Function decomposition for nomography

Dylan Holmes

2021 June 20

Abstract

I prove some important results about decompositions of the form $f(x,y) = \sum_i u_i(x) \cdot v_i(y)$, which are important for nomography.

In particular, I provide a recipe for taking any decomposition and reducing it to as few terms as possible. I show that no matter what decomposition you start with, all fully reduced decompositions of $f(x,y)$ have the same number of terms in the sum. I show that in fact, each function has basically *only one* fully reduced decomposition—the functions in any fully-reduced decomposition can all be written as linear combinations of the functions in any other fully-reduced decomposition. Finally, I present Warmus's algorithm for getting an initial decomposition[1]. The algorithm converts any function $f(x,y)$ into the form $\sum_i u_i(x) \cdot v_i(y)$.

These results are important to nomographers because they eliminate guesswork: In preparation for making a nomogram of a function $F(x,y,z)$, you often need to decompose it into the form $\sum_i f_i(x) \cdot g_i(y) \cdot h_i(z)$, and that form must have at most 6 terms. This document teaches you how to produce an initial decomposition and then how to reduce that decomposition to as few terms as possible. It then provides an ironclad verdict: If the resulting decomposition is short enough, you're all set to make a nomogram. If not, you can be sure that it's impossible—no other approach, decomposition, mathematical insight, or algebraic trickery will help.

1 Function decomposition

A *decomposition* of a two-variable function $f(x,y)$ splits it into a sum of products of one-variable functions:

$$f(x,y) = \sum_{i=1}^{n} u_i(x) v_i(y).$$

The number of terms in the sum is called the *rank* of the decomposition.

[1] Actually, this part is forthcoming in a later draft

1.1 Commentary We're starting with the two-variable case for simplicity. Decompositions with *three* variables are useful in nomography where, in order to draw a function $F(u,v,w)$ as a nomogram, you need to decompose it into a sum of one-variable products $\sum_i f_i(u)g_i(v)h_i(w)$.

1.2 Reducible decompositions Sometimes you can simplify a decomposition $f(x,y) = \sum_{i=1}^n u_i(x)v_i(y)$, converting it into a new sum with fewer terms. This happens just when there is linear dependence: when the u_i are linearly dependent and/or the v_i are.

For example, if the u_i are linearly dependent, this means by definition that we can find coefficients $c_1,\ldots,c_n$, at least one of which is nonzero, such that $\sum_i c_i u_i(x) = 0(x)$. This implies that you can rewrite one of the $u(x)$ in terms of the others, eliminating it from the sum.

1.3 Recipe If the u_i are linearly dependent[2], you can solve for one of the u_k in terms of the others $u_1,\ldots,u_n$. When you substitute that expression for u_k into the decomposition formula $f(x,y) = \sum_i u_i(x)v_i(y)$, you'll eliminate u_k from the formula. Grouping all the u_i terms together, you'll be able to simplify, getting a decomposition with one fewer term.

As long as the decomposition has linear dependence, we can repeatedly shorten the sum using this procedure. Otherwise, if the $u_1,\ldots,u_n$ are linearly independent and the $v_1,\ldots,v_n$ are linearly independent, the decomposition has been reduced as much as possible. We call such a decomposition *fully reduced* or *irreducible*.

Technical details The description above gives the flavor of the recipe. This section shows exactly how the calculations work. The key takeaway is that the simplification process is entirely rote and automatic. Like similar processes, such as Gaussian elimination, a computer can perform them without guesswork or human input. Feel free to skip or skim this section if you'd prefer.

Suppose c_k is nonzero. Then we can choose to rewrite the expression $\sum_i c_i u_i(x) = 0$ to solve for u_k. We find $u_k(x) = \sum_{i\neq k}(-c_i/c_k)u_i(x)$. We can substitute this expression for u_k in the original decomposition, which will eliminate $u_k(x)$ from the decomposition entirely, leaving us with one fewer term:

[2] If your functions are differentiable—which they often are in practice—you can check for linear dependence by forming the $n\times n$ Wrońskian determinant. The determinant has $u_1,\ldots,u_n$ in the first row, and each subsequent row is the derivative of the one above it. *If* the determinant is identically zero, the functions are linearly dependent.

