Learning Auto-Structured Regressor from Uncertain Nonnegative Labels

Shuicheng Yan[1], Huan Wang[2], Xiaoou Tang[2,3], and Thomas S. Huang[1]
[1] Beckman Institute, University of Illinois at Urbana-Champaign, USA
[2] Department of Information Engineering, the Chinese University of Hong Kong, HK
[3] Microsoft Research Asia, Beijing, China
Contact:scyan@ifp.uiuc.edu

Abstract

In this paper, we take the human age and pose estimation problems as examples to study automatic designing regressor from training samples with uncertain nonnegative labels. First, the nonnegative label is predicted as the *square norm* of a matrix, which is bilinearly transformed from the nonlinear mappings of the candidate kernels. Two transformation matrices are then learned for deriving such a matrix by solving a semidefinite programming (SDP) problem, in which the uncertain label of each sample is expressed as two inequality constraints. The objective function of SDP controls the ranks of these two matrices, and consequently automatically determines the structure of the regressor. The whole framework for automatic designing regressor from samples with uncertain nonnegative labels has the following characteristics: 1) SDP formulation makes full use of the uncertain labels, instead of using conventional fixed labels; 2) regression with matrix norm naturally guarantees the nonnegativity of the labels, and greater prediction capability is achieved by integrating the squares of the matrix elements, which act as weak regressors; and 3) the regressor structure is automatically determined by the pursuit of simplicity, which potentially promotes the algorithmic generalization capability. Extensive experiments on two human age databases, FG-NET and Yamaha, as well as the Pointing'04 pose database, demonstrate encouraging estimation accuracy improvements over conventional regression algorithms.

1. Introduction

Many computer vision tasks need to design regressors to estimate certain quantities. In this work, we take as examples the age and pose estimation problems to address the regression problem in the scenario where the label of each sample is provided as a nonnegative interval, instead of a conventional fixed value. A natural question is whether we can gain more from the enriched information by designing a new learning framework. Our answer to this question is positive.

A face image encodes different types of useful information, such as identity, expression, gender, age, and pose. It is commonly believed that human can provide satisfying and consistent ground truths about the identity, expression, and gender, by clues from the speech, hair style, and costume. But for age and pose, the ground truths labeled by variant individuals are often far from consistence due to the complex effects of living conditions, cosmetics, personal specialties, gender differences, facial geometry, and so on. Three examples of pose labeling from 13 different participants are shown in Figure 1, and large variations are observed among the labels from different participants [18].

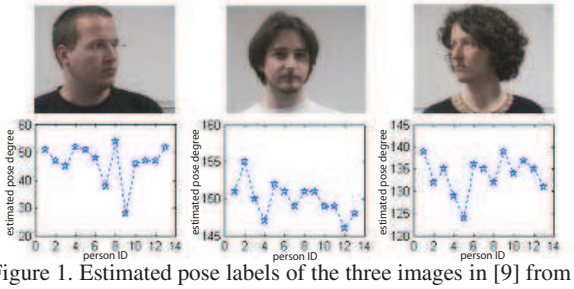


Figure 1. Estimated pose labels of the three images in [9] from 13 different observers by rotating a 3D head model. We can see that large standard deviations exist for these labeled ground truths [18].

Despite that age is an important characteristic of human, only a few works have been dedicated to the problem of age estimation [10][11]. The latest work is from Geng et al. [8], based on the statistical modeling of aging patterns. All these algorithms require the age label to be a fixed value. However, it is often difficult to acquire the accurate ages in real applications, instead, only rough age ranges are often obtained. Moreover, even the age of a certain person is labeled as a fixed value like 30, the actual age can be any real value within the interval $[30\ \ 31)$. Hence it makes more sense for the age to be expressed as an interval instead of a fixed value.

Head pose estimation has many useful applications such as gaze detection, safe driving, and auto-mouse in large screen. Much research [7][15] has been dedicated to this specific problem, and the latest work was proposed by Nallure et al [14], based on biased manifold learning. All these algorithms cannot directly handle the scenario with uncertain labels. Although the pose labels can be obtained from

978-1-4244-1631-8/07/$25.00 ©2007 IEEE

hardware device in certain experimental scenarios, it is still desirable and of great application importance to propose a new formulation for automatic designing regressor based on training samples with uncertain labels.

In this work, the estimation of age or pose information is considered as a nonlinear regression problem based on training samples with uncertain nonnegative labels. The nonlinearity from the input image features to the label is embodied with the kernel trick [13], and a set of kernels are used for designing a regressor. The age and pose values can be considered nonnegative and with lower-bound of zero, and we compute the age as the square norm of a dimension-flexible matrix. Its advantage over direct linear combination of features or kernels for regression is that it provides a flexible way to integrate a set of weak regressors, namely the square outputs of the matrix elements, for a better approximation. This flexible matrix is bilinearly transformed from the mappings of the candidate kernels. The learning of these two transformation matrices is formulated as a semidefinite programming (SDP) problem [4]. More specifically speaking, the uncertain label, namely the age or pose interval, of each sample is expressed as two inequality constraints in the SDP formulation, and its objectives function is used to pursue regressor with simple structure, which potentially promotes the algorithmic generalization capability.

