EE269
Signal Processing for Machine Learning

Lecture 4

Instructor : Mert Pilanci

Stanford University

Jan 9 2019

Recap: Changing the basis to Discrete Fourier

$$x[n] \overset{DFT}{\longleftrightarrow} X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}nk}$$

$$x[n] = \delta[n - n_0] \overset{DFT}{\longleftrightarrow} X[k] = e^{-j\frac{2\pi}{N}n_0 k}$$

$$x[n] = 1 \overset{DFT}{\longleftrightarrow} X[k] = N\delta[k]$$

$$x[n] = e^{j\frac{2\pi}{N}k_0 n} \overset{DFT}{\longleftrightarrow} X[k] = N\delta[k - k_0]$$

Distance based signal classification

Binary signal classification: training set
$x_1[n], x_2[n], \ldots, x_m[n]$, and labels $y_1, y_2, \ldots y_m \in \{0, 1\}^m$
Given unlabeled $x[n]$ find its label $y \in \{0, +1\}$

Distance based signal classification

Binary signal classification: training set
$x_1[n], x_2[n], \ldots, x_m[n]$, and labels $y_1, y_2, \ldots y_m \in \{0, 1\}^m$
Given unlabeled $x[n]$ find its label $y \in \{0, +1\}$

- Nearest Neighbor (NN) classifier
  Assign $x[n]$ the same label as the *closest* training signal

Distance based signal classification

Binary signal classification: training set
$x_1[n], x_2[n], \ldots, x_m[n]$, and labels $y_1, y_2, \ldots y_m \in \{0,1\}^m$
Given unlabeled $x[n]$ find its label $y \in \{0,+1\}$

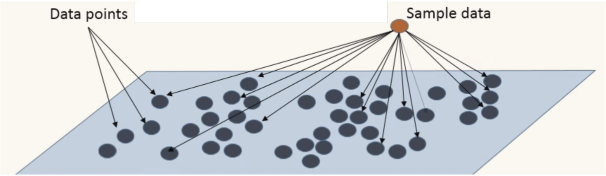
- Nearest Neighbor (NN) classifier
  Assign $x[n]$ the same label as the *closest* training signal

Example: Euclidean norm ($\sqrt{\text{signal energy}}$)

$$\min_{j=1,\ldots,m} \sqrt{\sum_{n=0}^{N-1} \left(x[n] - x_j[n]\right)^2}$$

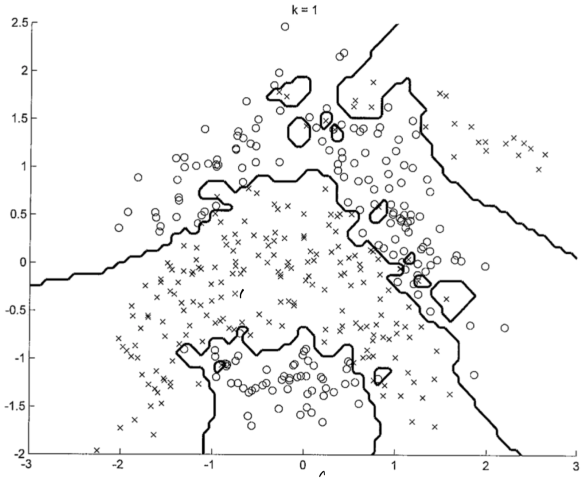
Other norms, metrics

Nearest Neighbor classifier



(Slide credit: R. Gunarathna)

- Nearest Neighbor rule defines a partitioning in the signal space Voronoi cells


k = 1
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-3
-2
-1
0
1
2
3

k-Nearest Neighbors

- Assign a label to $x[n]$ by taking a majority vote over the k training signals closest to $x[n]$.