$$
\begin{aligned}
f(x,y) &= \sum_{i=1}^{n} u_i(x)v_i(y) \\
&= u_k(x)v_k(y) + \sum_{i\neq k} u_i(x)v_i(y) \\
&= \left[\sum_i (-c_i/c_k)u_i(x)\right] v_k(y) + \sum_{i\neq k} u_i(x)v_i(y) \\
&= \sum_{i\neq k} u_i(x)[v_i(y) - (c_i/c_k)v_k(y)] \\
&= \sum_{i\neq k} u_i(x)\widehat{v}_i(y), \qquad \widehat{v}_i(y) \equiv v_i(y) - (c_i/c_k)v_k(y)
\end{aligned}
$$

1.4 Theorem *All fully reduced decompositions have the same rank.*

1.5 Commentary This is important: no matter which decomposition you begin with, no matter which factors the decomposition includes or how you choose to simplify it, when it is fully reduced it will have the same, minimal rank as any other fully reduced decomposition.

This is important for nomography because it eliminates guesswork. Given that you can only form a nomogram of a function $F(x,y,z)$ if that function can be decomposed into a sum of six terms, you often wonder if you could just algebraically manipulate it into a simpler form. You have too many terms and wonder if there's some algebraic insight you're missing. Linear dependence provides an algorithm for simplifying the expression as much as possible. This theorem provides rock bottom: it says that once you've eliminated linear dependence, the expression is as simplified as possible. No backtracking, no algebraic insight, no other approach can simplify it further.

In fact, there's an even stronger sense in which all irreducible decompositions of a function are related—they're all interconvertible.

1.6 Theorem *Any two irreducible decompositions $\sum_i u_i(x)v_i(y) = \sum_i p_i(x)q_i(y)$ are related via a unique invertible linear transformation. Specifically, there exists an invertible $\mathbf{A}$ such that $p_i = \mathbf{A}u_i$ and $q_i = (\mathbf{A}^\top)^{-1}v_i$.*

1.7 Commentary This is such an astonishing, nice result. It says, in other words, that each function has essentially only one irreducible decomposition, made of n pairs of independent functions: once you have one irreducible decomposition $f = \sum_i u_i v_i$, all other irreducible decompositions can be written as linear combinations of the u_i multiplied by linear combinations of the v_i.

This, again, eliminates guesswork in the production of nomograms. It dispels the feeling that you might've missed some crucial way of rewriting f in a more convenient form.

The first dimension theorem actually follows from this one, but it's so practically useful I made sure to state it on its own.

Proofs

The main task in this section is to prove the two theorems. They're not hard proofs, but they rely on some matrix manipulations that can be hard to follow.

To streamline the presentation, let me present a lemma whose proof contains most of the hard work.

1.8 Lemma Suppose $f(x,y)$ can be decomposed two ways, with

$$f(x,y) = \sum_{i=1}^{M} u_i(x)v_i(y) = \sum_{j=1}^{N} p_j(x)q_j(y).$$

(These decompositions may have different rank—M and N, respectively—and are possibly unreduced.)

If $u_1,\ldots,u_M$ are linearly independent, then the $v_j(y)$ are a linear combination of the $q_i(y)$. In other words, there is an $M \times N$ matrix $\mathbf{A}$ such that

$$\mathbf{v}(y) = \mathbf{A} \cdot \mathbf{q}(y).$$

1.9 Commentary I'll postpone the proof for now. Using the result of this lemma, we can straightforwardly prove our two theorems.

1.10 Proof of Theorem 1.4 *All fully reduced decompositions have the same rank.*

Proof. Suppose $f(x,y)$ has two fully reduced decompositions $\sum_{i=1}^{M} u_i(x)v_i(y) = \sum_{j=1}^{N} p_j(x)q_j(y)$ of rank M and N, respectively. We want to show that $M = N$.

Because $\sum_i u_i v_i$ is fully reduced, all our functions are linearly independent: $u_1,\ldots,u_M,\ v_1,\ldots,v_M,\ p_1,\ldots,p_N,\ q_1,\ldots,q_N$.

Because $u_1,\ldots,u_m$ is linearly independent, we can apply the lemma to it. Accordingly, we find a function $\mathbf{A}$ that relates the v_i to the q_j:

$$\mathbf{v}(y) = \mathbf{A} \cdot \mathbf{q}(y).$$

Using the fact that $v_1,\ldots,v_M$ and $q_1,\ldots,q_N$ are linearly independent, we see that this function $\mathbf{A}$ maps N linearly independent vectors onto M linearly independent vectors. According to linear algebra, this is only possible if $N \geq M$.

