Optimal Local Basis: A Reinforcement Learning Approach for Gender and Age Classification

Hujia Yu
Department of Management Science & Engineering
Stanford University
hujiay@stanford.edu

Abstract

This paper approaches the gender and age classification problem using reinforcement learning. Specifically, it uses an approach introduced by Harandi [2] named Optimal Local Basis (OLB), which are a set of basis derived by reinforcement learning to represent the face space locally. Our best-performing age classification learner is OLB-20/10/-5 with accuracy 28.85%, which learns from sampling 20 images per age per gender class, and performed Q-learning with reward of 10 and punishment of 5. Our best-performing gender classification learner is OLB-20/1/-1, which was trained from 20 images per gender per age group, with reward of 1 and punishment of -1. It reached 58.43% accuracy during testing. Our analysis shows that for the same train data, eigenface models with Q-learning perform better than without. For OLB-learner, more train images lead to better learning power, and higher accuracy. Finally, higher ratio of reward to punishment score during Q-learning leads to higher test performance.

1 Introduction

Automatic age and gender classification has become relevant to an increasing amount of applications, particularly since the rise of social platforms and social media. Nevertheless, performance of existing methods on real-world images is still significantly lacking, especially when compared to the tremendous leaps in performance reported for the related task of face recognition [1].

This project approaches the gender and age classification problem using reinforcement learning. Specifically, it uses an approach introduced by Harandi [2] named Optimal Local Basis (OLB), which are a set of basis derived by reinforcement learning to represent the face space locally. Unlike deep learning approaches that use a single basis for all individuals, learning OLB for each face image benefits from local information by incorporating different bases.

2 Background and Related Work

The same task has been investigated by Levi and Hassner[1], who attempt to close the gap between automatic face recognition capabilities and those of age and gender estimation methods using deep convolutional neural networks (CNN) [3]

There has been a tremendous amount of research in the highly-related field - face recognition. Among different approaches devised for face recognition, the widely studied ones are the statistical learning methods that try to derive an appropriate basis for face representation [5]. The reason behind deriving a basis is because a complete basis makes the derivation of unique image representations suitable for processes like image retrieval and object recognition.

Harandi [6] presented a novel learning approach for Face Recognition by introducing Optimal Local Basis, which serves as the main framework for this project. Optimal local bases are a set of basis derived by reinforcement learning to represent the face space locally. The reinforcement signal is designed to be correlated to the recognition accuracy. The optimal local bases are derived then by finding the most discriminant features for different parts of the face space, which represents either different individuals or different expressions, orientations, poses, illuminations, and other variants of the same individual.

In addition to the regular PCA methods being used largely on images, Feng [7] also introduced PCA on wavelet subband to tackle PCA's limitations: poor discriminatory power and large computational load. In the proposed method, wavelet transform is used to decompose an image into different frequency subbands, and a mid-range frequency subband is used for PCA representation. In comparison with the traditional use of PCA, the proposed method gives better recognition accuracy and discriminatory power.

This project did not use wavelet subband approach at this time, but it is definitely a promising method to try in the future.

The above papers are the main inspirations for the approaches and frameworks used in this project.

3 Approach

3.1 Reinforcement Learning Problem Formulation

Reinforcement Learning(RL) is a machine learning technique for solving sequential decision problems. In gender and age classification problems, input is a series of image dataset. Traditionally, a single basis is used to represent an image for classification models, realizing that this may not be efficient in all cases, I will approach this image classification problem with feature extraction using Q-learning, where feature pools are extracted from a holistic feature extraction method such as Principal Component Analysis (PCA), and the agent then learns to extract optimal feature basis from this feature pool through Q-learning.

Therefore, the environment of this RL problem becomes feature pools, and selected features, reward is defined to help to reinforce the subset of features that have high representational powers and to punish those that represent poorly of the original picture. In order for the learning process to be efficient, each time step is modeled as a nth order Markov Decision Process, where current state is defined to be last n selected features. During sorting and testing for the optimality of selected feature basis, K-nearest-neighbors(KNN) is used to cluster images represented similarly in each distinct feature space. The formulation of this image classification RL problem is further illustrated below:

3.1.1 Q-Learning

According to definition in [2], The value of agents state-action pairs are modeled by $n + 1$ Q-tables $(Q_0, Q_1, ..., Q_n)$ of size $(N, N \times N, N \times N \times N, ..., N \times N \times N \times N)$. The element n+1 $(Q_j(i_0, i_1, ..., i_{j-1}, i_j)$ of the Q-table Q_j demonstrates the expected value of received reward by selecting feature i_j when the features $(i_0, i_1, ..., i_{j-1})$ are already selected in j previous steps. For example in a first order MDP, element (i_0, i_1) of Q_1 demonstrate the expected reward of selecting direction i_1 when i_0 is previously selected. The updating equations for Q-learning algorithm is:

$$Q^j_{(i_0,i_1,...,i_{j-1},i_j)} = Q^j_{(i_0,i_1,...,i_{j-1},i_j)} + \alpha(r + \gamma \max_{l=1}^{N} Q^j_{(i_0,i_1,...,i_{j-1},i_j,i_l)} - Q^j_{(i_0,i_1,...,i_{j-1})}) \tag{1}$$

where $Q^j_{(i_0,i_1,...,i_{j-1},i_j)}$ is the expected reward of selecting feature i_j when the agent is in the state $(i_0, i_1, ..., i_{j-1})$; $Q^j_{(i_0,i_1,...,i_{j-1},i_j,i_l)}$ is the expected reward of selected i_l one step after selecting feature i_j, r is the received reward of the selecting feature i_j, $\alpha(0 < \alpha \leq 1)$ is the learning rate and $\gamma(0 \leq \gamma \leq 1)$ is the discount factor.

Figure 1: A schematic view of the learning system.

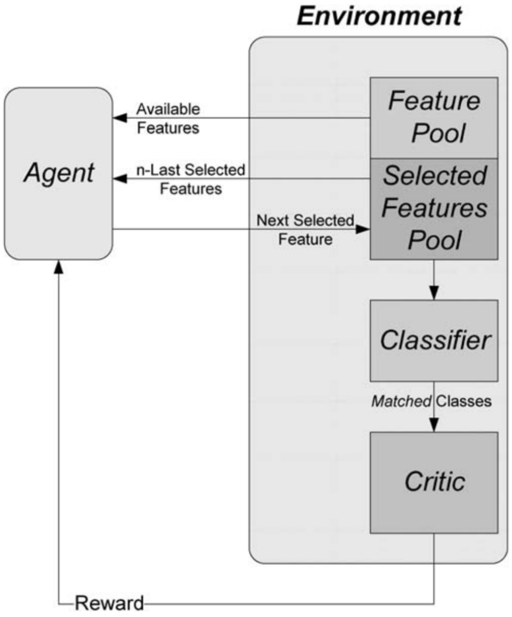
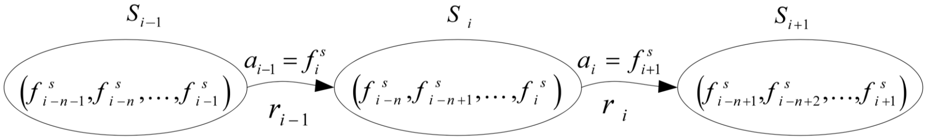


Figure 2: A nth order MDP model of the environment, where f_j^s is the j th selected feature and a_i is the agents action at state s_i


S_{i-1}
S_i
S_{i+1}
$(f^s_{i-n-1}, f^s_{i-n}, \cdots, f^s_{i-1})$
$a_{i-1} = f^s_i$
r_{i-1}
$(f^s_{i-n}, f^s_{i-n+1}, \cdots, f^s_i)$
$a_i = f^s_{i+1}$
r_i
$(f^s_{i-n+1}, f^s_{i-n+2}, \cdots, f^s_{i+1})$

3.1.2 Markov Decision Process (MDP)

As mentioned above, the environment is modeled as an nth order MDP where the current state S_i is represented by n last selected features $f = (f^s_{i-n}, f^s_{i-n+1}, ..., f^s_i)$ as shown in figure above. I also keep track of the selected states by keeping a binary vector **h** of length N, where N is the dimension of feature space. This is to ensure that a feature is not selected more than once. Whenever a feature is selected the corresponding element of vector **h** becomes one. The agent can select only those features that their corresponding elements in **h** are zero.

