Optimized Scoring Systems: Towards Trust in Machine Learning for Healthcare and Criminal Justice

Cynthia Rudin and Berk Ustun
Duke University and MIT

Questions of trust in machine learning models are becoming increasingly important, as these models are starting to be used widely for high-stakes decisions in medicine and criminal justice. Transparency of models is a key issue affecting trust. The topic of transparency in modeling is being heavily debated in the media, and there are conflicting legal trends on the use of black box models between the European Union and the United States. This paper reveals that: (1) There is new technology to build transparent machine learning models that are often as accurate as black box machine learning models. (2) These methods have had impact already in medicine and criminal justice. This work calls into question the overall need for black box models in these applications.

There has been an increasing trend in healthcare and criminal justice to leverage machine learning in high-stakes prediction problems such as detecting heart attacks (58), diagnosing Alzheimer's disease (40), and assessing recidivism risk (47). In many of these applications, practitioners are deploying black box machine learning models that do not explain their predictions in a way humans can understand. In other applications, model development is outsourced to private companies, who build and sell proprietary predictive models using confidential datasets and without regulatory oversight.

The lack of transparency and accountability of predictive models has severe consequences in domains where data-driven predictions can significantly affect human lives. In criminal justice, proprietary predictive models can lead to decisions that may violate due process or that may discriminate based on race or gender (59). In 2015, for instance, Billy Ray

Johnson was imprisoned based on evidence from software developed by a private company, TrueAllele, which refused to reveal how the software worked. This led to a landmark case (People v. Chubbs) where the California Appeals Court ruled that such companies were not required to reveal how their software worked. As a different example, consider the controversy surrounding the COMPAS recidivism prediction model (38), which is used for several applications in the U.S. criminal justice system, but does not provide clear reasons for its predictions, and may discriminate on the basis of race (2, 14). There have been cases such as that of Glen Rodriguez, a prisoner with a nearly perfect record, who was denied parole as a result of an incorrectly calculated COMPAS score (59), with little recourse to argue, or even to determine how his score was computed. Similar issues have led to regulations such as the European Union's "right to explanation" (20). This new law will allow individuals to receive explanations for decisions made about them by algorithms.

Because mistakes in healthcare and criminal justice can be serious, or even deadly, it can be beneficial for companies to keep their models hidden. If the model is allowed to be hidden, the company never needs to justify why any particular prediction was made, nor does it need to take responsibility for mistakes made by the model. This leads to misaligned incentives, where the users of the tools would strongly benefit from transparent predictive models, but this would equally undermine profits for selling predictive models. Since these industries have a strong disincentive from building transparent models, there has been little work done on determining the answers to the following questions:

1. *Are there interpretable predictive models that are as accurate as black box models?* When we trust companies to build black box models, we are implicitly assuming that these models are more accurate than transparent models. It is possible that for a given black box model, an alternative model exists that is just as accurate, but that is so simple

that it can fit on an index card. (see for instance the literature on the surprising performance of simple linear models 16, 25). A compelling argument of Breiman (8) called the *Rashomon effect* indicates that for many applications, there may exist a large number of models that predict almost equally well. Among this large class of models are those from the various black box machine learning methods (e.g., support vector machines, random forests, boosted decision trees, neural networks), and there is no inherent reason that this class would exclude interpretable models.

2. *What are the desired characteristics of an interpretable model, if one exists?* The answer to this question changes for each audience and application (33, 39, 18). We might desire accuracy in predictions, risks that are calibrated, and we might want the model to be calculated by a judge or a doctor without a calculator, which makes it easier to explain to a defendant or medical patient. We may want doctors to be able to easily memorize the model. A model with all of these characteristics may not exist for any given problem, but if it does, it would be easier to use than a black box.
3. *If interpretable models do exist, is it possible to find them?* Interpretability, transparency, usability, and other desirable characteristics in predictive models lead to computationally hard optimization problems, such as mixed-integer non-linear programs. It is much easier to find an accurate unintelligible model than an interpretable one.

The renaissance from proprietary predictive models back to interpretable predictive models can only be partially determined by regulations such as “right to explanation.” Instead, the restoration to interpretable models should fundamentally be driven by technology. It must be demonstrated that interpretable models can achieve performance comparable with black box models. That is what this work focuses on.

We will present two machine learning algorithms with medical and judicial applications. The machine learning algorithms, which we call Supersparse Linear Integer Models

(SLIM) and Risk-Calibrated SLIM (RiskSLIM), solve mixed-integer linear and nonlinear programs. Their sparse linear models are faithful to the century-old scoring-system model form, similar to the predictive models that humans have designed over the last century. RiskSLIM produces risk scores directly from data, and SLIM produces scoring systems optimized for particular true positive / false positive tradeoffs. These new methods leverage modern optimization tools and avoid well-known pitfalls of rounding methods. The models come with optimality guarantees, meaning that they allow one to test for the existence of interpretable models that are as accurate as black box models. RiskSLIM's models are risk-calibrated across the spectrum of true positives and false positives (or sensitivity and specificity), and both methods honor constraints imposed by the domain. Software for both methods is public, and could be used to challenge the use of black box models for high-stakes decisions.

SLIM and RiskSLIM are already challenging decision-making processes for applications in medicine and criminal justice. We will focus on three of them in this work. (i) *Sleep Apnea Screening*: In joint work with Massachusetts General Hospital (55), we determined that a scoring system built using a patient's medical history can be as accurate as one that relies on reported symptoms. This yields savings in the efficiency and effectiveness of medical care for sleep apnea patients. (ii) *ICU Seizure Prediction*: In joint work with Massachusetts General Hospital (45), we created the first scoring system that uses continuous EEG measurements to predict seizures, called 2HELPS2B. The model provides concise reasons why a patient may be at risk. (iii) *Recidivism Prediction*: The recent public debate regarding recidivism prediction, and whether COMPAS' proprietary predictions are racially biased (2) leads to the question of whether interpretable models exist for recidivism prediction. In our studies of recidivism (62, 52, 54), we used the largest publicly available dataset on recidivism, and showed that SLIM and RiskSLIM could produce small scoring systems that are as accurate as state-of-the-art machine learning models. This calls into question the necessity of tools like COMPAS, and the reasons for government expenditures for predictions from proprietary models.

Scoring Systems: Applications and Prior Art

The use of predictive models is not new to society, only the use of black box models is relatively new. Scoring systems, which are a widely used form of interpretable predictive model, have existed since at least work on parole violation by Burgess (10) in 1928. The CHADS$_2$ score, shown in Figure 1, predicts stroke in patients with atrial fibrillation, and is arguably the most widely used predictive model in medicine. Scoring systems are sparse linear models with small integer coefficients. The coefficients are the point scores: for CHADS$_2$, the coefficients are 1, 1, 1, 1 and 2. The vast majority of predictive models in the healthcare system and justice system are scoring systems. Other examples from healthcare include: SAPS I, II and III (35, 37); APACHE I, II and III to assess ICU mortality risk (32, 30, 31); TIMI to assess the risk of death and ischemic events (3), HEART (43) and EDACS (46) for cardiac events; PCL to screen for PTSD (57), and SIRS to detect system inflammatory response syndrome (7). Examples from criminal justice include the Ohio

1. *Congestive Heart Failure*	1 point		···
2. *Hypertension*	1 point	+	···
3. *Age ≥ 75*	1 point	+	···
4. *Diabetes Mellitus*	1 point	+	···
5. *Prior Stroke or Transient Ischemic Attack*	2 points	+	···
ADD POINTS FROM ROWS 1–5	**SCORE**	=	···

SCORE	0	1	2	3	4	5	6
STROKE RISK	1.9%	2.8%	4.0%	5.9%	8.5%	12.5%	18.2%

Figure 1 CHADS$_2$ score to assess stroke risk (19). For each patient, the score is computed as the sum of the patients' points. The score is translated into the 1-year stroke risk using the lower table.

Risk Assessment System (34), the Kentucky Pretrial Risk Assessment Instrument (4), the Salient Factor Score (24, 23), and the Criminal History Category (CHC) (49). These models exemplify why scoring systems are generally easy to use. One can add the points together without a calculator, and it is possible to memorize most of these scoring systems with little effort.

None of the scoring systems listed in the previous paragraph was optimized for predictive performance on data. Each scoring system was created using a different method. Some of them were built using domain expertise alone (no data), and some were created using rounding heuristics for logistic regression coefficients to obtain integer-valued point scores.

Serious problems with rounding heuristics are well documented. These are classic problems solved by integer programming, and the reason for existence of the field of discrete optimization. When we solve a relaxed problem and round values to integers afterward, we know that (unless the problem has specific properties) either the solutions become infeasible, or low quality. It is easy to find problems in discrete optimization textbooks where rounding leads to flawed solutions. In the case of linear regression models, regression coefficients that are small are all rounded to zero, and thus an important part of the signal can easily be lost. We should not be using rounding heuristics if we want a reliable high quality solution, despite the government's recommendation (21) to round logistic regression coefficients.

An additional set of challenges arises when models need to satisfy *operational constraints*, which are user-defined requirements for the model (e.g., false positive rate below 20%). It is extremely difficult to design rounding heuristics that produce accurate models that also obey operational constraints. Heuristics for model design lead to suboptimal models, which in turn could lead to poor decision-making for high-stakes applications.

Since its inception, the field of discrete optimization has been advancing, while all of the risk scores have been built without using discrete optimization technology. Let us describe the optimization problems that we actually desire to solve when building scoring systems.

Optimization Problems and Methods

We will discuss two kinds of scoring systems:

1. Decision rules, which are scoring systems for decision-making, produced by SLIM. Here, predictions are based on whether the total score exceeds a threshold value (i.e., predict "yes" if total score > 1). The choice of variables and points in the score function is optimized for accuracy at a specific decision point (a specific true positive rate or false positive rate). The desired choice of true positive rate (TPR) or false positive rate (FPR) depends on the application. For medical screening, one might desire a larger false positive rate so that the test is more likely to falsely identify someone as positive for a disease than to dismiss someone who has the disease by giving them a negative test result. The user could specify the maximum false positive rate they are willing to tolerate, and SLIM will optimize the true positive rate subject to that constraint.
2. Risk scores, which are scoring systems for risk assessment, produced by RiskSLIM. These models use the score to output a risk estimate. The choice of variables and points in the score function is optimized for risk calibration. A scoring system is risk calibrated when the predicted risk of the outcome (from the model) matches the risk of outcome in the data. These models do not optimize a specific TPR/FPR tradeoff, rather they aim to achieve the highest true positive rate for each false positive rate.

We illustrate the difference between these two types of scoring systems in Figure 2, where we show SLIM and RiskSLIM models for predicting whether a prisoner will be arrested within three years of being released from prison. Both models were built using the largest

SLIM scoring system

1.	Age at Release between 18 to 24	2 points	· · ·
2.	Prior Arrests ≥ 5	2 points	+ · · ·
3.	Prior Arrest for Misdemeanor	1 point	+ · · ·
4.	No Prior Arrests	-1 point	+ · · ·
5.	Age at Release ≥ 40	-1 point	+ · · ·
		SCORE	= · · ·

PREDICT ARREST FOR ANY OFFENSE IF SCORE > 1

RiskSLIM risk score

1.	Prior Arrests ≥ 2	1 point	· · ·
2.	Prior Arrests ≥ 5	1 point	+ · · ·
3.	Prior Arrests for Local Ordinance	1 point	+ · · ·
4.	Age at Release between 18 to 24	1 point	+ · · ·
5.	Age at Release ≥ 40	-1 points	+ · · ·
		SCORE	= · · ·

SCORE	-1	0	1	2	3	4
RISK	11.9%	26.9%	50.0%	73.1%	88.1%	95.3%

Figure 2 Optimized scoring systems for recidivism prediction built using SLIM (top) and RiskSLIM (bottom). The outcome variable for both models is whether a prisoner is arrested within 3 years of release from prison. The SLIM scoring system outputs a predicted outcome. It has a test TPR/FPR of 76.6%/44.5%, and a mean 5-fold cross validation TPR/FPR of 78.3%/46.5%. The RiskSLIM scoring system outputs a risk estimate. It has a 5-fold cross validation mean test CAL/AUC of 1.7%/0.697 and training CAL/AUC of 2.6%/0.701. We provide a definition of these performance metrics in the Evaluation section.

publicly available dataset on recidivism and perform in line with state-of-the-art machine learning models (as discussed in the applications section). The SLIM scoring system outputs a decision rule (predict "yes" if the total score exceed a threshold score), whereas the RiskSLIM scoring system outputs a table of risk estimates for each distinct score. In both cases, the choice of variables and the number of points are chosen to optimize the relevant performance metric by solving a discrete optimization problem.

SLIM solves one constrained optimization problem to produce decision rules, and RiskSLIM solves a different problem to produce risk scores. Solving these optimization problems directly is principled, obviates the need for rounding and other manipulation, and directly encodes what we desire in a scoring system. The optimization problems are described mathematically in the appendix. In particular:

- In both optimization problems (the decision rule optimization and risk score model optimization), hard constraints are used to force the coefficients to integer values.
- In both optimization problems, the objective we minimize includes a term that encourages the number of questions asked in the scoring system to be small (model sparsity).
- In the objective for SLIM, there is a term that encourages the point values to be small (e.g., it prefers value '1 point' rather than value '7 points'). This also encourages the point values to be co-prime, meaning they share no common prime factors. Thus, this formulation would never choose point scores '10, 10, 20, 10, 40', rather it would choose '1, 1, 2, 1, 4' to solve the same problem.
- In the formulation for RiskSLIM, the objective includes a term used in logistic regression (the *logistic loss*) that encourages the scores to be small and risk calibrated. As we define later, a model is risk calibrated when its predicted risks agree with risks calculated directly from the data.

Both optimization problems can accommodate constraints on the solution that are specific to the domain (operational constraints). Some types of constraints are in Table 1.

Both optimization problems are computationally hard, but theoretical results allow practical improvements in speed. As a result, both the decision rule optimization problem and the risk score optimization problem can be solved for reasonably large datasets in minutes.

