APS1070

Foundations of Data Analytics and Machine Learning

Summer 2020

Wed July 8 / Week 9:

- *Midterm review*
- *Analytical Geometry + Matrix Decompositions (continued)*
- *PCA + SVD*
- *Vector Caculus*

Jason Riordon, PhD

News

- Project 3 due Sunday, July 12, 11:00 pm
- Feedback – please!

Slide Attribution

These slides contain materials from various sources. Special thanks to Scott Sanner and Marc Deisenroth.

Midterm Review

Question 2 2 pts

Which of the following statements is false?

1. Basis vectors forming an orthogonal basis are always orthonormal.
2. Basis vectors forming an orthonormal basis are always orthogonal.
3. Basis vectors forming an orthonormal basis are always linearly independent.
4. All vectors in an orthonormal basis has length 1.

Question 3

2 pts

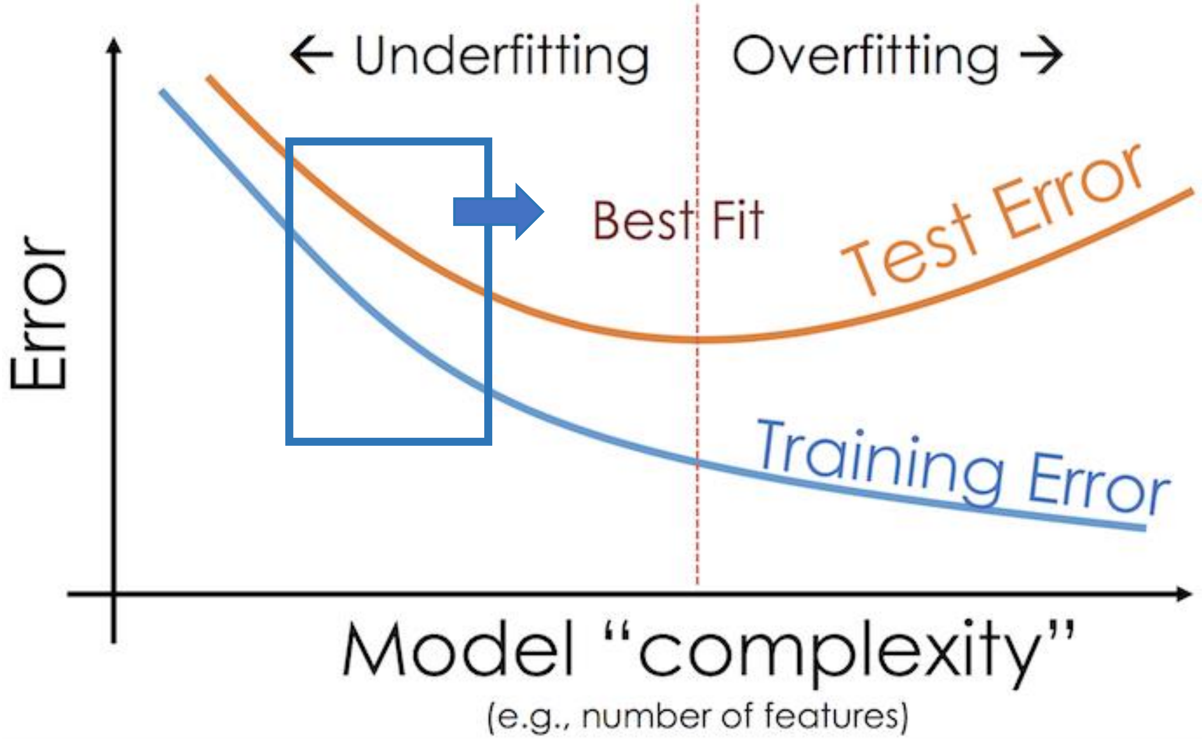
In lecture, we discussed decision trees – an intuitive classification model that splits on different attributes, creating a tree-like structure. A data scientist is given a large data set and uses part of the data to **train** a really big decision tree with many branches and nodes, that perfectly fits the data. When they apply it to the **validation** data, overall accuracy is only 78%.

1. Why is test performance so poor? **overfitting**
2. What can the data scientist do to improve the model? **multiple strategies - stop each branch based on some criteria during creation of the tree (minimum # of examples to continue), set max length for a branch, or "pruning" – removing branches based on least important features, or in such a way as to not hurt accuracy too much**

Question 4 2 pts

A data scientist has a data set with a lot of features and chooses to use some of these features to train a model on training data and evaluate performance on testing data. They find that both training and testing accuracy is poor. What would you recommend (i) removing a few features or (ii) adding more features? Explain.

Adding more features – the model is not sufficiently complex (underfitting).



https://www.textbook.ds100.org/ch/15/bias_cv.html

Question 5

2 pts

A data set with four features has the following covariance matrix:

	A	B	C	D
A	0.5	0.018	0.11	0.048
B	0.018	0.01	0.0025	0.14
C	0.11	0.0025	0.023	0.0055
D	0.048	0.14	0.0055	6

You're asked to remove a highly correlated feature from the data set. Which one would you remove?

If you calculate all correlations, you find that A and C are highly correlated.

Question 6

4 pts

You have two binary classification models (P_1 and P_2), that use a series of features to predict the probability of emails being spam. The computed probabilities are shown in the table below, along with actual labels, for six **validation** data.

