PORTFOLIO PERFORMANCE ATTRIBUTION VIA SHAPLEY VALUE

Nicholas Moehle[a,*], *Stephen Boyd*[b] *and Andrew Ang*[c]

We consider an investment process that includes a number of features, each of which can be active or inactive. Our goal is to attribute or decompose an achieved performance to each of these features, plus a baseline value. There are many ways to do this, which lead to potentially different attributions in any specific case. We argue that a specific attribution method due to Shapley is the preferred method, and discuss methods that can be used to compute this attribution exactly, or when that is not practical, approximately.

1 Introduction

Performance of an investment process. We consider an investment process guided by a portfolio manager, an automated process, or some combination. We are interested in some performance measure of the investment process over some past period, for example P&L, realized return, risk, tracking error, or turnover. Some of these performance measures we prefer to be large (e.g., P&L, realized return), and others we prefer to be small (e.g., risk, tracking error, turnover).

Features. The process has a number of variations or features, each of which can be active (or on) or inactive (or off). As a simple example, consider an investment process that relies on daily portfolio rebalancing using an optimization method. The features might be a leverage limit, an ESG constraint that limits the securities that can be held, and the use of a novel return forecast developed by some researchers. (In many practical applications, we would have far more than just three features.)

We assume that when the process was run over some time period, all the features were active. In our example above, this means that we ran the process with the leverage limit, the ESG constraint, and the forecasts.

[a]Goldman Sachs Quantitative Investment Strategies, 200 West Street New York, NY 10282. Phone: +1 (925) 989 5965, Email: nicholasmoehle@gmail.com.
[b]Stanford University, Packard 254, 350 Jane Stanford Way, Stanford, CA 94305. Phone: +1 (650) 721 2273, Email: boyd@stanford.edu
[c]BlackRock, 55 East 52nd Street, New York, NY 10055. Phone: +1 (212) 810-3525, Email: andrew.ang@ blackrock.com
*Corresponding author.

Attribution. The attribution problem asks the question: How much of the performance should we attribute to each of these features, and how much to a baseline or benchmark value? In attribution we are dividing up the performance that was achieved into an amount associated with each feature, and a baseline value, which corresponds to what the performance would have been with all the features off. The amount attributed to a feature can be negative, which means that the feature reduced the performance value. A positive attribution means that the feature increased the performance value. We seek *residual-free attribution*, which means that the sum of the amounts attributed to the features, plus the baseline value, equals the actual performance achieved.

Continuing our example above, suppose our investment process realizes a return of 8% over one year, with all three features active. An example attribution might be 1% to the leverage limit, −1% to the ESG constraint, 5% to the return forecast, and 3% to baseline performance. We interpret this as saying that the leverage limit was responsible for increasing our return by 1%, while our ESG restrictions depressed the return by 1%; our return forecast was responsible for 5% of the realized return, and 3% is attributed to baseline.

Attribution is closely tied to the idea of marginal performance gain, i.e., the change in performance when a feature is added. Unfortunately — and this is the crux of the problem — the marginal performance gain when adding a feature depends on which features have already been added. We can think of attribution as assigning a single marginal performance gain to a feature, independent of which other features are active.

We can carry out attributions for multiple performance objectives. For example, we can attribute realized risk, realized return, and realized turnover to our features and a baseline. (These are three separate attribution problems.) An example attribution is shown in Table 1.

Attribution has many applications. We can use it to allocate credit, for example, to determine bonus payments. For features that are associated with an additional cost, we can assess the cost per unit of performance delivered. For example, if our return forecast in our example incurs a data source cost, we can compare that cost with the return attributed to it. We can use attribution to makes changes moving forward; for example, we can consider dropping features that have a negative attribution to P&L. Attribution is a key method for explaining investment performance to clients, and is often required by law.

Challenges. There are two main challenges in carrying out attribution. The first is that it involves hypothetical or what-if situations. While

Table 1 Example attribution of three performance measures to two features plus baseline.

x_1	x_2	Risk	Return	Turnover		Baseline	x_1	x_2	*Total*
×	×	2.3	11	43	Risk	0.1	1.25	0.95	2.3
×		2	12	30	Return	5	5	1	11
	×	1.7	8	38	Turnover	2	16.5	24.5	43
		0.1	5	2					

The table on the left shows the data, i.e., the result of backtests with different combinations of the two features. The table on the right shows an example attribution; the first three columns show the amount attributed to the benchmark and two features, and the third shows the total attributed amount, i.e., the sum across the three columns.

we directly observe the performance achieved with all features active, we really do not know what the performance would have been with some of the features off. This is addressed by using a high-fidelity simulator or backtester, that can simulate what would have happened, had some of the features been off. Of course our attribution can only be as accurate as this simulator.

The second main challenge is conceptual or mathematical. Except in a very simple case, which we discuss below, it is hard to exactly define what attribution is. As a result, many different attribution methods are used in practice, leading to different attributions in any specific case. In this paper we will argue that a specific type of attribution, called Shapley attribution, is the best choice. Unfortunately, with more than just a handful of features, computing the Shapley attribution exactly requires carrying out an impractically large number of simulations or backtests. Fortunately, there are methods for computing it approximately, described below, which work well in practice.

The simple case.   We describe here, informally, the one case in which attribution is simple. (We describe this case mathematically below.) Attribution is simple when the performance measure is additive, i.e., a sum of terms, each associated with a feature plus a constant. Indeed, this particular form directly gives us the attribution. As a simple (and uninteresting) example, consider the total profit of a company over some period, with the features being the independent divisions of the company. Here the total profit is evidently the sum of the profits contributed by the divisions, plus some baseline profit for the company that is unrelated to the profits of the divisions.

Roughly speaking, in this simple case there is no interaction among the features; we simply add up the contributions of the features, which directly gives us the attribution, with the baseline

value being the difference between the achieved performance and the sum of that attributed to the features.

Interactions.   The challenge is when there are interactions among the features, in terms of how they affect the performance. As a simple example, suppose that feature one and feature two, taken individually, give no increase in performance; but when they are both active, give a substantial improvement in performance. In this case, how should we attribute the performance to these two features? Intuition suggests that they should share the credit equally, i.e., have attributions equal to half the increase in performance. As a variation on this situation, suppose that features 1 and 2 are the same, so having them both on is the same as having either of them on. Here too intuition suggests that they should share the credit.

1.1   Previous work

Cooperative game theory.   Our recommended attribution method uses the Shapley value, an idea from originated in cooperative game theory, which attempts to answer the question of how to allocate the earnings of a coalition to individual players (Shapley, 1953). It is derived axiomatically by showing that the only attribution method that satisfies a set of desirable properties is the Shapley value. Because of these desirable properties, the Shapley value is widely considered to be a fair approach to allocating value (Moulin, 2004). The Shapley value has seen a large number of extensions and variations; see Monderer *et al.* (2002) for a summary. One important extension assumes that certain coalitions of players are disallowed, which changes the resulting allocation (Hiller, 2018).

In general, the effort required to compute Shapley values is exponential in the number of players, which can be prohibitive when the number of players is large. In this case, the Shapley values

can be approximated using Monte Carlo (Castro *et al.*, 2009), with error bounds given in Maleki *et al.* (2013). When the coalition value function has specific properties (e.g., submodularity), more efficient methods may exist. (See Liben-Nowell, 2012).

Attribution in machine learning. Recently, Shapley values have been used to interpret the output of machine learning prediction models, such as random forests and neural networks (Lipovetsky and Conklin, 2001; Štrumbelj and Kononenko, 2014). In this case, the model inputs (or *features*) are modeled as players in a coalition, and the resulting prediction performance is the value of each coalition. As a result, mature theory and software exist for approximately computing the Shapley value for games with many players (Lundberg and Lee, 2017).

