Functional Attribution

Kun Joo Michael Ang

May 14, 2020

Abstract

We develop a framework for attributing the change of a multivariate function's value to changes in the input arguments. Starting with a few desirable basic properties, we discover a few interesting theoretical results and establish links to common discrete allocation algorithms. The paper establishes various algorithms for implementing functional attribution under different schemes. It introduces Functional Coordinate Descent, a function-space analog of the Coordinate Descent algorithm.

Keywords— functional attribution, functional coordinate descent, allocation problems, surface fitting, linear programming, linear analysis

1 Introduction

We consider C^1 functions of the form $f(x_1, x_2, ..., x_N) : \prod_{i=1}^{N}[a_i, b_i] = \Omega \to \mathbb{R}$ and points $\mathbf{x}, \mathbf{x}' \in \Omega$ for $N \geq 2$. In line with various applications in data science and finance, it is often desirable to attribute changes in f to changes in each x_i component. Mathematically, we seek a linear decomposition

$$f(\mathbf{x}') - f(\mathbf{x}) \approx \sum_{i=1}^{N} \psi_i(f, \mathbf{x}', \mathbf{x}) \tag{1}$$

and we call $\psi(f, -, -)$ a **functional attribution** of f. We now introduce two basic properties that are desirable in attributions: null-consistency and completeness.

If $x_i' = x_i \implies \psi_i(f, \mathbf{x}', \mathbf{x}) = 0$, then we say that the attribution ψ is null-consistent with respect to f. ψ is **null-consistent** if it is null-consistent with respect to all $f \in C^1(\Omega)$. If in Equation 1 we have strict equality, we say that ψ is complete with respect to f and similarly, ψ is **complete** if it is complete with respect to all $f \in C^1(\Omega)$.

Unfortunately, null-consistency and completeness are often insufficient in practice. Without imposing additional constraints, we can also see that the existence of one null-consistent and complete method implies the existence of infinitely many such methods and we lack a canonical method of deciding between them.

Let ψ be a null-consistent and complete decomposition. Pick $(\mathbf{x}', \mathbf{x})$ any two points that differ in two or more coordinates. i.e. $x'_i \neq x_i$, $x'_j \neq x_j$. Then

$$\phi_k(f, \mathbf{z}', \mathbf{z}) = \begin{cases} \psi_i(f, \mathbf{x}', \mathbf{x}) + 1 & \text{if } k = i, z' = x', z = x \\ \psi_j(f, \mathbf{x}', \mathbf{x}) - 1 & \text{if } k = j, z' = x', z = x \\ \psi_k(f, \mathbf{z}', \mathbf{z}) & \text{otherwise} \end{cases}$$

creates another decomposition ϕ that is also null-consistent and complete.

2 Scale-Invariance

Let $g : (x_1, x_2, ..., x_N) \longrightarrow (g_1(x_1), g_2(x_2), ..., g_N(x_N))$ be a transformation of the coordinate axes in which each g_i is a univariate, continuous and strictly increasing mapping onto some co-domain in $\mathbb{R}$. Define $\tilde{f}$ as the action of f in the transformed coordinates, i.e. $\tilde{f}(g(\mathbf{x}'), g(\mathbf{x})) \equiv f(\mathbf{x}', \mathbf{x})$. Then we can say that the attribution ψ is scale-invariant with respect to f if $\forall\, i, \mathbf{x}', \mathbf{x}$, we have $\psi_i(f, \mathbf{x}', \mathbf{x}) = \psi_i(\tilde{f}, g(\mathbf{x}'), g(\mathbf{x}))$. ψ is **scale-invariant** if it is scale-invariant with respect to all $f \in C^1(\Omega)$.

2.1 Stepwise Allocations

A common functional attribution method involves taking discrete steps from $\mathbf{x}$ to $\mathbf{x}'$ in only the coordinates where they differ. This stepwise method is null-consistent, complete and scale-invariant and its precise statement is as follows.

For any pair of points $(\mathbf{x}', \mathbf{x})$, let $S = \left\{S_1, S_2, ..., S_p \,|\, \mathbf{x}'_{\mathbf{S_i}} \neq \mathbf{x}_{\mathbf{S_i}}\right\}$ be the set of indices where their coordinate values differ, and the S_i are arranged in some pre-specified order. Starting from $\mathbf{x}^* = \mathbf{x}$, we change the S_1-coordinate value to match $\mathbf{x}'$, then assign the marginal change in f as the S_1-attribution. We repeat this process until $\mathbf{x}^* = \mathbf{x}'$ and arrive at the following attribution

$$\psi_k(\mathbf{x}', \mathbf{x}) = \begin{cases} f\left(\mathbf{x} + \sum\limits_{n \leq i} \left(x'_{S_n} - x_{S_n}\right)\right) - f\left(\mathbf{x} + \sum\limits_{n < i} \left(x'_{S_n} - x_{S_n}\right)\right) & \text{if } k = S_i \\ 0 & \text{if } k \notin S \end{cases}$$

It is easy to check that the stepwise allocation is, in general, dependent on the ordering of indices S_i (e.g. $f(x, y) = xy$, $\mathbf{x} = (0, 0)$, $\mathbf{x}' = (1, 1)$). But if the ordering is fixed or chosen in some coordinate-free manner, then the resulting allocation will be scale-invariant. Null-consistency and completeness are clear by construction.

In lieu of having to specify a natural ordering of the coordinate axes, one could instead ask when the stepwise allocation scheme generates functional attributions $\psi(f, -, -)$ that are independent of the coordinate ordering. This concept is closely related to the idea of 'path-invariance' that we will explore Section 4. In fact, one can also use the proof in Section 4 to show that the linear form $f(\mathbf{x}) = \sum_{i=1}^{N} f_i(x_i)$ is both a necessary and sufficient condition for the stepwise allocation to be order-independent for all $(\mathbf{x}', \mathbf{x})$ pairs.

The Shapley Allocation is a path-agnostic approach that first generates all possible permutations of indices in S, performs the functional attribution for each permutation, then computes each coordinate's attribution as its average across all permutations. An even faster approach is the Greedy Allocation, which uses only one permutation, but selects the S_i iteratively to minimize difference in function value. In the event of a tie between coordinates in the Greedy Algorithm, coordinates can either be chosen randomly, or taken to be the one which allows the minimum absolute function difference in the following step. Because the Greedy Algorithm relies only on the difference in function values and not the coordinate axes units, the algorithm is also scale-invariant.

