COMP3223 Foundations of ML: Dimensionality reduction

by

Srinandan Dasmah

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 min

the e

$$\sum_{i=1}^{p} w_i x_i^n$$

$$(r^n)^2 = (y^n - f(x^n; w))^2$$

$$\sum_{n=1}$$

$$r^n = y^n - (w_0 + w_1$$

Are more complex models better?

- With large number of features p, accidental correlations can obscure genuine patterns.
- In min
  the e

$$(r^n)^2 = (y^n - f(x^n; w))^2$$

$$= (y^n - (w_0 + w_1 \sum_{i=1}^{p} w_i x_i^n))^2$$

- In trying to minimise the residuals, you may be adjusti 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 min
  the e $\sum_{i=1}^{p} w_i x_i^n$

$$(r^n)^2 = (y^n - f(x^n; w)^2$$

$$\sum_{n=1}^{N} r^n = \sum_{n=1}^{N} (y^n - (w_0 + w_1 x^n)^2$$

- In trying to minimise the residuals, you may be adjusti 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:

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: $\frac{1}{N}$ $w_p x_p^n$)
- **Centering**: $\widehat{y}^n = y^n - \langle y \rangle$, $\widetilde{x}_i^n = x_i^n - \langle x_i \rangle$ $\sum_{i=1}$ from design matrix to form X with matrix $\langle x_j \rangle$.

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

N

$w_p x_p^n)$

$i \quad i$

$i=1$

- **Centering**: $\widetilde{y}^n = y^n - \langle y \rangle$, $\widetilde{x}_i^n = x_i^n$ … es from design matrix to form $\tilde{\boldsymbol{X}}$ with matrix … $\langle x_j \rangle$.

- $\frac{1}{N}\tilde{\boldsymbol{X}}^\top\left(\widetilde{\boldsymbol{y}} - \tilde{\boldsymbol{X}}\widehat{\boldsymbol{w}}\right) = 0$ implies $\widehat{\boldsymbol{w}} = \left(\frac{1}{N}\left(\tilde{\boldsymbol{X}}^\top\tilde{\boldsymbol{X}}\right)\right)^{-1}\left(\frac{1}{N}\tilde{\boldsymbol{X}}^\top\boldsymbol{y}\right)$, so

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

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

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

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

Covariance as trace suggests reduction of matrix size

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

$$\begin{pmatrix} & & x_j^n - \langle x_j \rangle \end{pmatrix}$$

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

- 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 $\tilde{X}$.

$$(\tilde{X}^T \tilde{X})_{ij} = \sum_{n=1}^{n} (\tilde{X}^T)_{in} (\tilde{X})_{n} \quad \begin{pmatrix} x_j^n - \langle x_j \rangle \\ \end{pmatrix}$$

- 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 X is the sum of eigenvalues of $\frac{1}{N}\text{tr}(\tilde{X}^T\tilde{X})$

Use of SVD in low rank approximation gives PCA

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

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 $\tilde{X} = USV^T$, $y = USV^T w$ 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.

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

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 $\tilde{X} = USV^T$, $y = USV^T w$ 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.

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

- **Discard small values** of σ_k, keep largest variation in data explained by r compone

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

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 $\tilde{X} = USV^T$, $y = USV^T w$ 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.

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

- **Discard small values** of σ_k, keep largest variation in data explained by r compone

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

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

SVD is a low rank approximation to any matrix

- Single image as matrix M

Reconstruction for r

SVD is a low rank approximation to any matrix

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

Reconstruction for r

SVD is a low rank approximation to any matrix

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

Reconstruction for r

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

First principal component captures greatest variation

- Data (mean subtracted) $x^n \in \mathbb{R}^p$ $i = n, \ldots, N$ arranged in data matrix 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 Xw.
- $\mathrm{var}(Xw) = w^T X^T X w = w^T S w$ 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 Xw.
- $\text{var}(Xw) = w^T X^T X w = w^T S w$ where $S = \text{var}(X)$.
- Seek w^T iance is w. Seek

$$\underset{w}{\text{argmax}} \; w^T S w - \lambda(w^T w$$

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 Xw.
- $\mathrm{var}(Xw) = w^T X^T X w = w^T S w$ where $S = \mathrm{var}(X)$.
- Seek w^T iance ise w. Seek

$$\underset{w}{\text{argmax}} \quad w^T S w - \lambda(w^T w$$

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

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.
- $\text{var}(Xw) = w^T X^T X w = w^T S w$ where $S = \text{var}(X)$.
- Seek w^T iance is w. Seek

