

Principal Components Analysis

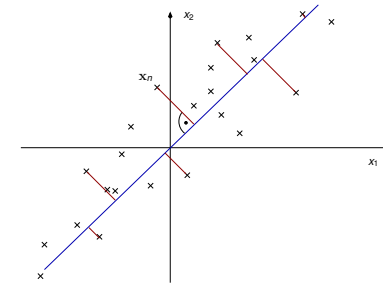
- ▶ Learning the linear manifold is known as Principal Components Analysis (PCA).
- ▶ There are many equivalent learning criteria leading to PCA.
- ▶ Two of the most well-known are
 1. find the subspace in which the projection of the training data has *maximal variance*
 2. maximize the *average distance* between the projections and the original points.

◀ ▶ ⏪ ⏩ 🔍 ↺

Roland Memisevic

Machine learning for vision

Principal Components Analysis



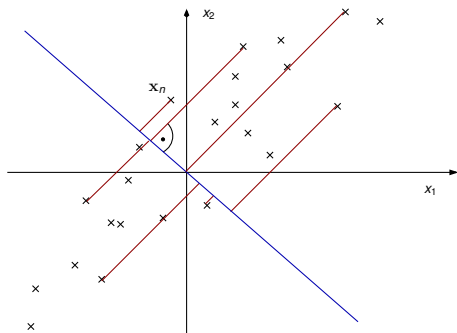
- ▶ The variance along the manifold is **large**.
- ▶ The average projection error is **small**.

◀ ▶ ⏪ ⏩ 🔍 ↺

Roland Memisevic

Machine learning for vision

Principal Components Analysis



- ▶ The variance along the manifold is **small**.
- ▶ The average projection error is **large**.

◀ ▶ ⏪ ⏩ 🔍 ↺

Roland Memisevic

Machine learning for vision

Principal Components Analysis

- ▶ To learn a *subspace* means we need to work under the assumption that the data is *mean-centered*:

$$\frac{1}{N} \sum_{n=1}^N \mathbf{x}_n = \mathbf{0}$$

- ▶ To derive PCA, we define an *orthonormal basis* for the subspace, consisting of vectors

$$\mathbf{u}_1, \dots, \mathbf{u}_M$$

where M is smaller than the dimensionality of the data.

- ▶ PCA amounts to *learning* this basis.

◀ ▶ ⏪ ⏩ 🔍 ↺

Roland Memisevic

Machine learning for vision

Principal Components Analysis

- ▶ It is convenient to stack the basis-vectors column-wise in matrix \mathbf{U} .
- ▶ By assuming we already learned the optimal basis, we can write the forward and backward mappings as:

Projecting data (backward mapping)

- ▶ The optimal coefficients that approximate \mathbf{x} within the subspace are given by

$$\mathbf{z} = \mathbf{U}^T \mathbf{x}$$

Reconstructing data (forward mapping)

- ▶ The approximation $\tilde{\mathbf{x}}$ of \mathbf{x} is given by

$$\tilde{\mathbf{x}} = \mathbf{U}\mathbf{z} \quad (= \mathbf{U}\mathbf{U}^T \mathbf{x})$$

Principal Components Analysis

Optimizing quadratic forms

- ▶ The maximizer of

$$\text{Tr}(\mathbf{U}^T \mathbf{A} \mathbf{U})$$

subject to

$$\mathbf{U}^T \mathbf{U} = \mathbf{I}$$

(where \mathbf{U} is $D \times M$) is given by the matrix whose columns are the eigenvectors of \mathbf{A} corresponding to the M largest eigenvalues.

- ▶ So to find principal components perform an **eigen-decomposition of the data covariance matrix**.

Principal Components Analysis

- ▶ One way to learn the subspace: minimize the reconstruction error

$$E(\mathbf{U}) = \sum_n \|\mathbf{x}_n - \mathbf{U}\mathbf{U}^T \mathbf{x}_n\|^2$$

under the constraint $\mathbf{U}^T \mathbf{U} = \mathbf{I}$

- ▶ To solve the problem, we stack the data row-wise in matrix \mathbf{X} and rewrite the objective function as a *quadratic form* in \mathbf{U} :