Algorithm 1 Shapley Allocation

```
Require: $\mathbf{x}' \neq \mathbf{x}$
  $S = \{S_i \,|\, \mathbf{x}'_{S_i} \neq \mathbf{x}_{S_i}\}, \quad k = |S|$
  for $\sigma \in \ Sym(k)$ do
    $\psi_{S_i}(\sigma) = f\Big(\mathbf{x} + \sum_{\sigma(n) \leq \sigma(i)} \big(x'_{S_n} - x_{S_n}\big)\Big) - f\Big(\mathbf{x} + \sum_{\sigma(n) < \sigma(i)} \big(x'_{S_n} - x_{S_n}\big)\Big)$
  end for
  return $\psi_j = \begin{cases} \frac{1}{|Sym(k)|} \sum_{\sigma} \psi_j(\sigma) & \text{if } j \in S \\ 0 & \text{otherwise} \end{cases}$
```

Algorithm 2 Greedy Allocation

```
Require: $\mathbf{x}' \neq \mathbf{x}$
  $S = \{i \,|\, \mathbf{x}'_i \neq \mathbf{x}_i\}$
  $\psi_i = 0$  if $i \notin S$
  $\mathbf{x}^* = \mathbf{x}$
  while $S \neq \emptyset$ do
    $k = \underset{k \in S}{\operatorname{argmin}} \Big| f\big(\mathbf{x}^* + (x'_k - x_k)\big) - f\big(\mathbf{x}^*\big) \Big|$
    $\psi_k = f\big(\mathbf{x}^* + (x'_k - x_k)\big) - f\big(\mathbf{x}^*\big)$
    $S = S \backslash \{k\}$
    $\mathbf{x}^* = \mathbf{x}^* + (x'_k - x_k)$
  end while
  return $\psi$
```

3 Transitivity

We say an attribution ψ is transitive with respect to a function f if, for all coordinates i and triplets $(\mathbf{x}, \mathbf{x}', \mathbf{x}'')$, we have $\psi_i(f, \mathbf{x}, \mathbf{x}'') = \psi_i(f, \mathbf{x}, \mathbf{x}') + \psi_i(f, \mathbf{x}', \mathbf{x}'')$. ψ is **transitive** if it is transitive with respect to all $f \in C^1(\Omega)$.

One important observation that will be useful in Section 4 is that transitivity and null-consistency are unrelated properties. To better demonstrate that fact, consider the following example.

Let $\Omega = [0,1]^2$ and $f((x_1,x_2)) = x_1+x_2$. Define $\psi_i(f,\mathbf{x},\mathbf{x}') = x_i'-x_i$ for $i = \{1,2\}$ as a functional attribution ψ of f. It is easy to verify that ψ is null-consistent and transitive with respect to f. Now define the regions $A = [0,0.5] \times [0,1]$ and $B = (0.5,1] \times [0,1]$ and the following functional attributions.

$$\phi_i(f,\mathbf{x},\mathbf{x}') = \begin{cases} 2 & i = 1,\ \mathbf{x} = (0,0),\ \mathbf{x}' = (1,1) \\ 0 & i = 2,\ \mathbf{x} = (0,0),\ \mathbf{x}' = (1,1) \\ \psi_i(f,\mathbf{x},\mathbf{x}') & \text{otherwise} \end{cases}$$

$$\theta_i(f,\mathbf{x},\mathbf{x}') = \begin{cases} \psi_i(f,\mathbf{x},\mathbf{x}') + \mathbb{1}_{\{i=1\}} - \mathbb{1}_{\{i=2\}} & \mathbf{x} \in A,\ \mathbf{x}' \in B \\ \psi_i(f,\mathbf{x},\mathbf{x}') - \mathbb{1}_{\{i=1\}} + \mathbb{1}_{\{i=2\}} & \mathbf{x} \in B,\ \mathbf{x}' \in A \\ \psi_i(f,\mathbf{x},\mathbf{x}') & \text{otherwise} \end{cases}$$

It can be verified that with respect to f, ϕ is null-consistent but not transitive and θ is transitive but not null-consistent. As a side remark, all three attributions are complete with respect to f and if we define the component attribution as the component-wise contributions to $f((x_1,x_2)) = x_1+x_2$, then it is possible to define ψ,ϕ,θ in a way that they are also scale-invariant with respect to f.

4 Line Integrals and Path-Invariance

Notice that in all the properties discussed thus far, all definitions still hold even for $f \notin C^1(\Omega)$. In this section, we will explore how the differentiability of f induces a natural functional attribution via a line integral.

Start by defining $C(\mathbf{x},\mathbf{x}')$ as a function that returns a continuous path from $\mathbf{x} \longrightarrow \mathbf{x}'$. Examples include a straight line in $\mathbb{R}^N$, or an ordered linear combination of the N basis lines. Since $f \in C^1(\Omega)$, we can express the differential $df = \sum_{i=1}^N \frac{\partial f}{\partial x_i} dx_i$, and integrate both sides to get

$$f(\mathbf{x}') - f(\mathbf{x}) = \sum_{i=1}^N \int_{C(\mathbf{x},\mathbf{x}')} \frac{\partial f}{\partial x_i} dx_i$$

which we notice forms a linear decomposition into N components. Therefore, by specifying C, a method of connecting points in the domain, this induces the attribution $\psi^C(-,-,-)$. We refer to this attribution as the **natural decomposition** of f with respect to a path function C. The stepwise allocation above is one such example.

