Proc. VIIth Digital Image Computing: Techniques and Applications, Sun C., Talbot H., Ourselin S. and Adriaansen T. (Eds.), 10-12 Dec. 2003, Sydney

Determining the Number of Clusters in a Mixture by Iterative Model Space Refinement - with Application to Free-swimming Fish Detection

Fiona H. Evans[1,2], Michael D. Alder[2], and Christopher J. S. deSilva[3]

[1] CSIRO Mathematical and Information Sciences,
Private Bag No. 5, PO Wembley 6913, Western Australia
Fiona.Evans@csiro.au
http://www.cmis.csiro.au/Fiona.Evans

[2] Department of Mathematics and Statistics,
University of Western Australia,
35 Stirling Highway, Crawley 6009, Western Australia
mike@maths.uwa.edu.au

[3] Australian Research Centre for Medical Engineering,
Murdoch University,
South St, Murdoch 6150, Western Australia
C.deSilva@murdoch.edu.au

Abstract. We present a clustering algorithm for use when the number of clusters is unknown. We first show that the EM algorithm for mixture modeling can be considered as an alternating minimization between the data space and the model space. We then show how data cleaning can be performed by alternating between the data space and two model spaces. Finally, we develop a mixture model approach that iteratively refines the model spaces, beginning with a coarse model and selecting finer models as indicated by the consistent Akaike information criterion.

1 Introduction

Image segmentation has been long considered as an unsupervised learning or clustering problem (see [3] for a comprehensive summary). However, few studies have used automated techniques for the segmentation of underwater images. Model-based clustering or mixture modeling [7, 9] assumes that the data derive from a mixture of probability measures or distributions, each of which corresponds to a different cluster. The EM algorithm for Gaussian mixture modeling has been shown to perform well when (i) the number of clusters is known in advance and (ii) the initialization is close to the true parameter values. However, determining the number of clusters and providing a good initialization are two problems that limit its application.

We approach the problem of mixture modeling as an alternation between the data space and, in the case where the number of components is known, the model space. In the case where the number of components is not known, we alternate between a hierarchy of data and model spaces. Using iterative refinement, we provide an algorithm that simultaneously finds the number of clusters and provides good initializations for the EM algorithm for refinement.

We apply our algorithm to the segmentation of video images of southern bluefin tuna. The Australian fishery operates by holding wild-caught fish for fattening after capture. Fish are held temporarily in towing cages and then released into *grow-out* cages where they are fed on a diet of baitfish and then harvested between three and eight months later. Given their high value, farmers are reluctant to cause the fish stress by removing them from the water. Therefore, monitoring is performed on a small sample of fish from each tow cage by taking length and weight measurements. An underwater video attached to the side of the gate is used to manually count the fish as they are transferred between the cages. We aim to automate the process and improve its accuracy by developing an algorithm that will use the video to detect and count the fish.

Image data for the project are being provided by Dr. Euan Harvey (Marine Biology Group, Department of Botany, University of Western Australia). Figure 1 shows typical frames collected by the underwater camera. Whilst most segmentation applications cluster in the RGB color space or some derived feature space, in order to visually assess our results, we apply our algorithm to the thresholded image data, using the X and Y pixel locations as input data. We model the image data as a mixture, with each individual fish modeled by a Gaussian distribution.

Fig. 1. Typical frames with intensity increasing from black to white.

