1 INTRODUCTION

Currently, data is accumulated at unprecedented rates. Access to detailed information of customers and their behavior is not an exception. We develop a model based segmentation approach that accommodates and exploits heterogeneous data. A finite mixture regression model is extended with variable selection abilities through likelihood penalization. This approach merges simultaneous estimation of a finite mixture model based on the EM algorithm with continuous variable selection into a single feasible procedure. The result is a flexible and powerful modeling algorithm that is able to deal with todays complexity and high-dimensionality of datasets. The model combines the value of mixture modeling and continuous feature selection resulting in a synergy of their advantages. The flexibility allows for finding groups of related observations while selecting the optimal subset of variables within these groups independently. A simulation study is done to examine performance and to display the advantages of this approach. The results indicate that extending a finite mixture model with variable selection abilities yields a powerful tool. Good performance is observed in terms of selecting the correct subset of variables to include while accurately estimating the effects of these variables. The model excels in high-dimensional settings where a relatively large amount of variables are of interest.

1.1 FINITE MIXTURE MODELS

We will now discuss possibilities to model heterogeneous data. Heterogeneity is often considered by grouping similar observations into groups. When dealing with data of individuals this can be seen as a segmentation of the customer-base. This concept emerged in the late 1950s. In the early days, segmentations were often based on simple and common characteristics such as gender or age. Although the idea of segmentation appears simple, it is one of the most researched topics in marketing science in terms of scientific development and methodology (Wedel and Kamakura, 2012). A segmentation can be achieved by means of a finite mixture model, which is simply put a combination of several distributions. The first influential analysis based on a mixture model originates from 1894 where the biometrician Pearson fitted a two component mixture of normal densities (Pearson, 1894). Since then major advances have been made to accommodate the need for methods that can handle large and complex datasets.

Meanwhile, a surge in popularity of machine learning approaches also increased the application of cluster analysis techniques. These clustering methods used for segmentation are often heuristic in nature. These algorithms are in general based on some arbitrary measure of distance to determine the similarity of observations (Tuma and Decker, 2013). The specific choice of distance measure significantly impacts the results of the analysis. This is especially true when categorical variables are included in the analysis. In this case a preliminary transformation of the data is required to allow application, such as Gower's distance (Gower, 1966).

Finite mixture models alleviate some of the common issues associated with heuristic methods. They provide a model based approach for segmentation (Wedel and Kamakura, 2012). In order to exploit differences in the customers we require a flexible model combined with an inference method to interpret the results (Allenby and Rossi, 1998). Finite mixture models have been expanded in the 1990s with practices composed of linear regression models and generalized linear models (Wedel and DeSarbo, 1995). The practical application, potential and theoretical attention of mixture models has grown considerably since 1995 (McLachlan and Peel, 2004). This growth can be explained by the immense flexibility to model unknown distributions in a convenient manner and secondly by advances in computational power. In addition, finite mixture models are particularly useful to capture and describe some type of grouping structure present within a complex dataset. These models have seen utility in various fields such as astronomy, biology, genetics, medicine, economics, engineering and marketing (McLachlan and Peel, 2004). Mixture models can also be combined with machine learning algorithms. An interesting present-day application is the speech of Siri on Apple devices. The technology behind Siri's voice is called a deep mixture density network (MDN) which combines deep neural networks with Gaussian mixture models (Apple, 2017). In short, finite mixtures can be seen as a more elegant approach compared to heuristic methods and have obtained an important position in modern market segmentation applications (Wedel and Kamakura, 2012; McLachlan and Peel, 2004).

Whether the data is simple or complex, the principle of segmentation is similar. The fundamental idea is that a single distribution or model fails to sufficiently describe a collection of data due to the presence of heterogeneity. A finite mixture model is based on a mixture of multiple parametric distributions to describe the underlying structure of some data. In our case, we assume the entire population of customers contains unidentified subgroups. This heterogeneity is called latent, meaning it is unobserved. The groups within the population can be interpreted as a finite number of latent classes also referred to as segments or components (Muthén and Shedden, 1999). Failure to recognize the presence of subpopulations and account for heterogeneity results in misleading or incorrect inference. Finite mixture models provide an effective method to consort population heterogeneity and provide a flexible and powerful way to model univariate or multivariate data. Specifying the parametric distribution of the latent structure in the data is not required to perform estimation. This is a highly attractive feature as it prevents bias in parameter estimation as a result from potential misspecification. An interesting fact is that normal mixture models can be used to test the performance of estimators with their ability to capture deviation from normality (McLachlan and Peel, 2004). Normal mixtures have helped in the development of robust estimators. For example the contaminated normal distribution proposed by Tukey where the density of a point is interpreted as a mixture of two normal distributions with different variances (Tukey, 1960). A more general incomplete contamination form is considered in the work of M-estimators by Huber et al. (1964). Finite mixtures are often labeled as a semi-parametric approach. Jordan and Xu describes them as an interesting niche between parametric and non-parametric. A parametric formulation of the mixture is determined whereas the number of components is allowed to vary which can be interpreted as non-parametric (Jordan and Xu, 1995). This description can be used to explain why a mixture model possesses the flexible properties of non-parametric approaches while retaining attractive analytical advantages of parametric approaches (McLachlan and Basford, 1988).

