BOOSTING-BASED FRAMEWORKS IN FINANCIAL MODELING: APPLICATION TO SYMBOLIC VOLATILITY FORECASTING

Valeriy V. Gavrishchaka

ABSTRACT

Increasing availability of the financial data has opened new opportunities for quantitative modeling. It has also exposed limitations of the existing frameworks, such as low accuracy of the simplified analytical models and insufficient interpretability and stability of the adaptive data-driven algorithms. I make the case that boosting (a novel, ensemble learning technique) can serve as a simple and robust framework for combining the best features of the analytical and data-driven models. Boosting-based frameworks for typical financial and econometric applications are outlined. The implementation of a standard boosting procedure is illustrated in the context of the problem of symbolic volatility forecasting for IBM stock time series. It is shown that the boosted collection of the generalized autoregressive conditional heteroskedastic (GARCH)-type models is systematically more accurate than both the best single model in the collection and the widely used GARCH(1,1) model.

Econometric Analysis of Financial and Economic Time Series/Part B
Advances in Econometrics, Volume 20, 123–151
Copyright © 2006 by Elsevier Ltd.
All rights of reproduction in any form reserved
ISSN: 0731-9053/doi:10.1016/S0731-9053(05)20024-5

1. INTRODUCTION

Increasing availability of the high-frequency and multisource financial data has opened new opportunities for quantitative modeling and problem solving in a wide range of financial and econometric applications. These include such challenging problems as forecasting of the financial time series and their volatilities, portfolio strategy discovery and optimization, valuation of complex derivate instruments, prediction of rare events (e.g., bankruptcy and market crashes), and many others.

Such data availability also allows more accurate testing and validation of the existing models and frameworks to expose their limitations and/or confirm their advantages. For example, analytical and simple parametric models are usually clear and stable, but lack sufficient accuracy due to simplified assumptions. Adaptive data-driven models (mostly non-parametric or semi-parametric) can be very accurate in some regimes, but lack consistent stability and explanatory power (“black-box” feature). Moreover, the multivariate nature of many problems and data non-stationarity results in “dimensionality curse” (Bishop, 1995) and incompleteness of the available training data that is prohibitive for a majority of the machine learning and statistical algorithms.

Recently, the large margin classification techniques emerged as a practical result of the statistical learning theory, in particular, support vector machines (SVMs) (Vapnik, 1995, 1998; Cristianini & Shawe-Taylor, 2000). A large margin usually implies good generalization performance. The margin is the distance of the example to the class separation boundary. In contrast to many existing machine learning and statistical techniques, a large margin classifier generates decision boundaries with large margins to almost all training examples. This results in superior generalization ability on out-of-sample data.

SVMs for classification and regression have been successfully applied to many challenging practical problems. Recent successful applications of the SVM-based adaptive systems include market volatility forecasting (Gavrishchaka & Ganguli, 2003), financial time series modeling (Van Gestel et al., 2001), bankruptcy prediction (Fan & M. Palaniswami, 2000), space weather forecasting (Gavrishchaka & Ganguli, 2001a, b), protein function classification (Cai, Wang, Sun, & Chen, 2003), cancer diagnostics (Chang, Wu, Moon, Chou, & Chen, 2003), face detection and recognition (Osuna, Freund, & Girosi, 1997) as well as many other financial, scientific, engineering, and medical applications. In most cases, SVMs demonstrated superior results compared to other advanced machine learning and statistical techniques, including different types of neural networks (NNs).

Although SVMs are significantly more tolerant to data dimensionality and incompleteness (Cristianini & Shawe-Taylor, 2000; Gavrishchaka & Ganguli, 2001a, b), they can still fail in many important cases where available data does not allow generating stable and robust data-driven models. Moreover, SVM framework, like many other machine learning algorithms (including NNs), largely remains to be a "black box" with its limited explanatory power. Also, it is not easy to introduce a priori problem-specific knowledge into the SVM framework, except for some flexibility in the problem-dependent choice of the kernel type.

One of the machine learning approaches to compensate for the deficiency of the individual models is to combine several models to form a committee (e.g., Bishop, 1995; Hastie, Tibshirani, & Friedman, 2001; Witten & Frank, 2000). The committee can compensate limitations of the individual models due to both incomplete data and specifics of the algorithms (e.g., multiple local minima in the NN error surface). A number of different ensemble learning (model combination) techniques to build optimal committees have been proposed over the years in several research communities (e.g., Granger, 1989; Clemen, 1989; Drucker et al., 1994; Hastie, Tibshirani, & Friedman, 2001, and references therein). The work of Bates and Granger (1969) is considered to be the first seminal work on forecast combining (model mixing) in econometrics and financial forecasting. In the early machine learning literature, the same area of research was called "evidence combination" (e.g., Barnett, 1981).

Recently, there was a resurgence of interest in model combination in machine learning community (e.g., Schapire, 1992; Drucker et al., 1994; Dietterich, 2000; Ratsch, 2001). Modern research is mainly focused on the novel techniques that are suited for challenging problems with such features as a large amount of noise, limited number of training data, and high-dimensional patterns. These types of problems often arise in financial applications dealing with high-frequency and heterogeneous data (e.g., Gavrishchaka & Ganguli, 2003 and references therein), bioinformatics and computational drug discovery (e.g., Scholkopf, Tsuda, & Vert, 2004), space weather forecasting (e.g., Gleisner & Lundstedt, 2001; Gavrishchaka & Ganguli, 2001a, b), medical diagnostic systems (e.g., Gardner, 2004), and many others. Many modern ensemble learning algorithms perform a special manipulation with the training data set to compensate for the data incompleteness and its numerous consequences. Among them are bagging, cross-validating committees, and boosting (e.g., Drucker et al., 1994; Ratsch, 2001; Hastie et al., 2001).

The advantage of the ensemble learning approach is not only the possibility of the accuracy and stability improvement, but also its ability to combine a variety of models: analytical, simulation, and data-driven. This latter feature can significantly improve explanatory power of the combined model if building blocks are sufficiently simple and well-understood models. However, ensemble learning algorithms can be susceptible to the same problems and limitations as standard machine learning and statistical techniques. Therefore, the optimal choice of both the base model pool and ensemble learning algorithms with good generalization qualities and tolerance to data incompleteness and dimensionality is very important.

A very promising ensemble learning algorithm that combines many desirable features is boosting (Valiant, 1984; Schapire, 1992; Ratsch, 2001). Boosting and its specific implementations such as AdaBoost (Freund & Schapire, 1997) have been actively studied and successfully applied to many challenging classification problems (Drucker, Schapire, & Simard, 1993; Drucker et al., 1994; Schwenk & Bengio, 1997; Opitz & Maclin, 1999). Recently, boosting has been successfully applied to such practical problems as document routing (Iyer et al., 2000), text categorization (Schapire & Singer, 2000), face detection (Xiao, Zhu, & Zang, 2003), pitch accent prediction (Sun, 2002), and others.

One of the main features that sets boosting aside from other ensemble learning frameworks is that it is a large margin classifier similar to SVM. Recently, connection between SVM and boosting was rigorously proven (Schapire et al., 1998). This ensures superior generalization ability and better tolerance to incomplete data compared to other ensemble learning techniques. Similar to SVM, the boosting generalization error does not directly depend on the dimensionality of the input space (Ratsch, 2001). Therefore, boosting is also capable of working with high-dimensional patterns. The other distinctive and practically important feature of boosting is its ability to produce a powerful classification system starting with just a "weak" classifier as a base model (Ratsch, 2001).

In most cases discussed in the literature, boosting is used to combine data-driven models based on machine learning algorithms such as classification trees, NNs, etc. In this article, I will stress out advantages of the boosting framework that allows an intelligent combination of the existing analytical and other simplified and parsimonious models specific to the field of interest. In this way, one can combine clarity and stability typical for analytical and parsimonious parametric models with a good accuracy usually achieved only by the best adaptive data-driven algorithms. Moreover, since the underlying components are well-understood and accepted models, the obtained ensemble is not a pure "black box."

