RANKING WITH UNCERTAIN LABELS

Shuicheng Yan[1], *Huan Wang*[2], *Thomas S. Huang*[1], *Qiong Yang*[3], *and Xiaoou Tang*[2,3]

[1]ECE Department, University of Illinois at Urbana Champaign, USA
[2]Information Engineering Department, Chinese University of Hong Kong, Hong Kong
[3]Microsoft Research Asia, Beijing, China

ABSTRACT

Most techniques for image analysis consider the image labels fixed and without uncertainty. In this paper, we address the problem of ordinal/rank label prediction based on training samples with *uncertain labels*. First, the core ranking model is designed as the bilinear fusing of multiple candidate kernels. Then, the parameters for feature selection and kernel selection are learned by maximum a posteriori for given samples and uncertain labels. The convergency provable Expectation-Maximization (EM) method is used for inferring these parameters. The effectiveness of the proposed algorithm is finally validated by the extensive experiments on age ranking task. The FG-NET and Yamaha aging database are used for the experiments, and our algorithm significantly outperforms those state-of-the-art algorithms ever reported in literature.

1. INTRODUCTION

Many image analysis tasks, e.g., age ranking, require to predict ordinal/rank labels of the data. This kind of tasks can be naturally formulated as general regression problems and solved with popular algorithms such as Multilayer Perceptrons (MLPs) [10]. In this work, we are interested in the ranking problems with uncertain labels, that is, the ordinal labels of the training samples are not fixed, but with uncertainty and maybe characterized with intervals. In real applications, it is often the case that we cannot exactly obtain the ordinal labels. For example, for age ranking problem, the exact age information is often difficult to obtain, but it is relatively much easier to obtain intervals to characterize the possible age ranges. Several works [2][3][4][7] have been done to solve the age ranking problem, but none of them can solve the age ranking problem with uncertain labels.

The contributions of this paper are two-fold. On the one hand, we propose a flexible ranking model by integrating the kernel trick and bilinear regression strategy. The kernel trick brings the potential to characterize the nonlinear relationship between the low-level features and high-level ordinal labels; and the bilinear regression strategy automatically selects the features and the way for kernel combination, which makes the algorithm flexible and adaptive for specific task. On the other hand, a general solution based Expectation-Maximization (EM) approach is proposed to solve the ranking problem with uncertain labels. Both the ranking model in Section 2 and solution approach in Section 3 are general for most ranking tasks, and the effectiveness of the whole framework is validated with the extensive experiments on age ranking problem as described in Section 4.

2. BILINEAR RANKING MODEL

As mentioned above, many image analysis problems, e.g., age ranking, need predict the ordinal label information, and often there exists uncertainty within the labels for the training samples. Here, we assume that the features extracted from the training samples (images or videos) are denoted as $X = \{x_1, x_2, \ldots, x_N\}, x_i \in \mathbb{R}^m$, where N is the sample number and m is the feature dimension. The uncertain ordinal label for the sample x_i is denoted as $[l_i^{min}, l_i^{max}]$, and the whole label set is denoted as $L = \{[l_i^{min}, l_i^{max}], i = 1, 2, \cdots, N\}$.

In this work, we integrate the kernel trick [9] and the bilinear regression strategy to build the ranking model, which maps the low-level features to the high-level rank label. The model is motivated from the following observations: 1) the mapping from low-level features to high-level rank label is often nonlinear and the kernel trick offers the potential to uncover this kind of nonlinear relationship; and 2) how to select proper kernel functions is still an open problem, and the bilinear regression strategy provides a solution for automatic kernel selection along with the feature selection process.

