Addressing Overfitting in Mixtures of Factor Analyzers

Phillip Shreeves[*] Jeffrey Andrews[†]

Abstract

A commonly used model for unsupervised machine learning, or cluster analysis, relies on the fitting of finite mixture distributions to data, and the most common method for estimating these models is the expectation-maximization (EM) algorithm. Unfortunately, this algorithm tends to experience issues related to overfitting, as well as the more commonly known problem of converging to a local maxima. Mixtures of factor analyzers allow for a factor analysis structure, thus implicitly reducing the dimensionality in the model. Unfortunately, these factor analyzers do not solve the problems with the EM stated above. In order to tackle said issues, we use an algorithm that combines the regular EM with the non-parametric bootstrap, and show its promise for addressing the problems discussed above on both real and simulated data.

1. Introduction

Finite mixture models (McLachlan and Peel, 2004) have become a prominent feature within the area of cluster analysis in statistics. A random vector $\mathbf{X}$ arises from a parametric finite mixture model if it follows a probability density function in the form of $f(\mathbf{x}) = \sum_{g=1}^{G} \pi_g f_g(\mathbf{x} \mid \boldsymbol{\vartheta_g})$ where π_g are the mixing proportions such that $\sum_{g=1}^{G} \pi_g = 1$ and $f_g(\mathbf{x} \mid \boldsymbol{\vartheta_g})$ is the density function of group g with $\boldsymbol{\vartheta_g}$ parameters. One very common approach for parameter estimation in mixture modelling is the use of the expectation maximization (EM) algorithm (McLachlan and Krishnan, 2008), which unfortunately suffers from issues to do with degeneracy (Ingrassia and Rocci, 2007, 2011), convergence to local maxima (Titterington et al., 1985; McLachlan and Krishnan, 2008) and overfitting (Andrews, 2018).

2. Background

2.1 Mixture Models and the EM Algorithm

Herein we focus on multivariate Gaussian distributions, taking the form

$$f(\mathbf{x}) = \sum_{g=1}^{G} \pi_g \phi(\mathbf{x} \mid \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$$

where $\boldsymbol{\mu}_g$ and $\boldsymbol{\Sigma}_g$ represent the mean and covariance matrix of the respective group g. To facilitate using mixture models for clustering, we introduce the missing cluster membership indicator variables. Specifically, these matrices are represented in the form of a Z_{ig} matrix, where given the i^{th} observation and the g^{th} group, we can find the conditional expectation of observation i belonging to group g:

$$E[Z_{ig} \mid \mathbf{x}_i, \vartheta] = \frac{\pi_g \phi_g(\mathbf{x}_i \mid \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)}{\sum_{j=1}^{G} \pi_j \phi_j(\mathbf{x}_i \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)},$$

where $\vartheta = (\pi_1, ..., \pi_g, \mu_1, ..., \mu_g, \boldsymbol{\Sigma}_1, ..., \boldsymbol{\Sigma}_g)$ is the entire parameter space.

[*] University of British Columbia, Okanagan Campus, 1177 Research Rd, Kelowna, British Columbia, V1V 1V7, Canada

[†] University of British Columbia, Okanagan Campus, 1177 Research Rd, Kelowna, British Columbia, V1V 1V7, Canada

The expectation maximization (EM) algorithm carries out parameter estimation through two specific steps: the expectation (E) step and the maximization (M) step. The E-step is used to calculate the conditional expectation of the missing data, given the observed data, with the assumption that the estimated parameters are valid. The M-step updates parameters by computing the maximum likelihood estimates (MLEs) from the complete-data log-likelihood's expected value. This algorithm then cycles through these two steps until a stable solution is met. Convergence is determined via "lack of progress" which occurs when the difference of the last two calculated log likelihoods is smaller than some small, non-negative ϵ value.

