Comparing Different Clustering Models on the Unit Hypercube

Dean, Nema
University of Glasgow, School of Mathematics & Statistics
15 University Gardens,Glasgow G12 8QQ, Scotland
E-mail: Nema.Dean@glasgow.ac.uk

Nugent, Rebecca
Carnegie Mellon University, Department of Statistics
Baker Hall 132, Pittsburgh, PA 15213, U.S.A.
E-mail: rnugent@stat.cmu.edu

Introduction

Cluster analysis (also known as unsupervised learning) is the counterpart to classification (or discrimination) where the object of interest is the group structure in the data. Unlike classification, which is a supervised learning technique, in clustering no *a priori* information is available about the number (if any) or nature of groups in the data. The goal of cluster analysis is to discover this information through exploration via graphical, algorithmic or modelling techniques. Many different methods for performing cluster analysis exist, arising from Statistics, Social Science, Computer Science and other areas. Regardless of the type of method, whether or not it is explicitly stated, there are always implicit shape or space restrictions on the groups that each method finds. Selecting a particular approach will then imply that this shape of group is the one that it has been decided is most useful/appropriate for the particular setting. It has been argued by Hennig (2010) that there is no perfect cluster analysis technique for every situation and that user knowledge, along with understanding of the shape restrictions of different approaches, must be used to select the appropriate method for each different context.

While little or nothing is known about group structure in the situation (although as stated before, expert knowledge may be used to make assumptions about group shape), it is often the case that the variables/measurements that the cluster analysis will be applied to, have restrictions of their own. The most obvious one is type of measurement, usually split into continuous or categorical (often further split into ordinal/nominal). There may be limits (either physical or due to the sensitivity of recorder) to the upper and/or lower range of the variable. Obvious examples of such include variables such as weight (inherently non-negative), proportions (lying between 0 and 1), etc. Other types of more complex restriction on the shape of the space the measurements fall in include hyperspheres or simplices.

The setting for cluster analysis in this paper is the unit hypercube of arbitrary dimension. For d variables/dimensions this means that all variables lie in the [0,1] interval. The problem with possible group shape in hypercubes lies not in the centre, but at the corners, where clouds of data points must by construction come to a point. Most standard clustering methods have elliptical or spherical shape restrictions which may not be suitable for this setting.

In addition to simulated data used to compare performance of the competing methods, a real data-set from the cognitive diagnosis setting will be examined. Cognitive diagnosis modelling is used to assess students' attainment levels of skills based on electronic testing

data.
Since an exhaustive examination of all possible cluster analysis methods would not be possible in one paper, this paper looks at one of the most popular algorithmic methods: k-means, both on the original hypercube data and arcsine transformed data, as well two model-based methods: model-based clustering (Gaussian mixture models) and a finite mixture of beta densities. The Methods section will give a brief review of the basics of k-means and model-based clustering, and a more in-depth explanation of the finite mixture of betas approach approach. These approaches will then be applied to simulated data in the Simulations section and the results compared. The Education Testing Example section will introduce a real life cognitive diagnosis dataset and discuss the difficulties inherent in this particular dataset. The different methods will then be applied and the results compared. The performance of the methods and recommendations based on this will be presented in the final Conclusions and Discussion section, as well as other possible approaches in the Education Testing setting.

Methods

K-Means

K-means is one of the mostly commonly used methods and often the first or only one taught in multivariate statistics classes. It is an algorithmic method which does not specify a statistical model to be fit, rather instead defines goodness in clustering in terms reducing the sums of squares between observations and their cluster centres. The objective function being minimized for observations $x_1, \ldots, x_n$ for G clusters is:

$$f(x_1, \ldots, x_n) = \sum_{g=1}^{G} \sum_{i:c(i)=g} (x_i - \overline{x}^{(g)})^2$$

