Preliminaries

Deep Learning (ENGG*6600*01)

School of Engineering,
University of Guelph, ON, Canada

Course Instructor: Benyamin Ghojogh

Summer 2023

Dataset, Learning Model, and Learning Tasks

Dataset

- Consider the measurement of a quantity. This quantity can be:
  - personal health data, including blood pressure, blood sugar, and blood fat,
  - images from a specific scene but taken from different perspectives,
  - images from several categories of animals, such as cat, dog, frog, etc.,
  - medical images, such as digital pathology image patches, including both healthy and tumorous tissues,
  - or any other measured signal.
- The quantity can be multidimensional, i.e., a set of values, and therefore, every quantity can be considered a multidimensional data point in a Euclidean space.
- Let the dimensionality of this space be d, meaning that every quantity is a d-dimensional vector, or data point, in $\mathbb{R}^d$. The set of d values for the quantity can be called **features** of the quantity.
- Multiple measurements of a quantity can exist, each of which is a d-dimensional data point. Therefore, there will be a set of d-dimensional data points, called a **dataset**.
- For example, the quantity can be an image, whose features are its pixels. The dataset can be a set of images from a specific scene but with different perspectives and angles.

Learning Model

x X

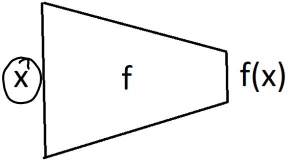
$\boldsymbol{x}$ X

$\boldsymbol{X}$

- Consider a dataset of n data points $\{\boldsymbol{x}_i \in \mathbb{R}^d\}_{i=1}^{n}$, each of which is a d-dimensional vector in the d-dimensional Euclidean space. We can put these vectors column-wise in a matrix $\boldsymbol{X} \in \mathbb{R}^{d \times n}$.
- Consider a learning model f which is a map from data space to some output space:

$$\begin{aligned} f &: \mathbb{R}^d \to \mathbb{R}^p, \\ f &: \boldsymbol{x} \mapsto f(\boldsymbol{x}). \end{aligned} \tag{1}$$

- Usually, $p \leq d$ but not necessarily.



Learning Tasks

- Learning model is like a new-born baby (it first knows nothing and we should teach it)
- **Supervised**:
  - **Regression**: example of learning EMG signals for artificial leg, example of weather prediction

$$f(\boldsymbol{x}) \in \underline{[0,1]^p} \text{ or } f(\boldsymbol{x}) \underline{\in \mathbb{R}}.$$

  - **Classification**: example of teaching apples and cucumbers to a baby

$$\underline{f(\boldsymbol{x})} \in \underline{\{\ell_1, \ell_2, \dots, \ell_m\}}. \leftarrow$$

- **Unsupervised**:
  - **Clustering**: example of clustering apples and cucumbers by a baby

$$f(\boldsymbol{x}) \in \underline{\{\overset{\checkmark}{\ell_1}, \overset{\checkmark}{\ell_2}, \dots, \overset{\checkmark}{\ell_m}\}.} \leftarrow$$

- **Environment (world)**:
  - **Reinforcement learning**: example of teaching a dog

$$f(\boldsymbol{x}) = a \in \mathcal{A},$$

where $\mathcal{A}$ is the set of possible actions.

Learning Tasks

- **Dimensionality reduction (manifold learning)**: learning an embedding space

$$\underline{f : \mathbb{R}^d \to \mathbb{R}^p.}$$

where $p \leq d$, and usually $p \ll d$.

  - **Unsupervised** dimensionality reduction: embedding similar patterns close to each other
  - **Supervised** dimensionality reduction: Decreasing the intra-class variances and increase the inter-class variances
  - For more information on dimensionality reduction , you can see our textbook [1]: `https://link.springer.com/book/10.1007/978-3-031-10602-6`

- **Numerosity processing**:
  - **Outlier (anomaly) detection**: detecting outliers in data
  - **Prototype selection** [2]: selecting important instances
  - **Prototype generation** [3]: selecting and generating important instances
  - For more information on dimensionality reduction and numerosity processing, you can see my PhD thesis: [4]: `https://uwspace.uwaterloo.ca/handle/10012/16813`

Other Fields of AI

Some other fields of Artificial Intelligence (AI):

- Soft computing:
  - Fuzzy logic and fuzzy control
  - Metaheuristic optimization and intelligent search
- Biological-inspired (third generation) neural networks - relation to neuroscience and cognitive science - Example: spiking neural network
- Feature engineering (pre-processing):
  - Feature selection
  - Feature extraction (dimensionality reduction)
- Application of AI in various fields of science and technology

