12. Linear algebra and data: Least squares and the Singular Value Decomposition (SVD)

Summary of last lecture

- Power iteration $\mathbf{x}_{n+1} = \mathbf{A}\,\mathbf{x}_n$
- Finds eigenvectors $\mathbf{A}\mathbf{x} = \lambda \mathbf{x}$
- Gram–Schmidt orthogonalisation
- QR factorisation
  - Orthogonal matrix Q such that $\mathsf{Q}^{\mathsf{T}}\mathsf{Q} = \mathsf{I}$
  - Upper-triangular matrix R

Goals for today

- Fitting functions to data

- Optimization problems
- Linear least-squares problems
- Solution using linear algebra

- The Singular-Value Decomposition (SVD)

Least squares: Fitting functions to data

Parametric functions

- Suppose we have data points (x_i, y_i)

Parametric functions

- Suppose we have data points (x_i, y_i)

- We suppose they can be fitted by a **model** $y = f(x)$

Parametric functions

- Suppose we have data points (x_i, y_i)
- We suppose they can be fitted by a **model** $y = f(x)$
- I.e. we suspect that they are related as

$$y_i = f_{\mathbf{p}}(x_i) + \epsilon_i$$

- With **noise** (randomness) ϵ_i on observation i

Parametric functions

- Suppose we have data points (x_i, y_i)
- We suppose they can be fitted by a **model** $y = f(x)$
- I.e. we suspect that they are related as

$$y_i = f_{\mathbf{p}}(x_i) + \epsilon_i$$

- With **noise** (randomness) ϵ_i on observation i
- $f_{\mathbf{p}}$ is a **parametric function** with parameter vector **p**

Parametric functions

- Suppose we have data points (x_i, y_i)

- We suppose they can be fitted by a **model** $y = f(x)$
- I.e. we suspect that they are related as

$$y_i = f_{\mathbf{p}}(x_i) + \epsilon_i$$

- With **noise** (randomness) ϵ_i on observation i
- $f_{\mathbf{p}}$ is a **parametric function** with parameter vector **p**
- Simplest case: straight line $f_{\alpha,\beta}(x) = \alpha + \beta x$

Collaboration I

Finding the best-fitting straight line

Suppose we have data (x_i, y_i) and a model $f(x) = \alpha + \beta x$.

1. How can we write down the problem to find the straight line $y = f(x)$ that “fits” the data best?
2. What type of problem is it?
3. How could we solve it?

Model of best fit

- We want to find the model that **best fits** the data
- How can we formalize this?

Model of best fit

- We want to find the model that **best fits** the data
- How can we formalize this?

- We want to find the values of the parameters $\mathbf{p}$ that **minimize** the “distance” of the data from the function

Model of best fit

- We want to find the model that **best fits** the data
- How can we formalize this?

- We want to find the values of the parameters **p** that **minimize** the “distance” of the data from the function

- To measure the distance we introduce a **loss function** or **cost function**

Model of best fit

- We want to find the model that **best fits** the data
- How can we formalize this?

- We want to find the values of the parameters $\mathbf{p}$ that **minimize** the “distance” of the data from the function

- To measure the distance we introduce a **loss function** or **cost function**
- This will measure the total “distance” between *all* the data and the function

Least squares problems

- **Loss** function:

$$\mathcal{L}(\mathbf{p}) = \sum_i \left[y_i - f_{\mathbf{p}}(t_i)\right]^2$$

Least squares problems

- **Loss** function:

$$\mathcal{L}(\mathbf{p}) = \sum_i \left[y_i - f_{\mathbf{p}}(t_i)\right]^2$$

- We want to minimise this over all vectors of parameters **p**
- E.g. over all α and β

Least squares problems

- **Loss** function:

$$\mathcal{L}(\mathbf{p}) = \sum_i \left[y_i - f_{\mathbf{p}}(t_i)\right]^2$$

- We want to minimise this over all vectors of parameters $\mathbf{p}$
- E.g. over all α and β
- **Least-squares problem**

Optimization

- This is an example of an **optimization** problem

Optimization

- This is an example of an **optimization** problem
- Optimization problems are usually **very hard**; huge field

Optimization

- This is an example of an **optimization** problem
- Optimization problems are usually **very hard**; huge field
- There are special methods for problems with certain **structure**

Optimization

- This is an example of an **optimization** problem
- Optimization problems are usually **very hard**; huge field
- There are special methods for problems with certain **structure**

- There are many numerical methods for optimization
- Sometimes analytical solutions are possible – e.g. linear least squares

