Homework 3 - Pattern Recognition

Dzvezdana Arsovska

April 29, 2017

Independent component analysis (ICA) is a statistical technique for revealing hidden factors that underlie sets of random variables, measurements, or signals. ICA can be seen as an extension to PCA and it is a much more powerful technique.

ICA defines a generative model for given large database of samples. In the model, the data variables are assumed to be linear or nonlinear mixtures of some unknown latent variables, and the mixing system is also unknown. The latent variables are assumed non Gaussian and mutually independent, and they are called the independent components of the observed data. These independent components, also called sources or factors, can be found by ICA.

The main concept of ICA applied to images is based on the idea that each image (subimage) may be perceived as linear superposition of features weighted by some coefficients. In case of ICA, features are represented by columns of mixing matrix A. In addition ICA features are localized and oriented and sensitive to lines and edges of varying thickness of images.

Support Vector Machine (SVM) is a supervised machine learning algorithm that is mostly used for classification. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick. SVMs are based on the idea of finding a hyperplane that best divides a dataset into two classes.Support vectors are the data points nearest to the hyperplane, the points of a data set that, if removed, would alter the position of the dividing hyperplane. Because of this, they can be considered the critical elements of a data set.

1 Task 1

Create a car recognition system using ICA and SVM to detect if a car is or is not present in an image.
Answer:

The training part of the system consists of the following steps:

1. In a training set, 500 car and non-car images are labeled as 1 or 0 respectively. The car images include various types of cars and non car images usually represent some outdoor scenery.

2. An ICA algorithm is applied to learn the independent image basis and generate the training ICA features.

3. Using the ICA features, SVM with quadratic kernel is trained for classification.

ICA is performed using the fastica function in Matlab. This function outputs the estimated separating matrix W and the corresponding mixing matrix A.

```
[icasig , A, W] = fastica(X);
```

For a given input data matrix Training Set (in our case the training images which are 500 x 4000 matrix, i.e. each row is an observation) unmixing can be done using: icasig = W*TrainingSet (TrainingSet is separated on independent 'sources' icasig) and mixing can be done using: TrainingSet = A*icasig.

A stores the independent components extracted from TrainingSet. icasig stores the sources or the projections of TrainingSet on the independent components or the projection of the input data on the ICs. W separates the sources or independent components from the mixtures.

icasig then is used to train SVM with quadratic kernel. Then a classification is made after the test data is projected in the same space as the training data.

Figure 1 shows a set of learned independent basis images obtained from training images. These basis images look like localized oriented filters which emphasize image edge information.

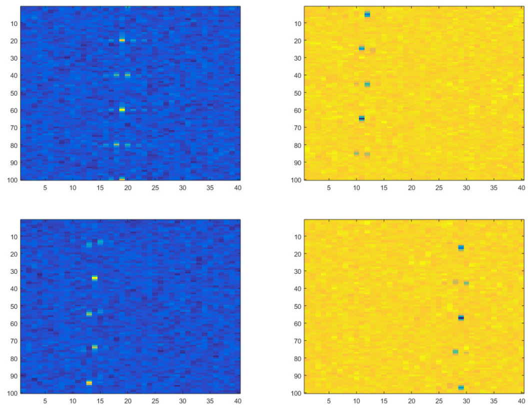

Figure 1: Components visualization

The accuracy was calculated when all, 350 and 250 components were used. The accuracy was used was 76.1%, 57.2% and 48.9% respectively. The results are shown in Figure 2.

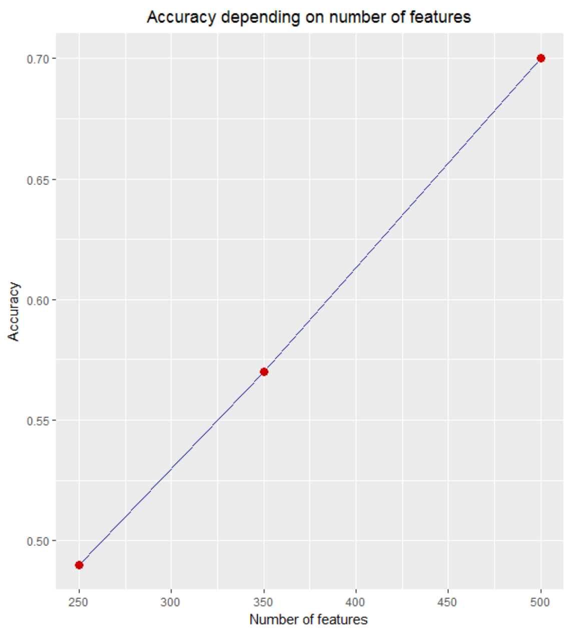

Figure 2: Accuracy plot

```matlab
clc;clear all;close all

% Directory for Training Data
myFolder = 'C:\Users\arsov\Downloads\CarData\Train';
filePattern = fullfile(myFolder, '*.pgm');
jpegFiles = dir(filePattern);

% New - 3D matrix to store images
imageMatrix = cell(1,numel(jpegFiles));

% Load Training data
for k = 1:length(jpegFiles)
    baseFileName = jpegFiles(k).name;
    fullFileName = fullfile(myFolder, baseFileName);
    fprintf(1, 'Now reading %s\n', fullFileName);
    imageArray50x50 = imread(fullFileName);

    imageMatrix{k} = double(imageArray50x50);

end

% Directory for Test data
myFolder1 = 'C:\Users\arsov\Downloads\CarData\Test';
filePattern1 = fullfile(myFolder1, '*.pgm');
jpegFiles1 = dir(filePattern1);

imageMatrix1 = cell(1,numel(jpegFiles1));

% Load Test data
for j = 1:length(jpegFiles1)
    baseFileName1 = jpegFiles1(j).name;
    fullFileName1 = fullfile(myFolder1, baseFileName1);
    fprintf(1, 'Now reading %s\n', fullFileName1);
    imageArray50x501 = imread(fullFileName1);
    imageArray50x50New1 = imresize(imageArray50x501, [100 40]);

    imageMatrix1{j} = double(imageArray50x50New1);

end

% Label the Training data
train_label= zeros(size(500,1),1);
train_label(1:250,1)   = 0;         % 0 = Not a car
train_label(251:500,1)  = 1;         % 1 = Car

% Prepare numeric matrix for svmtrain
Training_Set=[];
for m=1:length(imageMatrix)
    Training_Set_tmp   = reshape(imageMatrix{m}, 1, 100*40);
    Training_Set=[Training_Set;Training_Set_tmp];
end

Test_Set=[];
for n=1:length(imageMatrix1)
    Test_set_tmp   = reshape(imageMatrix1{n}, 1, 100*40);
    Test_Set=[Test_Set;Test_set_tmp];
end

% Default ica including all components.
[icasig, A, W] = fastica(Training_Set);

% Replace some of the rows with zeros
%for n=250:500
%F(n,:) = 0
%end

% Show recovered components
for iter=1:10,
 C{iter}=reshape(icasig(iter,:),[100 40]);
end;
figure(3);
set(gcf,'Name','ICA results');
```

```
subplot(2,3,1); imagesc(C{1});
subplot(2,3,2); imagesc(C{2});
subplot(2,3,3); imagesc(C{3});

% Project Test data set
A_test = icasig' \ Test_Set';
A_Test = A_test' * icasig;

% Perform first run of svm
SVMStruct = svmtrain(icasig , train_label, 'kernel_function', 'quadratic')

% Classify the test data
Group = svmclassify(SVMStruct,A_Test)
% Calculate the accuracy
acc = sum(train_label == Group) / numel(train_label)
```