Head Pose Estimation: Classification or Regression?

Guodong Guo
Computer Science
NCCU
Durham, NC, 27707
gdguo@nccu.edu

Yun Fu
Beckman Institute
UIUC
Urbana, IL 61801
yunfu2@uiuc.edu

Charles R. Dyer
Computer Sciences
UW-Madison
Madison, WI, 53706
dyer@cs.wisc.edu

Thomas S. Huang
Beckman Institute
UIUC
Urbana, IL 61801
t-huang1@uiuc.edu

Abstract

Head pose estimation has many useful applications in practice. How to estimate the head pose automatically and robustly is still a challenging problem. In pose estimation, different pose angles can be used as regression values or viewed as different class labels. Thus a question is raised in our study: which is proper for pose estimation – classification or regression? We investigate representative classification and regression methods on the same problem to see any difference. A method that combines regression and classification approaches is also examined. Preliminary experiments show some interesting results which might prompt further exploration of related issues in pose estimation.

1. Introduction

Head pose estimation is important for many real applications, such as multi-view face recognition, focus of attention, human computer interaction, and human-centered scene interpretation. It is challenging to estimate the head pose automatically and robustly [10]. A general approach to head pose estimation is to learn the head poses from a set of labelled face images with the known pan and tilt angles as the labels [2]. Then the learned results are applied to the unknown test face images to predict their head poses.

In head pose estimation, the training data can be denoted as $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)\}, \mathbf{x} \in R^n, y \in R^2$, where $\mathbf{x}_i$ is the representation of a face image, and y_i is a pose label, such as the horizontal and vertical angles. When each pose label is considered as a class, the head pose estimation is a classification problem [10] [8]. On the other hand, the pose angles are ordered, the problem can also be thought of as a regression problem [1]. Which is proper for head pose estimation – *classification* or *regression*? To our best knowledge, no previous work has addressed this issue.

In this paper, we will explicitly compare the regression and classification methods for the head pose estimation problem. The motivation of this study is our recent work on human age estimation [4] [3]. We found that the classification and regression approaches may have very different results on human age prediction, given the same training data. A promising method is to combine the regressor and the classifier for the best performance on age estimation. Here we want to investigate whether the same phenomenon could be observed from another problem – head pose estimation. Following the same idea as in [4], we choose to use the support vector machine (SVM) as our classifier and the support vector regression (SVR) method as our regressor. Both methods have demonstrated good performance on many real world problems.

To evaluate the head pose estimation results, we use the Pointing'04 head pose database [2]. The reason is that this public available database contains a large number of head poses, e.g., (-90, -75, -60, -45, -30, -15, 0, 15, 30, 45, 60, 75, 90) in horizontal direction, and (-90, -60, -30, -15, 0, 15, 30, 60, 90) in vertical direction. Lots of previous work on head pose estimation used only a limited number of poses, and mostly varying just in the horizontal direction [12] [6]. Li et al. [7] used the SVM and SVR for face detection and recognition, but not for head pose estimation.

In the remaining of the paper, we first briefly review the support vector machine and support vector regression and indicate how they are adapted to our pose estimation problem. Then an integration scheme, which was first introduced in [4], was simply presented in Section 4 and used for pose estimation. Experimental results are described in Section 6.

978-1-4244-2175-6/08/$25.00 ©2008 IEEE

2. Support Vector Machine

Given a set of training vectors belong to two separate classes, $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)$, where $\mathbf{x}_i \in R^n$, $y_i \in \{-1, +1\}$, the linear SVM learns an optimal separating hyperplane, $\mathbf{w}\mathbf{x} + b = 0$, that maximizes the margin [11]. The SVM learning is to find the saddle point of the Lagrange functional,

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2} \parallel \mathbf{w} \parallel^2 - \sum_{i=1}^{l} \alpha_i \{y_i [(\mathbf{w} \cdot \mathbf{x}_i) + b] - 1\} \tag{1}$$

where α_i are the Lagrange multipliers. The Lagrangian has to be minimized with respect to $\mathbf{w}$, b and maximized with respect to $\alpha_i \geq 0$. The optimization is usually transformed to its *dual* problem,

$$\max_{\alpha} W(\alpha) = \max_{\alpha} \left\{ \min_{\mathbf{w},b} L(\mathbf{w}, b, \alpha) \right\}, \tag{2}$$

and the optimal hyperplane is represented by the dual solution, α, so

$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i \tag{3}$$

The value of b can be estimated by plugging $\mathbf{w}$ into the original equation, $\mathbf{w}\mathbf{x} + b = 0$.

