Mathematically

- Two formulations
  - Maximum variance: find the directions that maximize the variance of the projected data
  - Minimum-error formulation: minimizes the reconstruction error of the projected data

Mathematically

- Consider a set of observations $\{x_n\}, n = 1, \ldots, N$ with x_n a vector of dimension D.
- We want to find the unit direction u_1 that maximizes the variance of the projection:

$$\arg\max_{u_1} \frac{1}{N} \sum_{n=1}^{N} ||u_1^T x_n - u_1^T \bar{x}||^2 = u_1^T C u_1$$

with $$||u_1|| = u_1^T u_1 = 1$$

$$C = \frac{1}{N} \sum_{n=1}^{N} (x_n - \bar{x})(x_n - \bar{x})^T$$

Mathematically

- Introducing lagrange multiplier:

$$u_1^T C u_1 + \lambda_1 (1 - u_1^T u_1)$$

- Setting the derivative with respect to u_1 equal to zero:

$$C u_1 = \lambda_1 u_1$$

$\Rightarrow$ u_1 must be an eigenvector of C .

- The variance is given by:

$$u_1^T C u_1 = \lambda_1$$

$\Rightarrow$ u_1 is the eigenvector corresponding to the highest eigenvalue λ_1

Mathematically

- The (M+1)th component is obtained by maximizing:

$$u_{M+1}^T C u_{M+1}$$

With the constraints
$$u_{M+1}^T u_{M+1} = 1$$
$$u_{M+1}^T u_i = 0 \quad \forall i = 1, \ldots, M+1$$

- Using lagragian multiplier:

$$u_{M+1}^T C u_{M+1} + \lambda_{M+1}(1 - u_{M+1}^T u_{M+1}) + \sum_{i=1}^{M} \eta_i u_{M+1}^T u_i$$

- At the optimum:

$$0 = 2Cu_{M+1} - 2\lambda_{M+1} u_{M+1} + \sum_{i=1}^{M} \eta_i u_i$$

- Multiplying by u_i^T at the left, one gets $\eta_i = 0$ and thus

$$Cu_{M+1} = \lambda_{M+1} u_{M+1}$$

$\Rightarrow u_{M+1}^T$ is the eigenvector with M+1 largest eigenvalue

Mathematically

- The *i*th principal component for objects x_j is computed by $x'_{ji} = u_i^T x_j$
- The reconstructed input is thus:

$$\hat{x}_j = \sum_{i=1}^{M} x'_{ji} u_i = \sum_{i=1}^{M} (u_i^T x_j) u_i$$

- PCA also minimizes the reconstruction error:

$$\arg \max_{u_1,\ldots,u_M} \frac{1}{N} \sum_{i=1}^{N} ||x_j - \hat{x}_j||^2$$

Algorithm 1

Recover basis: Calculate $XX^\top = \sum_{i=1}^{t} x_i x_i^\top$ and let U = eigenvectors of $XX^\top$ corresponding to the top d eigenvalues.

Encode training data: $Y = U^\top X$ where Y is a $d \times t$ matrix of encodings of the original data.

Reconstruct training data: $\hat{X} = UY = UU^\top X$.

Encode test example: $y = U^\top x$ where y is a d-dimensional encoding of x.

Reconstruct test example: $\hat{x} = Uy = UU^\top x$.

Table 1.1: *Direct PCA Algorithm*