Linear Projection

Column Space

- Consider p basis vectors. We can define a p-dimensional Euclidean space by these p basis vectors. For example, two vectors define a plane and three vectors define a 3D space.
- In terminology: The p basis vectors **span** the p-dimensional Euclidean space. Or the p-dimensional Euclidean space is **spanned by** the p basis vectors.
- Consider the p vectors $\{\boldsymbol{u}_1, \dots, \boldsymbol{u}_p\}$. These vectors can be stacked columnwise in matrix $\boldsymbol{U} = [\boldsymbol{u}_1, \dots, \boldsymbol{u}_p] \in \mathbb{R}^{d \times p}$.
- The space spanned by the columns of matrix $\boldsymbol{U}$ is called the column space of matrix $\boldsymbol{U}$, denoted by $\mathbb{C}\text{ol}(\boldsymbol{U})$:

$$\boxed{\mathbb{C}\text{ol}(\boldsymbol{U}) := \textbf{span}\{\boldsymbol{u}_1, \dots, \boldsymbol{u}_p\}.} \tag{2}$$

- In other words, the space whose bases are the columns of matrix $\boldsymbol{U}$ is called the column space of matrix $\boldsymbol{U}$.

Linear Projection

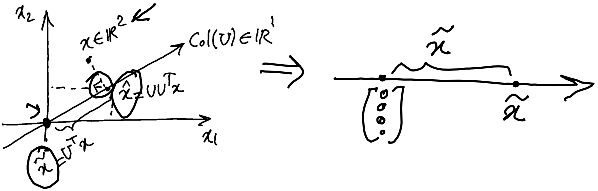
- Assume there is a data point $\boldsymbol{x} \in \mathbb{R}^d$. The aim is to project this data point onto the vector space spanned by p vectors $\{\boldsymbol{u}_1, \dots, \boldsymbol{u}_p\}$, where each vector is d-dimensional and usually $p \ll d$.
- These vectors can be stacked columnwise in matrix $\boldsymbol{U} = [\boldsymbol{u}_1, \dots, \boldsymbol{u}_p] \in \overbrace{\mathbb{R}^{d\times p}}$. In other words, the goal is to project $\boldsymbol{x}$ onto the column space of $\boldsymbol{U}$, denoted by $\mathbb{C}\mathrm{ol}(\boldsymbol{U})$.
- As $p < d$, this projection is projection onto a **subspace** because we are projecting from d-dimensional space onto a lower dimensional space.
- The **projection** of $\boldsymbol{x} \in \mathbb{R}^d$ onto $\mathbb{C}\mathrm{ol}(\boldsymbol{U}) \in \mathbb{R}^p$ is:

$$\boxed{\mathbb{R}^p} \ni \widetilde{\boldsymbol{x}} := \underbrace{\overset{p\times d}{\boldsymbol{U}^\top}\, \overset{d\times 1}{\boldsymbol{x}}}_{p\times 1}. \tag{3}$$

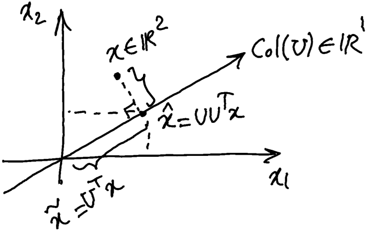
- The **reconstruction** of $\widetilde{\boldsymbol{x}} \in \mathbb{R}^p$ in the d-dimensional space is:

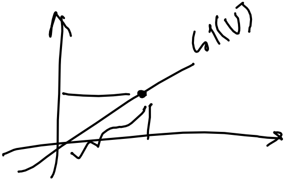
$$\boxed{\mathbb{R}^d} \ni \widehat{\boldsymbol{x}} := \overbrace{\underset{d\times p}{\boldsymbol{U}}\,\underset{p\times 1}{\widetilde{\boldsymbol{x}}}}^{d\times 1} = \boldsymbol{U}\underbrace{\boldsymbol{U}^\top \boldsymbol{x}}. \tag{4}$$

- Reconstruction is its representation in $\mathbb{R}^d$ again, but after projection.



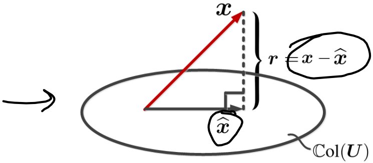
Linear Projection





- **Reconstruction error**: There is a residual/error between the original data $\boldsymbol{x}$ and its reconstruction (if the data point is already in the column space, this residual is zero):

$$\boxed{\boldsymbol{r} = \boldsymbol{x} - \widehat{\boldsymbol{x}}} = \underline{\boldsymbol{x} - \boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{x}}. \tag{5}$$


$\boldsymbol{x}$
$r = \boldsymbol{x} - \widehat{\boldsymbol{x}}$
$\widehat{\boldsymbol{x}}$
$\mathbb{C}\mathrm{ol}(\boldsymbol{U})$

Norm

Inner product

Definition (Inner product of vectors)

Consider two vectors $\underline{\boldsymbol{x} = [x_1, \dots, x_d]^{\boxed{\top}} \in \mathbb{R}^d}$ and $\underline{\boldsymbol{y} = [y_1, \dots, y_d]^\top \in \mathbb{R}^d}$. Their **inner product**, also called **dot product**, is:

$x_1 y_1 + x_2 y_2 + \cdots + x_d y_d$

$$\underbrace{\langle \boldsymbol{x}, \boldsymbol{y} \rangle} = \boxed{\boldsymbol{x}^\top \boldsymbol{y}} = \underbrace{\sum_{i=1}^{d} x_i \, y_i}_{\star}.$$

Definition (Inner product of matrices)

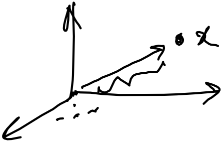
We also have inner product between matrices $\underline{\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}^{d_1 \times d_2}}$. Let $\boldsymbol{X}_{ij}$ denote the $\underline{(i,j)\text{-th element of matrix } \boldsymbol{X}}$. The inner product of $\boldsymbol{X}$ and $\boldsymbol{Y}$ is:

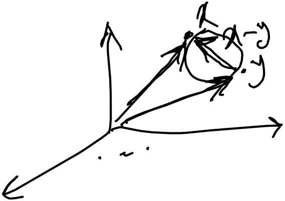
$$\underbrace{\langle \boldsymbol{X}, \boldsymbol{Y} \rangle} = \underbrace{\boxed{\mathbf{tr}}(\overline{\boldsymbol{X}^\top \boldsymbol{Y}})} = \underbrace{\sum_{i=1}^{d_1} \sum_{j=1}^{d_2} \boldsymbol{X}_{i,j} \, \boldsymbol{Y}_{i,j}}_{\star},$$

where **tr**(.) denotes the trace of matrix.

Norm

$\|x\|$



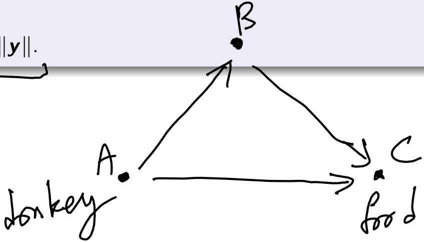


$\underbrace{\|x-y\|}$

Definition (Norm)

A function $\underline{\|\cdot\|} : \underline{\mathbb{R}^d \to \mathbb{R}}$, $\underline{\|\cdot\| : \boldsymbol{x} \mapsto \|\boldsymbol{x}\|}$ is a **norm** if it satisfies:

1. ★ $\boxed{\|\boldsymbol{x}\| \geq 0, \forall \boldsymbol{x}}$
2. ★ $\boxed{\|a\boldsymbol{x}\| = |a|\,\|\boldsymbol{x}\|,}\ \forall \underline{\boldsymbol{x}}$ and all scalars $\underline{a}$
3. ★ $\underline{\|\boldsymbol{x}\| = 0}$ if and only if $\underline{\boldsymbol{x} = 0}$
4. ★ Triangle inequality: $\underbrace{\|\boldsymbol{x}+\boldsymbol{y}\| \leq \|\boldsymbol{x}\| + \|\boldsymbol{y}\|.}$



Important norms for vectors

Some important norms for a vector $\boldsymbol{x} = [x_1, \dots, x_d]^\top$ are as follows.

- The ℓ_p **norm** is:

$$\boxed{\|\boldsymbol{x}\|_p := \big(|x_1|^p + \cdots + |x_d|^p\big)^{1/p},} = \sqrt[p]{|x_1|^p + \dots + |x_d|^p}$$

  where $p \geq 1$ and $|.|$ denotes the absolute value.

- Two well-known ℓ_p norms are ℓ_1 **norm** and ℓ_2 **norm** (also called the **Euclidean norm**) with $p = 1$ and $p = 2$, respectively:

$$★ \quad \|\boldsymbol{x}\|_1 := |x_1| + \cdots + |x_d| = \sum_{i=1}^{d} |x_i|,$$

$$★ ★ \quad \|\boldsymbol{x}\|_2 := \sqrt{x_1^2 + \cdots + x_d^2} = \sqrt{\sum_{i=1}^{d} x_i^2},$$

Important norms for matrices

Some important norms for a matrix $\boldsymbol{X} \in \mathbb{R}^{d_1 \times d_2}$ are as follows.

- The formulation of the **Frobenius norm** for a matrix is similar to the formulation of ℓ_2 norm for a vector:

$$\bigstar \quad \|\boldsymbol{X}\|_F := \sqrt{\underbrace{\sum_{i=1}^{d_1} \sum_{j=1}^{d_2}} \boldsymbol{X}_{i,j}^2},$$

where $\boldsymbol{X}_{ij}$ denotes the (i, j)-th element of $\boldsymbol{X}$.

