COMP3223 Foundations of ML: Dimensionality reduction

by

Principal Components Analysis (PCA)

Srinandan Dasmahapatra

Are more complex models better?

- With large number of features p, accidental correlations can obscure genuine patterns.

Are more complex models better?

- With large number of features p, accidental correlations can obscure genuine patterns.
- In minimising (say) SSR you may be trying to reduce residuals to "noise" at the expense of "signal". *Eg.*, linear model $y^n = f(\boldsymbol{x}^n; \boldsymbol{w}) := \sum_{i=1}^{p} w_i x_i^n$

$$\sum_{n=1}^{N} (r^n)^2 \quad = \quad \sum (y^n - f(\boldsymbol{x}^n; \boldsymbol{w}))^2$$

$$r^n = y^n - (w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_p x_p)$$

Are more complex models better?

- With large number of features p, accidental correlations can obscure genuine patterns.
- In minimising (say) SSR you may be trying to reduce residuals to "noise" at the expense of "signal". *Eg.*, linear model $y^n = f(\boldsymbol{x}^n; \boldsymbol{w}) := \sum_{i=1}^{p} w_i x_i^n$

$$\sum_{n=1}^{N} (r^n)^2 \;=\; \sum (y^n - f(\boldsymbol{x}^n; \boldsymbol{w}))^2$$

$$r^n = y^n - (w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_p x_p)$$

- In trying to minimise the residuals, you may be adjusting coefficients w_k that correspond to features irrelevant to the actual signal.

Are more complex models better?

- With large number of features p, accidental correlations can obscure genuine patterns.
- In minimising (say) SSR you may be trying to reduce residuals to "noise" at the expense of "signal". *Eg.*, linear model $y^n = f(\boldsymbol{x}^n; \boldsymbol{w}) := \sum_{i=1}^{p} w_i x_i^n$

$$\sum_{n=1}^{N} (r^n)^2 \;=\; \sum (y^n - f(\boldsymbol{x}^n; \boldsymbol{w}))^2$$

$$r^n = y^n - (w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_p x_p)$$

- In trying to minimise the residuals, you may be adjusting coefficients w_k that correspond to features irrelevant to the actual signal.
- This is the $p > N$ problem

Revision: linear regression

- Linear model: $y = Xw$, X design matrix, $\widehat{w} = \left(X^{\mathsf{T}}X\right)^{-1} X^{\mathsf{T}}y$; also w_0 is the difference between averages of inputs and outputs:

$$\begin{aligned} 0 = \sum_{n=1}^{N} r^n &= \sum_n y^n - N w_0 - \sum_n \left(w_1 x_1^n + \cdots + w_p x_p^n\right) \\ \implies w_0 &= \langle y \rangle - \sum_{i=1}^{p} w_i \langle x_i \rangle \end{aligned}$$

Revision: linear regression

- Linear model: $y = Xw$, X design matrix, $\widehat{w} = \left(X^{\mathsf{T}}X\right)^{-1}X^{\mathsf{T}}y$; also w_0 is the difference between averages of inputs and outputs:

$$\begin{aligned} 0 = \sum_{n=1}^{N} r^n &= \sum_n y^n - N w_0 - \sum_n \left(w_1 x_1^n + \cdots + w_p x_p^n\right) \\ \implies w_0 &= \langle y \rangle - \sum_{i=1}^{p} w_i \langle x_i \rangle \end{aligned}$$

- **Centering**: $\widetilde{y}^n = y^n - \langle y \rangle$, $\widetilde{x}_i^n = x_i^n - \langle x_i \rangle$ and drop the column of ones from design matrix to form $\widetilde{X}$ with matrix elements $\left(\widetilde{X}\right)_{nj} = x_j^n - \langle x_j \rangle$.

Revision: linear regression

- Linear model: $y = Xw$, X design matrix, $\widehat{w} = \left(X^{\mathsf{T}}X\right)^{-1}X^{\mathsf{T}}y$; also w_0 is the difference between averages of inputs and outputs:

$$0 = \sum_{n=1}^{N} r^n = \sum_n y^n - Nw_0 - \sum_n \left(w_1 x_1^n + \cdots + w_p x_p^n\right)$$

$$\implies w_0 = \langle y \rangle - \sum_{i=1}^{p} w_i \langle x_i \rangle$$

- **Centering**: $\tilde{y}^n = y^n - \langle y \rangle$, $\tilde{x}_i^n = x_i^n - \langle x_i \rangle$ and drop the column of ones from design matrix to form $\tilde{X}$ with matrix elements $\left(\tilde{X}\right)_{nj} = x_j^n - \langle x_j \rangle$.
- $\frac{1}{N}\tilde{X}^{\mathsf{T}}\left(\tilde{y} - \tilde{X}\widehat{w}\right) = 0$ implies $\widehat{w} = \left(\frac{1}{N}\left(\tilde{X}^{\mathsf{T}}\tilde{X}\right)\right)^{-1}\left(\frac{1}{N}\tilde{X}^{\mathsf{T}}y\right)$, so

$$\widehat{w} = [\mathrm{cov}(X)^{-1}][\mathrm{cov}(X, y)].$$

Covariance as trace suggests reduction of matrix size

- $\widetilde{y}^n = y^n - \langle y \rangle, \widetilde{x}_i^n = x_i^n - \langle x_i \rangle$ and drop the column of ones from design matrix to form $\tilde{X}$ with matrix elements $(\tilde{X})_{nj} = x_j^n - \langle x_j \rangle$.

