Creating Risk–Scores in very imbalanced datasets – Predicting extremely violent crime among criminal offenders following release from prison

Markus Breitenbach  
*Northpointe Institute for Public Management, U.S.A.*

William Dieterich  
*Northpointe Institute for Public Management, U.S.A.*

Tim Brennan  
*Northpointe Institute for Public Management, U.S.A.*

Adrian Fan  
*University of Colorado at Boulder, U.S.A.*

1 INTRODUCTION AND BACKGROUND

In this chapter, we will discuss the many benefits of the Area under the Curve metric (AUC), not only as a performance measure, but also as a tool for optimizing models on very imbalanced datasets. We first introduce the measure formally and then discuss a few modeling techniques that can be used specifically for imbalanced datasets. Then we will include techniques that optimize the AUC directly. We will discuss how to choose suitable cut-points on an AUC optimized score and present a case study on predicting violent felony offenses (VFO) on a parole population.

1.1 Background

The use of predictive modeling has become pervasive as a decision support tool in criminal justice organizations in the USA. Even before the current shift to technical methods, prediction and classification tasks were central in criminal justice decision-making. Until the late 1970’s, most risk estimations regarding criminal offenders were made in an informal manner largely relying on the subjective or “expert” judgment of judges, clinical psychologists, parole boards etc. These decision-makers usually had access to substantial data on criminal histories and other relevant social and psychological data; however, they generally produced their decisions intuitively without the aid of any predictive model.

The last two decades have seen a dramatic shift by most national and state criminal justice agencies to incorporate more reliable, objective, data-driven, and formal predictive models. The motivation for this shift included: the desire for higher predictive accuracy, for more reliable and defensible procedures that could be justified and replicated, and, at the policy level, a desire to appropriately balance public safety with the competing goals of equity and protecting the rights of
prisoners (Gottfredson, 1987, Brennan, 1987). A further motivation was the consistent finding in research studies that statistical and numerical models could systematically outperform the accuracy of human “expert” decision-makers, e.g. judges, prosecutors, trained prison classification officers, and so on (Quinsey, Harris, Rice, and Cormier, 1998b, Grove and Meehl, 1996).

The focus on criminal violence is critical since public safety is among the major goals of correctional agencies. Additionally, the scope of the task of estimating the risk of criminal violence is enormous. In recent years, state prisons in the USA admitted over 600,000 new inmates each year, and almost the same number were released each year from secure facilities. Thus, approximately 1,600 released prisoners each day were arriving back to communities across the country (Petersilia, 2001). Making estimations of the risk of criminal and violent behavior is thus a continual challenge for correctional/forensic professional staff. Risk estimations are also needed at several decision points that may involve different decision-makers across criminal justice. For example, probation officers must estimate the risk of future violence when preparing pre-sentence reports for judges. Judges, in turn, face similar predictive questions in struggling with sentencing decisions, i.e., is the expected risk so high that an offender should be locked up as an incapacitative or public safety strategy. Thus, the sheer number and the demand for timely decisions can overwhelm parole boards that must make and then justify such decisions. The performance and efficiency of numerical decision support risk estimations is thus of considerable value and importance to criminal justice agencies.

1.2 The changing array of methods
In the beginning of the shift to numerical and statistical risk assessment, the most widely used methods in criminal justice were simple unweighted additive scales (Dawes, 1979). These methods were followed by regression models, including ordinary least squares (OLS), logistic regression, and survival analysis. More recently, several methods emerging from computer science, machine learning, and data mining have been used in criminal justice risk assessments. Modern data mining methods include: regression trees (CART), Random Forests (Breiman, 2001), Support Vector Machines (SVM) (Schölkopf and Smola, 2002, Vapnik, 1998), Artificial Neural Networks, Ensemble and Mixed Method models, and so forth (Zeng, 1999, Caulkins, Cohen, Gorr, and Wei, 1996, Monahan, Steadman, Silver, Appelbaum, Robbins, and Mulvey, 2001, Silver and Miller, 2002, Brennan, Breitenbach, and Dieterich, 2008a, Berk, Kriegler, and Baek, 2006). While this changing array of methods has aimed at improving predictive accuracy, there have been very few comparative studies of the relative predictive accuracies of these methods - particularly for predicting violent offenses (Caulkins et al., 1996, Berk, 2008).

1.3 What measure of predictive accuracy should be used?
Much attention in violence prediction research has focused on how best to evaluate the accuracy of predictive methods. A good comparative measure must be clearly independent of the base-rates and selection ratios across diverse studies. Prior evaluative work on the comparative accuracy of risk assessment methods is now seen as being dubious in value and often misleading because the coefficients used were not independent of base-rates and selection ratios across studies (Rice and Harris, 1995, Smith, 1996). The field of criminal justice requires direct and uncontaminated comparisons across predictive factors and overall predictive models.

A consensus has emerged that the receiver operating characteristic (ROC) procedure from signal detection theory (Swets, 1988) and its accuracy coefficient the Area under the Curve (AUC) is the most useful measure for comparing predictive accuracy of different predictive models and factors. This is especially true for datasets with low base-rates as they are common in criminology. The ROC procedure is now used extensively in designing actuarial devices to predict violence.
2 MEASURING PERFORMANCE IN IMBALANCED DATASETS

The most commonly used metric for assessing a model’s performance for binary classification is the error rate on a hold-out dataset. In the case of unbalanced datasets, the error rate can be fairly misleading, because without knowing the base-rate of the majority class the number is meaningless. If the smaller class is only three percent of the dataset, a simple classifier predicting the majority class all the time will still result in a 97% correct prediction - a number that would be fairly good if the classes were evenly distributed.

Imbalanced classes are commonly addressed by either re-sampling (over-sampling of the rare class) or by assigning a high cost for misclassifying cases of the rare class. Re-sampling can skew the prediction too much in favor of the rare class while assigning costs requires either experimentation or knowing those application dependant costs of misclassification beforehand and is not supported by every classification algorithm. In these situations, the user often has to make a decision as to where to set the threshold for classifying an item into one class or the other.

2.1 Receiver Operator Characteristic (ROC)

However, a better way of measuring performance in imbalanced datasets is the receiver operating characteristic (or simply ROC curve). The ROC is a graphical plot of the true versus false positives (TPR vs. FPR, or True Positive Rate versus False Positive Rate) for a binary (two class) classification system as its discrimination threshold is varied. It was originally used in signal detection theory.

As an example, consider a two-class prediction problem (binary classification), in which the outcomes are labeled either as positive (p) or negative (n) class. There are four possible outcomes from a binary classifier. If the outcome from a prediction and the actual value are p, then it is called a true positive (TP). A true negative (TN) has occurred when both the prediction outcome and the actual value are n. However, if the actual value is n while being predicted as p, then it is said to be a false positive (FP). Likewise a false negative (FN) is when the prediction outcome is n while the actual value is p.

To draw a ROC curve, only the True Positive Rate ($TPR = \frac{TP}{TP+FN}$) and False Positive Rate ($FPR = \frac{FP}{FP+TN}$) are needed. The True Positive Rate determines the performance on classifying positive instances correctly among all positive samples available during the test. The False Positive Rate, on the other hand, defines how many incorrect positive results occur among all negative samples available during the test. A ROC space is defined by FPR and TPR as x and y axes in a line-plot showing relative trade-offs between true positive (benefits) and false positive (costs or false alarms).

