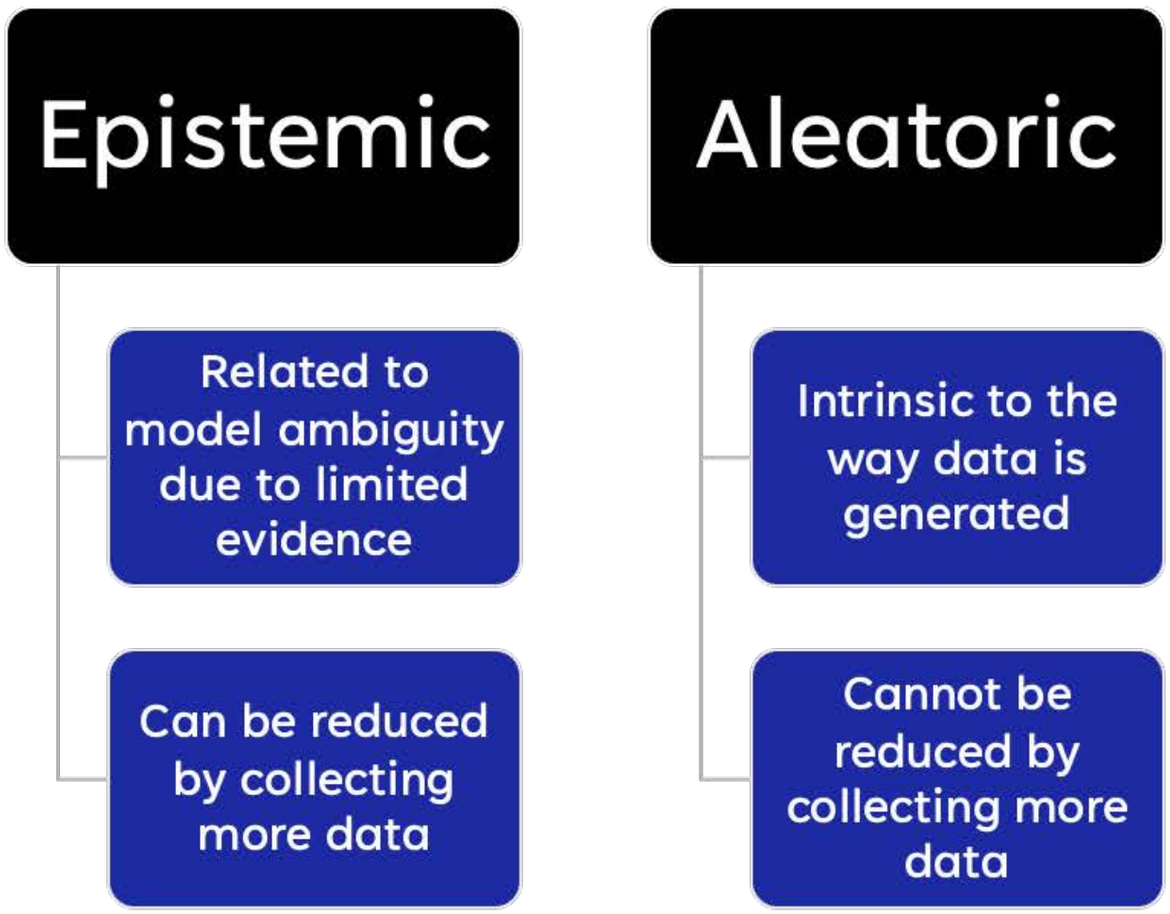
ENM 3600: Data-driven Modeling

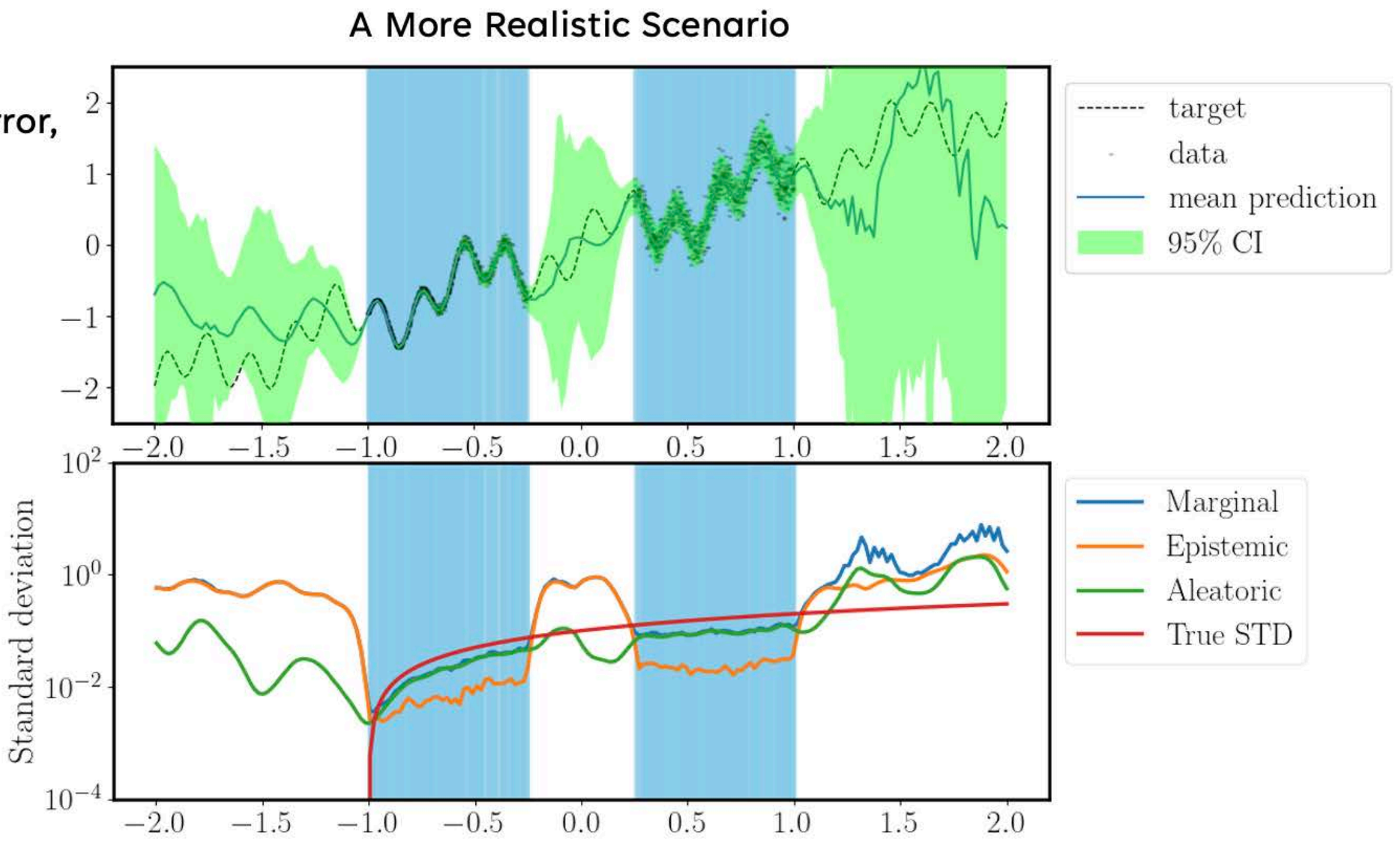
Lecture #19: Principal components analysis

Epistemic vs Aleatoric Uncertainty



Epistemic vs Aleatoric Uncertainty

High Epistemic Error,
Heteroscedastic
Aleatoric Error



A Taxonomy of UQ Tasks

Aleatory

$$\mathcal{D} = \{x_1, x_2, \ldots, x_n\} \mapsto p(x)$$

$$\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \mapsto p(y|x)$$

$$\mathcal{M} : \mathcal{X} \to \mathcal{Y}, \quad x \sim p(x) \mapsto p(\mathcal{M}(x)|x)$$

Epistemic

$$\mathcal{M}_\theta : \mathcal{X} \to \mathcal{Y}, \quad \mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \mapsto p(\theta|\mathcal{D})$$

