AI Meets Online Dating
Predicting Attractiveness in Profile Pictures

Tucker Leavitt (tuckerl), Duncan Wood (duncanw), Huyen Nguyen (huyenn)
Stanford University CS229 – Fall 2015

Abstract— **We construct a learning model to predict the attractiveness of facial images. We use a face detection API to tag several facial landmarks and use the pairwise distances between these landmarks as the features for our model. We find that our model rates images similarly to humans, making the same rating that an average human would nearly three quarters of the time. Our work may be useful for the online dating community and/or computer vision community.**

I. INTRODUCTION

Although judgments of beauty are subjective, the prevalence of beauty contests, cosmetic surgery and homogeneous ratings of photos on dating websites suggest there might be a common denominator for attractiveness perception. This project is an attempt to quantify this factor.

In this project, we explore the notion of facial attractiveness using machine learning. Our model attempts to predict attractiveness scores for facial images. The input for our model is a frontal image of a female subject. The output will be a discrete value between 1 and 5 which represents the attractiveness of the subject in the image.

There have been several experiments with a similar goal. In 2007, Yael Eisenthal and his team at Tel-Aviv University, Israel trained their model on 2 datasets. One contains 92 images of Austrian females, and one contains 92 images of Israeli females. Using K-nearest neighbors and SVM with both quadratic and linear kernels, their model achieved a correlation of 0.65 with the average human ratings[3]. The same year, Amit Kagian and his team, also from Tel-Aviv University, trained their model on a dataset of 91 facial images of American females. By using 90 principal components of 6972 distance vectors between 84 fiducial point locations, they achieved a Pearson correlation of 0.82 with human ratings[1]. In 2009, two students in CS229, Hefner and Lindsay used a training set of 147 photos and a test set of 50 photos, all obtained from HotOrNot and got the maximum correlation with linear regression estimation[4]. Most recently, in July 2015, Avi Singh from Indian Institute of Technology, Kanpur used Guassian Process Regression, K-nearest neighbor, Random Forest, SVM and Linear Regression with leave-one-out cross-validation on a dataset of 500 Asian females [2]. He achieved correlations of 0.52, 0.6, 0.64, 0.22, 0.64 respectively.

Its noticeable that most previous experiments use a relatively small number of carefully chosen geometric features (less than 100). Through this project, we want to begin with a large number of redundant geometric features, and systematically reduce the count to a smaller number of more independent components which will also allow for easier computation.

II. DESCRIPTION OF DATA

We use three different datasets: the SCUT-FBP dataset, Chicago Face dataset, and images scraped from the HotOrNot dating website.

The SCUT-FBP dataset contains 500 frontal images of 500 different Asian females with neutral expressions, simple backgrounds, and minimal occlusion. These factors are conducive to facial beauty perception in both geometry and appearance. Each image is rated by 75 users, both males and females, on a discrete scale from 1 to 5. We computed the average rating score for each image and used these as the labels.

The Chicago Face Dataset includes high-resolution frontal photographs of 597 male and female targets of varying ethnicity. Each target is represented with a neutral expression photo that has been normed by an independent rater sample. Each photo is rated by 20-30 raters with the average rating being a continuous value between 1 and 5.

The HotOrNot dataset was scraped off the website HotOrNot.com website. Each image has a continuous rating from 0 to 10, which is the ratio between the number of people who like her and the total number of people who rate her profile as like/dislike. For example, if 8 out of 10 people like a person, that person would have a rating of 8. We scaled the ratings of HotOrNot dataset to [1,5] range so that they can be the same as the ratings of the SCUT-FBP dataset and Chicago Face dataset. This dataset is used for testing purpose only.

