LECTURE 4 - CURVE FITTING & MATRICES FOR DERIVATIVES

1. Curve Fitting & Matrices for Derivatives

- Suppose we are given a training set comprising of N observations, $X = (x_1, x_2, \cdots, x_N)^T$, where $x_i = (x_{i1}, x_{i2}, ..., x_{iM})$, along with desired outputs $\mathbf{d} = (d_1, d_2, \cdots, d_N)^T$.
- Now we must assume a model (Figure 1). Let's assume a polynomial function as our model:

$$\begin{aligned} \mathbf{y}(\mathbf{x}) &= f(\mathbf{w}; \mathbf{x}) \\ &= \beta + w_1 x + w_2 x^2 + \cdots + w_M x^M \\ &= \beta 1 + w_1 x + w_2 x^2 + \cdots + w_M x^M \\ &= w_0 x^0 + w_1 x^1 + w_2 x^2 + \cdots + w_M x^M \\ &= \sum_{j=0}^{M} w_j x^j \end{aligned} \tag{1}$$

- The parameter β is often called *bias* and it is used to adjust the model to fluctuations of noise present in the data. For simplicity, β is taken as part of the parameter vector $\mathbf{w}$.
- In matrix notation, we can find two different ways to represent the same output. The reason is that the data matrix can be written in two different ways.

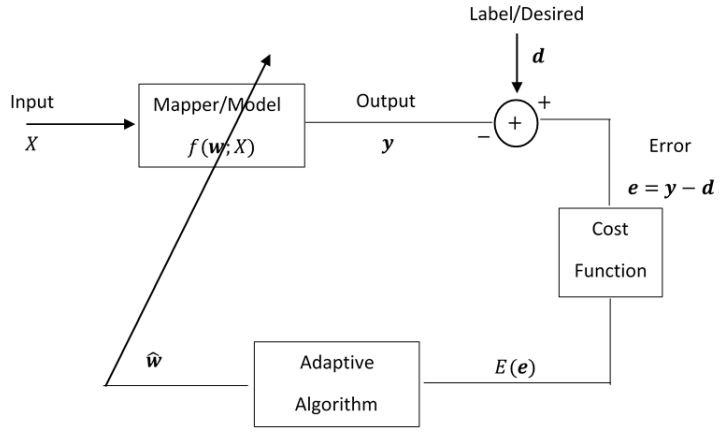


Figure 1. Block diagram of an adaptive filter.

1. The data matrix contains the samples in rows

$$X = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^M \\ 1 & x_2 & x_2^2 & \cdots & x_2^M \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_N & x_N^2 & \cdots & x_N^M \end{bmatrix} \tag{2}$$

This way, the output vector, in matrix notation, is written as a column vector in the form:

$$\mathbf{y} = X\mathbf{w} \tag{3}$$

2. The data matrix contains the samples in columns

$$X = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_N \\ x_1^2 & x_2^2 & \cdots & x_N^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^M & x_2^M & \cdots & x_N^M \end{bmatrix} \tag{4}$$

Again, the output vector, in matrix notation, is written as a column vector:

$$\mathbf{y} = X^T\mathbf{w} \tag{5}$$

- The information contained in both matrices is the same. It is important, though, to understand the notation for coding your algorithm. In many machine learning applications and software such as MATLAB, data samples are concatenated in the data matrix rows.
- Let's choose the first notation for convenience (samples in rows): X is an $N\times(M+1)$ matrix, $\mathbf{w}$ is a $(M+1)\times 1$ column vector, $\mathbf{d}$ is a $N\times 1$ column vector and $\mathbf{y} = X\mathbf{w}$ is a $N\times 1$ column vector.
- Now we must train this model by estimating the unknown parameters ($\mathbf{w}$) that maps the training data, X, to their desired values, $\mathbf{d}$, given some assumed value for M (model order).
  - Note that including the bias in the parameter vector $\mathbf{w}$ makes it such that its dimension becomes $(M+1)\times 1$.
- So, we have N points from which to estimate $\mathbf{w}$. We can minimize the **squared error** to estimate the parameters.
  - Recall that the *instantaneous* error is defined as $\mathbf{e} = \mathbf{y} - \mathbf{d}$, where $\mathbf{y}$ is the output vector of size $N\times 1$, $\mathbf{d}$ is the label vector of size $N\times 1$, therefore $\mathbf{e}$ is the error vector of size $N\times 1$. The **squared error** in vector form is defined as $\mathbf{e}^T\mathbf{e}$.

