Imperial College London

Lecture 2: Dimensionality reduction and PCA

Introduction to machine learning

Kevin Webster

Department of Mathematics
Imperial College London

Outline

Outline

Machine Learning definition revisited

'A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E.'

Feature selection

- The raw data can be in any format. There may be a lot of data attributes, and not all of them will be useful
- The first stage of the machine learning is to represent the data in a way that the algorithm can work with
- The number of features can end up being very large
- This is a problem for many machine learning algorithms, often referred to as the **curse of dimensionality**
- Consider the k-NN algorithm: in high dimensions the k-nearest neighbours grow further apart

Curse of dimensionality

- The k-NN algorithm works on the principle that similar points share similar labels
- As the feature dimension gets large, we need exponentially more data points to maintain an measure of 'closeness' between them
- The computation time also increases substantially in high dimensions, and the k-NN algorithm may become infeasible
- This problem is not unique to k-NN; all machine learning algorithms suffer from problems due to high dimensionality of the data features

Dimensionality reduction

- Dimensionality reduction techniques aim to transform the high-dimensional data to a space of fewer dimensions
- The transformation may be linear or nonlinear
- These techniques have the following benefits:
    - Removal of redundant features
    - Reduction of storage and computation costs
    - Data visualisation
    - Avoiding modelling problems due to the curse of dimensionality
- In this lecture we will study Principal Components Analysis (PCA), which is a popular linear method for dimensionality reduction

Dimensionality reduction methods

- Other methods for dimensionality reduction include:
    - Nonnegative matrix factorisation (NMF)
    - Kernel PCA
    - Linear discriminant analysis (LDA)
    - Generalised discriminant analysis (GDA)
    - Neural network autoencoders
- Note that these are unsupervised learning algorithms

Outline

Linear coding

- We first derive the PCA algorithm as a linear encoding algorithm
- Suppose we have the dataset of points

$$\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(m)}\} \text{ with each } \mathbf{x}^{(i)} \in \mathbb{R}^n$$

- For each $\mathbf{x}^{(i)} \in \mathbb{R}^n$ we want to map it to a code $\mathbf{c}^{(i)} \in \mathbb{R}^l$ with $l < n$
- The code is then a compressed representation of the data point
- Our coding is given by $\mathbf{c} = f(\mathbf{x})$ for some function $f : \mathbb{R}^n \mapsto \mathbb{R}^l$
- We are likely to lose some information in the compression, but we would like to find a decoding function that approximately reconstructs the datapoint: $\mathbf{x} \approx g(f(\mathbf{x}))$, for some $g : \mathbb{R}^l \mapsto \mathbb{R}^n$

Linear coding

- We choose a linear decoding function:

$$g(\mathbf{c}) = \mathbf{D}\mathbf{c}, \qquad \mathbf{D} \in \mathbb{R}^{n \times l}$$

- PCA constrains the matrix $\mathbf{D}$ to have orthogonal columns, so that $\mathbf{D}^T\mathbf{D} = \mathbf{I}_l$
- This means that the vector $\mathbf{c}$ is the coordinates for a set of orthogonal vectors in $\mathbb{R}^n$
- In addition, the columns of $\mathbf{D}$ are constrained to have unit norm—note this does not lose any generality, it just fixes the scale of the coordinates

Optimal code for a given input

- We would like to minimise the distance between a given data point $\mathbf{x}$ and it's reconstruction $g(\mathbf{c}^*)$:

$$\begin{aligned}
\mathbf{c}^* &:= \arg\min_{\mathbf{c}} ||\mathbf{x} - g(\mathbf{c})||_2^2 \\
&= \arg\min_{\mathbf{c}} \langle \mathbf{x} - g(\mathbf{c}), \mathbf{x} - g(\mathbf{c}) \rangle \\
&= \arg\min_{\mathbf{c}} \left(-2\mathbf{x}^T g(\mathbf{c}) + g(\mathbf{c})^T g(\mathbf{c}) \right) \\
&= \arg\min_{\mathbf{c}} \left(-2\mathbf{x}^T \mathbf{D}\mathbf{c} + \mathbf{c}^T \mathbf{D}^T \mathbf{D}\mathbf{c} \right) \\
&= \arg\min_{\mathbf{c}} \left(-2\mathbf{x}^T \mathbf{D}\mathbf{c} + \mathbf{c}^T \mathbf{c} \right)
\end{aligned}$$

