Unsupervised Analysis: Dimension Reduction

Why Dimension Reduction?

For Big-Data:
- Data visualization becomes very difficult! (Cannot draw 2D scatterplots between all pairs of features).
- Big-Data often has a high degrees of redundancy. (i.e. correlation among features).
- Many features may be uninformative for the particular problem under study (noise features).
- Dimension reduction ideally allows us retain information on most important features of the data, while reducing noise and simplifying visualization & analysis.

What is Dimension Reduction?

- Map the data into a new low-dimensional space where important characteristics of the data are preserved.
- The new space often gives a (linear or non-linear) transformation of the original data.
- Visualization and analysis (clustering/prediction/...) is then performed in the new space.
- In many cases, (especially for non-linear transformations) interpretation becomes difficult.

Principal Components Analysis (PCA)
PCA

Set-up:
- Data matrix: $X_{n \times p}$. $n$ observations and $p$ features.

Idea:
- Not all $p$ features are needed (much redundant info).
- Find low-dimensional representations that capture most of the variation in the data.

Uses:
- Ubiquitously used - Dimension reduction, data visualization, pattern recognition, exploratory analysis, etc.
- Best linear dimension reduction possible.

PCA - Main Idea

Question: What is a good 1D representation of the data?

Some Possibilities:
- Use one of the variables (e.g. $x_1$).
- Better idea: use a linear combination of the variables (i.e. a weighted average).

$$z_1 = v_1 x_1 + v_2 x_2 = v^T X$$

How to choose the weights ($v_1$ and $v_2$)?
PCA - Main Idea

Find line that maximizes the variance of the data projected onto the line:

Subsequent components orthogonal (perpendicular).

PCA minimizes orthogonal projection onto line: \[ Z = v_1 x_1 + v_2 x_2. \]
- Slope of line = \( v_2 / v_1 \) (if features centered).
- Note: Not same as OLS which minimizes projection of \( y \) onto \( x \)!

3D Projection onto a Hyperplane:
PCA - Criterion

PCA Criterion - PC 1 (Population):

\[
\begin{align*}
\text{maximize} & \quad \operatorname{Var}(Xv) \quad \text{subject to} \quad \|v\|_2 = 1 \\
\text{maximize} & \quad v^T \operatorname{Var}(X)v \quad \text{subject to} \quad \|v\|_2 = 1 \\
\text{maximize} & \quad v^T \Sigma v \quad \text{subject to} \quad \|v\|_2 = 1
\end{align*}
\]

where \( \Sigma = \operatorname{Cov}(X) \).

- Finds linear combination of features that maximizes the variance.

PCA - Criterion

PCA Criterion - Sample Version:

\[
\begin{align*}
\text{maximize} & \quad v_1^T X^T X v_k \quad \text{subject to} \quad \|v_k\|_2 = 1 \& \ v_k^T v_j = 0 \ \forall \ j < k
\end{align*}
\]

Replaces \( \Sigma \) with estimate \( X^T X / n \).

Solution: Eigenvalue decomposition of \( X^T X \). (eigen() in R)

PCA - Criterion

PCA Criterion - PC \( k \) (Population):

\[
\begin{align*}
\text{maximize} & \quad v_k^T \Sigma v_k \quad \text{subject to} \quad \|v_k\|_2 = 1 \& \ v_k^T v_j = 0 \ \forall \ j < k
\end{align*}
\]

- Subsequent linear combinations are orthogonal to previous combinations.
- Uncorrelated.

Equivalent PCA Criterion:

\[
\begin{align*}
\text{maximize} & \quad u_k^T X v_k \quad \text{subject to} \quad \|v_k\|_2 = 1 \& \ v_k^T v_j = 0 \ \forall \ j < k \\
\text{maximize} & \quad u_k^T X u_k \quad \text{subject to} \quad \|u_k\|_2 = 1 \& \ u_k^T u_j = 0 \ \forall \ j < k
\end{align*}
\]

- Finds left and right projection that maximize variance.