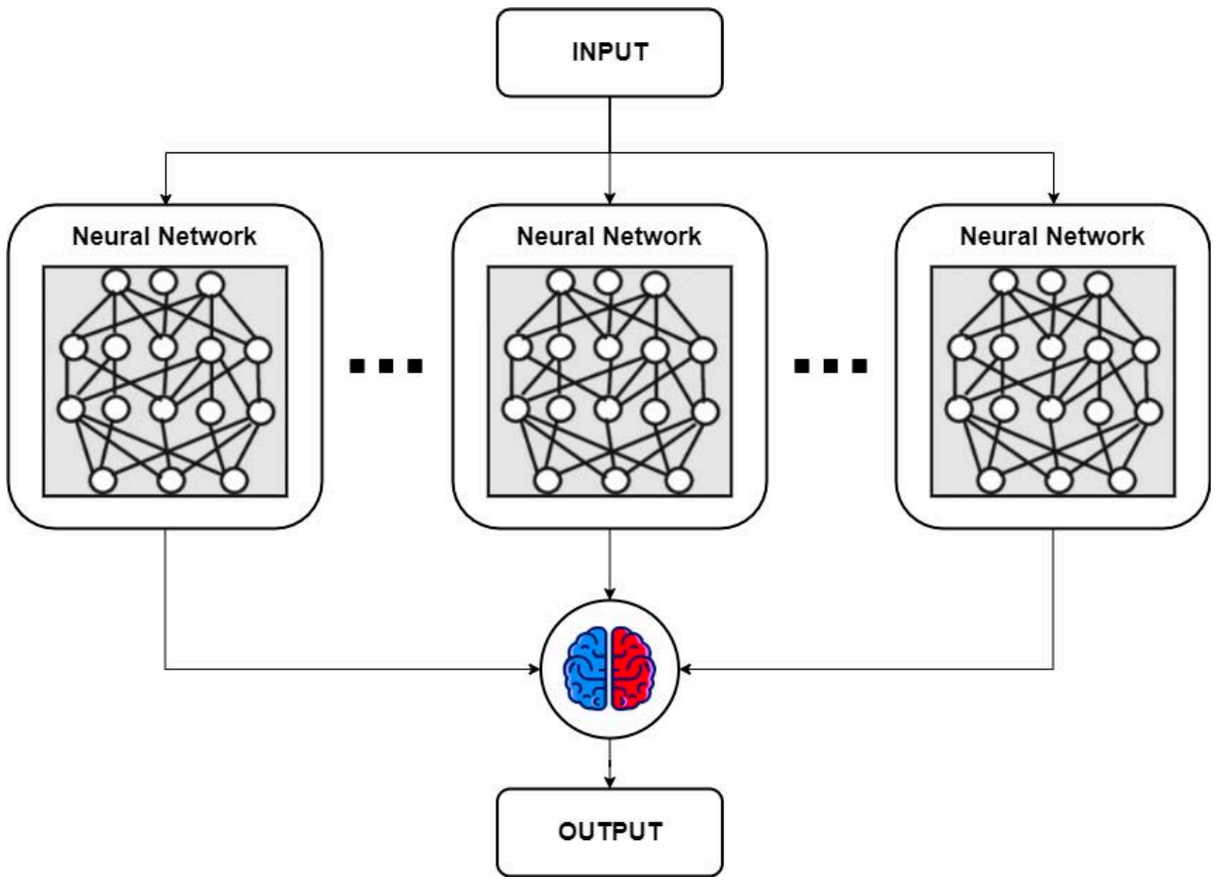
$$\mathcal{M}_\theta : \mathcal{X} \to \mathcal{Y}, \quad \mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \mapsto p(\mathcal{M}_\theta(x^*)|\mathcal{D}, x^*)$$

$$= \int p(\mathcal{M}_\theta|\mathcal{D}, x^*, \theta)\; p(\theta|\mathcal{D})\; d\theta$$

A need for robustness and uncertainty quantification

Becomes particularly important when:

- We are working with small data-sets (over-fitting regime).
- We need to make high-consequence decisions.
- We require performance/accuracy guarantees.
- We work under a limited budget.

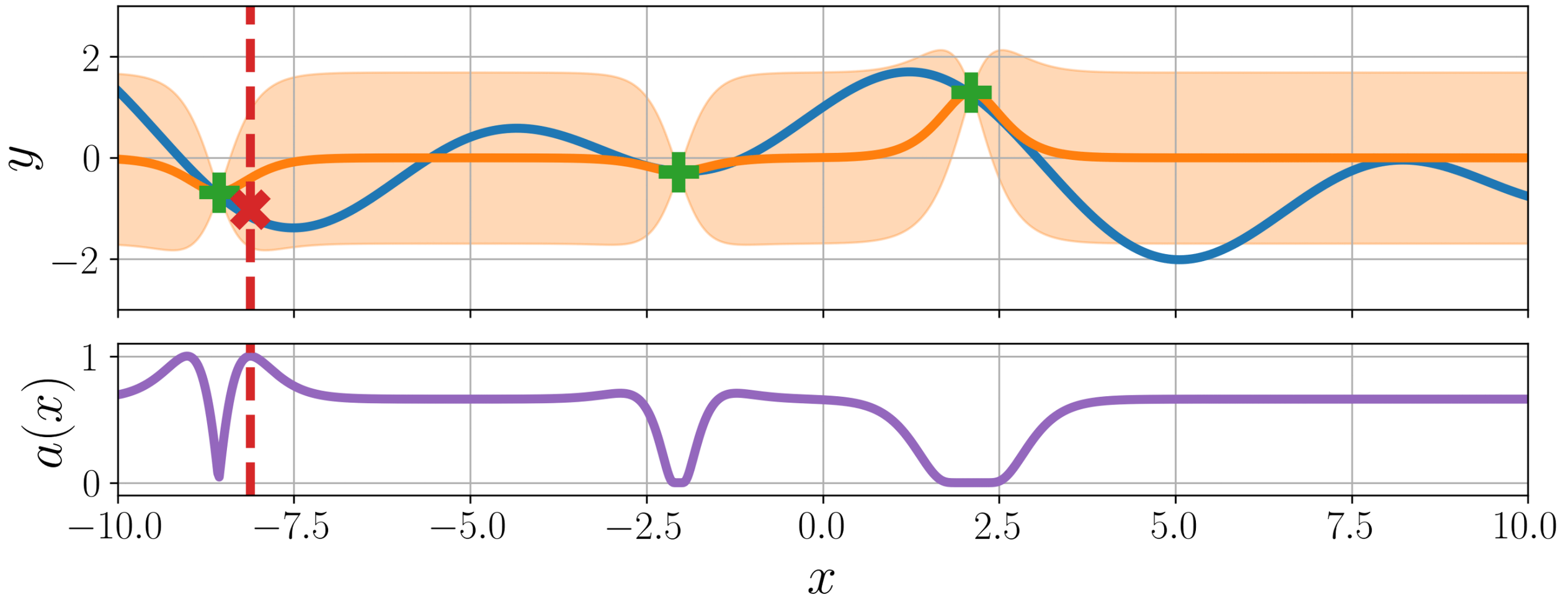


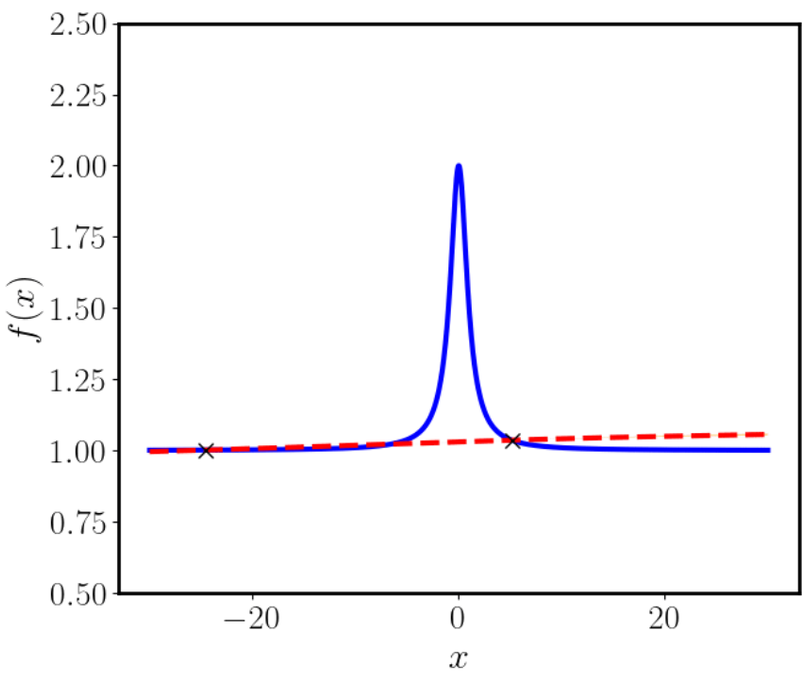
$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})} = \frac{p(\mathcal{D}|\theta)p(\theta)}{\int p(\mathcal{D}|\theta)p(\theta)d\theta}$$