Repeat this same argument, exchanging $u \leftrightarrow v$, $p \leftrightarrow q$, $M \leftrightarrow N$, and $x \leftrightarrow y$, to conclude that also $M \geq N$. Hence $M = N$.

□

1.11 Proof of Theorem 1.6 *Any two irreducible decompositions $\sum_i u_i(x)v_i(y) = \sum_i p_i(x)q_i(y)$ are related via an invertible linear transformation. Specifically, there exists an invertible $\mathbf{A}$ such that $p_i = \mathbf{A}u_i$ and $q_i = (\mathbf{A}^\top)^{-1}v_i$.*

Proof. Suppose there are two fully reduced decompositions $f(x,y) = \sum_i u_i(x) \cdot v_i(y) = \sum_i p_i(x) \cdot q_i(y)$. Note that they both have the same rank n, by the preceding theorem.

Because $\sum_i u_i(x)v_i(y)$ is fully reduced, both $u_1, \ldots, u_n$ and $v_1, \ldots, v_n$ are linearly independent. Apply the lemma to the $u_1, \ldots, u_n$ to find a matrix $\mathbf{A}$ such that

$$\mathbf{v}(y) = \mathbf{A} \cdot \mathbf{q}(y).$$

Apply the lemma to the $v_1, \ldots, v_n$ to find a matrix $\mathbf{B}$ such that

$$\mathbf{u}(x) = \mathbf{B} \cdot \mathbf{p}(x).$$

Incidentally, note that we can express the decomposition of $f(x, y)$ in vector notation as $f(x, y) = \mathbf{u}^\top \mathbf{v} = \mathbf{p}^\top \mathbf{q}$.

From the definition of $\mathbf{A}$ and $\mathbf{B}$, we also know that

$$\begin{aligned} f(x, y) &= \mathbf{u}^\top \mathbf{v} \\ &= (\mathbf{p}^\top \cdot \mathbf{B}^\top)(\mathbf{A} \cdot \mathbf{q}) \\ &= \mathbf{p}^\top \cdot (\mathbf{B}^\top \mathbf{A}) \cdot \mathbf{q}. \end{aligned}$$

Equating the two expressions for $f(x, y)$, we see that $\mathbf{B}^\top \mathbf{A}$ must be the identity matrix, and so the square matrices $\mathbf{B}^\top$ and $\mathbf{A}$ are inverses of one another. Rewriting $\mathbf{B} = (\mathbf{A}^\top)^{-1}$, we have our desired result:

$$\mathbf{u}(x) = (\mathbf{A}^\top)^{-1} \cdot \mathbf{p}(x)$$

$$\mathbf{v}(y) = \mathbf{A} \cdot \mathbf{q}(y)$$

□

1.12 Observation The converse of the theorem is also true: because invertible matrices send linearly independent sets to linearly independent sets, it follows that if $\mathbf{A}$ is invertible and $f(x, y) = \sum_i u_i v_i$ is irreducible, then we can obtain another irreducible decomposition by defining

$$[g_1(x), \ldots, g_n(x)] = \mathbf{A}^{-1}[u_1(x), \ldots, u_n(x)]$$

$$[h_1(y), \ldots, h_n(y)] = \mathbf{A}[v_1(y), \ldots, v_n(y)].$$

Indeed, then

$$\mathbf{g}^\top \mathbf{h} = \mathbf{u}^\top \mathbf{A}^{-1} \cdot \mathbf{A} \mathbf{v} = \mathbf{u}^\top \mathbf{v} = f(x, y),$$

as required.

1.13 Application Here's one sanity check: suppose you multiply each v_i by a certain amount $\alpha_i \neq 0$. To preserve the value of the function, you naturally want to divide each of the u_i by that same amount.

Note that this follows from the theorem: our scaling function amounts to a diagonal matrix $\mathbf{A} = \operatorname{diag}(\alpha_1, \ldots, \alpha_n)$ with α_i along the diagonal and zeroes elsewhere. Because it is a diagonal matrix, its inverse is $\mathbf{A}^{-1} = \operatorname{diag}(\alpha_1^{-1}, \ldots, \alpha_n^{-1})$ as expected.

