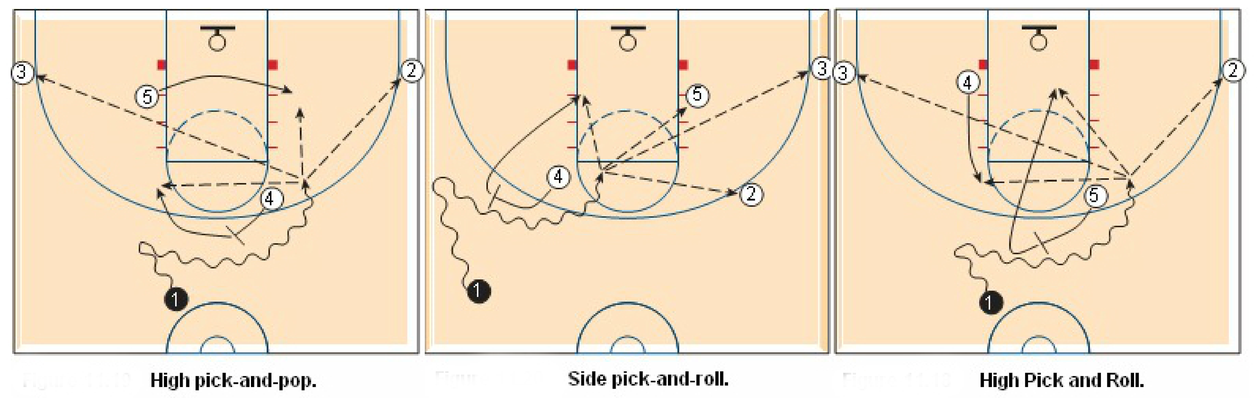
Automatic Classification of Pick-and-Roll plays in the NBA

Will Qiu

Introduction

The pick-and-roll is a basic and fundamental type of play frequently employed teams in the NBA. Knowing both how to execute as well as *stop* the execution of a pick-and-roll is an important skill that players and coaches must develop. This has led teams to develop a wide variety of pick-and-roll type of plays to use during game-time, and many teams have been come to known for the their particular style of play. Recognizing when a pick-and-roll is about to occur, its general type is vital for a defensive team's success in stopping its execution in a timely manner. More broadly, knowing *which kinds* of pick and roll tend to be effective is also important for teams as they learn to utilize their players on the court. Traditionally, this is a task relegated to the coaching staff and involves spending a tedious amount of time both in tape-watching sessions and scrimmages. Thus, a method which can take the large volumes of available game data and recognize the specific play-making strategies of teams in the league will help coaches better utilize their time when developing new styles of play.