2.2 Mixtures of Factor Analyzers

When performing the EM algorithm with respect to Gaussian mixture models, calculation of the covariance matrices becomes increasingly difficult when larger datasets are introduced. The general Gaussian mixture model EM has a total of $Gp(p+1)/2$ free parameters requiring estimation for the group covariance matrices. When p (the number of variables in the dataset) begins to become rather large, the number of parameters needed to estimate these matrices start to increase rather quickly. To reduce dimensionality, these group covariance matrices can be decomposed by assuming a mixtures of factor analyzers model (McLachlan and Peel, 2001) to give a structure of the form

$$\mathbf{\Sigma}_g \approx \mathbf{\Lambda}_g \mathbf{\Lambda}_g' + \mathbf{\Psi}_g \tag{1}$$

where $\mathbf{\Lambda}_g$ is a $p \times q$ loading matrix (where q is the number of factors) with $p << q$ and $\mathbf{\Psi}_g$ is a diagonal noise matrix. These covariance structures can have as few as $pq - q(q-1)/2 + p$ parameters and as many as $G[pq - q(q-1)/2 + p]$ parameters; either way providing a much smaller growth in the number of parameters with respect to the dimensionality p.

The factor analyzer structure requires several changes from the standard estimation scheme for mixtures of multivariate Gaussians via the EM. In particular, the factor analysis model introduces a new latent variable $\mathbf{u}$, providing two sources of missing data. A variant of the EM algorithm, the alternating expectation conditional maximization (AECM) algorithm is thus used for parameter estimation. In the interest of keeping this proceedings short, the reader is encouraged to review McLachlan and Peel (2001) and McLachlan and Krishnan (2008) for further insights on these matters.

2.3 Bootstrapping and the BootEM Algorithm

The process of bootstrapping (Efron, 1981, 1982) is a resampling-with-replacement scheme that is primarily used to facilitate finding standard errors for estimators. It has also been used to improve the predictive power of models through the process of bootstrap aggregation, or 'bagging' (Breiman, 1996). Importantly, some researchers have investigated the usage of the bootstrap within optimization procedures (Tibshirani and Knight, 1999; Wood, 2001).

In related and recent work, Andrews (2018) introduced a bootstrap-augmented EM-style (BootEM) algorithm specifically for performing model-based clustering with mixtures of multivariate Gaussians. Therein, Andrews shows that BootEM can address issues related to overfitting, as well as convergence to local maxima. However, among some of the drawbacks of the BootEM is that the resampling technique results in a requirement of larger sample sizes with respect to the dimensionality of the data. In other words, even data sets of moderate dimensionality need to be reduced prior to performing clustering. This fact severely reduces the practicality of the BootEM algorithm for most real data sets. We consider the research in this manuscript to primarily address this drawback.

3. The BootAECM Algorithm

With $\mathbf{x}$ being the observed data, $\mathbf{z}$ being the missing component indicators, $\mathbf{u}$ being the missing latent variables, and the parameter space $\vartheta = (\pi_1, ..., \pi_g, \boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_g, \boldsymbol{\Lambda}_1, ..., \boldsymbol{\Lambda}_g, \boldsymbol{\Psi}_1, ..., \boldsymbol{\Psi}_g)$:

1. Initial cluster membership $\hat{\mathbf{z}}_0$ and initial latent factor scores $\hat{\mathbf{u}}_0$, set $j = 1$ for the bootstrap index.

2. Find bootstrap sample $\mathbf{x}^{(\mathbf{j})}$ of size n, with corresponding cluster memberships $\hat{\mathbf{z}}^{(\mathbf{j})}$ and $\hat{\mathbf{u}}^{(\mathbf{j})}$found from $\hat{\mathbf{z}}_{(\mathbf{j-1})}$ and $\hat{\mathbf{u}}_{(\mathbf{j-1})}$, respectively.

3. Enter EM algorithm for bootstrap sample j, set $k = 0$ for the within-EM index and set $\hat{\mathbf{z}}^{(\mathbf{j})}_{\mathbf{jk}} = \hat{\mathbf{z}}^{(\mathbf{j})}$ and $\hat{\mathbf{u}}^{(\mathbf{j})}_{\mathbf{jk}} = \hat{\mathbf{u}}^{(\mathbf{j})}$.

   (a) CM-step 1: Find MLEs for $\boldsymbol{\mu}_g$ and $\pi_g\ \forall\ g = 1, \ldots, G$ which maximize $l_c(\vartheta|\mathbf{x}^{(\mathbf{j})}, \hat{\mathbf{z}}^{(\mathbf{j})}_{\mathbf{k}})$. Set $k = k + 1$.