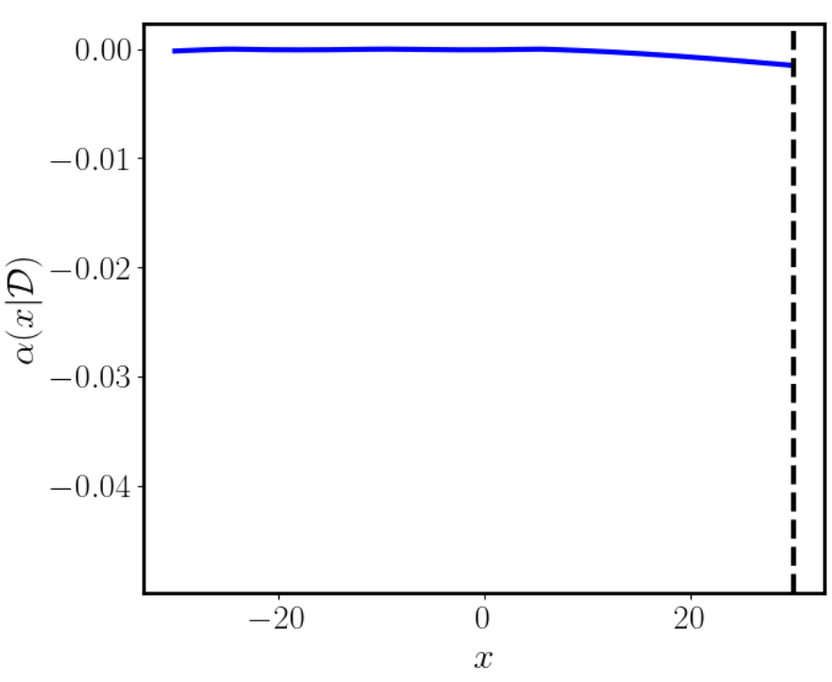
The frequentist approach:
Ensemble averaging

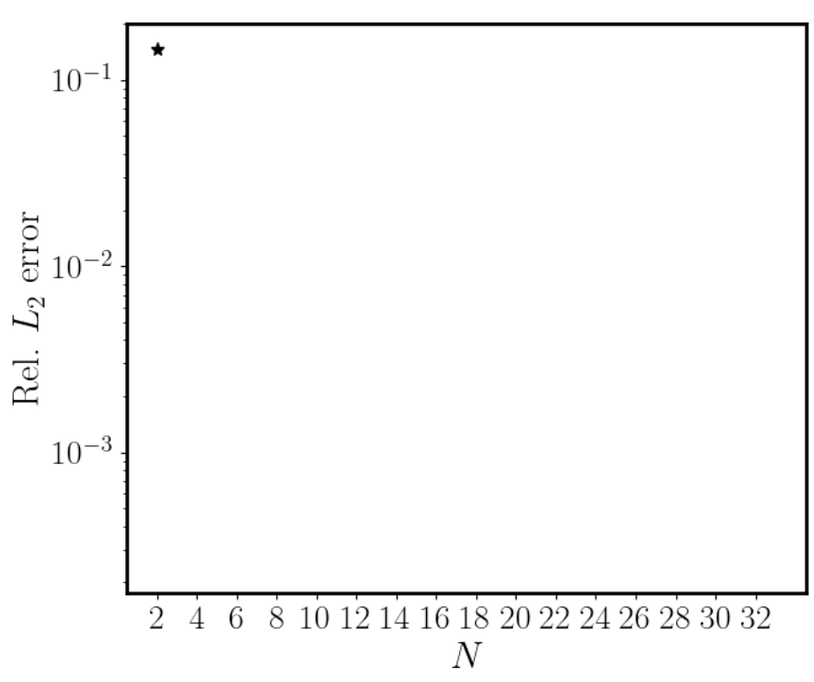
The Bayesian approach:
Probabilistic programming

Active Learning & Bayesian Optimization

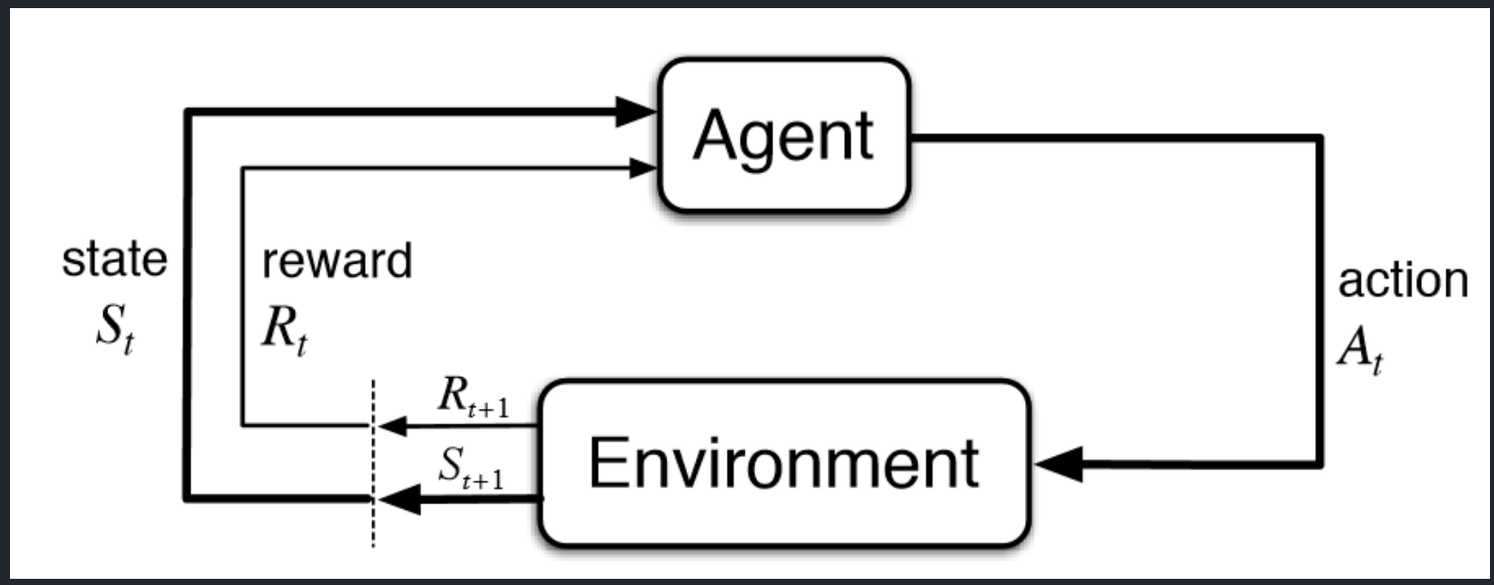


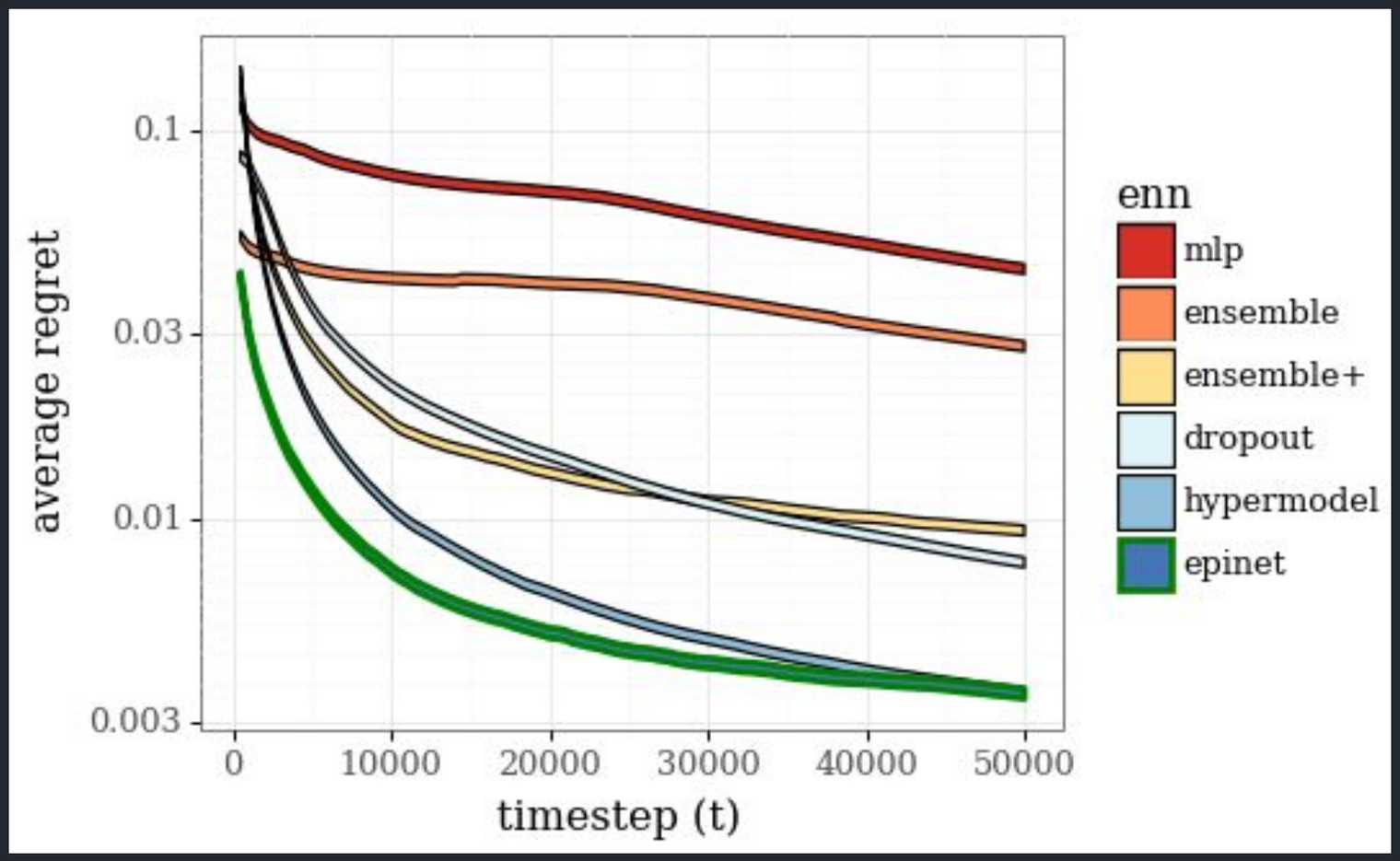






Accurate UQ leads to Good Decisions





Osband, I., Wen, Z., Asghari, S. M., Dwaracherla, V., Lu, X., Ibrahimi, M., ... & Van Roy, B. (2022). The neural testbed: Evaluating joint predictions. Advances in Neural Information Processing Systems,

