Prediction in Financial Markets: The Case for Small Disjuncts

VASANT DHAR, New York University

19

Predictive models in regression and classification problems typically have a single model that covers most, if not all, cases in the data. At the opposite end of the spectrum is a collection of models, each of which covers a very small subset of the decision space. These are referred to as "small disjuncts." The trade-offs between the two types of models have been well documented. Single models, especially linear ones, are easy to interpret and explain. In contrast, small disjuncts do not provides as clean or as simple an interpretation of the data, and have been shown by several researchers to be responsible for a disproportionately large number of errors when applied to out-of-sample data. This research provides a counterpoint, demonstrating that a portfolio of "simple" small disjuncts provides a credible model for financial market prediction, a problem with a high degree of noise. A related novel contribution of this article is a simple method for measuring the "yield" of a learning system, which is the percentage of in-sample performance that the learned model can be expected to realize on out-of-sample data. Curiously, such a measure is missing from the literature on regression learning algorithms. Pragmatically, the results suggest that for problems characterized by a high degree of noise and lack of a stable knowledge base it makes sense to reconstruct the portfolio of small rules periodically.

Categories and Subject Descriptors: H.0 [**Information Systems**]: General

General Terms: Algorithms, Experimentation, Measurement

Additional Key Words and Phrases: Machine learning, time-series prediction, financial markets, predictive modeling

ACM Reference Format:
Dhar, V. 2011. Prediction in financial markets: The case for small disjuncts. ACM Trans. Intell. Syst. Technol. 2, 3, Article 19 (April 2011), 22 pages.
DOI = 10.1145/1961189.1961191 http://doi.acm.org/10.1145/1961189.1961191

1. INTRODUCTION

Machine learning has seen successful published applications in a number of application areas that include the natural sciences, engineering, medicine, and business. The notable exception is financial markets, where published research on predictive modeling is scant. Some argue that to the extent there is predictability, there is little incentive to publish research that "works" since a good discovery can be very financially rewarding. Skeptics would argue that there is little scope for predictability since markets tend to be efficient. The reality is that there continues to be considerable interest and effort at finding structure and predictability in financial markets.

Financial time-series forecasting is difficult because of the inherently noisy nature of the domain. It is commonly known that most forecasting models do a very poor job in predicting future returns. It is also commonly known that return predictions of typical financial time-series forecasting models are usually very close to the mean because of

Author's address: V. Dhar, Stern School of Business, New York University, 44 West 4th Street, New York, NY 10012; email: vdhar@stern.nyu.edu.

DOI 10.1145/1961189.1961191 http://doi.acm.org/10.1145/1961189.1961191

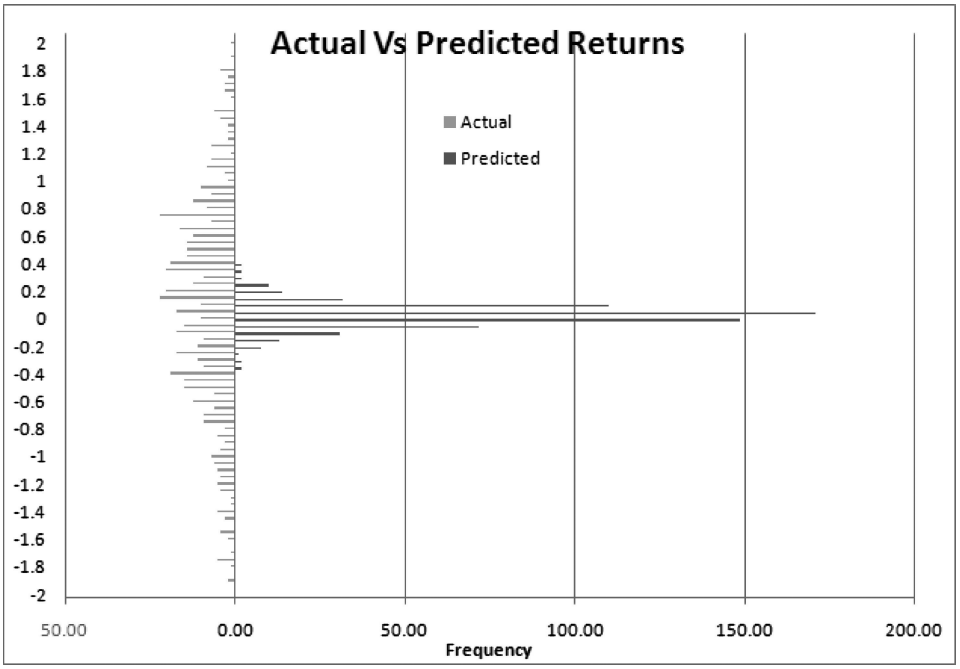

Fig. 1.

the inherent difficulty of making accurate forecasts, particularly for the larger values of returns. Figure 1 demonstrates this phenomenon showing actual returns from the S&P500 and predicted returns from a predictive model. Notice the large variance of actual returns relative to predicted returns, which means that a standard measure of error that compares actual to predicted returns (such as Mean Squared Error (MSE) or Mean Absolute Error (MAE) is dominated by the large actual values that contribute towards most of the error. This situation is particularly discouraging since it is the really the large future values that we really care about predicting well. Predictions around zero are relatively uninteresting in that we would not take any action in such instances.

Machine learning methods provide a potential solution to the problem of predictions being close to the mean because of their ability to easily partition the data into multiple sets and build a collection of models suited to each partitioning of the data. Some partitions can make large positive or negative predictions, while the others make predictions close to the mean, with the smaller partitions typically making the more extreme predictions. For example, Figure 2 shows a regression tree that recursively partitions a dataset into smaller subsets, with each subset representing the conjunction of conditions/splits that lead to it.

The dataset in this example consists of 1000 cases with mean of zero and standard deviation of 1. Notice that the rightmost partition covers only 3% of the dataset (30 cases out of 1000), but its mean is 1, indicating that on average the partition represents a significant positive return in the data. The partition represents the following rule.

IF "ret5 > 1" and "ret2 > 1" → Fret_mean=1

The independent variables are observed at the current time. These are derived from the original time series of prices. In the preceding example, one of the derived independent

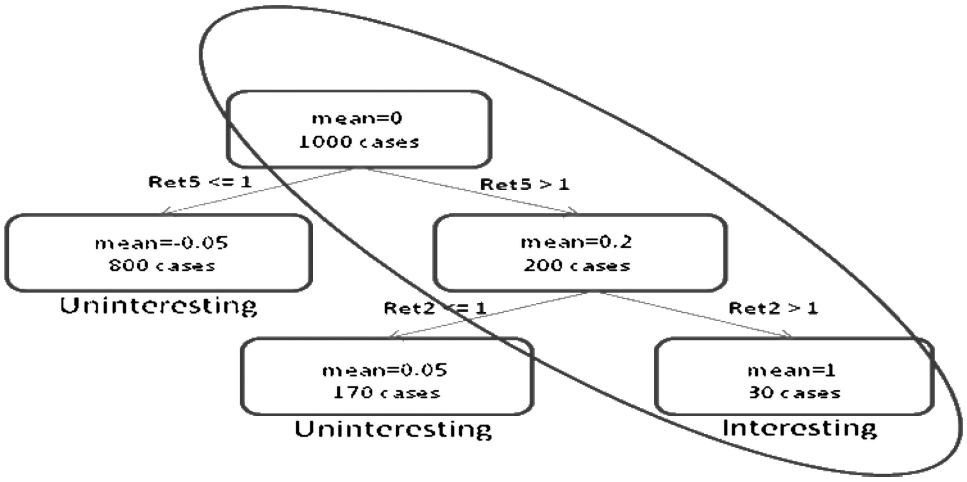

Fig. 2.

variables is the previous five period return (**Ret5**), and another is the previous two period return (**Ret2**). The prediction is the expected return in the next period, designated as **Fret_mean** in the previous rule. We can similarly follow the other paths in the tree and identify the corresponding rules that correlate combinations of the independent variables to the dependent variable. Some rules will have more extreme values of the dependent variable, which means that their means are distant from the overall mean of the data. The nodes on the left in Figure 1 are relatively uninteresting since the expected returns associated with these conditions are close to the overall mean of zero. These situations would result in no action on the part of a mechanical decision making system since the expected return in such situations is negligible.

The obvious drawback of this recursive partitioning approach is its tendency to find data partitions that make extreme predictions, but which "overfit" to the data in that they happen by chance alone to result in large absolute values for the dependent variable. This is known as "fitting a model to the noise" [Lee 1999]. The potential benefit, on the other hand, is its ability to model genuine but infrequently occurring situations as separate submodels.

A collection of low support situations such as the one in the bottom right node of Figure 2 are referred to in the literature as "small disjuncts" [Weiss 2000] since the learned patterns are alternative ways of making the same prediction. This article makes the case that good predictive models can be based on collections of certain types of small disjuncts.