The best possible prediction method would yield a point in the upper left corner of the ROC plot, representing the fact that all true positives were found with no false positives and the line going into the upper right corner indicating perfect classification for all discrimination thresholds. A completely random guess would give a point along a diagonal line (the so-called line of no-discrimination) from the left bottom to the top right corners.

In practice, the ROC is used to determine thresholds for the classification; that is, the user can choose a cut-point somewhere on the curve for cases to be classified as positive or negative. This choice can be made with respect to the costs of misclassification in the application at hand.

2.2 Area under Curve

The ROC is a more suitable way to determine the cut-points for thresholds than doing so using error rates. However, comparing plots of different models to each other is still difficult. In order to summarize the performance expressed in this 2D-plot in a single number, the Area under Curve (AUC) is often reported as a summary of the performance. A perfect classifier would have an AUC of 1, and a random guess would have an AUC of 0.5. For risk scores that have a reasonable performance for practical applications, the AUC is usually above 0.7 (Quinsey et al., 1998b).
The AUC is a non-parametric statistic (free of distributional assumptions) and is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one (Fawcett, 2006). It can be shown that the area under the ROC curve is equivalent to the Mann-Whitney statistic, which tests for the median difference between scores obtained in the two groups if the groups are of continuous data. Let \( n^+ \) and \( n^- \) denote the number of positive and negative examples respectively and let \( f() \) be a function computing the risk score, then

\[
AUC(f) = \frac{\sum_{i=1}^{n^+} \sum_{j=1}^{n^-} 1_{f(x_i^+) > f(x_j^-)}}{n^+ n^-}
\]

where 1 is the indicator function. This implies that the score is independent of thresholds, i.e. \( f(x) > f(x') \Rightarrow f(x) + c > f(x') + c, \forall c \)

An important property is that ROC graphs measure the ability of producing good relative instance scores; i.e., an algorithm does not have to produce accurate confidence probability estimates for its prediction. The AUC encourages the classifier to separate the two classes with a relative accurate score; i.e., the probability that the scores for true positives are larger than true negatives increases and measures the general ability to discriminate between classes. This is especially useful if the classifier produces a continuous score instead of a discrete class assignment as it is the case for risk-scores.

This can sometimes lead to confusion. For example, a classifier can have an accuracy of only 85 percent, yet still have an AUC of one. This can happen if the classifier ranks all the true positives over the true negatives (perfect ranking implies an AUC of one), but the threshold is set such that some cases are not assigned to the correct class. Remember that the AUC is independent of thresholds.

The independence of the threshold leads to an insensitivity to class skew. Thus, for example, this makes performance results easier to interpret as the base-rate must not be known. Classifiers that optimize the AUC directly are independent of the a priori distribution of the classes and hence it is unnecessary to assign penalty costs to encourage a classifier to learn non-trivial models.

### 2.3 Evaluating risk scores

Risk scores that originate from binary classification are harder to evaluate since they express a risk of failing, not necessarily the complete certainty. For example, we expect on average people with higher risk scores to fail earlier and more often than people with lower risk scores. In order to evaluate risk scores, one can use survival models to estimate three measures that are useful for evaluating the predictive value of the risk scales: failure probabilities, hazard ratios, and the concordance index - the later being the area under the receiver operating characteristic curve (AUC) for survival models.

- **Failure Probability.** In typical survival data without competing events, the Kaplan-Meier statistic is used as an estimate of survival or failure probability (1 - KM) at different time points. However, in the context of competing risks, 1 - KM is neither proper nor interpretable as a measure of failure probabilities. Hence the cumulative incidence functions are calculated for this purpose. We use a specialized proportional hazards survival model for competing risks (Fine and Gray, 1999) to estimate and plot failure probabilities (cumulative incidence curves) for each type of failure within the levels of the risk scores. The goal is to compare the **probability of the event of interest** (e.g. arrest) within different levels of the risk scores. The generality of the results from competing risk
models are limited to populations with similar characteristics and similar patterns of competing events (Pintilie, 2007).

- **Hazard Ratio.** We use the Cox survival model to model the effect of the respective risk scores on the cause-specific hazard for the outcome. In this type of model, failures due to competing events are censored. The cause-specific hazards reflect the risk of the event of interest (e.g. arrest) as if the competing events did not exist. The goal is to compare the **hazard rates for the event of interest** within different levels of the risk scores. The results are valid for any population with similar characteristics regardless of the pattern of competing events (Pintilie, 2007).

- **Concordance Index.** The concordance index is defined as the probability that the predictor values and survival times for a pair of randomly selected cases are concordant. A pair is concordant if the case with the higher predictor value has a shorter survival time. For survival models, the concordance index is equivalent to the AUC. The calculation is based on the number of all possible pairs of non-missing observations for which survival time can be ordered and the proportion of relevant pairs for which the predictor and survival time are concordant (Harrell, Califf, Pryor, Lee, and Rosati, 1982).

Optimizing the AUC directly has a couple of benefits such as being intuitive, making performance results easier to interpret, and being independent of the a priori distribution of the classes. Even if the two classes are not completely separable, optimizing for the AUC will “encourage” the algorithm to find a solution that separates the cases as well as possible. Optimizing the AUC directly is also beneficial if no cost ratio for misclassifications is known a priori. Note that optimizing for the AUC still leaves the choice of the cut-point up to the user. We will describe a simple method to determine the cut-point by minimizing the probability of misclassification in Section 4. The AUC is a good method for evaluating prognostic models because the estimate is not influenced by the base-rate (proportion of the sample that fails). This characteristic makes it easier to compare results across studies.

However, the base-rate can affect the precision of the AUC. The AUC ranges from .50 to 1.00. An AUC of .50 indicates no relationship between the risk scale and the outcome. An AUC of 1.00 indicates that the risk scale predicts the outcome perfectly. The vast majority of published studies for predicting criminal behavior in which the validity of instruments such as the COMPAS Reentry (Brennan and Dieterich, 2008) have been tested report AUCs in the range of .65 to .75. Criminal justice researchers have suggested an AUC between .60 and .70 indicates moderate predictive utility, and an AUC of .70 and above indicates strong predictive utility (Quinsey et al., 1998b). The performance of any instrument will vary depending on the population (e.g., prison releases or probation cases), measurement error in the scale or outcome, and type of outcome (e.g., arrests versus violations or returns).

### 3 ALGORITHMS

A risk score is often represented as a weighted sum of independents, i.e. $f(x) = \sum_i w_i x_i = \langle w, x \rangle$. Note that this effectively describes a hyperplane through the origin that the new data is mapped onto. Unlike decision tree or association rule based learning, the hyperplanes are not necessarily parallel to one of the axis. This property makes the rules more difficult to interpret, but often provides more accurate models in practice. Expressing a model as a linearly weighted sum, however, can give insights into the importance of variables. In general, the larger the absolute value of a weight is, the more influence the variable has on the classification. While this does not allow for clear-cut rules such as “IF A AND B THEN C”, it gives strong indications for major factors of recidivism. For some types of models, algorithms have been developed that extract readable rules (Thrun, Tesauro, Touretzky, and Leen, 1995). As we will show in our case-study, many of the factors in our equations are well established
associations of violent crime (e.g. prior violent crime, young age, poverty, low education, anti-social personality, etc). The weights for each factor in the formula represent the strength of association with violent crime and are easily interpretable. They can give additional insights into possible preventive measures by suggesting intervention programs that can reduce the risk factors with the most weight.