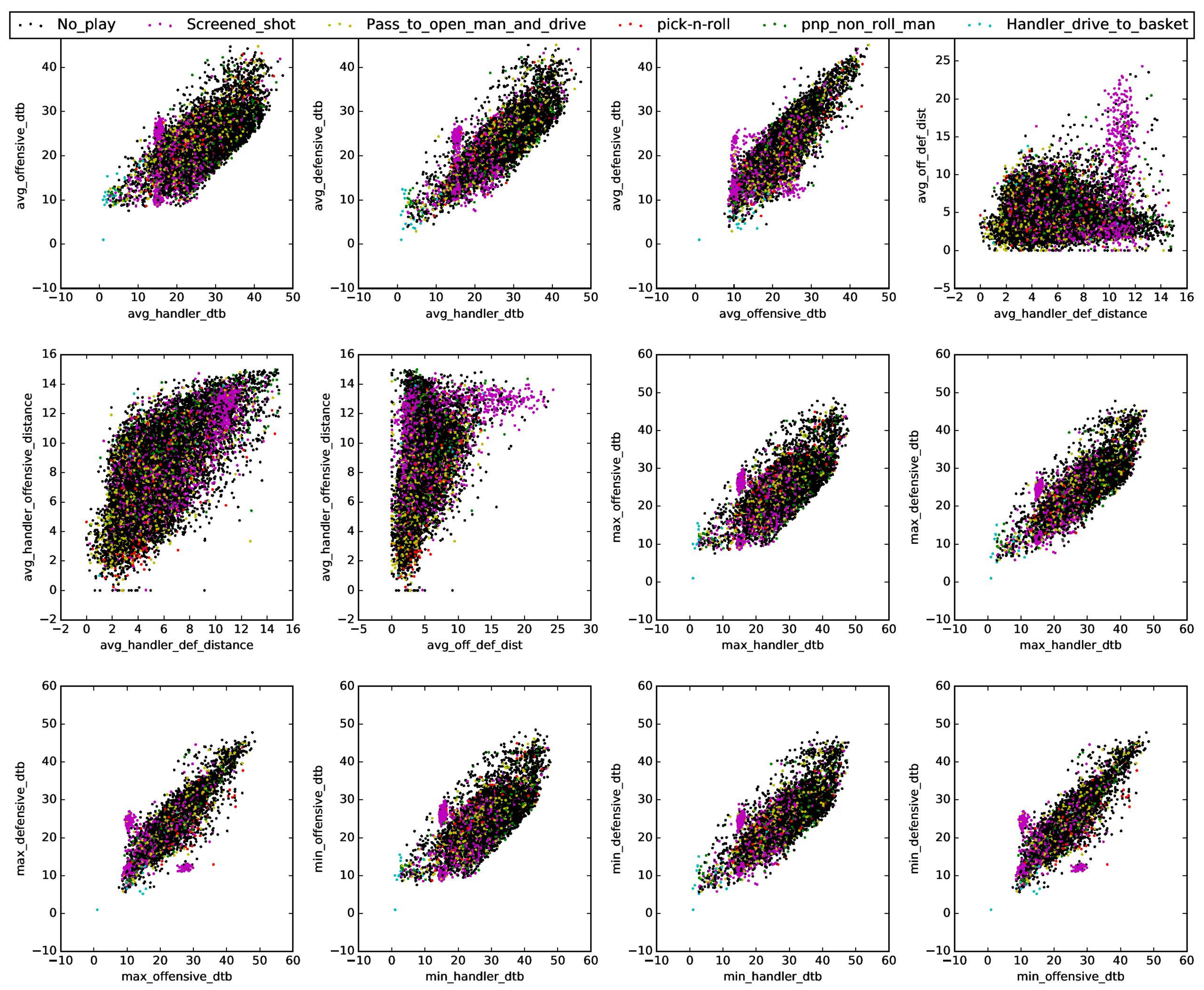


(coachesclipboard.net)

Data and Features

NBA SportsVu data of player and ball movements from 630 regular season games from the 2015-2016 season in the form of (x,y) coordinates was used. Data is discretized into individual events and further discretized into single frames (moments) at 25 frames per second. Relevant "pick-and-roll-able" moments and their features were extracted. Data were then labeled into several different categories for **prediction:** {no play, screened shot, pick-and-pop, classic pick-and-roll, pass to an open man}. I attempted both binary classification: {pick-and-roll, not pick-and-roll} as well as multi-label classification.

Feature	Representation
Moment Length (normalized to 1)	$m_{end} - m_{start}$
Time of ball-screen	$t_s = \arg\min_t d_t(p_d, p_s)$[1]
Ball handler's starting distance to basket	$d_0(p_h, b)$
Speed of player i	$S_i = \frac{\sum_t d_i}{t}$)
Min distance (all pairwise players and basket)	$min(\{d_1, d_2..., dt\})$
Max distance for (all pairwise players and basket)	$max(\{d_1, d_2..., dt\})$
Average distance (all pairwise players and basket)	$avg(\{d_1, d_2..., dt\})$
Total Distance Traveled by Player i	$sum(\{d_i(2-1), d_i(3-2)...d_i(t-t-1)\})$