Quadratic forms using norms

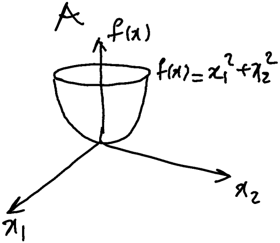
$$\|(x-y)\|_2^2 = (x-y)^\top (x-y)$$

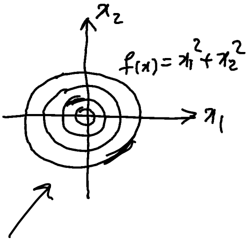
For $\boldsymbol{x} \in \mathbb{R}^d$ and $\boldsymbol{X} \in \mathbb{R}^{d_1 \times d_2}$, we have:

$$\bigstar \quad \|\boldsymbol{x}\|_2^2 = \underbrace{\boldsymbol{x}^\top \boldsymbol{x}} = \langle \boldsymbol{x}, \boldsymbol{x} \rangle = \underbrace{\sum_{i=1}^{d} x_i^2},$$

$$\bigstar \quad \|\boldsymbol{X}\|_F^2 = \underbrace{\mathbf{tr}(\boldsymbol{X}^\top \boldsymbol{X})} = \langle \boldsymbol{X}, \boldsymbol{X} \rangle = \underbrace{\sum_{i=1}^{d_1} \sum_{j=1}^{d_2} \boldsymbol{X}_{i,j}^2},$$

which are convex and in quadratic forms.





Preliminaries on Derivatives

Dimensionality of derivative

$$\frac{\partial \, tr(X^T X)}{\partial X} = 2X$$

- Consider a function $f : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$, $f : \boldsymbol{x} \mapsto f(\boldsymbol{x})$.
- Derivative of function $f(\boldsymbol{x}) \in \mathbb{R}^{d_2}$ with respect to (w.r.t.) $\boldsymbol{x} \in \mathbb{R}^{d_1}$ has dimensionality $(d_1 \times d_2)$.
- This is because tweaking every element of $\boldsymbol{x} \in \mathbb{R}^{d_1}$ can change every element of $f(\boldsymbol{x}) \in \mathbb{R}^{d_2}$. The (i, j)-th element of the $(d_1 \times d_2)$-dimensional derivative states the amount of change in the j-th element of $f(\boldsymbol{x})$ resulted by changing the i-th element of $\boldsymbol{x}$.

Examples

- The derivative of a scalar w.r.t. a scalar is a scalar.
- The derivative of a scalar w.r.t. a vector is a vector.
- The derivative of a scalar w.r.t. a matrix is a matrix.
- The derivative of a vector w.r.t. a vector is a matrix.
- The derivative of a vector w.r.t. a matrix is a rank-3 tensor.
- The derivative of a matrix w.r.t. a matrix is a rank-4 tensor.

Dimensionality of derivative

In more details:

- If the function is $f : \mathbb{R} \to \mathbb{R}, f : x \mapsto f(x)$, the derivative $(\partial f(x)/\partial x) \in \mathbb{R}$ is a scalar because changing the scalar $\boldsymbol{x}$ can change the scalar $f(\boldsymbol{x})$.
- If the function is $f : \mathbb{R}^d \to \mathbb{R}, f : \boldsymbol{x} \mapsto f(\boldsymbol{x})$, the derivative $(\partial f(\boldsymbol{x})/\partial \boldsymbol{x}) \in \mathbb{R}^d$ is a vector because changing every element of the vector $\boldsymbol{x}$ can change the scalar $f(\boldsymbol{x})$.
- If the function is $f : \mathbb{R}^{d_1 \times d_2} \to \mathbb{R}, f : \boldsymbol{X} \mapsto f(\boldsymbol{X})$, the derivative $(\partial f(\boldsymbol{X})/\partial \boldsymbol{X}) \in \mathbb{R}^{d_1 \times d_2}$ is a matrix because changing every element of the matrix $\boldsymbol{X}$ can change the scalar $f(\boldsymbol{X})$.
- If the function is $f : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}, f : \boldsymbol{x} \mapsto f(\boldsymbol{x})$, the derivative $(\partial f(\boldsymbol{x})/\partial \boldsymbol{x}) \in \mathbb{R}^{d_1 \times d_2}$ is a matrix because changing every element of the vector $\boldsymbol{x}$ can change every element of the vector $f(\boldsymbol{x})$.
- If the function is $f : \mathbb{R}^{d_1 \times d_2} \to \mathbb{R}^{d_3}, f : \boldsymbol{X} \mapsto f(\boldsymbol{X})$, the derivative $(\partial f(\boldsymbol{X})/\partial \boldsymbol{X})$ is a $(d_1 \times d_2 \times d_3)$-dimensional tensor because changing every element of the matrix $\boldsymbol{X}$ can change every element of the vector $f(\boldsymbol{X})$.
- If the function is $f : \mathbb{R}^{d_1 \times d_2} \to \mathbb{R}^{d_3 \times d_4}, f : \boldsymbol{X} \mapsto f(\boldsymbol{X})$, the derivative $(\partial f(\boldsymbol{X})/\partial \boldsymbol{X})$ is a $(d_1 \times d_2 \times d_3 \times d_4)$-dimensional tensor because changing every element of the matrix $\boldsymbol{X}$ can change every element of the matrix $f(\boldsymbol{X})$.