Finite mixture models can model the joint distribution of multiple variables, in contrast to non-parametric algorithms such as K-means or K-nearest neighbors. Although non-parametric methods are often fast and require no assumptions on the distribution of the data, there are some drawbacks associated with these methods. One cause of discussion is the fact that similarity between observations is based on a chosen distance measure. A finite mixture is based on a statistical model which requires to choose distribution. Yet, a result is that mixture models offer more extensive inference and interpretation possibilities. Uncertainty in the classification can be taken into account in contrast to non-parametric methods which result in hard grouping or classification. This means observations are assigned to components as if no certainty is involved in this membership. Often, this is a rough assumption as group memberships are in reality not fully certain. Moreover, the uncertainty in grouping may even be meaningful for interpretation of the cluster results. Furthermore, mixture models have the capability to handle groups with different sizes, correlation structures and overlapping of segments in contrast to many other techniques. On the contrary, non-parametric clustering techniques prefer groups of equal size and are not suited to handle overlapping segments due to hard classification. If an observation shares properties of multiple subgroups, this membership information is lost by hard clustering.

In this research we are interested in relating the response variable y with a set of explanatory features. DeSarbo and Cron introduced a methodology for cluster-wise linear regressions giving rise to finite mixture regression modeling (1988). Finite mixture regression models provides a flexible method to simultaneously estimate both group membership and separate regression functions to explain the response variable within each segment (Wedel and Kamakura, 2012). It has been proven that any continuous distribution can be estimate arbitrarily well by a finite mixture of Gaussian distributions (McLachlan and Peel, 2004; Lindsay, 1995). Consequently, a Gaussian or normal mixture regression constitutes the foundation of our model.

The density function of a general S-component finite mixture model can be formulated as

$$f(y|x,\Theta) = \sum_{s=1}^{S} \pi_s \cdot f(y|x,\theta_s), \tag{1}$$

where y is a vector of response variables, x is a vector

of regression variables, π_s is the prior probability of belonging to component s, each θ_s is a vector with component specific parameters for density f, and $\Theta = \{\theta_1, \ldots, \theta_p\}$ is a vector containing all parameters to specify the mixture. The prior probability π_s is also referred to as the mixing coefficient. The restrictions on the parameters are as follows. π_s is a probability, thus satisfying the follow conditions

$$\begin{aligned} &\sum_{s=1}^{S} \pi_s = 1, \\ &0 < \pi_s \leq 1 \; \forall s = 1, \ldots, S. \end{aligned} \tag{2}$$

For the component specific parameter vectors we have

$$\theta_s \neq \theta_k \; \forall s \neq k \text{ with } s, k \in \{1, \ldots, S\}. \tag{3}$$

Next, the group membership is the conditional probability of an observation belonging to segment s. This is also referred to as the posterior probability. We can compute this probability using Bayes' theorem as

$$z_{is} = \mathbb{P}(s|y_i, x_i, \Theta) = \frac{\pi_s \cdot f(y_i|x_i, \theta_s)}{\sum_{k=1}^{S} \pi_k \cdot f(y_i|x_i, \theta_k)}. \tag{4}$$

The corresponding log-likelihood function of the S-component mixture model is computed as

$$\begin{aligned} \mathcal{L}(\Theta) = \log f(y|x, \Theta) &= \log \prod_{i=1}^{N} f(y_i|x_i, \Theta) \\ &= \sum_{i=1}^{N} \log \sum_{s=1}^{S} \pi_s \cdot f(y_i|x_i, \theta_s) \end{aligned} \tag{5}$$

with corresponding maximum likelihood (ML) estimate

$$\begin{aligned} \widehat{\Theta}_{ML} &= \arg\max_{\Theta} \mathcal{L}(\Theta) \\ &= \arg\max_{\Theta} \left[\log f(y|x, \Theta)\right] \\ &= \arg\max_{\Theta} \left[\sum_{i=1}^{N} \log \sum_{s=1}^{S} \pi_s \cdot f(y_i|x_i, \theta_s)\right]. \end{aligned} \tag{6}$$

In this work we use a finite mixture regression model with Gaussian distributed components such that

$$f(y_i|x_i, \Theta) = \sum_{s=1}^{S} \pi_s \cdot \frac{1}{\sqrt{2\pi}\sigma_s} \exp\left(\frac{(y_i - x_i^{\mathrm{T}}\beta_s)^2}{2\sigma_s^2}\right) \tag{7}$$

where every component s has an independent vector of regression coefficients β_s and variance σ_s^2.

1.2 ESTIMATION

As the parameters of the mixture in Equation are unknown they need to be estimated from the data. Estimation options include method of moments, maximum likelihood (ML) and Bayesian approaches (McLachlan and Peel, 2004). ML estimation can be done with numerical methods such as Newton-Raphson's algorithm. However, the likelihood function as given in Equation 6 can be difficult to solve and generally contains multiple local maxima. Numerical optimization methods often do not perform smoothly. Alternatively, a Bayesian approach based on Markov Chain Monte Carlo (MCMC) sampling can be used to estimate the parameters (Diebolt and Robert, 1994). The likelihood function can also be solved with the Expectation-Maximization (EM) algorithm by Dempster et al. (1977). The EM algorithm is an iterative hill-climbing procedure to estimate the parameters that maximize the log-likelihood function. It is a prevalent approach for problems associated with incomplete data caused by missing variables or unobserved heterogeneity (Dempster et al., 1977). Usefulness of the EM algorithm in finite mixture models is reported by McLachlan and Basford among many others (1988).