The risk score problem is a mixed-integer non-linear program, because the logistic loss is nonlinear. However, since the logistic loss is convex, cutting planes would be a natural type of technique for this problem. Cutting plane techniques produce piecewise linear

Constraint Type	**Example**
Feature Selection	Choose up to 10 features
Group Sparsity	Include either *Male* or *Female*, not both
Optimal Thresholding	Use at most 3 thresholds for *Age*, e.g., (Age≤30, Age≤50, Age≤75).
Logical Structure	If *Male* is in model, then also include *Hypertension*
Probability	Predict $\Pr(y=+1\|\boldsymbol{x}) \geq 0.90$ when *Male* = TRUE
Fairness	Ensure that the predicted outcome $\hat{y}$ is $+1$ an equal number of times for *Male* and *Female*

Table 1 Examples of operational constraints that can be addressed. Both SLIM and RiskSLIM can handle constraints on model form. SLIM handles constraints related to error metrics (e.g., fairness). RiskSLIM handles constraints on risk estimates (e.g., probability constraints).

approximations to the objective (cuts), which produce a surrogate lower bound, as illustrated in Figure 3. However, traditional cutting plane methods fail badly for the risk score problem. Since the feasible region is the integer lattice, a traditional cutting plane method would need to solve a mixed-integer program (MIP) to optimality to develop each new cut. If this surrogate MIP is not solved to optimality, we have no way of knowing when we have reached the solution to the risk score problem. After several iterations, enough cuts would accumulate that the mixed integer program could not be solved to optimality in a reasonable amount of time, and the program would stall and fail to provide optimal scoring systems. This necessitates a new approach.

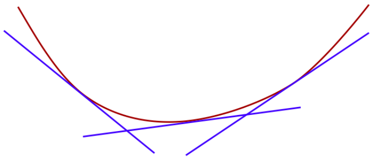


Figure 3 A convex loss function (smooth curve) and its surrogate lower bound (lines).

We developed a new branch-and-bound cutting plane method used in RiskSLIM for solving the risk score problem. This method does not stall, involves solving linear programs rather than mixed-integer programs, and can be implemented using standard callback functions in CPLEX (27). The method gracefully handles arbitrarily large datasets (even millions of observations), since computation scales linearly with the number of observations. The RiskSLIM model in Figure 2 was fit on a dataset with $N = 22,530$ observations in 20 minutes.

SLIM's decision rule problem (unlike the risk-score problem we just described for RiskSLIM) is a mixed-integer linear program. It can be solved with optimization software like CPLEX, but the solver is made more efficient with a specialized bound that we constructed, which reduces the amount of data we use without changing the solution to the optimization problem (discussed in 53).

In the appendix, we provide the formalism, algorithms, and implementation details for both SLIM and RiskSLIM.

SLIM and RiskSLIM have a strong theoretical basis in statistical learning theory, which is the main theoretical foundation underlying the discipline of machine learning. The appendix contains three *generalization bounds* for scoring systems, which are probabilistic guarantees on out-of-sample performance. Because the coefficients are co-prime, theoretical analyses of out-of-sample performance are able to leverage number theoretic concepts (Farey numbers). Statistical learning theory formalizes the principle of Occam's razor, which says that when there exist multiple competing theories that make exactly the same predictions, the simpler one is better.

Before we discuss applications, let us discuss means of evaluation.

Evaluation Methodology for Machine Learning Models

The fields of machine learning and data mining use rigorous empirical evaluation techniques. *Cross validation* is commonly used to provide a measure of uncertainty of prediction quality. To perform 5 fold cross-validation, the data are divided into five equal size folds. Four of the folds are used to train the algorithm, and predictions are made out-of-sample on the fifth "test" fold. The test fold rotates, and we report a mean and standard deviation (or range) across folds.

In this work, we are interested in the following evaluation measures for classification problems: The *true positive rate* (TPR) is the fraction of positive test observations predicted to be positive. *Sensitivity* is also the true positive rate. *Specificity* is the true negative rate, the fraction of negative test observations predicted to be negative. The *false positive rate* (FPR) is the fraction of negative test observations predicted to be positive, and FPR is equal to one minus the specificity. The *Receiver Operator Characteristic (ROC) curve* is a plot of true positive rate for each possible value of the false positive rate. The *area under the ROC curve* (AUC) is important, since if the true positive rate is high for each value of the false positive rate, the algorithm has a high AUC and is performing well. An AUC value of .5 would be obtained for random guessing, an AUC of 1 is perfect, and for most of the problems we consider here, an AUC value of .8 would be considered excellent. AUC is a useful evaluation measure particularly when the positive and negative classes are imbalanced, meaning that only a small fraction of the data are positive (or negative). For instance, for the seizure prediction problem we discuss below, only 13.5% of observations in the seizure prediction data correspond to true seizures, while the rest were non-seizures.

For risk score prediction, we are also interested in *calibration* (CAL), which is a measure of how closely the predicted positive rate from the model matches the empirical positive rate in the data. We will discuss CAL later.

In general we find that when the form or size of the model is not constrained, then for the majority of applications, AUC values for all machine learning algorithms tend to be similar. AUC's start to differ when operational constraints are imposed. We will see this in more depth for the sleep apnea and seizure examples below.

Applications and Insights

Both SLIM and RiskSLIM have had an impact on several applications in healthcare and criminal justice. In what follows, we discuss three applications, and insight gained by producing interpretable models.

Sleep Apnea Screening

Obstructive Sleep Apnea (OSA) is a serious medical condition that can lead to morbidity and mortality, and can severely affect quality of life. A major goal of every sleep clinic is to screen patients for this disease correctly. Testing for OSA is problematic. Preliminary screening is mainly based on patient-reported symptoms and scoring systems. However, surprisingly, patient-reported symptoms are not particularly reliably reported, nor are they useful for determining whether a patient has OSA. In particular, doctors often use the Epworth Sleepiness scale (28) or other scoring systems to screen for OSA, which are based on typical reported OSA symptoms like snoring, nocturnal gasping, witnessed apneas, sleepiness and other daytime complaints. Each of these predictive factors alone is weak; however, the comorbidities provided in medical records are much stronger. Hypertension, for instance, is a good predictor of OSA. Thus, it is reasonable that the staff of the Massachusetts General Hospital hypothesized that an accurate scoring system could be created that uses information from only routinely available medical records – without reported symptoms – that could be just as accurate as the widely used scoring systems.

The data provided for this study were records from all patients at the Massachusetts General Hospital Sleep Lab above 18 years old that underwent an definitive test for OSA called *polysomnography* (1,922 patients) between 2009 and 2013. Polysomnography is an expensive test for obstructive sleep apnea in which patients stay at the hospital overnight in order to collect information about brain activity, blood oxygen levels, heart rate, breathing patterns, eye movements and leg movements. Our goal was to predict OSA using only information that was available before the polysomnography. Such information included standard medical information (e.g. gender, age, BMI, past heart problems, hypertension, diabetes, smoking), as well as self-reported information on sleep patterns (e.g. caffeine consumption, insomnia, snoring, gasping, dry mouth in morning, leg jerks, falls back to sleep slowly). A full list of the features is provided in Table 1 of (55).

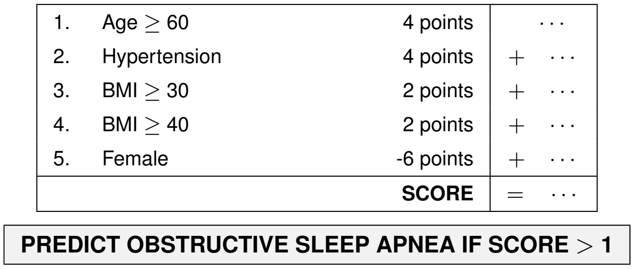
The domain experts also required several operational constraints on the form of the model, such as constraints on the size of the model, and the signs of the coefficients. The domain experts considered these constraints vital to their trust in the model.

If a scoring system could be developed that accurately screens patients for sleep apnea, using only the patient's medical records, without using the (misleading) patient-reported symptoms, it would create an actionable tool that could allow automatic (as opposed to manual) screening. This type of automated scoring would allow wise usage of limited resources available for direct patient encounters.

Results and Impact

Our domain experts (Brandon Westover and Matt Bianchi at Massachusetts General Hospital) had two important goals: (i) create an accurate transparent model for obstructive sleep apnea that obeyed operational constraints; (ii) determine the value of the patient-reported symptoms (e.g. gasping, insomnia, caffeine consumption) as compared with information that is already within the patient's medical record.

Prior to our work, the best previous scoring system for sleep apnea screening was arguably the STOP-BANG score (13). STOP-BANG relies on 8 features including self-reported snoring, tiredness, and breathing problems in addition to medical record information. Its sensitivity is 83.6% and specificity is 56.4%, which precludes it from being used as a screening tool. The specificity is the percentage of negatives identified correctly, meaning that the false positive rate is $100\% - 56.4\% = 43.6\%$, much higher than the goal on FPR that our domain experts were looking for, which was 20%. One of the models that our collaboration produced has sensitivity 61.4% and specificity 79.1%. The scoring system was produced by SLIM, and is in Figure 4.



1.	Age $\geq$ 60	4 points	$\cdots$
2.	Hypertension	4 points	$+ \cdots$
3.	BMI $\geq$ 30	2 points	$+ \cdots$
4.	BMI $\geq$ 40	2 points	$+ \cdots$
5.	Female	-6 points	$+ \cdots$
		SCORE	$= \cdots$

PREDICT OBSTRUCTIVE SLEEP APNEA IF SCORE $>$ 1

Figure 4 **SLIM scoring system for sleep apnea screening. This model achieves a 10-fold cross validation mean test TPR/FPR of 61.4/20.9%, and obeys all operational constraints. The model predicts OSA if the score exceeds 1. There are no common prime factors, since the threshold 1 is included in the set of factors; the coefficients are 1,4,4,2,2,-6, which are co-prime.**

In our experiments, we compared the performance of SLIM scoring systems to models from other machine learning algorithms, such as ℓ_1-penalized logistic regression (Lasso); ℓ_2-penalized logistic regression (Ridge); ℓ_1 and ℓ_2-penalized logistic regression (Elastic Net); C5.0 decision trees (C5.0T); C5.0 rule lists (C5.0R); support vector machines with a radial basis kernel (SVM RBF); and support vector machines with a linear kernel (SVM Linear).

Prediction Performance: There is no clear loss of prediction performance using SLIM than using the best of the black box machine learning models. In particular, the performance of most methods without operational constraints was very similar. In Figure 5, for example, we show the decision points of SLIM and SVM models across the full ROC curve using only features from medical records. The curves are almost identical, which means that SLIM and SVM produce predictions that attain the same levels of sensitivity across all levels of specificity.

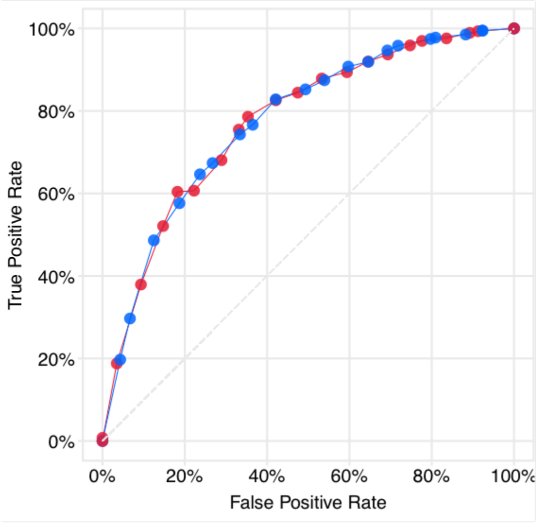


Figure 5 **Decision points for** SLIM **and SVM RBF across the full ROC curve using the set of features from medical records.** SLIM **(blue) has a 10-fold cross validation test AUC of 0.770. SVM RBF (red) has a 10-fold cross validation test AUC of 0.759.**

Patient-Reported Symptoms vs. Medical Record Information: Using any machine learning algorithm, it was easy to answer the second question of domain experts – that of measuring the importance of patient-reported symptoms. Patient-related symptoms are not nearly as important as medical history information. Across every machine learning method we tried, the models that used only patient-reported symptoms performed poorly, whereas models that used only medical record information performed almost as well (often as well) as the models that used both sets of information (see Table S2 in 55, for the AUC values of all machine learning methods). To illustrate this, Figure 6 shows the ROC curves for models built using all features (dashed curve), patient-reported symptoms only (lower solid curve), and features that were extracted from an electronic health record (gray curve).

This figure shows that performance does not degrade when omitting the patient-reported variables all together.

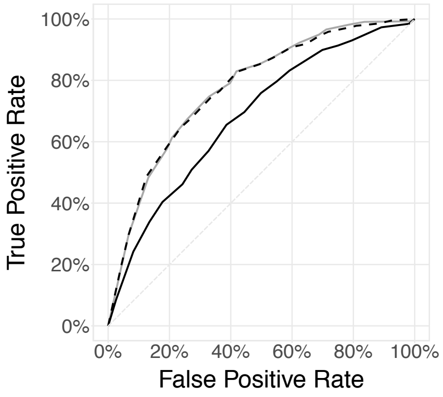


Figure 6 **Decision points of** SLIM **models over the full ROC curve for: (i) all features (gray, overlapping with dashed curve); (ii) features that can be extracted from an electronic health record (dashed); (iii) features related to patient-reported symptoms (black).**

Summary for Sleep Apnea: As we have seen:

- SLIM was able to find a model using medical-record information only – without the patient-reported symptoms – with prediction quality that is essentially identical to the models that use both types of information.
- If the operational constraints were not included, all machine learning methods had similar prediction performance.
- Shortly, we will discuss a third result: if the operational constraints were included, there were substantial differences between machine learning methods, and only our mathematical programming-based method was able to incorporate the constraints while maintaining accuracy.

Our work on this topic was published in the Journal of Clinical Sleep Medicine (55), which is the official journal of the American Academy of Sleep Medicine. More details can be found in SLIM's paper in the journal Machine Learning (53).

Insight from Sleep Apnea Application: Operational Constraints Are Challenging for Non-Mathematical-Programming-Based Machine Learning Algorithms

The experiment for the sleep apnea project revealed severe shortcomings for non-mathematical-programming-based machine learning methods, in that they are almost incapable of handling operational constraints. Our collaborators at Massachusetts General Hospital wanted a model fulfilling three simple operational constraints:

- *Max FPR*: Less than 20% false positive rate. Our goal was to correctly detect as many cases of OSA as possible, limiting the falsely detected cases to 20%.
- *Model Size*: Less than 5 terms in the model. Also small integer coefficients.
- *Sign Constraints*: Some point values needed to be constrained to be either positive or negative. For instance, it would not make sense to subtract points (predict lower risk of OSA) for patients that have hypertension, than for those who do not. This is because hypertension alone has significant risk for sleep apnea.

