Fund Asset Inference Using Machine Learning Methods: What's in that Portfolio?

David Byrd[*], Sourabh Bajaj[†], Tucker Hybinette Balch[‡]

Abstract

Given only the historic Net Asset Value of a large cap mutual fund, which members of some universe of stocks are held by the fund? Discovering an exact solution is combinatorially intractable as there are, for example, $C(500, 30)$ or 1.4×10^{48} possible portfolios of 30 stocks drawn from the S&P 500. The authors extend an existing linear clones approach and introduce a new sequential oscillating selection method to produce a computationally-efficient inference. Such techniques could inform efforts to detect fund "window dressing" of disclosure statements or to adjust market positions in advance of major fund disclosure dates. The authors test the approach by tasking the algorithm to infer the constituents of exchange traded funds for which the components can be later examined. Depending on the details of the specific problem, the algorithm runs on consumer hardware in 8 to 15 seconds while identifying the target portfolio constituents with an accuracy of 88.2% to 98.6%.

We address the problem of inferring the number and identity of constituents of an unknown portfolio given a time series of the portfolio's aggregate value. Consider a specific example: Given the historic net asset value of a large cap mutual fund, which members of the S&P 500 are held by the fund and in what proportions? Solutions to this problem enable a number of applications such as: Detecting "window dressing" where a fund might rearrange their portfolio just before a reporting deadline to show that they hold reputable stocks when in reality they had been holding risky assets; Or adopting market positions in advance of a significant fund disclosure date to earn profit from those pursuing a replication strategy after disclosure.

Other researchers have addressed related problems: Sharpe (1992) addressed the problem of inferring fund exposures to broad asset classes. Edirisinghe (2013), Chen and Kwon (2012), and others have addressed the index tracking problem where the constituents of the portfolio are known *a priori*. However, to our knowledge, ours is the first method to infer both the number of constituent assets and their identities without having any constituent information in advance.

[*]David Byrd is a research scientist at the Georgia Institute of Technology in Atlanta, GA. (db@gatech.edu)

[†]Sourabh Bajaj is a former student of the Georgia Institute of Technology in Atlanta, GA. (sbajaj9@gatech.edu)

[‡]Tucker Hybinette Balch is a professor of computer science at the Georgia Institute of Technology in Atlanta, GA. (tucker@cc.gatech.edu)

Instead we assume knowledge only of the daily close price for each stock in the S&P 500 plus the target exchange traded fund (ETF), or daily net asset value (NAV) for a target mutual fund.

The linear clones method of Hasanhodzic and Lo (2007), on which we base our extended linear clones method, addresses the problem of estimating and evaluating portfolio weights W if constituency C is known. Our extension is to infer C (including its size) prior to applying linear clones to obtain W. That is, the present state of the art takes known C and estimates W. Our extension addresses the question: "what if we don't know C?"

Discovering an exact solution by exhaustively checking all possibilities is combinatorially intractable. There are, for instance, $C(500, 30)$ or 1.4×10^{48} possible portfolios of 30 stocks drawn from the S&P 500. It would accordingly require millions of years using today's highest performance computer to infer the constituents of the Dow Jones Industrial index via exact computation. Accordingly, we have developed algorithms that find approximate solutions. Our approach leverages Sharpe's asset class exposure method in combination with a well known technique from Machine Learning, the sequential floating forward selection (SFFS) algorithm (Somol et al. 1999), to solve these problems in a few seconds. We demonstrate the approach in a series of example problems where we task the algorithm to infer the constituents of exchange traded funds (ETFs) for which the components are known. Because the motivation is to predict holdings of mutual funds, we treat the ETFs as a demonstrative proxy for other fund types, limiting ourselves to a single end-of-day NAV and periodic holdings disclosures, despite richer data being available for ETFs. Depending on the details of the specific problem, the algorithm runs on consumer hardware in 8 to 15 seconds, while identifying the constituents of the target portfolio with an accuracy of 88.2% to 98.6%.

1 Background and Related Work

In this paper we utilize a novel adaptation of the SFFS technique (Somol et al. 1999) to infer the holdings of a fund for which daily NAVs are known. A key improvement that our method provides is fitting a model of individual equity constituents rather than a broad clone based on risk factors or asset classes. To our knowledge, this is the first such solution. However, a number of related papers have given us valuable insight to the general class of problem.

