Overview

→ Overview of the course

→ Classification, Clustering, and Dimension reduction

→ The “curse of dimensionality”
Course Outline

Instructor:

Tianwei Yu
Office: GCR Room 334
Email: tianwei.yu@emory.edu
Office Hours: by appointment.

Teaching Assistant:

Yunchuan Kong, Teng Fei, Yanting Huang
Office Hours: TBA

Course Website:

http://web1.sph.emory.edu/users/tyu8/534
Overview

Focus of the course:
- Classification
- Clustering
- Dimension reduction

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
</tr>
<tr>
<td>2</td>
<td>Python Q &amp; A by TAs</td>
</tr>
<tr>
<td>3</td>
<td>Statistical background</td>
</tr>
<tr>
<td>4</td>
<td>Stat decision theory 1</td>
</tr>
<tr>
<td>5</td>
<td>Stat decision theory 2</td>
</tr>
<tr>
<td>6</td>
<td>Density estimation and KNN</td>
</tr>
<tr>
<td>7</td>
<td>Basis expansion 1</td>
</tr>
<tr>
<td>8</td>
<td>Basis expansion 2</td>
</tr>
<tr>
<td>9</td>
<td>Linear Machine</td>
</tr>
<tr>
<td>10</td>
<td>Support Vector Machine 1</td>
</tr>
<tr>
<td>11</td>
<td>Support Vector Machine 2</td>
</tr>
<tr>
<td>12</td>
<td>Boosting</td>
</tr>
<tr>
<td>13</td>
<td>Decision Tree</td>
</tr>
<tr>
<td>14</td>
<td>Random Forest</td>
</tr>
<tr>
<td>15</td>
<td>Bump hunting and forward stagewise regression</td>
</tr>
<tr>
<td></td>
<td>Overview</td>
</tr>
<tr>
<td>---</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>16</td>
<td>Hidden Markov Model 1</td>
</tr>
<tr>
<td>17</td>
<td>Hidden Markov Model 2</td>
</tr>
<tr>
<td>18</td>
<td>Neural networks 1</td>
</tr>
<tr>
<td>19</td>
<td>Neural networks 2</td>
</tr>
<tr>
<td>20</td>
<td>Neural networks 3</td>
</tr>
<tr>
<td>21</td>
<td>Model generalization 1</td>
</tr>
<tr>
<td>22</td>
<td>Model generalization 2</td>
</tr>
<tr>
<td>23</td>
<td>Clustering 1</td>
</tr>
<tr>
<td>24</td>
<td>Clustering 2 &amp; EM algorithm</td>
</tr>
<tr>
<td>25</td>
<td>Clustering 3</td>
</tr>
<tr>
<td>26</td>
<td>Dimension reduction 1</td>
</tr>
<tr>
<td>27</td>
<td>Dimension reduction 2</td>
</tr>
<tr>
<td>28</td>
<td>Dimension reduction 3</td>
</tr>
</tbody>
</table>
References:

Textbook:

The elements of statistical learning. Hastie, Tibshirani & Friedman.


Other references:

Pattern classification. Duda, Hart & Stork.

Data clustering: theory, algorithms and application. Gan, Ma & Wu.

An introduction to Statistical Learning: with Applications in R. James, Witten, Hastie, Tibshirani.
References:

Python:


Evaluation:

Four homeworks/projects (20% each for the first 3, and 30% final project)

**Requirement:** complete in Python. Submit code with results.

Class participation evaluated by 4 quizzes (10%)
Overview

Machine Learning/Data mining

- Supervised learning
  - "direct data mining"
  - Classification
  - Estimation
  - Prediction

- Unsupervised learning
  - "indirect data mining"
  - Clustering
  - Association rules
  - Description, dimension reduction and visualization

Semi-supervised learning

Modified from Figure 1.1 from <Data Clustering> by Gan, Ma and Wu
In supervised learning, the problem is well-defined: Given a set of observations \( \{x_i, y_i\} \), estimate the density \( Pr(Y, X) \). Usually the goal is to find the model/parameters to minimize a loss,

\[
L(Y, f(X))
\]

A common loss is Expected Prediction Error:

\[
EPE(f) = E(Y - f(X))^2 = \int [y - f(x)]^2 Pr(dx, dy),
\]

It is minimized at \( f(x) = E(Y | X = x) \)

Objective criteria exists to measure the success of a supervised learning mechanism.
In unsupervised learning, there is no output variable, all we observe is a set \{x_i\}. The goal is to infer \(Pr(X)\) and/or some of its properties.