A Taxonomy of UQ Tasks

Aleatory

$$\mathcal{D} = \{x_1, x_2, \ldots, x_n\} \mapsto p(x)$$

$$\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \mapsto p(y|x)$$

$$\mathcal{M} : \mathcal{X} \to \mathcal{Y}, \quad x \sim p(x) \mapsto p(\mathcal{M}(x)|x)$$

Epistemic

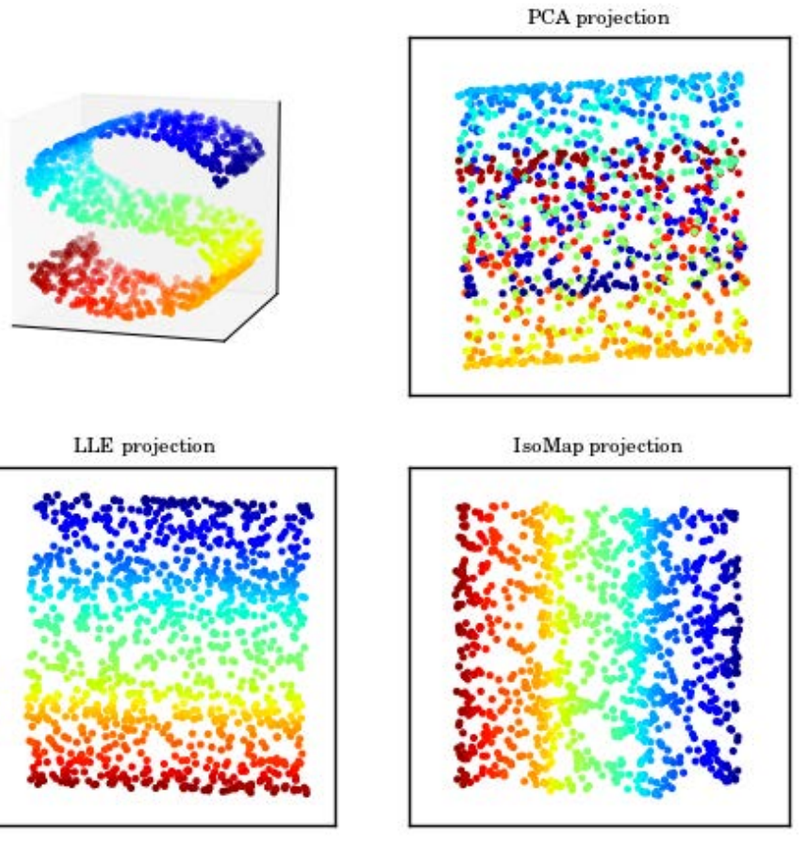
$$\mathcal{M}_\theta : \mathcal{X} \to \mathcal{Y}, \quad \mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \mapsto p(\theta|\mathcal{D})$$

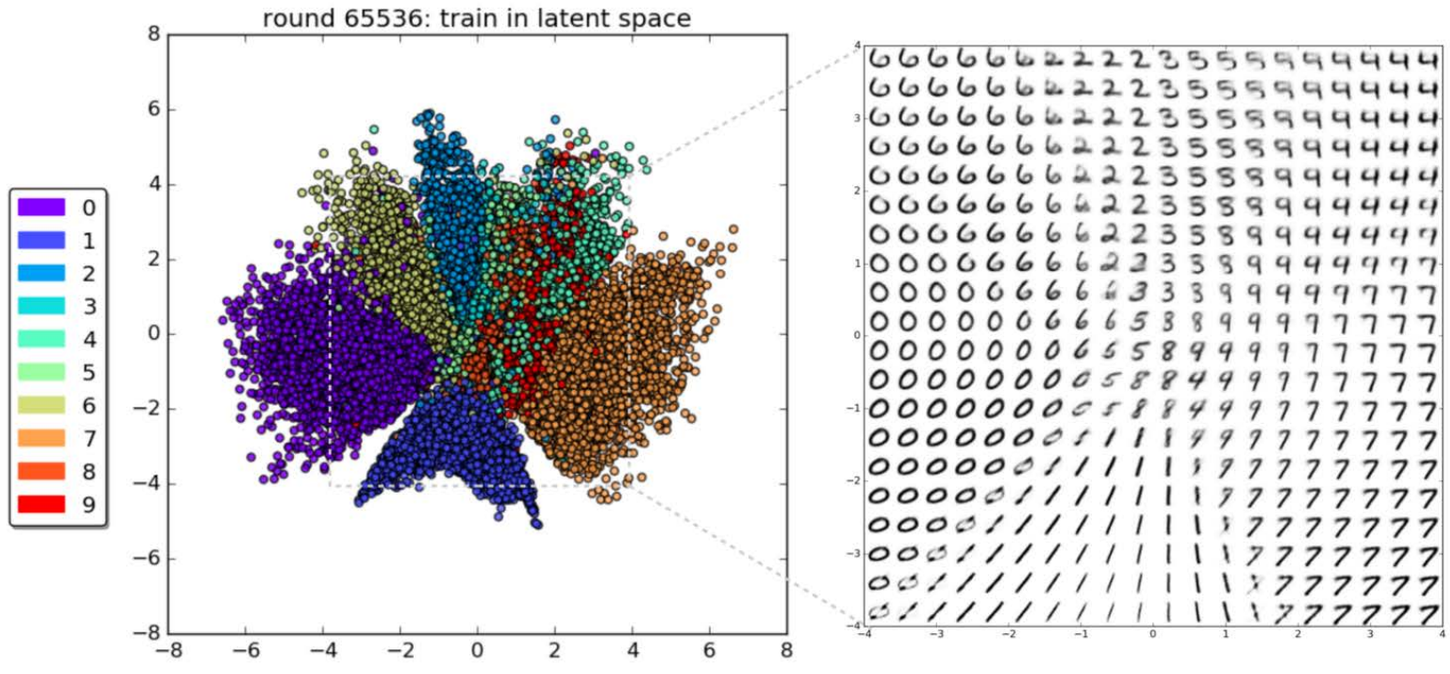
$$\mathcal{M}_\theta : \mathcal{X} \to \mathcal{Y}, \quad \mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \mapsto p(\mathcal{M}_\theta(x^*)|\mathcal{D}, x^*)$$

$$= \int p(\mathcal{M}_\theta|\mathcal{D}, x^*, \theta)\; p(\theta|\mathcal{D})\; d\theta$$

Introduction to unsupervised learning

- Density estimation
- Learning to draw samples from a distribution
- Learning to denoise samples from a distribution
- Find a low-dimensional manifold that the data lies near
- Cluster the data into groups of related examples





A classic unsupervised task:

Find the "best" representation of the data.

*By "best" we can mean different things, but generally speaking we are looking for a representation that preserves as much information about *x* as possible while obeying some penalty or constraint aimed at keeping the representation simpler or more accessible than *x* itself.