Gradient and Hessian

Definition (Gradient)

Consider a function $\underline{f : \mathbb{R}^d \to \mathbb{R},\ f : \boldsymbol{x} \mapsto f(\boldsymbol{x})}$. In optimizing the function f, the derivative of function w.r.t. its variable $\boldsymbol{x}$ is called the **gradient**, denoted by:

$$\underline{\nabla f(\boldsymbol{x})} := \frac{\partial f(\boldsymbol{x})}{\partial \boxed{\boldsymbol{x}}} \in \boxed{\mathbb{R}^d.}$$

Definition (Hessian)

Consider a function $f : \mathbb{R}^d \to \mathbb{R}$, $f : \boldsymbol{x} \mapsto f(\boldsymbol{x})$. The second derivative of function w.r.t. to its derivative is called the **Hessian** matrix, denoted by:

$$\boldsymbol{B} = \nabla^2 f(\boldsymbol{x}) := \frac{\partial^2 f(\boldsymbol{x})}{\partial \boxed{\boldsymbol{x}^2}} \in \underline{\mathbb{R}^{d \times d}}.$$

The Hessian matrix is symmetric. If the function is convex, its Hessian matrix is positive semi-definite.

Chain rule

- When having composite functions (i.e., function of function), we use **chain rule** for derivative. Example:

$$f(x) = \sqrt{x^3 + x^2 - x + 10} = \sqrt{g(x)}, \quad g(x) = x^3 + x^2 - x + 10,$$
$$\frac{\partial f(x)}{\partial x} = \frac{\partial f(x)}{\partial g(x)} \times \frac{\partial g(x)}{\partial x} = \frac{1}{2\sqrt{g(x)}} \times (3x^2 + 2x - 1) = \frac{3x^2 + 2x - 1}{2\sqrt{x^3 + x^2 - x + 10}}$$

- The chain rule in matrix derivatives is usually stated right to left in matrix multiplications while transpose is used for matrices in multiplication.
- Let **vec**(.) denote vectorization of a $\mathbb{R}^{a \times b}$ matrix to a $\mathbb{R}^{ab}$ vector.
- Let $\mathbf{vec}^{-1}_{a \times b}(.)$ be de-vectorization of a $\mathbb{R}^{ab}$ vector to a $\mathbb{R}^{a \times b}$ matrix.

Optimization

Optimization

- Lagrangian:

$$\begin{aligned}\underset{\boldsymbol{x}}{\text{minimize}} \quad & \boxed{f(\boldsymbol{x})} \\ \text{subject to} \quad & \underline{y_i(\boldsymbol{x}) \leq 0,}\ i \in \{1, \dots, m_1\}, \\ & \underline{h_i(\boldsymbol{x}) = 0,}\ i \in \{1, \dots, m_2\}.\end{aligned}$$

$$\longrightarrow \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) := \boxed{f(\boldsymbol{x})} + \underbrace{\sum_{i=1}^{m_1} \lambda_i y_i(\boldsymbol{x})} + \underbrace{\sum_{i=1}^{m_2} \nu_i h_i(\boldsymbol{x})} = f(\boldsymbol{x}) + \underbrace{\boldsymbol{\lambda}^\top \boldsymbol{y}(\boldsymbol{x})} + \underbrace{\boldsymbol{\nu}^\top \boldsymbol{h}(\boldsymbol{x})}.$$

$$\bigstar \quad \frac{\partial \mathcal{L}}{\partial \boldsymbol{x}} \overset{\text{set}}{=} \boldsymbol{0} \implies \dots$$

- $\underline{\text{Unconstrained}}$ optimization:

$$\underset{\boldsymbol{x}}{\text{minimize}} \qquad f(\boldsymbol{x}),$$

$$\boldsymbol{x}^{(k+1)} := \boldsymbol{x}^{(k)} + (\Delta \boldsymbol{x})^{(k)},$$

$$\underline{\text{Gradient method:}}\ \ \underline{\boldsymbol{x}^{(k+1)} := \boldsymbol{x}^{(k)} \boxed{- \eta^{(k)} \nabla f(\boldsymbol{x}^{(k)}),}}$$

$$\bigstar\ \underline{\text{Newton's method:}}\ \ \boldsymbol{x}^{(k+1)} := \boldsymbol{x}^{(k)} - \eta^{(k)} \underbrace{\left(\nabla^2 f(\boldsymbol{x}^{(k)})\right)^{-1}} \underbrace{\nabla f(\boldsymbol{x}^{(k)})}.$$

- Constrained optimization: We can use $\underline{\text{interior-point method}}$ or $\underline{\text{proximal methods}}$, ...

