Dimensionality Reduction

Ziping Zhao

School of Information Science and Technology
ShanghaiTech University, Shanghai, China

CS182: Introduction to Machine Learning (Fall 2022)
http://cs182.sist.shanghaitech.edu.cn

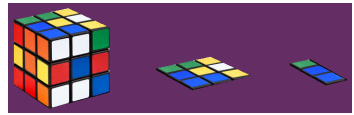
Ch. 6 of I2ML (Secs. 6.4, 6.6, and 6.12 – 6.13 excluded)

Outline

Outline

Why Dimensionality/Dimension Reduction?

- Whether for classification or regression problem, observation data that we believe are informative are taken as inputs and fed to the system for decision making.



- The number of inputs (input dimensionality of the feature) often affects the time and space complexity of the learning algorithm (either classifier or regressor):
  - Having less computation reduces time complexity.
  - Having fewer parameters reduces space complexity.
- Eliminating an input deemed unnecessary saves the cost of extracting/observing it.
- Simpler models are often more robust on small data sets.
- Simpler models are more interpretable, leading to simpler explanation.
- Data visualization in 2 or 3 dimensions facilitates the detection of structure and outliers.

Feature Selection vs. Extraction

- Two main methods for reducing dimensionality: feature selection and feature extraction.
- Feature selection:
  - Choosing $k < d$ important features and discarding the remaining $d - k$.
  - Subset selection algorithms (supervised methods)
- Feature extraction:
  - Projecting the original d dimensions to k ($< d$) new dimensions.
  - Unsupervised methods (without using output information):
    - Principal component analysis (PCA)
    - Factor analysis (FA)
    - Multidimensional scaling (MDS)
    - Canonical correlation analysis (CCA)
  - Supervised methods (using output information):
    - Linear discriminant analysis (LDA)
  - The linear methods above also have nonlinear extensions.
- These k features may be interpreted as hidden or latent factors that in combination generate the observed d features.

Outline

Subset Selection

- Goal: find the best subset of features.
- The best subset contains the least number of dimensions that most contribute to accuracy with respect to a certain task (e.g., classication, regression, visualization).
- There are $2^d - 1$ nonempty subsets of d features.
- Unless d is small, the search space is typically huge, making it impossible to conduct an exhaustive search for the best subset.
- Heuristic algorithms are often used to obtain reasonable (suboptimal) solutions in reasonable (polynomial) time.
- Two conventional approaches:
  - Forward search
  - Backward search
- More recent approach: using sparsity-inducing regularizers such as ℓ_1-norm.

Sequential Forward Search

- Start with no features and add them one by one, at each step adding the one that decreases the error measure the most, until the error cannot be further decreased.
- The error E (e.g., misclassication error for classification, mean squared error for regression) should be measured on a validation set distinct from the training set.
- Algorithm skeleton:
  - Initialize feature set as empty set: $\mathcal{F} = \emptyset$
  - At each iteration:
    - For each available feature x_i , train the model and calculate the error $E(\mathcal{F} \cup \{x_i\})$ incurred on the validation set.
    - Find the best feature x_j: $j = \arg\min_i E(\mathcal{F} \cup \{x_i\})$
    - If $E(\mathcal{F} \cup \{x_j\}) < E(\mathcal{F})$ then add x_j to F and continue; else exit.
- To select k features from d, we need to train and test the model $d + (d-1) + (d-2) + \cdots + (d-k+1)$ times, which is of the order $O(d^2)$.
- No guarantee for optimal subset with greedy search.
- We can add multiple features at a time (requires more computation) or backtrack to check which previously added feature can be removed.

Sequential Backward Search

- Start with all features and do a similar process as forward search except by removing features one at a time.
- Algorithm skeleton:
  - Initialize feature set $\mathcal{F}$ with all features.
  - At each iteration:
    - For each feature $x_i \in \mathcal{F}$, train the model and calculate the error $E(\mathcal{F} \backslash \{x_i\})$ incurred on the validation set.
    - Find the best feature x_j: $j = \arg\min_i E(\mathcal{F} \backslash \{x_i\})$
    - If $E(\mathcal{F} \backslash \{x_i\}) < E(\mathcal{F})$ then remove x_j from $\mathcal{F}$ and continue; else exit.
- We can stop if removing a feature does not decrease the error.
  - For model complexity reduction, we may decide to remove a feature if its removal causes only a slight increase in error. (similar procedure may apply to forward search)
- To select k features from d, we need to train and test the model $d + (d-1) + (d-2) + \cdots + (k+1)$ times.
- Backward search is more computationally demanding than forward search:
  - Usually $k \ll d$
  - Training a model with more features is more costly.