Optimal code for a given input

$$\mathbf{c}^* = \arg\min_{\mathbf{c}} \left(-2\mathbf{x}^T\mathbf{D}\mathbf{c} + \mathbf{c}^T\mathbf{c}\right)$$

- This optimisation problem is quadratic in $\mathbf{c}$. We solve it by setting the gradient to zero:

$$\begin{aligned} \nabla_{\mathbf{c}} \left(-2\mathbf{x}^T\mathbf{D}\mathbf{c} + \mathbf{c}^T\mathbf{c}\right) &= 0 \\ -2\mathbf{D}^T\mathbf{x} + 2\mathbf{c} &= 0 \\ \Rightarrow \quad \mathbf{c} = \mathbf{D}^T\mathbf{x} \end{aligned}$$

- Therefore our encoding function is given by $f(\mathbf{x}) = \mathbf{D}^T\mathbf{x}$ and the PCA reconstruction operation is $r(\mathbf{x}) := g(f(\mathbf{x})) = \mathbf{D}\mathbf{D}^T\mathbf{x}$

Finding the encoding matrix $\mathbf{D}$

- To choose the encoding matrix $\mathbf{D}$, we minimise the ℓ_2 distance between data points and their PCA reconstruction:

$$\mathbf{D}^* = \arg\min_{\mathbf{D}} \sum_{i,j} \left(x_j^{(i)} - r(\mathbf{x}^{(i)})_j \right)^2 \text{ subject to } \mathbf{D}^T\mathbf{D} = \mathbf{I}_l$$

- For ease of presentation, we consider the case $l = 1$, so that $\mathbf{D}$ is a vector $\mathbf{d} \in \mathbb{R}^n$. In this case we have

$$\mathbf{d}^* = \arg\min_{\mathbf{d}} \sum_{i} ||\mathbf{x}^{(i)} - \mathbf{d}\mathbf{d}^T\mathbf{x}^{(i)}||_2^2 \text{ subject to } ||\mathbf{d}||_2 = 1$$

Finding the encoding vector d

- We rewrite the problem by defining the **design matrix** $\mathbf{X} \in \mathbb{R}^{m \times n}$, formed by stacking all data points in the rows of $\mathbf{X}$
- The problem then becomes

$$\mathbf{d}^* = \arg\min_{\mathbf{d}} ||\mathbf{X} - \mathbf{X}\mathbf{d}\mathbf{d}^T||_F^2 \text{ subject to } ||\mathbf{d}||_2 = 1$$

- The Frobenius norm can be rewritten

$$\begin{aligned}
\mathbf{d}^* &= \arg\min_{\mathbf{d}} \mathrm{Tr}\left((\mathbf{X} - \mathbf{X}\mathbf{d}\mathbf{d}^T)^T(\mathbf{X} - \mathbf{X}\mathbf{d}\mathbf{d}^T)\right) \text{ s.t. } ||\mathbf{d}||_2 = 1 \\
&= \cdots \\
&= \arg\min_{\mathbf{d}} -\mathrm{Tr}\left(\mathbf{d}^T\mathbf{X}^T\mathbf{X}\mathbf{d}\right) \text{ s.t. } ||\mathbf{d}||_2 = 1 \\
&= \arg\max_{\mathbf{d}} \mathrm{Tr}\left(\mathbf{d}^T\mathbf{X}^T\mathbf{X}\mathbf{d}\right) \text{ s.t. } ||\mathbf{d}||_2 = 1
\end{aligned}$$

Finding the encoding vector d

$$\mathbf{d}^* = \arg\max_{\mathbf{d}} \mathrm{Tr}\left(\mathbf{d}^T \mathbf{X}^T \mathbf{X} \mathbf{d}\right) \text{ s.t. } ||\mathbf{d}||_2 = 1$$

