A Dynamic Modeling and Validation Framework for the Market Direction Prediction

Xugang Ye¹, Jingjing Li²

¹Johns Hopkins University, Baltimore, MD, USA
²University of Colorado, Boulder, CO, USA

ABSTRACT

There are many research papers talking about building various machine learning models to predict the market index. However, very few attention has been paid to effectively validating or calibrating the prediction results. The focus of this paper is to present a dynamic modeling and validation framework for the market direction prediction. The central idea is to calibrate the probabilistic prediction by estimating two conditional probabilities of correct forecast from the dynamic validation data set. The calibration method can be combined with any predictive model that generates probabilistic prediction of the market direction.

Keywords: market direction, machine learning, dynamic modeling, prediction, validation, calibration

INTRODUCTION

Modeling and forecasting market direction to facilitate trading strategies have been receiving more and more attention from academia and industry (Christoffersen, 2006; Granger, 1992; Tsay, 2010). Aiming at forecasting price movement — up or down — of general stock market, market direction prediction is also called “market timing” (Henriksson & Merton, 1981). Undoubtedly, this task is difficult because of the high market volatility. Many factors, such as political event, economic condition, trade mismatch, rumors, news and investors’ sentiment and mentality, all contribute to the high degree of uncertainty and fluctuation of the stock market (Harris, 2008).

Numerous efforts have been dedicated to find adequate modeling techniques to capture the market volatility. For example, the time series models, such as Autoregressive Conditional Heteroscedasticity (ARCH) (Engle, 1982) and generalized ARCH (GARCH) (Bollerslev, 1986), have been extensively used in economic and finance research. Those models, originated from the theories of the financial time series, often assume a stationary linear correlation structure (Tsay, 2010) among the time series data, which may not be able to capture the non-stationary nonlinear patterns and the impact of external events. On the contrary, machine learning approaches focus on finding patterns from data and usually make much less assumptions. Although various types of machine learning models including Artificial Neural Network (ANN) (Nicholas Refenes et al., 1994; Yoon & Swales, 1991; Zhang & Wu, 2009), Support Vector Machine (SVM) (Cao & Tay, 2003; Huang et al., 2005; Rao & Hong, 2010), Adaboosting (Rodríguez & Sosvilla-Rivero, 2006), and Hidden Markov Model (HMM) (Hassan & Nath, 2005; Rao & Hong, 2010) have been applied to the area of stock market forecasting, the machine learning methods became dominant only recently due to the fact that the technologies that enable people to generate and handle massive data are only recent stories.

Over the past few years, building machine learning models to predict financial markets has become more and more popular (Atsalakis & Valavanis 2009; Yoo et al., 2005). However, the existing market prediction literature shows that most work centers on static models, that is to build a model under a variety of parameter
settings from a training data set of a long time window, pick the best one according to its performance on the validation data set (as part of the training data set), and then test it on a hold-out data set of a relatively short time window for evaluation. Although this is a sound procedure for many machine learning applications, for market prediction, a problem is that there could be several market regime changes during a long time horizon so that a model trained and validated from the data collected from a long historical time window may not be able to catch the market dynamics. Another problem is that the models/parameters that are thrown away by the one-time validation will never participate the future recursive prediction tasks. Consequently, the market prediction is solely dependent on a narrow range of model settings, which are hardly adaptive to the future market changes. One fact that is often neglected is that for market prediction, a model that has very poor validation performance does not necessarily mean it should be discarded, one can use the opposite of the prediction. It actually follows that a really bad model is the one that has equal chance of predicting right or wrong. We say this kind of models do not generate “signal” and performs as same as random guess. Likewise, a model that performs very well on the validation set does not necessarily do as well in continual tests. One simple reason is that the market is driven by much more factors than the finite number of features that any model can accommodate. Tightly sticking to a model will likely lead to catastrophe sometime in the future (Harris, 2008).

Our remedy is to combine the rolling procedure (Tsay, 2010) that is widely used in time series analysis and a calibration step that is derived from the old results of Henriksson and Merton (1981). In short, we build machine learning models from the data collected from a rolling time window to predict the daily directional movement of the market. As the window slides forward, a validation set can be formed as long as there are enough observed targets. And when the window moves further, the validation set updates by absorbing a newly observed target and spitting out an old one. Dynamically, we calibrate the prediction generated by any machine learning model according to the validation performance. Depending on the validation results, we decide whether to stick to the prediction, use the opposite of the prediction, or follow the market — all of them could be reflected in the calibrated prediction.

The rest sections of this paper are organized as follows: we first introduce four machine learning models — logistic regression, linear regression, fisher’s linear discriminant analysis and SVM with soft margin — that are used in our work to predict the market direction. We then present the principal component analysis (PCA) method to perform the feature decorrelation and dimension reduction. Next, we derive our validation and calibration method. Specifically, we use the two conditional probabilities proposed by Henriksson and Merton (1981) to quantify the validation performance and calibrate the prediction. Subsequently, we discuss the ensemble method and how it is related to the conditional probabilities used in the calibration step. After introducing the evaluation metrics, serving as a proof-of-the-concept, we apply our approach to an S&P 500 index data from 2005 to 2011 to demonstrate the applicability and effectiveness of our method. Finally, we conclude our work by highlighting its strengths and limitations and pointing out the direction of future work.

