Exploratory Analysis
Dimensionality Reduction
Feature Extraction
Conclusion

Exploratory Analysis
Dimensionality Reduction

Davide Bacciu
Computational Intelligence & Machine Learning Group
Dipartimento di Informatica
Università di Pisa
bacciu@di.unipi.it

Introduzione all’Intelligenza Artificiale - A.A. 2012/2013
Lecture Outline

1. Exploratory Analysis
2. Dimensionality Reduction
   - Curse of Dimensionality
   - General View
3. Feature Extraction
   - Finding Linear Projections
   - Principal Component Analysis
   - Applications and Advanced Issues
4. Conclusion
Drowning into complex data

- Face images
- Documents
- Gene expression data
- MEG readings

According to media reports, a pair of hackers said on Saturday that the Firefox Web browser, commonly perceived as the safer and more customizable alternative to market leader Internet Explorer, is critically flawed. A presentation on the flaw was shown during the ToorCon hacker conference in San Diego.

Zambian President Levy Mwanawasa has won a second term in office in an election his challenger Michael Sata accused him of rigging. Official results showed on Monday.
Exploratory Data Analysis (EDA)

- Discover structure in data
  - Find unknown patterns in the data that cannot be predicted using current expert knowledge
  - Formulate new hypotheses about the causes of the observed phenomena
- A mix of graphical and quantitative techniques
  - Visualization
  - Finding informative attributes in the data
  - Finding natural groups in the data
- Interdisciplinary approach
  - Computer graphics
  - Machine learning
  - Data Mining
  - Statistics
Often an unsupervised learning task
- Dimensionality reduction
  - Feature Extraction
  - Feature Selection
- Clustering

Tackle with
- Large datasets..
- ...as well as high-dimensional data and small sample size

Exploiting tools and models beyond statistics
- E.g. non-parametric neural models
If the data lies in a high dimensional space, then an enormous amount of data is required to learn a model.

- **Curse of Dimensionality** (Bellman, 1961)
- Some problems become intractable as the number of the variables increases.
  - Huge amount of training data required
  - Too many model parameters (complexity)

Given a fixed number of training samples, the predictive power reduces as sample dimensionality increases (Hughes Effect, 1968)
Exploratory Analysis
Dimensionality Reduction
Feature Extraction
Conclusion

Curse of Dimensionality
General View

A Simple Combinatorial Example (I)

A toy 1-dimensional classification task with 3 classes

Classes cannot be separated well: lets add another feature..

Better class separation, but still errors. What if we add another feature?
A Simple Combinatorial Example (II)

Exponential growth in the complexity of the learned model with increasing dimensionality.

Exponential growth in the number of examples required to maintain a given sampling density:
- 3 samples per bin in 1-D
- 81 samples per bin in 3-D

Classes are well separated.
The **intrinsic dimension** of data is the **minimum number of independent parameters** needed to account for the observed properties of the data.

Data might live in a lower dimensional surface (**fold**) than expected.
What is the Intrinsic Dimension?

Might not be an easy question to answer...

It may increase due to noise
A data fold needs to be unfolded to reveal its intrinsic dimension
Informative Vs Uninformative Features

Data can be made of several dimensions that are either unimportant or comprise only noise

- Irrelevant information might distract the learning model
- Learning resources (memory) are wasted to represent irrelevant portions of the input space

Dimensionality reduction aims at automatically finding a lower-dimensional representation of high-dimensional data

- Counteracts the curse of dimensionality
- Reduces the effect of unimportant attributes
Why Dimensionality Reduction?

- Data Visualization
  - Projecting high-dimensional data to a 2D/3D screen space
  - Preserving topological relationships
  - E.g. visualize semantically related textual documents

- Data Compression
  - Reducing storage requirements
  - Reducing complexity
  - E.g. stopwords removal

- Feature ranking and selection
  - Identifying informative bits of information
  - Noise reduction
  - E.g. identify words correlated with document topics
**Flavors of Dimensionality Reduction**

- **Feature Extraction** - Create a lower dimensional representation of \( \mathbf{x} \in \mathbb{R}^D \) by combining the existing features with a given function \( f : \mathbb{R}^D \rightarrow \mathbb{R}^{D'} \)

\[
\mathbf{x} = \begin{bmatrix}
    x_1 \\
    x_2 \\
    \vdots \\
    x_D
\end{bmatrix} \quad \xrightarrow{y=f(x)} \quad \mathbf{y} = \begin{bmatrix}
    y_1 \\
    y_2 \\
    \vdots \\
    y_{D'}
\end{bmatrix}
\]