Solution: Singular Value Decomposition (SVD) of \( X \). (svd() in R)
PCA - Parts of the Solution

SVD: $X_{n \times p} = U_{n \times n} D_{n \times p} V^T_{p \times p}$

- Singular vectors: (left) $U$ and (right) $V$.
  - Orthonormal: $U^T U = I$ and $V^T V = I$.
- Singular values: Diagonals of $D$.
  - $d_1 \geq d_2 \geq \ldots \geq d_r$ where $r = \text{rank}(X)$.

SVD Solution to PCA:
- PCs: $Z = X V$ or $Z = U D$. ($U$ are un-scaled PCs).
  - $z_k = X v_k$ is $k^{th}$ PC.
  - $z_1, \ldots, z_K$ gives best $K$-dimensional projection of the data.
- PC Loadings: $V$.
  - $v_k$ is $k^{th}$ PC loading (feature weights).

PCA - Pattern Recognition

- $u_1$ - first column of $U$ encodes first major pattern in observation space.
- $v_1$ - first column of $V$ encodes the associated first pattern in feature space.
- $d_1$ gives strength of first pattern.
- Subsequent patterns are uncorrelated to first pattern (i.e. orthogonal).
- $X \approx \sum_{k=1}^{K} d_k u_k v_k^T$ - data is comprised of a series of patterns.

PCA - Properties

- Unique.
  - $U$ and $V$ unique up to a sign change.
  - $D$ unique.
- Global Solution.
PCA - Pattern Recognition

Patterns in feature space:

PCA - Data Visualization

PC Scatterplots:
- Problem: Can’t visualize
- Solution: Plot $u_1$ vs. $u_2$ and so forth.
- Advantages:
  - Dramatically reduces number of 2D scatterplots to visualize.
  - Focuses on patterns with most variance.

PC Loadings Plots:
- Scatterplots of $v_1$ vs. $v_2$.
- Visualizations of $v_k$.

Biplot:
- Scatterplot of PC 1 vs. PC 2 with loadings of $v_1$ vs. $v_2$ overlaid.
(See demo examples.)

PCA - Data Visualization

Scatterplots:

- Plotting Scatterplot PCs roughly equivalent to rotating axes of original plot.

PCA - Dimension Reduction

Best low-rank approximation to the data:

$$\min_{\hat{X}} \| X - \hat{X} \|_F^2 \quad \text{subject to } \text{rank}(\hat{X}) = K$$

Solution: $\hat{X} = \sum_{k=1}^{K} d_k u_k v_k^T$ - SVD / PCA solution!
- PCA also finds best data compression to minimize reconstruction error.
- PCA yields best linear dimension reduction possible!
PCA - Dimension Reduction

How much variance is explained? (i.e. extent of dimension reduction)

- Variance explained by \( k^{th} \) PC:
  \[ d_k^2 = v_k^T X^T X v_k. \]
- Total variance of data:
  \[ \sum_{k=1}^{n} d_k^2. \]
- Proportion of variance explained by \( k^{th} \) PC:
  \[ d_k^2 / \sum_{k=1}^{n} d_k^2. \]
- Cumulative variance explained by first \( r \) PCs:
  \[ \sum_{k=1}^{r} d_k^2 / \sum_{k=1}^{n} d_k^2. \]

(Extent of dimension reduction achieved by first \( r \) PC projections.)

---

PCA - Dimension Reduction

How to choose \( K \)?

- Elbow in screeplot.
- Take \( K \) that explains at least 90% (95%, 99%, etc.) variance.
- More sophisticated:
  - Cross-Validation done internally.
  - Validation via matrix completion.
  - Nuclear norm penalties.

---

PCA - Center and Scale?

- Typically, one should center features (i.e. columns of \( X \)).
  - Maximizing variance interpretation (assumes multivariate Gaussian model).
- Scaling changes PCA solution.
  - Features with large scale contribute more to variance, have large PC loadings.