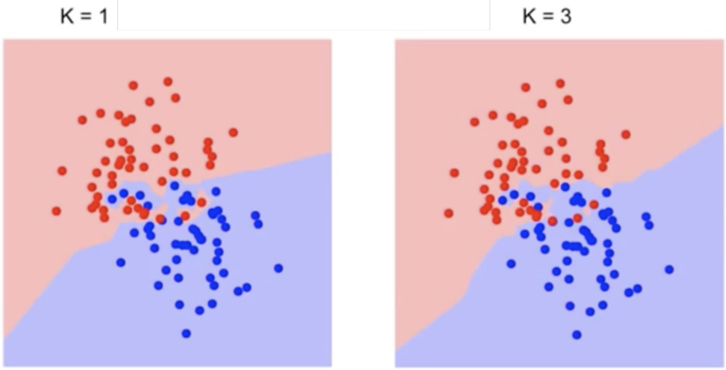
  - $k = 1$ is Nearest Neighbors
  - $k = m$ will always predict the majority class

k-Nearest Neighbors

- Assign a label to $x[n]$ by taking a majority vote over the k training signals closest to $x[n]$.
- Works for multi-class: $y_1, ..., y_m \in \{0, 1, 2, ..., L-1\}$

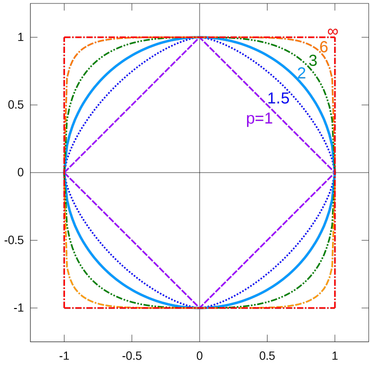
k-Nearest Neighbors

- Assign a label to $x[n]$ by taking a majority vote over the k training signals closest to $x[n]$.
- Works for multi-class: $y_1, ..., y_m \in \{0, 1, 2, ..., L-1\}$
- As $m \to \infty$, achieves at most $2\times$ minimum error possible for two classes



How to choose a distance function ?

- p-norm: $||x||_p := \left(\sum_{n=0}^{N-1} \left|x[n]\right|^p\right)^{1/p}$ for $p \geq 1$
  - ℓ_1-norm $(p = 1)$ $||x||_1 = \sum_{n=0}^{N-1} |x[n]|$
  - ℓ_2-norm $(p = 2)$ $||x||_2 = \sqrt{\sum_{n=0}^{N-1} x[n]^2}$



How to choose a distance function ?

- p-norm: $||x||_p := \left(\sum_{n=0}^{N-1} \left|x[n]\right|^p\right)^{1/p}$ for $p \geq 1$
  - ℓ_1-norm $(p = 1)$ $||x||_1 = \sum_{n=0}^{N-1} |x[n]|$
  - ℓ_2-norm $(p = 2)$ $||x||_2 = \sqrt{\sum_{n=0}^{N-1} |x[n]|^2}$
- Example:

$x_1[n] = [1, -1, 1, -1, ..., 1]$ length 100
$x_2[n] = [100, 0, 0, 0, ..., 0]$ length 100

$||x_1 - 0||_1 = 1 + 1 + ... + 1 = 100$
$||x_2 - 0||_1 = 100 + 0 + 0... + 0 = 100$

How to choose a distance function ?

- p-norm: $||x||_p := \left(\sum_{n=0}^{N-1} |x[n]|^p\right)^{1/p}$ for $p \geq 1$
  - ℓ_1-norm $(p = 1)$ $||x||_1 = \sum_{n=0}^{N-1} |x[n]|$
  - ℓ_2-norm $(p = 2)$ $||x||_2 = \sqrt{\sum_{n=0}^{N-1} |x[n]|^2}$
- Example:

$$x_1[n] = [1, -1, 1, -1, ..., 1] \quad \text{length } 100$$
$$x_2[n] = [100, 0, 0, 0, ..., 0] \quad \text{length } 100$$

$$||x_1 - 0||_1 = 1 + 1 + ... + 1 = 100$$
$$||x_2 - 0||_1 = 100 + 0 + 0... + 0 = 100$$

$$||x_1 - 0||_2 = \sqrt{1 + 1 + ... + 1} = \sqrt{100} = 10$$
$$||x_2 - 0||_2 = \sqrt{100^2 + 0 + 0...0} = 100$$