A partitioning of the data as in Figure 2 is based on "axis parallel splits," in that all instances of groups produced by splitting the data have identical values for one of the variables. The advantage of this method is simplicity and high level of explainability of the partitions and an efficient computational method for generating them. The disadvantage is that for problems where the groupings can be expressed more naturally in terms of combinations of variables, the axis parallel method can use too many splits to approximate the true groupings. Other methods such as "oblique partitions" can produce simpler and more accurate trees in such cases [Murthy et al. 1994]. Indeed, as one can imagine, there are other more complex splits possible at each node, including nonlinear multivariate ones, with methods such as genetic algorithms applied to discovering linear combinations of variables [Cantu-Paz and Kamath 2000] or Boolean conditions of independent variables [Dhar et al. 2000]. In this article we have limited

ourselves to axis parallel splits and simple explainable decision trees for a number of reasons. First, this is probably the most widely used and best understood data mining method and tends to work well on a remarkable range of problems. Second, our approach to financial prediction is to assemble many "simple" disjuncts of low complexity that are easily interpretable. Finally, our objective is to test whether this simple approach to predictive modeling has promise. If it does, future research can build on it and improve predictive accuracy using potentially more complex splitting criteria.

The remainder of this article is organized as follows. We first review the literature on small disjuncts. We discuss complexity, a key parameter in machine learning algorithms, as it applies to small disjuncts. We them describe the data and measure of performance. This is followed by a description our learning algorithm (a standard tree induction algorithm called CART [Breiman et al. 1984]) that is applied to various in- and out-of-sample datasets within and across time, and an interpretation of the corresponding results. We also define a simple construct called *yield* that measures the percentage of performance that a learned model realizes relative to an ideal model. We conclude with a discussion of the merits of using ensembles of small disjuncts to financial time-series forecasting. The basic argument is that for noisy problems that do not have a central structure, a distributed approach to forecasting works well when we keep the disjuncts simple. This approach is also suited to the problem because a system takes action only when it sees interesting opportunities and is silent at all other times. Apparently, this is also a characteristic of successful human traders.

2. SMALL DISJUNCTS: PRIOR RESEARCH

A small disjunct is a pattern that covers few training examples. In other words, the "coverage" or "support" of the pattern is limited, such as the lower rightmost node in Figure 2. A collection of small disjuncts, however, can cover a large part of the data, which means that they can have a significant impact on the overall quality of a learned model. A pattern can be represented using a rule, a neural network, a support vector machine, a regression model, or other representation that relates independent variables to a dependent variable. In Figure 2, the pattern is a rule.

We are agnostic about the representation of small disjuncts although we represent them as decision rules. A rule consists of a left- and right-hand side. The left-hand side is a conjunction of Boolean conditions, where each condition is expressed in terms of variables, relational operators, and constants. An example of a condition is "the 5 day return is greater than 1." The term "5 day return" is a standard measurement of the time series, like a "feature" in standard data mining applications. The right-hand side consists of an expected return that results by "applying" the left-hand side to the time series. This is the dependent variable we are trying to predict in the future.

Rules vary in several key ways. *Support* measures how much of the dataset the rule covers. Specifically, if N is the total number of examples in a dataset, then for a rule of the form $A \rightarrow B$:

$$\begin{aligned} \textbf{Support} &= (\text{Number of cases satisfying A})/\text{N} \\ &= \text{p(A)}. \end{aligned} \tag{1}$$

The right-hand side, B, is based on the type of problem. For classification problems, it specifies the majority class corresponding to the cases satisfying A. For regression problems, it is a numerical value, typically the average of the cases satisfying A, and is in Figure 2.

There has been a fair amount of research on small disjuncts in the machine learning literature. Much of it suggests that they produce a disproportionately large percentage of errors but collectively cover enough cases that they can't simply be ignored or

discarded [Holte et al. 1989; Quinlan 1991; Pazzani 1992; Danyluk and Provost 1993; Weiss 1995; Weiss and Hirsh 2000]. For example, Holte et al. showed that disjuncts that cover only 12% of the cases in the training set account for 57% of the errors. They noted that the "generality bias" used by learners like ID3 works well for large disjuncts but not small ones. On the other hand, a "specificity bias" produces lower error rates for the small disjuncts, but it increases the error rates for the large ones, making no difference to overall accuracy of the learner. Ting [1994] refined this approach by combining two learners: if a case is covered by a large disjunct, it is classified by a generality bias learner, otherwise by one with a specifity bias. The drawback of this method is the difficulty of defining practically (and in advance) what is meant by a "small" or "large" disjunct.

Quinlan [1991] proposed an interesting method for reducing the error rates for small disjuncts. Noting that class distributions are often skewed, disjuncts predicting the minority class are likely to have higher error rates. Quinlan [1991] incorporated these prior probabilities of the target class in classifying examples. Quinlan's method calculated the prior probabilities only on those training examples that are "close" to a small disjunct, meaning that they don't satisfy only *one* of the conditions of the pattern. His method produced lower error rates than the naïve model. In a similar vein, Pazzani [1992] assigned lower "reliability" to small disjuncts, effectively lowering their impact on the overall accuracy of a classifier.

A number of researchers have tried to provide explanations for when and why small disjuncts tend to provide erroneous predictions. Danyluk and Provost [1993] pointed to the difficulty of separating systematic noise from cases that are truly exceptional. Along these lines, Weiss and Hirsh [1998] showed that as class noise increases so does the number of small disjuncts, although with very high levels of noise the errors become more concentrated in the larger disjuncts. In a later paper, Weiss and Hirsh [2000] showed that as the size of the training set increases, errors tend to occur in the small disjuncts.

It is interesting that all the research on small disjuncts has been entirely on classification problems. What is notable about financial market prediction is that it represents a *regression* problem, not a classification one in which the dependent variable is discrete, often binary. When comparing models, the standard statistics used to evaluate classifiers such as "true positives" (which may be thought of as the proportion of winners) and "false positives" (proportion of losers) are meaningless. The quality of a system is driven by its ability to predict large values , even at the expense of being wrong on the small ones. A system can be profitable even if it wins a third of the time, and it can be unprofitable even it wins 90% of the time. The magnitude of winners and losers can be more important than their relative percentages. This makes regression problems inherently different from classification problems in terms of how performance of a model is calculated. We return to this issue after presenting the results.

An additional point of interest in understanding and interpreting prior literature on small disjuncts is that the problems studied, those from the UCI database and synthetic problems, have had a fair degree of known or controllable structure where for the most part the large disjuncts captured most of this structure. It is not surprising therefore that the small disjuncts would account for a larger proportion of the errors.

Last but far from least, the complexity of small disjuncts was not explicitly controlled for in prior studies. The small disjuncts typically generated had a large and variable number of conditions. Without a control on model complexity, it has been impossible to know definitively whether small disjuncts perform better if their complexity is limited and held constant. As we discuss shortly, we keep complexity as low as possible and fixed across disjuncts, making sure that the learner generates disjuncts that do not exceed a specific threshold of complexity.

3. COMPLEXITY

It is well known that if we let a learner partition the data without regard to the number or types of conditions it imposes on the independent variables, it will generate models that predict every example in the training set perfectly, but these models will fail miserably on data they have not encountered. In the example of Figure 2, if the left-hand side were something like

IF "1.01 > ret5 > 1" and "0 > ret2 > 0.01" and "2 > ret3 > 2.1"

we could probably regard it as too "complex" because of the large number of conditions, in this case six, and the fact that each condition "pinches" a very narrow range of values of the independent variables. Such disjuncts will *memorize* the data instead of *generalizing* from them and not perform well on future data. For this reason, it makes sense to limit the complexity to the extent possible while still allowing the disjuncts to represent interaction effects.

It is important to be able to represent interaction effects in financial time-series forecasting because these allow combinations of variables to be conditional. In Figure 2, for example, the rightmost leaf node says that high future returns can be expected when **Fret5** exceeds 1, conditional on **Fret2** also exceeding 1. This adds complexity to the model relative to simple additive models, but it gives it the power to model nonlinearities.

Complexity has been studied extensively in the machine learning literature from a number of angles. Linear models are considered "simple" in terms of structure compared to nonlinear models. The former are considered "high bias" since they impose a simple structure to which a learned model must conform in making predictions. In contrast, nonlinear models such as trees are considered more complex or "low bias" since they do not impose a structure on the model a priori (see Perlich et al. [2003] for a comparison of the two types of models).

Because nonlinear models allow a learner considerable degrees of freedom in fitting a model to data, a considerable amount of attention goes into specifying and controlling for it in learned models depending on the representation. This includes the VC dimension in support vector machines [Vapnik 1995], hidden nodes in neural networks [Poggio and Girosi 1989], and the number of conditions imposed on inputs to compute the output in decision/regression trees [Arora and Barak 2009].

Disjunct complexity has not been controlled for explicitly in prior studies, making it somewhat difficult to compare the results and to assess the impact of disjunct complexity on predictive accuracy. As we explain more fully next, we control explicitly for complexity.