In the next two sections, I provide a short overview of the boosting key features and its relation to existing approaches as well as the description of the algorithm itself. After that, potential applications of boosting in quantitative finance and financial econometrics are outlined. Finally, a realistic example of the successful boosting application to symbolic volatility forecasting is given. In particular, boosting is used to combine different types of GARCH models for the next day threshold forecasting of the IBM stock volatility. Ensemble obtained with a regularized AdaBoost is shown to be consistently superior to both single best model and GARCH(1,1) model on both training (in-sample) and test (out-of-sample) data.

2. BOOSTING: THE MAIN FEATURES AND RELATION TO OTHER TECHNIQUES

Since the first formulation of the adaptive boosting as a novel ensemble learning (model combination) algorithm (Schapire, 1992), researchers from different fields demonstrated its relation to different existing techniques and frameworks. Such an active research was inspired by the boosting robust performance in a wide range of applications (Drucker et al., 1993, 1994; Schwenk & Bengio, 1997; Opitz & Maclin, 1999; Schapire & Singer, 2000; Xiao et al., 2003). The most common conclusion in machine learning community is that boosting represents one of the most robust ensemble learning methods. Computational learning theorists also proved boosting association with the theory of margins and SVMs that belong to an important class of large-margin classifiers (Schapire et al., 1998).

Recent research in statistical community is showing that boosting can also be viewed as an optimization algorithm in function space (Breiman, 1999). Statisticians consider boosting as a new class of learning algorithms that Friedman named "gradient machines" (Friedman, 1999), since boosting performs a stage wise greedy gradient descent. This relates boosting to particular additive models and matching pursuit known within the statistics literature (Hastie et al., 2001). There are also arguments that the original class of model mixing procedures are not in competition with boosting but rather can coexist inside or outside a boosting algorithm (Ridgeway, 1999).

In this section, I review boosting key features in the context of its relation to other techniques. Since the structure and the most practical interpretation of the boosting algorithm naturally relates it to ensemble learning techniques, comparison of boosting with other approaches for model combination will bo my main focus. Conceptual similarity and differences with other algorithms will be demonstrated from several different angles whenever possible.

The practical value of model combination is exploited by practitioners and researchers in many different fields. In one of his papers, Granger (1989) summarizes the usefulness of combining forecasts: "The combination of forecasts is a simple, pragmatic, and sensible way to possibly produce better forecasts." The basic idea of ensemble learning algorithms including boosting is to combine relatively simple base hypotheses (models) for the final prediction. The important question is why and when an ensemble is better than a single model.

In machine learning literature, three broad reasons for possibility of good ensembles' construction are often mentioned (e.g., Dietterich, 2000). First, there is a pure statistical reason. The amount of training data is usually too small (data incompleteness) and learning algorithms can find many different models (from model space) with comparable accuracy on the training set. However, these models capture only certain regimes of the whole dynamics or mapping that becomes evident in out-of-sample performance.There is also a computational reason related to the learning algorithm specifics such as multiple local minima on the error surface (e.g., NNs and other adaptive techniques). Finally, there is a representational reason when the true model cannot be effectively represented by a single model from a given set even for the adequate amount of training data. Ensemble methods have a promise of reducing these key shortcomings of standard learning algorithms and statistical models.

One of the quantitative and explanatory measures for the analysis and categorization of the ensemble learning algorithms is an error diversity (e.g., Brown, Wyatt, Harris, & Yao, 2005 and references therein). In particular, the ambiguity decomposition (Krogh & Vedelsby, 1995) and bias variance–covariance decomposition (Geman, Bienenstack, & Dourstat, 1992) provide a quantification of diversity for linearly weighted ensembles by connecting it back to an objective error criterion: mean-squared error. Although these frameworks have been formulated for regression problems, they can also be useful for the conceptual analysis of the classifier ensembles (Brown et al., 2005).

For simplicity, I consider only ambiguity decomposition that is formulated for convex combinations and is a property of an ensemble trained on a single dataset. Krogh and Vedelsby (1995) proved that at a single data point the quadratic error of the ensemble is guaranteed to be less than or equal to the average quadratic error of the base models:

$$(f-y)^2 = \sum_t w_t(f_t - y)^2 - \sum_t w_t(f_t - f)^2 \tag{1}$$

where f is a convex combination ($\sum_t w_t = 1$) of the base models f_t:

$$f = \sum_t w_t f_t \tag{2}$$

This result directly shows the effect due to error variability of the base models. The first term in (1) is the weighted average error of the base models. The second is the ambiguity term, measuring the amount of variability among the ensemble member answers for a considered pattern. Since this term is always positive, it is subtractive from the first term. This means that the ensemble is guaranteed lower error than the average individual error.

The larger the ambiguity term (i.e., error diversity), the larger is the ensemble error reduction. However, as the variability of the individual models rises, so does the value of the first term. Therefore, in order to achieve the lowest ensemble error, one should get the right balance between error diversity and individual model accuracy (similar to the bias-variance compromise for a single model). The same is conceptually true for the classification ensembles (Hansen & Salamon, 1990).

The above discussion offers some clues about why and when the model combination can work in practice. However, the most important practical question is how to construct robust and accurate ensemble learning methods. In econometric applications, the main focus is usually on equal weight and Bayesian model combination methods (e.g., Granger, 1989; Clemen, 1989). These methods provide a simple and well-grounded procedure for the combination of the base models chosen by a practitioner or a researcher. The accuracy of the final ensemble crucially depends on the accuracy and diversity of the base models. In some cases, a priori knowledge and problem-dependent heuristics can help to choose an appropriate pool of the base models.

However, equal weight and Bayesian combination methods do not provide any algorithmic procedures for the search, discovery, and building of the base models that are suitable for the productive combination in an ensemble. Such procedures would be especially important in complex high-dimensional problems with limited a priori knowledge and lack of simple heuristics. Without such algorithms for the automatic generation of the models with significant error diversity, it could be difficult or impossible to create powerful and compact ensembles from the simple base models.

In modern machine learning literature, the main focus is on the ensemble learning algorithms suited for challenging problems dealing with a large amount of noise, limited number of training data, and high-dimensional patterns (e.g., Ratsch, 2001; Buhlmann, 2003). Several modern ensemble learning techniques relevant for these types of applications are based on training data manipulation as a source of base models with significant error diversity. These include such algorithms as bagging (“bootstrap aggregation”), cross-validating committees, and boosting (e.g., Ratsch, 2001; Witten & Frank, 2000; Hastie et al., 2001, and references therein).

Bagging is a typical representative of “random sample” techniques in ensemble construction. In bagging, instances are randomly sampled, with replacement, from the original training dataset to create a bootstrap set with the same size (e.g., Witten & Frank, 2000; Hastie et al., 2001). By repeating this procedure, multiple training data sets are obtained. The same learning algorithm is applied to each data set and multiple models are generated. Finally, these models are linearly combined as in (2) with equal weights. Such combination reduces variance part of the model error and instability caused by the training set incompleteness.

Unlike basic equal weight and Bayesian combination methods, bagging offers a direct procedure to build base models with potentially significant error diversity (the last term in (1)). Recently, a number of successful econometric applications of bagging have been reported and compared with other model combination techniques (e.g., Kilian & Inoue, 2004).

Bagging exploits the instability inherent in learning algorithms. For example, it can be successfully applied to the NN-based models. However, bagging is not efficient for the algorithms that are stable, i.e., whose output is not sensitive to small changes in the input (e.g., parsimonious parametric models). Bagging is also not suitable for a consistent bias reduction.

Intuitively, combining multiple models helps when these models are significantly different from one another and each one treats a reasonable portion of the data correctly. Ideally, the models should complement one another, each being an expert in a part of the domain where performance of other models is not satisfactory. The boosting method for combining multiple models exploits this insight by explicitly seeking and/or building models that complement one another (Valiant, 1984; Schapire, 1992; Ratsch, 2001).

Unlike bagging, boosting is iterative. Whereas in bagging individual models are built separately, in boosting, each new model is influenced by the performance of those built previously. Boosting encourages new models to become experts for instances handled incorrectly by earlier ones. Final difference is that boosting weights obtained models by their performance, i.e., weights are not equal as in bagging. Unlike bagging and similar “random sample” techniques, boosting can reduce both bias and variance parts of the model error.

