Linear Algebra Motivated by Linear Regression

Authors: Patrick Chao, Ajay Raj, and Danielle Sugrue

March 11, 2019

Contents

1 Motivation

What does linear algebra have to do with data science?

Linear algebra is the study of vector spaces, and it encompasses linear equations and functions represented by vector spaces and matrices. Vectors and matrices are essential for storing data, which is why we often use Python packages such as Numpy and Pandas. A common problem is linear regression, which we will delve into solving using matrices and linear algebra.

2 Notation

Common notation:

- $\mathbf{A}$: Bold capital letters represent matrices
- $\mathbf{x}$: Bold lowercase letters represent vectors
- θ: Non-bold values represent scalars

3 Matrix Terminology

- **Identity** matrix: A square matrix with diagonal elements equal to 1 and all off diagonal elements equal to zero. A $n \times n$ identity matrix is often denoted as I or I_n.
- **Order or Size** of matrix: If a matrix has m rows and n columns, the order of the matrix is $m \times n$. We denote the set of (real-valued) matrices $\mathbb{M}_{m,n}$
- **Trace**: The sum of all the diagonal elements of a square matrix.
- **Transpose** of a matrix: The transpose of matrix $\mathbf{A}$ satisfies the condition $\mathbf{A_{j,i}} = \mathbf{A_{i,j}}^T$. That is, the first row of $\mathbf{A}$ is the first column of $\mathbf{A}^T$.
- **Square** matrix: A matrix with the same number of rows as columns. This matrix is in the shape of a square.
- **Diagonal** matrix: A matrix with all the non-diagonal elements equal to 0 is called a diagonal matrix.
- **Upper triangular** matrix: A square matrix with all the elements below diagonal equal to 0.
- **Lower triangular** matrix: A square matrix with all the elements above diagonal equal to 0.
- **Scalar** matrix: A square matrix with all the diagonal elements equal to a constant.

- **Identity** matrix: A square matrix with all the diagonal elements equal to 1 and all the non-diagonal elements equal to 0.
- **Column** matrix: A matrix which consists of exactly 1 column. If it has m rows, it can be treated as a $m \times 1$ vector.
- **Row** matrix: A matrix which consists of exactly 1 column. If it has m rows, it can be treated as a $m \times 1$ vector.

4 Linear Transformations and the Determinant

What does a matrix represent? A matrix with m rows and n columns describes a linear transformation from a vector space in $\mathbb{R}^n$ to a vector space in $\mathbb{R}^m$.

A **linear transformation** is a mapping that preserves addition and scalar multiplication between vector spaces. The **determinant** of a square matrix is the volume scaling factor of the linear transformation that the matrix represents.

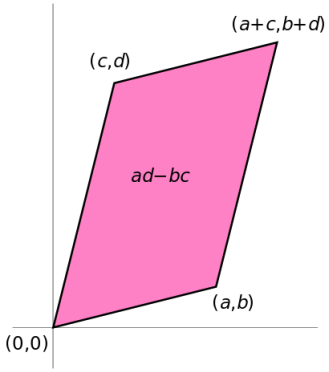

Figure 1: Determinant Scaling from `https://www.wikipedia.com/en/Determinant`

For a square matrix, the determinant is also known as the volume of the parallelepiped spanned by the columns of the matrix. The sign of the determinant is negative if the linear transformation reverses the orientation of a vector space.

For a 2×2 matrix, the determinant can be calculated in the following way:

$$\det \begin{bmatrix} a & b \\ c & d \end{bmatrix} = ad - bc$$

For a square matrix with more than 2 rows/columns, the determinant becomes more complicated to compute. Consider a 3×3 matrix:

$$\det \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} = aA_{11} - bA_{12} + cA_{13}$$

where A_{ij} = det(submatrix of A that has row i and columns j removed from A) which is called a **minor**. That is,

$$\det \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} = a \det \begin{bmatrix} e & f \\ h & i \end{bmatrix} - b \det \begin{bmatrix} d & f \\ g & i \end{bmatrix} + c \det \begin{bmatrix} d & e \\ g & h \end{bmatrix} = a(ei-hf) - b(di-gf) + c(dh-ge)$$

Luckily Python and R are useful for finding determinants and inverting matrices, so we don't have to worry too much about this for now.