3.2 Eigenfaces

The Eigenface method was introduced by Turk and Pentland in [6], where this approach treats face classification as a two-dimensional classification problem. Face images are first projected onto a feature space (face space) that best encodes the variation among known face images. The face space is defined by the eigenfaces, which are the eigenvectors of the set of faces; they do not necessarily correspond to isolated features such as eyes, ears, and noses. The framework provides the ability to learn to recognize faces in an unsupervised manner. This method has since been popular due to its great performance. Eigenface method will be used to benchmark model performance in this paper.

3.3 Optimal Local Basis (OLB)

I formulate this problem as a image classification problem where the feature vectors are represented by N dimensional vectors and each datum belongs to one of the classes $w = (w_1, ..., w_C)$ An OLB is defined as a three-tuple (x_i, w_i, T_i), where x_i is the OLB representative point, $w = (w_1, ..., w_C)$ is the OLB class label, T_i is a binary vector expressing the set of features associated with the OLB, and N is the set of all the N dimensional binary vectors excluding the null binary vector. For instance, in a face recognition task where the features are obtained by PCA, x_i is the representation of a sample face in the Eigenface space, N is the number of Eigenfaces, and $T_i)$ is the binary vector demonstrating which Eigenfaces are the optimal subset to describe $x_i)$. In this paper, features, eigenvectors, eigenfaces essentially mean the same thing and are used interchangeably.

4 Algorithms and Time Complexities

4.1 Phase 0: PCA - Feature Pools

This is a preparation part of the OLB-learning process. Let m be the size of train image data, to extract eigenfaces corresponding to the train images, we apply the following to the train data.

1. For each image, we change it from RGB to Grayscale, cropped the size to be $[224 \times 224]$ and then concatenate into a row vector, after preprocessing all, we have an input data of size $L = m \times 50176$.

2. We calculate its covariance matrix $C = L^T L$, resulting in dimension $[50176 \times 50176]$

3. Get eigenvectors and eigenvalues of dimension $[50176 \times m]$, $[1 \times m]$, where each column is an eigenvector, with its corresponding eigenvalue.

4. weights of all train images projected onto this eigenvector space can be simply calculated as $L \times eigenvectors = [m \times m]$, where each row i represents weights of ith image when projected onto this eigen-space.

Now we have initiated each image as an object of $OLB(x_i, w_i, T_i)$, where x_i is representation point of OLB, which is just the weight vector corresponding to image i, w_i is the class label of image i, and T_i is a binary vector of size N indicating the optimal feature basis of that OLB, and it is initialized to be 0.

4.2 Phase 1: OLB-Learning

The learning process is applied to each $OLB(x_i, w_i, T_i)$ where the classification problem is modeled by the set $OLB(x_1, w_1, T_1), OLB(x_2, w_2, T_2), ..., OLB(x_m, w_m, T_m)$ Here m is the number of $OLBs$ and $m \geq C$.

During reinforcement learning, The agents task is to learn those features for $OLB(x_i, w_i, T_i)$ that result in the collection of maximum expected reward. Basically, at each time step t_j, the agent chooses a feature f_j with the highest Q-value with ϵ greedy policy in the corresponding Q-table. It then updates that cell's q-value in the Q-table by looking ahead one step according to (1). In order to update all Q-tables, number of MDP iterations equivalent to the dimension of Q-table (N) is required. Also, in order for the q-values in all Q-tables to converge to its optimal values, number of epochs equivalent to number of eigenfaces is required. Therefore, time complexity for learning part of this model requires $O(m \times N \times N)$, where N is the number of eigenvectors extracted from sample data, and m is number of train data. Since for PCA, the maximum number of principal components is all of its eigenvectors, which is just size of train data, so if we set our N to be m, the time complexity for phase 1 of the model is $O(N^3)$. Notice that if without MDP, selection of the best feature subset in an N dimensional space demands for examining $2N1$ possibilities per OLB. MDP has improved efficiency greatly. Pseudo-code is presented in the table attached in Appendix.

4.3 Phase 2: OLB-Classifier

After completing the learning process, the set of optimal features for each OLB should be selected. Now, the question is how to determine the optimal features for each OLB. To do this, firstly the features are sorted according to their discrimination using the available Q-tables and then from the sorted features the optimal binary vector T_i is extracted.

To obtain a set of features in descending order of best performance, we use the already learned Q-tables in recall mode. For an nth order MDP, the first n appropriate features are obtained by selecting the position of the maximum in $Q_j, j = 0, 1, 2, ..., N$, respectively. In each selection step, the corresponding Q-table and the selected features are used. Pseudo-code is attached in Appendix below.