The initial motivation for boosting was a procedure that combines the outputs of many "weak" classifiers to produce a powerful committee (Valiant, 1984; Schapire, 1992; Ratsch, 2001). The purpose of boosting is to sequentially apply the weak classification algorithm to repeatedly modified versions of data, thereby producing sequence of weak classifiers. The predictions from all of them are then combined through a weighted majority vote to produce the final prediction. As iterations proceed, observations that are difficult to correctly classify receive an ever-increasing influence through larger weight assignment. Each successive classifier trained on the weighted error function is thereby forced to concentrate on those training observations that are missed by previous ones in the sequence.

Empirical comparative studies of different ensemble methods often indicate superiority of boosting over other techniques (Drucker et al., 1993, 1994; Schwenk & Bengio, 1997; Opitz & Maclin, 1999). Using probably approximately correct (PAC) theory, it was shown that if the base learner is just slightly better than random guessing, AdaBoost is able to construct ensemble with arbitrary high accuracy (Valiant, 1984). Thus, boosting can be effectively used to construct powerful ensembles from the very simplistic "rules of thumb" known in the considered field.

The distinctive features of boosting can also be illustrated through the error diversity point of view. In constructing the ensemble, the algorithm could either take information about error diversity into account or not. In the first case, the algorithm explicitly tries to optimize some measure of diversity during building the ensemble. This allows to categorize ensemble learning techniques as explicit and implicit diversity methods. While implicit methods rely on randomness to generate diverse trajectories in the model space, explicit methods deterministic ally choose different paths in the space.

In this context, bagging is an implicit method. It randomly samples the training patterns to produce different sets for each model and no measurement is taken to ensure emergence of the error diversity. On the other hand, boosting is an explicit method. It directly manipulates the training data distributions through specific weight changes to ensure some form of diversity in the set of base models. As mentioned before, equal weight and Bayesian model combination methods do not provide any direct algorithmic tools to manipulate error diversity of the base models.

Boosting has also a tremendous advantage over other ensemble methods in terms of interpretation. At each iteration, boosting trains new models or searches the pool of existing models that are complementary to the already chosen models. This often leads to the very compact but accurate ensemble with a clear interpretation in the problem-specific terms. This could also provide an important byproduct in the form of automatic discovery of the complementary models that may have value of their own in the area of application.

Boosting also offers a flexible framework for the incorporation of other ensemble learning techniques. For example, at each boosting iteration, instead of taking just one best model, one can use mini-ensemble of models that are chosen or built according to some other ensemble learning technique. From my empirical experience with boosting. I find it useful to form an equal weight ensemble of several comparable best models at each boosting iteration. Often, this leads to the superior out-of-sample performance compared to the standard boosted ensemble. Of course, there are many different ways to combine boosting with other ensemble techniques that could be chosen according to the specifics of the application.

So far, the main boosting features have been illustrated through its relation to other ensemble learning methods. However, one of the most distinctive and important features of boosting relates it to the large margin classifiers. It was found that boosted ensemble is a large margin classifier and that SVM and AdaBoost are intimately related (Schapire et al., 1998). Both boosting and SVM can be viewed as attempting to maximize the margin, except that the norm used by each technique and optimization procedures are different (Ratsch, 2001).

It is beyond the scope of this paper to review the theory of margins and structural risk minimization that is the foundation of the large margin classifiers including SVM (Vapnik, 1995, 1998). However, it should be mentioned that the margin of the data example is its minimal distance (in the chosen norm) to the classifier decision boundary. Large margin classifiers attempt to find decision boundary with large (or maximum) margin for all training data examples. Intuitively, this feature should improve the generalization ability (i.e., out-of-sample performance) of the classifier. Rigorous treatment of this topic can be found in (Vapnik, 1995, 1998; Ratsch, 2001).

Relation to the large margin classifiers partially explains robust generalization ability of boosting. It also clarifies its ability to work with high-dimensional patterns. Many traditional statistical and machine learning techniques often have the problem of "dimensionality curse" (Bishop, 1995), i.e., inability of handling high-dimensional inputs. The upper bound of the boosting generalization error depends on the margin and not on the dimensionality of the input space (Ratsch, 2001). Hence, it is easy to handle high-dimensional data and learn efficiently if the data can be separated with large margin. It has been shown that boosting superiority over other techniques is more pronounced in high-dimensional problems (e.g., Buhlmann, 2003).

3. ADAPTIVE BOOSTING FOR CLASSIFICATION

In this section, the technical details of a typical boosting algorithm and its properties are presented. For clarity, I will describe boosting only for two-class classification problem. A standard formalization of this task is to estimate a function $f : X \to Y$, where X is the input domain (usually R^n) and $Y = \{-1, +1\}$ is the set of possible class labels. An example (x_i, y_i) is assigned to the class $+1$ if $f(x) \geq 0$ and to the class -1 otherwise. Optimal function f from a given function set F is found by minimizing an error function (empirical risk) calculated on a training set $S = \{(x_1, y_1), ..., (x_N, y_N)\}$ with a chosen loss function $g(y, f(x))$:

$$\varepsilon[f, S] = \frac{1}{N}\sum_{n=1}^{N} g(y_n, f(x_n)) \tag{3}$$

Unlike the typical choice of squared loss in regression problems, here one usually considers the 0/1 loss:

$$g(y, f(x)) = I(-yf(x)) \tag{4}$$

where $I(z) = 0$ for $z < 0$ and $I(z) = 1$ otherwise. In a more general case, a cost measure of interest can be introduced through the multiplication of $g(y_n, f(x_n))$ by the normalized weight w_n of the training example.

Regularized AdaBoost for two-class classification consists of the following steps (Ratsch, 2001; Ratsch et al., 1999, 2001):

$$w_n^{(1)} = 1/N \tag{5}$$

$$\varepsilon_t = \sum_{n=1}^{N}[w_n^{(t)} I(-y_n h_t(x_n))] \tag{6}$$

$$\gamma_t = \sum_{n=1}^{N}[w_n^{(t)} y_n h_t(x_n)] \tag{7}$$

$$\alpha_t = \frac{1}{2}\log\frac{1+\gamma_t}{1-\gamma_t} - \frac{1}{2}\log\frac{1+C}{1-C} \tag{8}$$

$$w_n^{(t+1)} = w_n^{(t)} \exp[-\alpha_t y_n h_t(x_n)]/Z_t \tag{9}$$

$$f(x) = \frac{1}{\Sigma_{t=1}^{T}\alpha_t}\sum_{t=1}^{T}\alpha_t h_t(x) \tag{10}$$

Here N is a number of training data points, x_n a model/classifier input set of the nth data point, y_n the corresponding class label (i.e., -1 or $+1$), T the number of boosting iterations, $\omega_n^{(t)}$ a weight of the nth data point at tth iteration, Z_t the weight normalization constant at tth iteration, $h_t(x_n) \to [-1; +1]$ the best base hypothesis (model) at tth iteration, C a regularization (soft margin) parameter, and $f(x)$ is a final weighted linear combination of the base hypotheses (models).

Boosting starts with equal and normalized weights for all training data (step (5)). A base classifier (model) $h_t(x)$ is trained using weighted error function ε_t (step (6)). If a pool of several types of base classifiers is used, then each of them is trained and the best one (according to ε_t) is chosen at the current iteration. The training data weights for the next iteration are computed in steps (7)–(9). It is clear from step (9) that at each boosting iteration, data points misclassified by the current best hypothesis (i.e., $y_n h_t(x_n) < 0$) are penalized by the weight increase for the next iteration. In subsequent iterations, AdaBoost constructs progressively more difficult learning problems that are focused on hard-to-classify patterns. This process is controlled by the weighted error function (6).

Steps (6)–(9) are repeated at each iteration until stop criteria $\gamma_t \leq C$ (i.e.,$\varepsilon_t \geq \frac{1}{2}(1 - C)$) or $\gamma_t = 1$ (i.e., $\varepsilon_t = 0$) occurs. The first stop condition means that, for the current classification problem, a classifier with accuracy better than random guess (corrected by the regularization multiplier $(1 - C)$) cannot be found. The second stop condition means that the perfect classifier is found and formulation of the next iteration classification problem focusing on misclassified examples is not needed.

