Machine learning - HT 2016

9. Dimensionality Reduction & Multidimensional Scaling

Varun Kanade

University of Oxford

March 2, 2016

Supervised Learning: Summary

- Training data is of the form $\langle(\mathbf{x}_i, y_i)\rangle$ where $\mathbf{x}_i$ are features and y_i is target
- We formulate a probabilistic (or deterministic) model for $y \mid \mathbf{x}, \mathbf{w}$
- Choose a suitable loss function; minimize training loss
- Use regularization or other techniques to reduce overfitting
- Use trained classifier to predict targets/labels on unseen $\mathbf{x}_{\mathrm{new}}$

Unsupervised Learning

- Training data is of the form $\langle(\mathbf{x}_i)\rangle_{i=1}^{m}$
- Infer properties about the data
- Example: Clustering - can the data be grouped into categories?
- Example: Density Estimation
- **Today**: Dimensionality Reduction and Multi-dimensional Scaling (MDS)

Outline

Today, we'll study techniques for dimensionality reduction and multidimensional scaling

- Principal Component Analysis (PCA)
- Kernel PCA
- Multidimensional Scaling: Reconstruct data from similarity or dissimilarity measures

Dimensionality Reduction

Why perform dimensionality reduction?

- Computational Reasons - time/storage efficiency
- Statistical Resaons - better generalization guarantees
- Visualization - helps understand data

Objective

- Lower dimensional representation that preserves *essential* properties

Johnson-Lindenstrauss Lemma

Project data onto random k dimensional subspace

All pairwise distances are approximately preserved

Principal Component Analysis (PCA)

PCA is a *linear* dimensionality reduction technique

Find the directions of maximum variance in the data $\langle(\mathbf{x}_i)\rangle_{i=1}^{m}$

Assume that data is centered, *i.e.*, $\sum_i \mathbf{x}_i = \mathbf{0}$

Find a set of orthogonal vectors $\mathbf{v}_1, \ldots, \mathbf{v}_k$

- The first principal component (PC) $\mathbf{v}_1$ is the direction of largest variance
- The second PC $\mathbf{v}_2$ is the direction of largest variance orthogonal to $\mathbf{v}_1$
- The i^{th} PC $\mathbf{v}_i$ is the direction of largest variance orthogonal to $\mathbf{v}_1, \ldots, \mathbf{v}_{i-1}$

$\mathbf{V}_{n\times k}$ gives projection

$$\mathbf{z}_i = \mathbf{V}^T \mathbf{x}_i$$

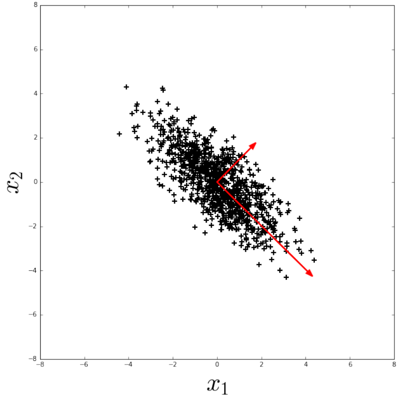
PCA: Directions that maximise variance

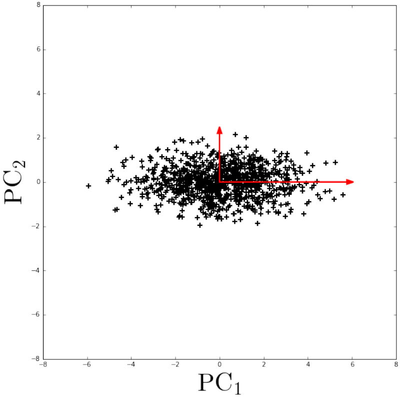
We are given i.i.d. data $\langle(\mathbf{x}_i)\rangle_{i=1}^{m}$; data matrix $\mathbf{X}$

Want to find $\mathbf{v}_1 \in \mathbb{R}^n$, $\|\mathbf{v}_1\| = 1$, that maximizes $\|\mathbf{X}\mathbf{v}_1\|^2$

Find $\mathbf{v}_2, \mathbf{v}_3, \ldots, \mathbf{v}_k$ that are all successively orthogonal to previous directions and maximise (as yet unexplained variance)

Principal Component Analysis (PCA)