$$
\begin{aligned}
\arg\min_{\mathbf{w}} E(\mathbf{e}) &= \frac{1}{2}\mathbf{e}^T\mathbf{e} \\
&= \frac{1}{2}\sum_{n=1}^{N} e_n^2 \\
&= \frac{1}{2}\sum_{n=1}^{N} (y_n - \mathbf{d}_n)^2 \\
&= \frac{1}{2}\sum_{n=1}^{N} (f(\mathbf{w}, \mathbf{x}_n) - \mathbf{d}_n)^2 \\
&= \frac{1}{2}\sum_{n=1}^{N} \left(\sum_{j=0}^{M} w_j \mathbf{x}_n^j - \mathbf{d}_n\right)^2 \\
&= \frac{1}{2}\sum_{n=1}^{N} (\mathbf{x}_n\mathbf{w} - \mathbf{d}_n)^2 \\
&= \frac{1}{2}\left(\left(\sum_{n=1}^{N} (\mathbf{x}_n\mathbf{w} - \mathbf{d}_n)^2\right)^{1/2}\right)^2 \\
&= \frac{1}{2}\left\|X\mathbf{w} - \mathbf{d}\right\|_2^2 \\
&= \frac{1}{2}\left\|\mathbf{y} - \mathbf{d}\right\|_2^2 \\
&= \frac{1}{2}\left\|\mathbf{e}\right\|_2^2
\end{aligned} \tag{6}
$$

where X is an $N \times (M+1)$ data matrix of the form

$$
X = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^M \\ 1 & x_2 & x_2^2 & \cdots & x_2^M \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_N & x_N^2 & \cdots & x_N^M \end{bmatrix} \tag{7}
$$

The parameter vector w is an $(M+1) \times 1$ vector of the form

$$
\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_M \end{bmatrix} = \begin{bmatrix} w_0 & w_1 & \cdots & w_M \end{bmatrix}^T \tag{8}
$$

and $\mathbf{d}$ is an $N \times 1$ vector of the output labels of the form

$$\mathbf{d} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix} = \begin{bmatrix} d_1 & d_2 & \cdots & d_N \end{bmatrix}^T \tag{9}$$

- What are the set of parameters $\mathbf{w} = \{w_i\}_{i=0}^M$ that fit the train data X to the labels $\mathbf{t}$?
  - *Answer:* The set of parameters $\mathbf{w} = \{w_i\}_{i=0}^M$ that fit the train data X to the labels $\mathbf{d}$ are the ones that minimize the cost function. The minimum of a convex cost function can be found by setting the gradient to zero, $\nabla E(\mathbf{w}) = 0$.
- Exercises: consider the square matrices A, B of size $N \times N$ scalars a, b of size 1×1 and vectors $\mathbf{v}$, $\mathbf{w}$ of size $N \times 1$. Let's simplify or solve the following terms:
  - $\left(A^T\right)^T$
  - $(AB)^T$
  - $\left(AB^T\right)^T$
  - $(A+B)^T$
  - $(AB+B)^T$
  - $\left(A^TB+B\right)^T$
  - $\frac{\partial A\mathbf{w}}{\partial \mathbf{w}}$
  - $\frac{\partial \mathbf{w}^T A}{\partial \mathbf{w}}$
  - $\frac{\partial \mathbf{w}^T A\mathbf{w}}{\partial \mathbf{w}}$
  - $\frac{\partial \mathbf{w}^T A^T A\mathbf{w}}{\partial \mathbf{w}}$
  - $\frac{\partial b^T Aa}{\partial A}$
  - $\frac{\partial A\mathbf{v}}{\partial B}$
  - $\frac{\partial A\mathbf{v}}{\partial A}$
  - $\frac{\partial \mathbf{w}^T A\mathbf{v}}{\partial A}$

- Let's now solve for the optimal parameters in matrix notation:

$$
\begin{aligned}
\nabla E(\mathbf{e}) &= 0 \\
\frac{\partial E(\mathbf{e})}{\partial \mathbf{w}} &= 0 \\
\frac{\partial}{\partial \mathbf{w}}\left[\frac{1}{2}\mathbf{e}^T\mathbf{e}\right] &= 0 \\
\frac{\partial}{\partial \mathbf{w}}\left[\frac{1}{2}\left(\mathbf{y}-\mathbf{d}\right)^T\left(\mathbf{y}-\mathbf{d}\right)\right] &= 0 \\
\frac{\partial}{\partial \mathbf{w}}\left[\frac{1}{2}\left(X\mathbf{w}-\mathbf{d}\right)^T\left(X\mathbf{w}-\mathbf{d}\right)\right] &= 0 \\
\frac{1}{2}\frac{\partial}{\partial \mathbf{w}}\left[\left(X\mathbf{w}-\mathbf{d}\right)^T\left(X\mathbf{w}-\mathbf{d}\right)\right] &= 0 \\
\frac{\partial}{\partial \mathbf{w}}\left[\left(X\mathbf{w}-\mathbf{d}\right)^T\left(X\mathbf{w}-\mathbf{d}\right)\right] &= 0 \\
\frac{\partial}{\partial \mathbf{w}}\left[\left(\mathbf{w}^TX^T-\mathbf{d}^T\right)\left(X\mathbf{w}-\mathbf{d}\right)\right] &= 0 \\
\frac{\partial}{\partial \mathbf{w}}\left[\mathbf{w}^TX^TX\mathbf{w}-\mathbf{w}^TX^T\mathbf{d}-\mathbf{d}^TX\mathbf{w}+\mathbf{d}^T\mathbf{d}\right] &= 0 \\
\left(X^TX\mathbf{w}\right)^T+\mathbf{w}^TX^TX-\left(X^T\mathbf{d}\right)^T-\mathbf{d}^TX &= 0 \\
2\mathbf{w}^TX^TX &= 2\mathbf{d}^TX \\
\left(\mathbf{w}^TX^TX\right)^T &= \left(\mathbf{d}^TX\right)^T \\
X^TX\mathbf{w} &= X^T\mathbf{d} \\
\mathbf{w} &= \left(X^TX\right)^{-1}X^T\mathbf{d}
\end{aligned}
\tag{10}
$$

- If the cost function is a convex function, then $\nabla E(\mathbf{w}) = 0$ gives the **global minimum** and the $\mathbf{w}$ that solves this equation is called the **optimal value**.

2. Error Surface - Hessian

- If the cost function is non-convex (Figure 2 on the right), we may have more than one *critical point* at which $\nabla E(\mathbf{w}) = \frac{\partial E(\mathbf{w})}{\partial \mathbf{w}} = 0$. How can you differentiate a minimum, maximum and saddle critical point if for all three the gradient is zero, $\nabla E(\mathbf{w}) = 0$?
- We compute the second derivative or **Hessian matrix** $H(E(\mathbf{w}))$ at each critical point. For a solution $\mathbf{w}^*$ of $\nabla E(\mathbf{w}) = 0$, if the determinant of $H(E(\mathbf{w}^*)) > 0$ than $\mathbf{w}^*$ is a local minimum, if $H(E(\mathbf{w}^*)) < 0$ than $\mathbf{w}^*$ is a local maximum, if $H(E(\mathbf{w}^*)) < 0$ than $\mathbf{w}^*$ is a saddle point.

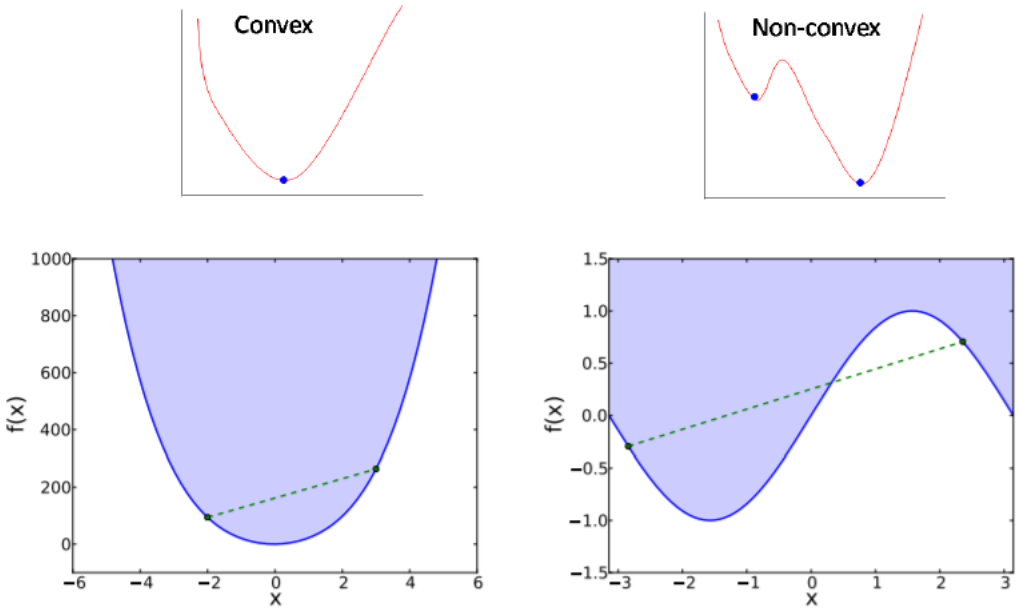


FIGURE 2. For any line connecting any two points, the line is contained within the boundaries of the function. Example of a convex function (left), and a non-convex function (right).