Portfolio performance attribution. Since the work of Jensen (1968), academics have sought to attribute returns of managers to skill (security selection) vs. exposure to rewarded risk factors, like the market portfolio (Fama and French, 2010; Sharpe, 1992). Many approaches for performance attribution exist. The simplest methods break up portfolio return into components, which sum to the portfolio return (Brinson *et al.*, 1986, 1991). Other standard approaches use the correlation between portfolio weights and returns (Grinold and Kahn, 2000; Grinold, 2006; Lo, 2007). These methods are scalable and are often used to attribute performance to a large number of predictive signals.

Many of these standard approaches use time series of returns or cross-sectional position holdings which result in unattributed value (or 'residuals'); one advantage of the Shapley value is *residual-free attribution*, which means the attributions of individual features and the baseline sum to the total portfolio return. In our exposition, we explicitly contrast Shapley attribution

to these more widely used techniques. While Shapley values have been applied to risk decompositions (Ortmann, 2016; Mussard and Terraza, 2008; Tarashev *et al.*, 2010; Colini-Baldeschi *et al.*, 2018), to our knowledge, ours is the first application of the Shapley value to the general portfolio performance attribution problem, and more generally to any statistic that is produced by an investment process.

2 Model

In this section we fix our notation and describe our model.

Investment process. We assume that there is an *investment process* which produces a dynamic (time-varying) portfolio allocation over some time window. The investment process may depend on market conditions, the prior portfolio holdings, the decisions of analysts or portfolio managers, and portfolio optimization techniques, to update the portfolio holdings over time. In practice, the investment process may be very complicated, and we intentionally leave the details unspecified.

Features and configuration. The investment process has n features that can be (or could have been) included or excluded, i.e., active or inactive. These features might represent the choice of a specific benchmark, choice of sector exposures or asset allocation, or the contributions of a specific analyst or signal. This inclusion or exclusion of feature i is denoted $x_i \in \{0, 1\}$, with $x_i = 0$ meaning that the feature is inactive, and $x_i = 1$ meaning the feature is active. The collection of these feature status values is called a *configuration* of the investment process, and is represented by the Boolean vector $x = (x_1, \ldots, x_n)$. For example, $x = (1, 0, 1, 1, 0)$ means that features 1, 3, and 4 are active, while features 2 and 5 are inactive.

We observe that there are 2^n possible configurations, which grows rapidly with the number of features n. For $n = 10$, there are around 1,000 possible configurations; for $n = 30$, the number is around 10^9.

The full and zero configurations.   The configuration $x = (1, 1, \ldots, 1) = \mathbf{1}$ (the vector of all ones) means that all features are active. We refer to this as the *full configuration* or *fully featured configuration*. We will assume that the full configuration is the one that was actually used. The other $2^n - 1$ configurations are hypothetical; we did not actually use them.

The configuration $x = (0, 0, \ldots, 0) = \mathbf{0}$ is called the *zero configuration* or the *baseline configuration* or the *benchmark configuration*. It corresponds to the investment process with all features inactive. In some cases it can be interpreted as investing in a benchmark portfolio.

Performance metric.   This investment process is evaluated using a real-valued performance metric $y \in \mathbf{R}$. (In practice, portfolios are evaluated using many metrics, which can be considered separately.) Examples of performance metrics include the portfolio's return, risk, risk-adjusted return, turnover, or average exposure to a particular risk factor, over some investment period. (These can be ex-ante values, evaluated using a contemporaneous model; or realized, ex-post values, evaluated using the actual data.) Note that large values of y can be good (as in return), or bad (as in risk).

We assume that when the full configuration $x = \mathbf{1}$ was used, the resulting realized performance was y^{real}, which can be directly observed. In cases when the baseline or zero configuration can be interpreted as investing in a benchmark portfolio, the performance value for this too can be directly observed.

Simulation and backtesting.   We use simulation to judge the performance using other, hypothetical configurations. This typically has the form of a backtesting engine, which can evaluate the performance under the hypothetical configurations. This process is represented by a function $f : \{0, 1\}^n \to \mathbf{R}$, with

$$y = f(x) = f(x_1, \ldots, x_n).$$

Evaluating the function f requires running a backtest of the investment process under configuration x, and recording the performance y. We assume that the backtests are calibrated so that $y^{\text{real}} = f(\mathbf{1})$, i.e., the backtest simulation result for the configuration we used agrees with the performance we actually observed.

Lift or marginal contribution.   We now introduce a natural concept in attribution, which is the change in performance when we add one feature. Suppose the configuration is x, with $x_i = 0$, i.e., feature i is inactive. Then $\tilde{x} = x + e_i$ (where e_i is the ith unit vector) is the configuration obtained by turning feature i on. We define the *lift* or *marginal contribution* as the change in performance obtained by adding feature i, i.e.,

$$f(x + e_i) - f(x).$$

This marginal contribution depends on the particular configuration x. In other words, the lift associated with adding a feature depends on which other features are active.

Geometric interpretation.   We can associate the 2^n different possible configurations with the corners of a unit (hyper)cube in $\mathbf{R}^n$. We can create a directed graph with configurations as vertices, by having an edge from configuration x to configuration $\tilde{x}$ if $\tilde{x} = x + e_i$ for some i. In words, an edge goes from one configuration to another that is obtained by adding one feature. We note for future use that we can associate with an edge from x to $\tilde{x} = x + e_i$ a marginal performance
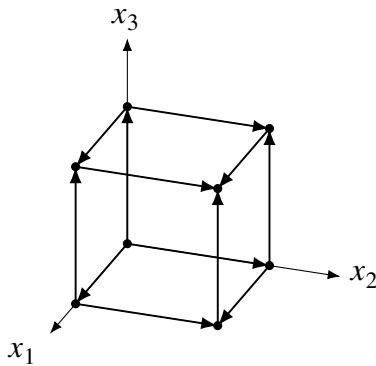
Figure 1 Visualization of configurations as vertices of a hypercube, for $n = 3$. The vertices, shown as dots, as configurations. The edges correspond to adding one feature to a configuration.

change $f(\tilde{x}) - f(x) = f(x + e_i) - f(x)$. The number of edges is $n2^{n-1}$.

An example with $n = 3$ features is depicted in Figure 1. The point $(0, 0, 0)$ represents the baseline configuration, and the point $(1, 1, 1)$ represents the fully featured configuration. The edge from $(0, 1, 0)$ to $(1, 1, 0)$ corresponds to adding feature 1 to the configuration with only feature 2 active. There are a total of 12 edges.

3 Attribution

We would like to attribute the realized performance y^{real} to the n features, i.e., to determine how much of the performance resulted from each feature. An attribution method determines real values $a_1, \ldots, a_n$ and b, where a_i is the amount attributed to feature i, and b is the baseline amount. The attribution and baseline amounts can be positive or negative. We will represent the attribution using a vector $a = (a_1, \ldots, a_n)$ and scalar b.

The attribution is derived from the feature performance function f, i.e., its values for the 2^n different configurations. An attribution method is an algorithm or method that determines the attribution based on evaluating f for some, or possibly all, configurations.

3.1 Attribution desiderata

We now describe several desirable properties of an attribution method.

Residual-free attribution.   We require *residual-free attribution*, which means that

$$y^{\text{real}} = f(\mathbf{1}) = a_1 + \cdots + a_n + b = \mathbf{1}^T a + b.$$

This means that the observed performance measure $f(\mathbf{1})$ is fully attributed to the n features, plus the baseline. Even though residual-free attribution is a crucial property of a good attribution method, we will see that commonly used attribution methods do not have it.