We will now investigate some properties of ψ^C. Completeness is easy to check. To show how null-consistency and transitivity are not guaranteed, consider the following scenario in Figure 1 where $f(x,y) = xy$, $\Omega = [0,1]^2$, $\mathbf{x} = (0,0)$ and $\mathbf{x}' = (1,1)$. Let $C(\mathbf{x},\mathbf{x}') = A$, $C(\mathbf{x}',\mathbf{x}) = B$ and $C(\mathbf{x},\mathbf{x}) = \{\mathbf{x}\}$. The path $\mathbf{x} \longrightarrow \mathbf{x}' \longrightarrow \mathbf{x}$ shows that ψ^C is not transitive. If we instead define $C'(\mathbf{x},\mathbf{x}) = A + B$, then $\psi^{C'}$ is not null-consistent. This example also shows how the sensitive the functional attributions the way C is specified. Using the same domainan and function, define $C(\mathbf{x},\mathbf{x}')$ to be the straight line connecting $\mathbf{x}$ to $\mathbf{x}'$, independent of f. In the example above this

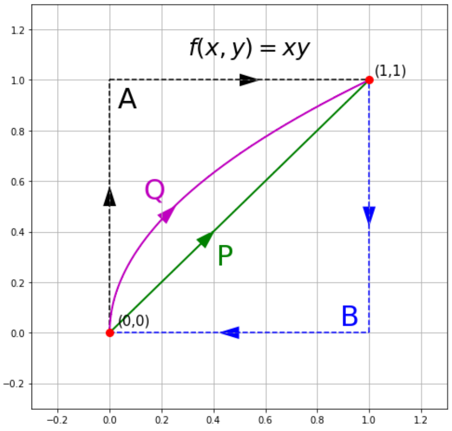

Figure 1: Paths from $(0, 0)$ to $(1, 1)$

corresponds to path P. Now apply a non-linear transformation $g(\mathbf{x}) = \tilde{\mathbf{x}}$ and let $\tilde{C}$ be the new straight-line path. The attributions are now

$$\int_{\tilde{C}} \frac{\partial f}{\partial \tilde{x}_i} d\tilde{x}_i = \int_{\tilde{C}} \frac{\partial f}{\partial \tilde{x}_i} \frac{\partial \tilde{x}_i}{\partial x_i} dx_i = \int_{\tilde{C}} \frac{\partial f}{\partial x_i} dx_i$$

which correspond to the original integral along a different path. In the example above, $g((x, y)) = (\sqrt{x}, y)$ and the new path corresponds to Q. We can verify that $\psi_1^C(f, \mathbf{x}, \mathbf{x}') = \frac{1}{2}$ and $\psi_1^{\tilde{C}}(\tilde{f}, g(\mathbf{x}), g(\mathbf{x}')) = \frac{2}{3}$ is not scale-invariant under the original definition of C. This can be fixed by having $C(\tilde{f})$ detect coordinate deformations relative to f and choosing the path that maps to P in the reference basis. But this is undesirable as it requires selecting a canonical basis, and in practice there might not be a strong reason to prefer one over another.

There are two solutions to restoring scale-invariance. One method is to restrict C to paths that are piecewise combinations of the N basis lines. These straight line paths map to the same straight line paths under any reparameterization because the coordinate axis transforms are univariate. Alternatively, we could restrict f to functions where the attribution integrals are path-independent. More precisely, $\forall i, C(-, -)$, $\int_C \frac{\partial f}{\partial x_i} dx_i = \psi_i^*(\mathbf{x}, \mathbf{x}')$. When this is true, we say that the natural decomposition of f is **path-invariant**. Note that path-invariance is a stronger condition than scale-invariance, since the path deformations are not restricted to continuous, strictly increasing rescalings of the coordinate axes, but over all path deformations with the same start and end points.

It turns out that if f can be written as $f(\mathbf{x}) = \sum_{i=1}^N f_i(x_i)$ for some $f_i \in C^1[a_i, b_i]$, this is both a necessary and sufficient condition for the natural decomposition to be path invariant. Sufficiency is easy to check as each coordinate's attribution becomes $f_i(x_i') - f_i(x_i)$. To show necessity, we prove the contrapositive.

Suppose f cannot be written in the form $f(\mathbf{x}) = \sum_{i=1}^N f_i(x_i)$. Recall the differential form $df = \sum_{i=1}^{N} \frac{\partial f}{\partial x_i} dx_i$ which holds for any path integral since $f \in C^1(\Omega)$. Then at least one of the partial derivatives $\frac{\partial f}{\partial x_i}$ must have dependence on other coordinates $x_{j_1}, ..., x_{j_k}$ where $i \notin \{j_1, ..., j_k\}$, otherwise integrating both sides of the differential leads to a contradiction. There therefore must exist points $\mathbf{x} \neq \mathbf{x}'$ such that $x_i = x_i'$ but $\frac{\partial f}{\partial x_i}(\mathbf{x}) \neq \frac{\partial f}{\partial x_i}(\mathbf{x}')$. Without loss of generality, let $\frac{\partial f}{\partial x_i}(\mathbf{x}') > \frac{\partial f}{\partial x_i}(\mathbf{x})$ and consider the following paths in Figure 2. $A : \mathbf{x} \longrightarrow (\mathbf{x} + \varepsilon x_i)$, $B : (\mathbf{x} + \varepsilon x_i) \longrightarrow (\mathbf{x}' + \varepsilon x_i)$, $B' : \mathbf{x} \longrightarrow \mathbf{x}'$ and $A' : \mathbf{x}' \longrightarrow (\mathbf{x}' + \varepsilon x_i)$. Along paths A and A', we change only the x_i components by increasing them monotonically, and along paths B and B', we do not change the x_i components. Then the functional attribution of $f(\mathbf{x}' + \varepsilon x_i) - f(\mathbf{x})$ in the x_i component is represented by the path integrals along A and A' in the two paths. Since $f_i \in C^1$, $\frac{\partial f}{\partial x_i}(\mathbf{x}') > \frac{\partial f}{\partial x_i}(\mathbf{x})$ means that for $\varepsilon > 0$ sufficiently small, $\int_{A'} \frac{\partial f}{\partial x_i} dx_i > \int_A \frac{\partial f}{\partial x_i} dx_i$ and by comparing ψ_i^C along paths $A + B$ and $B' + A'$, we have that the natural decomposition of f is not path-invariant.

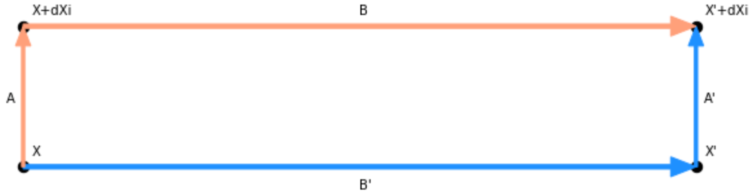

Figure 2: Paths from X to $X' + dX_i$

Lemma 1. *The following are equivalent:*

1. *The natural decomposition of f is path-invariant.*
2. *There exists a decomposition of the form $f(\mathbf{x}) = \sum_{i=1}^{N} f_i(x_i)$*
3. *$\psi^C(f, -, -)$ is null-consistent with respect to f for all $C \in \Theta$*
4. *$\psi^C(f, -, -)$ is transitive with respect to f for all $C \in \Theta$*

where $\Theta := \{M(\mathbf{x}, \mathbf{x}') : [0, 1] \to \Omega \,\big|\, M(0) = \mathbf{x},\, M(1) = \mathbf{x}',\, M_i \in C^0[a_i, b_i]\}$ is the set of path functions that connect any $(\mathbf{x}, \mathbf{x}')$ pair via a continuous path.