Covariance as trace suggests reduction of matrix size

- $\tilde{y}^n = y^n - \langle y \rangle, \tilde{x}_i^n = x_i^n - \langle x_i \rangle$ and drop the column of ones from design matrix to form $\tilde{X}$ with matrix elements $(\tilde{X})_{nj} = x_j^n - \langle x_j \rangle$.
- $(n \times n)$ and $(p \times p)$ matrices formed out of $\tilde{X}$:

$$(\tilde{X}\tilde{X}^T)_{mn} = \sum_{j=1}^{p} (\tilde{X})_{mj}(\tilde{X}^T)_{jn} = \sum_{j=1}^{p} (x_j^m - \langle x_j \rangle)(x_j^n - \langle x_j \rangle)$$

$$(\tilde{X}^T\tilde{X})_{ij} = \sum_{n=1}^{N} (\tilde{X}^T)_{in}(\tilde{X})_{nj} = \sum_{n=1}^{N} (x_i^n - \langle x_i \rangle)(x_j^n - \langle x_j \rangle)$$

Covariance as trace suggests reduction of matrix size

- $\tilde{y}^n = y^n - \langle y \rangle, \tilde{x}_i^n = x_i^n - \langle x_i \rangle$ and drop the column of ones from design matrix to form $\tilde{X}$ with matrix elements $(\tilde{X})_{nj} = x_j^n - \langle x_j \rangle$.
- $(n \times n)$ and $(p \times p)$ matrices formed out of $\tilde{X}$:

$$(\tilde{X}\tilde{X}^T)_{mn} = \sum_{j=1}^{p} (\tilde{X})_{mj}(\tilde{X}^T)_{jn} = \sum_{j=1}^{p} (x_j^m - \langle x_j \rangle)(x_j^n - \langle x_j \rangle)$$

$$(\tilde{X}^T\tilde{X})_{ij} = \sum_{n=1}^{N} (\tilde{X}^T)_{in}(\tilde{X})_{nj} = \sum_{n=1}^{N} (x_i^n - \langle x_i \rangle)(x_j^n - \langle x_j \rangle)$$

- Trace of $(\tilde{X}\tilde{X}^T)$ equals trace$(\tilde{X}^T\tilde{X})$:

$$\sum_{n=1}^{N}\sum_{j=1}^{p} (\tilde{X})_{mj}(\tilde{X}^T)_{jn}\delta_{mn} = \sum_{j=1}^{p}\sum_{n=1}^{N} (\tilde{X}^T)_{in}(\tilde{X})_{nj}\delta_{ij}$$

Covariance as trace suggests reduction of matrix size

- $\tilde{y}^n = y^n - \langle y \rangle, \tilde{x}_i^n = x_i^n - \langle x_i \rangle$ and drop the column of ones from design matrix to form $\tilde{X}$ with matrix elements $(\tilde{X})_{nj} = x_j^n - \langle x_j \rangle$.
- $(n \times n)$ and $(p \times p)$ matrices formed out of $\tilde{X}$:

$$(\tilde{X}\tilde{X}^T)_{mn} = \sum_{j=1}^{p} (\tilde{X})_{mj}(\tilde{X}^T)_{jn} = \sum_{j=1}^{p} (x_j^m - \langle x_j \rangle)(x_j^n - \langle x_j \rangle)$$

$$(\tilde{X}^T\tilde{X})_{ij} = \sum_{n=1}^{N} (\tilde{X}^T)_{in}(\tilde{X})_{nj} = \sum_{n=1}^{N} (x_i^n - \langle x_i \rangle)(x_j^n - \langle x_j \rangle)$$

- Trace of $(\tilde{X}\tilde{X}^T)$ equals trace$(\tilde{X}^T\tilde{X})$:

$$\sum_{n=1}^{N}\sum_{j=1}^{p} (\tilde{X})_{mj}(\tilde{X}^T)_{jn}\delta_{mn} = \sum_{j=1}^{p}\sum_{n=1}^{N} (\tilde{X}^T)_{in}(\tilde{X})_{nj}\delta_{ij}$$

- Work with p-dim matrix not N-dim.

Use of SVD in low rank approximation gives PCA

- Total variance of data $\boldsymbol{X}$ is the sum of eigenvalues of $\frac{1}{N}\text{tr}(\tilde{\boldsymbol{X}}^\top \tilde{\boldsymbol{X}})$

Use of SVD in low rank approximation gives PCA

- Total variance of data $\boldsymbol{X}$ is the sum of eigenvalues of $\frac{1}{N}\mathrm{tr}(\tilde{\boldsymbol{X}}^{\mathsf{T}}\tilde{\boldsymbol{X}})$
- If $\tilde{\boldsymbol{X}} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathsf{T}}$, $\boldsymbol{y} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathsf{T}}\boldsymbol{w}$ and $\widehat{\boldsymbol{w}} = \boldsymbol{V}(\boldsymbol{S})^{-1}\boldsymbol{U}^{\mathsf{T}}\boldsymbol{y}$.
  $\boldsymbol{S} \equiv \mathrm{diag}(\sigma_1, \ldots, \sigma_{\min(p,N)})$, $\boldsymbol{U}, \boldsymbol{V}$ contain singular vectors of size N and p respectively.

$$\widehat{\boldsymbol{w}} = \sum_{k=1}^{\min(p,N)} \frac{\boldsymbol{u}_k^{\mathsf{T}}\boldsymbol{y}}{\sigma_k}\boldsymbol{v}_k$$

Use of SVD in low rank approximation gives PCA

- Total variance of data $\boldsymbol{X}$ is the sum of eigenvalues of $\frac{1}{N}\mathrm{tr}(\tilde{\boldsymbol{X}}^T\tilde{\boldsymbol{X}})$
- If $\tilde{\boldsymbol{X}} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T$, $y = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T w$ and $\hat{w} = V(S)^{-1}U^T y$.
  $S \equiv \mathrm{diag}(\sigma_1, \ldots, \sigma_{\min(p,N)})$, U, V contain singular vectors of size N and p respectively.

$$\hat{w} = \sum_{k=1}^{\min(p,N)} \frac{u_k^T y}{\sigma_k} v_k$$

- Total variance is $\frac{1}{N}\mathrm{tr}(\tilde{\boldsymbol{X}}^T\tilde{\boldsymbol{X}}) = \frac{1}{N}\sum_{k=1}^{\min(p,N)} \sigma_k^2$

