RankBoost with $l_1$ regularization for Facial Expression Recognition and Intensity Estimation

Peng Yang$^1$, Qingshan Liu$^{1,2}$, Dimitris N. Metaxas$^1$

$^1$Computer Science Department, Rutgers University, Piscataway NJ 08854, USA,
$^2$National Laboratory of Pattern Recognition, Chinese Academy of Sciences Beijing, 100080, China

Abstract

Most previous facial expression analysis works only focused on expression recognition. In this paper, we propose a novel framework of facial expression analysis based on the ranking model. Different from previous works, it not only can do facial expression recognition, but also can estimate the intensity of facial expression, which is very important to further understand human emotion. Although it is hard to label expression intensity quantitatively, the ordinal relationship in temporal domain is actually a good relative measurement. Based on this observation, we convert the problem of intensity estimation to a ranking problem, which is modeled by the RankBoost. The output ranking score can be directly used for intensity estimation, and we also extend the ranking function for expression recognition. To further improve the performance, we propose to introduce $l_1$ based regularization into the Rankboost. Experiments on the Cohn-Kanade database show that the proposed method has a promising performance compared to the state-of-the-art.

1. Introduction

As early as 1970s, facial expression analysis has attracted some attention in the field of psychology. The two typical pioneer works are: categorizing facial expression into six basic expressions (happiness, sadness, disgust, surprise, anger, and fear) proposed by Izard [9] and Facial Action Coding System (FACS) designed by Ekman and Friesen [6]. FACS is a comprehensive standards to decompose each expression into several special active units (AUs). Although it seems more precise, the definitions of AUs are actually ambiguous semantic descriptions, which make automatic AUs detection and AUs-based expression analysis very difficulty in practices. Thus, in the communities of computer vision and pattern recognition, most of automatic facial expression analysis works are still based on six basic expressions [20][31][10][4] [8][30][25][29][3]. All these works aimed at automatically identifying an input facial image or sequence as one of six basic expressions, and some studies have obtained good performances in special cases. However, only classifying expression into such six basic categories is insufficient to further understand human emotion, which has been demonstrated by recent psychological studies [1]. Besides the categories of expression, facial expression dynamics is important to decipher its meaning, which is actually about expression intensity variation in temporal domain. Expression intensity estimation has lots of potential applications in human-robot interaction, patient monitoring, security surveillance and entertainment. So far, a few works addressed this issue. In this paper, we will focus on how to estimate facial expression intensity besides expression recognition.

1.1. The Problems

In FACS, the intensities of AUs are divided into 5 levels $A, B, C, D, E$. Figure 1 shows the 5 levels of AU 25, which are specified by how far the lips are parted.

![Figure 1. The example of AU25 on 5 levels (A, B, C, D, E).](image)

However, FACS does not give a computational definition on them, so the automatically labeling such 5 levels is still a open problem. To investigate the discrimination of 5 levels, [21] used the Gaussian models to simulate the distributions of 5 levels of AU 25 based on Line Local Embedding (LLE) shown in Figure 2. We can see that there exists big overlaps. The overlaps are actually caused by two main reasons: 1)
FACS does not give a quantitative measure between levels;
2) The variances exist in different subjects.

The above issues also exist in facial expression intensity estimation. Facial expression is a dynamic process from onset, to apex, then offset [12], but the intensities of the apexes are various due to different subjects. For example, someone shows the happiness with smiling, while someone may laugh loudly. It is hard to give a quantitative measure of intensity too. These issues make intensity estimation more difficult than recognition. Moreover, intensity estimation is not a hard decision problem, so conventional classification methods are unsuitable. Because we can not have ground-truths of absolute intensities, conventional regression methods can not be used too. In addition, expression intensity estimation has to rely on expression recognition, because they are coupled together.

1.2. The Proposed Framework

In this paper, we propose a novel ranking-based framework of facial expression analysis, which can do both expression recognition and its intensity estimation. It is based on this observation that the pair-wise ordinal relationship along the temporal domain is obvious, despite the short of quantitative measurement of intensity. Thus, it is easy to model the intensity estimation as a ranking problem. Figure 3 summarizes the proposed framework. It consists of three components: 1) facial appearance feature representation. We use the haar-like features to represent facial appearance due to its good properties, especially in facial appearance representation [26][27][30]; 2) Ordinal pair-wise data organization. It describes how to organize the data to make it suitable for the ranking model; 3) Building ranking model. This is the core component. Due to a large number of haar-like features existing, we propose to use the RankBoost [7] to select a subset of haar-like features to build a final strong ranker. In order to further improve the performance, we introduce the $l_1$ based regularization into the RankBoost. The final ranking score given by ranking function $H(x)$ can be used for expression intensity estimation and recognition. The details will be addressed in Section 3.