**PREDICTION MODELS**

Suppose the security’s price at time $t$ is observed as $s_t > 0$ and we use the notation $r_{t+1}$ to denote the simple return from time $t$ to time $t + 1$. That is $r_{t+1} = \frac{s_{t+1}}{s_t} - 1$. We’re interested in predicting the probability $P(r_{t+1} > 0)$ based on the information collected up to time $t$. Precisely, let $\mathbf{x}^{(t)}$ be the $n$-dimensional feature vector obtained by time $t$, the goal of a machine learning model is to find a time-dependent function $F_t: \mathbb{R}^n \rightarrow (0,1)$ such that $P(r_{t+1} > 0) = F_t(\mathbf{x}^{(t)})$. Under the rolling procedure, an estimate $\hat{F}_t$ of $F_t$ can be obtained from the training data $D = \{(\mathbf{x}^{(\tau)}, y_{t-\tau}) : \tau = 1, \ldots, I\}$, where $y_{t-\tau}$ is the target label corresponding to $r_{t-\tau}$, the width of the rolling window. How $y_{t-\tau}$ is constructed from observed $r_{t-\tau}$ depends on what model to use. In our work, we use four simple models as briefly introduced as follows.

**Logistic Regression**

For a logistic regression model (Bishop, 2006; Hastie et al., 2001), $y_{t-\tau}$ is an indicator function of $r_{t-\tau}$, that is $y_{t-\tau} = I(r_{t-\tau} > 0)$. The model can be simply expressed as

$$ F_t(\mathbf{x}^{(t)}) = \frac{\exp (\alpha_t + \beta^{(t)} \mathbf{x}^{(t)})}{1 + \exp (\alpha_t + \beta^{(t)} \mathbf{x}^{(t)})} . $$

(1)

where $\alpha_t, \beta^{(t)}$ are the parameters that can be estimated by maximizing the log-likelihood function:
\[ L(\alpha_t, \hat{\beta}^{(t)}; D) = \sum_{t=1}^{T} \left[ y_{t-\tau} \ln F_t(\hat{x}^{(t-\tau)}) + (1 - y_{t-\tau}) \ln \left(1 - F_t(\hat{x}^{(t-\tau)})\right) \right]. \quad (2) \]

And "\(^\top\)" denotes the matrix/vector transpose. By logistic regression model, the prediction problem is treated as a simple two-class classification problem. Note that the estimates of the parameters also depend on the length of the rolling window \( l \), which is a parameter that we deal with in the validation step.

Linear Regression

While the logistic regression is a popular and easy-to-implement one, one possible downside is by transforming the observed simple return into class label, the magnitude of the return is lost. Therefore, a regression model is derived from the Bayes formula as

\[ \alpha_t + \hat{\beta}^{(t)} \hat{x} + \epsilon_t, \]

where \( \epsilon_t \sim N(0, \sigma^2) \) is the uncorrelated Gaussian error. Under the normal distribution assumption, we have

\[ F_t(\hat{x}^{(t)}) = \Phi\left( \frac{\alpha_t + \hat{\beta}^{(t)} \hat{x}}{\sigma_t} \right), \]

where \( \Phi \) is the standard normal distribution function. In the training data \( D \) for a regression model, \( y_{t-\tau} = r_{t-\tau} + \epsilon_t \) and the parameters \( \alpha_t, \hat{\beta}^{(t)} \) can be estimated by minimizing the sum-of-squared-errors cost function:

\[ c(\alpha_t, \hat{\beta}^{(t)}; D) = \sum_{t=1}^{T} (y_{t-\tau} - \alpha_t - \hat{\beta}^{(t)} \hat{x})^2. \]

And the parameter \( \sigma_t \) can be estimated from the residuals. Note that the estimates of the parameters depend on \( l \).

Linear Discriminant Analysis

Another simple method is called Fisher’s linear discriminant analysis method (Bishop, 2006; Hastie et al., 2001; Mika et al., 1999; Vapnik, 1998). For binary classification, as same as the logistic regression, \( y_{t-\tau} = I(r_{t-\tau} > 0) \). The idea is to project the feature vector \( \hat{x}^{(t)} \) onto a direction \( \hat{w}^{(t)} \) and assume \( z_t = \hat{w}^{(t)^\top} \hat{x}^{(t)} \) follows an one-dimensional Gaussian mixture with two components:

\[ z_t \sim \pi_t N(\hat{w}^{(t)^\top} \mu^{(1, t)}, \sigma^2_{1,t}) + (1 - \pi_t) N(\hat{w}^{(t)^\top} \mu^{(2, t)}, \sigma^2_{2,t}) \]

such that the prediction model is derived from the Bayes formula as

\[ F_t(\hat{x}^{(t)}) = \frac{\pi_f(z_t | y_{t-1}; \hat{w}^{(t)^\top} \mu^{(1, t)}, \sigma^2_{1,t})}{\pi_f(z_t | y_{t-1}; \hat{w}^{(t)^\top} \mu^{(1, t)}, \sigma^2_{1,t}) + (1 - \pi_t) \pi_f(z_t | y_{t-1}; \hat{w}^{(t)^\top} \mu^{(2, t)}, \sigma^2_{2,t})}, \]