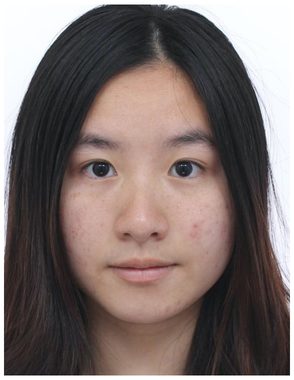

Fig. 1. A subject from SCUT-FBP Dataset with an attractiveness rating of 2.49

Fig. 2. A subject from Chicago Face Dataset with an attractiveness rating of 4.36

Due to time constraints, we only process frontal images of female subjects.

III. Methods

We used a multiclass logistic regression classifier without regularization to predict the average rating of each picture (as an integer between 1 and 5), using Python's sklearn library.

As an initial proof-of-concept test, we implemented a quick-and-dirty multilabel classifier using the SCUT-FBP dataset. To construct our feature set, we manually tagged 19 different points on 30 different faces using Python's ginput feature. We then used the pairwise distances between all pairs of points as the input features of our classifier.

To compute the testing error on our models, we used k-fold cross validation, choosing k to be $m/10$. The results of this simple model were promising, so we continued to refine the method to produce the final results.

Instead of manually tagging all 500 of our photos, we used two existing face detection APIs (faceplusplus.com, and facemarkapi (http://apicloud.me/apis/facemark/demo/)) to tag geometric landmarks on photos (seen in Fig. 3). Our API's produced a set of 68 landmarks, giving us a total of $\binom{68}{2} = 2278$ features. This is roughly six times the size of our data set. To avoid overfitting our data, we used Principal Component Analysis (PCA) to reduce the dimensionality of this feature space by an order of magnitude or more. Our analysis includes PCA-reduced feature numbers between 10 and 500. We ran linear regression on this reduced data, this time with $L1$ regularization.

We ran Support Vector Regression on the data without much success. This is described in the Results section.

IV. Results

Using regularized multi-class logistic regression, we were able to train a model with a maximum classification accuracy of $\mathbf{74.2\%}$. This put's our model's success rate at well above chance (i.e. 20%), and on par with the accuracies of other existing attractiveness-rating models.

Tables I and II show the confusion matrix and precision/recalls for our model.

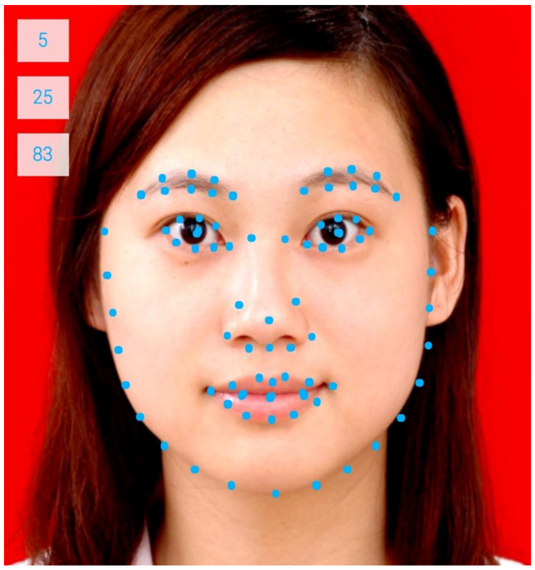

Fig. 3. Example picture from the SCUT-FBP dataset, with added labels indicated

TABLE I
Confusion matrix

	Actual 1	Actual 2	Actual 3	Actual 4
Predicted 1	13	15	0	0
Predicted 2	52	302	29	6
Predicted 3	1	4	20	4
Predicted 4	3	3	5	8

Note that while rating system allowed for values between 1 and 5, no average ratings of 5 were observed. Our model's most frequent error was in misclassifying 1's as 2's. In fact, the 2-1 entry in the confusion matrix is four times as large as the 1-1 entry. This might be related to the fact that we have so many instances of 2-rated profiles (324 of the 465 data points). Indeed, the 2-3 entry of the confusion matrix is also bigger than the 3-3 entry.