How would one obtain a model obeying these constraints with a standard machine learning algorithm that does not use mathematical programming? As it turns out, this is not trivial. For other methods, the only degrees of freedom given to the experimenter are parameters that govern the shape of the model. These parameters can be tuned until the constraints are obeyed, but this proved to be challenging in practice. In particular, our results showed that for the standard machine learning methods, even if we searched extensively through parameter values, we can rarely find feasible models (model that satisfy all constraints). Table 2 shows the number of parameter values we chose using a grid search, which is recorded in the “Total Instances Trained” column, and the parameter values we chose are in the “Free Parameters” column. For instance, we ran 975,000 instances of the standard machine learning algorithm called “elastic net.” Despite the large number of instances we trained, Table 2 indicates that the grid search rarely produced models that satisfied the constraints. The decision tree methods we tried (CART, C5.0 rules, C5.0 trees) had the worst problems: they were unable to produce any models with FPR<20% despite tuning. This can be seen in the column under "Percent of total instances satisfying" labeled "MaxFPR." SVM with either linear or RBF kernels were unable to produce models with simultaneously less than 5 terms and FPR<20%, while ridge regression had the same problem. The only algorithms that could be tuned to accommodate the constraints were elastic net, lasso, and SLIM. For SLIM, the constraints are directly incorporated into the solver, and every solution it produces is feasible.

Of the feasible models found from the standard machine learning methods, almost none are accurate predictive models. Figure 7 shows how elastic net, lasso and SLIM perform as we vary the model size. Here, both lasso and elastic net would need 8 variables to attain the accuracy of the 5-variable SLIM model.

What we have illustrated is a serious concern regarding the use of machine learning methods for practical problems: in almost all machine learning algorithms, user-defined constraints are not accommodated. Mathematical programming tools solve this issue.

Translating Better Predictions into Cost Benefit Analysis for Automated Sleep Apnea Screening

Machine learning predictions translate into a cost-benefit analysis for the application domain, but the cost and benefit can be difficult to quantify. The additional cost of the system comes from the increase in false positives and false negatives and the cost of implementing the system. The benefit comes from additional true positives and true negatives. To calculate cost and benefit, we are required to know how much each true positive, false positive, true negative and false negative is worth. These values are particularly difficult

Algorithm	Values for Free Parameters	Total Instances Trained	Percent of Total Instances Satisfying		
			Max FPR	Max FPR & Model Size	Max FPR, Model Size & Signs
CART	39 values of $w^+ \in \{0.025, 0.05, \ldots, 0.975\}$	39	0.0%	0.0%	0.0%
C5.0R	39 values of $w^+ \in \{0.025, 0.05, \ldots, 0.975\}$	39	0.0%	0.0%	0.0%
C5.0T	39 values of $w^+ \in \{0.025, 0.05, \ldots, 0.975\}$	39	0.0%	0.0%	0.0%
Lasso	39 values of $w^+ \in \{0.025, 0.05, \ldots, 0.975\}$ × 1000 values of λ chosen by **glmnet**	39000	19.6%	4.8%	4.8%
Ridge	39 values of $w^+ \in \{0.025, 0.05, \ldots, 0.975\}$ × 1000 values of α chosen by **glmnet**	39000	20.9%	0.0%	0.0%
Elastic Net	39 values of $w^+ \in \{0.025, 0.05, \ldots, 0.975\}$ × 1000 values of λ chosen by **glmnet** × 19 values of $\alpha \in \{0.05, 0.10, \ldots, 0.95\}$	975000	18.3%	1.0%	1.0%
SVM Linear	39 values of $w^+ \in \{0.025, 0.05, \ldots, 0.975\}$ × 25 values of $C \in \{10^{-3}, 10^{-2.75}, \ldots, 10^3\}$	975	18.7%	0.0%	0.0%
SVM RBF	39 values of $w^+ \in \{0.025, 0.05, \ldots, 0.975\}$ × 25 values of $C \in \{10^{-3}, 10^{-2.75}, \ldots, 10^3\}$	975	15.8%	0.0%	0.0%
SLIM	$w^+ = n^-/(1+n^-)$, $C_0 = 0.9w^-/nd$, $\lambda_0 \in \{-100, \ldots, 100\}$, $\lambda_j \in \{-10, \ldots, 10\}$	1	100.0%	100.0%	100.0%

Table 2 Classification methods used for sleep apnea screening. We show the parameter settings, total number of instances trained, and the percentage of instances that fulfilled various combinations of operational constraints. Each instance is a unique combination of free parameters for a given method. The w^+ parameter is a unit misclassification cost for positive points.

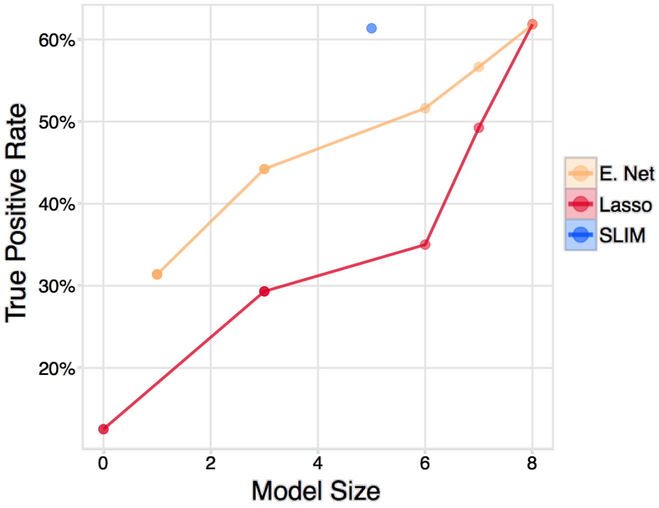


Figure 7 Sensitivity and model size of Lasso and Elastic Net models that satisfy the sign and FPR constraints. For each method, we plot the instance that attains the highest 10-fold cross validation mean test TPR at model sizes between 0 and 8. Lasso and Elastic Net need at least 8 coefficients to produce a model with the same sensitivity as SLIM.

to obtain for decision support systems that require a human in the loop (e.g., recidivism prediction) or systems for which there is not an established precedent for estimating early detection (seizure prediction in the ICU). If uncertainty in the monetary values is large (perhaps larger than the improvement in predictions due to the machine learning method), the cost-benefit analysis is not meaningful. This does not mean the predictive model is not useful, it simply means we cannot perform a meaningful cost-benefit analysis.

Sleep apnea is a serious medical condition that has many complications, including high blood pressure (which causes other medical conditions), type 2 diabetes, heart disease, liver problems, depression, and daytime fatigue leading to loss of productivity. The effect of early detection on these complications, and their downstream costs, is difficult to measure.

There is a well-studied, direct causal relationship we can attribute to sleep apnea, which is that sleep apnea causes car accidents. There have been at least 30 scientific papers studying the effects of obstructive sleep apnea on car accidents, many of which indicate a 2-5 times higher risk for apnea patients (17). Currently there is no automated scoring system for apnea in place within the U.S. since prior to our work, all scoring systems relied on patient-reported systems (e.g., snoring) that are not extractable from medical records. Thus, we can ask the question: *By implementing an automated scoring system for sleep apnea in the U.S., how much money would be saved through reduction in car accidents within one year?* Using a combination of well-studied estimates of the number and costs of car accidents in the U.S., along with assumptions that the at-risk population of Massachusetts General Hospital (MGH) apnea patients represents the U.S. at-risk population, we compute a rough back-of-the-envelope estimate of the cost and benefit for using our test on the scale of the U.S. population, over the period of one year. This calculation is in the appendix. According to our calculation, the cost (due to false positives and extra polysomnography tests) is approximately $1.6 billion and the benefit is $5.9 billion (in economic costs only) or $20.3 billion (including quality-of-life valuations) due to reduction in car accidents as a result of treatment for obstructive sleep apnea. This analysis indicates a savings of 4.3 billion in economic costs (conservatively estimated), or 18.7 billion including quality-of-life valuations. In each subsequent year, the savings would continue to be realized as the treated patients continue to have fewer car accidents, and there is no continued cost of polysomnography tests for those patients. Thus, the benefit we estimated for one year is similar to the estimated benefit for each subsequent year, and with no associated cost. This calculation is only for car accidents, and does not consider reduction in other medical conditions associated with sleep apnea.

Seizure Prediction in the ICU

Patients in the intensive care unit of a hospital who may be at risk for dangerous seizures are monitored using continuous electroencephalography cEEG, where electrodes monitor electrical signals in the brain. A clinician monitors the patient and identifies features in the cEEG signal that may be predictive of seizure. The clinician may determine that the patient requires an intervention to prevent seizures, which could be dangerous, or (expensive) continued monitoring. Rather than have clinicians estimate seizure risk manually from cEEG signals, Massachusetts General Hospital staff aimed to assist clinicians by estimating this risk in a transparent way. We worked with a dataset from the Critical Care EEG Monitoring Research Consortium, collected at several hospitals (Emory University Hospital, Brigham and Womens Hospital, and Yale University Hospital) over the course of 3 years. The database contains 5,427 cEEG recordings with 87 variables, and each patient had at least 6 hours of uninterrupted cEEG monitoring. The variables from cEEG included important pattern types: lateralized periodic discharges (LPD); lateralized rhythmic delta (LRDA); generalized periodic discharges (GPD); generalized rhythmic delta (GRDA); bilateral periodic discharges (BiPD). Additionally, we had medical history and secondary symptoms for each patient. The outcome we aimed to predict was whether the patient would have a seizure within 24 hours. A transparent automated tool to help with seizure risk prediction would be particularly helpful in preventing false negatives: situations where clinicians mistakenly label the patient as being not-at-risk.

Results and Impact for Seizure Prediction

1.	Any cEEG Pattern with Frequency **2 H**z	1 point	···
2.	**E**pileptiform Discharges	1 point	+ ···
3.	Patterns include [**L**PD, LRDA, BIPD]	1 point	+ ···
4.	**P**atterns Superimposed with Fast or Sharp Activity	1 point	+ ···
5.	Prior **S**eizure	1 point	+ ···
6.	**B**rief Rhythmic Discharges	**2** points	+ ···
		SCORE	= ···

SCORE	0	1	2	3	4	5	6+
RISK	<5%	11.9%	26.9%	50.0%	73.1%	88.1%	95.3%

Figure 8 **2HELPS2B scoring system, constructed by** RiskSLIM.

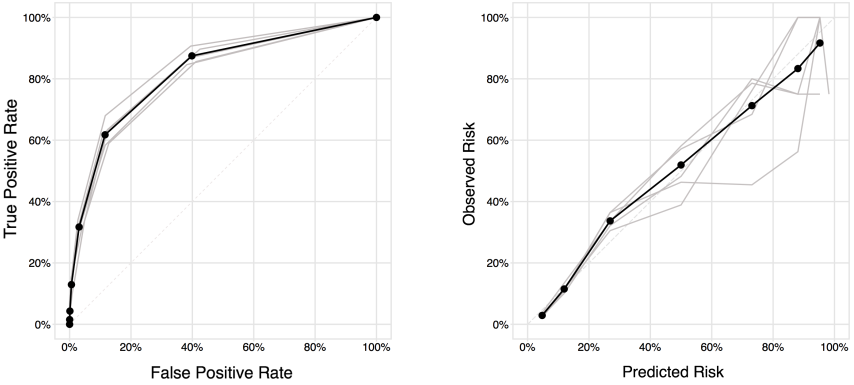


Figure 9 **ROC curves and calibration curves for the 2HELPS2B score produced by** RiskSLIM.

The model we created is called 2HELPS2B, which is shown in Figure 8. The "**2H**" stands for: "GRDAs, LRDAs, BiPDs, LPDs, or GPDs with a frequency > 2 **H**z" (1 point), "**E**" stands for **E**pileptiform discharges (1 point), "**L**" stands for **L**PD or LRDA or BiPD

(1 point), "**P**" stands for GRDAs, LRDAs, BiPDs, LPDs, or GPDs with plus features (superimposed rhythmic, fast, or sharp activity) (1 point); "**S**" is any history of **s**eizures (1 point), and "**2B**" is Brief Potentially Ictal Rhythmic Discharges (2 points). 2HELPS2B is similar to other typical medical scoring systems in that it is easy to memorize – the full model is contained in the acronym. The clinician need only remember the name of the score to recall the full model. Its mean AUC over 5 cross-validation folds is 0.819 (with a range of 0.776-0.849 over the 5 folds).

The 2HELP2B score has no predecessors; it is the first scoring system to be developed for cEEG monitoring for seizure prediction. It can be directly integrated into clinical workflow.

Calibration was an important concern for our collaborators – models were deemed unacceptable if they were poorly calibrated. While constructing the 2HELP2B score, it became apparent that the typical methods one might use to construct scoring systems had systematic problems with calibration. This is our second insight, which we now discuss.

Insight from Seizure Prediction Application: Risk Calibration Suffers when we use Rounding to Compute Risk Scores

Risk calibration (CAL) measures how closely the estimated risks from the model match risks in the data. Risk calibration is essential for practical use in risk-scoring applications.

Let us define CAL precisely. The estimated risks for each individual i are calculated using the scoring system (e.g., from 2HELPS2B), and the risk for patient i from the model is denoted by p_i. Separately, for each value of the score s, we estimate the probability of the outcome $y = 1$ given s, that is, $p(s) = P(y = 1|s)$. Then we compute the Euclidean distance between p_i and $p(s_i)$ across all patients i, and this is precisely CAL. A calibration plot is a plot of $p(s_i)$ vs p_i. If the plot is a diagonal line, the model is nicely calibrated.

RiskSLIM minimizes the logistic loss that is used for logistic regression. Logistic regression produces risk-calibrated models (12, 61) but when rounding or other post-processing steps are done to a logistic regression model, it can drastically alter calibration. As discussed earlier, rounding sends all small coefficients to zero (which eliminates part of the signal), and rounding coefficients upwards makes variables more important than they should be in a calibrated model. An extensive set of experiments in (54, 52) considered several types of rounding techniques. In particular, it considered naïve rounding (denoted Rd for *rounding*), which simply rounds coefficients to the nearest integer within the range $\{-5, -4, .., 0, ...4, 5\}$, and rescaled rounding (denoted RsRd for *rescaled rounding*), which scales all coefficients so that the largest one is ± 5, and then rounds to the nearest integer. Rescaled rounding tends to mitigate the problem of too many coefficients being rounding to zero.