- This optimisation problem can be solved by considering the eigensystem of the symmetric positive definite matrix $\mathbf{X}^T\mathbf{X}$
- Recall that symmetric matrices have real eigenvalues and orthogonal eigenvectors, and since $\mathbf{X}^T\mathbf{X}$ is positive definite, they will also all be positive
- Therefore $\mathbf{d}^*$ is given by the eigenvector corresponding to the largest eigenvalue
- It can be shown that the matrix $\mathbf{D}$ is given by the l eigenvectors corresponding to the largest l eigenvalues

Outline

PCA algorithm

- We have seen that the PCA dimensionality reduction can be computed by constructing the design matrix

$$\mathbf{X} \in \mathbb{R}^{m \times n}, \quad \text{with } \mathbf{X}_{(i,\cdot)} = \mathbf{x}^{(i)}$$

  and solving the eigensystem of the symmetric positive matrix $\mathbf{X}^T\mathbf{X} \in \mathbb{R}^{n \times n}$

- To compress the data to l dimensions, take the l eigenvectors $\mathbf{v}^{(j)}$ corresponding to the largest l eigenvalues of $\mathbf{X}^T\mathbf{X}$ and define

$$\mathbf{D} \in \mathbb{R}^{n \times l}, \quad \text{with } \mathbf{D}_{(\cdot,j)} = \mathbf{v}^{(j)}$$

- Each data point $\mathbf{x}^{(i)}$ is encoded to $\mathbf{c}^{(i)} = \mathbf{D}^T\mathbf{x}^{(i)}$
- The reconstruction $\tilde{\mathbf{x}}^{(i)}$ is given by $\tilde{\mathbf{x}}^{(i)} = \mathbf{D}\mathbf{D}^T\mathbf{x}^{(i)}$

Representation with decorrelated elements

- The PCA reduction can also be viewed as a representation that achieves the following purposes:
  - Reduction in dimension from the original input
  - Reduced representation has elements with no linear correlation
  - The variance of each element is maximised
- As such, the PCA reduction attempts to capture as much of the variability of the data as possible, whilst also minimising the redundancy in the representation

Covariance matrix

- Consider again the design matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$
- Suppose that the data has been centred, that is $\mathbb{E}[\mathbf{x}] = 0$. This can easily be enforced by subtracted the mean of each feature from every data point
- Recall the variance of a random variable $\text{Var}[x] = \mathbb{E}[x^2] - (\mathbb{E}[x])^2$
- Similarly, recall the covariance of two random variables is given by $\text{Cov}[x, y] = \mathbb{E}[xy] - \mathbb{E}[x]\mathbb{E}[y]$
- The variance of the data is given by the unbiased sample covariance matrix:

$$\text{Var}\left[\mathbf{x}\right] = \frac{1}{m-1}\mathbf{X}^T\mathbf{X}$$

Covariance matrix

- PCA is a linear projection, and so our reduced representation is given by $\mathbf{z} = \mathbf{W}^T\mathbf{x}$ for some matrix $\mathbf{W} \in \mathbb{R}^{n \times l}$
- Our criteria is that the elements of our reduced representation should be decorrelated
- Recall the correlation of two random variables is given by

$$\text{Corr}[x, y] = \frac{\text{Cov}[x, y]}{\sqrt{\text{Var}[x]\text{Var}[y]}} = \frac{\text{Cov}[x, y]}{\sigma_x \sigma_y},$$

  where σ_x and σ_y are the standard deviations of the variables x and y respectively
- Our requirement is that the covariance matrix $\text{Var}[\mathbf{z}]$ is diagonal

Covariance matrix for reduced representation

- The covariance matrix in the reduced representation can be written

$$\begin{aligned}\mathrm{Var}\left[\mathbf{z}\right] &= \frac{1}{m-1}\mathbf{Z}^T\mathbf{Z} \\ &= \frac{1}{m-1}\mathbf{W}^T\mathbf{X}^T\mathbf{X}\mathbf{W}\end{aligned}$$

  where $\mathbf{Z} = \mathbf{X}\mathbf{W}$
- Now recall that $\mathbf{X}^T\mathbf{X}$ is a positive definite symmetric matrix, so all eigenvalues are real and the eigenvectors are orthogonal