Correct baseline value.   We say that an attribution has the correct baseline value if

$$f(0) = b.$$

This means that the baseline value b matches the performance of the benchmark portfolio.

Symmetry.   We call an attribution method symmetric if, for any permutation of the features, the attributions are permuted the same way. This property implies that if two features are the same, i.e., they have the same effect on performance, then their attributions must be the same.

Monotonicity.   This means that if we change f in such a way that one feature's marginal contribution does not decrease (no matter which features are already active), then the attribution to this feature also does not decrease.

3.2 Additive case

We say the performance is *additive* if f has the form

$$f(x) = a^T x + b = b + \sum_{i:x_i=1} a_i,$$

for some vector a and scalar b. (If we consider x_i to be real numbers, and not just 0 or 1 as we do here, this corresponds to f being an affine function.) When f is additive, the baseline performance is b, and the marginal increase in the performance when adding feature i is always a_i, independent of what other features are already active.

For an additive function, a and b directly give an attribution which satisfies all four of the desiderata listed above: residual-free attribution, correct baseline, symmetry, and monotonicity. The case of additive performance is the easy, or even trivial case, for attribution. It is when f is not additive that it becomes more difficult to assign an attribution.

4 Attribution Methods

In this section, we describe several attribution methods, concluding with Shapley attribution method we recommend.

4.1 One-at-a-time attribution

We take $b = f(0)$, and

$$a_i = f(e_i) - f(0), \quad i = 1, \ldots, n.$$

In other words, we carry out a baseline simulation (if needed) and n additional simulations, each with exactly one feature enabled. We attribute to each feature the increase in performance when it is added to the baseline configuration, i.e., its lift or marginal performance increase from the baseline configuration $x = 0$. This method is natural, and requires carrying out only n (or $n + 1$ if we include the baseline) simulations.

One-at-a-time attribution satisfies correct baseline value, symmetry, and monotonicity. However, it can (and often does) fail to be residual free. To see this, consider the simple example with

$n = 2$ and

$$
\begin{aligned}
f(0) = 0, \quad & f(e_1) = 1, \\
f(e_2) = 1, \quad & f(\mathbf{1}) = 1.
\end{aligned}
\tag{1}
$$

In this example the presence of either feature 1 or feature 2 gives the full performance value 1. One-at-a-time attribution for this example is $b = 0$, $a_1 = 1$, $a_2 = 1$, so $b + a_1 + a_2 = 2$, whereas $f(\mathbf{1}) = 1$. Roughly speaking, one-at-a-time attribution over-allocates performance to the features in this example.

We note that one-at-a-time attribution coincides with the attribution described above for additive f. Here, however, the same formula is being applied to any f, not just additive f.

4.2 Leave-one-out attribution

The leave-one-out attribution method is closely related to one-at-a-time attribution As in one-at-a-time attribution, we set $b = f(0)$. We then carry out n simulations, with $x = \mathbf{1} - e_i, i = 1, \ldots, n$. In other words, for each feature we simulate the performance when it is left out, but all other features are present. We set

$$a_i = f(\mathbf{1}) - f(\mathbf{1} - e_i), \quad i = 1, \ldots, n,$$

which is the marginal performance increase when adding feature i when all other features are active. Like one-at-a-time attribution, leave-one-out attribution requires carrying out n simulations, plus a baseline simulation.

Like one-at-a-time, leave-one-out attribution satisfies correct baseline value, symmetry, and monotonicity, but it can fail to be residual free. To see this, consider the same example describe above in Equation (1). The leave-one-out attribution for this example is $b = 0$, $a_1 = 0$, and $a_2 = 0$, i.e., it allocates zero performance to each feature. Roughly speaking, it under-allocates performance in this example.

4.3 Sequential attribution

Another commonly used method is sequential attribution. We start by evaluating the baseline configuration performance $b = f(0)$. We then simulate the configuration $x = e_1$, i.e., we add the first feature. We continue adding features until we have all features active. We take

$$a_i = f(e_1 + \cdots + e_i) - f(e_1 + \cdots + e_{i-1}),$$
$$i = 1, \ldots, n,$$

which is the marginal contribution to performance from feature i, when the features $1, \ldots, i-1$ are active. Like one-at-time and leave-one-out attribution, this method requires n simulations, plus a baseline simulation.

Sequential attribution is residual free, since

$$b + a_1 + \cdots + a_n$$
$$= f(0) + (f(e_1) - f(0)) + (f(e_1 + e_2)$$
$$- f(e_1)) + \cdots + (f(1) - f(1 - e_n))$$
$$= f(1).$$

It also satisfies correct baseline and monotonicity.

But sequential attribution is not symmetric, since the attribution obtained depends on the order in which the features are added. The same example above given in Equation (1) illustrates this. Sequential attribution gives $b = 0$, $a_1 = 1$, and $a_2 = 0$; that is, the first feature gets attributed the full performance and the second gets none. If we swap the two features, we assign the full performance to feature 2.

Sequential attribution is also called off-the-top attribution, described in a different form. We first evaluate $f(1 - e_n)$, the performance when feature n is turned off, with all others on. We set $a_n = f(1) - f(1 - e_n)$, the marginal increase in adding feature n when all others are active. We

then evaluate $f(1 - e_n - e_{n-1})$, the performance when features n and $n - 1$ are both inactive, with all others active, and set

$$a_{n-1} = f(1 - e_n) - f(1 - e_n - e_{n-1}),$$

the marginal performance when we add feature $n - 1$, when features $1, \ldots, n - 2$ active. This is readily seen to result in the same attribution as sequential attribution. The only difference is in how we tell the story: In sequential attribution we add the features in order one at a time, whereas in off-the-top attribution, we are removing features in reverse order.

Geometric interpretation.   We can give a nice geometric interpretation of sequential attribution. We start at node or vertex $x = 0$ and follow a specific directed path to node e_1, then e_2, and so on, ending at the full configuration $x = 1$. The attribution to feature i is the marginal increase associated with the edge in which feature i is added. An example with $n = 3$ is shown (in red) in Figure 2.
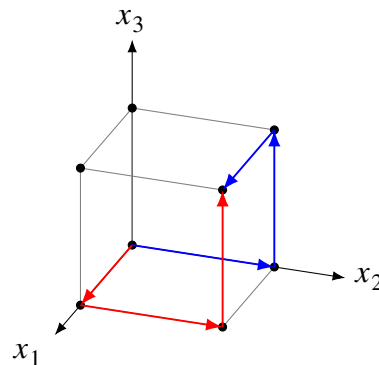
Figure 2 Two permutations for sequential attribution. The red path corresponds to the standard permutation $(0, 0, 0) \to (1, 0, 0) \to (1, 1, 0) \to (1, 1, 1)$, while the blue path corresponds to the permutation $(0, 0, 0) \to (0, 1, 0) \to (0, 1, 1) \to (1, 1, 1)$. Under the permuted attribution method, the lift associated with each edge in the path is the attribution to the feature added along that edge.

4.4 *Permuted sequential attribution*

Here we describe a simple extension of sequential attribution that will come up in the sequel. Let $\pi = (k_1, \ldots, k_n)$ be a permutation of $(1, \ldots, n)$, which means that each integer from 1 to n appears as one of the k_i. Define $\tilde{x}$ as $\tilde{x}_i = x_{k_i}$, $i = 1, \ldots, n$. This is the configuration vector when we permute the features using π. We define the permuted performance function as $\tilde{f}(\tilde{x}) = f(x)$.

Permuted sequential attribution permutes the original features to obtain $\tilde{f}$, then uses sequential attribution on $\tilde{f}$, and finally permutes the resulting attribution $\tilde{a}$ and $\tilde{b}$ back to the original ordering. In sequential attribution, we use the marginal performance contribution when the features are added one by one, in order. Permuted sequential attribution is the same, except that we add the features in the order $(k_1, k_2, \ldots, k_n)$.