Solving Equation 6 to obtain the maximum likelihood estimates is a difficult problem. This problem can be approached by assuming that we are dealing with incomplete observations that originate from non-observed complete data. In other words, we assume that our observations originate from a finite number of groups. However, the group membership variable is not part of the available data. In order to estimate the parameters in the mixture we augment our incomplete data with a group membership variable Z yielding the complete data. This approach allows to define a complete data log-likelihood function as

$$\begin{aligned} \mathcal{L}_c(\Theta) &= \log f(y, Z|x, \Theta) \\ &= \sum_{i=1}^{N} \sum_{s=1}^{S} z_{is} \cdot \log\left[\pi_s \cdot f(y_i|x_i, \theta_s)\right]. \end{aligned} \tag{8}$$

where the vector $Z = \{z_i, \ldots, z_N\}$ contains labels indicating group membership for every observation i. The complete likelihood function is also referred to as the classification likelihood in some cases.

Next, the EM algorithm is used to estimate the parameters by treating z_{is} as missing data. The algorithm

can be subdivided into two steps. The Expectation-step and the Maximization-step. Every iteration provides updated parameter estimates $\widehat{\Theta}$. The procedure is stopped if a predefined convergence criterion is met. The E-step computes the expectation of the complete data log-likelihood conditional on the data and the current estimates $\widehat{\Theta}^{(t)}$ as

$$\mathbb{E}\left[\mathcal{L}_c(\Theta)\right] = \mathbb{E}\left[\log f(y, Z|x, \widehat{\Theta}^{(t)})\right]. \tag{9}$$

In this step the group memberships, also called posterior probabilities, are calculated based on the current parameter values using Equation 4 such that

$$z_{is} = \frac{\pi_s^{(t)} \cdot f(y_i|x_i^{(t)}, \theta_s^{(t)})}{\sum_{k=1}^{S} \pi_k^{(t)} \cdot f(y_i|x_i^{(t)}, \theta_k^{(t)})}. \tag{10}$$

Consequently, the M-step maximizes the expected value seen in Equation 9 with respect to Θ

$$\begin{aligned}\widehat{\Theta}^{(t+1)} &= \arg\max_{\Theta} \mathbb{E}\left[\mathcal{L}_c(\widehat{\Theta}^{(t)})\right] \\ &= \arg\max_{\Theta} \mathbb{E}\left[\log f(y, Z|x, \widehat{\Theta}^{(t)})\right] \\ &= \arg\max_{\Theta} \sum_{i=1}^{N}\sum_{s=1}^{S} z_{is} \cdot \log\left[\pi_s \cdot f(y_i|x_i, \theta_s^{(t)})\right].\end{aligned} \tag{11}$$

The estimation procedure described above can be summarized as follows. First, we formulate our problem as a missing data setup. Second, we iteratively estimate the parameters with the EM algorithm.

Data Setup

- Observed data: the observations as available (y_i, x_i)
- Missing data: the group membership information of each observation z_{is}
- Complete data: the observations supplemented with the group memberships

Following this setup allows the likelihood function to be maximized with the following algorithm.

Algorithm 1 EM Algorithm for a Finite Mixture Regression

1. Determine a set of initial parameter estimates Θ^{ini} that define the mixture to start the algorithm.
2. E-step: Estimate the posterior probabilities based on the current set of parameter estimates

$$z_{is} = \frac{\pi_s \cdot f(y_i|x_i, \theta_s)}{\sum_{k=1}^{S} \pi_k \cdot f(y_i|x_i, \theta_k)}. \tag{12}$$

   Derive the prior class probabilities as

$$\pi_s = \frac{1}{N}\sum_{i=1}^{N} z_{is}. \tag{13}$$

3. M-step: Update the parameter estimates using the current posterior probabilities

$$\arg\max_{\Theta} \sum_{i=1}^{N}\sum_{s=1}^{S} z_{is} \cdot \log\left[\pi_s \cdot f(y_i|x_i, \theta_s)\right]. \tag{14}$$

4. Evaluate the complete log-likelihood function

$$\mathcal{L}_c(\Theta) = \sum_{i=1}^{N}\sum_{s=1}^{S} z_{is} \cdot \log\left[\pi_s \cdot f(y_i|x_i, \theta_s)\right]. \tag{15}$$

5. Repeat steps 2 to 4 until a defined convergence criterion is met.

A potential issue of finite mixture models is identifiability. For consistent estimation of the parameters identifiability is a necessary condition (Hennig, 2000). In some cases different sets of parameter estimates can describe the same density function. The model is identifiable if one unique set of parameters is able to define the distribution. In terms of the model as introduced in Equation 1 we need that for any two parameters Θ and Θ^*

$$\begin{aligned} f(y|x,\Theta) &= f(y|x^*,\Theta^*) \\ \sum_{s=1}^{S} \pi_s \cdot f(y|x,\theta_s) &= \sum_{s=1}^{S^*} \pi_s^* \cdot f(y^*|x^*,\theta_s^*) \end{aligned} \tag{16}$$

implies $\Theta = \Theta^*$ and $S = S^*$. It has been proven that given some mild conditions many finite mixture models are identifiable, including Guassian types (Titterington et al., 1985).

1.3 VARIABLE SELECTION

Like in almost any model, feature selection is an important aspect. Feature or variable selection has been given increasing attention in statistical research. The current era of high-dimensional problems require adequate techniques to deal with a large number of variables. Therefore, it is desirable to exclude irrelevant information from the model considering the goal of a parsimonious solution. In addition to increasing the goodness of fit, variable selection has the potential to improve the interpretability of our model (James et al., 2013). First we cover traditional approaches. Second, we review some developments in the field of feature selection based on regularization. Thereafter, we describe how to merge variable selection and simultaneous estimation of parameters with the EM algorithm into a single feasible mixture modeling procedure.