where $c(.)$ is the cluster allocation function, identifying which cluster each observation belongs to and $\overline{x}^{(g)}$ is the mean of all observations allocated to cluster g.
In practice Ward's method is used to find an approximate solution (rather than computing the objective function for all possible combinations of allocations). This is an iterative procedure, which begins with either an initial set of allocations of all observations to clusters or a set of all clusters' starting means, then alternates the following two steps until allocations no longer change. One step, given a set of cluster means, allocates each observation to the nearest cluster based on Euclidean distance from the cluster mean. The other step, given a set of observations' allocations to clusters, re-calculates the cluster means by averaging the allocated observations within each cluster. Since the algorithm is not guaranteed to find the global minimum sum of squares, it is prudent to run the algorithm with multiple random starts and use as the solution the best one in terms of sums of squares.
When using k-means there are 2 main caveats. Firstly, the number of clusters to be found must be specified in advance. While there may be guidance to what the number of cluster may be in some settings, often this is as much a question of interest as the actual make-up of the clusters to be found. Secondly, there is an implicit assumption of conditional independence of the variables within clusters and that each group to be found has spherical shape and the volume of space taken up is equal across clusters. Mainly, issues arise due to non-spherical groups, where larger numbers of clusters are needed to account for dependence within a group or occasionally there may be a larger amount of variation in one variable than others. Scaling/standardizing the data pre-analysis may help

with the later issue (but not always).

In order to select the number of clusters, the usual procedure is to run k-means separately for a range of numbers of clusters (each with multiple random starts to attempt to avoid local minima) and a line-plot of the sums of squares for each number is drawn. The number of clusters is usually selected where the sum of squares ceases decreasing sharply, the “elbow” of the plot.

Arcsine Transformation

Since many popular cluster algorithms will work well with data containing normally or nearly normally distributed groups, one approach is to transform data from the hypercube to another space.

The arcsine transformation is a common transform usually used with binomial(n, p) variables to make their distribution as normal as possible and to reduce the dependence of the variance of the (transformed) distribution on the p parameter. Given a realisation x_i from a $\text{Bin}(n_i, p)$ variable, the transformed realisation y_i is computed as follows:

$$y_i = \sin^{-1}\left(\sqrt{\frac{x_i}{n_i}}\right)$$

The distribution of y_i is then approximately normal with mean $\sin^{-1}(\sqrt{p})$ and variance $\frac{1}{4n_i}$. In practice we may not have information about binomials but merely an estimate $\hat{p}_{ij}$ for each observation i and each variable j so the transformation is simply: $y_{ij} = \sin^{-1}\left(\sqrt{\hat{p}_{ij}}\right)$.

If other methods fail due to lack of normality in the groups, using the arcsine transform before applying them may produce a better result.

Finite Mixture Clustering Approaches

When continuous data is involved, the usual model-based approach is a finite mixture of multivariate normals. A finite mixture model for a population allows each sub-population to be modelled with its own density and the overall population to be modelled by a weighted average of these densities. The weights give the proportions of each sub-population found in the population. The general equation for a finite mixture with G component densities is:

$$f(\mathbf{x}) = \sum_{g=1}^{G} \pi_g f_g(\mathbf{x}),$$

where $0 \leq \pi_g \leq 1$ and $\sum_{g=1}^{G} \pi_g = 1$. The π_g are the weights or mixture proportions and the f_g are the component densities. It is quite common for all the component densities to be from the same parametric family (e.g. Gaussian) and only to vary by parameters across components (e.g. means and/or covariance matrices).

Advantages of this approach include automatic methods based on model selection for allowing the data to dictate the number of clusters (by the number of mixture of components), or principled variable selection methods also based on model selection.

The most convenient method of estimation with a mixture model is the EM algorithm (Dempster et al 1977). Missing data is defined as the unknown group membership variables: $\mathbf{z}_i = (z_{i1}, \ldots, z_{iG})$, where $z_{ig} = 1$ if observation $\mathbf{x}_i$ belong to component g and 0 otherwise. The E-step, which estimates the missing data (given current parameter estimates) is given below:

$$\hat{z}_{ig} = \frac{\hat{\pi}_g \hat{f}_g(\mathbf{x}_i)}{\sum_{g'=1}^{G} \hat{\pi}_{g'} \hat{f}_{g'}(\mathbf{x}_i)}$$

where $\hat{\pi}_g$ are the current estimates of the mixture proportions and $\hat{f}_g(\mathbf{x}_i)$ is the g^{th} component density with the current parameter estimates plugged in, evaluated at $\mathbf{x}_i$.