Step (10) represents the final combined (boosted) model that is ready to use. The model classifies an unknown example as class $+1$ when $f(x) > 0$ and as -1 otherwise. It should be noted that the performance of the final ensemble (10) is evaluated according to the original error function given by (3) and (4) (not by (6)). Details of the more general versions of the regularized AdaBoost and its extensions are given in (Ratsch, 2001).

The original AdaBoost algorithm (Freund & Schapire, 1997) can be recovered from (5)–(10) for $C = 0$. Similar to SVM, regularization parameter C represents the so-called soft margin. Soft margin is required to accommodate the cases where it is not possible to find a boundary that fully separates data points from different classes. Regularization (soft margin) is especially important for financial applications where large noise-to-signal ratio is a typical case.

One of the most important properties of the AdaBoost is its fast convergence to a hypothesis, which is consistent with the training sample, if the base learning algorithm produces hypotheses with error rates consistently smaller than 1/2. A theorem on the exponential convergence of the original AdaBoost ($C = 0$) states that the error (given by (3) and (4)) of the ensemble (10) on the training set is bounded above by $2^T \Pi_{t=1}^{T} \sqrt{\varepsilon_t(1-\varepsilon_t)}$ (Freund & Schapire, 1997). Thus, if the error rate in each iteration is bounded from above by $\varepsilon_t \leq \frac{1}{2} - \frac{1}{2}\mu$ (for some $\mu > 0$), then the training error ε decreases exponentially in the number of iterations: $\varepsilon \leq \exp(-T\mu^2/2)$. This and the other theorems for the original AdaBoost have been generalized for the regularized case ($C > 0$) (Ratsch, 2001).

Motivated by boosting success in classification setting, a number of researchers attempted to transfer boosting techniques to regression problems. A number of different boosting-like algorithms for regression have been proposed (e.g., Drucker, 1997; Friedman, 1999; Ratsch, 2001). However, for clarity and owing to the existence of many practically important classification problems, I focus on boosting for classification in this article.

4. BOOSTING FRAMEWORKS IN FINANCIAL AND ECONOMETRIC APPLICATIONS

Several conceptually different boosting frameworks can be proposed for financial and econometric applications. In the most straightforward way, boosting can be employed as yet another advanced data-driven algorithm expanding collection of tools used for building empirical models. However, models, built in this way, would still have "black box" nature and potentially limited stability.

In our opinion, the most appealing feature of boosting, especially, for financial applications, is that it combines the power of the advanced learning algorithm with the ability to use existing models specific to the field of interest. This important feature has never been addressed in machine learning and related literature on boosting.

The mentioned combination can be easily achieved by using well-understood and accepted models as base models (hypotheses) in a boosting framework. This allows integrating a priori problem-specific knowledge in a natural way that could lead to enhancement of the final model performance and stability, especially, in the cases of severe incompleteness of the training data, Moreover, since the final model is a weighted linear combination of the industry-standard components, the conclusions of such a model will be better accepted by practitioners and decision makers compared to a pure "black-box" model. In the following, only examples of this type of boosting applications will be discussed.

Any mature field of quantitative finance has its own set of established and well-understood models. In many cases it is very desirable to improve accuracy of the models without losing their clarity and stability. In the following, I will give several examples from different fields. One of them (symbolic volatility forecasting) will be used as an illustration of the realistic boosting application in the next section. Encouraging results and all steps of boosting application to this practically important problem will be presented.

4.1. Typical Classification Problems

It would be convenient to distinguish four large groups of the potential boosting frameworks in quantitative finance and financial econometrics. The first group includes well-defined classification models that can be used as base hypotheses in a standard boosting framework ((5)–(10)).

For example, bankruptcy prediction models based on accounting and market data are used to classify companies as bankrupt/non-bankrupt (usually for the time horizon of 1 year). These models are based on linear discriminant analysis (LDA), logit, and other statistical and machine learning frameworks (e.g., Hillegeist, Keating, Cram, & Lundstedt, 2004; Fan & Palaniswami, 2000; Fanning & Cogger, 1994). Typical industry-standard models include Altman's Z-score (LDA) (Altman, 1968), Ohlson model (logit) (Ohlson, 1980), and variations of the Black–Scholes–Merton model (Hillegeist et al., 2004). One can use the same inputs (financial ratios) and framework (LDA, logit, etc.) as in well-accepted models and build classifier on a weighted training set at every boosting iteration. Obtained boosted combination will still be based on a trusted and simple framework while it may have significantly better performance compared to the single base model.

4.2. Symbolic Time Series Forecasting

Prediction of the financial time series (stocks, market indexes, foreign exchange rates, etc.) and their volatilities is one of the most challenging and generally unresolved problems. Time series predictor with sufficient accuracy is a desired component in both trading and risk management applications. In many practical cases forecasting of the actual future value of the time series is not required. Instead it is enough to forecast symbolically encoded value.

For example, the model can predict whether the future value will be supercritical or subcritical to some threshold value or just the direction of change (up or down). In many practical problems, switching from the full regression problem (i.e., actual value prediction) to the symbolic encoding allows to increase accuracy of the prediction, since practically unimportant small fluctuations and/or noise are removed by this procedure (Gavrishchaka & Ganguli, 2001b; Tino, Schittenkopf, Dorffner, & Dockner, 2000).

Classification problem that is obtained from the original time series forecasting problem by symbolic encoding is a second group suited for the standard boosting framework (5)–(10). Boosting can combine existing analytical and other parsimonious time series models. For example, GARCH-type framework (Engle, 1982; Bollerslev, 1986) is an industry standard for the volatility modeling. GARCH-type models can serve as base hypotheses pool in a boosting framework for the volatility forecasting. Details and preliminary results of the boosting application to the symbolic volatility forecasting will be discussed in the next section.

4.3. Portfolio Strategy Discovery and Optimization

Discovery and optimization of the portfolio strategy is an important area where boosting may also prove to be useful. Simple trading rules based on the well-known market indicators often have significant performance and stability limitations. They can also be used by many market participants easily making market efficient to these sets of trading strategies. On the other hand, adaptive strategies, capable of utilizing more subtle market inefficiencies, are often of the "black-box" type with limited interpretability and stability. Combining basic (intuitive) trading strategies in a boosting framework may noticeably increase return (decrease risk) while preserving-clarity of the original building blocks.

Straightforward application of the standard boosting framework (5)–(10) for the trading strategy optimization is not possible, since it is a direct optimization and not classification problem. However, for a large class of optimization objective functions, boosting for classification can be easily transformed into a novel framework that can be called "boosting for optimization."

For example, one can require returns (r) generated by the strategy on a given time horizon (τ) to be above certain threshold (r_c). By calculating strategy returns on a series of shifted intervals of length τ and using two-class encoding ($r >= r_c$ and $r < r_c$), one obtains symbolically encoded time series as previously discussed. Encoding can be based on a more sophisticated condition that combines different measures of profit maximization and risk minimization according to the utility function of interest.

Contrary to symbolic time series forecasting discussed earlier, here the purpose is not correct classification, but rather maximization of one class (i.e., $r >= r_c$) population. Formally, one still has classification problem where the boosting framework (5)–(10) can be applied. However, in this case, one has very uneven sample distribution between two classes. Therefore, not all theoretical results that have been rigorously proven for boosting could be automatically assumed to be valid here. Nevertheless, our preliminary empirical studies indicate consistent stability and robustness of the boosting for optimization framework when applied to technical stock/index trading.

In the case of trading strategy optimization, the usage of boosting output is also different. Instead of using weighted linear combination of the base hypotheses as a model for classification, one should use boosting weights for strategy portfolio allocation. It means that initial capital should be distributed among different base strategies in amounts according to weights obtained from the boosting. Details of boosting application to the portfolio strategy optimization will be given in our future work.

4.4. Regression Problems

Finally, the fourth group of applications explicitly requires boosting for regression (Ratsch, 2001). For example, time series forecasting models can be combined in their original form without casting into classification problem through symbolic encoding.