Use of SVD in low rank approximation gives PCA

- Total variance of data $\boldsymbol{X}$ is the sum of eigenvalues of $\frac{1}{N}\mathrm{tr}(\tilde{\boldsymbol{X}}^T\tilde{\boldsymbol{X}})$
- If $\tilde{\boldsymbol{X}} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T$, $\boldsymbol{y} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T\boldsymbol{w}$ and $\hat{\boldsymbol{w}} = \boldsymbol{V}(\boldsymbol{S})^{-1}\boldsymbol{U}^T\boldsymbol{y}$.
  $\boldsymbol{S} \equiv \mathrm{diag}(\sigma_1, \ldots, \sigma_{\min(p,N)})$, $\boldsymbol{U}, \boldsymbol{V}$ contain singular vectors of size N and p respectively.

$$\hat{\boldsymbol{w}} = \sum_{k=1}^{\min(p,N)} \frac{\boldsymbol{u}_k^T\boldsymbol{y}}{\sigma_k}\boldsymbol{v}_k$$

- Total variance is $\frac{1}{N}\mathrm{tr}(\tilde{\boldsymbol{X}}^T\tilde{\boldsymbol{X}}) = \frac{1}{N}\sum_{k=1}^{\min(p,N)} \sigma_k^2$
- **Discard small values** of σ_k, keep largest r singular vectors. Fraction of variation in data explained by r components is

$$\left(\sum_{k=1}^{r} \sigma_k^2\right) / \left(\sum_{k=1}^{\min(p,N)} \sigma_k^2\right)$$

Use of SVD in low rank approximation gives PCA

- Total variance of data X is the sum of eigenvalues of $\frac{1}{N}\mathrm{tr}(\tilde{X}^T\tilde{X})$
- If $\tilde{X} = USV^T$, $y = USV^Tw$ and $\hat{w} = V(S)^{-1}U^Ty$.
  $S \equiv \mathrm{diag}(\sigma_1, \ldots, \sigma_{\min(p,N)})$, U, V contain singular vectors of size N and p respectively.

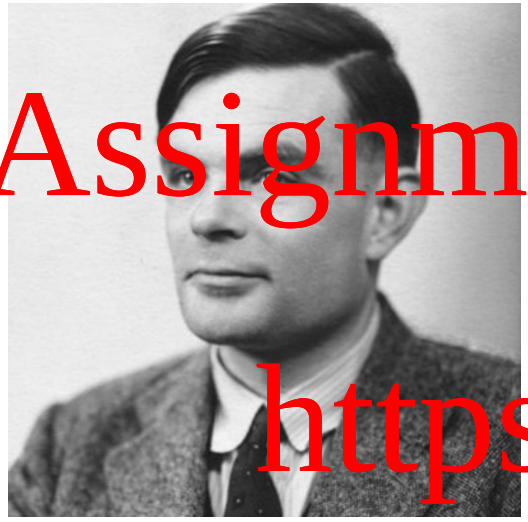
$$\hat{w} = \sum_{k=1}^{\min(p,N)} \frac{u_k^T y}{\sigma_k} v_k$$

- Total variance is $\frac{1}{N}\mathrm{tr}(\tilde{X}^T\tilde{X}) = \frac{1}{N}\sum_{k=1}^{\min(p,N)} \sigma_k^2$
- **Discard small values** of σ_k, keep largest r singular vectors. Fraction of variation in data explained by r components is

$$(\sum_{k=1}^{r} \sigma_k^2)/(\sum_{k=1}^{\min(p,N)} \sigma_k^2)$$

- $v_1, \ldots, v_r$ are the **principal components** (of variation).

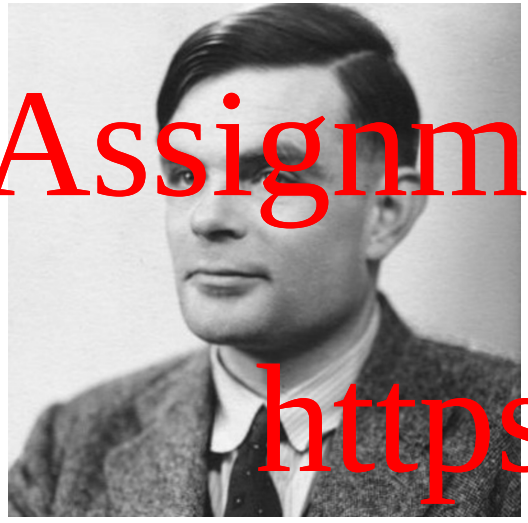
SVD is a low rank approximation to any matrix



- Single image as matrix M

Reconstruction for $r = 1, \ldots, 10$

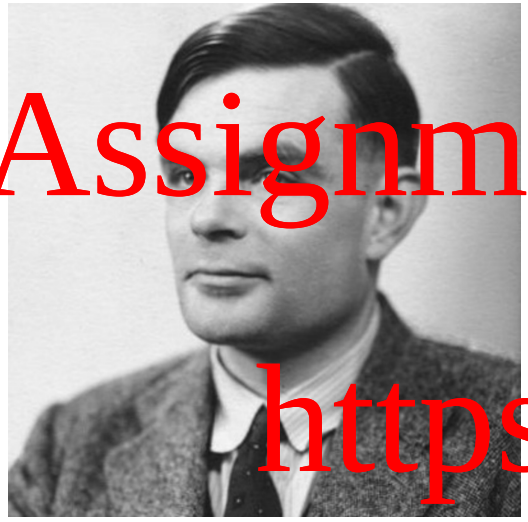
SVD is a low rank approximation to any matrix



