zhao Z, Morstatter F, Sharma S, Alelyani S, Anand A, Liu H (2010). Advancing feature selection research. ASU feature selection repository