Calibration curves should always go upwards: as the score increases, the risk should always increase. However, this does not hold for either Rd or RsRd. Our collaborators determined that this was problematic since it is unreasonable that (for instance) a patient with a score of 3 has a higher risk of seizure than a patient with a score of 4. Figure 10 shows results from a controlled cross-validation experiment, including ROC curves and calibration curves for RiskSLIM and also for the Rd and RsRd methods. The black curves in the figures are from a model computed across the 5 cross-validation folds, and models in gray are from each of the 5 folds. The problems with calibration are apparent: the curves simply do not always increase. Here, RiskSLIM's 5-fold mean CAL was 2.5% (the best is 0%), whereas Rd's was 3.7% and RsRd's was 11.5%. 2HELPS2B was determined separately from the controlled experiment, and its ROC and calibration curves are in Figure 9. It has mean CAL over the 5 folds of 2.7%.

These experiments with rounding are not surprising – when we move in an arbitrary direction in a high dimensional space, we know from integer programming textbooks (60) that there are problems with solution quality. Further, by using rounding, all guarantees of optimality are lost. This becomes problematic for applications like recidivism prediction, discussed shortly.

Recidivism Prediction

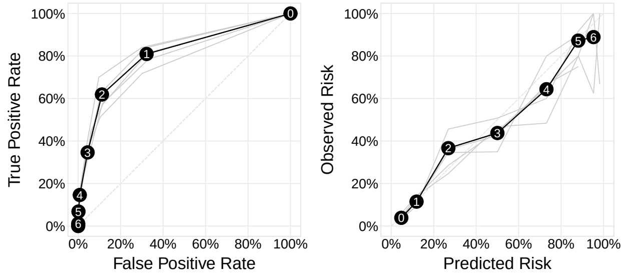
In the U.S., criminal sentencing is done according to a mandated federal guideline (e.g., the Criminal History Category, 50). One of the latest public guidelines for recidivism risk prediction in the U.S. is the Pennsylvania Commission on Sentencing (41), and other methods are used in Canada (22), the Netherlands (47), and the U.K. (26). There are a very large number of different risk scores for various applications, including sentencing, parole, and prison administration (see 62, for a longer list). These scores can be helpful: it is possible for a data-driven calculation to mitigate irregularities in decisions made by humans. No human can keep a database in their head and accurately calculate risks. In fact, the decision-making process of humans can have high variance and rely on arbitrary factors. For instance, there is evidence that judges are much less likely to make a favorable ruling just before a lunch break (29, 15). Worse than this, judges are not generally provided with feedback on the quality of their recidivism predictions, meaning they cannot learn from past mistakes.

Over the last few years, there has been an ongoing debate in the statistical community of criminologists. Some of them have claimed that traditional statistical methods are as accurate for predicting recidivism as modern machine learning tools, when the proper preprocessing has been done to create features (see e.g. 47, 5, 11). As we showed above, however, traditional statistical tools have serious flaws when paired with rounding methods, in terms of risk calibration and inability to incorporate operational constraints.

Optimized Risk Score (RiskSLIM)

1.	Brief Rhythmic Discharges	2 points	· · ·
2.	Patterns Include LPD	2 points	+ · · ·
3.	Prior Seizure	1 point	+ · · ·
4.	Epileptiform Discharge	1 point	+ · · ·
		SCORE	= · · ·

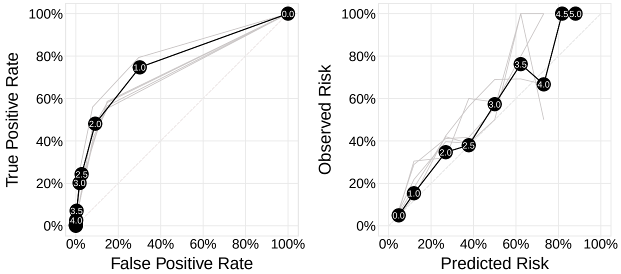
SCORE	0	1	2	3	4	5	6
RISK	4.7%	11.9%	26.9%	50.0%	73.1%	88.1%	95.3%



$\ell_1 + \ell_2$ Penalized Logistic Regression + Rounding

1.	Any Prior Seizure	1 point	· · ·
2.	Patterns Include BiPD, LRDA, LPD	1 point	+ · · ·
3.	MaxFrequency LPD	× 1 point per Hz	+ · · ·
		SCORE	= · · ·

SCORE	0.0	1.0	2.0	2.5	3.0	3.5	4.0	4.5	5.0
RISK	4.7%	11.9%	26.9%	37.8%	50.0%	62.2%	73.1%	81.8%	88.1%



$\ell_1 + \ell_2$ Penalized Logistic Regression + Scaling + Rounding

1.	AnyPriorSeizure	5 points	· · ·
2.	Patterns Include BiPD, LRDA, LPD	1 point	+ · · ·
3.	MaxFrequency LPD	× 5 points per Hz	+ · · ·
		SCORE	= · · ·

SCORE	0 to 10	12.5	15.0	20.0	20 to 25
RISK	< 5.0%	7.6%	50.0%	92.4%	> 95.0%

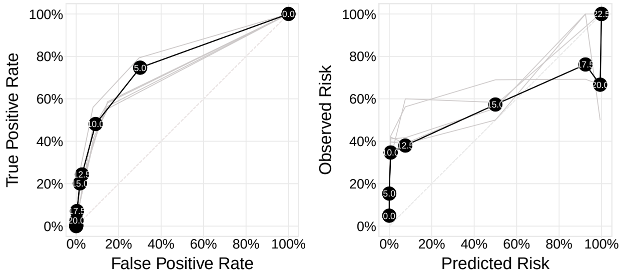


Figure 10 **Risk scores, ROC curves, and reliability diagrams for RiskSLIM and heuristic rounding techniques. We show the final model on training data in black, and fold-based models on test data in gray.**

At the same time as this debate is happening, companies like Northpointe (now called Equivant) are selling predictions to the government, which are used widely. These risk scores have the potential to be racially biased, as argued by ProPublica (2). In 2016, in the case State v. Loomis, the Wisconsin Supreme Court ruled that black box risk scores like Northpointe's COMPAS can be used by judges, but minimized the role that such scoring systems could play as evidence. An appeal was filed at the U.S. Supreme Court, who declined to hear the case in June 2017.

The goal of our project was to determine whether such black box scoring systems were needed at all for recidivism prediction. If we find a transparent model with the same accuracy as the best black box model, we no longer require the black box model.

We used the largest publicly available dataset on recidivism, which is the "Recidivism of Prisoners Released in 1994" dataset collected by the U.S. Department of Justice, Bureau of Justice Statistics (48). This dataset contains information that we used from 33,796 prisoners, including criminal history from record-of-arrest-and-prosecution (RAP) sheets, along with demographic factors such as gender and age. We omitted socioeconomic factors such as race for the main study, but conducted experiments using race afterwards (see 62). The outcomes we aimed to predict within three years of release were: (1) arrest for any crime, (2) arrest for drug-related crime, (3) arrest for violent crime (general_violence), (4) arrest for a domestic violence crime, (5) arrest for a sexual violence crime, and (6) arrest for a crime involving fatal violence.

The results were confirmed by Angelino et al. (1) using another publicly available dataset, namely the Propublica data, which was developed to study whether COMPAS is racially biased.

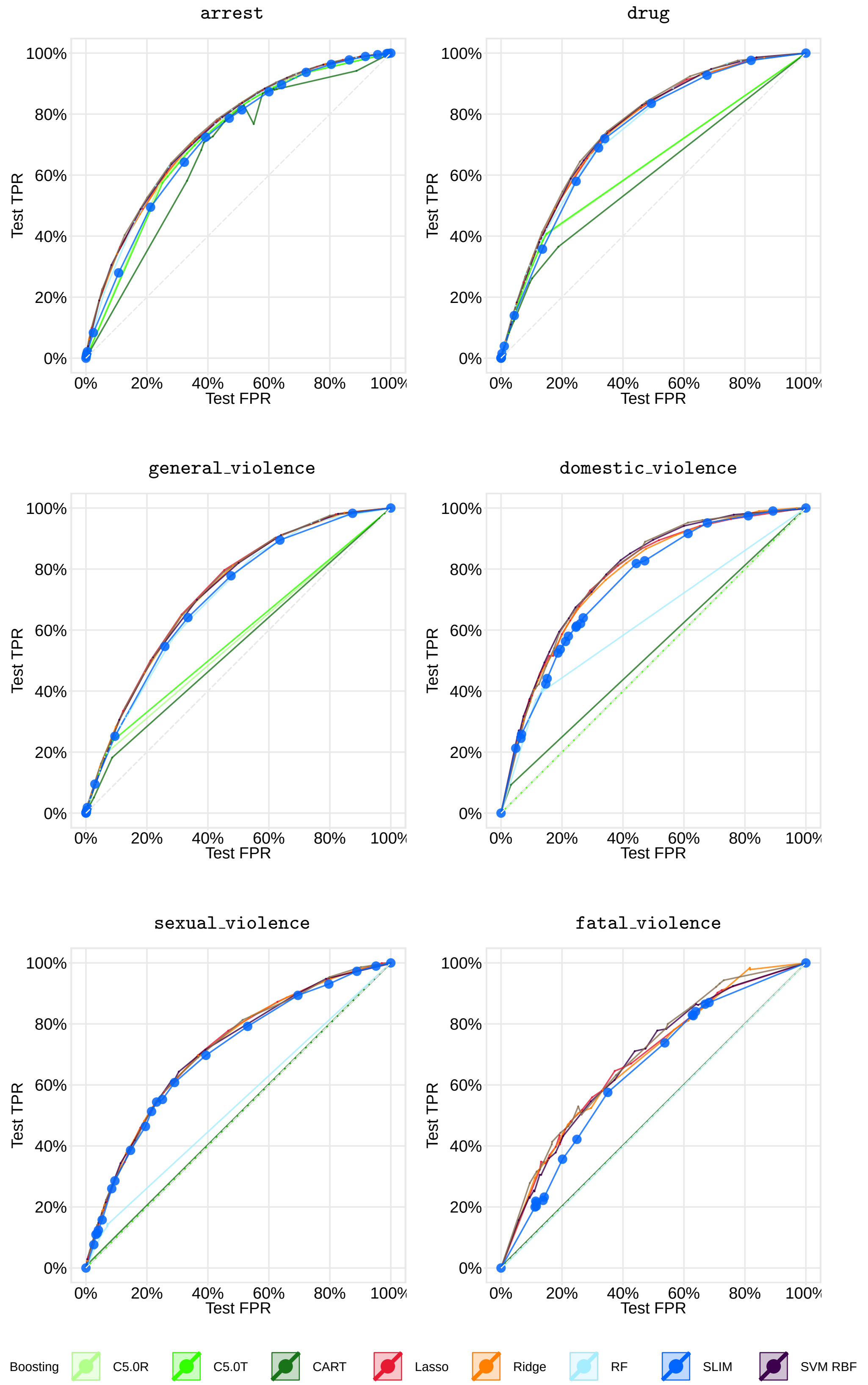


Figure 11 ROC curves for recidivism prediction problems. TPR/FPR for SLIM models are plotted using large blue dots. All models perform similarly except those from C5.0R, C5.0T, and CART.

Results for Recidivism Prediction

Our results were consistent with those from other applications, in that most machine learning algorithms performed almost identically across the full ROC curve, for all of the six prediction problems, as shown in Figure 11. The decision tree methods (CART, C5.0T, C5.0R – in green in Figure 11) sometimes performed poorly, particularly for imbalanced problems. This could potentially illustrate the reason why people often believe that an interpretable modeling algorithm does not perform as accurately as a black box method – methods that produce interpretable models like CART are indeed not as accurate as other methods. CART (9) is not based on optimization, and was designed to operate within the limits of computers from 1984. CART's poor performance is not a convincing reason as to why all interpretable modeling methods might perform poorly.

Our work on this problem was published in the Journal of the Royal Statistical Society (62). This paper won the 2015 Undergraduate Statistics Research Project Competition (USRESP) sponsored by the American Statistical Association (ASA) and the Consortium for Advancement of Undergraduate Statistics Education (CAUSE). Our work was presented at the Sackler Forum on Machine Learning at the National Academy of Sciences in February 2017. Currently the Laura and John Arnold Foundation is constructing a dataset on recidivism using a large amount of judicial data that is now being curated. They have requested to work with us to apply RiskSLIM to construct a risk scoring system.

Insight for Recidivism Prediction: Importance of Certifiable Optimality

Methods like SLIM and RiskSLIM produce certificates of optimality, or optimality gaps in the case where the problems are not fully solved to optimality. These types of guarantees are useful for answering questions such as: "*Does there exist an interpretable model (of a given form) that achieves a particular value for predictive performance?*"

While it is true that optimizing performance on the training set does not correspond exactly to performance on the test set, training and test performances are guaranteed to be similar by statistical learning theory. In fact, if a method cannot achieve high quality in-sample performance, it is difficult for it to achieve high quality out-of-sample performance.

Without the types of tools discussed in this work, it would be easier for companies to promote the use of black box tools in criminal justice. A black box model designer might argue that a comparison with CART, C5.0, or other interpretable modeling methods did not yield an accurate model, and thus, one should use black box models. This argument is a false dichotomy. It is possible that an accurate interpretable model exists, but that some of the older machine learning methods could not find one.

Other Applications

SLIM and RiskSLIM have been used for purposes besides those discussed above. SLIM has been used to detect cognitive impairment, such as Alzheimer's disease, dementia and Parkinson's disease. In particular, the Clock Drawing Test, which is a pen-and-paper test that has been used for a century to diagnose these disorders, has been updated to be digitized. Patients draw clocks with a digital pen, and this digitized test is automatically scored with a SLIM-based system (44). The new scoring system far surpasses the accuracy of all previously published scoring systems for the Clock Drawing Test, and is a promising non-invasive technique for early identification of cognitive impairment. Our work on this project, in conjunction with several collaborators, was published in the Machine Learning journal, and won the 2016 INFORMS Innovative Applications in Analytics Award.