In this study, we define disjunct complexity along two dimensions. The first of these is the number of conditions, which is a standard measure for trees and rules. The second is the number of variables allowed in the disjunct. Both dimensions deal with interaction effects, but the latter allows us to control for the dimensionality of the search space that the learner can consider. The famous expression "curse of dimensionality" coined by Bellman [1957] illustrates the exponential increase in examples required in a sample for every dimension (in this case, variable) that is added to the search space.

Our objective specifically is to keep the small disjuncts as simple as possible by limiting the interaction effects allowed between variables, thereby minimizing the "curse of dimensionality." To keep things simple, we therefore limit the small disjuncts to dyads, that is, involving two variables with axis parallel splits as shown in the example in Figure 2, with "Ret5 > 1" and "Ret2 > 1." These constraints keep the variable interaction effects as simple as possible while still giving the learner some room to model nonlinearities.

The dyad in the example in Figure 1 has one major limitation, namely, its inability to represent ranges of the form "0 > Ret2 >1." Considering the fact that correlations between variables can be nonlinear and hold within specific ranges, this is a severe hindrance. For this reason, we allow one other condition to be imposed in the dyad on one of the variables in terms of the standard relational operators. This representation enables us to learn rules of the form

$$\text{"Ret5} > 1\text{" and "0} > \text{Ret2} > 1\text{"} \rightarrow \textbf{Fret\_mean=0.3.}$$

In summary, we keep disjunct complexity as low as possible, but enabling a learner to capture conditional interaction effects. Limiting ourselves to dyads with a limit of three conditions for all disjuncts also keeps model complexity below a specific fixed threshold. By controlling for model complexity we can evaluate the impact of disjunct support size on performance. Do smaller disjuncts perform better as long as complexity is low? We can answer this question by varying a parameter, namely support, and testing for predictive performance. Before we do this, let us describe the data and the measure of performance.

4. DATA AND MEASUREMENT

An observation (datum) consists of a pair (Zfret, X), where X is a vector denoting the current state and Zfret is a continuous value we are trying to predict.

How should we measure the performance of a predictive model? A commonly used measure of performance in Finance is "risk adjusted return," which is typically a return divided by the risk involved in achieving it. Risk is generally measured in terms of volatility of returns, often calculated as a standard deviation of a return series.

A suitable time interval needs to be used to compute the volatility of each instrument for normalizing returns. For each instrument, performance is measured as follows: given a period of returns of the instrument $\{r_{i-N}, r_{i-N+1,} \ldots r_{i-2,} r_{i-1,} r_{i,} r_{i+1}\}$, where $i$is the current day, the risk adjusted future period return is defined as

$$\text{Zfret} = r_{i+1}/\text{Stdev}(r_{i-N}, r_{i-N+1,} \ldots r_i), \tag{2}$$

where N is a parameter used to standardize the returns. We use N=250, which uses roughly one year of historical data to calculate the volatility of returns for the instrument. In practice, N must be chosen to provide a representative distribution of returns.

Expressing performance in terms of risk adjusted returns makes it possible to compare all instruments using an identical measure. It lets us pool data across instruments. A volatile instrument, for example, would generate returns of high magnitude relative to one where volatility is low. In order to achieve the same dollar return per trade, one would therefore invest a larger dollar amount in the latter. We express performance of all instruments in terms of risk adjusted performance for learning and evaluation. The instruments chosen in this study were the most liquid global equity indices and include the S&P500, Dow Jones 30, Russell 2000, S&P Midcap 400, Nasdaq 100, Nikkei 225, Hang Seng, EuroStoxx 50, FTSE 100, CAC 40, IBEX 35, and the DAX 30. Daily open, high, low, and close prices were used from the date of inception of the futures contract for the instrument up to May 22, 2008. The total dataset consists of 44,473 records and can be found at the following URL.

http://w4.stern.nyu.edu/emplibrary/allData.equ2.intime_v69_posted.xls
http://w4.stern.nyu.edu/emplibrary/allData.equ2.outoftime_v69_posted.xls

The vector X consists of a standard set of indicators that represent the "state" of the market every day for each instrument [Kauffman 2004]. In this study, 68 indicators were included that are described in the Appendix. The properties of the indicators and

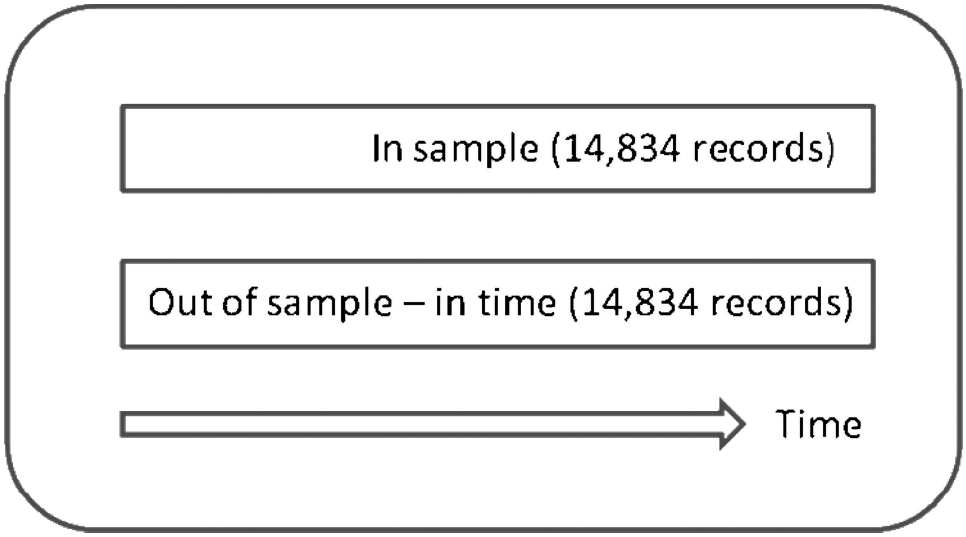

Fig. 3.

criteria for selection are beyond the scope of this study and peripheral to the focus of this article. For a detailed analysis of these and other typical indicators used to describe financial time series, the reader is referred to Kauffman [2004] and Achelis [2000].

For the first set of experiments, the data were partitioned randomly into two equal parts 50 times, covering the same length of time as illustrated in Figure 3. In this dataset, end date is June 30, 2003. The start date varies by instrument, depending on when it began trading. The random partitioning of the data provided 50 in-sample and 50 out-of-sample datasets consisting of an identical number of records (14,834 each). The learning algorithm was applied to each of the 50 training sets, and performance was measured for each learned model on the corresponding out-of-sample dataset.

5. LEARNING ALGORITHM

The learning algorithm applied to the in-sample data is CART [Breiman et al. 1984] which takes as input pairs (Zfret, X) of the form described in the previous section and generates trees of the type in Figure 2. Each path through the tree to a leaf node is a rule of the form A $\rightarrow$ B, where A is a conjunction expressed in terms of at most two independent variables (dyads), relational operators, and constants and B is the expected value of the dependent variable corresponding to the cases covered by A. Since input was restricted to consider only dyadic combinations from among the independent variables for reasons expressed earlier, for N variables, the learner was therefore invoked ${}^{N}C_{2}$times, thereby generating N.(N-1)/2 trees.

The first constraint applied to the learner is on the depth of trees. We limit the number of splits allowed to three in order to limit disjunct complexity. Since each terminal node represents a rule, the upper bound on the number of rules per tree is 2^{S}, where S is the number of splits or conditions allowed. This results in an upper bound of 2^{S-1}.N.(N-1) rules.[1] For 68 variables and the limit of three splits used in this study, this upper bound is 18,224 rules for each threshold of support.

The second parameter was used to control the unevenness of the split permitted. Depending on the data, it is not unusual for the induction algorithm to create highly

[1]The actual number of rules generated are a little lower than this upper bound since not all trees need to be balanced.

imbalanced splits, with only a single or few cases going into one partition. This is known as an "end-cut split" [Buja and Lee 2001]. These overly uneven trees generally result in nonactionable or highly fitted rules, although it has been observed that in less pathological cases, uneven splits can result in interpretable rules that represent useful outliers in the data [Torgo 2001; Buja and Lee 2001]. While we want small disjuncts to be created, we need to ensure that the learner generates rules that satisfy a specified minimum level of support. This requires the "minimum split" parameter (which we call **M**) to exceed the minimum percentage level of support (which we refer to as **L**) required as defined in Eq. (1). This condition guarantees that the first split will result in partitions that exceed the minimum required level of support for a disjunct.

It should be noted that trees generated using the different support thresholds (and fixed number of splits S) can vary considerably, depending on M and L. Consider that S splits with a threshold of M will result in a minimum support of M^S for a leaf node.

The support threshold L in our experiments was varied between 0.25% and 10%. Going any lower would not be meaningful considering that a 0.25% coverage would cover too few buying and selling cases to generate meaningful statistics. At the other end of the spectrum, a 10% support as the upper bound also seems reasonable since performance at higher levels of coverage would begin to resemble the overall average, namely, the market.

