Principal Component Analysis (PCA)

One way to deal with the curse of dimensionality is to project data down onto a space of low dimensions.

There are a number of different techniques for doing this, e.g., multidimensional scaling. Too many to deal with in this course.

Now we discuss the most basic method — Principal Component Analysis (PCA)
CONVENTION: $\mathbf{M}^T \mathbf{M}$ is a scalar $\mathbf{M}_1^2 + \mathbf{M}_2^2 + \ldots + \mathbf{M}_b^2$
$\mathbf{M} \mathbf{M}^T$ is a matrix $(\mathbf{M}_1^2 \mathbf{M}_2^2 \mathbf{M}_1 \mathbf{M}_3 \ldots)$.

Data samples $X_1, \ldots, X_N$

Compute the mean $\mu = \frac{1}{N} \sum_{i=1}^{N} X_i$ in $D$-dim. space.

Compute the covariance:

$\Sigma = \frac{1}{N} \sum_{i=1}^{N} (X_i - \mu) (X_i - \mu)^T$.

Next compute the eigenvalues and eigenvectors of $\Sigma$.

Solve $\Sigma \mathbf{e} = \lambda \mathbf{e}$.

$\lambda_1 > \lambda_2 > \ldots > \lambda_N$

Note: $\Sigma$ is symmetric, so eigenvalues are real, eigenvectors are orthogonal.

PCA reduces the dimension by PCA reduces the dimension by

by keeping the eigenvectors $\mathbf{e}_i$ with $\lambda_i > \tau$.

Let $M$ eigenvectors be kept. Let $M$ eigenvectors be kept. $\tau$ threshold

Then project data $X$ onto the subspace spanned by the first $M$ eigenvectors. Then project data $X$ onto the subspace spanned by the first $M$ eigenvectors. (After subtracting out the mean.)
(3) Formally:

\[
\text{Project: } \quad \mathbf{x} - \mathbf{\mu} = \sum_{v=1}^{D} a_v \mathbf{e}_v
\]

where the coefficients are given by

\[
a_v = (\mathbf{x} - \mathbf{\mu}) \cdot \mathbf{e}_v \quad \text{(Orthogonality, mean)}
\]

\[
\mathbf{e}_v \cdot \mathbf{e}_\mu = \delta_{v\mu}
\]

Kronecker delta

Here

\[
\mathbf{x} = \mathbf{\mu} + \sum_{v=1}^{D} (\mathbf{x} - \mathbf{\mu}) \cdot \mathbf{e}_v \mathbf{e}_v
\]

no dimension reduction
(no compression)

Then, approximate

\[
\mathbf{x} \approx \mathbf{\mu} + \sum_{v=1}^{m} (\mathbf{x} - \mathbf{\mu}) \cdot \mathbf{e}_v \mathbf{e}_v
\]

Projects the data into the \(m\)-dim subspace.

\[
\mathbf{\mu} + \sum_{v=1}^{m} b_v \mathbf{e}_v
\]
In 2-dimensions

Visually

The eigenvectors of $\Sigma$ correspond to the second order moments of the data if the data lies (almost) on a straight line, then $\lambda_1 > 0$, $\lambda_2 = 0$

\[ e_1 \rightarrow e_2 \]

PCA and Gaussian Distribution

PCA is equivalent to performing ML estimation of the parameters of a Gaussian

\[ P(x | \mu, \Sigma) = \frac{1}{\sqrt{2\pi|\Sigma|}} e^{-\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu)} \]

to get $\hat{\mu}, \hat{\Sigma}$. And then throw away the directions where the variance is small.
15) **Cost Function for PCA.**

\[
J(\mathbf{M}, \{a_k\}, \{e_i\}) = \sum_{k=1}^{N} \| \mathbf{M} + \sum_{c=1}^{M} a_k \mathbf{e}_i - \mathbf{x}_k \|^2
\]

Minimize \( J \) w.r.t. \( \mathbf{M}, \{a_k\}, \{e_i\} \)

Data \( \{\mathbf{x}_k : k=1\ldots N\} \).

