Bayesian Network-based Classifiers for Face Detection

Yan Zhao, Minjun Huang

Problem Statement

Face detection is a prerequisite of face recognition in many artificial intelligence systems. Because faces appear in different scales, illuminations, colors, orientations and different head poses in an image, fast and accurate detection remains a challenging task for profile faces or faces with perturbations such as shadows, glasses and beard. Therefore, our research concentrates on faces that are hard to detect with current algorithms [1]. Currently, major methods of face detection are based on maximum likelihood learning method, which is computationally efficient but might converge to suboptimal due to mismatch between its optimization criteria (parameter likelihood) and its actual goal of classification (label prediction accuracy) [2]. In contrast, Bayesian Network (BN) can reflect both the conditional independencies and dependencies among features of human face and therefore is more efficient and accurate [3].There are several algorithms to learn BN from data in an acceptable time and space complexity. Moreover, BN is robust to error or missing data. Therefore, we choose BN as our classification method.

Methods

1) Feature generation

1.1) Differential vector generation [4]: locate and separate every face in the FDDB and develop them into 5 groups: frontal face, 45-degree left oriented face, 45-degree right oriented face, left profile face and right profile face.
Each face in the CMU-MIT data set will be compared to the 5 mean faces, and then generate a differential image with the mean face that it is most similar to. This differential image will be transformed into a feature vector **Xd**.

1.2) Appearance vector generation [5]: The appearance of two human faces in gray scale are similar if their in-plane and out-of-plane rotation are similar. Moreover, each feature (mouth, nose, etc) has similar gray scales. Therefore we convert each image into an 8-level gray scale image and then transform it into a vector **Xa** representing the appearance of this face.

1.3) Haar Feature Vector generation[6]: Implement 2-D Haar wavelet transform to each image in the CMU-MIT data set, and translate the output image into a vector **Xh** representing the 2-D Haar feature of the face.

Last, we'll combine **Xd, Xa, Xh** into a vector **Xf,** representing these 3 features of human face.

1.4) Collect from internet a set of non-face images and generate a vector **Xnf,** based on the same process from 1.1 to 1.3.

2) Network training
A BN is a directed acyclic graph that encodes a joint probability distribution over a set of random variable X={X1, X2, ..., XN}[7]. It is defined by the pair B = {G, p}. G is the structure of the BN. P represents the probabilistic distribution of a set of nodes of G.

2.1) Structure Learning
The process of structure learning is finding a structure G best representing the probabilistic distribution of features **Xf** and **Xnf,** which means that in structure G **Xf** and **Xnf** has the greatest differences in all possible network structures.

2.2) Parameter Learning
the function of P is to accurately predict the label **Yf** or **Ynf** given the attributes of vector **X**. This is equal to maximizing the conditional likelihood of **Y** given **X**. However, for a generative model of G, a direct optimization approach requires computationally expensive numerical techniques. As a result we take the ensemble approach and maximize the classification performance of the ensemble BN classifier by minimizing ELF (exponential loss function). Then combined with the structure G, a BN classifier is created.

Data

We will be using 2 data sets. The first data set is the Face Detection Data Set and Benchmark (FDDB) from the University of Massachusetts [8]. This data set contains 5171 faces in a set of 2845 images, and would be used to generate the mean faces. The second data set is the CMU-MIT Face data set [9]. We would use the second data set to train and test our BN classifier. The pictures in the first data set are in RGB format, and will be converted to gray scale.

Timeline

1/24 Project proposal
1/31 Preprocessing of all images in the data sets
2/7 Image feature extraction
2/14 Bayesian network training and testing
2/19 Project milestone write up
2/26 Performance analysis and comparison with existing algorithms
3/7 Poster and final write up

References

[1] Pham, T. V.; Worring, M. & Smeulders, A. W. Face detection by aggregated Bayesian network classifiers. *Pattern Recognition Letters*, **2002**, *23*, 451 - 461

[2] Jing, Y.; Pavlović, V. & Rehg, J. Boosted Bayesian network classifiers. *Machine Learning, Springer US*, **2008**, *73*, 155-184

[3] Neapolitan, R. E. Learning Bayesian networks. *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM*, **2007**, 5:1-5:1

[4] Liu, C. A Bayesian discriminating features method for face detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **2003**, *25*, 725 - 740

[5] Kith, V.; El-Sharkawy, M.; Bergeson-Dana, T.; El-Ramly, S. & Elnoubi, S. A feature and appearance based method for eye detection on gray intensity face images. *Computer Engineering Systems, 2008. ICCES 2008. International Conference on*, **2008**, 41 -47

[6] Chai, C. & Wang, Y.Face detection based on extended Haar-like features. *Mechanical and Electronics Engineering (ICMEE), 2010 2nd International Conference on*, **2010**, *1*, V1-442 -V1-445

[7] Pearl, J. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. *Morgan Kaufmann*, **1988**

[8] http://vis-www.cs.umass.edu/fddb/

[9] http://vasc.ri.cmu.edu/idb/html/face/