2. Related Works

A lot of studies have been proposed for expression recognition, but a few works addressed the issue of expression intensity estimation. Some studies attempted to use the replacements of facial landmarks as features for estimating expression intensity. For example, Reilly [21][22] used Local Linear Embedding (LLE) and Kernel Principal Component (KPCA) to reduce the feature dimensions, and then used SVM to train the classifiers of different intensity levels. Although small errors were obtained on the training set, the performance on the unseen data was not good, because LLE has a "out of sample " problem and it is not easy to obtain robust facial landmarks. In [15], Ke Keung used the ISO-MAP for feature dimension reduction, and then SVM and Cascade Neutral Network were used to learn the classifier. Similar to [21][22], it could not handle the unseen data well yet. In [13], Potential Net Model was proposed to describe deformation of facial expression and tried to estimate the degrees of expression, but the experiment results were really limited. In [17], they combined facial feature point tracking, dense flow tracking, and high gradient component analysis together to do expression analysis. They claimed they could estimate expression estimation, but they did not show any experiments of intensity estimation.

Some researchers [18][14] tried to use the outputs of expression classifiers for intensity estimation. [18] used SVM for expression recognition, and the distance output of SVM, $\frac{w^T x + b}{\|w\|}$, is directly used to estimate the intensity of expression. This is based on the assumption that large distance to the hyperplane means the corresponding sample with high intensity. This is actually not guaranteed, for the hyperplane of SVM is formed by the support vectors, which have no constrains or logical relationships to the intensity.
Similar to [18], Koelstra [14] used the output of Gentleboost to predict the intensity of AUs, but he did not give quantitative analysis on the predicted results. The both methods took intensity estimation as a byproduct of the learned recognition classifier directly. However, as we mentioned in Section 1, intensity estimation is not a simple hard-decision problem, so conventional classification scheme does not work well at all. Figure 4 shows the distance output of SVM [18] on an image sequence of expression happiness from the Cohn-Kanade database. The sequence is from the onset to the apex. Its expression intensity should be monotonous increase, but the distance output of SVM does not guarantee intensity increasing monotonically. In this paper, we aim to handle expression recognition and intensity estimation together. Different from them, our framework is based on the ranking model, which focuses on the ordinal relationships between pair-wise data. The output of our ranking model can basically approximate the intensity variation, which is illustrated in the bottom row of Figure 4.

![Intensity vs. SVM Output](image)

**Figure 4.** The intensity changing on one sequence, the output of SVM, and the output of the proposed ranking model.

### 3. Our Work

Ranking is widely used in the fields of information retrieval [11] and econometric modeling [5]. For a ranking problem, it is assumed to have an outcome space \( \mathcal{Y} = \{r_1, ..., r_g\} \) with ordered ranks \( r_g \succ r_{g-1} \succ ... \succ r_1 \), where \( \succ \) denotes the order between different ranks. Generally, one latent continuous function \( U(x) \) can be learned to map one sample \( x \) into value \( r \) in the space \( \mathcal{Y} \). Ordinal regression is a classical ranking method. However, it is not easy to use ordinal regression to learn the map function \( U(x) \) for expression intensity estimation directly, because it is difficult to label each image with an absolute intensity value. We propose to use the RankBoost model for intensity estimation and recognition, which is based on the ordinal relationship between pairwise data, because we can easily organize the data in the ordinal pairwise format according to the temporal variation of expression.

In this section, we will first introduce how to organize the data, and then we will present the RankBoost and how to improve it with \( l_1 \) regularization. Finally we will discuss how to use the output ranking model for expression recognition.