The rest of this paper is organized as follows. Section 2 introduces the details of the SDP formulation for automatic designing regressor based on training samples with uncertain nonnegative labels. Its relationship to other SDP problems as well as traditional regression algorithms is discussed in Section 3. Section 4 provides comparison experiments on two human age databases and one pose database, and the concluding remarks are given in Section 5.

2. Nonlinear Regression with Uncertain Nonnegative Labels

For the human age or pose estimation problem, the image set for model training is denoted here as a matrix $X = [x_1, x_2, \ldots, x_N], x_i \in \mathbb{R}^m$, where N is the image number and m is the feature dimension. The uncertain nonnegative label for the image x_i is denoted as $[l_i, L_i]$ where $l_i \geq 0$. The task is to predict the nonnegative label of the new image x, and our solution to this general problem is as follows.

2.1. Problem Formulation

Nonlinear regression from image features to nonnegative label. Generally, rough age or pose estimation can be conducted by separating all possible ages or poses into several groups, and then formulated as a general multi-class classification problem. In this way, for a more precise estimate, a greater number of classes will be used, and consequently more samples will be required for learning a reliable estimator. In this work, we take the age/pose estimation problem as a nonlinear regression problem, and the nonlinear formulation is based on the kernel trick [13].

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 $\phi^o(x) : \mathbb{R}^m \rightarrow \mathfrak{F}^o, o = 1, 2, \cdots, n$, is the kernel mapping function 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 $\phi(x) = [\phi^1(x)^T, \phi^2(x)^T, \cdots, \phi^n(x)^T]^T$.

In this work, we model the nonnegative label as the square norm of a matrix, and specifically we have

$$a = \| W^T(\phi(x) \odot V) \|^2, \tag{1}$$

where $V \in \mathbb{R}^{n \times n'}$ is the matrix to give different weights for different kernels, and the operator $\odot$ is defined as

$$\phi(x) \odot V = \begin{bmatrix} \phi^1(x)V_{11} & \phi^1(x)V_{12} & \cdots & \phi^1(x)V_{1n'} \\ \phi^2(x)V_{21} & \phi^2(x)V_{22} & \cdots & \phi^2(x)V_{2n'} \\ \vdots & \vdots & \ddots & \vdots \\ \phi^n(x)V_{n1} & \phi^n(x)V_{n2} & \cdots & \phi^n(x)V_{nn'} \end{bmatrix}.$$

The symbol W is the transformation matrix that transforms the feature from the higher dimensional feature space into a lower dimensional one, and it is represented as the linear combination of $\{\phi(x_i), i = 1, 2, \cdots, N\}$, that is, $W = [\phi(x_1), \phi(x_2), \cdots, \phi(x_N)]\ U = \phi(X)\ U$ where $U \in \mathbb{R}^{N \times N'}$. The number n' and N' are the expected column numbers of the matrix V and W, and automatically determined as described afterward.

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

$$K_i(j, o) = k^o(x_i, x_j), \tag{2}$$

and for an image x, this kernel matrix is written as K_x. Then, the function in Eqn. (1) can be rewritten as

$$\|U^T\phi(X)^T \begin{bmatrix} \phi^1(x)V_{11} & \phi^1(x)V_{12} & \cdots & \phi^1(x)V_{1n'} \\ \phi^2(x)V_{21} & \phi^2(x)V_{22} & \cdots & \phi^2(x)V_{2n'} \\ \vdots & \vdots & \ddots & \vdots \\ \phi^n(x)V_{n1} & \phi^n(x)V_{n2} & \cdots & \phi^n(x)V_{nn'} \end{bmatrix} \|^2$$

$$= \|U^T K_x V\|^2, \tag{3}$$

that is,

$$a = \|U^T K_x V\|^2. \tag{4}$$

This formulation of the prediction function is much simpler than that in Eqn. (1).

Discussion: why do we use square norm of a flexible matrix for approximating label? The reasons are two-folds.

On the one hand, commonly the labels have lower-bound and can be assumed to be nonnegative after a translation, say $[0,\ +\infty)$ for age and $[0,\ 180]$ for pose in this work, and the norm operator can naturally guarantee the nonnegative property. On the other hand, Eqn. (4) equals to $\sum_{i=1}^{N'}\sum_{i=1}^{n'}\|U_i^T K_x V_j\|^2$, where $\|U_i^T K_x V_j\|^2$ can be considered as weak regressors, and hence the sum of these weak regressors may bring greater approximation capability than single weak regressor.

