# Recitation 4
# Linearity: The Only Thing Better Than Convexity

10-315: Introduction to Machine Learning

Fall 2020

## 1 Linear Algebra

**Definition 1.1.** A *matrix* is an object $M \in \mathbb{R}^{m \times n}$ representing a linear transformation from $\mathbb{R}^n \to \mathbb{R}^m$.

Recall a function is linear if

**Definition 1.2.** For vector spaces V,W, $f : V \to W$ is *linear* if $f(ax + by) = af(x) + bf(y)$ for all $a, b \in \mathbb{R}$ and $x, y \in V$.

Recall vector spaces are collections of vectors closed under addition and real multiplication. Every vector space of finite dimension $n$ admits a *basis* of size n ie. a collection of vectors $v_1, ..., v_n$ s.t. every other vector can be written as their linear combination.

The application of a matrix $A \in \mathbb{R}^{m \times n}$ to a vector in $v \in \mathbb{R}^n$ is the matrix vector multiplication $Av$. Two ways to think about this:

1) The ith component of $Av$ is given by the dot product of v with the ith row

2) $Av$ is given by a linear combination of the columns of A

**A Note on Type Checking:**

What dimension vector does $A \in \mathbb{R}^{m \times n}$ take in?

$v \in \mathbb{R}^n$

How about $ABC$ where $A \in \mathbb{R}^{m \times n}, B \in \mathbb{R}^{n \times k}, C \in \mathbb{R}^{k \times l}$. What will the output dimension be?

$v \in \mathbb{R}^l$ and $ABCv \in \mathbb{R}^m$

Simplest and quickest way to think about this(in my opinion) is to keep in mind is the adjacent dimensions of multiplied object must match:

If $A \in \mathbb{R}^{m \times n}, v \in \mathbb{R}^n$

$$A \times v \implies (m \times n) \times (n \times 1) \implies m \times 1$$

## 1.1 Normed Vector Spaces

**Definition 1.3.** A norm $|\cdot| : V \to \mathbb{R}$ on a vector space is defined as any function satisfying

i) $|v| \geq 0$ and $|v| = 0 \iff v = 0$ (Positivity)

ii) $|av| = |a||v|$ for $a \in \mathbb{R}$ (Homogeneity)

iii) $|v + w| \leq |v| + |w|$ (triangle inequality)

If we have a norm on V we say we have a normed vector space. These measure the "size" or "magnitude" of a vector

**Ex:** The $l^p$ norms

For $p > 1$ we define the $l^p$ norm of $v \in \mathbb{R}^n$ to be

$$|v|_p = (\sum_i |v_i|^p)^{1/p}$$

1. Convince yourself this is a norm.

   Exercise

2. Does $0 < p < 1$ define a norm?

   Exercise

3. Compute for $v = \langle -1, 2 \rangle$ the $l^1, l^2, l^3, l^4$ norms. Do you notice anything?

$$|v|_1 = 3$$
$$|v|_2 = \sqrt{5}$$
$$|v|_3 = 7^{1/3}$$
$$|v|_4 = 17^{1/4}$$

   The norms are increasing.

Often we work with the Euclidean norm $|\cdot|_2$ ie. the $l^2$ norm

**Definition 1.4.** The *inner product* on a vector space can be thought of as a generalization of the dot product $\cdot$ between two vectors:

$$x \cdot y = \sum_i x_i y_i$$

Generally we'll just be working with the dot product

**Fact:** Every inner product space admits a norm.

**Definition 1.5.** We say two vectors $v, w \in V$ are *orthogonal* if $v \cdot w = 0$

Geometrically this means they are "skew" or project onto one another as 0.

### 1.2 Matrix Operations

**Definition 1.6.** The *transpose* of a matrix $A$, written $A^T$, is defined coordinate wise as $A^T_{ij} = A_{ji}$, ie. indices are flipped. So $A \in \mathbb{R}^{m\times n}$ has $A^T \in \mathbb{R}^{n\times m}$.

**Definition 1.7.** The *inverse* of $A \in \mathbb{R}^{m\times n}$ satisfies $A^{-1}A = I$ where is the identity matrix.

### 1.3 Special Types of Matrices

**Definition 1.8.** *Diagonal* matrix is such that all the off diagonal entries are zero.

**Definition 1.9.** *Symmetric* matrix $A$ is s.t. $A = A^T$

**Definition 1.10.** *Orthogonal* matrix A is s.t. $A^{-1} = A^T$

Equivalently the set of orthogonal matrices are exactly those whose columns are unit norm, linearly independent, and pairwise orthogonal(orthonormal).