When the dimension is low, nonparametric density estimation is possible; When the dimension is high, may need to find simple properties without density estimation, or apply strong assumptions to estimate the density.

There is \textbf{no objective criteria} from the data itself; to justify a result: \(>\) Heuristic arguments, 
  \(>\) External information, 
  \(>\) Evaluate based on properties of the data
Classification

The general scheme.

An example.

FIGURE 1.1. The objects to be classified are first sensed by a transducer (camera), whose signals are preprocessed. Next the features are extracted and finally the classification is emitted, here either “salmon” or “sea bass.” Although the information flow is often chosen to be from the source to the classifier, some systems employ information flow in which earlier levels of processing can be altered based on the tentative or preliminary response in later levels (gray arrows). Yet others combine two or more stages into a unified step, such as simultaneous segmentation and feature extraction. From: Richard O. Duda, Peter E. Hart, and David G. Stork, Pattern Classification. Copyright © 2001 by John Wiley & Sons, Inc.
Classification

In most cases, a single feature is not enough to generate a good classifier.
Classification

Two extremes: overly rigid and overly flexible classifiers.

**FIGURE 1.4.** The two features of lightness and width for sea bass and salmon. The dark line could serve as a decision boundary of our classifier. Overall classification error on the data shown is lower than if we use only one feature as in Fig. 1.3, but there will still be some errors. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

**FIGURE 1.5.** Overly complex models for the fish will lead to decision boundaries that are complicated. While such a decision may lead to perfect classification of our training samples, it would lead to poor performance on future patterns. The novel test point marked ? is evidently most likely a salmon, whereas the complex decision boundary shown leads it to be classified as a sea bass. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.
**Classification**

Goal: an optimal trade-off between model simplicity and training set performance.

**FIGURE 1.6.** The decision boundary shown might represent the optimal tradeoff between performance on the training set and simplicity of classifier, thereby giving the highest accuracy on new patterns. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.
Classification

An example of the overall scheme involving classification:

**FIGURE 1.7.** Many pattern recognition systems can be partitioned into components such as the ones shown here. A sensor converts images or sounds or other physical inputs into signal data. The segmentor isolates sensed objects from the background or from other objects. A feature extractor measures object properties that are useful for classification. The classifier uses these features to assign the sensed object to a category. Finally, a post processor can take account of other considerations, such as the effects of context and the costs of errors, to decide on the appropriate action. Although this description stresses a one-way or “bottom-up” flow of data, some systems employ feedback from higher levels back down to lower levels (gray arrows). From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.
Classification

A classification project:

a systematic view.

FIGURE 1.8. The design of a pattern recognition system involves a design cycle similar to the one shown here. Data must be collected, both to train and to test the system. The characteristics of the data impact both the choice of appropriate discriminating features and the choice of models for the different categories. The training process uses some or all of the data to determine the system parameters. The results of evaluation may call for repetition of various steps in this process in order to obtain satisfactory results. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.
Clustering

Assign observations into clusters, such that those within each cluster are more closely related to one another than objects assigned to different clusters.