#### 3.1. Data Organization

Given an expression sequence, although it is difficult to label the intensity on each instance, which could be single image or sub-sequence, we can definitely label the ordinal relationship between a pair of instances according to temporal order easily. In order to separate different expressions, we train the ranking model by the strategy of one vs. all, i.e., when we learn the rank model for expression \( E_i \), we define the rank order of any data within the \( E_i \) is higher than that of the data from the other expressions. Without loss of generality, we take the expression \( E_i \) as the interested expression for example to discuss how to organize the data as follows:

Taking account of a subject with the expression \( E_i \) and the other expression \( E_j \), we label the intensity decreases from the apex to the start state in \( E_i \), and then connected it the sequence from the other expression \( E_j \) as, \( R(I_{E_i,Apex}) \succ R(I_{E_i,start}) \succ R(I_{E_j,start}) \succ R(I_{E_j,Apex}) \), where \( R(I) \) is the ranking score of the instance \( I \). According to this rule, we obtain the reordered sequences set \( \{S_{E_i}\} \), and based on \( \{S_{E_i}\} \), we build pairwise instances \( \{(x_k, x_{k+1})\} \) for the ranking model learning to satisfy \( R(x_{k+1}) \succ R(x_k) \). We also define the ranking orders of \( R(I_{E_i,start}) \) of one subject are always higher than those of \( R(I_{E_j,start}) \) of any subjects to produce some pairwise samples between different subjects. Figure 5 shows an example of the intensity ranking on a happiness sequence.

#### 3.2. RankBoost

In this paper, we use the haar-like features to represent facial appearance, so we have thousands of haar-like features. It is untractable to use all the haar-like features to build the ranking model. Moreover, each expression is only dominated by parts of facial appearances. Thus, we adopt the RankBoost to build the rank model over the ordinal pairwise data for intensity estimation. Similar to the boosting learning, the RankBoost [7] aims to select a set of weak rankers to build a strong ranker. Given pairwise sample sets \( \{x_{i,0}, x_{i,1}\} \), the RankBoost tries to find a ranking function

\[
H(x) = \sum_{t} \alpha_t h_t(x), \text{ where } h_t(x) \text{ is a weak ranker, based on the following loss function:}
\]
Algorithm 1 RankBoost Learning procedure
1: Give example image pairs \((x_{i,0},x_{i,1}),...,(x_{n,0},x_{n,1})\).
2: Initialize weight \(D_t(i) = 1/N\).
3: for \(t = 0,...,T\) do
4: Train weak learner using distribution \(D_t(i) = 1/N\).
5: Get weak ranking \(h_t : h_t(x) \to R\), s.t equation 1.
6: Choose \(\alpha_t \in R\).
7: Update: \(D_{t+1}(x_{i,0},x_{i,1}) = \frac{D_t(x_{i,0},x_{i,1})\exp(\alpha_t(h_t(x_{i,0}) - h_t(x_{i,1}))}{Z_t}\) where \(Z_t\) is a normalization factor.
8: end for
9: Output the final ranking \(H(x) = \sum_t \alpha_t h_t(x)\).

3.3. RankBoost with \(l_1\) Regularization

In supervised learning settings with many input features, overfitting is usually a potential problem. It is well known that sample complexity grows linearly with the VC dimension when using unregularized discriminative models to fit the samples via training error minimization. Further, the VC dimension for most models grows about linearly in the number of parameters, which typically grows at least linearly in the number of input features. Thus, unless the training set size is large enough against the dimension of the input, some special mechanism such as regularization, which encourages the fitted parameters to be small is usually needed to prevent overfitting [19]. \(l_1\) regularization is a good choice due to its sparse character, which takes the sum of the squares of the parameters as the penalty term. Thus, in this paper, we adopt the \(l_1\) regularization to further improve the performance of the RankBoost. Inspired by [28], we rewrite the loss function of the RankBoost as,

\[
loss(H) = \min_{x_0,x_1} \sum_{x_0,x_1} \exp(H(x_0) - H(x_1))
\]

\[
= \min_{x_0,x_1} \exp(\sum_{t=1}^T \alpha_t (h_t(x_0) - h_t(x_1)))
\]

The Rankboost additively selects weak rankers by minimizing the exponential loss function, in which the greedy optimization strategy is used to solve it. The detailed algorithm is presented in Algorithm 1.

