Memorial University of Newfoundland

Pattern Recognition

Lecture 12, June 13, 2006

http://www.engr.mun.ca/~charlesr

Office Hours: Tuesdays & Thursdays 8:30 - 9:30 PM

EN-3026

Calendar

June 2006

<table>
<thead>
<tr>
<th>Sunday</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
<th>Saturday</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>13</td>
<td>Lecture 12</td>
<td>Lecture 13</td>
<td>Assignment 3 Due</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>20</td>
<td>No Lecture Progress Report</td>
<td>Lecture 14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>27</td>
<td>Midterm Exam</td>
<td>Lecture 15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Bayesian Belief Networks

|        | $P(m_{c1}|ss,re)$ | $P(m_{c2}|ss,re)$ |
|--------|------------------|------------------|
| $ss_1$| 0.6              | 0.4              |
| $ss_2$| 0.05             | 0.95             |
| $ss_1$| 0.5              | 0.5              |
| $ss_2$| 0.01             | 0.99             |

$P(ss_1) = 0.1$, $P(ss_2) = 0.9$, $P(re_1) = 0.2$, $P(re_2) = 0.8$

Full Joint Probability Distribution:

$P(cc,mc,ss,re) = \ldots$

Exercise: Prove that you can ignore the ‘Classes Cancelled’ node when solving for $P(ss_1|mc_1)$.

See text book for some other practice questions based on the fish Belief Net.

Linear Discriminants

- Up until now, we’ve designed classifiers based on the probability density of the classes or distribution of the samples.
  - So we’ve tried to figure out the class distribution from the samples.
  - We shall now assume that we know the proper forms of the discriminant functions and use the samples to estimate the function parameters.
Linear Discriminants

Given \( \{ \mathbf{x} | \mathbf{x} \in \mathcal{C}_i, i = 1 \ldots k \} \)
find the discriminant (decision function) \( d(\mathbf{x}, \mathcal{C}_i) \) such that

The simplest discriminant is linear, which is a good solution because:
- computationally simple
- if you have very few samples, then you are not prepared to estimate probabilities or even covariance matrix.

However, it might not converge.

So we want some kind of linear discriminant, such as:

Let’s look at the simple case of 2 classes:

We need to find \( \mathbf{w}, w_0 \) from our samples.

Several approaches:
1. Trial and error
   a. Enumerate all possibilities (not simple)
   b. Guess with learning
2. Optimize a criterion function
**Guess with Learning**

*First*, add a dimension to account for $w_0$.

Define:

Then the decision function becomes

**Guess with Learning**

*Second*, change the values in class 2 by ‘normalizing’ them so that all values are negative. Using the augmented feature vector, we write:

Then we must satisfy the following condition for all of the samples in class 1 and the modified class 2:
Geometrically we have added a dimension to the feature space such that the decision boundary passes through the origin ($w_0$ term).

![Diagram showing feature space with decision boundary through origin](image)

‘Normalizing’ the samples in class 2 means that we want a plane with all the samples on the same side.

**Approach by Rosenblatt (1957)**

We can use an iterative approach to find a solution for $\mathbf{a}$:

1. $a_1 = y_i$
2. Try $a_1 \mathbf{y}_i$, $i = 2,3,...$ until $a_1 \mathbf{y}_i \leq 0$
3. Update $\mathbf{a}$: $a_2 = a_1 + y_i$
4. Continue with $a_{k+1} = a_k + y^k$, where $y^k$ is the first sample encountered such that $a_k y^k \leq 0$
5. Cycle through samples until all $N$ are correctly classified by $\mathbf{a}$

This is the Perceptron Algorithm.

More simply: $a_0 = 0$, $a_{k+1} = a_k + y^k$
Worked Example

Notes on Perceptron Algorithm

- Perceptron algorithm converges to a solution if the classes are linearly separable (Theorem 5.1 in Duda, Hart, and Stork)
- Rosenblatt devised the perceptron as a model of learning in the brain.

Each time a pattern from class 1 falls on the ‘retina’ and produces an output +1, nothing happens. If the output is -1, then weights are updated \( \mathbf{w}_{k+1} = \mathbf{w}_k + \mathbf{x}^k \).
Notes on Perceptron Algorithm

3. Could consider all of the samples at the same time.

This is the parallel form of the perceptron algorithm.
Converges quickly.

Note:

4. The perceptron discriminant is an example of a more general optimization approach.

Define a criterion function $J(w)$ (or $J(a)$) which is a minimum or maximum if $w$ (or $a$) is a solution.

$$J(a) \triangleq - \sum_{y \in Y} a^T y$$

$$(a^T y = 0)$$
Find \( a \) to minimize \( J(a) \).

Take

However, there is no dependence on \( a \), so we cannot set the equation equal to 0 to find a solution for \( a \) and thus get an analytic solution.

Resort to using steepest descent:

\[
\begin{align*}
\text{If we take the simple case of } & \quad \rho_k = 1 \\
\text{and recognize the direction of steepest descent is given by direction of } & \quad y_i \text{'s with } a^T y_i \leq 0, \text{ then }
\end{align*}
\]

This is the parallel form of the perceptron algorithm.
Minimum Square Error Function

What if the classes are not separable?

Use the criterion that gives good, but not necessarily perfect, separation of the classes.

- Eg. The Minimum Square Error

- There are both analytic and iterative methods discussed in section 5.8 of Duda, Hart, and Stork.