- **Feature Selection** - Choose a \( D' \)-dimensional subset of all the features (possibly the most-informative)

\[
\mathbf{x} = \begin{bmatrix}
    x_1 \\
    x_2 \\
    \vdots \\
    x_D
\end{bmatrix} \quad \text{select } i_1, \ldots, i_{D'} \quad \xrightarrow{\text{select } i_1, \ldots, i_{D'}} \quad \mathbf{y} = \begin{bmatrix}
    x_{i_1} \\
    x_{i_2} \\
    \vdots \\
    x_{i_{D'}}
\end{bmatrix}
\]
A Unique Formalization

Definition (Dimensionality Reduction)

Given an input feature space \( \mathbf{x} \in \mathbb{R}^D \) find a mapping \( f : \mathbb{R}^D \rightarrow \mathbb{R}^{D'} \) such that \( D' < D \) and \( \mathbf{y} = f(\mathbf{x}) \) preserves most of the informative content in \( \mathbf{x} \).

Often the mapping \( f(\mathbf{x}) \) is chosen as a linear function \( \mathbf{y} = \mathbf{Wx} \)

- \( \mathbf{y} \) is a linear projection of \( \mathbf{x} \)
- \( \mathbf{W} \in \mathbb{R}^{D' \times D} \) is the matrix of linear coefficients

\[
\begin{bmatrix}
  y_1 \\
  y_2 \\
  \vdots \\
  y_{D'}
\end{bmatrix}
= \begin{bmatrix}
  w_{11} & w_{12} & \cdots & w_{1D} \\
  w_{21} & w_{22} & \cdots & w_{2D} \\
  \vdots & \vdots & \ddots & \vdots \\
  w_{D'1} & w_{D'2} & \cdots & w_{D'D}
\end{bmatrix}
\begin{bmatrix}
  x_1 \\
  x_2 \\
  \vdots \\
  x_D
\end{bmatrix}
\]
The linear/nonlinear map $y = f(x)$ is learned from the data based on an error function that we seek to minimize.

- **Signal representation (Unsupervised)**
  - The goal is to represent the samples accurately in a lower-dimensional space
  - **Principal Component Analysis (PCA)**

- **Classification (Supervised)**
  - The goal is to enhance the class-discriminatory information in the lower-dimensional space
  - **Linear Discriminant Analysis (LDA)**
Objective - Create a lower dimensional representation of $x \in \mathbb{R}^D$ by combining the existing features with a given function $f : \mathbb{R}^D \rightarrow \mathbb{R}^{D'}$, while preserving as much information as possible.

\[
\begin{align*}
\mathbf{x} &= \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{bmatrix} \\
\mathbf{y} &= f(\mathbf{x}) \\
\mathbf{y} &= \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{D'} \end{bmatrix}
\end{align*}
\]

where $D' \ll D$ and, for visualization, $D' = 2$ or $D' = 3$. 
Linear Feature Extraction

- Signal Representation (Unsupervised)
  - Independent Component Analysis (ICA)
  - **Principal Component Analysis** (PCA)
  - Non-negative Matrix Factorization (NMF)

- Classification (Supervised)
  - Linear Discriminant Analysis (LDA)
  - Canonical Correlation Analysis (CCA)
  - Partial Least Squares (PLS)

**We focus on unsupervised approaches exploiting linear mapping functions**
Linear Methods Setup

Given $N$ samples $x_n \in \mathbb{R}^D$, define the input data as the matrix

$$X = \begin{bmatrix}
    x_{11} & \cdots & x_{N1} \\
    \vdots & \ddots & \vdots \\
    x_{1D} & \cdots & x_{ND}
\end{bmatrix} \in \mathbb{R}^D \times \mathbb{R}^N$$

Choose $D' \ll D$ projection directions $w_k$

$$W = \begin{bmatrix}
    w_{11} & \cdots & w_{D'1} \\
    \vdots & \ddots & \vdots \\
    w_{1D} & \cdots & w_{D'D}
\end{bmatrix} \in \mathbb{R}^D \times \mathbb{R}^{D'}$$

Compute the projection of $x$ along each direction $w_k$ as

$$y = [y_1, \ldots, y_{D'}]^T = W^T x$$

Linear methods only differ in the criteria used for choosing $W$
Linear Projection - A Graphical Interpretation

3D samples projected on an hyperplane generated by 2 projection directions

2D projection of the input samples on the hyperplane
**Principal Component Analysis (PCA)**

Orthogonal linear projection of high dimensional data onto a low dimensional subspace preserving as much variance information as possible.