Another practically important application of the boosting for regression may be construction of the pricing surface for complex derivative instruments using simplified analytical or numerical valuation models as base hypotheses. This may significantly improve representation accuracy of the real market pricing surface compared to a single pricing model. Boosting-based pricing framework could be especially useful for less common or novel derivative instruments where valuation models used by different market participants are not yet standardized. This type of boosting application can be considered as a hybrid derivative pricing contrary to pure empirical pricing approaches based on machine learning algorithms (e.g., Hutchinson, Lo, & Poggio, 1994; Gencay & Qi, 2001).

5. SYMBOLIC VOLATILITY FORECASTING

In this section, I consider boosting-based framework for symbolic volatility forecasting together with all the details of its implementation and application. Stock market index and foreign exchange rate volatility are very important quantities for derivative pricing, portfolio risk management, and as one of the components used for decision making in trading systems. It may be the main component for the quantitative volatility trading (e.g., Dunis & Huang, 2002). I start with a short overview of the existing deterministic volatility models and their limitations. Stochastic volatility models are not considered here.

A common example of the deterministic volatility models is autoregressive conditional heteroskedastic (ARCH)-type models (Engle, 1982; Bollerslev, 1986). These models assume a particular stochastic process for the returns and a simple functional form for the volatility. Volatility in these models is unobservable (latent) variable. The most widely used model of this family is generalized ARCH (GARCH) process (Bollerslev, 1986). GARCR(p,q) process defines volatility as

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i r_{t-i}^2 + \sum_{i=1}^{q} \beta_i \sigma_{t-i}^2 \tag{11}$$

where return process is defined as

$$r_t = \sigma_t \zeta_t \tag{12}$$

Here ζ_t is an identically and independently distributed (i.i.d.) random variable with zero mean and variance 1. The most common choice for the return stochastic model (ζ_t) is a Gaussian (Wiener) process. Parameters α_i and β_i from equation are estimated from historical data by maximizing the likelihood function (LF) which depends on the assumed return distribution.

To resolve some of the GARCH limitations, a number of extensions have been proposed and used by practitioners. For example, since GARCH model depends only on the absolute values of returns (r_i^2), it does not cover leverage effect. Different forms of leverage effect have been introduced in EGARCH (Nelson, 1991), TGARCH (Zakoian, 1994), and PGARCH (Ding, Granger, & Engle, 1993) models that are given below:

$$\ln(\sigma_t^2) = \alpha_0 + \sum_{i=1}^{p} \alpha_i \frac{|r_{t-i}| + \gamma_i r_{t-i}}{\sigma_{t-i}} + \sum_{i=1}^{q} \beta_i \ln(\sigma_{t-i}^2) \tag{13}$$

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i r_{t-i}^2 + \sum_{i=1}^{p} \gamma_i S_{t-i} r_{t-i}^2 + \sum_{i=1}^{q} \beta_i \sigma_{t-i}^2 \tag{14}$$

$$\sigma_t^d = \alpha_0 + \sum_{i=1}^{p} \alpha_i (|r_{t-i}| + \gamma_i r_{t-i})^d + \sum_{i=1}^{q} \beta_i \sigma_{t-i}^d \tag{15}$$

Here γ_i is a leverage parameter. For TGARCH model, $S_{t-i} = 1$ for $r_{t-i} < 0$ and $S_{t-i} = 0$ otherwise. For PGARCH, d is a positive number that can also be estimated together with α_i, β_i, and γ_i coefficients.

GARCH and its extensions are the most common choices of the volatility model by practitioners. GARCH process can reproduce a number of known stylized volatility facts, including clustering and mean reversion (e.g., Engle & Patton, 2001; Tsay, 2002). Explicit specification of the stochastic process and simplified (linear) functional form for the volatility allows to do simple analysis of the model properties and its asymptotic behavior.

However, assumptions of the ARCH-type models also impose significant limitations. Model parameter calculation from the market data is practical only for low-order models (small p and q), i.e., in general, it is difficult to capture direct long memory effects. Volatility multiscale effects are also not covered (Dacorogna, Gencay, Muller, Olsen, & Pictet, 2001). Finally, the model gives unobservable quantity that makes it difficult to quantify the prediction accuracy and comparison with other models. Some of these restrictions are relaxed in the GARCH model extensions. However, majority of the limitations mentioned cannot be resolved in a self-consistent fashion.

All advantages and limitations of the GARCH-type models, mentioned in the previous two paragraphs, suggest that the framework accepted by many practitioners is theoretically sound and clear. It can reproduce all major signatures empirically observed in real time series. On the other hand, its forecasting accuracy may be insufficient for a number of important applications and regimes (e.g., Gavrishchaka & Ganguli, 2003).

Accuracy can be improved by using more complex models that are often of the "black-box" type. For example, we have recently proposed the SVM-based volatility model (Gavrishchaka & Ganguli, 2003) with encouraging results for both foreign exchange and stock market data. Previously, different NN-based frameworks have also been proposed for the same purpose (Donaldson & Kamstra, 1997; Schittenkopf, Dorffner, & Dockner, 1998). However, limited stability and explanation facility of these "black-box" frameworks may restrict their usage only as complementary models (Gavrishchaka & Ganguli, 2003). Boosting offers an alternative way that allows to combine all the advantages of the accepted industry models such as GARCH with significant accuracy improvement.

In many practical applications, symbolic volatility forecasting may be acceptable and even desirable due to effective filtering of noise and unimportant small fluctuations. For example, for many trading and risk management purposes, it is important (and sufficient) to know whether the future volatility value will be above or below a certain threshold. This problem is of classification type and standard boosting framework (l)–(5) can be applied. Below I will give a detailed example of the boosting application to this problem.

I consider boosting-based framework for symbolic (threshold) volatility forecasting with a base hypotheses pool consisting of the GARCH-like models (Eqs. (11)–(15)) with different types and input dimensions. In the following, I will restrict the pool to 100 models that are obtained from all possible combinations of the model types (GARCH, EGARCH, TGARCH, and PGARCH), return input dimension (from 1 to 5), and σ input dimension (from 1 to 5). In our case, the number of the possible base hypothesis on each boosting iteration is countable and fixed. Therefore, the best model at each iteration is just one of the 100 models with a minimal classification error on a weighted training set. It is different from the more conventional usage of boosting where the available base models are not countable. Instead, at each iteration, the best model is obtained by training base models (NN, classification tree, etc.) on a weighted training set (Ratsch, 2001).

As a proxy for realized volatility that will be compared with a given threshold to give supercritical/subcritical classification (encoded as "+1" and "−1"), standard deviation of the daily returns computed on the last five business days will be used. Threshold value of the daily realized volatility is taken to be 2% that corresponds to 32% of the annualized volatility. Although the exact threshold value is problem dependent, the threshold considered here is a reasonable boundary between low volatility regime (usually taken to be 20–25% of annual volatility) and higher volatility regimes.

It is difficult to find an adequate measure of the realized volatility that can be used to test GARCH model performance (e.g., Tsay, 2002). The reason is that GARCH model outputs instantaneous (i.e., "true") volatility which is a latent (non-observable) variable. Any volatility measure computed on a real data set will be an approximation due to volatility time dependence. In many academic studies, performance of the GARCH-type models is measured by comparing model output σ_i^2 with a single day squared return r_i^2 through autocorrelation of r_i^2/σ_i^2 time series (Tsay, 2002).

However, although usage of r_i^2 is appealing due to its direct theoretical meaning (r_i^2/σ_i^2 should approach i.i.d. according to (12)), single day r_i^2 is never used as a volatility measure in practice. Daily r_i^2 time series is usually extremely noisy and can have large magnitudes that are unattainable by any practical volatility model. Volatility measure, based on a five-day standard deviation, significantly removes the noise, makes magnitude comparable to GARCH outputs, while still preserves instantaneous nature to some extent. On the other hand, standard deviation based on large data interval gives just unconditional averaged volatility that cannot be compared with GARCH conditional volatility.