PCA: Best Reconstruction

We are given i.i.d. data $\langle(\mathbf{x}_i)\rangle_{i=1}^{m}$; data matrix $\mathbf{X}$

Find a k-dimensional linear projection that best "models" the data

Suppose $\mathbf{V}_k \in \mathbb{R}^{n \times k}$ is such that columns of $\mathbf{V}_k$ are orthogonal

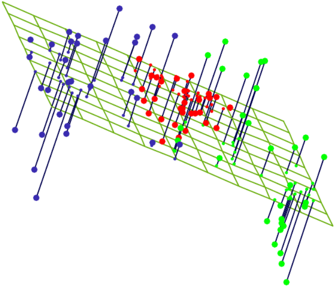
Project data $\mathbf{X}$ on to subspace defined by $\mathbf{V}$

$$\mathbf{Z} = \mathbf{X}\mathbf{V}_k$$

Minimize reconstruction error:

$$\sum_{i=1}^{m} \|\mathbf{x}_i - \mathbf{V}_k\mathbf{V}_k^T\mathbf{x}_i\|^2$$

Principal Component Analysis (PCA)



Equivalence between two objectives

Let $\mathbf{v}_1$ be the direction of projection

The point $\mathbf{x}$ is mapped to $\langle \mathbf{v}_1, \mathbf{x} \rangle \mathbf{v}_1$, where $\|\mathbf{v}_1\| = 1$

Finding Principal Components: SVD

Let $\mathbf{X}$ be the $m \times n$ data matrix (say $n < m$)

Pair of singular vectors $\mathbf{u} \in \mathbb{R}^m$, $\mathbf{v} \in \mathbb{R}^n$ and singular value $\sigma \in \mathbb{R}^+$ if

$$\sigma\mathbf{u} = \mathbf{X}\mathbf{v} \quad \text{and} \quad \sigma\mathbf{v} = \mathbf{X}^T\mathbf{u}$$

$\mathbf{v}$ is an eigenvector of $\mathbf{X}^T\mathbf{X}$ with eigenvalue σ^2

$\mathbf{u}$ is an eigenvector of $\mathbf{X}\mathbf{X}^T$ with eigenvalue σ^2

Finding Principal Components: SVD

$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$

Thin SVD: $\mathbf{U}$ is $m \times n$, $\mathbf{\Sigma}$ is $n \times n$, $\mathbf{V}$ is $n \times n$, $\mathbf{U}^T\mathbf{U} = \mathbf{V}^T\mathbf{V} = \mathbf{I}$

$\mathbf{\Sigma}$ is diagonal with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$

The first k principal components are first k columns of $\mathbf{V}$

PCA: Reconstruction Error

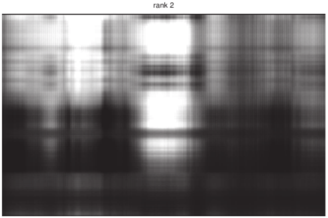
We have thin SVD: $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$

Let $\mathbf{V}_k$ be the matrix containing first k columns of $\mathbf{V}$

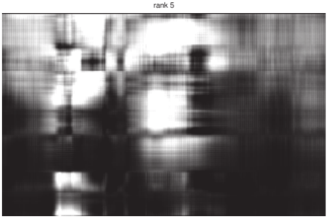
Projection $\mathbf{Z} = \mathbf{X}\mathbf{V}_k = \mathbf{U}_k\mathbf{\Sigma}_k$

Reconstruction error $= \sum_{i=1}^{m} \|\mathbf{x}_i - \mathbf{V}_k\mathbf{V}_k^T\mathbf{x}_i\|^2 = \sum_{j=k+1}^{n} \sigma_i^2$

(a)



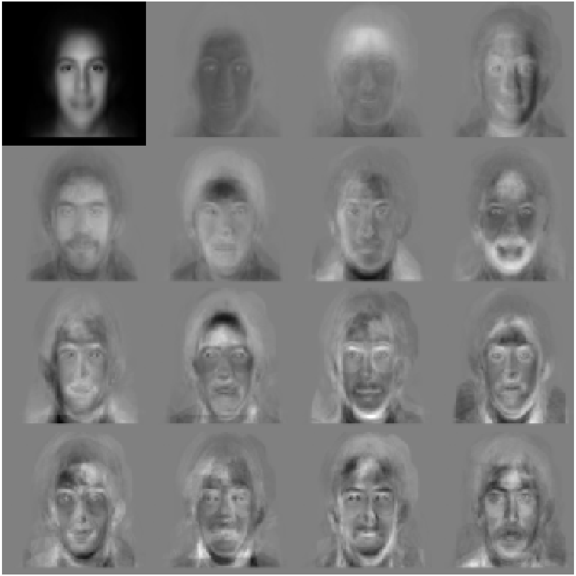
(b)



