Day 9: Unsupervised learning, dimensionality reduction

Introduction to Machine Learning Summer School
June 18, 2018 - June 29, 2018, Chicago

Instructor: Suriya Gunasekar, TTI Chicago

28 June 2018
Topics so far

• Linear regression

• Classification
  o Logistic regression
  o Maximum margin classifiers, kernel trick
  o Generative models
  o Neural networks
  o Ensemble methods

• Today and Tomorrow
  o Unsupervised learning – dimensionality reduction, clustering
  o Review
Unsupervised learning

• **Unsupervised learning:** Requires data \( x \in \mathcal{X} \), but no labels

• **Goal?** Compact representation of the data by detecting patterns
  - e.g. Group emails by topic

• Useful when we don’t know what we are looking for
  - makes evaluation tricky

• Applications in visualization, exploratory data analysis, semi-supervised learning
Clustering
Clustering languages

[Image from scienceinschool.org]
Clustering species (phylogeny)
Image clustering/segmentation

Goal: Break up the image into meaningful or perceptually similar regions

Current trend is to use datasets with labels for such task e.g., MS COCO
Dimensionality reduction

• Input data $x \in \mathcal{X}$ may have thousands or millions of dimensions!
  o e.g., text data represented as bag or words
  o e.g., video stream of images
  o e.g., fMRI data #voxels x #timesteps

• Dimensionality reduction: represent data with fewer dimensions
  o easier learning in subsequent tasks (preprocessing)
  o visualization
  o discover intrinsic patterns in the data
Manifolds

Slide from Yi Zhang
Embeddings

t-SNE visualization from Turian et al. (2010)

[Saul & Roweis ’03]
Low dimensional embedding

• Given high dimensional feature
  \[ x = [x_1, x_2, \ldots, x_d] \]
  find transformations
  \[ z(x) = [z_1(x), z_2(x), \ldots, z_k(x)] \]
  so that “almost all useful information” about \( x \) is retained in \( z(x) \)

• In general \( k \ll d \), and \( z(x) \) is not invertible

• Transformation learned from a dataset of examples of \( x \)
  \[ S = \{ x^{(i)} \in \mathbb{R}^d : i = 1, 2, \ldots, N \} \]
  o Note: typically no labels \( y \)
Linear dimensionality reduction

• Given high dimensional feature

\[ x = [x_1, x_2, ..., x_d] \]

find transformations

\[ z = z(x) = [z_1(x), z_2(x), ..., z_k(x)] \]

• Restrict \( z(x) \) to be a linear function of \( x \)

\[ z_1 = W_1 \cdot x \]
\[ z_2 = W_2 \cdot x \]
\[ \vdots \]
\[ z_k = W_k \cdot x \]

where

\[ z \in \mathbb{R}^k, \quad W \in \mathbb{R}^{k \times d}, \quad x \in \mathbb{R}^d \]

only question is which \( W \)?
Linear dimensionality 2D example

- Given points $S = \{x^{(i)}: i = 1, 2, ..., N\}$ in 2D, we want a 1D representation
  - project $\{x^{(i)}\}$ onto a line $w \cdot x = 0$
  - Find $w$ to minimizes the sum of squared distances to the line
Vector projections

- \( \mathbf{x} \cdot \mathbf{u} = \|\mathbf{x}\|\|\mathbf{u}\| \cos \theta \)
- Assuming \( \|\mathbf{u}\| = 1 \),
- \( \mathbf{x} \cdot \mathbf{u} = \|\mathbf{x}\|\cos \theta = z_u \rightarrow \text{value of } \mathbf{x} \text{ along } \mathbf{u} \)

- Distance of \( \mathbf{x} \) to projection is \( \|z_u \mathbf{u} - \mathbf{x}\| = \|(\mathbf{x} \cdot \mathbf{u})\mathbf{u} - \mathbf{x}\| \)
Principal component analysis

• For a 1D embedding along direction \( \mathbf{u} \), distance of \( \mathbf{x} \) to the projection along \( \mathbf{u} \) is given by
  \[
  \|z_u \mathbf{u} - \mathbf{x}\| = \|(x \cdot \mathbf{u}) \mathbf{u} - \mathbf{x}\|
  \]