- Single image as matrix M
- $M_{1200\times1200}$

Reconstruction for $r = 1, \ldots, 10$

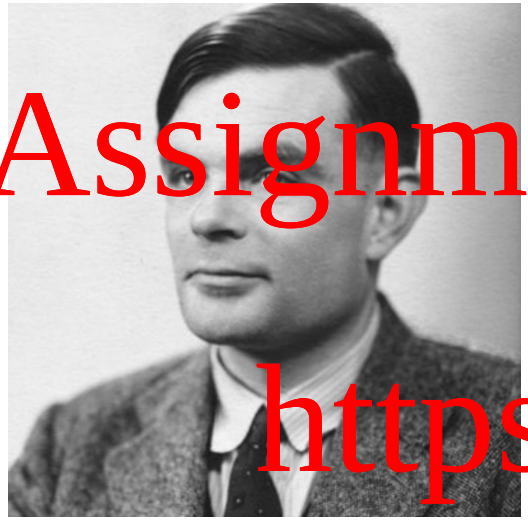
SVD is a low rank approximation to any matrix



- Single image as matrix M
- $M_{1200\times1200}$
- SVD $M = U\Sigma V^T$

Reconstruction for $r = 1, \ldots, 10$

SVD is a low rank approximation to any matrix



- Single image as matrix M
- $M_{1200\times1200}$
- SVD $M = U\Sigma V^T$
- Reconstruction: $\widetilde{M} = \sum_i^r \sigma_i u_i v_i^T$

Reconstruction for $r = 1, \ldots, 10$

First principal component captures greatest variation

- Data (mean subtracted) $x^n \in \mathbb{R}^p$ $i = n, \ldots,$ N arranged in data matrix $\boldsymbol{X}$.

First principal component captures greatest variation

- Data (mean subtracted) $x^n \in \mathbb{R}^p$ $i = n, \ldots, N$ arranged in data matrix X.
- Linear combinations of data vectors: $z^n = \sum_{j=1}^{p} w_j x_j^n$ written as $z = Xw$.

First principal component captures greatest variation

- Data (mean subtracted) $x^n \in \mathbb{R}^p$ $i = n, \ldots, N$ arranged in data matrix X.
- Linear combinations of data vectors: $z^n = \sum_{j=1}^{p} w_j x_j^n$ written as $z = Xw$.
- $\mathrm{var}(Xw) = \langle w^T X^T X w \rangle = w^T S w$ where $S = \mathrm{var}(X)$.

First principal component captures greatest variation

- Data (mean subtracted) $x^n \in \mathbb{R}^p$ $i = n, \ldots,$ N arranged in data matrix X.
- Linear combinations of data vectors $z^n = \sum_{j=1}^{p} w_j x_j^n$ written as $z = Xw$.
- $\mathrm{var}(Xw) = \langle w^T X^T X w \rangle = w^T S w$ where $S = \mathrm{var}(X)$.
- Seek components of linear combination $w =$ that maximise the variance $w^T Sw$ of the combined data points z^n, $i = 1, \ldots,$ N. Normalise w. Seek

$$\underset{w}{\mathrm{argmax}} \left\{ w^T S w - \lambda (w^T w - 1) \right\} \implies Sw = \lambda w.$$

First principal component captures greatest variation

- Data (mean subtracted) $x^n \in \mathbb{R}^p$ $i = n, \ldots,$ N arranged in data matrix X.
- Linear combinations of data vectors $z^n = \sum_{j=1}^{p} w_j x_j^n$ written as $z = Xw$.
- $\mathrm{var}(Xw) = \langle w^T X^T X w \rangle = w^T S w$ where $S = \mathrm{var}(X)$.
- Seek components of linear combination $w =$ that maximise the variance $w^T Sw$ of the combined data points z^n, $i = 1 \ldots,$ N. Normalise w. Seek

$$\underset{w}{\mathrm{argmax}} \left\{ w^T S w - \lambda (w^T w - 1) \right\} \implies Sw = \lambda w.$$

- Order eigenvalues $(\lambda_1, \ldots, \lambda_p)$ w_1 first principal component.

First principal component captures greatest variation

- Data (mean subtracted) $x^n \in \mathbb{R}^p$ $i = n, \ldots, N$ arranged in data matrix X.
- Linear combinations of data vectors $z^n = \sum_{j=1}^{p} w_j x_j^n$ written as Xw.
- $\mathrm{var}(Xw) = \langle w^T X^T X w \rangle = w^T S w$ where $S = \mathrm{var}(X)$.
- Seek components of linear combination $w =$ that maximise the variance $w^T S w$ of the combined data points z^n, $i = 1, \ldots, N$. Normalise w. Seek

$$\underset{w}{\mathrm{argmax}} \left\{ w^T S w - \lambda(w^T w - 1) \right\} \implies Sw = \lambda w.$$

- Order eigenvalues $(\lambda_1, \ldots, \lambda_p)$ w_1 first principal component.
- Linear combinations $z_1^n = w_{1,1} x_1^n + w_{1,2} x_2^n + \cdots + w_{1,p} x_p^n$ constitute representation of data in terms of first principal component. Instead of p components $(x_1^n, \ldots, x_p^n)$, **one** component z_1^n represents x^n.

First principal component captures greatest variation

- Data (mean subtracted) $x^n \in \mathbb{R}^p$ $i = n, \ldots, N$ arranged in data matrix X.
- Linear combinations of data vectors $z^n = \sum_{j=1}^{p} w_j x_j^n$ written as Xw.
- $\mathrm{var}(Xw) = \langle w^T X^T X w \rangle = w^T S w$ where $S = \mathrm{var}(X)$.
- Seek components of linear combination $w =$ that maximise the variance $w^T S w$ of the combined data points z^n, $i = 1, \ldots, N$. Normalise w. Seek

$$\underset{w}{\mathrm{argmax}} \left\{ w^T S w - \lambda(w^T w - 1) \right\} \implies Sw = \lambda w.$$

- Order eigenvalues $(\lambda_1, \ldots, \lambda_p)$ w_1 first principal component.
- Linear combinations $z_1^n = w_{1,1} x_1^n + w_{1,2} x_2^n + \cdots + w_{1,p} x_p^n$ constitute representation of data in terms of first principal component. Instead of p components $(x_1^n, \ldots, x_p^n)$, **one** component z_1^n represents x^n.
- Variance of $\{z_1^n\}$ is λ_1.