Eigenvalue and Singular Value Decomposition

Eigenvalue Problem

- Eigenvalue and generalized eigenvalue problems play important roles in different fields of science, including machine learning, physics, statistics, and mathematics.
- In the eigenvalue problem, the eigenvectors of a matrix represent the most important and informative directions of that matrix. For example, if the matrix is a covariance matrix of data, the eigenvectors represent the directions of the spread or variance of data and the corresponding eigenvalues are the magnitude of the spread in these directions [5].
- These directions are impacted by another matrix in the generalized eigenvalue problem. If the other matrix is the identity matrix, this impact is cancelled and the eigenvalue problem captures the directions of the maximum spread.
- The **eigenvalue problem** [6, 7] of a symmetric matrix $\boldsymbol{A} \in \mathbb{R}^{d \times d}$ is defined as:

$$\boxed{\underset{\text{matrix}}{\boldsymbol{A}}\underset{\text{vect}}{\boldsymbol{\phi}_i} = \underset{\text{scalar}}{\lambda_i}\boldsymbol{\phi}_i,} \quad \forall i \in \{1, \dots, d\}, \tag{6}$$

and in matrix form, it is:

$$\star \quad \boxed{\boldsymbol{A\Phi} = \boldsymbol{\Phi\Lambda},} \tag{7}$$

where the columns of $\mathbb{R}^{d \times d} \ni \boldsymbol{\Phi} := [\boldsymbol{\phi}_1, \dots, \boldsymbol{\phi}_d]$ are the eigenvectors and diagonal elements of $\mathbb{R}^{d \times d} \ni \boldsymbol{\Lambda} := \textbf{diag}([\lambda_1, \dots, \lambda_d]^\top)$ are the eigenvalues. Note that $\boldsymbol{\phi}_i \in \mathbb{R}^d$ and $\lambda_i \in \mathbb{R}$.

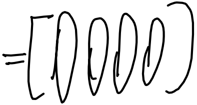
Eigenvalue Problem

- For the eigenvalue problem, the matrix $\boldsymbol{A}$ can be nonsymmetric. If the matrix is symmetric, its eigenvectors are orthogonal/orthonormal and if it is nonsymmetric, its eigenvectors are not orthogonal/orthonormal.
- Equation (7) can be restated as:

$$\bigstar\ \boxed{\boldsymbol{A}\boldsymbol{\Phi} = \boldsymbol{\Phi}\boldsymbol{\Lambda}} \implies \boldsymbol{A}\underbrace{\boldsymbol{\Phi}\boldsymbol{\Phi}^\top}_{I} = \boldsymbol{\Phi}\boldsymbol{\Lambda}\boldsymbol{\Phi}^\top \implies \boldsymbol{A} = \boldsymbol{\Phi}\boldsymbol{\Lambda}\boldsymbol{\Phi}^\top = \boxed{\boldsymbol{\Phi}\boldsymbol{\Lambda}\boldsymbol{\Phi}^{-1},}\ \bigstar \tag{8}$$

where $\boldsymbol{\Phi}^\top = \boldsymbol{\Phi}^{-1}$ because $\boldsymbol{\Phi}$ is an orthogonal matrix.

- There is always $\boldsymbol{\Phi}^\top\boldsymbol{\Phi} = \boldsymbol{I}$ for orthogonal $\boldsymbol{\Phi}$, but there is only $\boldsymbol{\Phi}\boldsymbol{\Phi}^\top = \boldsymbol{I}$ if "all" columns of the orthogonal $\boldsymbol{\Phi}$ exist (it is not truncated, i.e., it is a square matrix). Equation (8) is referred to as "**eigenvalue decomposition**", "eigen-decomposition", or "spectral decomposition".

$\Phi =$ 

$\rightarrow \Phi^T\Phi = \Phi\Phi^T = I$

$\rightarrow \Phi^T\Phi = I$

$\Phi\Phi^T \neq I$

Generalized Eigenvalue Problem

- The **generalized eigenvalue problem** [8, 7] of two symmetric matrices $\boldsymbol{A} \in \mathbb{R}^{d \times d}$ and $\boldsymbol{B} \in \mathbb{R}^{d \times d}$ is defined as:

$$\boxed{\boldsymbol{A}\boldsymbol{\phi}_i = \lambda_i \boldsymbol{B}\boldsymbol{\phi}_i,} \quad \forall i \in \{1, \dots, d\}, \tag{9}$$

  and in matrix form, it is:

$$\boxed{\boldsymbol{A}\boldsymbol{\Phi} = \boldsymbol{B}\boldsymbol{\Phi}\boldsymbol{\Lambda},} \tag{10}$$