Covariance matrix for reduced representation

- We can see from the above that if we construct $\mathbf{W}$ by stacking eigenvectors of $\mathbf{X}^T\mathbf{X}$ in its columns, then

$$\mathbf{W}^T\mathbf{X}^T\mathbf{X}\mathbf{W} = \mathbf{\Lambda}$$

  is an eigendecomposition of $\mathbf{X}^T\mathbf{X}$ using l of the eigenvalues/eigenvectors
- In particular, $\mathbf{\Lambda}$ is a diagonal matrix, with eigenvalues along the diagonal
- The eigenvalues are the sample variance of the elements of $\mathbf{z}$ (scaled by the multiplicative factor $m-1$)
- The elements of $\mathbf{z}$ are decorrelated
- Choosing the l largest eigenvalues gives the l directions of maximal variance, and these directions are given by the corresponding eigenvectors of $\mathbf{X}^T\mathbf{X}$

Derivation using singular value decomposition

- We can derive the same result for the principal components using singular value decomposition (SVD)
- Recall the SVD of a matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ is given by

$$\mathbf{X} = \mathbf{U\Sigma W}^T,$$

  where $\mathbf{U} \in \mathbb{R}^{m \times m}$, $\mathbf{W} \in \mathbb{R}^{n \times n}$ and $\Sigma \in \mathbb{R}^{m \times n}$
- $\mathbf{U}$ and $\mathbf{W}$ are orthogonal matrices
- Σ only has nonnegative entries on the diagonal: $\Sigma_{ii} = \sigma_i \geq 0$, called the singular values, and $\Sigma_{ij} = 0$ for $i \neq j$

Derivation using singular value decomposition

- We have that

$$\begin{aligned} \operatorname{Var}[\mathbf{x}] &= \frac{1}{m-1}\mathbf{X}^T\mathbf{X} \\ &= \frac{1}{m-1}(\mathbf{U}\boldsymbol{\Sigma}\mathbf{W}^T)^T\mathbf{U}\boldsymbol{\Sigma}\mathbf{W}^T \\ &= \frac{1}{m-1}\mathbf{W}\boldsymbol{\Sigma}^T\mathbf{U}^T\mathbf{U}\boldsymbol{\Sigma}\mathbf{W}^T \\ &= \frac{1}{m-1}\mathbf{W}\boldsymbol{\Sigma}^2\mathbf{W}^T \end{aligned}$$

where we used $\mathbf{U}^T\mathbf{U} = \mathbf{I}_m$ because $\mathbf{U}$ is orthogonal

Derivation using singular value decomposition

- Now we can use the above to show that the sample covariance matrix for $\mathbf{z}$ is diagonal, where $\mathbf{z} = \mathbf{W}^T\mathbf{x}$:

$$
\begin{aligned}
\mathrm{Var}[\mathbf{z}] &= \frac{1}{m-1}\mathbf{Z}^T\mathbf{Z} \\
&= \frac{1}{m-1}\mathbf{W}^T\mathbf{X}^T\mathbf{X}\mathbf{W} \\
&= \frac{1}{m-1}\mathbf{W}^T\mathbf{W}\boldsymbol{\Sigma}^2\mathbf{W}^T\mathbf{W} \\
&= \frac{1}{m-1}\boldsymbol{\Sigma}^2
\end{aligned}
$$

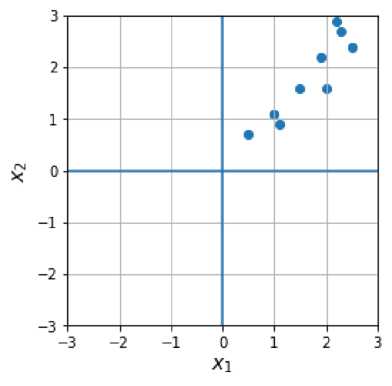
where we now use $\mathbf{W}^T\mathbf{W} = \mathbf{I}_n$

Outline

PCA example

Suppose we have the following set of data points for a random variable $\mathbf{x} \in \mathbb{R}^2$:

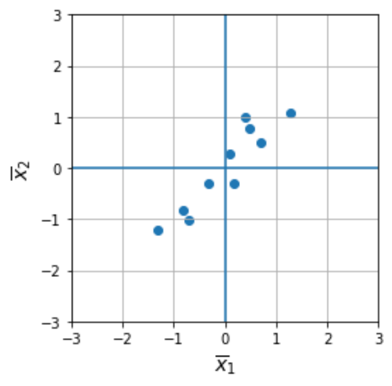
x_1	x_2
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9