In a separate project using RiskSLIM, we created a screening scale for adult ADHD (attention deficit hyperactivity disorder) in collaboration with a team of psychiatrists (51). The test allows for a quick, risk-calibrated diagnosis based on the answers to 6 questions on a self-reported questionnaire. The questions include: “How often do you have difficulty concentrating on what people say to you, even when they are speaking to you directly?” and “How often do you leave your seat in meetings and other situations in which you are expected to remain seated?” The prediction performance was optimized based on clinical diagnoses using DSM-5 criteria, which is the new standard for adult ADHD diagnosis. The work was published in JAMA Psychiatry in May 2017, and has 11,574 views as of Sept. 22, 2017.

Looking Forward

Within the foreseeable future, there will be a business need to keep the details of machine learning models as a trade secret. In some domains this may not be problematic, particularly when decisions have a minor effect on people’s lives. In other domains, such as healthcare and criminal justice, decisions are serious and actions need to be defensible. The machine learning algorithms presented here represent a fundamental change to the way transparent models are constructed, leveraging modern discrete optimization techniques (cutting planes, data reduction bounds, mixed-integer programming) and capabilities (callback functions, modern solvers). Code for SLIM and RiskSLIM is publicly available, at http://github.com/ustunb/slim-python and http://github.com/ustunb/risk-slim.

References

[1] Elaine Angelino, Nicholas Larus-Stone, Daniel Alabi, Margo Seltzer, and Cynthia Rudin. Learning certifiably optimal rule lists for categorical data. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017.

[2] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing, 2016.

[3] Elliott M Antman, Marc Cohen, Peter JLM Bernink, Carolyn H McCabe, Thomas Horacek, Gary Papuchis, Branco Mautner, Ramon Corbalan, David Radley, and Eugene Braunwald. The TIMI risk score for unstable angina/non–ST elevation MI. *The Journal of the American Medical Association*, 284(7):835–842, 2000.

[4] James Austin, Roger Ocker, and Avi Bhati. Kentucky pretrial risk assessment instrument validation. *Bureau of Justice Statistics. Grant*, (2009-DB), 2010.

[5] Richard A Berk and Justin Bleich. Statistical procedures for forecasting criminal behavior. *Criminology & Public Policy*, 12(3):513–544, 2013.

[6] Philip Bobko, Philip L Roth, and Maury A Buster. The usefulness of unit weights in creating composite scores. A literature review, application to content validity, and meta-analysis. *Organizational Research Methods*, 10(4):689–709, 2007.

[7] RC Bone, RA Balk, FB Cerra, RP Dellinger, AM Fein, WA Knaus, RM Schein, WJ Sibbald, JH Abrams, GR Bernard, et al. American college of chest physicians/society of critical care medicine consensus conference: Definitions for sepsis and organ failure and guidelines for the use of innovative therapies in sepsis. *Critical Care Medicine*, 20(6):864–874, 1992.

[8] Leo Breiman. Statistical modeling: The two cultures. *Statistical Science*, 16(3):199–231, 2001.

[9] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees.* CRC press Boca Raton, Florida, 1984.

[10] Ernest W Burgess. Factors determining success or failure on parole. *The workings of the indeterminate sentence law and the parole system in Illinois*, pages 221–234, 1928.

[11] Shawn D Bushway. Is there any logic to using logit. *Criminology & Public Policy*, 12(3):563–567, 2013.

[12] Rich Caruana and Alexandru Niculescu-Mizil. Data mining in metric space: an empirical analysis of supervised learning performance criteria. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 69–78. ACM, 2004.

[13] Liao P et al. Chung F, Yegneswaran B. Stop questionnaire: a tool to screen patients for obstructive sleep apnea. *Anesthesiology*, 108:812–821, 2008.

[14] Danielle Citron. (Un)fairness of risk scores in criminal sentencing. *Forbes, Tech section*, July 2016.

[15] Shai Danziger, Jonathan Levav, and Liora Avnaim-Pesso. Extraneous factors in judicial decisions. *Proceedings of the National Academy of Sciences*, 108(17):6889–6892, April 2011.

[16] Robyn M Dawes. The robust beauty of improper linear models in decision making. *American psychologist*, 34(7):571–582, 1979.

[17] Ruth Ellen, Shawn C Marshall, Mark Palayew, Frank J Molnar, Keith Wilson, and Malcolm Man-Son-Hing. Systematic review of motor vehicle crash risk in persons with sleep apnea. *Journal of Clinical Sleep Medicine*, 2:193–200, May 2006.

[18] Alex A Freitas. Comprehensible classification models: a position paper. *ACM SIGKDD Explorations Newsletter*, 15(1):1–10, March 2014.

[19] Brian F Gage, Amy D Waterman, William Shannon, Michael Boechler, Michael W Rich, and Martha J Radford. Validation of clinical classification schemes for predicting stroke. *The Journal of the American Medical Association*, 285(22):2864–2870, 2001.

[20] Bryce Goodman and Seth Flaxman. EU regulations on algorithmic decision-making and a "right to explanation". *arXiv preprint arXiv:1606.08813*, 2016. Presented at 2016 ICML Workshop on Human Interpretability in Machine Learning (WHI 2016).

[21] Don M. Gottfredson and Howard N. Snyder. The Mathematics of Risk Classification: Changing Data into Valid Instruments for Juvenile Courts. Technical Report NCJ 209158, Department of Justice, Washington, D.C. Office of Juvenile Justice and Delinquency Prevention, 2005.

[22] RK Hanson and D Thornton. Notes on the development of static-2002. *Ottawa, Ontario: Department of the Solicitor General of Canada*, 2003.

[23] Peter B Hoffman. Twenty years of operational use of a risk prediction instrument: The United States parole commission's salient factor score. *Journal of Criminal Justice*, 22(6):477–494, 1994.

[24] Peter B Hoffman and Sheldon Adelberg. The salient factor score: A nontechnical overview. *Fed. Probation*, 44:44, 1980.

[25] Robert C Holte. Very simple classification rules perform well on most commonly used datasets. *Machine learning*, 11(1):63–90, 1993.

[26] Philip Howard, Brian Francis, Keith Soothill, and Leslie Humphreys. OGRS 3: The revised offender group reconviction scale. Technical report, Ministry of Justice London, UK, 2009.

[27] ILOG. Cplex 11.0 user's manual. ILOG, Inc, 2007.

[28] Murray W Johns et al. A new method for measuring daytime sleepiness: the Epworth sleepiness scale. *Sleep*, 14(6):540–545, 1991.

[29] Daniel Kahneman. *Thinking, fast and slow*. Macmillan, 2011.

[30] William A Knaus, Elizabeth A Draper, Douglas P Wagner, and Jack E Zimmerman. APACHE II: a severity of disease classification system. *Critical Care Medicine*, 13(10):818–829, 1985.

[31] William A Knaus, DP Wagner, EA Draper, JE Zimmerman, Marilyn Bergner, PG Bastos, CA Sirio, DJ Murphy, T Lotring, and A Damiano. The APACHE III prognostic system. risk prediction of hospital mortality for critically ill hospitalized adults. *Chest Journal*, 100(6):1619–1636, 1991.

[32] William A Knaus, Jack E Zimmerman, Douglas P Wagner, Elizabeth A Draper, and Diane E Lawrence. APACHE-acute physiology and chronic health evaluation: a physiologically based classification system. *Critical Care Medicine*, 9(8):591–597, 1981.

[33] Y Kodratoff. The comprehensibility manifesto. *KDD Nugget Newsletter*, 94(9), 1994.

[34] Edward Latessa, Paula Smith, Richard Lemke, Matthew Makarios, and Christopher Lowenkamp. Creation and validation of the Ohio risk assessment system: Final report. *Center for Criminal Justice Research, School of Criminal Justice, University of Cincinnati, Cincinnati, OH. Retrieved from http://www. ocjs. ohio. gov/ORAS_FinalReport. pdf*, 2009.

[35] Jean-Roger Le Gall, Stanley Lemeshow, and Fabienne Saulnier. A new simplified acute physiology score (SAPS II) based on a European/North American multicenter study. *The Journal of the American Medical Association*, 270(24):2957–2963, 1993.

[36] J. Marklof. Fine-scale statistics for the multidimensional Farey sequence. *ArXiv e-prints*, July 2012.

[37] Rui P Moreno, Philipp GH Metnitz, Eduardo Almeida, Barbara Jordan, Peter Bauer, Ricardo Abizanda Campos, Gaetano Iapichino, David Edbrooke, Maurizia Capuzzo, and Jean-Roger Le Gall. SAPS 3 - from evaluation of the patient to evaluation of the intensive care unit. part 2: Development of a prognostic model for hospital mortality at icu admission. *Intensive Care Medicine*, 31(10):1345–1355, 2005.

[38] Northpointe. Correctional offender management profiling for alternative sanctions (COMPAS). http://www.northpointeinc.com/downloads/compas/Practitioners-Guide-COMPAS-Core-_031915.pdf, 2015.

[39] Michael J Pazzani. Knowledge discovery from data? *Intelligent systems and their applications, IEEE*, 15(2):10–12, 2000.

[40] Timoa Pekkala, Anettea Hall, Jyrkib Lötjönen, Jussic Mattila, Hilkkaa Soininen, Tiiae Ngandu, Tiinae Laatikainen, Miiaa Kivipelto, and Alina Solomon. Development of a late-life dementia prediction index with supervised machine learning in the population-based CAIDE study. *Journal of Alzheimer's Disease*, 55:1055–1067, 2017.

[41] Pennsylvania Commission on Sentencing. Interim Report 4: Development of Risk Assessment Scale. Technical report, June 2012.

[42] Paul A Rubin. Mixed integer classification problems. In *Encyclopedia of Optimization*, pages 2210–2214. Springer, 2009.

[43] AJ Six, BE Backus, and JC Kelder. Chest pain in the emergency room: value of the HEART score. *Netherlands Heart Journal*, 16(6):191–196, 2008.

[44] William Souillard-Mandar, Randall Davis, Cynthia Rudin, Rhoda Au, David J. Libon, Rodney Swenson, Catherine C. Price, Melissa Lamar, and Dana L. Penney. Learning classification models of cognitive conditions from subtle behaviors in the digital clock drawing test. *Machine Learning*, 102(3):393–441, 2016.

[45] Aaron F. Struck, Berk Ustun, Andres Rodriguez Ruiz, Jong Woo Lee, Suzette LaRoche, Lawrence J. Hirsch, Emily J. Gilmore, Cynthia Rudin, and Brandon M Westover. A practical risk score for EEG seizures in hospitalized patients. *Forthcoming in JAMA Neurology*, 2017.

[46] Martin Than, Dylan Flaws, Sharon Sanders, Jenny Doust, Paul Glasziou, Jeffery Kline, Sally Aldous, Richard Troughton, Christopher Reid, William A Parsonage, Christopher Frampton, Jaimi H Greenslade, Joanne M Deely, Erik Hess, Amr Bin Sadiq, Rose Singleton, Rosie Shopland, Laura Vercoe, Morgana Woolhouse-Williams, Michael Ardagh, Patrick Bossuyt, Laura Bannister, and Louise Cullen. Development and validation of the emergency department assessment of chest pain score and 2 h accelerated diagnostic protocol. *Emergency Medicine Australasia*, 26(1):34–44, 2014.

[47] Nikolaj Tollenaar and P.G.M. van der Heijden. Which method predicts recidivism best?: a comparison of statistical, machine learning and data mining predictive models. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 176(2):565–584, 2013.

[48] U.S. Department of Justice, Bureau of Justice Statistics. Recidivism of prisoners released in 1994. http://doi.org/10.3886/ICPSR03355.v8, 2014.

[49] U.S. Sentencing Commission. 2012 guidelines manual: Chapter four - criminal history and criminal livelihood, November 1987.

[50] U.S. Sentencing Commission. Measuring recidivism: The criminal history computation of the federal sentencing guidelines. 2004.

[51] Berk Ustun, Lenard A Adler, Cynthia Rudin, Stephen V Faraone, Thomas J Spencer, Patricia Berglund, Michael J Gruber, and Ronald C Kessler. The World Health Organization Adult Attention-Deficit/Hyperactivity Disorder Self-Report Screening Scale for DSM-5. *JAMA Psychiatry*, 74(5):520–526, 2017.

[52] Berk Ustun and Cynthia Rudin. Learning Optimized Risk Scores for Large-Scale Datasets. *arXiv:1610.00168*, 2016.

[53] Berk Ustun and Cynthia Rudin. Supersparse Linear Integer Models for Optimized Medical Scoring Systems. *Machine Learning*, 102(3):349–391, 2016.

[54] Berk Ustun and Cynthia Rudin. Optimized Risk Scores. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017.

[55] Berk Ustun, M.B. Westover, Cynthia Rudin, and Matt T. Bianchi. Clinical prediction models for sleep apnea: The importance of medical history over symptoms. *Journal of Clinical Sleep Medicine*, 12(2):161–168, 2016.

[56] Vladimir Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

[57] Frank W Weathers, Brett T Litz, Terence M Keane, Patrick A Palmieri, Brian P Marx, and Paula P Schnurr. The ptsd checklist for dsm-5 (pcl-5). *Scale available from the National Center for PTSD at www.ptsd.va.gov.*, 2013.

[58] Stephen F. Weng, Jenna Reps, Joe Kai, Jonathan M. Garibaldi, and Nadeem Qureshi. Can machine-learning improve cardiovascular risk prediction using routine clinical data? *PLoS ONE*, April 2017.

[59] Rebecca Wexler. When a computer program keeps you in jail: How computers are harming criminal justice. *New York Times*, June 2017.

[60] Laurence A Wolsey. *Integer programming*, volume 42. Wiley New York, 1998.

[61] Bianca Zadrozny and Charles 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. ACM, 2002.

[62] Jiaming Zeng, Berk Ustun, and Cynthia Rudin. Interpretable classification models for recidivism prediction. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 180(3):689–722, 2017.

Appendix: Optimization Problems

We start with a dataset of N i.i.d. training examples $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)_{i=1}^N\}$ where $\boldsymbol{x}_i \in \mathcal{X} \subseteq \mathbb{R}^{d+1}$ denotes a vector of features $[1, x_{i,1}, \ldots, x_{i,d}]^\top$ and $y_i \in \mathcal{Y} = \{-1, 1\}$ denotes a class label. We consider linear classification models of the form $\hat{y} = \text{sign}(\langle \boldsymbol{\lambda}, \boldsymbol{x} \rangle)$, where $\boldsymbol{\lambda} = [\lambda_0, \lambda_1, \ldots, \lambda_d]^\top$ represents a vector of coefficients and λ_0 represents an intercept.

