Dimension Reduction

Why reduce the dimensionality:
- The complexity of classifiers usually depends on the number of input dimensions.
- Helps learning and inference.
- Also fewer dimensions helps with knowledge extraction. Also easier to plot graphically.

Two main methods for reducing dimension:
1. Feature Selection (i.e., select k of d features).
2. Feature Extraction (i.e., find a new set of features that are a combination of the original ones).
   e.g., PCA/LDA

Subset selection - classifier + error measure.

There are $2^d$ possible subsets of d-dims.
We can’t test all of them.

Two strategies:
- Forward Selection - start with no features and then add them one by one. At each step select the feature that maximally decreases the error.
- Backward Selection - start with all features and remove them one by one, selecting the one that maximally decreases the error.
Subset Selection (Cont)

F - feature set, input dimensions \( X_i, i = 1, \ldots, d \)

\( E(F) \) error on validation sample when only using
the inputs to \( F \).

Sequential Forward Selection

Start with \( F = \emptyset \).

At each step, for all possible \( X_i \)

- train model and evaluate \( E(F \cup X_i) \)
on validation set.

Then choose \( X_j \) such that \( j = \arg \min_i E(F \cup X_i) \) and add \( X_j \) to \( F \) if \( E(F \cup X_j) < E(F) \).

Stop if adding more features does not decrease \( E \). (Or if decrease is too small)

Process may be costly, to get \( k \) features need to be

\( d + (d+1) + (d-2) + \ldots + (d-k) = O(d^2) \)

It is local search, no guarantee to find the
optimal subset. (e.g. \( X_i, X_j \) may be good together but bad individually.)

Sequential Backward Selection

Start, \( F = \{X_1, \ldots, X_d\} \). At each step \( j = \arg \min_i E(F - X_j) \).

Remove \( X_j \) from \( F \) if \( E(F - X_j) < E(F) \)
"Curse of Dimensionality" Bellman.

Amount of data needed to learn distribution or perform classification, rises with the dimension of the data.

Multivariate Gaussi $\mu, \Sigma \sim d^2$ parameter

needs $\sim 10x d^2$ data examples

$d = \text{dim} \text{ of space.}$

Histogram Distirbution $\uparrow \text{ in d-dim}$

- need $10xM$ in 1-dim
- $10xM^d$ in d-dimensions

Impractical.

Learning is possible in high dimensional spaces only because the real data lies in some low dimensional subspace

$\uparrow \leftrightarrow \text{ subspace}$

PCA $\rightarrow \text{ assumes that data lies on a linear subspace.}$
(4) Principal Component Analysis (PCA)

Seek a projection $z = w_1^T x$, $\omega_1$ first principal component

PCA is unsupervised (we don't use the output)

We seek $\omega_1$ to maximize the variance of projection

$\text{Var}(z) = \frac{\omega_1^T E E^T \omega_1}{\omega_1^T \omega_1} \leq \text{entire } \omega_1 \| = 1$

$\max \omega_1, \quad \frac{\omega_1^T E E^T \omega_1 - \lambda (\omega_1^T \omega_1 - 1)}{\omega_1^T \omega_1}$

Diff w.r.t. $\omega_1$

Lagrangian multiplier

$\frac{\partial}{\partial \omega_1} = \lambda \omega_1$, eigenvalue/eigenvector

$\omega_1^T E E^T \omega_1 = \lambda \omega_1$, choose $\lambda$ to be biggest as possible, biggest eigenvalue

$\omega_1$ second principal component.

$\omega_2^T \omega_1 = 0$ orthogonal.

$\max \omega_2, \quad \frac{\omega_2^T E E^T \omega_2 - \lambda (\omega_2^T \omega_2 - 1) - \beta (\omega_2^T \omega_1 - 0)}{\omega_2^T \omega_2}$

$\omega_2 = \lambda \omega_2$, choose $\lambda$ to be second biggest eigenvalue

$\omega_2$ can be orthogonal.

Note: because $E E^T$ is symmetric, the eigenvector

$m$ removed $z = W (x - \mu)$ is the projection

Columns of $W$ are eigenvectors of $E E^T$. 
Another derivation:

Find \( \mathbf{y} \) s.t. \( \mathbf{z} = \mathbf{y}^T \mathbf{x} \) has \( \text{Cov}(\mathbf{z}) = \mathbf{D}' \),

where \( \mathbf{D}' \) is diagonal.

\[ \mathbf{z} \sim \text{i.i.d.} \]

i.e., get uncorrelated \( \mathbf{z} \).

\[ \mathbf{z} \sim \mathbf{S} \mathbf{\xi} \]

\[ \mathbf{S} = \begin{pmatrix} s_1 & s_2 & \ldots & s_d \end{pmatrix} \]

\[ \mathbf{\xi} \sim \mathbf{S}^{-1} \mathbf{z} \]

\[ \mathbf{S}^{-1} \mathbf{S} = \begin{pmatrix} \lambda_1 & 0 & \ldots & 0 \\ 0 & \lambda_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \lambda_d \end{pmatrix} \]

Spectral Decomposition.

\[ \mathbf{S} = \mathbf{\Sigma} \mathbf{D} \mathbf{\Sigma}^T \]

\( \mathbf{D} \) diagonal matrix.