PCA example

The first step is to centre the data by subtracting the mean from each feature:

$\overline{x}_1$	$\overline{x}_2$
.69	.49
-1.31	-1.21
.39	.99
.09	.29
1.29	1.09
.49	.79
.19	-.31
-.81	-.81
-.31	-.31
-.71	-1.01



PCA example

The centred data now forms the design matrix $\mathbf{X} \in 10 \times 2$:

$$\mathbf{X} = \begin{bmatrix} .69 & .49 \\ -1.31 & -1.21 \\ .39 & .99 \\ .09 & .29 \\ 1.29 & 1.09 \\ .49 & .79 \\ .19 & -.31 \\ -.81 & -.81 \\ -.31 & -.31 \\ -.71 & -1.01 \end{bmatrix}$$

PCA example

The covariance matrix is given by

$$\begin{aligned}
\mathrm{Var}[\mathbf{x}] &= \frac{1}{m-1}\mathbf{X}^T\mathbf{X} \\
&= \frac{1}{9}\begin{bmatrix} 5.549 & 5.539 \\ 5.539 & 6.449 \end{bmatrix} \\
&= \begin{bmatrix} 0.616 & 0.615 \\ 0.615 & 0.716 \end{bmatrix}
\end{aligned}$$

Note the covariance matrix is symmetric (and positive definite)

PCA example

The covariance matrix has eigenvalues

$$\lambda_1 = 1.284, \quad \lambda_2 = 0.049$$

with corresponding eigenvectors

$$\mathbf{w}_1 = \begin{bmatrix} -0.678 \\ -0.735 \end{bmatrix}, \qquad \mathbf{w}_2 = \begin{bmatrix} -0.735 \\ 0.678 \end{bmatrix}$$

Equivalently, we have the decomposition

$$\frac{1}{m-1}\mathbf{X}^T\mathbf{X} = \mathbf{W}\mathbf{\Lambda}\mathbf{W}^T = [\mathbf{w}_1|\mathbf{w}_2]\,\mathrm{diag}[\lambda_1, \lambda_2]\,[\mathbf{w}_1|\mathbf{w}_2]^T$$

$$\begin{bmatrix} 0.616 & 0.615 \\ 0.615 & 0.716 \end{bmatrix} = \begin{bmatrix} -0.678 & -0.735 \\ -0.735 & 0.678 \end{bmatrix} \begin{bmatrix} 1.284 & 0 \\ 0 & 0.049 \end{bmatrix} \begin{bmatrix} -0.678 & -0.735 \\ -0.735 & 0.678 \end{bmatrix}$$

PCA example

Similarly, the design matrix $\mathbf{X}$ has the following singular value decomposition:

$$\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{W}^T$$

where $U \in \mathbb{R}^{10\times 10}$, $\boldsymbol{\Sigma} \in \mathbb{R}^{10\times 2}$ with diagonal elements

$$\begin{aligned}
\sigma_1 &= \sqrt{(m-1)\lambda_1} &= 3.399 \\
\sigma_2 &= \sqrt{(m-1)\lambda_2} &= 0.665
\end{aligned}$$

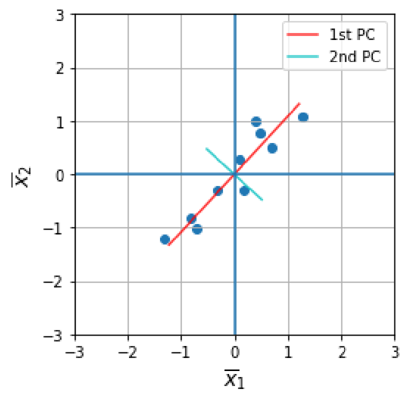
and

$$\mathbf{W} = [\mathbf{w}_1 | \mathbf{w}_2] = \begin{bmatrix} -0.678 & -0.735 \\ -0.735 & 0.678 \end{bmatrix}$$

Note that the right singular vectors are the same as the eigenvectors

PCA example

- The principal component directions are given by the eigenvectors of the covariance matrix (or the right singular vectors)
- The variance is given by the eigenvalues