Large hedge funds (over $150 million in assets under management) and all mutual funds are required by the SEC to report their significant holdings every quarter. Hedge funds not subject to mandatory reporting may also disclose some or all of their holdings as a matter of activism (Agarwal et al. 2015). The timing of these disclosures has been shown to be correlated with increased trading volume in the disclosed assets and a permanent price impact as the trading public rushes to buy or sell those equities added or dropped by high-performing funds (Croci and Petrella 2015; Agarwal et al. 2015). The ability to infer fund holdings from publicly available data prior to the announcement could allow a trader to enter or exit a position with less risk, greater potential reward, and no insider trading concerns (Frank et al. 2004).

Kacperczyk, Sialm, and Zheng (2006) studied the impact of unobserved actions in mutual funds. They calculated a projected return based on previously-disclosed holdings and compared it to the fund's actual return, considering the difference (the "return gap") as a measure of value added by the fund manager. Their work evaluates the correlation between the return gap, hidden costs like transaction fees, and hidden gains such as interim trades. It also explores the possibility of using this return gap to predict mutual fund performance.

Confounding factors when using portfolio holdings as a measure of performance are identified by Meier and Schaumburg (2004), including the lack of detailed enter/exit dates for holdings, lack of a model for holdings both opened and closed between disclosures, and the potential for funds to "window-dress" their portfolios, deliberately adjusting the contents around disclosure time to mask the fund's real investment strategy. Previous efforts evaluate the *effect* of asset disclosure, while we aim to provide algorithms to *predict* the contents of the disclosure.

Frank et al. (2004) analyzed "copycat" hedge funds which mimic the holdings of actively managed funds, analyzing the long-term difference in returns between the two. Our study differs in that we aim to infer pre-disclosure holdings and enable *profit* from the impact disclosure will have on the underlying equities, while Frank et al. (2004) execute post facto trades to *mimic* the target fund.

Sharpe (1992), from whom we draw inspiration, explored asset class factor models in which he decomposed mutual funds into asset classes such as growth stocks, bonds, large cap, etc. We use a similar approach, but proceed to the more granular level of individual equities. Fung and Hsieh (1997) applied the same technique to hedge funds using principal component analysis (PCA) to categorize the components. PCA produced well-fitting component axes, but still at the level of investment styles and asset class mixtures, rather than individual instruments.

The method of linear clones was explored by Hasanhodzic and Lo (2007) to perform a regression fit of certain risk classes to which a hedge fund may be exposed. Each factor in the regression represents an entire asset class such as the S&P500, the US bond market, or USD (similar to the earlier work by Sharpe) and the corresponding regression coefficient represents the allocation of that class. In their work, two variants were explored for the input data: using the entire fund history (called "fixed-weight") or using 24 months just prior to the prediction time (called "rolling window"). While most of the clones created were inferior to the actual hedge funds, performance is similar enough to justify further exploration. Our study follows on most closely to their efforts at adapting Sharpe's original work: One of our approaches extends their method of rolling linear clones to infer at a finer resolution – specific assets held by a fund – rather than the more coarse level of asset classes.

Bertsimas, Kogan, and Lo (1997) proved that any general payoff strategy such as mutual funds could be replicated using more liquid instruments. Kat and Palaro (2005) explored the possibility of distribution-based clones. Amenc, Goltz, and Le Sourd (2009) showed that such clones could only be successful with a training period of over 6 years. This is quite long for hedge funds and mutual funds that are more dynamic in nature. Amenc et al. (2010) also explored non-linear clones for hedge funds but the performance of these was worse than their linear counterparts,

hinting at over-fitting or bias arising from the greater flexibility of fit.

Our effort is to fill a gap in the body of work, providing methods to directly infer the quantity and identity of constituents in a liquid fund that is priced daily, using only that daily close price or NAV of a target fund, plus the daily close prices of a universe of stocks from which constituents could have been drawn.

2 The Portfolio Inference Problem

Here we state the problem formally and introduce some notation. Our problem is:

Given: A *target portfolio* P with known value over time but unknown constituents C, unknown allocations W, and a universe of candidate constituents U;

Find: A portfolio clone, $\hat{P}$ with constituents $\hat{C} \subseteq U$ with allocations $\hat{W}$ that maximizes the Matthews correlation coefficient or MCC (Matthews 1975) between C and $\hat{C}$.

