Statistical Machine Learning

Christian Walder

Machine Learning Research Group
CSIRO Data61

and

College of Engineering and Computer Science
The Australian National University

Canberra
Semester One, 2020.

(Many figures from C. M. Bishop, "Pattern Recognition and Machine Learning")

Outlines

Part IX

Principal Component Analysis

Motivation: Pre-training Deep Neural Networks

Empirical observations - pre 2006:

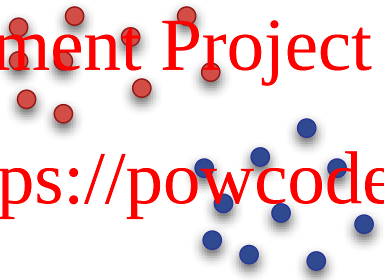
- Deep architectures get stuck in local minima or plateaus
- As architecture gets deeper, more difficult to obtain good generalisation
- Hard to initialise random weights well
- 1 or 2 hidden layers seem to perform better
- 2006: Unsupervised pre-training of each layer; deeper models possible
  - Usually based on auto-encoders (tomorrow's lecture)
  - Similar in spirit to PCA (today's lecture)

Motivation: Exploratory Analysis

Given a dataset of numerical features:

- Low dimensional data may be easy to plot
- High dimensional data is challenging
- Dimensionality reduction (e.g. PCA)
  - Try to explain with fewer dimensions
  - Enables visualisation
  - The new basis may yield insights
  - Aside: can simplify/speed up subsequent analysis e.g. regression

Supervised Learning



- Given are pairs of data $\mathrm{x_i} \in \mathcal{X}$ and targets $t_i \in \mathcal{T}$ in the form $(\mathrm{x_i}, t_i)$, where $i = 1 \dots N$.
- Learn a mapping between the data X and the target $\mathbf{t}$ which generalises well to new data.

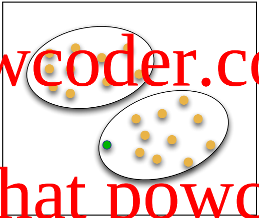
Unsupervised Learning

- Given only the data $\mathrm{x}_i \in \mathcal{X}$.
- Discover (=learn) some interesting structure inherent in the data X.

Unsupervised Learning

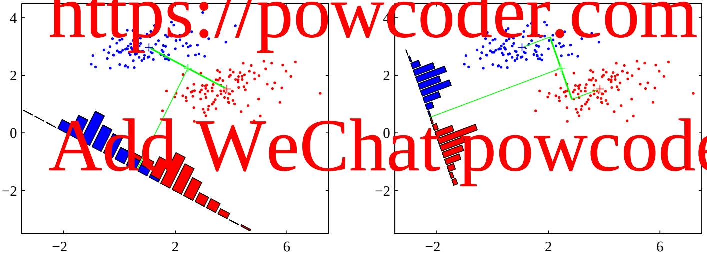
- Given only the data $x_i \in \mathcal{X}$.
- Discover (=learn) some interesting structure inherent in the data.

Testing - Supervised versus Unsupervised Learning



Recall: Fisher's Linear Discriminant

Samples from two classes in a two-dimensional input space and their histogram when projected to two different one-dimensional spaces.



Eigenvectors

- Every square matrix $A \in \mathbb{R}^{n \times n}$ has an Eigenvector decomposition

$$Ax = \lambda x$$

where $x \in \mathbb{R}^n$ and $\lambda \in \mathbb{C}$.

- Example:

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} x = \lambda x$$

$$\lambda = \{-\imath, \imath\}$$

$$x = \left\{ \begin{bmatrix} \imath \\ 1 \end{bmatrix}, \begin{bmatrix} -\imath \\ 1 \end{bmatrix} \right\}$$

Eigenvectors

- How many eigenvalue/eigenvector pairs?
- $$Ax = \lambda x$$

  is equivalent to

  $$(A - \lambda I)x = 0$$
- Has only non-trivial solution for $\det\{A - \lambda I\} = 0$
- polynom of nth order; at most n distinct solutions

Real Eigenvalues

- How can we enforce real eigenvalues?
- Let's look at matrices with complex entries $A \in \mathbb{C}^{n \times n}$.
- Transposition is replaced by Hermitian adjoint, e.g.

$$\begin{bmatrix} 1 + \imath 2 & 3 + \imath 4 \\ 5 + \imath 6 & 7 + \imath 8 \end{bmatrix}^H = \begin{bmatrix} 1 - \imath 2 & 5 - \imath 6 \\ 3 - \imath 4 & 7 - \imath 8 \end{bmatrix}$$

- Denote the complex conjugate of a complex number λ by $\overline{\lambda}$.

