I. Least Squares Optimization

Regression and least squares

A fundamental problem in science and engineering that we have already encountered is to estimate a function given point samples (that are possibly corrupted by noise). Recall that in this setting we observe pairs of points (x_m, y_m) for $m = 1, \ldots, M$, and want to find a function $f(x)$ such that

$$f(x_m) \approx y_m, \quad m = 1, \ldots, M.$$

Of course, the problem is not well-posed yet, since without any constraints on f, there are any number of functions for which $f(x_m) = y_m$ exactly. Thus, we typically specify a class that f belongs to.

Last time we considered the case where f was restricted to be a linear function, i.e., $f(x) = \alpha x$ for some scalar α. Today we'll begin by just slightly generalizing this to the case where f can have a nonzero intercept, i.e., it is of the form $f(x) = \alpha_1 x + \alpha_2$. In this case f is technically an *affine* function, although this approach is – somewhat confusingly – most commonly referred to as **linear regression**.

Following the same procedure as last time, we will (for now) assume that we wish to find the affine fit that minimizes the sum of squared errors. In this case, we can write our optimization problem as

$$\underset{\alpha_1, \alpha_2 \in \mathbb{R}}{\text{minimize}} \ \sum_{m=1}^{M} (y_m - \alpha_1 x_m - \alpha_2)^2. \tag{1}$$

Since (1) is quadratic with respect to both α_1 and α_2, we can find the minimum in a similar fashion as before, but now we must take *partial derivatives* with respect to both variables, setting both of these equal to zero, and solving for the minimizing α_1 and α_2.

Towards this end, let

$$g(\alpha_1, \alpha_2) = \sum_{m=1}^{M} (y_m - \alpha_1 x_m - \alpha_2)^2$$

and note that

$$\begin{aligned} \frac{\partial}{\partial \alpha_1} g(\alpha_1, \alpha_2) &= -2 \sum_{m=1}^{M} x_m (y_m - \alpha_1 x_m - \alpha_2) \\ \frac{\partial}{\partial \alpha_2} g(\alpha_1, \alpha_2) &= -2 \sum_{m=1}^{M} (y_m - \alpha_1 x_m - \alpha_2). \end{aligned}$$

Setting these both equal to zero and rearranging yields

$$\begin{aligned} \sum_{m=1}^{M} x_m y_m &= \alpha_1 \sum_{m=1}^{M} x_m^2 + \alpha_2 \sum_{m=1}^{M} x_m \\ \sum_{m=1}^{M} y_m &= \alpha_1 \sum_{m=1}^{M} x_m + M \alpha_2. \end{aligned}$$

We can write this as a 2×2 system of equations in matrix form as

$$\begin{bmatrix} \sum_{m=1}^{M} x_m^2 & \sum_{m=1}^{M} x_m \\ \sum_{m=1}^{M} x_m & M \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} \sum_{m=1}^{M} x_m y_m \\ \sum_{m=1}^{M} y_m \end{bmatrix}.$$

Thus, we can obtain the solution to our optimization problem, which we will denote by $(\widehat{\alpha}_1, \widehat{\alpha}_2)$, by simply inverting this system, i.e., computing

$$\begin{bmatrix} \widehat{\alpha}_1 \\ \widehat{\alpha}_2 \end{bmatrix} = \begin{bmatrix} \sum_{m=1}^{M} x_m^2 & \sum_{m=1}^{M} x_m \\ \sum_{m=1}^{M} x_m & M \end{bmatrix}^{-1} \begin{bmatrix} \sum_{m=1}^{M} x_m y_m \\ \sum_{m=1}^{M} y_m \end{bmatrix}.$$

