Bayesian Network-based Classifiers for Face Detection

Yan Zhao, Minjun Huang

Recap of Problem Statement

Face detection is a prior step of face. Accurate detection of human faces in any arbitrary scene is the most important prerequisite of face recognition. But face detection is easily subject to scales, illuminations, colors, orientations and head poses in an image. Also detection of for profile faces or faces with perturbations such as shadows, glasses and beard remains a challenging task. Therefore, our research concentrates on faces that are hard to detect with current algorithms [1]. Bayesian Network (BN) can reflect both the conditional independencies and dependencies among features of human face and therefore is more efficient and accurate [2]. Some existing algorithms [3] are able to learn BN from data in an acceptable time and space complexity. Therefore, we choose BN as our face detection classifier.

Progress

Matlab codes are developed for the following.

1) Image feature extraction and vectorization

After preprocessing, the training set and testing set each contain about: 454 faces and 70 non-faces. Then each image in both sets is transformed into a vector of the form **[Xh, Xd, Xa, C]**, where **Xh** is the 2-D Haar feature [4] of the image, **Xd** is the differential feature [5] when compared to the 5 mean faces, **Xa** is the appearance feature [6], and **C** is class label, where -1 represents non-face and 1 represents face.

2) Network training

A BN is a directed acyclic graph (see Fig 1) that encodes a joint probability distribution over a set of random variable $X=\{X_1, X_2, ..., X_N\}$[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.

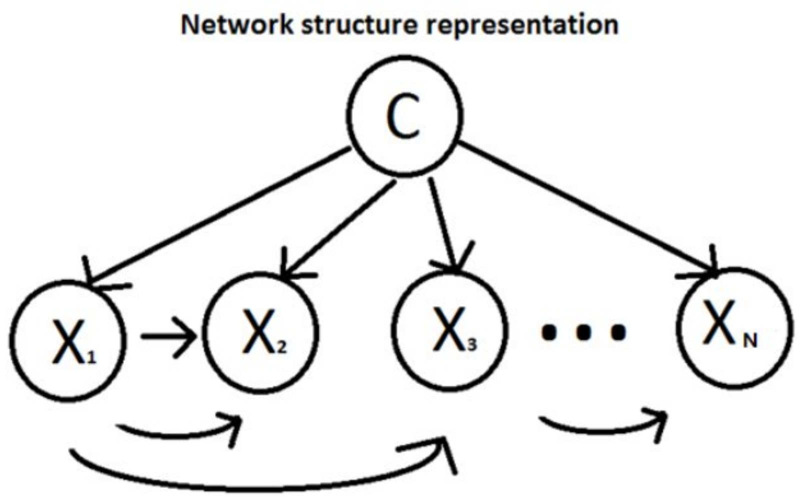

Fig 1 BN structure representation

2.1) Structure G Learning

We applied the hill-climbing algorithm in BN structure learning [8]. The steps are described as follows:

⓪ Initialize network (G_1) to naive Bayes (G_{NB})

① Evaluate the current classifier through misclassification rate by 10-fold cross validation on training set

② Consider adding every legal arc from Xi to Xj to the current classifier (Gs), where Xj belongs to Orphan node sets and Xi is not equal to Xj

③ Compute mutual conditional entropy (CCE) between Xj and Xi with each j. Add arc (j, i) that has Max CCE to current Gs. Remove Xi from Orphan set. Go to step ①

④ Repeat ①-③ until there is only one node in the Orphan set or no improvement can be made to Current Gs

2.2) Parameter P Learning

We applied the AdaBoost algorithm in BN parameter learning [2]. The steps can be described as follows:

⓪ Given a base structure G and the training data, initialize training data weights (averaged)

① Repeat for k =1, 2, … : Given G, θ_k is learning through ML on the weighted data D_k, compute the weighted error $Err_k = E_\omega\left[1_{x_c \neq f\theta_k(x_a)}\right]$, $\beta_k = 0.5log\frac{1-Err_k}{Err_k}$, where $f\theta_k(x_a)$ is classifier outcome on selecting label x_c given x_a during boosting iteration k, x_c is the true label of x_a, and β_k is the corresponding weight. Update weights $\omega_i = \omega_i \exp\{-\beta_k x_c^i f(x_a^i \mid \theta_k, G_k)\}$ and normalize.

② Ensemble output: sign $\sum_k \beta_k f(x_a \mid \theta_k, G_k)$

Then we combined the learned parameter θ with the structure G and obtain a BN classifier.

Preliminary result

The number of nodes in a network will affect the prediction accuracy and speed. We test our BN classifier with different network node number by resizing images into 4*4 (which would yield a smaller network size) and 20*20 (which would yield a larger network size) pixel. The prediction error rate is shown in Fig 2. Also we include the comparison of our BN classifier with Naive Bayes classifier.

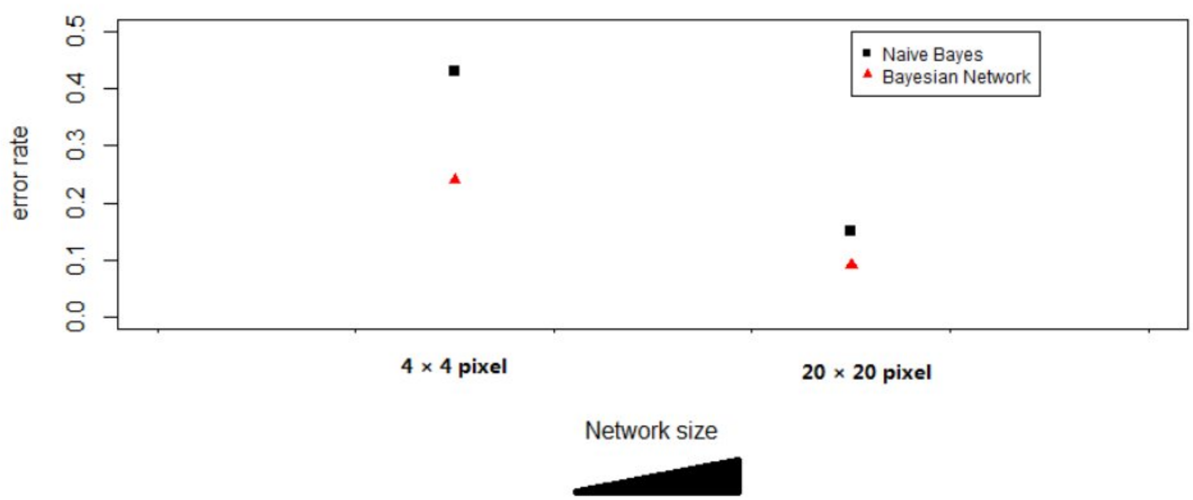

Fig 2 Comparison of error rate over (1) different network size and (2) different classifier

From Fig 2, we can see that larger network size would yield higher accuracy. However, as a trade-off, it will also lead to a longer training time, which is what we observed in our training of BN. Our next step is to find the balance between network size and training time. And then we will move on to face detection in an arbitrary image, which is our ultimate goal.

Timeline

2/26 Refine existing network learning algorithm and compare with existing algorithms
3/7 Write up poster and final report

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] 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
[5] Liu, C. A Bayesian discriminating features method for face detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* **2003**, *25*, 725 - 740
[6] 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
[7] Pearl, J. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. *Morgan Kaufmann,* **1988**
[8] Keogh, E. J. & Pazzani, M. J. Learning Augmented Bayesian Classifiers: A Comparison of Distribution-based and Classification-based Approaches, **1999**