Matrix formulation of linear least squares

- We want to minimise $\mathcal{L}(\mathbf{p}) = \sum_i r_i^2$
- Where $r_i := y_i - f_{\mathbf{p}}(t_i)$ is the ith **residual**

Matrix formulation of linear least squares

- We want to minimise $\mathcal{L}(\mathbf{p}) = \sum_i r_i^2$
- Where $r_i := y_i - f_{\mathbf{p}}(t_i)$ is the ith **residual**
- Look for a matrix formulation: $\mathcal{L} = \sum_i r_i^2 = \mathbf{r}^\mathsf{T}\mathbf{r}$

Matrix formulation of linear least squares

- We want to minimise $\mathcal{L}(\mathbf{p}) = \sum_i r_i^2$
- Where $r_i := y_i - f_{\mathbf{p}}(t_i)$ is the ith **residual**
- Look for a matrix formulation: $\mathcal{L} = \sum_i r_i^2 = \mathbf{r}^\mathsf{T}\mathbf{r}$
- The vector **r** of all residuals is given by

$$\mathbf{r} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_m \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_m \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} - \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} =: \mathsf{A}\mathbf{x} - \mathbf{b}$$

where $\mathbf{x} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$ are the unknowns

Matrix formulation II

- Ideally we would like to solve $A\mathbf{x} = \mathbf{b}$

Matrix formulation II

- Ideally we would like to solve $A\mathbf{x} = \mathbf{b}$
- But we have an **overdetermined system**:
- There are more equations (n) than unknowns (2)

Matrix formulation II

- Ideally we would like to solve $A\mathbf{x} = \mathbf{b}$
- But we have an **overdetermined system**:
- There are more equations (n) than unknowns (2)

- What can we do?

Matrix formulation II

- Ideally we would like to solve $A\mathbf{x} = \mathbf{b}$
- But we have an **overdetermined system**:
- There are more equations (n) than unknowns (2)

- What can we do?
- The best we can do is to *minimize* $\mathbf{r}^{\mathsf{T}}\mathbf{r}$

General linear least squares

- General case: want to “solve” $\mathsf{A}\mathbf{x} = \mathbf{b}$
- For a **tall and narrow** matrix A, i.e. with $m > n$

General linear least squares

- General case: want to "solve" $\mathsf{A}\mathbf{x} = \mathbf{b}$
- For a **tall and narrow** matrix A, i.e. with $m > n$
- We try to minimize

$$\left\|\mathsf{A}\mathbf{x} - \mathbf{b}\right\|^2$$

General linear least squares

- General case: want to “solve” A$\mathbf{x} = \mathbf{b}$
- For a **tall and narrow** matrix A, i.e. with $m > n$
- We try to minimize

$$\|A\mathbf{x} - \mathbf{b}\|^2$$

- Example: fit a polynomial of degree $< n$ to $(n+1)$ points

Geometrical interpretation

- Suppose **x** ranges over all of $\mathbb{R}^n$
- Then A**x** ranges over the column space / range of A

Geometrical interpretation

- Suppose **x** ranges over all of $\mathbb{R}^n$
- Then A**x** ranges over the column space / range of A
- The **column space** of A is the vector space spanned by the columns of A

Geometrical interpretation

- Suppose **x** ranges over all of $\mathbb{R}^n$
- Then A**x** ranges over the column space / range of A
- The **column space** of A is the vector space spanned by the columns of A
- The column space is a hyperplane

Geometrical interpretation

- Suppose **x** ranges over all of $\mathbb{R}^n$
- Then A**x** ranges over the column space / range of A
- The **column space** of A is the vector space spanned by the columns of A

- The column space is a hyperplane
- We are looking for an **x** whose image A**x** is **closest** to **b**

Collaboration II

Solving the least squares problem

We want $\mathbf{x}^*$ to be the solution that minimises $\|A\mathbf{x} - \mathbf{b}\|^2$.

1. Draw a sketch of this. Geometrically / intuitively, what condition should we satisfy?
2. How could we solve this problem?