Algorithm 2 RegRankBoost Learning procedure
1: Give example image pairs \((x_{i,0},x_{i,1}),...,(x_{n,0},x_{n,1})\), set \(U_0 = \emptyset\), \(r_0 = 0\).
2: Initialize weight \(D_t(i) = 1/N\).
3: for \(t = 1,...,T\) do
4: Train weak learner using distribution \(D_t(i) = 1/N\).
5: Get weak ranking \(h_k : h_k(x) \to R\).
6: Choose \(\alpha_t \in R\), update \(U_t = U_{t-1} \cup h_k, r_t = r_{t-1} + v\alpha_t\).
7: Solve the convex minimization problem over \(\{\alpha_j\}_{j \in U_t}:\)
\[
\min_{\sum_{j \in U_t} \alpha_j h_j(x_{i,0} - x_{i,1})} \sum_{j \in U_t} \alpha_j \leq r_t, \alpha_j \geq 0, \forall j \in U_t
\]
8: Update the coefficients:
\[
a_t(j) = \begin{cases} 
\alpha_j & \text{if } j \in U_t; \\
0 & \text{otherwise.}
\end{cases}
\]
9: Update: \(D_{t+1}(x_{i,0},x_{i,1}) = \frac{D_t(x_{i,0},x_{i,1})\exp(\alpha_t(h_t(x_{i,0}) - h_t(x_{i,1})))}{Z_t}\) where \(Z_t\) is a normalization factor.
10: end for
11: Output the final ranking \(H(x) = \sum_t \alpha_t h_t(x)\).

3.4. Recognition by the Ranking Model

Besides the output ranking score can be directly used to measure the intensity, the ranking function can also be extended for expression recognition. The conventional expression recognition methods generally assume samples of
interested expression to be disjoint with samples of other expressions, denoted as two disjoint subsets $X_0$ and $X_1$, and then they aim to find one function $F$ to make $F(X_1) = +1$ and $F(X_0) = -1$. Therefore, they can actually be regarded as bipartite ranking methods [7]. The ranks of all the instances in $X_1$ are above the ranks of all the instances in $X_0$, and the one feedback function $\Phi$ is built to make $\Phi(X_0, X_1) = 1$, $\Phi(X_1, X_0) = -1$, and $\Phi(X, Y) = 0$ for all the other pairs. Under this strategy, the subtle ranks in the sets of $X_0$ and $X_1$ are ignored, and the ranking model is degraded to a classifier. Based on this interpretation, we extend our ranking model for expression recognition by the following formula:

$$F(x) = \arg \max_m \sum_i \alpha_{m,i}$$ (4)

where $\{\alpha_{m,i}\}$ are the coefficients of selected weak rankers in the corresponding ranking function $H_m(x)$, and $m$ is the class label.

4. Experiment

Our experiments are conducted on the Cohn-Kanade facial expression database [16], which are widely used to evaluate the facial expression recognition algorithms. This database consists of 100 students aged from 18 to 30 years old. Subjects were instructed to perform a series of 23 facial displays, six of which were prototypic emotions mentioned above. For our experiments, we selected image sequences from 96 subjects. The selection criterion was that a sequence could be labeled as one of the six basic emotions, so we use one against all strategy to build 6 classifiers and train the SVM based on the selected features, in which the linear kernel is adopted. For [14], we use the AdaBoost to replace the Gentleboost for convenience, because these two boosting methods have similar performances. For simplicity, we denote them as the AdaSVM and the Adaboost respectively. There are six basic expressions, so we use one against all strategy to build 6 classifiers or ranking models for each method. In [18] the output distance of the SVM is directly for intensity estimation, and the output of boost is used to predict intensity in [14]. Three kinds of criteria are used to evaluate the performances: Detection Rate (DR), Recognition Rate (RR) and Relevant Accuracy (RA)[2]. RA is used for evaluating the performance of intensity estimation, and its definition of RA is:

$$RA = \frac{\text{number of correctly ranked relevant pairs}}{\text{number of all relevant pairs}}$$ (5)

Given a sequences with $n$ frames whose intensity increases monotonically, we build $C_n^2$ relevant pairs $(x_0, x_1)$, where intensity of $x_1$ is higher than $x_0$. Because the variations of some consecutive images are too subtle to be distinguished, we use an interval with 3 frames to build the pairwise data along the rebuilt sequences.

4.1. Training Error Analysis

Before we report the results on the testing set, we would like to compare the performances of four methods on the training set, by analyzing DA and RA. For the SVM and the Adaboost, we calculate DA based on all the frames in the positive set, and make sure the false alarm is zero. For the RankBoost and the RegRankBoost, we use the pairwise instances in the training set to calculate DA. For RA, all the methods are based on the same pairwise instance in the interested expression. Table 1 reports the results of DAs and RAs.