5 Linear Independence and Span

What does it mean for a set of vectors to be **linearly independent**? It is easier to define what it means to be linearly dependent. A set of vectors

$$\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_n}$$

is linearly dependent if there exist scalars $\alpha_1, \alpha_2, ..., \alpha_n$, not all equal to 0 such that

$$\sum_{i=1}^{n} \alpha_i x_i = 0.$$

In words, this means that there exists at least one vector that can be written as a linear combination of the remaining vectors.
We say that a set S **spans** a vector space V is every vector v in V can be written as a linear combination of vectors in S. Thus, we define the **span** of vectors

$$v_1, v_2, ..., v_n$$

as the set of linear combinations

$$c_1 v_1 + c_2 v_2 + ... + c_n v_n.$$

For example, the set

$$(0, 1, 1), (1, 1, 0), (1, 0, 1)$$

spans $\mathbb{R}^3$. Try to convince yourself this is true by writing an arbitrary vector (a, b, c) as a linear combination of the vectors in the set.

6 Rank and Null Space

We define rank as the **dimension of the row space of A**, where the **row space** of an $m \times n$ matrix $\mathbf{A}$ is the subspace of $\mathbb{R}^n$ spanned by the rows of $\mathbf{A}$. In other words, the rank of a matrix is equal to the maximum number of linearly independent row vectors of a matrix. For example, let's check if the following matrix is fully ranked:

$$\begin{bmatrix} 1 & 3 & 1 \\ 5 & 9 & -6 \\ 7 & 3 & 0 \end{bmatrix}$$

This matrix is indeed fully ranked because there is no linear combination of the rows that equals the zero vector.
The **nullspace** of an $m \times n$ matrix $\mathbf{A}$ is the set of all n-dimensional vectors $\mathbf{v}$ such that

$$\mathbf{Av} = 0.$$

That is, the nullspace of a matrix $\mathbf{A}$ is the set of vectors which solve the system

$$\mathbf{Av} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

When referring to an arbitrary linear transformation L, we call the nullspace the **kernel** of L. When referring to matrices specifically, we generally use the term null space, but kernel works too.
The **nullity** of a matrix is the dimension of the nullspace of the matrix.

Let's find the nullity of the following matrix:

$$\mathbf{A} = \begin{bmatrix} -1 & 0 & -1 & 2 \\ 2 & 0 & 2 & 0 \\ 1 & 0 & 1 & -1 \end{bmatrix}$$

We can use elementary row operations (which do not change the null space of the matrix) to get $\mathbf{A}$ into row echelon form:

$$ref(\mathbf{A}) = \begin{bmatrix} -1 & 0 & 1 & -2 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

.
Then solve

$$rref(\mathbf{A})\mathbf{v} = \begin{bmatrix} 1 & 0 & 1 & -2 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

We get the following equations.

$$v_1 + v_3 - 2v_4 = 0, v_4 = 0.$$

We can combine the equations to see

$$v_1 + v_3 = 0$$

which is equivalent to

$$-v_1 = v_3.$$

Thus we know that our vector solutions must be of the form

$$\begin{bmatrix} v_1 \\ v_2 \\ -v_1 \\ 0 \end{bmatrix}.$$

The vectors $(1, 0, -1, 0)$ and $(0, 2, 0, 0)$ form a basis of our nullspace $\mathrm{N}(\mathbf{A})$, so the nullity of $\mathbf{A}$ is 2.

There is a very important linear algebra theorem called the **Rank - Nullity Theorem**. The theorem states that the rank of a matrix $\mathbf{A} \in \mathbb{R}^{m,n}$ plus the nullity of the matrix $\mathbf{A}$ equals the number of columns of $\mathbf{A}$.

$$\mathrm{Rank}(\mathbf{A}) + \mathrm{Nullity}(\mathbf{A}) = n.$$

So if we are given a matrix $\mathbf{A}$, we can find the null space of $\mathbf{A}$, which gives us the nullity, and then apply the theorem to find the rank of $\mathbf{A}$.

For the example matrix above with 4 columns and nullity 2, what is the rank of $\mathbf{A}$?

7 Invertibility

The inverse of an $n \times n$ matrix $\mathbf{A}$, denoted as $\mathbf{A}^{-1}$, satisfies the following properties:

$$\mathbf{A}\mathbf{A}^{-1} = I_{n\times n},\ \mathbf{A}^{-1}\mathbf{A} = I_{n\times n}.$$

We may consider a concrete example with a 2×2 matrix.

$$\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$\mathbf{A}^{-1} = (\det \mathbf{A})^{-1} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

For inverting matrices of higher dimensions, the calculation is much more difficult. We will often want to use a computer to compute these for us. But

why do we care about inverse matrices in the first place? Why do they come up in linear regression?

