CMPE 422/Spring 07/Project 1: Cross–correlation

Samir Chettri

March 4, 2007

Abstract

In class we studied convolution. In this project we will see a similar concept called the cross–correlation and its use in object recognition.

Students will write a program in MATLAB to calculate the cross–correlation. They will apply their program to calculate the cross–correlation of some objects and perform object recognition. The maximum a student can get is one hundred points.

Keywords: Cross–correlation, MATLAB.

Contents

1 The problem

Cross–covariance is defined as:

$$s_{x,y}[k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n]y[n+k] \tag{1}$$

Note that the cross–covariance is defined for a single shift of the signal $y[n]$ by k units. It is only correct for mean–zero signals. For non–zero mean signal it is suitably modified as:

$$s_{x,y}[k] = \sum_{n=0}^{N-1} x[n]y[n+k] - \frac{\sum_{n=0}^{N-1} x[n]\sum_{n=0}^{N-1} y[n]}{N}. \tag{2}$$

The auto–covariance is not a function of k any more and is written as

$$s_{x,x} = \sum_{n=0}^{N-1} x[n]^2 - \frac{(\sum_{n=0}^{N-1} x[n])^2}{N}. \tag{3}$$

The cross–correlation is defined to be

$$\rho_{x,y}[k] = \frac{s_{x,y}[k]}{\sqrt{s_{x,x}s_{y,y}}}. \tag{4}$$

Clearly, from our lectures, $x[n], y[n]$ are digital signals of length N. Another point to understand is that the signals x and y must have the same length otherwise the definition doesn't work.

The student will write a program to calculate the cross–correlation between two signals. Here is what he program header will look like:

```
%
% Function: correlate.m - How much are two signals alike?
%
% Author: Your name here.
%
% Description: function [rho] = correlate(k, x, y)
%
%    input: k = shift of signal y
%           x, y = signals being correlated
%
%    output: rho = correlation coefficient in [-1 1].
%
```

2 An example

Consider the two signals

```
>> x = [0 0 1 5 1 -2 -3 -2 0 0];
>> y = [1 5 1 -2 -3 -2 0 0 0 0];
```

In order to help you with the calculations I will give you the values of $\rho_{x,y}[k]$ for $k = 0$ and $k = 1$. Remember, a shift of $k = n$ is a left–circular shift of n units. Here are the answers:

```
>> rho = correlate(0, x, y)

ans = -0.8000

>> rho = correlate(1, x, y}

ans = 4.4000
```

3 What to do?

MATLAB - 50%

1. Write a MATLAB function function [rho] = correlate(k, x, y) .
2. Calculate rho for every possible shift for the following signals x = [0 0 1 5 1 -2 -3 -2 0 0]; and y = [1 5 1 -2 -3 -2 0 0 0 0];. Your driver code will call correlate. Your code will put your cross–correlation for each shift in an array $ccr[k]$ and plot it versus shift k.

Code application 50%

This is the exciting part. You will apply your previously written code to see whether two geometric objects are similar. Consider two squares with corner coordinates (clockwise from upper left corner):

1. **Rectangle 1**: (87, 1), (87, 126), (12, 126), (12, 1).
2. **Rectangle 2**: (102, 312), (102, 371), (2, 371), (2, 312).

Note the each square is physcally separate (i.e., no overlap) and have different sizes. In order to make your program correlate work, clearly you'll have to convert each square from a two dimensional object to a one dimensional signal. How? Use the centroid of the square as the origin and measure the distance to the upper left corner. Continue clockwise till you reach the upper left again, sampling as you go along. Your samples should include the corners. This process converts 2–D to 1–D. Since each rectangle is not the same size, you'll have to take special care to make each 1–D signal have the same number of samples.

Your code will obtain the signals for the two rectangles, equalize their lengths, and use function [rho] = correlate(k, x, y) to calculate the cross-correlation for each shift. It will then plot the cross–correlation graph and print the maximum cross-correlation.

Due date: 11:59pm on 23 March 2007 by email only to Haleh Safavi.