Solving linear least squares

- We want to find $\mathbf{x}^*$ minimising $\|A\mathbf{x} - \mathbf{b}\|^2$

Solving linear least squares

- We want to find $\mathbf{x}^*$ minimising $\|A\mathbf{x} - \mathbf{b}\|^2$
- Intuitively this occurs when $\mathbf{r} := A\mathbf{x}^* - \mathbf{b}$ is **perpendicular** to the column space

Solving linear least squares

- We want to find $\mathbf{x}^*$ minimising $\|\mathrm{A}\mathbf{x} - \mathbf{b}\|^2$
- Intuitively this occurs when $\mathbf{r} := \mathrm{A}\mathbf{x}^* - \mathbf{b}$ is **perpendicular** to the column space
- Recall: $\|\mathbf{x}\|^2 = \mathbf{x}^\mathsf{T}\mathbf{x}$

Solving linear least squares

- We want to find $\mathbf{x}^*$ minimising $\|\mathrm{A}\mathbf{x} - \mathbf{b}\|^2$
- Intuitively this occurs when $\mathbf{r} := \mathrm{A}\mathbf{x}^* - \mathbf{b}$ is **perpendicular** to the column space
- Recall: $\|\mathbf{x}\|^2 = \mathbf{x}^\mathsf{T}\mathbf{x}$

- Let's look at other vectors in the subspace by taking any displacement $\mathbf{y}$
- Look at $\|\mathrm{A}(\mathbf{x}^* + \mathbf{y}) - \mathbf{b}\|^2$

Solving linear least squares II

- We have

$$\|\mathsf{A}(\mathbf{x}^* + \mathbf{y}) - \mathbf{b}\|^2$$

$$= \|(\mathsf{A}\mathbf{x}^* - \mathbf{b}) + \mathsf{A}\mathbf{y}\|^2 = [(\mathsf{A}\mathbf{x}^* - \mathbf{b}) + \mathsf{A}\mathbf{y}]^\mathsf{T}[(\mathsf{A}\mathbf{x}^* - \mathbf{b}) + \mathsf{A}\mathbf{y}]$$

Solving linear least squares II

- We have

$$\|A(\mathbf{x}^* + \mathbf{y}) - \mathbf{b}\|^2$$

$$= \|(A\mathbf{x}^* - \mathbf{b}) + A\mathbf{y}\|^2 = [(A\mathbf{x}^* - \mathbf{b}) + A\mathbf{y}]^\mathsf{T}[(A\mathbf{x}^* - \mathbf{b}) + A\mathbf{y}]$$

$$= (A\mathbf{x}^* - \mathbf{b})^\mathsf{T}(A\mathbf{x}^* - \mathbf{b}) + 2\mathbf{y}^\mathsf{T}A^\mathsf{T}(A\mathbf{x}^* - \mathbf{b}) + (A\mathbf{y})^\mathsf{T}(A\mathbf{y})$$

Solving linear least squares II

- We have

$$\|A(\mathbf{x}^* + \mathbf{y}) - \mathbf{b}\|^2$$

$$= \|(A\mathbf{x}^* - \mathbf{b}) + A\mathbf{y}\|^2 = [(A\mathbf{x}^* - \mathbf{b}) + A\mathbf{y}]^\mathsf{T}[(A\mathbf{x}^* - \mathbf{b}) + A\mathbf{y}]$$

$$= (A\mathbf{x}^* - \mathbf{b})^\mathsf{T}(A\mathbf{x}^* - \mathbf{b}) + 2\mathbf{y}^\mathsf{T}A^\mathsf{T}(A\mathbf{x}^* - \mathbf{b}) + (A\mathbf{y})^\mathsf{T}(A\mathbf{y})$$

- Here we used $\mathbf{y}^\mathsf{T}\mathbf{z} = \mathbf{z}^\mathsf{T}\mathbf{y}$, so $\mathbf{y}^\mathsf{T}\mathbf{z} + \mathbf{z}^\mathsf{T}\mathbf{y} = 2\mathbf{y}^\mathsf{T}\mathbf{z}$

$$= \|A\mathbf{x}^* - \mathbf{b}\|^2 + \|A\mathbf{y}\|^2 + 2\mathbf{y}^\mathsf{T}A^\mathsf{T}(A\mathbf{x}^* - \mathbf{b})$$

Solving linear least squares II

- We have $\|\mathsf{A}\mathbf{x}^* - \mathbf{b}\|^2 + \|\mathsf{A}\mathbf{y}\|^2 + 2\mathbf{y}^\mathsf{T}\mathsf{A}^\mathsf{T}(\mathsf{A}\mathbf{x}^* - \mathbf{b})$

Solving linear least squares II

- We have $\|A\mathbf{x}^* - \mathbf{b}\|^2 + \|A\mathbf{y}\|^2 + 2\mathbf{y}^\mathsf{T} A^\mathsf{T}(A\mathbf{x}^* - \mathbf{b})$
- Take $\mathbf{x}^*$ such that $A^\mathsf{T}(A\mathbf{x}^* - \mathbf{b}) = \mathbf{0}$