Proof. (1) $\iff$ (2): Shown above.

(2) $\iff$ (3): " $\Longrightarrow$ " is clear. " $\Longleftarrow$ " is shown by following the above construction and considering the path $C(\mathbf{x}, \mathbf{x}') = A + B - A'$ and checking that $\psi_i^C(\mathbf{x}, \mathbf{x}') < 0$.

(4) $\iff$ (2): " $\Longrightarrow$ " is clear. " $\Longleftarrow$ " is shown by following the above construction and considering the paths $C(\mathbf{x}, \mathbf{x}' + \varepsilon x_i) = A + B$, $C(\mathbf{x}' + \varepsilon x_i, \mathbf{x}') = -A'$, $C(\mathbf{x}, \mathbf{x}') = B'$, and checking that $\psi_i^C(\mathbf{x}, \mathbf{x}') = 0 \neq \psi_i^C(\mathbf{x}, \mathbf{x}' + \varepsilon x_i) + \psi_i^C(\mathbf{x}' + \varepsilon x_i, \mathbf{x}')$. □

As a final remark, functional forms $f(\mathbf{x}) = \sum_i f_i(x_i)$ are highly desirable because they induce a natural decomposition for functional attribution without the need to specify a path function connecting points in Ω. We get null-consistency and transitivity from Lemma 1, completeness by construction and scale-invariance from path-invariance.

5 Surface Fitting

Suppose now that we are seeking a functional attribution but our underlying function is not of the form $f(\mathbf{x}) = \sum_{i=1}^N f_i(x_i)$. One compromise is to somehow approximate f with the surface $f(\mathbf{x}) \approx \hat{f}(\mathbf{x}) = \sum_{i=1}^N \hat{f}_i(x_i)$. The functional attribution is induced by $\hat{f}$ and the residual $\varepsilon(\mathbf{x}) = f(\mathbf{x}) - \hat{f}(\mathbf{x})$ is treated as surface 'noise'. This approach allows our attribution to have all above properties except completeness. We will now investigate some techniques for computing $\hat{f}$ by minimizing $\|\varepsilon\|$ under various norms.

5.1 Weighted L_2 Norm - Generalized

Let Ω_i denote the domains for each of the coordinates x_i. We define the weighted inner product

$$(f,g) = \int_{\Omega_1} \cdots \int_{\Omega_N} f(X_1,\dots,X_N)g(X_1,\dots,X_N)w(X_1,\dots,X_N)dX_1\dots dX_N$$

where $w : \Omega \to \mathbb{R}_{>0}$ is positive, continuous and integrable over the full domain Ω. Minimizing the residual with respect to the weighted norm yields the following optimization:

$$\mathcal{L} = \min_{f_1,\dots,f_N} \int_{\Omega} \big(f - \sum_{k=1}^N f_k\big)^2 w \; dX \tag{2}$$

For notational convenience, we suppress the arguments of each function and denote $dX := dX_1 ... dX_N$. Furthermore, let Ω_{-i} denote the integral over all domains excluding Ω_i, and dX_{-i} its corresponding differential. The minimization can be written as

$$\min_{f_1,\dots,f_N} \int_{\Omega_i} \left[\int_{\Omega_{-i}} R_i^2 w \; dX_{-i}\right] - 2f_i \left[\int_{\Omega_{-i}} R_i w \; dX_{-i}\right] + f_i^2 \left[\int_{\Omega_{-i}} w dX_{-i}\right] dX_i$$

where $R_i := f - \sum\limits_{k\neq i} f_k$ are the fitted residuals excluding f_i.

Applying the Euler-Lagrange equation in each X_i coordinate means that at any extremum point $\sum f_i^*$,

$$f_i^*(X_i) = \frac{\displaystyle\int_{\Omega_{-i}} (f - \sum_{k\neq i} f_k^*) w \; dX_{-i}}{\displaystyle\int_{\Omega_{-i}} w dX_{-i}} \tag{3}$$

Unfortunately, the optimal f_i^* are implicit solutions to a system of N integral equations. Depending on the X_i-dependence in w, neat closed-form solutions might not exist. We will see later that if w is separable, i.e. $w = \prod_i w_i(X_i)$, then these integrals simplify somewhat. But for general weight functions, the implicit dependence motivates the creation of Algorithm 3, which we call Functional Coordinate Descent(FCD).

A few quick remarks about FCD in the context of our weighted L_2 norm: Using a small change of notation, $\mathcal{L}$ refers to the weighted error defined in Equation 2 as a function of $(f_1, f_2, ..., f_N)$, rather than the scalar minimum itself. In Line 5 of Algorithm 3, the minimum can be directly computed using Equation 3 as

$$f_i^s(X_i) = \left[\int_{\Omega_{-i}} \left(f - \sum_{k<i} f_k^s - \sum_{k>i} f_k^{s-1}\right) w \, dX_{-i}\right] \left[\int_{\Omega_{-i}} w dX_{-i}\right]^{-1}$$

Algorithm 3 Functional Coordinate Descent

```
Require: Initial $f_i^0$ for $i = 1, 2, ..., N$
         Functional $\mathcal{L} : (f_1, f_2, ..., f_N) \to \mathbb{R}$
         Error tolerance $\tau$
 1: $s = 0$, converged = False
 2: while ¬ converged do
 3:     $s = s + 1$
 4:     for $i \in [1, 2, ..., N]$ do
 5:         $f_i^s(X_i) = \underset{g(X_i)}{\operatorname{argmin}} \mathcal{L}(f_1^s, ..., f_{i-1}^s, g, f_{i+1}^{s-1}, ..., f_N^{s-1})$
 6:     end for
 7:     $\mathcal{L}_s = \mathcal{L}(f_1^s, f_2^s, ..., f_N^s)$
 8:     if $\mathcal{L}_{s-1} - \mathcal{L}_s < \tau$ then
 9:         converged = True
10:     end if
11: end while
12: return $f_i^s \quad i = 1, 2, ..., N$
```

We will now prove some convergence properties of FCD. In standard Coordinate Descent(CD), if $\mathcal{L}(x) = g(x) + \sum_i h(x_i)$ where g is convex, differentiable and each h_i convex, then CD converges to the true minimizer x^*. Not surprisingly, there is an analagous result in FCD, provided we impose a few additional constraints.