The \( \{a_k\} \) are projection coefficients.

**Intuition:** find the \( M \)-dimensional subspace s.t. the projections of the data onto this subspace have minimal error.

Minimizing \( J \), gives the \( \{\mathbf{e}_i\} \) to be the eigenvectors of the covariance matrix

\[
\mathbf{K} = \frac{1}{N} \sum_{k=1}^{N} (\mathbf{x}_k - \mathbf{\mu}) (\mathbf{x}_k - \mathbf{\mu})^T
\]

\[
\mathbf{\mu} = \frac{1}{N} \sum_{k=1}^{N} \mathbf{x}_k
\]

\( \hat{a}_k = (\mathbf{x}_k - \mathbf{\mu}) \cdot \mathbf{e}_i \) the projection coefficients.
To understand this fully, you must understand Singular Value Decomposition (SVD).

We can re-express the criteria as

\[ \frac{1}{N} \sum_{b=1}^{D} \sum_{k=1}^{K} \left( (y_{bk} - X_{ak} \phi_{vk}) + \frac{M}{\lambda} a_{ki} e_{ib} \right)^2 \]

where \( b \) denotes the vector components.

This is an example of a general class of problem:

\[ E(\Phi, e) = \sum_{a=1}^{D} \sum_{k=1}^{K} \left( X_{ak} - \sum_{v=1}^{M} \Phi_{av} \phi_{vk} \right)^2 \]

Goal: minimize \( E(\Phi, e) \) w.r.t. \( \Phi, e \).

This is a bilinear problem, that can be solved by SVD.

\[ \tilde{X}_{ak} = X_{ak} - \mu_a \]

Note: \( \tilde{X}_{ak} = X_{ak} - \mu_a \)

the position of the point, relative to the mean.
We can express any \( N \times D \) matrix \( X \) in the form
\[
X = E \cdot D \cdot F
\]
where \( D = (d_{\mu \nu}) \) is a diagonal matrix \( (d_{\mu \nu} = 0, \mu \neq \nu) \)
\[
D = \begin{pmatrix}
\sqrt{\lambda_1} & 0 \\
0 & \sqrt{\lambda_2}
\end{pmatrix}
\]
\( \lambda_i \) are eigenvalues of \( X \cdot X^T \)
\( \sqrt{\lambda_i} \) are eigenvectors of \( X \cdot X^T \)
\( E = \{ e_{\mu} \} \) are eigenvectors of \( X \cdot X^T \)
\( F = \{ f_{\nu} \} \) are eigenvectors of \( X^T \cdot X \)

Note: For \( \overline{X} \) defined on previous page, we get
\[
\overline{X} = \sum_{k=1}^{N} (X_k - \mu) (X_k - \mu)^T
\]

Note: If \( (X \cdot X^T) e = \lambda e \)
then \( (X^T \cdot X) (X^T e) = \lambda (X^T e) \)
This relates the eigenvectors of \( X \cdot X^T \) and \( X^T \cdot X \).
(Calculate the eigenvectors for the smaller matrix first, then deduce those of the bigger matrix - \( D > N \).)
Minimize:
\[ E[\Psi, e] = \sum_{a=1, k=1}^{a=D, k=N} \left( \hat{X}_{ak} - \sum_{n=1}^{m} \Psi_{an} \phi_{nk} \right)^2 \]

we set:
\[
\begin{cases}
\Psi_{an} = \sqrt{d_{an}} \cdot e_{a}
\
\phi_{nk} = \sqrt{d_{nk}} \cdot f_{n}
\end{cases}
\]

Take M biggest terms in the SVD expansion of X.

But there is an ambiguity:
\[ \sum_{n=1}^{M} \Psi_{an} \phi_{nk} = \Phi \Psi \Phi \text{ matrix unambiguous.} \]
\[ \Phi \Psi \Phi \text{ for any MxM invertible matrix } \Phi \]
This gets rid of the ambiguity.