where \( f(z_t | y_{t-1}; \hat{w}^{(t)^\top} \mu^{(1, t)}, \sigma^2_{1,t}) \) and \( f(z_t | y_{t-1}; \hat{w}^{(t)^\top} \mu^{(2, t)}, \sigma^2_{2,t}) \) respectively are the probability density functions of \( N(\hat{w}^{(t)^\top} \mu^{(1, t)}, \sigma^2_{1,t}) \) and \( N(\hat{w}^{(t)^\top} \mu^{(2, t)}, \sigma^2_{2,t}) \). In this model, \( \pi_t \) is the mixture parameter, \( \hat{\mu}^{(1, t)} \) is the mean of the class with label “1”, and \( \hat{\mu}^{(2, t)} \) is the mean of the class with label “0”. Respectively, they can be estimated as

\[ \hat{\pi}_t = \frac{1}{T} \sum_{t=1}^{T} I(y_{t-\tau}), \quad \hat{\mu}^{(1, t)} = \frac{\sum_{t=1}^{T} y_{t-\tau} \hat{x}^{(t-\tau)}}{\sum_{t=1}^{T} y_{t-\tau}}, \quad \text{and} \quad \hat{\mu}^{(2, t)} = \frac{\sum_{t=1}^{T}(1 - y_{t-\tau}) \hat{x}^{(t-\tau)}}{\sum_{t=1}^{T}(1 - y_{t-\tau})}. \]

The direction \( \hat{w}^{(t)} \) can be estimated as

\[ \hat{w}^{(t)} \propto S^{(t)} \hat{\mu}^{(2, t)} - \hat{\mu}^{(1, t)}, \]

where

\[ S^{(t)} = \frac{1}{\sum_{t=1}^{T} y_{t-\tau}} \left[ \left( \hat{x}^{(t-\tau)} - \hat{\mu}^{(1, t)} \right) \left( \hat{x}^{(t-\tau)} - \hat{\mu}^{(1, t)} \right)^\top + (1 - y_{t-\tau}) \left( \hat{x}^{(t-\tau)} - \hat{\mu}^{(2, t)} \right) \left( \hat{x}^{(t-\tau)} - \hat{\mu}^{(2, t)} \right)^\top \right] \]

is the total within-class covariance matrix. The parameters \( \sigma^2_{1,t} \) and \( \sigma^2_{2,t} \) can be respectively estimated as

\[ \hat{\sigma}^2_{1,t} = \frac{1}{\sum_{t=1}^{T} y_{t-\tau}} \sum_{t=1}^{T} (1 - y_{t-\tau}) \left( \hat{x}^{(t-\tau)} - \hat{\mu}^{(2, t)} \right)^2 \]

and

\[ \hat{\sigma}^2_{2,t} = \frac{1}{\sum_{t=1}^{T}(1 - y_{t-\tau})} \sum_{t=1}^{T}(1 - y_{t-\tau}) \left( \hat{x}^{(t-\tau)} - \hat{\mu}^{(1, t)} \right)^2. \]

Support Vector Machine

Another simple method that yields a prediction model similar to (7) is by support vector machine (Bishop, 2006; Hastie et al., 2001; Mika et al., 1999; Vapnik, 1998). For binary classification, \( y_{t-\tau} = I(r_{t-\tau} > 0) \). The basic idea is to find a hyperplane \( \hat{w}^{(t)^\top} \hat{x} + b = 0 \) to separate the two classes of the training data points. A separating plane can be found by solving the following quadratic programming problem

\[ \min \frac{1}{2} \sum_{t=1}^{T} \xi_t \]

Subject to

\[ 2(y_{t-\tau} - 0.5)(\hat{w}^{(t)^\top} \hat{x}^{(t-\tau)} + b) \geq 1 - \xi_t, \quad \tau = 1, ..., l; \]

\[ \sum_{t=1}^{T} y_{t-\tau} = m, \]

\[ \sum_{t=1}^{T}(1 - y_{t-\tau}) = n, \]

\[ 0 < \xi_t < C, \quad t = 1, ..., T. \]
\[ \xi_t \geq 0, \tau = 1, \ldots, l, \]
where \( C > 0 \) is the penalty parameter and \( \xi_t \)'s are slack variables.

Note that the quantity \( \eta_t = \bar{w}^{(t)} x^{(t)} + b \) determines whether the point \( x^{(t)} \) is located at one or the other side of the separating plane and how far it is away from the plane. Similar to the linear discriminant analysis, one can assume \( \eta_t \) follows an one-dimensional Gaussian mixture with two components:

\[
\eta_t \sim \pi_t \mathcal{N}(\bar{w}^{(t)} \mu^{(1,t)} + b, \sigma_x^{2,t}) + (1 - \pi_t) \mathcal{N}(\bar{w}^{(t)} \mu^{(2,t)} + b, \sigma_x^{2,t}),
\]
where each of the parameters \( \pi_t, \mu^{(1,t)}, \mu^{(2,t)}, \sigma_x^{2,t} \) has the same meaning and the estimation formula. And slightly changing (7) yields the prediction model as

\[
\hat{F}_t(x^{(t)}) = \pi^*_t f(\eta_t = 1; \bar{w}^{(t)} \mu^{(1,t)} + b, \sigma_x^{2,t}) + (1 - \pi^*_t f(\eta_t = 0; \bar{w}^{(t)} \mu^{(2,t)} + b, \sigma_x^{2,t})).
\]

In all the four models, the feature vector is first linearly mapped into a scalar and then nonlinealy mapped into the prediction. The first linear mapping ensures that the model complexity is basically at linear order of which respectively are the hit rates when the market is up and down. Since these two do not consider the effect of regularizing the model complexity.

