Math 484: Nonlinear Programming[1] Mikhail Lavrov

Chapter 4, Lecture 3: Orthogonality

February 25, 2019 *University of Illinois at Urbana-Champaign*

1 Projections and orthogonal vectors

Last time, we showed that for any $m \times n$ matrix A with full column rank (that is, with linearly independent columns), the matrix

$$P = AA^\dagger = A(A^\mathsf{T}A)^{-1}A^\mathsf{T}$$

is a projection matrix that maps any $\mathbf{y} \in \mathbb{R}^m$ to the point $P\mathbf{y}$ which is the closest point to $\mathbf{y}$ in the subspace $V = \{\mathbf{Ax} : \mathbf{x} \in \mathbb{R}^n\}$.

As a special case, let A be an $m \times 1$ matrix—that is, a vector $\mathbf{a} \in \mathbb{R}^m$. (We assume $\mathbf{a} \neq \mathbf{0}$.) Then the subspace $V = \{t\mathbf{a} : t \in \mathbb{R}\}$ is just the line through the origin in the direction of $\mathbf{a}$, and the projection matrix is

$$P = \mathbf{a}(\mathbf{a}^\mathsf{T}\mathbf{a})^{-1}\mathbf{a}^\mathsf{T} = \frac{1}{\|\mathbf{a}\|^2}\mathbf{a}\mathbf{a}^\mathsf{T}.$$

The product $P\mathbf{y}$ can be also written as

$$\operatorname{proj}_{\mathbf{a}}(\mathbf{y}) = P\mathbf{y} = \left(\frac{1}{\|\mathbf{a}\|^2}\mathbf{a}\mathbf{a}^\mathsf{T}\right)\mathbf{y} = \frac{\mathbf{a}\cdot\mathbf{y}}{\|\mathbf{a}\|^2}\mathbf{a},$$

which makes it easy to see that the result is a multiple of $\mathbf{a}$. Here, $\operatorname{proj}_{\mathbf{a}}(\mathbf{y})$ is a common notation for projecting onto the line through $\mathbf{0}$ and $\mathbf{a}$. The ratio $\frac{\mathbf{a}\cdot\mathbf{y}}{\|\mathbf{a}\|^2}$ measures how much of $\mathbf{y}$ is "pointing in the same direction as" $\mathbf{a}$.

This formula becomes nicer if we replace $\mathbf{a}$ by the unit vector $\mathbf{u} = \frac{\mathbf{a}}{\|\mathbf{a}\|}$, which we might as well do because it doesn't change the line we're projecting onto. If we do, then $\frac{1}{\|\mathbf{a}\|^2}\mathbf{a}\mathbf{a}^\mathsf{T}$ simplifies to $\mathbf{u}\mathbf{u}^\mathsf{T}$, and

$$\operatorname{proj}_{\mathbf{a}}(\mathbf{y}) = \operatorname{proj}_{\mathbf{u}}(\mathbf{y}) = \mathbf{u}\mathbf{u}^\mathsf{T}\mathbf{y} = (\mathbf{u}\cdot\mathbf{y})\mathbf{u}.$$

So for a one-dimensional subspace, the projection matrix is best written in terms of a unit vector. What's the best way to expess projections onto higher-dimensional subspaces?

The answer is that we need an orthonormal basis of the subspace. We say that vectors $\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \dots, \mathbf{u}^{(n)}$ are *orthonormal* if

- $\|\mathbf{u}^{(1)}\| = \|\mathbf{u}^{(2)}\| = \cdots = \|\mathbf{u}^{(n)}\| = 1$ (they are unit vectors), and
- $\mathbf{u}^{(i)} \cdot \mathbf{u}^{(j)} = 0$ when $i \neq j$ (they are orthogonal).

[1] This document comes from the Math 484 course webpage: https://faculty.math.illinois.edu/~mlavrov/courses/484-spring-2019.html

Theorem 1.1. *Suppose that V is a subspace of $\mathbb{R}^m$ with an orthonormal basis $\{\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \dots, \mathbf{u}^{(n)}\}$. Then the projection matrix onto V is given by the formula*

$$\mathbf{u}^{(1)}(\mathbf{u}^{(1)})^\mathsf{T} + \mathbf{u}^{(2)}(\mathbf{u}^{(2)})^\mathsf{T} + \cdots + \mathbf{u}^{(n)}(\mathbf{u}^{(n)})^\mathsf{T}.$$

(A note on notation: for vectors $\mathbf{u}, \mathbf{v}$, the product $\mathbf{u}\mathbf{v}^\mathsf{T}$ is sometimes called the *outer product* of $\mathbf{u}$ and $\mathbf{v}$, by analogy with the inner product $\mathbf{u}^\mathsf{T}\mathbf{v}$, and your textbook also writes it as $\mathbf{u} \otimes \mathbf{v}$.)