Remarks on Feature Selection

- In applications like face recognition, feature selection is not a good method for dimensionality reduction because individual pixels by themselves do not carry much discriminative information; it is the combination of values of several pixels together that carry information about the face identity.
- Dimensionality reduction in such cases is done by feature extraction methods that we will discuss next.

Outline

Principal Component Analysis

- Projection methods (feature extraction methods) aim to find a linear mapping from the d-dimensional input space ($\mathbf{x}$-space) to a k-dimensional space ($k \ll d$) ($\mathbf{z}$-space) with minimum information loss according to some criterion.
  - scalar projection of $\mathbf{x}$ on the direction of $\mathbf{w}$ s.t. $\|\mathbf{w}\| = 1$:

$$z = \mathbf{w}^T \mathbf{x}$$

- Principal component analysis (PCA) is one of the projection methods.
- The principal component is $\mathbf{w}_1$ such that the sample, after projection on to $\mathbf{w}_1$, is most spread out so that the difference between the sample points becomes most apparent and hence the criterion to be optimized is the variance.
- Finding the first principal component $\mathbf{w}_1$ s.t. the $\mathrm{Var}(z_1)$ is maximized:

$$\begin{aligned}\mathrm{Var}(z_1) &= \mathrm{Var}(\mathbf{w}_1^T \mathbf{x}) = \mathbb{E}[(\mathbf{w}_1^T \mathbf{x} - \mathbb{E}(\mathbf{w}_1^T \mathbf{x}))^2] = \mathbb{E}[(\mathbf{w}_1^T (\mathbf{x} - \boldsymbol{\mu}))^2] \\ &= \mathbb{E}[\mathbf{w}_1^T (\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T \mathbf{w}_1] = \mathbf{w}_1^T \mathbb{E}[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T] \mathbf{w}_1 = \mathbf{w}_1^T \boldsymbol{\Sigma} \mathbf{w}_1\end{aligned}$$

where

$$\boldsymbol{\mu} = \mathbb{E}[\mathbf{x}], \qquad \boldsymbol{\Sigma} = \mathrm{Cov}(\mathbf{x}) = \mathbb{E}[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T]$$

Optimization Problem for First Principal Component

- The optimization problem is given by

$$\begin{aligned} \underset{\mathbf{w}_1}{\text{maximize}} \quad & \mathbf{w}_1^T \mathbf{\Sigma} \mathbf{w}_1 \\ \text{subject to} \quad & \|\mathbf{w}_1\| = 1 \end{aligned}$$

  which is a constrained optimization problem and the Lagrangian is given by

$$\mathcal{L}(\mathbf{w}_1, \alpha) = -\mathbf{w}_1^T \mathbf{\Sigma} \mathbf{w}_1 + \alpha(\mathbf{w}_1^T \mathbf{w}_1 - 1)$$

  where α is the Lagrange multiplier.

- Taking the derivative of the Lagrangian w.r.t. $\mathbf{w}_1$ and setting it to $\mathbf{0}$, we get an eigenvalue equation for the (first) principal component $\mathbf{w}_1$:

$$\mathbf{\Sigma} \mathbf{w}_1^\star = \alpha^\star \mathbf{w}_1^\star$$

- Since

$$(\mathbf{w}_1^\star)^T \mathbf{\Sigma} \mathbf{w}_1^\star = \alpha^\star (\mathbf{w}_1^\star)^T \mathbf{w}_1^\star = \alpha^\star$$

  we choose the eigenvector corresponding to the largest eigenvalue λ_1 for the objective to be maximized.