- For example, consider the parabola function $f(\mathbf{x}) = \mathbf{x}^T\mathbf{x}$ where $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_M \end{bmatrix}^T$.

$$
\begin{aligned}
\nabla f(\mathbf{x}) &= 0 \\
\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} &= 0 \\
\begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} & \cdots & \frac{\partial f}{\partial x_M} \end{bmatrix}^T &= \mathbf{0} \\
\begin{bmatrix} \frac{\partial \mathbf{x}^T\mathbf{x}}{\partial x_1} & \frac{\partial \mathbf{x}^T\mathbf{x}}{\partial x_2} & \cdots & \frac{\partial \mathbf{x}^T\mathbf{x}}{\partial x_M} \end{bmatrix}^T &= \mathbf{0} \\
\begin{bmatrix} 2x_1 & 2x_2 & \cdots & 2x_M \end{bmatrix}^T &= \mathbf{0} \\
\mathbf{x}^* &= \mathbf{0} \\
\mathbf{x}^* &= \begin{bmatrix} 0 & 0 & \cdots & 0 \end{bmatrix}^T
\end{aligned}
\tag{11}
$$

- The solution for $\nabla f(\mathbf{x}) = 0$ is $\mathbf{x}^* = \begin{bmatrix} 0 & 0 & \cdots & 0 \end{bmatrix}^T$. Is this point a maximum or a minimum.

- Let's compute the Hessian matrix given by:

$$
\begin{aligned}
H(f(\mathbf{x})) &= \begin{bmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial x_1^2} & \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_1} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_M \partial x_1} \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_2} & \frac{\partial^2 f(\mathbf{x})}{\partial x_2^2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_M \partial x_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_M} & \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_M} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_M^2} \end{bmatrix} \\
H(f(\mathbf{x})) &= \begin{bmatrix} 2 & 0 & \cdots & 0 \\ 0 & 2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 2 \end{bmatrix}
\end{aligned} \tag{12}
$$

- The determinant of $H(f(\mathbf{x}))$ is $\det(H(f(\mathbf{x})) = 2^M > 0$, then the solution $\mathbf{x}^* = \begin{bmatrix} 0 & 0 & \cdots & 0 \end{bmatrix}^T$ is a local minimum.
- From last lecture, we have found the gradient of our cost function, $E(\mathbf{w}) = \frac{1}{2} \|X\mathbf{w} - \mathbf{d}\|_2^2$, to be:

$$
\begin{aligned}
\nabla E(\mathbf{w}) &= \begin{bmatrix} \frac{\partial E(\mathbf{w})}{\partial w_0} \\ \frac{\partial E(\mathbf{w})}{\partial w_1} \\ \vdots \\ \frac{\partial E(\mathbf{w})}{\partial w_M} \end{bmatrix} \\
&= \begin{bmatrix} \sum_{n=1}^N \left(\sum_{j=0}^M w_j x_n^j - d_n \right) x_n^0 \\ \sum_{n=1}^N \left(\sum_{j=0}^M w_j x_n^j - d_n \right) x_n^1 \\ \vdots \\ \sum_{n=1}^N \left(\sum_{j=0}^M w_j x_n^j - d_n \right) x_n^M \end{bmatrix}
\end{aligned} \tag{13}
$$

- The Hessian of our cost function is:

$$
\begin{aligned}
H(E(\mathbf{w})) &= \frac{\partial^2 E(\mathbf{w})}{\partial \mathbf{w}^2} \\
&= \begin{bmatrix} \frac{\partial^2 E(\mathbf{w})}{\partial w_0^2} & \frac{\partial^2 E(\mathbf{w})}{\partial w_1 \partial w_0} & \cdots & \frac{\partial^2 E(\mathbf{w})}{\partial w_M \partial w_0} \\ \frac{\partial^2 E(\mathbf{w})}{\partial w_0 \partial w_1} & \frac{\partial^2 E(\mathbf{w})}{\partial w_1^2} & \cdots & \frac{\partial^2 E(\mathbf{w})}{\partial w_M \partial w_1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 E(\mathbf{w})}{\partial w_0 \partial w_M} & \frac{\partial^2 E(\mathbf{w})}{\partial w_1 \partial w_M} & \cdots & \frac{\partial^2 E(\mathbf{w})}{\partial w_M^2} \end{bmatrix} \\
&= \begin{bmatrix} \sum_{n=1}^N (x_n^0)^2 & \sum_{n=1}^N x_n^1 x_n^0 & \cdots & \sum_{n=1}^N x_n^M x_n^0 \\ \sum_{n=1}^N x_n^0 x_n^1 & \sum_{n=1}^N (x_n^1)^2 & \cdots & \sum_{n=1}^N x_n^M x_n^1 \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{n=1}^N x_n^0 x_n^M & \sum_{n=1}^N x_n^1 x_n^M & \cdots & \sum_{n=1}^N (x_n^M)^2 \end{bmatrix} \\
&= X^T X
\end{aligned} \tag{14}
$$