In this section, we will discuss three algorithms that optimize the AUC directly and can be used to create a risk score from a labeled dataset (supervised learning). The first algorithm is a variation of the Support Vector Machine (SVM) family of algorithms. This family of learning algorithms has attracted attention for its robustness and good performance. Furthermore, the algorithms are easily generalized to others domains such as string matching for text classification or gene expressions for bio-informatics applications by the use of kernel functions.

The second algorithm is a simple gradient descent method that approximates the AUC step-function. Furthermore, the authors introduce a way to perform the gradient descent by a single linear scan over the data allowing this method to be applied to datasets that exceed the computer’s available RAM.

The third method is a neural network using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) methodology for training. This algorithm has a history of effective use in telecom churn analysis which contains highly imbalanced and often incomplete data. In general, neural networks are well known for their robustness amongst noisy inputs with variants of the BFGS methodology scaling well in both runtime and memory usage.

3.1 AUC optimizing support vector machines

Support vector machines (SVMs) are a family of related supervised learning methods used for binary classification and regression. They belong to a family of generalized classifiers learning models of the form \( f(x) = \sum \alpha_i K(x, v_i) + b \) where \( K() \) is a kernel function, \( \alpha \) are learned weights and \( v_i \) are the “support vectors” (SV). Kernel-functions behave like dot-products, but can be non-linear mappings and are used as “generalization devices” to build non-linear models (e.g. the non-linear Gaussian kernel). Sometimes they are represented with a mapping \( \phi \), which does not need to be explicitly known, that maps \( x \) to a higher-dimensional vector-space where the data becomes separable, i.e. \( K(x, y) = \langle \phi(x), \phi(y) \rangle \). The sign of the output of \( f(x) \) indicates to which class the new case is believed to belong. The set of support vectors are a subset of the training cases and is usually much smaller than the entire training set, i.e. a small number of examples is used to form the learned function (sparsity). A special property of SVMs is that they simultaneously minimize the empirical classification error and maximize the geometric margin; hence they are also known as maximum margin classifiers. Viewing the input data as two sets of vectors in a \( d \)-dimensional space, an SVM will construct a separating hyperplane, one which maximizes the “margin” between the two datasets. This hyperplane is represented by the weights \( \alpha_i \) and some examples from the training set. As an alternative representation, one can represent the function as \( f(x) = w^T x + b \) where \( w \) is \( d \)-dimensional vector of weights. This solution can be obtained from the dual of the optimization problem.

Support vector machines can be formulated as a quadratic programming problem (QP) for \( n \) training examples as follows:

\[
\text{minimize} \quad \frac{1}{2} \|w\|^2 + C \sum_i \xi_i \quad \text{s.t.} \quad c_i K(w, x_i) - b \geq 1 - \xi_i, 1 \leq i \leq n
\]
The slack-variables $\xi_i$ become larger than zero if the problem is not completely separable (e.g. mislabeled examples). The parameter $C$ specifies the tradeoff between growing slack-variables and a more complicated model and has to be determined by the user.

Support vector machines optimize for the empirical classification error and hence are very sensitive to very imbalanced classes. The “vanilla” SVM does not explicitly optimize the Area under Curve. For a more detailed overview over Support vector machines, see (Burges, 1998).

Brefeld and Scheffer (2005) introduce a support vector machine that optimizes a bound for the area under curve, i.e. by improving a lower bound the AUC is implicitly increased. The key idea is the observation that the AUC performance depends on the number of example pairs $(x_i^+, x_j^-)$ for all $i \in 1...n^+, j \in 1...n^-$ that satisfy $f(x_i^+) > f(x_j^-)$. This can be expressed equivalently as

$$\langle w, \phi(x_i^+) \rangle - \langle w, \phi(x_j^-) \rangle > 0$$

for a mapping $\phi$ (kernel) and a weight-vector $w$.

This QP problem can be turned into a convex “soft-margin” optimization problem. For $C > 0$ and $r \in \{1, 2\}$ one can solve the following quadratic programming problem:

$$\text{minimize} \quad \frac{1}{2} ||w||^2 + C \sum_i \xi_i, \quad \text{s.t.} \quad \langle w, \phi(x_i^+) \rangle - \langle w, \phi(x_j^-) \rangle \geq 1 - \xi_{ij} \quad (3)$$

One problem is that the execution time of this algorithm is prohibitive for larger datasets, because the number of constraints is quadratic in the number of examples. Given that solving a QP optimization problem is quadratic in the number of the constraints, we are looking at run-time of order $O(n^4)$. One way of speeding up the computation is to reduce the $n^+ n^-$ number of constraints to $n^+ n^-$ constraints by clustering the examples with k-Means. The pre-processing by clustering and QP-Solving are both quadratic in run-time, and reduce the overall run-time to $O(n^2)$ with a small loss in classification performance.

Joachims (2005) introduces a variation to a Quadratic Programming solver that limits the number of constraints being created. The method first reformulates the problem by treating it as an n-dimensional multi-output problem, i.e. each example is mapped to a binary output. This reformulation allows for measuring a sample-based loss (e.g. AUC) instead of an example-based loss (e.g. error rate). The reformulation results in only a single training example as a collection of all input-vectors with labels mapped to them. Since there is only one training example, there is only one slack-variable which directly bounds the AUC. Again, instead of optimizing AUC directly a bound on the AUC is improved with each step. This is the key difference with the previously discussed approach: instead of an example based approach which results in a quadratically increasing number of constraints the sample based approach learns a mapping for a multi-variate output and generally results in exponentially many constraints.

The exponentially many constraints seem like a bad trade-off, but allows for the use of sparse approximations. A variation of a quadratic programming solver is used which in each iteration introduces only the most violated constraint of the exponentially large set of all constraints. The run-time of finding the constraint is dominated by a sort operation to order examples by their current rank, i.e. $O(n \log n)$. It turns out that only relatively few constraints need to be added before convergence of the convex problems is reached. In practice, this is often limited (depending on the dataset) to under 100 constraints in our experience and is probably the fastest support vector based method for optimizing AUC directly.
3.2 Gradient descent

A more interesting approach would be to directly optimize the Area under Curve instead of improving upon a lower bound as the techniques in the previous section do. However, since equation (1) is not differentiable due to containing a step-function, it cannot be optimized for directly. A commonly used trick for approximating step-functions is to use a sigmoidal function that is fully differentiable (parameterized by $\beta$ determining the steepness of the slope) to approximate the function:

$$\text{SoftAUC}_\beta(f) = \sum_{i=1}^{n^+} \sum_{j=1}^{n^-} \frac{1}{n^+ n^-} \frac{1}{1+e^{-\beta(f(x_i^+)-f(x_j^-))}}$$

(4)

However, this approximation still requires full scans each time the weight-vector of a linear classifier is updated.