  where the columns of $\mathbb{R}^{d \times d} \ni \boldsymbol{\Phi} := [\boldsymbol{\phi}_1, \dots, \boldsymbol{\phi}_d]$ are the eigenvectors and diagonal elements of $\mathbb{R}^{d \times d} \ni \boldsymbol{\Lambda} := \textbf{diag}([\lambda_1, \dots, \lambda_d]^\top)$ are the eigenvalues. Note that $\boldsymbol{\phi}_i \in \mathbb{R}^d$ and $\lambda_i \in \mathbb{R}$.
- The generalized eigenvalue problem of Eq. (9) or (10) is denoted by $\underline{(\boldsymbol{A}, \boldsymbol{B})}$.
- The $(\boldsymbol{A}, \boldsymbol{B})$ is called a “pair” or “pencil” [8], and the order in the pair matters, according to Eq. (10).
- The $\boldsymbol{\Phi}$ and $\boldsymbol{\Lambda}$ are called the generalized eigenvectors and eigenvalues of $(\boldsymbol{A}, \boldsymbol{B})$.
- The $(\boldsymbol{\Phi}, \boldsymbol{\Lambda})$ or $(\boldsymbol{\phi}_i, \lambda_i)$ is called the “eigenpair” of the pair $\underline{(\boldsymbol{A}, \boldsymbol{B})}$ in the literature [8].
- Comparing Eqs. (6) and (9) or Eqs. (7) and (10) demonstrates that the eigenvalue problem is a special case of the generalized eigenvalue problem where $\boldsymbol{B} = \boldsymbol{I}$.

Singular Value Decomposition

- **Singular Value Decomposition (SVD)** [9] is one of the most well-known and effective matrix decomposition methods. There are different methods for obtaining this decomposition, one of which is Jordan's algorithm [9].
- SVD has two different forms, i.e., complete and incomplete.
- Consider a matrix $\boldsymbol{A} \in \mathbb{R}^{\alpha \times \beta}$. The **complete SVD** decomposes the matrix as:

$$\mathbb{R}^{\alpha \times \beta} \ni \boxed{\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top,} \\ \underbrace{\boldsymbol{U} \in \mathbb{R}^{\alpha \times \alpha}}_{}, \quad \underbrace{\boldsymbol{V} \in \mathbb{R}^{\beta \times \beta}}_{}, \quad \underbrace{\boldsymbol{\Sigma} \in \mathbb{R}^{\alpha \times \beta},}_{} \tag{11}$$

  where the columns of $\boldsymbol{U}$ and the columns of $\boldsymbol{V}$ are called *left singular vectors* and *right singular vectors*, respectively.
- In complete SVD, $\boldsymbol{\Sigma}$ is a *rectangular* diagonal matrix whose main diagonal includes the *singular values*. In the cases with $\alpha > \beta$ and $\alpha < \beta$, this matrix is in the following forms:

$$\ast \quad \boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1 & 0 & 0 \\ \vdots & \ddots & \vdots \\ 0 & 0 & \sigma_\beta \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \end{bmatrix} \text{ and } \begin{bmatrix} \sigma_1 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & 0 & \cdots & 0 \\ 0 & 0 & \sigma_\alpha & 0 & \cdots & 0 \end{bmatrix},$$

  respectively. In other words, the number of singular values is $\min(\alpha, \beta)$.

Singular Value Decomposition

- The **incomplete SVD** decomposes the matrix as:

$$\mathbb{R}^{\alpha\times\beta} \ni \boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top, \tag{12}$$
$$\underbrace{\boldsymbol{U} \in \mathbb{R}^{\alpha\times k}}, \quad \underbrace{\boldsymbol{V} \in \mathbb{R}^{\beta\times k}}, \quad \underbrace{\boldsymbol{\Sigma} \in \mathbb{R}^{k\times k}},$$

  where [10]:

$$k := \min(\alpha, \beta), \tag{13}$$

  and the columns of $\boldsymbol{U}$ and the columns of $\boldsymbol{V}$ are called *left singular vectors* and *right singular vectors*, respectively.

- In incomplete SVD, $\boldsymbol{\Sigma}$ is a *square* diagonal matrix whose main diagonal includes the *singular values*. The matrix $\boldsymbol{\Sigma}$ is in the form:

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1 & 0 & 0 \\ \vdots & \ddots & \vdots \\ 0 & 0 & \sigma_k \end{bmatrix}.$$

Singular Value Decomposition

- Note that in both complete and incomplete SVD, the left singular vectors are orthonormal and the right singular vectors are also orthonormal; therefore, $\boldsymbol{U}$ and $\boldsymbol{V}$ are both orthogonal matrices so:

$$\boxed{\begin{aligned} \boldsymbol{U}^\top \boldsymbol{U} &= \boldsymbol{I}, \\ \boldsymbol{V}^\top \boldsymbol{V} &= \boldsymbol{I}. \end{aligned}} \qquad \begin{aligned} &(14) \\ &(15) \end{aligned}$$

  If these orthogonal matrices are not truncated and thus are square matrices, e.g. for complete SVD, there are also:

$$\boxed{\begin{aligned} \boldsymbol{U}\boldsymbol{U}^\top &= \boldsymbol{I}, \\ \boldsymbol{V}\boldsymbol{V}^\top &= \boldsymbol{I}. \end{aligned}} \qquad \begin{aligned} &(16) \\ &(17) \end{aligned}$$