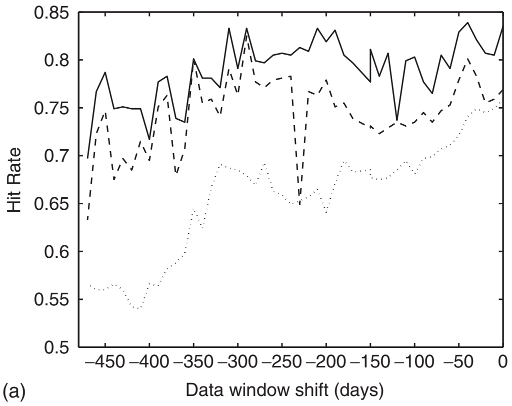
The choice of the realized volatility measure should be imposed by the requirements of the particular application. The purpose of boosting is not a "pure" test of the GARCH framework as a model for the "true" (latent) volatility. Instead, GARCH models are used as base experts that can capture the main generic features of the volatility, and boosting is used to make accurate forecasting of a particular practically important measure that is related but not identical to the "true" volatility.

I have applied boosting framework for symbolic volatility forecasting to last several years of IBM stock data. The daily closing prices adjusted for splits and dividends from finance.yahoo.com have been used. For benchmark purposes, boosting results are compared to the best model in the base hypothesis collection (as per training set) and industry-standard GARCH(1,1) model. To demonstrate result sensitivity to market conditions, data samples shifted in time are used.

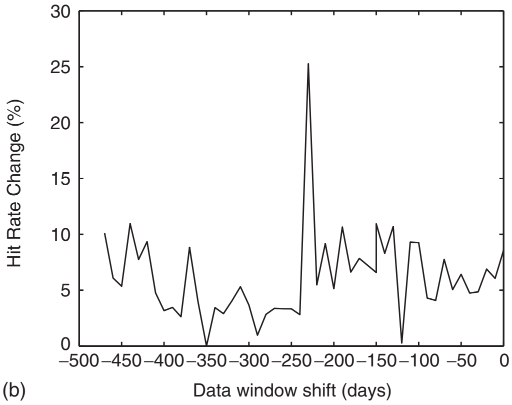
The first data sample, corresponding to shift = 0 in all figures, includes 3 years of IBM stock daily return data (April 18, 2001–March 4, 2004). Return is computed as $r_i = \ln(S_i/S_{i-1})$, where S_i is stock closing price at day i. First 2/3 of the data sample are used as training data and the remaining part as test data. All subsequent samples are obtained by 10 business days shifting backward in time. For simplicity and clarity, I do not use validation set that allows to choose best regularization parameters C in the regularized AdaBoost algorithm (5)–(10). Instead, I use a fixed value $C = 0.05$ that shows good performance across all samples. Adaptive choice of C should only improve the performance of the boosting-based model.

In Fig. 1a, I compare performance of the boosting model (solid line), single best model from the base model collection (dashed line), and GARCH(1, 1) model (dotted line) on the training sets of all shifted data intervals (samples). As a performance measure, I use hit rate defined as a number of correctly classified data points to the total number of data points (i.e., maximum possible hit rate is 1). This measure is directly related to the

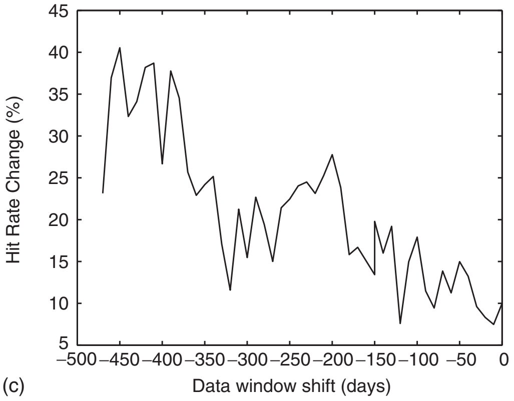
Fig. 1. (a) Hit Rate of the Boosting Model (Solid Line), Single Best Model from the Base Model Collection (Dashed Line), and GARCH(1,1) Model (Dotted Line) on the Training Sets of All Shifted Data Intervals, (b) Percentage Change of the Boosting Model Hit Rate Relative to that of the Best Single Model (Training Sets), and (c) Percentage Change of the Boosting Model Hit Rate Relative to that of the GARCH(1,1) Model (Training Sets).


Hit Rate
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
−450 −400 −350 −300 −250 −200 −150 −100 −50 0
Data window shift (days)

(a)


Hit Rate Change (%)
30
25
20
15
10
5
0
−500 −450 −400 −350 −300 −250 −200 −150 −100 −50 0
Data window shift (days)

(b)


Hit Rate Change (%)
45
40
35
30
25
20
15
10
5
−500 −450 −400 −350 −300 −250 −200 −150 −100 −50 0
Data window shift (days)

(c)

standard classification error function given by (3) and (4). It is clear that boosting models produce maximum hit rates at all times. For some samples, boosting performance is significantly superior compared to the other two models. Note that the best single model (i.e., its type and input dimensions) changes from sample to sample. The best model is the model with a maximum hit rate for a given sample.

To further quantify boosting model superiority over the two benchmark models, I compute the percentage change of the boosting model hit rate relative to the benchmark model. This number is given by $100 \times (HR - HR_0)/HR_0$, where HR and HR_0 are hit rates of the boosting and benchmark models, respectively. In Fig. 1b, this relative measure is shown for the best single model as a benchmark. One can see that in some regimes, boosting can enhance performance by up to 25%. Similar result for GARCH(1,1) model as a benchmark is shown in Fig. 1c. In this case, boosting can increase hit rate by up to 40%.

Although boosting demonstrated a superior performance on the training set, the true measure of the model performance is on the test (out-of-samplc) data. This demonstrates the generalization abilities of the model. In Fig. 2, I show exactly the same measures as in Fig. 1 but for the test sets of the shifted samples. The best single model is still the best model on the training set as would be the choice in a real application where performance on the test set is not known at the time of training. From Fig. 2, it is clear that the boosting model remains superior to both benchmark models on the test set. Moreover, the hit rate relative change even increases on the test set compared to the training set. Thus, boosting demonstrates robust out-of-sample performance and confirms its solid generalization ability.

Finally, I would like to illustrate what resources have been provided by the pool of the base models (hypotheses) such that boosting was able to achieve a significant performance improvement compared to the single base model. Figure 3 shows the number of the models that correctly classify a data point normalized to the number of all models in the pool. As an example, I consider just one sample with shift = 0 (a – training set, b – test set). It is clear that for all data points, there is a significant number of models that correctly classify this data point. However, if one considers even the best single model, a significant number of points will be misclassified. Fortunately, majority of points, misclassified by the best model, can be correctly classified by some complementary models (that may be noticeably inferior to the best model on the whole set). Thus, boosting robustly finds these complementary models at each iteration that results in a much more accurate final combined model.

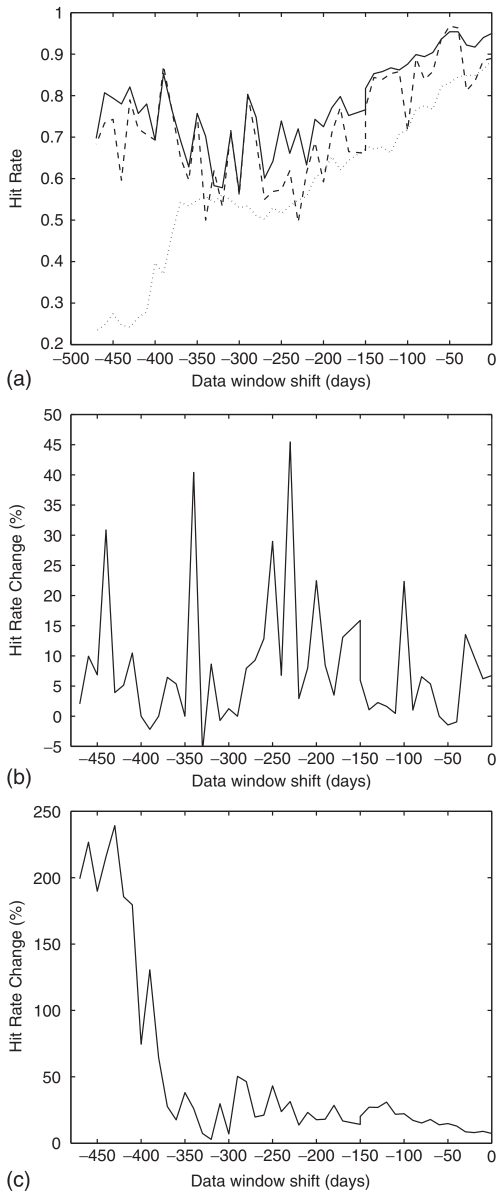


Fig. 2. The Same Measures as in Fig. 1, but for the Test (Out-of-Sample) Sets.