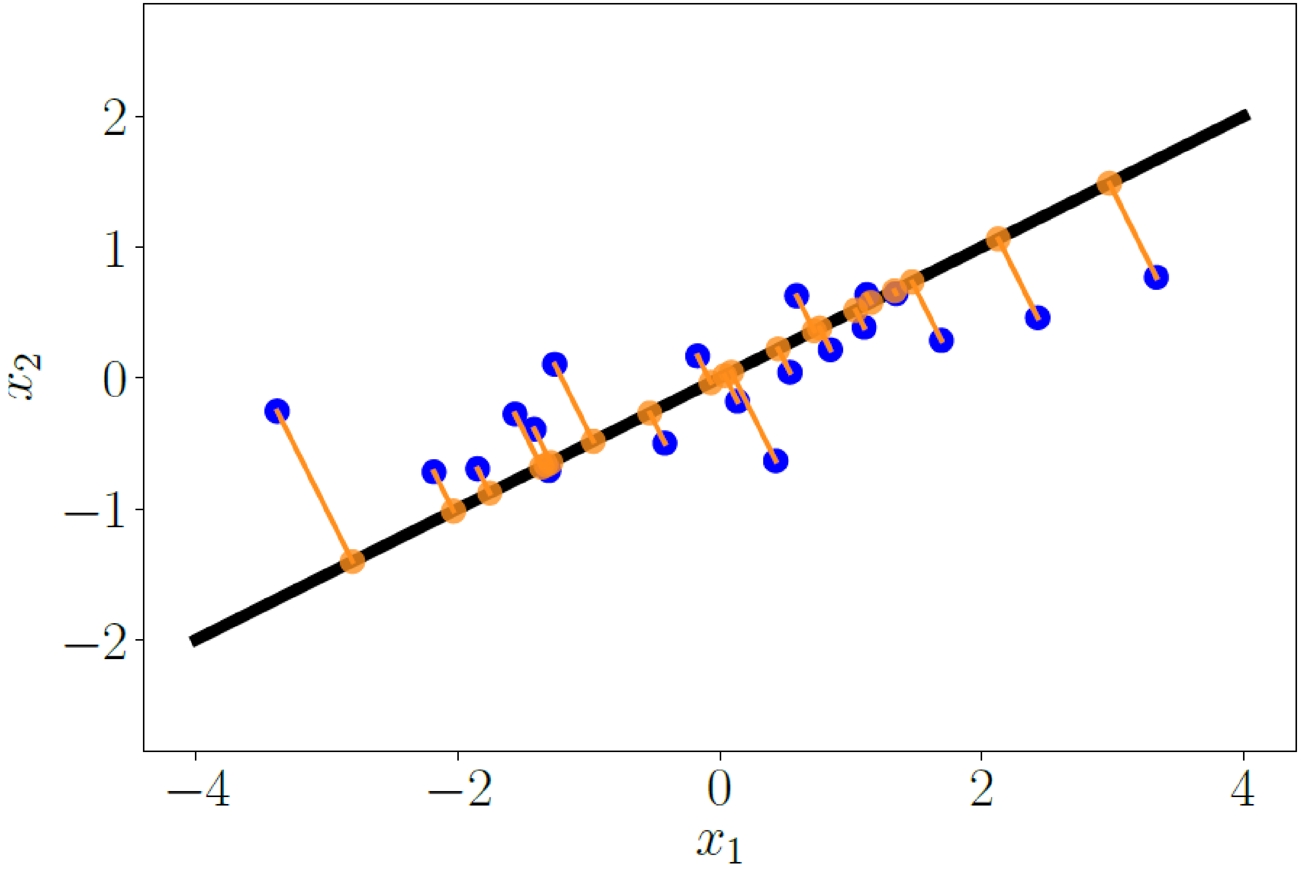
	Label	P_1	P_2
1	0	0.1	0.1
2	0	0.4	0.5
3	0	0.3	0.5
4	1	0.5	0.4
5	1	0.4	0.8
6	1	0.8	0.6

1. Calculate the AUC for each model. **AUC_1 = 0.94, AUC_2 = 0.77**
2. Assuming you value F1-score, which model would you choose? **P_1 F1 score = 0.85, P_2 F1 score = 0.79, choose P_1**
3. What is the precision, recall, accuracy and confusion matrix for this best model? **precision = 0.75, recall = 1, accuracy = 0.833 and CM=** $\begin{bmatrix} 2 & 1 \\ 0 & 3 \end{bmatrix}$

Analytical Geometry

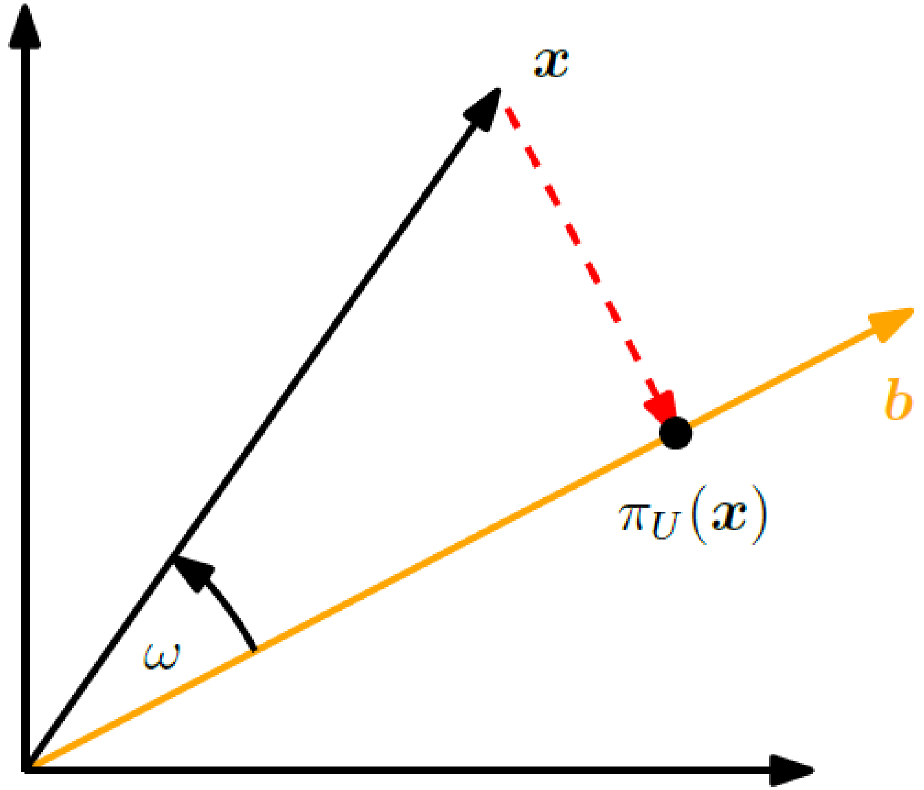
Orthogonal Projections

Figure 3.9 Orthogonal projection (orange dots) of a two-dimensional dataset (blue dots) onto a one-dimensional subspace (straight line).



Orthogonal Projections

3.8 *Orthogonal Projections*



(a) Projection of $\boldsymbol{x} \in \mathbb{R}^2$ onto a subspace U with basis vector $\boldsymbol{b}$.

Orthogonal Projections

Definition 3.10 (Projection). Let V be a vector space and $U \subseteq V$ a subspace of V. A linear mapping $\pi : V \to U$ is called a *projection* if $\pi^2 = \pi \circ \pi = \pi$.

Example 3.10 (Projection onto a Line)
Find the projection matrix $\boldsymbol{P}_\pi$ onto the line through the origin spanned by $\boldsymbol{b} = \begin{bmatrix}1 & 2 & 2\end{bmatrix}^\top$. $\boldsymbol{b}$ is a direction and a basis of the one-dimensional subspace (line through origin).

