21st International Conference on Pattern Recognition (ICPR 2012)
November 11-15, 2012. Tsukuba, Japan

Cluster-Classification Bayesian Networks for Head Pose Estimation

Mehran Kafai, Bir Bhanu, and Le An
Center for Research in Intelligent Systems, University of California, Riverside, USA
mkafai@cs.ucr.edu bhanu@cris.ucr.edu lan@ee.ucr.edu

Abstract

Head pose estimation is critical in many applications such as face recognition and human-computer interaction. Various classifiers such as LDA, SVM, or nearest neighbor are widely used for this purpose; however, the recognition rates are limited due to the limited discriminative power of these classifiers for discretized pose estimation. In this paper, we propose a head pose estimation method using a Cluster-Classification Bayesian Network (CCBN), specifically designed for classification after clustering. A pose layout is defined where similar poses are assigned to the same block. This increases the discriminative power within the same block when similar yet different poses are present. We achieve the highest recognition accuracy on two public databases (CAS-PEAL and FEI) compared to the state-of-the-art methods.

1. Introduction

Head pose estimation, the process of inferring the orientation of the human head, is critical in many applications such as face analysis and human-computer interaction. The human head is often considered as a rigid object; thus, three degree-of-freedoms (pitch, yaw, roll) need to be identified in order to describe the orientation of the head pose. The pitch and yaw angles describe the out-of-the-plane rotation and the roll angle accounts for the in-plane rotation.

The first type of head pose estimation methods is appearance based. Given an image with unknown pose, it is compared to a set of labeled data and the pose is determined by measuring the similarity between the image and the labeled data. Earlier works used normalized cross-correlation at different resolutions to determine the head pose. Gabor features [7] are also used to highlight the oriented features and perform pose estimation. In recent years, inspired by the success in pedestrian detection, Histogram of Oriented Gradients (HOG) has been widely used for head pose estimation. In [8] supervised local subspace learning is used to learn a local linear model from HOG features of the training data. Dong *et al.* [3] proposed a new image descriptor called Covariance of Oriented Gradients (COG) and reported higher recognition rates compared to other HOG based approaches. The advantage of an appearance based method is that only positive examples are required in the labeled data and the dataset can be easily extended.

Another approach is to employ a 3D model [2]. Due to the high computational cost to build the 3D model and the requirement for accurate detection and registration of facial features, this type of approach is less preferred compared to the appearance based methods. In [4] random forests are utilized to solve head pose estimation from 3D depth data and it is formulated as a regression problem.

In recent years, manifold embedding methods such as ℓ^1 graph regularization [9] have also attracted much interest. In these methods, it is assumed that the high-dimensional image sample lies on a low-dimensional manifold with the possible pose variations as constraints.

Being treated as a classification problem, the discriminative power of a classifier directly affects the classification accuracy. When the number of discrete poses becomes larger, the commonly used classifiers such as Linear Discriminant Analysis (LDA) and Support Vector Machine (SVM) cannot achieve sufficiently high performance to perform tasks such as face analysis.

In this paper, we introduce the *Cluster-Classification* Bayesian Network (CCBN) as part of a novel head pose estimation algorithm. The proposed CCBN is a hierarchical Bayesian network specifically designed to perform classification after clustering. The poses with similar appearance are first clustered into different blocks. Then a Bayesian network is built. Given a testing face, its pose is inferred from the Bayesian network. By assigning similar poses into the same block, the discriminative power is increased within the same block when similar yet different poses are present. We test the proposed method on two public datasets; CAS-PEAL [6] and FEI [1]. Our approach achieves the highest recognition accuracy on both datasets compared to the *state-of-the-art* methods.

In the rest of this paper, Section 2 discusses the technical approach. Experimental results are reported in Section 3, and the conclusions are drawn in Section 4.

978-4-9906441-0-9 ©2012 ICPR

2. Technical Approach

Bayesian networks graphically represent and factor joint probability distributions effectively. This important property makes them suitable for classification purposes. A Bayesian network is defined as a directed acyclic graph $G = (V, E)$ where the nodes represent random variables and the edges symbolize the direct dependencies between the random variables. For a Bayesian network with n nodes $X_1, X_2, \ldots, X_n$ the full joint distribution is defined as:

$$p(x_1, x_2, \ldots, x_n) = p(x_1) \times p(x_2|x_1) \times \ldots$$

$$\times p(x_n|x_1, x_2, \ldots, x_{n-1}) = \prod_{i=1}^{n} p(x_i|x_1, \ldots, x_{i-1}). \quad (1)$$