The answer to this will show up toward the end of this lecture when we consider the normal equations.

Remember for later: If a matrix $\mathbf{A}$ is invertible, so is its transpose, and the inverse of $\mathbf{A}^T$ is the transpose of the inverse of $\mathbf{A}$.

8 Eigenvalues and Eigenvectors

In linear algebra, an **eigenvector** of a matrix is defined as a non-zero vector that when multiplied by the matrix, changes only by a scalar factor. That is, for a (square) matrix $\mathbf{A}$ representing a linear transformation from a vector space, a nonzero vector $\mathbf{v}$ in V is an eigenvector of $\mathbf{A}$ if $\mathbf{Av}$ is a scalar multiple of $\mathbf{v}$. This can be written as:

$$\mathbf{Av} = \lambda\mathbf{v}$$

where λ is a scalar. λ is called an **eigenvalue** or characteristic value of $\mathbf{A}$ which is associated with the eigenvector $\mathbf{v}$. While eigenvalues exist where eigenvectors exist, we solve for eigenvalues before we find their associated eigenvectors.
Let's recall the equation

$$\mathbf{Av} = \lambda\mathbf{v}.$$

Suppose we have a matrix A and wish to solve for its eigenvalues. How might we approach this?

$$\mathbf{Av} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = \lambda \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = \lambda\mathbf{v}$$

How would we solve for $\mathbf{v}$? Since

$$\mathbf{Av} = \lambda\mathbf{v},$$

we can write

$$\mathbf{Av} - \lambda\mathbf{v} = 0$$

which is equivalent to

$$(\mathbf{A} - I\lambda)\mathbf{v} = 0.$$

It turns out that this has a nonzero solution for $\mathbf{v}$ if and only if the determinant of $\mathbf{A} - I\lambda$ has determinant 0. So we would take take the determinant, set it equal to 0 and solve for λ.

Example:

$$\mathbf{A} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}.$$

Then

$$\mathbf{A} - \lambda I = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 2-\lambda & 1 \\ 1 & 2-\lambda \end{bmatrix}$$

so $\det(\mathbf{A} - \lambda I) = (2-\lambda)^2 - 1 \cdot 1 = 3 - 4\lambda + \lambda^2$. Applying the quadratic formula or simply by factoring, we see that $\det(\mathbf{A} - \lambda I) = (\lambda - 1)(\lambda - 3)$ and conclude that our two eigenvalues are $\lambda_1 = 1$ and $\lambda_2 = 3$.

To solve for the associated eigenvectors, substitute the eigenvalues for the lambdas and solve for the vectors v_1 and v_2.

$$\begin{bmatrix} 2-\lambda_1 & 1 \\ 1 & 2-\lambda_1 \end{bmatrix} \begin{bmatrix} v_{11} \\ v_{12} \end{bmatrix} = \lambda_1 \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Which can be rewritten as:

$$\begin{bmatrix} 2-1 & 1 \\ 1 & 2-1 \end{bmatrix} \begin{bmatrix} v_{11} \\ v_{12} \end{bmatrix} = 1 \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Hence by multiplying:

$$\begin{bmatrix} v_{11} + v_{12} \\ v_{11} + v_{12} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Which gives us a system of equations:

$$v_{11} + v_{12} = 0$$

$$v_{11} + v_{12} = 0$$

Solving this returns $v_{11} = 1$ and $v_{22} = -1$, and any scalar multiples of this solution. Hence the eigenvector associated with $\lambda_1 = 1$ is given by

$$v_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

and all scalar multiples of this vector.

We can solve for the other eigenvector in the same manner. See if you can do this on your own.

Eigenvectors are especially useful in data science. For example, some of you may have heard of **Principal Component Analysis** (PCA), an algorithm used in machine learning. If you are given a dataset with lots of features, it is possible that there are redundant features in your data. For example, perhaps you are looking at a dataset of Berkeley residents and have a feature 'YearBorn' and another feature 'Age(Years).' In this case, you likely do not need both features because they are essentially informing you of the same

thing. PCA is used to cut out less important or redundant features, and it identifies these features by use of eigenvectors!