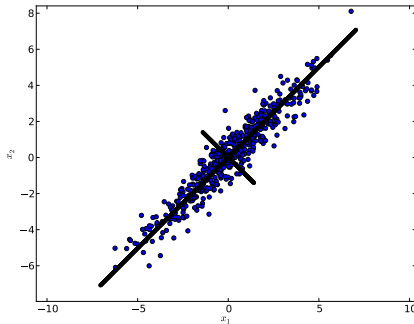
$$\begin{aligned} E(\mathbf{U}) &= \|\mathbf{X}^T - \mathbf{U}\mathbf{U}^T \mathbf{X}^T\|_F^2 \\ &= \text{Tr}((\mathbf{X}^T - \mathbf{U}\mathbf{U}^T \mathbf{X}^T)^T (\mathbf{X}^T - \mathbf{U}\mathbf{U}^T \mathbf{X}^T)) \\ &= \text{Tr}(\mathbf{X}\mathbf{X}^T) - \text{Tr}(\mathbf{U}^T \mathbf{X}^T \mathbf{X} \mathbf{U}) \\ &= -\text{Tr}(\mathbf{U}^T \mathbf{X}^T \mathbf{X} \mathbf{U}) + \text{const} \end{aligned}$$

Principal Components Analysis

Summary: Computing principal components

1. Mean-center the data.
2. Compute the covariance matrix $\mathbf{C} = \frac{1}{N} \mathbf{X}^T \mathbf{X}$.
3. Perform an eigen-decomposition of \mathbf{C} .
4. Sort the eigen-vectors according to the size of their eigenvalues.
5. Stack the leading M eigen-vectors in matrix \mathbf{U} .

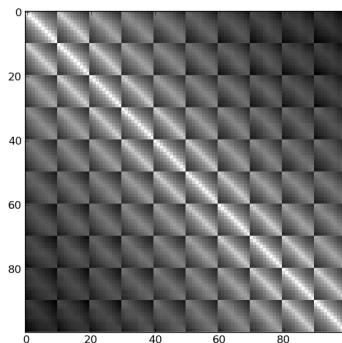
Principal Components Analysis



- ▶ A two-dimensional dataset and the two principal components.
- ▶ Projections onto the leading eigenvectors preserve most of the variability in the data. So PCA performs *lossy compression*.

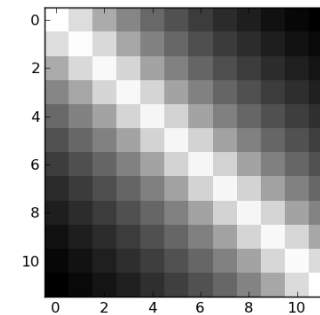
Navigation icons: back, forward, search, etc.

Translation invariance 2d



Navigation icons: back, forward, search, etc.

Translation invariance 1d



- ▶ The covariance between natural image pixels does not depend (much) on their absolute position.

Navigation icons: back, forward, search, etc.

PCA and Fourier analysis 1d

- ▶ A (covariance) matrix whose entries are translation invariant has phasors as eigenvectors:
- ▶ The dimension t of the projected phasor (ie., covariance matrix times phasor) takes the form:

$$\begin{aligned}
 & \sum_{t'} \text{cov}(t, t') e^{i\omega t'} \\
 &= \sum_{t'} c(t - t') e^{i\omega t'} \\
 &= \sum_z c(z) e^{i\omega t} e^{-i\omega z} \\
 &= \left[\sum_z c(z) e^{-i\omega z} \right] e^{i\omega t} =: \lambda_\omega e^{i\omega t}
 \end{aligned}$$

- ▶ In fact, this shows that multiplying by the covariance matrix is a convolution.

Navigation icons: back, forward, search, etc.

PCA and Whitening

- ▶ The components of the features, \mathbf{Z} , are uncorrelated (that is, \mathbf{Z} has a diagonal covariance matrix):

