Histogram Intersection Kernel

Even though the polynomial kernel function is more complex than the linear kernel function, as indicated by the testing accuracy, it does not make SVM model much more powerful. As a last exercise, we will use a more sophisticated kernel function refered to as the histogram intersection kernel. We will observe that using the histogram intersection kernel to train our SVM model will alow us to obtain a model that performs about 4% better than the previous model.

Formally, the histogram intersection kernel can be written as:

$$k(a, b) = \sum_{j=1}^{d} min(a_j, b_j)$$

where a, b are both d dimensional feature vectors. Inutitively, the histogram intersection kernel is trying to capture what the overlap between two given features is. For instance, if we have $a_3 = 0.66$ and $b_3 = 0.13$ then $min(a_3, b_3) = 0.13$, which means that their intersection is small. However, if we have $a_5 = 1.92$ and $b_5 = 1.97$ then $min(a_5, b_5) = 1.92$, which is a much higher value indicating that there is a much higher overlap between these two feature values. Empirically, we will show that such a kernel function works very well with SVMs for our particular people detection task.

Just like before, your function should take two feature matrices `X1` and `X2` as its inputs, and compute a Kernel matrix `K`. Every entry `K(i,j)` should store the value $k(X1(i,:), X2(j,:)$, where kernel function $k(a, b)$ is defined above.

Your SVM model using a histogram intersection kernel function should achieve 93.53% accuracy on the testing dataset, which is much better than our two previous models. Note that in this case, you are allowed to use the built-in 'for' loops.

Your Function

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

```
function K = KernelIntersection(X1, X2)
   % computes a histogram intersection 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 histogram intersection between the data poin

   r_1 = size(X1,1);
   r_2 = size(X2,1);
   K = zeros(r_1, r_2);

   % for each row in X1, create a new matrix
   % same size as X2
   % take min, then sum rows (i.e. across columns)

   for i = 1:r_1
     % for K(i,:)
     v_i = X1(i,:);
     V_i = repmat(v_i,r_2,1);

     Min_i = min(V_i, X2);
     col_sum = sum(Min_i,2);

     K(i,:) = col_sum';

   end

end
```

Code to call your function

Reset

```
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','KernelIntersection');
[preds,~] = predict(model,Xtest);
```

▶ Run Function ?

Previous Assessment: All Tests Passed

Submit ?

✓ **Is the Histogram Intersection Kernel Function Correct?**