We can express the solution to this system in closed form by explicitly computing the inverse. Using the notation

$$\bar{x} = \frac{1}{M}\sum_{m=1}^{M} x_m \qquad \bar{y} = \frac{1}{M}\sum_{m=1}^{M} y_m,$$

the solution to this system reduces to

$$\begin{bmatrix} \widehat{\alpha}_1 \\ \widehat{\alpha}_2 \end{bmatrix} = \frac{1}{\sum_{m=1}^M x_m^2 - M\bar{x}^2} \begin{bmatrix} \sum_{m=1}^M x_m y_m - M\bar{x}\bar{y} \\ \bar{y}\sum_{m=1}^M x_m^2 - \bar{x}\sum_{m=1}^M x_m y_m \end{bmatrix}.$$

Of course, while an affine function is a natural model in many settings, we may wish to consider functions f with higher-order features (e.g., a quadratic) or using any number of other features. We can extend the framework above whenever our model for f is that it consists of a linear combination of some set of functions $\phi_n(\cdot)$:

$$f(x) = \sum_{n=1}^{N} \alpha_n \phi_n(x).$$

The functions ϕ_n could be polynomials, sinusoids, exponentials, or anything else that might be appropriate given the application. We now fit our function by solving for the "best" coefficients $\alpha_1, \ldots, \alpha_N$. There is a classical complexity versus robustness trade-off in choosing the number N of functions that we are going to use to fit the data – generally speaking, letting N be large gives us a richer class of functions with more expressive power, but leads to a harder estimation problem requiring more data if we want our estimate to be accurate.

As before, we want to select the coefficients so that $f(x_m) \approx y_m$ for all m. We can express this using vector notation by letting $\boldsymbol{r} \in \mathbb{R}^M$

denote the vector whose m^{th} entry is

$$r_m = y_m - \sum_{n=1}^{N} \alpha_n \phi_n(x_m).$$

We want $\boldsymbol{r}$ to be "small". The approach we have taken so far has been to measure the "size" of $\boldsymbol{r}$ using the sum of the squares of the elements in $\boldsymbol{r}$. However, it is important to note that there are many other possible choices. There is an entire family of functions, called **norms**, that provide a way to talk about the size of a vector. (See the technical details at the end of these notes for a quick overview.) The approach we have taken so far has been to minimize $\|\boldsymbol{r}\|_2$, or equivalently, $\|\boldsymbol{r}\|_2^2$, where $\|\cdot\|_2$ is the *Euclidean norm*:

$$\|\boldsymbol{r}\|_2 = \sqrt{\sum_{m=1}^{M} r_m^2}.$$

Next, consider the $M \times N$ matrix $\boldsymbol{A}$ and the $N \times 1$ vector $\boldsymbol{\alpha}$:

$$\boldsymbol{A} = \begin{bmatrix} \phi_1(x_1) & \phi_2(x_1) & \cdots & \phi_N(x_1) \\ \phi_1(x_2) & \phi_2(x_2) & \cdots & \phi_N(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(x_M) & \phi_2(x_M) & \cdots & \phi_N(x_M) \end{bmatrix} \qquad \boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{bmatrix}$$

$\boldsymbol{A}$ maps a set of coefficients $\boldsymbol{\alpha} \in \mathbb{R}^N$ to a set of M predictions for the vector of observations $\boldsymbol{y} \in \mathbb{R}^M$. Using this notation, we can write

$$\boldsymbol{r} = \boldsymbol{y} - \boldsymbol{A}\boldsymbol{\alpha}.$$

Using this notation, finding the $\boldsymbol{\alpha}$ that minimizes the squared error is now reduced to the standard least squares problem:

$$\underset{\boldsymbol{\alpha} \in \mathbb{R}^N}{\text{minimize}} \ \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\alpha}\|_2^2.$$

Next time, we will consider how to actually solve this problem.

Linear Algebra Review I: Vector spaces, norms, and inner products

Linear algebra has become as basic and as applicable as calculus, and fortunately it is easier.

– Gilbert Strang

Linear algebra is the branch of mathematics that deals with solving systems of equations, with matrices and vectors being the key objects of study. But what exactly is a vector? Two intuitive ways of thinking about a vector might come to mind. First, the kind of vector we encounter in solving a system of equations is simply a list of numbers. However, the other place you have likely encountered this idea is in Euclidean geometry or physics, where a vector typically refers to a (directed) line segment between two points. The fact that we can use the same word for both of these concepts is chiefly due to the revolutionary idea of René Descartes that we can describe geometry via their coordinates, i.e., a list of numbers (a vector).