Assume that we have a set of kernel mapping functions, denoted as $\{\phi^1(x), \phi^2(x), \cdots, \phi^n(x)\}$, where n is the number of kernels, and $\phi^o(x) : \mathbb{R}^m \rightarrow \mathfrak{F}^o, o = 1, 2, \cdots, n$, are the kernel mapping functions with $\mathfrak{F}^o$ as the higher or infinite dimensional Hilbert space and the corresponding kernel function $k^o(x, y) =< \phi^o(x), \phi^o(y) >$. Meanwhile, let the combined kernel mapping function be $\phi(x) = [\phi^1(x)^T, \phi^2(x)^T, \cdots, \phi^n(x)^T]^T$. It is obvious that the $k(x, y) =< \phi(x), \phi(y) >$ is also a kernel function.

The core idea of our proposed ranking model is to map $\phi(x)$ to the desired rank label. One one hand, we utilize a linear regression model to map $\phi(x)$ to a scale value which tar-

1-4244-1017-7/07/$25.00 ©2007 IEEE

geting at the desired rank label, and the projection direction is set as the linear combination of the mappings from all training samples, namely $[\phi(x_1), \phi(x_2), \cdots, \phi(x_N)]u = \phi(X)u$ where $u \in \mathbb{R}^N$. One the other hand, we assign different weights to different kernel mapping functions for the sake of automatic kernel selection. The final bilinear ranking model is formally defined as

$$l = (\phi(X)u)^T(\phi(x). \times v) \tag{1}$$

where $v \in \mathbb{R}^n$ is the weighting vector to give different weights for different kernels, and the operator $.\times$ is defined as

$$\phi(x). \times v = [\phi^1(x)^T v_1, \phi^2(x)^T v_2, \cdots, \phi^n(x)^T v_n]^T. \tag{2}$$

Here, for each sample x, we define a data-specific kernel matrix $K_x \in \mathbb{R}^{N\times n}$ as

$$K_x(j, o) = k^o(x_j, x), \tag{3}$$

and for the training sample x_i, this kernel matrix is written as K_i. Then, Eqn. (1) can be rewritten as

$$l = u^T K_x v. \tag{4}$$

An intuitive explanation of this ranking model is that the low-level feature vector x is mapped into a matrix, and then two projection vectors are learned to bilinearly transform this matrix into the desired high-level rank label.

3. EM-BASED PARAMETER LEARNING

In this section, we introduce our solution to learn the parameters u and v based on the training sample set X and the corresponding uncertain label set L.

3.1. Likelihood Model

Here, we assume that the samples and their corresponding labels are independently sampled, and the likelihood model of u and v for given training sample set X and label set L is defined as

$$p(X, L|u, v) = \prod_{i=1}^{N} \max_{l\in[l_i^{min},\, l_i^{max}]} p(x_i, l|u, v). \tag{5}$$

Here, we have

$$p(x_i, l|u, v) \propto \exp\{-\|l - u^T K_i v\|^2/\delta_1^2\}, \tag{6}$$

where δ_1 is a constant and will be combined with another parameter θ_2 into a single parameter λ which is set experientially as described later.

3.2. Prior Model

To alleviate the possibility to overfit to the training data, we present a prior model for the model parameters. Commonly, the norm of parameters is utilized for controlling overfitting. The larger is the norm, the more possible that the model will overfit to the training data. Therefore, a reasonable prior model for the model parameters is

$$p(u, v) \propto \exp\{-(\|u\|^2 + \|v\|^2)/\delta_2^2\}, \tag{7}$$

where the parameter δ_2 is a constant.

3.3. Maximum a Posteriori

Based on the above likelihood model and the prior model, the ranking model parameters are derived by maximum a posteriori as

$$\arg\max_{u,v}\{p(u, v|X, L) \propto p(u, v)p(X, L|u, v)\}. \tag{8}$$

There does not exist closed-form solution for maximum a posteriori due to the maximization operator in the likelihood model. In this work, we consider this optimization problem as an incomplete data problem. The missing data is the desired rank label, denoted as l_i for sample x_i, and the desired label set is $L^m = [l_1, l_2, \cdots, l_N]$. The complete-data model is

$$p(u, v|X, L, L^m) = p(u, v|X, L^m) \propto p(u, v)p(X, L^m|u, v), \tag{9}$$

where

$$p(X, L^m|u, v) \propto \prod_{i=1}^{N} \exp\{-\|l_i - u^T K_i v\|^2/\delta_1^2\}. \tag{10}$$