PCA works in the following way: Suppose you have an n-dimensional dataset and would like to remove $n-k$ dimensions so you end up with a k-dimensional dataset.

First, we would normalize our data and scale our features.

Removing features means that some information will be lost. PCA aims to minimize this loss, or maximize variance while still removing features. This is performed using eigenvectors:

We first create a covariance matrix for our dataset, and then we solve for the eigenvectors of the matrix. Because our dataset has n dimensions, our covariance matrix will be an $n \times n$ matrix and hence we will have n eigenvalues and we can solve for n eigenvectors which correspond to these eigenvalues.

Then we select k of these eigenvectors which are associated with the k largest eigenvalues of our matrix. Then we will make a $k \times n$ matrix of the k eigenvectors.

Then to find our reduced data points for our k features, we can transpose our matrix of eigenvectors and multiply it by a given data point vector. The output will be the data point vector in k dimensions.

9 The 1-Dimensional Case

The purpose of regression is to fit a mathematical model to a set of observed points. The purpose of designing such a model is to predict a dependent variable given one or more explanatory variables. The simplest case of linear regression is given points $(x_1, y_1), ..., (x_n, y_n)$, find a line of best fit through the origin to describe the points. We call this **simple linear regression**.

For a motivating example, say that x is the temperature on any given day and y is the total ice cream sales you made for that day. We are using regression because we believe that there exists a mathematical relationship that maps daily temperature to daily ice cream sales. Let's define some terms.

We are trying to find a model $f_\alpha(x)$ such that $f_\alpha(x)$ will predict, given the temperature on a day, the ice cream sales you will get. Specifically, we will utilize a linear model with one parameter:

$$f_\alpha(x) = \alpha_0 + \alpha_1 x_i.$$

Now, let's describe the loss of this model. Previously, we covered **RSS (residual sum of squares)** loss for this model. The **method of least squares** aims to minimize the square of the distance between our estimated model and the true values.

$$\sum_{i=1}^{n} (y_i - \alpha_0 - \alpha_1 x_i)^2. \tag{1}$$

Q: Why would we want to minimize the squared distance?
Why is this preferred over distance itself, i.e.

$$\sum_{i=1}^{n} (y_i - \alpha_0 - \alpha_1 x_i). \tag{2}$$

And why is this preferred over absolute value of distance, i.e.

$$\sum_{i=1}^{n} |y_i - \alpha_0 - \alpha_1 x_i|. \tag{3}$$

9.1 Vector Reformulation

Now, let's express this using linear algebra. Instead of expressing that we have a set of points, we have two **vectors**:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

Since α is a **scalar** (just a number, not a vector), we can still say that our model is $f_\alpha(x)$, but now it looks like this:

$$f_\alpha(x_1, x_2, ..., x_n) = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1,p+1} \\ x_{21} & x_{22} & \dots & x_{2,p+1} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{n,p+1} \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ ... \\ \alpha_p \end{bmatrix}$$

Now, you can see that each element in the vector is the **predicted** value of x_i.
In the case of simple linear regression, we have:

$$f_\alpha(x_1, x_2, ..., x_n) = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ \vdots & \vdots \\ x_{n1} & x_{n2} \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix}$$

9.2 Vector Norms

Now, we need to express the loss of this model. To do this, we need to look at two fundamental concepts of linear algebra. In linear algebra, there exists a concept of a **norm** of a vector, which expresses various notions of **magnitude**. There are two **norms** that we will be discussing today.

$$\ell_1(\mathbf{x}) = \|\mathbf{x}\|_1 = \sum_{i=1}^{n} |x_i|$$

$$\ell_2(\mathbf{x}) = \|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$$

The bars represent the norm, and the subscript represents the type of norm. The ℓ_2 norm is what you may be familiar with, it describes the length of the vector in the Euclidean sense. This is equivalent to the distance formula representation.

Can you think of an expression with the norm that expresses the loss of our model?

Next, let's discuss the notion of a **standard inner product**, which is a function that turns two vectors into a number. In general, an inner product can be thought as the similarity between two vectors. The definition of an inner product is

$$\langle \mathbf{u}, \mathbf{v} \rangle = \|\mathbf{u}\|_2 \|\mathbf{v}\|_2 \cos(\theta).$$

where θ is the angle between the vectors. A more concise method of writing

this with vectors in the Euclidean case is

$$\begin{bmatrix} u_1 & u_2 & \dots & u_n \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = \mathbf{u}^T\mathbf{v}.$$

You may know this as the **dot product**.