Descartes initiated what might be called the "*algebraization*" of geometry: if we can describe geometry in terms of vectors, then we can reduce geometric problems to ones of algebra. Beginning in the 19th century, mathematics began trending in a different direction, leading to a "*geomertrization*" of algebra: extending the equivalence between geometry and vectors, we can apply geometric concepts to lists of numbers. It is now common to apply geometric notions such as length, distance, and angle from three-dimensional space to vectors that live in much higher-dimensional (even infinite-dimensional) spaces.

In order to do this, mathematicians needed to form a more abstract and precise definition of what we mean by a vector and the kind of sets of vectors where such geometric notions make sense. The basic building block here is the **vector space**. We will not worry too much about defining this in all of its abstract glory, but informally, a **vector space** $\mathcal{S}$ is a set of elements, called *vectors*, that has rules for adding vectors and multiplying them by scalars.[1]

These rules mostly just capture familiar properties that we would expect of addition (e.g., it is commutative and associative) and multiplication (e.g., distributive and associative). The most salient requirement is that the set $\mathcal{S}$ of vectors must be *closed* under vector addition and scalar multiplication, which simply means that adding two vectors (or multiplying a vector by a scalar) will produce another vector in $\mathcal{S}$. This requirement is called *linearity*, since it implies that we can take arbitrary linear combinations of vectors without producing nonsense, and as a result vector spaces are also often known as **linear vector spaces** or **linear spaces**.

The simplest example of a vector space, and the most important one for this course, is $\mathbb{R}^N$, i.e., the set of vectors consisting of lists of N real numbers, with the usual notions of vector (element-wise) addition and scalar multiplication (with real-valued scalars). To see the value of this more abstract definition, note that the following are also valid vector spaces:

- The set of infinite-length sequences.
- The set of polynomials of degree p.
- The set of continuous functions on the real line.
- The set of functions bandlimited to Ω.

[1]Most commonly, a scalar simply refers to a real or complex number.

An important detail to keep in mind is that not all sets of vectors actually qualify as a vector space – typically because the set fails to be *closed*. For example, the set of vectors in $\mathbb{R}^2$ that live within the unit circle is *not* a vector space. To see why not, think about whether all linear combinations of such vectors will also live within the unit circle.

While the definition of a vector space described above generalizes some of our intuition from Euclidean space, we gain much more by also defining a **norm** together with our vector space. A norm allows us to talk about the *length* of a vector or the *distance* between two vectors.[2]

Definition. A **norm** $\|\cdot\|$ on a vector space $\mathcal{S}$ is a function $\|\cdot\|$ that maps a vector in $\mathcal{S}$ to a real number with the following properties for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{S}$:

1. $\|\boldsymbol{x}\| \geq 0$, and $\|\boldsymbol{x}\| = 0 \Leftrightarrow \boldsymbol{x} = \mathbf{0}$.
2. $\|\boldsymbol{x} + \boldsymbol{y}\| \leq \|\boldsymbol{x}\| + \|\boldsymbol{y}\|$ (triangle inequality)
3. $\|a\boldsymbol{x}\| = |a| \cdot \|\boldsymbol{x}\|$ for any scalar a (homogeneity)

Other related definitions:

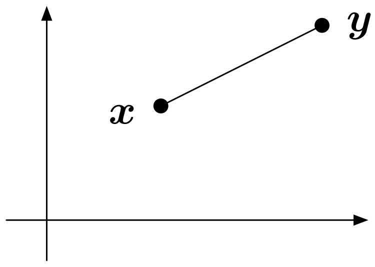
- The **length** of $\boldsymbol{x} \in \mathcal{S}$ is simply $\|\boldsymbol{x}\|$.
- The **distance** between $\boldsymbol{x}$ and $\boldsymbol{y}$ is $\|\boldsymbol{x} - \boldsymbol{y}\|$.
- A vector space in which we have defined a norm is called a **normed vector space**.