Our ensemble approach to the prediction problem through collections of trees is similar to that of random forests [Breiman 2001] and other similar methods that try to avoid the overfitting or bias that can result from single decision trees. Random forests do this by using bootstrap samples and considering a very small subset of the independent variables for consideration for every split. In our algorithm, we restrict this consideration to two variables and consider the combination of two variables at a time exhaustively. We selected our method because we wanted to give all variables equal opportunity in the rule generation process. These variables have been noted historically to be relevant based on past correlations with future returns, so some domain knowledge has gone into the variables which we would like to see reflected in the trees generated by the learning algorithm.

Rules are extracted from the trees as shown in Figure 2, with a path from root to leaf being a rule, represented as Boolean conditions over dyads in X. Each rule has a score, namely, its expected values of Zfret which is computed from the cases that satisfy the conditions specified by the rule. The rules are ordered by in-sample performance. This focus on nodes (rules) as the unit of analysis instead of the entire tree is important. It enables us to solve two important problems simultaneously.

The first advantage of rules as the unit of analysis is that it provides a good way to deal with the problem of model selection, which is a thorny one in machine learning. The problem is one of selecting one or more models from a large set of competing models. There are several measures for doing this, such as AIC [Akaike 1974], which typically rely on a basic measure such as MSE or MAD for assessing model error. The reality, however, is that most trees produce very similar error rates when applied to the entire dataset for reasons pointed out earlier, namely, that the majority of the error comes from the larger values of the dependent variable when the model predicts close to zero (from the nodes with high support and average prediction close to the mean of zero). Because competing models (in our case, trees) produce very similar error rates, ranking them reliably is difficult. In contrast, as illustrated in Figure 2, the means of individual nodes (especially those corresponding to small disjuncts or low support) are typically quite different from the overall mean of the entire dataset and from each other. In effect, we get a wide range of scores at the node or rule level. By comparing and ranking these, we get a well discriminated ranking of rules based on in-sample performance. We can then pick a subset of these rules as our model.

Focusing on the best rules (nodes) as the decision rules from across the large set of trees produced by the learning algorithm also solves a problem that is especially relevant to financial prediction, namely, that of making a decision to buy ("go long") or sell ("go short") only when there is a strong signal, and being agnostic otherwise. Unlike problems where a system must decide on every case it sees, in financial forecasting it often makes sense not to act because the expected return is close to zero. This is because of the inherent noise in the problem. By selecting only the "interesting" rules we have a decision making system that is opportunistic, acting only when the conditions are appropriate and being agnostic otherwise.

The ordered list of rules contains the largest positive scores (Zfret) at the top and the largest negative scores at the bottom. The top T rules are considered to be "long" rules. Whenever the left-hand side for a rule is satisfied for an instrument, a trading program would buy the instrument and sell it at the end of the next period. The bottom T rules are considered to be "short" rules. Whenever the left-hand side for a rule is satisfied for an instrument, a trading program would sell the instrument and buy it at the end of the next period.

The top T and bottom T rules make up the decision rules employed by a trading program. This type of top/bottom quantile structure is a common method in industry for how trading programs are assembled to make predictions in both directions. In our experiments T was set to 100.

We recognize that the simple selection procedure of the top T and bottom T rules is suboptimal from a portfolio optimization perspective, since it ignores correlations among the rules. If one takes correlation into account, we are faced with a massive combinatorial optimization problem. While there are several heuristics that could be used, we ignore this problem here since it introduces additional complexity into the experimentation. While this would be essential in formulating a real trading strategy, it isn't necessary for the comparative analysis of interest in this article.

6. RESULTS

Before running the main experiments, a baseline was established where the learner was allowed to generate rules without any limit on the complexity of rules, namely, the number of conditions used.

As expected, the result from this baseline experiment was zero correlation between rules' in-sample and out-of-sample performance. This result is consistent with prior literature on small disjuncts: if we don't control for their complexity, they overfit the data and generate large errors on out-of-sample data. In this case, their predictions are no better than random.

Next, the learning and testing was performed 50 times based on different random partitionings of the data (no duplicates in an in-sample or out-of-sample pair). For each run, the overall performance is calculated as follows. Each rule is applied to each case that satisfies its left-hand side and the return for that case is noted. The performance of a rule is the average value of the dependent variable across all cases that satisfy the rule. The performance of the set of the T long rules is the average across all the long rules. The performance of the set of the T short rules is computed similarly. Finally, the overall performance of the system of T long and T short rules is the difference of their respective averages.

Figure 4 shows the results on the in-sample and out-of-sample datasets for varying levels of support up to 10% for the 50 runs. The error bars show one standard deviation in performance for the 50 runs for each level of support.

Several things are noteworthy about the results in Figure 2. First, as expected, the in-sample performance increases with reduced support. However, it plateaus out at half

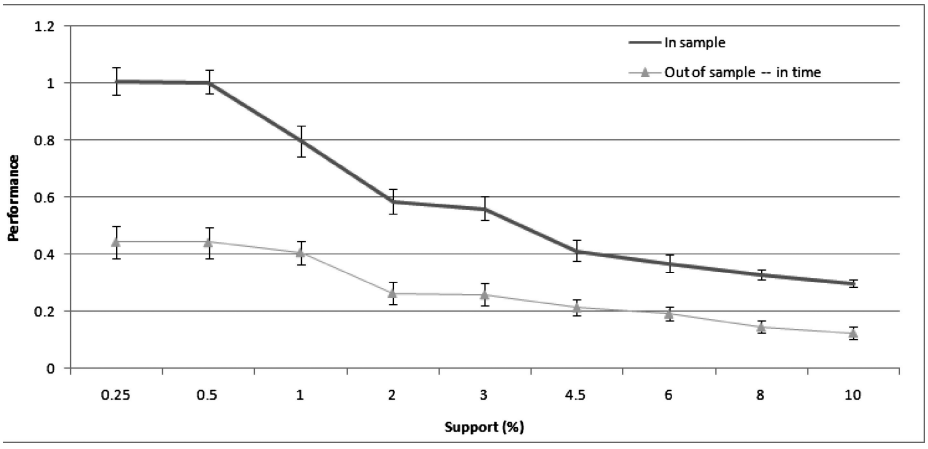

Fig. 4.

a percent, suggesting that the learner derives little additional benefit from making patterns more restrictive.

Secondly, and perhaps most significantly, the out-of-sample performance mirrors the in-sample performance for all levels of support. Surprisingly, it gets better monotonically as support gets smaller. We do not see the "inverted U" that one sees when model complexity is varied [Breiman et al. 1984]. Keeping the complexity low does appear to avoid gross overfitting that occurs when complexity is not controlled. This is a significant and novel result. It argues in favor of using Occam's razor for constructing ensembles of small disjuncts. We shall return to the significance of this result in the discussion section of the article.

Thirdly, the results show larger absolute performance drop for smaller disjuncts, indicating that the process of statistical induction does indeed result in more overfitting for smaller disjuncts than the larger ones. However, even though small disjuncts degrade more than large ones on out-of-sample data, they still perform significantly better. This is what we care about from a predictive standpoint.

Finally, there is a significant difference in performance between the in-sample and out-of-sample data at all levels of support. This is not surprising. It is common knowledge that the process of statistical induction leads to some invariable "memorization" of the data where some of the discovered patterns are really noise in the data. What is notable here is the extent of degradation, which is an indication of the inherent noise in the prediction problem.

It is important to recognize that in the absence of special knowledge about the problem, it is impossible for the machine to distinguish between the real and coincidental patterns, that is, the ones that represent memorization (or "noise") versus those that represent generalization (or "signal"). And even for the patterns representing "signal" there will be considerable variance in performance at the individual pattern level. The expectation, however, is that there is some degree of generalization captured in the aggregate collection of patterns.

Since some degree of memorization is inevitable in induction, we should expect the out-of-sample performance of a collection of patterns to be lower than the in-sample performance. For classification problems, the performance of a learner is defined using a confusion matrix which counts true and false positives and true and false negatives. For regression problems, however, evaluation depends on the magnitudes of the correct and erroneous predictions. As we mentioned earlier, systems that win a majority of the

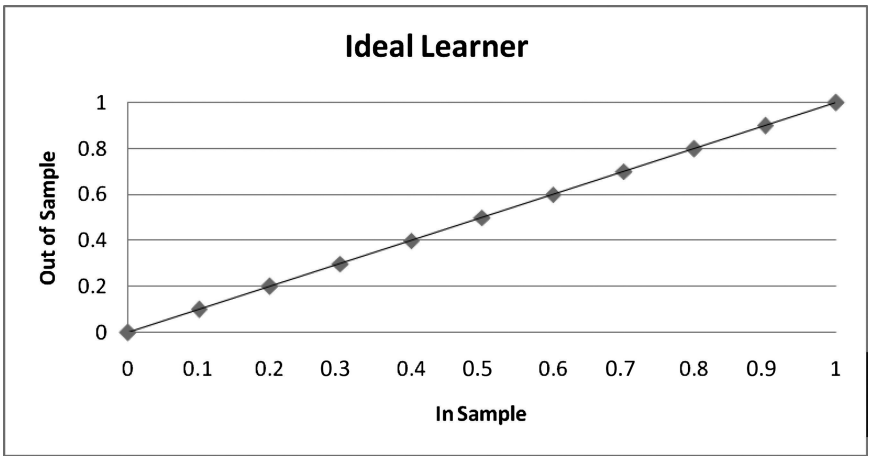

Fig. 5.

time can be unprofitable while those that win a minority of the time can be profitable. Performance is driven by the sizes of the winners and losers.