1.14 Commentary Our next job is to prove Lemma 1.8, which we used in the above proofs. To prove it, let me introduce a valuable result about linearly-independent functions[3]. It'll help with the proof (and is just generally nice to know, as well):

1.15 Lemma *The functions $f_1(x), \ldots, f_n(x)$ are linearly independent if and only if there exist points $x_1, \ldots, x_n$ such that the matrix*

$$\begin{bmatrix} f_1(x_1) & f_2(x_1) & \cdots & f_n(x_1) \\ f_1(x_2) & f_2(x_2) & \cdots & f_n(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ f_1(x_n) & f_2(x_n) & \cdots & f_n(x_n) \end{bmatrix}$$

is invertible (i.e. such that the determinant is nonzero).

1.16 Proof of Lemma 1.8 *Suppose $f(x, y)$ can be decomposed two ways, with*

$$f(x,y) = \sum_{i=1}^{M} u_i(x) v_i(y) = \sum_{j=1}^{N} p_j(x) q_j(y).$$

(These decompositions may have different rank—M and N, respectively—and are possibly unreduced.) If $u_1, \ldots, u_M$ are linearly independent, then the $v_j(y)$ are a linear combination of the $q_i(y)$. In other words, there is an $M \times N$ matrix $\mathbf{A}$ such that

$$\mathbf{v}(y) = \mathbf{A} \cdot \mathbf{q}(y).$$

Proof. Suppose we have two decompositions, possibly unreduced and possibly of different ranks $M \neq N$:

$$f(x,y) = \sum_{i=1}^{M} u_i(x) v_i(y) = \sum_{j=1}^{N} p_j(x) q_j(y).$$

Assume $u_1, \ldots, u_M$ are linearly independent. Then, applying the previous Lemma 1.15 to $u_1, \ldots, u_M$, we find points $a_1, \ldots, a_M$ such that the following matrix is invertible:

$$\mathbf{U} = \begin{bmatrix} u_1(a_1) & u_2(a_1) & \ldots & u_M(a_1) \\ \vdots & \vdots & \ddots & \vdots \\ u_1(a_M) & u_2(a_M) & \ldots & u_M(a_M) \end{bmatrix}$$

Using those *same* points $a_1, \ldots, a_M$, define the matrix

$$\mathbf{P} = \begin{bmatrix} p_1(a_1) & p_2(a_1) & \ldots & p_N(a_1) \\ \vdots & \vdots & \ddots & \vdots \\ p_1(a_M) & p_2(a_M) & \ldots & p_N(a_M) \end{bmatrix}.$$

[3] I'll prove this lemma next.

Note that $\mathbf{P}$ is an $M \times N$ matrix, not necessarily square like $\mathbf{U}$. In particular, because we're using points $a_1, \ldots, a_M$ chosen for $u_1, \ldots, u_M$, Lemma 1.15 doesn't apply so $\mathbf{P}$ is not necessarily invertible.

Finally, let us adopt shorthand notation $\mathbf{v}(y)$ for the vector of $v_1(y) \ldots v_M(y)$ functions, $\mathbf{q}(y)$ for the vector of $q_1(y), \ldots, q_M(y)$ functions, and $\mathbf{f_a}(y)$ for the vector of $f(a_1, y), \ldots, f(a_M, y)$ functions.

We can write:

$$\mathbf{U} \cdot \mathbf{v}(y) = \mathbf{f_a}(y) = \mathbf{P} \cdot \mathbf{q}(y)$$

Because $\mathbf{U}$ is invertible, we can multiply on the left by $\mathbf{U}^{-1}$:

$$\mathbf{v}(y) = \mathbf{U}^{-1} \cdot \mathbf{P} \cdot \mathbf{q}(y)$$

This expression shows that each $v_i(y)$ is a linear combination of the $q_1(y), \ldots, q_N(y)$, which was to be shown.

$\square$

1.17 Proof of Lemma 1.15 *The functions $f_1(x), \ldots, f_n(x)$ are linearly independent if and only if there exist points $x_1, \ldots, x_n$ such that the matrix*

$$\begin{bmatrix} f_1(x_1) & f_2(x_1) & \cdots & f_n(x_1) \\ f_1(x_2) & f_2(x_2) & \cdots & f_n(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ f_1(x_n) & f_2(x_n) & \cdots & f_n(x_n) \end{bmatrix}$$

is invertible (i.e., such that the determinant is nonzero).