2 Mixture modeling by data space - model space alternation

We define the mixture modeling problem as follows. We have an observed data set $Y = \{y_i : i = 1, ..., N\}$, where each $y_i \in {}^n$, and each y_i belongs to one of K clusters. We assume that each of the K mixture components has a model with known parametric form and densities . The probability density function for the mixture is then assumed to be a weighted sum of the component densities:

$$p\left(y|\,\theta\right) = \sum_{j=1}^{K} \alpha_j p_j \left(y|\theta_j\right)$$

where the weights, α_j sum to one:

The model parameters consist of the parameters of each of the component models and the K mixture weights $\theta = \left(\theta_1, ..., \theta_K, \alpha_1, ..., \alpha_K\right)^T$, where each θ_j may itself be a real-valued vector. We call the space of all possible parametrizations the model space,

$$\Theta = \left\{\theta = \left(\theta_1, ..., \theta_K, \alpha_1, ..., \alpha_K\right)^T\right\}$$

We apply the EM algorithm for finding the maximum likelihood estimate for incomplete data [5] as follows. For each y_i we define a K-dimensional vector, $w_i = \left(w_{i1}, ..., w_{iK}\right)^T$, whose elements sum to one. This vector contains the probabilities of the datum y_i belonging to each of the K clusters. We cannot observe the w_i directly; they are hidden. Thus, we define the hidden data space to be

$$\mathcal{W} = \left\{ w = \left(w_1, ...w_K\right)^T : \sum_{j=1}^{K} w_j = 1 \right\}$$

the (K-1)-dimensional Simplex; and the complete data space, $\mathcal{Z}$, to be the collection of sets:

$$\mathcal{Z} = Y \times \mathcal{W} = \{z_i = (y_i, w_i) : y_i \in Y, w_i \in \mathcal{W}\}_{i=1}^{N}$$

The EM algorithm is an iterative procedure, arbitrarily begun, where at iteration k, given an estimated model, $\theta^{(k)}$, we perform two steps.

The Expectation (E) step uses the model, $\theta^{(k)}$, and the observed data, $Y = \{y_i : i = 1, ..., N\}$, to find the weight vectors, $\left\{w_i^{(k+1)} : i = 1, ..., N\right\}$:

$$w_{il}^{(k+1)} = \frac{\alpha_l p\left(y_i|\theta_l^{(k)}\right)}{\sum_{j=1}^{K} \alpha_i p\left(y_i|\theta_j^{(k)}\right)}$$

This enables us to form the completed data $\left\{z_i^{(k+1)} = \left(y_i, w_i^{(k+1)}\right) : i = 1, ..., N\right\}$ thus finding a point in the data space, $Z^{(k+1)} \in \mathcal{Z}$.

The Maximization (M) step maximizes the normalized log-likelihood function using the completed data $Z^{(k+1)}$:

$$NLL\left(\theta\right) = \frac{1}{N}\sum_{i=1}^{N} \log p\left(y_i|\theta\right) = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K} \alpha_j p_j\left(y_i|\,\theta_j\right)$$

to get the maximum likelihood estimate

$$\theta^{(k+1)} = \arg\max_{\theta} NLL\left(\theta\right)$$

Thus, finding a point in the model space $\theta^{(k+1)} \in \Theta$.

The maximum likelihood estimate for the weights is simply

$$\alpha_j^{(k+1)} = \frac{n_j^{(k+1)}}{N} \qquad \text{where} \qquad n_j^{(k+1)} = \sum_{i=1}^{N} w_{ij}^{(k+1)}.$$

Since we are using Gaussian mixtures, we can also explicitly solve for the mean and covariance matrix of each component as follows:

$$\mu_j^{(k+1)} = \frac{\sum\limits_{i=1}^{N} z_{ij} y_i}{n_k^{(k+1)}}$$

$$\Sigma_j^{(k+1)} = \frac{\sum\limits_{i=1}^{N} z_j \left(y_i^{(k+1)} - \mu_j^{(k)}\right)^T \left(y_i^{(k+1)} - \mu_j^{(k)}\right)}{n_k^{(k+1)}}$$

Figure 2 shows the geometry of the EM algorithm considered as an alternation between the data space and model space.