The EM algorithm can be used to find estimates for a fixed number of components G. The question of selecting the "correct" number of components to fit the data is answered by model selection. The maximized log-likelihoods for the model with a range of different G's (from 1 up to some sensible maximum number) is found (via the EM estimates). The BIC score for each different G is then computed via the equation:

$$\text{BIC(model)} = 2 \times \max \text{log-likelihood} - \nu \log(n)$$

where ν is the number of independent parameters estimated in the model and n is the number of observations. The difference in BIC scores between two models approximates two times the Bayes factor between the two. In the case of normal mixtures (where the true model is a normal mixture), it has been shown to be consistent. However, in the presence of noise variables with no group information, Raftery and Dean (2006) has shown that BIC can perform poorly unless some form of variable selection is carried out. This issue is not addressed here but remains an important future consideration.

Model-based Clustering

In model-based clustering (also know by Gaussian mixture models and other titles), the component densities are Gaussians with parameters: mean vectors and covariance matrices either constrained or free to vary across components. More details about model-based clustering can be found in Fraley and Raftery (2002).

Since Gaussians have elliptical corners and we know that groups at the corners of the hypercube will not have this shape (also the range of the Gaussian is infinite and the hypercube has finite unit range) the arcsine transformation will be used on the data before application of this methodology (using BIC to select the number of components). For interpretation, the estimated component means can be transformed back to the original space.

Finite Mixture of Beta Distributions

In this applicatio,n since we are working in the unit hypercube of dimension d, i.e. $[0,1]^d$, we need the component densities to have a similar range. Therefore, the beta density is proposed. We use the assumption of conditional independence to allow each variable's beta density to be independent of the others (allowing the multivariate density to be a product of univariate densities) within each component. This means that for a d-dimensional hypercube and vector $\mathbf{x} = (x_1, \ldots, x_d)$, we have:

$$f_g(\mathbf{x}) = \prod_{j=1}^{d} \frac{\Gamma(\alpha_{jg} + \beta_{jg})}{\Gamma(\alpha_{jg})\Gamma(\beta_{jg})} x_j^{\alpha_{jg}-1}(1-x_j)^{\beta_{jd}-1},$$

and therefore the mixture model is:

$$f(\mathbf{x}) = \sum_{g=1}^{G} \left(\pi_g \prod_{j=1}^{d} \frac{\Gamma(\alpha_{jg} + \beta_{jg})}{\Gamma(\alpha_{jg})\Gamma(\beta_{jg})} x_j^{\alpha_{jg}-1}(1-x_j)^{\beta_{jd}-1} \right)$$

Aside from the issue of whether conditional independence is a sensible assumption to make, another consideration is the shape of the beta density. While hugely flexible (see Figure 1), if both α_{jg} and β_{jg} parameters are less than 1, the density becomes bimodal. Since the interpretation of such a cluster would be problematic, we constrain the pararmeters to be greater than or equal to 1 at all times (without too much loss of flexibility).