Outline

Eigenvalues and eigenvectors

- Eigenvalues and eigenvectors are important properties of a square matrix
- Let $\mathbf{A} \in \mathbb{R}^{n \times n}$. Then $\lambda \in \mathbb{R}$ is an eigenvalue of $\mathbf{A}$ with associated (nonzero) eigenvector $\mathbf{v} \in \mathbb{R}^n$ if we have

$$\mathbf{A}\mathbf{v} = \lambda \mathbf{v}$$

- Think of this equation in terms of a linear mapping from $\mathbb{R}^n$ to itself. It says that the vector subspace $\mathrm{Sp}(\mathbf{v})$ is invariant under the linear map $\mathbf{A} : \mathbb{R}^n \mapsto \mathbb{R}^n$

Non-uniqueness of eigenvectors

- Note that eigenvectors are not unique: if $\mathbf{v}$ is an eigenvector with eigenvalue λ, then $\alpha\mathbf{v}$ ($\alpha \neq 0$) is also an eigenvector with the same eigenvalue

$$\begin{aligned} \mathbf{A}(\alpha\mathbf{v}) &= \alpha\mathbf{A}\mathbf{v} \\ &= \alpha\lambda\mathbf{v} \\ &= \lambda(\alpha\mathbf{v}) \end{aligned}$$

- This makes sense since all vectors $\alpha\mathbf{v}$ belong to the same vector subspace $\mathrm{Sp}(\mathbf{v})$

Finding eigenvalues and eigenvectors

- We can rearrange the eigenvalue equation

$$\mathbf{Av} = \lambda \mathbf{v}$$

  to obtain

$$(\mathbf{A} - \lambda \mathbf{I}_n)\mathbf{v} = \mathbf{0}$$

- Note that if $(\mathbf{A} - \lambda \mathbf{I}_n)$ is invertible, then we can solve the above system of linear equations to obtain the unique solution $\mathbf{v} = \mathbf{0}$
- Therefore, for λ to be an eigenvalue, the matrix $(\mathbf{A} - \lambda \mathbf{I}_n)$ needs to be singular (non-invertible)

Finding eigenvalues and eigenvectors

- The matrix $(\mathbf{A} - \lambda \mathbf{I}_n)$ is singular if and only if

$$\det(\mathbf{A} - \lambda \mathbf{I}_n) = 0$$

- This is the *characteristic equation*. If we compute the determinant then this turns out to be a polynomial equation of degree n in the unknown variable λ
- The roots of this polynomial equation are the eigenvalues
- Recall that a degree n polynomial equation has n roots, so there are n eigenvalues
- Some of these eigenvalues may come in complex conjugate pairs

Diagonalisation

- Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ have n eigenvalues λ_i and eigenvectors v_i, $i = 1, \ldots, n$
- For simplicity we assume the all eigenvalues are real and distinct
- The set of eigenvectors $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ form a basis for $\mathbb{R}^n$
- We can simplify, or diagonalise the matrix $\mathbf{A}$ and corresponding linear map by making a change of basis to these eigenvectors
- This can be achieved by setting the matrix $\mathbf{P} := [\mathbf{v}_1 | \mathbf{v}_2 | \cdots | \mathbf{v}_n]$ and $\mathbf{x} = \mathbf{P}\mathbf{u}$
- Then the linear map

$$\mathbf{x} \mapsto \mathbf{A}\mathbf{x}$$

becomes

$$\mathbf{u} \mapsto \mathbf{P}^{-1}\mathbf{A}\mathbf{P}\mathbf{u}$$

Diagonalisation

- Looking more closely at the matrix $\mathbf{P}^{-1}\mathbf{A}\mathbf{P}$:

$$\begin{aligned}
\mathbf{P}^{-1}\mathbf{A}\mathbf{P} &= \mathbf{P}^{-1}[\mathbf{A}\mathbf{v}_1|\mathbf{A}\mathbf{v}_2|\cdots|\mathbf{A}\mathbf{v}_n] \\
&= \mathbf{P}^{-1}[\lambda_1\mathbf{v}_1|\lambda_2\mathbf{v}_2|\cdots|\lambda_n\mathbf{v}_n] \\
&= \mathbf{P}^{-1}[\mathbf{v}_1|\mathbf{v}_2|\cdots|\mathbf{v}_n]\begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix} \\
&= \mathbf{P}^{-1}\mathbf{P}\,\mathrm{diag}[\lambda_1, \lambda_2, \ldots, \lambda_n] \\
&= \mathrm{diag}[\lambda_1, \lambda_2, \ldots, \lambda_n] =: \mathbf{D}
\end{aligned}$$

  so we can see that the matrix has become diagonalised
- Complex and repeated eigenvalues can be treated in a similar way; we will not cover this here