With (3.46), we obtain

$$\boldsymbol{P}_\pi = \frac{\boldsymbol{b}\boldsymbol{b}^\top}{\boldsymbol{b}^\top\boldsymbol{b}} = \frac{1}{9}\begin{bmatrix}1\\2\\2\end{bmatrix}\begin{bmatrix}1 & 2 & 2\end{bmatrix} = \frac{1}{9}\begin{bmatrix}1 & 2 & 2\\2 & 4 & 4\\2 & 4 & 4\end{bmatrix}. \tag{3.47}$$

Let us now choose a particular $\boldsymbol{x}$ and see whether it lies in the subspace spanned by $\boldsymbol{b}$. For $\boldsymbol{x} = \begin{bmatrix}1 & 1 & 1\end{bmatrix}^\top$, the projection is

$$\pi_U(\boldsymbol{x}) = \boldsymbol{P}_\pi\boldsymbol{x} = \frac{1}{9}\begin{bmatrix}1 & 2 & 2\\2 & 4 & 4\\2 & 4 & 4\end{bmatrix}\begin{bmatrix}1\\1\\1\end{bmatrix} = \frac{1}{9}\begin{bmatrix}5\\10\\10\end{bmatrix} \in \mathrm{span}\left[\begin{bmatrix}1\\2\\2\end{bmatrix}\right]. \tag{3.48}$$

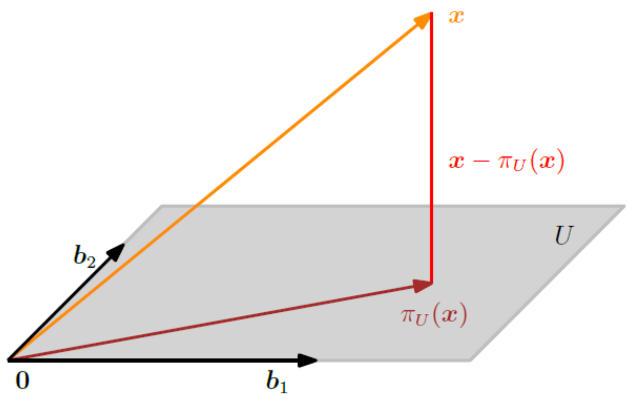
Note that the application of $\boldsymbol{P}_\pi$ to $\pi_U(\boldsymbol{x})$ does not change anything, i.e., $\boldsymbol{P}_\pi\pi_U(\boldsymbol{x}) = \pi_U(\boldsymbol{x})$. This is expected because according to Definition 3.10, we know that a projection matrix $\boldsymbol{P}_\pi$ satisfies $\boldsymbol{P}_\pi^2\boldsymbol{x} = \boldsymbol{P}_\pi\boldsymbol{x}$ for all $\boldsymbol{x}$.

Example 3.10

Orthogonal Projections

$$\boldsymbol{\lambda} = (\boldsymbol{B}^\top \boldsymbol{B})^{-1}\boldsymbol{B}^\top \boldsymbol{x}$$

coordinates



Example 3.11 (Projection onto a Two-dimensional Subspace)

For a subspace $U = \text{span}[\begin{bmatrix}1\\1\\1\end{bmatrix}, \begin{bmatrix}0\\1\\2\end{bmatrix}] \subseteq \mathbb{R}^3$ and $\boldsymbol{x} = \begin{bmatrix}6\\0\\0\end{bmatrix} \in \mathbb{R}^3$ find the coordinates $\boldsymbol{\lambda}$ of $\boldsymbol{x}$ in terms of the subspace U, the projection point $\pi_U(\boldsymbol{x})$ and the projection matrix $\boldsymbol{P}_\pi$.

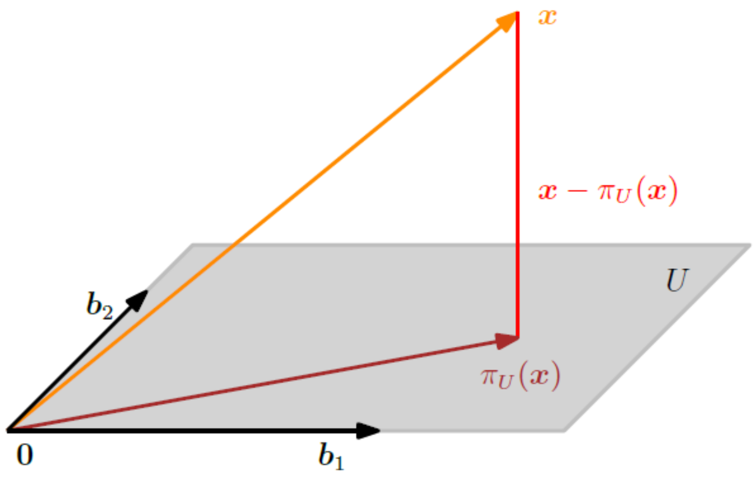
First, we see that the generating set of U is a basis (linear independence) and write the basis vectors of U into a matrix $\boldsymbol{B} = \begin{bmatrix}1 & 0\\1 & 1\\1 & 2\end{bmatrix}$.

Second, we compute the matrix $\boldsymbol{B}^\top \boldsymbol{B}$ and the vector $\boldsymbol{B}^\top \boldsymbol{x}$ as

$$\boldsymbol{B}^\top \boldsymbol{B} = \begin{bmatrix}1 & 1 & 1\\0 & 1 & 2\end{bmatrix}\begin{bmatrix}1 & 0\\1 & 1\\1 & 2\end{bmatrix} = \begin{bmatrix}3 & 3\\3 & 5\end{bmatrix}, \quad \boldsymbol{B}^\top \boldsymbol{x} = \begin{bmatrix}1 & 1 & 1\\0 & 1 & 2\end{bmatrix}\begin{bmatrix}6\\0\\0\end{bmatrix} = \begin{bmatrix}6\\0\end{bmatrix}. \tag{3.60}$$

Orthogonal Projections

$$\boldsymbol{\lambda} = (\boldsymbol{B}^\top \boldsymbol{B})^{-1}\boldsymbol{B}^\top \boldsymbol{x}$$



Example 3.11

Third, we solve the normal equation $\boldsymbol{B}^\top \boldsymbol{B}\boldsymbol{\lambda} = \boldsymbol{B}^\top \boldsymbol{x}$ to find $\boldsymbol{\lambda}$:

$$\begin{bmatrix} 3 & 3 \\ 3 & 5 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 6 \\ 0 \end{bmatrix} \iff \boldsymbol{\lambda} = \begin{bmatrix} 5 \\ -3 \end{bmatrix}. \tag{3.61}$$

Fourth, the projection $\pi_U(\boldsymbol{x})$ of $\boldsymbol{x}$ onto U, i.e., into the column space of $\boldsymbol{B}$, can be directly computed via

$$\pi_U(\boldsymbol{x}) = \boldsymbol{B}\boldsymbol{\lambda} = \begin{bmatrix} 5 \\ 2 \\ -1 \end{bmatrix}. \tag{3.62}$$

The corresponding *projection error* is the norm of the difference vector between the original vector and its projection onto U, i.e.,

$$\|\boldsymbol{x} - \pi_U(\boldsymbol{x})\| = \left\| \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}^\top \right\| = \sqrt{6}. \tag{3.63}$$

Fifth, the projection matrix (for any $\boldsymbol{x} \in \mathbb{R}^3$) is given by

$$\boldsymbol{P}_\pi = \boldsymbol{B}(\boldsymbol{B}^\top \boldsymbol{B})^{-1}\boldsymbol{B}^\top = \frac{1}{6}\begin{bmatrix} 5 & 2 & -1 \\ 2 & 2 & 2 \\ -1 & 2 & 5 \end{bmatrix}. \tag{3.64}$$

To verify the results, we can (a) check whether the displacement vector $\pi_U(\boldsymbol{x}) - \boldsymbol{x}$ is orthogonal to all basis vectors of U, and (b) verify that $\boldsymbol{P}_\pi = \boldsymbol{P}_\pi^2$ (see Definition 3.10).