Solving linear least squares II

- We have $\|A\mathbf{x}^* - \mathbf{b}\|^2 + \|A\mathbf{y}\|^2 + 2\mathbf{y}^T A^T (A\mathbf{x}^* - \mathbf{b})$
- Take $\mathbf{x}^*$ such that $A^T(A\mathbf{x}^* - \mathbf{b}) = \mathbf{0}$
- Then for any $\mathbf{y}$ we have

$$\|A(\mathbf{x}^* + \mathbf{y}) - \mathbf{b}\|^2 \geq \|A\mathbf{x}^* - \mathbf{b}\|^2$$

Solving linear least squares II

- We have $\|A\mathbf{x}^* - \mathbf{b}\|^2 + \|A\mathbf{y}\|^2 + 2\mathbf{y}^T A^T (A\mathbf{x}^* - \mathbf{b})$
- Take $\mathbf{x}^*$ such that $A^T(A\mathbf{x}^* - \mathbf{b}) = \mathbf{0}$
- Then for any $\mathbf{y}$ we have

$$\|A(\mathbf{x}^* + \mathbf{y}) - \mathbf{b}\|^2 \geq \|A\mathbf{x}^* - \mathbf{b}\|^2$$

- So $\mathbf{x}^*$ is the minimizer !

Solving linear least squares II

- We have $\|A\mathbf{x}^* - \mathbf{b}\|^2 + \|A\mathbf{y}\|^2 + 2\mathbf{y}^T A^T (A\mathbf{x}^* - \mathbf{b})$
- Take $\mathbf{x}^*$ such that $A^T(A\mathbf{x}^* - \mathbf{b}) = \mathbf{0}$
- Then for any $\mathbf{y}$ we have

$$\|A(\mathbf{x}^* + \mathbf{y}) - \mathbf{b}\|^2 \geq \|A\mathbf{x}^* - \mathbf{b}\|^2$$

- So $\mathbf{x}^*$ is the minimizer !
- Hence the unique solution of the least squares problem is given by the solution $\mathbf{x}$ of

$$A^T A\mathbf{x} = A^T\mathbf{b}$$

Solving linear least squares III

- We have reduced the linear least squares problem to solving a linear system:

$$A^T A\mathbf{x} = A^T\mathbf{b}$$

Solving linear least squares III

- We have reduced the linear least squares problem to solving a linear system:

$$A^T A\mathbf{x} = A^T\mathbf{b}$$

- Called the **normal equations**

Solving linear least squares III

- We have reduced the linear least squares problem to solving a linear system:

$$A^T A\mathbf{x} = A^T\mathbf{b}$$

- Called the **normal equations**
- $A^T A$ is a *square* matrix

Solving linear least squares III

- We have reduced the linear least squares problem to solving a linear system:

$$A^T A \mathbf{x} = A^T \mathbf{b}$$

- Called the **normal equations**
- $A^T A$ is a *square* matrix
- This is solvable if A is **full rank** – columns span the space

Solving linear least squares III

- We have reduced the linear least squares problem to solving a linear system:

$$A^T A\mathbf{x} = A^T\mathbf{b}$$

- Called the **normal equations**
- $A^T A$ is a *square* matrix
- This is solvable if A is **full rank** – columns span the space

- Note that the condition $A^T(A\mathbf{x} - \mathbf{b}) = 0$ corresponds to $A\mathbf{x} - \mathbf{b}$ being perpendicular to each column of A !

Pseudoinverse

- Mathematically: $\mathbf{x} = (\mathsf{A}^\mathsf{T}\mathsf{A})^{-1}\mathsf{A}^\mathsf{T}\mathbf{b}$

Pseudoinverse

- Mathematically: $\mathbf{x} = (\mathsf{A}^\mathsf{T}\mathsf{A})^{-1}\mathsf{A}^\mathsf{T}\mathbf{b}$
- Notation: $\mathbf{x} = \mathsf{A}^+\mathbf{b}$

- $\mathsf{A}^+ := (\mathsf{A}^\mathsf{T}\mathsf{A})^{-1}\mathsf{A}^\mathsf{T}$ is called the **pseudo-inverse** of A

Pseudoinverse

- Mathematically: $\mathbf{x} = (\mathsf{A}^\mathsf{T}\mathsf{A})^{-1}\mathsf{A}^\mathsf{T}\mathbf{b}$
- Notation: $\mathbf{x} = \mathsf{A}^{+}\mathbf{b}$

- $\mathsf{A}^{+} := (\mathsf{A}^\mathsf{T}\mathsf{A})^{-1}\mathsf{A}^\mathsf{T}$ is called the **pseudo-inverse** of A
- $\mathsf{A}^\mathsf{T}\mathsf{A}$ is invertible iff A has full rank