General Suggestions:
- Scale if features measured differently. (Example - US college data).
- Don’t scale if features measured in same way & scale has meaning. (Example - gene expression data).

---

Screeplot:
PCA - Applications

“EigenImages” or “EigenFaces”

![PCA Image]

PCA - Summary

Strengths:
- Best linear dimension reduction.
- Ordered / orthogonal components.
- Unique, global solution.
- Others?

Weaknesses:
- Non-linear patterns.
- Ultra-high-dimensional settings \( p \gg n \)
- Others?

Sparse PCA

Motivation:
- When \( p \gg n \), many features irrelevant.
- PCA can perform poorly.

Idea:
- Sparsity in \( V \): zero out irrelevant features from PC loadings.
- Advantage: Find important features that contribute to major patterns in the data.

How?
- Typically, optimize PCA criterion with sparsity-encouraging penalty of \( V \).
- Many methods - active area of research!
In R: SPC in PMA package.
Functional PCA

Motivation:
- Times series, ordered data, spatial data.

Idea:
- Want PC loadings to be smooth (vary continuously) over time or space.
- Advantage: Improve interpretation.

How?
- Typically, optimize PCA criterion with a penalty that encourages smoothness of $V$ over time or space.
- Many methods for both functional data (data in the form of curves) and discretely-sampled functional data (e.g. discrete time points or specific locations).

In R: package fpca.

Kernel PCA

- Data set $\{x_i\}, i = 1, \ldots, n$
- Consider a nonlinear transformation $h(x)$. We can perform standard PCA in the feature space, which implicitly defines a nonlinear principal component model in the original data space.
- For the moment, suppose $\sum_{i=1}^n h(x_i) = 0$.
- The “sample covariance” matrix in the feature space

$$
\Sigma = \frac{1}{n} \sum_{i=1}^n h(x_i)h^T(x_i)
$$

The eigen equations become

$$
\frac{1}{n} \sum_{i=1}^n h(x_i)\left( h^T(x_i)v_j \right) = \lambda_j v_j
$$

Notice that $v_j$ is a linear combination of the $h(x_i)$, so it can be written in the form

$$
v_j = \sum_{i=1}^n \alpha_{ji} h(x_i)
$$

The eigen equations become

$$
K^2 \alpha_j = \lambda_j nK\alpha_j
$$

where $K$ is the $n \times n$ matrix with $K(x_i, x_j) = \langle h(x_i), h(x_j) \rangle$. We can reduce the equation to

$$
K\alpha_j = \lambda_j n\alpha_j
$$

The normalization condition is

$$
\lambda_j n\alpha_j^T \alpha_j = 1
$$

The resulting principal component projection is

$$
z_j(x) = h^T(x) v_j = \sum_{i=1}^n \alpha_{ji}K(x, x_i)
$$

In general, $h(x_i)$ may not have zero mean, we use

$$
\tilde{K} = K - 1K - K1 + 1K
$$

where $1$ is the $n \times n$ matrix in which every element is $1/n$.  

Genevera I. Allen and Yufeng Liu ()  Unsupervised Learning  July 2015 35 / 1
Supervised Dimension Reduction

Partial Least Squares:
- Best dimension reduction of cross-covariance between $X$ and $Y$ such that factors are orthogonal to $X$.

Canonical Correlations Analysis:
- Best dimension reduction of cross-covariance between $X$ and $Y$ such that bi-projection is orthogonal to $X$ or $Y$.

Linear Discriminant Analysis (classification):
- Best dimension reduction of between class covariance matrix relative to within class covariance.

Non-Negative Matrix Factorization (NMF)

Idea: $X_{n \times p} \approx W_{n \times K} H_{K \times p}$ with $K << p$.
- $X_{ij} \geq 0$ - non-negative data matrix.
- $W_{ik} \geq 0$ - non-negative observation factors; often sparse (Basis Factors).
- $H_{kj} \geq 0$ - non-negative feature factors; often sparse (Mixture Factors).