Theorem 1. *Let $\vec{f} = (f_1, ..., f_N) \in \Theta$ denote a set of solution vectors where the arguments $f_i : \Omega_i \to \mathbb{R}$ are functions and Θ compact. Let $\| \cdot \|_\theta$ be a norm over Θ and $\mathcal{L} : \vec{f} \to \mathbb{R}$ a positive strictly convex, Gateaux differentiable, continuous functional on Θ. Assume Step 5 of the FCD algorithm always returns solution vectors in Θ provided $\vec{f^0} \in \Theta$. Then if $\exists \, \vec{f^*} \in \Theta$ such that either*

1. *$\exists N \in \mathbb{N}$ such that $\forall s \geq N$, $\vec{f^s} \equiv \vec{f^*}$*
2. *$\vec{f_i^s} \to \vec{f^*}$ and $\mathcal{L}$ Lipschitz.*

then $\vec{f^*}$ is the unique minimizer of $\mathcal{L}$ over Θ. Here $\vec{f^s}$ denotes the sequence of iterates from Step 7 of FCD and $\vec{f_i^s}$ denotes the sequence of sub-iterates from Step 5. Each while loop adds 1 and N terms to each sequence respectively.

Proof. Before we begin the proof, we must state that two solution vectors are defined to be equal if their inner product (defined by $\| \cdot \|_\theta$) is 0. This does not imply that their corresponding functions have to be pointwise identical in each of the N indices. For example, we can take $\| \cdot \|_\theta$ to be the L_2-norms summed over the indices and Θ as the space of solution vectors of N square-integrable functions. Then two solution vectors that differ at only finitely many points in each index will have 0 inner product. In the proof below, when we refer to a solution vector $\vec{f}$, we are referring to the conjugacy class of solutions which contains $\vec{f}$ and Θ as the space of conjugacy classes.

We can now make some statements about $\mathcal{L}$ over Θ. Since Θ is compact and $\mathcal{L}$ continuous and bounded below (by positivity), the Extreme Value Theorem tells us it attains a minimum over Θ at some $\vec{f}^*$. These compactness and boundedness properties can be relaxed if we instead directly assume a minimizer exists. $\vec{f}^*$ must also be unique by strict convexity of $\mathcal{L}$, otherwise $\mathcal{L}(\frac{1}{2}(\vec{f}_1^* + \vec{f}_2^*)) < \mathcal{L}(\vec{f}_1^*)$, violating the minimality of $\vec{f}_1^*$.

$\psi \in \Theta$ is said to be unidimensional if $\exists\, i \in \{1, 2, ..., N\}$ s.t. $\psi_i \not\equiv 0$ and $\forall j \neq i$, $\psi_j \equiv 0$. $\vec{f}$ is said to be a stationary point in Θ if its Gateaux derivative

$$d(\vec{f}, \psi) = \lim_{s \to 0} \frac{\mathcal{L}(\vec{f} + s\psi) - \mathcal{L}(\vec{f})}{s}$$

is 0 for all unidimensional ψ. Clearly, $\vec{f}^*$ must be a stationary point.

Additionally, $\vec{f}^*$ is the only stationary point in Θ. Suppose otherwise, that $\vec{g}$ is also stationary and define $\phi = (\vec{f}^* - \vec{g})$ and $\Delta = \mathcal{L}(\vec{f}) - \mathcal{L}(\vec{g}) < 0$. By convexity,

$$\forall\, \varepsilon \in (0, 1) \qquad \mathcal{L}(\vec{g} + \varepsilon\phi) < \mathcal{L}(\vec{g}) + \varepsilon\Delta$$

Now consider and $\mathcal{L}(\vec{g} + \varepsilon\phi)$ in the limit as $\varepsilon \to 0$ and we get $d(\vec{g}, \phi) = -\delta$ for some $\delta < 0$. This means we must have $\mathcal{L}(\vec{g} - \varepsilon\phi) = \mathcal{L}(\vec{g}) + \varepsilon\delta + o(\varepsilon)$. Now let $\phi \equiv \sum_{i=1}^{N} \phi_i$ be the unique unidimensional decomposition of ϕ. Then

$$\begin{aligned} \vec{g} + \varepsilon\phi = \frac{1}{N}\sum_{i=1}^{N}(\vec{g} - \varepsilon N\phi_i) \implies \mathcal{L}(\vec{g} - \varepsilon\phi) &< \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}(\vec{g} - \varepsilon N\phi_i) \\ &= \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}(\vec{g}) + o(\varepsilon N) \\ &= \mathcal{L}(\vec{g}) + o(\varepsilon) \end{aligned}$$

where we first use convexity of $\mathcal{L}$ and then the stationarity of $\vec{g}$. But this is a contradiction, since $\mathcal{L}(\vec{g} - \varepsilon\phi) - \mathcal{L}(\vec{g}) = \varepsilon\delta + o(\varepsilon)$.