Proof. For any vector $\mathbf{y}$, let $\mathbf{x} = P\mathbf{y}$ be the projection of $\mathbf{y}$ onto V, and let $\mathbf{z} = \mathbf{y} - \mathbf{x}$. As we showed yesterday, we have $\mathbf{z} \perp V$: that is, $\mathbf{z} \perp \mathbf{u}$ for all $\mathbf{u} \in V$.

Since $\mathbf{x} \in V$, we can write $\mathbf{x}$ in the orthonormal basis: $\mathbf{x} = x_1\mathbf{u}^{(1)} + x_2\mathbf{u}^{(2)} + \cdots + x_n\mathbf{u}^{(n)}$. So the original vector $\mathbf{y}$ can be written as

$$\mathbf{y} = x_1\mathbf{u}^{(1)} + x_2\mathbf{u}^{(2)} + \cdots + x_n\mathbf{u}^{(n)} + \mathbf{z}.$$

In this representation, we can compute $\mathbf{u}^{(i)} \cdot \mathbf{y}$ more easily:

$$\mathbf{u}^{(i)} \cdot \mathbf{y} = x_1(\mathbf{u}^{(i)} \cdot \mathbf{u}^{(1)}) + x_2(\mathbf{u}^{(i)} \cdot \mathbf{u}^{(2)}) + \cdots + x_n(\mathbf{u}^{(i)} \cdot \mathbf{u}^{(n)}) + \mathbf{u}^{(i)} \cdot \mathbf{z}$$

and a lot of cancellation happens: the term $x_i(\mathbf{u}^{(i)} \cdot \mathbf{u}^{(i)})$ simplifies to x_i, and every other term simplifies to 0. So $\mathbf{u}^{(i)} \cdot \mathbf{y} = x_i$.

This gives us a formula for x_i in terms of what we already know. So we can substitute that formula in for x_i in our expression for $\mathbf{x}$:

$$\begin{aligned}\mathbf{x} &= x_1\mathbf{u}^{(1)} + x_2\mathbf{u}^{(2)} + \cdots + x_n\mathbf{u}^{(n)} \\ &= (\mathbf{u}^{(1)} \cdot \mathbf{y})\mathbf{u}^{(1)} + (\mathbf{u}^{(2)} \cdot \mathbf{y})\mathbf{u}^{(2)} + \cdots + (\mathbf{u}^{(n)} \cdot \mathbf{y})\mathbf{u}^{(n)} \\ &= \mathbf{u}^{(1)}(\mathbf{u}^{(1)})^\mathsf{T}\mathbf{y} + \mathbf{u}^{(2)}(\mathbf{u}^{(2)})^\mathsf{T}\mathbf{y} + \cdots + \mathbf{u}^{(n)}(\mathbf{u}^{(n)})^\mathsf{T}\mathbf{y} \\ &= \left(\mathbf{u}^{(1)}(\mathbf{u}^{(1)})^\mathsf{T} + \mathbf{u}^{(2)}(\mathbf{u}^{(2)})^\mathsf{T} + \cdots + \mathbf{u}^{(n)}(\mathbf{u}^{(n)})^\mathsf{T}\right)\mathbf{y}.\end{aligned}$$

This proves the theorem: we multiply $\mathbf{y}$ by the expression in the statement of the theorem, and we get the projection $\mathbf{x}$. □

This translates into a second way to solve the least squares minimization problem from the previous lecture.

Corollary 1.1. *If we want to find the vector $\mathbf{x}^*$ that minimizes $\|Q\mathbf{x} - \mathbf{y}\|$, where the columns of Q are orthonormal, that vector is just $\mathbf{x}^* = Q^\mathsf{T}\mathbf{y}$.*

Proof. Let $\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \dots, \mathbf{u}^{(n)}$ be the columns of Q.

If $\mathbf{x}^* = Q^\mathsf{T}\mathbf{y}$, then its i^{th} component is $x_i^* = \mathbf{u}^{(i)} \cdot \mathbf{y}$. So

$$\begin{aligned}Q\mathbf{x}^* &= x_1^*\mathbf{u}^{(1)} + x_2^*\mathbf{u}^{(2)} + \cdots + x_n^*\mathbf{u}^{(n)} \\ &= (\mathbf{u}^{(1)} \cdot \mathbf{y})\mathbf{u}^{(1)} + (\mathbf{u}^{(2)} \cdot \mathbf{y})\mathbf{u}^{(2)} + \cdots + (\mathbf{u}^{(n)} \cdot \mathbf{y})\mathbf{u}^{(n)} \\ &= \left(\mathbf{u}^{(1)}(\mathbf{u}^{(1)})^\mathsf{T} + \mathbf{u}^{(2)}(\mathbf{u}^{(2)})^\mathsf{T} + \cdots + \mathbf{u}^{(n)}(\mathbf{u}^{(n)})^\mathsf{T}\right)\mathbf{y},\end{aligned}$$

which we know from the theorem is the closest point in $V = \{Q\mathbf{x} : \mathbf{x} \in \mathbb{R}^n\}$ to $\mathbf{y}$, confirming that $\mathbf{x}^*$ is the minimizer we want. □

