An Unsupervised Boosting Learning Algorithm for Finite Mixture Model-based Image Segmentation

Yu lin-Sen
Department of Computer Science and Technology, Harbin Institute of Technology. P.R.China
yulinsen@sohu.com

Zhang Tian-Wen
Department of Computer Science and Technology, Harbin Institute of Technology. P.R.China

Abstract

Finite normal mixture model-based image segmentation techniques can produce robust segmentation results. But the EM algorithm used for learning mixture parameters is very sensitive to initialization. Estimating the number of components and the optimal model parameters inevitably brings a heavy computation burden. In the view of boosting learning, the paper gives a weighted EM algorithm, which is a simple extension to traditional EM algorithm. Using the parameters estimated by the standard EM algorithm as feedback, a resampling-like technique can focus the EM algorithm on those samples which are incorrectly clustered. The experiment results show that the image segmentation results are insensitive to initialization and the number of classes to a certain extent.

1. Introduction

Image segmentation is now becoming one challenging problem in the field of Content-Based Image Retrieval (CBIR). Among many methods, clustering algorithms based on statistics of pixels can produce robust segmentation results. Perhaps the cleanest approach to segmenting points in feature space is based on mixture models in which one assumes the data were generated by multiple processes and estimates the parameters of the processes and the number of components in the mixture. But the frequently used EM algorithm often converges to a local maximum that depends on the initial conditions. A feasible way for solving this problem is to choose several sets of initial values, then proceed respectively with the EM algorithms, and finally choose the best outcome set as the estimation[2]. The selection of the initial parameter value is still an open question that was studied many times[5]. In addition, estimating the number of mixture components, which is also called model selection, is important and could have a significant effect on the quality of segmentation. Figueiredo and Jain combine model selection with parameter optimization[6], their algorithm start with a large number of mixture components and successively annihilate components with small mixing weights. As compared, the greedy EM[3] starts with optimal one-component mixture and searches the optimal component to insert. A problem of these methods, when applied to image segmentation, is the high-computational complexity involved in searching an optimum result. Although they can acquire more satisfied segmentation results, they cannot meet the real time demand of CBIR.

Under the compromise of speed and performance, the paper remains the number of mixture components unchanged. From the view of boosting learning, we introduce a weighted EM algorithm, which adopts a resampling-like technique to perform the next round EM algorithm after the convergence of the usual EM algorithm. Doing so can focus the algorithm on the some samples which has the worst estimate of the local density. The proposed method remains the simplicity of the usual EM algorithm, and is insensitive to initial conditions and cluster number to some extent.

In section 2,we review finite mixture models and the EM algorithm. In section 3,we present our boosting EM algorithm. The experiment results are presented in section 4. The paper's conclusions are summarized in section 4.

2. Finite mixture models and the EM algorithm

Consider a mixture model with M >1 components in R^n for $n \geq 1$:

$$p(x \mid \Theta) = \sum_{m=1}^{M} \pi_m p(x \mid \theta_m), \quad \forall x \in R^n \qquad (1)$$

where $\pi_m \in (0,1)$ ($\forall m = 1,2,\ldots,M$) are the mixing proportions subject to $\sum_{m=1}^{M} \pi_m = 1$, $\Theta = (\pi_1, \pi_2, \ldots, \pi_M, \theta_1, \theta_2, \ldots, \theta_M)$ denotes the parameter vector. For the Gaussian mixtures, each component density $p(x \mid \theta_m)$ is a normal probability

distribution:

$$p(x \mid \theta_m) = \frac{1}{(2\pi)^{1/2} \det(\Sigma_m)^{1/2}} \times \exp\left\{-\frac{1}{2}(x-\mu_m)^{\mathrm{T}} \Sigma_m^{-1}(x-\mu_m)\right\} \quad (2)$$

$\theta_m = (\mu_m, \Sigma_m)$ is component parameter.