It should be noted that, while we do obtain an estimated weight vector $\hat{W}$, our primary aim in the current work is an accurate estimate $\hat{C}$ of the portfolio constituents. Once $\hat{C}$ is obtained, $\hat{W}$ can be estimated using the linear clones method of Hasanhodzic and Lo (2007) or others, as described in the previous section.

The target portfolio P can be any fund for which daily prices are known. Examples include mutual funds, ETFs, indexes or other instruments for which daily information is available. In this initial work, our objective is to infer the constituents of a long-only equity portfolio. This allows experimentation and testing with publicly available data on index-tracking ETFs that provide a suitable ground truth for testing the approach. Note that if the target portfolio is drawn from U, the clone $\hat{P}$ that maximizes MCC is exactly P. By solving this problem we consequently discover the number of assets in P, their identities C, and portfolio weights W. Our method uses in sample tracking error as a loss function to optimize C and W. The current analysis focuses on in-class versus out-class prediction accuracy of C.

3 Approach

In this section we describe the various algorithmic methods we developed to solve the portfolio inference problem. In a later section we examine their performance.

1. Let U be the universe of candidate stocks, N be the number of constituents, and M be the maximum portfolio size.
2. For $N = 1 \rightarrow M$:
3. Apply linear regression to candidate universe U.
4. Let constituent vector estimate $\hat{C}$ contain the N candidate stocks with largest regression coefficients.
5. Apply linear regression to revised $\hat{C}$ to determine final portfolio weights $\hat{W}$.
6. After evaluating portfolios of size $1 \rightarrow M$, select candidate portfolio $\hat{P}$ that minimizes RMSE of NAV($\hat{P}$)vs NAV(P), the in-sample tracking error.

Exhibit 1: Algorithmic steps in extended linear clones method.

3.1 Extended Linear Clones method

We choose the linear clones method Hasanhodzic and Lo 2007 as a starting point to attack our problem. Recall that the linear clones method assumes knowledge of the portfolio constituents, so we must extend the approach to relax that assumption. We call our modification to the linear clones method Extended Linear Clones (ELC).

Let's first review the original linear clones method. Recall that we intend to find a set of constituents $\hat{C}$ and their weights $\hat{W}$ that will maximize Matthews' correlation with P. The linear clones method assumes that $\hat{C}$ is known and that the only issue is to find the weights $\hat{W}$.

The linear clones method treats this as a simple regression problem, as follows: For each day t and asset i, we define X_{it} to be the daily return of constituent i. We refer to X_t as a vector of all candidate asset returns on day t. Y_t is the daily return of the target portfolio. Over a year of 252 trading days, we have 252 data points: $< X_1, Y_1 >, < X_2, Y_2 >, \ldots, < X_{252}, Y_{252} >$. If we treat each day's data as an independent observation we can use linear regression to find the weights $\hat{W}$ that solve the equation $\hat{W}X = Y$ subject to the constraint $\sum \hat{W}_i = 1.0$. Daily returns are calculated from consecutive market days. This having been done, the sequence of data points given to the algorithm is now unimportant.

The linear clones method assumes that the constituent set $\hat{C}$ is known. For our problem, we do not know the constituents *a priori*. Accordingly we must modify the method. We perform the same linear regression fit, but allow a variable number of factor terms (each an equity in a portfolio of unknown size). Unlike Hasanhodzic and Lo, we do not constrain the sum of coefficients to be 1.0. Instead, we evaluate the coefficient sum during analysis as a test for the reasonableness of our interpretation of the coefficients, expecting that it should naturally be very near one, rather than normalizing it to one. Our modification to the linear clones method is summarized as a series of algorithmic steps in Exhibit 1.

The most significant change to the linear clones method is to accommodate a portfolio of unknown size, because the linear clones method assumes a fixed number of exposure factors. The computation time is very fast even for a non-trivial real world portfolio (portfolio size 30 to 80, universe size 500 or 1000), so we simply iterate over the possible range of portfolio sizes $1 \rightarrow \|U\|$ and select the size that produces the least in-sample tracking error, which we define as the root mean square error (RMSE) between series of daily NAVs for P and $\hat{P}$. To avoid that the method always select some weighted combination of all stocks in U, we mandate that any included stock must represent at least 0.1% of the portfolio by weight.