Permuted sequential attribution satisfies residual-free attribution, correct baseline, monotonicity, but not symmetry. Indeed, symmetry would require that the attribution obtained is the same for any permutation π. (This is the case if and only if f is additive.)

Geometric interpretation.   We can associate a permutation π with a directed path from 0 to **1** on the vertices of the hypercube, and vice versa, since any such path corresponds to a permutation. We allocate to each feature the marginal change in performance along the edge in which feature i is added. An example with $n = 3$ is shown (in blue) in Figure 2.

4.5 *Shapley attribution*

Finally, we come to the attribution method we endorse, the Shapley method. The Shapley attribution is simply the average of the permuted sequential attributions over all $n!$ permutations. Formally, let a_π and b denote the attribution

for permuted sequential attribution with permutation π. (The value of baseline attribution b does not depend on the permutation.) The Shapley attribution is

$$a = \frac{1}{n!} \sum_\pi a_\pi, \qquad (2)$$

where the sum is over all $n!$ permutations.

This method satisfies all the desiderata: residual-free attribution, correct baseline, symmetry, and monotonicity. Indeed, it has been shown that any attribution method that satisfies these four desiderata must coincide with the Shapley attribution (Young, 1985).

The bad news is that evaluating the Shapley attribution requires 2^n simulations, which for n larger than 10 or so is likely to be impractical. This is contrasted with the one-at-a-time, leave-one-out, sequential, and permuted sequential attribution methods, which require only $n + 1$ simulations. We will address this computational complexity issue in more detail below.

We summarize the properties of the different attribution methods in Table 2.

Simple example.   Consider the simple example given in Equation (1). There are only $2! = 2$ permutations. For $\pi = (1, 2)$, we get attribution $b = 0$, $a_1 = 1$, and $a_2 = 0$; for $\pi = (2, 1)$, we get attribution $b = 0$, $a_1 = 0$, and $a_2 = 1$. The Shapley attribution for this example is

$$b = 0, \quad a_1 = 1/2, \quad a_2 = 1/2.$$

Roughly speaking, in this example, features 1 and 2 are the same; the presence of either alone gives the full performance. Permuted sequential attribution gives all the credit to the first feature in the sequence, and none to the second. The Shapley attributions average over the two cases, and splits credit to the two features equally.

Geometric interpretation.   The Shapley attribution for feature i is the average of the marginal

Table 2 Properties of attribution methods. Righthand column gives number of simulations required to compute the attribution.

Method	Full attr.	Baseline	Fairness	Monotonicity	Simulations
One-at-a-time		×	×	×	$n+1$
Leave-one-out		×	×	×	$n+1$
Sequential	×	×		×	$n+1$
Permuted seq.	×	×		×	$n+1$
Shapley	×	×	×	×	2^n

performance change when feature i is added, over all directed paths from 0 to **1**.

The $n = 3$ case. We work out the general Shapley attribution for the case with $n = 3$. There are $2^n = 8$ configurations, and $3! = 6$ sequences. We derive formulas for a_1 here; attribution to other features has similar formulas. First we list the $n!$ sequences and the associated marginal performance change for feature 1 as

Permutation	Marginal performance change
$(1, 2, 3)$	$f(1, 0, 0) - f(0, 0, 0)$
$(1, 3, 2)$	$f(1, 0, 0) - f(0, 0, 0)$
$(2, 1, 3)$	$f(1, 1, 0) - f(0, 1, 0)$
$(2, 3, 1)$	$f(1, 1, 1) - f(0, 1, 1)$
$(3, 1, 2)$	$f(1, 0, 1) - f(0, 0, 1)$
$(3, 2, 1)$	$f(1, 1, 1) - f(0, 1, 1).$
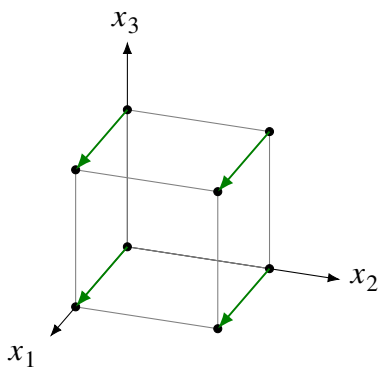
Figure 3 All green edges correspond to adding the first feature to a configuration without it.

Each line here corresponds to one edge in the graph, in which we add feature 1. Some of the marginal changes or edges are repeated; there are only four distinct edges. Expressing a_1 in terms of distinct edges or marginal changes, we have

$$a_1 = \frac{2}{6}(f(1, 0, 0) - f(0, 0, 0))$$
$$+ \frac{1}{6}(f(1, 1, 0) - f(0, 1, 0))$$
$$+ \frac{2}{6}(f(1, 1, 1) - f(0, 1, 1))$$
$$+ \frac{1}{6}(f(1, 0, 1) - f(0, 0, 1)).$$

The numerators 1 and 2 in each line correspond to the number of paths that include that edge. For example, there is only one path or permutation that includes the edge from $(0, 1, 0)$ to $(1, 1, 0)$, while there are two paths that include the edge from $(0, 0, 0)$ to $(1, 0, 0)$. In Figure 3 we show that (in green) the four edges corresponding to adding the first feature to a configuration not containing it.

5   Computing Shapley Attributions

In this section we focus on methods to compute the Shapley attribution exactly, or when that is not practical, approximately. We focus on general methods that work without any assumptions about f.

5.1 Exact computation

Computing the Shapley attribution directly using Equation (2) requires taking the average over all $n!$ permuted sequential attributions. In these sequential attributions, we end up evaluating $f(x)$ for the same value of x multiple times. To evaluate the Shapley attribution somewhat more efficiently, we use an alternative formula for the Shapley attribution, which sets the baseline value $b = f(0)$ and the attribution to feature i as

$$a_i = \left(\sum_{x \in \mathcal{X}_i} \frac{(\mathbf{1}^T x)!(n - \mathbf{1}^T x - 1)!}{n!} \right.$$

$$\left. (f(x + e_i) - f(x)) \right) - b. \qquad (3)$$

Here $\mathcal{X}_i$ is the set of configurations with feature i off, i.e., $\mathcal{X}_i = \{x \mid x_i = 0\}$. Using this formula for the Shapley attributions, it can be computed directly from the values of the all 2^n configurations.

We note for future use that the coefficients in the sum in Equation (3) sum to one, so they define a probability distribution on the set of configurations with feature i off. The formula shows that the ith Shapley attribution is the expected value or weighted average of the lift obtained by adding feature i.

5.2 Approximate evaluation

Computing the Shapley attribution requires 2^n simulations, which can be prohibitive when n is large, even just a few tens. In this case, we recommend approximating the Shapley attributions using Monte Carlo sampling methods. We can either sample over sequences of lifts, using the two formulas (2) and (3), each of which expresses the Shapley attributions as an expectation. The idea of sampling sequences has been proposed in Castro *et al.* (2009), but to our knowledge the

method based on sampling lifts has not appeared in the literature.

Sampling sequences. In this method, we use the sum in definition (2) as a basis for Monte Carlo sampling. We sample N permutations of the features, with replacement, and average the permuted sequential attributions corresponding to each. Computing each permuted sequential attribution requires $n + 1$ simulations. (We can reduce the number of simulations required a bit by caching previously computed values of $f(x)$, and using these when $f(x)$ is needed again.)

The expected value of the attributions corresponding to this method are the Shapley attributions. The Monte Carlo attributions are residual free (since each permuted sequential attribution is residual free). The attributions satisfy symmetry approximately, or in expectation.

