CS479 - Programming Assignment 4: Classifying gender based on faces using SVM (and comparing against Bayes Classifier baseline)

Tim Kwist and Shane Melton

Due May 2, 2016 - Submitted May 2, 2016

Contents

1 Technical Discussion

In this assignment we were tasked with using support vector machines (SVM) to classify individuals' faces to being either male or female. We then performed the same classification with a Bayesian classifier and maximum likelihood estimation, and compared these results with those of the SVM.

Our data consisted of 400 frontal images of 400 distinct people with different races, facial expressions, and lighting conditions. Every image has gone through the process of histogram equalization to each normalized image in order to account for the varying lighting conditions of the original images. The image data is evenly split with an equal number of male and female images. Additionally, the images came in two different sizes, 16x20 and 48x60, allowing us to test each classifier on different data sizes and compare their performances.

In this project, unlike the others, we performed our experiments using a three-fold cross-validation procedure to find the average error rate for each classifier and image resolution. The data provided to us was already pre-divided into the following data sets that can be seen in table 1.

Fold	Training	Test
1	69 male & 65 female	131 male & 135 female
2	62 male & 72 female	138 male & 128 female
3	71 male & 63 female	129 male & 137 female

Table 1: Data Fold information

1.1 Support Vector Machines

1.1.1 Overview

Support vector machines' primary goal is to find a discriminant that maximizes the margin, or space, between classes in order to improve generalization performance of the classifier. The margin of separation in SVMs is defined by the distance from the nearest training samples to the discriminant. These samples are typically the most difficult samples to classify, therefore, by maximizing the distance between the discriminant and these samples (known as support vectors) we improve the general performance of the classifier.

Support vector machines are primarily two-class classifiers that can be extended to multiple class classification (which the library we used in this project supports). However, in order to assist SVMs with finding an optimal discriminant we can transform the data to a higher dimension, using a kernel function, with the goal in mind that by moving the data to a higher dimension it will be more capable of being linearly separable. These kernel functions are relatively easy to compute and therefore do not significantly affect the complexity of the SVM.

1.1.2 Implementation

Our experiment implements SVMs using the LibSVM library which can be found at the following address `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`. The library allows for the use of multiple different customizable kernel functions with optional parameters. Our results from experimenting with these different kernel functions and options can be seen below in the Results section.

1.2 Bayesian and Maximum Likelihood Estimation

The theory for our Bayesian classifier and maximum likelihood estimation can been seen in our report for Programming Assignment 1 and 2.

2 Results

2.1 Experiment 1

Using the LibSVM library and two sets of male/female face data (including text data that already reduced the face images into eigenvectors and eigenvalues), we train and test an SVM classifier with many different training parameters.

The first difference in parameters is the type of kernel used. We try two different kernels: polynomial($(gamma * u' * v + coef0)^{degree}$) and radial basis function (rbf) ($exp(-gamma * |u - v|^2)$).

For the polynomial kernel, we use the same gamma (1) and coef0 (0) for all trials. We vary the degree, using d=1, d=2, and d=3.

For the rbf kernel, gamma is equivalent to $1/(2 * sigma^2)$. Thus, using sigma=1, sigma=10, and sigma=100, we set gamma to gamma=0.5, gamma=0.005, and gamma=0.00005, respectively.

Additionally, for both kernels, we use varied costs (c), c=1, c=10, c=100, and c=1000.

We train the model on one training set, then test on a 3-fold test set and take an average for the accuracy.

The tables below show our results for 16x20 images and for 48x60 images.

Table 2: 16x20 SVM results

Kernel	Degree (poly) or Sigma (rbf)	C	Fold 1	Fold 2	Fold 3	Average
Poly	1	1	85.34%	90.98%	89.85%	**88.72%**
		10	85.34%	90.98%	89.85%	**88.72%**
		100	85.34%	90.98%	89.85%	**88.72%**
		1000	85.34%	90.98%	89.85%	**88.72%**
	2	1	59.02%	69.17%	60.90%	**63.03%**
		10	59.02%	69.17%	60.90%	**63.03%**
		100	59.02%	69.17%	60.90%	**63.03%**
		1000	59.02%	69.17%	60.90%	**63.03%**
	3	1	63.16%	91.35%	73.68%	**76.06%**
		10	63.16%	91.35%	73.68%	**76.06%**
		100	63.16%	91.35%	73.68%	**76.06%**
		1000	63.16%	91.35%	73.68%	**76.06%**
RBF	1	1	49.25%	48.12%	48.50%	**48.62%**
		10	49.25%	48.12%	48.50%	**48.62%**
		100	49.25%	48.12%	48.50%	**48.62%**
		1000	49.25%	48.12%	48.50%	**48.62%**
	10	1	84.97%	90.97%	87.21%	**87.72%**
		10	87.22%	90.60%	88.72%	**88.85%**
		100	87.22%	90.60%	88.72%	**88.85%**
		1000	87.22%	90.60%	88.72%	**88.85%**
	100	1	49.25%	90.60%	48.49%	**62.78%**
		10	49.25%	48.12%	48.49%	**48.62%**
		100	86.46%	93.61%	88.72%	**89.60%**
		1000	85.71%	91.73%	89.85%	**89.10%**