- ℓ_2-norm is invariant to rotation
  - $||Rx||_2^2 = \langle Rx, Rx \rangle$
  - $x[n] \overset{DFT}{\longleftrightarrow} X[k] \implies ||x[n]||_2 = ||X[k]||_2$
- ℓ_1-norm is **not invariant** to rotation

Other distance metrics

- ℓ_∞-norm $\|x\|_\infty = \max_{n \in \{0,\dots,N-1\}} |x[n]|$
- ℓ_0-norm $\|x\|_0 = \sum_{n=0}^{N-1} 1_{x[n] \neq 0} =$ number of nonzero components of $x[n]$

How to choose k in k-Nearest Neighbors

- $f_k(x)$: classification output
  - validation set $x_{m+1}[n], ..., x_{m+r}[n]$ with known labels

$$R(f_k) := \frac{1}{r} \sum_{m+1}^{m+r} 1_{f_k(x_i) \neq y_i}$$

  - minimize over k

How to choose k in k-Nearest Neighbors

- $f_k(x)$: classification output
  - validation set $x_{m+1}[n], ..., x_{m+r}[n]$ with known labels

$$R(f_k) := \frac{1}{r} \sum_{m+1}^{m+r} 1_{f_k(x_i) \neq y_i}$$

  - minimize over k
- hold out method
- q-fold cross-validation

How to choose k in k-Nearest Neighbors

- $f_k(x)$: classification output
  - validation set $x_{m+1}[n], ..., x_{m+r}[n]$ with known labels

$$R(f_k) := \frac{1}{r} \sum_{m+1}^{m+r} 1_{f_k(x_i) \neq y_i}$$

  - minimize over k

How to choose k in k-Nearest Neighbors

- $f_k(x)$: classification output
  - validation set $x_{m+1}[n], ..., x_{m+r}[n]$ with known labels

$$R(f_k) := \frac{1}{r} \sum_{m+1}^{m+r} 1_{f_k(x_i) \neq y_i}$$

  - minimize over k
- hold out method
- q-fold cross-validation

How to choose k in k-Nearest Neighbors

- $f_k(x)$: classification output
  - validation set $x_{m+1}[n], ..., x_{m+r}[n]$ with known labels

$$R(f_k) := \frac{1}{r}\sum_{m+1}^{m+r} 1_{f_k(x_i)\neq y_i}$$

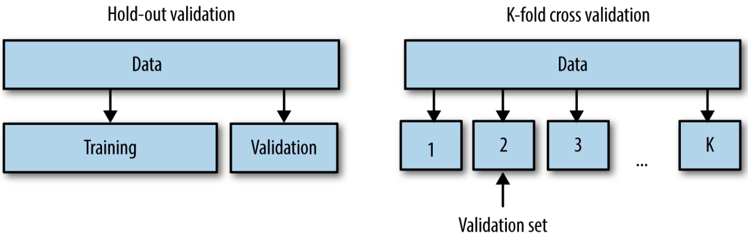
  - minimize over k

How to choose k in k-Nearest Neighbors

- $f_k(x)$: classification output
  - validation set $x_{m+1}[n], \ldots, x_{m+r}[n]$ with known labels