The ensemble of disjuncts provides a natural way to measure the expected degradation on future data in percentage terms. To understand how, consider first what an "ideal" learned model would look like, one where the degradation due to inductive generalization is zero. Such a learner would produce a zero intercept scatter plot as in Figure 5. The out-of-sample performance of this learned model matches its in-sample performance exactly by producing an identical value for the continuously valued dependent variable for both in-sample and out-of-sample data. Such a situation would occur if the model were specified correctly including all relevant variables and with the correct functional form, in which case it should perform similarly in and out-of-sample.

In practice, however, we typically do not know the perfect functional form, nor do we include all relevant variables. Accordingly, it is more common to observe scatter plots as in Figure 6. This particular plot shows the in- and out-of-sample performance of the top 100 long rules at the 1% support level. As we see, the in-sample data have a tight distribution, whereas the variance out-of-sample is much higher. Some patterns can actually do better out-of-sample, but the majority of them do worse.

Note that the regression line in Figure 6 shows the fit between in- and out-of-sample data, and its intercept is nonzero (–0.3146). This says that an in-sample performance of zero should result in a negative out-of-sample performance, of –0.3146. This can be viewed as the "bias" of the learned model. This interpretation of the regression model is clearly unnatural for certain types of extreme cases, such as in the limit when support approaches zero and no trading occurs. In such a case, when in-sample performance is zero, so should the out-of-sample performance. Consider a line going through the origin, which would represent this limit situation, with zero performance both in and out-of-sample . Forcing the line to go through the origin solves two problems simultaneously. First, we get a natural measure of degradation in percentage terms which we call yield. Specifically, if the 45 degree line passing through the origin represents a 100% yield as with the ideal learner, lower slopes represent correspondingly less than perfect fit between in- and out-of-sample performance. In effect, the slope of the line is the meaningful measure of the goodness of the learned model (in theory, it is possible for the line to be steeper than 45% where the learner has somehow managed to perform even better out-of-sample in the aggregate This should be extremely rare and correspond to a statistical fluke).

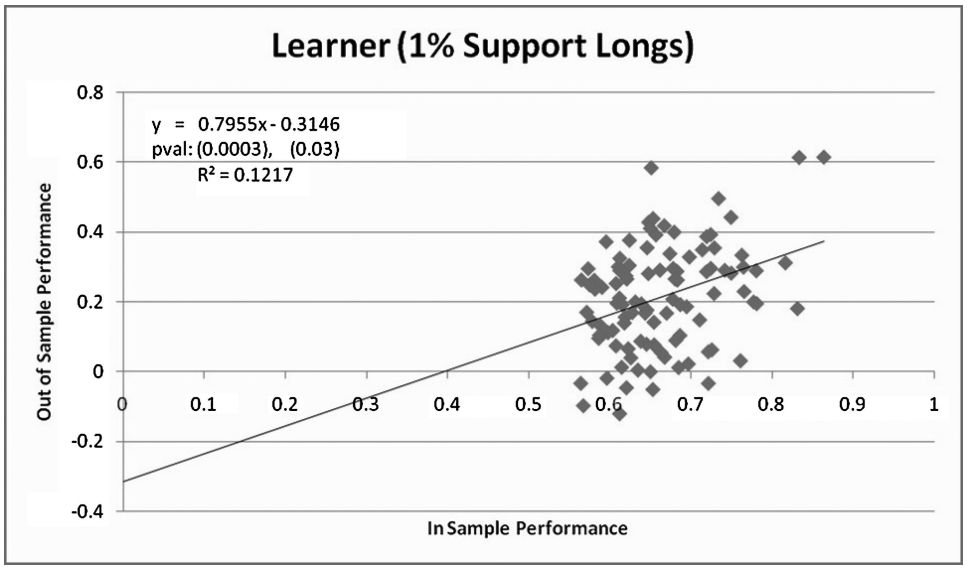

Fig. 6.

A nonzero intercept term also leads to a more serious problem with interpreting the regression that is best illustrated through an example. Specifically, imagine a scatter plot that is circular but tightly distributed around high values for both in- and out-of-sample performance. The R-square for such a line might be zero, in which case the regression line would be horizontal. But the learned model would actually be good one, measured by the high average out-of-sample performance of the disjuncts as a whole. Allowing an intercept would yield a flat line with a low R-square despite the high out-of-sample performance. In contrast, the slope of the line passing through the origin would represent the performance of the disjuncts corresponding to the scatter plot relative to the ideal, the 45 degree line.

In summary, the slope or tangent of the zero intercept provides a natural measure of yield, with flatter lines corresponding to lower yields. Figure 7 shows the line passing through the origin, corresponding to the data in Figure 6. For this ensemble, the yield is just over 32% relative to the ideal learner.[2] The R-square for this case is meaningless and does not represent fit and can therefore be ignored.

Unlike measures of model error such as MSE and MAD which are dominated by the large values and not easy to interpret, the yield measure has a clean and useful interpretation, namely, percentage degradation in relation to an ideal learner that a model can be expected to realize.

The preceding calculation of yield measures the degradation from inductive generalization based on the inputs provided to the learner, since the in-sample and out-of-sample data came from the same distribution. However, the same concept can be used to calculate the degradation of a model on future data which might come from a different distribution. Indeed, in time-series forecasting, one typically implements

[2]The limitation of this point estimate of yield, of course, is that it does not consider explicitly the dispersion of the data.

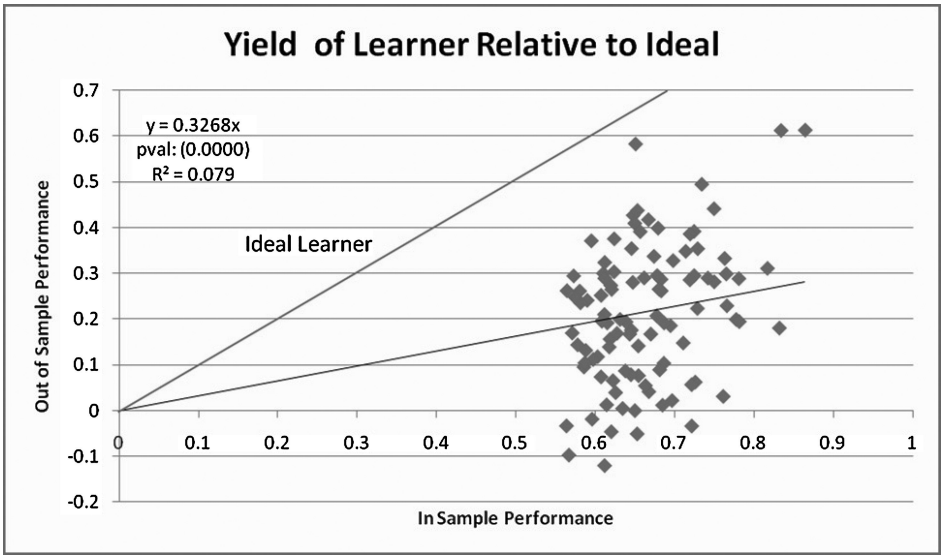

Fig. 7.

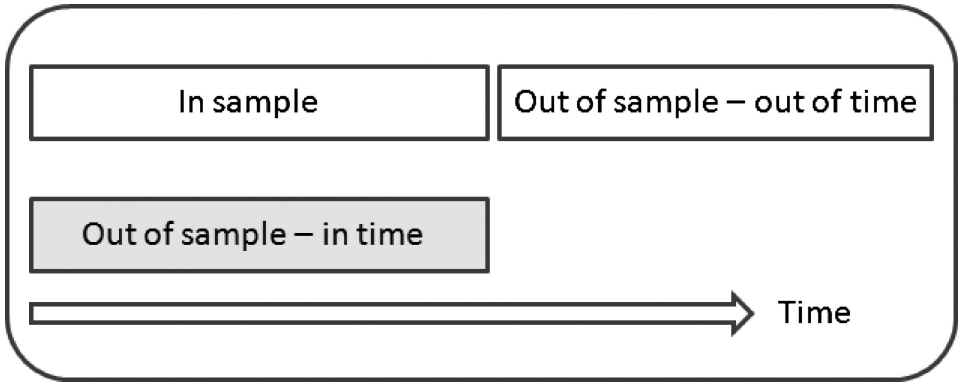

Fig. 8.

learned models for use on future data. In this situation, we should expect additional degradation in performance.