Alternative Problem Formulation for First Principal Component

- The optimization problem for the first principal component can also be written as

$$\underset{\mathbf{w}_1}{\text{maximize}} \quad \frac{\mathbf{w}_1^T \boldsymbol{\Sigma} \mathbf{w}_1}{\mathbf{w}_1^T \mathbf{w}_1}$$

  which is to maximize the Rayleigh quotient.
- The same solution can be obtained.

Optimization Problem for Second Principal Component

- The second principal component $\mathbf{w}_2$ defines the projection $z_2 = \mathbf{w}_2^T \mathbf{x}$, which should maximize $\text{Var}(z_2) = \mathbf{w}_2^T \mathbf{\Sigma} \mathbf{w}_2$ with z_2 uncorrelated to z_1, i.e.,

$$\text{Cov}(z_1, z_2) = \mathbb{E}[(\mathbf{w}_1^T \mathbf{x} - \mathbf{w}_1^T \boldsymbol{\mu})(\mathbf{w}_2^T \mathbf{x} - \mathbf{w}_2^T \boldsymbol{\mu})] = \mathbf{w}_1^T \mathbf{\Sigma} \mathbf{w}_2 = \lambda_1 \mathbf{w}_2^T \mathbf{w}_1 = 0$$

- The optimization problem is

$$\begin{aligned} \underset{\mathbf{w}_2}{\text{maximize}} \quad & \mathbf{w}_2^T \mathbf{\Sigma} \mathbf{w}_2 \\ \text{subject to} \quad & \|\mathbf{w}_2\| = 1,\ \mathbf{w}_2^T \mathbf{w}_1 = 0 \end{aligned}$$

- The Lagrangian:

$$\mathcal{L}(\mathbf{w}_2, \alpha, \beta) = -\mathbf{w}_2^T \mathbf{\Sigma} \mathbf{w}_2 + \alpha(\mathbf{w}_2^T \mathbf{w}_2 - 1) + \beta(\mathbf{w}_2^T \mathbf{w}_1 - 0)$$

- Taking the derivative of the Lagrangian w.r.t. $\mathbf{w}_2$ and setting it to $\mathbf{0}$, we get

$$2\mathbf{\Sigma} \mathbf{w}_2^\star - 2\alpha^\star \mathbf{w}_2^\star - \beta^\star \mathbf{w}_1^\star = \mathbf{0}$$

- We can show that $\beta = 0$ and hence have this eigenvalue equation $\mathbf{\Sigma} \mathbf{w}_2^\star = \alpha^\star \mathbf{w}_2^\star$, implying that $\mathbf{w}_2^\star$ is the eigenvector of $\mathbf{\Sigma}$ with the second largest eigenvalue.

Optimization Problem for Other Principal Components

- Similarly, we can show that the other dimensions are given by the eigenvectors of $\mathbf{\Sigma}$ with decreasing eigenvalues.

- The sample covariance $\mathbf{S} = \frac{1}{N}\mathbf{X}\mathbf{X}^T$ is symmetric, so, for two different eigenvalues, the eigenvectors are orthogonal.
  - If $\mathbf{S}$ is positive definite, then all its eigenvalues are positive.
  - If $\mathbf{S}$ is singular, then its rank, the effective dimensionality, is k with $k < d$ and λ_i, $i = k+1, \ldots, d$ are 0 (λ_i are sorted in descending order).
    - The k eigenvectors with nonzero eigenvalues are the dimensions of the reduced space.
- The first eigenvector (the one with the largest eigenvalue λ_1), $\mathbf{w}_1$, namely, the principal component, explains the largest part of the variance; the second explains the second largest; and so on.

- We have discussed obtain the principal components through a variance minimization formulation, which provides a statistical view for PCA.

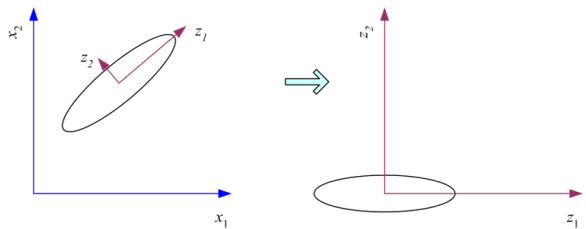
What PCA Does? – I

- Transformation of data:

$$\mathbf{z} = \mathbf{W}^T(\mathbf{x} - \mathbf{m})$$

where the k columns of $\mathbf{W} \in \mathbb{R}^{d \times k}$ are the k leading eigenvectors of the sample covariance $\mathbf{S}$, and $\mathbf{m}$ is the sample mean.