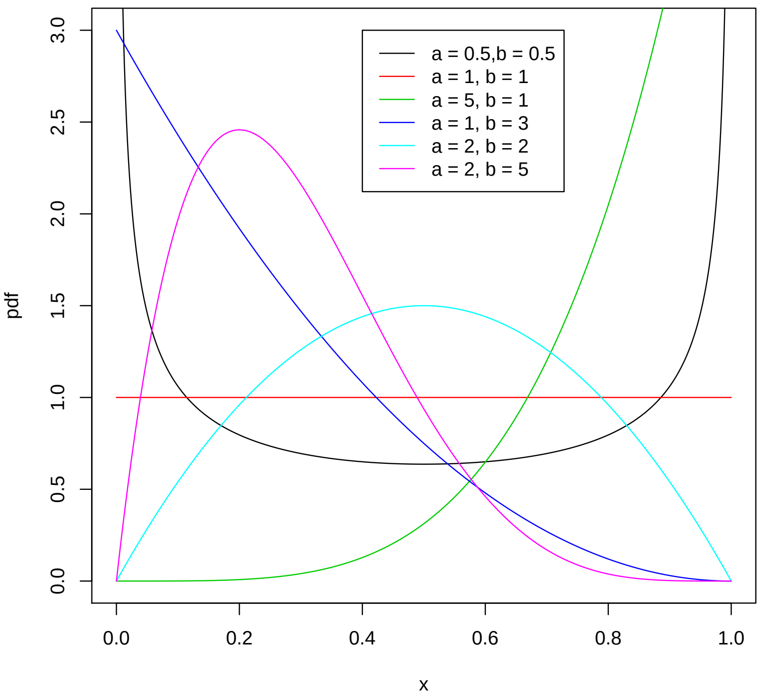

Figure 1: ***Examples of different beta densities***

The E-step for the EM algorithm is as before with $\hat{f}_g(\mathbf{x}_i) = \prod_{j=1}^{d} \frac{\Gamma(\hat{\alpha}_{jg}+\hat{\beta}_{jg})}{\Gamma(\hat{\alpha}_{jg})\Gamma(\hat{\beta}_{jg})} x_j^{\hat{\alpha}_{jg}-1}(1-x_j)^{\hat{\beta}_{jd}-1}$ where $\hat{\alpha}_{jg}, \hat{\beta}_{jg}$ are the current component parameter estimates. The M-step, which estimates the model parameters, both π_g's and α_{jg}, β_{jg}'s, given the current estimates of the group membership variables $\hat{\mathbf{z}}_i$, has no closed form for the solution, so constrained numerical maximization is used at each M-step.
Given a particular value of G, initial starting values are needed for the EM algorithm. The algorithm in this case is sensitive and random starting values (for either parameters or membership vectors) will result in poor output. K-means results can be used to initialize the EM algorithm. The method of moments is used to get rough initial estimates in the following way.

1. If we have K-means classification $c()$ where $c(i) = g$ means $\mathbf{x}_i$ belongs to cluster g then we can get j^{th} variable's sample cluster mean $\left(\bar{x}_g^{(j)} = \frac{1}{N}\sum_{i:c(i)=g} x_{ij}\right)$and variance $\left(v_g^{(j)} = \frac{1}{N}\sum_{i:c(i)=g}(x_{ij} - \bar{x}_g^{(j)})^2\right)$

2. Matching these to the mean $\left(\frac{\alpha_{jg}}{\alpha_{jg}+\beta_{jg}}\right)$ and variance $\left(\frac{\alpha_{jg}\beta_{jg}}{(\alpha_{jg}+\beta_{jg})^2(\alpha_{jg}+\beta_{jg}+1)}\right)$ of the beta distribution with parameters α_{jg} and β_{jg} gives us:

$$\begin{aligned}\hat{\alpha}_{jg} &= \bar{x}_g^{(j)}\left(\frac{\bar{x}_g^{(j)}(1-\bar{x}_g^{(j)})}{v_g^{(j)}} - 1\right)\\ \hat{\beta}_{jg} &= (1-\bar{x}_g^{(j)})\left(\frac{\bar{x}_g^{(j)}(1-\bar{x}_g^{(j)})}{v_g^{(j)}} - 1\right)\end{aligned}$$

Simulations

The first simulation is a simple 1-d example with 2 groups. This data is generated as a 50:50 mixture from two betas, one with shape parameters 1 and 3 and the other with parameters 5 and 1. A histogram of the data with the true mixture density superimposed as a line is given on the left of Figure 2. The arcsine transformed data is given on the right. In both, one can clearly see two peaks and the transformation has done a good job of normalizing the groups. Results of the various clusterings are shown in Table 1