From this corollary, we also see that the projection matrix $\mathbf{u}^{(1)}(\mathbf{u}^{(1)})^{\mathsf{T}}+\mathbf{u}^{(2)}(\mathbf{u}^{(2)})^{\mathsf{T}}+\dots+\mathbf{u}^{(n)}(\mathbf{u}^{(n)})^{\mathsf{T}}$ can be written more concisely as QQ^{T}.

The reverse product $Q^{\mathsf{T}}Q$ also has a relevant interpretation: we can check if $\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \dots, \mathbf{u}^{(n)}$ are orthonormal by checking if $Q^{\mathsf{T}}Q = I$.

2 The Gram–Schmidt process

Now we know that if Q has orthonormal columns, then we get a much nicer formula for the projection matrix and for the least-squares minimization problem. How do we make Q have orthonormal columns?

One method for doing this is the Gram–Schmidt process. (There are other, fancier methods, too.) This is an algorithm that:

- takes in as input some vectors $\mathbf{a}^{(1)}, \mathbf{a}^{(2)}, \dots, \mathbf{a}^{(n)}$,
- returns as output some orthonormal vectors $\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \dots, \mathbf{u}^{(k)}$ with

$$\operatorname{span}\{\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \dots, \mathbf{u}^{(k)}\} = \operatorname{span}\{\mathbf{a}^{(1)}, \mathbf{a}^{(2)}, \dots, \mathbf{a}^{(n)}\}.$$

(When we minimize $\|A\mathbf{x} - \mathbf{y}\|$, then the input vectors are the columns of A, and we make the output vectors be the columns of Q.)

The algorithm constructs $\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \dots, \mathbf{u}^{(k)}$ in order. For simplicity, let's first assume that the input vectors $\mathbf{a}^{(1)}, \mathbf{a}^{(2)}, \dots, \mathbf{a}^{(n)}$ are linearly independent; in this case, $n = k$ and the output vectors $\mathbf{u}^{(i)}$ is computed starting from $\mathbf{a}^{(i)}$.

Along the way, we'll also produce a list of vectors $\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \dots, \mathbf{v}^{(n)}$, which are orthogonal but not orthonormal: they're not unit vectors. We get the final output by normalizing them, setting $\mathbf{u}^{(i)} = \frac{\mathbf{v}^{(i)}}{\|\mathbf{v}^{(i)}\|}$, either at the end of the algorithm or as we go.

1. To begin the process, let $\mathbf{v}^{(1)} = \mathbf{a}^{(1)}$ and let $\mathbf{u}^{(1)} = \frac{\mathbf{v}^{(1)}}{\|\mathbf{v}^{(1)}\|}$.
2. When we've computed $\mathbf{v}^{(1)}, \dots, \mathbf{v}^{(j-1)}$ and possibly $\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(j-1)}$, we define $\mathbf{v}^{(j)}$ to be

$$\begin{aligned}\mathbf{v}^{(j)} &= \mathbf{a}^{(j)} - (\mathbf{u}^{(1)} \cdot \mathbf{a}^{(j)})\mathbf{u}^{(1)} - (\mathbf{u}^{(2)} \cdot \mathbf{a}^{(j)})\mathbf{u}^{(2)} - \dots - (\mathbf{u}^{(j-1)} \cdot \mathbf{a}^{(j)})\mathbf{u}^{(j-1)} \\ &= \mathbf{a}^{(j)} - \frac{\mathbf{v}^{(1)} \cdot \mathbf{a}^{(j)}}{\mathbf{v}^{(1)} \cdot \mathbf{v}^{(1)}}\mathbf{v}^{(1)} - \frac{\mathbf{v}^{(2)} \cdot \mathbf{a}^{(j)}}{\mathbf{v}^{(2)} \cdot \mathbf{v}^{(2)}}\mathbf{v}^{(2)} - \dots - \frac{\mathbf{v}^{(j-1)} \cdot \mathbf{a}^{(j)}}{\mathbf{v}^{(j-1)} \cdot \mathbf{v}^{(j-1)}}\mathbf{v}^{(j-1)}\end{aligned}$$

   and $\mathbf{u}^{(j)} = \frac{\mathbf{v}^{(j)}}{\|\mathbf{v}^{(j)}\|}$.
3. Repeat step 2 until we've found $\mathbf{v}^{(n)}$ and $\mathbf{u}^{(n)}$.