Relation of SVD and EVD

$(xy)^\top = y^\top x^\top$

- In both complete and incomplete SVD of matrix $\boldsymbol{A}$, the left and right singular vectors are the eigenvectors of $\boldsymbol{A}\boldsymbol{A}^\top$ and $\boldsymbol{A}^\top\boldsymbol{A}$, respectively, and the singular values are the square root of eigenvalues of either $\boldsymbol{A}\boldsymbol{A}^\top$ or $\boldsymbol{A}^\top\boldsymbol{A}$.
- Proof: There is:

$$\bigstar\ \underbrace{\boldsymbol{A}\boldsymbol{A}^\top} = \underbrace{(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top)}\underbrace{(\overset{\downarrow}{\boldsymbol{U}}\overset{\downarrow}{\boldsymbol{\Sigma}}\overset{\downarrow}{\boldsymbol{V}^\top})^{\top}} = \overbrace{\boldsymbol{U}\boldsymbol{\Sigma}\,\boldsymbol{V}^\top}\overbrace{\boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{U}^\top} = \underbrace{\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{\Sigma}\boldsymbol{U}^\top} = \underbrace{\boldsymbol{U}\boldsymbol{\Sigma}^2\boldsymbol{U}^\top},$$

$$\underbrace{\boldsymbol{V}^\top\boldsymbol{V}}_{I}$$

which is the eigen-decomposition [11] of $\boldsymbol{A}\boldsymbol{A}^\top$ where the columns of $\boldsymbol{U}$ are the eigenvectors and the diagonal of $\boldsymbol{\Sigma}^2$ are the eigenvalues so the diagonal of $\boldsymbol{\Sigma}$ are the square root of eigenvalues.

- Also:

$$\bigstar\ \boldsymbol{A}^\top\boldsymbol{A} = \underbrace{(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top)^\top}\underbrace{(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top)} = \overbrace{\boldsymbol{V}\boldsymbol{\Sigma}\,\underbrace{\boldsymbol{U}^\top\boldsymbol{U}}_{I}}\boldsymbol{\Sigma}\boldsymbol{V}^\top = \underbrace{\boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{\Sigma}\boldsymbol{V}^\top} = \underbrace{\boldsymbol{V}\boldsymbol{\Sigma}^2\boldsymbol{V}^\top},$$

which is the eigenvalue decomposition of $\boldsymbol{A}^\top\boldsymbol{A}$ where the columns of $\boldsymbol{V}$ are the eigenvectors and the diagonal of $\boldsymbol{\Sigma}^2$ are the eigenvalues, so the diagonal of $\boldsymbol{\Sigma}$ are the square root of eigenvalues.

Acknowledgement

- Some slides are based on our textbook: “Elements of Dimensionality Reduction and Manifold Learning” [1]
- For more information on optimization, refer to my course “Optimization Techniques” in University of Guelph. Link in my YouTube channel: `https://www.youtube.com/playlist?list=PLPrxGIUWsqP3ZBM4Zy5YqfCh1BqM5sJov`

References

[1] B. Ghojogh, M. Crowley, F. Karray, and A. Ghodsi, *Elements of Dimensionality Reduction and Manifold Learning*.
Springer Nature, 2023.

[2] S. Garcia, J. Derrac, J. Cano, and F. Herrera, "Prototype selection for nearest neighbor classification: Taxonomy and empirical study," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 3, pp. 417–435, 2012.

[3] I. Triguero, J. Derrac, S. Garcia, and F. Herrera, "A taxonomy and experimental study on prototype generation for nearest neighbor classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 1, pp. 86–100, 2011.

[4] B. Ghojogh, "Data reduction algorithms in machine learning and data science," 2021.

[5] I. Jolliffe, *Principal component analysis*.
Springer, 2011.

[6] J. H. Wilkinson, *The algebraic eigenvalue problem*, vol. 662.
Oxford Clarendon, 1965.

[7] G. H. Golub and C. F. Van Loan, *Matrix computations*, vol. 3.
The Johns Hopkins University Press, 2012.

[8] B. N. Parlett, "The symmetric eigenvalue problem," *Classics in Applied Mathematics*, vol. 20, 1998.

References (cont.)

[9] G. W. Stewart, “On the early history of the singular value decomposition,” *SIAM review*, vol. 35, no. 4, pp. 551–566, 1993.

[10] G. H. Golub and C. Reinsch, “Singular value decomposition and least squares solutions,” *Numerische mathematik*, vol. 14, no. 5, pp. 403–420, 1970.

[11] B. Ghojogh, F. Karray, and M. Crowley, “Eigenvalue and generalized eigenvalue problems: Tutorial,” *arXiv preprint arXiv:1903.11240*, 2019.