$$\underset{w}{\text{argmax}} \; w^T S w - \lambda(w^T w$$

- Order eigenvalues $(\lambda_1, \ldots, \lambda_p)$ w_1 first
- Linear combinations $z_1^n = w_{1,1} x_1^n + w_{1,2} x_2 + \cdots + w_{1,p} x_p$ titute 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.
- $\mathrm{var}(Xw) = w^T X^T X w = w^T S w$ where $S = \mathrm{var}(X)$.
- Seek iance
  w^T ise w. Seek

$$\underset{w}{\text{argmax}} \quad w^T S w - \lambda(w^T w$$

- Order eigenvalues $(\lambda_1, \ldots, \lambda_p)$ w_1 first
- Linear combinations $z_1^n = w_{1,1} x_1^n + w_{1,2} x_2 + \cdots + w_{1,p} x_p$ titute 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 $z^n = w_{k,1}x^n + w_{k,2}x^n + \quad + w_{k,p}x^n$ decorrelated from z_1.

cov(/

PCA: principal components from variance decomposition

- Other eigenvectors w_k (from $Sw_k = \lambda_k w_k$) represent combinations that create $z^n = w_{k,1}x^n + w_{k,2}x^n + \quad + w_{k,p}x^n$ decorrelated from z_1.
  cov(/
- Larg e data and
  acco

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

PCA: principal components from variance decomposition

- Other eigenvectors w_k (from $Sw_k = \lambda_k w_k$) represent combinations that create $z^n = w_{k,1}x^n + w_{k,2}x^n + \quad + w_{k,p}x^n$ decorrelated from z_1.
  cov(/
- Larg e data and
  acco

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

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

Scale independence

To make P e first standardi ing 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 (e

represent and more robu

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 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 W (one column for each data point):

and t

$$X\widehat{W} = X$$

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 W (one column for each data point):

and t $$X\widehat{W} = X$$

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

PCA as low-rank regression: input as target output

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

and t $$\boldsymbol{X}\widehat{\boldsymbol{W}} = \boldsymbol{X}$$

- Choose a set of q orthonormal vectors $\{$ s $\boldsymbol{z}^n = \sum_{i=1}^{q} \alpha_i^n \boldsymbol{v}_i$ minimise $\|\boldsymbol{x}^n - \boldsymbol{z}^n$

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

Exercise

- Minimising with respect to $\begin{pmatrix} \nu_1 \\ \nu_2 \end{pmatrix}_{(1)}$ and $\begin{pmatrix} \nu_1 \\ \nu_2 \end{pmatrix}_{(2)}$,

$$\left\| \begin{pmatrix} x_1^{(n)} \\ x^{(n)} \end{pmatrix} - \begin{matrix} \nu_{1;1} & \nu_{2;1} \\ \nu_{1;2} & \nu_{2;2} \end{matrix} \begin{pmatrix} \alpha_1^{(n)} \\ \alpha^{(n)} \end{pmatrix} \right\|^2$$

(in ν

$$\alpha_2^{(n)} = \nu_{1;2} \quad \nu .$$

Exercise

- Minimising with respect to $\begin{pmatrix} \nu_1 \\ \nu_2 \end{pmatrix}_{(1)}$ and $\begin{pmatrix} \nu_1 \\ \nu_2 \end{pmatrix}_{(2)}$,

$$\left\| \begin{pmatrix} x_1^{(n)} \\ x^{(n)} \end{pmatrix} - \begin{matrix} \nu_{1;1} & \nu_{2;1} \\ \nu_{1;2} & \nu_{2;2} \end{matrix} \begin{pmatrix} \alpha_1^{(n)} \\ \alpha^{(n)} \end{pmatrix} \right\|^2$$

(in ν

$$\alpha_2^{(n)} = \quad \nu_{1;2} \quad \nu \qquad .$$

- Extend to q-dimensional representation of

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

PCA: Low rank projection

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

$$\alpha_1^{(n)} \qquad x_1^{(n)}$$

PCA: Low rank projection

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

$$\alpha_1^{(n)} \qquad x_1^{(n)}$$

- Sinc $(V_q V_q^T)(V_q V_q^T) = (V_q V_q^T)$ P$_q$ $V_q V_q^T$ n onto q-dim subspace spanned by columns of V

PCA: Low rank projection

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

$$\alpha_1^{(n)} \qquad x_1^{(n)}$$

- Sinc $(V_q V_q^T)(V_q V_q^T) = (V_q V_q^T)$ P$_q$ $V_q V_q^T$ n onto q-dim subspace spanned by columns of V
- $Y = (X)_q = V_q^T X$ is the matrix of N

PCA: Low rank projection

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

$$\alpha_1^{(n)} \qquad x_1^{(n)}$$

- Sinc $(V_q V_q^T)(V_q V_q^T) = (V_q V_q^T)$ P$_q$ $V_q V_q^T$ n onto q-dim subspace spanned by columns of V
- $Y = (X)_q = V_q^T X$ is the matrix of N

