

Additional (non-required,
more advanced) material on
kernel PCA, computing the
principal components, and ICA

Kernel PCA (Kernel Principal Component Analysis)

Useful when data lies on or near a low d -dimensional linear subspace of the ϕ -space associated with a kernel

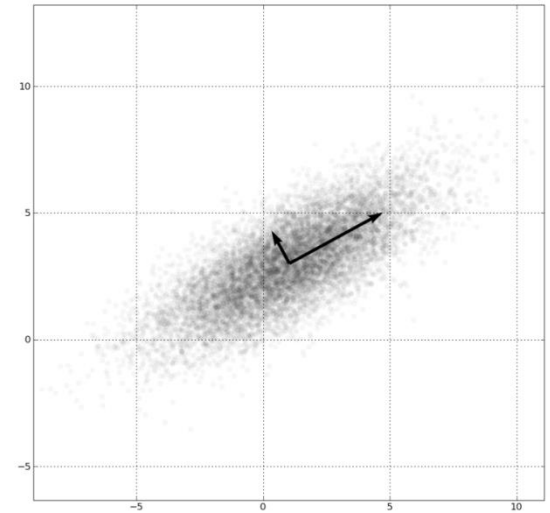
Properties of PCA

- Given a set of n centered observations $x_i \in R^D$, 1st PC is the direction that maximizes the variance

- $X = (x_1, x_2, \dots, x_n)$

- $v_1 = \operatorname{argmax}_{\|v\|=1} \frac{1}{n} \sum_i (v^\top x_i)^2$
 $= \operatorname{argmax}_{\|v\|=1} \frac{1}{n} v^\top X X^\top v$

- Covariance matrix $C = \frac{1}{n} X X^\top$
- v_1 can be found by solving the eigenvalue problem:
 - $C v_1 = \lambda v_1$ (of maximum λ)

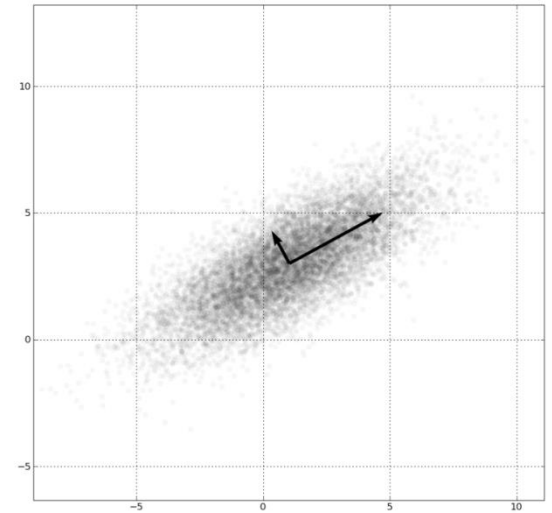


Properties of PCA

- Given a set of n centered observations $x_i \in \mathbb{R}^D$, 1st PC is the direction that maximizes the variance

- $X = (x_1, x_2, \dots, x_n)$

- $$v_1 = \underset{\|v\|=1}{\operatorname{argmax}} \frac{1}{n} \sum_i (v^\top x_i)^2$$
$$= \underset{\|v\|=1}{\operatorname{argmax}} \frac{1}{n} v^\top X X^\top v$$



- Covariance matrix $C = \frac{1}{n} X X^\top$ is a $D \times D$ matrix
the (i,j) entry of $X X^\top$ is the correlation of the i -th coordinate of examples with j th coordinate of examples
- To use kernels, need to use the inner-product matrix $X^\top X$.

Alternative expression for PCA

- The principal component lies in the span of the data

$$v_1 = \sum_i \alpha_k x_i = X\alpha$$

Why? 1st PC is direction of largest variance, and for any direction outside of the span of the data, only get more variance if we project that direction into the span.

- Plug this in we have

$$C v_1 = \frac{1}{n} X X^T X \alpha = \lambda X \alpha$$

- Now, left-multiply the LHS and RHS by X^T .