Singular value decomposition

- SVD is an important matrix decomposition, that applies to any matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$
- The decomposition is of the form

$$\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$$

  where $\mathbf{U} \in \mathbb{R}^{m \times m}$ and $\mathbf{V} \in \mathbb{R}^{n \times n}$ are square matrices and $\boldsymbol{\Sigma} \in \mathbb{R}^{m \times n}$
- Furthermore, $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices
- $\boldsymbol{\Sigma}$ only has nonnegative entries on the diagonal: $\Sigma_{ii} = \sigma_i \geq 0$, the **singular values**, and $\Sigma_{ij} = 0$ for $i \neq j$

Singular value decomposition

- The decomposition can be visualised as follows for $m > n$:

$$
\underset{m \times n}{\begin{bmatrix} & & \\ & \mathbf{A} & \\ & & \end{bmatrix}} = \underset{m \times m}{\begin{bmatrix} | & & | \\ \mathbf{u}_1 & \cdots & \mathbf{u}_m \\ | & & | \end{bmatrix}} \underset{m \times n}{\begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & 0 \\ 0 & \cdots & \sigma_n \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \end{bmatrix}} \underset{n \times n}{\begin{bmatrix} - & \mathbf{v}_1 & - \\ & \vdots & \\ - & \mathbf{v}_n & - \end{bmatrix}}
$$

Singular value decomposition

and for $m < n$:

$$\begin{bmatrix} & & \\ & \mathbf{A} & \\ & & \end{bmatrix} = \begin{bmatrix} | & & | \\ \mathbf{u}_1 & \cdots & \mathbf{u}_m \\ | & & | \end{bmatrix} \begin{bmatrix} \sigma_1 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & 0 & \cdots & 0 \\ 0 & \cdots & \sigma_m & \cdots & 0 \end{bmatrix} \begin{bmatrix} - & \mathbf{v}_1 & - \\ & \vdots & \\ - & \mathbf{v}_n & - \end{bmatrix}$$

$$m \times n \qquad m \times m \qquad m \times n \qquad n \times n$$

- The number of singular values is $\min(m, n)$
- The vectors $\mathbf{u}_1, \ldots, \mathbf{u}_m$ are an orthonormal basis for $\mathbb{R}^m$ and are the **left singular vectors**
- The vectors $\mathbf{v}_1, \ldots, \mathbf{v}_m$ are an orthonormal basis for $\mathbb{R}^n$ and are the **right singular vectors**

SVD: coordinate frames

- The SVD can be understood in terms of analysing the map $\mathbf{A} : \mathbb{R}^n \to \mathbb{R}^m$ in convenient coordinate bases

$$\mathbf{A}\mathbf{x} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{x}$$

- Recall that for an orthogonal matrix, $\mathbf{V}^{-1} = \mathbf{V}^T$
- Therefore the vector $\mathbf{V}^T\mathbf{x}$ gives the coordinates of $\mathbf{x}$ with respect to the basis vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$
- In this coordinate frame, the action of $\mathbf{A}$ is simplified to the diagonal matrix $\mathbf{\Sigma} : \mathbb{R}^n \to \mathbb{R}^m$
- The vector $\mathbf{\Sigma}\mathbf{V}^T\mathbf{x}$ is then the coordinates of $\mathbf{A}\mathbf{x}$ with respect to the basis vectors $\mathbf{u}_1, \ldots, \mathbf{u}_m$
- A simple observation to illustrate the above is $\mathbf{A}\mathbf{v}_i = \sigma_i\mathbf{u}_i$