A. Parameter Tuning

We attempted training our models directly on our distance features, but found that the resulting models overfit our data. With logistic regression, training error hovered at around 0.4% while testing error was 41%. This is likely because we had many more features (~ 3000) than we had data points (~ 500).

To mitigate overfitting, we ran PCA on our data and trained our model on only the top several principal components. Figures 4 and 5 show the projections of our data onto the top few principal components.

The number of principal components used had a significant impact on the training and testing errors. Figure 6 shows that testing error generally increases with the number of principal components used, while training error decreases. However, testing error reaches a local minimum at 32 principal components, as shown in figure 7.

In training our model, we performed a grid search to

TABLE II
PRECISION AND RECALL

Label	1	2	3	4
Precision	0.464	0.776	0.69	0.421
Recall	0.188	0.932	0.37	0.444

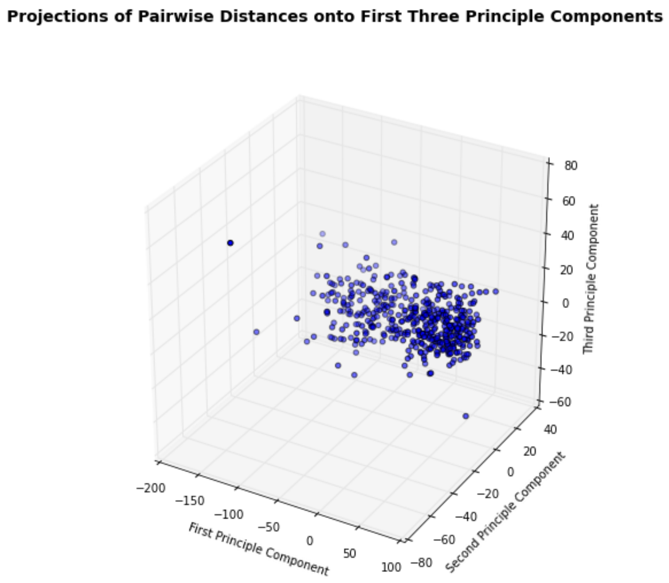

Fig. 4. Projections of our data points onto the first three principal components

optimize the coefficient of the $L1$ penalty term, C. We found that testing error seemed to increase monotonically with increasing C. The global minimum was for a value of $C = 0.08$. We chose a slightly larger value ($C = 0.15$) to further discourage overfitting of the data.

B. Visualizing the Principal Components

The well-known Eigenfaces method allows for a way of visualizing prototypical faces using principal component analysis. We were curious if we could visualize the principal components of our distance features in the same way. Each of our data points represents a planar embedding of points in the plane. We wanted to see what happens if we tried visualizing our principal components in the same way.

Our features represent pairwise distances between points in the plane. To reconstruct the set of points from their distance matrix, we computed the matrix M such that:

$$M_{ij} = (x^i - x^1) \cdot (x^j - x^1) = \frac{D_{i1}^2 + D_{j1}^2 - D_{ij}^2}{2}$$

where $D_{ij} = ||x^i - x^j||$. Then, the matrix M can be written as $M = XX^T$, where the ith row of X is $x^i - x^1$. X can then be found by computing the eigenvalues and eigenvectors of M.[5]

If the distance matrix D truly represents a planar embedding of points, then all but the first two columns of X will be zero. We computed our entries of D for the principal components by "unnormalizing" them, i.e. multiplying them by the dataset's standard deviation and adding the dataset's mean.