- Detect data relations
- Find natural hierarchy
- Ascertained the data consists of distinct subgroups
- ......
Clustering

Mathematically, we hope to estimate the number of clusters $k$, and the membership matrix $U$

$$U = \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ u_{21} & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ u_{k1} & u_{n2} & \cdots & u_{kn} \end{pmatrix},$$

$u_{ji} \in \{0, 1\}$, $1 \leq j \leq k$, $1 \leq i \leq n$,

$$\sum_{j=1}^{k} u_{ji} = 1, \quad 1 \leq i \leq n,$$

$$\sum_{i=1}^{n} u_{ji} > 0, \quad 1 \leq j \leq k.$$

In fuzzy clustering, we have

$$u_{ji} \in [0, 1]$$
Clustering

Some clusters are well-represented by center+spread model; Some are not.

Figure 1.2. Three well-separated center-based clusters in a two-dimensional space.

Figure 1.3. Two chained clusters in a two-dimensional space.
The purpose of dimension reduction:

- Data simplification
- Data visualization
- Reduce noise (if we can assume only the dominating dimensions are signals)
- Variable selection for prediction
## Dimension reduction

<table>
<thead>
<tr>
<th>Outcome variable $y$ exists (learning the association rule)</th>
<th>Data separation</th>
<th>Dimension reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classification, regression</td>
<td>SIR, Class-preserving projection, Partial least squares...</td>
<td></td>
</tr>
</tbody>
</table>

| No outcome variable (learning intrinsic structure)       | Clustering       | PCA, MDS, Factor Analysis, ICA, NCA... |
Curse of Dimensionality


In p-dimensions, to get a hypercube with volume $r$, the edge length needed is $r^{1/p}$.

In 10 dimensions, to capture 1% of the data to get a local average, we need 63% of the range of each input variable.
Curse of Dimensionality

In other words,

To get a “dense” sample, if we need $N=100$ samples in 1 dimension, then we need $N=100^{10}$ samples in 10 dimensions.

In high-dimension, the data is always sparse and do not support density estimation.

More data points are closer to the boundary, rather than to any other data point $\rightarrow$ prediction is much harder near the edge of the training sample.
Curse of Dimensionality

Estimating a 1D density with 40 data points.

Standard normal distribution.
Curse of Dimensionality

Estimating a 2D density with 40 data points.

2D normal distribution; zero mean; variance matrix is identity matrix.
Curse of Dimensionality

Another example – the EPE of the nearest neighbor predictor.
To find $E(Y|X=x)$, take the average of data points close to a given $x$, i.e. the top $k$ nearest neighbors of $x$

$$\hat{f}(x) = \text{Ave}(y_i|x_i \in N_k(x))$$

Assumes $f(x)$ is well-approximated by a locally constant function

When $N$ is large, the neighborhood is small, the prediction is accurate.
Curse of Dimensionality

Data:
Uniform in $[-1, 1]^p$

$$f(X) = e^{-8||X||^2}$$
FIGURE 2.8. A simulation example with the same setup as in Figure 2.7. Here the function is constant in all but one dimension: \( F(X) = \frac{1}{2}(X_1 + 1)^3 \). The variance dominates.
Curse of Dimensionality

We have talked about the curse of dimensionality in the sense of density estimation.

In a classification problem, we do not necessarily need density estimation.

Generative model --- care about the mechanism: class density function.
   Learns $p(\mathbf{X}, y)$, and predict using $p(y|\mathbf{X})$.
   In high dimensions, this is difficult.

Discriminative model --- care about boundary.
   Learns $p(y|\mathbf{X})$ directly, potentially with a subset of $\mathbf{X}$. 
Curse of Dimensionality

Example: Classifying belt fish and carp. Looking at the length/width ratio is enough. Why should we care how many teeth each kind of fish have, or what shape fins they have?
Curse of Dimensionality

Modern problems are almost always high-dimensional. Training data is often limited.

**Restrictive models:**

More assumptions (that may be wrong)

Less vulnerable to curse of dimensionality

Require less training samples

**Flexible (adaptive) models:**

Less assumptions

More vulnerable to curse of dimensionality

Require more training samples (?)

**The ideal models:**

Flexible to capture complex data structures

Resistant to curse of dimensionality, can train well with limited samples.

Can tell us about important predictors and their interactions