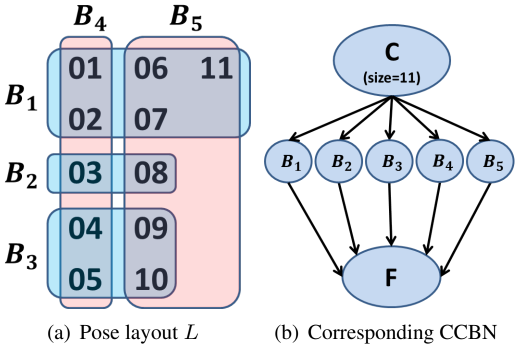
A node in a Bayesian network is only conditional on its parent's values; thus,

$$p(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} p(x_i|parents(X_i)), \quad (2)$$

where $p(x_1, x_2, \ldots, x_n)$ is an abbreviation for $p(X_1 = x_1 \wedge \ldots \wedge X_n = x_n)$. In other words, a Bayesian network models a probability distribution if each variable is conditionally independent of all its non-descendants in the graph given the value of its parents.

In this paper, we introduce the CCBN, a modified Bayesian network with a specific structure capable of performing classification after clustering. Say we have m poses $P_1, P_2, \ldots, P_m$, each with a unique ID (each pose corresponds to a class). Initially, we define a layout L for the m poses such that similar poses are located in neighboring positions. Depending on the images, the layout can be one, two, or three dimensional. Thereafter, the layout is partitioned into blocks where each block holds similar poses. The layout L is not unique and the partitioning may be performed using systematic or heuristic methods. A simple way to define the blocks is to group the poses based on their visual similarity by just looking at the data. The number of blocks, m, are predetermined (similar to k-means). Each pose is a member of at least one block. Let's clarify this with an example. Say 11 poses are available and the blocks are determined as $B_1 = \{1, 2, 6, 7, 11\}$, $B_2 = \{3, 8\}$, $B_3 = \{4, 5, 9, 10\}$, $B_4 = \{1, 2, 3, 4, 5\}$, and $B_5 = \{6, 7, 8, 9, 10, 11\}$. Figure 1(a) illustrates this layout. Pruning can be used to optimize the CCBN and reduce the computational complexity without changing the probability distributions on the variables of interest. A corresponding CCBN is generated after the pose blocks are defined. Figure 1(b) presents the CCBN corresponding to the layout in Figure 1(a).

A CCBN is a hybrid hierarchical Bayesian network with three different types of nodes:



(a) Pose layout L (b) Corresponding CCBN

Figure 1. Sample pose layout and CCBN

1. *Class node*. This is the top layer node and holds the probabilities of the data belonging to each class. The class node is a discrete node with a node size equal to the number of classes (poses). This node is represented by the random variable C.

2. *Features node*. This is the bottom layer node and corresponds to the feature vector representing the data. Depending on the data, the feature node can be discrete or continuous. The node size is equal to the dimensionality of the data. This node is represented by the random variable F.

3. *Block nodes*. These nodes are discrete, binary, and define the middle layer. Each block node represents a block on the pose layout. A block node B_i determines the membership probability of the data to block i vs. all the other blocks. The block nodes are represented by random variables $B_1, B_2, \ldots, B_m$.

To validate the CCBN structure in Figure 1(b), we use the K2 algorithm described in [5] to determine a sub-optimal structure (learning the best structure/topology takes exponential time and a sub-optimal structure is a good approximation here). K2 is a greedy algorithm that incrementally adds parents to a node according to a score function. In this paper we use the Bayesian Information Criterion (BIC) function as the scoring function.

By utilizing such a classification-after-clustering structure, similar yet different poses have dissimilar probability distributions over the block node they belong to; thus, the discriminative power is increased within the same block by assigning similar poses into the same block.

The block nodes in a CCBN can be combined to create cluster nodes using join tree algorithms. This is done to improve efficiency of computing posterior probabilities for all random variables in a BN. We avoid doing so because we do not require computing the posterior probability for all nodes; our goal is to compute

$$max_{class} P(C|F), \quad (3)$$

which represents the class label with the highest probability.