(The first formula for $\mathbf{v}^{(j)}$ is in terms of the $\mathbf{u}$ vectors, the second is in terms of the $\mathbf{v}$ vectors. In practice, the advantage of the first formula is that we don't need to remember the $\mathbf{v}$ vectors; the advantage of the second formula is that (for rational inputs) we can stick to using rational numbers and not have to deal with square roots.)

We see from step 2 of this process that there is some expression

$$\mathbf{a}^{(j)} = r_{1j}\mathbf{u}^{(1)} + r_{2j}\mathbf{u}^{(2)} + \cdots + r_{jj}\mathbf{u}^{(j)}$$

that gives us the original vectors in terms of the output vectors. This can be written as a matrix product:

$$A = \begin{bmatrix} | & & | \\ \mathbf{a}^{(1)} & \cdots & \mathbf{a}^{(n)} \\ | & & | \end{bmatrix} = \begin{bmatrix} | & & | \\ \mathbf{u}^{(1)} & \cdots & \mathbf{u}^{(n)} \\ | & & | \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ & r_{22} & \cdots & r_{2n} \\ & & \ddots & \vdots \\ & & & r_{nn} \end{bmatrix}$$

or $A = QR$, where Q is the matrix whose columns are the orthonormal vectors we've found, and R is an upper triangular matrix.

In the general case, where $\mathbf{a}^{(1)}, \mathbf{a}^{(2)}, \dots, \mathbf{a}^{(n)}$ are not linearly independent, step 2 will sometimes give us $\mathbf{v}^{(j)} = \mathbf{0}$. In that case, we omit the j^{th} vector: what this tells us is that $\mathbf{a}^{(j)}$ is not necessary to span the subspace. When we do this, we still get a factorization $A = QR$, but R is no longer square; it has extra columns for the vectors we omitted in the final output.

After doing the hard work of the Gram–Schmidt process and finding Q and R, solving the least-squares minimization problem is easy: $\|A\mathbf{x} - \mathbf{y}\| = \|QR\mathbf{x} - \mathbf{y}\|$ is minimized at the solution(s) to $R\mathbf{x} = Q^{\mathsf{T}}\mathbf{y}$. Since R is upper-triangular, this system of equations takes less work to solve.

Okay, so it's only less work because we did lots of work in advance. Mostly, people are interested in this method for numerical stability reasons; when working with orthonormal vectors, fewer numerical errors arise where we divide by a 0.0001 that should have been a 0.0002 and get an answer that's off by 10000.

2.1 An example of Gram–Schmidt

Let's find an orthonormal basis of the subspace spanned by $\mathbf{a}^{(1)} = (1, -1, 0)$, $\mathbf{a}^{(2)} = (0, 1, -1)$, and $\mathbf{a}^{(3)} = (1, 0, -1)$.

1. We set $\mathbf{v}^{(1)} = (1, -1, 0)$. Since $\mathbf{v}^{(1)} \cdot \mathbf{v}^{(1)} = 2$, we set $\mathbf{u}^{(1)} = \frac{1}{\sqrt{2}}\mathbf{v}^{(1)} = (\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}, 0)$.
2. We set
$$\mathbf{v}^{(2)} = \mathbf{a}^{(2)} - \frac{\mathbf{v}^{(1)} \cdot \mathbf{a}^{(2)}}{\mathbf{v}^{(1)} \cdot \mathbf{v}^{(1)}}\mathbf{v}^{(1)} = (0, 1, -1) - \frac{-1}{2}(1, -1, 0) = (\tfrac{1}{2}, \tfrac{1}{2}, -1).$$
Since $\mathbf{v}^{(2)} = \frac{3}{2}$, we set $\mathbf{u}^{(2)} = (\frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, -\sqrt{\frac{2}{3}})$.
3. We set
$$\begin{aligned} \mathbf{v}^{(3)} &= \mathbf{a}^{(3)} - \frac{\mathbf{v}^{(1)} \cdot \mathbf{a}^{(3)}}{\mathbf{v}^{(1)} \cdot \mathbf{v}^{(1)}}\mathbf{v}^{(1)} - \frac{\mathbf{v}^{(2)} \cdot \mathbf{a}^{(3)}}{\mathbf{v}^{(2)} \cdot \mathbf{v}^{(2)}}\mathbf{v}^{(2)} \\ &= (1, 0, -1) - \frac{1}{2}(1, -1, 0) - \frac{3/2}{3/2}(\tfrac{1}{2}, \tfrac{1}{2}, -1) = (0, 0, 0). \end{aligned}$$
This means we don't get an $\mathbf{u}^{(3)}$ vector.

Our final output is $\mathbf{u}^{(1)} = (\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}, 0)$ and $\mathbf{u}^{(2)} = (\frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, -\sqrt{\frac{2}{3}})$.