$$R(f_k) := \frac{1}{r} \sum_{m+1}^{m+r} 1_{f_k(x_i) \neq y_i}$$

  - minimize over k
- hold out method
- q-fold cross-validation

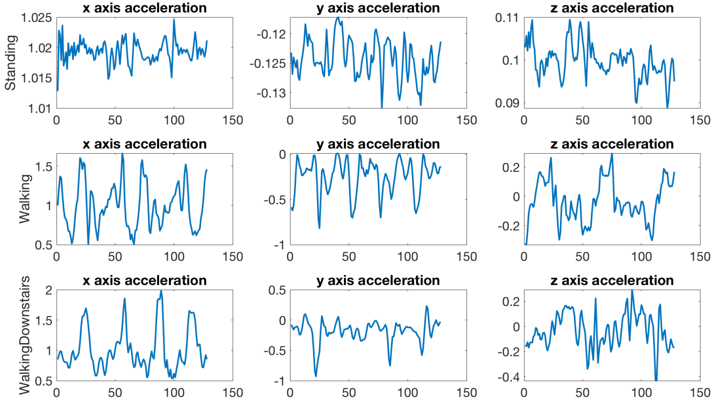


Application

- Human Activity Recognition Using Smartphones Dataset (Reyes-Ortiz et al, 2012)

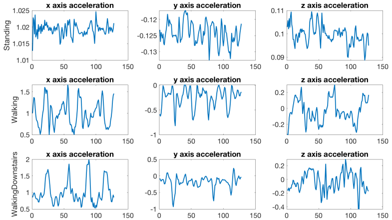
Application

- Human Activity Recognition Using Smartphones Dataset (Reyes-Ortiz et al, 2012)
- Time domain training signals $x_1[n], x_2[n], ...x_m[n]$



Problems with time domain representation

- Human Activity Recognition Using Smartphones Dataset (Reyes-Ortiz et al, 2012)
- Time domain training signals $x_1[n], x_2[n], ...x_m[n]$

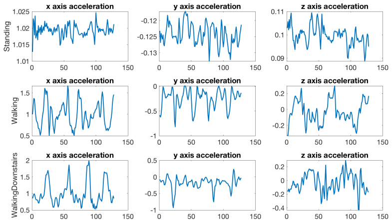


Discrete Fourier Transform basis

- Human Activity Recognition Using Smartphones Dataset (Reyes-Ortiz et al, 2012)
- Time domain training signals $x_1[n], x_2[n], \ldots x_m[n]$

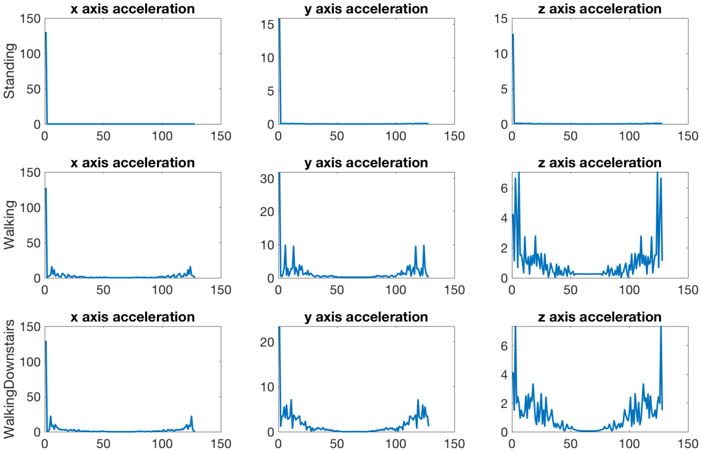
$$x[n] = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X[k] e^{-j\frac{2\pi}{N}nk} \; .$$

$$x[(n - n_0) \mod N] \quad \overset{DFT}{\longleftrightarrow} \quad e^{-j\frac{2\pi}{N}n_0 k} X[k]$$