The usual choice for obtaining ML or MAP estimates of the mixture parameters is the EM algorithm. The EM algorithm consists of an E-step and M-step. Given a set of samples $\chi = (x_1, x_2, \ldots, x_K)$, suppose that $\Theta^{(t)}$ denotes the estimation of Θ obtained after the t th iteration of the algorithm. Then at the $(t+1)$th iteration, the E-step computes the expected complete data $\log$-likelihood function

$$Q(\Theta \mid \Theta^{(t)}) = \sum_{k=1}^{K}\sum_{m=1}^{M}\{\log \pi_m \, p(x_k \mid \theta_m)\} p(m \mid x_k; \Theta^{(t)}), \quad (3)$$

where $p(x_k \mid \Theta_m^{(t)})$ is a posterior probability

$$p(m \mid x_k; \Theta^{(t)}) = \frac{\pi_m^{(t)} p(x_k \mid \theta_m^{(t)})}{\sum_{l=1}^{M} \pi_l^{(t)} p(x_k \mid \theta_l^{(t)})} \quad (4)$$

and the M-step finds the $(t+1)$th estimation $\Theta^{(t+1)}$ of Θ by maximizing $Q(\Theta \mid \Theta^{(t)})$

$$\pi_m^{(t+1)} = \frac{1}{K}\sum_{k=1}^{K} p(m \mid x_k; \Theta^{(t)}), \quad (5)$$

$$\mu_m^{(t+1)} = \frac{\sum_{k=1}^{K} x_k P(m \mid x_k; \Theta^{(t)})}{\sum_{k=1}^{K} P(m \mid x_k; \Theta^{(t)})}, \quad (6)$$

$$\Sigma_m^{(t+1)} = \frac{\sum_{k=1}^{K} P(m \mid x_k; \Theta^{(t)}) \{(x_k - \mu_m^{(t+1)}) \cdot (x_k - \mu_m^{(t+1)})^{\mathrm{T}}\}}{\sum_{k=1}^{K} P(m \mid x_k; \Theta^{(t)})} \quad (7)$$

3. Boosting EM algorithm

Boosting is a general method for improving the accuracy of any given learning algorithm. AdaBoost[1] places most weight on the examples most often misclassified by the preceding weak rules; this has the effect of forcing the base learner to focus its attention on the "hardest" example. Motivated by the idea, we adjust the usual EM algorithm to a weighted version. In unsupervised learning situation, we first perform the usual EM algorithm, after the convergence of the algorithm, we use the parameters estimated as feedback, and put higher weight on those samples which were incorrectly grouped. Doing so, the weighted EM algorithm can pay attention to the most informative samples. For a Gaussian mixtures model, the iterative procedure of the weighted EM algorithm is

$$\mu_m^{(t+1)} = \frac{\sum_{k=1}^{K} x_k P(m \mid x_k; \Theta^{(t)}) \cdot w(x_k)}{\sum_{k=1}^{K} P(m \mid x_k; \Theta^{(t)}) \cdot w(x_k)}, \quad (8)$$

$$\Sigma_m^{(t+1)} = \frac{\sum_{k=1}^{K} P(m \mid x_k; \Theta^{(t)}) \cdot w(x_k) \cdot \{(x_k - \mu_m^{(t+1)}) \cdot (x_k - \mu_m^{(t+1)})^{\mathrm{T}}\}}{\sum_{k=1}^{K} P(m \mid x_k; \Theta^{(t)}) \cdot w(x_k)} \quad (9)$$

where $w(x_k)$ denotes the weight of the sample x_k. But the iterative procedure of mixture proportion is remained as the usual EM:

$$\pi_m^{(t+1)} = \frac{1}{K}\sum_{k=1}^{K} p(m \mid x_k; \Theta^{(t)}). \quad (10)$$

So the mixture proportions are inconsistent with the real volume of the mixture components. The inconsistency compels the weighted mixture components to shrink or to extend in a more reasonable way.