Uncertain labels to inequality constraints. According to Eqn. (4), the uncertain nonnegative label of the sample x_i, *i.e.*, $[l_i\ \ L_i]$, can be expressed as the two inequalities:

$$\|U^T K_i V\|^2 \le L_i, \tag{5}$$
$$-\|U^T K_i V\|^2 \le -l_i. \tag{6}$$

Objective function: avoiding overfitting and pursuing simplicity. As described above, $\|U_i^T K_x V_j\|^2$ can be considered as a weaker regressor. A reasonable way to reduce the possibility of overfitting and promote the algorithmic generalization capability is to control the rank of the transformation matrices U and V and consequently reduce the number of weak regressors. Moreover, there may be infinite solutions that satisfy all the constraints in Eqn. (5-6). It is desirable to provide a criterion for guiding the selection of the optimal solution, and the pursuing of lower ranks of the parameter matrices is a feasible strategy.

In the following, we present a method for controlling the ranks of the transformation matrices based on sparsity property. Similar to the Sparse Support Vector Machine [3] method, our algorithm employs the L_1 norm to control the sparsity of the parameters.

As we expect to control the ranks of the transformation matrices, we do not compute the matrix U and V directly; instead, we compute the matrices

$$S^u = UU^T,\ S^v = VV^T. \tag{7}$$

Here, we take S^u as an example to demonstrate how to control the matrix rank. Take the singular value decomposition of S^u as

$$S^u = U_r \Lambda^u {U_r}^T, \tag{8}$$

where $U_r \in \mathbb{R}^{N\times N}$ is a square orthogonal matrix and $\Lambda^u = diag\{\lambda_1^u, \lambda_2^u, \cdots, \lambda_N^u\}$. Then, controlling the rank of the matrix U is equivalent to controlling the sparsity of the diagonal elements of matrix Λ^u. Enlightened by the Sparse SVM, we simply minimize the L_1 norm of the diagonal elements of Λ^u, that is,

$$\min_{S^u} \sum_{i=1}^{N} |\lambda_i^u| = \min_{S^u} Tr(S^u), \tag{9}$$

where $Tr(\cdot)$ is the trace of a square matrix. The equality is satisfied owing to the fact that all λ_i^u's from positive semidefinite matrix S^u are nonnegative.

Remark. $Tr(S^u)$ can also be considered as the L_2 norm of $(\tau_i^u, \tau_2^u, \cdots, \tau_N^u)$, which are the singular values of U with $\lambda_i^u = {\tau_i^u}^2$. But as discussed later, the optimization problem takes S^u and S^v as variables directly and the constraints are all linear inequalities. Hence it is reasonable to control the sparsity of $(\lambda_1^u, \lambda_2^u, \cdots, \lambda_N^u)$ directly. Similarly, we can control the rank of the matrix V by minimizing $Tr(S^v)$.

According to the definition in Eqn. (7), the left-hand items in the constraints (5-6) can be expressed by S^u and S^v as

$$\begin{aligned}\|U^T K_i V\|^2 &= Tr(U^T K_i V V^T K_i^T U)\\ &= Tr(K_i V V^T K_i^T U U^T)\\ &= Tr(K_i S^v K_i^T S^u),\end{aligned} \tag{10}$$

where the first equality is obtained from the fact that $\|A\|^2 = Tr(AA^T)$ for any matrix A, and the second equality is obtained from the fact that $Tr(AB) = Tr(BA)$ for any matrices A and B with proper dimensions.

Based on the above constraints and the objective function, the regression problem with uncertain nonnegative labels can be formally defined as

$$\begin{aligned}&(S^u, S^v)^* = \arg\min_{S^u, S^v} Tr(S^u) + Tr(S^v),\ s.t.\\ &1: S^u \succeq 0, S^v \succeq 0;\\ &2:\ \ Tr(K_i S^v K_i^T S^u) \le L_i,\ i = 1, 2, \cdots, N;\\ &3: Tr(-K_i S^v K_i^T S^u) \le -l_i, i = 1, 2, \cdots, N;\end{aligned} \tag{11}$$

where $S^u \succeq 0$ and $S^v \succeq 0$ mean that S^u and S^v are positive semidefinite.

In this problem, the objective function is convex, yet the feasible solution set is possibly nonconvex; hence it is essentially a non-convex optimization problem and consequently no closed-form solution exists. Naturally, we present a procedure to optimize S^u and S^v iteratively, and in each step, the problem is then converted into a convex optimization problem. The semidefinite programming toolbox can be applied for the step-wise optimization.

2.2. Iterative Parameter Optimization

Iterative optimization along different axes is very common in the non-convex optimization literature. Here, we solve the optimization problem with respect to (S^u, S^v) by iteratively optimizing one parameter matrix while fixing the other one.

For the given S^v, the constraints 2-3 in the optimization problem (11) are changed to

$$Tr(C_i^u S^u) \le\ \ L_i, i = 1, 2, \cdots, N, \tag{12}$$
$$Tr(-C_i^u S^u) \le -l_i, i = 1, 2, \cdots, N, \tag{13}$$

where $C_i^u = K_i S^v K_i^T$.