Another challenge is that we must regress over all equities in the universe as factors to determine which are the most significant (maximum positive weights). The coefficients of the selected equities were determined in the presence of a large number of other, potentially discarded, factors. To address this issue, we run the regression one more time using only the selected top-N ($N =$ portfolio size) significant factors. The coefficients of this second run are interpreted as the portfolio allocation weights.

3.2 Sequential Oscillating Selection (SOS) method

Note that in the ELC method the linear coefficient for each equity is used implicitly as a measure of importance of the asset to the portfolio. The coefficient is essentially a "score" and only the highest N scoring assets are retained. During exploratory work of other approaches to portfolio inference, multiple methods of scoring individual equity significance to the portfolio were examined (e.g.: *coefficient* $\times$ *volatility*, *coefficient* $\times$ ***RMSE*** of stock time series). In some cases, the scoring formula improved the results, but this was not the case with ELC, so the most simple *score* = *weight* method was retained for those studies.

A potential weakness in ELC is that the fitness of a portfolio is tested in the presence of extraneous members of the universe that will not be included in the final portfolio. For example, when selecting the best portfolio for $N = 2$, the regression process is performed using the full universe, with the two highest coefficients selected. If instead, every possible combination of two candidate constituents were independently tested, the resulting selection of $\hat{C}$ could differ. The challenge presented by this observation is the aforementioned computational intractability of testing every possible combination of candidate constituents for every $N = 1 \rightarrow M$.

The Sequential Oscillating Selection (SOS) algorithm was developed to improve the inferential power of the ELC method by retaining the underlying time-independent linear methods but allowing for more exploration of potential candidate portfolios. SOS is related to a family of methods from the field of machine learning to solve the feature selection problem (Pudil, Novovičová, and Kittler 1994). That is, given a set of features that may be predictive of a future outcome, which subset of those features, when utilized together provide the most predictive power?

SOS is derived from SFFS as described in Pudil, Novovičová, and Kittler (1994). SOS works by first trying each feature individually to discover which one, by itself, is most predictive. It adds that feature to the set of features to be used. It then tries each remaining feature in combination

with the first feature to see which one augments the set with the most predictive power. The method augments the feature set one feature at a time until results fail to improve, then diminishes one feature at a time until results fail to improve, oscillating direction until no further greedy improvement is possible.

SOS starts with an input consisting of historical data (daily returns) of all the constituent stocks in the candidate universe U (e.g., S&P 500) individually and the target fund in aggregate for an arbitrarily-selected twelve month period. The daily return of each stock is treated as a feature. The corresponding daily return of the target portfolio is assigned as the dependent variable.

In a manner similar to ELC, SOS always works from daily price changes and ignores any temporal ordering of the data. SOS tests each equity in the universe using linear regression against the target portfolio to find the single equity that most closely matches the index by Root Mean Squared Error (RMSE). This is the initial portfolio. The forward process of the algorithm iteratively augments the current size N portfolio by each candidate equity in turn, keeping only the best size $N+1$ portfolio. When no augmentation improves the RMSE over the current portfolio, the backward process is engaged, iteratively trimming the current size N portfolio by each constituent in turn, keeping only the best size $N-1$ portfolio. When no trimmed portfolio improves the RMSE over the current portfolio, the forward process is engaged again. When neither process improves the RMSE, the final estimated portfolio is determined.

The SOS algorithm receives the same input data as the previously-described ELC algorithm, and proceeds as described in Exhibit 2.

4 Experimental Methodology

The ELC and SOS algorithms are implemented in Python 3.5 leveraging the numerical libraries *numpy* and *scipy* (Jones, Oliphant, Peterson, et al. 2001). Twelve months of market data and ETF constituency information are drawn from Compustat Capital IQ, Select Sector SPDRs, and Yahoo! Finance (Compustat 2018; ALPS Portfolio Solutions Distributor, Inc 2018; Yahoo! Finance 2018). We selected several sector ETFs as target portfolios P, with the S&P 500 as our candidate asset universe U. The algorithms also work with target portfolios containing unknown constituents (such as mutual funds), but we chose sector ETFs because we can validate the output of our algorithms by comparing them with the known constituents of the ETFs.