In testing, the classification is given by

$$f(\mathbf{x}) = \text{sign}\left(\mathbf{w} \cdot \mathbf{x} + b\right), \tag{4}$$

for any new data point $\mathbf{x}$. If the training data are non-separable, slack variables ξ_i can be introduced [11].

The standard SVMs [11] deal with the two-class classification problem. For a multi-class classification problem, such as the 93 classes in our head pose estimation [2], there are three possible ways: 1) learning classifiers for each pair of classes, and taking a binary tree search in testing [9] [5]; 2) training SVMs for each class against all the remaining classes; and 3) training SVMs for all classes simultaneously. The last two schemes are not flexible when the number of classes changes, since the SVMs have to be re-trained from scratch. While in the first scheme there is no need to re-train the SVMs. All pair-wise SVM classifiers can be trained off-line, and only a limited number of pairs are involved in the binary tree search in testing.

3. Support Vector Regression

The basic idea of SVR is to find a function $f(\mathbf{x})$ that has most ϵ deviation from the actually obtained target y_i for the training data $\mathbf{x}_i$, and at the same time is as flat as possible. In other words, we do not care errors as long as they are less than ϵ.

Consider the problem of approximating the set of data $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)\}, \mathbf{x} \in R^n, y \in R$, with a linear function,

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b. \tag{5}$$

The optimal regression function [11] is given by

$$\begin{array}{ll} \min\limits_{\mathbf{w},\xi} & \frac{1}{2} \parallel \mathbf{w} \parallel^2 + C \sum_{i=1}^{l} (\xi_i^+ + \xi_i^-) \\ & y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle - b \leq \epsilon + \xi_i^+ \\ \text{subject to} & \langle \mathbf{w}, \mathbf{x}_i \rangle + b - y_i \leq \epsilon + \xi_i^- \\ & \xi_i^+, \xi_i^- \geq 0 \end{array} \tag{6}$$

where constant $C > 0$ determines the trade-off between the flatness of f and data deviations, and ξ_i^+, ξ_i^- are slack variables to cope with otherwise infeasible constraints on the optimization problem of (6). The ϵ-insensitive loss function is defined by

$$L_\epsilon(\mathbf{x}, y) = \begin{cases} 0 & \text{if } |f(\mathbf{x}) - y| < \epsilon \\ |f(\mathbf{x}) - y| - \epsilon & \text{otherwise} \end{cases} \tag{7}$$

The *primal* problem of (6) can be solved more efficiently in its *dual* formulation. A nonlinear regression function can be obtained by using kernels [11]. Different kernels, such as polynomials, sigmoid, or Gaussian radial basis functions, can be used depending on the tasks. In our evaluation of head pose estimation, the Gaussian radial basis function kernel was adopted.

The standard SVR usually deals with the regression problem where y_i is a scalar value. In our pose estimation, the poses can vary by angles in both horizontal and vertical directions. To adapt the standard SVR for this problem without changing its formulation, we do pose regression in two directions separately. That is, all training data are re-grouped by horizontal angles for the SVR learning, and then the same data are re-grouped again by vertical angles for the SVR learning. Two different regression functions are learned separately. In testing, the new input $\mathbf{x}$ will go through two SVR functions and its pose is obtained from the two SVR outputs.

4. Combining Regression and Classification

As demonstrated in [4], a much better estimation performance might be obtained by integrating the SVR and SVMs for human age estimation. A method, called Locally Adjusted Robust Regressor (LARR), was proposed in [4] for combining the SVR and SVMs. The key idea is to adjust the regression values delivered by the SVR locally so that the estimated values can be "dragged" towards the true values. The

idea of the LARR is illustrated in Figure 1. Suppose the predicted value by SVR is $f(\mathbf{x})$, corresponding to the input data $\mathbf{x}$. The point $f(\mathbf{x})$ is displayed by the black dot on the regression curve. The estimated value, $f(\mathbf{x})$, may be far away from the true value, L, shown as the red dot on the true trajectory curve. The LARR method slides the estimated value, $f(\mathbf{x})$, up and down (corresponding to greater and smaller age values in age estimation) by checking different possible values, $t \in [f(\mathbf{x}) - d, f(\mathbf{x}) + d]$, to see if it can come up with a better estimation. The value d indicates the range of possible values for local search. Hopefully the true value, L, is within this range, i.e., $L \in [f(\mathbf{x}) - d, f(\mathbf{x}) + d]$.