Scatter Plot of Main Features (Handler, Screener and On Ball Defender)

Methods

I train and assess the performance of two classifiers using gradient boosting and SVM –specifically, two separate groups of models. The first group attempts simple binary classification of pick-and-roll plays. The second group attempts a multi-class classification task whereby different types of plays in addition to the classic pick-and-roll are predicted. The data is split 70-30 in order to compute validation errors. The training set contains $n_{training} = 134,165$ total unique plays and the test set contains $n_{test} = 4816$ total unique plays.

Gradient Boosting Classification

Gradient boosting is a boosting method that uses gradient descent to train a number of weak classifiers which are then combined to form a single more robust classifier. For the case of binary classification, gradient boosting is formulated in the following way:

Start with initial models for a given class i F_i, then, for each iteration until convergence:

Calculate negative gradient for i:

$$-g_i(x_i) = -\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}$$

Fit a regression tree h_i to $-g_i(x_i)$.

Compute F_t:

$$F_{t+1} := F_t + \rho h$$

Here, L is a loss function and $0 < \rho \leq 1$ is the learning rate. Here, L is chosen as the logistic loss function. To get at the best classifier, I experimented with different learning rates ρ and assess each one's performance.

Support Vector Machines

Support Vector Machines is a more traditional method which attempts to find the maximal separating hyper-plane among two sets of points. For classes where $y = \{-1, 1\}$, The typical SVM classifier is classified as the following:

$$h_{w,b} = g(\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x} + b)$$

Where:

$$\boldsymbol{w} = \sum_i h_i y_i K(\boldsymbol{x_i}, \boldsymbol{x})$$

$$b = \frac{1}{|SV|}\sum_{i \in SV}(y_i - \sum_j h_j y_j K(\boldsymbol{x_j}, \boldsymbol{x_i}))$$