Our goal now is to show that the FCD algorithm converges to this unique stationary point. Since the N sub-iterates in Steps 4-6 of the algorithm are decreasing in value, $\mathcal{L}_s$ must also form a positive, decreasing sequence. We now assert that if at some s we have $\mathcal{L}_{s-1} = \mathcal{L}_s$, then $\vec{f}^k \equiv \vec{f}^{s-1}$ for all $k \geq s$. First label the sub-iterates as $\vec{f}^s(k) = (f_1^s, ..., f_k^s, f_{k+1}^{s-1}, ..., f_N^{s-1})$, where $\vec{f}^s(0) = \vec{f}^{s-1}$ and $\vec{f}^s(N) = \vec{f}^s$. If $\mathcal{L}_{s-1} = \mathcal{L}_s$, then $\mathcal{L}(\vec{f}^s(k))$ must also be constant in k. If $\vec{f}^{s-1} \not\equiv \vec{f}^s$, then there exists some smallest index i where $\vec{f}^s(i) \neq \vec{f}^s(i-1)$. But since all $\vec{f}^s(k) \in \Theta$ and $\mathcal{L}$ strictly convex by assumption, the solution $\vec{f}^+ = \frac{1}{2}[\vec{f}^s(i) + \vec{f}^s(i-1)]$ must have $\mathcal{L}(\vec{f}^+) < \mathcal{L}(\vec{f}^s(i-1))$. But since $\vec{f}^+$ differs from $\vec{f}^s(i-1)$ in only the i^{th} functional, this contradicts the minimality of Step 5. Therefore, $\vec{f}^{s+1} \equiv \vec{f}^s$. It is easy to see that $\vec{f}^s$ is stationary, $\vec{f}^k \equiv \vec{f}^{s-1}$ for all $k \geq s$ and we have converged to the unique minimizer.

If instead $\mathcal{L}_s$ is strictly decreasing, apply the assumption that $\vec{f}_i^s \to \vec{f}^+$ for some cluster point $\vec{f}^+ \in \Theta$. We will now show that this cluster point is stationary and therefore minimal. Suppose otherwise, then there exists some scalar k, coordinate i and unidimensional ψ_i such that $\mathcal{L}(\vec{f}^+ + k\psi_i) < \mathcal{L}(\vec{f}^+)$. Let K be the Lipschitz constant of $\mathcal{L}$. Then by picking $\varepsilon = \frac{\mathcal{L}(\vec{f}^+)-\mathcal{L}(\vec{f}^+ + k\psi_i)}{3K}$, and applying the Lipschtiz Inequality, $\exists\, N$ such that $\forall s > N$, $|\mathcal{L}(\vec{f}_i^s) - \mathcal{L}(\vec{f}^+)| \leq \frac{\mathcal{L}(\vec{f}^+)-\mathcal{L}(\vec{f}^+ + k\psi_i)}{3}$. But applying the same Lipschtiz inequality also yields $|\mathcal{L}(\vec{f}_i^s + k\psi_i) - \mathcal{L}(\vec{f}^+ + k\psi_i)| \leq \frac{\mathcal{L}(\vec{f}^+)-\mathcal{L}(\vec{f}^+ + k\psi_i)}{3}$ and we have $\mathcal{L}(\vec{f}_i^s + k\psi_i) < \mathcal{L}(\vec{f}_i^s)$, which directly contradicts Step 5 of the FCD algorithm.

If the coordinate axes are orthogonal, which is to say that for any $f = (f_1, f_2, ..., f_N)$, $\|f\|_\Theta = \sum_{k=1}^N \|\psi_k\|_\Theta$ where $\psi_i = (0, ..., 0, f_i, 0, ..., 0)$, then convergence of the sub-iterates $\vec{f}_i^s \to \vec{f}^*$ can be replaced by convergence of the iterates $\vec{f}^s \to \vec{f}^*$. □

Let's apply the FCD algorithm with $\mathcal{L}$ as the weighted L_2 error defined in Equation 2. Pick Θ to be the space of solution vectors that are L_2 integrable in each coordinate and $\|(f_1, ..., f_N)\|_\Theta = \sum_i \|f_i\|_{L^2}$. We can check that $\mathcal{L}$ is indeed positive, strictly convex, Gateaux-differentiable and continuous. It is also easy to intuit from the continuity of $\mathcal{L}$ and the observation that $\mathcal{L}(\vec{f}) \to \infty$ as $\|\vec{f}\|_\Theta \to \infty$ that a minimizer $\vec{f}^*$ exists. If f is L_2-integrable over Ω and f_i^0 L_2-integrable in each coordinate i, then $\forall\, s, i$, the sub-iterates $f_i^s(X_i)$ will also be integrable. These conditions allow us to apply the results of Theorem 1.

5.2 Weighted L_2 Norm- Separable

If the weighting function is separable, $w(X_1, X_2, ..., X_N) = \prod_i w_i(X_i)$, then the f_i^* can instead be computed explicitly. Starting with Equation 3,

$$\begin{aligned}
f_i^*(X_i) &= \frac{\displaystyle\int_{\Omega_{-i}} (f - \sum_{k\neq i} f_k^*) w \, dX_{-i}}{\displaystyle\int_{\Omega_{-i}} w dX_{-i}} \\
&= \frac{\displaystyle\int_{\Omega_{-i}} fw \, dX_{-i}}{\displaystyle\int_{\Omega_{-i}} w dX_{-i}} - \frac{\cancel{w_i} \displaystyle\int_{\Omega_{-i}} \sum_{k\neq i} f_k^* \left(\prod_{s\neq i} w_s \right) dX_{-i}}{\cancel{w_i} \displaystyle\prod_{s\neq i} \left[\int_{\Omega_s} w_s dX_s \right]} \\
&= \frac{\displaystyle\int_{\Omega_{-i}} fw \, dX_{-i}}{\displaystyle\int_{\Omega_{-i}} w dX_{-i}} - \lambda_i
\end{aligned}$$

where in the second line we observe that the second term is independent of X_i and replace it with a scalar λ_i. This is well-defined, as positivity in w guarantees that w_i and $\int_{\Omega_s} w_s dX_s$ are non-zero. The solutions $f_i^*(X_i)$ can be interpreted as the weighted average of $f(-, X_i, -)$ over the other coordinate axes.

For the purpose of functional attribution, it is sufficient to solve f_i^* as above with

$\lambda_i \equiv 0$. To recover the optimal fit, we can write $\hat{f} = \sum_i f_i^* + \lambda$ and recover $\lambda = \Big[\int_\Omega (f - \sum_i f_i^*) w dX\Big]\Big[\int_\Omega w dX\Big]^{-1}$.