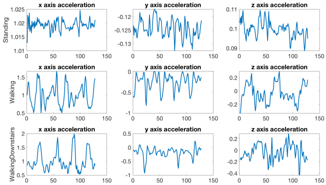


Application

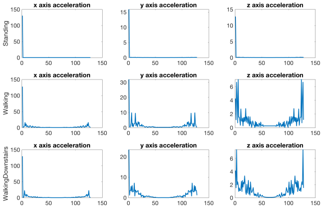
- Human Activity Recognition Using Smartphones Data Set (Reyes-Ortiz et al, 2012)
- Compote DFT of the training signals $X_1[k], X_2[k], \ldots X_m[k]$ DFT Magnitude $|X_1[k]|, |X_2[k]|, \ldots |X_m[k]|$



Results: training set: 7724 signals, test set: 2575 signals

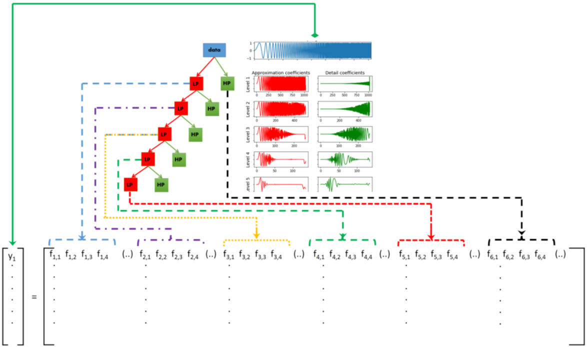


3-Nearest Neighbors, ℓ_2-norm distance on $x[n]$. **Accuracy** : 0.77



3-Nearest Neighbors, ℓ_2-norm distance on $|X[k]|$. **Accuracy** : 0.85

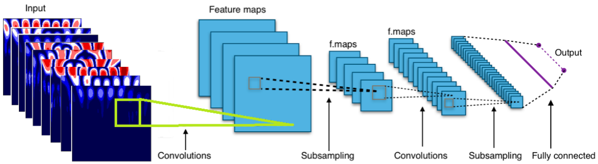
Human Activity Recognition dataset



Wavelet Transform. **Accuracy** : 0.91

(Slide credit: Taspinar)

Human Activity Recognition dataset



Wavelet Transform + Deep Learning. **Accuracy** : 0.96

(Slide credit: Taspinar)

Ingredients for a Hilbert space

Ingredient 1. We start with a collection of vectors (signals) and some scalars (real or complex numbers) and closure under the following two operations:

(a) Vector addition:

if x, y are in the vector space, $x + y$ is in the vector space.

(b) Scalar multiplication:

for a scalar α and a vector x in the vector space,

αx is in the vector space.

Ingredient 2. Next we need an *inner product*. A function $\langle x, y\rangle$ of two vectors x, y is an inner product if it satisfies the following properties. Note this function outputs a complex number.

(a) The self-inner product is positive and zero if and only if $x = 0$.

$$\langle x, x\rangle \geq 0 \text{ and } \langle x, x\rangle = 0 \Leftrightarrow x = 0$$

(b) When we scale one of the vectors, the inner product scales as follows

$$\langle x, \alpha y\rangle = \alpha\langle x, y\rangle \text{ and } \langle \alpha x, y\rangle = \alpha^*\langle x, y\rangle$$

(c) The inner product is distributive:

$$\langle x + y, z\rangle = \langle x, z\rangle + \langle y, z\rangle$$

(d) And finally the inner product is commutative with conjugation:

$$\langle x, y\rangle = \langle y, x\rangle^*$$

Ingredient 3. Finally, we need the property of *completeness*. Completeness means that if you take a limit of a sequence of vectors, and the limit exists, then the limit is also in the vector space.

- For a space to be complete, every Cauchy sequence converges.

Let's take a basic example: $\mathbb{R}^2$ is a Hilbert space when we use real-valued scalars.

Ingredient 1. Is $\mathbb{R}^2$ a vector space? Yes. We can add and scale vectors and they are still in $\mathbb{R}^2$.

Ingredient 2. We said the inner product for $\mathbb{R}^2$ is:

$$\langle x, y \rangle = x_0 y_0 + x_1 y_1 \ .$$

Does this satisfy our rules for the inner product?