Solving the normal equations

- It is possible to solve the normal equations using elimination, but to do so we need to construct $A^T A$

Solving the normal equations

- It is possible to solve the normal equations using elimination, but to do so we need to construct $A^T A$
- Slightly more efficient version of elimination for **symmetric** matrices: **Cholesky factorization**

Solving the normal equations

- It is possible to solve the normal equations using elimination, but to do so we need to construct $A^T A$
- Slightly more efficient version of elimination for **symmetric** matrices: **Cholesky factorization**
- Alternative: factorize $A = QR$

Solving the normal equations

- It is possible to solve the normal equations using elimination, but to do so we need to construct $\mathsf{A}^\mathsf{T}\mathsf{A}$
- Slightly more efficient version of elimination for **symmetric** matrices: **Cholesky factorization**

- Alternative: factorize $\mathsf{A} = \mathsf{QR}$
- Need generalization of QR when A tall and narrow, $m \times n$

Solving the normal equations

- It is possible to solve the normal equations using elimination, but to do so we need to construct $\mathsf{A}^\mathsf{T}\mathsf{A}$
- Slightly more efficient version of elimination for **symmetric** matrices: **Cholesky factorization**

- Alternative: factorize $\mathsf{A} = \mathsf{QR}$
- Need generalization of QR when A tall and narrow, $m \times n$
- **Thin** QR factorization: $\mathsf{A} = \mathsf{QR}$ where
  - Q is $(m \times n)$ with orthonormal columns
  - R is $(n \times n)$ upper triangular

Solving the normal equations

- It is possible to solve the normal equations using elimination, but to do so we need to construct $\mathsf{A}^\mathsf{T}\mathsf{A}$
- Slightly more efficient version of elimination for **symmetric** matrices: **Cholesky factorization**

- Alternative: factorize $\mathsf{A} = \mathsf{QR}$
- Need generalization of QR when A tall and narrow, $m \times n$
- **Thin** QR factorization: $\mathsf{A} = \mathsf{QR}$ where
  - Q is $(m \times n)$ with orthonormal columns
  - R is $(n \times n)$ upper triangular
- If A is of full rank then R is non-singular

Solving normal equations II

- We want to solve $A^T A\mathbf{x} = A^T\mathbf{b}$
- Substitute $A = QR$

Solving normal equations II

- We want to solve $A^T A\mathbf{x} = A^T\mathbf{b}$
- Substitute $A = QR$
- So $R^T Q^T Q R\,\mathbf{x} = R^T Q^T\mathbf{b}$

Solving normal equations II

- We want to solve $A^T A\mathbf{x} = A^T\mathbf{b}$
- Substitute $A = QR$

- So $R^T Q^T QR\,\mathbf{x} = R^T Q^T\mathbf{b}$
- So $R^T R\,\mathbf{x} = R^T Q^T\mathbf{b}$

Solving normal equations II

- We want to solve $A^T A\mathbf{x} = A^T\mathbf{b}$
- Substitute $A = QR$

- So $R^T Q^T QR\,\mathbf{x} = R^T Q^T\mathbf{b}$
- So $R^T R\,\mathbf{x} = R^T Q^T\mathbf{b}$
- So $R\,\mathbf{x} = Q^T\mathbf{b}$

Solving normal equations II

- We want to solve $A^T A\mathbf{x} = A^T\mathbf{b}$
- Substitute $A = QR$

- So $R^T Q^T QR\,\mathbf{x} = R^T Q^T\mathbf{b}$
- So $R^T R\,\mathbf{x} = R^T Q^T\mathbf{b}$
- So $R\,\mathbf{x} = Q^T\mathbf{b}$
- So $\mathbf{x} = R^{-1} Q^T\mathbf{b}$

Solving normal equations II

- We want to solve $A^T A\mathbf{x} = A^T\mathbf{b}$
- Substitute $A = QR$

- So $R^T Q^T QR\,\mathbf{x} = R^T Q^T\mathbf{b}$
- So $R^T R\,\mathbf{x} = R^T Q^T\mathbf{b}$
- So $R\,\mathbf{x} = Q^T\mathbf{b}$
- So $\mathbf{x} = R^{-1} Q^T\mathbf{b}$

- Solve $R\mathbf{x} = Q^T\mathbf{b}$ by backsubstitution

Backslash for solving least squares

- Backslash in Julia is overloaded to give this least-squares solution when A is an $(m \times n)$ matrix (
- I.e. `A \ b` gives $\mathsf{A}^{+}\mathbf{b} = (\mathsf{A}^{\mathsf{T}}\mathsf{A})^{-1}\mathsf{A}^{\mathsf{T}}\mathbf{b}$

Backslash for solving least squares

- Backslash in Julia is overloaded to give this least-squares solution when A is an $(m \times n)$ matrix (
- I.e. `A \ b` gives $\mathsf{A}^{+}\mathbf{b} = (\mathsf{A}^{\mathsf{T}}\mathsf{A})^{-1}\mathsf{A}^{\mathsf{T}}\mathbf{b}$
- It is not necessarily clear that overloading \ in this way is a good idea!