In this setup, the coefficient vector $\boldsymbol{\lambda}$ determines all parameters of a scoring system. In particular, the coefficient λ_j represents the *points* for feature j for $j = 1, \ldots, d$. Given an example with features $\boldsymbol{x}_i$, users first tally the points for all features such that $\lambda_j \neq 0$ to obtain a total *score* $\sum_{j=1}^d \lambda_j x_{i,j}$ then use the total score to obtain a predicted label (i.e. for decision-making) or a estimate of predicted risk (i.e. for risk assessment).

SLIM's **Optimization Framework for Decision-Making**

In decision-making applications, we use the score to output a predicted label $\hat{y} \in \{-1, 1\}$ through a decision rule of the form:

$$\hat{y}_i = \begin{cases} +1 & \text{if } \displaystyle\sum_{j=1}^{d} \lambda_j x_{i,j} + \lambda_0 > 0, \\ -1 & \text{if } \displaystyle\sum_{j=1}^{d} \lambda_j x_{i,j} + \lambda_0 \leq 0. \end{cases}$$

In this setting, we learn the values of coefficients by solving a discrete optimization problem that we refer to as the *decision rule problem*. The optimal solution to the decision rule problem is a *Supersparse Linear Integer Model*. The decision rule problem is a discrete optimization problem of the form:

$$\begin{aligned} \min_{\boldsymbol{\lambda}} \quad & l_{01}(\boldsymbol{\lambda}) + C_0 \left\|\boldsymbol{\lambda}\right\|_0 \\ \text{s.t.} \quad & \boldsymbol{\lambda} \in \mathcal{L}, \\ & \gcd(\boldsymbol{\lambda}) = 1, \end{aligned} \tag{1}$$

where:

- $l_{01}(\boldsymbol{\lambda}) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left[\hat{y}_i \neq y_i\right]$ is the fraction of misclassified observations;
- $\|\boldsymbol{\lambda}\|_0 = \sum_{j=1}^{d} \mathbb{1}\left[\lambda_j \neq 0\right]$ is the count of non-zero coefficients, ℓ_0-seminorm;
- $\mathcal{L} \subset \mathbb{Z}^{d+1}$ is a finite user-provided set of feasible coefficient vectors, usually chosen to be small integers, $\mathcal{L} = \{-10, \ldots, 10\}^{d+1}$;
- $C_0 > 0$ is a user-chosen trade-off parameter to balance accuracy and sparsity;
- $\text{gcd}(\boldsymbol{\lambda}) = 1$ is a symmetry-breaking constraint to ensure coefficients are co-prime. Here "gcd" stands for greatest common divisor.

Here, the objective minimizes the empirical probability of misclassification, and penalizes the number of non-zero terms to encourage the model to be sparse. The feasible region can be customized to include additional operational constraints (see Table 1). In practice the fraction of misclassifications is replaced with a weighted sum of false positives and false negatives, for applications where one of these is more important to reduce than the other.

To implement the decision rule problem as a mathematical program, there is a simple trick for encoding the constraint that the gcd of the coefficients is 1. In particular, if we add a term to the objective that is the sum of the absolute coefficients, multiplied by a very small number (ϵ in the formulation below), it forces the gcd to be 1 without influencing either accuracy or sparsity. The reason this trick works is because the loss and sparsity terms take on only discrete values. Among all models that are equally accurate and equally sparse, the formulation will choose the one with the smallest absolute sum of terms, $\sum_j |\lambda_j|$. Since the values of the λ_j are also integers, they must be co-prime.

Using this additional term in the objective, we formulate the decision rule problem as a mixed integer linear program as follows to obtain a SLIM scoring system, as follows.

$$\min \quad V = L + C_0 R + \epsilon \sum_{j=1}^{d} \beta_j \qquad \textit{objective value} \qquad (2a)$$

$$\text{s.t.} \quad L = \frac{w^+}{N} \sum_{i \in \mathcal{I}^+} z_i + \frac{w^-}{N} \sum_{i \in \mathcal{I}^-} z_i \qquad \textit{weighted misclassification error} \qquad (2b)$$

$$R = \sum_{j=1}^{d} \alpha_j \qquad \qquad \textit{model size} \tag{2c}$$

$$M_i z_i \geq \gamma - y_i \left(\sum_{j=0}^{d} \lambda_j x_{i,j} \right) \qquad i = 1,\ldots,N \qquad \textit{definition of } z_i \textit{ as misclassification} \tag{2d}$$

$$\Lambda^{\max} j \alpha_j \geq \lambda_j \qquad j = 1,\ldots,d \qquad \alpha_j \textit{ is 1 if } \lambda_j \textit{ is nonzero} \tag{2e}$$

$$\Lambda^{\min} j \alpha_j \geq -\lambda_j \qquad j = 1,\ldots,d \qquad \alpha_j \textit{ is 1 if } \lambda_j \textit{ is nonzero} \tag{2f}$$

$$\beta_j \geq \lambda_j \qquad j = 1,\ldots,d \qquad \beta_j \textit{ is } |\lambda_j| \tag{2g}$$

$$\beta_j \geq -\lambda_j \qquad j = 1,\ldots,d \qquad \beta_j \textit{ is } |\lambda_j| \tag{2h}$$

$$\lambda_j \in \mathcal{L}_j \qquad j = 0,\ldots,d \qquad \textit{coefficient set}$$

$$z_i \in \{0,1\} \qquad i = 1,\ldots,N \qquad \textit{error indicators}$$

$$\alpha_j \in \{0,1\} \qquad j = 1,\ldots,d \qquad \ell_0 \textit{ variables}$$

$$\beta_j \in \mathbb{R}_+ \qquad j = 1,\ldots,d \qquad \textit{co-primeness variables.}$$

In Equation 2d, the value $y_i \left(\sum_j \lambda_j x_{i,j} \right)$ has the same sign as $y_i \cdot \hat{y}_i$, by the definition of $\hat{y}$. If $\hat{y}_i$ and y_i do not have the same sign, it means that observation i has been misclassified. When that happens, $y_i \cdot \hat{y}_i$ is negative, $y_i \left(\sum_j \lambda_j x_{i,j} \right)$ is negative, and for small γ, variable z_i is forced to be 1 by Equation 2d. Thus, z_i is an indicator that observation i is misclassified. Equation 2d is a Big-M constraint that depends on scalar parameters γ and M_i (see e.g., 42). The value of M_i represents the maximum score when observation i is misclassified, and can be set as $M_i = \max_{\boldsymbol{\lambda} \in \mathcal{L}} (\gamma - y_i(\boldsymbol{\lambda}^\top \boldsymbol{x}_i))$ which is easy to compute since $\mathcal{L}$ is finite. The value of γ represents the traditional "margin" in machine learning. It is the smallest value of $y_i(\boldsymbol{\lambda}^\top \boldsymbol{x}_i)$ that could be considered as a correct classification. When the features are binary, γ can be set to any value between 0 and 1, with any choice leading to the same solution. In other cases, the lower bound is difficult to calculate exactly so we set $\gamma = 0.1$, which makes an implicit assumption on the values of the features. (Usually, however, we choose to binarize the features beforehand to avoid this arbitrary choice.) The model size is set to R in constraint (2c) via the indicator variables $\alpha_j := \mathbb{1}[\lambda_j \neq 0]$. These variables are defined by Big-M constraints in (2e) – (2f), and $\beta_j := |\lambda_j|$ is defined by the constraints in (2g)–(2h).

The choices for w^+ and w^- in the objective are the relative importance of false positives and false negatives. These values should generally be chosen by the user, depending on how much a false positive is worth relative to a false negative in the application. Often, we try many possible values of w^+ and w^- to create several models that are optimized for specific points on the ROC curve. We have finished discussing the formulation for SLIM, now we move on to RiskSLIM.

RiskSLIM's **Optimization Framework for Risk Assessment**

In risk assessment applications, we use the score to estimate of predicted risk. Specifically, we estimate the *predicted risk* that example i belongs to the positive class using the logistic link function as:

$$\Pr\left(y_i = +1 \mid \boldsymbol{x}_i\right) = \frac{1}{1 + \exp(-\boldsymbol{\lambda}^T \boldsymbol{x}_i)}.$$

We learn the values of the coefficients from data by solving the following mixed integer nonlinear program (MINLP), which we refer to as the *risk score problem* or RiskSlim-MINLP:

$$\begin{array}{ll} \min\limits_{\boldsymbol{\lambda}} & l(\boldsymbol{\lambda}) + C_0 \left\|\boldsymbol{\lambda}\right\|_0 \\ \text{s.t.} & \boldsymbol{\lambda} \in \mathcal{L}, \end{array} \tag{3}$$

where:

- $l(\boldsymbol{\lambda}) = \frac{1}{N}\sum_{i=1}^{N} \log(1 + \exp(-\boldsymbol{\lambda}^T y_i \boldsymbol{x}_i))$ is the logistic loss function;
- $\left\|\boldsymbol{\lambda}\right\|_0 = \sum_{j=1}^{d} \mathbb{1}\left[\lambda_j \neq 0\right]$ is the ℓ_0-seminorm;
- $\mathcal{L} \subset \mathbb{Z}^{d+1}$ is a set of feasible coefficient vectors (user-provided);
- $C_0 > 0$ is a trade-off parameter to balance fit and sparsity (user-provided);

The optimal solution to the risk score problem is a scoring system that we refer to as a *Risk-calibrated Supersparse Linear Integer Model.*

Here, the objective minimizes the *logistic loss* from logistic regression in order to achieve high values of the area under the ROC curve (AUC) and to achieve risk calibration. The objective penalizes the ℓ_0-seminorm for sparsity. The trade-off parameter C_0 controls the balance between these competing objectives, and represents the maximum log-likelihood that is sacrificed to remove a feature from the optimal model. The feasible region restricts coefficients to a small set of bounded integers such as $\mathcal{L} = \{-10, \ldots, 10\}^{d+1}$, and may be further customized to include operational constraints, such as those in Table 1.

In order to fit a RiskSLIM scoring system, we need to solve a mixed integer non-linear program (MINLP). This MINLP is difficult to solve using any commercial solver. Cutting plane algorithms are a natural choice for this problem because the objective is continuous and convex, but we were not able to use a traditional cutting plane algorithm because of the discrete domain of the optimization problem. Instead, we designed a specialized cutting plane technique that creates a series of branches, where we compute cutting planes on each branch. This allows us to solve very large problems and parallelize easily. Let us discuss this in more depth. We will first show that traditional cutting plane methods stall because of the discrete domain, and present the Lattice Cutting Plane method, which does not stall.

Cutting Plane Algorithms Stall in Non-Convex Domains

Let us explain why traditional cutting plane algorithms have trouble with discrete domains, even when the objective function is convex and differentiable. A traditional cutting plane method iteratively creates a piecewise linear lower bound (a surrogate) to the objective function, as shown in Figure 12. To create the surrogate, the algorithm alternates between two steps: creating a cutting plane that is a tangent plane to the objective, and minimizing over the surrogate to determine the next location to create a cutting plane.

The first step, creating the cutting plane, is a calculation that uses the data, and does not involve optimization. The second step, minimizing over the surrogate, does not involve data, and solves an optimization problem using the cutting planes discovered thus far.

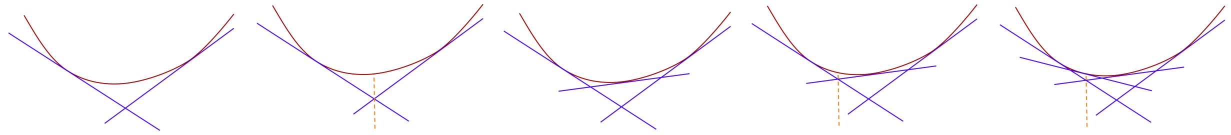


Figure 12 **Illustration of adding cuts in a traditional cutting plane algorithm. Starting with cuts found previously (first figure), we find the minimum of the piecewise linear lower bound surrogate (in blue) and evaluate the function at that value (second figure). We then construct a new cutting plane (third figure), again find the minimum of the surrogate (fourth figure), and construct a new cutting plane (fifth figure). This continues until the minimum of the surrogate matches the function value, in which case the minimum of the function has been attained.**

The algorithm completes when the minimum of the surrogate function is equal to the value of the objective. Thus, in order to determine that the algorithm has found an optimal solution, we must be able to minimize the surrogate function to provable optimality. If we cannot minimize the surrogate to provable optimality, we will have no way of knowing when we have reached the optimal solution, nor how far we are from optimality.

If the domain is discrete, it can be difficult to minimize the surrogate. Minimizing over the surrogate requires solving a mixed integer program (MIP) whose complexity grows rapidly with the number of iterations. After several iterations, the solver tends to have difficulty solving the MIP. Minimization of the surrogate gets harder as iterations increase, so that if we cannot solve the optimization at one iteration, the problem only gets worse later on.

Figure 13 shows the stalling of a traditional cutting plane algorithm on a RiskSLIM MINLP instance where $d = 20$ (in black). As shown, the time to minimize the surrogate

MIP tends to increase exponentially over iterations. The algorithm stalls on iteration $k = 87$ when the surrogate MIP cannot be solved. Even after 6 hours, the best feasible solution found is highly suboptimal, with a large optimality gap (which makes sense as the solution optimizes a cutting-plane approximation that uses at most 86 cuts). The value of the loss determines the performance of the model, and thus the risk score we obtain after 6 hours performs poorly.

There is not an easy solution to this problem of stalling.

