# COMS30035, Machine learning: Principal components analysis (PCA)

James Cussens
james.cussens@bristol.ac.uk

Department of Computer Science, SCEEM
University of Bristol

October 28, 2022

# Agenda

- PCA (standard presentation)

# Dimensionality reduction

- Sometimes it is obvious we can throw away a dimension (i.e. a variable).

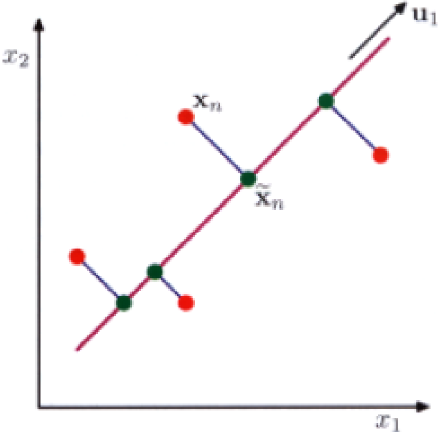
```
[5.1, 3.5, 1.4, 0.2, 1],
[4.9, 3. , 1.4, 0.2, 1],
[4.7, 3.2, 1.3, 0.2, 1],
[4.6, 3.1, 1.5, 0.3, 1],
[5. , 3.6, 1.4, 0.2, 1],
....
```

- The idea with PCA is to rotate the data (i.e. choose a different co-ordinate system) so that we end up with dimensions with low variance …
- …which we can throw away without losing much information.

# Motivations for PCA

- We can either view PCA as looking for projections with maximum variance [Bis06, §12.1.1],
- or looking for projections which minimise the distance from the original points to their projections [Bis06, §12.1.2].
- These are equivalent (we get the same projections)
- I will present the derivation in terms of maximising variance.

# PCA in a picture (Bishop Fig 12.2)




James Cussens
james.cussens@bristol.ac.uk

# From $D$ dimensions to 1

- A projection from $D$ dimensions down to 1 is defined by a $D$ dimensional vector $\mathbf{u}_1$ (which we can choose to be a unit vector so $\mathbf{u}_1^T\mathbf{u}_1 = 1$).
- The projection of $\mathbf{x}$ is simply $\mathbf{u}_1^T\mathbf{x}$.
- So which projection (which $\mathbf{u}_1$) is ‘best’?

# Eigenvector projections

Given a bunch of $N$ data points $\mathbf{x}_n$, the sample covariance matrix is:

$$\mathbf{S} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^T$$

- The variance of the *projected data* is $\mathbf{u}_1^T\mathbf{S}\mathbf{u}_1$.
- By the usual method of differentiating (w.r.t. to $\mathbf{u}_1$) and setting to 0 we [Bis06, p. 562] find that

$$\mathbf{S}\mathbf{u}_1 = \lambda_1\mathbf{u}_1 \tag{1}$$

- So $\mathbf{u}_1$ is an eigenvector of $\mathbf{S}$ (with eigenvalue $\lambda_1$).
- Since $\mathbf{u}_1^T\mathbf{S}\mathbf{u}_1 = \lambda_1$, we maximise variance by setting $\mathbf{u}_1$ to be the eigenvector with the biggest eigenvalue.
- This eigenvector is the called *the first principal component*.

# And so on

- The second principal component is that direction which maximises projected variance **subject to being orthogonal to the first principal component**.
- Each subsequent principal component is chosen to maximise variance subject to being orthogonal to all previous principal components.
- It can be shown that the principal components are the eigenvectors of the covariance matrix ordered by eigenvalue.

# New co-ordinates

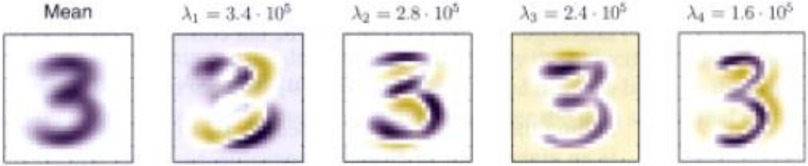
We have

$$\mathbf{x}_n = \sum_{i=1}^{D} (\mathbf{x}_n^T \mathbf{u}_i)\mathbf{u}_i = \sum_{i=1}^{D} \alpha_{ni}\mathbf{u}_i \quad (2)$$

- So each datapoint is a linear combination of principal components (= eigenvectors),
- but we (typically) only keep $M < D$ of these dimensions.
- When approximating a $D$-dimensional datapoint $\mathbf{x}_n$ by an $M$-dimensional vector $\tilde{\mathbf{x}}_n$ the best PCA approximation accounts for the mean $\bar{\mathbf{x}}$ by adding a constant vector $\bar{\mathbf{x}} - \sum_{i=1}^{M}(\bar{\mathbf{x}}^\top \mathbf{u}_i)\mathbf{u}_i$:

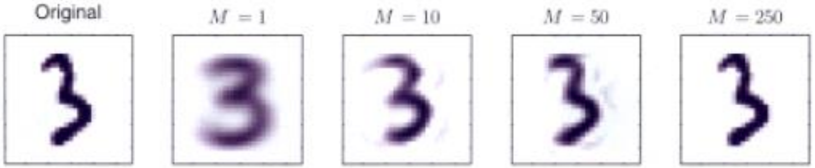
$$\begin{aligned}
\tilde{\mathbf{x}}_n &= \bar{\mathbf{x}} + \sum_{i=1}^{M} (\mathbf{x}_n^\top \mathbf{u}_i - \bar{\mathbf{x}}^\top \mathbf{u}_i)\mathbf{u}_i \\
&= \sum_{i=1}^{M} (\mathbf{x}_n^\top \mathbf{u}_i)\mathbf{u}_i + \bar{\mathbf{x}} - \sum_{i=1}^{M} (\bar{\mathbf{x}}^\top \mathbf{u}_i)\mathbf{u}_i
\end{aligned}$$

# Seeing the eigenvectors (Bishop Fig 12.3)




The mean vector $\overline{\mathbf{x}}$ along with the first four PCA eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_4$ for the off-line digits data set, together with the corresponding eigenvalues.

# Seeing PCA reconstructions (Bishop Fig 12.5)




An original example from the off-line digits data set together with its PCA reconstructions obtained by retaining $M$ principal components for various values of $M$. As $M$ increases the reconstruction becomes more accurate and would become perfect when $M = D = 28 \times 28 = 784$.

# Now do the quiz!

Yes, please do the quiz for this lecture on Blackboard!

Christopher M. Bishop.
*Pattern Recognition and Machine Learning*.
Springer, 2006.