7. LEARNING AND TIME-SERIES FORECASTING

The purpose of the second set of experiments was to isolate and quantify the degradation arising from the fact that the future is always different from the past. This set of experiments assesses whether small disjuncts have any predictive ability.

In order to quantify this second type of loss, the learned patterns were tested not on data covering the same period as the data on which they were discovered, but on *future* periods. Figure 8 illustrates how the data were selected. This dataset (labeled "out-of-sample out of time" in the figure) corresponds to the time period between July 2003 and May 2008 and consists of 14,805 records. In this experiment, we generated the out-of-time performance by applying the rules learned on the in-sample data to the

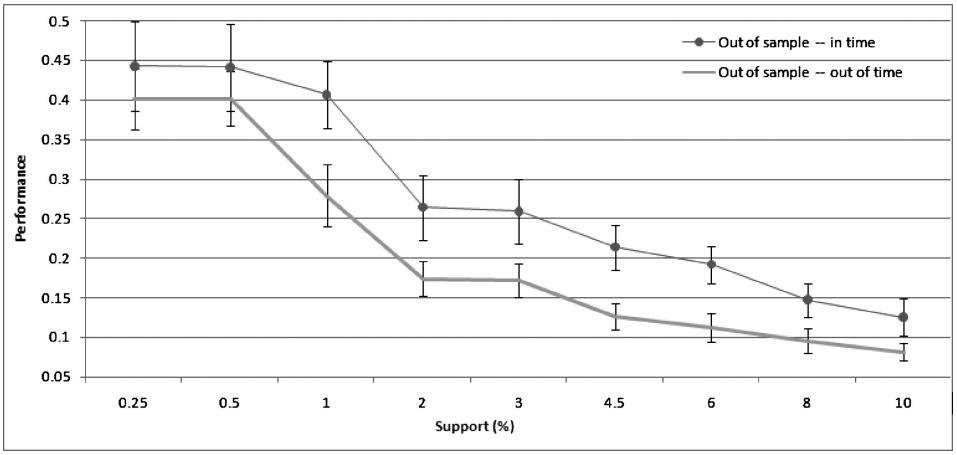

Fig. 9.

"out of time" data. The results from evaluating the learned patterns out of time are shown in Figure 9. Each level of support shows the average performance of the patterns with the corresponding error bars similar to Figure 4. In order to avoid clutter, we do not show the in sample performance again (for which the reader is referred to Figure 4), and only compare the two out-of-sample performances: in time versus out of time. (The lower line from Figure 4 is the upper line in Figure 9.)

Interestingly, the out-of-sample performance out of time also mirrors the out-of-sample performance in time. As expected, the performance out of time is consistently worse. The yield achieved out-of-sample can be measured as before, by plotting the in sample performance versus the out-of-time performance and calculating the slope of the zero intercept regression line. We do not show this plot since it adds little to the discussion, but what is of interest is the fact that the degradation on future data both in absolute and percentage terms is not worse than that due to inductive generalization. Indeed, it is lower in percentage terms, especially for the lower levels of support. This is encouraging since it suggests that the collection of small disjuncts do indeed have some predictive power that can persist over time.

In concluding this discussion on out-of-time degradation, it is natural to expect that the degradation can be severe enough to result in a *negative* performance in the future. Indeed, the plot in Figure 10 (the performance of the "short ensemble" at the 1% support level) shows such a situation. In contrast to Figure 7 where the "long ensemble" at the 1% support level had a roughly 32% yield, this time the overall yield is negative, showing that the learned patterns that make negative predictions for Zfret perform poorly in the future.

The previous result illustrates two interesting concepts that are worth mentioning briefly. First, constructing a learning system on noisy financial time-series data can result in models that perform poorly in the future. A learning algorithm will lose performance not just in the process of induction, but also if the future turns out to be sufficiently different from the past, on which the model is based. Second, the figure illustrates a peculiarity of financial equity markets, namely the difficulty of finding good "short" models, because markets trend up most of the time, so a short seller is on average swimming against the current. This is a well known phenomenon in equities markets.

Would anyone implement a model with a potential negative yield? Normally, one would not. However, many portfolio managers are not comfortable with implementing

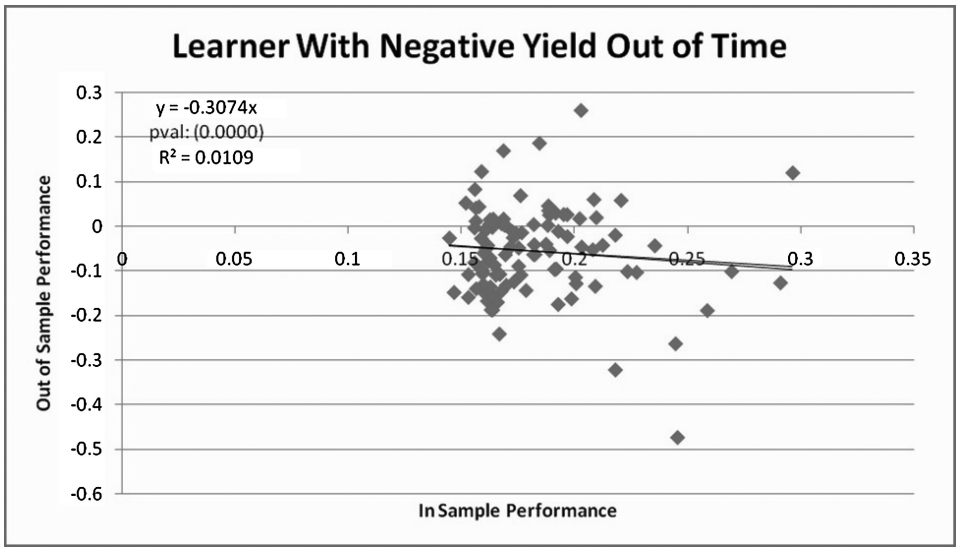

Fig. 10.

Table I.

support%	0.5	1	2	3	4.5	6	8	10
Dow 30	0.61	0.53	0.10	0.08	0.00	0.05	0.03	0.00
CAC40	0.19	0.29	0.12	0.19	0.13	0.04	0.14	0.08
DAX30	−0.10	−0.15	−0.18	−0.20	−0.19	−0.18	−0.02	−0.04
FTSE100	0.36	0.31	0.11	0.17	0.15	0.05	0.10	0.02
Hang Seng	0.19	0.28	0.22	0.26	0.22	0.16	0.13	0.05
Midcap 400	0.29	0.20	0.05	−0.05	0.00	−0.04	0.04	−0.03
IBEX35	0.30	0.23	0.05	0.13	0.06	−0.02	−0.02	−0.08
Nasdaq100	0.31	0.26	0.19	0.11	0.04	0.09	0.07	0.07
Russel 2000	0.31	0.24	0.24	0.18	0.16	0.11	0.14	0.06
SP500	0.60	0.65	0.27	0.29	0.23	0.22	0.15	0.17
Nikkei 225	0.26	0.35	0.12	0.02	−0.02	0.00	0.05	−0.04
EuroStoxx 50	0.20	0.19	0.14	0.12	0.10	0.09	0.07	0.08
Average	**0.29**	**0.28**	**0.12**	**0.11**	**0.07**	**0.05**	**0.07**	**0.02**
Max	**0.61**	**0.65**	**0.27**	**0.29**	**0.23**	**0.22**	**0.15**	**0.17**
Min	**−0.10**	**−0.15**	**−0.18**	**−0.20**	**−0.19**	**−0.18**	**−0.02**	**−0.08**

The table shows means only to avoid clutter. The average standard deviation was approximately 0.1 at the higher levels of support and approximately 0.2 at the lowest levels of support.

"long only" models, assuming that the market will go up at all times. For this reason it is typical to "hedge" long exposure to the market through a short strategy which may lose money on average, but nevertheless provides protection during those times when the market drops unexpectedly.

As a final check on the generality of the preceding results, we tested the learned models on each of the *individual* equity indices out of time. The results are presented in Table I which shows average performance for each equity index, long minus short, as before. Not surprisingly, there is considerable variance in results, with significant difference in performance across the individual instruments. However, other than the DAX30 for which the learned models do not work at any level of support, there is a pattern of better performance for the smaller disjuncts, similar to what we observed at the overall asset class level. It is also notable that there is a considerable drop in performance above the 1% level of support.

The previous results suggest that the results at the asset class level are not driven by outliers, but from across the asset class. They bolster the earlier findings, suggesting that the patterns described by the small disjuncts exist at the individual instrument level, and not based on outliers.

It is worth concluding this section with the question we raised at the outset, namely, whether ensembles of small disjuncts are a useful predictive model for problems that have a high level of noise. Interestingly, our results run counter to previous research, which was primarily on classification problems, and where complexity was not controlled for explicitly. We find that their absolute performance is better than that of large disjuncts, and this difference is statistically significant.

