Overview

• Motivation for dimensionality reduction
• Feature selection
  • Wrappers
  • Filters
  • Embedded methods
• Feature transformation
  • Principal Component Analysis (PCA)
  • Autoencoders
  • Clustering
Overview

• Motivation for dimensionality reduction
• Feature selection
  • Wrappers
  • Filters
  • Embedded methods
• Feature transformation
  • Principal Component Analysis (PCA)
  • Autoencoders
  • Clustering
Motivation for dimensionality reduction

Examples of large feature spaces
Predicting recurrence of lung cancer

Only a few genes actually matter!

Need small, interpretable subset to help doctors!
Motivation for dimensionality reduction

Examples of large feature spaces

Text classification

```
<REUTERS TOPICS="YES" INDEX="TRUTH"
CHISPLIT="TRAINING-SET" OLDID="12901" NEWID="798"
<DATE>2-MAR-1987 16:11:43.12</DATE>
<TOPICS><DD>livestock</DD><DD>boq</DD></TOPICS>
<TITLE>AMERICAN PORK CONGRESS KICKS OFF TOMORROW</TITLE>
<DATELINE>CHICAGO, March 2 - </DATELINE><BODY>The American Pork Congress kicks off tomorrow, March 3, in Indianapolis with 140 of the nations pork producers from 44 member states determining industry positions on a number of issues, according to the National Pork Producers Council, NPPC. Delegates to the three day Congress will be considering 26 resolutions concerning various issues, including the future direction of farm policy and the tax law as it applies to the agriculture sector. The delegates will also debate whether to endorse concepts of a national FVR (pseudorabies virus) control and eradication program, the NPPC said. A large trade show, in conjunction with the congress, will feature the latest in technology in all areas of the industry, the NPPC added. Reuter
\n\n</BODY></TEXT></REUTERS>
```

“Bag-of-Words” representation:

\[ x = \{0, 3, 0, 0, 1, \ldots, 2, 3, 0, 0, 0, 1\} \]

one entry per word!

Easily 50,000 words! Very sparse - easy to overfit!
What is curse of dimensionality

Number of cells grows exponentially as dimensionality increases

- Large number of cells, even if $D$ is moderately large
- So to cover the whole space reasonably well, you need exponentially number of training data points
Dimensionality Reduction

Broader question

• How can we detect low dimensional structure in high dimensional data?

Motivations

• Exploratory data analysis & visualization: you can plot data now
• Compact representation: small memory/computational footprint, lossy data compression
• Robust statistical modeling: curse of dimensionality
Dimensionality Reduction

General rules of dimensionality reduction

- **Relevant** features: the features that we need to perform well
- **Irrelevant** features: the features that are unnecessary
- **Redundant** features: the features that become irrelevant in the presence of others
Overview

• Motivation for dimensionality reduction
• Feature selection
  • Wrappers
  • Filters
  • Embedded methods
• Feature transformation
  • Principal Component Analysis (PCA)
  • Autoencoders
  • Clustering
Feature selection

Wrappers

• Rely on a feature search strategy to find an “optimal” subset of features based on the performance of the classifier

• Pros
  • High accuracy
  • Specific to the classifier of interest

• Cons
  • Computationally expensive
Feature selection

Wrappers - Possible feature search strategies

• Exhaustive search
  • Try all possible feature combinations
  • $M$ features $\rightarrow 2^M$ possible subsets

• Sequential forward selection
  • Greedy incremental selection of best performing features

• Recursive backward elimination
  • Starting from the full feature set, greedy selection of features which hurt performance

• Genetic algorithms
  • Random selection of features
  • Update of feature selection probabilities based on performance metrics
Feature selection

Wrappers: Sequential Feature Selection

• Cost is $M + (M - 1) + \ldots + 1 = \frac{M(M+1)}{2}$, instead of $2^M$
Overview

• Motivation for dimensionality reduction
• Feature selection
  • Wrappers
  • Filters
  • Embedded methods
• Feature transformation
  • Principal Component Analysis (PCA)
  • Autoencoders
  • Clustering
Feature selection

Filters

• We only pick the most informative features for the outcome
• We do not run a machine learning model
• We rank features according to their information and choose a cut-off point

• Pros
  • Computationally cheap

• Cons
  • No feature interaction is taken into account
  • Machine learning model is not taken into account

<table>
<thead>
<tr>
<th>$k$</th>
<th>$J(X_k)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>35</td>
<td>0.846</td>
</tr>
<tr>
<td>42</td>
<td>0.811</td>
</tr>
<tr>
<td>10</td>
<td>0.810</td>
</tr>
<tr>
<td>654</td>
<td>0.611</td>
</tr>
<tr>
<td>22</td>
<td>0.443</td>
</tr>
<tr>
<td>59</td>
<td>0.388</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>212</td>
<td>0.09</td>
</tr>
<tr>
<td>39</td>
<td>0.05</td>
</tr>
</tbody>
</table>
Feature selection