Each sector ETF is designed to track a specific S&P sector index, and accordingly contains the constituents listed in the sector index. ETFs were chosen, as opposed to sector index values, because ETFs and our historical stock price data account for dividends while sector index values are based on price only. We referred to Standard & Poor for lists of ETF and index constituents, which we use to judge the accuracy of our algorithms.

We consider each stock in the S&P 500 as a possible candidate for our target portfolio $\hat{P}$. Each candidate stock and the target portfolio are treated as an unordered "bag of returns" input to

1. Initialize $\hat{C}$ to contain the single equity from the candidate universe U that minimizes RMSE of daily returns with respect to target portfolio P. Initialize algorithm direction to forward (additive).

2. Tentatively augment $\hat{C}$ with each currently-excluded candidate equity in turn, judging fit by RMSE of daily returns of the augmented hypothesis portfolio versus the target portfolio.

   (a) If at least one augmented hypothesis portfolio scores better than the base hypothesis portfolio, retain the best augmented hypothesis portfolio as the new $\hat{C}$ and continue from step 2.

   (b) If no augmented hypothesis portfolio scores better than the base hypothesis portfolio, retain the previous base hypothesis portfolio and change the algorithm direction to backward (subtractive). Continue with step 3.

3. Tentatively diminish the base hypothesis portfolio by each currently-included candidate in turn, judging fit by RMSE as during augmentation.

   (a) If at least one diminished hypothesis portfolio scores better than the base hypothesis portfolio, retain the best decremented hypothesis portfolio as the new $\hat{C}$ and continue from step 3.

   (b) If no diminished hypothesis portfolio scores better than the base hypothesis portfolio, retain the previous base hypothesis portfolio and change the algorithm direction to forward (additive). Continue from step 2.

4. HALT: If neither augmentation nor diminishment can further improve the portfolio, emit the current $\hat{C}$ as the final inferred portfolio.

5. Apply linear regression to final $\hat{C}$ to determine final portfolio weights $\hat{W}$.

Exhibit 2: Algorithmic steps in sequential oscillating selection method.

a linear regression model to iteratively choose stocks belonging to the target portfolio.

Experiments are executed on a BSD-based UNIX system powered by a 2.6 GHz Intel Core i7 processor and 16 GB of 1600MHz DDR3 internal memory.

4.1 Experimental Results

Nine popular ETFs during the twelve-month period beginning in October 2013 were used as inference targets. Our objective was to discover the constituents and allocation weights for an inferred portfolio $\hat{P}$ at the end of the study period. We evaluated the accuracy of the inferred portfolio using two metrics: classification accuracy and Matthews Correlation Coefficient or MCC

			ELC		SOS	
Index Tracked	Symbol	Actual	Pred.	Correct	Pred.	Correct
Dow Jones Industrial Average	DIA	30	206	29	54	27
S&P Materials Sector	XLB	29	496	29	57	26
S&P Energy Sector	XLE	41	173	38	57	36
S&P Financial Sector	XLF	85	358	84	74	50
S&P Industrial Sector	XLI	63	313	60	79	48
S&P Technology Sector	XLK	69	333	58	56	39
S&P Consumer Staples Sector	XLP	37	501	37	38	29
S&P Utilities Sector	XLU	29	458	29	36	29
S&P Health Care Sector	XLV	48	456	48	53	32

Exhibit 3: Count of actual, predicted, and correct constituents per target symbol per approach

	Accuracy		MCC	
Symbol	ELC	SOS	ELC	SOS
DIA	0.645	0.940	0.285	0.645
XLB	0.068	0.932	0.025	0.611
XLE	0.725	0.948	0.365	0.719
XLF	0.451	0.882	0.274	0.561
XLI	0.489	0.908	0.257	0.629
XLK	0.429	0.906	0.149	0.575
XLP	0.074	0.966	0.000*	0.755
XLU	0.144	0.986	0.076	0.891
XLV	0.186	0.926	0.102	0.594
MEAN	0.357	0.933	0.170	0.664

* *ELC placed the entire universe into the candidate portfolio for XLP, resulting in an undefined MCC. We treat this as zero correlation.*

Exhibit 4: Success rate and MCC of predictions per target symbol per approach

(Matthews 1975). The target ETFs included DIA (Dow Jones Industrial Average tracking fund) and eight sector ETFs (tracking S&P market sectors) listed in Exhibit 3. Runtime for ELC varied from 3.24 to 3.61 seconds, with a mean runtime of 3.38 seconds. Runtime for SOS varied from 8.82 to 15.41 seconds, with a mean runtime of 11.48 seconds.