From Table 1, we can see that even DAs of the AdaSVM and the AdaBoost are very high, their RAs are low. It means that the output distances of the SVM and the AdaBoost can not precisely describe the intensities. The RankBoost and the RegRankBoost obtain good DAs and RAs, because they are based on the ordinal pairwise data and their outputs are directly related to the intensities. In order to further investigate the performances of the RankBoost and the RegRankBoost, we illustrate their training errors in Figure 7, where the error rate is equal to $1 - RA$. We can see that the RegRankBoost slightly outperforms the RankBoost. For the RegRankBoost, almost 400 weak rankers are enough to build a final strong ranker, while the RankBoost needs at least 800 weak rankers to achieve similar performance to the RegRankBoost.

To evaluate the proposed method, we compare it to two related works: the SVM based method [18] and the boost based method [14]. For fair comparison, all the methods are based on the haar-like features. For the SVM based method, following by [18], we first use the Adaboost to select some discriminative features and train the SVM based on the selected features, in which the linear kernel is adopted. For [14], we use the AdaBoost to replace the Gentleboost for convenience, because these two boosting methods have similar performances. For simplicity, we denote them as the AdaSVM and the Adaboost respectively. There are six basic expressions, so we use one against all strategy to build 6 classifiers or ranking models for each method. In [18] the output distance of the SVM is directly for intensity estimation, and the output of boost is used to predict intensity in [14]. Three kinds of criteria are used to evaluate the performances: Detection Rate (DR), Recognition Rate (RR) and Relevant Accuracy (RA)[2]. RA is used for evaluating the performance of intensity estimation, and its definition of RA is:

$$RA = \frac{\text{number of correctly ranked relevant pairs}}{\text{number of all relevant pairs}}$$ (5)

Given a sequences with $n$ frames whose intensity increases monotonically, we build $C_n^2$ relevant pairs $(x_0, x_1)$, where intensity of $x_1$ is higher than $x_0$. Because the variations of some consecutive images are too subtle to be distinguished, we use an interval with 3 frames to build the pairwise data along the rebuilt sequences.

4.1. Training Error Analysis

Before we report the results on the testing set, we would like to compare the performances of four methods on the training set, by analyzing DA and RA. For the SVM and the Adaboost, we calculate DA based on all the frames in the positive set, and make sure the false alarm is zero. For the RankBoost and the RegRankBoost, we use the pairwise instances in the training set to calculate DA. For RA, all the methods are based on the same pairwise instance in the interested expression. Table 1 reports the results of DAs and RAs.

From Table 1, we can see that even DAs of the AdaSVM and the AdaBoost are very high, their RAs are low. It means that the output distances of the SVM and the AdaBoost can not precisely describe the intensities. The RankBoost and the RegRankBoost obtain good DAs and RAs, because they are based on the ordinal pairwise data and their outputs are directly related to the intensities. In order to further investigate the performances of the RankBoost and the RegRankBoost, we illustrate their training errors in Figure 7, where the error rate is equal to $1 - RA$. We can see that the RegRankBoost slightly outperforms the RankBoost. For the RegRankBoost, almost 400 weak rankers are enough to build a final strong ranker, while the RankBoost needs at least 800 weak rankers to achieve similar performance to the RegRankBoost.
Table 1. Performances on the training set

<table>
<thead>
<tr>
<th>Expression</th>
<th>AdaSVM(linear)</th>
<th>AdaBoost</th>
<th>RankBoost</th>
<th>RegRankBoost</th>
</tr>
</thead>
<tbody>
<tr>
<td>DR</td>
<td>RA</td>
<td>DR</td>
<td>RA</td>
<td>DR</td>
</tr>
<tr>
<td>Angry</td>
<td>1.00</td>
<td>0.56</td>
<td>0.98</td>
<td>0.61</td>
</tr>
<tr>
<td>Disgust</td>
<td>1.00</td>
<td>0.74</td>
<td>0.98</td>
<td>0.71</td>
</tr>
<tr>
<td>Fear</td>
<td>1.00</td>
<td>0.66</td>
<td>0.98</td>
<td>0.62</td>
</tr>
<tr>
<td>Happiness</td>
<td>1.00</td>
<td>0.69</td>
<td>0.98</td>
<td>0.64</td>
</tr>
<tr>
<td>Sadness</td>
<td>1.00</td>
<td>0.55</td>
<td>0.98</td>
<td>0.55</td>
</tr>
<tr>
<td>Surprise</td>
<td>1.00</td>
<td>0.73</td>
<td>0.98</td>
<td>0.67</td>
</tr>
<tr>
<td>mean</td>
<td>1.00</td>
<td>0.655</td>
<td>0.980</td>
<td>0.633</td>
</tr>
</tbody>
</table>