### FEATURE PROCESSING

It's very common that the dimension of the feature vectors \( x^{(t)} \) is much larger than \( l \), the number of training data points. There could be a lot of redundant variables in the sense that they are highly correlated to each other. We apply the following steps for the de-correlation and dimension reduction within the rolling window from time \( t - l \) to time \( t \).

1. Remove the variables with identical values in the rolling window;
2. Scale the values of each remaining variable to \((0, 1)\) by subtracting min and dividing range;
3. Standardize each remaining variable by subtracting mean and then dividing by standard deviation, without losing generality, suppose the resulting feature vectors are \( \tilde{x}^{(t-l)}, \ldots, \tilde{x}^{(t-1)}, \tilde{x}^{(t)} \);
4. Compute the sample covariance matrix \( \tilde{S} = \frac{1}{t+1} \sum_{t-l}^{t} \tilde{x}^{(t)} \tilde{x}^{(t)\prime} = \frac{1}{t+1} XX' \), where \( X = [\tilde{x}^{(t-l)} \ldots \tilde{x}^{(t-1)} \tilde{x}^{(t)}]' \) is the feature matrix;
5. Compute the eigen values and eigen vectors of the sample covariance matrix;
6. Form the projection matrix \( V \) by picking \( k \) eigen vectors corresponding to the largest \( k \) eigen values;
7. Obtain the decorrelated feature vectors \( V' \tilde{x}^{(t)} \) for \( i = t \), \( \ldots, t \), that is the new feature matrix is \( XV \).

According to the notes in page 569 of Bishop (2006), \( \frac{1}{t+1} XX' \) and \( \frac{1}{t+1} X'X \) have the same positive eigenvalues and in the case of no degeneracy, if \( \tilde{u}^{(1)}, \ldots, \tilde{u}^{(k)} \) are the eigenvectors of \( \frac{1}{t+1} X'X \) respectively corresponding to the first \( k \) largest eigenvalues \( \lambda_1, \ldots, \lambda_k \), then \( \tilde{v}^{(1)} = \frac{1}{\sqrt{\lambda_1+1}} X' \tilde{u}^{(1)}, \ldots, \tilde{v}^{(k)} = \frac{1}{\sqrt{\lambda_k+1}} X' \tilde{u}^{(k)} \) are the corresponding eigenvectors of \( \frac{1}{t+1} X'X \). Obviously, \( k \leq l + 1 \), meaning there are no more than \( l + 1 \) principal components although there could be a very large number of feature variables. Also, properly choosing \( k \) has the effect of regularizing the model complexity.

### VALIDATION AND CALIBRATION

Given a prediction model, when the rolling window slides forward, a sequence of probabilistic predictions are generated as \( \tilde{y}_t = \hat{F}_t(x^{(t)}) \), \( \ldots, \tilde{y}_{t-1} = \hat{F}_{t-1}(x^{(t-1)}) \), where the feature vectors are supposed already decorrelated and dimensionally reduced. When the returns \( r_{t+l}, \ldots, r_{t-1} \) are realized, we can calculate two conditional accuracies:

\[
h_1^{(t+1)} = \frac{\sum_{t+l}^{t} \left(r_{t+l} > 0 \right) \tilde{y}_{t+l} + \sum_{t+l}^{t} \left(r_{t+l} < 0 \right) 
\]

\[
h_2^{(t+1)} = \frac{\sum_{t+l}^{t} \left(r_{t+l} > 0 \right) \left(1 - \tilde{y}_{t+l} \right) + \sum_{t+l}^{t} \left(r_{t+l} < 0 \right) \tilde{y}_{t+l}}{\sum_{t+l}^{t} \left(r_{t+l} \right)},
\]

which respectively are the hit rates when the market is up and down. Since these two do not consider the magnitudes of the returns, we also define the “weighted” version of the two hit rates as
By respectively substituting \( X \) under the same modeling method. Our ensemble problem is how to combine different models and the different parameter settings. The idea is to use \( H = h(X) \), where \( h(X) \) are the portions of sum of returns caught when the market is up and down.

To show how to use the two hit rates to calibrate the prediction \( \hat{y}_T = F_T(\varepsilon(T)) \) at time \( T \), we define an 0-1 binary random variable \( X_T \) to indicate whether the market from time \( T \) to \( T + 1 \) is predicted as up and consider computing \( P(X_T = 1) \) via conditioning as

\[
P(X_T = 1) = P(X_T = 1| r_{T:T+1} > 0) P(r_{T:T+1} > 0) + P(X_T = 1| r_{T:T+1} \leq 0) P(r_{T:T+1} \leq 0)
\]

\[
= p_1^T P(r_{T:T+1} > 0) + (1 - p_1^T)(1 - P(r_{T:T+1} > 0))
\]

\[
= (p_1^T + p_2^T - 1)P(r_{T:T+1} > 0) + 1 - p_2^T.
\]