Real Eigenvalues

- How can we enforce real eigenvalues?
- Let's assume $A \in \mathbb{C}^{n \times n}$, Hermitian ($A^H = A$).
- Calculate

$$x^H A x = \lambda x^H x$$

for an eigenvector $x \in \mathbb{C}^n$ of A.

- Another possibility to calculate $x^H A x$

$$\begin{aligned} x^H A x &= x^H A^H x && \text{(}A\text{ is Hermitian)} \\ &= (x^H A x)^H && \text{(reverse order)} \\ &= (\lambda x^H x)^H && \text{(eigenvalue)} \\ &= \overline{\lambda} x^H x \end{aligned}$$

- and therefore

$$\lambda = \overline{\lambda} \quad (\lambda \text{ is real}).$$

- If A is Hermitian, then all eigenvalues are real.
- Special case: If A has only real entries and is symmetric, then all eigenvalues are real.

Singular Value Decomposition

Every matrix $A \in \mathbb{R}^{n \times p}$ can be decomposed into a product of three matrices

$$A = U \Sigma V^T$$

where $U \in \mathbb{R}^{n \times n}$ and $V \in \mathbb{R}^{p \times p}$ are orthogonal matrices ($U^T U = I$ and $V^T V = I$), and $\Sigma \in \mathbb{R}^{n \times p}$ has nonnegative numbers on the diagonal.

Linear Curve Fitting - Least Squares

$$N = 10$$

$$\mathbf{x} \equiv (x_1, \dots, x_N)^T$$

$$\mathbf{t} \equiv (t_1, \dots, t_N)^T$$

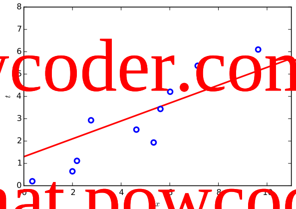
$$x_i \in \mathbb{R} \quad i = 1,\dots, N$$

$$t_i \in \mathbb{R} \quad i = 1,\dots, N$$

$$y(x, \mathbf{w}) = w_1 x + w_0$$

$$X \equiv [\mathbf{x} \quad 1]$$

$$w^* = (X^T X)^{-1} X^T \mathbf{t}$$



Inverse of a matrix

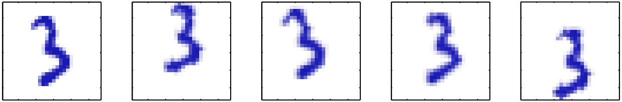
- Assume a full rank symmetric real matrix A.
- Then $A = U^T \Lambda U$ where
- Λ is a diagonal matrix with real eigenvalues
- U contains the eigenvectors

$$
\begin{aligned}
A^{-1} &= (U^T \Lambda U)^{-1} \\
&= U^{-1} \Lambda^{-1} U^{-T} && \text{inverse changes order} \\
&= U^T \Lambda^{-1} U && U^T U = I
\end{aligned}
$$

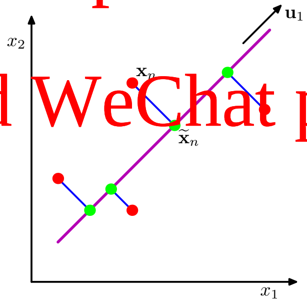
- The inverse of a diagonal matrix is the inverse of its elements.

Dimensionality Reduction

- Main goal of Principal Component Analysis: dimensionality reduction
- Many applications in visualisation, feature extraction, signal processing, data compression . . .
- Example: Use hand-written digits (binary data) and place them into a larger frame (100 $\times$ 100) varying the position and the rotation angle.
- Data space size $= 10\,000$.
- But data live on a three-dimensional manifold (x, y, and the rotation angle).
- FYI only: this manifold is not linear and requires bleeding edge models like capsule networks (Hinton 2017); still we can locally approximate with PCA.

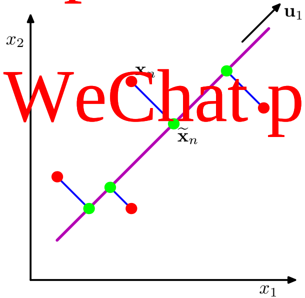


- Idea: Linearly project the data points onto a lower dimensional subspace such that
  - the variance of the projected data is maximised, or
  - the distortion error from the projection is minimised.
- Both formulation lead to the same result.
- Need to find the lower dimensional subspace, called the principal subspace.