8. DISCUSSION

In many problem domains, there is usually a "main" or "first order" model that explains a bulk of the cases in the data, and the minority of cases that are hard to classify are treated as noise or exceptions that may require a "second order" model to classify them correctly. When rule induction algorithms are applied to such problems, it is not surprising that the small disjuncts would produce a disproportionately large number of errors. In trying to explain the residual cases, these models end up modeling much of the noise as well. Previous research supports this view.

Our assertion is that no such first-order model exists when it comes to predicting future returns of asset classes in financial markets. In the absence of such a model, an approach that attempts to find a single model will perform poorly. It will model the noise and its predictions will tend to be close to the mean for all cases as shown in Figure 1. In contrast, a distributed approach to prediction using an ensemble of small disjuncts can do a better job of avoiding the inherent noise in such problems and finding the islands of structure in the data. While this will invariably model some of the noise as well, the expectation is that in the aggregate, it will find structure.

As support for the alternative approach, we have provided new evidence on the performance of "simple" small disjuncts as predictive models in financial markets. This type of model represents a new way to think about market prediction where the prediction task is distributed across a large number of simple independent models instead of a single one.

We have shown that the distributed approach works well when the complexity of the learner is low. Indeed, keeping disjuncts simple tends to partition the search space into large segments that are "uninteresting," where a model makes no prediction, and the small "interesting" ones where it makes predictions. This model fits naturally with financial markets compared to problems where a model must always make a decision (such as accepting or rejecting an application for credit, and so on). This property of the problem lets us focus on the interesting submodels (tree nodes) while ignoring those whose average forecast is close to the mean.

Prior research with small disjuncts has been almost entirely on classification problems, where it has been demonstrated that they account for a large proportion of errors and degrade the overall performance of a learned model. In contrast, our research shows that ensembles of simple small disjuncts perform well on data they have not seen before. Indeed the ensemble of small disjuncts *is* the predictive model. The results appear consistent with the conjecture that markets for the most part constitute noise, with infrequent opportunities presenting favorable risk-reward trade-offs. It is important to point out, however, that higher performance of small disjuncts on regression problems doesn't mean "higher accuracy" as in classification problems. Rather, the performance of the small disjuncts comes from their ability, on average, to predict accurately the higher values of the dependent variable. This is a subtle but important distinction.

A natural by-product of the ensemble approach, where the ensemble is a mix of signal and noise which is indistinguishable, is the quantification of "yield" of a learned model. Virtually all models induced by machine learning methods are known to degrade on future data, but other than standard measures of error, which are of limited value, and expected error bounds of learned models on data corresponding to the distribution of the training data [Vapnik 1995], no one has heretofore proposed a metric for quantifying the yield from an induced regression model relative to an ideal benchmark.

We also show that the degradation in performance of a learned model is decomposable into two distinct components: that resulting from the inevitable overfitting that occurs in the process of statistical induction, and that resulting from the fact that the future is inherently different from the past. This seems like a useful way to break down the expected degradation of a learned model. It can be applied to any problem where decision making is distributed across an ensemble of decision rules, each of which recognizes a small set of conditions under which to act. The relative magnitudes of the two degradations provide useful information to the model builder for performing cost/benefit and other types of analyses.

Financial markets are at the same time recurrent and evolutionary. Old patterns repeat, albeit unpredictably, and new patterns emerge constantly. This makes it difficult to find a single stable predictive model. A practical benefit of our approach is that it makes it possible to find patterns on a large scale efficiently, deploy them, and move on, without investing large amounts of effort in theory building only to find the theory obsolete. This approach is consistent with the observation that "patterns emerge before the reasons for them become apparent." An implication of the result is that small disjuncts provide a promising approach towards finding emerging patterns and assembling a predictive model automatically.

The evidence presented, that small disjuncts have predictive power, raises the obvious question as to why they are not discovered by market players and therefore disappear. The high degree of competition among players in financial markets can be expected to dissipate any obvious advantages that may occasionally arise.

The issue of efficient markets is hotly debated in the literature. The debate and its evolution is much too extensive to review in this article, however, it is worth addressing briefly the concept of "market anomalies," since it is possible that small disjuncts might represent a type of anomaly. It is well known that certain strategies lead to abnormal returns by exploiting market anomalies. An example of an anomaly is that stocks with low market capitalization (small stocks) have abnormally high average returns [Banz 1981]. Similarly, stocks with high ratios of book value to the market value of equity also have unusually high average returns [Rosenberg et al. 1985; Chan et al. 1991; Fama and French 1993]. Another example is that more profitable firms have higher average stock returns [Haugen and Baker 1996; Cohen et al. 2002]. Companies that deliver an earnings surprise see subsequent price movement in the direction of the surprise [Ball 1995]. Similarly, "momentum investing," a strategy of buying high and selling even higher which can be implemented as described in the preceding paragraph, also seems to persist and generate abnormal returns [Jegadeesh and Titman 1993; Schwager 1992; Soros 1987]. Similarly, a strategy of buying a higher interest rate yielding currency and selling a lower rate yielding one, known as the "carry trade," is another well known one. There are several other known market anomalies.

As summarized by Ball [1995], a proponent of the efficient market hypothesis,

> "the theory of efficient markets is, like all theories, an imperfect and limited way of viewing stock markets. The issue will be impossible to solve conclusively while there are so many binding limitations to the asset pricing models that underlie empirical tests of market efficiency" [Ball 1995].

In other words, financial markets are a complex phenomenon, not subject to a clean and simple interpretation.

A plausible explanation for the persistence of anomalies is that the excess returns realized by applying them entail taking on some sort of risk, so there is no "free lunch" after all [Till 2001]. For example, the carry trade strategy takes on the risk that the higher yielding currency will be devalued or that a sudden shift in risk taking preference will cause participants to reverse their default positions, in which case it performs badly. Similarly, a strategy that only buys "value" stocks (those with high book value to price ratios) takes on a business cycle risk that others don't want to assume. A strategy that sells deep out of the money options (a "short-option" strategy) makes a steady return on the small premium it makes selling catastrophic protection to others and works well until a catastrophe actually occurs (such events occurred in September and October of 2008). In summary, each of these types of strategies associated with anomalies can be viewed as taking on some sort of risk that others are unwilling to take, for which they can make superior returns than the market for significant periods of time.

Small disjuncts may have a similar interpretation. Markets are a complex social phenomenon where there is tremendous competition among participants to place a value today on future outcomes which are uncertain. This entails risk taking. But risk is not easy to calculate, nor is it static [Damodaran 2007]. During "normal" times, the human emotions of risk and greed are balanced in the aggregate, with large numbers of participants buying and selling, and prices move smoothly. However, fear and greed often go out of balance, causing people to become more or less risk averse. Such markets can experience rapid price changes and illiquidity, driven by human emotion. Market activity in October 2008 was an example of the rapid changes in investors' appetite for taking risk, with extreme fear quickly pervading markets, represented by a relative absence of buyers.

Small disjuncts by definition represent "unusual" situations in financial markets. It is plausible that in these outlier-like situations represented by the small disjuncts, market participants' propensity for risk is imbalanced. Accordingly, when fear dominates, the situations appear more risky than they have been historically or they would be during "normal" situations. There could be a number of fundamental reasons for why the outlier situations might be abnormal from a risk bearing standpoint. One is that these situations occur after major market dislocations when fear dominates. When this is the case, participants are frozen into inaction. Or they may have held positions previously that they were forced to liquidate. Or they may have had stricter risk limits imposed on their trading activity through institutional risk managers who tend to become highly risk averse in outlier situations. Schleifer [2001] provides an extensive discussion of the reasons why markets cease to be "efficient" during time of market stress. Whatever the reasons may be for the inefficiency, small disjuncts may represent such "anomalies" similar to the ones discussed earlier. These can be accounted for by the fact that each of these anomalies represents a specific risk that is being rewarded by the market, or some "residual" nonrisk bearing reason. In either case, taking risk when others are not should be correspondingly rewarded.

Regardless of the interpretation one might favor for the existence and persistence of small disjuncts, this research suggests that small disjuncts represent an interesting phenomenon in financial markets. Indeed, unlike previous domains in which they represent much of the "noise" in the problem, our research suggests that they collectively represent "the signal" in a domain that otherwise consists largely of noise.

More generally, the approach of considering a portfolio of small disjuncts as a predictive model is likely to be useful for problems that are noisy and "nonstationary," that is, where the rules for prediction shift over time. For such problems, it appears

promising to construct ensembles of rules that serve as prediction models that are valid for limited periods of time, and thereby reconstructed periodically.