Principal component analysis

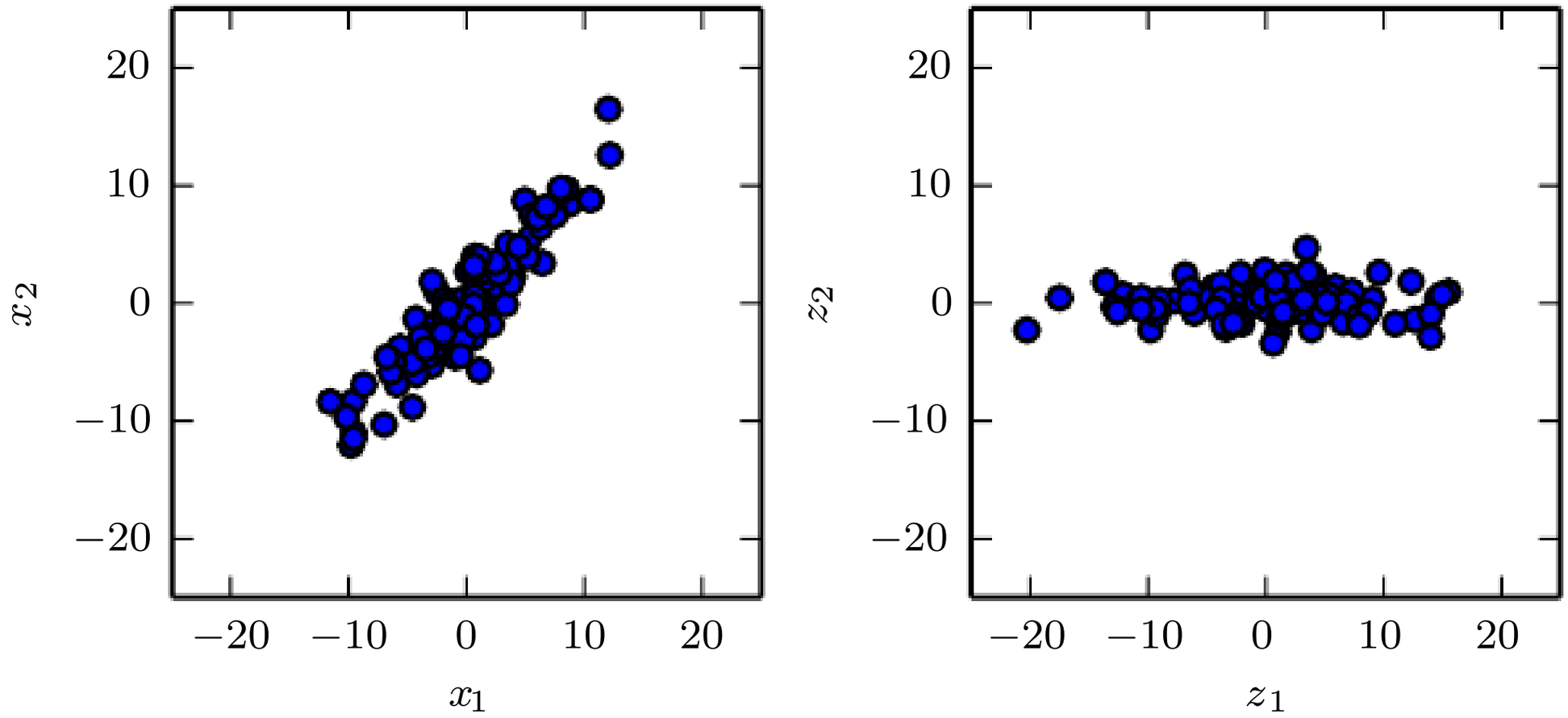
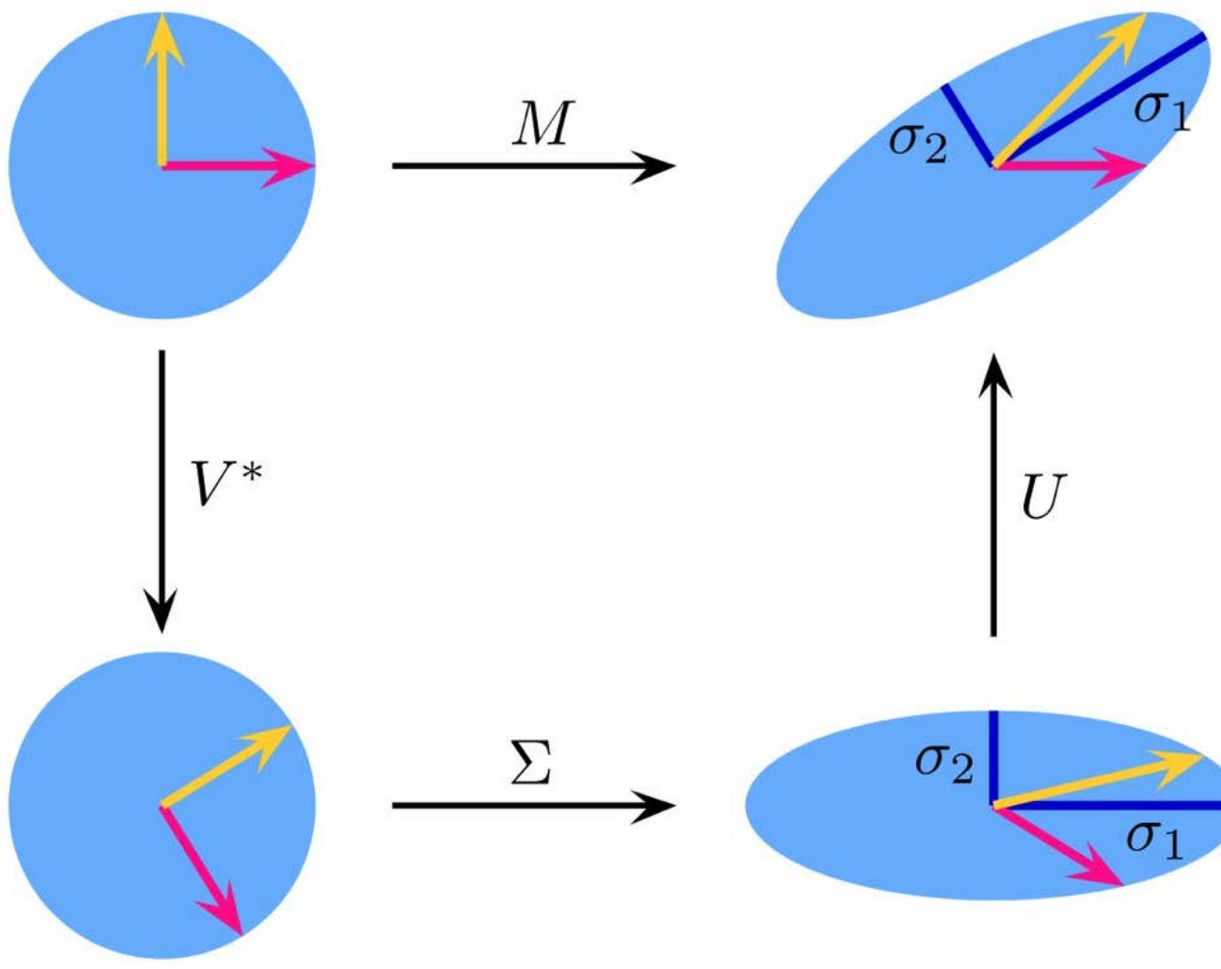


Figure 5.8: PCA learns a linear projection that aligns the direction of greatest variance with the axes of the new space. *(Left)*The original data consists of samples of $\boldsymbol{x}$. In this space, the variance might occur along directions that are not axis-aligned. *(Right)*The transformed data $\boldsymbol{z} = \boldsymbol{x}^\top \boldsymbol{W}$ now varies most along the axis z_1. The direction of second most variance is now along z_2.

Singular value decomposition

The SVD decomposes the action of a matrix into rotations and scalings along the axes:

$$M = U\,\mathrm{diag}(\sigma_i)V^\top$$



Linear algebra recap: Singular value decomposition

The **singular value decomposition** (SVD) provides another way to factorize a matrix, into **singular vectors** and **singular values**. The SVD allows us to discover some of the same kind of information as the eigendecomposition. However, the SVD is more generally applicable. Every real matrix has a singular value decomposition, but the same is not true of the eigenvalue decomposition. For example, if a matrix is not square, the eigendecomposition is not defined, and we must use a singular value decomposition instead.