Orthogonal Projections

Orthonormal basis

Remark. We just looked at projections of vectors $\boldsymbol{x}$ onto a subspace U with basis vectors $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k\}$. If this basis is an ONB, i.e., (3.33) and (3.34) are satisfied, the projection equation (3.58) simplifies greatly to

$$\pi_U(\boldsymbol{x}) = \boldsymbol{B}\boldsymbol{B}^\top \boldsymbol{x} \tag{3.65}$$

since $\boldsymbol{B}^\top \boldsymbol{B} = \boldsymbol{I}$ with coordinates

$$\boldsymbol{\lambda} = \boldsymbol{B}^\top \boldsymbol{x} \,. \tag{3.66}$$

This means that we no longer have to compute the inverse from (3.58), which saves computation time. $\diamondsuit$

Matrix Decompositions

Determinant and Trace

$$\det(\boldsymbol{A}) = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix}$$

The *determinant* of a square matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is a function that maps $\boldsymbol{A}$ onto a real number.

Determinant and trace

Example 4.1 (Testing for Matrix Invertibility)
Let us begin with exploring if a square matrix $\boldsymbol{A}$ is invertible (see Section 2.2.2). For the smallest cases, we already know when a matrix is invertible. If $\boldsymbol{A}$ is a 1×1 matrix, i.e., it is a scalar number, then $\boldsymbol{A} = a \implies \boldsymbol{A}^{-1} = \frac{1}{a}$. Thus $a\,\frac{1}{a} = 1$ holds, if and only if $a \neq 0$.

For 2×2 matrices, by the definition of the inverse (Definition 2.3), we know that $\boldsymbol{A}\boldsymbol{A}^{-1} = \boldsymbol{I}$. Then, with (2.24), the inverse of $\boldsymbol{A}$ is

$$\boldsymbol{A}^{-1} = \frac{1}{a_{11}a_{22} - a_{12}a_{21}} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix}. \tag{4.2}$$

Hence, $\boldsymbol{A}$ is invertible if and only if

$$a_{11}a_{22} - a_{12}a_{21} \neq 0\,. \tag{4.3}$$

This quantity is the determinant of $\boldsymbol{A} \in \mathbb{R}^{2\times 2}$, i.e.,

$$\det(\boldsymbol{A}) = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21}\,. \tag{4.4}$$

Determinant and trace

For $n = 3$ (known as Sarrus' rule),

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11}a_{22}a_{33} + a_{21}a_{32}a_{13} + a_{31}a_{12}a_{23} \qquad (4.7)$$

$$- a_{31}a_{22}a_{13} - a_{11}a_{32}a_{23} - a_{21}a_{12}a_{33} \, .$$

Determinant and trace

Definition 4.4. The *trace* of a square matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is defined as

$$\mathrm{tr}(\boldsymbol{A}) := \sum_{i=1}^{n} a_{ii} \,,$$

i.e. , the trace is the sum of the diagonal elements of $\boldsymbol{A}$.

Eigenvalues and Eigenvectors

characteristic roots

Definition 4.6. Let $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ be a square matrix. Then $\lambda \in \mathbb{R}$ is an *eigenvalue* of $\boldsymbol{A}$ and $\boldsymbol{x} \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}$ is the corresponding *eigenvector* of $\boldsymbol{A}$ if

$$\boldsymbol{A}\boldsymbol{x} = \lambda \boldsymbol{x} \,. \tag{4.25}$$

We call (4.25) the *eigenvalue equation*.

When you perform a linear transformation A, direction doesn't change, scales as lambda. This is unique! Usually , when you transform a vector, it changes direction.

Eigenvalues and Eigenvectors

How to calculate

Example 4.5 (Computing Eigenvalues, Eigenvectors, and Eigenspaces)
Let us find the eigenvalues and eigenvectors of the 2×2 matrix

$$\boldsymbol{A} = \begin{bmatrix} 4 & 2 \\ 1 & 3 \end{bmatrix} . \tag{4.28}$$

Step 1: Characteristic Polynomial. From our definition of the eigenvector $\boldsymbol{x} \neq \boldsymbol{0}$ and eigenvalue λ of $\boldsymbol{A}$, there will be a vector such that $\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x}$, i.e., $(\boldsymbol{A} - \lambda\boldsymbol{I})\boldsymbol{x} = \boldsymbol{0}$. Since $\boldsymbol{x} \neq \boldsymbol{0}$, this requires that the kernel (null space) of $\boldsymbol{A} - \lambda\boldsymbol{I}$ contains more elements than just $\boldsymbol{0}$. This means that $\boldsymbol{A} - \lambda\boldsymbol{I}$ is not invertible and therefore $\det(\boldsymbol{A} - \lambda\boldsymbol{I}) = 0$. Hence, we need to compute the roots of the characteristic polynomial (4.22a) to find the eigenvalues.

Step 2: Eigenvalues. The characteristic polynomial is

$$\begin{aligned} p_{\boldsymbol{A}}(\lambda) &= \det(\boldsymbol{A} - \lambda\boldsymbol{I}) & (4.29a)\\ &= \det\left(\begin{bmatrix} 4 & 2 \\ 1 & 3 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}\right) = \begin{vmatrix} 4-\lambda & 2 \\ 1 & 3-\lambda \end{vmatrix} & (4.29b)\\ &= (4-\lambda)(3-\lambda) - 2 \cdot 1 . & (4.29c) \end{aligned}$$

We factorize the characteristic polynomial and obtain

$$p(\lambda) = (4-\lambda)(3-\lambda) - 2 \cdot 1 = 10 - 7\lambda + \lambda^2 = (2-\lambda)(5-\lambda) \tag{4.30}$$

giving the roots $\lambda_1 = 2$ and $\lambda_2 = 5$.

Eigenvalues and Eigenvectors

Step 3: Eigenvectors and Eigenspaces. We find the eigenvectors that correspond to these eigenvalues by looking at vectors $\boldsymbol{x}$ such that

$$\begin{bmatrix} 4-\lambda & 2 \\ 1 & 3-\lambda \end{bmatrix} \boldsymbol{x} = \boldsymbol{0}\,. \tag{4.31}$$

For $\lambda = 5$ we obtain

$$\begin{bmatrix} 4-5 & 2 \\ 1 & 3-5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -1 & 2 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \boldsymbol{0}\,. \tag{4.32}$$

We solve this homogeneous system and obtain a solution space

$$E_5 = \text{span}[\begin{bmatrix} 2 \\ 1 \end{bmatrix}]\,. \tag{4.33}$$

This eigenspace is one-dimensional as it possesses a single basis vector.