Proof. ($\Leftarrow$). If the functions $f_1, \ldots, f_n$ are linearly *dependent*, then there's a nonzero linear combination of them that sums to zero: $\sum_i c_i f_i(x) = 0(x)$. That linear combination sums to zero everywhere, so it sums to zero at every particular point $x = x_i$; hence no matter which points you choose, if you combine the rows $\begin{bmatrix} f_i(x_1) & f_i(x_2) & \ldots & f_i(x_n) \end{bmatrix}$ of the matrix using that linear combination, you'll get all zeroes. This shows that the rows are linearly dependent for any choice of $x_1, \ldots, x_n$, and so the matrix is not invertible either.

($\Rightarrow$). We proceed by induction. If we have only one linearly independent function f_1, it follows from the definition of linear independence that f_1 is not identically zero. Hence we can find some x_1 at which $f_1(x_1) \neq 0$. But then the 1×1 matrix $[f_1(x_1)]$ is invertible because its determinant is nonzero. Hence the statement is true for $n = 1$ linearly independent functions.

Next, let's proceed by induction. Suppose we have linearly independent functions $f_1, \ldots, f_n, g$ and that the theorem is true for $f_1, \ldots, f_n$ but fails when g is included. We'll find a contradiction.

If the statement is true for $f_1, \ldots, f_n$, we can find $x_1, \ldots, x_n$ to make the matrix $[f_i(x_j)]$ invertible, i.e. such that its determinant is nonzero.

Define the function

$$\mathbf{D}(x) \equiv \det\left(\begin{bmatrix} f_1(x_1) & f_2(x_1) & \cdots & f_n(x_1) & g(x_1) \\ f_1(x_2) & f_2(x_2) & \cdots & f_n(x_2) & g(x_2) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ f_1(x_n) & f_2(x_n) & \cdots & f_n(x_n) & g(x_n) \\ f_1(x) & f_2(x) & \cdots & f_n(x) & g(x) \end{bmatrix}\right).$$

If the statement fails when g is included, then this determinant $\mathbf{D}(x)$ is zero everywhere. There is no value of x which makes the matrix invertible.

We can rewrite $\mathbf{D}(x)$ by expanding the determinant along the bottom row[4]. When we do, we get a linear combination of the functions $f_1(x), \ldots, f_n(x), g(x)$. The coefficients on the functions are determinants of various submatrices. In particular, the coefficient on the g term is the determinant $\det([f_i(x_j)])$. It is nonzero because we picked the $x_1, \ldots, x_n$ so that it would be.

Hence we have two results: on the one hand, $\mathbf{D}(x)$ is zero everywhere. On the other hand, it is equal to a nontrivial linear combination of $f_1, \ldots, f_n, g$. This contradicts our assumption that $f_1, \ldots, f_n, g$ are linearly independent. It follows that $\mathbf{D}(x)$ can't be zero everwhere, and in particular there's some choice of $x = x_{n+1}$ which makes its matrix invertible. $\square$

2 Warmus decomposition algorithm

See Warmus's publication *Nomographic Functions* (1959) for more details.

2.1 Recipe You can convert a function $F(x,y)$ into a decomposition $\sum_i g_i(x) \cdot h_i(y)$ as follows.

In the base case, the function $F(x,y)$ is identically zero, and we're done—we don't need any terms in the sum.

Otherwise, we can find points (a,b) such that $F(a,b) \neq 0$. Accordingly, define $g_1(x) \equiv F(x,b)/F(a,b)$ and $h_1(y) \equiv F(a,y)$.

The first term in the decomposition is therefore $g_1(x) \cdot h_1(y) = \frac{F(x,b)}{F(a,b)} \cdot F(a,y)$. To get the remaining terms, recursively apply this algorithm to the function

$$G(x,y) \equiv F - g_1 h_1$$

2.2 Example Try this with the function $F(x,y) = \cos(x+y)$. Choose the values for a and b carefully so that the expressions for $g_i(x)$ and $h_i(y)$ simplify as much as possible. You'll find that the function decomposes nicely into exactly two terms—and a form that is perhaps familiar.

[4]For more details, see Laplace expansion.