Table 3: 48x60 SVM results

Kernel	Degree (poly) or Sigma (rbf)	C	Fold 1	Fold 2	Fold 3	Average
Poly	1	1	88.72%	91.73%	89.47%	**89.97%**
		10	88.72%	91.73%	89.47%	**89.97%**
		100	88.72%	91.73%	89.47%	**89.97%**
		1000	88.72%	91.73%	89.47%	**89.97%**
	2	1	58.64%	66.92%	65.04%	**63.53%**
		10	58.64%	66.92%	65.04%	**63.53%**
		100	58.64%	66.92%	65.04%	**63.53%**
		1000	58.64%	66.92%	65.04%	**63.53%**
	3	1	69.92%	84.96%	81.20%	**78.69%**
		10	69.92%	84.96%	81.20%	**78.69%**
		100	69.92%	84.96%	81.20%	**78.69%**
		1000	69.92%	84.96%	81.20%	**78.69%**
RBF	1	1	49.25%	48.12%	48.50%	**48.62%**
		10	49.25%	48.12%	48.50%	**48.62%**
		100	49.25%	48.12%	48.50%	**48.62%**
		1000	49.25%	48.12%	48.50%	**48.62%**
	10	1	49.25%	48.12%	48.50%	**48.62%**
		10	49.25%	48.12%	48.50%	**48.62%**
		100	49.25%	48.12%	48.50%	**48.62%**
		1000	49.25%	48.12%	48.50%	**48.62%**
	100	1	49.62%	48.12%	50.38%	**49.37%**
		10	88.72%	94.36%	88.72%	**90.60%**
		100	88.72%	92.11%	89.10%	**89.98%**
		1000	88.72%	92.11%	89.10%	**89.98%**

For 16x20 images, we found the RBF kernel with sigma=100 and C=100 to give the best averaged accuracy (89.60%).

For 48x60 images, we found the RBF kernel with sigma=100 and C=10 to give the best averaged accuracy (90.60%).

2.2 Experiment 2

For Experiment 2, we repeat Experiment 1 but use Maximum Likelihood Estimation + Bayes Classifier to train and test gender classification. Maximum Likelihood Estimation is used to find the mean and covariance matrix of the training set, which is then used as the model for a Bayesian Classifier to test accuracy on a 3-fold test set.

Table 4: MLE + Bayesian Classifier results

Image-Size	Fold 1	Fold 2	Fold 3	Average
16x20	50.75%	48.12%	51.50%	**50.12%**
48x60	50.75%	48.12%	51.50%	**50.12%**

The Bayesian Classifier turned out to have identical results for images of size 16x20 and 48x60 which leads us

to conclude that the classifier may be relatively size invariant. Regardless, the results of the Bayesian classifier are virtually no better than randomly choosing or guessing the gender for each face, and are much worse than that of the SVM.

3 Division of Work

Shane:

- 'Clean' given data files to be easier to read in / utilize
- Setup data files for Bayes-Classifier
- Program Bayes-Classifier for this experiment
- Theory write-up

Tim:

- Setup data files for SVM scripts
- Write bash scripts to utilize libsvm
- Results write-up

4 Program Listings

Source will be sent by email and can be found on Github at the following URL:
`https://github.com/timkwist/CS479/tree/master/PA3-Eigenfaces`

Eigen library can be found at the following URL:
`http://eigen.tuxfamily.org/index.php?title=Main_Page`

Box-Muller Transformation C++ Code:
`ftp://ftp.taygeta.com/pub/c/boxmuller.c`

Libsvm:
`http://www.csie.ntu.edu.tw/~cjlin/libsvm/`