(17)

where \( p_1^T = P(X_T = 1| r_{T:T+1} > 0) \) and \( p_2^T = P(X_T = 0| r_{T:T+1} \leq 0) \) respectively are the two conditional probabilities of predicting right under the up and down markets. When \( p_1^T + p_2^T \neq 1 \), the probability \( P(r_{T:T+1} > 0) \) can be expressed as

\[
P(r_{T:T+1} > 0) = \frac{P(X_T = 1) + p_2^T - 1}{p_1^T + p_2^T - 1}.
\]

(18)

By respectively substituting \( P(X_T = 1) \), \( p_1^T \), and \( p_2^T \) with \( \hat{y}_T \), \( \hat{p}_1^T \), and \( \hat{p}_2^T \), we have a calibrated prediction at time \( T \) as

\[
\hat{y}_T^c = \frac{\hat{y}_T + \hat{p}_1^T - 1}{\hat{p}_1^T + \hat{p}_2^T - 1}.
\]

(19)

where the estimates \( \hat{p}_1^T \) and \( \hat{p}_2^T \) can be calculated either by (13-14) or (15-16).

It’s worthwhile to understand \( p_1^T + p_2^T \). We may compare \( P(r_{T:T+1} > 0| X_T = 1) \) and \( P(r_{T:T+1} > 0) \). Note that by (17), we have

\[
P(r_{T:T+1} > 0| X_T = 1) - P(r_{T:T+1} > 0) = (p_1^T + p_2^T - 1)(1 - P(r_{T:T+1} > 0)).
\]

(20)

Hence as \( P(r_{T:T+1} > 0) < 1 \), \( P(r_{T:T+1} > 0| X_T = 1) \) ≥ \( P(r_{T:T+1} > 0) \) if and only if \( p_1^T + p_2^T \geq 1 \), with equality holds when \( p_1^T + p_2^T = 1 \). This means when \( p_1^T + p_2^T > 1 \), \( X_T = 1 \) is a “supporting” evidence of \( r_{T:T+1} > 0 \). When \( p_1^T + p_2^T < 1 \), \( X_T = 1 \) is opposite to the up market. When \( p_1^T + p_2^T = 1 \), we have

\[
P(r_{T:T+1} > 0| X_T = 1) = P(r_{T:T+1} > 0),
\]

which says the market result is independent of the prediction and therefore the prediction is random. Basically, the quantity \( p_1^T + p_2^T \) indicates whether the signal exists. \( p_1^T + p_2^T > 1 \) means the right predictions are more than the wrong predictions; while \( p_1^T + p_2^T < 1 \) means the opposite. Both cases indicate that signal exists. The main idea of calibrating the new prediction is to estimate \( p_1^T \) and \( p_2^T \) using the historical data in a sliding window, as (13-14) and (15-16) show. In plain words, signal exists if a predictor is more right than wrong or more wrong than right. It’s undesirable that a predictor has equal chance of being right or wrong.

**ENSEMBLE**

Ensemble (Bishop, 2006; Hastie et al., 2001) is a model averaging technique that combines the prediction results of multiple models, which are resulted from multiple methodologies or the various parameter settings under the same modeling method. Our ensemble problem is how to combine different \( \hat{y}_T^c \) at time \( T \) under both the different models and the different parameter settings. The idea is to use \( \hat{p}_1^T \) and \( \hat{p}_2^T \), the estimates of the two conditional accuracies, to determine how \( \hat{y}_T^c \) from one model under one parameter setting is combined into the final prediction result. Our combining formula is

\[
\hat{y}_T^c = \frac{1}{M} \sum_{j=1}^{M} \hat{p}_j
\]

where \( \rho_j = \langle \hat{y}_T^c \rangle \) if \( \langle \hat{p}_1^T \rangle + \langle \hat{p}_2^T \rangle - 1 > \delta_p \); \( \rho_j = \frac{1}{T-t} \sum_{t=0}^{T-t-1} y_{t+1} \) otherwise. That is for the \( j \)-th prediction \( \langle \hat{y}_T^c \rangle \), if the validation result shows non-randomness, then we take it into the model averaging; otherwise we replace it with the market’s up-prior in the validation time window from time \( t \) to time \( T - 1 \) for the model averaging. And \( \delta_p > 0 \) is a threshold.

In our setting, we apply the four models as logistic regression, linear regression, Fisher’s discriminant analysis, and the soft-margin SVM. We also vary the value of \( l \), which is the length of the rolling window of
the training data, from \( l_1 \) to \( l_2 (\geq l_1) \). This gives us totally \( 4(l_2 - l_1 + 1) \) results to ensemble.

**EVALUATION METRICS**

Suppose the final online predictions are \( \hat{y}_T^{\tau}, \ldots, \hat{y}_{T+l-1}^{\tau} \), when the returns \( r_{T:T+1}, \ldots, r_{T+l-1:T+l} \) are realized, we can evaluate the performance of the system from time \( T \) to time \( T + L - 1 \). We adopt the virtual trading performance measures used in Harris (2008). At time \( T + \tau \) for \( \tau = 0, \ldots, L - 1 \), we define the decision function of the prediction \( \hat{y}_{T+t}^{\tau} \) as

\[
d_{T+t} = \begin{cases} 
1 & \text{if } \hat{y}_{T+t}^{\tau} \geq 0.5 + \delta_y; \\
0 & \text{if } |\hat{y}_{T+t}^{\tau} - 0.5| < \delta_y; \\
-1 & \text{if } \hat{y}_{T+t}^{\tau} \leq 0.5 - \delta_y,
\end{cases}
\]

where “1” stands for “long”, “-1” stands for “short”, and “0” stands for “cash”. The decision cut-off is 0.5 and \( \delta_y \) is a threshold. In plain words, the decision is to long if the final prediction is greater than 0.5 + \( \delta_y \); or to short if the final prediction is less than 0.5 – \( \delta_y \); or to hold cash if the final prediction is close to 0.5. From the decision sequence \( d_{T}, \ldots, d_{T+l-1} \) and the realized return sequence \( r_{T:T+1}, \ldots, r_{T+l-1:T+l} \), we can calculate two common metrics as follows.