Backslash for solving least squares

- Backslash in Julia is overloaded to give this least-squares solution when A is an $(m \times n)$ matrix (
- I.e. `A \ b` gives $\mathsf{A}^{+}\mathbf{b} = (\mathsf{A}^{\mathsf{T}}\mathsf{A})^{-1}\mathsf{A}^{\mathsf{T}}\mathbf{b}$
- It is not necessarily clear that overloading \ in this way is a good idea!
- E.g. For a simple linear fit, use \ with above matrix

The Singular-Value Decomposition (SVD)

Action of a matrix

- Consider an arbitrary $(m \times n)$ matrix A

Action of a matrix

- Consider an arbitrary $(m \times n)$ matrix A
- Let's study its action on the unit sphere $\mathbb{S}_n$ in $\mathbb{R}^n$
- $\mathbb{S}_n := \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_2 = 1\}$

Action of a matrix

- Consider an arbitrary $(m \times n)$ matrix A
- Let's study its action on the unit sphere $\mathbb{S}_n$ in $\mathbb{R}^n$
- $\mathbb{S}_n := \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_2 = 1\}$

- A visualisation suggests that $A(\mathbb{S}_n)$ is an **ellipsoid** (hyper-ellipse)
- I.e. a sphere that is stretched and then rotated

Action of a matrix

- Consider an arbitrary $(m \times n)$ matrix A
- Let's study its action on the unit sphere $\mathbb{S}_n$ in $\mathbb{R}^n$
- $\mathbb{S}_n := \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_2 = 1\}$

- A visualisation suggests that $A(\mathbb{S}_n)$ is an **ellipsoid** (hyper-ellipse)
- I.e. a sphere that is stretched and then rotated

- Key quantities:
  - Lengths σ_i of semi-axes – stretches
  - Directions $\mathbf{u}_i$ of stretches

Direction of maximum stretch

- Which is the direction with *maximal* stretch?
- This is a special direction associated to the matrix

Direction of maximum stretch

- Which is the direction with *maximal* stretch?
- This is a special direction associated to the matrix
- We need to solve

$$\operatorname{argmax}_{\mathbf{x}} \frac{\|A\mathbf{x}\|}{\|\mathbf{x}\|}$$

Direction of maximum stretch

- Which is the direction with *maximal* stretch?
- This is a special direction associated to the matrix

- We need to solve

$$\operatorname{argmax}_{\mathbf{x}} \frac{\|A\mathbf{x}\|}{\|\mathbf{x}\|}$$

- Another optimisation problem
- Also solvable using linear algebra

Action of a matrix II

- Let's think about the image of a sphere under a matrix

Action of a matrix II

- Let's think about the image of a sphere under a matrix
- Suppose $\mathbf{x} \in \mathbb{S}_n$; define $\mathbf{y} := A\mathbf{x}$ to be its image

Action of a matrix II

- Let's think about the image of a sphere under a matrix
- Suppose $\mathbf{x} \in \mathbb{S}_n$; define $\mathbf{y} := \mathsf{A}\mathbf{x}$ to be its image
- Suppose that A is square and invertible for now

Action of a matrix II

- Let's think about the image of a sphere under a matrix
- Suppose $\mathbf{x} \in \mathbb{S}_n$; define $\mathbf{y} := \mathsf{A}\mathbf{x}$ to be its image
- Suppose that A is square and invertible for now
- Then $\quad \|\mathsf{A}^{-1}\mathbf{y}\|^2 = 1$
- So $\quad \mathbf{y}^\mathsf{T}(\mathsf{A}^{-1})^\mathsf{T}\mathsf{A}^{-1}\mathbf{y} = 1$

Action of a matrix II

- Let's think about the image of a sphere under a matrix
- Suppose $\mathbf{x} \in \mathbb{S}_n$; define $\mathbf{y} := \mathsf{A}\mathbf{x}$ to be its image
- Suppose that A is square and invertible for now
- Then $\|\mathsf{A}^{-1}\mathbf{y}\|^2 = 1$
- So $\mathbf{y}^\mathsf{T}(\mathsf{A}^{-1})^\mathsf{T}\mathsf{A}^{-1}\mathbf{y} = 1$