For the PCA problem - we have constraints that the projection directions one or orthogonal unit eigenvectors.
Relate SVD to PCA  (Linear Algebra)

Start with an \( n \times m \) matrix \( X \).

\[ X^T X \] is a symmetric \( n \times n \) matrix

\[ X^T X \] is a symmetric \( m \times m \) matrix.

\( (X^T X)^T = X^T X \)

By standard linear algebra,

\[ X^T X \mathbf{e}_m = \lambda_m \mathbf{e}_m \quad \text{for eigenvalues } \lambda_m \]

eigenvectors are orthogonal \( \mathbf{e}_m \cdot \mathbf{e}_n = 0 \).

Similarly, \( X^T \mathbf{f}_u = \lambda_u \mathbf{f}_u \) for eigenvalues \( \lambda_u \).

The \( \{ \mathbf{e}_m \} \) and \( \{ \mathbf{f}_u \} \) are related because

\[ (X^T X) (X^T \mathbf{e}_m) = \lambda_m (X^T \mathbf{e}_m) \]

\[ (X^T X) (X^T \mathbf{f}_u) = \lambda_u (X^T \mathbf{f}_u) \]

Here:

\[ X^T \mathbf{e}_m \propto \mathbf{f}_u \]

\[ X^T \mathbf{f}_u \propto \mathbf{e}_m \]

\[ \lambda_m = \lambda_u \]

If \( n > m \), then there are \( n \) eigenvectors \( \{ \mathbf{e}_m \} \) and \( m \) eigenvector \( \{ \mathbf{f}_u \} \). So some \( \mathbf{f}_u \) relate to several \( \{ \mathbf{e}_m \} \).
Claim: we can express
\[
X = \sum \frac{q^\mu}{\mu} e^\mu f^\mu \quad \text{for some } q^\mu
\]
\[
X^T = \sum \frac{q^\mu}{\mu} e^\mu f^\mu \quad \text{(we will solve } q^\mu \text{ for } \mu \text{ later.)}
\]

Verify the claim
\[
X f^\mu = \sum \frac{q^\mu}{\mu} e^\mu f^\mu f^\mu = \sum 2^\mu \delta_{\mu
u} e^\mu = 2^\nu e^\nu.
\]
\[
X^T X = \sum \frac{2^\nu}{\nu} e^\nu f^\mu f^\mu f^\nu = \sum \frac{2^\nu}{\nu} \delta_{\mu \nu} e^\mu e^\mu e^\nu.
\]
\[
X^T X = \sum \frac{2^\nu}{\nu} \delta_{\mu \nu} e^\mu e^\mu e^\nu = \sum \left(\frac{2^\mu}{\mu}\right)^2 e^\mu e^\mu e^\nu.
\]
Similarly
\[
X^T X = \sum \frac{2^\nu}{\nu} \delta_{\mu \nu} e^\mu e^\mu e^\nu,
\]
so \(2^\mu = \frac{X^\mu}{X}
\]

(Because we can express a symmetric matrix
in form \(X^\mu X_{\mu \nu} e^\mu e^\nu = \))

\[
X = \sum \frac{q^\mu}{\mu} e^\mu f^\mu \quad \text{is the SVD of } X
\]

In coordinates:
\[
X_{\alpha i} = \sum \frac{q^\mu}{\mu} e^\mu a^\mu f_i
\]
\[
X_{\alpha i} = \sum \frac{q^\mu}{\mu} e^\mu a^\mu \delta_{\mu \nu} f_i
\]
\[
X = E D F^T \quad E_{\alpha \mu} = e^\mu a^\mu, \quad D_{\mu \nu} = 2^\mu \delta_{\mu \nu}
\]
\[
F_{\nu i} = f_i
\]
(11) Effectiveness of PCA.

In practice, PCA is often effective at unnecessary reduction of data dimension. But it will not be effective for some problems.

For example, if the data is a set of strings

\( (1, 0, 0, 0, \ldots, 1) = \mathbf{x}_1 \)
\( (0, 1, 0, 0, \ldots, 1) = \mathbf{x}_2 \)
\( (0, 0, 0, \ldots, 1, 1) = \mathbf{x}_n \)

Thus the eigenvalues do not fall off as PCA requires.