- PCA intuition: centering the data at the origin and rotating the axes:



If $\text{Var}(z_2)$ is too small, it can be ignored to reduce the dimensionality from 2 to 1.

- After the linear transformation, we get a k-dimensional space whose dimensions are the eigenvectors, and the variances over them are equal to the eigenvalues.

What PCA Does? – II

- The eigenvalue decomposition or spectral decomposition of the sample covariance $\mathbf{S}$ is given by

$$\mathbf{S} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$$

where $\mathbf{\Lambda} \in \mathbb{R}^{d \times d}$ and $\mathbf{Q} \in \mathbb{R}^{d \times d}$ with $\mathbf{Q}\mathbf{Q}^T = \mathbf{I}$ are the eigenvalue matrix and eigenvector matrix, respectively, and hence

$$\mathbf{Q}^T\mathbf{S}\mathbf{Q} = \mathbf{\Lambda}$$

- We have

$$\text{Cov}(\mathbf{z}) = \mathbf{W}^T\mathbf{S}\mathbf{W} = \mathbf{\Lambda}_k$$

which is a diagonal matrix.

- PCA intuition: find a matrix $\mathbf{W}$ s.t. the linear transformed data $\mathbf{z} = \mathbf{W}^T(\mathbf{x} - \mathbf{m})$ has diagonal covariance; that is, we would like to get uncorrelated z_i.
- PCA does not use output information and hence is a one-group procedure.

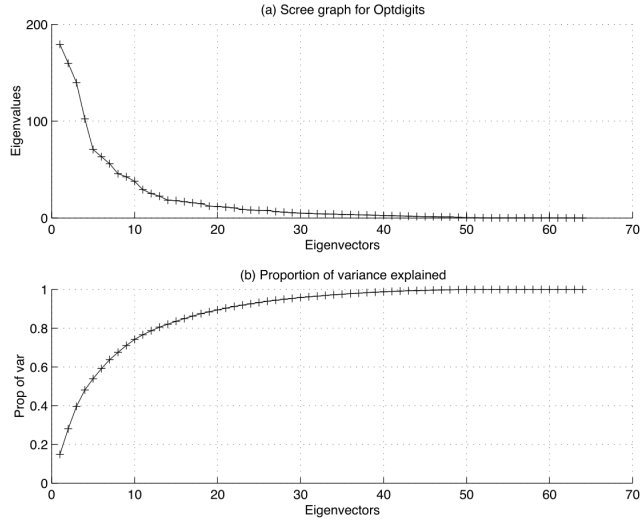
How to Choose k?

- Proportion of variance (PoV) explained (or cumulative explained variance):

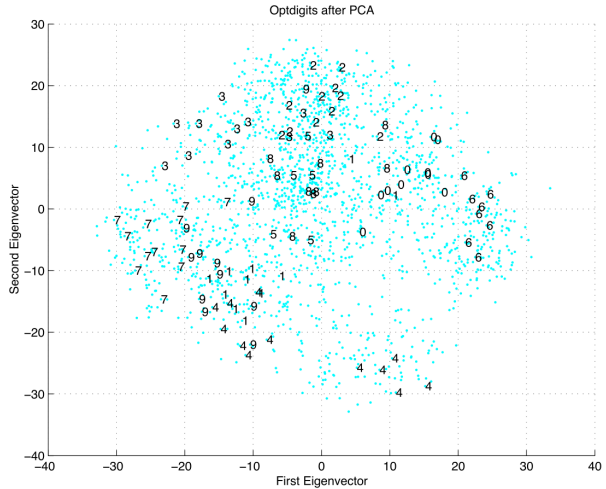
$$\frac{\lambda_1 + \lambda_2 + \cdots + \lambda_k}{\lambda_1 + \lambda_2 + \cdots + \lambda_k + \cdots + \lambda_d}$$

  where λ_d are sorted in descending order.
- Typically, stop at PoV > 0.9.
- Scree graph plotting PoV against k; stop at "elbow".