$$\frac{1}{n} X^T X X^T X \alpha = \lambda X^T X \alpha$$

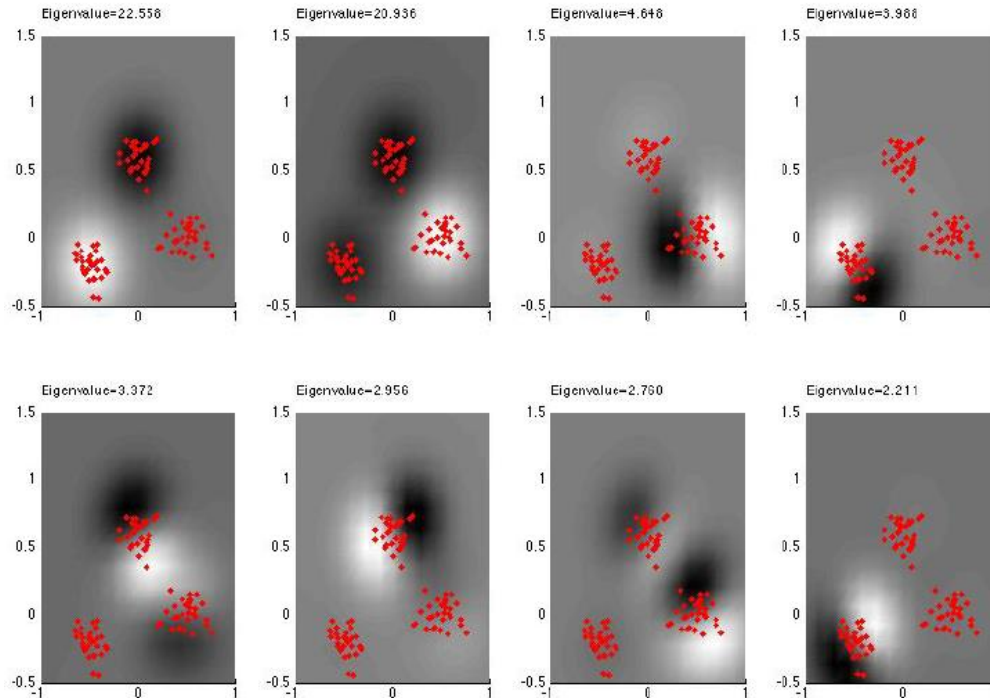
Only depends on
the inner product
matrix

Kernel PCA

- **Key Idea:** Replace inner product matrix by kernel matrix
 - PCA: $\frac{1}{n} X^T X X^T X \alpha = \lambda X^T X \alpha$
 - Let $K = [K(x^i, x^j)]_{ij}$ be the matrix of all dot-products in the ϕ -space.
 - Kernel PCA: replace " $X^T X$ " with K .
 $\frac{1}{n} K K \alpha = \lambda K \alpha$, or equivalently, $\frac{1}{n} K \alpha = \lambda \alpha$
- **Key computation:** form an n by n kernel matrix K , and then perform eigen-decomposition on K .

Kernel PCA Example

- Gaussian RBF kernel $\exp\left(-\frac{\|x-x'\|^2}{2\sigma^2}\right)$ over 2 dimensional space
- Eigenvector evaluated at a test point x is a function
 $w^\top \phi(x) = \sum_i \alpha_i \langle \phi(x^i), \phi(x) \rangle = \sum_i \alpha_i k(x^i, x)$



What You Should Know

- Principal Component Analysis (PCA)
 - What PCA is, what is useful for.
 - Both the maximum variance subspace and the minimum reconstruction error viewpoint.
- Kernel PCA

Power method for computing PCs

Given matrix $X \in R^{D \times n}$, compute the top eigenvector of XX^T

Initialize with random $\hat{v} \in R^D$

Repeat

$$\hat{v} \leftarrow XX^T \hat{v}$$

$$\hat{v} \leftarrow \hat{v} / \|\hat{v}\|$$

Claim

For any $\epsilon > 0$, whp over choice of initial vector, after $O\left(\frac{1}{\epsilon} \log \frac{d}{\epsilon}\right)$ iterations, we have $\hat{v}^T XX^T \hat{v} \geq (1 - \epsilon)\lambda_1$.

Then can subtract the $\hat{v}$ component off of each example and repeat to get the next.

Eigendecomposition

Any symmetric matrix $A = XX^T$ is guaranteed to have an eigendecomposition with real eigenvalues: $A = V \Lambda V^T$.

$$\begin{array}{c} \boxed{} \\ A \\ (D \times D) \end{array} = \begin{array}{c} \boxed{} \\ V \\ (D \times D) \end{array} \begin{array}{c} \boxed{\begin{array}{ccc} \lambda_1 & & 0 \\ & \lambda_2 & \\ 0 & & \lambda_3 \\ & & \dots \end{array}} \\ \Lambda \\ (D \times D) \end{array} \begin{array}{c} \boxed{} \\ V^T \\ (D \times D) \end{array} = \sum_i \lambda_i v_i v_i^T$$

Matrix Λ is diagonal with eigenvalues $\lambda_1 \geq \lambda_2 \geq \dots$ on the diagonal. Matrix V has the eigenvectors as the columns.

Singular Value Decomposition (SVD)

Eigendecomposition of XX^T is closely related to SVD of X .

Given a matrix $X \in \mathbb{R}^{D \times n}$, the SVD is a decomposition: $X^T = USV^T$

$$\begin{array}{c} \boxed{} \\ X^T \\ (n \times D) \end{array} = \begin{array}{c} \boxed{} \\ U \\ (n \times d) \end{array} \begin{array}{c} \boxed{\begin{array}{cc} \sigma_1 & 0 \\ 0 & \sigma_2 & \dots \end{array}} \\ S \\ (d \times d) \end{array} \begin{array}{c} \boxed{} \\ V^T \\ (d \times D) \end{array} = \sum_i \sigma_i u_i v_i^T$$

- S is a diagonal matrix with the singular values $\sigma_1, \dots, \sigma_d$ of X .
- Columns of U, V are orthogonal, unit length.
- So, $XX^T = VSU^TUSV^T = VS^2V^T =$ eigendecomposition of XX^T .