With the above constraints and the objective function in Eqn. (11), the matrix S^u can be obtained by optimizing

Algorithm 1 Procedure to learn matrix S^u

Minimize $Tr(S^u)$

1: $S^u \succeq 0$;
2: $Tr(C_i^u S^u) \leq L_i,\ i = 1, 2, \cdots, N$;
3: $Tr(-C_i^u S^u) \leq -l_i, i = 1, 2, \cdots, N$.

Algorithm 2 Procedure to learn matrix S^v

Minimize $Tr(S^v)$

1: $S^v \succeq 0$;
2: $Tr(C_i^v S^v) \leq L_i,\ i = 1, 2, \cdots, N$;
3: $Tr(-C_i^v S^v) \leq -l_i, i = 1, 2, \cdots, N$.

a Semidefinite Programming problem as shown in Algorithm 1. The objective function in Algorithm 1 is convex, and the optimization does not suffer from the local optimum issue [17]. There are several general-purpose toolboxes and polynomial-time solvers available for solving the semidefinite programming problem. In this work, we utilize the solver SeDuMi and the CSDP 4.9 toolbox in MATLAB [4].

Similarly, for the given S^u, the constraints 2-3 in the optimization problem (11) are changed to

$$Tr(C_i^v S^v) \leq \ L_i, i = 1, 2, \cdots, N, \tag{14}$$

$$Tr(-C_i^v S^v) \leq -l_i, i = 1, 2, \cdots, N, \tag{15}$$

where $C_i^v = K_i^T S^u K_i$.

Then, the optimization problem in Eqn. (11) is converted into a standard semidefinite programming problem as listed in Algorithm 2. Similarly, it can be solved with the general-purpose toolbox for the SDP problem.

The Algorithm 1 and 2 are iteratively conducted to obtain the stepwise result (S_t^u, S_t^v) until satisfying the following stop criteria:

$$\begin{cases} \|S_t^u - S_{t+1}^u\| < \varepsilon N^2 \\ \|S_t^v - S_{t+1}^v\| < \varepsilon n^2 \end{cases}, \tag{16}$$

where ε is a manually defined threshold and is empirically set to be 10^{-6} in this work.

Convergency Discussion: The optimization problem in Eqn. (11) is non-convex due to the non-convexity of the feasible solution set, and hence we cannot guarantee that the solution will be globally optimal. Here, instead we prove that the iterative algorithm will converge to a local optimum. Denote the objective function as $F(S^u, S^v) = Tr(S^u) + Tr(S^v)$, then we have

$$F(S_t^u, S_t^v) \geq F(S_{t+1}^u, S_t^v) \geq F(S_{t+1}^u, S_{t+1}^v). \tag{17}$$

Therefore, the objective function is non-increasing, and we have $f(S^u, S^v) \geq 0$, which means that the objective function has a lower-bound. Then we can conclude that the objective function will converge to a local optimum.

After the convergence of the iterative procedure, the transformation matrices U and V can be obtained from the singular value decomposition of the obtained matrices S^u and S^v,

$$S^u = U_r diag\{\lambda_1^u, \lambda_2^u, \cdots, \lambda_{N'}^u, 0, \cdots, 0\} {U_r}^T, \tag{18}$$

$$S^v = V_r diag\{\lambda_1^v, \lambda_2^v, \cdots, \lambda_{n'}^v, 0, \cdots, 0\} {V_r}^T, \tag{19}$$

where $\lambda_i^u > 0$ and $\lambda_i^v > 0$. We then have

$$U = U_r(:, 1 : N') diag\{\lambda_1^u, \lambda_2^u, \cdots, \lambda_{N'}^u\}^{1/2}, \tag{20}$$

$$V = V_r(:, 1 : n')\, diag\{\lambda_1^v, \lambda_2^v, \cdots, \lambda_{n'}^v\}^{1/2}, \tag{21}$$

where $U_r(:, 1 : N')$ means the submatrix consisting of the left N' column vectors of the matrix U_r and similarly $V_r(:, 1 : n')$ is the submatrix consisting of the left n' column vectors of the matrix V_r.

Remark: The column numbers of the matrix U and V are automatically determined, and consequently the number of weak regressors as well as the structure of the regressor are determined in an automatic manner.

3. Algorithmic Analysis

In this section, we analyze the relationship between the traditional regression formulation and that defined by SDP in this paper. We then compare our algorithm with other SDP based algorithms.

3.1. Traditional Regressor vs. SDP Formulation

A direct approach for age or pose estimation is to design proper regressor. In [11], the relationship between the label and the features is modeled with quadratic regression, namely,

$$a = c + w_1^T x + w_2^T (x. * x), \tag{22}$$

where x and $(x. * x)$ are the vectors containing the image features and the squares of the features respectively; c, w_1, and w_2 are parameters to learn. This simple regression method works well for simple quadratic fitting problems, but for a complex nonlinear regression problems like age and pose estimation, it cannot produce a satisfactory fit.