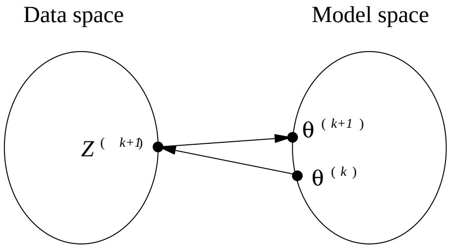

Fig. 2. The EM algorithm for mixture modeling as a data space - model space alternation.

3 Alternating minimization

Having shown that the EM algorithm alternates between points in the (complete) data space and the model space, we can also show that the EM algorithm minimizes a non-symmetric distance between the two spaces [4]. We first observe that for any observed data set $X = \{x_i : i = 1, ..., N\}$, we can define a unique singular measure, the empirical measure, corresponding to X as follows. For any open subset $u \subset {}^{n}$,

$$\delta_X(u) = \frac{k}{N}$$

where k is the number of $x_i \in X$ contained in u. For any $x_i \in X$, we have that the value of the measure at the singleton set containing only x_i is

$$\delta_X(\{x_i\}) = \frac{1}{N}$$

For convenience, we write $\delta_X(x_i)$ for $\delta_X(\{x_i\})$ from hereon.

For any model θ we define the data-model divergence to be

$$D(X \parallel \theta) = \sum_{i=1}^{N} \delta_X(x_i) \log \frac{\delta_X(x_i)}{p(x_i|\theta)}$$

We can think of this as the divergence of the data from the given model. We note that if we constrain θ to be defined only over the data X (and equal to zero for any $x \notin X$), then $D(X \parallel \theta)$ is the Kullback-Leibler information divergence, also known as the cross-entropy [11].

We can easily show that the M-step of the EM algorithm (maximizing the likelihood) is equivalent to minimizing the divergence as follows. We have

$$\begin{aligned} D(X \parallel \theta) &= \sum_{i=1}^{N} \delta_X(x_i) \log \frac{\delta_X(x_i)}{p(x_i|\theta)} \\ &= \sum_{i=1}^{N} \delta_X(x_i) \log \delta_X(x_i) - \sum_{i=1}^{N} \delta_X(x_i) \log p(x_i|\theta) \\ &= \sum_{i=1}^{N} \delta_X(x_i) \log \delta_X(x_i) - \frac{1}{N} \sum_{i=1}^{N} \log p(x_i|\theta) \\ &= \mathcal{F}(X) - NLL(\theta) \end{aligned}$$

where the first term $\mathcal{F}(X)$, the entropy of the data, is constant for any given data set and the second term $NLL(\theta)$ is the normalized log-likelihood.

Csiszár, and Tusnády [4] show that the E-step is also equivalent to minimizing the divergence.

4 Data cleaning by model space refinement

To introduce the idea of model space refinement, we consider the problem of cleaning noisy data: detecting and removing noise from an observed data sample. For example, Figure 3 shows the thresholded data for a typical frame collected from the underwater video camera used to monitor southern bluefin tuna. The

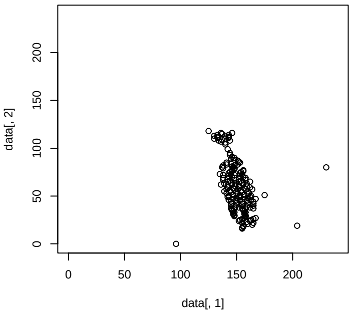

Fig. 3. Image data showing a single fish and some spatial noise.

quality of the data is poor and we can clearly see spatial noise caused by suspended particles and matter in the water. We want to remove the noise and fit a model to the fish.

We model the data by a bivariate Gaussian and the noise by a bivariate uniform distribution.. Let $X = \{x_i : i = 1, ..., N\}$ be the observed data set and let the data space be the set of all possible observed data sets. We approach the data-cleaning problem as an alternation between the data space and iteratively refined model spaces as a three-step algorithm:

1. Find the maximum likelihood estimate, θ_u, for the uniform parameters using the observed data sample, and the maximum likelihood estimate, θ_g, for the Gaussian parameters using the observed data sample. This gives a point, (θ_u, θ_g) in the model space $\Theta_u \times \Theta_g$ where Θ_u is the space of uniform parameters and θ_g is the space of Gaussian parameters.
2. Remove the noise from the observed data set by deleting the data more likely to have been generated by the uniform distribution. This gives a new point, X^* , in the data space.
3. Find the maximum likelihood estimate for the Gaussian parameters using the new data set. This corresponds to finding a point θ_g^* in the model space, Θ_g.

