Dimensionality reduction

Subhransu Maji

CMPSCI 689: Machine Learning

9 April 2015

Motivation

- Data visualization
  - Hard to visualize data that lives in high dimensions — reduce it two two or three dimensions for visualization

- Curse of dimensionality
  - Some learning methods don’t scale well with the number of features (e.g., kNN, kernel density estimators)
  - Lower memory overhead and training/testing time
  - Fewer dimensions is a form of regularization

The goal is to reduce the dimension of the data in high-dimensions (say 10000) to low dimensions (say 2) while retaining the “important” characteristics of the data

Unsupervised setting, so the notion of important characteristics is hard to define

Closely related to clustering
  - Clustering: reduce the number of data
  - Dimensionality reduction: reduce the number of features

All you can do is project the data onto a vector and use the projected distances as the embeddings

Example: projecting two dimensional data to one

Linear dimensionality reduction

$x_i \in \mathbb{R}^D, i = 1, 2, \ldots, N$

data matrix $= R^{N \times D}$

Maximum variance

Minimum variance

Linear projection
Find a **linear projection** that **maximizes** the variance of the projection

Assume we have data \( x_1, x_2, \ldots, x_N \in \mathbb{R}^D \) of zero mean

Let \( u \) be the **projection vector**

Let the **projections** of the data \( p_1, p_2, \ldots, p_N \)

The mean of the projections is zero

Maximize the variance of the projection:

\[
\max_u \sum_i (x_i^T u)^2 \quad \text{subject to: } ||u|| = 1
\]

Compute the **data covariance matrix** \( X^T X \)

\[
[X^T X]_{ij} = \sum_n x_{ni} x_{nj}
\]

The optimal (maximal variance) projection direction is the first eigenvector of the data covariance matrix

\[
(X^T X) u = \lambda u
\]

What about learning a second projection direction?

For non-redundancy additionally require that \( v^T u = 0 \)

\[
\max_v ||Xv||^2 \quad \text{subject to: } v^T v = 1, v^T u = 0
\]

This is the second eigenvector of the data covariance matrix

Let \( X \) be the NxD data matrix (each row is data point)

The **projection vector** \( u \) is a Dx1 matrix

The vector of **projections** is given by \( Xu \), a Nx1 matrix

We can rewrite the optimization as:

\[
\max_u ||Xu||^2 \quad \text{subject to: } u^T u = 1
\]

The corresponding **Lagrangian** is:

\[
L(u, \lambda) = ||Xu||^2 - \lambda(u^T u - 1)
\]

At maxima:

\[
\Delta u = 2X^T Xu - 2\lambda u
\]

\[
\implies (X^T X) u = \lambda u \quad \text{eigenvalue problem}
\]

Generalizing this argument leads to **principal component analysis**

The eigenvectors give you the projection directions — to compute the embeddings you have to multiply the data by the projections

For completeness here is the **Matlab** code:
Application: Eigenfaces

- **Eigenfaces** — a linear basis for face images [Turk, Pentland '91]
- Each face is a weighted linear combination of eigenfaces
- Compare faces by comparing the weights

**Input images**

**Principal components**

What should K be?

- For visualization K=2 or 3
- For dimensionality reduction it depends on the problem
  - Option: ignore projections that correspond to small eigenvalues
  - Option: based on computational and memory constraints

Kernel PCA

- We can use the kernel trick to learn linear projections in feature space
- PCA representer theorem: the projection direction is a linear combination of the data points
  \[ u = \sum_i \alpha_i \phi(x_i) \]

- PCA using only dot products between the data
  \[
  \max_u \sum_i \left( \phi(x_i)^T u \right)^2 \text{ subject to: } ||u|| = 1
  \]

\[
\begin{align*}
\sum_i \left( \phi(x_i)^T \left( \sum_j \alpha_j \phi(x_j) \right) \right)^2 &= \sum_i \left( \sum_j \phi(x_i)^T \phi(x_j) \alpha_j \right)^2 \\
&= \sum_i \left( \sum_j K_{ij} \alpha_j \right)^2 = ||K\alpha||^2
\end{align*}
\]

\[
\begin{align*}
\left( \sum_i \phi(x_i) \alpha_i \right)^T \left( \sum_j \phi(x_j) \alpha_j \right) &= \sum_i \sum_j \phi(x_i)^T \phi(x_j) \alpha_i \alpha_j \\
&= \alpha^T K \alpha
\end{align*}
\]