Another popular regression algorithm is Multilayer Perceptrons (MLP) with back propagation learning [16]. It has been widely applied in various applications, such as face detection and recognition. Yet, MLP has several parameters that need to be set beforehand, such as the number of layers and the node number for each layer; the tuning of such parameters is often time consuming and needs extra validation data set.

Although simple quadratic regression and MLP can be used for age or pose estimation, they cannot directly handle regression problems with uncertain labels like what our proposed SDP based formulation does. Another disadvantage of quadratic regression and MLP is that the output from

Algorithm 3 Procedure to learn matrix S^u with Relaxation

Minimize $Tr(S^u) + \gamma \sum_{i=1}^{N} (\epsilon_i^1 + \epsilon_i^2)$

1: $S^u \succeq 0, \epsilon_i^1 \geq 0, \epsilon_i^2 \geq 0, i = 1, 2, \cdots, N;$
2: $Tr(C_i^u S^u) - \epsilon_i^1 \leq L_i^2,\ i = 1, 2, \cdots, N;$
3: $Tr(-C_i^u S^u) - \epsilon_i^2 \leq -l_i^2, i = 1, 2, \cdots, N.$

them can be negative, which is inconsistent with the nonnegative assumption in this work.

3.2. Other Problems Formulated with SDP

In the past few years, the optimization tool SDP has been used for problem formulation in several works. Weinberger et al. [17] formulated the manifold embedding task as a semidefinite programming problem and provided a new perspective beyond spectral analysis for manifold learning. The sparse Principal Component Analysis problem was also formulated with SDP as demonstrated in [2]. Though our proposed algorithm is also formulated as a SDP problem, the objective function and the constraints in our algorithm are unique. Our proposed algorithm utilizes SDP as a tool for formulating the nonlinear regression problem with uncertain nonnegative labels.

4. Experiments

In this section, we first introduce the implementation details of our algorithm for Nonlinear **R**egression with **U**ncertain **N**onnegative Labels, referred to as RUN hereafter. Then the superiority of uncertain labels over fixed labels is justified with a toy problem; and the algorithmic convergency property is verified by the experiments on the FG-NET [1] database. Finally, the human age databases, FG-NET and Yamaha databases, and Pointing'04 pose database, are used to systematically evaluate the effectiveness of the RUN algorithm in estimation accuracy by comparisons with the state-of-the-art algorithms [8].

4.1. Implementation Details

In our implementation, two strategies are applied to facilitate the RUN algorithm. It is possible that not all the constraints in Algorithm 1 and 2 can be satisfied in real applications, and hence we add relaxation parameters to ensure that the feasible solution set is not empty. Following [4], we take Algorithm 1 as an example to introduce how to add relaxation parameters, and the details are listed in Algorithm 3.

As described in Section 2.1, the column vectors of the matrix W lie within the space spanned by the kernel mappings of all training samples. Thus the size of U increases along with the growth of the training set, and consequently the size of S^u will be very large. To improve the scaling capability, we constrain the column vectors of W to be the combination of certain prototypes from the training set, namely $W = [\phi(x_1^p), \phi(x_2^p), ..., \phi(x_M^p)]U$ where x_i^p is the selected prototype, $U \in \mathbb{R}^{M \times N'}$, and M is the prototype number. We conduct the K-means algorithm for clustering the training samples, and then the samples near the cluster centers are selected as prototypes. For all the age estimation experiments, the prototype number is set as 400, and the parameter γ in Algorithm 3 is set to 1. 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. For the pose estimation experiments, the prototype number is set as 40, and the other parameters are the same as those for age estimation.

In this work, RUN is compared with the traditional regression algorithms Quadratic Models (QM) and supervised Neural Networks [11]. According to [8], the Least Squares Fit (LSF) optimization algorithm commonly gives better performance than the genetic algorithm, thus the Least Square Fit optimization is utilized for the Quadratic Models. And for the Neural Networks, we adopt the same configuration as in [12], *i.e.*, MLP with the back propagation learning algorithm, and the network architecture and parameters are also set the same as in [12].

Estimation output. For an image x, its estimated age is output as $\hat{a} = \|U^T K_x V\|$.

Accuracy Measurements. Two measures are used to evaluate the algorithmic performance. The first one is the Mean Absolute Error (MAE) criterion used in [11][12] and [8]. MAE is defined as the average of the absolute errors between the estimated labels and the ground truth labels, *i.e.*,

$$MAE = \sum_{i=1}^{N_t} |\hat{a}_i - a_i|/N_t,$$

where $\hat{a}_i$ is the estimated age for the i-th testing sample, a_i is the ground truth age and N_t is the number of testing images. Another popular measure is the cumulative score [8]:

$$CumScore(\theta) = N_{e \leq \theta}/N_t \times 100\%,$$

where $N_{e \leq \theta}$ is the number of samples on which the estimator makes an absolute error not higher than θ.