- Hence $\mathbf{y}^\mathsf{T}\mathsf{S}\mathbf{y} = 1$
- Where $\mathsf{S} := (\mathsf{A}^{-1})^\mathsf{T}(\mathsf{A}^{-1})$ is symmetric

Spectral theorem and eigen-decomposition

- The spectral theorem for symmetric matrices tells us that there is a basis of orthogonal eigenvectors $\mathbf{v}_i$ for S
- Hence we have $\mathsf{SQ} = \mathsf{QL}$, where

$$\mathsf{Q} := \left(\mathbf{v}_1 \middle| \mathbf{v}_2 \middle| \cdots \middle| \mathbf{v}_n\right)$$

$$\mathsf{L} := \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_m \end{bmatrix}$$

Spectral theorem and eigen-decomposition

- The spectral theorem for symmetric matrices tells us that there is a basis of orthogonal eigenvectors $\mathbf{v}_i$ for S
- Hence we have $\mathsf{SQ} = \mathsf{QL}$, where

$$\mathsf{Q} := (\mathbf{v}_1 | \mathbf{v}_2 | \cdots | \mathbf{v}_n)$$

$$\mathsf{L} := \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_m \end{bmatrix}$$

- So $\quad \mathsf{S} = \mathsf{QLQ}^{\mathsf{T}}$

Action of a matrix III

- So $\mathbf{y}^T QLQ^T \mathbf{y} = 1$

Action of a matrix III

- So $\mathbf{y}^{\mathsf{T}}\mathsf{QLQ}^{\mathsf{T}}\mathbf{y} = 1$
- Hence $\mathbf{z}^{\mathsf{T}}\mathsf{L}\mathbf{z} = 1$

  where $\mathbf{z} := \mathsf{Q}^{\mathsf{T}}\mathbf{y}$

Action of a matrix III

- So $\mathbf{y}^{\mathsf{T}}\mathsf{Q}\mathsf{L}\mathsf{Q}^{\mathsf{T}}\mathbf{y} = 1$
- Hence $\mathbf{z}^{\mathsf{T}}\mathsf{L}\mathbf{z} = 1$

  where $\mathbf{z} := \mathsf{Q}^{\mathsf{T}}\mathbf{y}$
- So $\lambda_1 z_1^2 + \cdots + \lambda_n z_n^2 = 1$

Action of a matrix III

- So $\mathbf{y}^{\mathsf{T}}\mathsf{QLQ}^{\mathsf{T}}\mathbf{y} = 1$
- Hence $\mathbf{z}^{\mathsf{T}}\mathsf{L}\mathbf{z} = 1$

  where $\mathbf{z} := \mathsf{Q}^{\mathsf{T}}\mathbf{y}$
- So $\lambda_1 z_1^2 + \cdots + \lambda_n z_n^2 = 1$

- Can show $(\mathsf{A}^{\mathsf{T}}\mathsf{A})^{-1}$ has eigenvalues $\lambda_i > 0$ (exercise)

Action of a matrix III

- So $\mathbf{y}^{\mathsf{T}}\mathsf{QLQ}^{\mathsf{T}}\mathbf{y} = 1$
- Hence $\mathbf{z}^{\mathsf{T}}\mathsf{L}\mathbf{z} = 1$

  where $\mathbf{z} := \mathsf{Q}^{\mathsf{T}}\mathbf{y}$
- So $\lambda_1 z_1^2 + \cdots + \lambda_n z_n^2 = 1$

- Can show $(\mathsf{A}^{\mathsf{T}}\mathsf{A})^{-1}$ has eigenvalues $\lambda_i > 0$ (exercise)
- Hence $\mathbf{z}$ lies on an **ellipse**, with semi-axis lengths $\sigma_i := \sqrt{\lambda_i}$

Action of a matrix III

- So $\mathbf{y}^\mathsf{T} \mathsf{Q} \mathsf{L} \mathsf{Q}^\mathsf{T} \mathbf{y} = 1$
- Hence $\mathbf{z}^\mathsf{T} \mathsf{L} \mathbf{z} = 1$

  where $\mathbf{z} := \mathsf{Q}^\mathsf{T} \mathbf{y}$
- So $\lambda_1 z_1^2 + \cdots + \lambda_n z_n^2 = 1$

- Can show $(\mathsf{A}^\mathsf{T} \mathsf{A})^{-1}$ has eigenvalues $\lambda_i > 0$ (exercise)
- Hence $\mathbf{z}$ lies on an **ellipse**, with semi-axis lengths $\sigma_i := \sqrt{\lambda_i}$
- Thus $\mathbf{y}$ lies on a rotated ellipse !