ℓ_0 penalization is a fundamental concept in various model selection methods. This penalization provides a clear interpretation for subset selection while having convenient sampling properties (Barron et al., 1999). Common feature selection methods are stepwise procedures where variables are iteratively added or removed to find the best subset of features. Often applied examples are stepwise selection, forward selection and backward elimination. The resulting models are compared based on goodness-of-fit measures such as AIC or BIC. However, due to increasing data complexity and size, these stepwise procedures quickly explode to the point of computational infeasibility. Even when a mixture consists of a moderate number of components and variables, classical subset selection approaches are intensive (Khalili and Chen, 2007). In addition, these algorithms are greedy and do not provide any guarantee in finding the optimal subset of variables. Moreover, subset selection approaches are shown to be unstable and further limitations are evident (Breiman, 1995).

As consequence, recent advances have given rise to multiple new forms of penalized likelihood methods with the ability to perform feature selection. The purpose of these methods is to control the number of variables included in the model while taking parsimony and therewith computational intensity into account (Fan and Lv, 2010). Some of these developments are sparked by ultra-high dimension problems where the number of variables p is larger than the number of observations N such that $p > N$. This situation is currently no exception in various fields such as genomics, web analysis, health sciences, finance, economics and machine learning (Fan and Lv, 2010). Hence, it is no surprise that regularization techniques have obtained an important place in modern statistical research and applications.

It is ideal to obtain a parsimonious and well interpretable model while capturing the structure of our data in a satisfactory manner. Naturally, this is very often the goal. This trade-off accounts to finding a good balance in the amount of information needed to explain the structure of the data. Hence, our goal is to estimate variable effects while simultaneously selecting the important ones by excluding irrelevant variables from the model. This is a complicated optimization problem as we are iteratively estimating a mixture of models instead of a single model. As explained, we assume the data originates from multiple subpopulations. A key consequence stems from this assumption. Namely, the presence of subgroups implies that variables may also vary across components. In turn, this gives rise to a particular interest in selecting the optimal subset of features within each separate segment while correctly estimating the effects of these variables. The variation in features across components can surface in two ways. Firstly, through a difference in the optimal subset of variables and secondly, through a varying importance of the selected variables within a component. In

order to achieve this high amount of flexibility, we need to combine estimation of our model with a continuous variable selection algorithm that has the freedom to operate independently across components.

We now introduce several forms of penalization methods from the starting point of Ordinary Least Squares (OLS). Thereafter, we formulate an approach that combines a finite mixture model with penalization. OLS minimizes the residual sum of squares (RSS) formulated as

$$\beta_{OLS} = \min_{\beta} \underbrace{\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2}_{\text{RSS}}. \quad (17)$$

In order to obtain an estimation method that can perform feature selection we extend the model with penalization. The principle of ℓ_0 penalization can be seen as part of the general families of ℓ_q penalties, also referred to as bridge functions (Frank and Friedman, 1993). This form of penalties is given by

$$\lambda \sum_{j=1}^{p} |\beta_j|^q \quad (18)$$

where $0 < q \leq 1$ in order to achieve variable selection abilities. For $q = 0$ we obtain the AIC or BIC penalty depending on λ. This function of families can be used to introduce penalization methods starting with ridge regression by Hoerl and Kennard (1970). Ridge regression has lead to more recent advances such as the lasso by Tibshirani (1996) and the elastic net by Zou and Hastie (2005). The lasso and elastic net both posses the ability to perform continuous variable selection which is further discussed in the next sections.

1.3.1 RIDGE REGRESSION

Ridge regression is the foundation of many modern penalization methods (Hoerl and Kennard, 1970). It is also known as Tikhonov regularization (Tikhonov et al., 1977) or as weight decay in neural networks in the field of machine learning (Friedman et al., 2001). Instead of ℓ_0 penalization it is based on the ℓ_2 norm (Hoerl and Kennard, 1970). This form of penalization is obtained by setting $q = 2$ in Equation 18 resulting in the following objective function

$$\beta_{RDG} = \min_{\beta} \underbrace{\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2}_{\text{RSS}} + \underbrace{\lambda \sum_{j=1}^{p} |\beta_j|^2}_{\text{Penalty}}. \quad (19)$$

Ridge regression has the property to decrease the effect of non-important variables, this is referred to as shrinking. The amount of shrinkage is controlled by the λ parameter (Friedman et al., 2001). In addition, the variance of the coefficient estimates can be significantly decreased as result of shrinking (James et al., 2013). Although the effect of a variable can be decreased with ridge regression, it cannot be nullified. In other words, ridge regression cannot perform feature selection to obtain a more parsimonious model (Zou and Hastie, 2005). Yet, shrinking to exactly zero is highly desirable when the goal is to select the most important variables in the model. A similar procedure that does possess the ability to perform feature selection is the least absolute shrinkage and selection operator (lasso) introduced by Tibshirani (1996).

1.3.2 LASSO

In contrast to ridge regression the lasso is based on ℓ_1 instead of ℓ_0 penalization. This is achieved by setting $q = 1$ in Equation 18 yielding the following objective function

$$\beta_{LAS} = \min_{\beta} \underbrace{\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)}_{\text{RSS}} + \underbrace{\lambda \sum_{j=1}^{p} |\beta_j|}_{\text{Penalty}}. \quad (20)$$

The lasso can be described as a continuous subset selection algorithm with the ability to shrink the effect of unimportant variables similar to ridge regression (Tibshirani, 1996). The algorithm constrains the total magnitude of the coefficients resulting in the scaling of a variables effect based on importance. In contrast to ridge regression, the lasso possesses variable selection properties. This is achieved by the ability to shrink the effect of a certain variable all the way down to zero. This can be interpreted as exclusion of this respective variable from the model. A numerical advantage of the lasso is a convex penalty function. This is very convenient from a computational viewpoint.