Calders and Jaroszewicz (2007) propose an approximation for both equations (1) and (4) by approximating the step-function $H()$ by a polynomial of degree $d$ of the form $\sum_{k=0}^{d} c_k x^k$ as follows:

$$H(f(x_i^+)-f(x_j^-)) \approx \sum_{k=0}^{d} c_k (f(x_i^+)-f(x_j^-))^k$$

(5)

using the Binomial theorem this results in

$$= \sum_{k=0}^{d} \sum_{l=0}^{k} c_k \binom{k}{l} (-1)^{k-l} f(x_i^+)^{k-l} f(x_j^-)^l$$

(6)

which can be used in equation (1):

$$AUC(f) \approx \frac{\sum_{k=0}^{d} \sum_{l=0}^{k} c_k \binom{k}{l} (-1)^{k-l} \left( \sum_{i}^{n} f(x_i^+) \right)^{k-l} \left( \sum_{j}^{n} f(x_j^-) \right)^l}{n^+ n^-}$$

(7)

Note that these sums can be computed in one scan and independently. The gradient can be similarly approximated by computing the partial derivatives for the polynomial approximations.

With both these approximations we can now update a weight-vector $w$ iteratively, i.e. $w \leftarrow w + \gamma g$ for a learning rate $\gamma$ determining the size of each step in the direction indicated by the derivative. One other outstanding feature of this algorithm is the ability to determine the learning rate automatically by computing the resulting AUC directly for various values of $\gamma$ without having to rescan the data. This can be done because the AUC only depends on the direction of the weight-vector, but not on the length. This allows for arbitrary re-scaling of the weight-vector $w$ since re-scaling only affects the magnitude of the scores. This property also makes it easier to find the ideal angle between the old weight vector and the new direction indicated by the gradient instead of finding the right learning rate. After the gradient has been computed, one scan over the database is needed to compute the output of the approximated step-functions. Further updates to the weight-vector in this iteration can be done without a re-scan of the data.
3.3 BFGS neural networks

Neural networks are a class of algorithms that, despite being discovered more than 30 years ago, still produce excellent results in classification tasks due to their robustness to noise within real-world datasets (Haykin, 1998).

In a neural network model, simple nodes consisting of a summation and transform function, are connected together to form a network. These in turn propagate a signal forward and adapt to a target through altering the strength of weights within nodes. Neural network models can be used to infer a function, including classification functions, by adapting to a given target and propagating error throughout the network. The most common technique is to retrace the flow of the network and propagate the error backwards from the expected target, altering weights in each node in a manner similar to gradient descent. This method is known as the back-propagation approach to neural network training.

Neural networks are a class of algorithms which also optimize Area under Curve through the use of a sigmoidal instead of a step function. First, we set the binary classification target in the range of 0 and 1. Given an error in the final node of a network \( \text{error} = (y_i - g(x))^2 \) where

\[
g(x) = \frac{1}{1 + \exp \left( \sum_i x_i w_i \right)}
\]

in any node, we perform gradient descent in order to optimize our sigmoidal output with respect to the AUC.

The performance of neural networks can be greatly augmented by pre-processing the data. For preparation of the data, many setup tricks such as standardization of inputs, using logarithms of inputs, and weighting examples are used to optimize our search space. Standardization and outlier removal are common sanitization procedures. Adding in the logarithms of inputs affords us an interaction between variables which can be learned through training. The logarithm equalities

\[
\log(a) - \log(b) = \log \left( \frac{a}{b} \right)
\]

can be used to compute ratios of inputs. For example, given two factors a and b within our input array, we find that during our calculation of \( \sum_i x_i w_j \) we, in fact, have the terms

\[
\log(a)w_{\log(a)} - \log(b)w_{\log(b)}
\]

or \( \log \left( \frac{a}{b} \right) w_{\log(a)} \) where \( w_{\log(a)} \) indicates the weight associated with this input.

The weighting of examples in the gradient descent rule is due to our earlier concern regarding imbalanced datasets. Underrepresented classes receive variable higher weights to tailor our model in regards to a specific classification. For example, if class A consists of 98% of the population and class B consists of 2% of the population, we may weight class B to contribute 10 times what class A contributes during neural network learning in order to prevent a classifier that assumes everything is class A, for a 2% error rate.

Modifying the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method (Fletcher, 1970, Broyden, 1970, Goldfarb, 1970, Shanno, 1970), we solve a binary classification task with competitive results. The BFGS method is a derivative of Newton’s method in optimization literature. The main advantages of this flavor of neural network are mostly logistical as the BFGS method has a low memory footprint and fast runtime, while maintaining a convergence very similar to Quasi-Newton methods.
4 FINDING CUT-POINTS ON AUC OPTIMIZED RISK SCORES

Where to set the threshold for classifying a case into one class or another is often dependent on the application and the costs associated with either of the two error types.

Picking the threshold can be done by training another classifier that optimizes for accuracy with respect to misclassification-costs on the scores produced by the AUC optimized classifier. For example, the C4.5 decision tree could be built on the score and the first split of the tree could be used as the threshold (Quinlan, 1986).

Another method would be to use deciles to group cases of a normative sample into bins. The score of new cases can then be assigned to a decile group and this organization allows for a much more fine-grained approach to managing risks than a strict yes/no decision. This approach is particularly popular in the criminal justice context as varying levels of supervision are available.

4.1 Minimizing the probability of misclassification

In the following, we will describe an approach that simply minimizes the probability of misclassification. This criterion has been successfully used for building optimization based classifiers. Minimizing the probability of misclassification has led to the family of the minimax probability machine classifiers with competitive results in practice (Lanckriet, Ghaoui, Bhattacharyya, and Jordan, 2002, Strohmann, Belitski, Grudic, and DeCoste, 2004). By choosing a cut-point that minimizes misclassification probability, we obtain a classifier that is optimal in maximizing the AUC and minimizing the probability of misclassification. The method is free of any distributional assumptions and a priori information of the initial class distribution.

Let \( F(w), w \in \mathbb{R}^d \), be the original hypothesis learned by an ROC-optimized classifier. Let \( \mu_x \) and \( \mu_y \) be the mean of the scores produced by \( F \) for the two classes \( X,Y \). Let \( \sigma_x \) and \( \sigma_y \) be the standard deviation of the score for the two classes \( \{x,y\} \). To find an optimal cut that minimizes the probability of misclassification for a decision hyperplane \( aF(w)=b \), one computes

\[
\alpha = 1 - \frac{\kappa^2}{1 + \kappa^2} \tag{12}
\]

The classifier is then \( \text{sign}(aF(w)−b) \). This minimizes the probability of misclassification \( \alpha \) for this hyper-plane classifier. Note that the quality of the solution depends on the estimates of the mean and variance for both of the classes. This can be a problem when one is dealing with
imbalanced problems with both extremely few cases, as well as a large variance in one of the classes. The variable \(\alpha\) gives an upper bound on the misclassification probability.

5 CONTEXTUAL ISSUES IN CRIMINAL JUSTICE REGARDING VIOLENT CRIME PREDICTION

In our work to develop risk estimation models for serious violent offenders, several contextual issues emerged that increased the difficulty of both implementing quantitative decision support procedures and in achieving higher levels of accuracy:

5.1 Outcome measures - Diversity and definitions of violent crimes

The first issue is that predictive modeling that addresses violent crime forces researchers to carefully define the outcome variable. A key issue is the specificity or generality of how the violent outcome criterion is defined and measured. For example, a very broad range of violent crimes is included in the USA Uniform Crime Code definitions, e.g. murder, voluntary manslaughter, forcible rape, robbery, aggravated assault, simple assault, burglary (of an occupied dwelling with a weapon), possession of dangerous weapons, sex offenses, extortion, arson, and kidnap. Choices must often be made in aggregating or separating these offenses. Secondly, violent offenses have ranges of seriousness, i.e. less serious misdemeanors and or serious felonies. Clearly, a broad or more inclusive definition - that may aggregate all of the above - will obviously be a composite of several forms of violence. Such broader variables are useful in increasing the base-rate for predictive modeling. However, the potential cost is that unrelated kinds of violence are mixed within a single common class. This may be hazardous if different causal processes with differing predictor variables are mixed in the class, e.g. felony drug violence may have a quite different causal process from domestic violence homicide.