Like PCA except finds patterns with same direction of correlation.
NMF Interpretation

Topic Modeling:
- \( X \) a matrix of news articles (rows) by words (columns) whose entries are word counts.
  - \( X \approx \sum_{k=1}^{K} W_k H_k \) - sum of topics.
  - \( X_{ij} = W_i^T H_j^T = \sum_{k=1}^{K} W_{ik} H_{kj} \).
- Topic \( k \): Outer-product of \( k^{th} \) column of \( W \) (\( W_{.:k} \)) and \( k^{th} \) row of \( H \) (\( H_{k:.} \)).
  - E.g. Gay marriage.
- \( H_{k:.} \): non-zeros- words contributing to topic \( k \).
  - E.g. marriage, gay, Supreme, Court, district, equal, etc.
- \( W_{.:k} \) non-zeros - news articles belonging to topic \( k \).

NMF Criterion - Continuous Data

\[
\minimize_{W,H} \| X - WH \|^2_F \\
\text{subject to } W_{ik} \geq 0 & H_{kj} \geq 0
\]

(PCA criterion except with non-negativity constraints.)
Algorithm Updates: (Alternating Non-negative Least Squares)
\[
\hat{W} = \left( X H^T (H^T H)^{-1} \right)_+ \\
\hat{H} = \left( (W^T W)^{-1} W^T X \right)_+
\]

Local Solution.

NMF Criterion - Count Data

\[
\minimize_{W,H} \sum_{i=1}^{n} \sum_{j=1}^{p} [X_{ij} \log(W_{ij} H_{ij}) - W_{ij} H_{ij}]
\]

subject to \( W_{ik} \geq 0 & H_{kj} \geq 0 \)

Algorithm Updates:
\[
\hat{W}_{ik} = W_{ik} \left( \frac{\sum_{j=1}^{p} \hat{H}_{kj} X_{ij} / \hat{W}_{ij}^T \hat{H}_j}{\sum_{j=1}^{p} \hat{H}_{kj}} \right) \\
\hat{H}_{kj} = H_{kj} \left( \frac{\sum_{i=1}^{n} W_{ik} X_{ij} / \hat{W}_{ij}^T \hat{H}_j}{\sum_{i=1}^{n} W_{ik}} \right)
\]

Local solution.

NMF - Uses

1. Dimension Reduction / Pattern Recognition.
   - Similar to PCA (e.g. component scatterplots) except that patterns of correlation found in the same direction.
2. Archetypal Analysis.
   - Caricatures (segments; contrastive categorization) vs. Prototypes (averages).
   - Discussed Next Lecture!
PCA vs. NMF

Similarities:
- Linear Dimension Reduction.
- Interpretation.

Differences:
- Factors are unordered.
- Factors NOT orthogonal.
- Changing $K$ can fundamentally change factors.
- Non-unique, non-global solution.
- Depends on initialization. (Run several times and take the best).

Choosing $K$

Choice depends on goal:
- Dimension Reduction:
  - Residual sums of squares (or dispersion) - Screeplot.
- Clustering:
  - Consensus, silhouette, etc. (Discussed next lecture!).
- Archetypal Analysis:
  - Sparsity, factor purity, etc.
NMF - Summary

Strengths:
- Interpretation (often more appealing than PCA!).
- Applications - Clustering & Archetypal Analysis.
- Pattern Recognition.
- Others?

Weaknesses:
- Local solutions that depend strongly on $K$.
- Others?
- In R: NMF package.

ICA

Pre-processing Step: Reduce $X_{n \times p}$ to $\tilde{X}_{K \times p}$ with $K < n \neq$ independent sources. (Typically via PCA!)

Idea: $\tilde{X}_{K \times p} \approx A_{K \times K} S_{K \times p}$.
- Assumption: $\tilde{X}$ a matrix of $K$ scrambled independent signals.
- $A_{K \times K}$ Mixing Matrix - denotes how signals are scrambled to form sources in data.
- $S_{K \times p}$ Signal Matrix - each row of $S$ is an independent signal.