When the training phase is finished and the optimal features for each OLB are selected, we are ready to build an OLB-based Classifier. To this end, we need to assess the similarity of a query image to all stored classes. Simple similarity judgment based on ordinary distance measures on feature space does not work here since $OLBs$ have different dimensions and features.

In order to make different bases comparable, we use the reward signal as the similarity measure. For a given query x_q, for every learnt OLB, We project it and every other train image into that OLB's optimal local basis, and calculate its similarity by summing the correct predictions based on projection.

$$S(x_q, OLB(x_i, w_i, T_i)) = \sum_{j=1}^{K} R_C(j) f_j(w_j)$$

Then, class similarity measure is defined by fusing the similarity measures of all the OLBs belonging to it, i.e. $S(x_q, OLB(x_i, w_i, T_i))$. For fusing the $S(x_q, OLB(x_i, w_i, T_i))$:

$$S_{Class} = \sum_{i=1}^{n} S(x_q, OLB(x_i, Class, T_i))$$

5 Experiments

5.1 Data

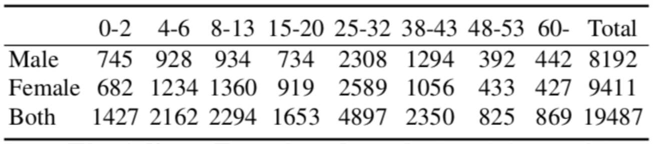
Many previous methods has focused on constrained images taken in lab settings. Such settings do not adequately reflect appearance variations common to the real-world images in social websites and online repositories. Therefore, the dataset that I chose is Adience [4] collection of unfiltered faces for gender and age classification. The sources of the images included in the set are Flickr albums, assembled by automatic upload from smart-phone devices, and released by their authors to the general public under the Creative Commons (CC) license [3]. Im using the same dataset that Levi and Hassner used in [2], which had reached best gender classification accuracy of 86.8%, and best age classification of 50.7% [2].

This dataset contains 26,580 number of photos, 2,284 number of subjects (users). It contains gender label for each photo as well as the age group that the subject falls into.

The gender has two class labels: ['m', 'f'], and age is separated into 8 groups: [0-2, 4-6, 8-13, 15-20, 25-32, 38- 43, 48-53, 60-].

In the original paper where Harandi introduced Optimal Local Basis, he used 5 images per class during training/Q-leaning phase. Therefore, i decided to sample m number of images per age group per gender group, that is, a total of $m \times 2 \times 8 = 16m$ images, where m varies between $[5, 10, 15, 20]$, and I split them into $60 : 40$ ratio for train and test dataset. The same train and test images were used for both age and gender classification for consistency.

The specific breakdowns of the number of images in each group is shown in the table below.



	0-2	4-6	8-13	15-20	25-32	38-43	48-53	60-	Total
Male	745	928	934	734	2308	1294	392	442	8192
Female	682	1234	1360	919	2589	1056	433	427	9411
Both	1427	2162	2294	1653	4897	2350	825	869	19487

Table 1. **The AdienceFaces benchmark.** Breakdown of the AdienceFaces benchmark into the different Age and Gender classes.

5.1.1 Data Preprocessing

Each image is cropped to be of dimension size 224, RGB is converted to gray scale to reduce the dimension to $[224 \times 224]$, sample mean feature value is subtracted from each image before obtaining its feature space representation using PCA. Evaluation metrics are top-1 and top-2 accuracies.

5.2 Models

I explored a variety of models with different number of images for RL-learning. Specifically, we started with small images per group, and small rewards to punishment ratio for prediction during Q-learning step, and slowly increased to more images per class used and higher ratio of reward to punishment. In order to maximize the number of features the agent can learn from, I did not discriminate number of principal feature components. I set all eigenvectors to be equal to the total number of principal components corresponding to each training group. Since during PCA reduction, the number of principal components is limited by number of samples, so the principal feature pools is always of the same size as the training data size.

I also used top-1 as accuracies. I performed Eigenfaces classification introduced by Turk [5] as my baseline model, which is simply classify an input image to the class of its closest neighbor in the eigenface space. The results are shown in table below.