- We can also solve for the Hessian in matrix form as we did for the gradient in Equation 10:

$$
\begin{aligned}
H(E(\mathbf{w})) &= \frac{\partial^2 E(\mathbf{w})}{\partial \mathbf{w}^2} \\
&= \frac{\partial^2}{\partial \mathbf{w}^2} \left[\frac{1}{2} \left\| X\mathbf{w} - \mathbf{d} \right\|_2^2 \right] \\
&= \frac{1}{2} \frac{\partial^2}{\partial \mathbf{w}^2} \left[(X\mathbf{w} - \mathbf{d})^T (X\mathbf{w} - \mathbf{d}) \right] \\
&= \frac{1}{2} \frac{\partial}{\partial \mathbf{w}} \left[\frac{\partial}{\partial \mathbf{w}} \left[\mathbf{w}^T X^T X \mathbf{w} - \mathbf{w}^T X^T \mathbf{d} - \mathbf{d}^T X \mathbf{w} + \mathbf{d}^T \mathbf{d} \right] \right] \\
&= \frac{1}{2} \frac{\partial}{\partial \mathbf{w}} \left[(X^T X \mathbf{w})^T + \mathbf{w}^T X^T \mathbf{w} - (X^T \mathbf{d})^T - \mathbf{d}^T X \right] \\
&= \frac{1}{2} \frac{\partial}{\partial \mathbf{w}} \left[2\mathbf{w}^T X^T X - 2\mathbf{d}^T X \right] \\
&= \frac{1}{2} \left[2(X^T X)^T \right] \\
&= X^T X
\end{aligned} \tag{15}
$$

- **Example:** What is the gradient and the Hessian of the function $f : \mathbb{R}^2 \to \mathbb{R}$ given as $f(\mathbf{x}) = x_1^3 + x_2^3 - 3x_1 x_2$?

3. Eigenvectors and Eigenvalues

- Let us refresh some concepts....
  For any transformation $T : \mathbb{R}^n \to \mathbb{R}^n$, we are interested to find the vectors $\mathbf{v}$ that

only get **scaled** by the transformation T, that is $T(\mathbf{v}) = \lambda\mathbf{v}$. These vectors have important properties for reasons we will later understand.

- Consider now the transformation $T : \mathbb{R}^2 \to \mathbb{R}^2$ that flips vectors across the line $y = 2x$. For example, the transformation of the vector $\mathbf{v}_2$ in Figure 3 is $T(\mathbf{v}_2)$. This vector's direction changed! Now consider the vector $\mathbf{v}_1$, the transformation that flips vectors across the line $y = 2x$ will not affect this vector, so $T(\mathbf{v}_1) = \mathbf{v}_1 = 1\mathbf{v}_1$. Or $\mathbf{v}_1$ was **only scaled** by a factor of 1.
- Consider now the vector $\mathbf{v}_3 = [-2, 1]^T$, if you flip it across the line, the results is $T(\mathbf{v}_3) = [1, 2]^T = -1\mathbf{v}_3$, and again this vector only got scaled by a factor of -1.
- It is important to note that $\mathbf{v}_1$ and $\mathbf{v}_3$ are orthogonal and are called the **eigenvectors** of the transformation T. The scales that each vector undergo by the transformation T are called the **eigenvalues**.

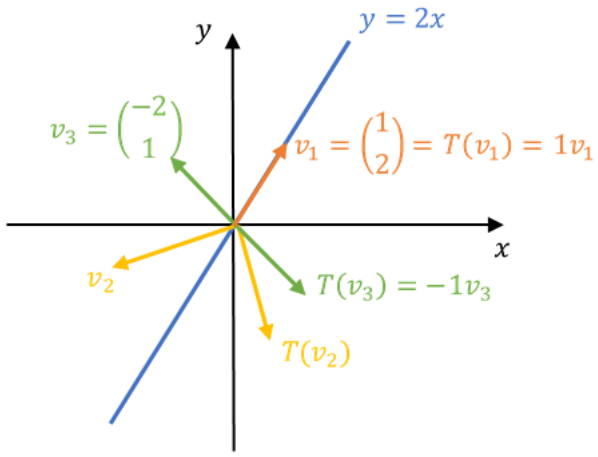


FIGURE 3. Example of linear transformation and its eigenvectors.