Predictive accuracy: Perhaps the simplest method with which to evaluate a boolean classification algorithm is accuracy. While having limitations discussed below, classification accuracy is easy to compute and has a straightforward interpretation, making it popular in financial literature, where it is referred to as "classification error" in the negative case (Aitken and Frino 1996), or "predictive accuracy" (Edmister 1972) or "prediction success rate" (Henry 2006) in the positive case. In the field of statistics, it is also known as the Rand Index (Rand 1971). Terminology aside, in the general case, it analyzes the similarity of two partitions X, Y of a set S. When X, Y represent the predicted class and actual class of each element in a set of predictions, it can be expressed as:

$$R = \frac{TP + TN}{TP + TN + FP + FN}$$

In this interpretation, X is the set of predicted element-wise classes and Y is the set of actual element-wise classes. The two classes in the experiment are $True$ (the equity is in the portfolio) and $False$ (the equity is not in the portfolio). Thus we can define TP as the set of elements labeled $True$ in both X and Y, TN as the set of elements labeled $False$ in both X and Y, FP as the set of elements labeled $True$ in X but $False$ in Y, and FN as the set of elements labeled $False$ in X but $True$ in Y. R then represents the accuracy of the partitions, or in this context simply "fraction correct" (Hubert and Arabie 1985).

The ELC method achieved predictive accuracy 0.068 to 0.725 with a mean of 0.357 when using the S&P 500 as the candidate universe. The SOS method achieved predictive accuracy of 0.882 to 0.986 with a mean of 0.933 on the same universe. These results are summarized in Exhibits 3 and 4. In the mean, we found that SOS misclassified 90% fewer equities than ELC.

An issue with this simple application of predictive accuracy is sensitivity to unequal class sizes in the "ground truth" data, called the *class imbalance* problem (Japkowicz and Stephen 2002). In the face of substantial class imbalance, for example when the correct label for an element of the universe is $True$ far more often than it is $False$, the simple accuracy calculation can produce high similarity values that do not properly account for the frequency with which each class occurs (Hubert and Arabie 1985). In our case, *most* equities from the candidate universe are *not* in the target portfolio. The proper partitioning does not come close to a 50/50 split of $True$ and $False$ labels, so the algorithm should predict $False$ far more often than it predicts $True$.

Matthews Correlation Coefficient: Because of the potential problems with simple predictive accuracy in the face of significant class imbalance, we chose also to assess our algorithms using the Matthews Correlation Coefficient or MCC (Matthews 1975). MCC assesses binary classification performance even in the face of unbalanced class size (Baldi et al. 2000) by accounting for the size of the true negative prediction set: Information not captured by precision, recall, and the F-score. MCC is a contingency method of calculating the Pearson product-moment correlation coefficient and therefore has the same interpretation (Pearson 1895; Powers 2011). We follow the customary interpretation of $abs(r)$ where values above 0.1 indicate weak accuracy, above 0.3 indicate medium accuracy, above 0.5 strong, and above 0.7 very strong. Negative values indicate similar anti-correlation (Evans 1996). The results of our methods in terms of MCC are listed in Exhibit 4.

The ELC method achieved an MCC of 0.000 to 0.365 with a mean MCC of 0.170 indicating a weak correlation. The SOS method achieved an MCC of 0.561 to 0.891 with a mean MCC of 0.664 (strong to very strong).

4.2 Discussion

We introduced *portfolio inference*, a new problem, the solution for which has a number of potential applications in finance including: Detection of "window dressing" by fund managers and development of arbitrage strategies based on the inferred constituents of large funds. It is important to distinguish *portfolio inference* from the simpler *index tracking* problem: The objectives for each are different. In the case of index tracking the goal is to minimize tracking error, while for portfolio inference the objective is to accurately infer the constituents of a portfolio. When applying boolean classification techniques, we defined the null class as those members of the universe not included in the target portfolio.