Sampling lifts. In this method, we use Equation (3) as the basis for Monte Carlo sampling. As noted above, the sum in Equation (3) can be expressed as

$$a_i = \mathbf{E}(f(x + e_i) - f(x)) - b, \qquad (4)$$

where the configuration x is random variable supported over $\mathcal{X}_i$ with probability distribution

$$\mathbf{Prob}(x = x') = \frac{(\mathbf{1}^T x')!(n - \mathbf{1}^T x' - 1)!}{n!}. \qquad (5)$$

To approximate the Shapley attribution of feature i, we first compute $b = f(0)$. We then sample configurations from $\mathcal{X}_i$ with distribution (5), and compute the lift of adding feature i to this configuration. The approximate Shapley attribution is the average of all lifts obtained minus the baseline value. This process is then repeated with each feature. (To sample from distribution (5), first sample the number of active features $\mathbf{1}^T x$, which has a multinomial distribution with outcome probabilities $p_i = (i!(n - i - 1)!)/n!$.

Then randomly sample $\mathbf{1}^T x$ of the n features to be active).

The advantage of this method is that it tends to produce better approximations with fewer simulations than by sampling sequences, because it samples more frequently terms in the sum (Equation (3)) with larger coefficients, therefore forming a more precise approximation of the sum quickly. (This is a form of importance sampling.) Unfortunately, these approximate Shapley attributions are residual free only in the limit as the number of samples grows, or in expectation. This can be remedied by scaling the approximate Shapley attributions to eliminate the residual even for a finite number of samples.

Caching simulations. For both sampling methods, some configurations may appear repeatedly. It is therefore useful to cache the values of configurations, so that they can be re-used in future sampled sequences. If we are asked to evaluate f for an x that has already been evaluated, we simply use the already computed value.

5.3   Example

We now demonstrate the approximation techniques for a simple numerical example in which the metric is the convex quadratic function

$$f(x) = x^T P x,$$

where P is symmetric and positive semidefinite. It can be shown that its Shapley attribution is $b = 0$ and $a = 2P\mathbf{1}$.

We generate P randomly as $P = Z^T Z$, where the entries of Z are independently drawn from a standard normal distribution. (Thus, P has a Wishart distribution with n degrees of freedom and scale matrix I.) We approximate the Shapley attribution using the two methods given above, sampling sequences and sampling lifts, and compare the accuracy as a function of the number of unique configurations x at which we evaluate $f(x)$. (This gives the number of configurations for which we evaluate f, using caching as described above.) When sampling lifts, we consider two versions, the basic (unscaled) version and the version in which we scale to eliminate the residual. We compare the results using the relative error

$$e_{\text{rel}} = \frac{\|\hat{a} - a\|_2}{\|a\|_2},$$

where $a = 2P\mathbf{1}$ is the true Shapley attribution, and $\hat{a}$ is the sampling-based estimate.
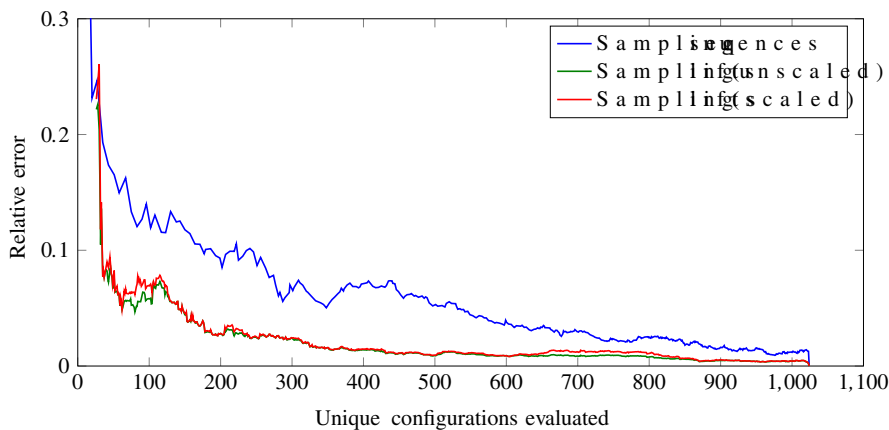
Figure 4   Relative error of the approximate Shapley attributions as a function of the number of unique configurations evaluated, when sampling sequences (blue), sampling lifts (green), and when sampling lifts and scaling to eliminate the residual (red).

The results are shown in Figure 4, for a problem instance with $n = 10$. When sampling sequences, there is no estimate (and therefore no error) until one entire sequence has been evaluated; similarly, when sampling lifts, there is no estimate or error until all features have at least one lift sampled. Both methods converge to the true values once all $2^{10} = 1,024$ configurations have been evaluated. We see that for this example, sampling lifts obtains a lower error than sampling sequences, regardless of the number of configurations evaluated. We also note that when sampling lifts, scaling the approximate Shapley values decreases accuracy, but only slightly. The results of this particular problem instance are typical of many others we have evaluated.

6   Examples

6.1   Simple return attribution

Here we consider a simple return attribution example from Bacon (2008, §5). The performance metric $f(x)$ is the portfolio return over some time period, expressed in percent. We have $n = 2$ features: feature 1 is the country allocation decision (the decision of how much to invest in which country) and feature 2 is the stock selection decision (i.e., the decision of which individual stocks to hold within these countries). (When feature 1 is not active, we invest in each country in proportion to its benchmark weight.

Table 3 Data for the simple returns-based attribution example.

	Benchmark $(0, 0)$	Country alloc. $(1, 0)$	Stock sel. $(0, 1)$	Full portfolio $(1, 1)$
$f(x)$	6.4	5.2	9.4	8.3

When feature 2 is not active, then within each country we invest in each security in proportion to its benchmark weight.) Table 3 shows example data for a single year. We revisit this example in Appendix A, where we further decompose returns by country.

We now discuss, in detail, how to apply the one-at-a-time, sequential, and Shapley attribution methods to this example, with a geometric interpretation given in Figure 5. We will see that the one-at-a-time and sequential methods recover classical attribution methods known in the literature.

One-at-a-time attribution.   One-at-a-time attribution chooses

$$b = f(0, 0), \quad a_1 = f(1, 0) - f(0, 0),$$

$$a_2 = f(0, 1) - f(0, 0).$$

For this example, one-at-a-time attribution is exactly the classical Brinson–Hood–Beebower method (Brinson *et al.*, 1986). As discussed in
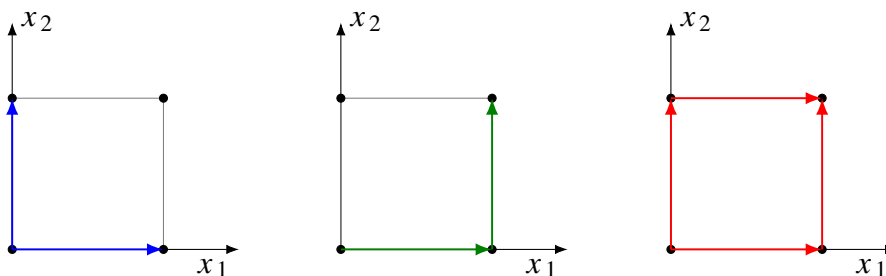
Figure 5 Geometric interpretation of the returns-based attribution example. The one-at-a-time method attributes based on the lifts $f(0, 0) - f(1, 0)$ and $f(0, 0) - f(0, 1)$, shown on the left. The sequential method attributes based on the lifts $f(0, 0) - f(1, 0)$ and $f(1, 0) - f(1, 1)$, shown in the middle. The Shapley attribution averages over all possible lifts, shown on the right.

Section 3.1, this method is not residual free; in fact, the unattributed value is $f(1, 1) - f(1, 0) - f(0, 1) + f(0, 0)$.