PCA: principal components from variance decomposition

- Other eigenvectors w_k (from $Sw_k = \lambda_k w_k$) represent combinations that create $z_k^n = w_{k,1}x_1^n + w_{k,2}x_2^n + \cdots + w_{k,p}x_p^n$ decorrelated from z_1. $\text{cov}(z_i^n, z_j^n) = 0$ for $i \neq j$.

PCA: principal components from variance decomposition

- Other eigenvectors w_k (from $Sw_k = \lambda_k w_k$) represent combinations that create $z_k^n = w_{k,1}x_1^n + w_{k,2}x_2^n + \cdots + w_{k,p}x_p^n$ decorrelated from z_1. $\text{cov}(z_i^n, z_j^n) = 0$ for $i \neq j$.
- Largest r eigenvalues give new "co-ordinates" $(z_1^n, \ldots, z_r^n)$ of the data and account for fraction of variation

$$\left(\sum_{k=1}^{r} \lambda_k\right) / \left(\sum_{k=1}^{p} \lambda_k\right)$$

PCA: principal components from variance decomposition

- Other eigenvectors w_k (from $Sw_k = \lambda_k w_k$) represent combinations that create $z_k^n = w_{k,1}x_1^n + w_{k,2}x_2^n + \cdots + w_{k,p}x_p^n$ decorrelated from z_1. $\text{cov}(z_i^n, z_j^n) = 0$ for $i \neq j$.

- Largest r eigenvalues give new 'co-ordinates' $(z_1^n, \ldots, z_r^n)$ of the data and account for fraction of variation

$$\left(\sum_{k=1}^{r} \lambda_k\right) / \left(\sum_{k=1}^{p} \lambda_k\right)$$

- $w_1, \ldots, w_r$ are the **principal components** (of variation).

Scale independence

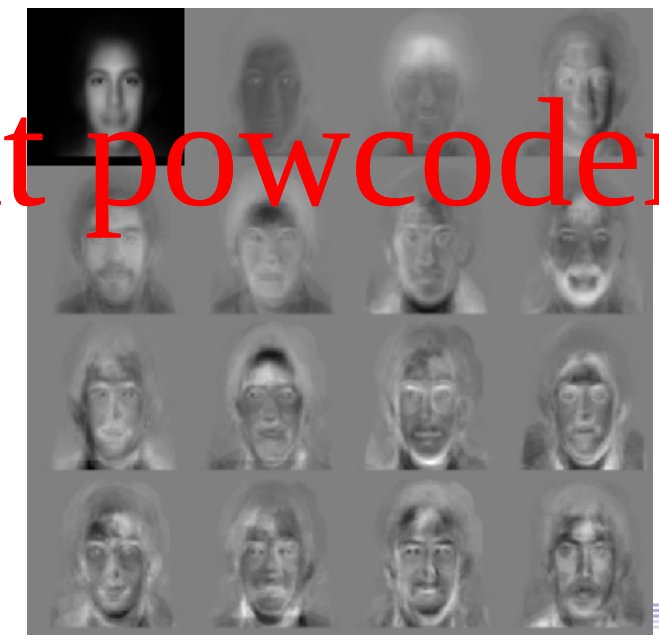
To make PCA independent of the *scale* of the data x^n, they might have to be first standardised such that all data entries x_k^n have equal range, eg by transforming them to unit variance.

Eigenfaces as features

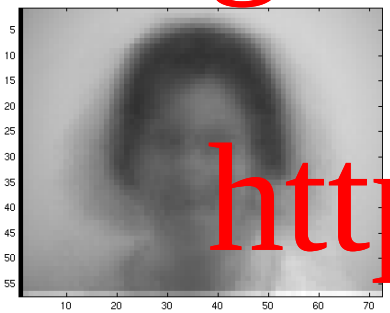
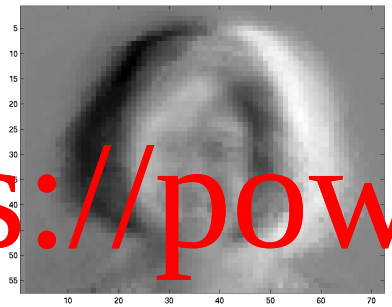
We will express arbitrary data vectors as linear combinations $v^n = \sum_{i=1}^{r} \alpha_i^n w_i$. of a set of eigenvectors $\{w_i\}$.

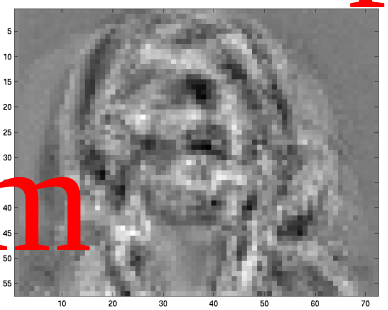
These are obtained from a certain data dependent matrix. Often most of the w_i are small (except for a small set of 'important' eigenvectors). In such a case we can represent a high dimensional data vector v^n effectively by a much smaller (and more robust) set of new 'coordinates' α_j^n

For a face recognition problem the vector v^n and the dominant **eigenfaces** may look like this:



Mean and eigenfaces



Mean face eigenface 1 eigenface 2 eigenface 30

Higher eigenfaces show only some random structure.

PCA as low-rank regression: input as target output

- In regression $y = Xw$

PCA as low-rank regression: input as target output

- In regression $y = Xw$
- Take output y is matrix of x^n, $n = 1, \ldots, N$. Instead of vector $\widehat{w}$ we have matrix $\widehat{W}$ (one column for each data point):

$$\widehat{W} = (X^T X)^{-1} X^T X = \mathbb{I}$$

and the "predicted" output is $X\widehat{W} = X$.

PCA as low-rank regression: input as target output

- In regression $y = Xw$
- Take output y is matrix of x^n, $n = 1, \ldots, N$. Instead of vector $\hat{w}$ we have matrix $\widehat{W}$ (one column for each data point):

$$\widehat{W} = (X^T X)^{-1} X^T X = \mathbb{I}$$

and the "predicted" output is $X\widehat{W} = X$.