When we computed the matrix X for each of our principal components in this way, we found that its first two columns contained large nonzero values (on the order of 10^{-1}) and the subsequent columns contained smaller nonzero values

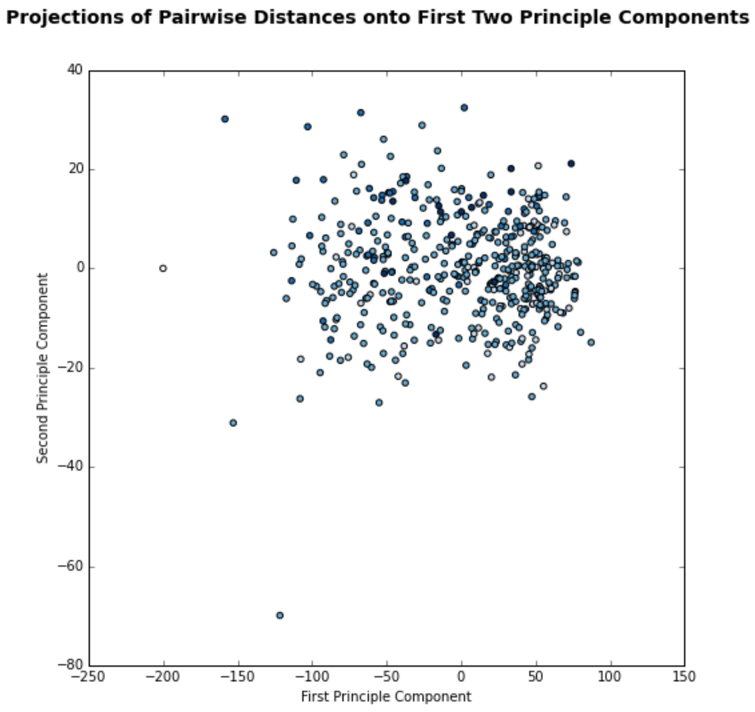

Fig. 5. Projections of our data points onto the first two principal components. Darker colors indicate higher attractiveness scores. It is not immediately obvious what is special about the outliers.

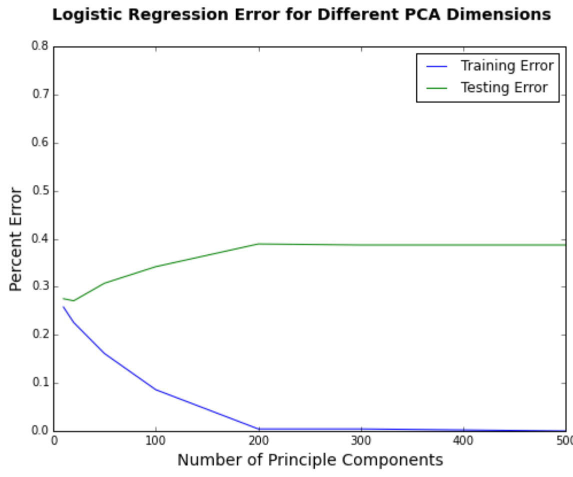

Fig. 6. Training and testing error as a function of the number of principal components used

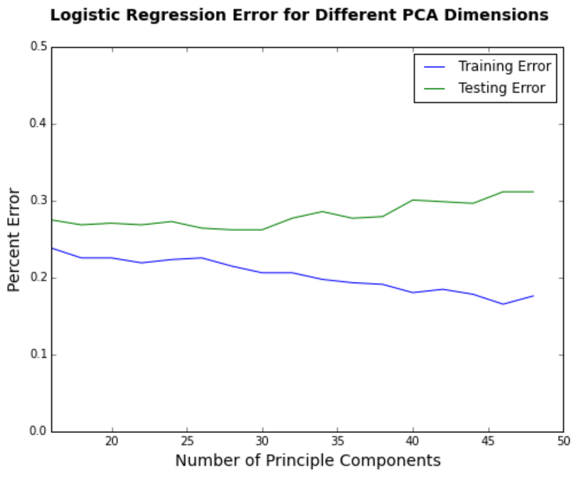

Fig. 7. Training and testing error, near the global minimum of testing error.

(on the order of 10^{-3}). Surprisingly, plotting the first two columns of X yields face-like images, as in figures 8 and 9.