• More generally for \( k \) dimensional embedding:
  o find orthonormal basis of the \( k \) dimensional subspace \( \mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k \in \mathbb{R}^d \), i.e., \( \mathbf{u}_i \cdot \mathbf{u}_j = 1 \) if \( i = j \), and 0 otherwise
  o let \( \mathbf{U} \in \mathbb{R}^{k \times d} \) be the matrix with \( \mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k \) along rows
  o distance of projection of \( \mathbf{x} \) to \( \text{span}\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k\} \)
    \[
    \|\mathbf{U}^\top \mathbf{U} \mathbf{x} - \mathbf{x}\|
    \]
  o also from orthonormality of \( \mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k \), check \( \mathbf{U} \mathbf{U}^\top = \mathbf{I} \)

• PCA objective
  \[
  \min_{\mathbf{U} \in \mathbb{R}^{k \times d}} \sum_{i=1}^{N} \left\|\mathbf{U}^\top \mathbf{U} \mathbf{x}^{(i)} - \mathbf{x}^{(i)}\right\|^2 \quad \text{s.t.} \quad \mathbf{U} \mathbf{U}^\top = \mathbf{I}
  \]
PCA

• PCA objective

\[
\min_{U \in \mathbb{R}^{k \times d}} \frac{1}{N} \sum_{i=1}^{N} \| U^T U x^{(i)} - x^{(i)} \|^2 \quad \text{s.t. } UU^T = I
\]

• Also, for all \( UU^T = I \)

\[
\| U^T U x - x \|^2 = \| x \|^2 + x^T U^T U U^T U x - 2 x^T U^T U x
\]

\[
= \| x \|^2 - x^T U^T U x = \| x \|^2 - \| Ux \|^2
\]

• Equivalent PCA objective

\[
\max_{U} \frac{1}{N} \sum_{i=1}^{N} \| U x^{(i)} \|^2 = \sum_{j \in [k]} u_j^T \hat{\Sigma}_{xx} u_j \quad \text{s.t. } UU^T = I
\]

where \( \hat{\Sigma}_{xx} = \frac{1}{N} \sum_{i=1}^{N} x^{(i)} x^{(i)^T} \) (derivation in board)

• This is the same as finding top k eigenvectors of \( \hat{\Sigma}_{xx} \)
PCA algorithm

• Given $S = \{x^{(i)} \in \mathbb{R}^d : i = 1, 2, \ldots, N\}$

• Let $X \in \mathbb{R}^{N \times d}$ be data matrix
  
  o make sure $X$ is re-centered so that column mean is 0

• $\hat{\Sigma}_{xx} = \frac{1}{N} \sum_{i=1}^{N} x^{(i)} x^{(i)\top} = \frac{1}{N} X\top X \in \mathbb{R}^{d \times d}$

• $u_1, u_2, \ldots, u_k \in \mathbb{R}^d$ are top $k$ eigenvectors of $\hat{\Sigma}_{xx}$
How to pick $k$?

- Data assumed to be low dimensional projection + noise
- Only keep projections onto components with large eigenvalues and ignore the rest

*Slide credit: Arti Singh*
Eigenfaces

• Input images:

• Principal components:

• Turk and Pentland ’91
SVD version

• Given $S = \{x^{(i)} \in \mathbb{R}^d: i = 1,2, \ldots, N\}$

• Let $X \in \mathbb{R}^{N \times d}$ be data matrix
  o make sure $X$ is re-centered so that column mean is 0

• $X = \overline{V} \overline{S} \overline{U}^\top$ be the Singular Value Decomposition (SVD) of $X$, where
  o $\overline{V} \in \mathbb{R}^{N \times d}$ have orthonormal columns, i.e., $\overline{V}^\top \overline{V} = I$
    ▪ columns of $\overline{V}$ are called left singular vectors
  o $\overline{U} \in \mathbb{R}^{d \times d}$ also has orthonormal columns, i.e., $\overline{U}^\top \overline{U} = I$
    ▪ columns of $\overline{U}$ are called right singular vectors
  o $\overline{S} = \text{diagonal}(\sigma_1, \sigma_2, \ldots, \sigma_d) \in \mathbb{R}^{d \times d}$
    ▪ $\sigma_1, \sigma_2, \ldots, \sigma_d$ are called the singular values

• First $k$ columns of $\overline{U}$ are the $u_1, u_2, \ldots, u_k$ we want.