How to weight the samples is the crucial step of the weighted EM algorithm. For the present, we adopt a heuristic way to choose samples. We first perform the usual EM algorithm, then distant between the components can be acquired. If there is a component whose distribution is close to all other component, we believe the samples in this component are incorrectly grouped. So we should give more weight to these samples. Distant d between two component m and l is defined as

$$d(m,l) = (\mu_l - \mu_m)^{\mathrm{T}} \cdot \Sigma_m^{-1} \cdot (\mu_l - \mu_m), \quad (11)$$

d is not a metric. Find component m for which the distant to all other component is minimum

$$m = \arg\min_{m'} \sum_{l=1, l\neq m'}^{M} d(m', l). \quad (12)$$

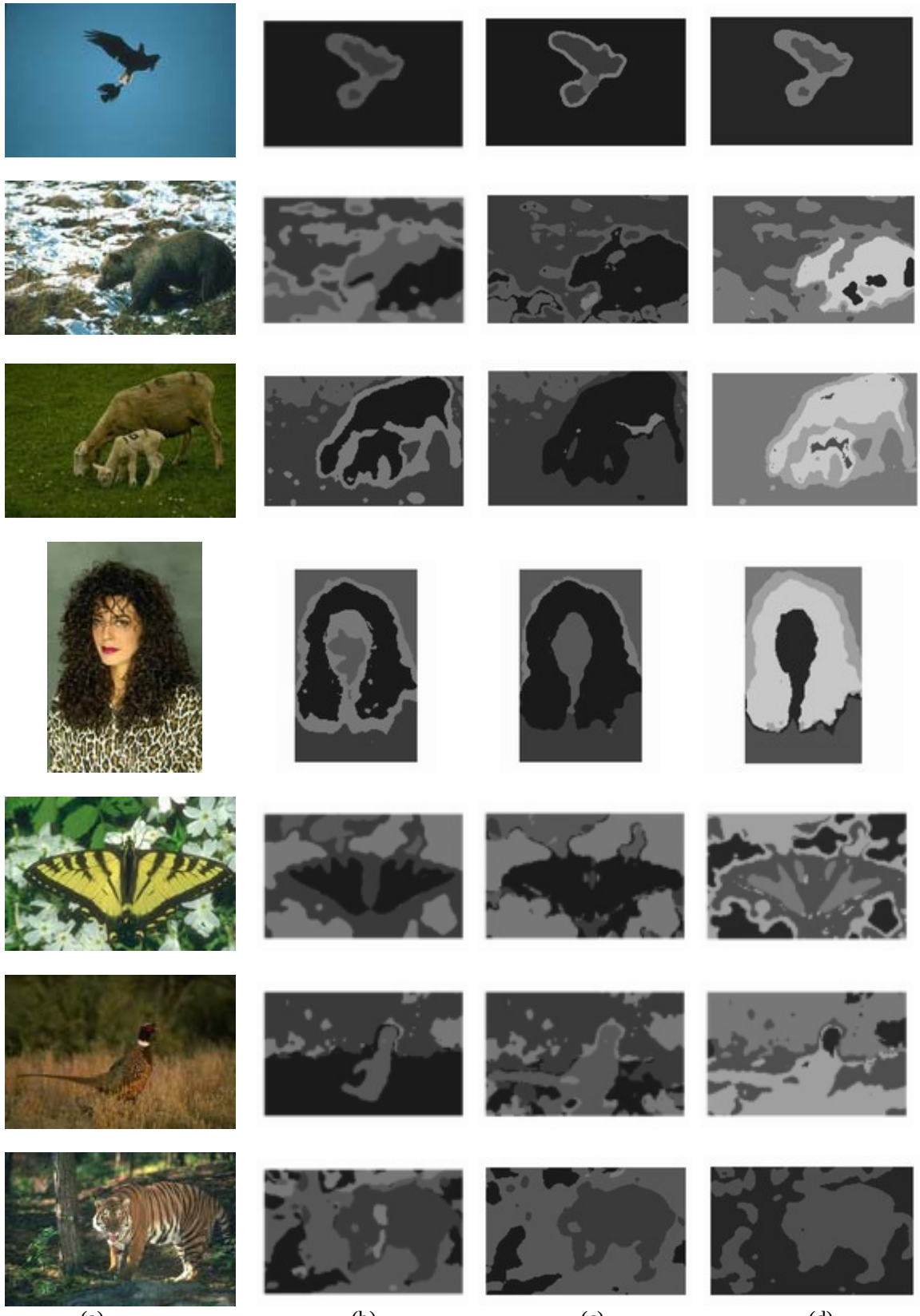
Assign weight to sample x_k according to the posterior probability $p(m \mid x_k; \Theta)$

$$w(x_k) = p(m \mid x_k; \Theta) \cdot W, \quad (13)$$

where W is a constant.