(c)

(d)

Eigenfaces



Source: http://vismod.media.mit.edu/vismod/demos/facerec/basic.html

Latent Semantic Analysis

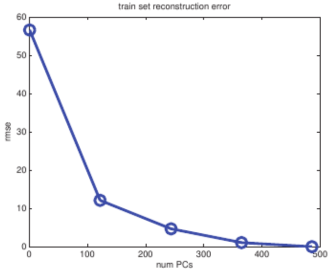
$\mathbf{X}$ is an $m \times n$, n is the size of dictionary

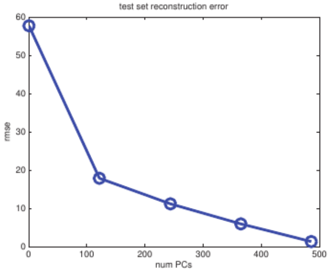
$\mathbf{x}_i$ is a vector of word counts (bag of words)

Reconstruction using k eigenvectors $\mathbf{X} \approx \mathbf{Z}\mathbf{V}_k^T$, where $\mathbf{Z} = \mathbf{X}\mathbf{V}_k$

$\langle \mathbf{z}_i, \mathbf{z}_j \rangle$ is probably a better notion of similarity than $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$

How many principal components to pick?





PCA Summary

Algorithm: We've expressed PCA as SVD of data matrix $\mathbf{X}$

Equivalently, we can use eigendecomposition of co-variance matrix $\mathbf{X}^T\mathbf{X}$

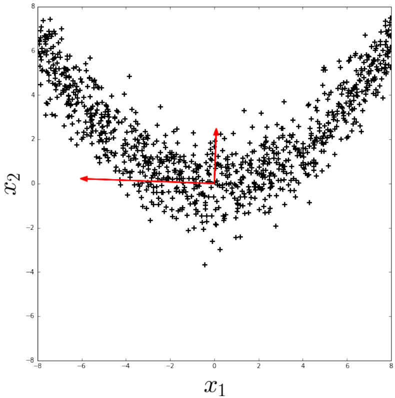
Running Time: $O(mnk)$ to compute k principal components (avoid computing covariance matrix)

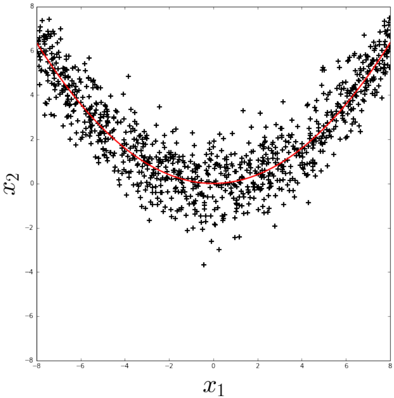
PCs are uncorrelated, but there may be non-linear (higher-order) effects

PCA depends on scale or units of measurement; it may be a good idea to standardize data

PCA is sensitive to outliers

PCA: Going beyond linearity





We can perform basis expansion $\phi(\mathbf{x}) = (x_1, x_1^2, x_1x_2, \ldots,)^T$

Kernel PCA

Representation:

PCs can be expressed in terms of the datapoints $\mathbf{x}_i$. Why?

Suppose $\mathbf{v}_1 = \mathbf{X}^T \boldsymbol{\alpha}$, *i.e.*,$\mathbf{v}_1 = \sum_{i=1}^m \alpha_i \mathbf{x}_i$

Objective

$$\max_{\|\mathbf{v}_1\|=1} \mathbf{v}_1^T \mathbf{X}^T \mathbf{X} \mathbf{v}_1 = \max_{\|\boldsymbol{\alpha}^T \mathbf{X}\mathbf{X}^T \boldsymbol{\alpha}\|=1} \boldsymbol{\alpha}^T (\mathbf{X}\mathbf{X}^T)^2 \boldsymbol{\alpha}$$

We only need $\mathbf{K} = \mathbf{X}\mathbf{X}^T$ to compute $\boldsymbol{\alpha}$

Kernel PCA

Objective

$$\max_{\|\boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha}\|=1} \boldsymbol{\alpha}^T \mathbf{K}^2 \boldsymbol{\alpha},$$

where $\mathbf{K} = \mathbf{X}\mathbf{X}^T$. What is the solution $\boldsymbol{\alpha}$?