- In this example, $\mathbf{v}_1$ and $\mathbf{v}_3$ are eigenvectors of the transformation T with corresponding eigenvalues 1 and -1, respectively.
- Now, any square matrix A is an example of a linear transformation, $T(\mathbf{v}) = A\mathbf{v}$, that is, A is just a matrix representation of the transformation T. So we can find the eigenvectors and eigenvalues of A that satisfy:

$$A\mathbf{v} = \lambda\mathbf{v} \tag{16}$$

- We can rewrite this equation as follows:

$$\begin{aligned} A\mathbf{v} &= \lambda\mathbf{v} \\ A\mathbf{v} - \lambda\mathbf{v} &= 0 \\ A\mathbf{v} - \lambda I\mathbf{v} &= 0 \\ (A - \lambda I)\,\mathbf{v} &= 0, \forall\mathbf{v} \neq \overrightarrow{0} \end{aligned} \tag{17}$$

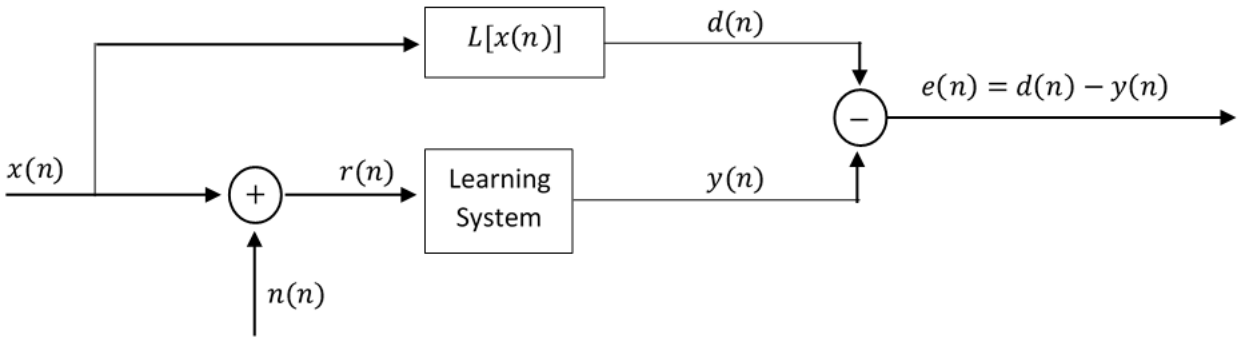
- Again, all vectors $\mathbf{v}$ that solve the previous equation are called the eigenvectors. A trivial solution is having $\mathbf{v} = \overrightarrow{0}$, however the 0 vector is not much useful. So the eigenvectors $\mathbf{v}$ of the matrix A are elements of the *null space* of $A - \lambda I$, $\mathbf{v} \in N(A - \lambda I) = \left\{\mathbf{v} \in \mathbb{R}^N : (A - \lambda I)\mathbf{v} = 0\right\}$.
- We know that for a certain matrix A, the columns of A are *linearly independent* **if and only if** (iff) $N(A) = \{\overrightarrow{0}\}$.
- So if we have the matrix $A - \lambda I$ whose null space includes more than just the zero vector, $\overrightarrow{0}$, then $\lambda I - A$ must contain *linearly dependent columns*, which also means that this matrix is not invertible or that $\det(A - \lambda I) = 0$.
  - $A\mathbf{v} = \lambda\mathbf{v}$ for non-zero $\mathbf{v}$ **iff** $\det(\lambda I - A) = 0$.
  - I is the identity matrix the same size as A.
- The eigenvectors are also called the **principal axes**.
- The principal axes of the surfaces of equal error of MSE, $E(\mathbf{e})$, correspond to the eigenvectors of the auto-correlation function $R = E[X^T X]$ and the rate of change of the gradient along the principal axes of the error surface contour correspond to the eigenvalues.
- **Exercise:** For the matrix $A = \begin{bmatrix} 1 & 2 \\ 4 & 3 \end{bmatrix}$, find the eigenvectors $\mathbf{v}$ and its corresponding eigenvalues λ. Hint: The determinant of a 2×2 matrix, $D = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$, is $\det(D) = ad - bc$.

4. Auto- and Cross-Correlation Functions

- During the war, Wiener worked on aircraft fire control. (a plane for 10,000 rounds of ammunition). Wiener worked on a electronic gunsight.
  - Where to aim the gun? We must predict the place where the plan will be in the future!!
  - The contribution was to think of the plane position in statistical terms. From all possible paths choose the one that is more probable.
  - This was a break with deterministic thinking. Helped establish the conceptual framework for information theory.
- We will formulate the problem of pure prediction (but it can be extended also to estimation).
- A random signal $x(n)$ in an additive random noise $n(n)$, is going to be sent through some linear system, such that the output $y(n)$ is an approximation to some linear operator $L[x(n)]$ on the input signal.
- The filter is designed such that the **mean square error** (MSE)

$$J = E\left[e^2(n)\right] = E\left[\left(L[x(n)] - y(n)\right)^2\right] \tag{18}$$

is to be minimized.