9.3 Transposes

The notation v^T represents the transpose of a vector or matrix. In general, this is defined as

$$\mathbf{A}^T_{ij} = \mathbf{A}_{ji}.$$

The transpose operation means the elements in the ith row and jth column become the elements in the jth row and ith column, or all elements are reflected over the main diagonal (top left to bottom right). In the case of a matrix $\mathbf{A}$ that is $n \times p$, n rows and p columns, $\mathbf{A}^T$ is of dimension $p \times n$. The transpose of a vector $\mathbf{x}$ of dimension $n \times 1$ becomes a row vector of dimension $1 \times n$. Since we may consider transposing matrices/vectors as reflecting them over the diagonal,

$$(\mathbf{A}^T)^T = \mathbf{A}.$$

A key identity of transposes is how they affect matrix-matrix products and matrix-vector products.

$$(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T. \tag{4}$$

Notice that $\mathbf{u}^T\mathbf{v} = \mathbf{v}^T\mathbf{u}$. A very common technique is rewriting norms using transposes.

$$\mathbf{x}^T\mathbf{x} = \|\mathbf{x}\|_2^2 \tag{5}$$

We say that a vector $\mathbf{x}$ has **unit norm** if $\|\mathbf{x}\|_2 = 1$. This may be interpreted as the length of the vector being 1. We can turn any vector $\mathbf{x}$ into one that is unit norm by dividing by its norm:

$$\mathbf{v} = \frac{\mathbf{x}}{\|\mathbf{x}\|_2}$$

$$\mathbf{v}^T\mathbf{v} = \left(\frac{\mathbf{x}}{\|\mathbf{x}\|_2}\right)^T \frac{\mathbf{x}}{\|\mathbf{x}\|_2} = \frac{\mathbf{x}^T\mathbf{x}}{\|\mathbf{x}\|_2^2} = 1$$

9.4 Vectorized Loss

We are now equipped to solve for θ. We would like to rephrase the RSS in equation 3. Our loss is:

$$\textbf{Loss}(\theta) = \sum_{i=1}^{n} (y_i - \theta x_i)^2.$$

Let's express the loss in terms of **inner products**.

$$\textbf{Loss}(\theta) = \sum_{i=1}^{n}(y_i - \theta x_i)^2 = \|\mathbf{y} - \theta\mathbf{x}\|_2^2.$$

Next, we may represent the ℓ_2 norm squared with equation 5.

$$\begin{aligned}\textbf{Loss}(\theta) &= (\mathbf{y} - \theta\mathbf{x})^T(\mathbf{y} - \theta\mathbf{x}) \\ &= \mathbf{y}^T\mathbf{y} - \theta\mathbf{y}^T\mathbf{x} - \theta\mathbf{x}^T\mathbf{y} + \theta^2\mathbf{x}^T\mathbf{x} \\ &= \mathbf{y}^T\mathbf{y} - 2\mathbf{y}^T\mathbf{x}\theta + \mathbf{x}^T\mathbf{x}\theta^2. \end{aligned} \tag{6}$$

There is an important step between the second and third line. Since the loss is a scalar, we may consider it as a 1×1 matrix. Thus the transpose of a 1×1 matrix is itself, or more concretely $\mathbf{A} = \mathbf{A}^T$. In the case of $\theta\mathbf{y}^T\mathbf{x}$, we know this is equivalent to $(\theta\mathbf{y}^T\mathbf{x})^T$. Now applying the rule of switching order with transposition in equation 4,

$$(\theta\mathbf{y}^T\mathbf{x})^T = \theta(\mathbf{y}^T\mathbf{x})^T = \theta(\mathbf{x}^T(\mathbf{y}^T)^T) = \theta\mathbf{x}^T\mathbf{y}.$$

From the first to the second equation, we can pull out θ since it is a scalar.

9.5 Minimizing Loss in the 1-Dimensional Case

It is our goal, now, to minimize that loss. You may have remembered from calculus that in order to minimize a function, you take the derivative and set it equal to zero. Let's take the derivative with respect to θ. Notice that this is the same as deriving an equation that looks like $a\theta^2 + b\theta + c$, since we may consider the vector terms as constants.