   (b) E-step: Find conditional expectations $\hat{z}_{jk}$ and $\hat{u}_{jk}$ for all observed data $\mathbf{x}$ and $\vartheta^{(\mathbf{j})}_{\mathbf{k}}$. Find $\hat{\mathbf{z}}^{(\mathbf{j})}_{\mathbf{jk}}$ and $\hat{\mathbf{u}}^{(\mathbf{j})}_{\mathbf{jk}}$ for bootstrap sample j from $\hat{\mathbf{z}}_{\mathbf{jk}}$ and $\hat{\mathbf{u}}_{\mathbf{jk}}$ respectively.

   (c) CM-step 2: Find MLEs for $\boldsymbol{\Lambda}_g$ and $\boldsymbol{\Psi}_g\ \forall\ g = 1, \ldots, G$ which maximize $l_c(\vartheta|\mathbf{x}^{(\mathbf{j})}, \hat{\mathbf{z}}^{(\mathbf{j})}_{\mathbf{k}}, \hat{\mathbf{u}}^{(\mathbf{j})}_{\mathbf{k}})$. Set $k = k + 1$.

   (d) E-step: Find conditional expectations $\hat{z}_{jk}$ and $\hat{u}_{jk}$ for all observed data $\mathbf{x}$ and $\vartheta^{(\mathbf{j})}_{\mathbf{k}}$. Find $\hat{\mathbf{z}}^{(\mathbf{j})}_{\mathbf{jk}}$ and $\hat{\mathbf{u}}^{(\mathbf{j})}_{\mathbf{jk}}$ for bootstrap sample j from $\hat{\mathbf{z}}_{\mathbf{jk}}$ and $\hat{\mathbf{u}}_{\mathbf{jk}}$ respectively.

   (e) Calculate $l(\vartheta^{(\mathbf{j})}_{\mathbf{k}}|\mathbf{x}^{(\mathbf{j})})$, check for within bootstrap convergence, if not converged return to Step 3a, else set $\hat{\mathbf{z}}_{\mathbf{j}} = \hat{\mathbf{z}}^{(\mathbf{j})}_{\mathbf{jk}}$ and $\hat{\mathbf{u}}_{\mathbf{j}} = \hat{\mathbf{u}}^{(\mathbf{j})}_{\mathbf{jk}}$ continue.

4. Calculate $l(\vartheta^{(\mathbf{j})}_{\mathbf{k}}|\mathbf{x})$, check for total algorithm stopping criterion, if not converged set $j = j + 1$ and return to Step 2.

Note that Step 3 of the BootAECM still constitutes a standard AECM algorithm, allowing us to use standard convergence criteria such as lack of progress. The log-likelihood calculated during Step 4 will not be monotonically increasing. We utilize the same stopping criteria as Andrews (2018), checking for a null result of the Durbin-Watson test for autocorrelation (Durbin and Watson, 1951) across the 500 most recent bootstrap samples.

3.1 Averaged Parameter Space

With the application of the bootstrap to the mixture model, this allows one to consider examining the model in a different way. A suggested application is the use of the averaged parameter space, in which one takes the last 500 parameters, averages them, and then reports the averaged model as the algorithm's output. In Figure 3.1, the black circles provide caution on naively averaging the parameter space from the BootAECM algorithm for mixtures of factory analyzers.

The reason we see a precipitous drop for the log-likelihood of the naively averaged parameter space for this has to do with one of the main features of the factor analysis model: namely, the non-uniqueness of the solutions for $\boldsymbol{\Lambda}_g$. Thus, while the BootAECM progresses, we may see arbitrary orthogonal rotation of the $\boldsymbol{\Lambda}_g$ occur. When this happens, any averaging across $\boldsymbol{\Lambda}_g$ resulting from different bootstrap samples will provide factor loadings without any particular relation to the data being fit — hence a drop in the model-fitting measure.

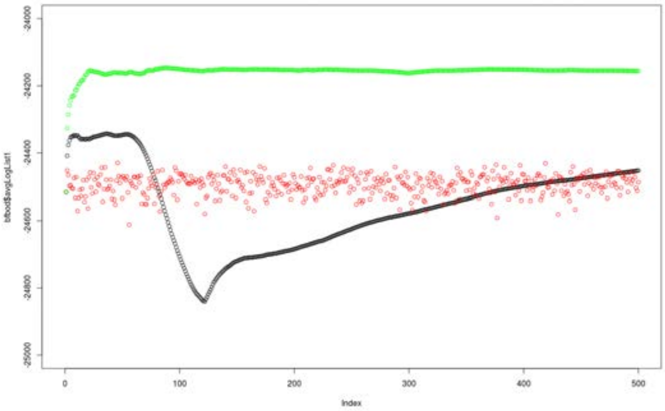

Figure 1: Consider the following plots of log likelihoods taken from a run on the body data set. In red are the log-likelihoods specified in Step 4 of the BootAECM algorithm, while in green and black are differing approaches to the averaged parameter spaces. Black represents a naive averaging of all the parameters in ϑ, where the green represents the averaged space where $\mathbf{\Lambda}_g$ and $\mathbf{\Psi}_g$ are combined to provide an estimate of $\mathbf{\Sigma}_g$.