Figure 7. Error rates on the training data. (Anger, Disgust, Fear, Happiness, Sadness and Surprise)

4.2. Testing Error Analysis

On the testing data, we use $RR$ and $RA$ to evaluate the performances of four methods. Because there are six classifiers, we use the maximal normalized distance as the final recognition results for the AdaSVM and the AdaBoost. For the RankBoost and the RegRankBoost, the equation 4 is used to do recognition. The detailed results of $RR$s and $RA$s are shown in Table 2. The results on the testing set are similar to the results on the training set. The RankBoost and the RegRankBoost are better than the AdaSVM and the AdaBoost. Although four methods obtain high $DR$s on the training set, their $RR$s have some drops on the testing set, because there are no overlaps between the training set and the testing set. $RR$s in our experiments seem not as good as the results in [23], but our experiment setting is totally different from [23]. In [23], only the last tree peak frames are picked, while we use all the frames from the onset to the apex. Especially for the frames around the onset, i.e., the frames with low intensities, it is hard to identify their categorizations.

The RegRankBoost outperforms the RankBoost in both $RR$s and $RA$s. In order to further investigate their performances, we also evaluate their testing errors with the increase of the weak rankers shown in Figure 8, where the testing error is equal to $1 - RA$. We can see the RegRankBoost is a little better than the RankBoost. It demonstrates that the introduction of $l_1$ regularization can efficiently improve the performance of the RankBoost.

Comparing $RA$s in Table 2 and 1, we can see $RA$ drops a little for the RankBoost and the RegRankBoost. Besides non-overlaps exist between the training set and the testing set, some pairs of the data are really hard to evaluate their
Table 2. Performances on the testing set

<table>
<thead>
<tr>
<th>Expression</th>
<th>AdaSVM(linear)</th>
<th>AdaBoost</th>
<th>RankBoost</th>
<th>RegRankBoost</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RR</td>
<td>RA</td>
<td>RR</td>
<td>RA</td>
</tr>
<tr>
<td>Angry</td>
<td>0.68</td>
<td>0.64</td>
<td>0.87</td>
<td>0.68</td>
</tr>
<tr>
<td>Disgust</td>
<td>0.50</td>
<td>0.50</td>
<td>0.61</td>
<td>0.74</td>
</tr>
<tr>
<td>Fear</td>
<td>0.70</td>
<td>0.49</td>
<td>0.72</td>
<td>0.66</td>
</tr>
<tr>
<td>Happiness</td>
<td>0.88</td>
<td>0.62</td>
<td>0.94</td>
<td>0.70</td>
</tr>
<tr>
<td>Sadness</td>
<td>0.52</td>
<td>0.44</td>
<td>0.59</td>
<td>0.58</td>
</tr>
<tr>
<td>Surprise</td>
<td>0.86</td>
<td>0.61</td>
<td>0.90</td>
<td>0.74</td>
</tr>
<tr>
<td>mean</td>
<td>0.690</td>
<td>0.550</td>
<td>0.771</td>
<td>0.683</td>
</tr>
</tbody>
</table>

Figure 8. Error rates on the testing data (Anger, Disgust, Fear, Happiness, Sadness and Surprise)

Figure 9. Mis-classified examples.

ranks, because their intensity variations are too subtle to discriminate. Figure 9 shows some pairs which are wrongly ranked by the ranking model. We can see that they are very similar.

Additionally, the Adaboost gets a slightly higher RA in the testing set than in the training set as shown in Table 1 and Table 2. It shows that the Adaboost does not take account of intensity values during the training, so it can not work well on intensity estimation.

5. Conclusion

In this paper, we proposed to use the RankBoost for facial expression recognition and intensity estimation. Different from the previous work, we converted the intensity estimation problem to a ranking problem, and the intensity level is scored by the ranking function. Also we extended the ranking score for expression recognition. In order to further improve the performance of the RankBoost, the RegRankBoost is proposed, in which $l_1$ regularization is integrated into the RankBoost. Extensive experiments conducted on the Cohn-Kanade facial expression database demonstrated the power of the proposed method.
References