One is the realized hit rate, which can be expressed as

\[
H^{(T:T+L-1)} = \frac{\sum_{t=T}^{T+L-1} d_{t+T-t} r_{t+T-t+1} > 0 + 0.5 |d_{t+T-t} = 0|}{L},
\]

where the nominator is number times the decision coincides with the market direction. The second metric is the cumulative return, which can be expressed as

\[
r^{(\tau:T+L-1)} = \prod_{t=T}^{T+L-1} (1 + d_{t+T-t} r_{t+T-t+1} - \gamma_{t+T-t}) - 1.
\]

**EXPERIMENTAL RESULTS**

We present our results on a raw market data subscribed from Thomson Reuters. The data contains the published information of all S&P 500 stocks from the year 2005 to the year 2011. The basic information contains the prices, volumes, common outstanding shares, market caps, dividends, EPS estimates, EPS reports, revenues, net incomes, debts, cash flows, etc. We generated more than 1000 technical and fundamental feature variables on a daily base and formed the daily target by using the simple return as the function of the prediction. Calibrated prediction happens on the 181st day. We use the model selection threshold \( \hat{\delta}_p = 0.01 \) and the decision threshold \( \delta_y = 0.001 \). For the evaluation metrics, we calculated the realized semiannual hit rates and cumulative returns after transaction costs. We assume that \( \gamma_{T+t} = 0.001 |d_{t+T-t} - d_{T+t-1}| \), which means if a position is as same as the previous one, then there is no transaction cost. If a position is switched from long to short or from short to long, the transaction cost is 0.2%. If a position is switched from long or short to cash or from cash to long or short, the transaction cost is 0.1%. In our virtual trading experiments, we assume that the starting cash position is 10k dollars so that the assumption of no market impact could hold.

For convenience we let \( H-S&P500 \), \( H-Logistic \), \( H-LinReg \), \( H-FDA \), \( H-SVM \), \( H-4Modles \) respectively denote the S&P 500 index’s up-proportion and the realized hit rates of logistic regression, linear regression, Fisher’s discriminant analysis, SVM, and the four models combined, under the validation/calibration step that uses (13-14). We let \( H^w-S&P500 \), \( H^w-Logistic \), \( H^w-LinReg \), \( H^w-FDA \), \( H^w-SVM \), \( H^w-4Modles \) respectively denote the counterparts under the validation/calibration step that uses (15-16). Likewise, we use the notations \( r-S&P500 \), \( r-Logistic \), \( r-LinReg \), \( r-FDA \), \( r-SVM \), \( r-4Modles \) for the cumulative returns, with “\( r \)” corresponding to (13-14) and “\( r^w \)” corresponding to (15-16). Note that for
each individual model, we aggregate the 41 predictions corresponding to \( l = 50, \ldots, 90 \), using the ensemble formula (21) and for the four models combined, we aggregate all the 164 predictions, also using (21).

Table 1. The semiannual test performance comparison between the models and the market benchmark