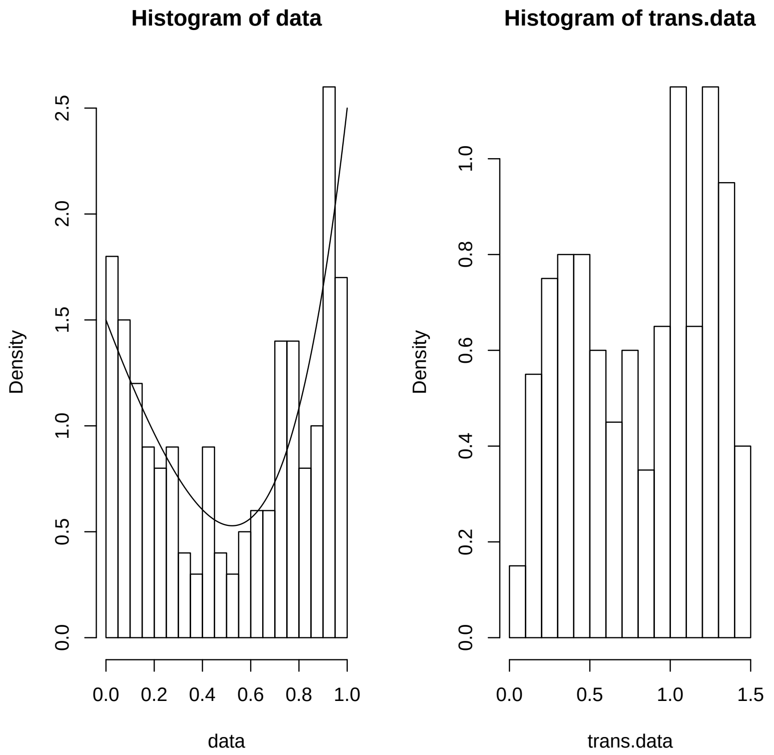

Figure 2: ***One-dimensional two-component beta mixture example***

All methods (sometimes once they were fixed on the true number of groups) performed similarly. The beta mixture does best (as would be expected given the generation of the data), automatically selecting the best number of clusters and with the lowest number of misclassifications. The shape parameters estimated from EM were 1.00, 4.07 and 5.56, 1.3 for the two components respectively.

The second example has 100 2-dimensional datapoints generated by truncated normals with means at the four corners of the hypercube (unequal numbers of points in different groups). The data (original and arc-sine transformed) is plotted in Figure 3. Results are given in Table 2. This shows a similar pattern of results to the previous example, with the transformation working well to improve clustering results compared to the original data.

Educational Testing Example

In educational testing, the main goal is to estimate students' current mastery of skills.

Table 1: ***Clustering results on one-dimensional two-component beta mixture example***

Method and data used	Number of clusters/ components chosen (* denotes fixed in advance)	Number of points misclassified
K-means on original data	2*	12
K-means on transformed data	2*	13
Model-Based Clustering on original data	4	98
Model-Based Clustering on original data	2*	13
Model-Based Clustering on transformed data	2	13
Beta-mixture on original data	2	11

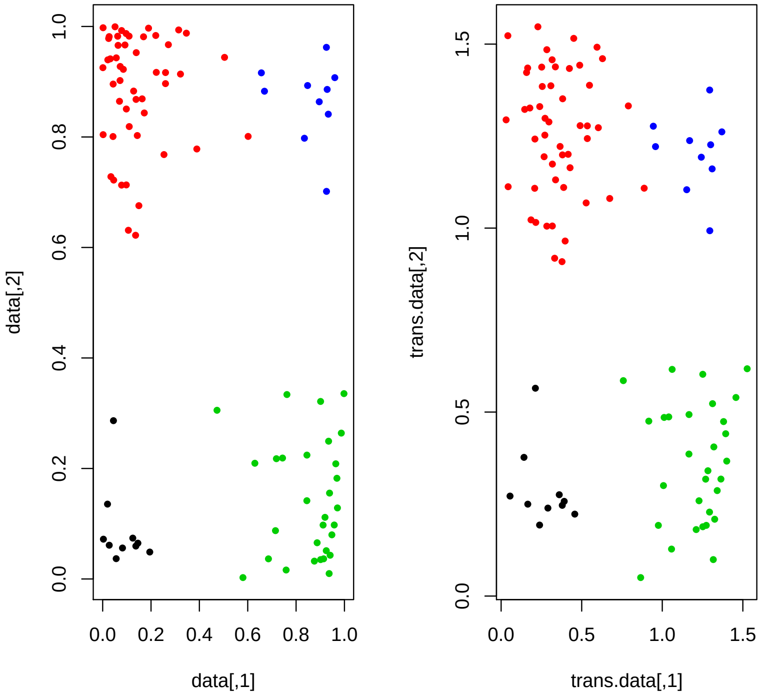

Figure 3: ***Two-dimensional four group example***

Table 2: ***Clustering results on two-dimensional four group example***

Method and data used	Number of clusters/ components chosen (* denotes fixed in advance)	Number of points misclassified
K-means on original data	4*	2
K-means on transformed data	4*	13
Model-Based Clustering on original data	6	30
Model-Based Clustering on original data	4*	2
Model-Based Clustering on transformed data	4	1
Beta-mixture on original data	4	1