- Among all rank q matrices A, quantity $\|$ $A = V_q V_q^T X$, 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:

$$\alpha_1^{(n)} \qquad x_1^{(n)}$$

- Sinc $(V_q V_q^T)(V_q V_q^T) = (V_q V_q^T)$ P$_q$ $V_q V_q^T$ n onto q-dim subspace spanned by columns of V
- $Y = (X)_q = V_q^T X$ is the matrix of N

- Among all rank q matrices A, quantity $\|$ $A = V_q V_q^T X$, 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 $\ell(\boldsymbol{w})$

$$\boldsymbol{w}_{\text{ridge}} = \text{argmin}_{\boldsymbol{w}} \ \boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\ ^2 + \lambda\ \boldsymbol{w}\ ^2$$

$$= \sum_{k=1} \boldsymbol{u}_k \boldsymbol{y} \ \frac{}{2} \boldsymbol{v}_k$$

Compare PCA pre-processed regression with regularisation

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

$$\boldsymbol{w}_{\text{ridge}} = \text{argmin}_{\boldsymbol{w}} \ \boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\ ^2 + \lambda\ \boldsymbol{w}\ ^2$$

$$= \sum_{k=1} \boldsymbol{u}_k \boldsymbol{y} \ \frac{}{2} \boldsymbol{v}_k$$

- PCA drops all terms with small variance contributi 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 $\ell(\boldsymbol{w})$

$$\boldsymbol{w}_{\text{ridge}} = \text{argmin}_{\boldsymbol{w}} \ \boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\ ^2 + \lambda\ \boldsymbol{w}\ ^2$$

$$= \sum_{k=1} \boldsymbol{u}_k \boldsymbol{y} \ \frac{}{2} \boldsymbol{v}_k$$

- PCA drops all terms with small variance contributi 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. $\lambda \sum_i |w_i| < \text{t}$.

Compare PCA pre-processed regression with regularisation

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

$$\boldsymbol{w}_{\text{ridge}} = \text{argmin}_{\boldsymbol{w}} \ \boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\ ^2 + \lambda\ \boldsymbol{w}\ ^2$$

$$= \sum_{k=1} \boldsymbol{u}_k \boldsymbol{y} \ \frac{}{2} \boldsymbol{v}_k$$

- PCA drops all terms with small variance contributi 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. $\lambda \sum_i |w_i| < \text{t}$.

Compare PCA pre-processed regression with regularisation

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

$$\boldsymbol{w}_{\text{ridge}} = \text{argmin}_{\boldsymbol{w}} \ \boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\ ^2 + \lambda\ \boldsymbol{w}\ ^2$$

$$= \sum_{k=1} \boldsymbol{u}_k \boldsymbol{y} \ \frac{}{2} \boldsymbol{v}_k$$

- PCA drops all terms with small variance contributi 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. $\lambda \sum_i |w_i| < \text{t}$. ($\|\boldsymbol{w}\|_1 \triangleq \sum_i |w_i|$, 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**: $w_{lasso} = \text{argmin}_w \|y - Xw\|^2$ 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**: $\boldsymbol{w}_{\text{lasso}} \in \text{argmin}_{\boldsymbol{w}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|^2$ s.t. $\lambda \sum_i |w_i| \leq 1$.
  - **Elastic net**: $\text{argmin}_{\boldsymbol{w}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|^2 + \lambda_2 \|\boldsymbol{w}\|^2 + \lambda_1 \|\boldsymbol{w}\|_1$.

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

- Using ideas from
  - **Lasso**: $w_{lasso} \equiv \text{argmin}_w \|y - Xw\|^2$ s.t. $\lambda \sum_i |w_i| < 1$.
  - **Elastic net**: $\text{argmin}_w \|y - Xw\|^2 + \lambda_2 \|w\|^2 + \lambda_1 \|w\|_1$.
- **Spa**

  A^T),

  $\lambda_1 \|\beta\|_1$

  $v_i = \widehat{\beta}_i / \|\widehat{\beta}\|$

  v_i sparse PCA components

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

- Using ideas from
  - **Lasso**: $w_{lasso} \equiv \text{argmin}_w \|y - Xw\|^2$ s.t $\lambda \sum_i |w_i| < 1$.
  - **Elastic net**: $\text{argmin}_w \|y - Xw\|^2 + \lambda_2 \|w\|^2 + \lambda_1 \|w\|_1$.
- **Spa**),

A^T $\lambda_1 \|\beta\|_1$

$$v_i = \widehat{\beta}_i / \|\widehat{\beta}\|$$

v_i sparse PCA components

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