### 1.4 Good Matrix Vector Identities to Know

We have

$$(AB)^T = B^TA^T$$

$$A = (A^T)^T$$

$$x \cdot y = y \cdot x$$

$$x \cdot Ay = x^TAy = y^TA^Tx = y \cdot A^Tx$$

$$(A^{-1})^{-1} = A$$

$$(A^{-1})^T = (A^T)^{-1}$$

$$(A+B)^T = A^T + B^T$$

## 2 Vector and Matrix Derivatives

In the following discussion I will differentiate matrix quantities with respect to the elements of the referenced matrices. Although no new concept is required to carry out such operations, the element-by-element calculations involve cumbersome manipulations and, thus, it is useful and often much more efficient to derive these results and have them readily available(particularly come neural networks).

**Definition 2.1.** Vector Derivative Let $y = f(x)$ where $f : \mathbb{R}^n \to \mathbb{R}^m$. Then define

$$\frac{\partial y}{\partial x} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \cdots \frac{\partial y_1}{\partial x_n} & \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \frac{\partial y_m}{\partial x_2} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix}$$

This is the jacobian.

Note this definition of the jacobian is the transpose of how I defined it in the first recitation. For a function $f : \mathbb{R}^n \to \mathbb{R}$ this is the same thing as the gradient $\nabla$.

**Theorem 1.** Let $y = Ax$ where $A \in \mathbb{R}^{m \times n}$ and $x \in \mathbb{R}^n$. Then

$$\frac{\partial y}{\partial x} = A(x) = A$$

where A does not depend on x.

*Proof.* 1. Prove this

□

## 2.1 More Good Matrix Vector Identities to Know

$$\frac{\partial}{\partial x} w^T x = w$$

$$\frac{\partial}{\partial x} y^T A x = y^T A$$

$$\frac{\partial}{\partial x} x^T A x = x^T (A + A^T) \text{ and if A symmetric } 2x^T A$$

$$\frac{\partial}{\partial z} y^T(z) x(z) = x^T \frac{\partial y}{\partial z} + y^T \frac{\partial x}{\partial z}$$

$$\frac{\partial}{\partial z} y^T(z) A x(z) = x^T A^T \frac{\partial y}{\partial z} + y^T A \frac{\partial x}{\partial z}$$

For proofs/written explanations look up Matrix Calculus and some other stuff by Randal Barnes or The Matrix Cookbook. Many more good identities(involving more operators like trace and determinant) can be found there.

**A Note on Dimensions of Derivatives:**

Given $\frac{\partial y}{\partial x}$ where $y : \mathbb{R}^n \to \mathbb{R}^m$, we know $\frac{\partial y}{\partial x}$ will have $mn$ components and be $m \times n$

### 2.2 Chain Rule: The Reason Neural Networks Are a Thing

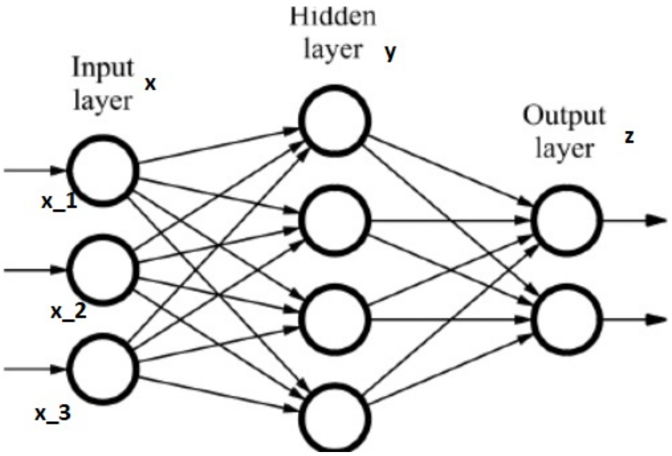
In the single variable setting chain rule tells us

$$(f(g(x)))' = f'(g(x))g'(x)$$

or in leibniz form

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial g}\frac{\partial g}{\partial x}$$

massively more helpful for taking complex chain rules. Consider:



1. Compute $\frac{\partial z_1}{\partial x}$

   We can write this as a function $z_1(y(x))$. Then $\frac{\partial z_1}{\partial x} = \frac{\partial z_1}{\partial y}^T \frac{\partial y}{\partial x}$

   Alternatively can compute component wise:

   $\frac{\partial z_1}{\partial x_i} = \frac{\partial z_1}{\partial y}^T \frac{\partial y}{\partial x_i} = \sum_j \frac{\partial z_1}{\partial y_j}\frac{\partial y_j}{\partial x_i}$

Moral: Make sure things type check!

