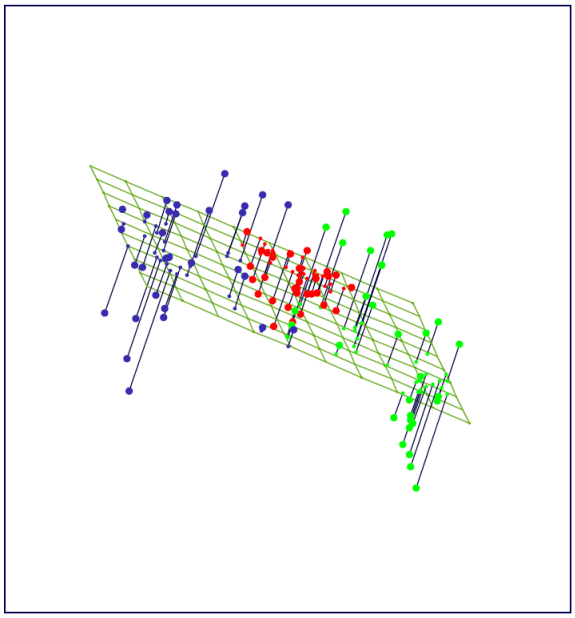
Principal Components Analysis

Principal Components Analysis (PCA) is a standard technique for **dimensionality reduction** of data sets. It is a way to automatically find a **subspace** which approximates the data. It is used everywhere in signal processing, machine learning, and statistics.

There are actually two equivalent ways to think about PCA. The first is statistical: we are trying to find a transform that is carefully tuned to the (second-order) statistics of the data. The second perspective, which is what we will adopt in this course, is more geometrical: given a set of vectors, we are trying to find a subspace of a certain dimension that comes closest to containing this set.

Specifically, suppose that we have data points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^D$, and want to find the K-dimensional affine space (subspace plus offset) that comes closest to containing them. Here is a picture[1]



[1]From Ch. 14 of Tibshirani and Hastie's *Elements of Statistical Learning*.

Our goal is to find an offset $\boldsymbol{\mu} \in \mathbb{R}^D$ and a matrix $\boldsymbol{Q}$ with orthonormal columns such that

$$\boldsymbol{x}_n \approx \boldsymbol{\mu} + \boldsymbol{Q}\boldsymbol{\theta}_n \quad \text{for all} \;\; n = 1, \ldots, N,$$

for some $\boldsymbol{\theta}_n \in \mathbb{R}^K$. We cast this as the following optimization problem. Given $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, solve

$$\underset{\boldsymbol{\mu}, \boldsymbol{Q}, \{\boldsymbol{\theta}_n\}}{\text{minimize}} \;\; \sum_{n=1}^{N} \|\boldsymbol{x}_n - \boldsymbol{\mu} - \boldsymbol{Q}\boldsymbol{\theta}_n\|_2^2 \quad \text{subject to} \quad \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Q} = \mathbf{I}.$$

Note that if we fix $\boldsymbol{\mu}$ and define $\widetilde{\boldsymbol{x}}_n = \boldsymbol{x}_n - \mu$, then we can recast the optimization with respect to $\boldsymbol{Q}$ and the $\boldsymbol{\theta}_n$ as

$$\underset{\boldsymbol{Q}, \{\boldsymbol{\theta}_n\}}{\text{minimize}} \;\; \sum_{n=1}^{N} \|\widetilde{\boldsymbol{x}}_n - \boldsymbol{Q}\boldsymbol{\theta}_n\|_2^2 \quad \text{subject to} \quad \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Q} = \mathbf{I}.$$

If $\widetilde{\boldsymbol{X}}$ and $\boldsymbol{\Theta}$ denote the matrices whose columns are given by $\widetilde{\boldsymbol{x}}_1, \ldots, \widetilde{\boldsymbol{x}}_n$ and $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n$ respectively, then we can also write this as

$$\underset{\substack{\boldsymbol{Q}:D\times K \\ \boldsymbol{\Theta}:K\times N}}{\text{minimize}} \;\; \|\widetilde{\boldsymbol{X}} - \boldsymbol{Q}\boldsymbol{\Theta}\|_F^2 \quad \text{subject to} \quad \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Q} = \mathbf{I}.$$