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>S&amp;P500</td>
<td>47.89%</td>
<td>48.00%</td>
<td>50.00%</td>
<td>50.00%</td>
<td>63.28%</td>
<td>54.03%</td>
<td>60.16%</td>
<td>56.00%</td>
<td>51.55%</td>
</tr>
<tr>
<td>H-Logistic</td>
<td>57.75%</td>
<td>48.80%</td>
<td>58.20%</td>
<td>50.00%</td>
<td>63.28%</td>
<td>60.48%</td>
<td>60.16%</td>
<td>54.40%</td>
<td>44.85%</td>
</tr>
<tr>
<td>H-LinReg</td>
<td>57.75%</td>
<td>48.00%</td>
<td>58.20%</td>
<td>50.00%</td>
<td>63.28%</td>
<td>63.71%</td>
<td>60.94%</td>
<td>54.40%</td>
<td>45.88%</td>
</tr>
<tr>
<td>H-FDA</td>
<td>53.52%</td>
<td>50.40%</td>
<td>57.42%</td>
<td>51.61%</td>
<td>64.06%</td>
<td>62.10%</td>
<td>58.59%</td>
<td>54.40%</td>
<td>45.88%</td>
</tr>
<tr>
<td>H-SVM</td>
<td>56.34%</td>
<td>48.00%</td>
<td>58.20%</td>
<td>50.00%</td>
<td>62.50%</td>
<td>59.68%</td>
<td>59.38%</td>
<td>56.00%</td>
<td>50.00%</td>
</tr>
<tr>
<td>H-4Models</td>
<td>57.75%</td>
<td>48.00%</td>
<td>58.20%</td>
<td>50.00%</td>
<td>63.28%</td>
<td>62.90%</td>
<td>61.72%</td>
<td>54.40%</td>
<td>47.94%</td>
</tr>
<tr>
<td>H-Logistic</td>
<td>57.75%</td>
<td>48.80%</td>
<td>62.89%</td>
<td>51.61%</td>
<td>63.28%</td>
<td>60.48%</td>
<td>60.16%</td>
<td>54.40%</td>
<td>47.94%</td>
</tr>
<tr>
<td>H-LinReg</td>
<td>57.75%</td>
<td>48.00%</td>
<td>62.89%</td>
<td>51.61%</td>
<td>63.28%</td>
<td>63.71%</td>
<td>60.94%</td>
<td>54.40%</td>
<td>48.97%</td>
</tr>
<tr>
<td>H-FDA</td>
<td>53.52%</td>
<td>50.40%</td>
<td>62.11%</td>
<td>53.23%</td>
<td>64.06%</td>
<td>62.10%</td>
<td>58.59%</td>
<td>54.40%</td>
<td>48.97%</td>
</tr>
<tr>
<td>H-SVM</td>
<td>56.34%</td>
<td>48.00%</td>
<td>62.89%</td>
<td>52.42%</td>
<td>62.50%</td>
<td>59.68%</td>
<td>59.38%</td>
<td>56.00%</td>
<td>53.09%</td>
</tr>
<tr>
<td>H-4Models</td>
<td>57.75%</td>
<td>48.00%</td>
<td>62.89%</td>
<td>51.61%</td>
<td>63.28%</td>
<td>62.90%</td>
<td>61.72%</td>
<td>54.40%</td>
<td>51.03%</td>
</tr>
<tr>
<td>r-S&amp;P500</td>
<td>-4.69%</td>
<td>-11.17%</td>
<td>-27.74%</td>
<td>-0.85%</td>
<td>22.95%</td>
<td>-9.27%</td>
<td>23.87%</td>
<td>5.24%</td>
<td>-9.21%</td>
</tr>
<tr>
<td>r-Logistic</td>
<td>6.47%</td>
<td>-5.07%</td>
<td>35.57%</td>
<td>34.00%</td>
<td>22.95%</td>
<td>15.02%</td>
<td>20.05%</td>
<td>-0.16%</td>
<td>-3.96%</td>
</tr>
<tr>
<td>r-LinReg</td>
<td>6.47%</td>
<td>-1.49%</td>
<td>35.57%</td>
<td>24.40%</td>
<td>22.95%</td>
<td>21.54%</td>
<td>18.72%</td>
<td>-1.12%</td>
<td>4.46%</td>
</tr>
<tr>
<td>r-FDA</td>
<td>0.43%</td>
<td>1.11%</td>
<td>33.24%</td>
<td>37.84%</td>
<td>19.76%</td>
<td>14.93%</td>
<td>14.22%</td>
<td>-0.62%</td>
<td>-5.32%</td>
</tr>
<tr>
<td>r-SVM</td>
<td>5.59%</td>
<td>-7.21%</td>
<td>35.57%</td>
<td>37.12%</td>
<td>16.69%</td>
<td>12.19%</td>
<td>14.69%</td>
<td>2.68%</td>
<td>11.26%</td>
</tr>
<tr>
<td>r-4Models</td>
<td>6.47%</td>
<td>-1.49%</td>
<td>123.49%</td>
<td>37.68%</td>
<td>22.95%</td>
<td>14.93%</td>
<td>20.64%</td>
<td>-0.16%</td>
<td>7.32%</td>
</tr>
<tr>
<td>r-Logistic</td>
<td>6.47%</td>
<td>-5.07%</td>
<td>123.49%</td>
<td>37.68%</td>
<td>22.95%</td>
<td>15.02%</td>
<td>20.05%</td>
<td>-0.16%</td>
<td>10.82%</td>
</tr>
<tr>
<td>r-LinReg</td>
<td>6.47%</td>
<td>-1.49%</td>
<td>123.49%</td>
<td>32.20%</td>
<td>22.95%</td>
<td>21.54%</td>
<td>18.72%</td>
<td>-1.12%</td>
<td>20.53%</td>
</tr>
<tr>
<td>r-FDA</td>
<td>0.43%</td>
<td>1.11%</td>
<td>119.64%</td>
<td>41.62%</td>
<td>19.76%</td>
<td>14.93%</td>
<td>14.22%</td>
<td>-0.62%</td>
<td>9.69%</td>
</tr>
<tr>
<td>r-SVM</td>
<td>5.59%</td>
<td>-7.21%</td>
<td>123.49%</td>
<td>40.88%</td>
<td>16.69%</td>
<td>12.19%</td>
<td>14.69%</td>
<td>2.68%</td>
<td>28.38%</td>
</tr>
<tr>
<td>r-4Models</td>
<td>6.47%</td>
<td>-1.49%</td>
<td>123.49%</td>
<td>26.70%</td>
<td>22.95%</td>
<td>14.93%</td>
<td>20.64%</td>
<td>-0.16%</td>
<td>23.84%</td>
</tr>
</tbody>
</table>

![Figure 1. The semiannual test value path plots of the model ensemble vs. the market benchmark](image)

The table 1 shows the realized semiannual hit-rates and the cumulative returns (after the transaction costs) of the 10 model settings, evaluated on the 9 semiannual test periods. As the benchmark, we also list the semiannual up-proportions and the cumulative returns of the S&P 500 index for the same 9 periods. Due to the data limitation, the first semiannual test period is partial (from 09/20/2007 to 12/31/2007) and the last semiannual test period is also partial (from 07/01/2011 to 11/16/2011). The figure 1 shows the semiannual test value paths of the four model combined under the use of the (15-16) and the S&P 500 index. The plot
corresponds to the bold rows $H^{\text{Modles}}$ and $r^{\text{Modles}}$ in table 1.