1. $\langle x, x \rangle = x_0^2 + x_1^2$ and this is always positive or zero, and it is only zero if $x_0 = 0$ and $x_1 = 0$, i.e., if $x = zero$.

2 Recalling we are using real-valued scalars, and that the complex conjugate of a real number is itself, we can show this property:

$$\langle x, \alpha y\rangle = x_0(\alpha y_0) + x_1(\alpha y_1) = \alpha\left(x_0 y_0 + x_1 y_1\right) = \alpha\langle x, y\rangle$$

and

$$\langle \alpha x, y\rangle = (\alpha x_0)y_0 + (\alpha x_1)y_1 = \alpha(x_0 y_0 + x_1 y_1) = \alpha\langle x, y\rangle\ .$$

3 Distributivity:

$$\begin{aligned}\langle x + y, z\rangle &= (x_0 + y_0)z_0 + (x_1 + y_1)z_1 \\ &= x_0 z_0 + y_0 z_0 + x_1 z_1 + y_1 z_1 \\ &= (x_0 z_0 + x_1 z_1) + (y_0 z_0 + y_1 z_1) = \langle x, z\rangle + \langle y, z\rangle\ .\end{aligned}$$

4 And again recalling we are using real-valued scalars, and that the complex conjugate of a real number is itself, we can show commutativity:

$$\langle x, y\rangle = x_0 y_0 + x_1 y_1 = y_0 x_0 + y_1 x_1 = \langle y, x\rangle$$

1 $\mathbb{R}^N$ using scalar field $\mathbb{R}$ and inner product $\sum_{n=0}^{N-1} x^*[n]y[n]$ is a Hilbert space

1 $\mathbb{R}^N$ using scalar field $\mathbb{R}$ and inner product $\sum_{n=0}^{N-1} x^*[n]y[n]$ is a Hilbert space

2 $\mathbb{C}^N$ using scalar field $\mathbb{C}$ is a Hilbert space

1. $\mathbb{R}^N$ using scalar field $\mathbb{R}$ and inner product $\sum_{n=0}^{N-1} x^*[n]y[n]$ is a Hilbert space
2. $\mathbb{C}^N$ using scalar field $\mathbb{C}$ is a Hilbert space
3. $L_2[-1,1]$: Continuous-time square-integrable signals on the interval $[-1,1]$, i.e.,

$$\int_{-1}^{1} |x(t)|^2 dt < \infty \ .$$

and the inner product $\langle x, y \rangle = \int_{-1}^{1} x^*(t)y(t)dt$ is a Hilbert space.

1. $\mathbb{R}^N$ using scalar field $\mathbb{R}$ and inner product $\sum_{n=0}^{N-1} x^*[n]y[n]$ is a Hilbert space
2. $\mathbb{C}^N$ using scalar field $\mathbb{C}$ is a Hilbert space
3. $L_2[-1,1]$: Continuous-time square-integrable signals on the interval $[-1,1]$, i.e.,

$$\int_{-1}^{1} |x(t)|^2 dt < \infty \ .$$

   and the inner product $\langle x, y \rangle = \int_{-1}^{1} x^*(t)y(t)dt$ is a Hilbert space.
4. $\ell_2(\mathbb{Z})$: Infinite-length discrete time square summable complex-valued sequences and $\sum_{k=-\infty}^{\infty} |x[k]|^2 < \infty$, using scalar field $\mathbb{C}$ and inner product $\sum_{k=-\infty}^{\infty} x[k]^* y[k]$, using scalar field $\mathbb{C}$ is a Hilbert space.

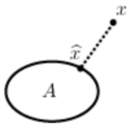
Hilbert space geometry

Definition: A set $\mathcal{A}$ is convex iff $\lambda x + (1 - \lambda) y \in \mathcal{A}$ for all points $x, y \in \mathcal{A}$.