4. Experiment results

We start by mapping each pixel in the original image to a 6-dimensional feature vector, which consists of the same texture and color features used in Blobword[2].



(a) (b) (c) (d)

fig.1 image segmentation results: (a)original image; (b)EM initialized by k-means;(c)boosting EM;(d)Blobworld

First, the image $I(x,y)$ is convolved with Gaussian smoothing kernels of several scales $\sigma: M_\sigma(x,y) = G_\sigma(x,y) * (\nabla I(x,y))(\nabla I(x,y))^{\mathrm{T}}$ .Then compute the polarity at each pixel location: $P_\sigma = |E_+ - E_-| / (E_+ - E_-)$,where E_+ and E_- represent the number of gradient vectors in the window $G_\sigma(x,y)$ that are on the positive and negative sides of the dominant orientation respectively. For each pixel, an optimal scale σ^* is selected. The three texture features are p_{σ^*}, anisotropy $\alpha = 1 - \lambda_2/\lambda_1$, where λ_1 and λ_2 are the eigenvalues of $M_{\sigma^*}(x,y)$, and normalized texture contrast: $c = 2\sqrt{\lambda_1 + \lambda_2}$. The three color features are the $L^*a^*b^*$ coordinates of the color image computed after smoothing the image with a Gaussian kernel at the selected optimal scale.

We condense the original 6-dimensional feature space to a 3-dimensional one by applying PCA, which can avoid covariance matrices becoming singular or near-singular. Although dimensionality reduction via PCA lose some original cluster structure, we find, in our experiment, 3 dimensional subspace can capture at least 95% of the data variance on the average. Without model selection, the number of mixture components is fixed a priori. For our experiments on the unsupervised image segmentation here, the number of regions M=4 is a reasonable choice[4].we use the k-means method as the initialization technique to perform the usual EM. After the convergence of the usual EM, we perform the weighted EM based on the result of the usual EM. Fig. 1 (a) show the original image. The usual EM result using K-mean initialization are shown in fig. 1(b). Fig. 1(c) show our boosting result based on the result of fig.1(b). We also give the segmentation result of Blobworld[2] in fig. 1(d). But unlike [2], as a comparison, here we use the compressed features, and the position features are rejected. In fig.1 (b), (c) and (d), different gray levels correspond to different segmentation regions.

5. Conclusions

This paper presents a modified EM algorithm using unsupervised boosting and is applied to image segmentation for CBIR. The proposed boosting method has the advantages that it is insensitive to initial choice of parameter vector and chance of oversegmentation due to larger number of initial parameters is reduced. Secondly, conventional EM may be trapped in local optimum, whereas the proposed method, though does not guarantee to attain global optimum, has a possibility to give better result. The experimental result also conform the claim. Further work will focus on finding a more reasonable way to weight samples.

References

[1] Freund, Y. and Schapire, R. A decision-theoretic generalization of on-line learning and application to boosting. *Journal of Computer and System Sciences*, 55(1):119−139.1997.

[2] Chad Carson, Serge Belongie, Hayit Greenspan and Jitendra Malik, Blobworld: Color- and Texture-Based Image Segmentation Using EM and Its Application to Image Querying and Classification , *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(8): 1026-1038, August 2002.

[3] J.J. Verbeek N. Vlassis B. KrÄose. Efficient Greedy Learning of Gaussian Mixture Models, *Neural Computation*, 5(2): 469-485, Feb 2003.

[4] J. Chen, T. N. Pappas, A. Mojsilovic, and B. E. Rogowitz, Image segmentation by spatially adaptive color and texture features, *Proc. Int. Conf. Image Processing*, vol. 3, pp. 777--780, Sept. 2003.

[5] Ueda, N., Nakano, R., Ghahramani, Z., and Hinton, G. E. SMEM algorithm for mixture models. *Neural Computation*,12:2109-2128. 2000.

[6] Figueiredo, M. A. T. and Jain, A. K. Unsupervised learning of finite mixture models.*IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):381-396.2002.