Principal Component Analysis (PCA)

- Given N observations $\mathbf{x}_n \in \mathbb{R}^D$, $n = 1, \dots, N$.
- Project onto a space with dimensionality $M < D$ while maximising the variance.
- More advanced : How to calculate M from the data. Therefore here: M is fixed.
- Consider a 1-dimensional subspace spanned by some unit vector $\mathbf{u}_1 \in \mathbb{R}^D$, $\mathbf{u}_1^T \mathbf{u}_1 = 1$.



PCA - Maximise Variance

- Each data point $\mathbf{x}_n$ is then projected onto a scalar value $\mathbf{u}_1^T\mathbf{x}_n$.
- The mean of the projected data is $\mathbf{u}_1^T\bar{\mathbf{x}}$ where $\bar{\mathbf{x}}$ is the sample mean

$$\bar{\mathbf{x}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n.$$

- The variance of the projected data is then

$$\frac{1}{N}\sum_{n=1}^{N}\left\{\mathbf{u}_1^T\mathbf{x}_n - \mathbf{u}_1^T\bar{\mathbf{x}}\right\}^2 = \mathbf{u}_1^T\mathbf{S}\mathbf{u}_1$$

with the covariance matrix

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

- Maximising $\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1$ under the constraint $\mathbf{u}_1^T \mathbf{u}_1 = 1$ (why do we need to bound $\mathbf{u}_1$?) leads to the Lagrange equation

- Maximising $\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1$ under the constraint $\mathbf{u}_1^T \mathbf{u}_1 = 1$ (why do we need to bound $\mathbf{u}_1$?) leads to the Lagrange equation

$$L(\mathbf{u}_1, \lambda_1) = \mathbf{u}_1^T \mathbf{S} \mathbf{u}_1 + \lambda_1 (1 - \mathbf{u}_1^T \mathbf{u}_1)$$

which has a stationary point

PCA - Maximise Variance

- Maximising $\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1$ under the constraint $\mathbf{u}_1^T \mathbf{u}_1 = 1$ (why do we need to bound $\mathbf{u}_1$?) leads to the Lagrange equation

$$L(\mathbf{u}_1, \lambda_1) = \mathbf{u}_1^T \mathbf{S} \mathbf{u}_1 + \lambda_1 (1 - \mathbf{u}_1^T \mathbf{u}_1)$$

which has a stationary point if $\mathbf{u}_1$ is an eigenvector of $\mathbf{S}$ with eigenvalue λ_1.

$$\mathbf{S} \mathbf{u}_1 = \lambda_1 \mathbf{u}_1.$$

- The variance is then $\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1 = \lambda_1$.
- Variance is maximised if $\mathbf{u}_1$ is the eigenvector of the covariance $\mathbf{S}$ with the largest eigenvalue.

PCA - Maximise Variance

- Continue maximising the variance amongst all possible directions orthogonal to those already considered.
- The optimal linear projection onto a M-dimensional space for which the variance is maximised is defined by the M eigenvectors $\mathbf{u}_1, \dots, \mathbf{u}_M$ of the covariance matrix $\mathbf{S}$ corresponding to the M largest eigenvalues $\lambda_1, \dots, \lambda_M$.
- Is this subspace always uniquely defined?
- Not if $\lambda_M = \lambda_{M+1}$.

PCA - Minimise Distortion Error

- The distortion between data points $\mathbf{x}_n$ and their projection $\widetilde{\mathbf{x}}_n$

$$J = \frac{1}{N} \sum_{n=1}^{N} \|\mathbf{x}_n - \widetilde{\mathbf{x}}_n\|^2$$

  is minimised if the variance is maximised.

- The distortion error is then

$$J = \sum_{i=M+1}^{D} \lambda_i$$

  where $\lambda_i, i = M+1, \dots, D$ are the smallest eigenvalues of the covariance matrix $\mathbf{S}$.

- In signal processing we speak of the signal space (principal subspace) and the noise space (orthogonal to the principal subspace).