Hilbert space geometry

Definition: A set $\mathcal{A}$ is convex iff $\lambda x + (1-\lambda)y \in \mathcal{A}$ for all points $x, y \in \mathcal{A}$.

Theorem (The Fundamental Theorem of Approximation) : Let $\mathcal{A}$ be a nonempty, closed convex set in a Hilbert space H. For any $x \in H$ there is a **unique** point in $\mathcal{A}$ that is closest to x, i.e., x has a unique **best approximation** in $\mathcal{A}$.



As a consequence, there is a unique minimum norm (minimum energy) point.

Applications: Signal denoising, Kernel regression, classification...

Hilbert (vector) space structure

Subspace: A subspace of a vector space V is a set of vectors $P \subset V$ that are closed under addition and scalar multiplication:

For all $x, y \in P$ and for all scalars $\alpha, \beta, \quad \alpha x + \beta y \in P$.

Example:

Hilbert (vector) space structure

Span: Given a set of M vectors $\{x^{(0)}[n], \ldots, x^{(M-1)}[n]\} = W \subset V$, the *span* of the vectors is

$$\text{span}(W) = \left\{ x : x = \sum_{m=0}^{M-1} \alpha_m x^{(m)}, \quad \text{for all scalars } \alpha \right\} .$$

Hilbert (vector) space structure

Span: Given a set of M vectors $\{x^{(0)}[n], \ldots, x^{(M-1)}[n]\} = W \subset V$, the *span* of the vectors is

$$\text{span}(W) = \left\{ x : x = \sum_{m=0}^{M-1} \alpha_m x^{(m)}, \quad \text{for all scalars } \alpha \right\} .$$

Examples:

In $\mathbb{R}^N$, the span of $\delta_0[n], \cdots, \delta_{N-1}[n]$

In $\mathbb{C}^N$, the span of $m < N$ vectors from the Fourier basis

Hilbert (vector) space structure

- **Dimension:** The dimension of a space is the number of linearly independent vectors required in order to span the whole space.

  Since length-N signals require N different δ signals to span $\mathbb{R}^N$, then by definition $\mathbb{R}^N$ is an N-*dimensional* space.

- **Parseval's Identity:** For any orthonormal basis for length-N signals, $y^{(0)}[n], \ldots, y^{(N-1)}[n]$, we have the following equality:

$$\|x\|^2 = \sum_{k=0}^{N-1} \left| \langle y^{(k)}[n], x[n] \rangle \right|^2 .$$

- **Parseval's Identity:** For any orthonormal basis for length-N signals, $y^{(0)}[n], \ldots, y^{(N-1)}[n]$, we have the following equality:

$$\|x\|^2 = \sum_{k=0}^{N-1} \left| \langle y^{(k)}[n], x[n] \rangle \right|^2 .$$

- The Fourier basis is orthonormal, this implies the classical Parseval's identity

$$\|x\| = \sum_{n=0}^{N-1} |x[n]|^2 = \sum_{k=0}^{N-1} |X[k]|^2 = \sum_{k=0}^{N-1} \left| \langle y^{(k)}[n], x[n] \rangle \right|^2$$

for $y^{(k)}[n] = w_k[n] = \frac{1}{\sqrt{N}} e^{-j\frac{2\pi}{N}nk}$.

Null space

Null space of a complex valued matrix $A \in \mathbb{C}^{M \times N}$

$$N(A) = \{x \in \mathbb{C}^N \ : \ Ax = 0\}$$

Null space

Null space of a complex valued matrix $A \in \mathbb{C}^{M \times N}$

$$N(A) = \{x \in \mathbb{C}^N \ : \ Ax = 0\}$$

$$A = \begin{bmatrix} a_1 & a_2 & \dots & a_N \end{bmatrix}$$

$a_1, a_2, \dots, a_N$ are linearly independent iff $N(A) = \{0\}$.