**Objective**

1. Minimize the projection error, i.e., the error of the reconstructed sample $\|\mathbf{x}_n - \tilde{\mathbf{x}}_n\|$.

2. Maximize the variance of the projected data $\mathbf{Y}$.

The good news is that both objectives are equivalent!!
PCA - Two Operations

**Encode**  Project data onto the principal components

\[ \mathbf{y} = \mathbf{W}^T \mathbf{x} \text{ for } k \text{ - th component } y_k = w_k^T \mathbf{x} \]

**Decode**  Reconstruct the projected data

\[ \tilde{\mathbf{x}} = \mathbf{W} \mathbf{y} = \sum_{k=1}^{D'} y_k w_k \]
Given $N$ samples $\{x_n\}_{n=1}^{N}$ and $x_n \in \mathbb{R}^D$

Goal
Project data into a $D' < D$ dimensional space such that the variance of the projected data is maximized

For simplicity consider $D' = 1$

- A single projection direction $w_1$
- Assume normalized vectors $\|w_1\|_2 = 1$
  - Orthonormal basis from numerical analysis
  - Serves to select a single solution among infinite $w$
Variance Maximization - Input Space

- Compute the **means of the input data** \( \{ \mathbf{x}_n \}_{n=1}^{N} \)

\[
\bar{\mathbf{x}} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}_n
\]

- Compute the **covariance of the input data**

\[
\mathbf{S} = \frac{1}{N} \sum_{n=1}^{N} (\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^T
\]
Compute the means of the projected data as \( \mathbf{w}_1^T \mathbf{x} \)

Compute the variance of the projected data as

\[
\frac{1}{N} \sum_{n=1}^{N} \left( \mathbf{w}_1^T \mathbf{x}_n - \mathbf{w}_1^T \mathbf{x} \right)^2 = \frac{1}{N} \sum_{n=1}^{N} \left( \mathbf{w}_1^T (\mathbf{x}_n - \mathbf{x}) \right)^2
\]

\[
= \frac{1}{N} \sum_{n=1}^{N} \mathbf{w}_1^T (\mathbf{x}_n - \mathbf{x})(\mathbf{x}_n - \mathbf{x})^T \mathbf{w}_1
\]

\[
= \mathbf{w}_1^T \mathbf{S} \mathbf{w}_1
\]
**Goal** - Maximizing the variance of the projected data

\[
\mathcal{L} = \max_{\mathbf{w}} \left\{ \mathbf{w}^T \mathbf{S} \mathbf{w} \right\}
\]

subject to the normalization constraint

\[
\|\mathbf{w}\|_2 = 1
\]

**How?** Don’t panic! No theoretical explanation. For that you will need to take the Machine Learning course

Basically, it is an optimization problem that it is solved by differentiating \( \mathcal{L} \) to find its maximum
PCA - Variance Maximization Solution

For $D' = 1$ the solution is the **first principal component** $w = u_1$ such that

$$Su_1 = \lambda_1 u_1$$

where

- $\lambda_1 \in \mathbb{R}$ is the first eigenvalue of $S$ (i.e. the largest)
- $u_1 \in \mathbb{R}^D$ is the associate first eigenvector
- $\lambda_1$ is the variance of the projected data, i.e.

$$\lambda_1 = u_1^T Su_1$$

Maximize the variance $\Rightarrow$ choose eigenvector $u$ with largest associated eigenvalue $\lambda$
What if I want more than 1 projection direction ($D' > 1$)?

- Choose each new direction $w_k$ as one that
  - Maximizes the variance of projected data
  - Is subject to the normalization constraint $\|w_k\|_2 = 1$
  - Is orthogonal to those already selected, i.e. $w_1, \ldots, w_{k-1}$

The solution is in the eigenvectors of the input covariance $S$

- The covariance $S$ of a $D$-dimensional input space has $D$ eigenvectors
  - The eigenvector $u_1$ of the largest eigenvalue $\lambda_1$ is the first principal component
  - The eigenvector $u_2$ of the second-largest eigenvalue $\lambda_2$ is the second principal component
  - The eigenvector $u_3$ of the third-largest eigenvalue $\lambda_3$ is the third principal component
  - \ldots
PCA Solution - Eigenvalue Decomposition

The PCA solution reduces to finding the **eigenvalue decomposition** of the covariance matrix of input data

$$
S = U \Lambda U^T
$$

where

- $U = [u_k]_{k=1}^{D}$ is the $D \times D$ matrix of **eigenvectors** $u_k$
- $\Lambda$ is the $D \times D$ **diagonal matrix** whose diagonal element $\lambda_k$ is the $k$-th eigenvalue

A $D' < D$ dimensional projection space is created by choosing $D'$ eigenvectors $\{u_k\}_{k=1}^{D'}$ corresponding to the $D'$ largest eigenvalues $\{\lambda_k\}_{k=1}^{D'}$
Practical PCA (I)

Step 1  **Organize Data** - Put your $N$ samples into a $D \times N$ matrix $X$

Step 2  **Compute Means** - Calculate the empirical means of your data

$$\bar{x} = \frac{1}{N} \sum_{n=1}^{N} x_n$$

Step 3  **Preprocess Data** - Subtract means $\bar{x}$ to each input sample

$$\bar{X} = X - \bar{x}$$
Practical PCA (Ia)