The concept of the lasso is influenced by Breiman's non-negative garrotte (Breiman, 1995). A drawback

of the non-negative garrotte is that it is not defined when a problem involves more parameters p than observations N which is not uncommon present-day. The lasso is still valid in this case but shrinkage of the non-zero coefficient causes non-ignorable bias towards zero yielding inconsistent estimates (Fan and Li, 2001). The bias can be reduced by a modification of the penalty function such that large coefficients are shrunken less (Fan et al., 2004). This idea is used in another variable selection algorithm known as the smoothly clipped absolute deviation (SCAD) (Fan et al., 2004).

Alternatively, the lasso can be extended by including data-dependent weights which is known as the adaptive lasso (Zou, 2006). Now, the strength of penalization is allowed to vary across different coefficients due to adding adaptive weights in the penalty giving the following objective function

$$\beta_{ALS} = \min_{\beta} \underbrace{\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)}_{\text{RSS}} + \underbrace{\lambda \sum_{j=1}^{p} \hat{w}_j |\beta_j|}_{\text{Penalty}} . \tag{21}$$

where $\hat{w}_j$ are the coefficient dependent weights with the power to control penalty strength per coefficient. This weighting vector is determined by

$$\hat{w}_j = \frac{1}{|\hat{\beta}_j^{\text{ini}}|^{\gamma}} \tag{22}$$

where $\hat{\beta}_j^{\text{ini}}$ are initial estimates of the coefficients which can be obtained from a consistent estimator for $\hat{\beta}_j$ such as OLS or ridge regression. In order for the adaptive lasso to be consistent $\hat{\beta}_j^{\text{ini}}$ need to be consistent. Coefficients with lower initial estimates are penalized more through the weights vector $\hat{w}_j$. It has been shown that this extension yields the oracle property (Zou, 2006; Fan and Li, 2001; Fan et al., 2004). An estimator has the oracle property if it has the ability to be consistent in both parameter estimation as well as variable selection. This is further examined in the Simulation study in section 2. On the contrary, the regular lasso does not posses the oracle property which has been shown to be associated with the bias problem (Zou, 2006). The adaptive lasso consistently estimates parameters while retaining the desirable convexity property (Friedman et al., 2001).

Recent studies have discovered that the lasso is related to the maximum margin explanation which is key in support vector machines (SVM) and boosting algorithms (AdaBoost, XGBoost) in the field of machine learning (Rosset et al., 2004). The lasso has been used to explain the success of boosting which can be interpreted as a high-dimensional lasso without explicit use of the ℓ_1 penalty (Friedman et al., 2004, 2001). However, a drawback of both lasso algorithms is the performance in presence of multicollinearity. In practice, variables can be highly correlated especially when the number of variables is relatively large. In this situation the lasso has the tendency to select merely one of these correlated variables in an arbitrary fashion while ignoring the others. Zou and Hastie have shown the lasso path to be unstable in case of multicollinearity yielding unsatisfactory results (2005). These difficulties are overcome by a more recent regularization technique called the elastic net (Zou and Hastie, 2005). For this reason, we select the elastic net as variable selection algorithm in our model.

1.3.3 ELASTIC NET

A relatively new regularization and variable selection method is the elastic net (Zou and Hastie, 2005). This method is closely related to the lasso which has proven to be a valuable asset in modern model fitting and covariate selection. Some of the limitations of the lasso are solved by combining the ℓ_1- and ℓ_2 norm into a new penalty function given by

$$\xi_{NET}(\beta_j) = \lambda \sum_{j=1}^{p} \left(\alpha |\beta_j| + \frac{(1-\alpha)}{2} |\beta_j|^2 \right) \tag{23}$$

such that the following problem is solved

$$\begin{aligned} \beta_{NET} = \min_{\beta} & \underbrace{\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)}_{\text{RSS}} \\ & + \underbrace{\lambda \sum_{j=1}^{p} \left(\alpha |\beta_j| + \frac{(1-\alpha)}{2} |\beta_j|^2 \right)}_{\text{Penalty}} \end{aligned} \tag{24}$$

where α is a parameter that determines the mix of the penalties. Setting $\alpha = 0$ results in ridge regression whereas setting $\alpha = 1$ results in the lasso. Hence, this method can be seen as a dynamic blend of ridge regression and the lasso. The elastic net possesses all the desirable properties of the lasso, it can perform automatic variable selection through continuous shrinkage while overcoming the issues regarding multicollinearity. The second term in Equation 23

causes variables with high correlation to be averaged, whereas the first term encourages a parsimonious solution and stabilizes the solution (Friedman et al., 2001). Zou and Hastie describes this method as a stretchable fishing net with the ability to retain all the big fish (2005). It has been shown that elastic net often yields better results than the lasso in simulations and real world data (Zou and Hastie, 2005). To implement the elastic net we make use of the *glmnet* algorithm developed by (Friedman et al., 2009) which is specifically designed for speed and dealing with relatively large datasets.

1.4 EXTENDED FINITE MIXTURE MODEL (MIXNET)

We have first discussed the fundamentals regarding the formulation and estimation of finite mixture models. Second, we introduced penalized estimation methods. The methodology is now expanded by merging these two principles into a single estimation and variable selection algorithm. This approach is inspired by Khalili and Chen who makes use of the lasso to perform variable selection in mixture models (2007). Khalili and Chen have shown that this procedure is consistent and yields equal or better performance than traditional methods such as BIC in terms of model selection whilst greatly reducing computational burden.