[2]A fancy mathematical way to say this is that a norm adds a layer of *topological structure* on top of the algebraic structure defining a vector space.

Examples:

1. $\mathcal{S} = \mathbb{R}^N$,
$$\|\boldsymbol{x}\|_2 = \left(\sum_{n=1}^{N} |x_n|^2 \right)^{1/2}$$
This is called the "ℓ_2 norm", or "standard Euclidean norm"

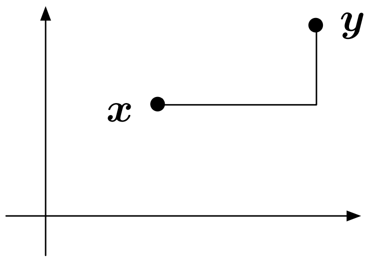
In $\mathbb{R}^2$:



$$\|\boldsymbol{x}-\boldsymbol{y}\|_2 = \sqrt{(x_1-y_1)^2 + (x_2-y_2)^2}$$

2. $\mathcal{S} = \mathbb{R}^N$
$$\|\boldsymbol{x}\|_1 = \sum_{n=1}^{N} |x_n|$$
This is the "ℓ_1 norm" or "taxicab norm" or "Manhattan norm"

In $\mathbb{R}^2$:



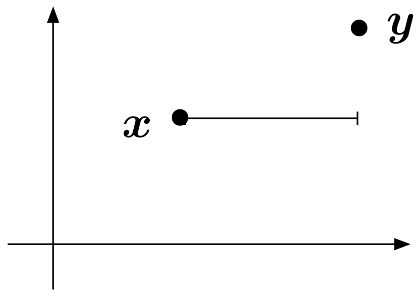
$$\|\boldsymbol{x}-\boldsymbol{y}\|_1 = |x_1-y_1| + |x_2-y_2|$$

3. $\mathcal{S} = \mathbb{R}^N$

$$\|\boldsymbol{x}\|_\infty = \max_{n=1,\dots,N} |x_n|$$

This is the "ℓ_∞ norm" or "Chebyshev norm"

In $\mathbb{R}^2$:



$$\|\boldsymbol{x}-\boldsymbol{y}\|_\infty = \max\left(|x_1 - y_1|,\ |x_2 - y_2|\right)$$

4. $\mathcal{S} = \mathbb{R}^N$

$$\|\boldsymbol{x}\|_p = \left(\sum_{n=1}^{N} |x_n|^p\right)^{1/p} \qquad \text{for some } 1 \le p < \infty$$

This is the "ℓ_p norm".

5. The same definitions extend to infinite sequences:
$\mathcal{S}$ = discrete-time signals $x[n]$ indexed by the integers $n \in \mathbb{Z}$

$$\|x[n]\|_p = \left(\sum_{n=-\infty}^{\infty} |x[n]|^p\right)^{1/p}$$

We also call this the "ℓ_p norm" – the fact that $x[n]$ is an infinite sequence is generally understood from the context.

6. The same idea can extend to continuous signals:
   $\mathcal{S}$ = continuous-time signals on the real line
$$\|x(t)\|_2 = \left(\int_{-\infty}^{\infty} |x(t)|^2 \ \mathrm{d}t \right)^{1/2}$$
   This is called the L_2 norm.[3] Similarly, we can define the more general L_p norms as
$$\|x(t)\|_p = \left(\int_{-\infty}^{\infty} |x(t)|^p \ \mathrm{d}t \right)^{1/p}.$$

A norm gives us a way to think about distances in a vector space. We can also talk about the *angle* between two vectors if we introduce the notion of an **inner product**.