4.2. Toy Data: Effectiveness of Uncertain Labels

To examine the effectiveness of uncertain labels, we introduce a toy example where we know the exact labels. In this toy problem, the feature dimension is 2, and the underlying relationship between the data and the label is

$$l(x) = l(x_1, x_2) = 100 \times cos(\|x\|) + 100 \times exp(-\|x\|^2),$$

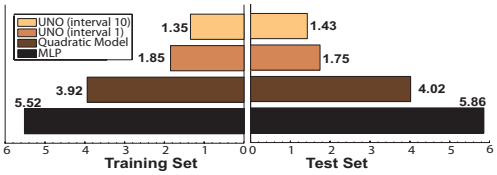


Figure 2. The MAEs of the RUN algorithms with uncertain label interval of width 10 and width 1, QM, and MLPs on the toy data.

Figure 3. Sample aging images from one person in FG-NET Aging Database.

where x_1 and x_2 independently follow the uniform distribution on the unit interval $[0,\ 1]$, and $l(x)$ is the exact label from data x with $l(x) > 0$.

To obtain the uncertain labels, the observed label is assumed to be $l^n(x) = l(x) + \varepsilon$, where ε is random noise evenly distributed within the interval $[-5, 5]$ such that $l^n(x) > 0$. 200 data samples are randomly sampled as the training set and the labels with noise are used for model training; also another 200 data are sampled for testing, and the exact labels are used for computing both training and testing accuracy. For comparison, the results from QM and MLP are also reported.

For our proposed RUN algorithm, we compare two types of uncertainty scales: one is the uncertain labels with an interval of width 10, namely $[l^n(x)\text{-}5, l^n(x)\text{+}5]$, and the other is the uncertain labels with a smaller interval of width 1, namely $[l^n(x)\text{-}0.5, l^n(x)\text{+}0.5]$ which is very close to the fixed labels. In these experiments, 100 prototypes are used, the kernel number and γ are set as for age estimation experiments. Typically it takes about 70 seconds for one iteration on a computer with 2.8GHz CPU and 2G memory. The comparison results are displayed in Figure 2, from which we have two observations: 1) RUN is better than the other two regression algorithms in label estimation accuracy; 2) the uncertain labels which consider the noise of the labels can further promote estimation accuracy of the RUN algorithm.

4.3. Databases for Age and Pose Estimation

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. Some sample images of a person are displayed in Figure 3. The evaluation method for the FG-NET database is the Leave-One-Person-Out (LOPO).

The other age database, Yamaha[1], contains 8000 Japanese facial images of 1600 persons with ages ranging from 0 to 93. Each person has 5 images and the Yamaha database is divided into two parts with 4000 images from 800 males and another 4000 images for 800 females. Our experiments are carried out separately on female and male subsets. For each subset, 1000 images are randomly selected for model training while the remaining 3000 samples are used for testing. To the best of our knowledge, Yamaha is the largest aging database ever reported.

For the FG-NET database, each person has multiple images of different ages, and hence many other algorithms such as *aging patterns subspace* (AGES) [8] and *Weighted Appearance Specific* (WAS) [12] are applicable. For comparison, we use the same feature set as in [8] for the FG-NET database. First, the first 200 appearance parameters [6] based on the 68 key facial points are used as input for age estimation. For detailed information on shape, texture and appearance parameters, please refer to [5]. For the Yamaha database, the positions of the key facial points are not provided, and hence the original gray level values are used instead as features by normalizing the images to size of 64-by-64 pixels and fixing the locations of the two eyes.

The Pointing'04 head pose database [9] consists of 15 subjects. The nose tips are manually marked, and we crop the faces to the size of 64×64 pixels. In our experiments, 52 images from four subjects are used, and 13 observers are invited for labeling the poses by rotating a 3D face head [18][2]. As shown in Figure 1, uncertainty exists in the labels from different observers.

4.4. Convergency Justification

In this subsection, we systematically evaluate the convergency property of the RUN algorithm from three aspects: 1) the convergency of the objective function value; 2) the convergency of the transformation matrix S^u, which is characterized by $\|S^u_t - S^u_{t+1}\|$; and 3) the convergency of the transformation matrix S^v, which is characterized by $\|S^v_t - S^v_{t+1}\|$.

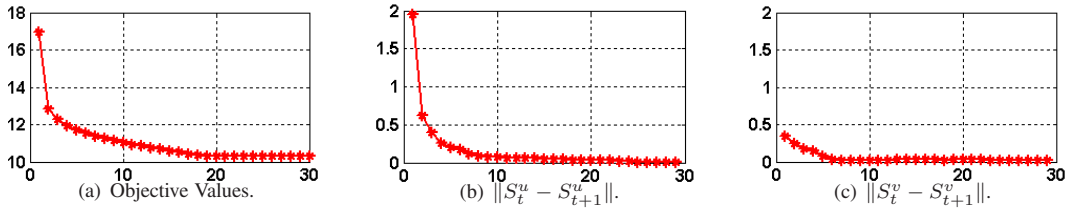
The above properties are evaluated on the FG-NET database, and detailed results are shown in Figure 4. In this experiment, we can see that the objective value monotonically decreases and converges together with the parameter matrix S^u and S^v after about 20 iterations.