6 Results

6.1 Quantitative Performance

Model	AGE TOP-1	GENDER TOP-1
Eigenface-10	21.43	55.35
OLB-5/1/-1	21.46	53.57
OLB-5/1/-5	21.95	53.57
OLB-10/1/-1	21.98	55.64
OLB-10/1/-5	22.64	57.14
OLB-10/5/-1	22.34	57.14
OLB-15/1/-5	23.91	57.04
OLB-15/10/-1	28.77	57.17
OLB-20/1/-1	25.93	**58.43**
OLB-20/10/-5	**28.85**	58.12

First of all, the majority of the OLB-learners achieved better performance than my baseline model, which is an Eigenface classification model based on 10 images per gender per age group. Only two learners did not perform as well, which are the OLB-5/1/-1, and OLB-5/1/-5 that achieved lower accuracy score on gender classification task. This could be because of its small amount of train data. This indicates that Q-learning helps with image classification tasks on facial images.

According to our results above, overall, the accuracies increase as you go down the table, indicating a positive correlation between number of images used during Q-learning, and its test accuracies on classification problems. This makes sense because even though the process of Q-learning itself only looks at the current state and next feature for each image, which does not seem to be directly related to the size of total sample images, the total number of images plays an important role in the earlier step of PCA where eigenfaces are extracted from the sample data.

Basically, since the number of eigenvectors is limited by the input data size, number of principal components extracted from covariance of m images is at most m because that is the number of total eigenvectors. This means that more input images will result in a larger dimension of feature basis available for Q-learning in the next step. More eigenfaces to perform Q-learning can be beneficial due to possible higher representational power than lower dimensional feature spaces. Therefore, more images during the Q-learning process will help improve model accuracy.

Our best-performing age classification learner is OLB-20/10/-5 with accuracy 28.85%, which learns from sampling 20 images per age per gender class, and performed Q-learning with reward of 10 and punishment of 5. The second-best age classification learner is OLB-15/10/-1 with an accuracy of 28.77%, this learner learns from a smaller sample size - 15 images per age per gender, and it also has less punishment for incorrect predictions.

Our best-performing gender classification learner is OLB-20/1/-1, which was trained from 20 images per gender per age group, with reward of 1 and punishment of -1. It reached 58.43% accuracy during testing.

From each comparison group with the same train data size but different reward/punishment ratio. For example, OLB-15/1/-1 and OLB-15/10/-5, reward/punishment ratio doubled, but dataset is still the same, accuracy on age classification increased from 23.91% to 28.77%, and that on gender classification increased slightly. The same trend can be observed for other groups having same train data size. Higher ratio of reward to punishment seems to have a positive effect on learner's performance. This point is further examined in later sections.

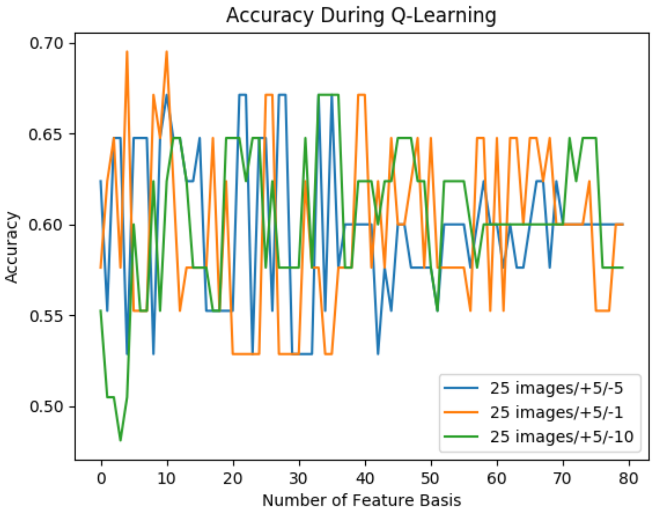
Overall, the results indicate the follows. For the same size of train data, eigenface models with Q-learning perform better than without. For OLB-learner, more train images lead to better learning power, and therefore higher accuracy. Finally, higher ratio of reward to punishment leads to higher performance.

6.2 Analysis

6.2.1 Reward and Punishment Scores

As shown in the results table above, the ratio of reward and punishment parameters affects how the agent learns. Therefore, I kept track of fluctuations of prediction accuracy during gender classification Q-learning phase with train sample size of 240, and compared the curve for the learners with same sample data but different reward to punishment score ratio.