Diagonal elements are eigenvalues \( \lambda_1, \ldots, \lambda_d \) of \( \mathbf{S} \).

\[ \mathbf{\Sigma}^T \mathbf{S} = \mathbf{D} \]

Now, if \( \mathbf{z} = \mathbf{y}^T \mathbf{x} \), \( \text{Cov}(\mathbf{z}) = \mathbf{y}^T \mathbf{S} \mathbf{S}^T \mathbf{y} \) which we want to be diagonal — so set \( \mathbf{y} = \mathbf{\xi} \).

Even if all eigenvalues are positive, only some make significant contribution.

Proportion of variance explained by \( k \) PCs is

\[ \frac{\lambda_1 + \lambda_2 + \ldots + \lambda_k}{\lambda_1 + \lambda_2 + \ldots + \lambda_d} \]

Scre screening — plot eigenvalues.
In F.A., we assume there is a set of unobservable latent factors $z_j$, $j = 1, \ldots, k$ which act together to generate $x$.

Assume factors are unit normal $E(z_j) = 0$, $\text{Var}(z_j) = 1$

$\text{Cov}(z_i, z_j) = 0$, $i \neq j$.

Also $\epsilon_i$, with $E(\epsilon_i) = 0$ and $\text{Var}(\epsilon_i) = \psi_i$ (unknown)

$\text{Cov}(\epsilon_i, \epsilon_j) = 0$, $i \neq j$, $\text{Cov}(\epsilon_i, z_j) = 0$, $i \neq j$.

FA assumes each input dimension can be expressed

\[ x_i - \mu_i = \sum_{j=1}^{k} \lambda_{ij} z_j + \epsilon_i \]

\[ x - \mu = \Lambda z + \epsilon \]

\[ \Lambda = \begin{pmatrix} \lambda_{11} & \cdots & \lambda_{1k} \\ \vdots & \ddots & \vdots \\ \lambda_{k1} & \cdots & \lambda_{kk} \end{pmatrix} \]

\[ \text{Var}(x_i) = \sum_{j=1}^{k} \lambda_{ij}^2 + \psi_i \]

\[ \sum_{j=1}^{k} \lambda_{ij}^2 = \text{Cov}(z_i, x_i) = \text{Cov}(z_i, \sum_{j=1}^{k} \lambda_{ij} z_j + \epsilon_i) = \sum_{j=1}^{k} \lambda_{ij}^2 + \psi_i \]
(7) **Factor Analysis (cont)**

How to find the factor loadings and the specific variances?

By spectral decomposition:

\[ \Sigma = \Sigma \Lambda \Lambda^T \]

Thus \( \Lambda = \Sigma \Lambda^{1/2} \),

we find \( \psi_j \) by \( \psi_i = \sigma^2 - \sum_{j=1}^{K} \psi_j^2 \).

Two uses of factor analysis:

1) Knowledge extraction when we find the loadings and try to express the variables with fewer factors.

2) It can also reduce dimensionality when \( p < d \).

**Dimension Reduction**

\[ \Sigma = \Lambda \Lambda^T \Rightarrow \Sigma^{-1} \Lambda^T \Lambda \Sigma^{-1} \]
Fisher's Linear Discriminant Analysis

Find projection to perform classifier

\[ z = \omega^T x \quad \text{sample} \quad \chi = \{ x^i, r^i \} \]

Projected mean of two classes

\[ m_1 = \frac{1}{Z} \sum_{r^i} \omega^T x^i r^i = \omega^T m_1 \]

\[ m_2 = \frac{1}{Z} \sum_{(1-r^i)} \omega^T x^i (1-r^i) = \omega^T m_2 \]

Scatter

\[ S_1 = \frac{1}{Z} \sum_{r^i} (\omega^T x^i - m_1)^2 r^i \]

\[ S_2 = \frac{1}{Z} \sum_{(1-r^i)} (\omega^T x^i - m_2)^2 (1-r^i) \]

Fisher's Criterion

\[ J(\omega) = \frac{(m_1 - m_2)^2}{S_1 + S_2} \]

Goal: find projection to maximize the between class scatter \((m_1 - m_2)^2\) and to minimize the within class scatter \(S_1 + S_2\)
(4) 

**Fisher's LDA.**

Re-express \( J(\omega) = \omega^T S_\omega \omega \) \[ \frac{\omega^T S_\omega \omega}{\omega^T S S \omega} \] as \( S_\omega = S_1 + S_2 \) with \( S_1 = \sum_t v_t (x_t - \mu_1)(x_t - \mu_1)^T \) \( S_2 = \sum_t (1 - v_t) (x_t - \mu_2)(x_t - \mu_2)^T \) \( S_{\beta} = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T \).

Maximize \( J(\omega) \) is equivalent to log-likelihood

\[
\min \quad \omega^T S_\omega \omega - 2 \left( \omega^T S_{\beta} \omega - 1 \right)
\]

\( S_\omega \omega = \lambda S_{\beta} \omega \)

generalized eigenvalue equation.

Solve to get \( \omega^* = S_{\beta}^{-1} (\mu_1 - \mu_2) \) (since

Why Fisher is better than PCA

\( \mu_1 - \mu_2 \)

\( \mu_1 - \mu_2 \)

PCA: favoring the wrong projection (in this case)

\( S_{\beta} \) projection

Fisher can be extended to multiple classes (see book).