Linear Algebra and Optimisation – Tutorial

Machine Learning

Linear Algebra I

If a non-zero $\mathbf{v}$ is an eigenvector of the matrix $\mathbf{A}$, then

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

for some scalar λ. This scalar is called an eigenvalue of $\mathbf{A}$. This can be written as:

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{I}\mathbf{v}$$

which then becomes:

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

The matrix $\mathbf{A} - \lambda\mathbf{I}$ must be singular. What are the values of λ such that this matrix becomes singular. These values are the roots to the equation

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

This is called the **characteristic equation**. The scalar λ is called the eigenvalue of $\mathbf{A}$ and $\mathbf{v}$ is called the eigenvector. There is an eigenvector for each eigenvalue.

1. Find out the eigenvalues for the matrix

$$\mathbf{A} = \begin{bmatrix} -6 & 3 \\ 4 & 5 \end{bmatrix}$$

Answer. The eigenvalues are the roots of of the equation

$$\det\left(\begin{bmatrix} -6 & 3 \\ 4 & 5 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) = 0$$

That is

$$\det\left(\begin{bmatrix} -6-\lambda & 3 \\ 4 & 5-\lambda \end{bmatrix}\right) = 0$$

or, $(-6-\lambda)(5-\lambda) - 3 \times 4 = 0$ or, $\lambda = -7, 6$.

2. Find out the eigenvectors of $\mathbf{A}$ for the eigenvalues you obtained in the previous question.

   Answer. Let $\mathbf{v} = [x, y]^\mathsf{T}$ be the eigenvector. Now, we have to solve

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

or,

$$\begin{bmatrix} -6 & 3 \\ 4 & 5 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \lambda \begin{bmatrix} x \\ y \end{bmatrix}$$

So, we get two equations for $\lambda = 6$:

$$-6x + 3y = 6x$$
$$4x + 5y = 6y$$

Simplifying this we get $y = 4x$. So, we can fix: $x = 1, y = 4$.
(Similarly, find the eigenvectors for $\lambda = -7$.)

Least-Squares Estimator I

Recall that we looked at the special case of describing a set of N data points using a linear model:

$$y = f(x_1, x_2, \ldots, x_d) = w_0 + w_1 x_1 + \cdots + w_d x_d$$

Here, $f(\mathbf{x})$ is a scalar function of $\mathbf{x}$, and y is the (true) output. In a vectorised notation, this equation is

$$y = f(\mathbf{x}) = w_0 + \mathbf{w} \cdot \mathbf{x}$$

- We want our function f to be able to correctly output a value that is equal to y. The function as parameters w_i. So, we want a set of w_is such that $f(\mathbf{x})$ outputs y.
- This problem of determining a function that will correctly describe data using this linear model is called **linear regression**. The function parameters w_i are called **regression coefficients**.

Least-Squares Estimator II

We will extend the data representation to include a 1 so that we can write $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x}$ instead of $1 + \mathbf{w} \cdot \mathbf{x}$. So, any ith data point is now written as $[1, x_{i,1}, x_{i,2}, \ldots, x_{i,d}]^\mathsf{T}$.

An extended representation of the data is then:

$$\mathbf{X} = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \ldots & x_{1,d} \\ 1 & x_{2,1} & x_{2,2} & \ldots & x_{2,d} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & x_{N,1} & x_{N,2} & \ldots & x_{N,d} \end{bmatrix}$$

The coefficient vector is

$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{bmatrix}$$

Least-Squares Estimator III

and, the (true) output in vectorised notation:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

Summary of dimensions:

$\mathbf{X}$	$N \times (d+1)$
$\mathbf{w}$	$(d+1) \times 1$
$\mathbf{y}$	$N \times 1$

Least-Squares Estimator IV

Let's re-write the linear regression equation for the data in vectorised form:

$$\begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \dots & x_{1,d} \\ 1 & x_{2,1} & x_{2,2} & \dots & x_{2,d} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_{N,1} & x_{N,2} & \dots & x_{N,d} \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

Or,

$$\mathbf{X}\mathbf{w} = \mathbf{y} \tag{1}$$

(Notice that: l.h.s is a matrix multiplication)

To obtain the unknown coefficient vector **w** for Eq. 1, it is necessary that $N \geq d$.