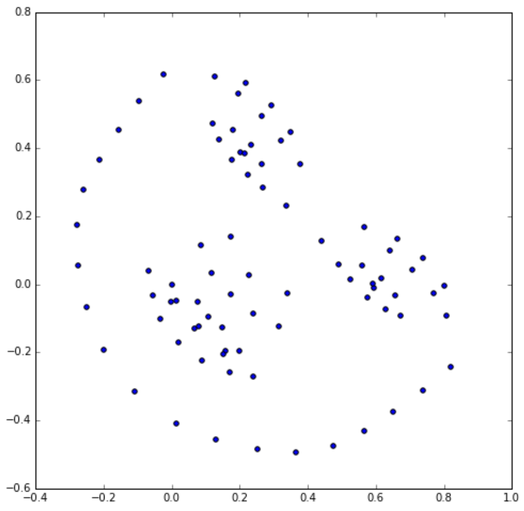

Fig. 8. Embedding of the first principal component

Upon further inspection, however, it was seen that all the principal components produce nearly identical embeddings. For example, the coordinate-wise variation between the first and last principal components was on the order of 10^-3. So, the embeddings of these principal components all resemble the average embedding of the dataset.

As an alternative way of visualizing the principal components, we posed the following optimization problem:

$$\min_{\zeta,x} \sum_{ij} \zeta_i j^2$$
$$\text{s.t. } ||x_i - x_j||^2 - D_{ij}^2 = \zeta_{ij}$$

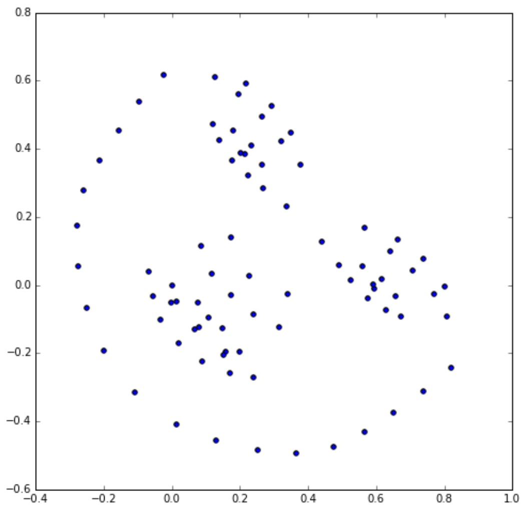

Fig. 9. Embedding of the last principal component. It's visually indistinguishable from the embedding of the first principal component.

Here, the ζ's are slack variables that represent how much our principal component deviates from a planar embedding. This is a quadratic optimization problem with quadratic constraints, which is in general an NP-hard problem. We attempted to use the "COBYLA" (Constrained Optimization BY Linear Approximation) optimization method using Python built-ins to solve the problem, but were not able to obtain good convergence.

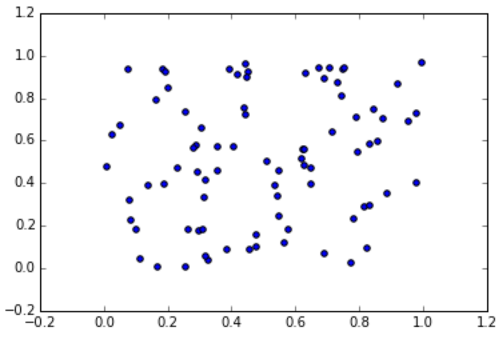

Fig. 10. Embedding of the fourth principal component via the optimization method, showing some signs of structure

It is unknown if visualizing the first principal components in this manner will yield any meaningful results. The principal components represent the directions in which our data points vary most significantly. This could correspond to how much a subject is rotating her head, how assymetrical her features are, or how significantly her features differ from some "prototypical" face. Pursuing the planar embedding of these principal components further would be an interesting topic for further research.