This is exactly the optimization problem that we looked at previously in our Subspace Approximation Lemma! Thus the solution is given by computing the SVD of $\widetilde{\boldsymbol{X}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}$ and then taking as our solution

$$\begin{aligned} \widehat{\boldsymbol{Q}} &= \boldsymbol{U}_K, \\ \widehat{\boldsymbol{\Theta}} &= \boldsymbol{U}_K^{\mathrm{T}}\widetilde{\boldsymbol{X}}, \end{aligned}$$

where $\boldsymbol{U}_K = \begin{bmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 & \cdots & \boldsymbol{u}_K \end{bmatrix}$ contains the first K columns of $\boldsymbol{U}$.

Finally, let us return to the question of how to set $\boldsymbol{\mu}$. For any given $\boldsymbol{\mu}$, the solution for $\boldsymbol{Q}$ and $\boldsymbol{\Theta}$ is given by the Subspace Approximation Lemma. This results in setting

$$\boldsymbol{\theta}_n = \boldsymbol{Q}^{\mathrm{T}}(\boldsymbol{x}_n - \boldsymbol{\mu}).$$

Plugging this in for $\boldsymbol{\theta}_n$ in our objective function, we have that

$$\begin{aligned}\boldsymbol{x}_n - \boldsymbol{\mu} - \boldsymbol{Q}\boldsymbol{\theta}_n &= \boldsymbol{x}_n - \boldsymbol{\mu} - \boldsymbol{Q}\boldsymbol{Q}^{\mathrm{T}}(\boldsymbol{x}_n - \boldsymbol{\mu}) \\ &= (\mathbf{I} - \boldsymbol{Q}\boldsymbol{Q}^{\mathrm{T}})(\boldsymbol{x}_n - \boldsymbol{\mu}).\end{aligned}$$

Hence, the problem of selecting $\boldsymbol{\mu}$ reduces to the optimization problem

$$\underset{\boldsymbol{\mu}}{\text{minimize}} \ \sum_{n=1}^{N} \|(\mathbf{I} - \boldsymbol{Q}\boldsymbol{Q}^{\mathrm{T}})(\boldsymbol{x}_n - \boldsymbol{\mu})\|_2^2$$

The vector $\boldsymbol{\mu}$ is unconstrained; we can solve for the optimal $\boldsymbol{\mu}$ by taking a gradient and setting it equal to zero. To make this easier, note that

$$\|(\mathbf{I} - \boldsymbol{Q}\boldsymbol{Q}^{\mathrm{T}})(\boldsymbol{x}_n - \boldsymbol{\mu})\|_2^2 = (\boldsymbol{x}_n - \boldsymbol{\mu})^T(\mathbf{I} - \boldsymbol{Q}\boldsymbol{Q}^{\mathrm{T}})(\boldsymbol{x}_n - \boldsymbol{\mu})$$

by simply expanding out the norm squared as an inner product and then using the fact that $\mathbf{I} - \boldsymbol{Q}\boldsymbol{Q}^{\mathrm{T}}$ is a projector, i.e., it is symmetric and $(\mathbf{I} - \boldsymbol{Q}\boldsymbol{Q}^{\mathrm{T}})^2 = \mathbf{I} - \boldsymbol{Q}\boldsymbol{Q}^{\mathrm{T}}$. Thus, by taking a gradient and setting it equal to zero we have

$$\begin{aligned}\mathbf{0} &= -2\sum_{n=1}^{N}(\mathbf{I} - \boldsymbol{Q}\boldsymbol{Q}^{\mathrm{T}})(\boldsymbol{x}_n - \boldsymbol{\mu}) \\ &= -2(\mathbf{I} - \boldsymbol{Q}\boldsymbol{Q}^{\mathrm{T}})\left(\left(\sum_{n=1}^{N}\boldsymbol{x}_n\right) - N\boldsymbol{\mu}\right).\end{aligned}$$