- Choose a set of q orthonormal vectors $\{v_i\}$ whose linear combinations $z^n = \sum_{i=1}^{q} \alpha^n v_i$ minimise $\|x^n - z^n\|^2$.

PCA as low-rank regression: input as target output

- In regression $y = Xw$
- Take output y is matrix of x^n, $n = 1, \ldots, N$. Instead of vector $\hat{w}$ we have matrix $\widehat{W}$ (one column for each data point):

$$\widehat{W} = (X^T X)^{-1} X^T X = \mathbb{I}$$

and the "predicted" output is $X\widehat{W} = X$.

- Choose a set of q orthonormal vectors $\{v_i\}$ whose linear combinations $z^n = \sum_{i=1}^{q} \alpha_i^n v_i$ minimise $\|x^n - z^n\|^2$.

- If $q < p$, dimensional reduction. Instead of basis vectors $e_k = (0, \cdots, 0, \underbrace{1}_{k}, 0, \ldots, 0)^T$, $k = 1, \ldots, p$, use v_i, $i = 1, \ldots, q$. The co-ordinates are α_i^n

Exercise

- Minimising with respect to $\begin{pmatrix} v_1 \\ v_2 \end{pmatrix}_{(1)}$ and $\begin{pmatrix} v_1 \\ v_2 \end{pmatrix}_{(2)}$,

$$\left\| \begin{pmatrix} x_1^{(n)} \\ x_2^{(n)} \end{pmatrix} - \begin{pmatrix} v_{1;1} & v_{2;1} \\ v_{1;2} & v_{2;2} \end{pmatrix} \begin{pmatrix} \alpha_1^{(n)} \\ \alpha_2^{(n)} \end{pmatrix} \right\|^2$$

(in $v_{j;i}$, j the label for the vector, i the row index),

$$\begin{pmatrix} \alpha_1^{(n)} \\ \alpha_2^{(n)} \end{pmatrix} = \begin{pmatrix} v_{1;1} & v_{2;1} \\ v_{1;2} & v_{2;2} \end{pmatrix}^{-1} \begin{pmatrix} x_1^{(n)} \\ x_2^{(n)} \end{pmatrix}.$$

Exercise

- Minimising with respect to $\begin{pmatrix} v_1 \\ v_2 \end{pmatrix}_{(1)}$ and $\begin{pmatrix} v_1 \\ v_2 \end{pmatrix}_{(2)}$,

$$\left\| \begin{pmatrix} x_1^{(n)} \\ x_2^{(n)} \end{pmatrix} - \begin{pmatrix} v_{1;1} & v_{2;1} \\ v_{1;2} & v_{2;2} \end{pmatrix} \begin{pmatrix} \alpha_1^{(n)} \\ \alpha_2^{(n)} \end{pmatrix} \right\|^2$$

(in $v_{j;i}$, j the label for the vector, i the row index),

$$\begin{pmatrix} \alpha_1^{(n)} \\ \alpha_2^{(n)} \end{pmatrix} = \begin{pmatrix} v_{1;1} & v_{2;1} \\ v_{1;2} & v_{2;2} \end{pmatrix}^T \begin{pmatrix} x_1^{(n)} \\ x_2^{(n)} \end{pmatrix}.$$

- Extend to q-dimensional representation of $x^n \in \mathbb{R}^p$,

$$\begin{pmatrix} \alpha_1^{(n)} \\ \vdots \\ \alpha_q^{(n)} \end{pmatrix} = \underbrace{\begin{pmatrix} v_{1;1} & v_{2;1} & \cdots & v_{q;1} \\ \vdots & \vdots & \ddots & \vdots \\ v_{1;p} & v_{2;p} & \cdots & v_{q;p} \end{pmatrix}^T}_{\text{columns of matrix } V_q \text{ are orthogonal vectors}} \begin{pmatrix} x_1^{(n)} \\ \vdots \\ x_p^{(n)} \end{pmatrix}.$$

PCA: Low rank projection

- q-dimensional representation of $x^n \in \mathbb{R}^p$, with orthonormal vectors as columns of V_q:

$$\begin{pmatrix} \alpha_1^{(n)} \\ \vdots \\ \alpha_q^{(n)} \end{pmatrix} = V_q^T \begin{pmatrix} x_1^{(n)} \\ \vdots \\ x_p^{(n)} \end{pmatrix}.$$

PCA: Low rank projection

- q-dimensional representation of $x^n \in \mathbb{R}^p$, with orthonormal vectors as columns of V_q:

$$\begin{pmatrix} \alpha_1^{(n)} \\ \vdots \\ \alpha_q^{(n)} \end{pmatrix} = V_q^T \begin{pmatrix} x_1^{(n)} \\ \vdots \\ x_p^{(n)} \end{pmatrix}.$$

- Since $(V_q V_q^T)(V_q V_q^T) = (V_q V_q^T)$, $P_q \triangleq V_q V_q^T$ is a projection onto q-dim subspace spanned by columns of V_q.

PCA: Low rank projection

- q-dimensional representation of $x^n \in \mathbb{R}^p$, with orthonormal vectors as columns of V_q:

$$\begin{pmatrix} \alpha_1^{(n)} \\ \vdots \\ \alpha_q^{(n)} \end{pmatrix} = V_q^T \begin{pmatrix} x_1^{(n)} \\ \vdots \\ x_p^{(n)} \end{pmatrix}.$$

- Since $(V_q V_q^T)(V_q V_q^T) = (V_q V_q^T)$, $P_q \triangleq V_q V_q^T$ is a projection onto q-dim subspace spanned by columns of V_q.
- $Y = (X)_q = V_q^T X$ is the matrix of N q-dim vectors $z^n \in \mathbb{R}^q$

PCA: Low rank projection

- q-dimensional representation of $x^n \in \mathbb{R}^p$, with orthonormal vectors as columns of V_q:

$$\begin{pmatrix} \alpha_1^{(n)} \\ \vdots \\ \alpha_q^{(n)} \end{pmatrix} = V_q^T \begin{pmatrix} x_1^{(n)} \\ \vdots \\ x_p^{(n)} \end{pmatrix}.$$