Figure 3: Gender classification accuracies during Q-learning for learner formulated by 25 images per age group per gender group (240 train images in total), with correct reward and punishment of (5,-5), (5,-1),(5,-10) respectively.



Since this is a snapshot of accuracy changes for one epoch of iterations, the x-axis is number of iterations, which is also the number of feature basis selected, since at each iteration, the agent selects the next best feature vector that could maximize its long-term reward.

From this graph, there is no obvious relationship between difference in reward, punishment scores and accuracies for learner of sample size 240 for at least one epoch of training. Potentially there might be more obvious effect as number of epochs increases, but the possible trend is not captured.

Also, according to this graph, the variance of accuracies for all three learners seem to decrease as more number of feature vectors are chosen. This trend strongly makes sense, since when dimension of feature space is too low, it is not representative of the original image and is less likely to make correct prediction, resulting in low accuracies. Therefore, as dimension of feature space increases, more information of the original image can be captured and thus better decisions can be made.

6.3 Qualitative Performance

6.3.1 PCA-Reconstruction

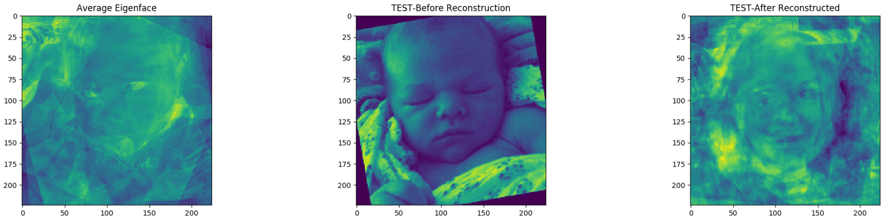
Eigenfaces can be extracted from a group of sample data, I extracted the average eigenface of the input sample data of 10 images per gender per age group, therefore from a total of $10 \times 8 \times 2 = 160$ images. PCA is implemented on $160 \times (224 \times 224)$ sample data, and the corresponding eigenvectors is of dimension 160×50176 (each row eigenvector represents an eigenface) with eigenvalues(weights) of dimension 1×160. I then reconstructed the average eigenface image by adding all eigenvectors with weights set to be 1.

One thing I need to point out is that our dataset contains a large variety of images, including different expressions, orientations, poses, illuminations, number of faces, and other variants, making it very challenging to extract eigenfaces. As a result, the extracted average eigenface looks very different from an actual face. Yet it still maintains the major facial features such as location of eyes, mouth, nose, and ears, etc.

I have tested performance on PCA-reconstruction on one image using the top 50 eigenfaces, the reconstructed image is plotted in the figure below.

From the reconstructed image, one can clearly see that it resembles the average face significantly more than the original face, but still maintains some distinctive features of the original face.

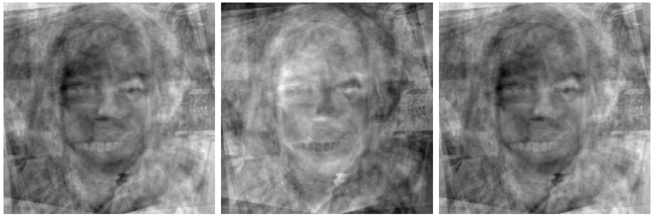
Figure 4: (From left to right) Average eigenfaces of the sample data. Sample image before and after PCA reconstruction based on the sample mean face



6.3.2 Feature Pools

To visualize what feature pool looks like, I have also extracted the top 3 eigenfaces from 160 samples, shown in figure above. Due to the small sample size and the large variance in between images, the extracted faces are not representative of any specific face, rather they are just overlaps of different facial features according to where they frequently appear to be in the sample images. The extracted eigenface shows overlapping layers of facial features such as hair, eyes, and teeth, etc. The images can be scary to loot at times, thus eigenfaces also have a name referred to by others as ’ghostface’.