SVD: kernel

$$\begin{bmatrix} & & \\ & \mathbf{A} & \\ & & \end{bmatrix} = \begin{bmatrix} | & & | \\ \mathbf{u}_1 & \cdots & \mathbf{u}_m \\ | & & | \end{bmatrix} \begin{bmatrix} \sigma_1 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & 0 & \cdots & 0 \\ 0 & \cdots & \sigma_m & \cdots & 0 \end{bmatrix} \begin{bmatrix} - & \mathbf{v}_1 & - \\ & \vdots & \\ - & \mathbf{v}_n & - \end{bmatrix}$$

- When $m < n$, the matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ is mapping from one vector space ($\mathbb{R}^n$) into a smaller vector space ($\mathbb{R}^m$)
- Therefore there are necessarily directions in $\mathbb{R}^n$ that are mapped to zero (the kernel of $\mathbf{A}$)
- The kernel of $\mathbf{A}$ is spanned by $\mathbf{v}_{n-m+1}, \ldots, \mathbf{v}_n$, plus any other right singular vectors with corresponding to zero singular values

SVD: range

$$\begin{bmatrix} & & \\ & \mathbf{A} & \\ & & \end{bmatrix} = \begin{bmatrix} | & & | \\ \mathbf{u}_1 & \cdots & \mathbf{u}_m \\ | & & | \end{bmatrix} \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & 0 \\ 0 & \cdots & \sigma_n \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} - & \mathbf{v}_1 & - \\ & \vdots & \\ - & \mathbf{v}_n & - \end{bmatrix}$$

- When $m > n$, the matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ is mapping from one vector space ($\mathbb{R}^n$) into a larger vector space ($\mathbb{R}^m$)
- Therefore the range of $\mathbf{A}$ necessarily has dimension less than n (in fact, the dimension of the range is equal to the number of positive singular values)
- The vector subspace in $\mathbb{R}^m$ that is orthogonal to the range is spanned by $\mathbf{u}_{m-n+1}, \ldots, \mathbf{u}_m$ plus any other left singular vectors corresponding to zero singular values

SVD: relation to eigendecomposition

- The SVD exists for all matrices $\mathbf{A} \in \mathbb{R}^{m \times n}$, whereas the eigendecomposition only applies to square matrices. However, the two decompositions are related
- Note the the matrix $\mathbf{A}^T\mathbf{A}$ is an $m \times m$ square matrix
- Using the SVD decomposition, we can see that

$$\begin{aligned} \mathbf{A}^T\mathbf{A} &= (\mathbf{U\Sigma V}^T)^T(\mathbf{U\Sigma V}^T) \\ &= \mathbf{V\Sigma^T U}^T\mathbf{U\Sigma V}^T \\ &= \mathbf{V\Sigma}^T\mathbf{\Sigma V}^T \end{aligned}$$

- The matrix $\mathbf{D} := \mathbf{\Sigma}^T\mathbf{\Sigma}$ is an $m \times m$ diagonal matrix with squared singular values σ_i^2 on the diagonal
- $\mathbf{VDV}^T$ is therefore the eigendecomposition of $\mathbf{A}^T\mathbf{A}$
- The eigenvalues are σ_i^2 (possibly with additional zeros if $m > n$) with corresponding eigenvectors $\mathbf{v}_i$

SVD: relation to eigendecomposition

- A similar derivation follows for $\mathbf{A}\mathbf{A}^T$:

$$\begin{aligned}\mathbf{A}\mathbf{A}^T &= (\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T)(\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T)^T \\ &= \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{V}\boldsymbol{\Sigma}^T\mathbf{U}^T \\ &= \mathbf{U}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^T\mathbf{U}^T\end{aligned}$$

- Now the matrix $\hat{\mathbf{D}} := \boldsymbol{\Sigma}\boldsymbol{\Sigma}^T$ is an $n \times n$ matrix with the squared singular values σ_i^2 on the diagonal
- $\mathbf{U}\hat{\mathbf{D}}\mathbf{U}^T$ is the eigendecomposition of $\mathbf{A}\mathbf{A}^T$
- The eigenvalues are σ_i^2 (possibly with additional zeros if $m > n$) with corresponding eigenvectors $\mathbf{u}_i$
- These relations also show how the SVD can be computed, by computing the eigensystems for $\mathbf{A}^T\mathbf{A}$ and $\mathbf{A}\mathbf{A}^T$