Recall that the eigendecomposition involves analyzing a matrix $\boldsymbol{A}$ to discover a matrix $\boldsymbol{V}$ of eigenvectors and a vector of eigenvalues $\boldsymbol{\lambda}$ such that we can rewrite $\boldsymbol{A}$ as

$$\boldsymbol{A} = \boldsymbol{V}\text{diag}(\boldsymbol{\lambda})\boldsymbol{V}^{-1}.$$

The singular value decomposition is similar, except this time we will write $\boldsymbol{A}$ as a product of three matrices:

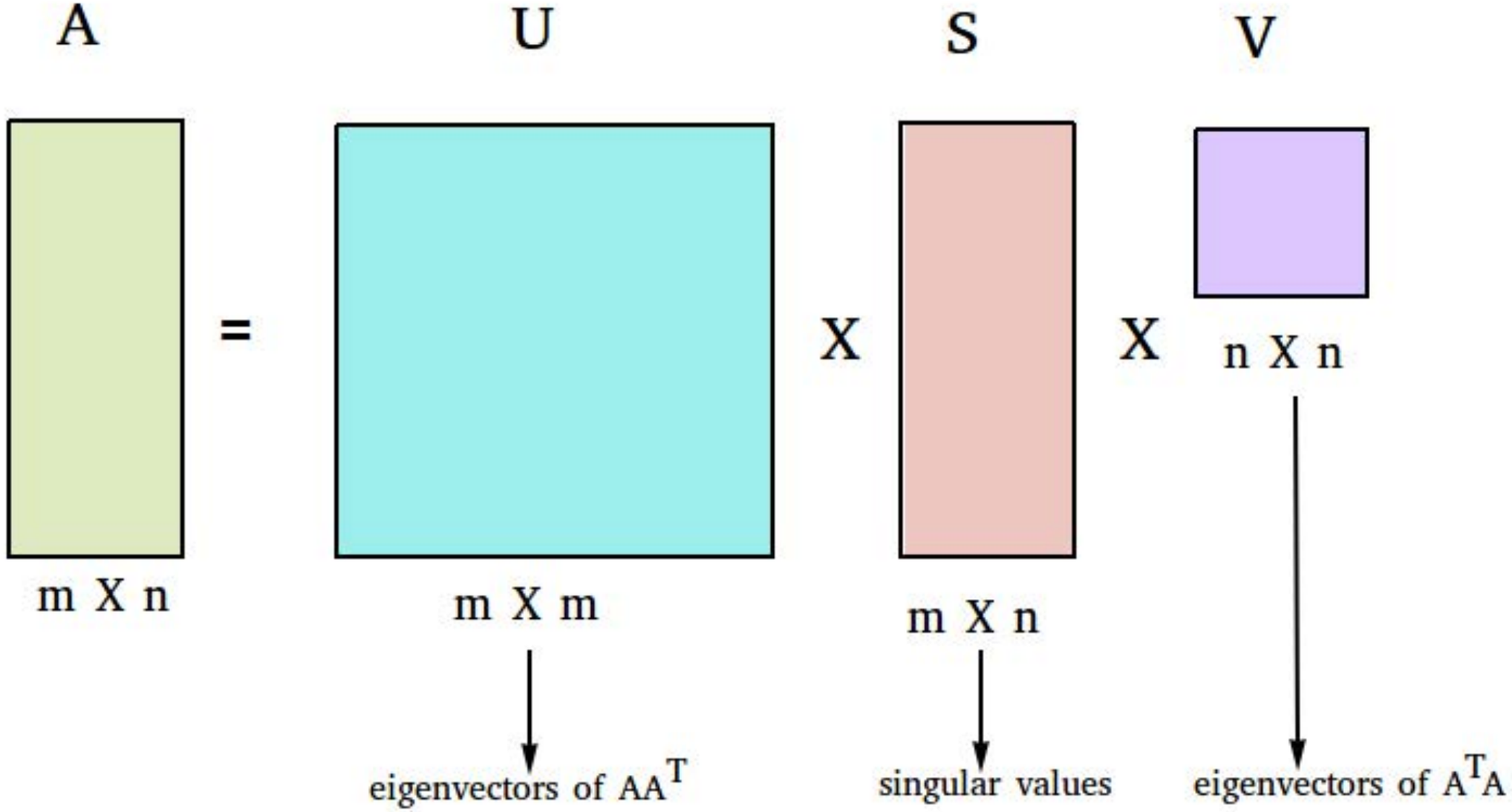
$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^{\top}.$$

Suppose that $\boldsymbol{A}$ is an $m \times n$ matrix. Then $\boldsymbol{U}$ is defined to be an $m \times m$ matrix, $\boldsymbol{D}$ to be an $m \times n$ matrix, and $\boldsymbol{V}$ to be an $n \times n$ matrix.

Each of these matrices is defined to have a special structure. The matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ are both defined to be orthogonal matrices. The matrix $\boldsymbol{D}$ is defined to be a diagonal matrix. Note that $\boldsymbol{D}$ is not necessarily square.

The elements along the diagonal of $\boldsymbol{D}$ are known as the **singular values** of the matrix $\boldsymbol{A}$. The columns of $\boldsymbol{U}$ are known as the **left-singular vectors**. The columns of $\boldsymbol{V}$ are known as as the **right-singular vectors**.

Singular value decomposition



https://blog.paperspace.com/dimension-reduction-with-principal-component-analysis/

Linear algebra recap: Eigendecomposition

One of the most widely used kinds of matrix decomposition is called **eigendecomposition**, in which we decompose a matrix into a set of eigenvectors and eigenvalues.

An **eigenvector** of a square matrix $\boldsymbol{A}$ is a non-zero vector $\boldsymbol{v}$ such that multiplication by $\boldsymbol{A}$ alters only the scale of $\boldsymbol{v}$:

$$\boldsymbol{A}\boldsymbol{v} = \lambda \boldsymbol{v}.$$

The scalar λ is known as the **eigenvalue** corresponding to this eigenvector. (One can also find a **left eigenvector** such that $\boldsymbol{v}^\top \boldsymbol{A} = \lambda \boldsymbol{v}^\top$, but we are usually concerned with right eigenvectors).

If $\boldsymbol{v}$ is an eigenvector of $\boldsymbol{A}$, then so is any rescaled vector $s\boldsymbol{v}$ for $s \in \mathbb{R}, s \neq 0$. Moreover, $s\boldsymbol{v}$ still has the same eigenvalue. For this reason, we usually only look for unit eigenvectors.

Suppose that a matrix $\boldsymbol{A}$ has n linearly independent eigenvectors, $\{\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(n)}\}$, with corresponding eigenvalues $\{\lambda_1, \ldots, \lambda_n\}$. We may concatenate all of the eigenvectors to form a matrix $\boldsymbol{V}$ with one eigenvector per column: $\boldsymbol{V} = [\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(n)}]$. Likewise, we can concatenate the eigenvalues to form a vector $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_n]^\top$. The **eigendecomposition** of $\boldsymbol{A}$ is then given by

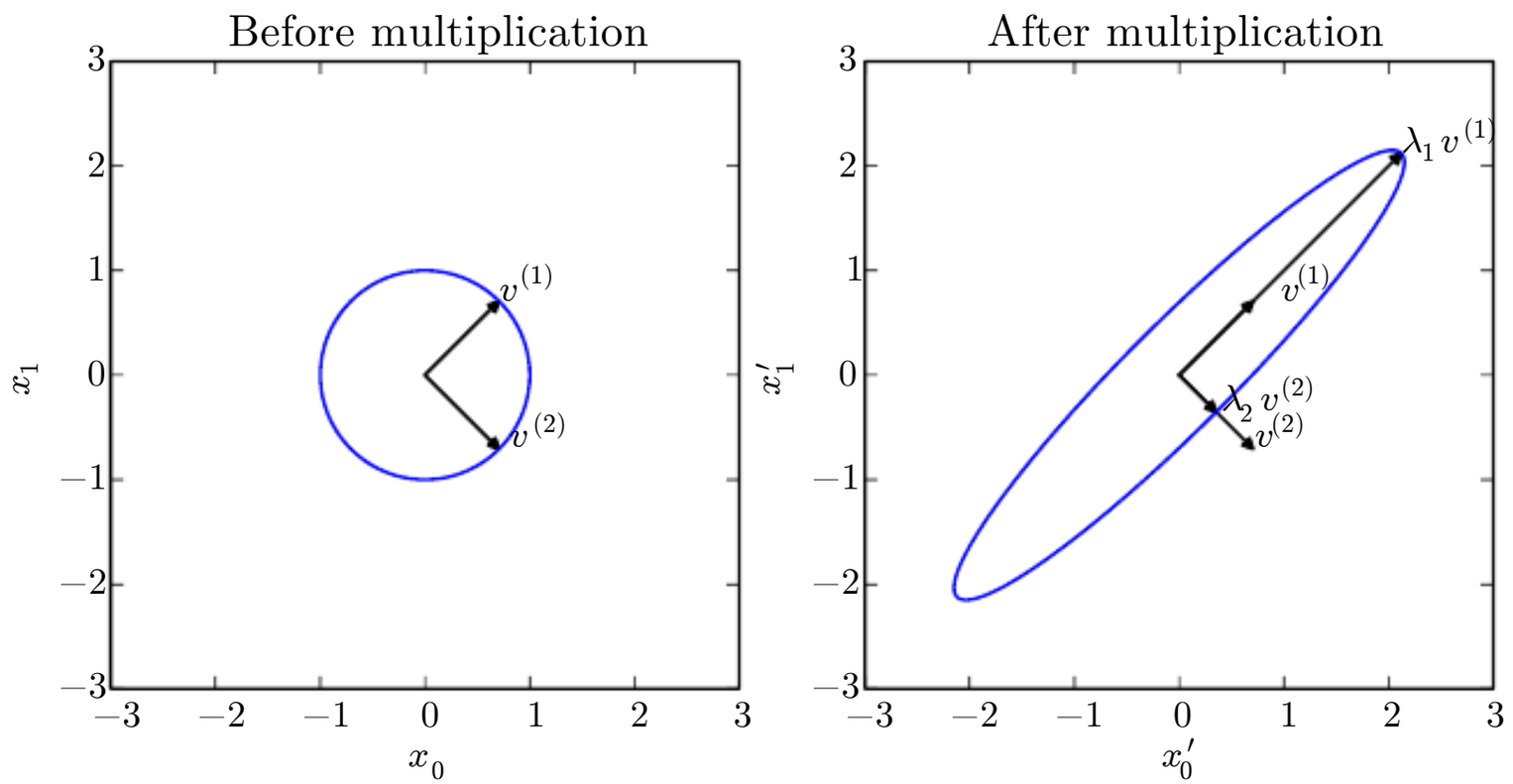
$$\boldsymbol{A} = \boldsymbol{V}\text{diag}(\boldsymbol{\lambda})\boldsymbol{V}^{-1}.$$