4.5. Age Estimation Results

In this subsection, we conduct detailed age estimation experiments on the aforementioned age databases. According to [8], AGES achieves the best performance for age esti-

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

[2]These data labels are collected for comparison research of man and machine in pose estimation capability.



(a) Objective Values. (b) $\|S_t^u - S_{t+1}^u\|$. (c) $\|S_t^v - S_{t+1}^v\|$.

Figure 4. Convergence of the objective function, S^u and S^v. (a) objective value vs. iteration number. (b) $\|S_t^u - S_{t+1}^u\|$ vs. iteration number. (c) $\|S_t^v - S_{t+1}^v\|$ vs. the iteration number.

mation, followed by the WAS [12]. Thus we compare RUN with AGES and WAS for the experiments on the FG-NET aging database. For the RUN algorithm, the fixed age label is replaced with the corresponding uncertain one. As aforementioned, when we say that the label is a_i (integer) for the sample x_i, his/her exact age can be any real value within the interval $[a_i,\ a_i{+}1)$, and hence for RUN, the uncertain label for sample x_i is set as $[a_i,\ a_i{+}1\text{-}\epsilon]$ where ϵ is the minimal positive number that a computer can encode. Of course a more reliable estimator can be achieved if we are able to obtain specific uncertainty information for each training image.

Figure 5 displays the cumulative scores of different algorithms, Figure 6 displays the MAEs of different age groups for RUN with uncertain labels and RUN with fixed labels, and Table 1 lists the detailed MAEs of different algorithms. In our implementation, the vectors in the matrix U and V corresponding to the singular values less than 10^{-6} are removed. For the FG-NET database, the rank of the derived matrix U ranges from 9 to 33 with respect to different training sets in the LOPO strategy, and the derived matrix U takes the rank of 10 and 11 for the Yamaha Female and Male databases respectively. The final rank of matrix V turns out to be 1 or 2 for these two aging databases.

From these results, we can have several interesting observations:

1. RUN reaches the lowest MAEs across both databases. On the FG-NET database, RUN brings about 15% deduction of MAE compared with the state-of-the-art algorithm, *AGES* [8]. Also, the uncertain labels bring extra accuracy improvement over fixed labels for RUN algorithm as reported in Figure 6 on the FG-NET database.

2. The Yamaha database is more challenging for age estimation than the FG-NET database, and the estimation accuracies from the evaluated algorithms are relatively lower than those in the FG-NET database.

3. The average MAE from the QM based estimator is about 27% lower than that of the average MLP based estimator in the FG-NET database; while the MLP generally performs better than the QM in the Yamaha database.

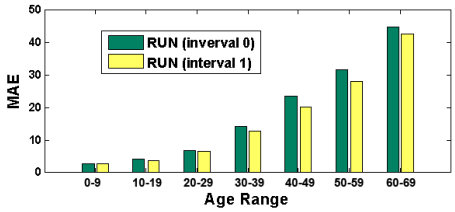


Figure 6. MAEs for different age groups of FG-NET database compared between RUN with age interval of width 1 and RUN with interval of width 0, namely fixed labels.

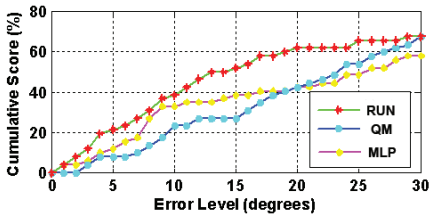


Figure 7. Cumulative scores of the pose estimation for Quadratic Model (QM), MLPs, and RUN at error levels from 0 to 30 degrees.

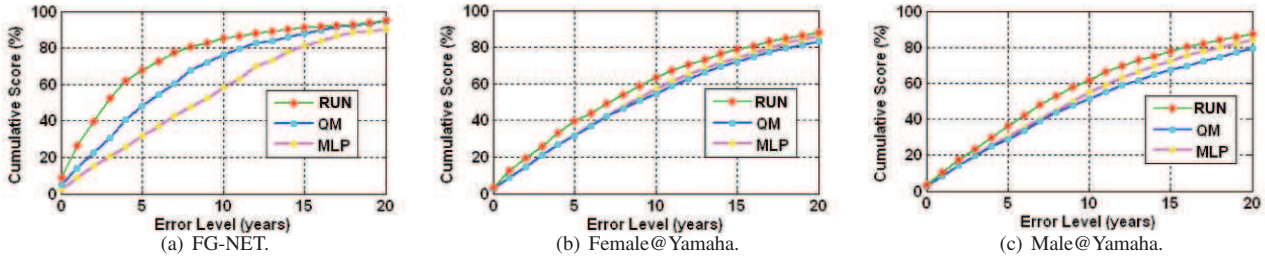
4.6. Pose Estimation Result

For pose estimation, all algorithms are conducted 13 times, and each time, the labels from one observer are used for training QM and MLP, and the labels with the interval length as the standard deviation of the labels from all 13 observers are used for training RUN algorithm. The leave-one-image-out experimental results are shown in Figure 7, which illustrates that RUN is better than both QM and MLP in accumulative scores for almost all error levels.