## 3 Discriminant Analysis

Linear discriminant analysis vs. Logistic Regression:
- Statistically similar formulation
- Uses least squares estimation
- Assumes classes are characterized by normal densities (strong assumption)
- More sensitive to outliers
- LDA assumes homoskedasticity (very strong assumption)

When assumptions are met, LDA usually outperforms LR, but assumptions are very strict, often making it impractical. Most instances of LDA outperforming LR are in asymptotic cases where improvement is negligible.

# 4 Linear Regression (Matrix Form)

So far, we have not used any notions, or notation, that goes beyond basic algebra and calculus (and probability). This has forced us to do a fair amount of book-keeping, as it were by hand. This is just about tolerable for the simple linear model, with one predictor variable. It will get intolerable if we have multiple predictor variables. Fortunately, a little application of linear algebra will let us abstract away from a lot of the book-keeping details, and make multiple linear regression hardly more complicated than the simple version[1].

These notes will not remind you of how matrix algebra works. However, they will review some results about *calculus* with matrices, and about expectations and variances with vectors and matrices.

Throughout, bold-faced letters will denote matrices, as **a** as opposed to a scalar $a$.

# 1 Least Squares in Matrix Form

Our data consists of $n$ paired observations of the predictor variable $X$ and the response variable $Y$, i.e., $(x_1, y_1), \ldots (x_n, y_n)$. We wish to fit the model

$$Y = \beta_0 + \beta_1 X + \epsilon \tag{1}$$

where $\mathbb{E}\left[\epsilon | X = x\right] = 0$, $\mathrm{Var}\left[\epsilon | X = x\right] = \sigma^2$, and $\epsilon$ is uncorrelated across measurements[2].

## 1.1 The Basic Matrices

Group all of the observations of the response into a single column ($n \times 1$) matrix $\mathbf{y}$,

$$\mathbf{y} = \left[\begin{array}{c} y_1 \\ y_2 \\ \vdots \\ y_n \end{array}\right] \tag{2}$$

Similarly, we group both the coefficients into a single vector (i.e., a $2 \times 1$ matrix)

$$\beta = \left[\begin{array}{c} \beta_0 \\ \beta_1 \end{array}\right] \tag{3}$$

We'd also like to group the observations of the predictor variable together, but we need something which looks a little unusual at first:

$$\mathbf{x} = \left[\begin{array}{cc} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{array}\right] \tag{4}$$

[1] Historically, linear models with multiple predictors evolved before the use of matrix algebra for regression. You may imagine the resulting drudgery.

[2] When I need to also assume that $\epsilon$ is Gaussian, and strengthen "uncorrelated" to "independent", I'll say so.

This is an $n \times 2$ matrix, where the first column is always 1, and the second column contains the actual observations of $X$. We have this apparently redundant first column because of what it does for us when we multiply $\mathbf{x}$ by $\beta$:

$$\mathbf{x}\beta = \left[ \begin{array}{c} \beta_0 + \beta_1 x_1 \\ \beta_0 + \beta_1 x_2 \\ \vdots \\ \beta_0 + \beta_1 x_n \end{array} \right] \tag{5}$$

That is, $\mathbf{x}\beta$ is the $n \times 1$ matrix which contains the point predictions.

The matrix $\mathbf{x}$ is sometimes called the **design matrix**.

## 1.2 Mean Squared Error

At each data point, using the coefficients $\beta$ results in some error of prediction, so we have $n$ prediction errors. These form a vector:

$$\mathbf{e}(\beta) = \mathbf{y} - \mathbf{x}\beta \tag{6}$$

(You can check that this subtracts an $n \times 1$ matrix from an $n \times 1$ matrix.)

When we derived the least squares estimator, we used the mean squared error,

$$MSE(\beta) = \frac{1}{n}\sum_{i=1}^{n} e_i^2(\beta) \tag{7}$$

How might we express this in terms of our matrices? I claim that the correct form is

$$MSE(\beta) = \frac{1}{n}\mathbf{e}^T\mathbf{e} \tag{8}$$

To see this, look at what the matrix multiplication really involves:

$$[e_1 e_2 \ldots e_n] \left[ \begin{array}{c} e_1 \\ e_2 \\ \vdots \\ e_n \end{array} \right] \tag{9}$$

This, clearly equals $\sum_i e_i^2$, so the MSE has the claimed form.