- We can now design our model just like we did for spatial features. Using matrix notation, our input vector is

$$\mathbf{x}(n) = \begin{bmatrix} x(n) & x(n-1) & \cdots & x(n-M) \end{bmatrix}^T \tag{19}$$

the parameters $\mathbf{w}$ are

$$\mathbf{w} = \begin{bmatrix} w_0 & w_1 & \cdots & w_M \end{bmatrix}^T \tag{20}$$

and so the output can be written as

$$\mathbf{y}(n) = \sum_{k=0}^{M} x(n-k)w(k) = \mathbf{x}^T(n)\mathbf{w} \tag{21}$$

- Now the MSE is

$$\begin{aligned} \arg\min_{\mathbf{w}} J &= \arg\min_{\mathbf{w}} E[e^2(n)] \\ &= \arg\min_{\mathbf{w}} E\left[\left(\sum_{k=0}^{M} x(n-k)w(k) - d(n)\right)^2\right] \end{aligned} \tag{22}$$

- To minimize the error, we equate the gradient to zero:

$$\begin{aligned} \nabla J &= 0 \\ \frac{\partial J}{\partial \mathbf{w}} &= 0 \end{aligned} \tag{23}$$

For every element of the gradient:

(24)
$$\begin{aligned}
\frac{\partial J}{\partial w_i} &= 0 \\
\frac{\partial}{\partial w_i}\left[E\left[\left(\sum_{k=0}^{M} x(n-k)w(k) - d(n)\right)^2\right]\right] &= 0 \\
E\left[\frac{1}{2}\frac{\partial}{\partial w_i}\left[\left(\sum_{k=0}^{M} x(n-k)w(k) - d(n)\right)\right]\left(\sum_{k=0}^{M} x(n-k)w(k) - d(n)\right)\right] &= 0 \\
\frac{1}{2}E\left[x(n-i)\left(\sum_{k=0}^{M} x(n-k)w(k) - d(n)\right)\right] &= 0 \\
E\left[x(n-i)\left(\sum_{k=0}^{M} x(n-k)w(k) - d(n)\right)\right] &= 0 \\
E\left[\sum_{k=0}^{M} x(n-i)x(n-k)w(k) - x(n-i)d(n)\right] &= 0 \\
E\left[\sum_{k=0}^{M} x(n-i)x(n-k)w(k)\right] - E\left[x(n-i)d(n)\right] &= 0 \\
E\left[\sum_{k=0}^{M} x(n-i)x(n-k)w(k)\right] &= E\left[x(n-i)d(n)\right] \\
\sum_{k=0}^{M} E\left[x(n-i)x(n-k)\right]w(k) &= E\left[x(n-i)d(n)\right]
\end{aligned}$$

- Calling

(25)
$$\mathbf{p}(i) = E\left[x(n-i)d(i)\right]$$

the **cross-correlation function** between $d(n)$ and the input, and

(26)
$$R(i-k) = E\left[x(n-i)x(n-k)\right]$$

the **auto-correlation function** of the input, we get

(27)
$$\mathbf{p}(i) = \sum_{k=0}^{M} R(i-k)w(k), i = 0, 1, \cdots, M$$

- This set of equation $\mathbf{p}(i)$ is known as the **normal equations**, or the **Wiener-Hopf equations**.

- In our matrix notation, $\mathbf{p}$ is a column vector and $\mathbf{w}$ also a column vector, we can write

$$\begin{aligned} \mathbf{p}(i) &= \sum_{k=0}^{M} R(i-k)w(k) \\ \mathbf{p} &= R\mathbf{w} \end{aligned}$$

$$\begin{bmatrix} p(0) \\ p(1) \\ \vdots \\ p(M) \end{bmatrix} = \begin{bmatrix} r(0) & r(1) & \cdots & r(M) \\ r(1) & r(0) & & r(M-1) \\ \vdots & \vdots & \ddots & \vdots \\ r(M) & r(M-1) & \cdots & r(0) \end{bmatrix} \begin{bmatrix} w(0) \\ w(1) \\ \vdots \\ w(M) \end{bmatrix} \tag{28}$$

$$\mathbf{w} = R^{-1}\mathbf{p}$$

- Each element of R can be written as

$$\begin{aligned} r(i) &= E\left[x(n-i)x(n-k)\right] \\ r(i) &= \frac{1}{N}\sum_{k=1}^{N} x(n-i)x(n-k) \end{aligned} \tag{29}$$