5.3 Other Norms

Denoting the surface residual $E = f - \sum_i f_i$, we might wish to penalize not just its absolute error, but also its derivative. This introduces the weighted H^1-norm

$$\min_{f_1,\ldots,f_N} \int_\Omega E^2 w_1(X) + \left[\left(\frac{\partial E}{\partial X_1}\right)^2 + \ldots + \left(\frac{\partial E}{\partial X_N}\right)^2\right] w_2(X)\, dX$$

where we allow $w_1 \neq w_2$. Once again, we rewrite the integral.

$$\min_{f_1} \int_{\Omega_i} \left[A(X_i) f_i^2 - 2B(X_i, R_i) f_i + C(X_i)\left(\frac{\partial f_i}{\partial X_i}\right)^2 - 2D(X_i)\frac{\partial f_i}{\partial X_i}\right] dX$$

$$A = \int_{\Omega_i} w_1\, dX_{-i} \qquad B = \int_{\Omega_i} w_1 R_i\, dX_{-i}$$

$$C = \int_{\Omega_i} w_2\, dX_{-i} \qquad D = \int_{\Omega_i} w_2 \frac{\partial f}{\partial X_i}\, dX_{-i}$$

Because the integrand now involves the first derivative, we must be more careful with our boundary conditions. Let's impose that A, B, C, D all have continuous partial X_i-derivatives and that all the first-order partial derivatives in every coordinate vanish at the boundaries. Then an application of the Euler-Lagrange equation yields

$$A f_i^* - B(R_i^*) = C\ddot{f}_i^* + \dot{C}\dot{f}_i^* - \dot{D} \tag{4}$$

where a dot denotes the derivative with respect to X_i. Much like in Section 5.1, solutions to Equation 4 have implicit dependence on R_i^*. furthermore, there are no general methods for solving second-order linear ODEs with variable coefficients, although it can be proved that a unique solution exists if we impose vanishing first derivatives at the boundaries. Numerical schemes for approximate solutions exist and one such example can be found in [2]. These schemes allows us to use the FCD algorithm where in Step 5 we apply the numerical solution to Equation 4. This guarantees an iterative reduction of the weighted H^1 error, and if instead of L_2 integrability we enforce H^1-integrability in each coordinate function, we can once again apply Theorem 1 to recover convergence.

Unfortunately, the surface fitting technique using the weighted L_2 or H^1 norm is scale-dependent. For example, if we were to rescale $X_1 \longrightarrow \ln X_1$ in the L_2 norm, the optimization becomes

$$\mathcal{L} = \min_{f_1,\ldots,f_N} \int_\Omega \big(f - \sum_{k=1}^N f_k\big)^2 w X_1^{-1}\, dX$$

In general, any reparameterization will introduce the determinant of the Jacobian into the integrand. One solution is to minimize the residuals under the L_∞ norm,

$$\mathcal{L} = \min_{f_1,\ldots,f_N} \sup_{X\in\Omega} \Big|\big[f(X) - \sum_{k=1}^N f_k(X_k)\big] w(X)\Big|$$

and both the fitted surface $\sum f_k^*$ and the residual $\mathcal{L}^*$ are scale-invariant.

5.4 Implementation

In practice, FCD is difficult to implement in its analytic form and we instead choose to discretize the problem. For each coordinate's domain Ω_i, pick a set of basis functions $\psi_1^i, ..., \psi_{m_i}^i$ associated with a finite mesh $x_1^i < x_2^i < ... < x_{n_i}^i$. We then construct f_i^* as a linear combination of these basis functions. This is especially useful in the L_2 norm, where the coefficients can be recovered as the solution to a system of linear equations.

For example, we might choose to construct each f_i^* as a cubic spline within the knots $x_1^i < x_2^i < ... < x_{n_i}^i$ and 0 outside the boundary. The free variables in our optimization are therefore $y_1^i, ..., y_{n_i}^i$ and $d_1^i, ..., d_{n_i}^i$, the values and derivatives of the spline at the knots. Between two consecutive nodes x_a, x_b, this corresponds to four basis functions.

$$\begin{aligned}
\psi_+^{y_a} &= 2\left(\frac{x - x_a}{x_b - x_a}\right)^3 - 3\left(\frac{x - x_a}{x_b - x_a}\right)^2 + 1 \\
\psi_+^{d_a} &= \left(\frac{x - x_a}{x_b - x_a}\right)^3 - 2\left(\frac{x - x_a}{x_b - x_a}\right)^2 + \left(\frac{x - x_a}{x_b - x_a}\right) \\
\psi_-^{y_b} &= -2\left(\frac{x - x_a}{x_b - x_a}\right)^3 + 3\left(\frac{x - x_a}{x_b - x_a}\right)^2 \\
\psi_-^{d_b} &= \left(\frac{x - x_a}{x_b - x_a}\right)^3 - \left(\frac{x - x_a}{x_b - x_a}\right)^2
\end{aligned}$$

These basis functions are 0 for $x \notin [x_a, x_b]$ and each function sets exactly one of the variables y_a, y_b, d_a, d_b to 1 and the others to 0. For general $\mathcal{L}$ we perform the optimization in Step 5 of the FCD algorithm over this set of bases. However, under the weighted L_2 norm, we can go a step further.

$$\begin{aligned}
&\min_{f_k} \int_\Omega \left(\sum_k f_k\right)^2 w - 2f\left(\sum_k f_k\right) w dX \\
&= \min_{y_a^k, d_a^k} \int_\Omega \left(\sum_{k=1}^N \sum_{a=1}^{n_k} \left[y_a^k \psi_\pm^{y_a} + d_a^k \psi_\pm^{d_a}\right]\right)^2 w - 2f\left(\sum_{k=1}^N \sum_{a=1}^{n_k} \left[y_a^k \psi_\pm^{y_a} + d_a^k \psi_\pm^{d_a}\right]\right) w dX \\
&= \min_{\alpha_a^k} \sum_{\alpha,\beta,k,l,a,b} \alpha_a^k \beta_b^l A_{\alpha\beta klab} - 2 \sum_{\alpha,k,a} \alpha_a^k F_{\alpha ka}
\end{aligned}$$

where

$$A_{\alpha\beta klab} = \sum_{p,q} \int_\Omega \psi_p^{\alpha_a^k} \psi_q^{\beta_b^l} w \, dX \qquad \text{and} \qquad F_{\alpha ka} = \sum_p \int_\Omega \psi_p^{\alpha_a^k} f w \, dX$$

and $\alpha, \beta \in \{y, d\}$, $k, l \in \{1, ..., N\}$, $a \in \{1, ..., i_k\}$, $b \in \{1, ..., i_l\}$. The signs $p, q \in \{+, -\}$ except at the boundary points, where $p \equiv +$ if $a = 1$ and $p \equiv -$ if $a = i_k$. The optimal coefficients α_a^k can then be recovered implicitly as the solution to the following linear system.

$$\sum_{\alpha,k,a} A_{\alpha\beta klab} \, \alpha_a^k = F_{\beta lb}$$

If each of the N dimensions has M nodes, we can reparameterize $F \to \mathbb{R}^{2MN}$ a single vector and $A \to \mathbb{R}^{2MN \times 2MN}$ a dense matrix with $2M(N-1)+6$ diagonal entries. Inverting this matrix is computationally difficult.

