

CS281B Project: Statistical Learning of Human Attractiveness

Ryan White

May 16, 2004

Abstract

We apply computer vision methods to the task of automatically predicting human attractiveness from frontal face images. A dataset of thousands of images and corresponding scores was obtained from a popular website that asks viewers to rate the attractiveness of the people appearing in the images. We applied several kernel regression techniques to the problem and found that while all techniques showed evidence of overfitting, partial least squares provided the best results on a test dataset. Overall statistical learning results are mediocre, but human performance isn't much better.

1 Introduction

Since this project is a largely empirical study of statistical learning techniques on actual data, the first section is devoted to understanding the dataset and the relevant computer vision work necessary to apply statistical methods.

1.1 The Dataset

Our dataset consists of images and scores extracted from the website hotornot.com [2]. Using a webcrawler to download images from the website, we collected over 30000 images split roughly evenly between the two genders. Figure 1 shows a snapshot of the hotornot website. We extracted the following items: the image of the subject being rated, the average score for the image, and the number of votes. The website allows users who post images to self categorize based on two criteria. Because most users (and thus most of the images) are young, we limited our work to the male and female 18 to 25 year old categories. While we were able to download 30,000 images over the course of several days, the website claims to have had 7 billion votes and 9.1 million photos submitted.

We pruned this dataset based on a number of metrics:


- Quality of face detector response. The faces were found using the Mikolajczyk-Schmid face finder [3]. By selecting only images where the face detector responded highly, we can be reasonably sure that the face had been found accurately. Scores of 10 or greater were used.
- Size of face in image. Faces at least 100 pixels (and less than 400 pixels) on a side were selected so that the future steps would have reasonably high resolution images to process.
- High rectification score. Rectification code based on [1] and [4] found important facial features to rectify all of the faces to a common view using an affine transform. Only reliable rectifications (with scores above 2.5) were used. Again, this is mostly to guarantee that subsequent processing will have reliable images with which to work.
- More than 50 votes. According to the hotornot website, "Scores do not tend to change much after as few as 30 to 50 votes."

HOT or NOT.
As seen in People, Time, Newsweek, NY Times, and USA Today

[Meet Me at HOT or NOT - click here](#)

Please select a rating to see the next picture.
 1 2 3 4 5 6 7 8 9 10
 NOT HOT

Show me ages




You can share this picture with a friend:
<http://www.hotornot.com/r/?eid=GROYHRR&key=VGX>

Please help keep this site **FUN, CLEAN, and REAL.**
[Click here](#) if the picture above is broken, copyrighted, or inappropriate.
 If the pictures take more than 15 seconds to load, [click here](#)

Are you HOT or NOT ?
 Submit your picture and find out!

HOT or NOT
 What others thought
7.9
 based on 933 votes
 You rated him: 5
[Click Here to Meet Me](#)



He last checked his score:
7 hours ago

If you are geeky like us, you may find [this](#) funny and interesting! -jim and james

Thanks to the San Francisco 49ers for inviting us to a game!!! It was incredible. [Click Here](#) to see pics of who we got to meet! :)

[Meet the guys that run HOT or NOT - Jim and James!](#)

Figure 1: An example page from the website www.hotornot.com. Note that this includes several key pieces of information for building a dataset: an image of the person, their average score and the number of votes that they received.

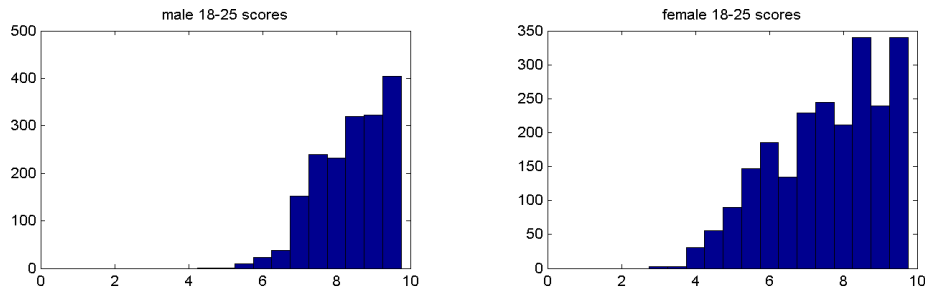


Figure 2: A histogram of the scores received in the rectified dataset. Note that the distribution differs by gender.

From these pruning techniques, we reduced our dataset to approximately 4000 images, divided almost evenly by gender. Most of the experiments presented in this paper are drawn from the rectified face images, as opposed to the original images. The rectified images were all downsampled to 86 pixels by 86 pixels with three color channels.

