COMS30035, Machine learning: Principal components analysis (PCA)

James Cussens
james.cussens@bristol.ac.uk

School of Computer Science,
University of Bristol

October 11, 2023

Dimensionality reduction

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

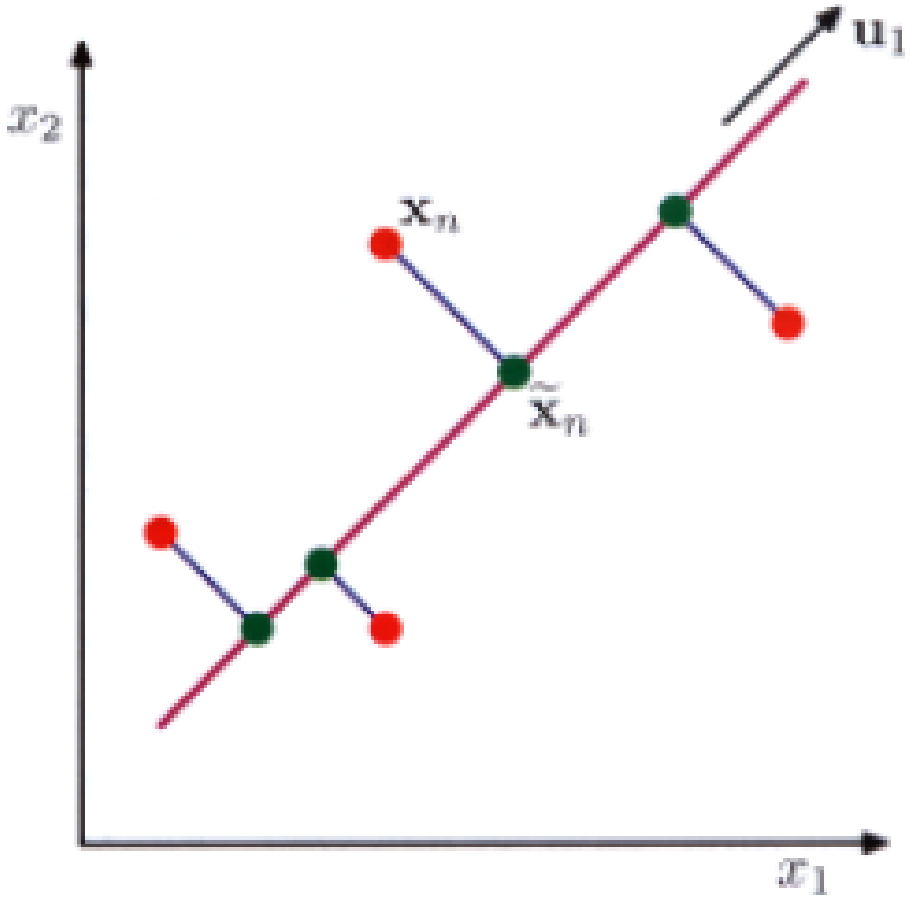
```
[5.1, 3.5, 1.4, 0.2, 1],
[4.9, 3. , 1.4, 0.2, 1],
[4.7, 3.2, 1.3, 0.2, 1],
[4.6, 3.1, 1.5, 0.3, 1],
[5. , 3.6, 1.4, 0.2, 1],
....
```

- 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)



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 $\mathbf{u}_1^T\mathbf{S}\mathbf{u}_1$.
- We want to find the $\mathbf{u}_1$ that maximises this subject to $\mathbf{u}_1^T\mathbf{u}_1 = 1$.
- Using some simple calculus (see [Bis06, p. 562]) we find that $\mathbf{u}_1$ must satisfy (1) for some scalar λ_1:

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

- So $\mathbf{u}_1$ is an eigenvector of $\mathbf{S}$ (with eigenvalue λ_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