Given that even within the felony classes crimes can vary in degree of seriousness, the use of a crude binary variable (i.e. violent/non-violent) may in some situations seriously oversimplify the dependent variable. Binary measures are especially problematic in cases where the degree of violence is of key importance (e.g. sexual offenses). Another problem with the outcome variable is that local criminal justice practices may introduce noise into its measure. For example, police may arrest a person on several charges, but court prosecutors then “plea-bargain” with defense lawyers and charge the offender only with one of the lesser crimes. This will erode the reliability and validity of the criterion variable (violent offending) and may weaken the ability to identify valid predictors and establish overall predictive accuracy.

5.2 The base-rate problem

Many prediction situations in criminal justice, particularly of serious violence, confront the complicated issue of low base-rates in the outcome behavior. In predictive modeling this increases the difficulty of finding levels of accuracy that are sufficiently high for practical use. In reviewing criminal justice prediction studies, Quinsey, Harris, Rice, and Cormier (1998a) concluded that for the outcome follow-up periods usually used in violence studies, the base-rates for the crimes of concern were usually fewer than 20% and often much lower. Also, for very serious violent crimes (murder, armed robbery, etc.) the base-rates are typically lower; i.e., prevalence of violent crime is inversely related to its seriousness. Prevalence will also be larger with longer follow-up periods. For example, when longer follow-up periods are used (e.g., 10 years) with specific high-risk samples (e.g., violent sex offenders) the base-rates can approach or exceed the 50% level (Quinsey et al., 1998a). Thus, a simple strategy to achieve higher base-rates is to use the broader category of “any person offense” which as noted earlier has the hazard of mixing different forms of violence with possibly different underlying causal processes.
5.3 Do our model assumptions match the complexity of the violent behaviors being modeled?

It is possible that criminal populations are heterogeneous in the causal processes generating their criminal violence (Lykken, 1995, Brennan, Breitenbach, and Dieterich, 2008b). This possibility will be a challenge to any “global” model (e.g., regression, classification tree, survival model) as to whether it appropriately matches this heterogeneity. For example, Lykken argues for a structural heterogeneity among diverse kinds of offenders and proposes several distinct pathways to crime, each with its own explanatory variables (Lykken, 1995). An implication is that any risk estimation study may require a preliminary dis-aggregation of the overall population of criminally violent offenders into homogeneous sub-populations and then appropriate models can be applied for each sub-population.

A related critique is that any global numerical model will be suspect if it ignores the individuality of specific offenders or selected groups. This challenge - often from defendants, defense lawyers and psychological counselors - is when a specific offender or specific offender category (e.g., women) does not “fit” the global model. Again, it is asserted that the violent behavior of a selected person or group is driven by different factors than those used in the model and that more specific models are required for these groups (Daly, 1992, Owen, 1998). This complex issue is not yet resolved. There are advocates on both sides, and recent research is addressing the issue of gender and racial differences in predictive models (Blanchette and Brown, 2006, Brennan et al., 2008b).

5.4 Limited technical knowledge of criminal justice decision-makers

A widespread problem when implementing technical decision support procedures is that many criminal justice decision-makers have very limited training in the technical issues related to risk estimation methods. They are, however, often comfortable with traditional “subjective” decision-making and may resent, mistrust, or feel threatened by technical approaches. They naturally prefer their subjective judgments since these represent their own personal “experiences”, hunches, assumptions, and personal biases (Hastie and Dawes, 2001). Unfortunately, these assumptions, hunches, and implicit mental models are mostly untested and often wrong. In addition, these decision makers, including judges, often cannot articulate the logic or rational basis for their decisions (Tata, 2002). Thus, a major challenge in successful implementation of decision support technologies is to achieve trust and cooperation among the users (Walton, 1989, Brennan, 1999).

5.5 Ethical issues and predictive technologies

A final contextual issue in criminal justice violence prediction is that although predictive methods come from the world of technology they will typically be used in people-processing justice organizations embedded in a world of values and politics. Thus technology, policy, and ethics become intertwined. This is particularly so in decisions involving the loss of freedom for offenders. The limitations of predictive methods, particularly in low base-rate events such as serious violence may result in ethical conflicts and challenging lawsuits against correctional agencies for apparent “errors” of prediction. Social and political pressures largely focus on two broad consequences of error: 1) the threat to public safety from false negative errors and 2) the loss of fairness and equity from false positive errors. A policy of applying longer sentences to offenders who have higher risk probabilities of future violence may raise ethical considerations and, in some cases, legal challenge. The political balance between false positives and false negatives typically requires intensive policy debate so that acceptable decision thresholds are implicit in policies. Increasingly, the different “costs” or “stakes” of false positive and false negative errors are now factored into the final balancing of these two errors (Quinsey et al., 1998a).
6 CASE STUDY

In the following, we will conduct a case study using criminal justice data of parolees, as well as build and evaluate a risk score predicting violent felony offenses (VFO). We will evaluate the methods we have introduced in Section 3 and compare against a few established machine learning methods as baseline that have shown overall good performance in empirical comparisons (Caruana and Niculescu-Mizil, 2006), but do not optimize the AUC directly. Furthermore, we will show using survival analysis that the risk score actually sufficiently differentiates between high, medium, and low risk groups.

6.1 Methods and data

The data for this study were collected through COMPAS Reentry (Northpointe, 2006) assessments conducted by parole staff with 874 soon-to-be-released inmates at 25 correctional facilities in an eastern state. The assessments were conducted between December 2005 and July 2006. The COMPAS Reentry assessment tool covers several risk- and need domains relevant for parole decision-making with inmates preparing for the transition from prison to community (see table 2).

We developed new risk models to predict violent felony offenses. To develop the models, we started with a set of candidate variables that included 13 COMPAS Reentry scales plus age, age at first arrest, and gender. We regard this approach of working with the established scales as conservative because there is less opportunity to capitalize on chance, as opposed to data mining at the item level.

6.2 Sample

A stratified cluster sampling methodology was used to select the study sample. The sampling frame consisted of soon-to-be-released inmates who had been incarcerated in prison for at least 12 months, clustered in prisons, stratified by parole cluster, security level (minimum, medium, maximum) and gender (male, female). At the first stage, we randomly selected facilities from each stratum of the sampling frame. This resulted in 24 prisons selected - four from each of the six parole clusters. In the second stage, field parole officers periodically visited the selected prisons and assessed soon-to-be-released inmates with the COMPAS Reentry. The parole officers who conducted the assessments attended a short training in data collection prior to the start of the pilot study. Selected inmates were actively consented. Approximately 10% of selected inmates declined to participate.

The estimation sample used in the outcomes analyses consists of the original pilot sample with completed assessments (n=866) plus 30 randomly selected cases from a sample of 200 assessments conducted at a short-term holding and training facility for parole violators. These data were originally collected for a supplemental study. A small subset is incorporated into our main sample. From the combined sample of 896 cases, 22 cases were excluded because they were not released onto parole within 180 days of the COMPAS Reentry assessment. The mean time from COMPAS Reentry assessment date to parole release date is 15 days (median = 4 days). An additional 67 cases were excluded due to missing data on either the General Recidivism Risk scale or the Violent Recidivism Risk scale. There are 800 cases remaining in the estimation sample.