2 Evaluating Performance

Since the goal of this paper is to build a program that can predict human attractiveness scores, it is important to define a reasonable error metric. Our metric is the Mean Squared Error Ratio (MSER):

$$MSER = \frac{\frac{1}{N} \sum_{i=1}^N (P_i - S_i)^2}{\frac{1}{N} \sum_{i=1}^N (\bar{S} - S_i)^2}$$

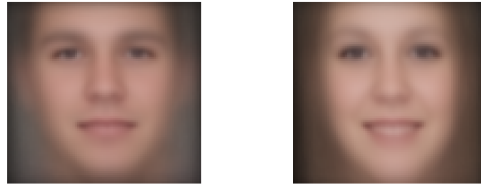


Figure 3: The average male (left) and female (right) faces. These are pixel by pixel averages on the rectified images.



Figure 4: The average attractive (right) and unattractive (left) male (bottom) and female (top) faces. The images on the right reflect the average taken over the 40 images with the lowest attractiveness scores and the images on the right reflect the average taken over the 40 images with the highest scores.

$$= \frac{\sum_{i=1}^N (P_i - S_i)^2}{\sum_{i=1}^N (\bar{S} - S_i)^2}$$

where S_i is the score of image i , P_i is the predicted score for image i and \bar{S} is the average score over all images. In this metric, a 1 indicates that the prediction algorithm does nothing and a 0 indicates perfect prediction.

3 Human Performance

To judge the difficulty of the dataset, experiments were run with a human subject told to guess the score provided by the hotnot website. The human was given a training set of images to study, and a test set to predict on.

Two separate experiments were run: one where only the face was provided to the human subject and one where the entire original image was provided to the human subject. (Only images from the female 18-25 category were used, since empirical evidence suggested that the male 18-25 category was more difficult)

For the whole image, the human subject achieved an MSER of 0.50. For the rectified face images, the subject achieved an MSER of 0.63. Clearly some information is lost by cutting out the face and rectifying it, but vision techniques for the rest of the body aren't well studied.

4 Regression Techniques

The basic approach for the statistical portion of this project is to treat the raw pixel values of the rectified images as the features, and use the scores as the desired output. This naturally falls into the structure of a linear regression.

Applying least squares directly on the pixel values is undesirable. First, there are $86 * 86 * 3$ or 22188 values. After all of the pruning of the dataset described above, we are left with approximately 2000 images in each category - far fewer than the number of coefficients.

Excluding Figure 5, all of the results reported from here are apply only to the female 18-25 category. Figure 5 supports this focus, showing that the male portion of the dataset is extremely difficult.

4.1 Comparing Faces

In order to cope with the large number of dimensions, several kernel techniques were tried. The simplest setup was to use a gaussian kernel to compare faces. Evidence from the computer vision literature suggests that using a spacial gaussian centered in the middle of the face to weight the pixel values improves results. This fact was confirmed on this dataset.

The gaussian kernel has the free parameter σ^2 . Using cross validation over a wide range of values and a wide range of regression techniques, empirical evidence suggests that values between 10^3 and 10^6 gave similar good results.

Restricting the kernel matrix to the first 60 columns (called exemplars later in this report) was found to be roughly equivalent to comparing all of the faces to three 'canonical' faces: the average face, the average attractive face and the average un-attractive face. (computed as pixel averages across all images, the 40 with the highest scores and the 40 with the lowest scores respectively)

Domain specific knowledge suggests that individual facial features can be separated and compared independently. Defining kernels that roughly correspond to the mouth, nose, eyes, cheek and jaw and using a greedy algorithm for feature selection, the following order of importance was obtained (from most important to least important): cheek, mouth, nose, eyes, jaw.

4.2 Linear and Ridge Regression

The first regression technique we tried was linear regression, solving for the coefficients using least squares. Results were worse than guessing on the test dataset.

<i>Kernel Feature</i>	<i>Male MSER</i>	<i>Female MSER</i>
Average Faces	0.995	0.93
60 Random Faces	0.969	0.93
200 KPCA coeffs	0.953	0.86
Upper Cheek	0.997	0.94
Lower Cheek	0.996	0.98
Left Eye	0.989	0.97
Mouth	0.993	0.94
Nose	0.995	0.97

Figure 5: Some kernelizing results. The last five (Upper Cheek, Lower Cheek, Left Eye, Mouth and Nose) were created by selecting 50 images from the dataset at random and then kernelizing the rest of the dataset with respect to these images. All values computed on a test dataset using ridge regression.

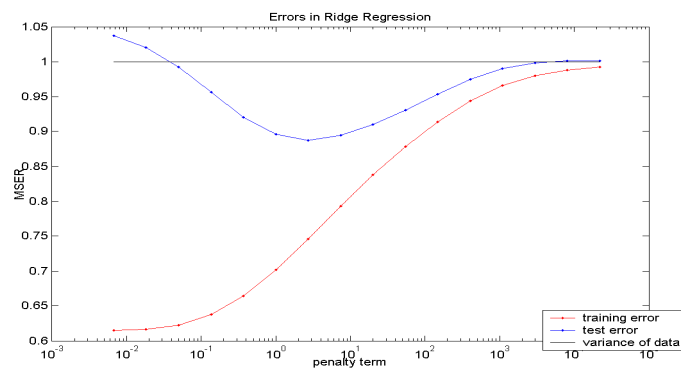


Figure 6: Ridge regression on the kernel matrix. An MSER of 1 is equivalent to always guessing the mean, an MSER of 0 indicates perfect prediction.