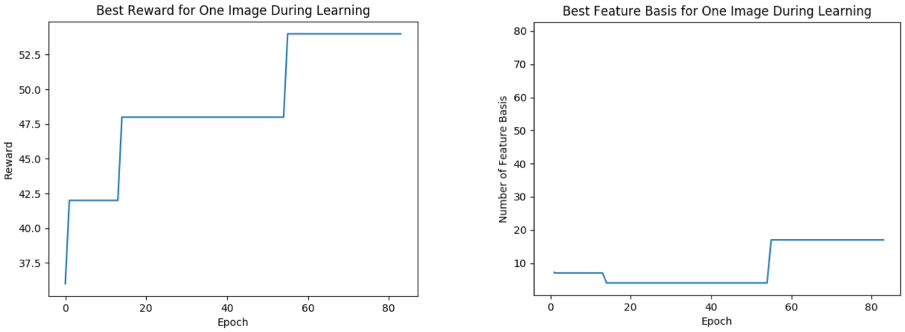
Figure 5: (From left to right) Top 3 eigenfaces from sample data



6.3.3 Best Reward vs. Best Number of Features

One thing that would be interesting to look at is the relationship between reward and dimension of feature space, namely do more features necessarily lead to higher reward during learning process. In order to answer this question, I kept track of best reward and its corresponding best-performing feature space at the end of each epoch, and plotted in figure below. Best number of features is calculated as the dimension of feature space corresponding to the best reward received during **current epoch** of iterations.

Figure 6: (From left to right) Best reward, and best number of feature basis during one epoch of age classification training for one training data over 84 epochs with 10 images per age per gender and positive score of 5 and punishment score of -1



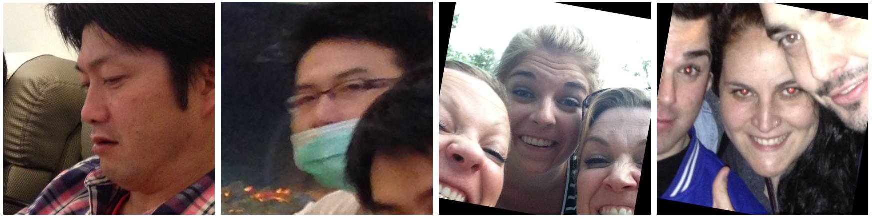
From the figure on best reward, we can see clearly from the graph that best reward increases with step sizes as more epochs of training pass. More specifically, it takes significant number of iterations for the agent to get higher reward. It took on average 20 epochs of training for best reward to improve by one more correct prediction on average. Whereas for the best feature basis, for the learning problem with parameters as stated here (10 images per class group, and correct score +5, punishment score -1), the best feature space does not necessarily correspond to the best reward, a lower dimension of feature space is also likely to produce high reward, which is as expected.

7 Error Analysis

7.1 Challenging Dataset

One great challenge is that the dataset is diverse. Unlike traditional facial recognition image data that are usually prepared in a lab setting. The Adience dataset are images uploaded by users on social media and thus they can have very different expressions, orientations, poses, illuminations, number of faces, and other variants, making it very challenging to extract eigenfaces. Examples of some challenging sample images are shown in the figure below.

Figure 7: (From left to right) Shot from the Side; face with mask; multiple faces;multiple faces



Since Eigenface-based image classification models rely heavily on overlapping features in between different images to extract features that are more representative, having a diverse train image dataset does not help in improving accuracies of Eigenface-based models. More train images can possibly offset this disadvantage brought by image variance, due to its higher representational power.

8 Conclusion and Future Work

In this paper, we discussed our approaches to building a Reinforcement Learning model to image classification tasks on gender and age classification using optimal local basis (OLB). OLBs are a set of basis derived by reinforcement learning to represent the face space locally. The optimal local bases are derived by finding the optimal subset of features for different parts of the face space that correspond to the class. Therefore, unlike most of the existing approaches that solve the recognition problem by using a single basis for all individuals, our proposed method benefits from local information by incorporating different bases for its decision.

Our best-performing age classification learner is OLB-20/10/-5 with accuracy 28.85%, which learns from sampling 20 images per age per gender class, and performed Q-learning with reward of 10 and punishment of 5. Our best-performing gender classification learner is OLB-20/1/-1, which was trained from the same train data, but with reward of 1 and punishment of -1. It reached 58.43% accuracy during testing.

Our results show the follows. For the same size of train data, eigenface models with Q-learning perform better than without. For OLB-learner, more train images lead to better learning power, and higher accuracy. Finally, higher ratio of reward to punishment leads to higher performance.

In the future, I'm interested to explore with different image dataset that are more consistent within class groups, such as AR, PIE, ORL and YALE databases.

