Lecture 20
Implementing Dog/Cat Recognition in MATLAB

Jason J. Bramburger

In the previous two lectures we saw that we could take 80 pictures of dogs and 80 pictures of cats and create a computer program to classify the images as either dogs or cats. It wasn't perfect, but out of 160 total images we only got about 6 or 8 incorrect - that's pretty successful! Here's the problem. It's not good enough with supervised machine learning to do a good job on the training data. You could easily get them all right by having the computer memorize the image and look them up in a table. So our algorithm is only useful if it is able to classify new pictures of dogs and cats. Let's start by calling the functions we created in the previous lecture to train our model on the training set of 160 images.

```
load('catData.mat')
load('dogData.mat')

dog_wave = dc_wavelet(dog);
cat_wave = dc_wavelet(cat);

feature = 20;
[U,S,V,threshold,w,sortdog,sortcat] = dc_trainer(dog_wave,cat_wave,feature);
```

That's all we have to do for training! Now, let's test the performance. We will load in the file 'PatternRecAns.mat' which contains a mixture of new pictures of dogs and cats. When you load this MATLAB file you will see it contains 38 images of dogs and cats in the 'TestSet' matrix. You will also see an array called 'hiddenlabels' which is a vector of 0's and 1's (0 for dog and 1 for cat) which we can use to check our performance against without having to look through each image individually. Let's load in the .mat file and look at some example images (on the next page).

```
load('PatternRecAns')

figure(1)
for k = 1:9
   subplot(3,3,k)
   test = reshape(TestSet(:,k+5),64,64);
   imshow(test)
end
```

We can now use our classification procedure to determine which images are dogs and which images are cats. Remember that we need to first perform a wavelet transform, then PCA, then project onto the line we found from LDA.

```
TestNum = size(TestSet,2);
Test_wave = dc_wavelet(TestSet); % wavelet transform
TestMat = U'*Test_wave; % PCA projection
pval = w'*TestMat;
```

We can now check which images were classified as dogs and which were classified as cats using the threshold found during our training. We want 0 if we are below the threshold (a dog) and 1 if we are above it (a cat).

```
% Cat = 1, dog = 0
ResVec = (pval > threshold)
```

We can check for errors using the hiddenlabels vector included with the testing data set. We can also check what our success rate was on the test data.

```
% 0s are correct and 1s are incorrect
err = abs(ResVec - hiddenlabels)
errNum = sum(err);
sucRate = 1 - errNum/TestNum;
```

We can see that we got 2 out of 38 wrong. That gives us a success rate of more than 94%! That's pretty good in my opinion! Let's see which animals were misclassified.

```
k = 1;
TestNum = length(pval);
figure(2)
for j = 1:TestNum
   if ResVec(j) ≠ hiddenlabels(j)
      S = reshape(TestSet(:,j),64,64);
      subplot(1,2,k)
      imshow(S)
      k = k+1;
   end
end
```

Notice that the two failures were dogs that were misclassified as cats. In particular, both dogs had pointy ears. We have already seen that it is the ears that our algorithm has identified is the defining characteristic that differentiates dogs and cats. So, it shouldn't be surprising that these dogs were misclassified as cats.

Classifying Your Pets

The thing that is on all of our minds now is whether we can classify our own pets. Below is a script that will format your pictures to the right size so that we can classify them as dogs or cats. At the top you must enter the name of the image file. The code will then show the original image. This will perform best if you provide a photo of the face only (or cropped) and taken straight on - just like the training data. We convert the image to grayscale (assuming the image is in colour), resize to 64 × 64 pixels, perform a wavelet transform, PCA, LDA, and then classify them.

Aside: Bailey.jpg is a picture of my childhood dog Bailey. We don't really know what kind of dog he was aside from a good one. Lola.jpg is a picture of my sister's Yorkie named Lola. I like to tease her that Lola is basically a cat. As you will see, our program disagrees with my assessment. Dingbat.jpg is my partner's childhood cat Dingbat II. Unfortunately, I was not able to get a good picture of Dingbat I.

```
% Clear pval
pval = [];

% Read in image
I = imread('Bailey.jpg'); % Change the file name for your photo
%I = imread('Lola.jpg');
%I = imread('Dingbat.jpg');
imshow(I)

% Convert to grayscale and resize
I = rgb2gray(I);
I = im2double(I);
I = imresize(I,[64,64]);
I = reshape(I,64*64,1);

%  Classify the image
I_wave = dc_wavelet(I); % wavelet transform
IMat = U'*I_wave; % PCA projection
pval = w'*IMat;

if pval > threshold
    disp('Cat')
else
    disp('Dog')
end
```