We note that steps 1 and 3 are maximizing the likelihood and thus performing the M-step of the EM algorithm. Step 2 is implicitly finding the EM weights vectors for each datum, $w_i = (w_{ig}, w_{iu})$, thus performing the E-step of the EM algorithm. We therefore have an alternating minimization algorithm for data cleaning.

Figure 4 shows the geometry of the data cleaning problem considered as an alternation between the data space and model spaces. There is clearly a projection between the two data spaces: $\pi_2 : \Theta_u \times \Theta_g \to \Theta_g$.

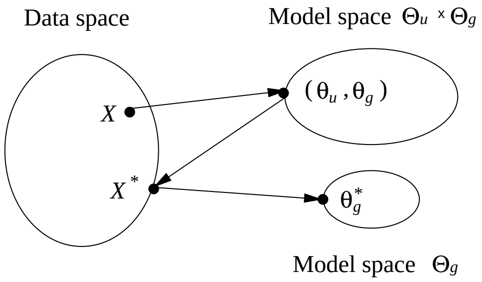

Fig. 4. Data cleaning as an alternation between the data space and model spaces.

5 How many clusters?

Various criteria have been developed for comparing mixtures with different numbers of components. These include Akaike's information criterion (AIC) [1], Bozdogan's consistent AIC [2], the Bayesian information criterion [10] and the minimum message length criterion [12]. In most applications, the EM algorithm is randomly initialized for $K = 1, ..., K_{\max}$, where $K_{\max}$ is some pre-specified maximum number of components. The algorithm is run for each K, and the criteria evaluated. The model that achieves the minimum value is then selected. The major limitation of this approach is the problem of initialization.

In an effort to avoid poor initializations, Figueirdo and Jain [6] incorporate the MML criteria into the EM algorithm. The algorithm is initialized with a large number of components, and components are collapsed as it iterates. Hierarchical agglomeration methods [8] also first fit a large number of clusters, successively merging clusters according to some criteria.

A problem with these techniques arises with the selection of the maximum number of clusters. If it is too small, the model may be too coarse for the data; if it is too large, the computational time may become very large. In the next section, we will present an algorithm that initially models the data using a coarse model, splitting clusters until our criteria is minimized. This allows us to minimize the amount of computation required. It also suggests a logical way of initializing the EM algorithm at each split based upon the previous level in the hierarchy.

6 Mixture modeling by iterative model space refinement

We now apply the data space - model space alternation approach to the problem of mixture modeling where the number of components is not known in advance. We first clean the data using the approach from Section 4, and we therefore have an initial Gaussian component with parameters $g^{(1)}$. We can think of this as a mixture model, where the number of components is equal to one; that is, a 1-dimensional mixture, $\theta^{(1)}$.

We test whether the component can be split by considering two potential models: the original model, $g^{(1)}$, and a two-component mixture that results from using the EM algorithm (10 iterations) with the initializations derived from $g^{(1)}$ as follows. We decompose $g^{(1)}$ into its mean and covariance matrix. We perform an eigenanalysis on the covariance matrix and initialize the two components at either end of the larger eigenvector from the mean, at a distance of two standard deviations. We retain the model that achieves the minimum value of the consistent AIC [2].

