Linear Kernel

Now that we have standardized the data, we are ready to use an SVM model. For the rest of this exercise we will be using Matlab's built in SVM toolbox. Specifically, to train the model we will use the function `model = fitcsvm(Xtrain,Ytrain,'KernelFunction','<function_name>')` that takes two variables as an input: 1) a feature matrix `Xtrain` storing all of our training data features, and 2) the ground truth label vector `Ytrain` storing the true values whether a given image contains a person or not. Furthermore, as an additional parameter, this function takes a 'KernelFunction' , which we will have to code up for this assignment.

Intuitively, a kernel function can be viewed as a similarity metric between two given feature vectors. Given feature vectors of two data instances, the kernel function needs to output how similar these two data instances are. You can read more about kernel functions at https://en.wikipedia.org/wiki/Kernel_method (https://en.wikipedia.org/wiki/Kernel_method).

Your first task is to write a linear kernel function, which can be implemented as a dot product between the feature vectors. Your function should take two feature matrices `X1` and `X2` as its inputs, and compute a Kernel matrix `K.` Note that matrices `X1` and `X2` may have a different number of rows (i.e. observations) but they will always have the same number of columns (i.e. features). Then, the computed kernel matrix `K` should have the same number of rows as `X1,` and the same number of columns as there are rows in `X2.` Every entry `K(i,j)` should store the dot product between the feature vector from row `i` in matrix `X1,` and the feature vector from row `j` in matrix `X2.`

Your function should be implemented as a one line matrix multiplication without using `for` loops, which would make SVM training more computationally expensive. Your SVM model using a linear kernel function should achieve 89.15% accuracy on the testing dataset.

Your Function

Save Reset MATLAB Documentation (https://www.mathworks.com/help/)

```matlab
function K = KernelLinear(X1, X2)
   % computes a linear kernel
   %
   % Input:
   % - X1: an n x d dimensional feature matrix where n is the number of observations, and d is the number of fe
   % - X2: an m x d dimensional feature matrix where m is the number of observations, and d is the number of fe
   % Output:
   % - K: an n x m dimensional kernel matrix where K(i,j) stores a dot product between the data point i in X1,
   K = X1 * X2';
end

```

Code to call your function

Reset

```matlab
load('ImageDataTrain.mat');
Xtrain=StandardizeData(data.trainX);
Ytrain=data.trainY;

load('ImageDataTest.mat');
Xtest=StandardizeData(data.testX);
Ytest=data.testY;

model = fitcsvm(Xtrain,Ytrain,'KernelFunction','KernelLinear');
[preds,~] = predict(model,Xtest);
```

▶ Run Function ?

Previous Assessment: All Tests Passed

Submit ?