• Representation of $x \in \mathbb{R}^d$ as $z(x) \in \mathbb{R}^k$ is given by
  $z(x)_j = \sigma_j \, u_j \cdot x$ for $j = 1,2, \ldots, k$
Other linear dimensionality reduction

- **PCA:** given data $x \in \mathbb{R}^d$, find $U \in \mathbb{R}^{k \times d}$ to minimize
  $$\min_{U} ||U^T Ux - x||_2^2 \quad \text{s.t.} \quad UU^T = I$$

- **Canonical correlation analysis:** given two “views” of data $x \in \mathbb{R}^d$ and $x' \in \mathbb{R}^{d'}$, find $U \in \mathbb{R}^{k \times d}, U' \in \mathbb{R}^{k \times d'}$ to minimize
  $$||Ux - U' x'||_2^2 \quad \text{s.t.} \quad UU^T = U'U'^T = I$$

- **Sparse dictionary learning:** learn a sparse representation of $x$ as a linear combination of over-complete dictionary
  $$x \rightarrow Dz \text{ where } D \in \mathbb{R}^{d \times m}, z \in \mathbb{R}^m$$
  - unlike PCA, here $m \gg d$ so $z$ is higher dimensional, but learned to be sparse!

- **Independent component analysis**
- **Factor analysis**
- **Linear discriminant analysis**
Non linear dimensionality reduction

- Isomap
- Autoencoders
- Kernel PCA
- Local linear embedding
- Check out t-SNE for 2D visualization
- ...

21
Isomap

[Tenenbaum, Silva, Langford. Science 2000]
Isomap – algorithm

• Dataset of \( N \) points \( S = \{ x^{(i)} \in \mathbb{R}^d : i = 1, 2, \ldots, N \} \)

• Represent the points as a kNN-graph with weights proportional to distance between the points

• The geodesic distance \( d(x, x') \) between points in the manifold is the length of shortest path in the graph

• Use any shortest path algorithm can be used to construct a matrix \( M \in \mathbb{R}^{N \times N} \) of \( d(x^{(i)}, x^{(j)}) \) for all \( x^{(i)}, x^{(j)} \in S \)

• MDS: Find a (low dimensional) embedding \( z(x) \) of \( x \) so that distances are preserved

\[
\min_z \sum_{i,j \in [N]} (\|z(x^{(i)}) - z(x^{(j)})\| - M_{ij})^2
\]

\[
\text{sometimes } \min_z \sum_{i,j \in [N]} \left( \frac{\|z(x^{(i)}) - z(x^{(j)})\| - M_{ij})^2}{M_{ij}} \right)
\]
Autoencoders

• Recall neural networks as feature learning

  o was learned for some supervised learning task
  o weights learned by minimizing $\ell(v_{out}, y)$
  o but we don’t have $y$ anymore!
Autoencoders

• Recall neural networks as feature learning

 o was learned for some supervised learning task
 o weights learned by minimizing $\ell(v_{out}, y)$
 o but we don’t have $y$ anymore!
 o instead use another “decoder” network to reconstruct $x$
Autoencoders

\[ \phi(x) = f_{W_1}(x) \]
\[ \tilde{x} = f_{W_2}(\phi(x)) \]
• some loss \( \ell(\tilde{x}, x) \)
\[
\hat{W}_1, \hat{W}_2 = \min_{W_1, W_2} \sum_{i=1}^{N} \ell \left( f_{W_2} \left( f_{W_1}(x^{(i)}) \right), x^{(i)} \right)
\]
• learn using SGD with backpropagation