Figure 5 shows how the iterative model refinement algorithm alternates between the data space and the higher-dimensional model spaces as they are iteratively refined. The model hierarchy is shown in tree form, to indicate which components have been split. In this example, we split the original 1-dimensional mixture, $\theta^{(1)}$ into two components, so that we have a 2-dimensional mixture, $\theta^{(2)}$, with components $g^{(2)}$ and $g^{(3)}$. We then test whether the components $g^{(2)}$ and $g^{(3)}$ can be split, by applying the same procedure. The algorithm is iterated until all components have been tested.

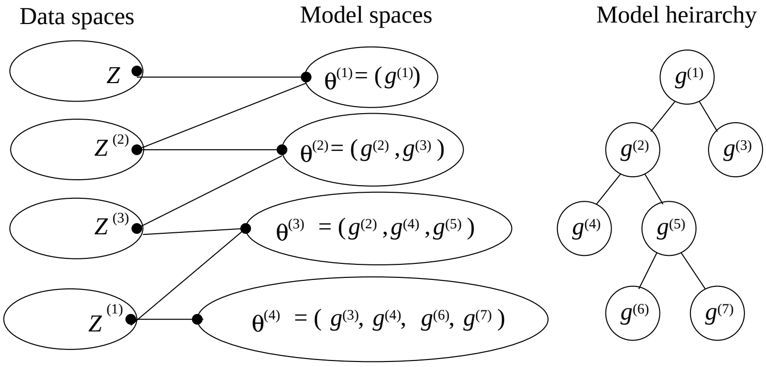
Data spaces
Model spaces
Model heirarchy
Z
$\theta^{(1)} = (g^{(1)})$
$Z^{(2)}$
$\theta^{(2)} = (g^{(2)}, g^{(3)})$
$Z^{(3)}$
$\theta^{(3)} = (g^{(2)}, g^{(4)}, g^{(5)})$
$Z^{(1)}$
$\theta^{(4)} = (g^{(3)}, g^{(4)}, g^{(6)}, g^{(7)})$
$g^{(1)}$
$g^{(2)}$
$g^{(3)}$
$g^{(4)}$
$g^{(5)}$
$g^{(6)}$
$g^{(7)}$

Fig. 5. Iterative model space refinement.

7 Results and conclusions

We tested the algorithm on eighty frames of video. The results are shown in Table 1. For frames with less than three fish, the algorithm worked well - sometimes finding two clusters for a single fish. This is shown in Figure 6 (three successive frames containing two fish). In two of the frames, the algorithm has over-segmented the data, finding both the head and tail of the second fish. On closer examination of the algorithm (see Figure 7), we see that this problem occurs when the initialization for the two-component model leads to the EM algorithm converging to a suboptimal local minima of the divergence.

For frames with three or more fish, the algorithm performed poorly. In most cases, the fish were over-segmented, as seen in Figure 6. However, in some cases,

Table 1. Results for free-swimming fish detection.

	Number of fish detected					
Number of fish	**0**	**1**	**2**	**3**	**4**	**≥ 5**
0	6					
1		22				
2		1	24	12	1	
3		3	2			1
4					2	4

the initializations caused images containing three fish to cluster into only a single fish. This occurred when the initial split into two clusters failed to converge because the *real* clusters were located along the minor rather than the major axis of the original Gaussian.

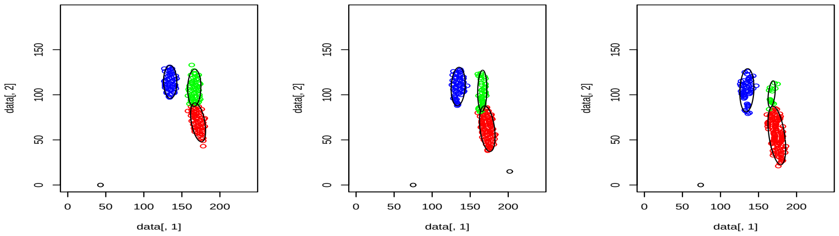

Fig. 6. Successive frames (31, 32 and 33) with two fish.