Appendix: Description of independent variables

There are 68 independent variables in the dataset that belong to 8 variable types that are based on 8 historical intervals. The first variable type is historical returns that indicate the "trend" of a time series over the different intervals. (Figure 2 shows "Ret2" and "Ret5" which measure returns over the last 2 and 5 intervals respectively. There are 6 other returns measured over different historical intervals.) If "current" is the end of the current time period, the N day historical return, Ret(N) is defined as

$$\text{Ret(N)} = (\text{Price (current)} - \text{Price (current-N)})/\text{Price (current –N)}.$$

Similarly, an additional variable type, volatility, measures the historical dispersion in returns and ranges for a series. The "standard" measure of T day volatility is

$$\text{Vol(T)} = \text{Stdev (Ret(T),Ret(T-1), . . . Ret(1), where Stdev returns the standard deviation of the given series.}$$

Two additional types of volatility are computed identically. The first is based on intraperiod range (i.e., high minus low for a period), and the second is known as the Garman-Klass volatility, described in Garman and Glass [1980]. All variables are "normalized" using a 1 year time window. For example, a distribution of Vol(T) is generated using a year of history, and the current Vol(T) is expressed as a Z-score using the distribution.

Four additional variables are computed that indicate "position" within a range (commonly referred to as "stochastics" in the trade literature). For example, if the current "close price" of a series is the highest closing price over the last N periods, the position for it is +1 indicating the highest position, whereas if it is the lowest, its position is -1, and so on. In addition to the close, we measure this value for the open, high, and the low. Suppose we wish to compute the position of a variable (say the close) relative to the last T intervals. Let us call this variable V. This formula for computing the position of V is

$$\text{Position(V,T)} = (\text{Current (V)–Low(V,T)})/(\text{High(V,T)–Low(V,T)},$$

where

Current(V) is the last period's value of the variable V, Low(V,T)) is the low of the variable V over the last T periods, High(V,T)) is the high of the variable V over the last T periods.

In addition to the 64 variables defined before (8 types over 8 intervals), the database consists of four additional variables that calculate volatility-adjusted returns over four periods. Adjusting returns by volatility, also known as the Sharpe ratio, provides an indication of the smoothness of a trend.

REFERENCES

ACHELIS, S. 2000. *Technical Analysis from A to Z*. McGraw-Hill, New York, NY.

AKAIKE, H. 1974. A new look at the statistical model identification. *IEEE Trans. Autom. Control* 19,6, 716–723.

ALEXANDER, J. 1991. Earnings surprise, market efficiency, and expectations. *Finan. Rev.* 27, 4, 475–502.

ALI, K. M. AND PAZZANI, M. J. 1992. Reducing the small disjunct problem by learning probabilistic concept descriptions. In *Computational Learning Theory and Natural Learning Systems,* T. Petsche, Ed., The MIT Press, Cambridge, MA, 183–199.

ARORA, S. AND BARAK, B. 2009. *Computational Complexity: A Modern Approach.* Cambridge University Press.

BALL, R. 1995. The theory of stock market efficiency: Accomplishments and limitations. *J. Corp. Finan. 8,* 1, 4–18.

BANZ, R. W. 1981. The relationship between return and market value of common stocks. *J. Finan. Econ. 9*, 3–18.

BAUER, R. 1994. *Genetic Algorithms and Investment Strategies.* Wiley Finance.

BELLMAN, R. E. 1957. *Dynamic Programming*. Princeton University Press, Princeton, NJ.

BREIMAN, L. 2001. Random forests. *Mach. Learn. 45,* 1, 5–32.

BUJA, A. AND LEE, Y. 2001. Data mining criteria for tree-based regression and classification. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

BUNGE, M. 1963. *The Myth of Simplicity: Problems of Scientific Philosophy*. Prentice-Hall, Englewood Cliffs, NJ.

CANTU-PAZ, E. AND KAMATH, C. 2000. Combining evolutionary algorithms with oblique decision trees to detect bent double galaxies. In *Proceedings of the International Symposium on Optical Science and Technology*. SPIE Annual Meeting. 63–71. (Also available as Lawrence Livermore National Laboratory Tech. rep. UCRL-JC-138979.)

CHAN, L. K. C., HAMAO, Y., AND LAKONISHOK, J. 1991. Fundamentals and stock returns in Japan. *J. Finan. 46*, 1739–1789.

CHERNICK, M. R. 1999. *Bootstrap Methods, A Practitioner's Guide*. Wiley Series in Probability and Statistics.

COHEN, R. B., GOMPERS, P. A., AND VUOLTEENAHAO, T. 2002. Who underreacts to cash-flow news? Evidence from trading between individuals and institutions. *J. Finan. Econ. 66,* 409–462.

DAMODARAN, A. 2007. *Strategic Risk Taking: A Framework*. Wharton School Publishing, Upper Saddle River, NJ.

DANYLUK, A. AND PROVOST, F. 1993. Small disjuncts in action: Learning to diagnose errors in the local loop of the telephone network. In *Proceedings of the 10th International Conference on Machine Learning.* Morgan Kauffman Publishers.

DUNIS, C., TIMMERMAN, A., AND MOODY, J. 2001. *Developments in Forecast Combination and Portfolio Choice.* Wiley Financial Economics, London.

FAMA, E. F. AND FRENCH, K. R. 1993. Common risk factors in the returns on stocks and bonds. *J. Finan. Econ. 33*, 3–56.

GARMAN, M. AND KLASS, M. 1980. On the estimation of security price volatility using historical data. *J. Bus. 53,* 1, 67–78.

GRINOLD, R. C. AND KAHN, R. 1999. *Active Portfolio Management*. Irwin.

HOLTE, R. C., ACKER, L., AND PORTER, B. 1989. Concept learning and the problem with small disjuncts. In *Proceedings of the 11th International Conference on Artificial Intelligence*.

JEGADEESH, N. AND TITMAN, S. 1993. Returns to buying winners and selling losers: Implications for stock market efficiency. *J. Finan. 48*, 65–91.

KAUFFMAN, P. 2004. *New Trading Systems and Methods.* 4th Ed. Wiley.

KEARNS, M. J., NEVMYVAKA, Y., AND FENG, Y. 2006. Reinforcement learning for optimized trade execution.In *Proceedings of the International Conference on Machine Learning.*

KEIM, D. 1983. Size-Related anomalies and stock return seasonality: Further empirical evidence. *J. Finan. Econ. 12*, 13–32.

LEE, H. K. H. 1999. *Bayesian Nonparametrics via Neural Networks*. Cambridge University Press.

MOODY, J. AND SAFFELL, M. 2001. Learning to trade via direct reinforcement. *IEEE Trans. Neural Netw. 12*, 4.

MURTHY, S. K., KASIF, S., AND SALZBERG, S. 1994. A system for induction of oblique decision trees. *J. Artif. Intell. Res.* 2,1, 1–32.

PERLICH, C., PROVOST, F., AND SIMONOFF, J. 2003. Tree induction versus logistic regression: A learning curve analysis. *J. Mach. Learn. Res. 4.*

POGGIO, T. AND GIROSI, F. 1989. A theory of networks for approximation and learning. MIT Artificial Intelligence Laboratory Paper 1140.

QUINLAN, J. R. 1991. Technical note: Improved estimates for the accuracy of small disjuncts. *Mach. Learn. 6,* 1.

ROSENBERG, B., REID, K., AND LANSTEIN, R. 1985. Persuasive evidence of market inefficiency. *J. Portf. Manag. 11*, 9–17.

SHLEIFER, A. 2001. *Inefficient Markets: An Introduction to Behavioral Finance*, *Clarendon Lectures in Economics*. Oxford University Press.

SCHWAGER, J. D. 1992. *The New Market Wizards: Conversations With America's Top Traders*. John Wiley and Sons, New York, 224.

Soros, G. 1987. *The Alchemy of Finance*. Simon and Shuster, New York.

Till, H. 2001. *Life at Sharpe's End*, *Risk and Reward*. FOW Publication.

Ting, K. M. 1994. The problem of small disjuncts: Its remedy in decision trees. In *Proceedings of the 10th Canadian Conference on Artificial Intelligence*.

Torgo, L. 2001. A study of end-cut preferences in regression trees. Tech. rep., Department of Computer Science, University of Porto, Portugal.

Tversky, A., and Kahnemann, D. 1974. Judgment under uncertainty: Heuristics and biases. *Sci. 185,* 4187.

Webb, G. 1996. Further experimental evidence against the utility of Occam's razor. *J. Artif. Intell. Res*. 4, 397–417.

Vapnik, V. 1995. *The Nature of Statistical Learning Theory*. Springer, New York.

Weiss, G. M. 1998. Learning with rare cases and small disjuncts. In *Proceedings of the 12th International Conference on Machine Learning.* Morgan Kauffman Publishers.

Weiss, G. M. and Hirsh, H. 1998. The problem with noise and small disjuncts. In *Proceedings of the 15th International Conference on Machine Learning.* Morgan Kauffman Publishers.

Weiss, G. M. and Hirsh, H. 2000. A quantitative study of small disjuncts. In *Proceedings of the 7th National Conference on Artificial Intelligence.*

Received April 2010; revised June 2010; accepted September 2010