From Equation (2) the joint probability distribution for a given CCBN with class node C, feature node F, and block nodes $B_1, B_2, \ldots, B_m$ is defined as:

$$P(C, B_1, \ldots, B_m, F) =$$

$$P(C) \times P(F|B_1, \ldots, B_m) \times \prod_{i=1}^{m} P(B_i|C). \qquad (4)$$

All CCBN parameters on the right hand side of Equation (4) are computed during training of the CCBN. Thereafter, inference is performed where a probability distribution over the set of pose classes is assigned to the feature vector representing a face image and the class with the highest posterior probability is selected as the classification result. The probability of a given data f being from class c_k is formulated as:

$$P(C = c_k|F = f) = \frac{P(C = c_k, F = f)}{P(F = f)}, \qquad (5)$$

where

$$P(C = c_k, F = f) = \sum_{B_1,\ldots,B_m} P(C = c_k)$$

$$\times \prod_{i=1}^{m} P(B_i|C = c_k) P(F = f|B_1, \ldots, B_m) \qquad (6)$$

and

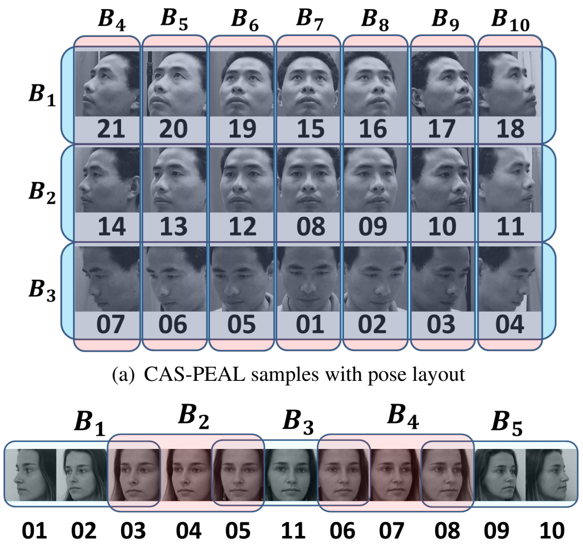
$$P(F = f) = \sum_{C,B_1,\ldots,B_m} P(C) \prod_{i=1}^{m} P(B_i|C)$$
$$\times P(F = f|B_1, \ldots, B_m).$$

3. Experimental Results

We use face images from the FEI [1] and the CAS-PEAL [6] databases. For our experiments, we use 2200 images from the FEI database representing 200 individuals under 11 different poses from full profile left to full profile right, and 4200 images from the CAS-PEAL of 200 individuals under 21 various poses. The CAS-PEAL images were selected according to the experimental set up in [3, 10] for the purpose of fair comparison. The CAS-PEAL poses have IDs 1 to 21, and FEI database poses have IDs 1 to 11. All images are resized to 32×32 and aligned. Each image is divided into 8×8 blocks and there are 15 bins in each histogram for each block, resulting in a 240 dimensional HOG feature vector. k-fold cross validation is used to evaluate the performance with k=10. For each fold, 150 individuals are used for training and 50 individuals for testing.

3.1. Pose Layouts

Figure 2 presents the pose layouts for both CAS-PEAL and FEI databases. Each pose layout is overlaid on sample images from its corresponding database.



(a) CAS-PEAL samples with pose layout

(b) FEI samples with pose layout

Figure 2. Pose layout and samples for CAS-PEAL and FEI databases

The CCBN structure for the CAS-PEAL pose layout has ten block nodes $B_1, \ldots, B_{10}$ (Figure 3), and for the FEI pose layout five block nodes $B_1, \ldots, B_5$.

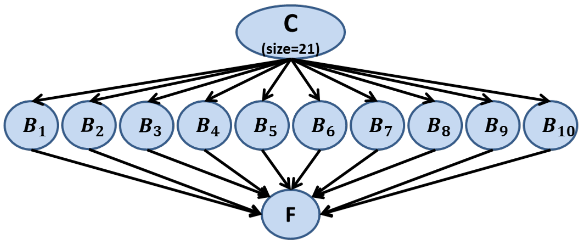


Figure 3. CCBN structure for CAS-PEAL pose layout

3.2. Comparison with other methods

Results on CAS-PEAL database: Table 1 presents pose estimation accuracy for different methods on the CAS-PEAL database. To show how CCBN compares to other classifiers, we report the accuracy for 4 other classifiers: Nearest Neighbor (NN), SVM (linear), LDA, and Naive Bayes (NB). Three representative feature descriptors are used for each classifier: Local Binary Patterns (LBP), HOG , and GaFour [10]. The results show that using CCBN improves the accuracy compared to the NN, LDA and SVM classifiers for all the three descriptors. CCBN with the HOG descriptor achieves the highest accuracy 96.91% compared to all other classifiers and descriptors reported in Table 1. In this case, HOG successfully encodes the head poses. To the best of our knowledge, the highest recognition rate on the same CAS-PEAL database is reported as 95.33% in [3] using Covariance of Oriented Gradients (COG) features and Nearest Centroid (NC) classifier. This is inferior to the performance of CCBN+HOG.