We conclude that whilst the iterative refinement algorithm seems to provide good clustering results when the clusters are distinct, it has failed to adequately detect more than two free-swimming fish in an image. The main problem occurs because the Gaussian distributions do not model the shapes of tuna sufficiently well. Further adaptations to the algorithm may be worth investigating. For instance, we could use the facts that the tuna have an elliptical rather than round shape and are all moving in a roughly upward direction as they are released from the tow-cage by adopting a Bayesian approach. Thus, we would define a prior distribution over the component models that lends higher probability to components with the preferred shapes and orientations. Because our initializations at each refinement step are giving poor results, we also need to consider a better way of moving from the data space to a higher-dimensional model space and thus providing better initializations.

In addition, we believe that the failure of the algorithm is likely caused by the unsuitability of the data provided to it. In order to visually assess the algorithm's performance, we have provided two-dimensional data in the form of the locations of pixels in the threshholded images. The data are thus over-simplified and do

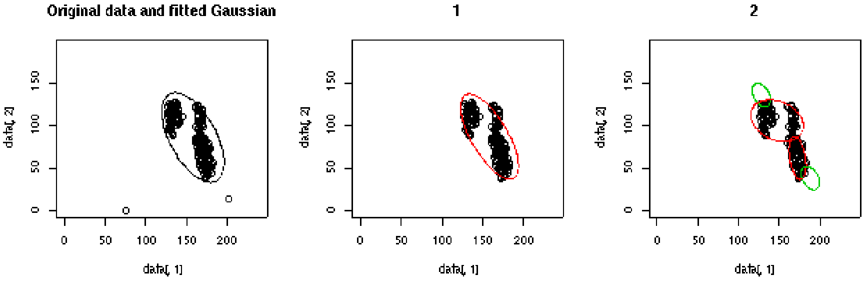

Fig. 7. Original data and two potential models for frame 32. The green ellipses show the initialization

not include colour or temporal information. We are continuing to work on this application with better feature selection.

Further work is also required on comparing the algorithm with other clustering approaches.

References

1. Akaike, H.: A new look at the statistical model identification, IEEE Transactions on Automatic Control, 19, 6 (1974) 716-723
2. Bozdogan, H.: Model selection and Akaike's information criterion (AIC): The general theory and its analytical extensions, Psychometrika, 52, 3 (1987) 345-370
3. Cheng, H.D., Jiang, X.H., Sun, Y., Wang, J.: Color image segmentation: advances and prospects, Pattern Recognition, 34, 12 (2001) 2259-2281
4. Csiszár, I. and Tusnády, G.: Information geometry and alternating minimization procedures. Statistics and Decisiosn, Supplement Issue No. 1 (1984) 205-237
5. Dempster, A.P., Laird, N.M., and Rubin, D.A.: Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society, Series B, 39 (1977) 1-38
6. Figueiredo, M.A.T and Jain, A.K.: Unsupervised learning of finite mixture models, IEEE Transactions on Pattern Analysis and Machine Learning, 24, 3 (2002) 381-396
7. Fraley, C. and Raftery, A.E.: How many clusters? Which clustering method? Answers via model-based cluster analysis. Technical Report 329, Department of Statistics, University of Washington (1998)
8. Fraley, C. and Raftery, A.E.: Model-based clustering, discriminant analysis and density estimation, University of Washington, USA (2000)
9. McLachlan, G.J., Peel, D., and Prado, P.: Clustering via normal mixture models, Proceedings of the American Statistical Society (Bayesian Statistical Science Section). (1997) 98-103
10. Schwarz, G.: Estimating the dimension of a model, Annals of Statistics, 6 (1978) 461-464
11. Shannon, C.E. and Weaver, W.: A mathematical theory of communication. Urbaba: The University of Illinois Press (1949)
12. Wallace, C. and Freeman, P.: Estimation and inference via compact coding, The Computer Journal, 42, 4 (1987) 241-252