Unsupervised Dimension Reduction and Matrix Factorisation

David Barber

Learning Objectives

Lectures

- Why we want to find low-dimensional representations of data.
- How we can do this using Principal Components Analysis.
- How to go beyond PCA, including Non-Negative matrix factorisation.
- Example applications in Recommender systems, signal processing, text analysis.

Practicals

- PCA
- Non-negative Matrix Factorisation

High-Dimensional Spaces – Low Dimensional Manifolds

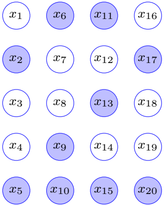
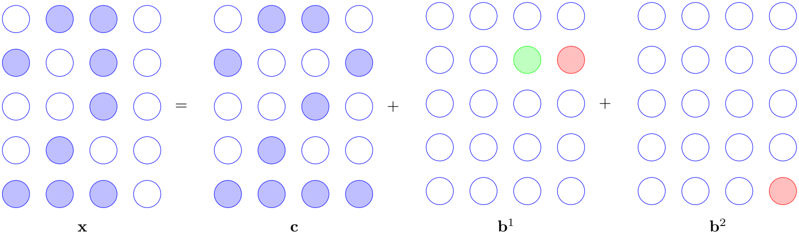

x_1 x_6 x_{11} x_{16}
x_2 x_7 x_{12} x_{17}
x_3 x_8 x_{13} x_{18}
x_4 x_9 x_{14} x_{19}
x_5 x_{10} x_{15} x_{20}

Image represented by a 20 dimensional vector:
$\mathbf{x} = (0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1)^{\mathsf{T}}$.

- Data can be high dimensional – images (here a ‘2’).
- As we move through the 20 dimensional space, is each point in the space likely to look like a ‘2’?
- No – only a very limited part of the space correspond to images that look like ‘2s’ – the space is highly constrained. There are only certain directions within the space that correspond to sensible ‘2s’ – we want to find this small number of relevant directions.

demoLowDManifold.m

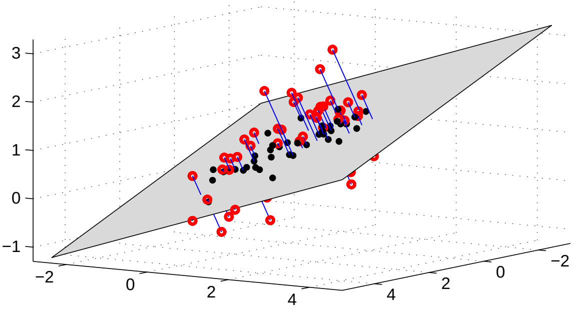
High-Dimensional Spaces – Low Dimensional Manifolds



- We can describe vectors by first finding the mean $\mathbf{c}$ of all images of ‘2s’.
- Then we can add on vectors in directions $\mathbf{b}^1$ and $\mathbf{b}^2$ to build up other images that look like ‘2s’.

High-Dimensional Spaces – Low Dimensional Manifolds

Provided there is some ‘structure’, data will typically lie close to a much lower dimensional ‘manifold’. Here we concentrate on computationally efficient linear dimension reduction techniques.



Principal Components Analysis (PCA)

Full rank representation

Any D-dimensional vector $\mathbf{x}$ can be written as

$$\mathbf{x} = \sum_{j=1}^{D} y_j \mathbf{b}^j$$

for linearly independent basis vectors $\mathbf{b}$.

Reduced rank approximation

$$\mathbf{x} \approx \mathbf{c} + \sum_{j=1}^{M} y_j \mathbf{b}^j \equiv \tilde{\mathbf{x}}$$

- The vector $\mathbf{c}$ defines a fixed point in the subspace and $M < D$.
- The $\mathbf{b}^j$ are 'basis' vectors that span the subspace. Collectively we can write $\mathbf{B} = \left[\mathbf{b}^1, \ldots, \mathbf{b}^M\right]$.
- The y_i are the low dimensional co-ordinates of the data.
- We can write

$$\tilde{\mathbf{x}} = \mathbf{c} + \mathbf{B}\mathbf{y}$$

Minimal square loss approximation

- We have a collection of N D-dimensional data vectors $\mathbf{x}^1, \ldots, \mathbf{x}^N$.
- To determine the best low rank approximation we can minimise the square distance error between $\mathbf{x}^n$ and its approximation $\tilde{\mathbf{x}}^n$:

$$E(\mathbf{B}, \mathbf{Y}, \mathbf{c}) = \sum_{n=1}^{N} \sum_{i=1}^{D} \left[x_i^n - \tilde{x}_i^n\right]^2$$

Redundancy

- Formally, the least squares criterion doesn't uniquely define the directions.
- Trivially, we could use $-\mathbf{b}^i$ instead of $\mathbf{b}^i$, but more generally we can use any vectors which span the same space as $\mathbf{B}$.
- These solutions all have the same square loss – they are all 'optimal'.
- We can make the solution essentially unique by requiring that all the $\mathbf{b}$ are orthogonal to each other and of unit length.

Least Squares solution

- Compute the mean of the data:

$$\mathbf{m} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}^n$$

- Compute the matrix:

$$\hat{\mathbf{X}} = \left[\mathbf{x}^1 - \mathbf{m}, \ldots, \mathbf{x}^N - \mathbf{m}\right]$$

- Find the Singular Value Decomposition

$$\hat{\mathbf{X}} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathsf{T}}$$

- The first M columns of $\mathbf{U}$ form the matrix $\mathbf{B}$.
- The lower dimensional representations are then given by $\mathbf{Y} = \mathbf{B}^{\mathsf{T}}\hat{\mathbf{X}}$.
- The approximate reconstructions are given by $\tilde{\mathbf{X}} \equiv \mathbf{B}\mathbf{Y} + \mathbf{M}$ where $\mathbf{M} = [\mathbf{m} \ldots \mathbf{m}]$.

Eigenvectors and PCA

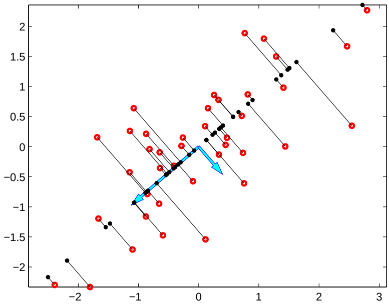
- One can show that the above SVD approach is equivalent to finding an eigen-decomposition of the sample covariance matrix

$$\frac{1}{N-1}\sum_{n=1}^{N}\left(\mathbf{x}^n - \mathbf{m}\right)\left(\mathbf{x}^n - \mathbf{m}\right)^{\mathsf{T}}$$

  and taking the leading M eigenvectors (which form the columns of $\mathbf{B}$) and their corresponding eigenvalues λ_i.
- The singular values and eigenvalues are related by

$$\lambda_i = D_{ii}^2$$

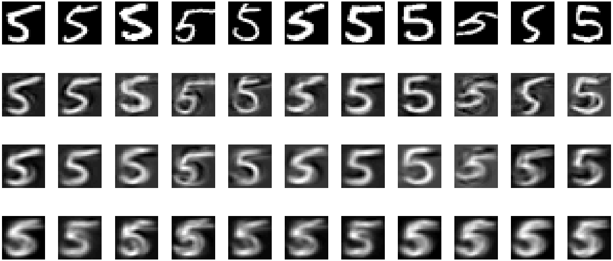
PCA: Reducing 2D data to a 1D approximation



The original datapoints $\mathbf{x}$ (larger rings) and their reconstructions $\tilde{\mathbf{x}}$ (small dots) using 1 dimensional PCA. The lines represent the orthogonal projection of the original datapoint onto the first eigenvector. The arrows are the two eigenvectors scaled by the square root of their corresponding eigenvalues. For each datapoint $\mathbf{x}$, the 'low dimensional' representation $\mathbf{y}$ is given by the distance (possibly negative) from the origin along the first eigenvector direction to the corresponding orthogonal projection point.

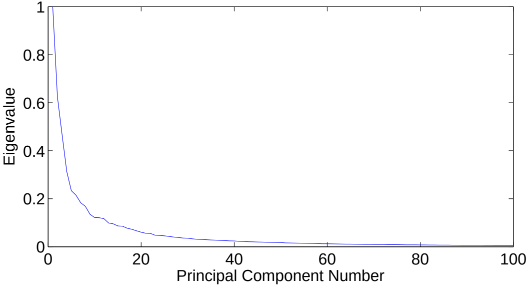
Reducing the dimension of digits

Each digit image consists of $28 \times 28 = 784$ pixels, each pixel in the range $0, \ldots, 255$. For each image, we first form a vector of the image by concatenating the columns of the image matrix into a 784 dimensional vector.



Top row : a selection of '5s' taken from the database of 892 examples. Plotted beneath each digit is the reconstruction using 100, 30 and 5 principal components (from top to bottom). Note how the reconstructions for fewer PCs express less variability from each other, and resemble more a mean 5 digit.

Eigen-spectrum of the ‘5’ digits



- 100 largest eigenvalues (scaled so that the largest is 1).
- If we order the eigenvalues $\lambda_1 \geq \lambda_2, \ldots$, the squared error is then given by the sum of the neglected eigenvalues

$$E = (N - 1) \sum_{i=M+1}^{D} \lambda_i$$

- The ‘latent’ dimensionality of data can be defined by a ‘kink’ in the spectrum (where the eigenvalues stop rapidly decreasing)

PCA via Singular Value Decomposition

- In practice, it can be impractical to first compute the covariance matrix and then the eigenvalues.
- A mathematically equivalent but computationally more convenient approach is to consider the SVD decomposition of the data matrix:

  $$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^\mathsf{T}$$

  where $\mathbf{U}^\mathsf{T}\mathbf{U} = \mathbf{I}_D$ and $\mathbf{V}^\mathsf{T}\mathbf{V} = \mathbf{I}_N$ and $\mathbf{D}$ is a diagonal matrix of the (positive) singular values.
- We can then approximate

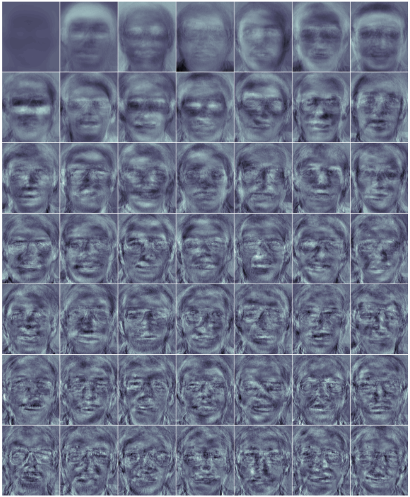
  $$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^\mathsf{T} \approx \mathbf{U}_M\mathbf{D}_M\mathbf{V}_M^\mathsf{T}$$

  where $\mathbf{U}_M$, $\mathbf{D}_M$, $\mathbf{V}_M$ correspond to taking only the first M singular values of the full matrices.
- PCA corresponds to setting $\mathbf{B} = \mathbf{U}_M$ and the eigenvalues are the diagonal elements of $\mathbf{D}_M$ squared.

Finding a low dimensional representation of Faces

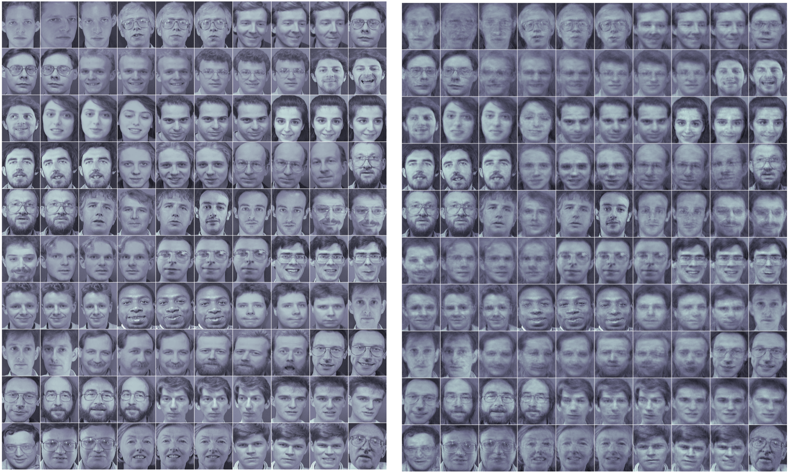
100 of the 120 training images (40 people, with 3 images of each person). Each image consists of $92 \times 112 = 10304$ non-negative greyscale pixels.

Eigenfaces: the 49 largest Principal Components



The eigenvectors (each plotted as an image) corresponding to the largest 49 eigenvalues.

Reconstruction using PCA with 49 components



(a) (b)

Figure : **(a)**: Original Data. **(b)**: PCA reconstruction of the images using a combination of the 49 principal components.

PCA Mathematics

Least Squares Objective

To determine the best lower dimensional representation it is convenient to use the square distance error between $\mathbf{x}$ and its reconstruction $\tilde{\mathbf{x}}$:

$$E(\mathbf{B}, \mathbf{Y}, \mathbf{c}) = \sum_{n=1}^{N} \sum_{i=1}^{D} \left[x_i^n - \tilde{x}_i^n\right]^2$$

Centering

The optimal bias $\mathbf{c}$ is given by the mean of the data $\sum_n \mathbf{x}^n / N$. We therefore assume that the data has been centred (has zero mean $\sum_n \mathbf{x}^n = \mathbf{0}$), so that we can set $\mathbf{c}$ to zero, and concentrate on finding the optimal basis $\mathbf{B}$ below.

PCA Mathematics (2): Rotation Invariance

We wish to minimize the sum of squared differences between each vector $\mathbf{x}$ and its reconstruction $\tilde{\mathbf{x}}$:

$$E(\mathbf{B},\mathbf{Y}) = \sum_{n=1}^{N}\sum_{i=1}^{D}\left[x_i^n - \sum_{j=1}^{M} y_j^n b_i^j\right]^2 = \mathrm{trace}\left((\mathbf{X}-\mathbf{B}\mathbf{Y})^{\mathsf{T}}(\mathbf{X}-\mathbf{B}\mathbf{Y})\right)$$

where $\mathbf{X} = \left[\mathbf{x}^1, \ldots, \mathbf{x}^N\right]$.

Orthonormality constraint

Consider an invertible transformation $\mathbf{Q}$ of the basis $\mathbf{B}$ so that $\tilde{\mathbf{B}} \equiv \mathbf{B}\mathbf{Q}$ is an orthonormal matrix, $\tilde{\mathbf{B}}^{\mathsf{T}}\tilde{\mathbf{B}} = \mathbf{I}$. Since $\mathbf{Q}$ is invertible, we may write $\mathbf{B}\mathbf{Y} = \tilde{\mathbf{B}}\mathbf{Q}^{-1}\mathbf{Y} \equiv \tilde{\mathbf{B}}\tilde{\mathbf{Y}}$, which is of then same form as $\mathbf{B}\mathbf{Y}$, albeit with an orthonormality constraint on $\tilde{\mathbf{B}}$. Hence, without loss of generality, we may impose the orthonormality constraint $\mathbf{B}^{\mathsf{T}}\mathbf{B} = \mathbf{I}$, namely that the basis vectors are mutually orthogonal and of unit length.

PCA mathematics: Finding the optimal $\mathbf{Y}$

$$-\frac{1}{2}\frac{\partial}{\partial y_k^n}E(\mathbf{B},\mathbf{Y}) = \sum_i \left[x_i^n - \sum_j y_j^n b_i^j \right] b_i^k = \sum_i x_i^n b_i^k - \sum_j y_j^n \underbrace{\sum_i b_i^j b_i^k}_{\delta_{jk}}$$

The squared error $E(\mathbf{B},\mathbf{Y})$ therefore has zero derivative when

$$y_k^n = \sum_i b_i^k x_i^n, \qquad \text{which can be written as} \quad \mathbf{Y} = \mathbf{B}^\mathsf{T}\mathbf{X}$$

The objective $E(\mathbf{B})$ becomes

$$E(\mathbf{B}) = \operatorname{trace}\left((\mathbf{X} - \mathbf{B}\mathbf{Y})^\mathsf{T} (\mathbf{X} - \mathbf{B}\mathbf{Y}) \right) = \operatorname{trace}\left(\mathbf{X}^\mathsf{T} \left(\mathbf{I} - \mathbf{B}\mathbf{B}^\mathsf{T}\right)^2 \mathbf{X} \right)$$

Since $\left(\mathbf{I} - \mathbf{B}\mathbf{B}^\mathsf{T}\right)^2 = \mathbf{I} - \mathbf{B}\mathbf{B}^\mathsf{T}$, (using $\mathbf{B}^\mathsf{T}\mathbf{B} = \mathbf{I}$)

$$E(\mathbf{B}) = \operatorname{trace}\left(\mathbf{X}\mathbf{X}^\mathsf{T} \left(\mathbf{I} - \mathbf{B}\mathbf{B}^\mathsf{T}\right) \right)$$

PCA mathematics: The role of the sample covariance

Hence the objective becomes

$$E(\mathbf{B}) = (N-1)\left[\mathrm{trace}\left(\mathbf{S}\right) - \mathrm{trace}\left(\mathbf{S}\mathbf{B}\mathbf{B}^{\mathsf{T}}\right)\right]$$

where $\mathbf{S}$ is the sample covariance matrix of the centred data. Since we assumed the data is zero mean, this is

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

General case

If we hadn't centred the data, we would have

$$\mathbf{S} = \frac{1}{N-1}\sum_{n=1}^{N}\left(\mathbf{x}^{n} - \mathbf{m}\right)\left(\mathbf{x}^{n} - \mathbf{m}\right)^{\mathsf{T}}$$

where $\mathbf{m}$ is the sample mean of the data.

PCA maths: Enforcing orthonormality

To minimise $E(\mathbf{B})$ under the constraint $\mathbf{B}^\mathsf{T}\mathbf{B} = \mathbf{I}$ we use a set of Lagrange multipliers $\mathbf{L}$, so that the objective is to minimize

$$-\mathrm{trace}\left(\mathbf{S}\mathbf{B}\mathbf{B}^\mathsf{T}\right) + \mathrm{trace}\left(\mathbf{L}\left(\mathbf{B}^\mathsf{T}\mathbf{B} - \mathbf{I}\right)\right)$$

Since the constraint is symmetric, we can assume that $\mathbf{L}$ is also symmetric. Differentiating with respect to $\mathbf{B}$ and equating to zero we obtain that at the optimum

$$\mathbf{S}\mathbf{B} = \mathbf{B}\mathbf{L}$$

We need to find matrices $\mathbf{B}$ and $\mathbf{L}$ that satisfy this equation. One solution is given when $\mathbf{L}$ is diagonal in which case this is a form of eigen-equation and the columns of $\mathbf{B}$ are the corresponding eigenvectors of $\mathbf{S}$.

In this case, $\mathrm{trace}\left(\mathbf{S}\mathbf{B}\mathbf{B}^{\mathsf{T}}\right) = \mathrm{trace}\left(\mathbf{L}\right)$, which is the sum of the eigenvalues corresponding to the eigenvectors forming $\mathbf{B}$.

$$\frac{1}{N-1}E(\mathbf{B}) = -\mathrm{trace}\left(\mathbf{L}\right) + \mathrm{trace}\left(\mathbf{S}\right) = -\sum_{i=1}^{M}\lambda_i + \text{const.}$$

Since we wish to minimise $E(\mathbf{B})$, we therefore define the basis using the eigenvectors with largest corresponding eigenvalues.

Residual error

If we order the eigenvalues $\lambda_1 \geq \lambda_2, \ldots$, the squared error is then given by

$$\frac{1}{N-1}E(\mathbf{B}) = \mathrm{trace}\left(\mathbf{S}\right) - \mathrm{trace}\left(\mathbf{L}\right) = \sum_{i=1}^{D}\lambda_i - \sum_{i=1}^{M}\lambda_i = \sum_{i=M+1}^{D}\lambda_i$$

Hence the sum of the neglected eigenvalues equals the squared reconstruction error.

Uniqueness

Whilst the solution to this eigen-problem is unique, this only serves to define the solution subspace since one may rotate and scale $\mathbf{B}$ and $\mathbf{Y}$ such that the value of the squared loss is exactly the same

PCA: Breaking rotational Invariance

To break the invariance of least squares projection with respect to rotations and rescaling, we need an additional criterion. One such is given by first searching for the single direction $\mathbf{b}$ such that the variance of the data projected onto this direction is maximal amongst all possible such projections. For a single vector $\mathbf{b}$ we have

$$y^n = \sum_i b_i x_i^n$$

The projection of a datapoint onto a direction $\mathbf{b}$ is $\mathbf{b}^\mathsf{T}\mathbf{x}^n$ for a unit length vector $\mathbf{b}$. Hence the sum of squared projections is

$$\sum_n \left(\mathbf{b}^\mathsf{T}\mathbf{x}^n\right)^2 = \mathbf{b}^\mathsf{T}\left[\sum_n \mathbf{x}^n \left(\mathbf{x}^n\right)^\mathsf{T}\right]\mathbf{b} = (N-1)\mathbf{b}^\mathsf{T}\mathbf{S}\mathbf{b}$$

The optimal single $\mathbf{b}$ which maximises the projection variance is given by the eigenvector corresponding to the largest eigenvalue of $\mathbf{S}$. Under the criterion that the next optimal direction $\mathbf{b}^{(2)}$ should be orthonormal to the first, one can readily show that $\mathbf{b}^{(2)}$ is given by the second largest eigenvector, and so on. These maximal variance directions found by PCA are called the principal directions.

PCA With Missing Data

- Often like to use PCA when there are elements of the data missing.
- There is no 'quick fix' PCA solution when some of the x_i^n are missing.
- One approach is to require the squared reconstruction error to be small only for the existing elements of $\mathbf{X}$. That is

$$E(\mathbf{B}, \mathbf{Y}) = \sum_{n=1}^{N} \sum_{i=1}^{D} \gamma_i^n \left[x_i^n - \sum_j y_j^n b_i^j \right]^2$$

  where $\gamma_i^n = 1$ if the i^{th} entry of the n^{th} vector is available, and is zero otherwise.
- There are efficient iterative schemes to find $\mathbf{B}$ and $\mathbf{Y}$.

Matrix Completion

- Given a matrix $\mathbf{X}$ with missing entries, we would like to 'infer' the missing values.
- If the elements of the matrix can be well approximated by a low-rank factorisation, we can use PCA to 'fill in' the missing values.

PCA with missing data

Optimize $\mathbf{Y}$ for fixed $\mathbf{B}$

$$E(\hat{\mathbf{B}}, \mathbf{Y}) = \sum_{n=1}^{N} \sum_{i=1}^{D} \gamma_i^n \left[x_i^n - \sum_j y_j^n \hat{b}_i^j \right]^2$$

For fixed $\hat{\mathbf{B}}$ the above $E(\hat{\mathbf{B}}, \mathbf{Y})$ is a quadratic function of the matrix $\mathbf{Y}$, which can be optimised directly. By differentiating and equating to zero, one obtains the fixed point condition

$$\sum_i \gamma_i^n \left(x_i^n - \sum_l y_l^n \hat{b}_i^l \right) \hat{b}_i^k = 0$$

Defining

$$\left[\mathbf{y}^{(n)}\right]_l = y_n^l, \qquad \left[\mathbf{M}^{(n)}\right]_{kl} = \sum_i \hat{b}_i^l \hat{b}_i^k \gamma_i^n, \qquad \left[\mathbf{c}^n\right]_k = \sum_i \gamma_i^n x_i^n \hat{b}_i^k,$$

in matrix notation, we then have a set of linear systems:

$$\mathbf{c}^{(n)} = \mathbf{M}^{(n)} \mathbf{y}^{(n)}, \qquad n = 1, \ldots, N$$

PCA with missing data

Optimize $\mathbf{B}$ for fixed $\mathbf{Y}$

One now freezes $\hat{\mathbf{Y}}$ and considers the function

$$E(\mathbf{B}, \hat{\mathbf{Y}}) = \sum_{n=1}^{N} \sum_{i=1}^{D} \gamma_i^n \left[x_i^n - \sum_j \hat{y}_j^n b_i^j \right]^2$$

For fixed $\hat{\mathbf{Y}}$ the above expression is quadratic in the matrix $\mathbf{B}$, which can again be optimised using linear algebra. This corresponds to solving a set of linear systems for the i^{th} row of $\mathbf{B}$:

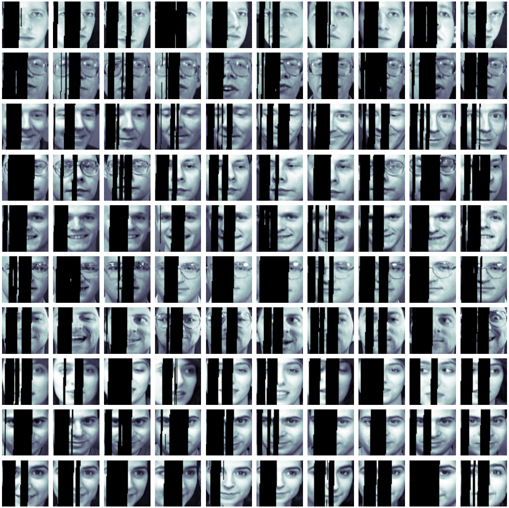
$$\mathbf{m}^{(i)} = \mathbf{F}^{(i)} \mathbf{b}^{(i)}$$

where

$$\left[\mathbf{m}^{(i)}\right]_k = \sum_n \gamma_i^n x_i^n \hat{y}_k^n, \qquad \left[\mathbf{F}^{(i)}\right]_{kj} = \sum_n \gamma_i^n \hat{y}_j^n \hat{y}_k^n$$

Mathematically, this is $\mathbf{b}^{(i)} = {\mathbf{F}^{(i)}}^{-1} \mathbf{m}^{(i)}$. In this manner one is guaranteed to iteratively decrease the value of the squared error loss until a minimum is reached.

PCA With Missing Data: matrix completion



The training dataset consists of 400 images with randomly removed segments in each image.

PCA With Missing Data: matrix completion

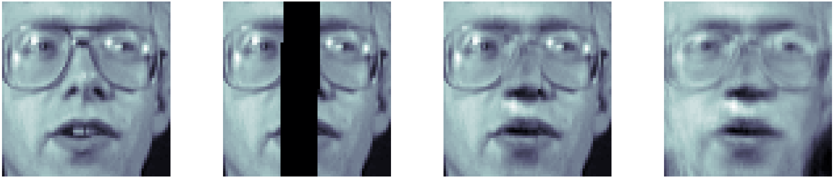


Figure : Left: original complete datapoint (one of the 400 in the original complete training set). Second image: data vector with missing entries. The new training set is composed of such images with randomly selected strips of the image missing. Third image: reconstruction of the missing data vector entries using 50 components. Right: reconstruction of the whole image (from the missing data images).

`demoSVDmissingFace.m`

Collaborative Filtering: Recommending Movies

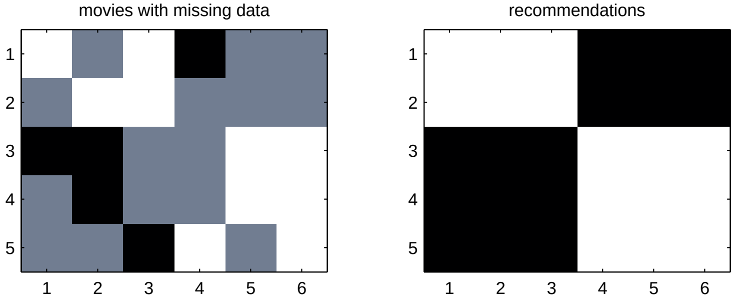
User

	1	2	3	4	5	6
Terminator 1	🟢		🟢	🔴		
Terminator 2		🟢	🟢			
Toy Story 1	🔴	🔴			🟢	🟢
Toy Story 2		🔴			🟢	🟢
Invincibles			🔴	🟢		🟢

This is a matrix completion problem.

Collaborative Filtering: Recommending Movies

5 movies and 6 users:



Left: gray represents missing data. White is $+1$ (like), black is -1 (don't like).
Right: Reconstruction using PCA with 2 components and taking the sign of the reconstructions.

`demoSVDmissingMovie.m`

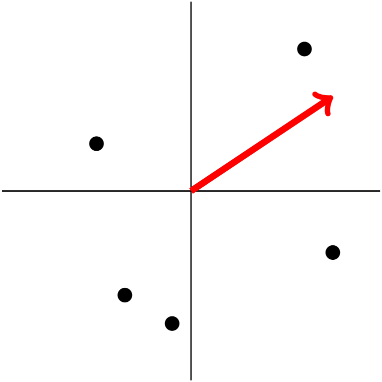
Matrix Decompositions

Given a data matrix $\mathbf{X}$ for which each column represents a datapoint, an approximate matrix decomposition is of the form $\mathbf{X} \approx \mathbf{B}\mathbf{Y}$ into a basis matrix $\mathbf{B}$ and weight (or coordinate) matrix $\mathbf{Y}$. Symbolically, matrix decompositions are of the form

$$\underbrace{\begin{pmatrix} \\ X : \text{Data} \\ \\ \end{pmatrix}}_{D \times N} \approx \underbrace{\begin{pmatrix} \\ B : \text{Basis} \\ \\ \end{pmatrix}}_{D \times M} \underbrace{\begin{pmatrix} Y : \text{Weights/Components} \end{pmatrix}}_{M \times N}$$

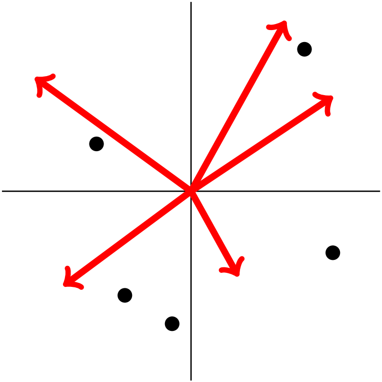
Based on the SVD of the data matrix, we see that PCA is in this class. Many methods can be considered as matrix decompositions under specific constraints.

Under-complete decompositions



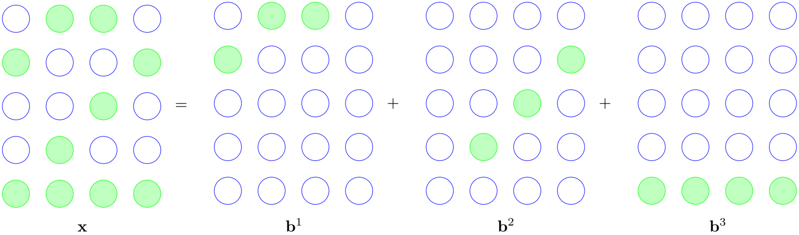
When $M < D$, there are fewer basis vectors than dimensions. The matrix $\mathbf{B}$ is then called 'tall' or 'thin'. In this case the matrix $\mathbf{Y}$ forms a lower dimensional approximate representation of the data $\mathbf{X}$, PCA being a classic example.

Over-complete decompositions



For $M > D$ the basis is over-complete, there being more basis vectors than dimensions. In such cases additional constraints are placed on either the basis or components. For example, one might require that only a small number of the large number of available basis vectors is used to form the representation for any given $\mathbf{x}$. Such sparse-representations are common in theoretical neurobiology where issues of energy efficiency, rapidity of processing and robustness are of interest.

Non-Negative Matrix Factorisation (NNMF)



- Can describe a positive data vector by adding up positive parts of a '2'.
- The parts can be more interpretable than those given by PCA.
- If applied to Item-Basket analysis, it would find essentially what kinds of things are bought together. For example a basis vector (or ‘component’) might be analogous to say ‘vegetables’ and another might be ‘dairy’. We might then expect a customer basket to be able to be described by the amount that it contains vegetables and the amount it contains dairy.

Non-Negative Matrix Factorisation

- In some cases it is more natural to consider that the data is composed of positive amounts of positive objects.
- We therefore seek an approximate decomposition

  $$X_{ij} \approx \sum_k B_{ik} Y_{kj}$$

  for non-negative B and non-negative Y.
- For example the columns of the data matrix $\mathbf{X}$ might be images. Then the columns of the matrix $\mathbf{B}$ are the 'basis' vectors that we will use to reconstruct these images. The positive weights of the reconstruction are given by $\mathbf{Y}$.

NNMF: Probabilistic latent semantic analysis

Consider two objects, x and y, where $\mathrm{dom}(x) = \{1, \ldots, I\}$ and $\mathrm{dom}(y) = \{1, \ldots, J\}$ and a dataset $(x^n, y^n, n = 1, \ldots, N)$. We have a count matrix with elements C_{ij} which describes the number of times the joint state $x = i, y = j$ was observed in the dataset. We can transform this count matrix into a frequency matrix p with elements

$$p(x = i, y = j) = \frac{C_{ij}}{\sum_{ij} C_{ij}}$$

Our interest is to find a decomposition of this frequency matrix of the form

$$\underbrace{p(x = i, y = j)}_{X_{ij}} \approx \sum_k \underbrace{\tilde{p}(x = i|z = k)}_{B_{ik}} \underbrace{\tilde{p}(y = j|z = k)\tilde{p}(z = k)}_{Y_{kj}} \equiv \tilde{p}(x = i, y = j)$$

where all quantities $\tilde{p}$ are distributions. This is then a form of matrix decomposition into positive basis $\mathbf{B}$ and positive coordinates $\mathbf{Y}$. This has the interpretation of discovering latent topics z that describe the joint behaviour of x and y.

NNMF: An EM style training algorithm

For probabilities, a useful measure of discrepancy is the Kullback-Leibler divergence

$$\mathrm{KL}(p|\tilde{p}) = \left\langle \log p \right\rangle_p - \left\langle \log \tilde{p} \right\rangle_p$$

Since p is fixed, minimising the Kullback-Leibler divergence with respect to the approximation $\tilde{p}$ is equivalent to maximising the 'likelihood' term $\left\langle \log \tilde{p} \right\rangle_p$. This is

$$L \equiv \sum_{x,y} p(x,y) \log \tilde{p}(x,y)$$

It's convenient to derive an EM style algorithm to learn $\tilde{p}(x|z)$, $\tilde{p}(y|z)$ and $\tilde{p}(z)$.

Consider

$$\begin{aligned}\mathrm{KL}(q(z|x,y)|\tilde{p}(z|x,y))&\\ &= \sum_z q(z|x,y)\log q(z|x,y) - \sum_z q(z|x,y)\log \tilde{p}(z|x,y) \geq 0\end{aligned}$$

where $\sum_z$ implies summation over all states of the variable z. Using

$$\tilde{p}(z|x,y) = \frac{\tilde{p}(x,y,z)}{\tilde{p}(x,y)}$$

and rearranging, this gives the bound,

$$\log \tilde{p}(x,y) \geq -\sum_z q(z|x,y)\log q(z|x,y) + \sum_z q(z|x,y)\log \tilde{p}(z,x,y)$$

Plugging this into the 'likelihood' term above, we have the bound

$$\begin{aligned}L \geq -\sum_{x,y} p(x,y)\sum_z q(z|x,y)\log q(z|x,y)&\\ + \sum_{x,y} p(x,y)\sum_z q(z|x,y)\left[\log \tilde{p}(x|z) + \log \tilde{p}(y|z) + \log \tilde{p}(z)\right]&\end{aligned}$$

NNMF: M-step

For fixed $\tilde{p}(x|z), \tilde{p}(y|z)$, the contribution to the bound from $\tilde{p}(z)$ is

$$\sum_{x,y} p(x,y) \sum_z q(z|x,y) \log \tilde{p}(z)$$

Up to a constant, this is $\mathrm{KL}\Big(\sum_{x,y} q(z|x,y)p(x,y)|\tilde{p}(z)\Big)$ so that, optimally,

$$\tilde{p}(z) = \sum_{x,y} q(z|x,y)p(x,y)$$

Similarly, optimally

$$\tilde{p}(x|z) \propto \sum_y p(x,y)q(z|x,y)$$

and

$$\tilde{p}(y|z) \propto \sum_x p(x,y)q(z|x,y)$$

NNMF: E-step

The optimal setting for the q distribution at each iteration is

$$q(z|x,y) = \tilde{p}(z|x,y)$$

which is fixed throughout the M-step.

Convergence

L is guaranteed to increase (and the Kullback-Leibler divergence decrease) under iterating between the E and M-steps, since the method is analogous to an EM procedure.

NNMF: Conditional PLSA

In some cases it is more natural to consider a conditional frequency matrix

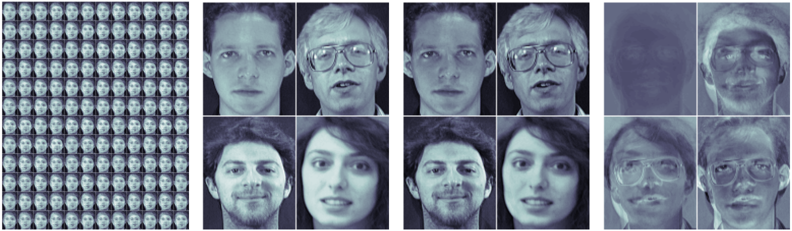
$$p(x = i|y = j)$$

and seek an approximate decomposition

$$\underbrace{p(x = i|y = j)}_{X_{ij}} \approx \sum_k \underbrace{\tilde{p}(x = i|z = k)}_{B_{ik}} \underbrace{\tilde{p}(z = k|y = j)}_{Y_{kj}}$$

Deriving an EM style algorithm for this is straightforward, being equivalent to the non-negative matrix factorisation algorithm.

Learning a positive basis



(a) (b) (c) (d)

Figure : **(a)**: Training data, consisting of a positive (convex) combination of the base images. **(b)**: The chosen base images from which the training data is derived. **(c)**: Basis learned using conditional PLSA on the training data. This is virtually indistinguishable from the true basis. **(d)**: Eigenbasis (sometimes called ‘eigenfaces’).

Positive reconstruction

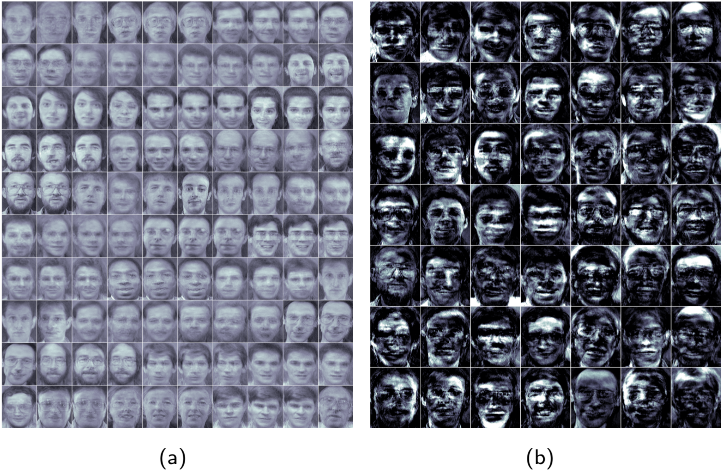
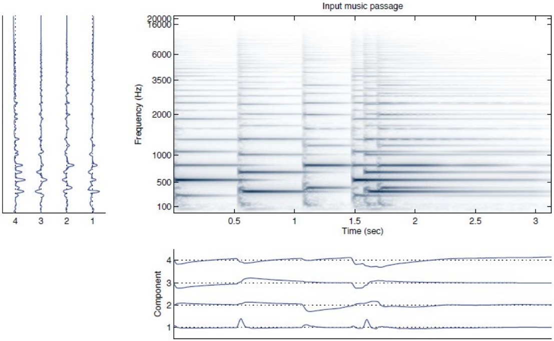


Figure : **(a)**: Conditional PLSA reconstruction of the images in using a positive convex combination of the 49 positive base images in **(b)**. The root mean square reconstruction error is 1.391×10^{-5}. The base images tend to be more ‘localised’ than the corresponding eigen-images. Here one sees local structure such as foreheads, chins, *etc.*

Decomposing Music Signals: PCA

- Compute the spectrogram of a music signal – gives a matrix $\mathbf{X}$ with elements $M_{f,t}$.
- Compute the PCA decomposition using 4 components.
- Doesn't give any nice interpretation.

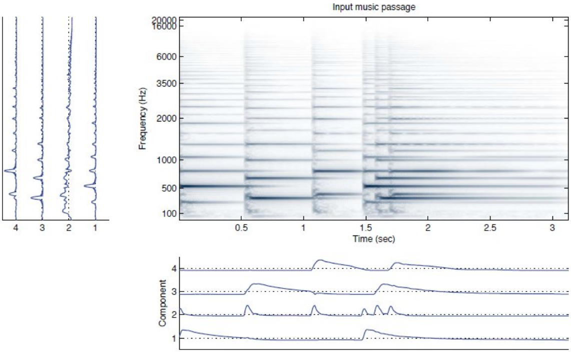


[1]

[1] Images from Paris Smaragdis WASPAA keynote 2013

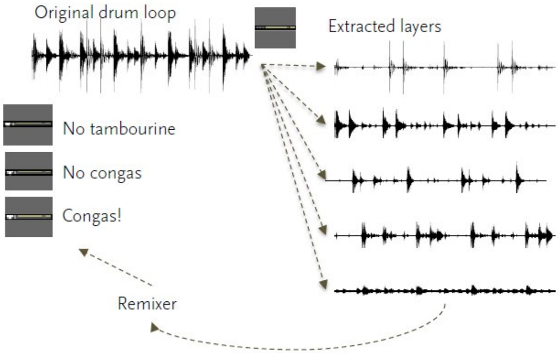
Decomposing Music Signals: NNMF

- Compute the spectrogram of a music signal – gives a matrix $\mathbf{X}$ with elements $M_{f,t}$.
- Compute the NNMF decomposition using 4 components.
- Gives a very nice interpretation – can see what each note corresponds to and when it is played.



Decomposing Music Signals: NNMF

- Can use this to isolate the component instruments in music.
- Eg. isolate the individual drums in a piece.



See Paris' demo

Topic Modelling

- Have a collection of documents.
- Each document may refer to more than one topic
- Eg. A document might discuss the role of climate change in Polynesia. `climate change` and `Polyneisia` might be two topics that we would like to identify.
- We want to do this in an unsupervised way – no one tells us what the topics are. We just have a collection of unlabelled documents and want to discover automatically what are the 'latent' topics.
- Since a single document could contain more than one topic – 'mixed membership' model.

Ways to do this

- NNMF (also called PLSA in the topic modelling community) is very easy and fast to implement.
- Latent Dirichlet Allocation is an alternative approach which is very similar mathematically but difficult to implement – not clear what the practical advantages are over NNMF.

Frequency Representation

- Define a fixed dictionary of words $d_1 = \texttt{aardvark}, \ldots, d_{10,000} = \texttt{zorro}$.
- For a given document n represent the document by the 10,000 dimensional frequency vector (number of occurrences of each word)

$$\mathbf{f}^n = (0, 0, 2, 0, 0, 1, \ldots, 1, 0)$$

This will be a very sparse vector.

- This is also called a ‘bag of words’ representation.
- We may also wish to remove so-called ‘stop words’ (common words such as ‘and’, ’the’, *etc.* which would otherwise dominate statistics and are anyway common to any topic).
- We may also wish to re-weight the frequencies using the so-called TF-IDF which means that we use instead a value that is high if the word appears many times in a document but gets scaled down if the same word appears in many documents.

Topic Modelling Example: NNMF

16,333 documents are taken from Associated Press corpus with a dictionary of 23,075 unique terms. Fit a topic model (NNMF) containing 4 topics.

Arts	Budgets	Children	Education
new	million	children	school
film	tax	women	students
show	program	people	schools
music	budget	child	education
movie	billion	years	teachers
play	federal	families	high
musical	year	work	public
best	spending	parents	teacher
actor	new	says	bennett
first	state	family	manigat
york	plan	welfare	namphy
opera	money	men	state
theater	programs	percent	president
actress	government	care	elementary
love	congress	life	haiti

(a)

The William Randolph Hearst Foundation will give $ 1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services, Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Centers share will be $200,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive $400,000 each. The Juilliard School, where music and the performing arts are taught, will get $250,000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual $100,000 donation, too.

(b)

Modelling citations

We have a collection of research documents which cite other documents. For example, document 1 might cite documents $3, 2, 10$, *etc.* Given only the list of citations for each document, can we identify key research papers and the communities that cite them?

A probabilistic formulation

We use the variable $d \in \{1, \ldots, D\}$ to index documents and $c \in \{1, \ldots, D\}$ to index citations (both d and c have the same domain, namely the index of a research article). If document $d = i$ cites article $c = j$ then we set the entry of the matrix $C_{ij} = 1$. If there is no citation, C_{ij} is set to zero. We can form a 'distribution' over documents and citations using

$$p(d = i, c = j) = \frac{C_{ij}}{\sum_{ij} C_{ij}}$$

and use PLSA to decompose this matrix into citation-topics.

Modelling citations

The Cora corpus contains an archive of around 30,000 computer science research papers. From this archive the papers in the machine learning category are extracted, consisting of 4220 documents and 38,372 citations.

Using PLSA

The joint PLSA method is fitted to the data using $z = 7$ topics. From the trained model the expression $p(c = j|z = k)$ defines how authoritative paper j is according to community $z = k$.

Modelling citations

factor 1	(Reinforcement Learning)
0.0108	Learning to predict by the methods of temporal differences. Sutton.
0.0066	Neuronlike adaptive elements that can solve difficult learning control problems. Barto et al.
0.0065	Practical Issues in Temporal Difference Learning. Tesauro.
factor 2	(Rule Learning)
0.0038	Explanation-based generalization: a unifying view. Mitchell et al.
0.0037	Learning internal representations by error propagation. Rumelhart et al.
0.0036	Explanation-Based Learning: An Alternative View. DeJong et al.
factor 3	(Neural Networks)
0.0120	Learning internal representations by error propagation. Rumelhart et al.
0.0061	Neural networks and the bias-variance dilemma. Geman et al.
0.0049	The Cascade-Correlation learning architecture. Fahlman et al.
factor 4	(Theory)
0.0093	Classification and Regression Trees. Breiman et al.
0.0066	Learnability and the Vapnik-Chervonenkis dimension. Blumer et al.
0.0055	Learning Quickly when Irrelevant Attributes Abound. Littlestone.
factor 5	(Probabilistic Reasoning)
0.0118	Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Pearl.
0.0094	Maximum likelihood from incomplete data via the em algorithm. Dempster et al.
0.0056	Local computations with probabilities on graphical structures. Lauritzen et al.
factor 6	(Genetic Algorithms)
0.0157	Genetic Algorithms in Search, Optimization, and Machine Learning. Goldberg.
0.0132	Adaptation in Natural and Artificial Systems. Holland.
0.0096	Genetic Programming: On the Programming of Computers by Means of Natural Selection. Koza.
factor 7	(Logic)
0.0063	Efficient induction of logic programs. Muggleton et al.
0.0054	Learning logical definitions from relations. Quinlan.
0.0033	Inductive Logic Programming Techniques and Applications. Lavrac et al.

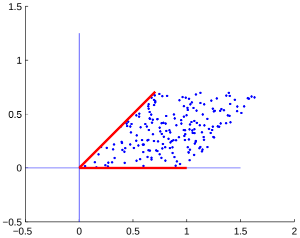
Table : Highest ranked documents according to $p(c|z)$. The factor topic labels are manual assignments based on similarity to the Cora topics.

Independent Components Analysis

- As we saw, PCA finds interesting subspaces, but does not really find interesting directions in the subspace.
- A useful coordinate system would be one in which the data we have could be generated by sampling independently along directions.
- ICA finds 'basis' vectors $\mathbf{b}^i$ and coefficients y such that the data can be approximately expressed as

$$\mathbf{x}^n \approx \sum_i y_i^n \mathbf{b}^i$$

where empirically the y_i^n are uncorrelated.



We can generate these datapoints by drawing a random value y_1 from the uniform distribution and independently drawing y_2 from the uniform distribution. A datapoint is then given by taking a linear combination $y_1\mathbf{b}^1 + y_2\mathbf{b}^2$ of the two red basis vectors.

ICA versus PCA

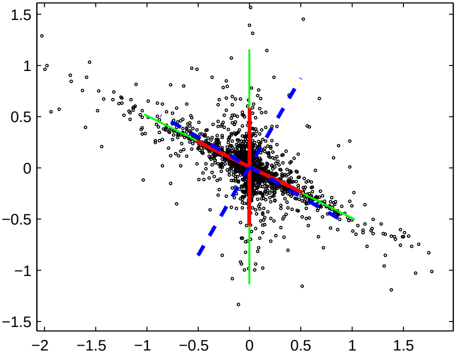


Figure : The red lines are the basis vectors estimated by ICA. For comparison, PCA produces the blue (dashed) components. As expected, PCA finds the orthogonal directions of maximal variation. ICA however, correctly estimates the directions in which the components were independently generated.

ICA versus PCA for faces

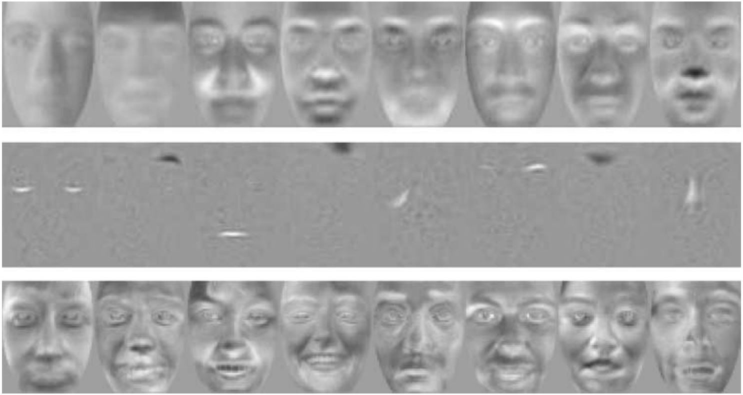


Figure : The top row are the 8 largest principal components. The second row is ICA in which the coefficients y are independent – this identifies things like the eyes, mouth, nose. The bottom row is ICA such that the basis vector elements are independent. From Draper etal, 2003.

ICA applications

Training

- ICA is often accomplished using the FastICA package.

Applications

- Finding independent causes in time series data.
- Often used in audio processing.
- Commonly also used to find the components of natural images.
- Finding independent ‘purchase directions’ in basket-item data.

`ICA audio unmixing`

Summary

Dimension reduction looks for low dimensional structure in data and can be used to compress, interpret and predict missing values. It is often also used a preprocessing step to remove irrelevant ‘noise’ in the data.

PCA

- Most common approach 🟢
- Fast to use – efficient scalable algorithms exist for sparse data 🟢
- Versions exist for missing data 🟢
- Hard to interpret the components 🔴

NNMF

- Constrained PCA with the basis vectors and their weights all positive.
- Can be more interpretable than PCA 🟢
- Interesting applications in text/topic modelling (and elsewhere) 🟢

State of the Art

Deep Learning

- Also called Neural Nets – we will discuss more in another section
- Can give very good results 🟢
- Can be hard to train 🔴

Tensor Factorisation

- Not really tensors (in the differential geometry sense) but multi-dimensional arrays
- E.g decompose $X_{ijk} \approx \sum_{mn} A_{im} B_{mjn} C_{nk}$

Topic Modelling

- Latent Dirichlet Allocation is a probabilistic model that is popular in topic modelling.
- LDA does not really account well for the typical word distribution (Zipf's law) that is common of English language texts. More recent approaches attempt to address this.