Analogously, we find the eigenvector for $\lambda = 2$ by solving the homogeneous system of equations

$$\begin{bmatrix} 4-2 & 2 \\ 1 & 3-2 \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} 2 & 2 \\ 1 & 1 \end{bmatrix} \boldsymbol{x} = \boldsymbol{0}\,. \tag{4.34}$$

This means any vector $\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, where $x_2 = -x_1$, such as $\begin{bmatrix} 1 \\ -1 \end{bmatrix}$, is an eigenvector with eigenvalue 2. The corresponding eigenspace is given as

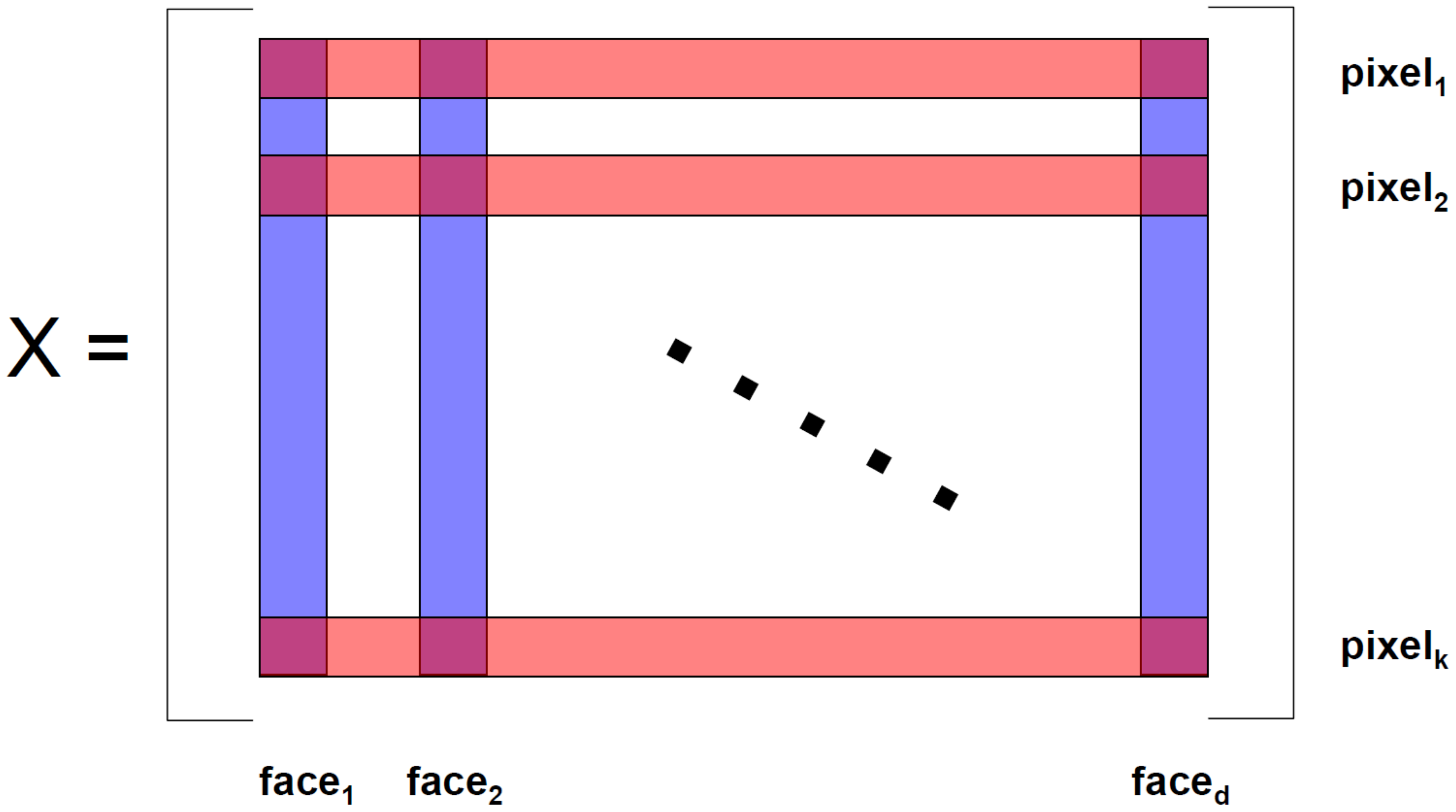
$$E_2 = \text{span}[\begin{bmatrix} 1 \\ -1 \end{bmatrix}]\,. \tag{4.35}$$

PCA + SVD

Normalized Face Data

The Data Matrix X

- Matrix with columns as faces, rows as pixels



The Covariance Matrix

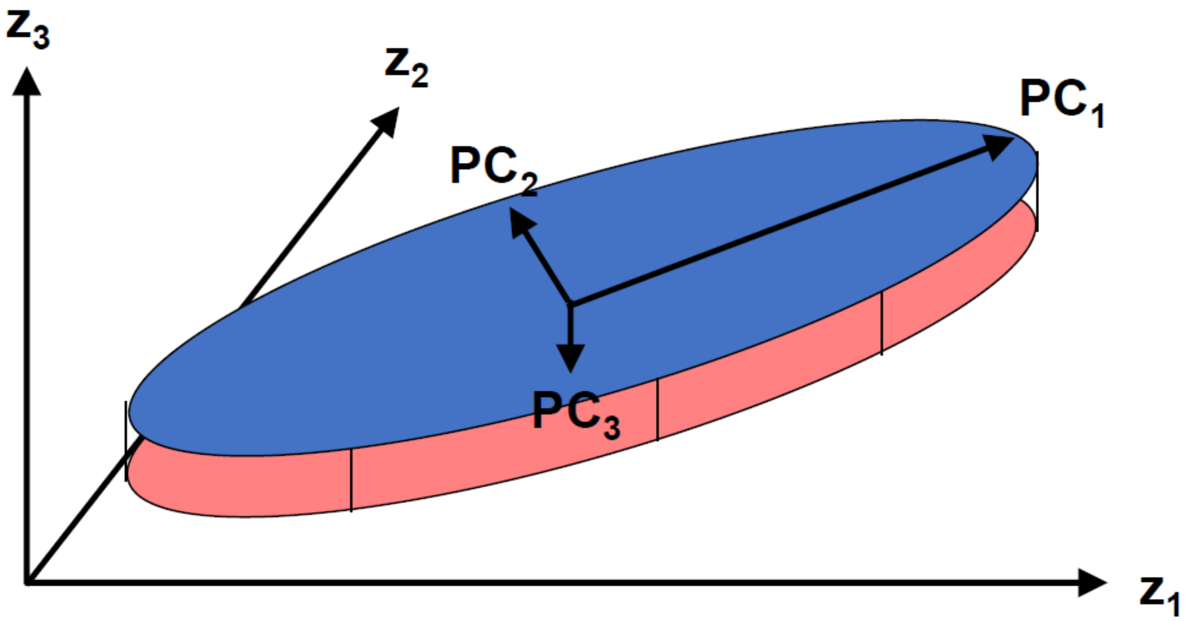
- Compute mean: $m = 1/d \, \Sigma_{j=1..d} \, x_j$
- Center each face: $x_i := x_i - m$
- Look at Cov. for a single face $f = x_i$

$$\mathrm{Cov}(f) = f\,f^T = \begin{bmatrix} f_i^2 & f_j f_i & \\ f_i f_j & & \\ & & \ddots \\ & & \end{bmatrix}$$