We presented two potential solutions, ELC and SOS, and we evaluated their performance along a number of dimensions. The ELC method, which represents a natural extension of existing index tracking solutions, is a sensible approach for determining allocations to assets to minimize tracking error, but it performs poorly at inferring portfolio constituents. According to statistical evaluations, ELC did not provide better accuracy than the random assignment of the equity universe to the target portfolio or the null class. The Linear Clones method proposed by Hasanhodzic and Lo (2007) and its predecessor (Sharpe 1992) have been demonstrated to work well for portfolio performance replication and inference of exposure to a fixed number of asset classes. They do not seem to extend well to portfolio composition inference with a large, unknown number of asset classes. This is no failing of those efforts — they were designed to solve the allocation problem in which C is already known.

The new approach presented here, SOS provides substantially better accuracy than ELC and random assignment. SOS explores a larger subset of the candidate portfolio search space in an "intelligent" manner. As a practical example, in identifying the 30 constituents of the Dow Jones Industrial Average ETF (DIA), the SOS method assigned each member of the S&P 500 universe to the correct group (in portfolio or not in portfolio) 94.0% of the time. It predicted a portfolio of 54 constituents (correct size: 30) of which 27 were correct, and excluded 457 stocks (correct size: 471) of which 444 were correct. In comparison, the ELC method identified 206 constituents and suffers correspondingly lower measures of accuracy. The SOS method is computationally slower than ELC, with a 3.5x runtime cost multiplier, but in typical applications it completes in as little as 10 seconds on a consumer-grade laptop, and the improvement in classification accuracy is significant.

In our evaluations, we noticed that the ELC method substantially overestimated the actual membership size of most portfolios. In fact, without any constraints applied, the method would frequently suggest that the entire universe of stocks were constituents (e.g., that the S&P Consumer Staples ETF contains all of the S&P 500 stocks). Because allocations to some constituents were very small, we added a minimum inclusion threshold to the ELC process. The addition of the minimum threshold did improve performance of the ELC method, but it was still significantly worse than SOS.

Even though SOS performs much better than ELC in this task, there is still room for improvement. Anecdotally, when the experiments were repeated with a maximum portfolio size of

100 and a minimum inclusion weight of 1%, which were not considered fair initial assumptions, both methods performed substantially better. This suggests better performance could be achieved in future work with constrained weights, a dynamic rather than fixed minimum inclusion threshold, or some fairly-derived "hint" concerning the correct portfolio size. One way to obtain such a "hint" would be to use knowledge of prior fund disclosures: We could presume the size will remain similar, or use the previous holdings disclosure as an initialization state for the algorithm. Another limitation is the requirement for a fund to be highly liquid and priced daily due to the periodicity of our data. This requirement could be relaxed in a future study using intraday pricing information for funds where that is available.

4.3 Acknowledgements

This material is based upon research supported by the National Science Foundation under Grant No. 1741026.

References

Agarwal, Vikas, Kevin A Mullally, Yuehua Tang, and Baozhong Yang. 2015. "Mandatory portfolio disclosure, stock liquidity, and mutual fund performance." *Journal of Finance* 70 (6): 2733–2776.

Aitken, Michael, and Alex Frino. 1996. "The accuracy of the tick test: Evidence from the Australian stock exchange." *Journal of Banking & Finance* 20 (10): 1715–1729.

ALPS Portfolio Solutions Distributor, Inc. 2018. *Select Sector SPDRs.* Obtained via Select Sector SPDRs at http://www.sectorspdr.com/sectorspdr/.

Amenc, Noel, Felix Goltz, and Veronique Le Sourd. 2009. "The Performance of Characteristics-based Indices." *European Financial Management* 15 (2): 241–278.

Amenc, Noël, Lionel Martellini, Jean-Christophe Meyfredi, and Volker Ziemann. 2010. "Passive hedge fund replication–Beyond the linear case." *European Financial Management* 16 (2): 191–210.

Baldi, Pierre, Søren Brunak, Yves Chauvin, Claus AF Andersen, and Henrik Nielsen. 2000. "Assessing the accuracy of prediction algorithms for classification: an overview." *Bioinformatics* 16 (5): 412–424.