The sample is 78.4% male. Note that women represent only 7% of the parolee population in the research jurisdiction; they were oversampled for the pilot study. The average age is 35.2 years (SD = 9.9) and ranges from 17.9 years to 83.3 years. The median age is 34.9. The median age in the parole population is 34 years. The ethnicity breakdown in the sample is 29.1% Caucasian, 53.7% African American, 15.9% Latino/a, and 1.3% other ethnic group. In the parolee population
the breakdown is 20% Caucasian, 52% African American, 26% Latino/a, and 2% other ethnic group. The mean age-at-first-arrest in the sample is 19.9 years (SD = 6.7); the mean number of prior arrests is 8.9 (SD = 13.6); and the mean number of felony convictions is 1.5 (SD = 1.8). Table 1 shows the types of original conviction offenses for the individuals in the sample. Two types of conviction offense predominate, legislative violent felony offenses [VFO] (33.2%) and drug offenses (35.9%). In the parolee population in the study jurisdiction, 36% are VFO and 39% are drug offenses. For 74% of the individuals in the sample, this is their first release onto parole for the current sentence.

<table>
<thead>
<tr>
<th>Original Conviction Offense</th>
<th>Frequency</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1 Violent</td>
<td>7</td>
<td>0.9</td>
</tr>
<tr>
<td>Legislative VFO</td>
<td>263</td>
<td>32.9</td>
</tr>
<tr>
<td>Other Coercive</td>
<td>58</td>
<td>7.2</td>
</tr>
<tr>
<td>Drug Offenses</td>
<td>286</td>
<td>35.8</td>
</tr>
<tr>
<td>Major Property</td>
<td>108</td>
<td>13.5</td>
</tr>
<tr>
<td>Other Felony</td>
<td>59</td>
<td>7.4</td>
</tr>
<tr>
<td>YO/JO</td>
<td>19</td>
<td>2.4</td>
</tr>
</tbody>
</table>

Table 1: Original Conviction Offense

6.3 Outcomes measures
We developed a model to predict violent felony arrests (VFO). For the binary models, the criterion is defined as the first occurrence of an arrest for a violent felony offense within 600 days of release from prison. The follow-up time is adjusted for competing events (parole discharge, death, and return to prison) as described in Section 6.5.

6.4 Predictive modeling strategy
To develop the model predicting VFOs, we started with a set of candidate variables that included 13 COMPAS Reentry scales plus age, age-at-first-arrest, gender, and a re-parole indicator. The re-parole indicator is coded 1 if the current release onto parole is a re-release on the same sentence after a return to prison for a parole violation or 0 if it is the first release onto parole on the current sentence. Table 2 shows the candidate variables along with their respective number of items and alpha reliability coefficients in the pilot data.

A step-wise selection approach with cross-validation was used to identify a subset of variables listed in table 6. It is well-known that variable selection procedures capitalize on chance and are prone to problems with over-fitting. We feel that problems with over-fitting are not as serious in our case because we limited the candidate pool to only a small number of variables, the majority of which are previously established scales. Additionally, we adjusted the AUC estimates to account for shrinkage in new data. Finally, we found similar results using several different approaches to variable selection and penalized estimation, including bootstrapped backward stepwise variable selection and the lasso method for variable selection in survival models.
Table 2: Candidate Variables Used in Model Development

<table>
<thead>
<tr>
<th>Variable</th>
<th>Items</th>
<th>Alpha</th>
</tr>
</thead>
<tbody>
<tr>
<td>Early Onset</td>
<td>4</td>
<td>0.73</td>
</tr>
<tr>
<td>Criminal Involvement</td>
<td>4</td>
<td>0.81</td>
</tr>
<tr>
<td>Violence History</td>
<td>6</td>
<td>0.69</td>
</tr>
<tr>
<td>Prison Misconduct</td>
<td>7</td>
<td>0.52</td>
</tr>
<tr>
<td>Substance Abuse</td>
<td>7</td>
<td>0.75</td>
</tr>
<tr>
<td>Housing Problems</td>
<td>4</td>
<td>0.64</td>
</tr>
<tr>
<td>Financial Problems</td>
<td>6</td>
<td>0.67</td>
</tr>
<tr>
<td>Vocational/Educational Assets</td>
<td>9</td>
<td>0.70</td>
</tr>
<tr>
<td>Family Support</td>
<td>5</td>
<td>0.77</td>
</tr>
<tr>
<td>Social Isolation</td>
<td>5</td>
<td>0.64</td>
</tr>
<tr>
<td>Purpose/Direction</td>
<td>7</td>
<td>0.61</td>
</tr>
<tr>
<td>Criminal Personality</td>
<td>20</td>
<td>0.84</td>
</tr>
<tr>
<td>Self-Efficacy</td>
<td>15</td>
<td>0.82</td>
</tr>
<tr>
<td>Re-paroled</td>
<td>1</td>
<td>0.00</td>
</tr>
<tr>
<td>Gender</td>
<td>1</td>
<td>0.00</td>
</tr>
<tr>
<td>Age-at-Assessment</td>
<td>1</td>
<td>0.00</td>
</tr>
<tr>
<td>Age-at-first-Arrest</td>
<td>1</td>
<td>0.00</td>
</tr>
</tbody>
</table>

6.4.1 ROC analysis

The tables below show AUCs for a variety of outcomes, including VFOs. Table 3 shows the Area under Curve based on the survival model estimates for each risk score and outcome. Eight out of thirteen evaluations had AUCs at 0.70 or above. Table 4 shows the results of using the risk scores calculated from the survival model estimates as predictors in logistic regression models. Twelve out of thirteen studies using logistic regression had AUCs above 0.70. These results were particularly encouraging.

In Table 5, we compare different algorithm capabilities in modeling our VFO data. The data was randomly split into training and test sets (90/10), a model was built on the training set, and then the ROC was measured on the test set. The random split was done on the entire dataset and not per class, i.e., some of the splits might by chance have far less of the VFO examples in it allowing for a more conservative estimate. Using a corrected two-tailed test the results of 100 test-iterations were averaged and found to be significant. We used the implementations of a few classical classification algorithms in WEKA (software version 3.5.7) (Garner, 1995). We will also compare the results to the algorithms described in Section 3. Algorithms were run with the default parameters from Weka, and no tweaking of parameters or settings was performed. The neural network in Weka performs an automated discovery of the number of neurons in the hidden layer, though. We set the tradeoff parameter $C$ in SVMperf to one (default).

We first try all the scales we have introduced earlier and note that, as expected, the classical algorithms are outperformed by the AUC optimizing algorithms. Some of the classical algorithms do not perform better than chance. To our surprise Random Forests (Breiman, 2001) performs fairly well selecting features in this task and outperforms even the AUC optimizing techniques.

Once we select a subset of the scales, the performance generally increases. ADTrees (Freund and Mason, 1999) also demonstrated a fairly good performance in both cases. We can see that many classical algorithms do not perform better than chance in this imbalanced setting, while the
AUC optimizing algorithms all result in the best performance we were able to obtain on these data. One surprise was that logistic regression and multilayer perceptron perform almost as well as the AUC optimizing methods.