5. Conclusions and Future Works

In this paper, we presented a semidefinite programming formulation for automatic designing regressor based on training samples with uncertain nonnegative labels. Encouraging experimental results were achieved on two human aging databases, one of which is the largest one ever reported, and one pose database compared with the state-of-the-art regression algorithms.

Our proposed algorithm is general for regression problems with uncertain nonnegative labels, and we are planning to further investigate two aspects: 1) how to select the parameter γ in a rational way such that it acts as the weighting parameter in Support Vector Machines [13]; and 2) how to efficiently solve the SDP problem when the size of the training set is very large.



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

Figure 5. Cumulative scores of the age estimation for QM, MLP, and RUN at error levels from 0 to 20 years for different data sets.

Table 1. MAEs of different algorithms on two different databases and over different age ranges. Note that *#Samples* means the number of samples for each age group of the FG-NET database.

FG-NET					Female@Yamaha				Male@Yamaha			
Range	#Samples	RUN	QM	MLP	Range	RUN	QM	MLP	Range	RUN	QM	MLP
0-9	371	2.51	6.26	11.63	0-9	11.21	11.97	14.33	0-9	9.86	13.42	14.08
10-19	339	3.76	5.85	3.33	10-19	6.23	9.58	8.85	10-19	7.52	10.33	9.46
20-29	144	6.38	7.10	8.81	20-29	7.95	9.29	9.70	20-29	8.85	10.21	9.35
30-39	79	12.51	11.56	18.46	30-39	8.17	9.85	9.66	30-39	7.76	9.35	8.60
40-49	46	20.09	14.80	27.98	40-49	8.64	10.45	8.78	40-49	8.67	11.71	9.10
50-59	15	28.07	24.27	37.20	50-59	9.43	10.15	9.53	50-59	11.10	13.38	10.08
60-69	8	42.50	37.38	49.13	60-69	11.12	13.49	10.88	60-69	12.49	15.99	13.44
Average		**5.78**	7.57	10.39	70-93	15.56	19.66	16.52	70-93	16.60	20.44	19.69
Average	AGES: 6.77 [8]		WAS : 8.06 [8]		Average	**9.79**	11.80	11.03	Average	**10.36**	13.10	11.72

Acknowledgment

This work was funded in part by the U.S. Government VACE program. The views and conclusions are those of the authors, not of the U.S. Government or its agencies.

References

[1] The fg-net aging database: http://sting.cycollege.ac.cy/∽alanitis/fgnetaging/index.htm.

[2] A. Aspremont, L. Ghaoui, M. Jordan, and G. Lanckriet. A direct formulation for sparse pca using semidefinite programming. *Advances in Neural Information Processing Systems 17*, pp. 41–48, 2005.

[3] J. Bi, K. Bennett, M. Embrechts, C. Breneman, and M. Song. Dimensionality reduction via sparse support vector machines. *Journal of Machine Learning Research*, vol. 3, pp. 1229–1243, 2003.

[4] B. Borchers. Csdp, a C library for semidefinite programming. *Optimization Methods and Software*, vol. 11, no. 1, pp. 613–623, 1999.

[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] Y. Fu and T. Huang. Graph Embedded Analysis for Head Pose Estimation. *Proceedings of International Conference Automatic Face and Gesture Recognition*, pp. 3-8, 2006.

[8] X. Geng, Z. Zhou, Y. Zhng, G. Li, and H. Dai. Learning from facial aging patterns for automatic age estimation. *Proceedings of ACM Multimedia*, pp. 307–316, 2006.

[9] N. Gourier, D. Hall, and J. Crowley. Estimating face orientation from robust detection of salient facial features. *Proceedings of Pointing, ICPR, International Workshop on Visual Observation of Deictic Gestures, Cambridge, UK*, 2004.

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

[11] A. Lanitis, C. Draganova, and C. Christodoulou. Comparing different classifiers for automatic age estimation. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 34, no. 1, pp. 621–628, 2004.

[12] 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.

[13] 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.

[14] V. Nallure, J. Ye, and S. Panchanathan. Biased Manifold Embedding: A framework For Peson-Independent Head Pose Estimation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[15] B. Raytchev, I. Yoda, and K. Sakaue. Head pose estimation by nonlinear manifold learning. *Proceedings of the 17th International Conference on Pattern Recognition*, pp. 23-26, 2004.

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

[17] K. Weinberger and L. Saul. Unsupervised learning of image manifolds by semidefinite programming. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 988–995, 2004.

[18] Z. Zhang, Y. Hu, and T. Huang. Pose estimation comparison for man and machine. Internal Report of IFP, UIUC.