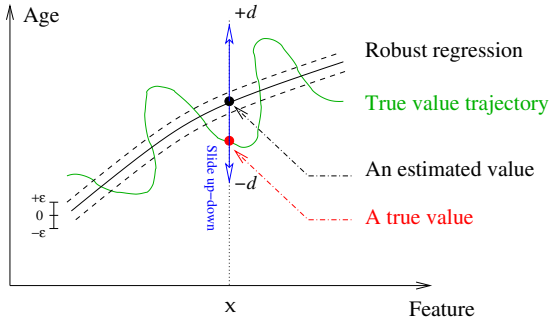

Figure 1. Illustration of the LARR idea [4].

To use the LARR method [4] for head pose estimation, we modify the local adjust behavior from one dimension (which is proper for age estimation) to two dimensions (horizontal and vertical pose angles). Specifically, given an input pattern $\mathbf{x}$, two SVRs are used for estimating the horizontal and vertical angles, θ_v and θ_h, and the two dimensional Euclidean distances are computed between the estimated angles with all possible 93 poses and sorted in ascending order. Then the closest classes are used for the local adjustment by the SVMs, working on a much smaller number of pairs of classes.

5. Experiments

The experiments are performed on the Pointing'04 head pose database [2], which is a public database containing a large span of head poses, -90 to 90 in both horizontal and vertical directions. The data set consists of 15 subjects, each performs 13 pose variations in horizontal and 7 in vertical, together with two extreme cases in the vertical 90 and -90 degrees. The subjects wear glasses or not and have various skin colors. The database is divided into the training and test sets, each with 93 images of the same person at different poses. There are totally 1395x2=2790 images in the database.

Table 1. MAEs in degrees.

Various Setup	Vertical	Horizontal	Average
SVR	9.37	7.84	8.61
linear SVM	4.90	59.74	32.32
kernel SVM	4.73	59.91	32.32
LARR2	7.69	9.23	8.46
LARR4	7.91	13.34	10.63
LARR8	8.59	20.28	14.44
LARR16	7.98	30.31	19.15
LARR32	6.95	42.09	24.52
LARR64	5.18	54.92	30.05

We manually marked the nose-tips for each face image, and cropped the image patch of size 18x18, similar to that in [10]. Here our focus is to evaluate the classification and regression methods for head pose estimation, the manual marking avoids possible errors caused by misalignment. Cropped image pixel values are normalized and each image patch is reshaped into a vector of dimension 324. Then the same training data are used for the SVM and SVR learning, and also for the LARR method [4].

Similar to our work on age estimation [4], the performance of head pose estimation is measured by two different measures: the Mean Absolute Error (MAE) and the Cumulative Score (CS). The MAE is defined as the average of the absolute errors between the estimated poses and the ground truth, $MAE = \sum_{k=1}^{N} |\hat{l_k} - l_k|/N$, where l_k is the ground truth pose for the test image k, $\hat{l_k}$ is the estimated pose, and N is the total number of test images. The MAE is computed for the horizontal and vertical angles separately in order to see the performance in detail. The two MAEs can be averaged to get the estimate of errors in both directions. We believe that the MAEs can deliver more information than the simple error rates, because MAEs can measure how far away the estimated results are from the ground truth. The cumulative score is defined as $CS(j) = N_{e \leq j}/N \times 100\%$, where $N_{e \leq j}$ is the number of test images on which the pose estimation makes an absolute error no higher than j degrees.

The MAEs are shown in Table 1, from which we can get some interesting observations. The SVR performs well in either horizonal or vertical poses, while the SVM performs better than the SVR in the horizontal direction, but much worse in the vertical direction. The reason may be that the distributions of the face patterns have large overlapping in the horizonal directions but less serious in the vertical direction, given 93 classes to discriminate. The SVR can be viewed as a global

method that uses all input data for learning, and thus can deal with the overlapping to some extent.

The average MAEs of SVM is 32.32 degrees which is much higher than the 8.61 degrees of the SVR. The SVR uses a Gaussian kernel, while the SVM uses a linear kernel. To see if non-linear SVMs could improve the performance, we also used a Gaussian kernel for the SVMs, but not too much difference as shown in row 3 of Table 1.