<table>
<thead>
<tr>
<th>Outcome Risk Score</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arrest Risk Score</td>
<td>.72</td>
</tr>
<tr>
<td>VFO Risk Score</td>
<td>.77</td>
</tr>
<tr>
<td>Abscond Risk</td>
<td>.72</td>
</tr>
<tr>
<td>Noncompliance</td>
<td>.68</td>
</tr>
<tr>
<td>Prison Return</td>
<td>.68</td>
</tr>
</tbody>
</table>

*Table 3: Survival model estimates of the area under the receiver operating characteristic curve for each risk scale model and different outcomes.*

<table>
<thead>
<tr>
<th>Outcome Risk Score</th>
<th>AUC All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arrest Risk Score</td>
<td>.72</td>
</tr>
<tr>
<td>VFO Risk Score</td>
<td>.77</td>
</tr>
<tr>
<td>Abscond Risk Score</td>
<td>.72</td>
</tr>
<tr>
<td>Noncompliance</td>
<td>.75</td>
</tr>
<tr>
<td>Prison Return</td>
<td>.70</td>
</tr>
</tbody>
</table>

*Table 4: Logistic regression estimates of the area under the receiver operating characteristic curve for each risk scale model and different outcomes.*

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>AUC all features</th>
<th>AUC feature subset</th>
</tr>
</thead>
<tbody>
<tr>
<td>J48</td>
<td>0.50</td>
<td>0.52</td>
</tr>
<tr>
<td>SVM (SMO)</td>
<td>0.50</td>
<td>0.50</td>
</tr>
<tr>
<td>Random Forest</td>
<td>0.70</td>
<td>0.58</td>
</tr>
<tr>
<td>ADTree</td>
<td>0.60</td>
<td>0.65</td>
</tr>
<tr>
<td>Neural Network</td>
<td>0.58</td>
<td>0.74</td>
</tr>
<tr>
<td>Logistic</td>
<td>0.63</td>
<td>0.74</td>
</tr>
<tr>
<td>AUCOpt</td>
<td>0.66</td>
<td>0.76</td>
</tr>
<tr>
<td>Neural Network</td>
<td>0.63</td>
<td>0.74</td>
</tr>
<tr>
<td>(BFGS)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SVMperf</td>
<td>0.60</td>
<td>0.72</td>
</tr>
</tbody>
</table>

*Table 5: Comparison of different algorithms on the VFO data.*
6.4.2 VFO risk score

In this section, we present results of validity tests for the VFO risk score for predicting the first occurrence of a VFO arrest during the follow-up.

The components entering into the VFO Risk Score are shown in Table 6. None of the input variables were transformed. Only mean-centering was applied. The risk score requires 33 Reentry scale items.

<table>
<thead>
<tr>
<th>Items</th>
<th>Description of VFO Risk Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>miscon.c</td>
<td>Misconduct</td>
</tr>
<tr>
<td>hxviol.c</td>
<td>History of Violence</td>
</tr>
<tr>
<td>crimpers.c</td>
<td>Criminal Personality</td>
</tr>
<tr>
<td>age.c</td>
<td>Age at assessment</td>
</tr>
<tr>
<td>gender</td>
<td>Gender</td>
</tr>
</tbody>
</table>

Table 6: VFO Risk Score Items.

6.5 Survival models

We fit survival models to assess the effects of the COMPAS risk scales on parole outcomes. Survival models are appropriate for this data because we are interested in both the occurrence and timing of the parole outcomes. Our analysis focuses on VFO arrests as well as any arrest, absconding, noncompliance, and return to prison. We model return to prison and discharge from parole as competing events (see below). Survival time begins on the release from prison date. The risk set at this point contains all inmates in the estimation sample (valid N = 800). Survival time is measured in days from release date to the point of the first failure of interest, first competing event, or end of the follow-up, whichever occurs first. The failure point is determined by the warrant date associated with the failure of interest. For returns to prison the most proximate warrant date prior to the return date is used as the failure time point. Cases that do not fail by the failure of interest date or a competing event date during the follow-up are censored at the end of the study (December 31, 2007). Cases remain in the risk set and contribute information to the analysis until the point of failure of interest, occurrence of a competing event, or the end of follow-up, whichever occurs first.

Cases can fail by an event of interest, such as a new arrest, as well as the competing events return to prison or discharge from parole. These competing events alter the probability of observing the event of interest. For example, if a parolee is revoked and returned to prison, they cannot fail by the event of interest (e.g., new arrest), at least not while in prison for the competing reason. Even though a parolee could reoffend after discharge or after return and expiration of their time allocation, we only tracked cases while they were on parole. Thus, within our survival models discharge and return are terminal competing events.

Parole jail time is controlled for during the follow-up by removing cases from the risk set during these periods. The models only account for jail time that is associated with parole warrants where the jail begin- and end dates are known.
6.6 Description of events of interest during the follow-up

Table 7 shows the number of events observed in each failure-specific survival model. The number failed gives the number of first failures. Some cases had multiple arrests and warrants, so the actual total number of arrests and abscond warrants observed in the sample over the follow-up was much higher. Table 8 shows the reasons for removal from parole for the cases in the estimation sample.

Figure 1 shows a plot of the cumulative crude incidence function for return to prison in the presence of parole discharge. The jumps in the discharge curve reflect groups of inmates being discharged at one year and two years. Here parole discharge alters the probability of return to prison. Both discharge and return to prison compete with the other study outcomes any arrest, VFO, absconding, and noncompliance.

<table>
<thead>
<tr>
<th>Event</th>
<th>Censored</th>
<th>Failed</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abscond</td>
<td>600</td>
<td>200</td>
<td>800</td>
</tr>
<tr>
<td>Arrest</td>
<td>496</td>
<td>304</td>
<td>800</td>
</tr>
<tr>
<td>Return</td>
<td>539</td>
<td>261</td>
<td>800</td>
</tr>
<tr>
<td>VFO</td>
<td>743</td>
<td>57</td>
<td>800</td>
</tr>
<tr>
<td>Noncompliance</td>
<td>358</td>
<td>440</td>
<td>798</td>
</tr>
</tbody>
</table>

Table 7: Distribution of Events for Each Failure-Specific Model.

Figure 1: Estimates of the probability of return to prison in the presence of parole discharge, based on cumulative incidence functions.
### Table 8: Distribution of Reasons for Removal from Parole Supervision.

<table>
<thead>
<tr>
<th>Reason</th>
<th>Woman</th>
<th>Men</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Expiration</td>
<td>40</td>
<td>81</td>
<td>121</td>
</tr>
<tr>
<td>Board Action</td>
<td>0</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Revoked-PV</td>
<td>48</td>
<td>213</td>
<td>261</td>
</tr>
<tr>
<td>Death</td>
<td>3</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>Merit Termination</td>
<td>9</td>
<td>13</td>
<td>22</td>
</tr>
<tr>
<td>Mandatory Termination</td>
<td>2</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>CDME</td>
<td>1</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>Revoked-PVNT</td>
<td>2</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>New Felony Conviction</td>
<td>4</td>
<td>26</td>
<td>30</td>
</tr>
<tr>
<td>Sum</td>
<td>109</td>
<td>363</td>
<td>472</td>
</tr>
</tbody>
</table>

6.6.1 Failure probability

Table 9 shows the distribution of cases across the deciles of the VFO risk score, including the proportion of cases that fail and the crude cumulative incidence of failure. The crude cumulative incidence of VFO arrest in the first, fifth, and tenth deciles of the VFO risk score are 0.01, 0.03, and 0.16, respectively. With the cuts at deciles D5 and D9, there are 156 (20%) inmates classified as high-risk on this scale. There is a consistent trend of increasing probability of a VFO arrest across the deciles. With the cuts at D5 and D9, there is a sevenfold difference in the failure probability between the high-risk (.14) and low-risk (.02) levels.