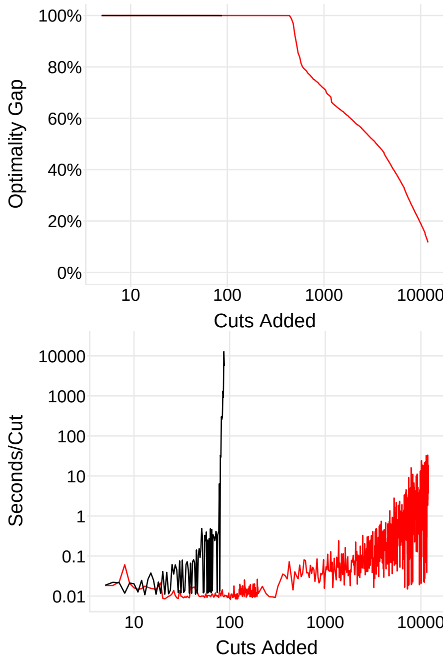


Figure 13 **Traditional cutting plane algorithm (black) and LCPA (red) on a simulated dataset with $d = 20$ and $N =$ 50,000. We show the optimality gap (top) and the time to add a new cut (bottom; log-scale) over 6 hours. The traditional cutting plane algorithm stalls after adding 86 cuts as the time to optimize RiskSlimMIP increases exponentially. The resulting solution corresponds to a risk score with poor performance. In contrast, LCPA does not stall, finding a near-optimal solution in 9 minutes, and the optimal solution in 234 minutes. LCPA uses the remaining time to reduce the optimality gap.**

Lattice Cutting Plane Algorithm

To avoid the stalling behavior of existing cutting-plane algorithms in non-convex settings, we solve the risk score problem using the *lattice cutting plane algorithm* (LCPA) shown in Algorithm 1), which we developed for producing risk scores. LCPA is a cutting-plane algorithm that recovers the optimal solution to RiskSlimMINLP via *branch-and-bound* (B&B) search. The search recursively splits the feasible region of RiskSlimMINLP into disjoint partitions, discarding partitions that are infeasible or provably suboptimal. LCPA solves a *surrogate linear program* (LP) over each partition. In this approach, the cutting-plane approximation is updated whenever the surrogate LP yields an integer feasible solution. The lower bound is set as the smallest possible value of the surrogate LP over the remaining search region.

As shown in Figure 13, LCPA (in red) does not stall. This is because LCPA does not need to optimize a non-convex surrogate to add cuts or to compute a valid lower bound. In what follows, we describe the main elements of LCPA.

Branch and Bound Search

In Algorithm 1, we represent the state of the branch and bound (B&B) search using a B&B tree. This tree is composed of nodes (i.e. leaves) in the *node set* $\mathcal{N}$. Each *node* $(\mathcal{P}_n, v_n) \in \mathcal{N}$ consists of a *partition* of the convex hull of constraint set $\mathcal{P}_n \subseteq \text{conv}(\mathcal{L})$, and a lower bound for the optimal value of the surrogate over this partition, v_n.

Each iteration of LCPA starts by removing a node $(\mathcal{P}_n, v_n)$ from the node set $\mathcal{N}$ and solving the surrogate over $\mathcal{P}_n$ (Steps 2-6 in the algorithm). The next several steps depend on the feasibility of RiskSlimLP$(\hat{l}^k(\cdot), \mathcal{P}_n)$:

- If RiskSlimLP$(\hat{l}^k(\cdot), \mathcal{P}_n)$ yields an integer solution $\boldsymbol{\lambda}^{\text{LP}} \in \mathcal{L}$, LCPA updates the cutting plane approximation $\hat{l}^k(\cdot)$ with a cut at $\boldsymbol{\lambda}^{\text{LP}}$ in Step 8. If the solution found is the best

Algorithm 1 Lattice Cutting Plane Algorithm (LCPA)

```
Input
    $(\boldsymbol{x}_i, y_i)_{i=1}^N$                        training data
    $\mathcal{L}$                                            constraint set for RiskSlimMINLP
    $C_0$                                                    $\ell_0$ penalty parameter
    $\varepsilon^{\text{stop}} \in [0,1]$                    optimality gap of acceptable solution
    RemoveNode                                               rule to pick a node from a node set (provided by MIP solver)
    SplitPartition                                           rule to split a partition into disjoint subsets (provided by MIP solver)

Initialize
    $k \leftarrow 0$                                         number of cuts
    $\hat{l}^0(\boldsymbol{\lambda}) \leftarrow \{0\}$      cutting-plane approximation of loss function
    $(V^{\min}, V^{\max}) \leftarrow (0, \infty)$            bounds on the optimal value
    $\varepsilon \leftarrow \infty$                          optimality gap
    $\mathcal{P}_0 \leftarrow \text{conv}(\mathcal{L})$      partition for initial node
    $v_0 \leftarrow V^{\min}$                                lower bound for initial node
    $\mathcal{N} \leftarrow \{(\mathcal{P}_0, v_0)\}$        initial node set

1:  while $\varepsilon > \varepsilon^{\text{stop}}$ do
2:      $(\mathcal{P}_t, v_t) \leftarrow$ RemoveNode $(\mathcal{N})$                    ▷$t$ is index of removed node
3:      solve RiskSlimLP$(\hat{l}^k(\cdot), \mathcal{P}_t)$
4:      $\boldsymbol{\lambda}^{\text{LP}} \leftarrow$ coefficients from optimal solution to RiskSlimLP$(\hat{l}^k(\cdot), \mathcal{P}_t)$
5:      $v^{\text{LP}} \leftarrow$ optimal value of RiskSlimLP$(\hat{l}^k(\cdot), \mathcal{P}_t)$
6:      if optimal solution is integer feasible then
7:          compute cut parameters $l(\boldsymbol{\lambda}^{\text{LP}})$ and $\nabla l(\boldsymbol{\lambda}^{\text{LP}})$
8:          $\hat{l}^{k+1}(\boldsymbol{\lambda}) \leftarrow \max\{\hat{l}^k(\boldsymbol{\lambda}), l(\boldsymbol{\lambda}^{\text{LP}}) + \langle \nabla l(\boldsymbol{\lambda}^k), \boldsymbol{\lambda} - \boldsymbol{\lambda}^{\text{LP}} \rangle\}$      ▷update approximation $\forall \boldsymbol{\lambda}$
9:          if $v^{\text{LP}} < V^{\max}$ then
10:             $V^{\max} \leftarrow v^{\text{LP}}$                                      ▷update lower bound
11:             $\boldsymbol{\lambda}^{\text{best}} \leftarrow \boldsymbol{\lambda}^{\text{LP}}$          ▷update best solution
12:             $\mathcal{N} \leftarrow \mathcal{N} \setminus \{(\mathcal{P}_s, v_s) \mid v_s \geq V^{\max}\}$      ▷prune suboptimal nodes
13:           end if
14:           $k \leftarrow k + 1$
15:     else if optimal solution is not integer feasible then
16:         $(\mathcal{P}', \mathcal{P}'') \leftarrow$ SplitPartition$(\mathcal{P}_t, \boldsymbol{\lambda}^{\text{LP}})$      ▷$\mathcal{P}'$, $\mathcal{P}''$ are disjoint subsets of $\mathcal{P}_t$
17:         $(v', v'') \leftarrow (v^{\text{LP}}, v^{\text{LP}})$                         ▷$v^{LP}$ is lower bound for $\mathcal{P}', \mathcal{P}''$
18:         $\mathcal{N} \leftarrow \mathcal{N} \cup \{(\mathcal{P}', v'), (\mathcal{P}'', v'')\}$      ▷add child nodes to $\mathcal{N}$
19:     end if
20:     $V^{\min} \leftarrow \min_{\mathcal{N}} v_s$                    ▷lower bound is smallest lower bound among nodes in $\mathcal{N}$
21:     $\varepsilon \leftarrow 1 - V^{\min}/V^{\max}$                  ▷update optimality gap
22: end while
Output: $\boldsymbol{\lambda}^{\text{best}}$                            $\varepsilon$-optimal solution to RiskSlimMINLP
```

RiskSlimLP$(\hat{l}(\cdot), \mathcal{P})$ is a LP relaxation of RiskSlimMIP$(\hat{l}(\cdot))$ over the partition $\mathcal{P} \subseteq \text{conv}(\mathcal{L})$:

$$
\begin{aligned}
\min_{\theta, \boldsymbol{\lambda}, \boldsymbol{\alpha}} \quad & \theta + C_0 \sum_{j=1}^{d} \alpha_j \\
\text{s.t.} \quad & \boldsymbol{\lambda} \in \mathcal{P} \\
& \theta \geq \hat{l}(\boldsymbol{\lambda}) \\
& \alpha_j = \max(\lambda_j, 0)/\Lambda_j^{\max} + \min(\lambda_j, 0)/\Lambda_j^{\min} \text{ for } j = 1 \ldots d.
\end{aligned}
\tag{4}
$$

so far, we update the current best solution in Step 11 and prune suboptimal nodes in Step 12.

- If RiskSlimLP$(\hat{l}^k(\cdot), \mathcal{P}_n)$ yields a continuous solution $\boldsymbol{\lambda}^{\text{LP}} \notin \mathcal{L}$, then LCPA splits the partition $\mathcal{P}_n$ into disjoint subsets $\mathcal{P}'$ and $\mathcal{P}''$. Each subset is paired with the optimal value of the surrogate LP to yield the child nodes $(\mathcal{P}', v^{\text{LP}})$ and $(\mathcal{P}'', v^{\text{LP}})$. The child nodes are added back into $\mathcal{N}$ in Step 18.

- If $\textsc{RiskSlim}\text{LP}(\hat{l}^k(\cdot), \mathcal{P}_n)$ is infeasible, the node is discarded.

The search process uses two rules that are typically provided by a MIP solver:

- RemoveNode, which takes as input the node set $\mathcal{N}$ and outputs a node $(\mathcal{P}_n, v_n)$ (e.g., the node with the smallest v_n).
- SplitPartition, which takes as input a partition $\mathcal{P}_n$ and the current solution $\boldsymbol{\lambda}^{\text{LP}}$ and outputs disjoint partitions that do not cover $\mathcal{P}_n$ (e.g. split on a fractional component of the solution λ_j^{LP}, which returns $\mathcal{P}' = \{\boldsymbol{\lambda} \in \mathcal{P}_{\text{LP}} \,|\, \lambda_j^{\text{LP}} \geq \lceil \lambda_j^{\text{LP}} \rceil\}$ and $\mathcal{P}'' = \{\boldsymbol{\lambda} \in \mathcal{P}_{\text{LP}} \,|\, \lambda_j^{\text{LP}} \leq \lceil \lambda_j^{\text{LP}} \rceil\}$). These output conditions ensure that: (i) the partitions of all nodes in the node set remain disjoint; (ii) the search region shrinks even if the solution to the surrogate is not integer feasible; (iii) the number of nodes is finite.

Convergence

LCPA checks convergence using bounds on the optimal value of $\textsc{RiskSlim}\text{MINLP}$. The upper bound $V^{\max}$ is set as the objective value of the best integer feasible solution in Step 11. The lower bound $V^{\min}$ is set as the smallest lower bound among all nodes in Step 20. The quantity $V^{\min}$ is a lower bound on the optimal value of the surrogate over the *remaining search region* $\bigcup_n \mathcal{P}_n$; that is, the optimal value of $\textsc{RiskSlim}\text{LP}(\hat{l}^k(\cdot), \bigcup_n \mathcal{P}_n)$. Thus, $V^{\min}$ improves when we add cuts or reduce the remaining search region.

Each iteration of LCPA reduces the remaining search region as it either finds an integer feasible solution, identifies an infeasible partition, or splits a partition into disjoint subsets. Thus, $V^{\min}$ increases monotonically as the search region becomes smaller, and cuts are added at integer feasible solutions. Likewise, $V^{\max}$ decreases monotonically, and the search is guaranteed to find the optimal solution. Since there are a finite number of nodes, LCPA terminates after a finite number of iterations.

Implementation

We implemented LCPA using a MIP solver that provides *control callbacks*, such as CPLEX. The solver handles all B&B related steps in Algorithm 1 and control callbacks let us update the cutting-plane approximation by intervening in the search. In a basic implementation, we use a control callback to intervene when Algorithm 1 reaches Step 6. Our code retrieves the integer feasible solution, computes the cut parameters, adds a cut, and returns control back to solver by Step 9.

We have finished describing the RiskSLIM algorithm and its cutting plane algorithm LCPA. The methodological paper on RiskSLIM (52, 54) was awarded the 2017 INFORMS Computing Society (ICS) student paper prize.

Appendix: Generalization Bounds

One of the key properties of scoring systems is that they generalize well out of sample. This observation is supported by the empirical results in our applications and often mentioned in the literature from applied domains (see e.g. work on the out-of-sample performance of linear models with unit weights, 16, 6).

We can motivate the generalization of these models from a machine learning perspective using ideas from statistical learning theory, and in particular, structural risk minimization (56). The main types of results from statistical learning theory are probabilistic bounds on out-of-sample performance. These bounds imply that there are two key ingredients for good out-of-sample generalization: (i) good in-sample performance, and (ii) using low-complexity functions. Statistical learning theory is a formal statement of why simple models that perform well on the data tend to generalize well, and is a mathematical formalism for the principle of Occam's Razor.

In what follows, we will state a series of statistical learning theoretic bounds that are specialized for scoring systems. The first bound is a traditional Occam's Razor style bound. The second and third bounds improve the first bound. Statistical learning theoretic bounds are not generally tight enough to be used in practice, but they motivate specific choices within algorithms. Learning-theoretic bounds often are proved using arguments involving Hoeffding's inequality or other related tail bounds, such as McDiarmid's inequality.

Consider fitting a classifier $f:\mathcal{X}\rightarrow\mathcal{Y}$ with data $\mathcal{D}_N=(\boldsymbol{x}_i,y_i)_{i=1}^N$, where $\boldsymbol{x}_i\in\mathcal{X}\subseteq\mathbb{R}^d$ and $y_i\in\mathcal{Y}=\{-1,1\}$. In Theorem 1, we present a basic uniform generalization guarantee on the predictive accuracy of an algorithm that chooses a function from a class of functions $\mathcal{F}$. This guarantee bounds the *true risk*,

$$R^{\text{true}}(f)=\mathbb{E}_{\mathcal{X},\mathcal{Y}}\mathbb{1}\left[f(\boldsymbol{x})\neq y\right],$$

by the *empirical risk*,

$$R^{\text{emp}}(f) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left[f(\boldsymbol{x}_i) \neq y_i\right],$$

and other quantities important to the learning process. The true risk represents out-of-sample performance, and the empirical risk measures in-sample performance.