Definition: An **inner product** on a real-valued vector space $\mathcal{S}$ is a function $\langle \cdot, \cdot \rangle$ that that maps a pair of vectors in $\mathcal{S}$ to a real number that satisfies the following properties for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{S}$:

1. $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \langle \boldsymbol{y}, \boldsymbol{x} \rangle$

2. For any $a, b \in \mathbb{R}$
$$\langle a\boldsymbol{x} + b\boldsymbol{y}, \boldsymbol{z} \rangle \ = \ a\langle \boldsymbol{x}, \boldsymbol{z} \rangle + b\langle \boldsymbol{y}, \boldsymbol{z} \rangle$$

3. $\langle \boldsymbol{x}, \boldsymbol{x} \rangle \geq 0 \quad \text{and} \quad \langle \boldsymbol{x}, \boldsymbol{x} \rangle = 0 \Leftrightarrow \boldsymbol{x} = \boldsymbol{0}$

Note that by "real-valued", we mean the scalar associated with scalar multiplication is a real number. You can easily extend these definitions to a complex-valued vector space, but since we will not need

[3]The L is for Lebesgue, the mathematician who formalized the modern theory of integration in the early 1900s.

this level of generality in this course we will stick with the simpler real-valued case.

Standard Examples:

- $\mathcal{S} = \mathbb{R}^N$,
$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle \;=\; \sum_{n=1}^{N} x_n y_n \;=\; \boldsymbol{y}^{\mathrm{T}} \boldsymbol{x}$$

- $\mathcal{S} =$ continuous-time signals on the real line
$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle \;=\; \int_{-\infty}^{\infty} x(t) y(t) \; \mathrm{d}t$$

Slightly less standard example:

- $\mathcal{S} = \mathbb{R}^{M \times N}$ (the set of $M \times N$ matrices with real entries)
$$\langle \boldsymbol{X}, \boldsymbol{Y} \rangle = \operatorname{trace}(\boldsymbol{Y}^{\mathrm{T}} \boldsymbol{X}) = \sum_{m=1}^{M} \sum_{n=1}^{N} X_{m,n} Y_{m,n}$$

  (Recall that trace($\boldsymbol{X}$) is the sum of the entries on the diagonal of $\boldsymbol{X}$.) This is called the *trace inner product* or *Frobenius inner product* or *Hilbert-Schmidt inner product.*

A vector space equipped with an inner product is called an **inner product space**. Inner products allow us to think about angles between vectors in arbitrary vector spaces – most importantly, we can generalize the notion of orthogonality from Euclidean space to an arbitrary inner product space: two vectors $\boldsymbol{x}, \boldsymbol{y}$ are **orthogonal** if $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = 0$.

Induced norms

A valid inner product induces a valid norm by

$$\|\boldsymbol{x}\| = \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}$$

(You can check that $\sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}$ indeed satisfies the properties required of a norm on your own as an exercise.)

It is not hard to see that in $\mathbb{R}^N$, the standard inner product induces the ℓ_2 norm, i.e., $\sqrt{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}} = \|\boldsymbol{x}\|_2^2$.

Properties of induced norms

In addition to the triangle inequality,

$$\|\boldsymbol{x} + \boldsymbol{y}\| \leq \|\boldsymbol{x}\| + \|\boldsymbol{y}\|,$$

which all norms must obey, induced norms obey some very handy inequalities. Below I give two of the most famous and useful ones. Note that these are not necessarily true for norms in general, only for norms induced by an inner product:

- **Pythagorean Theorem**

  $$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = 0 \;\Rightarrow\; \|\boldsymbol{x} + \boldsymbol{y}\|^2 = \|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2$$

  The left-hand side above also implies that $\|\boldsymbol{x} - \boldsymbol{y}\|^2 = \|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2$.

- **Cauchy-Schwarz Inequality**

  $$|\langle \boldsymbol{x}, \boldsymbol{y} \rangle| \;\leq\; \|\boldsymbol{x}\|\,\|\boldsymbol{y}\|$$

  Equality is achieved above when (and only when) $\boldsymbol{x}$ and $\boldsymbol{y}$ are **colinear**:

  $$\exists\, a \in \mathbb{R} \quad \text{such that} \quad \boldsymbol{y} = a\boldsymbol{x}.$$

A note on matrix multiplication

Although you have certainly encountered these concepts already, now is a good time to review some basic notions involving how we think about matrix multiplication. When dealing with simple matrix-vector products, these ideas are straightforward, but as we get to increasingly complex matrix factorizations, it can be harder to break down what is happening.