3.4. Parameter estimation with EM

Here, we use the Expectation Maximization (EM) algorithm [11] for parameter estimation with incomplete data. Denote the parameter estimate obtained at the n-th step by $\theta_n = \{u_n, v_n\}$. The Q-function [11] can then be obtained as

$$Q(\theta \mid \theta_n) \quad = E_{L^m}\{p(\theta|X, L, L^m) \mid X, L, \theta_n\} \tag{11}$$

where the expectation is taken w.r.t. $p(L^m|X, L, \theta_n)$.

3.4.1. E-Step

For given X, L and θ_n, the desired rank label for a certain data is fixed, namely with probability of 1 for a value while probability of 0 for all other values. More specifically speaking, the desired rank label of sample x_i has probability of 1 to be

$$\tilde{l}_i^m = \begin{cases} l_i^{min}, & \text{if } u_n^T K_i v_n < l_i^{min}, \\ l_i^{max}, & \text{if } u_n^T K_i v_n > l_i^{max}, \\ u_n^T K_i v_n, & \text{else.} \end{cases} \tag{12}$$

Denote $\tilde{L}^m = [\tilde{l}_1^m, \tilde{l}_2^m, \cdots, \tilde{l}_N^m]$, then we have

$$p(L^m | X, L, \theta_n) = \begin{cases} 1, \text{ if } L^m = \tilde{L}^m, \\ 0, \text{ else.} \end{cases} \quad (13)$$

Consequently, the Q-function can be simplified as

$$Q(\theta \mid \theta_n) = p(\theta | X, \tilde{L}^m) \quad (14)$$

3.4.2. M-Step

In the M-step, we maximize the Q-function

$$Q(\theta \mid \theta_n) \propto \exp\{-(\|u\|^2 + \|v\|^2)/\delta_2^2\} \times \prod_{i=1}^{N} \exp\{-\|\tilde{l}_i^m - u^T K_i v\|^2/\delta_1^2\}, \quad (15)$$

which is equivalent to minimize

$$F(u,v) = (\|u\|^2 + \|v\|^2)/\delta_2^2 + \sum_{i=1}^{N} \|\tilde{l}_i^m - u^T K_i v\|^2/\delta_1^2. \quad (16)$$

The objective function is quartic and commonly there does not exist closed-form solution. In this work, we present an iterative solution to search for the local optimum.

For given u, we set the derivative of $F(u,v)$ w.r.t v as zero to obtain the corresponding optimal value of v, namely

$$\frac{\partial F(u,v)}{\partial v} = \frac{2}{\delta_2^2} v - \sum_{i=1}^{N} \frac{2}{\delta_1^2} (u^T K_i)^T (\tilde{l}_i^m - u^T K_i v) = 0.$$

Then, we have

$$v = (\lambda I + \sum_{i=1}^{N} K_i^T u u^T K_i)^{-1} \sum_{i=1}^{N} K_i^T u \tilde{l}_i^m, \quad (17)$$

where $\lambda = \delta_1^2/\delta_2^2$. λ is the parameter to balance two terms on prior model and likelihood model, and in this paper, λ is experientially set as 0.0005 in all the experiments.

Similarly, for given v, we set the derivative of $F(u,v)$ w.r.t u as zero to obtain the corresponding optimal value of u, namely

$$\frac{\partial F(u,v)}{\partial u} = \frac{2}{\delta_2^2} u - \sum_{i=1}^{N} \frac{2}{\delta_1^2} (v^T K_i^T)^T (\tilde{l}_i^m - u^T K_i v) = 0.$$

Then, we have

$$u = (\lambda I + \sum_{i=1}^{N} K_i v v^T K_i^T)^{-1} \sum_{i=1}^{N} K_i v \tilde{l}_i^m, \quad (18)$$

We iteratively optimize u and v until the norms of differences between the solutions of two successive steps are both smaller than a manually set threshold (set as 10^{-4} in this work), or after a predefined number of loops (set as 20 in this work). Meanwhile, the whole algorithm iterates between the E-step and M-step until converged.