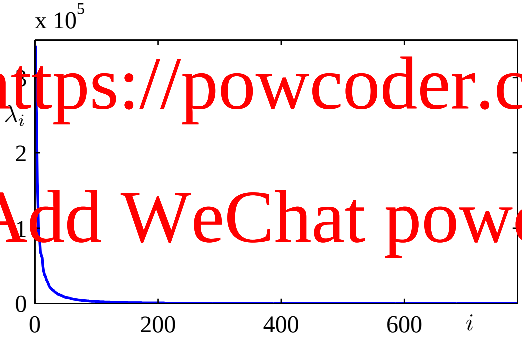
PCA - Applications

- The eigenvectors of the covariance matrix are elements of the original vector space $u_i \in \mathbb{R}^D$.
- If the input data are images, the eigenvectors are also images.

The mean and the first four eigenvectors $\mathbf{u}_1, \dots, \mathbf{u}_4$ of a set of handwritten digits of ‘three’.
Blue corresponds to positive values, white is zero and yellow corresponds to negative values.

PCA - Applications

- The eigenvalues of the covariance matrix express the variance of the data set in the direction of the corresponding eigenvectors.



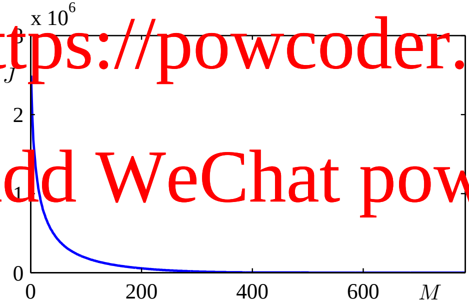
(a)

Plot of the eigenvalue spectrum for the digits of three data set.

PCA - Applications

- The sum of the eigenvalues of the covariance matrix of the discarded directions express the distortion error.

$$J = \frac{1}{N} \sum_{n=1}^{N} \|\mathbf{x}_n - \tilde{\mathbf{x}}_n\|^2$$



(b)

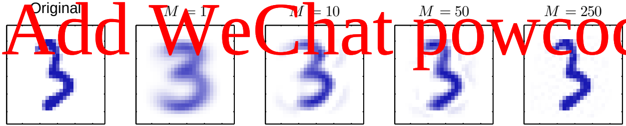
Plot of the distortion error versus the number of dimension of the subspace considered for projection.

PCA - Compression

- The approximated data vector $\widetilde{\mathbf{x}}_n$ can be written in the form

$$\widetilde{\mathbf{x}}_n = \bar{\mathbf{x}} + \sum_{i=1}^{M} \left(\mathbf{u}_i^T (\mathbf{x}_n - \bar{\mathbf{x}}) \right) \mathbf{u}_i$$

- Codebook : $M + 1$ vectors of dimension D ($\bar{\mathbf{x}}$ and $\mathbf{u}_i$).
- Compressed $\mathbf{x}_n$: M factors $\mathbf{u}_i^T (\mathbf{x}_n - \bar{\mathbf{x}})$



Reconstruction of an image retaining M principal components.

PCA - Data Preprocessing

- Standardise certain features of a data set (for instance as a preprocessing step to subsequent algorithms expecting these features).
- Usually, individual standardisation: each variable (dimension) has zero mean and unit variance. But variables are still correlated.
- PCA can do more: create decorrelated data (covariance is the identity, also called whitening or sphering of the data)
- Write the eigenvector equation for the covariance matrix $\mathbf{S}$

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

where $\mathbf{L}$ is the diagonal matrix of (positive!) eigenvalues.
- Transform the original data by

$$\mathbf{y}_n = \mathbf{L}^{-1/2}\,\mathbf{U}^T(\mathbf{x}_n - \bar{\mathbf{x}})$$

- The set $\{\mathbf{y}_n\}$ has mean zero and covariance given by the identity.

PCA - Data Preprocessing

- Transform the original data by

$$\mathbf{y}_n = \mathbf{L}^{-1/2}\,\mathbf{U}^T(\mathbf{x}_n - \bar{\mathbf{x}})$$

- Mean of the set $\{\mathbf{y}_n\}$

$$\frac{1}{N}\sum_{n=1}^{N}\mathbf{y}_n = \frac{1}{N}\sum_{n=1}^{N}\mathbf{L}^{-1/2}\,\mathbf{U}^T(\mathbf{x}_n - \bar{\mathbf{x}})$$

$$= \mathbf{L}^{-1/2}\,\mathbf{U}^T\frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n - \bar{\mathbf{x}}) = 0$$