For example,

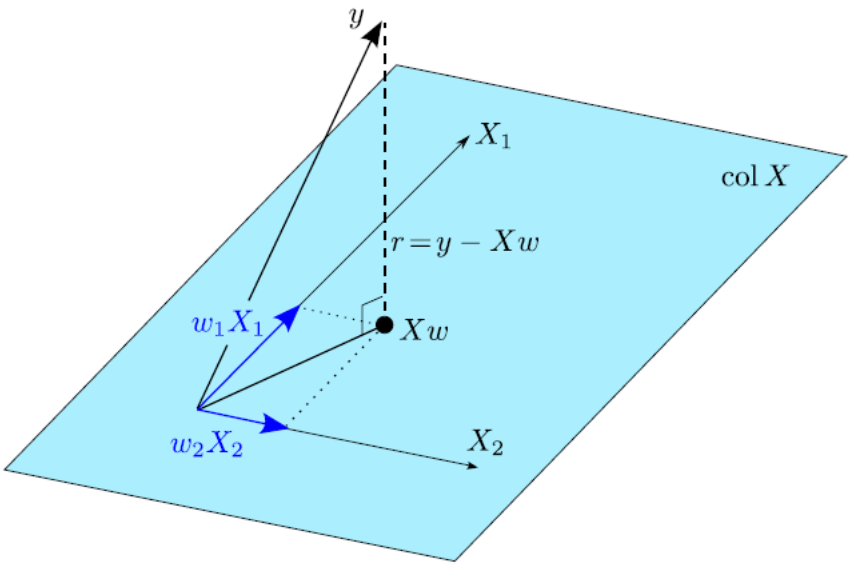
$$\begin{aligned} r(0) &= E\left[x(n)x(n-k)\right] \\ r(M-1) &= E\left[x(n-(M-1))x(n-k)\right] \\ r(M) &= E\left[x(n-M)x(n-k)\right] \end{aligned} \tag{30}$$

- The auto-correlation function is one of the tools used to find patterns in the data X. In other words, R computes how correlated the data is with itself. Its size is $M \times M$.
- The cross-correlation $\mathbf{p} = X^T\mathbf{d}$ computes how correlated the data is with the desired output values and its dimension is $M \times 1$.
- *When we have a large model order M and a small number of data points N, what happens to the auto-correlation matrix R term in our code?*
  - If $M > N$ then there exists multiple solutions that solve the normal equations, $\mathbf{p}(i) = \sum_{k=0}^{M} R(i-k)w(k), i = 0, 1, \cdots, M$, because we will have more unknown coefficients $\mathbf{w}$ than sample points $x(n)$.
  - In other words, the auto-correlation function R will have linearly dependent columns, which also means the matrix is not full rank, the $\det(R) = 0$ and R is not invertible. If R is not invertible, than we cannot compute the optimal weights $w = R^{-1}\mathbf{p}$.
  - A possible solution to this problem is to add a small perturbation to R such that R^{-1} exists, $R_{new} = R + \epsilon * I$, $\epsilon \in \mathbb{R}$ is small and I is the identity matrix. This is called *regularization*.
  - Another solution is to add more samples. The number of sample points needed to densely populate a space grows quickly with dimensionality. If you add

informative, discriminating features – more features can be helpful – but you will need exponentially more sample points to fully understand that space.

4.1. Geometrical Interpretation.

- If you think about the output vector without the noise, $X\mathbf{w}$, it has to belong in the same subspace as the input X. After all, the output is a function of X, $X\mathbf{w}$.



- *What happens when the input is corroborated with noise?* For example, instrumentation measuring errors, electrical noise from power source, thermal vibrations of atoms in conductors, unwanted signals, etc.
  - The output signal is now $y = X\mathbf{w} + \mathbf{e}$ will lay outside the space spanned by X unless the error (or residuals) $e = \mathbf{y} - X\mathbf{w}$ also belongs in the space spanned by X.
  - If you look carefully, by imposing the L-2 norm as our cost function, $E[\mathbf{e}^T\mathbf{e}] = \left\|\mathbf{e}^T\mathbf{e}\right\|_2^2$, it implies that we are performing a projection of $\mathbf{y} = X\mathbf{w} + \mathbf{e}$ onto the space spanned by X. In particular, our chosen cost function MSE is characterized by the property that the vector of error $\mathbf{e} = \mathbf{y} - X\mathbf{w}$ is **orthogonal** to the space spanned by the input X.

5. For next class...

- To prepare for next class, please read chapters 6.1-6.4 of the recommended textbook.
  - S. Theodoridis and K. Koutroumbas, Pattern Recognition. Academic Press: Cambridge, MA, 2009.