- Verify covariance for all data is XX^T
  - Make sure X is centered

Principle Components

- **Why look at eigenvectors of covariance?**



- **If data lives in linear subspace…**
  - Covariance indicates principle data dimensions
  - Then eigenvectors = ‘principle data components’

A First Approach

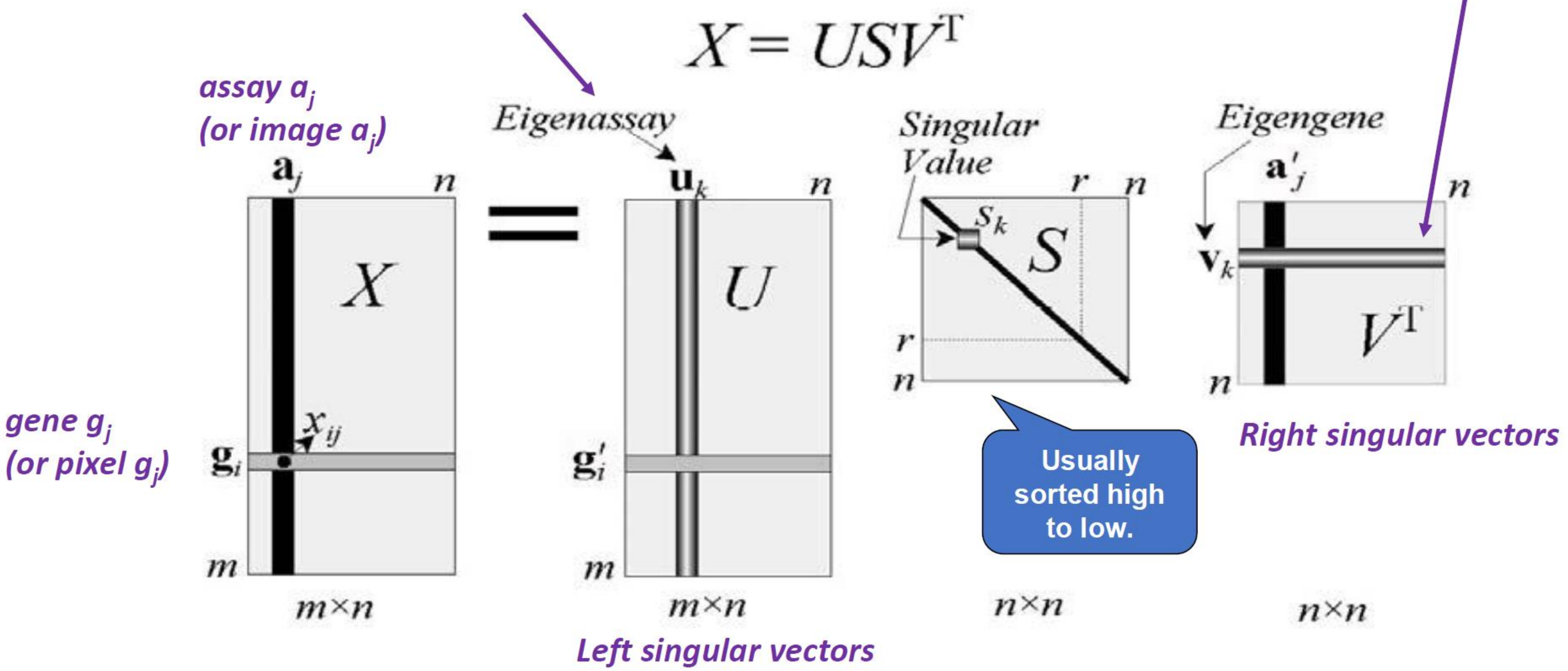
- **Do eigenvalue decomp: $XX^T = U\Sigma U^T$**
  - Σ is diagonal (eigenvalues)
  - U is orthogonal matrix of eigenvectors (cols)
- **What if data dimension (k) is large?**
  - Will require $O(k^3)$ time!
  - Impractical for large k

SVD to the Rescue!

The columns of U are called the *left singular vectors*, $\{\mathbf{u}_k\}$, and form an orthonormal basis for the assay expression profiles

The rows of V^T contain the elements of the *right singular vectors*, $\{\mathbf{v}_k\}$, and form an orthonormal basis for the gene transcriptional responses

$$X = USV^T$$



Taken from Wall et al, "[Singular value decomposition and principal component analysis]", 2003.

See also… http://web.mit.edu/be.400/www/SVD/Singular_Value_Decomposition.htm and https://www.cc.gatech.edu/~dellaert/pubs/svd-note.pdf

A Second Approach

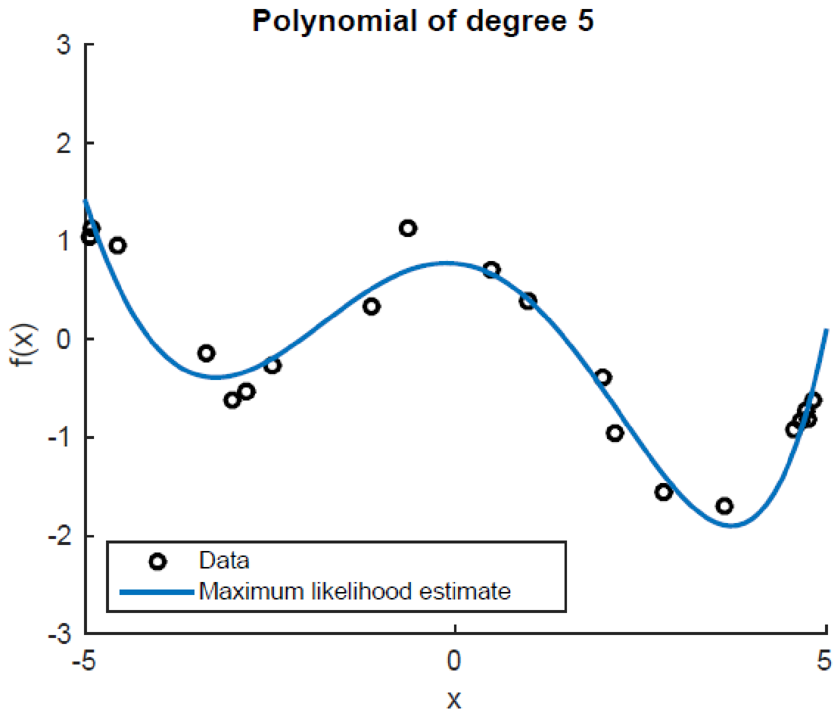
If X is a square, symmetric matrix = SVD is equivalent to diagonalization, or solution to the eigenvalue problem

- **Do SVD: $X = USV^T$**
  - S is diagonal (sqrt. of eigenvalues)
  - U is orthogonal matrix of “input” eigenvectors (in cols)
    - If x has dim k x d, U has dim k x d
    - Much more computationally tractable
  - V is orthogonal matrix of “output” eigenvectors