C. Attempts at Support Vector Regression

Our ratings are continuous, not categorical, so it makes sense to describe them with a regression model. We attempted to use Support Vector Regression to fit our data, but found that it drastically overfit our data, even when trained on the dataset's principal components and even when we performed a grid search to optimize its parameters. As shown in figure 11, our model's R^2 value for its training data is nearly 1 regardless of the number of principal components used, while the R^2 value for the testing data never rises above 0.3

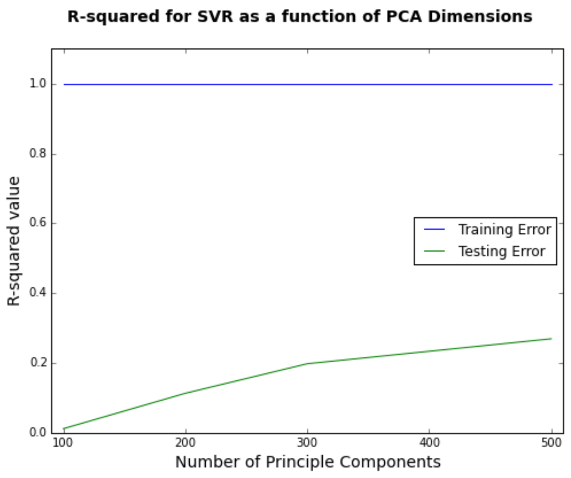

Fig. 11. R^2 values for training and testing data using SVR.

This is consistent with the results found in Hefner and Lindsay's 2006 machine learning project. More data or a different approach to selecting features may help make regression models a viable option.

V. Discussion

We found that based solely on the set of distances between key points on a photo, we can predict with as much as 74% the score on a scale from 1-5 of the attractiveness of a woman's face. This result is extremely significant considering the subjectivity involved in determining attractiveness, in addition to the many features not captured by this model including complexion and hair and eye color. This suggests that structure and positioning is the most important factor in finding an attractive face, assuming our data is reasonably distributed in other unmeasured parameters. This means that for women choosing which photos to post online, the way the face is depicted spatially can be a principally important consideration.

The most logical expansion of this project would be to analyze a more diverse dataset. More data should increase the accuracy with which one could predict the attractiveness of any face in general. It may also be interesting to be able to cluster faces based on differentiating features such as sex, ethnicity, and age. It is conceivable that certain feature distributions will be more attractive in certain demographics; perhaps there is a universally appealing facial structure, or perhaps different groups have different optimally attractive facial structures.

It is our hope that the results of this paper demonstrate an effective and relatively simple method for classifying faces based on attractiveness, as well as contribute to the understanding of the components of structure that comprise to this attractiveness. This research could be useful especially when combined with analysis of other non-spacial components mentioned above.

References

[1] Amit Kagian, Gideon Dror, Tommer Leyvand, Daniel Cohen-Or, Eytan Ruppin, *A Humanlike Predictor of Facial Attractiveness*, School of Computer Sciences, Tel-Aviv University, http://papers.nips.cc/paper/3111-a-humanlike-predictor-of-facial-attractiveness.pdf, 2007. Accessed November 2015.
[2] Avi Singh, *Computer Vision for Predicting Factial Attractiveness*, http://www.learnopencv.com/computer-vision-for-predicting-facial-attractiveness/. July 27, 2015. Accessed November 2015.
[3] Yael Eisenthal, Gideon Dror, Eytan Ruppin, *Facial Attractiveness: Beauty and the Machine*, School of Computer Science, Tel-Aviv University, Tel-Aviv, 2006.
[4] Jim Hefner, Roddy Lindsay, *Are you Hot or Not?*, Stanford University, http://cs229.stanford.edu/proj2006/HefnerLindsay-AreYouHotOrNot.pdf December 2006. Accessed December 2015.
[5] Bruno Bruck, (http://math.stackexchange.com/users/33347/bruno-bruck), *Finding the coordinates of points from distance matrix*, (version: 2012-06-09): http://math.stackexchange.com/q/156161