PCA finds uncorrelated, but not independent signals.

Independent Components Analysis (ICA)

ICA Uses

   - Assume $K$ independent signals got scrambled, but record $K$ scrambled versions of the signal.
   - Cocktail Party Problem.

   http://research.ics.aalto.fi/ica/cocktail/cocktail_en.cgi

2. Denoising.
   - Noise - independent from true signals.
ICA vs. PCA

Blind Source Separation:

Source Signals  
\[
\begin{array}{l}
\text{PCA Solution} \\
\end{array}
\]

\[
\begin{array}{l}
\text{ICA Solution} \\
\end{array}
\]

Measured Signals

FIGURE 14.37. Illustration of ICA vs. PCA on artificial time-series data. The upper left panel shows the two source signals, measured at 1000 uniformly spaced time points. The upper right panel shows the observed mixed signals. The lower two panels show the principal components and independent component solutions.

ICA Algorithms

Fast ICA:
- Finds rotations of \( X \) that are “non-Gaussian”.
- Uses non-Gaussian contrast functions:
  - \( g(x) = x^4 \).
  - \( g(x) = \tanh(x) \).
- Generalization of projection pursuit.

Others:
- Infomax (entropy).

Not Statistically Independent!

PCA vs. ICA

Similarities:
- Linear Dimension Reduction.
- Interpretation.

Differences:
- Factors are unordered.
- Factors NOT invariant - same solution by applying a permutation.
- Factors NOT orthogonal.
- Changing \( K \) can fundamentally change factors.
- Non-unique.
- No optimization criterion to evaluate solution.

ICA Applications - EEG

FIGURE 14.41. Fifteen seconds of EEG data (of 1917 seconds) at nine (of 100) scalp channels (top panel), as well as nine ICA components (lower panel). While nearby electrodes record nearly identical mixtures of brain and non-brain activity, ICA components are temporally distinct. The colored scalps represent the ICA unmixing coefficients \( \hat{a}_j \) as a heatmap, showing brain or scalp location of the source.
ICA Summary

Strengths:
- Interpretation.
- Applications - Blind Source Separation & Denoising.
- Others?

Weaknesses:
- Solutions that depend strongly on $K$.
- Solutions can be rotated.
- Others?
In R: fastICA package.
Multidimensional Scaling (MDS)

Idea:
- Visually represent proximities (similarities or distances) between objects in a lower dimensional space.
- Input: Matrix of similarities or dissimilarities, $D_{n \times n}$ (don't need the data itself!).
- Goal: Find projections ($z_1, \ldots, z_K$ where $z \in \mathbb{R}^n$) that preserve original distances in $D$ in a lower dimensional space ($K << n$).
- Distances preserved by optimizing a stress function.
- Non-linear dimension reduction.

MDS - Example

Consider the distances between nine American cities:

<table>
<thead>
<tr>
<th></th>
<th>BOS</th>
<th>CHI</th>
<th>DC</th>
<th>DEN</th>
<th>LA</th>
<th>MIA</th>
<th>NY</th>
<th>SEA</th>
<th>SF</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOS</td>
<td>0</td>
<td>963</td>
<td>429</td>
<td>1949</td>
<td>2979</td>
<td>1504</td>
<td>206</td>
<td>2976</td>
<td>3095</td>
</tr>
<tr>
<td>CHI</td>
<td>963</td>
<td>0</td>
<td>671</td>
<td>1616</td>
<td>2054</td>
<td>1329</td>
<td>802</td>
<td>2013</td>
<td>2142</td>
</tr>
<tr>
<td>DC</td>
<td>429</td>
<td>671</td>
<td>0</td>
<td>1616</td>
<td>2631</td>
<td>1329</td>
<td>802</td>
<td>2013</td>
<td>2142</td>
</tr>
<tr>
<td>DEN</td>
<td>1949</td>
<td>1616</td>
<td>0</td>
<td>1616</td>
<td>2631</td>
<td>1329</td>
<td>802</td>
<td>2013</td>
<td>2142</td>
</tr>
<tr>
<td>LA</td>
<td>2979</td>
<td>2054</td>
<td>2631</td>
<td>0</td>
<td>2687</td>
<td>2786</td>
<td>1331</td>
<td>379</td>
<td>1235</td>
</tr>
<tr>
<td>MIA</td>
<td>1504</td>
<td>1329</td>
<td>2037</td>
<td>2687</td>
<td>0</td>
<td>1308</td>
<td>3273</td>
<td>3053</td>
<td>1235</td>
</tr>
<tr>
<td>NY</td>
<td>206</td>
<td>802</td>
<td>233</td>
<td>1771</td>
<td>2786</td>
<td>1308</td>
<td>0</td>
<td>2815</td>
<td>2934</td>
</tr>
<tr>
<td>SEA</td>
<td>2976</td>
<td>2013</td>
<td>2684</td>
<td>1331</td>
<td>3273</td>
<td>2815</td>
<td>0</td>
<td>808</td>
<td>0</td>
</tr>
<tr>
<td>SF</td>
<td>3095</td>
<td>2142</td>
<td>2786</td>
<td>3273</td>
<td>3053</td>
<td>2815</td>
<td>808</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Can we represent these cities in a 2D space like a map?

MDS - Example

```
cmdscale(cities)
```

![MDS Example](image)

Flip the sign (MDS solution can be flipped or rotated.)
MDS - Stress Functions

- Input: \( D_{n \times n} : d_{ii'} \) denotes distance between object \( i \) and \( i' \).
- Output: Projections, \( z_1, \ldots, z_k \), \( z_k \in \mathbb{R}^n \), that preserve distances.

Stress Functions:

- Least squares or Kruskal-Shephard Scaling:
  \[
  S_D(z_1, z_2, \ldots, z_K) = \sqrt{\sum_{i \neq i'} (d_{ii'} - \|z_i - z_i'\|)^2}.
  \]

- Sammon mapping: preserve smaller pairwise distances
  \[
  \sum_{i \neq i'} \frac{(d_{ii'} - \|z_i - z_i'\|^2)}{d_{ii'}}.
  \]

- Shepard-Kruskal nonmetric scaling (\( \theta(\cdot) \): an increasing function):
  \[
  \sum_{i \neq i'} \left[ \theta(\|z_i - z_i'\|) - d_{ii'} \right]^2 \\sum_{i \neq i'} d_{ii'}^2.
  \]

MDS - Properties

- Data not needed - only dissimilarities.
- Algorithm - gradient descent.
- Choosing \( K \):
  - Scree plot (like PCA).
  - Shepard Diagram - plot proximities against distances in \( Z \).

- Interpreting MDS maps:
  - Axes and orientation arbitrary.
  - Can be rotated.
  - Only relative locations important.
  - Typically looks for objects close in the MDS map.
MDS vs. PCA

Similarities:
- Dimension reduction for visualization.

Differences:
- Non-linear vs. Linear.
- Local solution & arbitrary map.
- Non-unique & local solution.

MDS - Summary

Strengths:
- Visualizing proximities.
- Only need dissimilarities.
- Others?

Weaknesses:
- Arbitrary maps.
- Which stress function?
- High-dimensional settings? ($p >> n$ - more features than objects)
- Others?

In R: `dist; cmdscale` - classical MDS; `isoMDS` - Kruskals’s MDS and `sammon` in MASS package.

Dimension Reduction Wrap-Up

Techniques Covered:
- PCA.
- NMF.
- ICA.
- MDS.

Dimension Reduction Wrap-Up

Comparative Strengths & Weaknesses:

<table>
<thead>
<tr>
<th>Property</th>
<th>PCA</th>
<th>NMF</th>
<th>ICA</th>
<th>MDS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Textbooks:
- Elements of Statistical Learning by Hastie, Tibshirani & Friedman. 
Some of the figures in this presentation are taken from this textbook with permission from the authors.