Bertsimas, Dimitris P, Leonid Kogan, and Andrew Wen-Chuan Lo. 1997. *Pricing and Hedging Derivative Securities in Incomplete Markets: An [epsilon]-arbitrage Approach.* National Bureau of Economic Research.

Chen, Chen, and Roy H Kwon. 2012. "Robust portfolio selection for index tracking." *Computers & Operations Research* 39 (4): 829–837.

Compustat. 2018. *Compustat - Capital IQ - Index Constituents.* Obtained via Wharton Data Research Services at https://wrds-web.wharton.upenn.edu/wrds/.

Croci, Ettore, and Giovanni Petrella. 2015. "Price changes around hedge fund trades: disentangling trading and disclosure effects." *Journal of Management & Governance* 19 (1): 25–46.

Edirisinghe, NCP. 2013. "Index-tracking optimal portfolio selection." *Quantitative Finance Letters* 1 (1): 16–20.

Edmister, Robert O. 1972. "An empirical test of financial ratio analysis for small business failure prediction." *Journal of Financial and Quantitative analysis* 7 (2): 1477–1493.

Evans, James D. 1996. *Straightforward statistics for the behavioral sciences.* Brooks/Cole.

Frank, Mary Margaret, James M Poterba, Douglas A Shackelford, and John B Shoven. 2004. "Copycat funds: Information disclosure regulation and the returns to active management in the mutual fund industry." *Journal of Law and Economics* 47 (2): 515–541.

Fung, William, and David Hsieh. 1997. "Investment style and survivorship bias in the returns of CTAs: the information content of track records." *Journal of Portfolio Management* 24 (1): 30–41.

Hasanhodzic, Jasmina, and Andrew W. Lo. 2007. "Can Hedge-Fund Returns Be Replicated?: The Linear Case." *Journal of Investment Management* 5 (2).

Henry, Elaine. 2006. "Market reaction to verbal components of earnings press releases: Event study using a predictive algorithm." *Journal of Emerging Technologies in Accounting* 3 (1): 1–19.

Hubert, Lawrence, and Phipps Arabie. 1985. "Comparing partitions." *Journal of Classification* 2 (1): 193–218.

Japkowicz, Nathalie, and Shaju Stephen. 2002. "The class imbalance problem: A systematic study." *Intelligent data analysis* 6 (5): 429–449.

Jones, Eric, Travis Oliphant, Pearu Peterson, et al. 2001. *SciPy: Open source scientific tools for Python.* Obtained online. `http://www.scipy.org/`.

Kacperczyk, Marcin, Clemens Sialm, and Lu Zheng. 2006. "Unobserved actions of mutual funds." *Review of Financial Studies* 21 (6): 2379–2416.

Kat, Harry M, and Helder P Palaro. 2005. "Hedge Fund Returns: You Can Make Them Yourself!" *Journal of Wealth Management* 8 (2): 62–68.

Matthews, Brian W. 1975. "Comparison of the predicted and observed secondary structure of T4 phage lysozyme." *Biochimica et Biophysica Acta (BBA)-Protein Structure* 405 (2): 442–451.

Meier, Iwan, and Ernst Schaumburg. 2004. "Do funds window dress? Evidence for US equity mutual funds." *Working paper, Northwestern University.*

Pearson, Karl. 1895. "Note on regression and inheritance in the case of two parents." *Proceedings of the Royal Society of London* 58:240–242.

Powers, David Martin. 2011. "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation." *Journal of Machine Learning Technologies.*

Pudil, Pavel, Jana Novovičová, and Josef Kittler. 1994. "Floating search methods in feature selection." *Pattern Recognition Letters* 15 (11): 1119–1125.

Rand, William M. 1971. "Objective criteria for the evaluation of clustering methods." *Journal of the American Statistical association* 66 (336): 846–850.

Sharpe, William F. 1992. "Asset allocation: Management style and performance measurement." *Journal of Portfolio Management* 18 (2): 7–19.

Somol, Petr, Pavel Pudil, Jana Novovičová, and Pavel Paclık. 1999. "Adaptive floating search methods in feature selection." *Pattern Recognition Letters* 20 (11): 1157–1163.

Yahoo! Finance. 2018. *Yahoo! Finance Interactive Charts.* Obtained online. `http://ichart.finance.yahoo.com/`.