We now introduce a model which combines a finite mixture model with the elastic net algorithm. We refer to this model as MIXNET in short. MIXNET combines the power of statistical based finite mixture modeling with the convenience of automatic variable selection. The result is a highly feasible and relatively fast procedure in terms of computational intensity. Variable selection is achieved by shrinkage of parameters through the elastic net algorithm. As consequence, all desirable properties of the elastic net are adopted. MIXNET has the ability to deal with a large number of variables while simultaneously performing continuous selection of the relevant ones. We would like to emphasize the power of this algorithm as it possesses the ability to operate independently within components. Hence, both estimation and variable selection is done in a component specific manner. This increases both the flexibility and potential interpretability of groups in comparison to a variable selection procedure that takes the entire population into account as a whole. Moreover, in case the problem contains more variables than observations, such that $p > N$, MIXNET can still be applied in contrast to a regular likelihood approach.

We now cover the mathematical formulation of this model. To obtain the ability to perform feature selection through shrinkage we take the log-likelihood function as given in Equation 5 and extend it with a penalty term such that we have a penalized log-likelihood function defined as

$$\widetilde{\mathcal{L}}(\Theta) = \mathcal{L}(\Theta) - \text{Penalty}(\Theta). \tag{25}$$

To obtain the MIXNET model we employ the elastic net penalty as given in Equation 23

$$\xi_{NET}(\Theta) = \sum_{s=1}^{S} \lambda_s \sum_{j=1}^{p} \left(\alpha|\beta_{sj}| + \frac{(1-\alpha)}{2}|\beta_{sj}|^2 \right) \tag{26}$$

resulting in a penalized log-likelihood function

$$\begin{aligned} \widetilde{\mathcal{L}}(\Theta) = & \underbrace{\sum_{i=1}^{N} \log \sum_{s=1}^{S} \pi_s \cdot f(y_i|\theta_s)}_{\text{Log-Likelihood}} \\ & \underbrace{- \sum_{s=1}^{S} \lambda_s \sum_{j=1}^{p} \left(\alpha|\beta_j| + \frac{(1-\alpha)}{2}|\beta_j|^2 \right)}_{\text{Penalty}}. \end{aligned} \tag{27}$$

The corresponding maximum likelihood (ML) is then computed by

$$\begin{aligned} \widehat{\Theta}_{ML} &= \arg\max_{\Theta} \widetilde{\mathcal{L}}(\Theta) \\ &= \arg\max_{\Theta} \left[\log f(y|\Theta) - \xi_{NET}(\Theta) \right] \\ &= \arg\max_{\Theta} \left[\sum_{i=1}^{N} \log \sum_{s=1}^{S} \pi_s \cdot f(y_i|\theta_s) \right. \\ &\qquad \left. - \sum_{s=1}^{S} \lambda_s \sum_{j=1}^{p} \left(\alpha|\beta_j| + \frac{(1-\alpha)}{2}|\beta_j|^2 \right) \right]. \end{aligned} \tag{28}$$

Lastly, the complete data log-likelihood function is defined as

$$\begin{aligned} \widetilde{\mathcal{L}}_c(\Theta) &= \log f(y, Z|\Theta) - \xi_{NET}(\Theta) \\ &= \sum_{i=1}^{N} \sum_{s=1}^{S} z_{is} \cdot \log \left[\pi_s \cdot f(y_i|\theta_s) \right] - \xi_{NET}(\Theta). \end{aligned} \tag{29}$$

To obtain estimates of the parameters $\widehat{\Theta}$ the EM algorithm as described in Algorithm 1 is used. The algorithm can be subdivided into two separate steps, the Expectation-step and the Maximization-step. Every iteration provides new parameter estimates $\widehat{\Theta}$. The E-step computes the expectation of the complete data log-likelihood conditional on y and the current estimate $\widehat{\Theta}^{(t)}$. The E-step is given by

$$\mathbb{E}\left[\widetilde{\mathcal{L}}_c(\Theta)\right] = \mathbb{E}\left[\log f(y, Z|\widehat{\Theta}^{(t)}) - \xi_{NET}(\widehat{\Theta}^{(t)})\right] \quad (30)$$

Consequently, the M-step maximizes the expected value in Equation 30 with respect to Θ such that

$$\begin{aligned}\widehat{\Theta}^{(t+1)} &= \arg\max_{\Theta} \mathbb{E}\left[\widetilde{\mathcal{L}}_c(\widehat{\Theta}^{(t)})\right] \\ &= \arg\max_{\Theta} \mathbb{E}\left[\log f(y, Z|\widehat{\Theta}^{(t)}) - \xi_{NET}(\widehat{\Theta}^{(t)})\right]\end{aligned} \quad (31)$$

yielding updated parameter estimates $\widehat{\Theta}^{(t+1)}$. The two steps are repeated until convergence is met resulting in a final solution. The log-likelihood function can be extended with the different penalties introduced above in a similar manner. For instance, to obtain a log-likelihood function with the adaptive lasso penalty.