See notebook, whiteboard

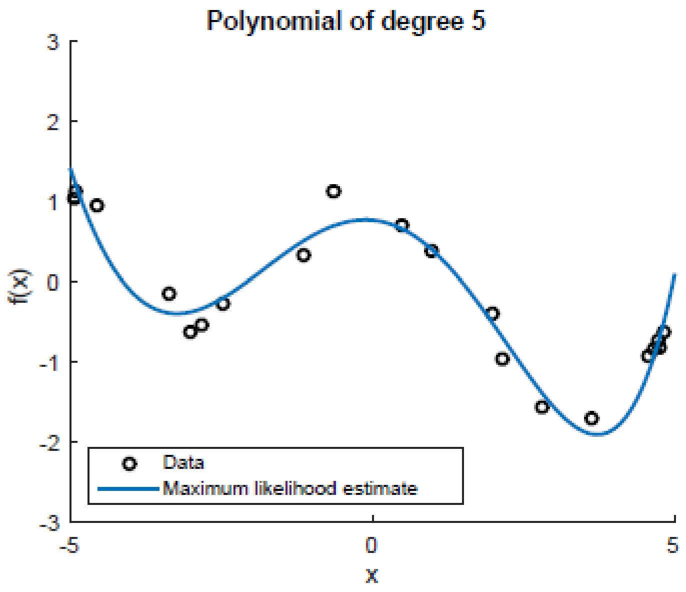
Vector Calculus

Curve Fitting (Regression) in Machine Learning



- Setting: Given inputs $\boldsymbol{x}$, predict outputs/targets $\boldsymbol{y}$
- Model f that depends on parameters $\boldsymbol{\theta}$. Examples:
  - Linear model: $f(\boldsymbol{x}, \boldsymbol{\theta}) = \boldsymbol{\theta}^\top \boldsymbol{x}, \quad \boldsymbol{x}, \boldsymbol{\theta} \in \mathbb{R}^D$
  - Neural network: $f(\boldsymbol{x}, \boldsymbol{\theta}) = NN(\boldsymbol{x}, \boldsymbol{\theta})$

- Training data, e.g., N pairs $(\boldsymbol{x}_i, y_i)$ of inputs $\boldsymbol{x}_i$ and observations y_i
- Training the model means finding parameters $\boldsymbol{\theta}^*$, such that $f(\boldsymbol{x}_i, \boldsymbol{\theta}^*) \approx y_i$



- Define a loss function, e.g., $\sum_{i=1}^{N}(y_i - f(\boldsymbol{x}_i, \boldsymbol{\theta}))^2$, which we want to optimize
- Typically: Optimization based on some form of gradient descent
  ▶▶ Differentiation required

Types of Differentiation

1. Scalar differentiation: $f : \mathbb{R} \to \mathbb{R}$

   $y \in \mathbb{R}$ w.r.t. $x \in \mathbb{R}$

2. Multivariate case: $f : \mathbb{R}^N \to \mathbb{R}$

   $y \in \mathbb{R}$ w.r.t. vector $\boldsymbol{x} \in \mathbb{R}^N$

3. Vector fields: $f : \mathbb{R}^N \to \mathbb{R}^M$

   vector $\boldsymbol{y} \in \mathbb{R}^M$ w.r.t. vector $\boldsymbol{x} \in \mathbb{R}^N$

4. General derivatives: $f : \mathbb{R}^{M \times N} \to \mathbb{R}^{P \times Q}$

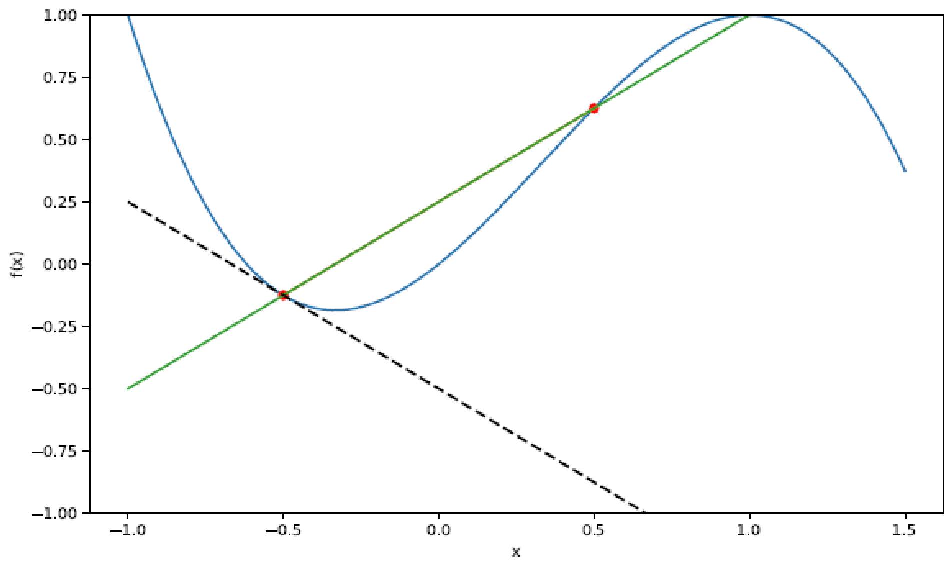
   matrix $\boldsymbol{y} \in \mathbb{R}^{P \times Q}$ w.r.t. matrix $\boldsymbol{X} \in \mathbb{R}^{M \times N}$

Scalar Differentiation $f : \mathbb{R} \to \mathbb{R}$

- Derivative defined as the limit of the difference quotient

$$f'(x) = \frac{df}{dx} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

▶▶ Slope of the secant line through $f(x)$ and $f(x+h)$



Some examples

$$
\begin{aligned}
f(x) &= x^n & \qquad f'(x) &= nx^{n-1} \\
f(x) &= \sin(x) & \qquad f'(x) &= \cos(x) \\
f(x) &= \tanh(x) & \qquad f'(x) &= 1 - \tanh^2(x) \\
f(x) &= \exp(x) & \qquad f'(x) &= \exp(x) \\
f(x) &= \log(x) & \qquad f'(x) &= \tfrac{1}{x}
\end{aligned}
$$

Rules

- Sum Rule

$$(f(x)+g(x))' = f'(x)+g'(x) = \frac{df(x)}{dx} + \frac{dg(x)}{dx}$$

- Product Rule

$$(f(x)g(x))' = f'(x)g(x) + f(x)g'(x) = \frac{df(x)}{dx}g(x) + f(x)\frac{dg(x)}{dx}$$

- Chain Rule

$$(g \circ f)'(x) = (g(f(x)))' = g'(f(x))f'(x) = \frac{dg(f(x))}{df}\frac{df(x)}{dx}$$

- Quotient Rule

$$\left(\frac{f(x)}{g(x)}\right)' = \frac{f(x)'g(x) - f(x)g(x)'}{(g(x))^2} = \frac{\frac{df}{dx}g(x) - f(x)\frac{dg}{dx}}{(g(x))^2}$$

Example: Scalar Chain Rule

$$(g \circ f)'(x) = (g(f(x)))' = g'(f(x))f'(x) = \frac{dg}{df}\frac{df}{dx}$$

$$\begin{aligned} g(z) &= 6z + 3 \\ z = f(x) &= -2x + 5 \\ (g \circ f)'(x) &= \underbrace{(6)}_{dg/df} \underbrace{(-2)}_{df/dx} \\ &= -12 \end{aligned}$$

Multivariate Differentiation $f : R^N \rightarrow R$

$$y = f(\boldsymbol{x}), \quad \boldsymbol{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} \in \mathbb{R}^N$$

- Partial derivative (change one coordinate at a time):

$$\frac{\partial f}{\partial x_i} = \lim_{h \to 0} \frac{f(x_1, \ldots, x_{i-1}, x_i + h, x_{i+1}, \ldots, x_N) - f(\boldsymbol{x})}{h}$$

- Jacobian vector (gradient) collects all partial derivatives:

$$\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{x}} = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \cdots & \frac{\partial f}{\partial x_N} \end{bmatrix} \in \mathbb{R}^{1 \times N}$$

Note: This is a row vector.

Example: Multivariate Differentiation

$$f : \mathbb{R}^2 \to \mathbb{R}$$

$$f(x_1, x_2) = x_1^2 x_2 + x_1 x_2^3 \in \mathbb{R}$$

Partial derivatives

$$\frac{\partial f(x_1, x_2)}{\partial x_1} = 2x_1 x_2 + x_2^3$$

$$\frac{\partial f(x_1, x_2)}{\partial x_2} = x_1^2 + 3x_1 x_2^2$$

Gradient $\quad \frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{x}} = \begin{bmatrix} \frac{\partial f(x_1,x_2)}{\partial x_1} & \frac{\partial f(x_1,x_2)}{\partial x_2} \end{bmatrix} \in \mathbb{R}^{1\times 2}$

$$\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{x}} = \begin{bmatrix} 2x_1 x_2 + x_2^3 & x_1^2 + 3x_1 x_2^2 \end{bmatrix}$$

Vector Field Differentiation $f : R^N \rightarrow R^M$

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) \in \mathbb{R}^M , \quad \boldsymbol{x} \in \mathbb{R}^N$$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} f_1(\boldsymbol{x}) \\ \vdots \\ f_M(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} f_1(x_1, \ldots, x_N) \\ \vdots \\ f_M(x_1, \ldots, x_N) \end{bmatrix}$$

- Jacobian matrix (collection of all partial derivatives)

$$\begin{bmatrix} \frac{dy_1}{d\boldsymbol{x}} \\ \vdots \\ \frac{dy_M}{d\boldsymbol{x}} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_N} \\ \vdots & & \vdots \\ \frac{\partial f_M}{\partial x_1} & \cdots & \frac{\partial f_M}{\partial x_N} \end{bmatrix} \in \mathbb{R}^{M \times N}$$

Example: Vector Field Differentiation

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x}\,, \qquad \boldsymbol{f}(\boldsymbol{x}) \in \mathbb{R}^M, \quad \boldsymbol{A} \in \mathbb{R}^{M\times N}, \quad \boldsymbol{x} \in \mathbb{R}^N$$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} f_1(\boldsymbol{x}) \\ \vdots \\ f_M(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} A_{11}x_1 + A_{12}x_2 + & \cdots & + A_{1N}x_N \\ \vdots \quad\quad \vdots & \vdots & \vdots \\ A_{M1}x_1 + A_{M2}x_2 + & \cdots & + A_{MN}x_N \end{bmatrix}$$

- Compute the gradient $\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}}$
  - Gradient:

$$f_i(\boldsymbol{x}) = \sum_{j=1}^{N} A_{ij}x_j \qquad \Longrightarrow \quad \frac{\partial f_i}{\partial x_j} = A_{ij}$$

$$\Longrightarrow \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_N} \\ \vdots & & \vdots \\ \frac{\partial f_M}{\partial x_1} & \cdots & \frac{\partial f_M}{\partial x_N} \end{bmatrix} = \begin{bmatrix} A_{11} & \cdots & A_{1N} \\ \vdots & & \vdots \\ A_{M1} & \cdots & A_{MN} \end{bmatrix} = \boldsymbol{A} \in \mathbb{R}^{M\times N}$$

Dimensionality of the Gradient

In general: A function $f : \mathbb{R}^N \to \mathbb{R}^M$ has a gradient that is an $M \times N$-matrix with

$$\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}} \in \mathbb{R}^{M \times N}, \qquad \mathrm{d}\boldsymbol{f}[m,n] = \frac{\partial f_m}{\partial x_n}$$

Gradient dimension: # target dimensions $\times$ # input dimensions

Chain Rule

$$\frac{\partial}{\partial \boldsymbol{x}}(g \circ f)(\boldsymbol{x}) = \frac{\partial}{\partial \boldsymbol{x}}\big(g(f(\boldsymbol{x}))\big) = \frac{\partial g(f)}{\partial f}\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}}$$

Example: Chain Rule

- Consider $f : \mathbb{R}^2 \to \mathbb{R}, \quad \boldsymbol{x} : \mathbb{R} \to \mathbb{R}^2$

$$f(\boldsymbol{x}) = f(x_1, x_2) = x_1^2 + 2x_2\,,$$

$$\boldsymbol{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} \sin(t) \\ \cos(t) \end{bmatrix}$$

- What are the dimensions of $\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{x}}$ and $\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t}$?

1×2 and 2×1

- Compute the gradient $\frac{\mathrm{d}f}{\mathrm{d}t}$ using the chain rule:

$$\frac{\mathrm{d}f}{\mathrm{d}t} = \frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{x}}\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} \end{bmatrix} \begin{bmatrix} \frac{\partial x_1}{\partial t} \\ \frac{\partial x_2}{\partial t} \end{bmatrix} = \begin{bmatrix} 2\sin t & 2 \end{bmatrix} \begin{bmatrix} \cos t \\ -\sin t \end{bmatrix}$$

$$= 2\sin t \cos t - 2\sin t = 2\sin t(\cos t - 1)$$

Derivatives with Respect to Matrices

- Recall: A function $f : \mathbb{R}^N \to \mathbb{R}^M$ has a gradient that is an $M \times N$-matrix with

$$\frac{\mathrm{d}f}{\mathrm{d}x} \in \mathbb{R}^{M\times N}, \qquad \mathrm{d}f[m,n] = \frac{\partial f_m}{\partial x_n}$$

Gradient dimension: # target dimensions $\times$ # input dimensions

- This generalizes to when the inputs (N) or targets (M) are **matrices**
- Function $f : \mathbb{R}^{M\times N} \to \mathbb{R}^{P\times Q}$, has a gradient that is a $(P\times Q) \times (M\times N)$ object (tensor)

$$\frac{\mathrm{d}f}{\mathrm{d}X} \in \mathbb{R}^{(P\times Q)\times(M\times N)}, \qquad \mathrm{d}f[p,q,m,n] = \frac{\partial f_{pq}}{\partial X_{mn}}$$

Additional Reading:
Mathematics for Machine Learning – Chapters 4, 5, 10