To prevent some of the overfitting, we tried ridge regression. As shown in Figure 6, ridge regression significantly improved the overfitting problem.

4.3 Lasso Regression

Continuing through the regression chapter of the course textbook [5], the next method we tried was lasso regression. As shown in Figure 7, lasso regression performed significantly worse than ridge regression. In addition, since lasso regression involves solving a quadratic program, it runs significantly slower than ridge regression.

However, since lasso regression forces many of the coefficients to zero, we can use it to select features. Retraining the ridge regression using these features resulting in almost identical performance to the entire set of features.

4.4 Partial Least Squares

Again, following the course textbook [5], we implemented partial least squares. Trying several different versions of the kernel matrix, we found that comparing to a small set of 'exemplars' that aren't used in the actual regression performed better than providing the whole kernel matrix. Effectively, this reduces to breaking the dataset into three parts: exemplars, training and test. The exemplars are used to create a pseudo kernel matrix, the partial least squares is trained on the training data and then the whole thing is tested on the test dataset.

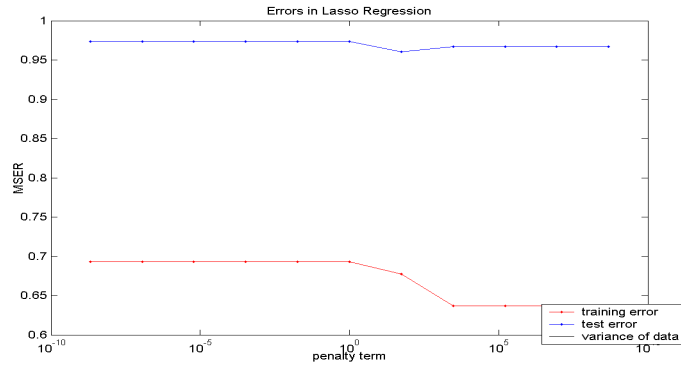


Figure 7: Lasso regression on the kernel matrix. An MSER of 1 is equivalent to always guessing the mean, an MSER of 0 indicates perfect prediction.

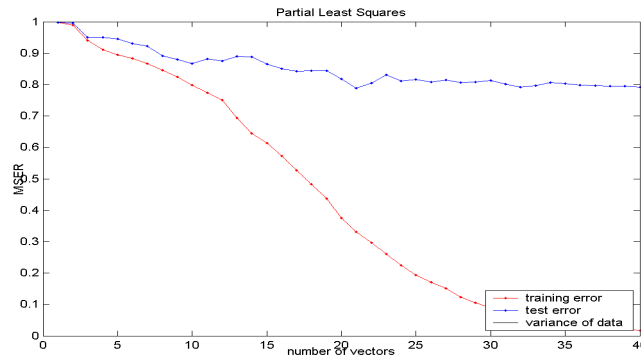


Figure 8: Partial least squares on the kernel matrix. An MSER of 1 is equivalent to always guessing the mean, an MSER of 0 indicates perfect prediction.

Results of this reduced kernel partial least squares are shown in Figure 8. Partial least squares yields an effective MSER on the training set of 0.80 - beating all other methods.

5 Conclusion

At the most basic level, the results are lackluster: the kernelized regression reduced the variance by 20 percent. However, the problem is hard, with a human subject only able to reduce the variance by 37 percent. As expected, partial least squares performed better than all other methods.

Two other interesting results are revealed from the dataset. First, the female dataset showed significantly better results. Second, defying traditional social stereotypes, the cheek and mouth proved to be much more effective at predicting the score than the eyes.

References

- [1] Alexander Berg and Jitendra Malik. Geometric blur for template matching. In *Computer Vision and Pattern Recognition*, 2001.
- [2] James Hong and Jim Young. <http://www.hotornot.com>.

- [3] K. Mikolajczyk. *Detection of local features invariant to affine transformations*. PhD thesis, INPG Grenoble, 2002.
- [4] Tamara Miller, Alexander Berg, Jaety Edwards, Michael Maire, Ryan White, Yee-Whye Teh, Erik Learned-Miller, and D.A. Forsyth. Faces and names in the news. Submitted, Computer Vision and Pattern Recognition, 2004.
- [5] John Shawe-Taylor and Nello Christiani. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004. Preprint used as course textbook for CS281b at UC Berkeley.