- Covariance of the set $\{\mathbf{y}_n\}$

$$\frac{1}{N}\sum_{n=1}^{N}\mathbf{y}_n\mathbf{y}_n^T = \frac{1}{N}\sum_{n=1}^{N}\mathbf{L}^{-1/2}\,\mathbf{U}^T(\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^T\mathbf{U}\mathbf{L}^{-1/2}$$

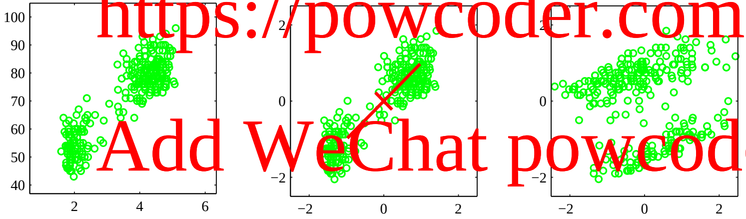
$$= \mathbf{L}^{-1/2}\,\mathbf{U}^T\mathbf{S}\mathbf{U}\mathbf{L}^{-1/2}$$

$$= \mathbf{L}^{-1/2}\,\mathbf{U}^T\mathbf{U}\mathbf{L}\mathbf{L}^{-1/2}$$

$$= \mathbf{I}$$

PCA - The Effect of Whitening

- Compare standardising and whitening of a data set.
- (b) also shows the principal axis of the normalised data set plotted as red lines over the range $\pm\lambda_i^{1/2}$.



Original data (note the different axis).

Standardising to zero mean and unit variance.

Whitening to achieve unit covariance.

- Kernel PCA
    - Use $\Phi(x)$ as features, and express in terms of kernel matrix $\mathbf{K}$
    - The covariance matrix $\mathbf{S}$ and the (centered) kernel matrix $\mathbf{K}$ has the same eigenvalues.

- Probabilistic PCA
    - Explicitly model latent variable $\mathbf{z} \sim \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I})$.
    - Mean value of observed variable is given by $\mathbf{W}\mathbf{z} + \mu$
    - Conditional distribution of observed variable

$$\mathbf{x} \sim \mathcal{N}\left(\mathbf{x}|\mathbf{W}\mathbf{z} + \mu, \sigma^2\mathbf{I}\right)$$

Independence versus Uncorrelatedness

- Independence

$$p(x_1, x_2) = p(x_1)\, p(x_2)$$

- Uncorrelatedness (defined via a zero covariance)

$$\mathbb{E}\left[x_1 x_2\right] - \mathbb{E}\left[x_1\right]\, \mathbb{E}\left[x_2\right] = 0$$

- Independence implies Uncorrelatedness (prove it!).
- BUT Uncorrelatedness does NOT imply Independence.
- Example: Draw the pair (x_1, x_2) with equal probability from the set $\{(0, 1), (0, -1), (1, 0), (-1, 0)\}$.
- Then x_1 and x_2 are uncorrelated because $\mathbb{E}\left[x_1\right] = \mathbb{E}\left[x_2\right] = \mathbb{E}\left[x_1 x_2\right] = 0$.
- But x_1 and x_2 are NOT independent

$$p(x_1 = 0, x_2 = -1) = \frac{1}{4}$$

$$p(x_1 = 0)\, p(x_2 = -1) = \frac{1}{2} \times \frac{1}{4}$$

Independent Component Analysis - Overview

- Assume we have K signals and K recordings, each recording containing a mixture of the signals.
- 'Cocktail party' problem : K people speak at the same time in a room, and K microphones pickup a mixture of what they say.
- Given unknown source signals $S \in \mathbb{R}^{N \times K}$ and an unknown mixing matrix $\mathbf{A}$, producing the observed data $X \in \mathbb{R}^{N \times K}$

$$X = S\mathbf{A}$$

- Can we recover the original signals (Blind Source Separation)?
- Yes, under the assumption that
  - at most one of the signals is Gaussian distributed.
  - we don't care for the amplitude (including the sign).
  - we don't care for the order of the recovered signals.
  - we have at least as many observed mixtures as signals, the matrix $\mathbf{A}$ has full rank and can be inverted.

Independent Component Analysis - Overview

- Uncorrelated variables are not necessarily independent.
- ICA maximises the statistical independence of the estimated components.
- Find A in such a way that the columns of

$$S = XA^{-1}$$

  are maximally independent.
- Several definitions for statistical independence possible.

- Central Limit Theorem: The distribution of a sum of independent random variables tends toward a Gaussian distribution (under certain conditions).
- FastICA algorithm.