- Since $(V_q V_q^T)(V_q V_q^T) = (V_q V_q^T)$, $P_q \triangleq V_q V_q^T$ is a projection onto q-dim subspace spanned by columns of V_q.
- $Y = (X)_q = V_q^T X$ is the matrix of N q-dim vectors $z^n \in \mathbb{R}^q$

- Among all rank q matrices A, quantity $\|X - A\|^2$ is minimised is $A = V_q V_q^T X$, q leading vectors (in order of singular values σ_i in $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_{\min p,N}$ in V in $X = U\Sigma V^T$

PCA: Low rank projection

- q-dimensional representation of $x^n \in \mathbb{R}^p$, with orthonormal vectors as columns of V_q:

$$\begin{pmatrix} \alpha_1^{(n)} \\ \vdots \\ \alpha_q^{(n)} \end{pmatrix} = V_q^T \begin{pmatrix} x_1^{(n)} \\ \vdots \\ x_p^{(n)} \end{pmatrix}.$$

- Since $(V_q V_q^T)(V_q V_q^T) = (V_q V_q^T)$, $P_q \triangleq V_q V_q^T$ is a projection onto q-dim subspace spanned by columns of V_q.

- $Y = (X)_q = V_q^T X$ is the matrix of N q-dim vectors $z^n \in \mathbb{R}^q$
- Among all rank q matrices A, quantity $\|X - A\|^2$ is minimised is $A = V_q V_q^T X$, q leading vectors (in order of singular values σ_i in $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_{\min p,N}$ in V in $X = U\Sigma V^T$
- $\|X - V_q V_q^T X\|^2$ is the minimum residual.

Compare PCA pre-processed regression with regularisation

- Regularisation using the 2-norm of $\boldsymbol{w}$ involves minimising the loss function $\ell(\boldsymbol{w})$

$$
\begin{aligned}
\widehat{\boldsymbol{w}}_{\text{ridge}} &= \operatorname{argmin}_{\boldsymbol{w}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|^2 + \lambda\|\boldsymbol{w}\|^2 \\
\implies \widehat{\boldsymbol{w}}_{\text{ridge}} &= (\boldsymbol{X}^{\mathsf{T}}\boldsymbol{X} + \lambda\mathbb{I})^{-1}\boldsymbol{X}^{\mathsf{T}}\boldsymbol{y} \\
&= \boldsymbol{V}(\boldsymbol{S}^2 + \lambda\mathbb{I})^{-1}\boldsymbol{S}\boldsymbol{U}^{\mathsf{T}}\boldsymbol{y} \\
&= \sum_{k=1}^{\min(p,N)} (\boldsymbol{u}_k^{\mathsf{T}}\boldsymbol{y})\,\frac{\sigma_k}{\sigma_k^2 + \lambda}\boldsymbol{v}_k
\end{aligned}
$$

Compare PCA pre-processed regression with regularisation

- Regularisation using the 2-norm of $\boldsymbol{w}$ involves minimising the loss function $\ell(\boldsymbol{w})$

$$\begin{aligned} \widehat{\boldsymbol{w}}_{\text{ridge}} &= \text{argmin}_{\boldsymbol{w}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|^2 + \lambda\|\boldsymbol{w}\|^2 \\ \implies \widehat{\boldsymbol{w}}_{\text{ridge}} &= (\boldsymbol{X}^\mathsf{T}\boldsymbol{X} + \lambda\mathbb{I})^{-1}\boldsymbol{X}^\mathsf{T}\boldsymbol{y} \\ &= \boldsymbol{V}(\boldsymbol{S}^2 + \lambda\mathbb{I})^{-1}\boldsymbol{S}\boldsymbol{U}^\mathsf{T}\boldsymbol{y} \\ &= \sum_{k=1}^{\min(p,N)} (\boldsymbol{u}_k^\mathsf{T}\boldsymbol{y}) \frac{\sigma_k}{\sigma_k^2 + \lambda}\boldsymbol{v}_k \end{aligned}$$

- PCA drops all terms with small variance contributions, ridge regression performs a "soft" re-weighting.

Compare PCA pre-processed regression with regularisation

- Regularisation using the 2-norm of $\boldsymbol{w}$ involves minimising the loss function $\ell(\boldsymbol{w})$

$$\begin{aligned} \widehat{\boldsymbol{w}}_{\text{ridge}} &= \text{argmin}_{\boldsymbol{w}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|^2 + \lambda\|\boldsymbol{w}\|^2 \\ \implies \widehat{\boldsymbol{w}}_{\text{ridge}} &= (\boldsymbol{X}^\mathsf{T}\boldsymbol{X} + \lambda\mathbb{I})^{-1}\boldsymbol{X}^\mathsf{T}\boldsymbol{y} \\ &= \boldsymbol{V}(\boldsymbol{S}^2 + \lambda\mathbb{I})^{-1}\boldsymbol{S}\boldsymbol{U}^\mathsf{T}\boldsymbol{y} \\ &= \sum_{k=1}^{\min(p,N)} (\boldsymbol{u}_k^\mathsf{T}\boldsymbol{y}) \frac{\sigma_k}{\sigma_k^2 + \lambda}\boldsymbol{v}_k \end{aligned}$$

- PCA drops all terms with small variance contributions, ridge regression performs a "soft" re-weighting.
- **Lasso**: regularisation method that drops components $\widehat{\boldsymbol{w}}_{\text{lasso}} = \text{argmin}_{\boldsymbol{w}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|^2$ s.t. $\lambda \sum_i |w_i| < t$.