Sequential attribution. Sequential attribution chooses

$$b = f(0, 0), \quad a_1 = f(1, 0) - f(0, 0),$$

$$a_2 = f(1, 1) - f(1, 0).$$

This method coincides with the modified Brinson–Hood–Beebower method given in Bacon (2008), where the authors justify choosing feature 1 first in the sequence because the sector allocation decision is often made before security selection decisions. This method eliminates the unattributed value, but violates symmetry by prioritizing the sector allocation decision over the stock selection decision during attribution.

Shapley attribution. Shapley attribution chooses $b = f(0, 0)$ and

$$a_1 = \frac{1}{2}(f(1, 1) - f(0, 1) + f(1, 0) - f(0, 0)),$$

$$a_2 = \frac{1}{2}(f(1, 1) - f(1, 0) + f(0, 1) - f(0, 0)).$$

This is the average of the sequential attributions produced by the two sequences $(0, 0)$–$(1, 0)$–$(1, 1)$, which is the sequence used above, and $(0, 0)$–$(0, 1)$–$(1, 1)$, which is the sequence obtained by making the stock selection decisions *before* the sector allocation decisions.

Results. Table 4 shows the results of applying all three attribution methods. As expected, the one-at-a-time method has a non-zero unattributed return. We can see that in the sequential method, this unattributed return is entirely allocated to stock selection. In the Shapley attribution method, the unattributed term is allocated half to the sector allocation and half to stock selection. Like the sequential method, it has no unattributed component. However, unlike the sequential method, it treats country allocation and stock selection equally, instead of prioritizing country allocation over stock selection. In this simple and small example, the differences in attribution by the different methods are not very significant. In the next section, we will see an example where Shapley attribution is a substantial improvement over competing methods.

6.2 Tax-aware portfolio management

Here we give an example of attribution of multiple performance metrics for a tax-aware portfolio management process. To avoid the wash-sale rule (in which certain capital losses are disallowed), rebalance trades are carried out monthly.

Metrics. We focus on four performance metrics: realized post-tax return, ex-ante risk, realized capital gains, and portfolio turnover. The return, risk, and turnover are annualized. The realized capital gains are reported in dollars over the five-year simulation.

Trading strategy. We simulate an investment strategy based on Markowitz portfolio optimization. In this case, the features correspond to

Table 4 Attribution results for the simple returns-based attribution example.

	Benchmark b	Country alloc. a_1	Stock sel. a_2	Unattributed $y - a_1 - a_2 - b$
One-at-a-time	6.4	−1.2	3.0	0.1
Sequential	6.4	−1.2	3.1	0
Shapley	6.4	−1.15	3.05	0

different terms in the optimization problem that can be on or off. More specifically, given the configuration x with $n = 7$ features, we determine the tradelist by solving the optimization problem

$$\text{maximize} \quad \sum_{i=1}^{5} x_i h^T \alpha^{(i)} - \gamma \sigma^2(h)$$
$$- x_6 \ell(h - h_0) - s^T |h - h_0|.$$
$$\text{subject to} \quad x_7 \sigma(h) \leq \sigma_{\text{lim}} \tag{6}$$
$$\mathbf{1}^T h = 1, \quad h \geq 0.$$

Here the decision variable is the post-trade portfolio h, expressed as a fraction of the account total; the pre-trade portfolio (which is given) is h_0. We describe the objective function and constraints in more detail below.

The first term in the objective function is an expected return forecast, which is divided into the five alpha vectors $\alpha^{(1)}, \ldots, \alpha^{(5)}$, corresponding to the momentum, size, quality, value, and minimum volatility factors. The first five components of x control whether these five alpha vectors are on or off. The second term is the (scaled) squared active risk, defined as

$$\sigma^2(h) = (h - h_b)^T \Sigma (h - h_b),$$

where Σ is the return covariance matrix and h_b is the benchmark portfolio, and $\gamma > 0$ is the risk-aversion parameter. The third term $\ell(h - h_0)$ is the immediate tax liability, due to capital gains, required to reach the post-trade portfolio h, and is parametrized by the long- and short-term capital gains rates, and the tax lots comprising the initial portfolio. (For more details on ℓ, see Moehle *et al.* (2020, §3).) This tax-awareness term can be on or off, depending on x_6. The fourth and last term in the objective is a model of transaction cost, where s is the vector of bid–ask spreads for each asset.

The first constraint is a risk limit with parameter $\sigma_{\text{lim}} > 0$. (When $x_7 = 0$, this constraint is deactivated.) The second constraint is a full-investment constraint, and the last constraint specifies that the portfolio is long only.

Note that when $x = 0$, the portfolio aims to simply track the benchmark portfolio. The full configuration $x = \mathbf{1}$ means that all seven features are on, i.e., we use all five alpha sources, the capital gain objective term, and the risk limit.

Backtests. All of our simulations use the S&P 500 as the benchmark portfolio, with data over the period 2002 to 2019. The alpha was obtained using methods similar to those of Kimura *et al.* (2020). We use the Barra US Equity model (Menchero *et al.*, 2011) to define Σ and h_b, and used the risk-aversion parameter $\gamma = 80$. The tax rates used in ℓ were 0.238 and 0.408, which reflect the current highest marginal tax rates in the United States for long and short-term capital gains, respectively. The simulations take into account transaction costs with the conservative value $s = 0.00051$, i.e., the bid–ask spread is 10 basis points for all assets. The risk limit is $\sigma_{\text{lim}} = 2\%$.

Results. Figures 6 and 7 show the attribution results using Shapley, one-at-a-time, and leave-one-out methods. For each metric, the leftmost set of bars, labeled 'Base', shows the baseline attribution b for each of the three methods. (The attribution to the baseline is the same for all methods, as described in Section 4.) The following seven sets of bars are the attributions a_1, ..., a_7 corresponding to the seven features for each of the three methods. Table 5 shows the unattributed amount $f(\mathbf{1}) - \mathbf{1}^T a - b$ for each of the three methods and four metrics. For comparison, we show the metrics for the baseline and full configuration.

By and large, we see that the same phenomenon occur for all four metrics: one-at-a-time attribution over-attributes, i.e., it overestimates the contribution of each feature, because when only a single feature is included, it drives the portfolio selection process. One the other hand, the leave-one-out attribution under-attributes, i.e., it

underestimates the contribution of each feature, because each single feature makes little difference when competing with the other six. The degree of over- or under-attribution depends on the specific metric and feature in question.

For example, when attributing the risk, this leads to serious problems with the one-at-a-time and leave-one-out attributions that are resolved by Shapley attribution. Under one-at-a-time attribution, the risk limit does not get any 'credit' for risk reduction. This is because the attribution of risk to the risk limit feature is the change in risk by adding it to the benchmark portfolio. Because the benchmark portfolio already has low risk, the risk limit has no effect. On the other hand, each of the five signals, when added to the benchmark portfolio, results in a high risk. Therefore, with one-at-a-time attribution, risk is severely over-attributed to the five signals. This problem is also apparent in Table 5; with the one-at-a-time method, the risk is overattributed, i.e., the sum of the attributions to the features and baseline is 14.7%, which is much greater than the true (full configuration) value of 2.0%.