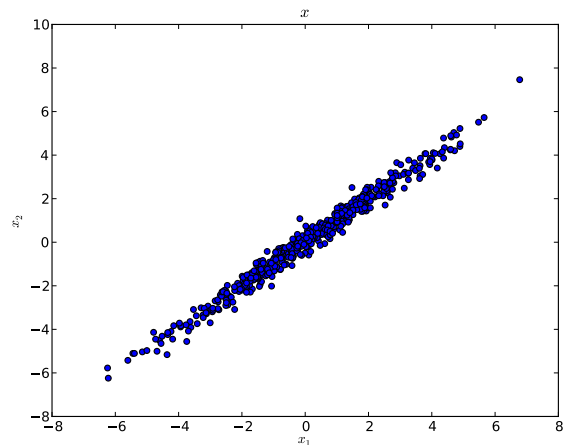
$$\begin{aligned}\frac{1}{N} \sum_n \mathbf{z}_n \mathbf{z}_n^T &= \frac{1}{N} \sum_n \mathbf{U}^T \mathbf{x}_n \mathbf{x}_n^T \mathbf{U} \\ &= \mathbf{U}^T \left(\frac{1}{N} \sum_n \mathbf{x}_n \mathbf{x}_n^T \right) \mathbf{U} \\ &= \mathbf{U}^T \mathbf{C} \mathbf{U} \\ &= \mathbf{L}\end{aligned}$$

where the diagonal matrix \mathbf{L} contains the eigenvalues of \mathbf{C} on its diagonal.

- ▶ (The last step follows from the eigenvalue definition: $\mathbf{C} \mathbf{u}_i = \lambda_i \mathbf{u}_i$)

Navigation icons

Whitening example



Navigation icons

PCA and Whitening

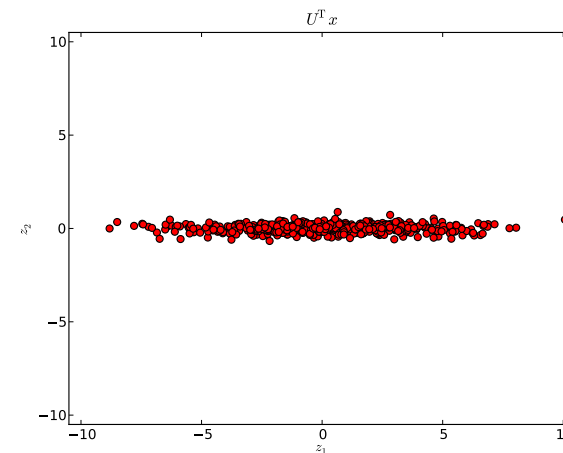
- ▶ We can obtain the *identity* as the covariance matrix for \mathbf{Z} , if instead of \mathbf{U}^T we use the following backward mapping:

$$\mathbf{V} = \mathbf{L}^{-\frac{1}{2}} \mathbf{U}^T$$

- ▶ Data with identity covariance matrix is known as **white**; multiplying data by \mathbf{V} as **whitening**.
- ▶ Whitening may be performed without reducing the dimensionality.
- ▶ This amounts to just rotating the coordinate system of the data, followed by independently “stretching” or “squeezing” the dimensions to obtain unit variance in each.

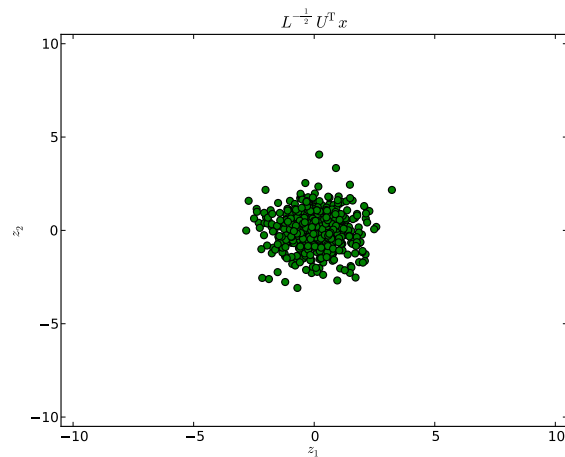
Navigation icons

Whitening example

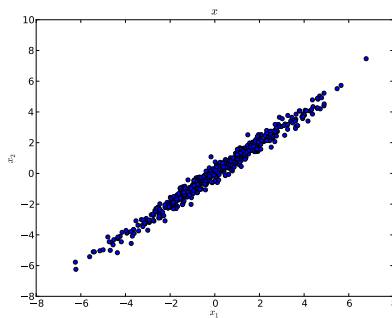


Navigation icons

Whitening example



Original data



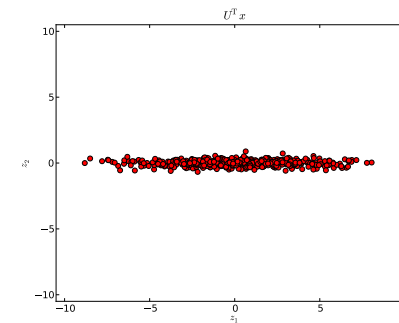
ZCA Whitening

- ▶ Multiplying whitened data by an orthonormal matrix leaves the data white (exercise).
- ▶ Thus, the whitening matrix $V = L^{-\frac{1}{2}} U^T$ is not the only whitening matrix. Any matrix AV with orthonormal A will be one.
- ▶ One way to define a canonical whitening matrix is to choose the *symmetric* one. To get the symmetric whitening matrix W pre-multiply V with the orthonormal matrix U :

