Reducing Data Dimension

Required reading:
• “A Tutorial on PCA,’ J. Schlens, copy available on class website

Recommended reading:
• Wall et al., 2003

Machine Learning 10-701
November 14, 2006

Tom M. Mitchell
Machine Learning Department
Carnegie Mellon University
Outline

• Feature selection
  – Single feature scoring criteria
  – Search strategies

• Unsupervised dimension reduction using all features
  – Principle Components Analysis
  – Singular Value Decomposition
  – Independent components analysis

• Supervised dimension reduction
  – Fisher Linear Discriminant
  – Hidden layers of Neural Networks
Dimensionality Reduction

Why?

- Learning a target function from data where some features are irrelevant - reduce variance, improve accuracy
- Wish to visualize high dimensional data
- Sometimes have data whose “intrinsic” dimensionality is smaller than the number of features used to describe it - recover intrinsic dimension
Supervised Feature Selection
Supervised Feature Selection

Problem: Wish to learn $f: X \rightarrow Y$, where $X=\langle X_1, \ldots, X_N \rangle$

But suspect not all $X_i$ are relevant

Approach: Preprocess data to select only a subset of the $X_i$
  • Score each feature, or subsets of features
    – How?
  • Search for useful subset of features to represent data
    – How?
Scoring Individual Features $X_i$

Common scoring methods:

• Training or cross-validated accuracy of single-feature classifiers $f_i: X_i \rightarrow Y$

• Estimated mutual information between $X_i$ and $Y$:

$$\hat{I}(X_i, Y) = \sum_k \sum_y \hat{P}(X_i = k, Y = y) \log \frac{\hat{P}(X_i = k, Y = y)}{\hat{P}(X_i = k) \hat{P}(Y = y)}$$

• $\chi^2$ statistic to measure independence between $X_i$ and $Y$

• Domain specific criteria
  – Text: Score “stop” words (“the”, “of”, …) as zero
  – fMRI: Score voxel by T-test for activation versus rest condition
  – …
Choosing Set of Features to learn $F: X \rightarrow Y$

Common methods:

Forward1: Choose the $n$ features with the highest scores

Forward2:

- Choose single highest scoring feature $X_k$
- Rescore all features, conditioned on the set of already-selected features
  - E.g., $\text{Score}(X_i | X_k) = I(X_i, Y | X_k)$
  - E.g., $\text{Score}(X_i | X_k) = \text{Accuracy}(\text{predicting } Y \text{ from } X_i \text{ and } X_k)$
- Repeat, calculating new scores on each iteration, conditioning on set of selected features
Choosing Set of Features

Common methods:

Backward1: Start with all features, delete the n with lowest scores

Backward2: Start with all features, score each feature conditioned on assumption that all others are included. Then:
- Remove feature with the lowest (conditioned) score
- Rescore all features, conditioned on the new, reduced feature set
- Repeat
Feature Selection: Text Classification

Approximately $10^5$ words in English

[Rogati & Yang, 2002]

Figure 2: Top 3 feature selection methods for Reuters-21578 (Macro F1)

IG = information gain, chi = $\chi^2$, DF = doc frequency,
Impact of Feature Selection on Classification of fMRI Data

[ Pereira et al., 2005 ]

Table 1: Average accuracy across all pairs of categories, restricting the procedure to use a certain number of voxels for each subject. The highlighted line corresponds to the best mean accuracy, obtained using 300 voxels.

Voxels scored by p-value of regression to predict voxel value from the task
Summary: Supervised Feature Selection

Approach: Preprocess data to select only a subset of the $X_i$

- Score each feature
  - Mutual information, prediction accuracy, …

- Find useful subset of features based on their scores
  - Greedy addition of features to pool
  - Greedy deletion of features from pool
  - Considered independently, or in context of other selected features

Always do feature selection using training set only (not test set!)

- Often use nested cross-validation loop:
  - Outer loop to get unbiased estimate of final classifier accuracy
  - Inner loop to get unbiased feature scores for feature selection
Unsupervised Dimensionality Reduction
Unsupervised mapping to lower dimension

Differs from feature selection in two ways:

• Instead of choosing subset of features, create new features (dimensions) defined as functions over all features

• Don’t consider class labels, just the data points
Principle Components Analysis

• Idea:
  – Given data points in d-dimensional space, project into lower dimensional space while preserving as much information as possible
    • E.g., find best planar approximation to 3D data
    • E.g., find best planar approximation to $10^4$ D data
  – In particular, choose projection that minimizes the squared error in reconstructing original data
PCA: Find Projections to Minimize Reconstruction Error