<table>
<thead>
<tr>
<th>Decile</th>
<th>Cases in Level</th>
<th>Failing</th>
<th>Incidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>79</td>
<td>1</td>
<td>0.01</td>
</tr>
<tr>
<td>D2</td>
<td>79</td>
<td>3</td>
<td>0.04</td>
</tr>
<tr>
<td>D3</td>
<td>79</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>D4</td>
<td>78</td>
<td>1</td>
<td>0.01</td>
</tr>
<tr>
<td>D5</td>
<td>79</td>
<td>2</td>
<td>0.03</td>
</tr>
<tr>
<td>D6</td>
<td>79</td>
<td>8</td>
<td>0.10</td>
</tr>
<tr>
<td>D7</td>
<td>78</td>
<td>3</td>
<td>0.04</td>
</tr>
<tr>
<td>D8</td>
<td>79</td>
<td>10</td>
<td>0.13</td>
</tr>
<tr>
<td>D9</td>
<td>79</td>
<td>17</td>
<td>0.22</td>
</tr>
<tr>
<td>D10</td>
<td>77</td>
<td>12</td>
<td>0.16</td>
</tr>
</tbody>
</table>

Table 9: Distribution of VFO risk score deciles and crude cumulative incidence of a VFO within each decile.

Figure 2 shows a plot of the fitted failure probabilities for the levels of the VFO risk score from the competing risks model for VFO arrest. The plot indicates good separation of the fitted probability curves across the low, medium, and high levels of the scale. There is about a
sevenfold increase in the failure probability of the high-risk group compared to the low-risk group.

Figure 2: Predicted cumulative incidence of first VFO arrest within the levels of the VFO risk score.

### 6.6.2 Hazard ratios

Table 10 shows the results from the Cox proportional hazards regression on the cause-specific hazards. The point estimate for the hazard ratio of the high-risk category relative to the low-risk category is 13.27, which indicates inmates classified as high-risk on the VFO risk score have a hazard for VFO arrest that is 13 times higher than the hazard for inmates classified as low-risk. The *p*-value is less than .001, indicating the effect is significantly different than zero. The confidence interval suggests the hazard ratio could be as low as 5.8 or as high as 30.35. The hazard for the medium category relative to the low category is 5.45 (*p*-value <.001). This indicates that inmates classified as medium-risk have a failure hazard that is 5 times higher than the hazard for inmates classified as low-risk.

We also set up contrast comparisons with the medium-risk group as the reference category to test if the hazard for the high-risk category was different than the hazard for the medium-risk category. The results indicate that the hazard for inmates classified as high-risk is 2.44 times higher relative to the hazard for inmates classified as medium-risk (*p*-value <.002). This result is consistent with the plot of the fitted failure probabilities above.

<table>
<thead>
<tr>
<th>Risk Level</th>
<th>Coeff.</th>
<th>SE</th>
<th>p-value</th>
<th>Hazard Ratio</th>
<th>95% CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Medium</td>
<td>1.695</td>
<td>0.436</td>
<td>0</td>
<td>5.45</td>
<td>[2.32,12.82]</td>
</tr>
<tr>
<td>High</td>
<td>2.586</td>
<td>0.422</td>
<td>0</td>
<td>13.27</td>
<td>[5.80,30.35]</td>
</tr>
</tbody>
</table>

Table 10: Cox proportional hazards model regressing the hazard for VFO arrest on the levels of the VFO risk score: Estimated cause-specific coefficients, standard errors, *p*-values, and hazard ratios with 95% confidence intervals. Note: The reference category for the test of medium- and high-risk categories is the low-risk category.
6.7 Comparison to other instruments

The AUCs of the other main instruments often used for offender risk prediction may help to contextualize the findings in our study. Perhaps the best known instruments are the Violence Risk Appraisal Guide [VRAG] (Quinsey et al., 1998b); the Level of Services Inventory-Revised [LSI-R] (Andrews and Bonta, 1995); and the Psychopathy Checklist-Revised [PCL-R] (Hare, 1991). The AUC values for these instruments in recent studies are quite varied according to the specific populations, outcome periods, and dependent variables used in specific studies.

- **VRAG**: Quinsey et al. (1998b) found an AUC of 0.76 in a large scale, multiyear recidivism study. Barbaree, Seto, Langton, and Peacock (2001) reported AUCs of 0.69 in predicting serious re-offending and 0.77 when predicting any re-offense for sex offenders. Kroner, Stadtland, Eidt, and Nedopil (2007) obtained an AUC of 0.703 in a study of re-offending among mentally ill offenders.

- **LSI-R**: The recent review by Andrews, Bonta, and Wormith (2006) did not provide AUCs for the LSI-R. However, Barnoski and Aos (2003) found AUCs of 0.64-0.66 for the LSI-R in predicting felony and violent recidivism among Washington State prisoners. Flores, Lowenkamp, Smith, and Latessa (2006) found an AUC of 0.689 using the LSI-R to predict reincarceration among federal probationers. Dahle (2006) reported an AUC of 0.65 using the LSI-R to predict violent recidivism. Barnoski (2006) reported an AUC of 0.65 using the LSI-R to predict felony sex recidivism.

- **PCL-R**: AUC levels again varied across studies. For example, in a Swedish study of mentally ill violent offenders, Grann, Belfrage, and Tengstrom (2000) found AUC levels of 0.64-0.75 based on various follow-up timeframes. Barbaree et al. (2001) reported AUCs of 0.61, 0.65, and 0.71 for the PCL-R in predicting various recidivism outcomes among sex offenders. The above findings clearly do not exhaust the full range of studies in this area. As more studies report AUCs for specific instruments, varying populations, outcome variables, and timeframes, it may become possible to identify which instruments perform well in these varying conditions.

7 CONCLUSION

In this chapter, we explored AUC as an error-metric suitable for imbalanced data, argued for its effectiveness, and surveyed methods of optimizing this metric directly. We introduced three techniques that scale to large datasets and are based on different paradigms of machine learning: support vector machines, neural networks, and a simple gradient descent-based method. We also address the issue of cutting-point thresholds for practical decision-making.

We performed a case study that examines predictive rule development and validation procedures for establishing risk levels for violent felony crimes committed when criminal offenders are released from prison in the USA. We were estimating the risk of “violent felony” recidivism as the key outcome since these crimes are a major public safety concern, have a low base-rate (around 7 percent), and represent the most extreme forms of violence.

We compared the performance of different algorithms on the dataset, including classical methods like decision trees and found that the AUC-based methods perform much better in the presence of more (potentially irrelevant) features than other methods. Random Forests, to our surprise, outperformed every algorithm in this category. Once we selected a more relevant subset of features, we obtained a better result with the algorithms, but also noticed that some classical techniques like logistic regression performed almost equally well to the AUC optimizing techniques.

By using survival analysis we showed that the risk scores produced by these techniques are computing estimates that are consistent with the recidivism observed in the different low-, medium-, and high-risk groups.
REFERENCES