Let us expand this a little for further use.

$$\begin{aligned} MSE(\beta) &= \frac{1}{n}\mathbf{e}^T\mathbf{e} && (10)\\ &= \frac{1}{n}(\mathbf{y} - \mathbf{x}\beta)^T(\mathbf{y} - \mathbf{x}\beta) && (11)\\ &= \frac{1}{n}(\mathbf{y}^T - \beta^T\mathbf{x}^T)(\mathbf{y} - \mathbf{x}\beta) && (12)\\ &= \frac{1}{n}\left(\mathbf{y}^T\mathbf{y} - \mathbf{y}^T\mathbf{x}\beta - \beta^T\mathbf{x}^T\mathbf{y} + \beta^T\mathbf{x}^T\mathbf{x}\beta\right) && (13) \end{aligned}$$

Notice that $(\mathbf{y}^T\mathbf{x}\beta)^T = \beta^T\mathbf{x}^T\mathbf{y}$. Further notice that this is a $1 \times 1$ matrix, so $\mathbf{y}^T\mathbf{x}\beta = \beta^T\mathbf{x}^T\mathbf{y}$. Thus

$$MSE(\beta) = \frac{1}{n}\left(\mathbf{y}^T\mathbf{y} - 2\beta^T\mathbf{x}^T\mathbf{y} + \beta^T\mathbf{x}^T\mathbf{x}\beta\right) \tag{14}$$

## 1.3 Minimizing the MSE

First, we find the gradient of the MSE with respect to $\beta$:

$$\begin{aligned}
\nabla MSE(\beta &= \frac{1}{n}\left(\nabla\mathbf{y}^T\mathbf{y} - 2\nabla\beta^T\mathbf{x}^T\mathbf{y} + \nabla\beta^T\mathbf{x}^T\mathbf{x}\beta\right) && (15)\\
&= \frac{1}{n}\left(0 - 2\mathbf{x}^T\mathbf{y} + 2\mathbf{x}^T\mathbf{x}\beta\right) && (16)\\
&= \frac{2}{n}\left(\mathbf{x}^T\mathbf{x}\beta - \mathbf{x}^T\mathbf{y}\right) && (17)
\end{aligned}$$

We now set this to zero at the optimum, $\widehat{\beta}$:

$$\mathbf{x}^T\mathbf{x}\widehat{\beta} - \mathbf{x}^T\mathbf{y} = 0 \tag{18}$$

This equation, for the two-dimensional vector $\widehat{\beta}$, corresponds to our pair of normal or estimating equations for $\hat{\beta}_0$ and $\hat{\beta}_1$. Thus, it, too, is called an estimating equation.

Solving,

$$\widehat{\beta} = (\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\mathbf{y} \tag{19}$$

That is, we've got one matrix equation which gives us both coefficient estimates.

If this is right, the equation we've got above should in fact reproduce the least-squares estimates we've already derived, which are of course

$$\hat{\beta}_1 = \frac{c_{XY}}{s^2_X} = \frac{\overline{xy} - \bar{x}\bar{y}}{\overline{x^2} - \bar{x}^2} \tag{20}$$

and

$$\hat{\beta}_0 = \overline{y} - \hat{\beta}_1\overline{x} \tag{21}$$

Let's see if that's right.

As a first step, let's introduce normalizing factors of $1/n$ into both the matrix products:

$$\widehat{\beta} = (n^{-1}\mathbf{x}^T\mathbf{x})^{-1}(n^{-1}\mathbf{x}^T\mathbf{y}) \tag{22}$$

Now let's look at the two factors in parentheses separately, from right to left.

$$\begin{aligned}
\frac{1}{n}\mathbf{x}^T\mathbf{y} &= \frac{1}{n}\left[\begin{array}{cccc} 1 & 1 & \ldots & 1 \\ x_1 & x_2 & \ldots & x_n \end{array}\right]\left[\begin{array}{c} y_1 \\ y_2 \\ \vdots \\ y_n \end{array}\right] && (23)\\
&= \frac{1}{n}\left[\begin{array}{c} \sum_i y_i \\ \sum_i x_i y_i \end{array}\right] && (24)\\
&= \left[\begin{array}{c} \overline{y} \\ \overline{xy} \end{array}\right] && (25)
\end{aligned}$$