Theorem 1 **(Generalization of Discrete Linear Classifiers)**. *Let $\mathcal{F}$ denote the set of linear classifiers with coefficients $\boldsymbol{\lambda} \in \mathcal{L}$:*

$$\mathcal{F} = \left\{ f : \mathcal{X} \rightarrow \mathcal{Y} \;\middle|\; f(\boldsymbol{x}) = \text{sign}\left(\langle \boldsymbol{\lambda}, \boldsymbol{x} \rangle\right) \text{ and } \boldsymbol{\lambda} \in \mathcal{L} \right\}.$$

For every $\delta > 0$, with probability at least $1 - \delta$, every classifier $f \in \mathcal{F}$ obeys:

$$R^{\text{true}}(f) \leq R^{\text{emp}}(f) + \sqrt{\frac{\log(|\mathcal{L}|) - \log(\delta)}{2N}}.$$

That is, with high probability, the out-of-sample error $R^{\text{true}}(f)$ (which we cannot measure) is not more than the in-sample error $R^{\text{emp}}(f)$ (which we can measure) plus a quantity that relates to the size of the hypothesis space $|\mathcal{L}|$. The result shows that more restrictive hypothesis spaces (smaller $\mathcal{L}$'s) lead to better generalization. This bound provides motivation for using sparse linear models with integer coefficients (as oppose to more complex model classes). The proofs of all theorems are in (53).

In what follows, we improve (tighten) the generalization bound from Theorem 1 for SLIM scoring systems. In Theorem 2, we improve the generalization bound from Theorem 1 by excluding models from the hypothesis space that are provably suboptimal. Here, we exploit the fact that we can bound the number of non-zero coefficients using the trade-off parameter C_0.

In the theorem's notation, we indicate $l_{01}(\boldsymbol{\lambda}, S)$ to mean that the misclassification error is being computed on dataset S using a linear scoring system with coefficients $\boldsymbol{\lambda}$.

Theorem 2 **(Generalization of Sparse Discrete Linear Classifiers)**.

Let $\mathcal{F}$ denote the set of linear classifiers with coefficients $\boldsymbol{\lambda}$ from a finite set $\mathcal{L}$. Here $\mathcal{L}$ contains the trivial model, $\mathbf{0} \in \mathcal{L}$. $\mathcal{F}$ is the class of models such that:

$$\mathcal{F} = \left\{ f_{\boldsymbol{\lambda}} : \mathcal{X} \to \mathcal{Y} \,\middle|\, f_{\boldsymbol{\lambda}}(\boldsymbol{x}) = \operatorname{sign}\left(\langle \boldsymbol{\lambda}, \boldsymbol{x} \rangle\right) \right\}, \quad \textit{(linear models with coefficients from } \mathcal{L}\textit{)}$$

$$\boldsymbol{\lambda} \in \operatorname*{arg\,min}_{\boldsymbol{\lambda}' \in \mathcal{L}} l_{01}(\boldsymbol{\lambda}', S) + C_0 \left\|\boldsymbol{\lambda}'\right\|_0 \textit{ for at least one dataset } S.$$

Then for every $\delta > 0$, with probability at least $1 - \delta$, every classifier $f_{\boldsymbol{\lambda}} \in \mathcal{F}$ obeys:

$$R^{\text{true}}(f_{\boldsymbol{\lambda}}) \leq R^{\text{emp}}(f_{\boldsymbol{\lambda}}) + \sqrt{\frac{\log(|\mathcal{H}_{d,C_0}|) - \log(\delta)}{2N}},$$

where:

$$\mathcal{H}_{d,C_0} = \left\{ \boldsymbol{\lambda} \in \mathcal{L} \,\middle|\, \|\boldsymbol{\lambda}\|_0 \leq \left\lfloor \frac{1}{C_0} \right\rfloor \right\}.$$

This theorem relates the trade-off parameter C_0 in the SLIM objective to the generalization of SLIM scoring systems. It indicates that increasing the value of the C_0 parameter will produce a model with better generalization properties.

In Theorem 3, we will show an alternative generalization bound by exploiting the fact that SLIM scoring systems use co-prime integer coefficients. In particular, we express the generalization bound from Theorem 1 using the d-dimensional Farey points of level Λ (see 36, for a definition).

Theorem 3 **(Generalization of Co-prime Discrete Linear Classifiers)**.

Let $\mathcal{F}$ denote the set of linear classifiers with co-prime integer coefficients bounded by Λ:

$$\mathcal{F} = \left\{ f : \mathcal{X} \to \mathcal{Y} \,\middle|\, f(\boldsymbol{x}) = \operatorname{sign}\left(\langle \boldsymbol{\lambda}, \boldsymbol{x} \rangle\right) \text{ and } \boldsymbol{\lambda} \in \mathcal{L} \right\},$$

$$\mathcal{L} = \left\{ \boldsymbol{\lambda} \in \hat{\mathbb{Z}}^d \,\middle|\, |\lambda_j| \leq \Lambda \text{ for } j = 1, \ldots, d \right\},$$

$$\hat{\mathbb{Z}}^d = \left\{ \boldsymbol{z} \in \mathbb{Z}^d \,\middle|\, \gcd(\boldsymbol{z}) = 1 \right\}.$$

For every $\delta > 0$, with probability at least $1-\delta$, every classifier $f \in \mathcal{F}$ obeys:

$$R^{\text{true}}(f) \leq R^{\text{emp}}(f) + \sqrt{\frac{\log(|\mathcal{C}_{d,\Lambda}|) - \log(\delta)}{2N}},$$

where $\mathcal{C}_{d,\Lambda}$ denotes the set of Farey points of level Λ:

$$\mathcal{C}_{d,\Lambda} = \left\{ \frac{\boldsymbol{\lambda}}{q} \in [0,1)^d : (\boldsymbol{\lambda}, q) \in \hat{\mathbb{Z}}^{d+1} \text{ and } 1 \leq q \leq \Lambda \right\}.$$

The proof of Theorem 3 involves a counting argument over co-prime integer vectors, using the definition of Farey points from number theory.

These bounds can be difficult to interpret directly. The importance of each bound is not in its exact values but in its form. As discussed above, these bounds are formalizations of the principle of Occam's razor, which is that an accurate yet simple explanation will generalize well to new situations. If the right hand side of the bound is small, we have a better probabilistic guarantee on out-of-sample error R^{true}.

There are two ingredients to making these bounds small: (i) We must choose a model with small in-sample error, meaning that R^{emp} is small. SLIM directly minimizes R^{emp}. (ii) The complexity term on the right hand side of each bound, called the *generalization error term*, should be small. In the generalization error term, the simplicity of the model class $\mathcal{F}$ is measured by its size. In Theorem 1, we directly used the size of the set $\mathcal{F}$ in the bound. In Theorem 2, we used the trade-off parameter C_0 to eliminate functions from $\mathcal{F}$ that are not sparse, thus these are functions that SLIM would never choose. Theorem 3 keeps only functions from $\mathcal{F}$ whose coefficients are co-prime, and SLIM chooses only classifiers with co-prime coefficients. All of these theorems are simultaneously valid, and tighter bounds could be produced by trivially combining ideas of Theorem 2 and Theorem 3 to count the set of sparse integer models with co-prime coefficients. Thus, according to these three theorems, by minimizing in-sample error while limiting our model class to linear models with integer co-prime coefficients, we have a theoretical foundation explaining why SLIM tends to have good out-of-sample performance.

Appendix: Back-of-the-Envelope Cost/Benefit Analysis for Automated Sleep Apnea Screening

Here we estimate the cost and benefit from introducing an automated screening test for sleep apnea on the U.S. population. We use the following values (not including uncertainty in our calculations, the fact that different populations are affected differently, etc.):

i There are approximately 25 million Americans with sleep apnea,[1] roughly 7% of 327 million total Americans. (The estimate of 7% varies from 4% to 9% in the literature.) Conservatively, 4% of Americans have untreated apnea, which is 13 million.

ii People with undiagnosed obstructive sleep apnea are at least 2.5 times more likely to have a car accident than usual. This result came from a review over 30 papers (17), and in all but one of the relevant 19 studies the the odds ratio was above 2, often much higher.

iii The cost of a definitive polysomnography sleep apnea test at MGH ranges from $300-$1800 not counting downstream costs. We use the value $1000.

iv The economic costs of car accidents in the U.S. in 2010 was $242 billion, according to the U.S. National Traffic Highway Safety Administration,[2] and including quality-of-life valuations, the cost is $836 billion. They report a total of 13,565,773 crashes that same year. This means each crash costs over $17.8K on average or $61.6K including quality-of-life valuations.

v There is an approximately 70% adoption rate for sleep apnea treatment, and it works fast, sometimes returning patients approximately to normal within a day.

vi We assume that of the people who are risk for obstructive sleep apnea, a subset of them chosen uniformly at random visits a clinic similar to that of MGH's Sleep Clinic.

[1] https://aasm.org/rising-prevalence-of-sleep-apnea-in-u-s-threatens-public-health/

[2] https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812013

Implementing our screening test is essentially free, as it is a simple calculation using existing medical records.

First, we determine the probability of a crash for an undiagnosed apnea patient, denoted $\Pr(\text{crash}|\text{apnea})$. From (i), the probability of having undiagnosed apnea is $\Pr(\text{apnea}) = 0.04$, thus $\Pr(\text{no apnea}) = 1 - 0.04$, including those being diagnosed and treated. Using the law of total probability and substituting the values above:

$$\begin{aligned}\Pr(\text{crash}) &= \Pr(\text{crash}|\text{apnea}) \times \Pr(\text{apnea}) + \Pr(\text{crash}|\text{no apnea}) \times \Pr(\text{no apnea})\\ \frac{13.6 \text{ million crashes}}{327 \text{ million people}} &= (2.5 \times \Pr(\text{crash}|\text{no apnea})) \times 0.04 + \Pr(\text{crash}|\text{no apnea}) \times (1-0.04).\end{aligned}$$

Solving this yields $\Pr(\text{crash}|\text{no apnea}) = 0.039$, and thus $\Pr(\text{crash}|\text{apnea}) = 2.5 \times 0.039 = 0.098$.

From here we can calculate the cost and benefit of using the automated screening test. The cost comes from false positives, which is when our screening test encourages people to get a polysomnography test that will ultimately come out negative. This calculation is what we will do next.

Cost Calculation of False Positive Screening. We first need to determine the number of people who are at risk for sleep apnea. We prove two facts in order to do this.

According to (i) and (vi) above, $9\frac{1}{3}\%$ of people are at risk for apnea. Further, $1\frac{1}{3}\%$ of the population is at risk but does not have apnea and does not go to a clinic. To show this, we use assumption (vi), which is that the at-risk population visits the clinic at random. As we will show, an assumption of $9\frac{1}{3}\%$ of people being at risk for apnea allows for all of the figures within (i) above to hold. Of those at risk, 4% of the population goes to the clinic, and of those, 3/4 of them (3% of the population) test positive (in agreement with (i)). This figure matches the proportions of true positive patients we get from the Massachusetts

General Hospital (MGH) Sleep Lab (in particular, 1478 of the 1922 MGH patients have OSA, which is approximately 3/4). That leaves $5\frac{1}{3}\%$ of the population who is at-risk and does not go to a clinic. Of those, 4% have undiagnosed OSA (in agreement with (i)), and $1\frac{1}{3}\%$ of them do not have apnea (but are at risk, in that they possess risk factors similar to those who do have apnea). The fraction of the at-risk no-clinic population with OSA is then $4/(4+1\frac{1}{3}) = 3/4$, which is identical to the ratio of true OSA patients in the MGH population. Thus, we have shown that our assumption of $9\frac{1}{3}\%$ of people being at risk for apnea agrees with (vi), (i), and MGH's observed population.

The number of false positives is 7.63 million × *FPR.* Let us show this. For the population that is not at risk, $100\% - 9\frac{1}{3}\% = 90\frac{2}{3}\%$, we assume they will not screen positive, since they are not at risk. Since the population visiting the clinic is chosen randomly from the at-risk population, the TPR and FPR of the test will be the same for those at-risk individuals not visiting the clinic. According to the previous paragraph, $1\frac{1}{3}\%$ of the population is at risk, does not go to the clinic, and does not have apnea. They could potentially screen positive as long as our screening test is launched using medical records – they do not need to go to the clinic to screen positive. With the additional 1% of the population who do not have apnea and attend the clinic, the screening test will find $\text{FPR} \times 2\frac{1}{3}\%$ of the population screening positive who do not actually have apnea. Thus, the number of false positives is approximately 0.0233 × 327 million × FPR, or 7.63 million × FPR false positive screens.

Conservatively assuming everyone who screens positive will get a polysomnography test,

$$\begin{aligned}\text{cost of false positives} &= 7.63 \text{ million } \times \text{ FPR } \times \text{ cost of test},\\ &= 7.63 \text{ million } \times \text{ FPR } \times \$1000.\end{aligned}$$

Benefit Calculation of Additional True Positives. The benefit comes from the reduction in car accidents within one year. Earlier we calculated the risks for accidents with and without undiagnosed apnea to be $\Pr(\text{crash}|\text{no apnea}) = 0.039$, and thus

$\Pr(\text{crash}|\text{apnea}) = 2.5 \times 0.039 = 0.098$. Thus, we can calculate the value of accidents reduced as a result of the screening test:

$$\begin{aligned}
&\text{Value of accidents reduced}\\
&\ = \text{reduction in risk for treated patients}\\
&\quad \times \text{number of patients treated as a result of the screening test}\\
&\quad \times \text{cost of each accident}\\
&\ = (0.098 - 0.039) \times \text{number of patients treated as a result of the screening test}\\
&\quad \times (\text{either \$17.8K or \$61.6K}).
\end{aligned}$$

The number of patients treated as a result of the screening test is computed as follows. The number of undiagnosed apnea patients who screen positive is: (13 million undiagnosed apnea patients) × TPR of the test. The TPR is precisely $\Pr(\text{screen positive}|\text{apnea})$. Of these people, 70% adopt treatment. The number treated as a result of the screening test is then 13 million × TPR × .7, so:

$$\begin{aligned}
&\text{Value of accidents reduced}\\
&\ = (0.098 - 0.039) \times 13 \text{ million } \times \text{TPR } \times .7 \times (\text{either \$17.8K or \$61.6K})\\
&\ = \text{ either \$9.56 billion} \times \text{TPR or 33.1 billion} \times \text{TPR}.
\end{aligned}$$

Estimated Cost and Benefit. For a model with 61.4% TPR and 20.9% FPR, which are the rates from the model in Figure 4, the cost is $1.53 billion and the benefit is either $5.87 billion (economic costs) or $20.3 billion (including quality-of-life valuations).

This estimate is only for the first year that the automated screening test is offered. In the following years, patients diagnosed with OSA who obtain treatment would also have fewer accidents, thus, the benefits would continue into the following years, without the cost of additional polysomnography testing. As discussed earlier, there are many other health and economic benefits for reducing conditions associated with sleep apnea that are not able to be quantified as easily.