We can satisfy this condition by taking the offset $\boldsymbol{\mu}$ to be the sample mean (average of all the observed vectors):

$$\widehat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}_n.$$

Note that this choice is not unique – any choice of $\boldsymbol{\mu}$ that results in $\sum(\boldsymbol{x}_n - \boldsymbol{\mu})$ living in the nullspace of $\mathbf{I} - \boldsymbol{Q}\boldsymbol{Q}^{\mathrm{T}}$ would also suffice – but $\boldsymbol{\mu}$ is the easy and obvious choice, and also what is usually done in practice, because it makes computing the solution to the PCA problem straightforward.

Computing the PCA solution

Specifically, in practice you would typically proceed by first computing the mean $\widehat{\boldsymbol{\mu}}$ of your data as described above. Given $\widehat{\boldsymbol{\mu}}$, you can then form the matrix $\widetilde{\boldsymbol{X}}$ whose columns are given by

$$\widetilde{\boldsymbol{x}}_n = \boldsymbol{x}_n - \widehat{\boldsymbol{\mu}}.$$

Alternatively, if you know a priori that your columns of zero mean (or should have zero mean) based on the underlying process generating the data, then you can skip this step, setting $\widetilde{\boldsymbol{X}} = \boldsymbol{X}$.

In either case, once you have formed $\widetilde{\boldsymbol{X}}$, you simply compute the SVD of $\widetilde{\boldsymbol{X}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}$ and then set

$$\widehat{\boldsymbol{Q}} = \boldsymbol{U}_K,$$
$$\widehat{\boldsymbol{\theta}}_n = \boldsymbol{U}_K^{\mathrm{T}}\widetilde{\boldsymbol{x}}_n,$$

where $\boldsymbol{U}_K = \begin{bmatrix}\boldsymbol{u}_1 & \boldsymbol{u}_2 & \cdots & \boldsymbol{u}_K\end{bmatrix}$ contains the first K columns of $\boldsymbol{U}$. We can think of $\widehat{\boldsymbol{\theta}}_n$ as a representation of $\boldsymbol{x}_n$ is a K-dimensional

subspace, with $\widehat{\boldsymbol{Q}}$ giving us a basis for that subspace (which is useful for projecting vectors $\boldsymbol{x} \in \mathbb{R}^N$ into the subspace).

Note that if you look up a discussion of PCA in most textbooks or online, you will typically see a slightly different presentation. Specifically, most texts describe an approach to the problem that involves forming the matrix

$$\boldsymbol{S} = \sum_{n=1}^{N} (\boldsymbol{x}_n - \widehat{\boldsymbol{\mu}})(\boldsymbol{x}_n - \widehat{\boldsymbol{\mu}})^{\mathrm{T}},$$

taking and eigenvalue decomposition $\boldsymbol{S} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{\mathrm{T}}$, and then taking

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_K \end{bmatrix},$$

where $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_K$ are the eigenvectors of $\boldsymbol{S}$ corresponding to the K largest eigenvalues.

This approach is **completely equivalent** to our approach above.[2] The reason that PCA is typically presented in this was is that $\boldsymbol{S}$ can be interpreted as a scaled version of an empirical estimate of the covariance matrix for the underlying distribution generating the data. While this provides a nice connection with the other (statistical) interpretation of PCA, I personally find the SVD approach more intuitive. In PCA, we are simply trying to find a low-rank approximation to our dataset, which is directly and optimally handled by computing a truncated SVD.

[2] Recall the relationship between the SVD of $\widetilde{\boldsymbol{X}}$ and the eigendecomposition of $\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{X}}^{\mathrm{T}}$.