Kernel PCA

- Formulation using kernels:
  \[
  \max_{\alpha} ||K\alpha||^2 \text{ subject to: } \alpha^T K \alpha = 1
  \]
- The corresponding Lagrangian is:
  \[
  \mathcal{L}(\alpha, \lambda) = ||K\alpha||^2 - \lambda (\alpha^T K \alpha - 1)
  \]
- At optimality:
  \[
  \Delta_{\alpha} \mathcal{L}(\alpha, \lambda) = 2K^T K \alpha - 2\lambda K \alpha \\
  \implies K \alpha = \lambda \alpha , \text{ since: } K^T = K
  \]
- Hence \( \alpha \) is an eigenvector of the kernel matrix \( K \)
- Different eigenvectors correspond to different projections
Subhransu Maji (UMASS)
CMPSCI 689  

Kernel PCA

- How do we center the data in kernel space?
  - Recall that PCA requires zero mean data
- Centering be written in terms of kernels as well:
  \[
  \text{dot product} = (\phi(x_i) - \mu)^T (\phi(x_j) - \mu) = \phi(x_i)^T \phi(x_j) - \mu \phi(x_i)^T - \mu^T \phi(x_j) + \mu^T \mu
  \]
  \[
  \Rightarrow K' = K - 1K - K1 + 1K1
  \]
- Where the matrix 1 is defined as:
  \[
  1_{ij} = \frac{1}{N}
  \]
- Perform PCA on the $K'$ matrix and compute the eigenvectors $\alpha$
- Projections of the data are $K\alpha$

Subhransu Maji (UMASS)
CMPSCI 689  

Spectral clustering revisited

- Normalized cuts objective:
  \[
  \min_x \text{NCut}(x) = \min_y \frac{y^T(D - W)y}{y^T Dy}
  \]
  subject to: $y^T D 1 = 0$
  $y(i) \in \{1, -b\}$
- Relax the integer constraint on $y$:
  \[
  \min_y y^T (D - W)y; \text{ subject to: } y^T Dy = 1
  \]
- Same as: $(D - W)y = \lambda Dy$ \textit{(generalized eigenvalue problem)}
- Note that $(D - W)1 = 0$, so the first eigenvector is $y_1 = 1$, with the corresponding eigenvalue of 0
- The eigenvector corresponding to the second smallest eigenvalue is the solution to the relaxed problem

Subhransu Maji (UMASS)
CMPSCI 689  

Spectral clustering example

- Spectral clustering = spectral embedding + thresholding (or k-means)
- Recall the earlier example
  - Gaussian weighted edges connected to 3 nearest neighbors
  - Below are the components of the eigenvector corresponding to the second smallest eigenvalue
Spectral embedding examples

- Image segmentation: multiple eigenvalues reshaped into an image


- Toy dataset

Summary

- Dimensionality reduction for visualization or preprocessing

  - Linear methods
    - PCA — linear projections of data
      - solve $(X^TX)x = \lambda x$ — eigenvectors of covariance matrix
  
  - Non-linear methods
    - kernel PCA — linear projections in kernel space
      - solve $Kx = \lambda x$ — eigenvectors of the kernel matrix
    - Spectral embedding — graph partitions
      - solve $(D - W)x = \lambda Dx$ — eigenvectors of the Graph laplacian

- There are several methods that we didn’t discuss
  - ISOMAP, locally linear embedding, tSNE, etc

Slides credit

- Some of the slides are based on CIMAL book by Hal Daume III
- The example for kernel PCA is from: http://sebastianraschka.com/Articles/2014_kernel_pca.html