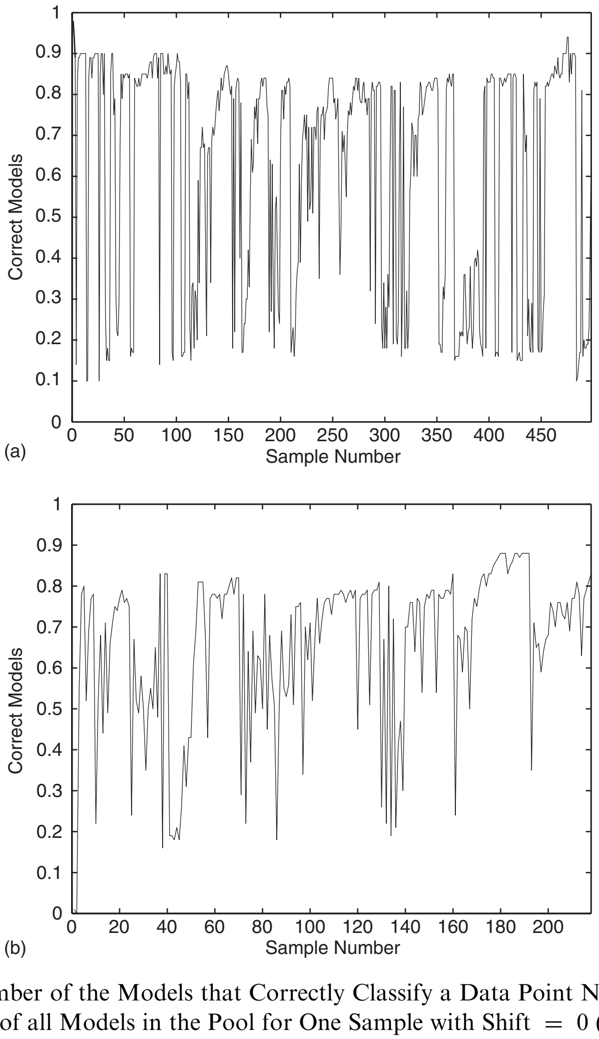


Fig. 3. Number of the Models that Correctly Classify a Data Point Normalized to the Number of all Models in the Pool for One Sample with Shift $= 0$ (a – Training Set, b – Test Set).

In the presented illustrative application, I have considered rather a restrictive pool of base volatility models. However, even with that pool, boosting was able to achieve significant improvement in performance. In practice, one can expand this pool by including more advanced and flexible models such as heterogeneous ARCH (HARCH) (Dacorogna et al., 2001) (accounts for multiscale effects), fractionally integrated GARCH (FIGARCH) (Baillie, Bollerslev, & Mikkelson, 1996) (models short and long-term memory), and other models, including those outside the GARCH family. In addition, hybrid volatility models (e.g., Day & Lewis, 1992), that combine delayed historical inputs with available implied volatility data, could also be included in the base model pool. More detailed study of the boosting-based models for volatility forecasting is warranted.

6. DISCUSSION AND CONCLUSION

Limitations of the existing modeling frameworks in quantitative finance and econometrics, such as low accuracy of the simplified analytical models and insufficient interpretability and stability of the best data-driven algorithms have been outlined. I have made the case that boosting (a novel, ensemble learning technique) can serve as a simple and robust framework for combining the clarity and stability of the analytical models with the accuracy of the adaptive data-driven algorithms.

A standard boosting algorithm for classification (regularized AdaBoost) has been described and compared to other ensemble learning techniques. I have pointed out that boosting combines the power of the large margin classifier characterized by a robust generalization ability and open framework that allows to use simple and industry-accepted models as building blocks. This distinguishes boosting from other powerful machine learning techniques, such as NNs and SVMs, that often have "black-box" nature.

Potential boosting-based frameworks for different financial applications have been outlined. Some of them can use standard boosting framework without modifications. Others require reformulation of the original problem to be used with boosting. Detailed description of these frameworks and results of their application will be presented in our future works.

Details of a typical boosting operation have been demonstrated on the problem of symbolic volatility forecasting for IBM stock time series. Two-class threshold encoding of the predicted volatility has been used. It has been shown that the boosted collection of the GARCH-type models performs consistently better compared to both the best single model in the collection and the widely used GARCH(1,1) model. Depending on the regime (period in time), classification hit rate of the boosted collection on the test (out-of-sample) set can be up to 240% higher compared to GARCH(l, l) models and up to 45% higher compared to the best single model in the collection. For a training set, these numbers are 40% and 25%, respectively.

Boosting performance is also very stable. In the worst case/regime, performance of the boosted collection just becomes comparable to the best single model in the collection.

Detailed comparative studies of boosting and other ensemble learning (model combination) methods for typical econometric applications are beyond the scope of this work. However, to understand the real practical value of boosting and its limitations, such studies should be performed in the future.

ACKNOWLEDGMENTS

This work is supported by Alexandra Investment Management. I thank the referee who evaluated this paper and the editor for the valuable comments and suggestions.

REFERENCES

Altman, E. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *Journal of Finance*, *23*, 589–609.

Baillie, R. T., Bollerslev, T., & Mikkelsen, H.-O. (1996). Fractionally integrated generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, *74*, 3.

Barnett, J. A. (1981). Computational methods for a mathematical theory of evidence. *Proceedings of IJCAI*, *868*.

Bates, J. M., & Granger, C. W. J. (1969). The combination of forecasts. *Operational Research Quarterly*, *20*, 451.

Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford: Clarendon Press.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, *31*, 307.

Breiman, L. (1999). Prediction games and arcing algorithms. *Neural Computation*, *11*(7), 1493–1518.

Brown, G., Wyatt, J., Harris, R., & Yao, X. (2005). Diversity creation methods: A survey and categorisation. *Journal of Information Fusion*, *6*, 1.

Cai, C. Z., Wang, W. L., Sun, L. Z., & Chen, Y. Z. (2003). Protein function classification via support vector machine approach. *Mathematical Biosciences*, *185*, 111.

Chang, R. F., Wu, W.-J., Moon, W. K., Chou, Y.-H., & Chen, D.-R. (2003). Support vector machine for diagnosis of breast tumors on US images. *Academy of Radiology*, *10*, 189.

Clemen, R. T. (1989). Combining forecasts: A review and annotated bibliography. *International Journal of Forecasting*, *5*, 559.

Cristianini, N., & Shawe-Taylor, J. (2000). *Introduction to support vector machines and other kernel-based learning methods*. Cambridge: Cambridge University Press.

Dacorogna, M. M., Gencay, R., Muller, U., Olsen, R. B., & Pictet, O. V. (2001). *An introduction to high-frequency finance*. San Diego: Academic Press.

Day, T. E., & Lewis., C. M. (1992). Stock market volatility and the information content of stock index options. *Journal of Econometrics*, *52*, 267.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In: J. Kittler & F. Roli (Eds), *First international workshop on multiple classifier systems, Lecture Notes in Computer Science* (pp. 1–15). Heidelberg: Springer Verlag.

Ding, Z., Granger, C. W., & Engle, R. F. (1993). A long memory property of stock market returns and a new model. *Journal of Empirical Finance*, *1*, 83.

Donaldson, R. G., & Kamstra, M. (1997). An artificial neural network-GARCH model for international stock return volatility. *Journal of Empirical Finance*, *4*, 17.

Drucker, H. (1997). Improving regressors using boosting techniques. In: D. H. Fisher (Ed.), *Proceedings of the fourteenth international conference on machine learning (ICML 1997)* (pp. 107–115). Nashville, TN, USA, July 8–12. Morgan Kaufmann, ISBN 1–55860–486–3.