4. EXPERIMENTS

In this section, we take the age rank problem as an example to illustrate the effectiveness of our proposed algorithm for ranking with uncertain labels. The age label is not easy to obtain in real applications, and often there exists uncertainty within the label. Moreover, the age information is often present as an integer, but the age can actually be any real value. For example, when we say a person is of age n, its real age can be any value within $[n, n{+}1)$. In this paper, we set the age label as $[n, n{+}1{-}\epsilon]$ for our algorithm, where ϵ is the smallest positive value that a computer can encode.

The following two algorithms are systematically compared with our algorithm. In [4], the relationship between the age labels and the feature vectors is modeled as a quadratic regression model (QM), namely,

$$a = c + w_1^T x + w_2^T (x.^2), \quad (19)$$

where x and $(x.^2)$ are the vectors containing the features and the squares of the features respectively, c, w_1, and w_2 are parameters. Another popular regression algorithm is Multilayer Perceptrons (MLPs) with the back propagation learning [10].

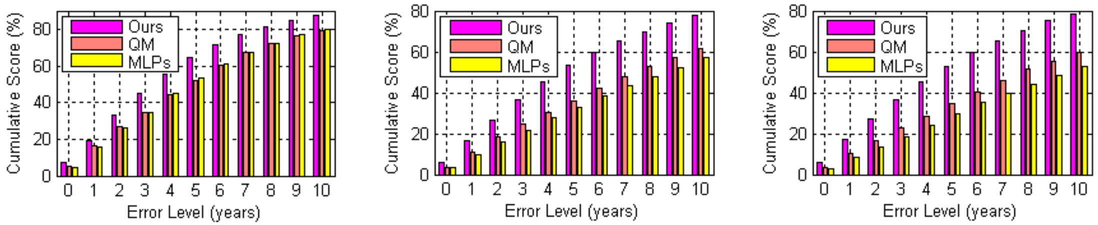
4.1. Experiment Configurations

Two aging face databases are used in our experiments. One is the FG-NET aging database [1], which contains 1002 face images of 82 persons with ages ranging from 0 to 69. The evaluation framework for the FG-NET database is the Leave-One-Person-Out (LOPO). The other data set Yamaha face database[1] contains 800 males and 800 females, and 8000 images with ages ranging from 0 to 93. The experiments are carried out separately on female and male subsets respectively. For each subset, the images are randomly divided into 4 folds, and 4-fold cross-validation is performed for the evaluation of different algorithms.

In the FG-NET database, each person has multiple images of different ages, and hence it is relatively easy to extract features characterizing aging process; while for the Yamaha database, the images of the same person are of the same age, and hence many algorithms such as *aging patterns subspace* (AGES) [7] and *Weighted Appearance Specific* (WAS) [8] are inapplicable due to their requirement of multiple images of different ages for each person.

For comparison, we use the same feature set as in [7] for the FG-NET database, and the first 200 appearance parameters [6] based on the 68 key facial points are used as input for age ranking. For detailed information on shape, texture and appearance parameters, please refer to [5]. For the Yamaha database, as we do not have the positions of the key facial points as in the FG-NET database, the face images are cropped and normalized to size of 64-by-64 pixels by fixing

[1] To protect the portrait rights of the participants, sample images of the Yamaha face database are not shown here.



(a) FG-NET. (b) Female@YAMAHA. (c) Male@YAMAHA.

Fig. 1. Cumulative scores for Quadratic Model, MLPs and our algorithm at absolute error levels from 0 to 10 years.