Given the infiltration of technology into schools and the increasing use of computers for testing and tutorials, teachers want to get skill estimates quickly from large amounts of data. The current assumption in the cognitive diagnosis community is that student's skill mastery at a given time is either 0 or 1 (i.e. discrete and only in the corners of the hypercube). We believe it is not unreasonable for the skill mastery to fall in the range [0,1]. A student may have complete (1) or non-mastery (0) of a skill but may also be able to use the skill only in some, not all, settings (an intuitive estimate would be between 0 and 1). Therefore for d skills, the data-space will be a d-dimensional unit hypercube.

Given estimates of the students' skill mastery profiles, the goal is to group students so that tailored group-level tutoring can be possible. Further examination of the identified clusters may be able to identify which skills are weakest/strongest in each cluster. The algorithm needs to work in as close to real-time as possible. Current MCMC approaches in this area are only feasible for approximately 15 skills which is limited. The mixture model should be able to handle a higher number without too much of a bottle-neck in computing time.

The estimate, based on Ayers et al (2008) for skill mastery is called the capability matrix, with an entry for each of N students and each of K skills. For each student and each skill the ratio of correct answers for questions involving the skill to number of questions involving the skill attempted, gives the capability estimate. The data from the Assistment project (Ayers et al 2008) gives information on 3 mathematical skills: evaluating functions (8 questions), multiplication (20 questions) and unit conversion (2 questions). There are 26 questions of 3 types: 6 involving only evaluating functions, 18 involving only multiplication and 2 involving both unit conversion and multiplication. 551 students sat this test. Since not all students saw all questions, there will be missing data in the response. However, as long as the student has attempted at least one question per skill, a capability estimate is possible. Since only 2 out of 26 questions involved unit conversion, it is likely that many students will not have seen either and no estimate of unit conversion skill can be made. For this example, a complete case analysis is used (only students with all 3 skill capability estimates are analyzed) but because of conditional independence, it would be easily possible to allow observations with missing data in the beta mixture model without need for imputation. The skill profile estimates in the 3-dimensional unit hypercube are plotted in Figure 4.