Assume data is set of \( d \)-dimensional vectors, where \( n \)th vector is

\[
x^n = \langle x^n_1 \ldots x^n_d \rangle
\]

We can represent these in terms of any \( d \) orthogonal basis vectors

\[
x^n = \sum_{i=1}^{d} z^n_i u_i; \quad u^T_i u_j = \delta_{ij}
\]

PCA: given \( M < d \). Find \( \langle u_1 \ldots u_M \rangle \)

that minimizes \( E_M = \sum_{n=1}^{N} \|x^n - \hat{x}^n\|^2 \)

where \( \hat{x}^n = \bar{x} + \sum_{i=1}^{M} z^n_i u_i \)

\[
\bar{x} = \frac{1}{N} \sum_{n=1}^{N} x^n
\]
PCA: given $M<d$. Find $\langle u_1 \ldots u_M \rangle$

that minimizes $E_M \equiv \sum_{n=1}^{N} ||x^n - \hat{x}^n||^2$

where $\hat{x}^n = \bar{x} + \sum_{i=1}^{M} z_i^n u_i$

Note we get zero error if $M=d$, so all error is due to missing components.

Therefore,

$$E_M = \sum_{i=M+1}^{d} \sum_{n=1}^{N} [u_i^T (x^n - \bar{x})]^2$$

$$= \sum_{i=M+1}^{d} u_i^T \Sigma u_i$$

This minimized when $u_i$ is eigenvector of $\Sigma$, the covariance matrix of $X$. i.e., minimized when:

$$\Sigma u_i = \lambda_i u_i$$

Covariance matrix:

$$\Sigma = \sum_{n} (x^n - \bar{x})(x^n - \bar{x})^T$$
PCA

Minimize \( E_M = \sum_{i=M+1}^{d} \mathbf{u}_i^T \Sigma \mathbf{u}_i \)

\[ \rightarrow \sum \mathbf{u}_i = \lambda_i \mathbf{u}_i \]

\( \text{Eigenvector of } \Sigma \)

\( \text{Eigenvalue (scalar)} \)

\[ \rightarrow E_M = \sum_{i=M+1}^{d} \lambda_i \]

**PCA algorithm 1:**
1. \( X \leftarrow \) Create \( N \times d \) data matrix, with one row vector \( x^n \) per data point
2. \( X \leftarrow \) subtract mean \( \bar{x} \) from each row vector \( x^n \) in \( X \)
3. \( \Sigma \leftarrow \) covariance matrix of \( X \)
4. Find eigenvectors and eigenvalues of \( \Sigma \)
5. PC’s \( \leftarrow \) the \( M \) eigenvectors with largest eigenvalues
PCA Example

\[ \hat{x}^n = \bar{x} + \sum_{i=1}^{M} z_i^n u_i \]
PCA Example

\[ \hat{x}^n = \bar{x} + \sum_{i=1}^{M} z_i^n u_i \]

Reconstructed data using only first eigenvector (M=1)
Very Nice When Initial Dimension Not Too Big

What if very large dimensional data?
• e.g., Images \((d \geq 10^4)\)

Problem:
• Covariance matrix \(\Sigma\) is size \((d \times d)\)
• \(d=10^4 \rightarrow |\Sigma| = 10^8\)

Singular Value Decomposition (SVD) to the rescue!
• pretty efficient algs available, including Matlab SVD
• some implementations find just top \(N\) eigenvectors
SVD

Data $X$, one row per data point

$US$ gives coordinates of rows of $X$ in the space of principle components

$S$ is diagonal, $S_k > S_{k+1}$, $S_k^2$ is kth largest eigenvalue

Rows of $VT$ are unit length eigenvectors of $XTX$

If cols of $X$ have zero mean, then $XTX = c \Sigma$ and eigenvects are the Principle Components

[from Wall et al., 2003]
Singular Value Decomposition

To generate principle components:

• Subtract mean $\bar{x} = \frac{1}{N} \sum_{n=1}^{N} x^n$ from each data point, to create zero-centered data

• Create matrix $X$ with one row vector per (zero centered) data point

• Solve SVD: $X = USV^T$

• Output Principle components: columns of $V$ (= rows of $V^T$)
  – Eigenvectors in $V$ are sorted from largest to smallest eigenvalues
  – $S$ is diagonal, with $s_k^2$ giving eigenvalue for kth eigenvector
Singular Value Decomposition