Table 1. MAEs of different algorithms on FG-NET and Yamaha aging databases. Note that AGES was the best algorithm ever reported on the FG-NET database.

Algorithm	FG-NET	Female@YA	Male@YA
WAS [7]	8.06	N/A	N/A
AGES [7]	6.77	N/A	N/A
QM	6.55	9.96	10.51
MLPs	6.98	10.99	12.0
Ours	**5.33**	**6.95**	**6.95**

the locations of the two eyes. The original gray level values are used as features for age ranking. In both databases, the gaussian kernels $k_o(x, y) = \exp\{-\|x - y\|^2/\delta_o^2\}$ are applied as the kernel candidates and we use 4 kernels with parameters $\delta_o = 2^{(o-10)/2.5}\delta, o = 0, 1, 2, 3$ in all the experiments, where δ is the standard deviation of the sample data.

4.2. Age Ranking Results

Two measures are used to evaluate algorithmic performance. The first one is the Mean Absolute Error (MAE) criterion used in [4][8][7]. MAE is defined as an average of the absolute errors between the estimated ages and the ground truth ages, *i.e.*, MAE = $\sum_{i=1}^{N_t} |\hat{l}_i - l_i|/N_t$, where $\hat{l}_i$ is the estimated age for the i-th sample, l_i is the ground truth age for the testing images and N_t is the number of testing images. Another popular measure is the cumulative score [7]: $CS(\theta) = N_{e\leq\theta}/N_t \times 100\%$, where $N_{e\leq\theta}$ is the number of samples on which the estimator makes an absolute error no higher than θ.

A detailed comparison of the age estimation accuracy is displayed in Figure 1 and listed in Table 1. Table 1 lists the MAEs of different algorithms on both FG-NET and Yamaha databases. From these results, we can have the observations: 1) our algorithm significantly outperforms the state-of-the-art age ranking algorithms which consider the age labels fixed; and 2) AGES is the best algorithm ever reported for age ranking, and it performs better than MLPs and WAS, yet worse than QM in our experiment. A possible explanation is that many aging patters are incomplete in FG-NET database.

5. ACKNOWLEDGMENT

This research was funded in part by the U.S. Government VACE program.

6. REFERENCES

[1] The FG-NET aging database: http://sting.cycollege.ac.cy/∼alanitis/fgnetaging/index.htm.

[2] J. Hayashi, M. Yasumoto, H. Ito and H. Koshimizu, “A method for estimating and modeling age and gender using facial image processing,” *Seventh International Conference on Virtual Systems and Multimedia*, pp. 439–448, 2001.

[3] Y. Kwon and N. Lobo, “Age classification from facial images,” *Computer Vision and Image Understanding*, vol. 74, no. 1, pp. 1–21, 1999.

[4] A. Lanitis, C. Draganova and C. Christodoulou, “Comparing different classifiers for automatic age estimation,” vol. 34, no. 1, pp. 621–628, February 2004.

[5] T. Cootes, G. Edwards and C. Taylor, “Active appearance models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 681–685, 2001.

[6] G. Edwards and A. Lanitis. Statistical face models: Improving specificity. *Image and Vision Computing*, vol. 16, no. 3, pp. 203–211, 1998.

[7] X. Geng, Z. Zhou, Y. Zhng, G. Li, and H. Dai, “Learning from facial aging patterns for automatic age estimation,” *Proceedings of ACM Multimedia’06*, 2006.

[8] A. Lanitis, C. Taylor and T. Cootes, “Toward automatic simulation of aging effects on face images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 442–455, 2002.

[9] K. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, “An introduction to kernel-based learning algorithms,” *IEEE Transactions on Neural Networks*, vol. 12, no. 2, pp. 181–202, 2001.

[10] G. Hinton, D. Rumelhart and R. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, pp. 533–536, 1986.

[11] J. Bilmes. A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Technical report, International Computer Science Institute, Berkeley, 1998.