Least-Squares Estimator V

Case $N = d$:

That is, if $\mathbf{X}$ is square and non-singular, then

$$\mathbf{w} = \mathbf{X}^{-1}\mathbf{y}$$

Here, we also assumed that there is no noise in data, and there we could write Eq. 1.

Least-Squares Estimator VI

Case $N > d$:

- In reality, the number of data points is more than the number of function parameters.
- An exact solution satisfying all N equations is not possible.
- Data might be noisy.
- The model is not appropriate for describing the target system.

So, we would write a modified version of the equation:

$$\mathbf{X}\mathbf{w} \approx \mathbf{y} \tag{2}$$

That means, there is some amount of error between $f(\mathbf{x})$ and y for any given pair $(\mathbf{x}, y)$. We want to minimise this error.

Least-Squares Estimator VII

The commonly used error that is minimised is called squared-error function or squared-loss function, denoted as $\mathcal{L}$:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \mathbf{w}^\top \mathbf{x}_i\right)^2 \tag{3}$$

This is an averaged version of the squared-loss called **mean-squared** loss.

To find the w_i that minimise $\mathcal{L}$:

- We have to obtain partial derivative of $\mathcal{L}$ w.r.t. the w_i, and setting each partial derivative to 0.
- With d-dimensional data, this will result in d equations.
- Solving these d equations simultaneously, will give us the values of the w_i.

Least-Squares Estimator VIII

- But:
  - Solving the equations will require values from the data.
  - With large d, this representation is cumbersome.

Least-Squares Estimator IX

Let's look at some useful definitions:

Definition (Dot product)

For an N-dimensional vector $\mathbf{v} = [v_1, \ldots, v_N]^\mathsf{T}$, recall:

$$\mathbf{v} \cdot \mathbf{v} \; = \; \mathbf{v}^\mathsf{T}\mathbf{v} \; = \; \sum_{i=1}^{N} v_i^2$$

where $\mathbf{u} \cdot \mathbf{v}$ denotes the inner-product of the vectors $\mathbf{u}$ and $\mathbf{v}$.

Least-Squares Estimator X

Gradient of a scalar function Let $\mathbf{x} = [x_1, \ldots, x_n]^\mathsf{T}$ and let $f(\mathbf{x})$ be a **scalar function** of $\mathbf{x}$. Then the derivative of $f(\mathbf{x})$ w.r.t. $\mathbf{x}$, called the **gradient vector** or **gradient** of $f(\mathbf{x})$ is a column vector denoted by

$$\nabla_{\mathbf{x}} f(\mathbf{x}) \text{ or } \nabla f(\mathbf{x}) = \left[\frac{\partial f(\mathbf{x})}{\partial x_1}, \ldots, \frac{\partial f(\mathbf{x})}{\partial x_n}\right]^\mathsf{T}$$

Least-Squares Estimator XI

Gradient of a vector function Let $\mathbf{x} = [x_1, \ldots, x_n]^\mathsf{T}$ and let $\mathbf{f}(\mathbf{x})$ be a **vector function** of $\mathbf{x}$, denoted by $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}, \ldots, f_m(\mathbf{x}]^\mathsf{T}$. Then, the derivative of $\mathbf{f}(\mathbf{x})$ w.r.t. $\mathbf{x}$, called the **Jacobian matrix** or **Jacobian** of $\mathbf{f}(\mathbf{x})$, is an $m \times n$ matrix denoted by

$$\mathbf{J_f} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \vdots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \nabla_\mathbf{x}^\mathsf{T} f_1(\mathbf{x}) \\ \vdots \\ \nabla_\mathbf{x}^\mathsf{T} f_m(\mathbf{x}) \end{bmatrix}$$