Matrix-vector multiplication

Let's begin by considering simple matrix-vector multiplication $\boldsymbol{Ax}$. There are really two ways to think about this. First, suppose that $\boldsymbol{A}$ is an $M \times N$ matrix. We can think of $\boldsymbol{A}$ as the concatenation of N columns, denoted by $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_N$:

$$\boldsymbol{A} = \begin{bmatrix} | & | & & | \\ \boldsymbol{a}_1 & \boldsymbol{a}_2 & \cdots & \boldsymbol{a}_N \\ | & | & & | \end{bmatrix}.$$

The first way of thinking of a matrix-vector multiplication is that $\boldsymbol{Ax}$ is a weighted combination of the columns of $\boldsymbol{A}$:

$$\begin{bmatrix} | & | & & | \\ \boldsymbol{a}_1 & \boldsymbol{a}_2 & \cdots & \boldsymbol{a}_N \\ | & | & & | \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = x_1\boldsymbol{a}_1 + x_2\boldsymbol{a}_2 + \cdots + x_N\boldsymbol{a}_N.$$

To describe the second way of thinking of a matrix-vector multiplication, we are going to slightly abuse our notation. Even though we just said $\boldsymbol{a}_n$ was a column of $\boldsymbol{A}$, which is what we will do for the rest of the course, note that we can also break up a matrix by along its rows, which we will (just for now) denote by $\boldsymbol{a}_1^{\mathrm{T}}, \boldsymbol{a}_2^{\mathrm{T}}, \ldots, \boldsymbol{a}_M^{\mathrm{T}}$:

$$\boldsymbol{A} = \begin{bmatrix} — & \boldsymbol{a}_1^{\mathrm{T}} & — \\ — & \boldsymbol{a}_2^{\mathrm{T}} & — \\ & \vdots & \\ — & \boldsymbol{a}_M^{\mathrm{T}} & — \end{bmatrix}.$$

We use a transpose when referring to the rows since normally $\boldsymbol{a}_m$ always refers to a column vector. This way of writing $\boldsymbol{A}$ suggests an alternative way to think about matrix-vector multiplication. That is, each entry of $\boldsymbol{Ax}$ is the inner product between the rows of $\boldsymbol{A}$ and the vector $\boldsymbol{x}$:

$$\begin{bmatrix} — & \boldsymbol{a}_1^{\mathrm{T}} & — \\ — & \boldsymbol{a}_2^{\mathrm{T}} & — \\ & \vdots & \\ — & \boldsymbol{v}_M^{\mathrm{T}} & — \end{bmatrix} \begin{bmatrix} | \\ \boldsymbol{x} \\ | \end{bmatrix} = \begin{bmatrix} \boldsymbol{a}_1^{\mathrm{T}}\boldsymbol{x} \\ \boldsymbol{a}_2^{\mathrm{T}}\boldsymbol{x} \\ \vdots \\ \boldsymbol{a}_N^{\mathrm{T}}\boldsymbol{x} \end{bmatrix}.$$

Matrix-matrix multiplication

Likewise, the product of an $M \times N$ matrix $\boldsymbol{A}$ and a $N \times P$ matrix $\boldsymbol{B}$ can be thought of in two different ways. The traditional way (how you would compute the elements of $\boldsymbol{AB}$ one entry at a time) is as a collection of the inner products between all of the rows of $\boldsymbol{A}$ and all of the columns of $\boldsymbol{B}$. If we continue to be a little loose with our notation, we can visualize this as:

$$\begin{bmatrix} — & \boldsymbol{a}_1^{\mathrm{T}} & — \\ — & \boldsymbol{a}_2^{\mathrm{T}} & — \\ & \vdots & \\ — & \boldsymbol{a}_M^{\mathrm{T}} & — \end{bmatrix} \begin{bmatrix} | & | & & | \\ \boldsymbol{b}_1 & \boldsymbol{b}_2 & \cdots & \boldsymbol{b}_P \\ | & | & & | \end{bmatrix} = \begin{bmatrix} \boldsymbol{a}_1^{\mathrm{T}}\boldsymbol{b}_1 & \boldsymbol{a}_1^{\mathrm{T}}\boldsymbol{b}_2 & \cdots & \boldsymbol{a}_1^{\mathrm{T}}\boldsymbol{b}_P \\ \boldsymbol{a}_2^{\mathrm{T}}\boldsymbol{b}_1 & \boldsymbol{a}_2^{\mathrm{T}}\boldsymbol{b}_2 & \cdots & \boldsymbol{a}_2^{\mathrm{T}}\boldsymbol{b}_P \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{a}_M^{\mathrm{T}}\boldsymbol{b}_1 & \boldsymbol{a}_M^{\mathrm{T}}\boldsymbol{b}_2 & \cdots & \boldsymbol{a}_M^{\mathrm{T}}\boldsymbol{b}_P \end{bmatrix}.$$

However, just as with matrix-vector multiplication, there are alternative perspectives. We think of each column of $\boldsymbol{AB}$ as being the result of a matrix-vector product that combines the columns of $\boldsymbol{A}$:

$$\boldsymbol{A}\begin{bmatrix} | & | & & | \\ \boldsymbol{b}_1 & \boldsymbol{b}_2 & \cdots & \boldsymbol{b}_P \\ | & | & & | \end{bmatrix} = \begin{bmatrix} | & | & & | \\ \boldsymbol{A}\boldsymbol{b}_1 & \boldsymbol{A}\boldsymbol{b}_2 & \cdots & \boldsymbol{A}\boldsymbol{b}_P \\ | & | & & | \end{bmatrix}.$$

Similarly, we can also think of each row of $\boldsymbol{AB}$ as a linear combination of the rows of $\boldsymbol{B}$:

$$\begin{bmatrix} - & \boldsymbol{a}_1^{\mathrm{T}} & - \\ - & \boldsymbol{a}_2^{\mathrm{T}} & - \\ & \vdots & \\ - & \boldsymbol{a}_M^{\mathrm{T}} & - \end{bmatrix}\boldsymbol{B} = \begin{bmatrix} - & \boldsymbol{a}_1^{\mathrm{T}}\boldsymbol{B} & - \\ - & \boldsymbol{a}_2^{\mathrm{T}}\boldsymbol{B} & - \\ & \vdots & \\ - & \boldsymbol{a}_M^{\mathrm{T}}\boldsymbol{B} & - \end{bmatrix}.$$

Finally, there is a fourth way to think of matrix-matrix multiplication, as a sum of the rank 1 matrices formed by taking the outer product of the columns of $\boldsymbol{A}$ with the rows of $\boldsymbol{B}$:

$$\begin{bmatrix} | & | & & | \\ \boldsymbol{a}_1 & \boldsymbol{a}_2 & \cdots & \boldsymbol{a}_N \\ | & | & & | \end{bmatrix}\begin{bmatrix} - & \boldsymbol{b}_1^{\mathrm{T}} & - \\ - & \boldsymbol{b}_2^{\mathrm{T}} & - \\ & \vdots & \\ - & \boldsymbol{b}_N^{\mathrm{T}} & - \end{bmatrix} = \sum_{n=1}^{N} \boldsymbol{a}_n\boldsymbol{b}_n^{\mathrm{T}}.$$

To see this, consider just $\boldsymbol{a}_1$ and what it contributes to $\boldsymbol{AB}$. Each column of $\boldsymbol{AB}$ will potentially have some contribution from $\boldsymbol{a}_1$. Just how much? Well, it's the first row of $\boldsymbol{B}$ that will determine this for each column. The same argument follows for each of the N columns of $\boldsymbol{A}$ and rows of $\boldsymbol{B}$.