Input data Compute means Rescale data
Step 4 Compute Covariance - Calculate the covariance of input data

\[ S = \frac{1}{N} XX^T \]

Step 5 Eigenvalue Decomposition - Compute the eigenvalue decomposition of the covariance

\[ S = U \Lambda U^T \]

where \( \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_D) \) and \( \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_D \)

Eigenvalue decomposition can be obtained using standard vector algebra or numerical routines (e.g. Singular Value Decomposition (SVD))
Practical PCA (III)

Step 6  **Model Selection** - Select $D' < D$ projection directions, associated to the first $D'$ eigenvalues, so as to maximize the amount of variance retained in the projection

$$W = U_{D'} \begin{bmatrix} u_1 & \cdots & u_{D'} \\ \vdots & \ddots & \vdots \\ \vdots & & \vdots \end{bmatrix} \in \mathbb{R}^D \times \mathbb{R}^{D'}$$

Step 7  **Encoding** - Transform the normalized data $\overline{X}$ by projecting it onto the $D'$ principal components

$$Y = W^T \overline{X}$$

where $Y \in \mathbb{R}^{D'} \times \mathbb{R}^N$ is a compressed representation of the input data
Principal Components Data projected in the principal components’ plane
Given $N'$ new input samples in $X' \in \mathbb{R}^D \times \mathbb{R}^{N'}$ they can be projected into the reduced space by

1. Subtracting the means

$$\bar{X}' = X' - \bar{x}$$

2. Projecting onto the known principal components

$$Y' = W^T \bar{X}'$$
Each sample $x_n \in \mathbb{R}^D$ is a face picture with $D$ pixels.

The value of the $d$-th feature $x_n(d)$ is the intensity level of the corresponding pixel.

Application Example - Eigenfaces (II)

What is a principal component? Clearly, an eigenface $u_k$

Eigenvectors can be shown as images depicting primitive facial features
Application Example - Eigenfaces (III)

We can easily visualize the reconstruction of an image projected onto its eigenfaces

\[ \tilde{x}_n = W W^T x_n \]

\[ x_n \]

D’ = 10  
D’ = 20  
D’ = 30  

D’ = 50  
D’ = 100  
D’ = 200
How Many Principal Components?

Eigenvalues measure the fraction of variance captured by the projection.

Can be used to define a distortion measure.

- Suppose we have selected $K < D$ principal components.
- The resulting distortion is

$$J = \sum_{k=K+1}^{D} \lambda_k$$

that is the proportion of variance neglected by the projection.
Is Variance so much Informative?
Linear Discriminant Analysis

Adding *supervised* class information into the projection function

- Linear Discriminant Analysis (LDA)
- Perform dimensionality reduction while preserving as much of the *class discriminatory information* as possible

- Maximum separation between the means of the projection
- Minimum variance within each projected class
To solve this problem either you un-fold the roll (manifold approaches) or you change the data representation (kernel methods)
Nonlinear Feature Extraction

- **Signal Representation (Unsupervised)**
  - Manifold learning algorithms: e.g. ISOMAP
  - **Kernel** Principal Component Analysis (KPCA)

- **Classification (Supervised)**
  - **Kernel** Discriminant Analysis (KDA)
  - **Kernel** Canonical Correlation Analysis (KCCA)

**Kernels** allow to use a linear model for a nonlinear problem

A kernel **induces a new space** by means of a **non-linear mapping**, where the original linear operations can be performed.

*E.g. KPCA performs a linear PCA in the space created by the kernel rather than in the original data space.*
Take-home Messages

- **Exploratory data analysis**
  - Find new patterns in data
  - Formulate new hypotheses

- **Two key concepts**
  - *Curse of dimensionality* - Intractable problems
  - *Intrinsic dimension* - Data lies in lower dimensional space

- **Dimensionality Reduction**
  - Feature Extraction - Create new features by combining input data
  - Feature Selection - Extract a subset of informative input dimension

- **Linear feature extraction** \( \Rightarrow y = Wx \)
  - Models differentiate by the criteria used to chose \( W \)
Wrapping up PCA...

- PCA is a linear transformation
  - Defined by the matrix of eigenvectors \( W \) of data covariance \( S \)
  - Preserves as much variance as possible, measured by the eigenvalues \( \Lambda \)
- A general linear transformation produces a rotation, translation and scaling of the space
  - PCA rotates the data so that is maximally decorrelated (orthonormal principal components)
- PCA is linear
  - It cannot fit well curved surfaces
  - Nonlinear models
- PCA does not account for class information
  - Supervised models (LDA)