2 SIMULATION STUDY

Lastly, we compare the performance of Gaussian mixture regression models with different penalization methods by means of a simulation study. The included modeling approaches are:

- **MIXREG**: Regular Gaussian mixture model without variable selection
- **MIXRDG**: Gaussian mixture model combined with the ridge penalty function
- **MIXNET**: Gaussian mixture model combined with the elastic net penalty function
- **MIXLAS**: Gaussian mixture model combined with the lasso penalty function
- **MIXALS**: Gaussian mixture model combined with the adaptive lasso penalty function

The penalty specifications are given in Equation 19 for ridge, Equation 23 for the elastic net, Equation 20 for the lasso and Equation 21 for the adaptive lasso. For the elastic net we consider two penalty mixing proportions, $\alpha = 0.5$ and $\alpha = 0.9$. The choice of alpha is indicated with a subscript. $\alpha = 0.9$ results in stricter regularization as it tends more towards the lasso than the ridge penalty. For completeness we list the (penalized) log-likelihood function of each tested modeling approach. In MIXREG we do not add a penalty term. For MIXALS the coefficient dependent weights $\hat{w}_{sj}$ are obtained through a preliminary ridge regression.

- **MIXREG**:

$$\mathcal{L}(\Theta) = \underbrace{\sum_{i=1}^{N} \log \sum_{s=1}^{S} \pi_s \cdot f(y_i|x_i, \theta_s)}_{\text{Log-Likelihood}} . \tag{32}$$

- **MIXRDG**:

$$\widetilde{\mathcal{L}}(\Theta) = \underbrace{\sum_{i=1}^{N} \log \sum_{s=1}^{S} \pi_s \cdot f(y_i|x_i, \theta_s)}_{\text{Log-Likelihood}} - \underbrace{\sum_{s=1}^{S} \lambda_s \sum_{j=1}^{p} |\beta_{sj}|^2}_{\text{Penalty}} . \tag{33}$$

- **MIXNET$_5$**:

$$\widetilde{\mathcal{L}}(\Theta) = \underbrace{\sum_{i=1}^{N} \log \sum_{s=1}^{S} \pi_s \cdot f(y_i|x_i, \theta_s)}_{\text{Log-Likelihood}} - \underbrace{\sum_{s=1}^{S} \lambda_s \sum_{j=1}^{p} \left(0.5|\beta_{sj}| + 0.25|\beta_{sj}|^2\right)}_{\text{Penalty}} . \tag{34}$$

- **MIXNET$_9$**:

$$\widetilde{\mathcal{L}}(\Theta) = \underbrace{\sum_{i=1}^{N} \log \sum_{s=1}^{S} \pi_s \cdot f(y_i|x_i, \theta_s)}_{\text{Log-Likelihood}} - \underbrace{\sum_{s=1}^{S} \lambda_s \sum_{j=1}^{p} \left(0.9|\beta_{sj}| + 0.05|\beta_{sj}|^2\right)}_{\text{Penalty}} . \tag{35}$$

- **MIXLAS**:

$$\widetilde{\mathcal{L}}(\Theta) = \underbrace{\sum_{i=1}^{N} \log \sum_{s=1}^{S} \pi_s \cdot f(y_i|x_i, \theta_s)}_{\text{Log-Likelihood}} - \underbrace{\sum_{s=1}^{S} \lambda_s \sum_{j=1}^{p} |\beta_{sj}|}_{\text{Penalty}} . \tag{36}$$

- **MIXALS**:

$$\widetilde{\mathcal{L}}(\Theta) = \underbrace{\sum_{i=1}^{N} \log \sum_{s=1}^{S} \pi_s \cdot f(y_i|x_i, \theta_s)}_{\text{Log-Likelihood}} - \underbrace{\sum_{s=1}^{S} \lambda_{sj} \sum_{j=1}^{p} \hat{w}_{sj} |\beta_{sj}|}_{\text{Penalty}} . \tag{37}$$

A highly desirable property of regularization algorithms is shrinking the influence of the least important variables. Or in case of algorithms with variable selection abilities such as the elastic net, we wish to select the optimal subset of variables while excluding the non-influential variables from the model by shrinking their effect down to zero. The elastic net and lasso approaches both have feature selection abilities while the regular and ridge approach yield non-zero estimates for all coefficients. Secondly, besides selecting the correct subset of variables we wish to obtain accurate estimate of the effects of the non-zero coefficients. Thirdly, another fundamental challenge in mixture modeling is determining the optimal amount of components to describe the data. The number of components is in general unknown and must be extracted from the data. Hence, the ability to select the correct amount of components is also investigated.

In short, we compare performance of the different modeling approaches with a simulation in which three aspects are examined:

- Selection of correct subset of variables
- Accuracy of non-zero coefficient estimates
- Recovery of true component amounts

We specify the following general 2-component finite mixture form to generate a response variable y

$$\pi \cdot \phi \left(y_1; x^\top \beta_1, \sigma^2 \right) + (1 - \pi) \cdot \phi \left(y_2; x^\top \beta_2, \sigma^2 \right) \quad (38)$$

with $\sigma^2 = 1$. Three different prior probabilities are tested, $\pi_1 = \{0.15, 0.3, 0.6\}$ implying $\pi_2 = 1 - \pi_1 = \{0.85, 0.7, 0.4\}$. The covariates x are generated from a multivariate normal distribution with mean 0, variance 1 and a correlation structures ρ_{ij} such that

$$x \sim \mathcal{N} \left(\mu = 0, \sigma^2 = 1 \right) \quad (39)$$

with $\rho_{ij} = cor(x_i, x_j) = 0.6^{|i-j|}$.