Leave-one-out attribution also fails to produce a satisfactory result. In this case, leaving out one
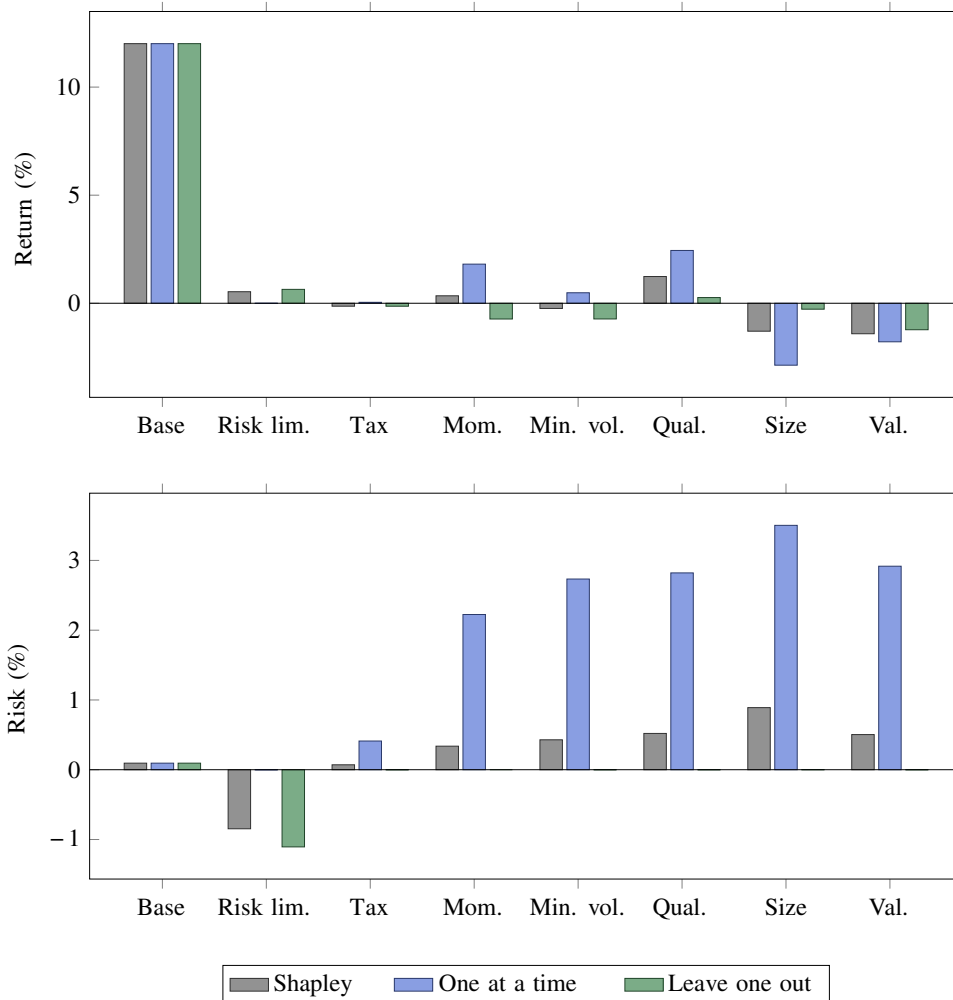
Figure 6 Attributions of return and risk for the tax-aware portfolio management example.
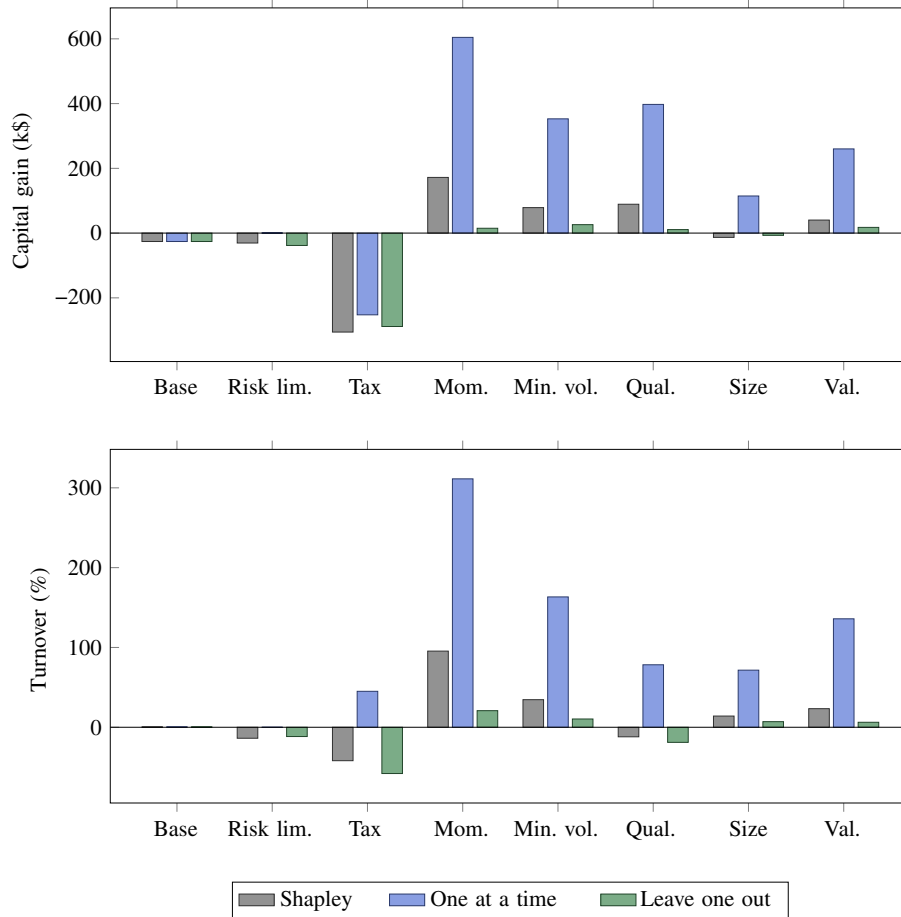
Figure 7 Attributions of capital gains and turnover for the tax-aware portfolio management example.

Table 5 The baseline value $b = f(0)$, full configuration value $f(1)$, as well as the unattributed components $f(1) - 1^T a - b$ for all three attribution methods.

	Return (%)	Risk (%)	Tax (k$)	Turnover (%)
Baseline value $f(0)$	12.0	0.1	−26.0	0.0
Full configuration value $f(1)$	11.0	2.0	4.6	100.2
Unattributed, Shapley	0.0	0.0	0.0	0.0
Unattributed, one-at-a-time	−1.0	−12.7	−1447.1	−705.6
Unattributed, leave-one-out	1.2	3.0	294.6	143.8

signal while keeping the other four does not result in a risk reduction at all, due to the active risk limit. We therefore do not attribute any risk to any of the signals. In fact, we reach the paradoxical conclusion that even though the benchmark has low risk and our realized portfolio has high risk, none of the features are attributed any risk at all. This problem is again reflected in Table 5. With the leave-one-out attribution, the risk is severely underattributed: the sum of the attributions to the

features baseline is -1.0%, while the true risk was 2.0%.

In some cases, the attribution of a metric to a feature has different signs under two different methods. For example, the tax-awareness feature increases turnover when it is the only feature used, because it causes the portfolio to realize losses and keep gains, leading to a positive attribution of turnover to tax awareness under one-at-a-time attribution. When many other features are active, however, the desire to hold onto tax lots with low basis means that tax awareness *decreases* turnover; this is reflected in the negative attribution of turnover to tax awareness with Shapley and leave-one-out methods.

Run time. We note that carrying out a single backtest took around 1 minute for our example. This means that sequentially carrying out the $2^7 = 128$ backtests required to exactly compute the Shapley value take around 2 hours. However, because each backtest can be carried out in parallel, the Shapley value could potentially be computed in a number of minutes.

7 Conclusion

We propose the use of the Shapley value for portfolio performance attribution. Shapley attribution is the only method that possesses four properties that we believe are crucial for satisfactory portfolio performance attribution: symmetry, correct baseline, residual-free attribution, and monotonicity. (A fifth property, additivity, is discussed in Appendix A.) We then compare Shapley attribution with other well-known attribution methods. Compared to other attribution methods, the only disadvantage of Shapley attribution is computational: the number of simulations required to carry out Shapley attribution is exponential in the number of features we attribute to. To overcome this, we recommend two Monte Carlo methods to approximate the Shapley attribution. One of these

is known from the literature, and one is novel, to the best of our knowledge.