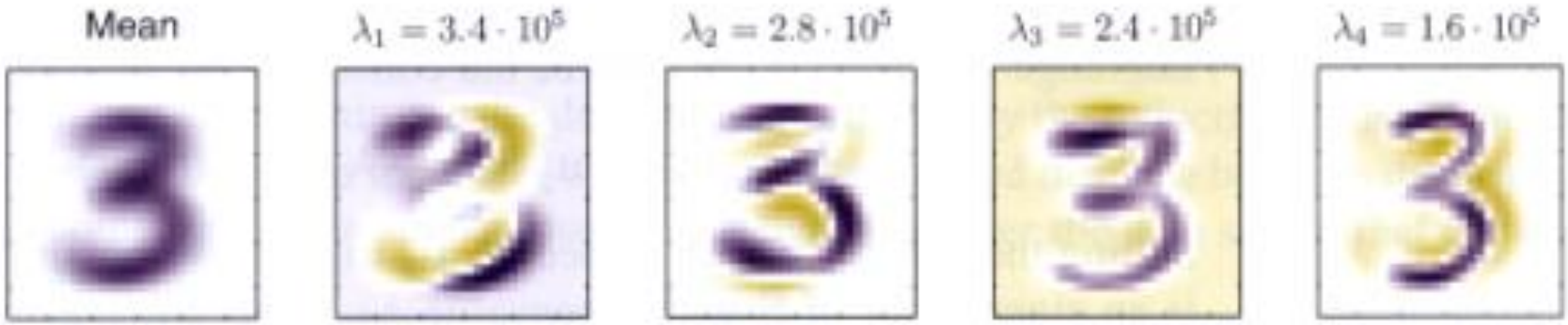
We have

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

- 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 $\bar{\mathbf{x}} - \sum_{i=1}^{M} (\bar{\mathbf{x}}^\top \mathbf{u}_i)\mathbf{u}_i$:

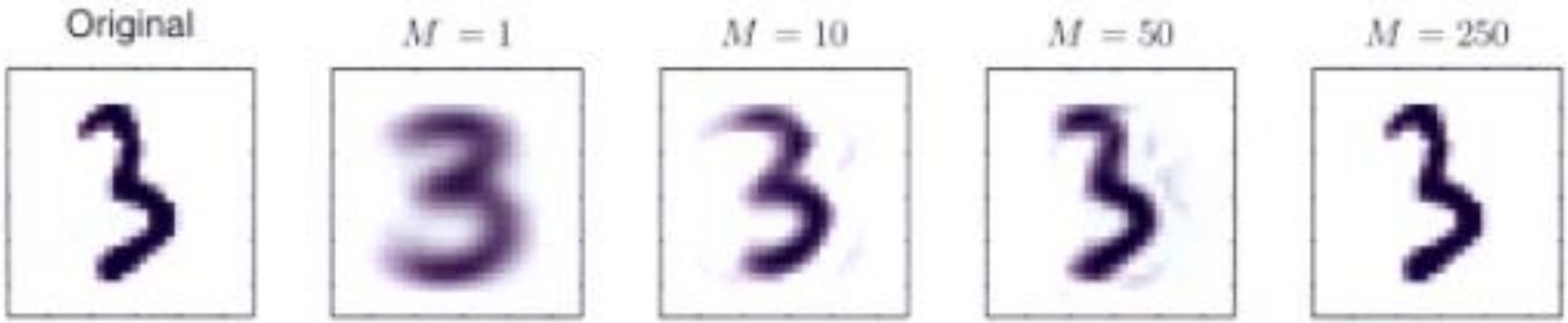
$$\begin{aligned} \tilde{\mathbf{x}}_n &= \bar{\mathbf{x}} + \sum_{i=1}^{M} (\mathbf{x}_n^\top \mathbf{u}_i - \bar{\mathbf{x}}^\top \mathbf{u}_i)\mathbf{u}_i \\ &= \sum_{i=1}^{M} (\mathbf{x}_n^\top \mathbf{u}_i)\mathbf{u}_i + \bar{\mathbf{x}} - \sum_{i=1}^{M} (\bar{\mathbf{x}}^\top \mathbf{u}_i)\mathbf{u}_i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

Seeing the eigenvectors (Bishop Fig 12.3)



The mean vector $\bar{\mathbf{x}}$ 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 $M = D = 28 \times 28 = 784$.

Probabilistic PCA

- Basic idea: reformulate PCA as the maximum likelihood solution to a latent variable model.
- Unlike with mixtures of Gaussians the latent variable here is continuous.
- That's why PCA is in the *Continuous Latent Variables* chapter of Bishop.

The PPCA model

The latent variable $\mathbf{z}$ has a zero-mean unit-covariance Gaussian distribution:

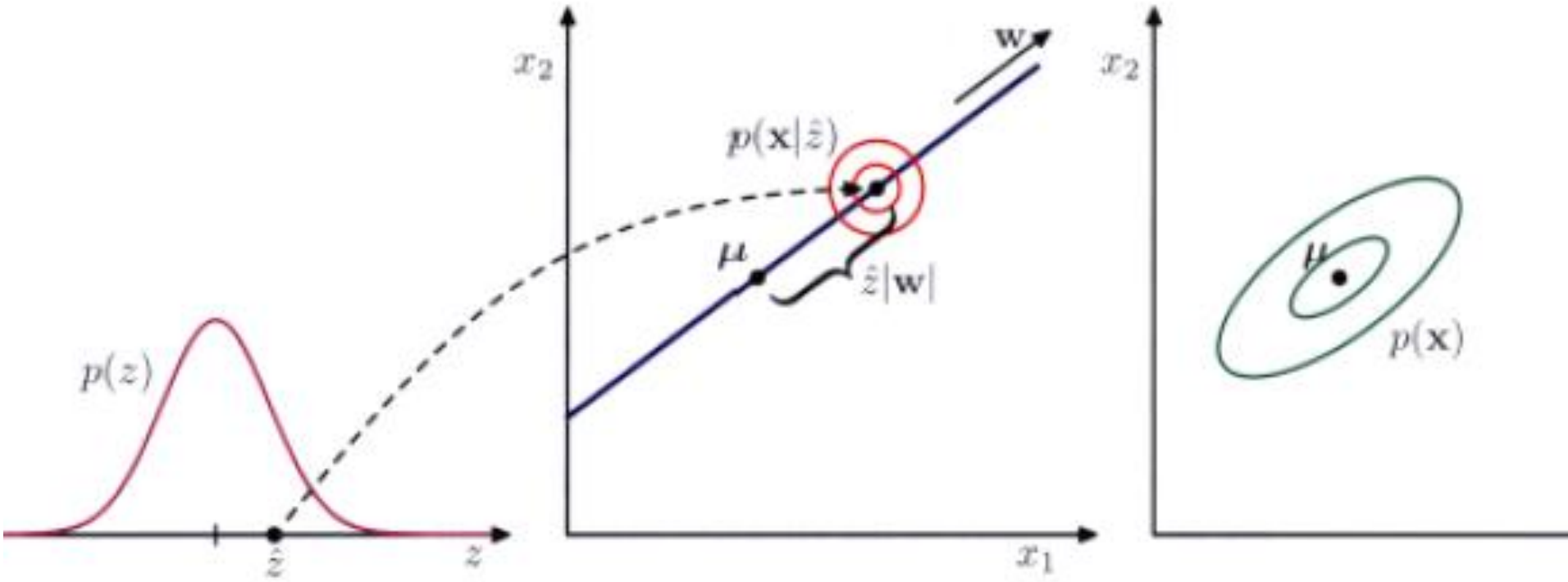
$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I}) \tag{3}$$

The distribution of the observed data conditional on this latent variable is another Gaussian:

$$p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}|\mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \sigma^2\mathbf{I}) \tag{4}$$

- So the parameters (to learn) are: $\mathbf{W}$, $\boldsymbol{\mu}$ and σ^2.
- $\mathbf{W}$ is an $D \times M$ matrix where D is the dimension of the data and M is the dimension of the PCA space, where $M \leq D$.

The generative view of PPCA (Bishop Fig 12.9)



Maximum likelihood for PPCA

- The good (if unsurprising) news is that the MLE parameters for PPCA can be computed exactly in closed form [Bis06, §12.2.1].
- And can be 'read off' from the (M first) eigenvectors and eigenvalues of the sample covariance matrix.
- The MLE estimate for $\boldsymbol{\mu}$ is just the sample mean.
- We might still resort to EM if the sample covariance matrix is huge, or if we have to deal with missing values in the data.

Why PPCA?

- Choosing M: since we now have a likelihoood, we can use cross-validation or a Bayesian approach with a special prior on $\mathbf{W}$.
- We can make connections to closely related models like *factor analysis* (which is just a small generalisation of PPCA.)
- We can generate data from a given PPCA model.


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