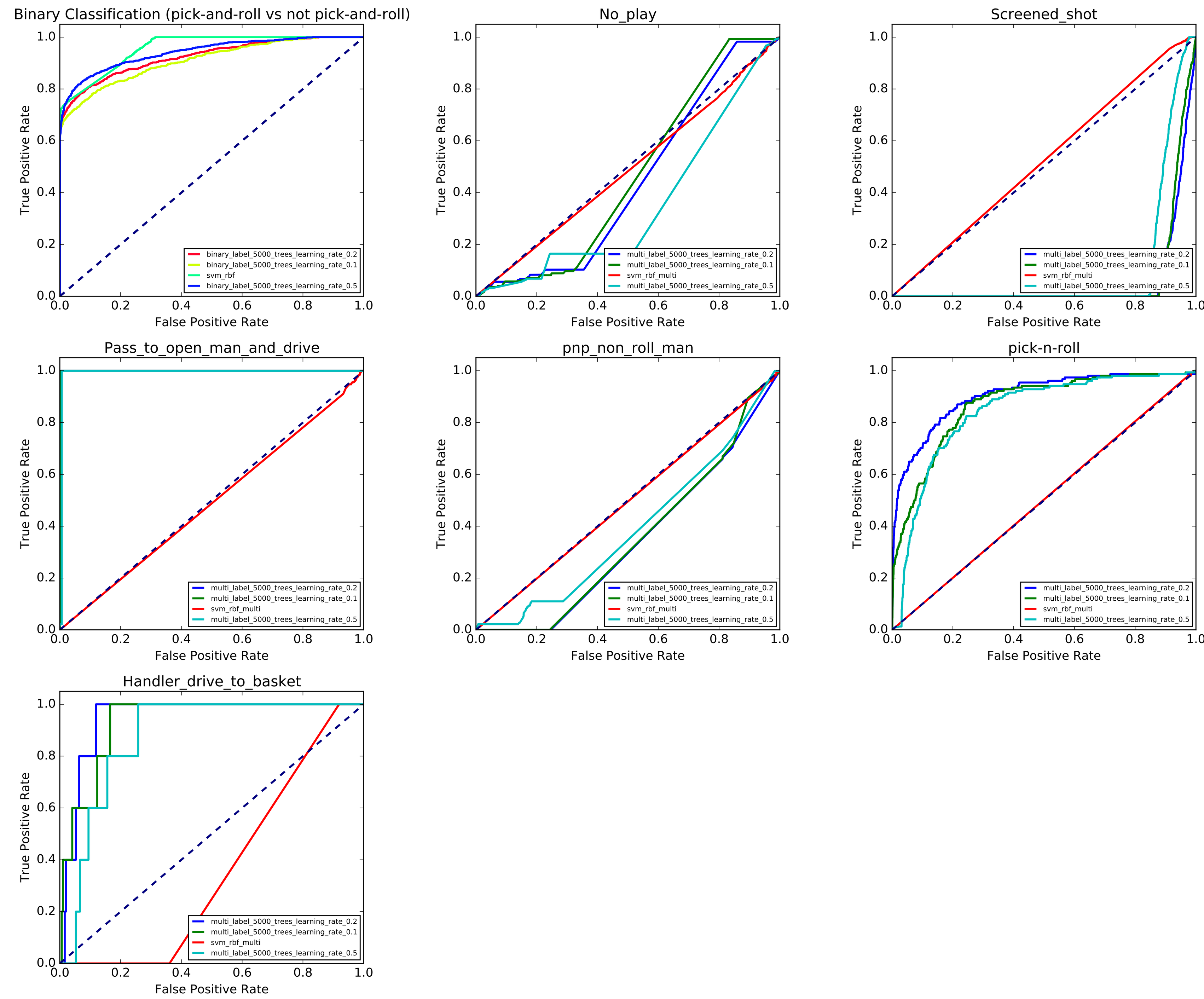
and K is the Kernel function. For the classification task here, the Radial Basis Function is used as the Kenel. The RBF Kernel is defined as:

$$K(\mathbf{x}, \mathbf{x'}) = \exp\left(-\frac{||\mathbf{x} - \mathbf{x'}||^2}{2\sigma^2}\right)$$

Results and Error Analysis

Model Training and Test Error (Training size: 134,165, Test size: 4,816)

Model	Training Accuracy	Test Accuracy
Binary SVM with RBF Kernel	0.9459	0.946
Multi-Class SVM with RBF Kernel	0.9918	0.695
Gradient Boosting (Binary) ($\rho = 0.1$)	0.937	0.930
Gradient Boosting (Binary) ($\rho = 0.2$)	0.941	0.933
Gradient Boosting (Binary) ($\rho = 0.5$)	0.946	0.940
Gradient Boosting (Multi-Class) ($\rho = 0.1$)	0.939	0.931
Gradient Boosting (Multi-Class) ($\rho = 0.2$)	0.947	0.941
Gradient Boosting (Multi-Class) ($\rho = 0.5$)	0.918	0.90



ROC Curves For Different Methods (Binary and Multi-Label Classifiers)

Discussion and Extension

All of the methods perform very well for binary classification and their differences are negligible. There is however greater variation in multi-label classification between methods depending on the specific label being predicted. Surprisingly, the multi-label SVM had terrible performance overall for multi-label classification against gradient boosting.

Gradient boosting is a much better method overall. But the high false positive rates for some of the labels suggest that the extracted features do not provide adequate predictive value for distinguishing between more nuanced types of plays.

One possible future extension is to take into account the sequential nature of the subject matter, and use as input, a sequence of values of player positions to make predictions about the next likely action or set of sequences.

References

[1] Armand McQueen, Jenna Wiens, and John Guttag. "Automatically recognizing on-ball screens". In: *2014 MIT Sloan Sports Analytics Conference*. 2014.