We will differentiate the expanded form of the loss in equation 6.

$$\frac{\partial}{\partial\theta}\left(\mathbf{y}^T\mathbf{y} - 2\mathbf{y}^T\mathbf{x}\theta + \mathbf{x}^T\mathbf{x}\theta^2\right) = -2\mathbf{y}^T\mathbf{x} + 2\mathbf{x}^T\mathbf{x}\theta$$

Now, we set it equal to zero to solve for the extrema.

$$\begin{aligned} 0 &= -2\mathbf{y}^T\mathbf{x} + 2\mathbf{x}^T\mathbf{x}\theta \\ 2\mathbf{y}^T\mathbf{x} &= 2\mathbf{x}^T\mathbf{x}\theta \\ \mathbf{y}^T\mathbf{x} &= \mathbf{x}^T\mathbf{x}\theta \\ \theta &= \frac{\mathbf{x}^T\mathbf{y}}{\mathbf{x}^T\mathbf{x}} \end{aligned}$$

$$\boxed{\theta = \frac{\mathbf{x}^T\mathbf{y}}{\mathbf{x}^T\mathbf{x}} = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^2}}$$

10 The d-Dimensional Case

Now say that instead of one feature, we have p features (note that this is usually the case—for example, ice cream sales don't just depend on temperature). We have a model $f_{\boldsymbol{\theta}}(x_1, x_2, ..., x_d)$. We would still like to model a single scalar output y_i. We continue our assumption that our model is linear, meaning we know that $f_{\boldsymbol{\theta}}$ looks like this:

$$f_{\boldsymbol{\theta}}(x_1, x_2, ..., x_d) = \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_d x_d$$

Because mathematicians are efficient (not lazy), we can write f_θ as an inner product, where θ is now a **vector** and x is now a d-dimensional vector:

$$f_{\boldsymbol{\theta}}(x) = \mathbf{x}^T \boldsymbol{\theta}$$

Notice that now we are attempting to calculate d values in θ instead of a single value. Now, let's express our loss. In general, our loss is

$$\mathbf{Loss}(\theta) = \sum_{i=1}^{n} (y_i - f_\theta(\mathbf{x}_i))^2.$$

Previously, we were considering a single scalar input x to our model (temperature for the day). However, now we now have a vector $\mathbf{x}$ corresponding to the set of d features. We can consider each observation $\mathbf{x}_i$ as a vector of length p. With our current model, it looks like this:

$$\mathbf{Loss}(\boldsymbol{\theta}) = \sum_{i=1}^{n} (y_i - \mathbf{x}^T \boldsymbol{\theta})^2$$

10.1 Matrix-ized Loss

Let's try to express this without a sum. Earlier, we were able to construct a vector such that each element is the individual loss for that specific data point. We were able to stack each individual temperature into a vector corresponding to all the temperatures. In this case, we may stack each row into a matrix.

Let $\mathbf{X}$ be a matrix where each row is a single data point (with p features). That is, $\mathbf{X}$ is an $n \times p$ matrix. The vector specified above is $\mathbf{y} - \mathbf{X}\boldsymbol{\theta}$. Notice that $\mathbf{y}$ is the same since the dimension of our output (ice cream sales) does not change. Our loss is now:

$$\mathbf{Loss}(\boldsymbol{\theta}) = \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_2^2 \tag{7}$$

As we did earlier, let's express this loss using inner products.

$$\begin{aligned}
\|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_2^2 &= (\mathbf{y} - \mathbf{X}\boldsymbol{\theta})^T (\mathbf{y} - \mathbf{X}\boldsymbol{\theta}) \\
&= \mathbf{y}^T\mathbf{y} - \mathbf{y}^T\mathbf{X}\boldsymbol{\theta} - (\mathbf{X}\boldsymbol{\theta})^T\mathbf{y} + (\mathbf{X}\boldsymbol{\theta})^T(\mathbf{X}\boldsymbol{\theta}) \\
&= \mathbf{y}^T\mathbf{y} - \mathbf{y}^T\mathbf{X}\boldsymbol{\theta} - \boldsymbol{\theta}^T\mathbf{X}^T\mathbf{y} + \boldsymbol{\theta}^T\mathbf{X}^T(\mathbf{X}\boldsymbol{\theta}) \\
&= \mathbf{y}^T\mathbf{y} - 2\mathbf{y}^T\mathbf{X}\boldsymbol{\theta} + \boldsymbol{\theta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\theta}.
\end{aligned}$$