Least-Squares Estimator XII

Hessian of a scalar function Let $\mathbf{x} = [x_1, \ldots, x_n]^{\mathsf{T}}$ and let $f(\mathbf{x})$ be a **scalar function** of $\mathbf{x}$. Then the second derivative of $f(\mathbf{x})$, called the **Hessian matrix** or **Hessian** of $f(\mathbf{x})$, is an $n \times n$ matrix denoted by

$$\mathbf{H}_f = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

which is:

$$\mathbf{H}_f = \begin{bmatrix} \frac{\partial}{\partial x_1}\left(\frac{\partial f}{\partial x_1}\right) & \frac{\partial}{\partial x_1}\left(\frac{\partial f}{\partial x_2}\right) & \cdots & \frac{\partial}{\partial x_1}\left(\frac{\partial f}{\partial x_n}\right) \\ \frac{\partial}{\partial x_2}\left(\frac{\partial f}{\partial x_1}\right) & \frac{\partial}{\partial x_2}\left(\frac{\partial f}{\partial x_2}\right) & \cdots & \frac{\partial}{\partial x_2}\left(\frac{\partial f}{\partial x_n}\right) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial}{\partial x_n}\left(\frac{\partial f}{\partial x_1}\right) & \frac{\partial}{\partial x_n}\left(\frac{\partial f}{\partial x_2}\right) & \cdots & \frac{\partial}{\partial x_n}\left(\frac{\partial f}{\partial x_n}\right) \end{bmatrix} = \begin{bmatrix} \nabla_{\mathbf{x}}^{\mathsf{T}} \frac{\partial f}{\partial x_1} \\ \vdots \\ \nabla_{\mathbf{x}}^{\mathsf{T}} \frac{\partial f}{\partial x_n} \end{bmatrix}$$

Least-Squares Estimator XIII

Gradient of a function (1) Let $\mathbf{c} = [c_1, \ldots, c_n]^\mathsf{T}$ and $\mathbf{x} = [x_1, \ldots, x_n]^\mathsf{T}$. Then the gradient of a linear scalar function $f(\mathbf{x}) = \mathbf{c}^\mathsf{T}\mathbf{x} = \mathbf{x}^\mathsf{T}\mathbf{c}$ w.r.t. $\mathbf{c}$

$$\nabla_{\mathbf{c}} f(\mathbf{x}) = \mathbf{x}$$

Gradient of a function (2) If $f(\mathbf{x}) = \mathbf{x}^\mathsf{T}\mathbf{x}$, then

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = 2\mathbf{x}$$

Gradient of a function (3) If $f(\mathbf{x}) = \mathbf{x}^\mathsf{T}\mathbf{A}\mathbf{x}$, then

$$\nabla_{\mathbf{x}} f = 2\mathbf{A}\mathbf{x}$$

Least-Squares Estimator XIV

3. If $\mathbf{X}^\mathsf{T}\mathbf{X}$ is non-singular (i.e. $\det(\mathbf{X}^\mathsf{T}\mathbf{X}) \neq 0$), show that $\mathbf{w} = (\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y}$ minimises the mean-squared loss $\mathcal{L}$.

   Answer. We have

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \mathbf{w}^\mathsf{T}\mathbf{x}_i\right)^2$$

   We can re-write this as

$$\mathcal{L} \;=\; \frac{1}{N}(\mathbf{y} - \mathbf{X}\mathbf{w})^\mathsf{T}(\mathbf{y} - \mathbf{X}\mathbf{w})$$