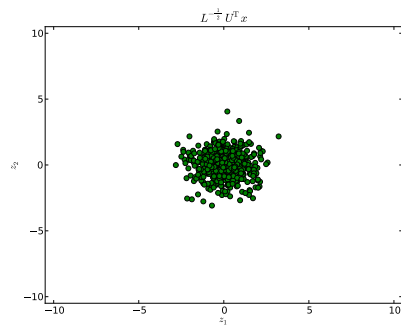
$$W := UV = UL^{-\frac{1}{2}}U^T$$

- ▶ Transforming data with this matrix is known as ZCA (zero-phase components analysis).

PCA projection



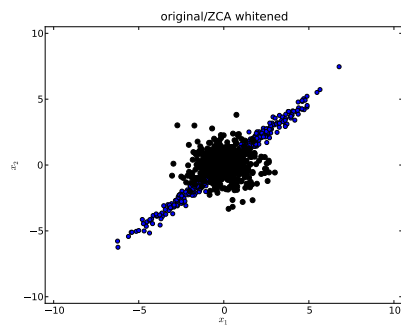
PCA whitened data



Roland Memisevic

Machine learning for vision

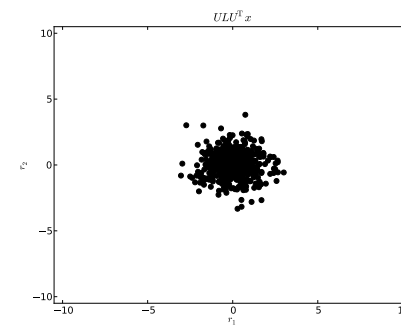
ZCA whitened data and original data



Roland Memisevic

Machine learning for vision

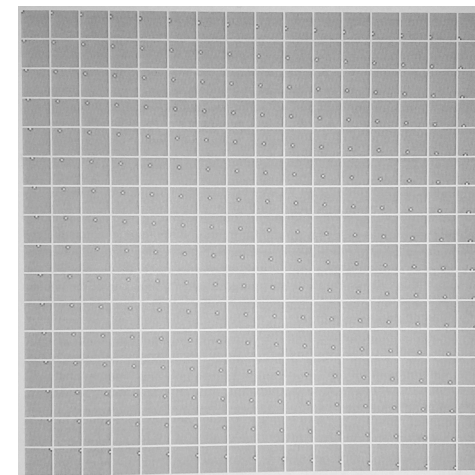
ZCA whitened data



Roland Memisevic

Machine learning for vision

ZCA example (all columns of W)



from: Natural Image Statistics (Hyvarinen, Hurri, Hoyer; 2009)

Roland Memisevic

Machine learning for vision

De-whitening

- ▶ How to get the original images back from the whitened images?

De-whitening

- ▶ How to get the original images back from the whitened images?
- ▶ (Pseudo-)invert the whitening matrix:
inverse PCA whitening:

$$\mathbf{x} = \mathbf{V}^{-1} \mathbf{z} = (\mathbf{L}^{\frac{1}{2}} \mathbf{U}^T)^{-1} \mathbf{z} = \mathbf{U} \mathbf{L}^{\frac{1}{2}} \mathbf{z}$$

inverse ZCA whitening:

$$\mathbf{x} = \mathbf{W}^{-1} \mathbf{z} = (\mathbf{U} \mathbf{L}^{-\frac{1}{2}} \mathbf{U}^T)^{-1} \mathbf{z} = \mathbf{U} \mathbf{L}^{\frac{1}{2}} \mathbf{U}^T \mathbf{z}$$

where $\mathbf{L}^{\frac{1}{2}}$ is diagonal with $(\mathbf{L}^{\frac{1}{2}})_{ii} = \sqrt{\lambda_i}$