We use the same trick of merging the middle terms. thus our final equation is

$$\mathbf{Loss}(\boldsymbol{\theta}) = \mathbf{y}^T\mathbf{y} - 2\mathbf{y}^T\mathbf{X}\boldsymbol{\theta} + \boldsymbol{\theta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\theta}. \tag{8}$$

10.2 Matrix Calculus Identities

How do we take the derivative of this? Since $\boldsymbol{\theta}$ is a vector now, it becomes a little more difficult. We won't go into why these properties are true today, but these tools help use derive this loss function. In vector calculus, we use $\nabla_{\mathbf{x}}$ instead of $\frac{d}{dx}$. The idea of a gradient is that you take the partial derivative with respect to each coordinate individually.

$$\nabla_{\mathbf{x}} = \begin{bmatrix} \frac{\partial}{\partial x_1} \\ \frac{\partial}{\partial x_2} \\ \vdots \\ \frac{\partial}{\partial x_n} \end{bmatrix}$$

There are some essential identities that you can perform with gradient operations.

$$\nabla_{\mathbf{x}}(\mathbf{a}^T\mathbf{x}) = \mathbf{a} \tag{9}$$

$$\nabla_{\mathbf{x}}(\mathbf{A}\mathbf{x}) = \mathbf{A}^T \tag{10}$$

$$\nabla_{\mathbf{x}}(\mathbf{x}^T\mathbf{A}\mathbf{x}) = (\mathbf{A} + \mathbf{A}^T)\mathbf{x} \tag{11}$$

Notice that if $\mathbf{A}$ is symmetric, meaning $\mathbf{A} = \mathbf{A}^T$, then equation 11 becomes $\nabla_{\mathbf{x}}\mathbf{x}^T\mathbf{A}\mathbf{x} = 2\mathbf{A}\mathbf{x}$.

We recommend that you go attempt to prove the first two identities yourself.

10.3 Minimizing Loss in the p-Dimensional Case

Now equipped with these identities, we may solve for the optimal value of $\boldsymbol{\theta}$.

$$\begin{aligned} \nabla_{\boldsymbol{\theta}}\left(\mathbf{y}^T\mathbf{y} - 2\mathbf{y}^T\mathbf{X}\boldsymbol{\theta} + \boldsymbol{\theta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\theta}\right) &= \nabla_{\boldsymbol{\theta}}\left(\mathbf{y}^T\mathbf{y}\right) - \nabla_{\boldsymbol{\theta}}\left(2\mathbf{y}^T\mathbf{X}\boldsymbol{\theta}\right) + \nabla_{\boldsymbol{\theta}}\left(\boldsymbol{\theta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\theta}\right) \\ &= 0 - 2\mathbf{X}^T\mathbf{y} + 2\mathbf{X}^T\mathbf{X}\boldsymbol{\theta} \end{aligned}$$

We may now set this equal to 0 and solve for $\boldsymbol{\theta}$.

$$\begin{aligned} 0 &= -2\mathbf{X}^T\mathbf{y} + 2\mathbf{X}^T\mathbf{X}\boldsymbol{\theta} \\ 2\mathbf{X}^T\mathbf{y} &= 2\mathbf{X}^T\mathbf{X}\boldsymbol{\theta} \\ \mathbf{X}^T\mathbf{y} &= \mathbf{X}^T\mathbf{X}\boldsymbol{\theta} \\ (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{X}\boldsymbol{\theta} \end{aligned}$$

$$\boxed{\boldsymbol{\theta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}}$$

The final equation is incredibly important, it is known as the **normal equation**. This boils down the entire linear regression problem into a simple and one line equation.

11 Closing Remarks

We have covered the 1-dimensional case intuition and built up to the general p-dimensional case. There are a few important key points to consider.

- Assuming the data has mean 0, what statistical interpretation do you have for the formula for the 1-dimensional case?
- Is $\mathbf{X}^T\mathbf{X}$ always invertible?
- Does this construction of linear regression contain constant terms? For example, in the 1-dimensional case, do we predict models of the form $\theta x + b$ where b is a constant? If not, how can we adjust our construction? Can this be expanded to the p-dimensional case?
- How does this change if we adjust our loss function to also include an ℓ_2 penalty to the magnitude of $\boldsymbol{\theta}$? Consider the following situation for $\lambda \geq 0$.

$$\mathbf{Loss}(\boldsymbol{\theta}) = \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_2^2 + \lambda\|\boldsymbol{\theta}\|_2^2.$$