A Additivity

In addition to the desiderata of Section 3.1, Shapley attribution is *additive*. This means that if the metric can be decomposed into multiple components, such that $f(x) = f^1(x) + \cdots + f^k(x)$, then the Shapley attribution is given by $a = a^1 + \cdots + a^k$ and $b = b^1 + \cdots + b^k$, where a^i and b^i are the attribution of metric f^i to the features.

This is especially useful when the metric is separable across time. In this case, $f(x)$ is the value of the metric across a large time window (such as a year), and each $f^i(x)$ is the value of the same metric over a shorter time window (such as a month or quarter). Examples of time-separable metrics are log-returns and squared risk.

A.1 Returns-based attribution

Here we return to the returns-based attribution example from Section 6.1, where we now decompose the returns by country. Take $f^{uk}(x)$ to be the weighted return on UK stocks, i.e., it is the portfolio weight in UK stocks multiplied by the return in UK stocks. (Equivalently, it is the value of UK stocks at the end of the investment period divided by the initial portfolio value.) Define $f^{jp}(x)$ and

Table A.1 Data for the returns-based attribution example, when further sub-divided by country.

	Benchmark $(0, 0)$	Country alloc. $(1, 0)$	Stock sel. $(0, 1)$	Full portfolio $(1, 1)$
$f^{uk}(x)$	4	4	8	8
$f^{jp}(x)$	-0.8	-1.2	-1	-1.5
$f^{us}(x)$	3.2	2.4	2.4	1.8
Total, $f(x)$	6.4	5.2	9.4	8.3

Table A.2 Attribution results for the returns-based attribution example, when further subdivided by country.

		Benchmark b	Country alloc. a_1	Stock sel. a_2	Unattr. $f(x) - a_1 - a_2 - b$
One-at-a-time	UK	4	0	4	0
	Japan	−0.8	−0.4	−0.2	−0.1
	US	3.2	−0.8	−0.8	0.2
	Total	6.4	−1.2	3.0	0.1
Sequential	UK	4	0	4	0
	Japan	−0.8	−0.4	−0.3	0
	US	3.2	−0.8	−0.6	0
	Total	6.4	−1.2	3.1	0
Shapley	UK	4	0	4	0
	Japan	−0.8	−0.45	−0.25	0
	US	3.2	−0.7	−0.7	0
	Total	6.4	−1.15	3.05	0

$f^{\text{us}}(x)$ similarly. This means that

$$f(x) = f^{\text{uk}}(x) + f^{\text{jp}}(x) + f^{\text{us}}(x).$$

Table 3 shows example data, which are from Bacon (2008). In particular, the benchmark portfolio weights are 40% (UK), 30% (Japan), and 30% (US), and the portfolio country allocation was 40%, 20%, and 40%, respectively. The benchmark returns, by country, were 10%, −4%, and 8%, respectively, and the by-sector portfolio returns, after stock selection, were 20%, −5%, and 6%, respectively. Combining the data, we obtain the performance metrics shown in Table A.1.

Results. In Table A.2, we show the results of using the three attribution methods from Section 6.1, but now decomposed by country.

Acknowledgments

References

Bacon, C. R. (2008). *Practical Portfolio Performance Measurement and Attribution.* Vol. 546 (John Wiley & Sons).

Brinson, G. P., Hood, L. R., and Beebower, G. L. (1986). "Determinants of Portfolio Performance," *Financial Analysts Journal* **42**(4), 39–44.

Brinson, G. P., Singer, B. D., and Beebower, G. L. (1991). "Determinants of Portfolio Performance II: An Update," *Financial Analysts Journal* **47**(3), 40–48.

Colini-Baldeschi, R., Scarsini, M., and Vaccari, S. (2018). "Variance Allocation and Shapley Value," *Methodology and Computing in Applied Probability* **20**(3), 919–933.

Castro, J., Gómez, D., and Tejada, J. (2009). "Polynomial Calculation of the Shapley Value Based on Sampling," *Computers & Operations Research* **36**(5), 1726–1730.

Fama, E. F. and French, K. R. (2010). "Luck versus Skill in the Cross-Section of Mutual Fund Returns," *The Journal of Finance* **65**(5) 1915–1947.

Grinold, R. C. and Kahn, R. N. (2000). *Active Portfolio Management*, (McGraw Hill New York, NY).

Grinold, R. C. (2006). "Attribution," *The Journal of Portfolio Management* **32**(2), 9–22.

Hiller, T. (2018). "Excluded Coalitions and the 2013 German Federal Election," *Applied Economics Letters* **25**(13), 936–940.

Jensen, M. C. (1968). "The Performance of Mutual Funds in the Period 1945–1964," *The Journal of Finance* **23**(2), 389–416.

Kimura, K., Schwaiger, K., Sharma, D., and Ang, A. (2020). "Factors with Style," SSRN preprint.

Lipovetsky, S. and Conklin, M. (2001). "Analysis of Regression in Game Theory Approach," *Applied Stochastic Models in Business and Industry* **17**(4), 319–330.

Lundberg, S. M. and Lee, S.-I. (2017). "A Unified approach to Interpreting Model Predictions," *Advances in Neural Information Processing Systems*, pp. 4765–4774.

Liben-Nowell, D., Sharp, A., Wexler, T., and Woods, K. (2012). "Computing the Shapley Value in Supermodular Coalitional Games," *International Computing and Combinatorics Conference* (Springer), pp. 568–579.

Lo, A. W. (2007). "Where Do Alphas Come From?: A New Measure of the Value of Active Investment Management," *A New Measure of the Value of Active Investment Management* (May 8, 2007).

Maleki, S., Tran-Thanh, L., Hines, G., Rahwan, T., and Rogers, A. (2013). "Bounding the Estimation Error of Sampling-Based Shapley Value Approximation," arXiv Preprint.

Moehle, N., Kochenderfer, M. J., Boyd, S., and Ang, A. (2020). "Tax-Aware Portfolio Construction via Convex Optimization," arXiv Preprint.

Moulin, H. (2004). *Fair Division and Collective Welfare* (MIT press).

Menchero, J., Orr, D., and Wang, J. (2011). "The Barra US Equity Model (USE4)," *Methodology Notes. English* (MSCI).

Monderer, D., Samet, D. *et al.* (2002). "Variations on the Shapley Value," *Handbook of Game Theory* **3**, 2055–2076.

Mussard, S. and Terraza, V. (2008). "The Shapley Decomposition for Portfolio Risk," *Applied Economics Letters* **15**(9), 713–715.

Ortmann, K. M. (2016). "The Link between the Shapley Value and the Beta Factor," *Decisions in Economics and Finance* **39**(2), 311–325.

Shapley, L. S. (1953). "A Value for n-Person Games," *Contributions to the Theory of Games* **2**(28), 307–317.

Sharpe, W. F. (1992). "Asset Allocation: Management Style and Performance Measurement," *Journal of Portfolio Management* **18**(2), 7–19.

Štrumbelj, E. and Kononenko, I. (2014). "Explaining Prediction Models and Individual Predictions with Feature Contributions," *Knowledge and Information Systems* **41**(3), 647–665.

Tarashev, N., Borio, C., and Tsatsaronis, K. (2010). "Attributing Systemic Risk to Individual Institutions," *Tech. rep. Bank for International Settlements*.

Young, H. P. (1985). "Monotonic Solutions of Cooperative Games," *International Journal of Game Theory* **14**(2), 65–72.