Filters - Possible feature evaluation metrics

- **Correlation** of feature $x_k$ with target variable $y$

$$
\rho(x_k, y) = \frac{\text{Cov}(x_k, y)}{\text{Var}(x_k) \text{Var}(y)}
$$

Measures linear dependencies

- **Mutual information** of feature $x_k$ with target variable $y$

$$
I(x_k, y) = \sum_i \sum_j P(x_k = i, y = j) \frac{P(x_k = i, y = j)}{P(x_k = i)P(y = j)}
$$

where $P$ is the probability estimate from the data

Assumes known probability distribution of the data.
Feature selection

Filters - Possible feature evaluation metrics

- **Fisher’s criterion** of feature $d$ of sample $x_n$ from class $k$

$$F(d) = \frac{\sum_k \sum_{x_n \in C_k} (x_{nd} - \mu_{kd})^2}{\sum_k (\mu_d - \mu_{kd})^2}$$

where $\mu_{kd}$ is the mean of feature $d$ from class $k$, and $\mu_d$ is the mean of feature $d$ from all samples.

Measures within-class scatter in relation to between-class scatter.
Feature selection

Filters

- A lot less expensive
Overview

• Motivation for dimensionality reduction
• Feature selection
  • Wrappers
  • Filters
  • Embedded methods
• Feature transformation
  • Principal Component Analysis (PCA)
  • Autoencoders
  • Clustering
Feature selection

Embedded methods

• The classifier performs feature selection as part of the learning procedure
• Regularization is a great example

\[ J(w) = EC(w) + \lambda \sum_{d=1}^{D} w_d^2 = EC(w) + \lambda ||w||_2^2 \]

• Pros
  • Feature selection is part of learning the procedure

• Cons
  • Computationally demanding
Overview

• Motivation for dimensionality reduction
• Feature selection
  • Wrappers
  • Filters
  • Embedded methods
• Feature transformation
  • Principal Component Analysis (PCA)
  • Autoencoders
  • Clustering
Feature transformation

Linear feature transformation

• $x \in \mathbb{R}^D \rightarrow y \in \mathbb{R}^M$, $D \gg M$

• linear transformation of original space: $y = U^T x$, $U \in \mathbb{R}^{D \times M}$
Feature transformation

Linear feature transformation: Example

Assuming an input vector $\mathbf{x} = [x_1, x_2] \in \mathbb{R}^2$ and a transformation $\mathbf{Ux}$, what transformation do each of the following matrices perform?

If $\mathbf{Ux} = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix}$, then $\mathbf{Ux} = \begin{bmatrix} u_{11}x_1 + u_{12}x_2 \\ u_{21}x_1 + u_{22}x_2 \end{bmatrix}$.

If $\mathbf{Ux} = \begin{bmatrix} u_{11} & u_{12} \end{bmatrix}$, then $\mathbf{Ux} = \begin{bmatrix} u_{11}x_1 + u_{12}x_2 \end{bmatrix}$.

1. $\mathbf{U} = [1, 0; 0, 1]$
2. $\mathbf{U} = [1.5, 0; 0, 1.5]$
3. $\mathbf{U} = [0, 1; 1, 0]$
4. $\mathbf{U} = [1, 0]$
5. $\mathbf{U} = [1, 1]$
6. $\mathbf{U} = [1, 1; 0, 1]$
Feature transformation

Linear feature transformation: Example

Assuming an input vector $\mathbf{x} = [x_1, x_2] \in \mathbb{R}^2$ and a transformation $U\mathbf{x}$, what transformation do each of the following matrices perform?

If $U\mathbf{x} = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix}$, then $U\mathbf{x} = \begin{bmatrix} u_{11}x_1 + u_{12}x_2 \\ u_{21}x_1 + u_{22}x_2 \end{bmatrix}$.

If $U\mathbf{x} = \begin{bmatrix} u_{11} & u_{12} \end{bmatrix}$, then $U\mathbf{x} = \begin{bmatrix} u_{11}x_1 + u_{12}x_2 \end{bmatrix}$.

1. $U = [1, 0; 0, 1]$ Identity
2. $U = [1.5, 0; 0, 1.5]$ Dilation
3. $U = [0, 1; 1, 0]$ Flipping of axes
4. $U = [1, 0]$ Preserving only first dimension
5. $U = [1, 1]$ Substituting the first dimension by the sum of the two. Removing the second dimension.
6. $U = [1, 1; 0, 1]$ Substituting the first dimension by the sum of the two. Preserving the second dimension.
**Principal Component Analysis (PCA): Representation**

- **Input:** Data \( \mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \), \( \mathbf{x}_n \in \mathbb{R}^D \), centered inputs
- **Output:** Projected data \( \{\mathbf{y}_1, \ldots, \mathbf{y}_N\} \), \( \mathbf{y}_n \in \mathbb{R}^M \), \( D \gg M \)
- **Projection into subspace:** \( \mathbf{U} \in \mathbb{R}^{D \times M} \)

\[
\mathbf{y}_n = \mathbf{U}^T \mathbf{x}_n , \quad \mathbf{U}^T \mathbf{U} = \mathbf{I}
\]

- **Evaluation metric:** many possible metrics yielding the same solution
  - **Derivation 1:** Maximize captured variance
  - **Derivation 2:** Minimize projection error
Covariance Matrix