Kernel PCA

As in the case of SVM, we can use many different types of kernels $\kappa(\mathbf{x}, \mathbf{x}')$

Examples

- Linear kernel: $\kappa(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}'$
- Polynomial kernel: $\kappa(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x}^T \mathbf{x}')^d$
- Gaussian (RBF) kernel: $\kappa(\mathbf{x}, \mathbf{x}') = \exp(\|\mathbf{x}^T - \mathbf{x}'\|^2)$
- Kernels useful for combinatorial objects: cosine, string kernel, etc.

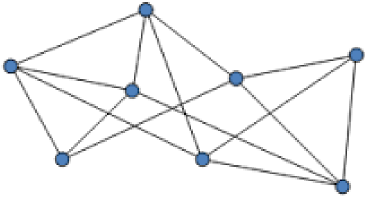
Mercer's Theorem

As long as κ always results in a positive definite Gram matrix, there exists a high-dimensional feature space ϕ, such that $\kappa(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x})$

Multidimensional Scaling

Suppose for some m points in $\mathbb{R}^n$ we are given all pairwise distances in a matrix $\mathbf{D}$

Can we reconstruct $\mathbf{x}_1, \ldots, \mathbf{x}_m$, i.e., all of $\mathbf{X}$?



Multidimensional Scaling

Distances are preserved under translation, rotation, reflection, etc.

We cannot recover $\mathbf{X}$ exactly; we can determine $\mathbf{X}$ up to these transformations

If D_{ij} is the distance between points $\mathbf{x}_i$ and $\mathbf{x}_j$, then

$$\begin{aligned} D_{ij}^2 &= \|\mathbf{x}_i - \mathbf{x}_j\|^2 \\ &= \mathbf{x}_i^T \mathbf{x}_i - 2\mathbf{x}_i^T \mathbf{x}_j + \mathbf{x}_j^T \mathbf{x}_j \\ &= M_{ii} - 2M_{ij} + M_{jj} \end{aligned}$$

Here $\mathbf{M} = \mathbf{X}\mathbf{X}^T$ is the $m \times m$ matrix of dot products

Exercise: Show that assuming $\sum_i \mathbf{x}_i = \mathbf{0}$, $\mathbf{M}$ can be recovered from $\mathbf{D}$

Multidimensional Scaling

Consider the (non-thin) SVD: $\mathbf{X} = \mathbf{U\Sigma V}^T$

We can write $\mathbf{M}$ as

$$\mathbf{M} = \mathbf{X}\mathbf{X}^T = \mathbf{U\Sigma\Sigma}^T\mathbf{U}^T$$

To reconstruct $\tilde{\mathbf{X}}$, consider the eigendecomposition of $\mathbf{M}$

$$\mathbf{M} = \mathbf{U\Lambda U}^T$$

Because, $\mathbf{M}$ is symmetric and positive semi-definite, $\mathbf{U}^T = \mathbf{U}^{-1}$ and all entries of (diagonal matrix) $\mathbf{\Lambda}$ are non-negative

Let $\tilde{\mathbf{X}} = \mathbf{U\Lambda}^{1/2}(= \mathbf{U\Sigma}$ *[after truncation]*$)$

If we are satisfied with approximate reconstruction, we can use truncated eigendecomposition

Multidimensional Scaling: Comments

If the similarity matrix $\mathbf{M}$ is not positive semi-definite, cannot necessarily find a Euclidean embeddding

Minimize stress function: Find $\mathbf{z}_1, \ldots, \mathbf{z}_m$ that minimizes

$$S(\mathbf{Z}) = \sum_{i \neq j} (D_{ij} - \|\mathbf{z}_i - \mathbf{z}_j\|)^2$$

Many other types of stress functions