Least-Squares Estimator XV

Simplifying

$$\begin{aligned}
\mathcal{L} &= \frac{1}{N}(\mathbf{y}-\mathbf{X}\mathbf{w})^\top(\mathbf{y}-\mathbf{X}\mathbf{w}) \\
&= \frac{1}{N}(\mathbf{X}\mathbf{w}-\mathbf{y})^\top(\mathbf{X}\mathbf{w}-\mathbf{y}) \\
&= \frac{1}{N}((\mathbf{X}\mathbf{w})^\top-\mathbf{y}^\top)(\mathbf{X}\mathbf{w}-\mathbf{y}) \\
&= \frac{1}{N}\left[(\mathbf{X}\mathbf{w})^\top\mathbf{X}\mathbf{w}-\mathbf{y}^\top\mathbf{X}\mathbf{w}-(\mathbf{X}\mathbf{w})^\top\mathbf{y}+\mathbf{y}^\top\mathbf{y}\right] \\
&= \frac{1}{N}\mathbf{w}^\top\mathbf{X}^\top\mathbf{X}\mathbf{w}-\frac{2}{N}\mathbf{w}^\top\mathbf{X}^\top\mathbf{y}+\frac{1}{N}\mathbf{y}^\top\mathbf{y}
\end{aligned}$$

(The terms $\mathbf{y}^\top\mathbf{X}\mathbf{w}$ and $\mathbf{w}^\top\mathbf{X}^\top\mathbf{y}$ are transposes of each other and scalars, and therefore equal)

Least-Squares Estimator XVI

$\mathcal{L}$ is a scalar function. Differentiating it w.r.t $\mathbf{w}$:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \frac{\partial}{\partial \mathbf{w}} \left[\frac{1}{N} \mathbf{w}^{\mathsf{T}} \mathbf{X}^{\mathsf{T}} \mathbf{X} \mathbf{w} - \frac{2}{N} \mathbf{w}^{\mathsf{T}} \mathbf{X}^{\mathsf{T}} \mathbf{y} + \frac{1}{N} \mathbf{y}^{\mathsf{T}} \mathbf{y} \right]$$

simplifying,

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \frac{2}{N} \mathbf{X}^{\mathsf{T}} \mathbf{X} \mathbf{w} - \frac{2}{N} \mathbf{X}^{\mathsf{T}} \mathbf{y}$$

Equating to $\mathbf{0}$ gives:

$$\mathbf{X}^{\mathsf{T}} \mathbf{X} \mathbf{w} = \mathbf{X}^{\mathsf{T}} \mathbf{y}$$

or:

$$\mathbf{w} = (\mathbf{X}^{\mathsf{T}} \mathbf{X})^{-1} \mathbf{X}^{\mathsf{T}} \mathbf{y}$$

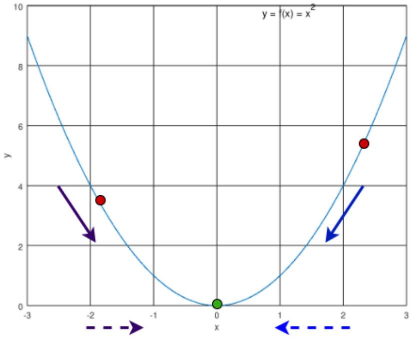
Optimisation I

Let's look at minimisation problems for functions that are continuous and differentiable.

- If the derivative of the function is positive, the function is increasing.
  - Don't move in that direction, because you'll be moving away from a minimum.
- If the derivative of the function is negative, the function is decreasing.
  - Keep going, since you're getting closer to a minimum.

Optimisation II

Let $f(x) = x^2$. The function looks like this:



The arrows show movement of next functional value, and the dotted arrows show the corresponding direction of movement of x.

Optimisation III

Here is a very simple gradient descent procedure:

1. Initialize x to some value
2. **while** stopping criterion is not met
   2.1 Calculate the gradient of the function, $\nabla_x f$
   2.2 $x := x - \eta \nabla_x f$
3. **return** x

Notice step 2.2. above: x will move right, if $\nabla_x f$ is negative, and it will move left, if $\nabla_x f$ is positive.