So, $\lambda_i = \sigma_i^2$ and can read off the solution from the SVD.

Singular Value Decomposition (SVD)

Eigendecomposition of XX^T is closely related to SVD of X .

Given a matrix $X \in \mathbb{R}^{D \times n}$, the SVD is a decomposition: $X^T = USV^T$

$$\begin{array}{c} \boxed{} \\ X^T \\ (n \times D) \end{array} = \begin{array}{c} \boxed{} \\ U \\ (n \times d) \end{array} \begin{array}{c} \boxed{\begin{array}{cc} \sigma_1 & 0 \\ 0 & \sigma_2 & \dots \end{array}} \\ S \\ (d \times d) \end{array} \begin{array}{c} \boxed{} \\ V^T \\ (d \times D) \end{array} = \sum_i \sigma_i u_i v_i^T$$

- In fact, can view the rows of US as the coordinates of each example along the axes given by the d eigenvectors.

So, $\lambda_i = \sigma_i^2$ and can read off the solution from the SVD.

Independent Component Analysis (ICA)

Find a linear transformation

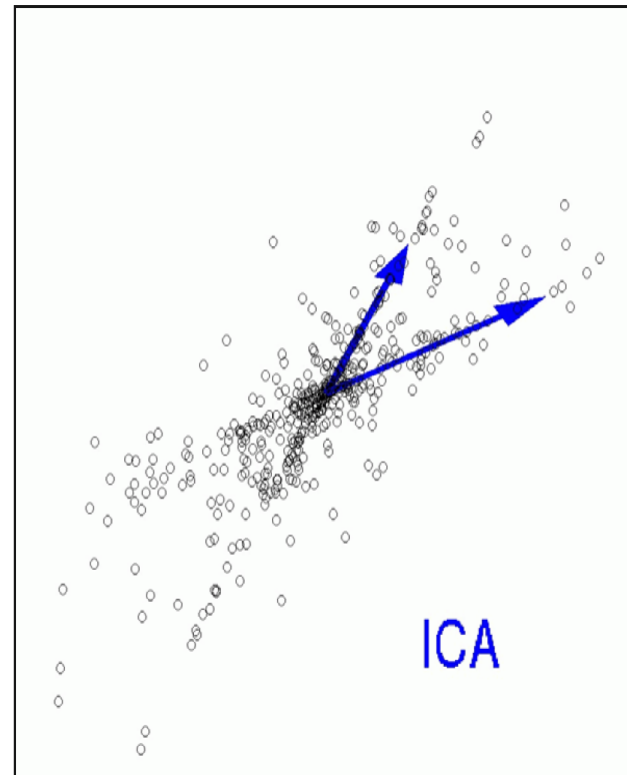
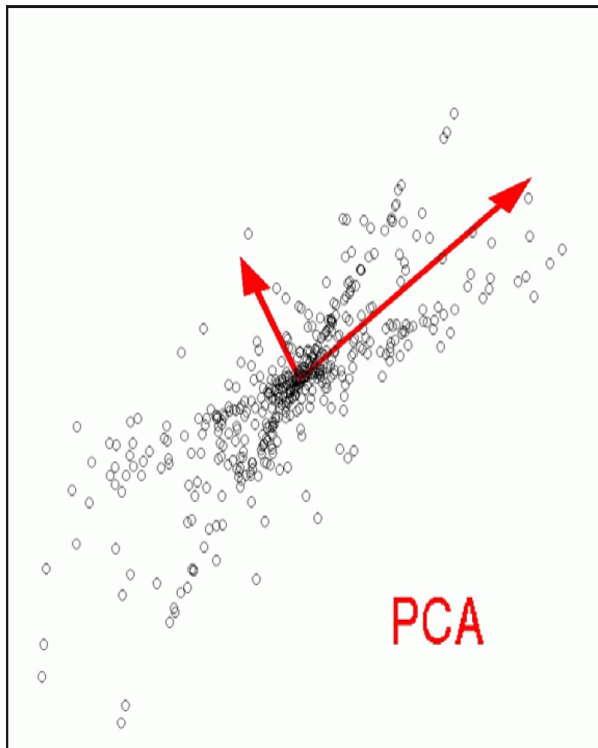
$$\mathbf{x} = \mathbf{V} \cdot \mathbf{s}$$

for which coefficients $\mathbf{s} = (s_1, s_2, \dots, s_D)^T$ are **statistically independent**

$$p(s_1, s_2, \dots, s_D) = p_1(s_1)p_2(s_2) \dots p_n(s_D)$$

Algorithmically, we need to identify matrix $\mathbf{V}$ and coefficients $\mathbf{s}$, s.t. under the condition $\mathbf{x} = \mathbf{V}^T \cdot \mathbf{s}$ the **mutual information** between $s_1, s_2, \dots, s_D$ is minimized:

$$I(s_1, s_2, \dots, s_D) = \sum_{i=1}^D H(s_i) - H(s_1, s_2, \dots, s_D)$$



PCA finds directions of maximum variation,
ICA would find directions most "aligned" with data.