CSI 436/536
Introduction to Machine Learning

Dimension reduction and total LLSE

Professor Siwei Lyu
Computer Science
University at Albany, State University of New York
Linear least squares

- Fitting a line (linear model) by minimizing prediction error
- The error is on the y-axis only
Total least squares estimation

• Fitting a line (linear model) by minimizing the total error
  • The error is for both x- and y- coordinates
TLSE as encoder-decoder

- The TLSE model can be understood with an encoder-decoder model

- The encoder takes the input and reduces it to a code
- The decoder takes the code and reconstructs it to an output
- The low dimensional code is the “information bottleneck”
- Learning is achieved through “self-supervision”, i.e., reducing the error between the input and the reconstructed output
The encoder-decoder interpretation of TLSE also suggests that TLSE can be viewed as a data compression procedure.

- Input and output have dimension \( d \)
- The code has dimension \( 1 \)
- Compression
Total least squares

- given m data vector of dimensions \( X = (x_1, \ldots, x_m) \).
- assumption: data are centered, i.e., \( \sum_i x_i = 0 \).
- find the best one-dimensional approximation to \( X \) minimizing \( \ell_2 \) errors.
- specifically, find a unit vector \( v \) (why?), and scaling factors \((s_1, \ldots, s_m)\), s.t.,

\[
\min_{v: \|v\|_2 = 1, s_1, \ldots, s_m} \sum_{i=1}^{m} \|x_i - s_i v\|_2^2.
\]
solution

First, given \( \mathbf{v} \), find optimal solution to \( s_i \).

\[
\frac{\partial}{\partial s_i} \sum_{i=1}^{m} \| \mathbf{x}_i - s_i \mathbf{v} \|^2_2 = \frac{\partial}{\partial s_i} \| \mathbf{x}_i - s_i \mathbf{v} \|^2_2 = 0
\]

\[
\Rightarrow \frac{\partial}{\partial s_i} (\mathbf{x}_i - s_i \mathbf{v})^T (\mathbf{x}_i - s_i \mathbf{v}) = 0
\]

\[
\Rightarrow \frac{\partial}{\partial s_i} (s_i^2 \mathbf{v}^T \mathbf{v} - 2s_i \mathbf{v}^T \mathbf{x}_i + \mathbf{x}_i^T \mathbf{x}_i) = 0
\]

\[
\Rightarrow s_i = \mathbf{x}_i^T \mathbf{v}.
\]

\[
\min_{\mathbf{v}: \|\mathbf{v}\|_2 = 1} \sum_{i=1}^{m} \| \mathbf{x}_i - (\mathbf{x}_i^T \mathbf{v}) \mathbf{v} \|^2_2.
\]
solution (continued)

\[
\sum_{i=1}^{m} \| x_i - (x_i^T v)v \|^2 = \sum_{i=1}^{m} (x_i - (x_i^T v)v)^T (x_i - (x_i^T v)v) \\
= \sum_{i=1}^{m} (x_i^T x_i - 2(x_i^T v)x_i^T v + (x_i^T v)^2 v^T v) \\
= \sum_{i=1}^{m} (x_i^T x_i - (x_i^T v)^2) = \sum_{i=1}^{m} (x_i^T x_i - v^T x_i x_i^T v)
\]
Furthermore

\[
\sum_{i=1}^{m} v^T x_ix_i^T v = v^T \left( \sum_{i=1}^{m} x_ix_i^T \right) v = v^T (XX^T)v.
\]

Recall $XX^T$ is the covariance matrix of data matrix $X$ because $X$ is centered.

Equivalently, in PCA, we seek

\[
\max_v v^T (XX^T)v
\]

s.t. $\|v\|_2^2 - 1 = 0$
Total LLSE

Constrained optimization:

\[
\begin{align*}
\max_v & \quad v^T (XX^T)v \\
\text{s.t.} & \quad v^T v - 1 = 0
\end{align*}
\]

Lagrangian

\[
L(v, \lambda) = v^T (XX^T)v - \lambda(v^T v - 1)
\]

Derivative w.r.t. \( x \) sets to zero

\[
(XX^T)v = \lambda v.
\]
Example

- data
  - $X = \{(1,2),(3,3),(3,5),(5,4),(5,6),(6,5),(8,7),(9,8)\}$

- centering

- covariance

- EVD
  - $\lambda_1 = 9.34$
  - $\lambda_2 = 0.41$
  - $v_1 = [0.81, 0.59]$, $v_2 = [0.81, -0.59]$, 
Dimension reduction

- Total LLSE fits a 1D line to a set of multi-dimensional vectors with minimum distortion
- This can be equivalently viewed as finding a low dimensional approximation (in this case 1D) of a high-dimensional data point
- The procedure is known as dimension reduction, and it is behind image compression algorithms
- We will talk about the more general version of dimension reduction known as principal component analysis (PCA) later