To project a point (column vector $x$) into PC coordinates:

$$VTx$$

If $x_i$ is $i$th row of data matrix $X$, then

- $(i$th row of $US) = VTx_i^T$
- $(US)^T = VTXT$

To project a column vector $x$ to M dim Principle Components subspace, take just the first M coordinates of $VTx$
PCA Example

faces

Eigenfaces

Thanks to Christopher DeCoro
see http://www.cs.princeton.edu/~cdecoro/eigenfaces/
Reconstructing a face from the first $N$ components (eigenfaces)

In this next image, we show a similar picture, but with each additional face representing an additional 8 principle components. You can see that it takes a rather large number of images before the picture looks totally correct.
Independent Components Analysis

- PCA seeks directions $<Y_1 \ldots Y_M>$ in feature space $X$ that minimize reconstruction error.

- ICA seeks directions $<Y_1 \ldots Y_M>$ that are most statistically independent. I.e., that minimize $I(Y)$, the mutual information between the $Y_j$:

  $$I(Y) = \left[ \sum_{j=1}^{J} H(Y_j) \right] - H(Y)$$

  Which maximizes their departure from Gaussianity!
Supervised Dimensionality Reduction
1. Fisher Linear Discriminant

- A method for projecting data into lower dimension to hopefully improve classification

- We’ll consider 2-class case

Project data onto vector that connects class means?
Fisher Linear Discriminant

Project data onto one dimension, to help classification

\[ y = w^T x \]

Define class means: 

\[ m_i = \frac{1}{N_i} \sum_{n \in C_i} x^n \]

Could choose \( w \) according to: 

\[ \arg \max_w w^T (m_2 - m_1) \]

Instead, Fisher Linear Discriminant chooses: 

\[ \arg \max_w \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} \]

\[ m_i \equiv w^T m_i \quad s_i^2 \equiv \sum_{n \in C_i} (y^n - m_i)^2 \]
Fisher Linear Discriminant

Project data onto one dimension, to help classification

\[ y = w^T x \]

Fisher Linear Discriminant: \( \arg \max_w \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} \)

is solved by: \( w \propto S_W^{-1}(m_2 - m_1) \)

Where \( S_W \) is sum of within-class covariances:

\[ S_W \equiv \sum_{n \in C_1} (x^n - m_1)(x^n - m_1)^T + \sum_{n \in C_2} (x^n - m_2)(x^n - m_2)^T \]
Fisher Linear Discriminant

Fisher Linear Discriminant: \[ \arg \max_w \frac{(m_2 - m_1)^2}{s^2_1 + s^2_2} \]

Is equivalent to minimizing sum of squared error if we assume target values are not +1 and -1, but instead \( \frac{N}{N_1} \) and \( -\frac{N}{N_2} \)

Where \( N \) is total number of examples, \( N_i \) is number in class \( i \)

Also generalized to \( K \) classes (and projects data to \( K-1 \) dimensions)
Summary: Fisher Linear Discriminant

- Choose n-1 dimension projection for n-class classification problem
- Use within-class covariances to determine the projection
- Minimizes a different sum of squared error function (the projected within-class variances)
2. Hidden Layers in Neural Networks

When \# hidden units < \# inputs, hidden layer also performs dimensionality reduction.

Each synthesized dimension (each hidden unit) is logistic function of inputs

\[
h_k(x) = \frac{1}{1 + \exp(w_0 + \sum_{i=1}^{N} w_i x_i)}
\]

Hidden units defined by gradient descent to (locally) minimize squared output classification/regression error

\[
E = \sum_{n=1}^{N} \sum_{k} (\hat{y}_k(x^n) - y_k(x^n))^2
\]

Also allow networks with multiple hidden layers
→ highly nonlinear components (in contrast with linear subspace of Fisher LD, PCA)
A target function:

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>10000000</td>
<td>10000000</td>
</tr>
<tr>
<td>01000000</td>
<td>01000000</td>
</tr>
<tr>
<td>00100000</td>
<td>00100000</td>
</tr>
<tr>
<td>00010000</td>
<td>00010000</td>
</tr>
<tr>
<td>00001000</td>
<td>00001000</td>
</tr>
<tr>
<td>00000100</td>
<td>00000100</td>
</tr>
<tr>
<td>00000010</td>
<td>00000010</td>
</tr>
<tr>
<td>00000001</td>
<td>00000001</td>
</tr>
</tbody>
</table>

Can this be learned??
Learning Hidden Layer Representations

A network:

Learned hidden layer representation:

<table>
<thead>
<tr>
<th>Input Values</th>
<th>Hidden Values</th>
<th>Output Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>10000000</td>
<td>.89 .04 .08</td>
<td>10000000</td>
</tr>
<tr>
<td>01000000</td>
<td>.01 .11 .88</td>
<td>01000000</td>
</tr>
<tr>
<td>00100000</td>
<td>.01 .97 .27</td>
<td>00100000</td>
</tr>
<tr>
<td>00010000</td>
<td>.99 .97 .71</td>
<td>00010000</td>
</tr>
<tr>
<td>00001000</td>
<td>.03 .05 .02</td>
<td>00001000</td>
</tr>
<tr>
<td>00000100</td>
<td>.22 .99 .99</td>
<td>00000100</td>
</tr>
<tr>
<td>00000010</td>
<td>.80 .01 .98</td>
<td>00000010</td>
</tr>
<tr>
<td>00000001</td>
<td>.60 .94 .01</td>
<td>00000001</td>
</tr>
</tbody>
</table>
Neural Nets for Face Recognition

left  str  rght  up

30x32 inputs

Typical input images

90% accurate learning head pose, and recognizing 1-of-20 faces
Learned Hidden Unit Weights

left  str  rght  up

Learned Weights

30x32 inputs

Typical input images

http://www.cs.cmu.edu/~tom/faces.html
Semantic Memory Model Based on ANN’s

No hierarchy given.
Train with assertions, e.g., Can(Canary,Fly)
Humans act as though they have a hierarchical memory organization

1. Victims of Semantic Dementia progressively lose knowledge of objects. But they lose specific details first, general properties later, suggesting hierarchical memory.

2. Children appear to learn general categories and properties first, following the same hierarchy, top down.*

* some debate remains on this.
Memory deterioration follows semantic hierarchy


---

**a**

<table>
<thead>
<tr>
<th>Item</th>
<th>Sept. 91</th>
<th>March 92</th>
<th>March 93</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bird</td>
<td>+</td>
<td>+</td>
<td>Animal</td>
</tr>
<tr>
<td>Chicken</td>
<td>+</td>
<td>+</td>
<td>Animal</td>
</tr>
<tr>
<td>Duck</td>
<td>+</td>
<td>Bird</td>
<td>Dog</td>
</tr>
<tr>
<td>Swan</td>
<td>+</td>
<td>Bird</td>
<td>Animal</td>
</tr>
<tr>
<td>Eagle</td>
<td>Duck</td>
<td>Bird</td>
<td>Horse</td>
</tr>
<tr>
<td>Ostrich</td>
<td>Swan</td>
<td>Bird</td>
<td>Animal</td>
</tr>
<tr>
<td>Peacock</td>
<td>Duck</td>
<td>Bird</td>
<td>Vehicle</td>
</tr>
<tr>
<td>Penguin</td>
<td>Duck</td>
<td>Bird</td>
<td>Part of animal</td>
</tr>
<tr>
<td>Rooster</td>
<td>Chicken</td>
<td>Chicken</td>
<td>Dog</td>
</tr>
</tbody>
</table>

---

**b**

- Proportion omitted
- **DS**
- **DC**
- **IF**

---

**c** IF’s delayed copy of a camel

---

**d** DC’s delayed copy of a swan
The process of differentiation of conceptual representations. The representations are those seen in the feedforward network model shown in Fig. 3. a | Acquired patterns of activation that represent the eight objects in the training set at three points in the learning process (epochs 250, 750 and 2,500). Early in learning, the patterns are undifferentiated; the first difference to appear is between plants and animals. Later, the patterns show clear differentiation at both the superordinate (plant–animal) and intermediate (bird–fish/tree–flower) levels. Finally, the individual concepts are differentiated, but the overall hierarchical organization of the similarity structure remains. b | A standard hierarchical clustering analysis program has been used to visualize the similarity structure in the
ANN Also Models Progressive Deterioration


average effect of noise in inputs to hidden layers
What you should know

• Feature selection
  – Single feature scoring criteria
  – Search strategies
    • Common approaches: Greedy addition of features, or greedy deletion

• Unsupervised dimension reduction using all features
  – Principle Components Analysis
    • Minimize reconstruction error
  – Singular Value Decomposition
    • Efficient PCA
  – Independent components analysis

• Supervised dimension reduction
  – Fisher Linear Discriminant
    • Project to n-1 dimensions to discriminate n classes
  – Hidden layers of Neural Networks
    • Most flexible, local minima issues
Further Readings

LANL LA-UR-02-4001