A simple fix is to instead consider the estimates of $\mathbf{\Sigma}_g$, that is $\mathbf{\Lambda}_g\mathbf{\Lambda}_g' + \mathbf{\Psi}_g$, resulting from each bootstrap sample and average those instead. We see the loglikelihood arising from that process in green in Figure 3.1, showing a much more expected result — that is, generally increasing while the algorithm progresses.

4. Application to a Benchmark Data Set

The new BootAECM algorithm was applied to a real data set as a preliminary investigation of its performance compared to the standard AECM algorithm. We chose the 'body' data set available in the gclus library (Hurley, 2012) as a commonly used benchmark data set for clustering that also happens to have moderate dimensionality. It contains contains 23 measurements on 507 people, in addition to the recording of their age and sex. Known groups are based on the sex variable (260 females and 247 males). A classification table resulting from 50 random runs of each algorithm is shown in Table 1. We find a substantial decrease on the aggregate misclassification rate (0.26 vs 0.41) when using the proposed BootAECM algorithm instead of the standard AECM approach to model fitting.

Table 1: Results from 50 random initializations of the AECM and BootAECM algorithms for mixtures of factor analyzers

Method	Group	Gender	
		Male	Female
AECM	1	5908	3957
	2	6442	9043
BootAECM	1	8628	2759
	2	3722	10241

5. Summary

A bootstrap-augmented alternating expectation conditional-maximization algorithm with application to mixtures of factor analyzers was implemented in an attempt to address issues of overfitting and convergence to local maxima while simultaneously performing implicit dimensionality reduction. Our proposed algorithm has herein been shown to address at least the underfitting aspect by improving on the clustering results of a benchmark data set. Future work will develop a simulation framework to investigate its actual performance *vis a vis* overfitting. Additional work is expected to address issues such as model selection, alternative convergence criteria, and application to more real benchmark and/or novel clustering data sets.

References

Andrews, J. L. (2018). Addressing overfitting and underfitting in gaussian model-based clustering. *Computational Statistics & Data Analysis 127*, 160–171.

Breiman, L. (1996). Bagging predictors. *Machine learning 24*(2), 123–140.

Durbin, J. and G. S. Watson (1951). Testing for serial correlation in least squares regression. ii. *Biometrika 38*(1-2), 159–178.

Efron, B. (1981). Nonparametric estimates of standard error: the jackknife, the bootstrap and other methods. *Biometrika*, 589–599.

Efron, B. (1982). *The jackknife, the bootstrap and other resampling plans.* SIAM.

Hurley, C. (2012). *gclus: Clustering Graphics.* R package version 1.3.1.

Ingrassia, S. and R. Rocci (2007). Constrained monotone EM algorithms for finite mixture of multivariate gaussians. *Computational Statistics & Data Analysis 51*(11), 5339–5351.

Ingrassia, S. and R. Rocci (2011). Degeneracy of the EM algorithm for the MLE of multivariate gaussian mixtures and dynamic constraints. *Computational statistics & data analysis 55*(4), 1715–1725.

McLachlan, G. and T. Krishnan (2008). *The EM algorithm and extensions, Second Edition.* John Wiley & Sons.

McLachlan, G. and D. Peel (2004). *Finite mixture models.* John Wiley & Sons.

McLachlan, G. J. and D. Peel (2001). Mixtures of factor analyzers. *Journal of Computational Statistics 16.*

Tibshirani, R. and K. Knight (1999). Model search by bootstrap "bumping". *Journal of Computational and Graphical Statistics 8*(4), 671–686.

Titterington, D. M., A. F. Smith, and U. E. Makov (1985). *Statistical analysis of finite mixture distributions.* Wiley,.

Wood, S. N. (2001). Minimizing model fitting objectives that contain spurious local minima by bootstrap restarting. *Biometrics 57*(1), 240–244.