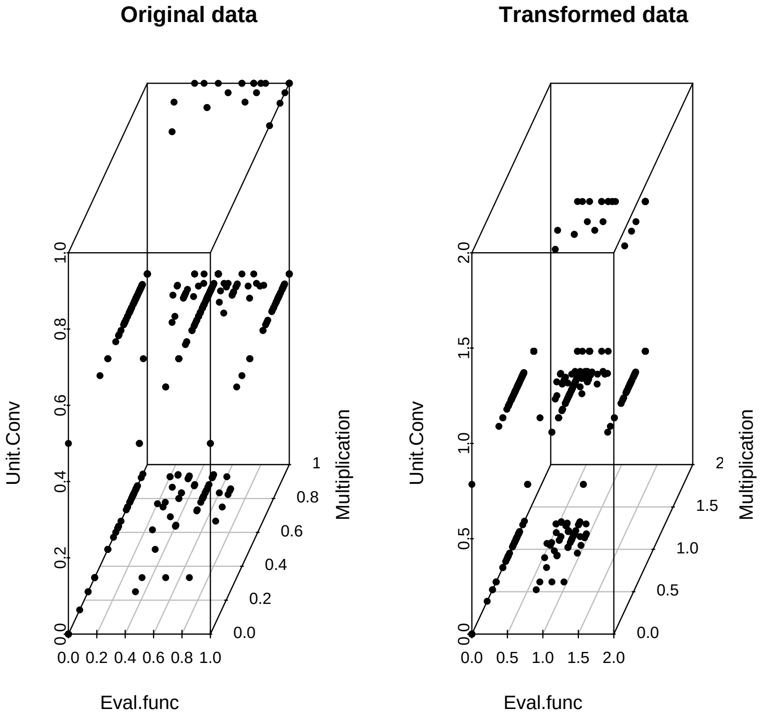

Figure 4: ***Assistment mathematics skills example***

The initial issue that is apparent, is that the data lie in the three 2-dimensional subplanes of the space. This is due to the fact that there were only 2 questions involving unit conversion, therefore only 3 values were possible for this skill estimate: 0 (0 out of 1 or out of 2 questions right), 1/2 (1 out of 2 right) and 1 (1 out of 1 or 2 out of 2 right). The cluster methods are applied to each plane in turn (101 observations in plane 0, 415 in plane 0.5 and 35 in plane 3), the issue of dimension reduction being left to another paper. The results for all methods (except for k-means which we don't have a cluster number to be set) are given in Table 3.

Table 3: ***Clustering results on Assistment Data***

Method and data used	Number of clusters/components chosen (Unit Conversion plane 0, plane 0.5, plane 1)
Model-Based Clustering on original data	21 (19, 1, 1)
Model-Based Clustering on transformed data	31 (15, 1, 15)
Beta-mixture on original data	5 (2, 1, 2)

While the true number of groups is not know, both model-based clustering results seem high (particularly for planes 0 in both and 1 in the transformed data result). Some of this is down to the fact, that due to the granularity of the data (evaluating functions has only

8 questions), a large number of observations lie on top of each other on the same points. Model-based clustering attempts to model these with their own components while the beta mixture model does not.

Conclusions and Discussion

The arc-sine transform seems to do an excellent job of normalizing the data groups, allowing the use of more traditional clustering methods and model-based clustering. While k-means does well, there is no automatic method for choosing number of clusters, model-based clustering on the transformed data gets around this. In practice, there will likely be a limited number of special helpers to allow group-work within the classroom. Therefore, the goal may be to find the best clusters for a given number of groups G which would be possible in all approaches listed here by fixing the number of clusters/components. In addition, there may be an interest in the hierarchical structure of the groups (which are closest together and in what way) which cannot be addressed by k-means or mixtures directly (although distance between cluster means could be used to provide a post-hoc hierarchy) but can be dealt with via hierarchical agglomerative clustering. This can be based on distances or mixture models as needed.

The mixture model approach is attractive because of its flexibility and automatic decision-making. Variable selection could be incorporated into the mixture model approaches discussed in this paper without much difficulty (allowing the selection of variables/skills in the set that differentiate between groups). If it is suspected that correlation in the groups may be causing an overestimation in the number of groups by number of clusters, mixture models can use the methodologies proposed in Hennig (2010) to look at combining components to estimate more complicated group estimation than is allowed by the conditional independence assumption.

In the Assistment example, the capability estimate of Ayers et al (2008) which gives data in a unit hypercube was used as the basis of all cluster approaches. However, this quantity allows for no confidence to be given for the estimate, e.g. an estimate of 0.7 is much more confident for 70 out of 100 correct than for 7 out of 10, but this information is lost if the only reported number is the capability without context as to the number of questions/trials that produced it. Instead of a finite mixture of betas, a more appropriate approach may be a finite mixture of binomials, looking at the number of questions attempted by each student and the number of correct answers.

REFERENCES (RÉFERENCES)

Ayers, E., Nugent, R., and Dean, N. (2008) Skill Set Profile Clustering Based on Student Capability Vectors Computed from Online Tutoring Data *Educational Data Mining 2008: 1st International Conference on Educational Data Mining, Proceedings (refereed,* 210–217

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society - Series B 39*(1), 1–38.

Fraley, C. and A. E. Raftery (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association 97*(458), 611–631.

Hennig, C. (2010). Methods for merging Gaussian mixture components. *Advances in Data Analysis and Classification 4*, 3–34.

Raftery, A. E. and Dean, N. (2006). Variable Selection for Model-Based Clustering. *Journal of the American Statistical Association 101*(473), 168–178.