Next, we tried the LARR method with different ranges to search. From Table 1, one can observe that the LARR2 method gives better results than using other different ranges. When a larger range is used to search, the SVMs will have a pose estimate very different from the ground truth. A smaller range really constrains the SVMs from being trapped into a "bad" local optimum in the binary tree search.

The cumulative scores of the linear SVM, SVR, and the LARR2 methods are shown in Figure 2.

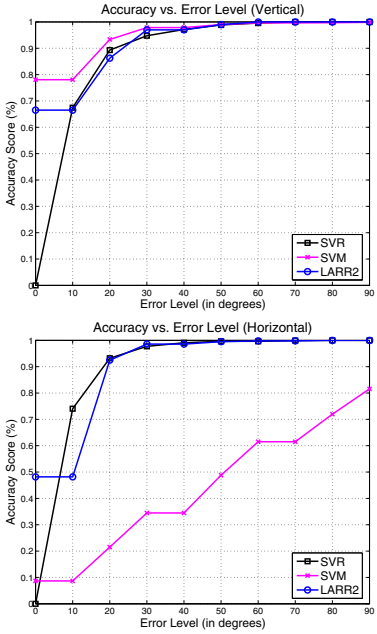

Figure 2. Comparisons of the SVM, SVR, and the LARR methods in terms of cumulative scores in the vertical (top) and horizontal (bottom) directions, respectively.

We summarize the main observations from our experiments: 1) The SVM performs the best in the vertical direction, but worst in the horizontal direction; 2) Linear or non-linear SVMs do not show much differences; 3) The SVR with a Gaussian kernel performs well in both directions; and 4) the LARR method performs well for pose estimation using small ranges to search, while large ranges make the performance worse towards the SVMs.

In the future, we will further investigate the related issues by using more advanced features such as the tensor model [10].

6. Conclusion

We have evaluated the classification (SVM) and regression (SVR) methods for head pose estimation. Preliminary experiments on the Pointing'04 database show some interesting results. The SVR performs well in either horizontal or vertical pose variations. The SVM performs better in vertical direction but much worse in the horizontal direction. The LARR method that combines the SVM and SVR performs well when the local search range is small. The interesting observations might inspire further exploration of related issues in head pose estimation.

References

[1] S. Y. et al. Learning auto-structured regressor from uncertain nonnegative labels. In *ICCV*, 2007.

[2] N. Gourier, D. Hall, and J. L. Crowley. Estimating face orientation from robust detection of salient facial features. In *ICPR Pointing'04 Workshop*, 2004.

[3] G. Guo, Y. Fu, C. Dyer, and T. Huang. Image-based human age estimation by manifold learning and locally adjusted robust regression. *IEEE Trans. on Image Processing*, 17:1178–1188, July 2008.

[4] G. Guo, Y. Fu, T. Huang, and C. Dyer. Locally adjusted robust regression for human age estimation. In *IEEE WACV*, January 2008.

[5] G. Guo, S. Li, and K. Chan. Support vector machines for face recognition. *Image and Vision Computing*, 19:631–638, 2001.

[6] S. Li, Q. Fu, L. Gu, B. Scholkopf, Y. Cheng, and H. Zhang. Kernel machine based learning for multi-view face detection and pose estimation. In *ICCV*, 2001.

[7] Y. Li, S. Gong, and H. Liddell. Support vector regression and classification based multi-view face detection and recognition. In *IEEE FGR*, pages 300–305, 2000.

[8] Z. Li, Y. Fu, J. Yuan, T. Huang, and Y. Wu. Query driven localized linear discrimination models for head pose estimation. In *ICME*, pages 1810–1813, 2007.

[9] M. Pontil and A. Verri. Support vector machines for 3-d object recognition. *IEEE PAMI*, 20:637–646, 1998.

[10] J. Tu, Y. Fu, Y. Hu, and T. Huang. Evaluation of head pose estimation for studio data. In *CLEAR06*, pages 281–290, 2006.

[11] V. N. Vapnik. *Statistical Learning Theory*. John Wiley, New York, 1998.

[12] Y. Wei, L. Fradet, and T. Tan. Head pose estimation using gabor eigenspace modeling. In *ICIP*, pages 281–284, 2002.