Drucker, H., Cortes, C., Jackel, L. D., LeCun, Y., & Vapnik, V. (1994). Boosting and other ensemble methods. *Neural Computation*, *6*, 1289–1301.

Drucker, H., Schapire, R. E., & Simard, P. Y. (1993). Boosting performance in neural networks. *International Journal of Pattern Recognition and Artificial Intelligence*, *7*, 705.

Dunis, C. L., & Huang, X. (2002). Forecasting and trading currency volatility: An application of recurrent neural regression and model combination. *Journal of Forecasting*, *21*, 317.

Engle, R. F. (1982). Autoregressive conditional heteroskedasticity with estimates of the variance of UK inflation. *Econometrica*, *50*, 987.

Engle, R. F., & Patton, A. J. (2001). What good is a volatility model? *Quantitative Finance*, *1*, 237.

Fan, A., & Palaniswami, M. (2000). Selecting bankruptcy predictors using a support vector machine approach. *Proceedings of the IEEE-INNS-ENNS international joint conference on neural networks*, *IJCNN 2000, Neural computing: New challenges and perspectives for the new millennium* (Vol. 6, pp. 354–359). Como, Italy, July 24–27.

Fanning, K., & Cogger, K. (1994). A comparative analysis of artificial neural networks using financial distress prediction. *International Journal of Intelligent Systems in Accounting, Finance, and Management*, *3*(3), 241–252.

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of online learning and an application to boosting. *Journal of Computer and System Sciences*, *55*, 119.

Friedman, J. H. (1999). *Greedy function approximation*. Technical report (February). Department of Statistics, Stanford University.

Gardner, A. (2004). *A novelty detection approach to seizure analysis from intracranial EEG*. Ph.D. thesis, School of Electrical and Computer Engineering, Georgia Institute of Technology.

Gavrishchaka, V. V., & Ganguli, S. B. (2001a). Support vector machine as an efficient tool for high-dimensional data processing: Application to substorm forecasting. *Journal of Geophysical Research*, *106*, 29911.

Gavrishchaka, V. V., & Ganguli, S. B. (2001b). Optimization of the neural-network geomagnetic model for forecasting large-amplitude substorm events. *Journal of Geophysical Research*, *106*, 6247.

Gavrishchaka, V. V., & Ganguli, S. B. (2003). Volatility forecasting from multiscale and high-dimensional market data. *Neurocomputing*, *55*, 285.

Geman, S., Bienenstock, E., & Dourstat, R. (1992). Neural networks and the bias/variance dilemma. *Neural Computation*, *4*, 1.

Gencay, R., & Qi, M. (2001). Pricing and hedging derivative securities with neural networks: Bayesian regularization, early stopping and bagging. *IEEE Transactions on Neural Networks*, *12*(4), 726–734.

Gleisner, H., & Lundstedt, H. (2001). Auroral electrojet predictions with dynamic neural networks. *Journal of Geophysical Research*, *106*, 24, 541.

Granger, C. W. J. (1989). Combining forecasts – Twenty years later. *Journal of Forecasting*, *8*, 167.

Hansen, L. K., & Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*, 993.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning: Data mining, inference, and prediction*. Berlin: Springer.

Hillegeist, S. A., Keating, E. K., Cram, D. P., & Lundstedt, K. G. (2004). Assessing the probability of bankruptcy. *Review of Accounting Studies*, *9*, 5–34. Kluwer Academic Publishers.

Hutchinson, J. M., Lo, A., & Poggio, A. W. (1994). A nonparametric approach to pricing and hedging derivative securities via learning networks. *Journal of Finance*, *31*, 851.

Iyer, R. D., Lewis, D. D., Schapire, R. E., Singer, Y., & Singhal, A. (2000). Boosting for document routing, In: *Proceedings of the ninth international conference on information and knowledge management*.

Kilian, L., & Inoue, A. (2004). Bagging time series models. *Econometric society 2004 North American summer meetings*.

Krogh, A., & Vedelsby, J. (1995). Neural network ensembles, cross validation, and active learning. *Neural Information Processing Systems (NIPS)*, *7*, 231.

Nelson, D. B. (1991). Conditional heteroskedasticity in asset returns: A new approach. *Econometrica*, *59*, 347.

Ohlson, J. (1980). Financial ratios and the probabilistic prediction of bankruptcy. *Journal of Accounting Research*, *19*, 109.

Opitz, D., & Maclin, R. (1999). Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, *11*, 169.

Osuna, E., Freund, R., & Girosi, F. (1997). Training support vector machines: An application to face detection. Conference on Computer Vision and Patterns Recognition (CVPR'97). IEEE Computer Society, 130.

Ratsch, G. (2001). *Robust boosting via convex optimization: Theory and applications*. Ph.D. thesis, Potsdam University.

Ratsch, G., et al. (1999). Regularizing AdaBoost. In: M. S. Kearns, S. A. Solla & D. A. Cohn (Eds), *Advances in neural information processing systems*, (Vol. 11, pp. 564–570). Cambridge, MA: MIT Press.

Ratsch, G., et al. (2001). Soft margins for AdaBoost. *Machine Learning*, *42*, 287.

Ridgeway, G. (1999). The state of boosting. *Computing Science and Statistics*, *31*, 172.

Schapire, R. E. (1992). *The design and analysis of efficient learning algorithms*. Ph.D. thesis, Cambridge, MA: MIT Press.

Schapire, R. E., & Singer, Y. (2000). BoostTexter: A boosting-based system for text categorization. *Machine Learning*, *39*, 135.

Schapire, R. E., Freund, Y., Bartlett, P. L., & Lee, W. S. (1998). Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, *26*, 1651.

Schittenkopf, C., Dorffner, G., & Dockner, E. J. (1998). Volatility prediction with mixture density networks. In: L. Niklasson, M. Boden & T. Ziemke (Eds), *ICANN '98 – Proceedings of the 8th international conference on artificial neural networks* (p. 929). Berlin: Springer.

Scholkopf, B., Tsuda, K., & Vert, J. -P. (Eds) (2004). *Kernel methods in computational biology (computational molecular biology)*. Cambridge, MA: MIT Press.

Schwenk, H., & Bengio, Y. (1997). AdaBoosting neural networks. In: W. Gerstner, A. Germond, M. Hasler & J.-D. Nicoud (Eds), *Proceedings of the Int. Conf. on Artificial Neural Networks (ICANN'97)* (Vol. 1327 of LNCS, pp. 967–972). Berlin, Springer.

Sun, X. (2002). Pitch accent prediction using ensemble machine learning. In: *Proceedings of the 7th international conference on spoken language processing (ICSLP)*. Denver, CO, USA, September 16–20.

Tino, P., Schittenkopf, C., Dorffner, G., & Dockner, E. J. (2000). A symbolic dynamics approach to volatility prediction. In: Y. S. Abu-Mostafa, B. LeBaron, A. W. Lo & A. S. Weigend (Eds), *Computational finance 99* (pp. 137–151). Cambridge, MA: MIT Press.

Tsay, R. S. (2002). *Analysis of Financial Time Series*. New York: Wiley.

Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM*, *27*, 1134.

Van Gestel, T., Suykens, J., Baestaens, D., Lambrechts, A., Lanckriet, G., Vandaele, B., De Moor, B., Vandewalle, J. (2001). Financial time series prediction using least squares support vector machines within the evidence framework. *IEEE Transactions on Neural Networks*, Special Issue on Neural Networks in Financial Engineering, *12*, 809

Vapnik, V. (1995). *The nature of statistical learning theory*. Heidelberg: Springer Verlag.

Vapnik, V. (1998). *Statistical learning theory*. New York: Wiley.

Witten, I. H., & Frank, E. (2000). *Data mining*. San Francisco, CA: Morgan Kaufmann Publishers.

Xiao, R., Zhu, L., & Zhang, H. -J. (2003). Boosting chain learning for object detection. In: *Proceedings of the ninth IEEE international conference on computer vision* (*ICCV03*). Nice, France.

Zakoian, J.-M. (1994). Threshold heteroskedastic models. *Journal of Economic Dynamics Control*, *18*, 931.