Furthermore, due to inefficient computation of PCA method, I'm interested to apply wavelet subband method proposed by Feng [7] to tackle PCA's problems of poor discriminatory power and large computational load. In wavelet subband method, wavelet transform is used to decompose an image into different frequency subbands, and a mid-range frequency subband is used for PCA representation. In comparison with the traditional use of PCA, the proposed method is said to give better recognition accuracy and discriminatory power.

References

[1] Gil Levi and Tal Hassner, Age and Gender Classification using Convolutional Neural Networks, Department of Mathematics and Computer Science, The Open University of Israel

[2] Mehrtash T. Harandi, Majid Nili Ahmadabadi and Babak N. Araabi, "Optimal Local Basis: A Reinforcement Learning Approach for Face Recognition,

[3] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. Neural compu- tation, 1(4):541551, 1989. 1, 3

[4] Adience Dataset: http://www.cslab.openu.ac.il/

[5] Turk, M., Pentland, A. (1991). Eigenfaces for recognition. Cognitive Neuroscience, 3, 7186.

[6] Optimal Local Basis: A Reinforcement Learning Approach for Face Recognition

[7] Feng, G. C., Yuen, P. C., Dai, D. Q. (2002). Human face recognition using PCA on wavelet subband. Electronic Imaging, 9, 226233.

[8]O. M. Parkhi, A. Vedaldi, A. Zisserman, Deep Face Recognition, British Machine Vision Conference, 2015

[9] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, Martin Riedmiller, DeepMind Technologies. Playing Atari with Deep Reinforcement Learning

[10] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, 2014

9 Appendix

Algorithm OLB Learning

```
Select randomly an OLB OLB($\mathbf{x}_i, \omega_i, \mathbf{T}_i$) from the training dataset
{OLB($\mathbf{x_1}, \omega_1, \mathbf{T}_1$), OLB($\mathbf{x}_2, \omega_2, \mathbf{T}_2$), . . . , OLB($\mathbf{x}_i, \omega_i, \mathbf{T}_i$), . . . ,
OLB($\mathbf{x}_m, \omega_m, \mathbf{T}_m$)}.
Initialize the corresponding $Q$-tables randomly.
for iteration = 1 to _number_of_Episodes do

$\mathbf{h} = \left(\overbrace{0, 0, \ldots, 0}^{N}\right)$

  repeat
      Select a feature $a_i = f^s_{i+1}$ by $\varepsilon$ greedy policy.
      Update the selected feature vector, $\mathbf{h}(f^s_{i+1}) = 1$.
      Project all the representative points $\mathbf{x}_j,\ j = 1, \ldots, m$
      in training dataset into space defined by $\mathbf{h}$ using
      $\mathbf{p}_j = \mathrm{diag}(\mathbf{x}_j \otimes \mathbf{h})$.
      Find the class labels of the $K$-nearest neighbors of the
      projected data from $\mathbf{p}_i$.
      Update the corresponding cell of $Q$-table according to (4).
  until in $C_{hits}$ consequent steps, the agent receives the maximum
       reward or if all the features are selected.
end for
```

Algorithm Sorting the features according to their appropriateness

```
$fC = 1$

$h = \left(\overbrace{1, 1, \ldots 1}^{N}\right)$

$a_0 = \arg\max_j (Q^0(a_j))$

$ordered\_features = (a_0)$

$h(a_0) = 0$

$cState = (a_0)$

while $fC < n$ do
    Select all the $Q$ values described by $cState$ in $Q^{fC}$.
    This is a vector of length $N$ which is called
    $Row(Q^{fC}) = Q^{fC}(cState, a_j),\ j = 1, \ldots, N$.
    $a_{fC} = \arg\max_j (Row(Q^{fC}) | h(j) = 1)$
    $ordered\_features = (ordered\_features, a_{fC})$
    $rem\_features(a_{fC}) = 0$
    $cState = (a_0, a_1, \ldots, a_{fC})$
    $fC = fC + 1$
end while
while $fC \le N$ do
    Select all the $Q$ values described by $State$ of $Q^n$, $Row(Q^n)$
    $= (cState, a_j),\ j = 1, \ldots, N$. This vector has length $N$.
    $a_{fC} = \arg\max_j (Row(Q^n) | h(j) = 1)$
    $ordered\_features = (ordered\_features, a_{fC})$
    $h(a_{fC}) = 0$
    $cState = (a_{fC-n}, a_{fC-n+1}, \ldots, a_{fC})$
    $fC = fC + 1$
end while
```