Compare PCA pre-processed regression with regularisation

- Regularisation using the 2-norm of $\boldsymbol{w}$ involves minimising the loss function $\ell(\boldsymbol{w})$

$$\begin{aligned} \widehat{\boldsymbol{w}}_{\text{ridge}} &= \text{argmin}_{\boldsymbol{w}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|^2 + \lambda\|\boldsymbol{w}\|^2 \\ \implies \widehat{\boldsymbol{w}}_{\text{ridge}} &= (\boldsymbol{X}^\mathsf{T}\boldsymbol{X} + \lambda\mathbb{I})^{-1}\boldsymbol{X}^\mathsf{T}\boldsymbol{y} \\ &= \boldsymbol{V}(\boldsymbol{S}^2 + \lambda\mathbb{I})^{-1}\boldsymbol{S}\boldsymbol{U}^\mathsf{T}\boldsymbol{y} \\ &= \sum_{k=1}^{\min(p,N)} (\boldsymbol{u}_k^\mathsf{T}\boldsymbol{y}) \frac{\sigma_k}{\sigma_k^2 + \lambda}\boldsymbol{v}_k \end{aligned}$$

- PCA drops all terms with small variance contributions, ridge regression performs a "soft" re-weighting.
- **Lasso**: regularisation method that drops components $\widehat{\boldsymbol{w}}_{\text{lasso}} = \text{argmin}_{\boldsymbol{w}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|^2$ s.t. $\lambda \sum_i |w_i| < t$.

Compare PCA pre-processed regression with regularisation

- Regularisation using the 2-norm of $\boldsymbol{w}$ involves minimising the loss function $\ell(\boldsymbol{w})$

$$\begin{aligned} \widehat{\boldsymbol{w}}_{\text{ridge}} &= \operatorname{argmin}_{\boldsymbol{w}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|^2 + \lambda\|\boldsymbol{w}\|^2 \\ \implies \widehat{\boldsymbol{w}}_{\text{ridge}} &= (\boldsymbol{X}^{\mathsf{T}}\boldsymbol{X} + \lambda\mathbb{I})^{-1}\boldsymbol{X}^{\mathsf{T}}\boldsymbol{y} \\ &= \boldsymbol{V}(\boldsymbol{S}^2 + \lambda\mathbb{I})^{-1}\boldsymbol{S}\boldsymbol{U}^{\mathsf{T}}\boldsymbol{y} \\ &= \sum_{k=1}^{\min(p,N)} (\boldsymbol{u}_k^{\mathsf{T}}\boldsymbol{y})\,\frac{\sigma_k}{\sigma_k^2 + \lambda}\boldsymbol{v}_k \end{aligned}$$

- PCA drops all terms with small variance contributions, ridge regression performs a "soft" re-weighting.
- **Lasso**: regularisation method that drops components
  $\widehat{\boldsymbol{w}}_{\text{lasso}} = \operatorname{argmin}_{\boldsymbol{w}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|^2$ s.t. $\lambda \sum_i |w_i| < t$.
  ($\|\boldsymbol{w}\|_1 \triangleq \sum_i |w_i|$, 1-norm.)

Refine representation: rotation in q-dim PC space for interpretable/sparse combinations; introducing non-linearity

- Using ideas from

Refine representation: rotation in q-dim PC space for interpretable/sparse combinations; introducing non-linearity

- Using ideas from
- **Lasso**: $\hat{w}_{lasso} = \arg\min_w \|y - Xw\|^2$ s.t. $\lambda \sum_i |w_i| < t$.

Refine representation: rotation in q-dim PC space for interpretable/sparse combinations; introducing non-linearity

- Using ideas from
  - **Lasso**: $\hat{w}_{\text{lasso}} \in \text{argmin}_{w} \|y - Xw\|^2$ s.t. $\lambda \sum_i |w_i| < t$.
  - **Elastic net**: $\text{argmin}_{w} \|y - Xw\|^2 + \lambda_2 \|w\|^2 + \lambda_1 \|w\|_1$.

Refine representation: rotation in q-dim PC space for interpretable/sparse combinations; introducing non-linearity

- Using ideas from
  - **Lasso**: $\widehat{w}_{\text{lasso}} \equiv \text{argmin}_w \|y - Xw\|^2$ s.t. $\lambda \sum_i |w_i| < 1$.
  - **Elastic net**: $\text{argmin}_w \|y - Xw\|^2 + \lambda_2 \|w\|^2 + \lambda_1 \|w\|_1$.
- **Sparse PCA**: $A_{p\times q} := (\alpha_1, \ldots, \alpha_q)$, $B_{p\times q} := (\beta_1, \ldots, \beta_q)$, $A^T A = \mathbb{I}_q$, so that $AA^T = P_q$ a q-dim projector:

$$\begin{aligned} \widehat{\beta} &= \text{argmin}_{A,B} \sum_{i=1}^{N} \|x_i - AB^T x_i\|^2 + \lambda_2 \|\beta\|^2 + \lambda_1 \|\beta\|_1 \\ v_i &= \widehat{\beta}_i / \|\widehat{\beta}\| \end{aligned}$$

v_i sparse PCA components

Refine representation: rotation in q-dim PC space for interpretable/sparse combinations; introducing non-linearity

- Using ideas from
  - **Lasso**: $\widehat{w}_{\text{lasso}} = \text{argmin}_w \|y - Xw\|^2$ s.t. $\lambda \sum_i |w_i| \leq 1$.
  - **Elastic net**: $\text{argmin}_w \|y - Xw\|^2 + \lambda_2 \|w\|^2 + \lambda_1 \|w\|_1$.
- **Sparse PCA**: $A_{p\times q} := (\alpha_1, \ldots, \alpha_q)$, $B_{p\times q} := (\beta_1, \ldots, \beta_q)$, $A^T A = \mathbb{I}_q$, so that $AA^T = P_q$ a q-dim projector:

$$\begin{aligned} \widehat{\beta} &= \text{argmin}_{A,B} \sum_{i=1}^{N} \|x_i - AB^T x_i\|^2 + \lambda_2 \|\beta\|^2 + \lambda_1 \|\beta\|_1 \\ v_i &= \widehat{\beta}_i / \|\widehat{\beta}\| \end{aligned}$$

v_i sparse PCA components

- Implement PCA via neural network with linear activation function. If non-linear, we arrive at an **autoencoder**, trained via backprop.