Singular values and singular vectors

- σ_i are **singular values** of A – this is a terrible name!

Singular values and singular vectors

- σ_i are **singular values** of A – this is a terrible name!

- The $\mathbf{u}_i$, the columns of Q, are the **left singular vectors**
- I.e. the directions in which the ellipse is stretched

Singular values and singular vectors

- σ_i are **singular values** of A – this is a terrible name!

- The $\mathbf{u}_i$, the columns of Q, are the **left singular vectors**
- I.e. the directions in which the ellipse is stretched

- $\mathbf{v}_i$ such that $A\mathbf{v}_i = \sigma_i \mathbf{u}_i$ are the **right singular vectors**

Singular-value decomposition (SVD)

- *Any* $(m \times n)$ matrix A has an SVD:

$$\mathsf{A} = \mathsf{U}\Sigma\mathsf{V}^{\mathsf{T}}$$

  - the columns of U are $\mathbf{u}_i$
  - the columns of V are $\mathbf{v}_i$
  - Σ is diagonal matrix with σ_i on diagonal

Singular-value decomposition (SVD)

- *Any* $(m \times n)$ matrix A has an SVD:

$$\mathsf{A} = \mathsf{U}\Sigma\mathsf{V}^{\mathsf{T}}$$

  - the columns of U are $\mathbf{u}_i$
  - the columns of V are $\mathbf{v}_i$
  - Σ is diagonal matrix with σ_i on diagonal

- Note: only square, diagonalisable matrices have eigen-decompositions
- But *any* matrix has an SVD

Singular-value decomposition (SVD)

- *Any* $(m \times n)$ matrix A has an SVD:

$$\mathsf{A} = \mathsf{U}\Sigma\mathsf{V}^\mathsf{T}$$

  - the columns of U are $\mathbf{u}_i$
  - the columns of V are $\mathbf{v}_i$
  - Σ is diagonal matrix with σ_i on diagonal

- Note: only square, diagonalisable matrices have eigen-decompositions
- But *any* matrix has an SVD
- The SVD is often more useful than an eigen-decomposition

Calculating the SVD

- We can calculate (most of) the SVD

Calculating the SVD

- We can calculate (most of) the SVD
- Form $B := A^T A$

Calculating the SVD

- We can calculate (most of) the SVD

- Form $B := A^T A$
- Calculate eigenvalues and eigenvectors of B

Calculating the SVD

- We can calculate (most of) the SVD

- Form $B := A^T A$
- Calculate eigenvalues and eigenvectors of B
- **Exercise**: Relate these to the singular values and singular vectors of A

Rank of a matrix and best approximation

- Number of non-zero singular values = column rank = **rank**

Rank of a matrix and best approximation

- Number of non-zero singular values = column rank = **rank**

- The SVD gives the **best approximation** to A by a matrix of rank k:

$$\mathsf{A}_k := \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^{\mathsf{T}}$$

Rank of a matrix and best approximation

- Number of non-zero singular values = column rank = **rank**

- The SVD gives the **best approximation** to A by a matrix of rank k:

$$\mathsf{A}_k := \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^{\mathsf{T}}$$

- For a data matrix the SVD gives the best approximation by a k-dimensional subspace

Rank of a matrix and best approximation

- Number of non-zero singular values = column rank = **rank**

- The SVD gives the **best approximation** to A by a matrix of rank k:

$$\mathsf{A}_k := \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^\mathsf{T}$$

- For a data matrix the SVD gives the best approximation by a k-dimensional subspace
- Use for **low-rank approximation** of data
- PCA (Principal Component Analysis) in statistics

Rank of a matrix and best approximation

- Number of non-zero singular values = column rank = **rank**

- The SVD gives the **best approximation** to A by a matrix of rank k:

$$\mathsf{A}_k := \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^{\mathsf{T}}$$

- For a data matrix the SVD gives the best approximation by a k-dimensional subspace
- Use for **low-rank approximation** of data
- PCA (Principal Component Analysis) in statistics

Summary

- Linear least squares for overdetermined systems
- Solvable using linear algebra
- Solution given by normal equations – linear system
- Solve using QR decomposition

- The SVD gives a matrix as rotation + stretch + rotation
- Closely related to eigendecomposition of $A^T A$

- Applications to data compression