Linear algebra recap: Eigendecomposition

Not every matrix can be decomposed into eigenvalues and eigenvectors. In some cases, the decomposition exists, but may involve complex rather than real numbers. Fortunately, in this book, we usually need to decompose only a specific class of matrices that have a simple decomposition. Specifically, every real symmetric matrix can be decomposed into an expression using only real-valued eigenvectors and eigenvalues:

$$\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top,$$

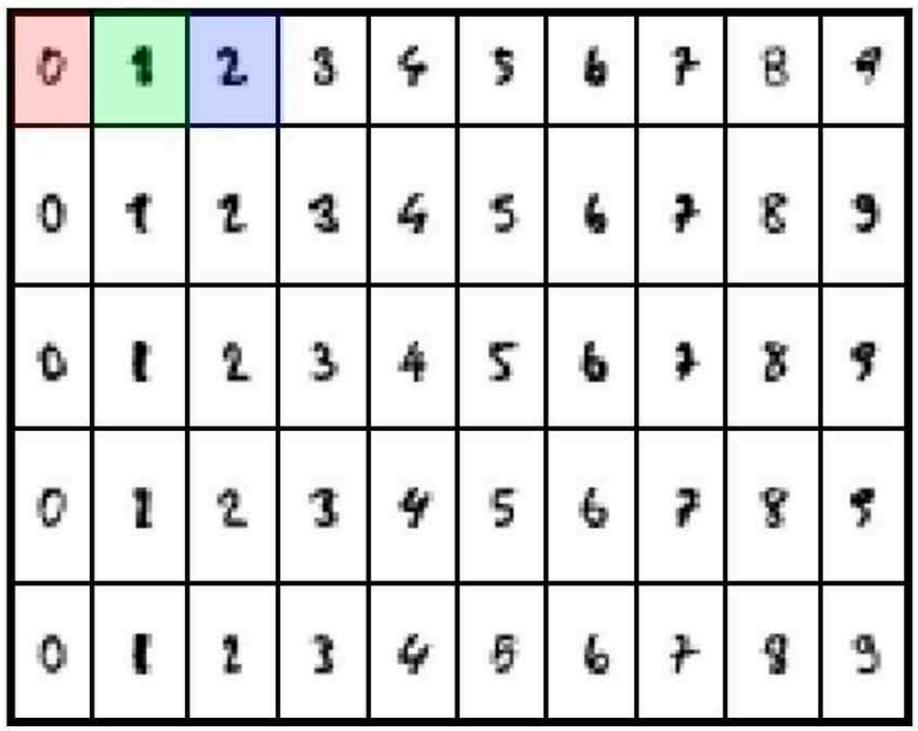
where $\boldsymbol{Q}$ is an orthogonal matrix composed of eigenvectors of $\boldsymbol{A}$, and $\boldsymbol{\Lambda}$ is a diagonal matrix. The eigenvalue $\Lambda_{i,i}$ is associated with the eigenvector in column i of $\boldsymbol{Q}$, denoted as $\boldsymbol{Q}_{:,i}$. Because $\boldsymbol{Q}$ is an orthogonal matrix, we can think of $\boldsymbol{A}$ as scaling space by λ_i in direction $\boldsymbol{v}^{(i)}$.

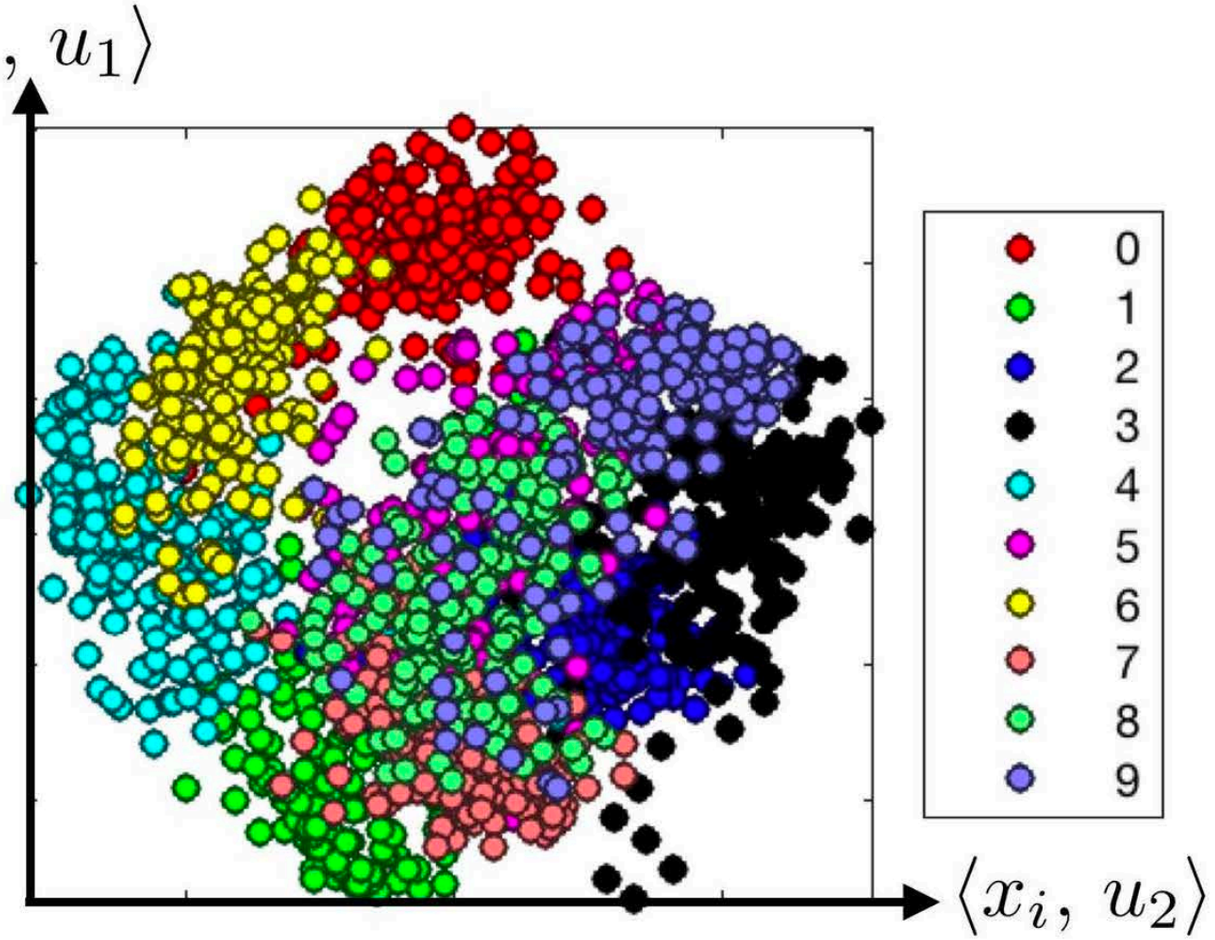


Principal component analysis

Input data:	$X = (x_i)_{i=1}^n \in \mathbb{R}^{n \times p}, x_i \in \mathbb{R}^p$
Remove mean:	$x_i \leftarrow x_i - \frac{1}{n} \sum_j x_j$
Covariance:	$C \overset{\text{def.}}{=} \frac{1}{n} X^\top X \in \mathbb{R}^{p \times p}$
Eigen-decomposition:	$C = U \operatorname{diag}(\sigma_k^2) U^\top, U = (u_k)_{k=1}^p$