But if the weighting function in the L_2 norm is separable, then using a piecewise-constant basis in each dimension, we can recover explicit solutions as in the continuous case. Although this introduces discontinuities in the fitted surface, the piecewise-constant solution converges to the continuous version in the limit as the meshes become finer. For each dimension i and mesh $x_0^i < x_1^i < ... < x_{n_i}^i$, define the basis functions $\phi_s^i = \mathbb{1}\{X_i \in (x_{s-1}^i, x_s^i)\}$. As before, we seek solutions λ_i^k to the system of linear equations

$$\sum_{l,j} A_{klij}\, \lambda_j^l = F_{ki}$$

where

$$A_{klij} = \int_\Omega \phi_i^k \phi_j^l w \, dX \qquad \text{and} \qquad F_{ki} = \int_\Omega \phi_i^k f w \, dX$$

Applying separability of $w = w_1 w_2 ... w_N$, define

$$W = \int_\Omega w \, dX \qquad \text{and} \qquad \eta_i^k = \frac{\int_{\Omega_i} \phi_i^k w_i \, dX_i}{\int_{\Omega_i} w_i \, dX_i}$$

then

$$\begin{aligned} F_{ki} &= \sum_{\substack{l,j \\ l \neq k}} W \eta_i^k \eta_j^l \, \lambda_j^l + \sum_j \lambda_j^k \int_\Omega \phi_i^k \phi_j^k w \, dX \\ &= \sum_{\substack{l,j \\ l \neq k}} W \eta_i^k \eta_j^l \, \lambda_j^l + W \eta_i^k \lambda_i^k \\ &= W \left(\sum_{\substack{l,j \\ l \neq k}} \eta_j^l \, \lambda_j^l + \lambda_i^k \right) \eta_i^k \\ &= W \left(c^k + \lambda_i^k \right) \eta_i^k \end{aligned}$$

where in the second line we use orthogonality of ϕ_i^k, ϕ_j^k for $i \neq j$ and the fourth line we replace the sum with a constant dependent only on dimension. The explicit solution is therefore

$$\hat{f} = \sum_{k,i} \lambda_i^k \phi_i^k + \frac{1}{W} \left[\int_\Omega f w dX + \sum_{k,i} \lambda_i^k F_{ki} \right] \qquad \text{where} \qquad \lambda_i^k = \frac{F_{ki}}{W \eta_i^k}$$

by once again setting $c^k \equiv 0$ for all k and solving for the final shift.

If we want to implement the L_∞ norm minimization numerically, we need to compress the N-dimensional mesh $x_1^i, ..., x_{n_i}^i$ for $i = 1, ..., N$ into a single vector of length $M = \prod_{i=1}^N n_i$. The basis functions $\psi_s^i(\mathbf{x}) := \mathbb{1}_{\{\mathbf{x}_i = x_s^i\}}$ are also represented in this M-dimensional vector as basis elements $\beta_1, ..., \beta_K$ where $K = \sum_i n_i$. The linear programming problem can then be written as follows:

$$\min_A \max_{i \in \{1,..,M\}} \Big| f - A_1 \beta_1 - A_2 \beta_2 - ... - A_K \beta_K \Big|_i$$

or using the conventional notation,

$$\text{Minimize } \lambda \qquad \text{subject to} \qquad \begin{aligned} A^T\beta + \lambda e &\geq f \\ A^T\beta - \lambda e &\leq f \end{aligned}$$

where e denotes the vector of ones. This problem has a dual which can be written in the form

$$\text{Maximize } f^T(\pi' + \pi'') \qquad \text{subject to} \qquad \begin{aligned} &A(\pi' + \pi'') = 0 \\ &e^T(\pi' - \pi'') = 1 \\ &\pi' \geq 0,\ \pi'' \leq 0 \end{aligned}$$

which can be solved using the traditional simplex method. Further details on this technique can be found in [1].

6 Final Remarks

Functional attribution is an interesting sub-field of mathematics that combines various ideas in Group Theory, Linear Analysis and Numerical Analysis. Although we have laid the groundwork and discovered some preliminary results, there are still plenty of interesting extensions to investigate.

On the theoretical side, there are several open questions. If we relax the $f \in C^1$ condition to C^0, what additional constraints do we need so that we can still recover meaningful results? Can we relax some conditions in the FCD algorithm and still guarantee convergence? Are there other natural ways to define an attribution?

More practical examples include data-driven problems such as portfolio performance attribution in finance, where we lack explicit knowledge of the underlying function f. With only observations $(\mathbf{x}_1, f(\mathbf{x}_1)), ..., (\mathbf{x}_S, f(\mathbf{x}_S))$, we need to determine a suitable method to infer f from the data before we can apply functional attribution. A simple fitted model that allows us to do both simultaneously is

$$f(\mathbf{x}) = \sum_i f_i(\mathbf{x}_i) + \varepsilon$$

where the ε are i.i.d. errors. This motivates the development of a more rigorous statistical framework that allows for non-parametric estimates of f_i, heteroskedastic errors, asymptotic results for convergence, etc.

If there is some Markovian structure governing the evolution of $\mathbf{x}$, such as a diffusion process, then we can relax the path invariance requirement. The functional attribution can be computed by integrating along the path corresponding to the geodesic between two points, or as an average over all connecting paths weighted by their likelihoods. Under such a structure, transitivity is no longer as desirable. There is room to discover meaningful properties and useful results in this space.

References

[1] R. D. Armstrong and M. G. Sklar. A linear programming algorithm for curve fitting in the l_∞ norm. *Numerical Functional Analysis and Optimization*, 2(2-3):187–218, 1980.

[2] A. W. III and G. B. Costa. Solving second-order differential equations with variable coefficients. *International Journal of Mathematical Education in Science and Technology*, 39(2):238–243, 2008.