Table 1. Accuracy percentages comparison

Classifier→ Descriptor↓	NN	LDA	SVM	NB	CCBN
LBP	84.67	86.12	87.20	86.44	**89.74**
HOG	86.85	91.86	95.04	91.27	**96.91**
GaFour [10]	82.96	88.29	92.76	89.81	**94.33**

Results on FEI database: Based on the results in Table 1, we choose to use HOG as the feature descriptor for the experiments on the FEI database. Table 2 shows the pose estimation results of each pose for CCBN, NN, LDA, and SVM on the FEI database. CCBN has greater accuracy than the other three classifiers for 9 out of the total 11 poses. The average accuracy for CCBN is 3.48% more than SVM, 5.81% more than LDA, and 11.21% more than NN.

Table 2. Accuracy percentage comparison for CCBN vs. NN, LDA, and SVM

Pose↓	NN	LDA	SVM	This paper
1	81.41	91.43	94.65	**97.63**
2	86.68	88.67	**95.76**	95.34
3	83.22	88.70	90.88	**95.04**
4	85.20	91.21	92.06	**94.69**
5	83.91	92.44	92.46	**95.93**
6	85.61	93.09	**95.20**	94.99
7	83.80	89.67	92.44	**98.93**
8	85.98	91.37	94.40	**98.04**
9	86.48	90.84	88.90	**96.96**
10	88.42	87.13	92.29	**96.93**
11	84.61	90.12	91.35	**95.40**
avg.	85.03	90.43	92.76	**96.24**

Figure 4 presents the Receiver Operating Characteristic (ROC) curves for all of the four classifiers from Table 2. The results show how each classifier performs on the FEI database using HOG as the descriptor. Clearly, CCBN outperforms NN, LDA, and SVM.

4. Conclusions

In this paper, we introduced a novel pose estimation method using a Cluster-Classification Bayesian Network (CCBN). Before defining the CCBN, a pose layout is generated where similar poses are grouped in blocks. Given a face, the class label is determined as the one with the highest probability conditioned on the feature descriptor. By clustering similar poses into the same block, the trained classifier is more discriminative in these similar poses. The CCBN is tested on two public datasets CAS-PEAL and FEI. The comparisons are made among different classifiers and different features. The experimental results show that the CCBN improves the classification accuracy compared to the other classifiers. Also, the CCBN classifier with HOG as the feature descriptor provides the best performance.

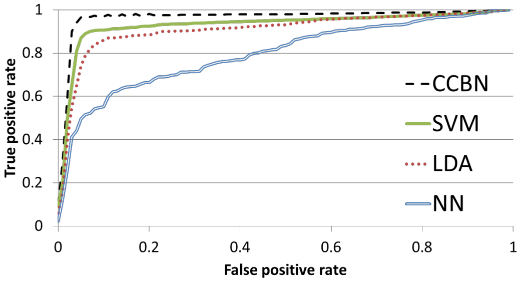


Figure 4. Performance ROC plot

5. Acknowledgments

We thank Suresh Kumar for his discussions. This work was supported in part by NSF grant number 0905671.

References

[1] FEI face database. www.fei.edu.br.
[2] J. Dai and R. Chung. Head pose estimation by imperceptible structured light sensing. In *ICRA*, pages 1646–1651, 2011.
[3] L. Dong, L. Tao, and G. Xu. Head pose estimation using covariance of oriented gradients. In *Proc. ICASSP*, 2010.
[4] G. Fanelli, J. Gall, and L. Van Gool. Real time head pose estimation with random regression forests. In *CVPR*, pages 617–624, June 2011.
[5] N. Friedman. Being bayesian about network structure. In *Machine Learning*, pages 201–210, 2000.
[6] W. Gao, B. Cao, S. Shan, X. Chen, D. Zhou, X. Zhang, and D. Zhao. The CAS-PEAL large-scale chinese face database and baseline evaluations. *IEEE TSMC B*, 2008.
[7] C. Huang, X. Ding, and C. Fang. Head pose estimation based on random forests for multiclass classification. In *ICPR*, pages 934–937, Aug. 2010.
[8] D. Huang, M. Storer, F. De la Torre, and H. Bischof. Supervised local subspace learning for continuous head pose estimation. In *CVPR*, pages 2921–2928, June 2011.
[9] H. Ji, F. Su, and Y. Zhu. Robust head pose estimation via semi-supervised manifold learning with ℓ^1-graph regularization. In *IJCB*, pages 1–6, Oct. 2011.
[10] B. Ma, S. Shan, X. Chen, and W. Gao. Head yaw estimation from asymmetry of facial appearance. *IEEE TSMC B*, 2008.