For 2-dimensional samples $x_n = [x_{n1}, x_{n2}]^T$, we assume means $\mu_1$ and $\mu_2$ for dimensions 1 and 2.

$$\Sigma = \begin{bmatrix}
\sum_{n=1}^{N} (x_{n1} - \mu_1)^2 & \sum_{n=1}^{N} (x_{n1} - \mu_1)(x_{n2} - \mu_2) \\
\sum_{n=1}^{N} (x_{n1} - \mu_1)(x_{n2} - \mu_2) & \sum_{n=1}^{N} (x_{n2} - \mu_2)^2
\end{bmatrix}$$
Matrix Diagonalization

• Converting a square matrix into a special type of matrix, i.e. diagonal, which shares the same fundamental properties of the underlying matrix

• Eigen-decomposition theorem: A square matrix $A \in \mathbb{R}^{D \times D}$ can be decomposed into

$$A = P \Lambda P^{-1}$$

• $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_D)$: diagonal constructed from eigenvalues of $A$

• $P = \begin{bmatrix} e_1 & \ldots & e_D \end{bmatrix} \in \mathbb{R}^{M \times M}$: matrix decomposed from the eigenvectors of $A$
Matrix Diagonalization

- Assuming that we have a matrix equation $AX = Y$
- This can be written as $PΛP^{-1}X = Y$
- Multiplying both sides by $P^{-1}$ we get $ΛP^{-1}X = P^{-1}Y$
- The same linear transformation $P^{-1}$ is being applied to $X$ and $Y$, therefore we can transform the equation into another space $X' = P^{-1}X$ and $Y' = P^{-1}Y$
- Now the new system that we have to solve it $PX' = Y'$
- The most important advantage of this transformation is that it decorrelates the matrices (i.e., “canonicializes” the system), therefore some computations might be easier of less expensive
Principal Component Analysis (PCA): Optimization

• Compute covariance matrix

\[ S = \frac{1}{N} X^T X, \quad X = \begin{bmatrix} -\tilde{x}_1^T & - \\ \vdots & \vdots \\ -\tilde{x}_N^T & - \end{bmatrix} \]

\[ U \in \mathbb{R}^{D \times M} \]

• Diagonalize \( S \), i.e. compute eigenvalues and eigenvectors

\[ S = P \Lambda P^{-1}, \quad P = \begin{bmatrix} u_1 & \ldots & u_D \end{bmatrix} \in \mathbb{R}^{D \times D} \]

• Use the eigenvectors corresponding to the \( M \) largest eigenvalues

\[ U = \begin{bmatrix} u_1 & \ldots & u_M \end{bmatrix} \in \mathbb{R}^{D \times M} \]
Principal Component Analysis (PCA): Algorithm

- **Step 0:** Mean normalize input features
- **Step 1:** Compute covariance matrix \( S = \frac{1}{N} X^T X = \frac{1}{N} \sum_n x_n x_n^T \)
- **Step 2:** Diagonalize \( S \) and find eigenvector matrix \( P \)
- **Step 3:** Take the first \( M \ll D \) eigenvectors or *principal components* (corresponding to the \( M \) largest eigenvalues) and form reduced matrix \( U \)
- **Step 3:** Project data into reduced space: \( z_n = U^T x_n \)
Principal Component Analysis (PCA): Example
Principal Component Analysis (PCA)

Original Images

Eigenvectors

they look like blurred original images

Mean $\lambda_1 = 3.4 \cdot 10^5$ $\lambda_2 = 2.8 \cdot 10^5$ $\lambda_3 = 2.4 \cdot 10^5$ $\lambda_4 = 1.6 \cdot 10^5$

Used to centralize inputs
Principal Component Analysis (PCA)

How to determine the number of principal components $M$?
Plot eigenspectrum

$\sum_{d=1}^{M} \lambda_d \geq \text{threshold}$, where common choices are 95%, 99%

Overview

• Motivation for dimensionality reduction
• Feature selection
  • Wrappers
  • Filters
  • Embedded methods
• Feature transformation
  • Principal Component Analysis (PCA)
  • Autoencoders
  • Clustering
Autoencoders

- Unsupervised algorithm that tries to learn an approximation of the identity function $h_{W,b}(x) \approx x$
- The middle layer of the autoencoder can be used as the transformed feature set
Overview

- Motivation for dimensionality reduction
- Feature selection
  - Wrappers
  - Filters
  - Embedded methods
- Feature transformation
  - Principal Component Analysis (PCA)
  - Autoencoders
  - Clustering
Clustering

- Grouping of “similar” instances in the data sample
- Replacing a high dimensional data entry with a cluster label
- Deterministic clustering (e.g., K-Means) gives only one label per input
- Soft clustering gives probability of a sample belonging to each cluster
- More in the next lecture!
What have a learned so far

• Dimensionality reduction for visualization, compression, avoid curse of dimensionality
• Feature selection to select the most informative features
• Feature transformation to transform the features into a reduced space
• **Readings:** Alpaydin 6.1-6.3