We can see that, under the virtual trading, the market can be beaten in the six periods 09/20/2007-12/31/2007, 01/02/2008-06/30/2008, 07/01/2008-12/31/2008, 01/02/2009-06/30/2009, 01/04/2010-06/30/2010, and 07/01/2011-11/16/2011. While it’s not the case in the three periods 07/01/2009-12/31/2009, 07/01/2010-12/31/2010, and 01/03/2011-06/30/2011 for all the 10 model settings. It’s not clear whether an individual model is superior or four models combined together is better, however, using (15-16) seems to generate better results than using (13-14) for the three periods 07/01/2008-12/31/2008, 01/02/2009-06/30/2009, and 07/01/2011-11/16/2011.

It can be observed that in the three periods 07/01/2009-12/31/2009, 07/01/2010-12/31/2010, and 01/03/2011-06/30/2011, the market was very bullish. And the best model could perform as same as the market. One explanation is that when the market enters bull days, it’s difficult for a model’s hit rate to match the chance that the market will go up. And it’s a well-known observation that for the S&P 500 index, the average daily rally in a bull market is small compared with the average daily plunge in a bear market. To illustrate how difficult it is to beat the bull market, consider a simple hypothetical scenario in which a model has at most 60% accuracy in a 100 days bull market that has 60% chance of going up 0.5% and 40% chance of going down 0.5%. The market’s cumulative return is 10.49%. For the model, each hit generates 0.5% return and each miss leads to 0.5% loss, obviously even without considering the transaction cost, 10.49% is the most return that can be achieved. Without any doubt, in a bull market, an investor should follow the market. Unfortunately, it’s very difficult to know when a bull market starts and ends. As the figure 1 shows, the market posted sustained gain from March 2009 to beyond the end of the year, however, the results of ensemble model show a continual draw-down during some period of that year.

CONCLUSIONS

In this paper, we proposed a rolling-based dynamic modeling and validation framework for the market direction prediction. In this framework, a newly developed calibration method that is based on two historically estimated conditional accuracies is combined with multiple machine learning models that generate probabilistic predictions of the S&P 500 index’s move direction. For the reason that there are much less trading data points than the created feature variables, we entertain the simple models with linear structure. And we use the principal component analysis for the feature vector’s decorrelation and dimension reduction in the rolling windows. Four models were applied in our experiments. They are the logistic regression, the linear regression, the Fisher’s linear discriminant analysis, and the SVM with soft margin. In the ensembles, we aggregate different prediction results either at the level of individual models or at the higher level of multiple models. In averaging the multiple prediction results, we replace the predictions that are deemed as “random” with the market’s up-prior in the validation rolling window.

The experimental results seem to be encouraging although it hardly beats the market during the bull periods. It should be pointed out that any backtesting result is determined by both the modeling methods and the constructed training data. We don’t claim that we have found a way to beat the market based on the limited experiments. We just want to argue that identifying the market’s signals may be possible if the models are properly designed and the training data is well built. It’s a well-accepted truth that data is always the first decisive factor. The better the available market data, the greater the chance that a model built upon it can catch the market patterns. We feel that the feature generation from the raw data is also very important. And the steps of feature decorrelation and dimension reduction via PCA did help a lot for solving the potential numerical issues in the model training.

We have not tried more complicate models like the neural networks, boosting, and random forests mainly for the concern that building models with highly nonlinear structure and much more parameters from very few number of training data points might incur severe over-fitting. But there are certainly many more simple models and even complicated models can be simplified. We do suggest some attempts on combining more predictive machine learning models with the calibration method that we proposed therein. However, we want to point out that the further improved results from extensive experiments on the fixed set of data do not guarantee the better performance in the future. It will be interesting to see how a historically successful model system performs on an extended time horizon, and what happens when the system begins to fail.
ACKNOWLEDGEMENTS

We thank the two anonymous referees and the editor-in-chief of the journal for their valuable comments and suggestions that helped improve the quality of this paper significantly.

REFERENCES


Xugang Ye is a research scientist at Microsoft and an adjunct instructor at the Department of Applied Mathematics and Statistics, the Johns Hopkins University, where he received his Ph.D. Dr. Ye has broad research interests in Machine Learning, Operations Research, and Statistics. His main research areas include quantitative finance, forecasting, Bayesian statistics, probabilistic generative models, and deep learning etc. He was a Charles Counselman fellow at the Johns Hopkins University and a Stephen Altschul fellow at the National Institutes of Health.

Jingjing Li is an applied scientist at Microsoft. She obtained her Ph.D. from the University of Colorado at Boulder. Her research focuses on combining behavioral theories and machine learning models to push the forefront of big data analytics and solve challenging issues in both academia and industry, such as dynamic forecasting, personalized recommendation, automated ontology learning, media search relevance and autosuggest. She contributes her expertise on dynamic behavior prediction and machine learning to this paper.