Scree Graph


(a) Scree graph for Optdigits
Eigenvalues
Eigenvectors
0
100
200
0
10
20
30
40
50
60
70
(b) Proportion of variance explained
Prop of var
Eigenvectors
0
0.2
0.4
0.6
0.8
1
0
10
20
30
40
50
60
70

PCA for Visualization: Scatterplot in Lower-Dimensional Space



An Alternative Equivalent Formulation for PCA

- Given the transformation of data $\mathbf{z} = \mathbf{W}^T(\mathbf{x} - \mathbf{m})$, where $\mathbf{W} \in \mathbb{R}^{d \times d}$ with $\mathbf{W}\mathbf{W}^T = \mathbf{I}$. We have

$$\mathbf{x} = \mathbf{m} + \mathbf{W}\mathbf{z}$$

- If $\mathbf{W} \in \mathbb{R}^{d \times k}$ with columns to be the principal components, the reconstruction of $\mathbf{x}^t$ from its representation in the lower-dimensional $\mathbf{z}$-space is

$$\hat{\mathbf{x}}^t = \mathbf{m} + \mathbf{W}\mathbf{z}^t \quad \text{or} \quad \mathbf{x}^t = \mathbf{m} + \mathbf{W}\mathbf{z}^t + \boldsymbol{\epsilon}^t$$

- It can be proved that among all orthogonal linear projections, PCA minimizes the reconstruction error (a geometric view of PCA), i.e.,

$$\begin{aligned} \underset{\mathbf{m},\mathbf{W},\{\mathbf{z}^t\}}{\text{minimize}} \quad & \frac{1}{2}\sum_{t=1}^{N} \|\mathbf{x}^t - (\mathbf{m} + \mathbf{W}\mathbf{z}^t)\|_2^2 = \frac{1}{2}\|\mathbf{X} - \mathbf{m}\mathbf{1}^T - \mathbf{W}\mathbf{Z}\|_F^2 \\ \text{subject to} \quad & \mathbf{W}^T\mathbf{W} = \mathbf{I} \end{aligned}$$

  - We can pre-subtract the sample mean from $\mathbf{x}^t$ or constrain $\sum_t \mathbf{z}^t = \mathbf{0}$ if we expect $\mathbf{m}$ to be the sample mean estimate of $\mathbf{x}^t$.

Probabilistic PCA

- PCA model is not a generative model, since the low-dimensional representation $\{\mathbf{z}^t\}$ and the error $\{\boldsymbol{\epsilon}^t\}$ are not treated as random variables. As a consequence, the PCA model cannot be used to generate new samples of the random variable $\mathbf{x}$.
- To address this issue, the probabilistic PCA (PPCA) assume that $\mathbf{z}$ and $\boldsymbol{\epsilon}$ are independent random variables with some pdfs, then it generates an $\mathbf{x}$ by

$$\mathbf{x} = \mathbf{m} + \mathbf{W}\mathbf{z} + \boldsymbol{\epsilon}$$

- Let the mean and covariance of $\mathbf{z}$ be denoted by $\boldsymbol{\mu}_z$ and $\boldsymbol{\Sigma}_z$ (commonly assuming $\boldsymbol{\Sigma}_z = \mathbf{I}_k$), respectively and the mean and covariance of $\boldsymbol{\epsilon}$ be denoted by $\mathbf{0}$ and $\boldsymbol{\Sigma}_\epsilon$ (commonly assuming $\boldsymbol{\Sigma}_\epsilon = \psi^2 \mathbf{I}_d$). Then we have

$$\boldsymbol{\mu} = \mathbf{m} + \mathbf{W}\boldsymbol{\mu}_z \quad \text{and} \quad \boldsymbol{\Sigma} = \mathbf{W}\boldsymbol{\Sigma}_z\mathbf{W}^T + \boldsymbol{\Sigma}_\epsilon$$

- Then we can estimate $\mathbf{m}$, $\mathbf{W}$, $\boldsymbol{\mu}_z$, $\boldsymbol{\Sigma}_z$, and $\boldsymbol{\Sigma}_\epsilon$ from the estimates of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ or directly from the sample $\{\mathbf{x}^t\}$ through, say, MLE.