x_1 x_2 x_3

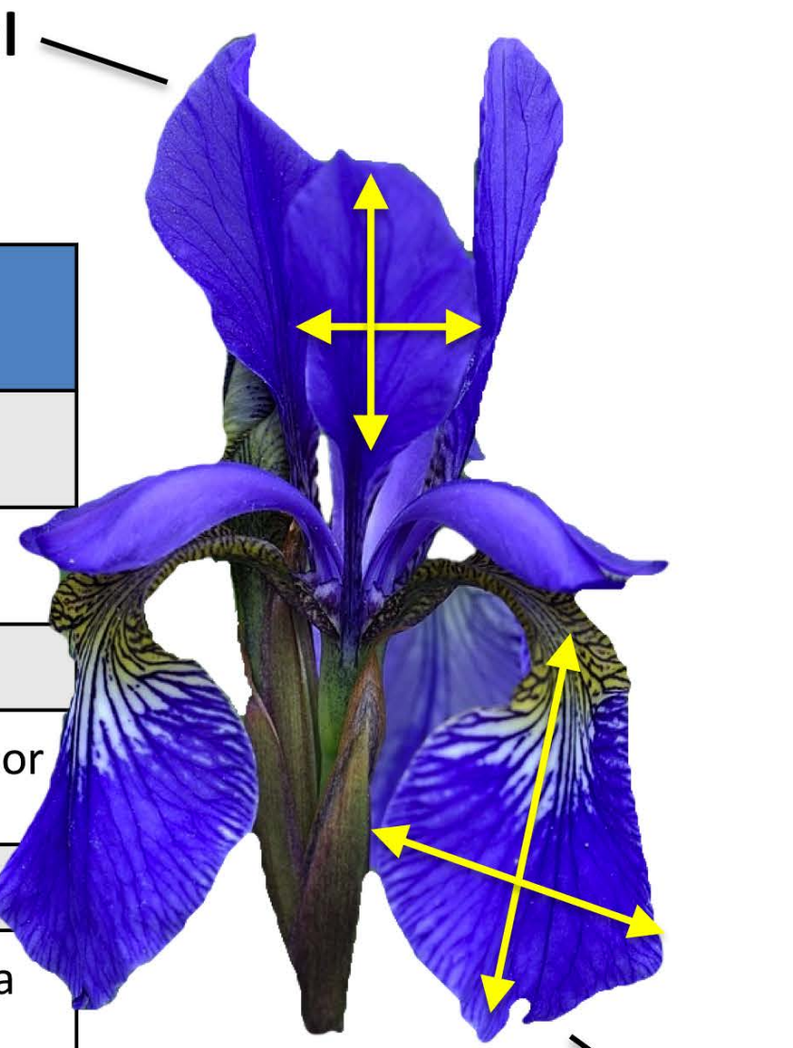




Example: the Iris data set

Samples
(instances, observations)

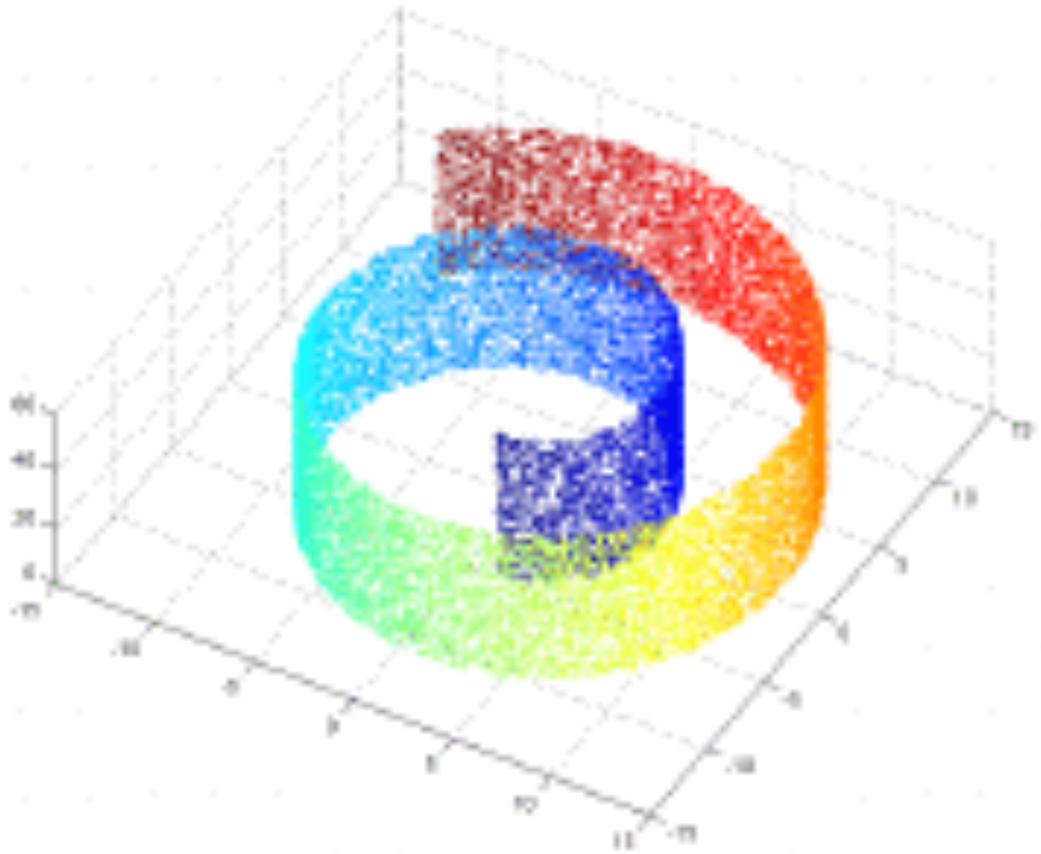
	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
			...		
50	6.4	3.5	4.5	1.2	Versicolor
			...		
150	5.9	3.0	5.0	1.8	Virginica

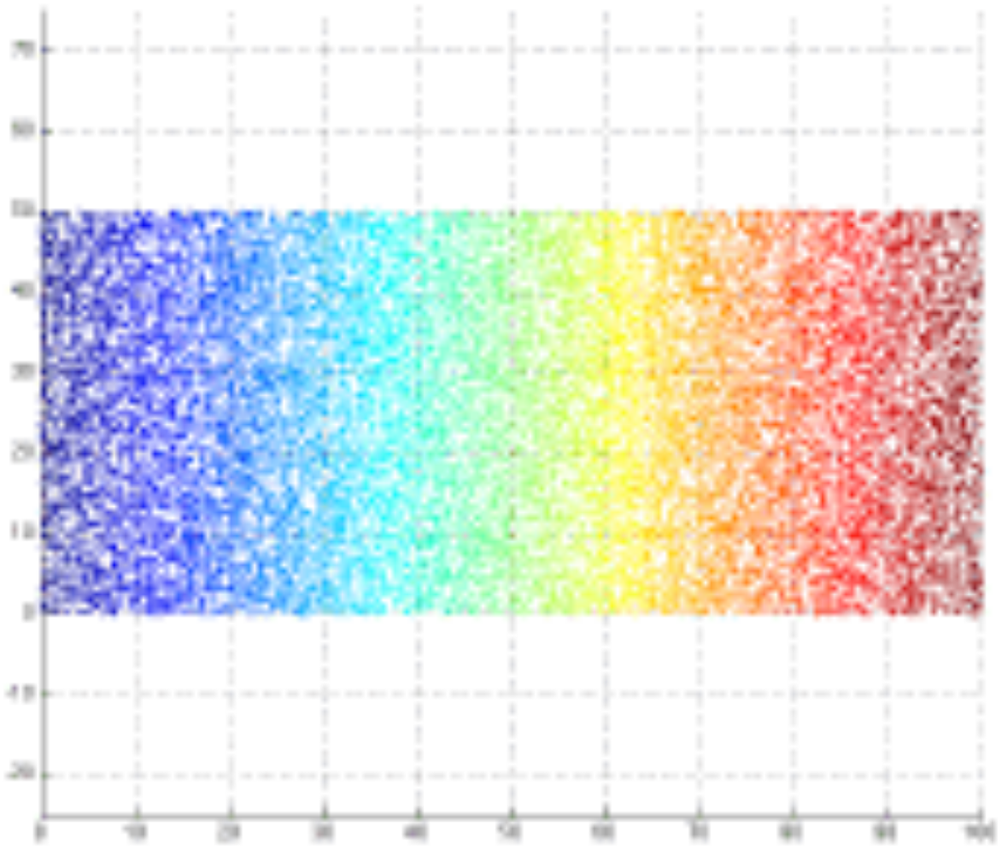


Features
(attributes, measurements, dimensions)

Class labels
(targets)

Kernel PCA





Linear PCA

$$X_{Nxd}$$

$$C = E[X^T X] = V \Lambda V^T$$

$$\hat{X} = X . \hat{V}^k . \hat{V}^{kT}$$

Kernel PCA

$$C = \frac{1}{N} \sum_{i=1}^{N} \phi(\mathbf{x_i}) \phi(\mathbf{x_i})^T$$

$$\kappa(\mathbf{x_i}, \mathbf{x_j}) = exp\Big(- \gamma \, \|\mathbf{x_i} - \mathbf{x_j}\|_2^2 \Big)$$

$$K = K - \mathbf{1_N} K - K \mathbf{1_N} + \mathbf{1_N} K \mathbf{1_N}$$