Optimisation IV

4. Using gradient descent, obtain the value of x that minimizes $f(x) = (x-2)^2 - 5$. Starting value of $x = 3$ and $\eta = 1$.

   Answer. Derivative of f w.r.t. x: $\nabla f = 2(x-2)$

   - $x = 3$: $\nabla f|_{x=3} = 2$; $x = 3 - 2 = 1$; $f(1) = -4$
   - $x = 1$: $\nabla f|_{x=1} = -2$; $x = 1 - (-2) = 3$; $f(3) = -4$.
   - ... gets repeated.

5. Solve the same question with same starting point, but with $\eta = 0.5$.

Answer. Derivative of f w.r.t. x: $\nabla f = 2(x - 2)$

- $x = 3$: $\nabla f|_{x=3} = 2$; $x = 3 - 0.5 \times 2 = 2$; $f(2) = -5$
- $x = 2$: $\nabla f|_{x=2} = 0$; $x = 2 - 0.5 \times 0 = 2$; $f(2) = -5$.
- $x = 2$: $\nabla f|_{x=2} = 0$; $x = 2 - 0.5 \times 0 = 2$; $f(2) = -5$.
- Value of f doesn't change further. So, stopping criterion met. Return $x = 2$. This is same as the exact solution i.e. Find root of $\nabla f = 0$.

Optimisation VI

Gradient descent is guaranteed to eventually find a local minimum if:

- the learning rate is set appropriately (sometimes, using adaptive learning rate); $\eta \in [0.0001, 1]$.
- a finite local minimum exists (i.e. the function doesn't keep decreasing forever).

Optimisation VII

Various stopping criteria for gradient descent:

- Stop when the norm of the gradient is below some threshold, θ

$$||\nabla f|| < \theta$$

  This is checking the distant the gradient is from the origin, $\mathbf{0}$.
- Maximum number of iterations is reached.

Optimisation VIII

It is straightforward to extend the gradient descent procedure to scalar functions with multiple variables.

6. Let $f(x_1, x_2) = 3x_1^2 - 2x_1x_2 + x_2^2 - 5$. Initial values $x_1 = 1$, $x_2 = 1$. Fix $\eta = 1$.

   Answer. Present value of f: $f(1, 1) = 3 - 2 + 1 - 5 = -3$.
   The partial derivatives are:

$$\nabla_{x_1} f = 6x_1 - 2x_2$$
$$\nabla_{x_2} f = 2x_2 - 2x_1$$

Optimisation IX

Update the present $x_{1,2}$:

$$
\begin{aligned}
x_1 &= x_1 - \eta \nabla_{x_1} f \\
&= 1 - (6 - 2) = -3 \\
x_2 &= x_2 - \eta \nabla_{x_2} f \\
&= 1 - (2 - 2) = 1
\end{aligned}
$$

New value of f: $f(-3, 1) = 29$. Update the present $x_{1,2}$ using gradients:

$$
\begin{aligned}
x_1 &= x_1 - \eta \nabla_{x_1} f \\
&= -3 - (-18 - 2) = 17 \\
x_2 &= x_2 - \eta \nabla_{x_2} f \\
&= 1 - (-6 - 2) = 9
\end{aligned}
$$

New value of f: $f(17, 9) = 637$.

7. Solve the above question with $\eta = 0.1$.

Answer. Update the present $x_{1,2}$:

$$x_1 = 1 - 0.1(6 - 2) = 0.6$$
$$x_2 = 1 - 0.1(2 - 2) = 1$$

New value of f: $f(0.6, 1) = -4.12$. Update the present $x_{1,2}$ using gradients:

$$x_1 = 0.6 - 0.1(3.6 - 2) = 0.44$$
$$x_2 = 1 - 0.1(2 - 1.2) = 0.92$$

New value of f: $f(0.44, 0.92) = -4.38$.

Optimisation XI

This is function surface and its contour: