Week 1

Introduction to Geometric Data Analysis

What is Geometric Data Analysis?

Geometric Data Analysis (GDA) uses tools and intuition from geometry, topology, and linear algebra to analyze high-dimensional data. A guiding principle in GDA is that *data has shape*, and this shape can be used to extract meaningful information. GDA uses the shape of data to extract meaningful information for visualization and machine learning. Applications of the information extracted using geometric techniques include:

- **Dimensionality Reduction**: Data is often high-dimensional. Dimensionality reduction techniques provide tools for projecting high dimensional data onto fewer dimensions while preserving as much information as possible.

  The kind of information we wish to preserve is what leads to different dimensionality reduction techniques. The dimensionality reduction techniques belonging to GDA are those that aim to preserve the geometry of the original dataset.

  Dimensionality reduction serves many purposes:

  - **Visualization**: Reducing data down to two or three dimensions make it possible to create plots for visualization.
  - **Compression**: With compression, we reduce the dimension of the data down to a manageable size for storage or transmission, with the goal of being able to reconstruct the original data as faithfully as possible.
  - **Exploratory Data Analysis**: Dimensionality reduction can simplify data, which can make it easier to uncover patterns and relationships in the data that may be difficult to detect in the raw dataset.
  - **Machine Learning**: Dimensionality reduction can be used to engineer features for supervised learning or can reveal patterns in the data, and hence be regarded as a form of unsupervised learning.

- **Unsupervised Learning**: Unsupervised learning is a framework in machine learning that attempts to find patterns in data without relying on prior context or labels.

  An example is **clustering**, where data points are organized into clusters in such a way that points in the same cluster are in some sense more similar to one another than they are to points in different clusters. Clustering techniques often rely on the geometry of the data, and thus such techniques belong to GDA. For instance, we often cluster data based on distances between or density of the data points.

Dimensionality reduction techniques can also be employed as a form of unsupervised learning, since they tend to reduce dimension by finding patterns inherent in the data. For example, **multidimensional scaling (MDS)** reduces the dimensional of data while attempting to preserve distances between data points as much as possible, often revealing clusters in the process.

- **Supervised Learning**: In supervised learning, a model is trained on labeled data and then this model is used to predict the labels of unseen data.

  Examples of supervised learning are classification and regression. In classification, each data point in the training set belongs to a specific class, and the model has access to the class labels for each data point. Our goal is to use this information to build a model that can predict the class label of new, unlabeled data points. In regression, each point in the data set is assigned a value, and our goal is to use this information to build a model to predict this value for unseen data points.

  Geometry can be used to build supervised learning models for classification. For example, **support vector machines (SVM)** use the geometry of a dataset to find a hyperplane that separate the data into different classes. Similarly, **linear discriminant analysis (LDA)** finds a separating hyperplane to perform classification (though it finds the optimal hyperplane through statistical considerations rather than just geometry).

- **Feature Engineering** Feature engineering is the process of transforming data into simpler representative features that can then be used as the input to other machine learning models. For example, in an image dataset, the average pixel intensity of each image could serve as a feature. The goal is to extract features that better represent the data with fewer variables. Engineering good features can improve model performance and reduce overfitting.

  In GDA, an example of feature engineering is **persistent homology**, which is a technique from topological data analysis (TDA). Persistent homology summarizes the shape of a dataset into a *persistence diagram*, which can then serve as input for dimensionality reduction, supervised learning, or unsupervised learning algorithms.

  Often, features extracted using more than one method are combined before being passed to machine learning algorithms. This allows us to incorporate geometric and topological information into more traditional machine learning pipelines.

Geometric techniques have been to applied to every type of data. While GDA techniques are especially suited to data sets where geometry plays an obvious role, they can be applied to almost any data set to uncover patterns that aren't immediately apparent. Examples of applications include:

- **Image processing**: Principal components analysis can be used for image compression.

- **Natural language processing**: Text documents are often represented as vectors which count the number of occurrences of each word. NLP algorithms often use dimensionality reduction techniques to embed these representations into lower dimensional spaces.

- **Medical imaging**: Techniques from TDA have been used to classify cancer cell type and were used to discover a new prostate cancer subtype.

Linear Algebra Review

Much of GDA relies on the language and tools of linear algebra. As a starting point for GDA, we usually assume that our data lies in some high dimensional vector space, typically a Euclidean space $\mathbb{R}^p$. Each data point is a vector in this space, and we refer to the whole data set collectively as a **point cloud**. We can use linear algebra to uncover structure in this point cloud.

I'll assume you're familiar with vectors, matrices, linear combinations, bases, and the range and nullspace/kernel of a matrix (though this is not necessarily an exhaustive list of concepts we might need later).

We'll review the following concepts here as they'll be crucial later: the dot product, vector and matrix norms, orthogonality, eigenvalues, eigenvectors, and the eigendecomposition. All of this builds up to the singular value decomposition (SVD), which we will cover next week.

Conventions

We will denote vectors by lowercase, bold-faced letters such as $\mathbf{x}, \mathbf{u}, \mathbf{s}$, etc. We'll use uppercase, bold-faced letters for matrices, e.g., $\mathbf{A}$, $\mathbf{X}$, $\mathbf{M}$, etc. The set of all $m \times n$ matrices with real entries is denoted $\mathbb{R}^{m \times n}$. Vectors will always be regarded as *column vectors*: that is, an n-dimensional vector $\mathbf{x}$ is the same thing as an $n \times 1$ matrix

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \tag{1.1}$$

We'll denote the set of all n-dimensional vectors with real entries by $\mathbb{R}^n$. Since all our vectors are assumed to be column vectors, $\mathbb{R}^n$ is the same thing as $\mathbb{R}^{n \times 1}$.

Recall that the **transpose** of a matrix interchanges its rows with its columns. For example

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \quad \Longrightarrow \quad A^T = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}.$$

The transpose of a column vector is a row vector, and vice versa. For this reason, we'll often denote a column vector as the transpose of a row vector, e.g., $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^T$ is the same vector as (1.1) (this is essentially just to save vertical space on the page). Note that, when regarding vectors as matrices, the vector $\begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}$ is a different vector from $\mathbf{x}$, since the former is $1 \times n$ and the latter is $n \times 1$.

The Dot Product

Let $\mathbf{x} = \begin{bmatrix} x_1 & \dots & x_n \end{bmatrix}^T$ and $\mathbf{y} = \begin{bmatrix} y_1 & \cdots & y_n \end{bmatrix}^T$ be vectors in $\mathbb{R}^n$. The **dot product** (or **inner product**) of $\mathbf{x}$ and $\mathbf{y}$ is defined by

$$\mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^{n} x_i y_i = x_1 y_1 + x_2 y_2 + \cdots + x_n y_n.$$

In matrix notation $\mathbf{x} \cdot \mathbf{y} = \mathbf{x}^T \mathbf{y}$.

Vector Norms

Norms are used to quantity the *magnitude* (or *size* or *length*) of a vector. There are many different norms which measure magnitude in different ways. We'll review the most common ones.

The **Euclidean norm** (or 2-**norm**) of a vector $\mathbf{x} = \begin{bmatrix} x_1 & \dots & x_n \end{bmatrix}^T$ is defined by

$$\|\mathbf{x}\|_2 = \left(\sum_{i=1}^{n} x_i^2 \right)^{\frac{1}{2}} = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}.$$

Note that $\mathbf{x} \cdot \mathbf{x} = x_1^2 + x_2^2 + \cdots + x_n^2 = \|\mathbf{x}\|_2^2$, i.e., the dot product of a vector with itself gives the square of the 2-norm of $\mathbf{x}$. Equivalently, we have $\|\mathbf{x}\|_2 = \sqrt{\mathbf{x} \cdot \mathbf{x}}$.

More generally, for $p \geq 1$, we define the p-**norm** of $\mathbf{x}$ by

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}}.$$

The Euclidean norm corresponds to using $p = 2$. Another common choice is $p = 1$, in which case we obtain the 1-**norm**, which reduces to

$$\|\mathbf{x}\|_1 = \sum_{i=1}^{n} |x_i| = |x_1| + |x_2| + \cdots + |x_n|.$$

Another common norm occurs by considering the limit $\lim_{p \to \infty} \|\mathbf{x}\|_p$. It can be shown that this limit is the maximum value of the absolute values of the entries of $\mathbf{x}$. This is called the ∞-**norm**, and can be expressed as

$$\|\mathbf{x}\|_\infty = \max_{1 \leq i \leq n} |x_i| = \max(|x_1|, |x_2|, \dots, |x_n|).$$

Example 1. Let $\mathbf{x} = \begin{bmatrix} 1 & -2 & 3 \end{bmatrix}^T \in \mathbb{R}^3$. Then

- $\|\mathbf{x}\|_2 = \sqrt{1^2 + (-2)^2 + 3^2} = \sqrt{14}$,
- $\|\mathbf{x}\|_1 = |1| + |-2| + |3| = 6$,
- $\|\mathbf{x}\|_\infty = \max(|1|, |-2|, |3|) = 3$.

A vector $\mathbf{x}$ is a *unit vector* (with respect to a given norm $\|\cdot\|$) if $\|\mathbf{x}\| = 1$. Note that the property of being a unit vector is norm-dependent. For example, the vector $\mathbf{x} = \begin{bmatrix}1 & 1\end{bmatrix}^T$ is a unit vector with respect to $\|\cdot\|_\infty$, since $\|\mathbf{x}\| = \max(|1|, |1|) = 1$, but it is not a unit vector with respect to $\|\cdot\|_2$, since $\|\mathbf{x}\|_2 = \sqrt{1^2 + 1^2} = \sqrt{2} \neq 1$. Given a vector $\mathbf{x}$, we can **normalize** $\mathbf{x}$ to obtain a unit vector $\hat{\mathbf{x}}$ by dividing $\mathbf{x}$ by its norm: $\hat{\mathbf{x}} = \frac{\mathbf{x}}{\|\mathbf{x}\|}$. If $\mathbf{x}$ is a unit vector with respect to the 2-norm then $\|\mathbf{x}\|_2 = \sqrt{\mathbf{x} \cdot \mathbf{x}} = 1$. This is the same thing as saying that $\mathbf{x} \cdot \mathbf{x} = 1$, i.e., a unit vector (with respect to the 2-norm) is a vector whose dot product with itself is 1.

Dot products and norms are related by the following formula: for vectors $\mathbf{x}$ and $\mathbf{y}$, we have

$$\mathbf{x} \cdot \mathbf{y} = \|\mathbf{x}\|_2 \|\mathbf{y}\|_2 \cos(\theta),$$

where θ denotes the angle between $\mathbf{x}$ and $\mathbf{y}$. Solving for $\cos(\theta)$ gives

$$\cos(\theta) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\|_2 \|\mathbf{y}\|_2} = \frac{\mathbf{x}}{\|\mathbf{x}\|_2} \cdot \frac{\mathbf{y}}{\|\mathbf{y}\|_2} = \hat{\mathbf{x}} \cdot \hat{\mathbf{y}},$$

i.e., the cosine of the angle between $\mathbf{x}$ and $\mathbf{y}$ is computed by normalizing both vectors and then computing the dot product of the result. Further, solving for θ using arccos, we obtain the following formula for θ:

$$\theta = \arccos\left(\frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\|_2 \|\mathbf{y}\|_2}\right). \tag{1.2}$$

Matrix Norms

Just as vector norms are a way to measure the magnitude of a vector, **matrix norms** give a way to measure the magnitude of a matrix.

Let $\mathbf{A} \in \mathbb{R}^{m \times n}$. For $p \geq 1$, the **matrix p-norm** is defined by

$$\|\mathbf{A}\|_p = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^p\right)^{1/p}.$$

In the special case $p = 2$, this is usually called the **Frobenius norm** and is denoted $\|\mathbf{A}\|_F$ instead of $\|\mathbf{A}\|_2$, i.e.,

$$\|\mathbf{A}\|_F = \|\mathbf{A}\|_2 = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}^2}.$$

The Frobenius norm is a very popular matrix norm. We'll make use of it next week when we discuss the singular value decomposition.

Orthogonality

Two vectors $\mathbf{x}$ and $\mathbf{y}$ are said to be *orthogonal* (synonymous with *perpendicular*) if $\mathbf{x} \cdot \mathbf{y} = 0$. By Formula (1.2), this means $\theta = \arccos(0) = \frac{\pi}{2}$ (or 90°). Thus two vectors are orthogonal if they form a right angle. We write $\mathbf{x} \perp \mathbf{y}$ to indicate that $\mathbf{x}$ and $\mathbf{y}$ are orthogonal. A set of vectors $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k\}$ is an *orthogonal set* if each pair of vectors in this set is orthogonal, i.e., if $\mathbf{u}_i \cdot \mathbf{u}_j$ for all $i \neq j$. This set is said to be an *orthonormal set* if it is an orthogonal set and each of the vectors $\mathbf{u}_i$ is a unit vector.

Let $\mathbf{U} \in \mathbb{R}^{n\times n}$ be a *square* matrix. $\mathbf{U}$ is said to be *orthogonal* if $\mathbf{U}^T\mathbf{U} = \mathbf{I}$, where $\mathbf{I}$ denotes the $n \times n$ identity matrix. Let's break this down: if we write

$$\mathbf{U} = \left[\begin{array}{c|c|c|c} \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_n \end{array}\right]$$

then

$$\mathbf{U}^T\mathbf{U} = \left[\begin{array}{c} \mathbf{u}_1^T \\ \hline \mathbf{u}_2^T \\ \hline \vdots \\ \hline \mathbf{u}_n^T \end{array}\right] \left[\begin{array}{c|c|c|c} \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_n \end{array}\right] = \mathbf{I}.$$

This means that

$$\mathbf{u}_i \cdot \mathbf{u}_j = \begin{cases} 1 & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases}.$$

Thus $\mathbf{u}_i \perp \mathbf{u}_j$ when $i \neq j$, i.e., each column of $\mathbf{U}$ is orthogonal to every other column, and each column is a unit vector (with respect to the 2-norm). Orthogonal matrices are therefore those square matrices whose columns form an *orthonormal set* of vectors. Orthogonal matrices are automatically invertible, with $\mathbf{U}^{-1} = \mathbf{U}^T$. It follows that we also automatically get $\mathbf{U}\mathbf{U}^T = \mathbf{I}$.

Sometimes, we will encounter an $n \times r$ matrix $\mathbf{U}$, with $r < n$, whose columns form an orthonormal set. This still means that $\mathbf{U}^T\mathbf{U} = \mathbf{I}_{r\times r}$, but it will generally not be the case that $\mathbf{U}^T\mathbf{U} = \mathbf{I}_{n\times n}$. We will call such matrices **semi-orthogonal**.

Example 2. The matrix $\mathbf{U} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ is orthogonal since

$$\mathbf{U}^T\mathbf{U} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The matrix $\mathbf{V} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \\ 0 & 0 \end{bmatrix}$ is semi-orthogonal since

$$\mathbf{V}^T\mathbf{V} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Note that $\mathbf{V}$ fails to be an orthogonal matrix because it is not square.

Eigenvalues and Eigenvectors

Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be an $n \times n$ square matrix. A vector $\mathbf{x} \in \mathbb{R}^n$ is an *eigenvector* for $\mathbf{A}$ if there is is a (possibly complex) number $\lambda \in \mathbb{C}$, called an *eigenvalue*, such that

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}.$$

If λ is an eigenvalue of $\mathbf{A}$ with corresponding eigenvalue $\mathbf{x}$, then $\mathbf{A}\mathbf{x} - \lambda\mathbf{x} = \mathbf{0}$, and hence $(\mathbf{A} - \lambda\mathbf{I})\mathbf{x} = \mathbf{0}$, where $\mathbf{I}$ denotes the $n \times n$ identity matrix. This means that $\mathbf{x}$ is in the *nullspace* of $\mathbf{A}-\lambda\mathbf{I}$. So to find the eigenvalues of A, we need to find the values of λ for which the matrix $\mathbf{A}-\lambda\mathbf{I}$ has nontrivial nullspace. For $\mathbf{A}-\lambda\mathbf{I}$ to have nontrivial nullspace, it is equivalent that $\mathbf{A}-\lambda\mathbf{I}$ be singular (i.e., non-invertible), and this is equivalent to having $\det(\mathbf{A}-\lambda\mathbf{I})=0$.

The determinant $\det(\mathbf{A}-\lambda\mathbf{I})$ is always a polynomial in λ and is called the *characteristic polynomial of* $\mathbf{A}$, denoted $p_{\mathbf{A}}(\lambda)$. Therefore, to find the eigenvalues of $\mathbf{A}$, we find λ for which $p_{\mathbf{A}}(\lambda)=0$, i.e., we find the roots of $p_{\mathbf{A}}$.

Example 3. Let $\mathbf{A}=\begin{bmatrix}-1 & 3\\ 3 & -1\end{bmatrix}$. The characteristic polynomial of $\mathbf{A}$ is

$$p_{\mathbf{A}}(\lambda)=\det(\mathbf{A}-\lambda I)=\begin{vmatrix}-1-\lambda & 3\\ 3 & -1-\lambda\end{vmatrix}=(-1-\lambda)^2-9=-8+2\lambda+\lambda^2.$$

This factors as $p_{\mathbf{A}}(\lambda)=(\lambda+4)(\lambda-2)$, so the eigenvalues of $\mathbf{A}$ are $\lambda_1=2$ and $\lambda_2=-4$.

Once we've found an eigenvalue λ, we can compute a corresponding eigenvector $\mathbf{x}$ by solving the linear system $(\mathbf{A}-\lambda\mathbf{I})\mathbf{x}=\mathbf{0}$ for $\mathbf{x}$. We can do this using *Gaussian elimination.*

Let's find an eigenvector corresponding to $\lambda_1=2$. We want to solve the system $(\mathbf{A}-2\mathbf{I})\mathbf{x}=\mathbf{0}$. To solve this system, we set up the *augmented matrix*

$$\left[\begin{array}{c|c}\mathbf{A}-2\mathbf{I} & \mathbf{0}\end{array}\right]=\left[\begin{array}{cc|c}-3 & 3 & 0\\ 3 & -3 & 0\end{array}\right].$$

We now perform *elementary row operations* to get the augmented matrix into *row-reduced echelon form*:

$$\left[\begin{array}{cc|c}-3 & 3 & 0\\ 3 & -3 & 0\end{array}\right]\xrightarrow[\frac{1}{3}R_2]{-\frac{1}{3}R_1}\left[\begin{array}{cc|c}1 & -1 & 0\\ 1 & -1 & 0\end{array}\right]\xrightarrow{R_2-R_1}\left[\begin{array}{cc|c}\boxed{1} & -1 & 0\\ 0 & 0 & 0\end{array}\right]$$

The first row of the reduced matrix tells us that $x_1-x_2=0$, i.e., $x_1=x_2$. Thus an eigenvector for $\lambda_1=2$ has the form

$$\mathbf{x}=\begin{bmatrix}x_1\\ x_2\end{bmatrix}=\begin{bmatrix}x_2\\ x_2\end{bmatrix}=x_2\begin{bmatrix}1\\ 1\end{bmatrix}.$$

The variable x_2 is a *free variable*, and so we are free to choose it. Taking $x_2=1$, we get the eigenvector

$$\mathbf{x}_1=\begin{bmatrix}1\\ 1\end{bmatrix}$$

corresponding to the eigenvalue $\lambda_1=2$.

We can verify that this is indeed an eigenvector for λ_1 by checking that $\mathbf{A}\mathbf{x}_1=2\mathbf{x}_1$. We have

$$\mathbf{A}\mathbf{x}_1=\begin{bmatrix}-1 & 3\\ 3 & -1\end{bmatrix}\begin{bmatrix}1\\ 1\end{bmatrix}=\begin{bmatrix}2\\ 2\end{bmatrix}=2\begin{bmatrix}1\\ 1\end{bmatrix}=2\mathbf{x}_1.$$

Eigendecomposition

Let $\mathbf{A}\in\mathbb{R}^{n\times n}$ is a square matrix. Suppose that $\mathbf{A}$ has n linearly independent eigenvectors $\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n$ (for example, this happens when $\mathbf{A}$ has n distinct eigenvalues, but can happen even if this is not the case). Let

$$\mathbf{X}=\left[\begin{array}{c|c|c|c}\mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_n\end{array}\right]$$

be the matrix whose columns are the eigenvectors of A. Then, using properties of matrix multiplication and the definition of eigenvectors and eigenvalues, we have

$$\begin{aligned}\mathbf{AX} &= \mathbf{A}\left[\begin{array}{c|c|c|c} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_n \end{array}\right] \\ &= \left[\begin{array}{c|c|c|c} \mathbf{Ax}_1 & \mathbf{Ax}_2 & \cdots & \mathbf{Ax}_n \end{array}\right] \\ &= \left[\begin{array}{c|c|c|c} \lambda_1\mathbf{x}_1 & \lambda_2\mathbf{x}_2 & \cdots & \lambda_n\mathbf{x}_n \end{array}\right] = \mathbf{X\Lambda}\end{aligned}$$

where $\mathbf{\Lambda}$ is the diagonal matrix

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix}.$$

Since we assumed that the eigenvectors of $\mathbf{A}$ are linearly independent, $\mathbf{X}$ is invertible. Thus, we can solve for $\mathbf{A}$ by inverting $\mathbf{X}$ to obtain

$$\mathbf{A} = \mathbf{XAX}^{-1}. \tag{1.3}$$

This is called the **eigendecomposition** of $\mathbf{A}$ (since we are *decomposing* $\mathbf{A}$ in terms of its eigenvalues and eigenvectors). A matrix with an eigendecomposition is also called *diagonalizable*, since (1.3) says that, after a change of basis, $\mathbf{A}$ is represented by the diagonal matrix $\mathbf{\Lambda}$. We will use the eigendecomposition (and related singular value decomposition) frequently.

Not every matrix admits an eigendecomposition. The following proposition tells us exactly when a matrix admits an eigendecomposition:

Proposition 4. *A square matrix* $\mathbf{A}$ *admits an eigendecomposition* $\mathbf{A} = \mathbf{X\Lambda X}^{-1}$ *if and only if* $\mathbf{A}$ *has n linearly independent eigenvectors.*

Example 5. Let's show that the matrix

$$\mathbf{A} = \begin{bmatrix} -1 & 3 \\ 3 & -1 \end{bmatrix}$$

from Example 3 has an eigendecomposition and then compute it. We computed the eigenvalues to be $\lambda_1 = 2$ and $\lambda_2 = -4$.

We computed an eigenvector for λ_1 to be $\mathbf{x_1} = \begin{bmatrix} 1 & 1 \end{bmatrix}^T$. You can check for yourself that an eigenvector for λ_2 is $\mathbf{x}_2 = \begin{bmatrix} 1 & -1 \end{bmatrix}^T$.

From the Proposition above, $\mathbf{A}$ admits an eigendecomposition if and only $\{\mathbf{x}_1, \mathbf{x}_2\}$ is a linearly independent set. To check this, we form the matrix

$$\mathbf{X} = \left[\begin{array}{c|c} \mathbf{x}_1 & \mathbf{x}_2 \end{array}\right] = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

The eigenvectors are linearly independent if and only if the matrix $\mathbf{X}$ is invertible. To see that $\mathbf{X}$ is invertible, note that $\det(\mathbf{X}) = -2 \neq 0$. Thus our matrix $\mathbf{A}$ does indeed have an eigendecomposition.

The matrix $\mathbf{\Lambda}$ is then the diagonal matrix

$$\mathbf{\Lambda} = \begin{bmatrix} 2 & 0 \\ 0 & -4 \end{bmatrix}$$

with the eigenvalues on the diagonal.

Finally, we need to invert $\mathbf{X}$, which we can do by using the formula for the inverse of an invertible 2×2 matrix:

$$\mathbf{X}^{-1} = \frac{1}{\det(\mathbf{X})} \begin{bmatrix} -1 & -1 \\ -1 & 1 \end{bmatrix} = \frac{1}{-2} \begin{bmatrix} -1 & -1 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 \\ 1/2 & -1/2 \end{bmatrix}.$$

Thus, an eigendecomposition of $\mathbf{A}$ is

$$\mathbf{A} = \mathbf{X}\mathbf{\Lambda}\mathbf{X}^{-1} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & -4 \end{bmatrix} \begin{bmatrix} 1/2 & 1/2 \\ 1/2 & -1/2 \end{bmatrix}.$$

A problem is that matrices typically don't have n linearly independent eigenvectors, and so typically do not admit an eigendecomposition. We get around this by considering the *Gram matrix*:

Definition 6. Let $\mathbf{A}$ be an $m \times n$ matrix. The **Gram matrix** for $\mathbf{A}$ is the $n \times n$ matrix

$$\mathbf{G} = \mathbf{A}^T\mathbf{A}.$$

The following theorem tells us that the Gram matrix always admits an eigendecomposition, and its eigendecomposition is particularly nice:

Theorem 7. *Let $\mathbf{A}$ be a real $m \times n$ matrix and let $\mathbf{G} = \mathbf{A}^T\mathbf{A}$ be the corresponding Gram matrix. Then $\mathbf{G}$ has real, non-negative eigenvalues and the eigenvectors of $\mathbf{G}$ form an orthonormal basis for $\mathbb{R}^n$. Therefore, $\mathbf{G}$ admits an eigendecomposition of the form*

$$\mathbf{G} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T,$$

where $\mathbf{V}$ is an $n \times n$ orthogonal matrix and $\mathbf{\Lambda}$ is an $n \times n$ diagonal matrix with diagonal entries $\lambda_i \geq 0$.

This theorem tells us the eigenvectors of a Gram matrix $\mathbf{G}$ (i.e., the columns of $\mathbf{V}$) form an *orthonormal set*, and that the corresponding eigenvalues are all non-negative. It also tells us that the eigenvalues are real numbers and are non-negative. We will always order the eigenvalues of the Gram matrix in descending order $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$. This will be important in applications, where the eigenvalues will give a measure of importance of the corresponding eigenvectors.

Example 8. Let's compute the eigendecomposition of the Gram matrix for the matrix

$$\mathbf{A} = \begin{bmatrix} -1 & 3 \\ 3 & 1 \end{bmatrix}$$

from the previous examples. The Gram matrix for $\mathbf{A}$ is

$$\mathbf{G} = \mathbf{A}^T\mathbf{A} = \begin{bmatrix} -1 & 3 \\ 3 & -1 \end{bmatrix} \begin{bmatrix} -1 & 3 \\ 3 & -1 \end{bmatrix} = \begin{bmatrix} 10 & -6 \\ -6 & 10 \end{bmatrix}.$$

The characteristic polynomial of $\mathbf{G}$ is

$$p_{\mathbf{G}} = \det(\mathbf{G} - \lambda\mathbf{I}) = (10-\lambda)^2 - 36 = \lambda^2 - 20\lambda + 64 = (\lambda - 16)(\lambda - 4).$$

Thus the eigenvalues of $\mathbf{G}$ (ordered from largest to smallest) are $\lambda_1 = 16$ and $\lambda_2 = 4$. You can check for yourself that corresponding eigenvectors are $\mathbf{x}_1 = \begin{bmatrix} 1 & -1 \end{bmatrix}^T$ and $\mathbf{x}_2 = \begin{bmatrix} 1 & 1 \end{bmatrix}^T$.

The eigenvectors are orthogonal, but to obtain an orthonormal set of eigenvectors as in Theorem 7, we need to normalize them. We have

$$\mathbf{v}_1 = \hat{\mathbf{x}}_1 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}, \quad \mathbf{v}_2 = \hat{\mathbf{x}}_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}.$$

Therefore,

$$\mathbf{V} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}.$$

Note that in the case of a Gram matrix, we do not need to compute $\mathbf{V}^{-1}$ to get the eigendecomposition since we know that $\mathbf{V}$ is orthogonal and hence $\mathbf{V}^{-1} = \mathbf{V}^T$. Therefore, an eigendecomposition for $\mathbf{G}$ is

$$\mathbf{G} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 16 & 0 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}.$$

As guaranteed by Theorem 7, the eigenvalues of $\mathbf{G}$ are all non-negative, and the normalized eigenvectors form an orthonormal set.