Next, we use Equation 38 to define two different models M_1 and M_2 to simulate a set of data. The component specifications of both models are given in Table 1. The first model, M_1, contains $p = 10$ covariates of which 5 zero coefficients in component 1 and 6 in component 2. The second model, M_2, presents a higher-dimensional and more realistic variable selection problem and includes $p = 25$ covariates. Component 1 contains 10 zero coefficients while component 2 contains 15 zero coefficients. Hence, in both models component 2 contains more zero coefficients than component 1. In each case, a sample size of $N = 100$ observations is used.

A widely used performance metric is the hit-rate. Hit-rate is simply the ratio of correct predictions to the total of observations. Indeed, this is an intuitive and effective measure when dealing with symmetric data. However, in case of an unbalanced class distribution the hit-rate may fail to provide a proper indication of performance. In order to compare the detection of true zero coefficients we consider the following metrics; precision (specificity), recall (sensitivity), and F1 score. We define precision as the ratio of correctly estimated zero coefficients (true positives) to the total estimated number of zero coefficients (true positives and false positives) such that

$$\text{Precision} = \frac{\text{TP}_0}{\text{TP}_0 + \text{FP}_0}. \quad (40)$$

Next, recall is given by the ratio of correctly estimated zero coefficients to the true number of zero coefficients defined as

$$\text{Recall} = \frac{\text{TP}_0}{\text{TP}_0 + \text{FN}_0}. \quad (41)$$

The F1 score (Van Rijsbergen, 1979) is the weighted average of precision and recall given by

$$\text{F1} = \frac{2 \cdot (\text{Recall} \times \text{Precision})}{(\text{Recall} + \text{Precision})}. \quad (42)$$

A flawless performance would results in a ratio of 1 for all three metrics. By combining the precision and recall we can take true and false positives and negatives into account simultaneously. This allows us to quickly compare the subset selection performance of each model and specification with a single metric. Each scenario is repeated for 100 iterations.

The component-wise results for the detection of zero coefficients are reported in Table 2. We exclude the MIXREG and MIXRDG in this part of the simulation as they both do not have variable selection abilities. The numbers are rounded to four decimal points. In case of an unbalanced mix of the component sizes for $\pi_1 = 0.15$, the first component contains a small amount of observations. In general, the performance drops when the amount of observations in the second component decreases. There is no universal best method. In case of the higher-dimensional variable

selection problem in M_2 MIXREG would fail to obtain a solution for $\pi_1 = 0.15$. In this scenario 25 coefficients are to be estimated based on approximately 15 observations which is not possible with the regular likelihood approach. All other models have no estimation issues in this scenario when $p > N$. This is a very attractive property of the penalized likelihood approaches.

There is no indication of a single overall superior shrinkage algorithm in this part of simulation. In general we find that MIXNET performs well when the amount of observations in component 2 is larger. The lasso based models perform better when the amount of observations in component 2 decreases. In the higher dimensional problem in model M_2 MIXALS is the best method when component 2 contains little observations. This situation is the most challenging in terms of selecting the correct subset of variables.

Table 1 Simulation model specifications M_1 and M_2.

Parameters	**Model M**$_1$ $(p = 10)$	**Model M**$_2$ $(p = 25)$
$\beta_{s=1}$	(2, -0.8, 1, 0, 0, 1.2, 0, 0, 1.2, 0)	(0, 2, -24, 1, 0, 3, 15, 22, -5, 28, 0, 0, 14, 29, 0, 0, 19, -6, 0, 21, 31, 0, 0, -19, 0)
$\beta_{s=2}$	(0, 0, 0, 1, 2, 0, 0, -1.5, 0, 1.2)	(-6, 0, 0, 15, 0, 0, 0, 8, 0, 22, 0, -3, 0, 17, 0, 0, 5, 0, 13, 0, 0, -19, 0, 0, 1)
ρ_{ij}	$0.6^{\lvert i-j\rvert}$	$0.6^{\lvert i-j\rvert}$
π_1	0.15, 0.3, 0.6	0.15, 0,3, 0.6

Table 2 Detection of zero coefficients based on 100 simulation repetitions.

	Model M$_1$ $(p = 10)$						**Model M**$_2$ $(p = 25)$					
	Component 1			Component 2			Component 1			Component 2		
Method	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
$\pi_1 = 0.15$												
MIXNET$_5$	.8202	.9400	**.8745**	.8333	1.000	**.9091**	.7271	.8473	.7809	.9487	.9832	.9655
MIXNET$_9$	.8220	.9340	.8714	.8333	1.000	**.9091**	.7327	.8513	**.7861**	.9493	.9853	.9668
MIXLAS	.8430	.7640	.7846	.8333	1.000	**.9091**	.7790	.7933	.7829	.9995	.9432	**.9697**
MIXALS	.8163	.9340	.8698	.8290	.9960	.9048	.7315	.8513	.7854	.9483	.9842	.9658
$\pi_1 = 0.3$												
MIXNET$_5$	.8188	.9760	.8898	.8246	.9920	.9004	.7886	.8820	.8307	.9292	.9268	.9256
MIXNET$_9$	.8247	.9740	.8925	.8289	1.000	.9062	.8013	.8900	**.8417**	.9253	.9347	**.9289**
MIXLAS	.8380	.9040	.8661	.8307	.9980	**.9067**	.8372	.8020	.8153	.9667	.8489	.9006
MIXALAS	.8303	.9820	**.8993**	.8333	1.000	.9091	.7884	.8740	.8265	.9178	.9053	.9096
$\pi_1 = 0.6$												
MIXNET$_5$	.8217	.9900	.8978	.8293	.982	.8980	.9073	.8826	.8904	.8343	.9067	.8675
MIXNET$_9$	.8217	.9840	.8954	.8259	.994	**.9019**	.8936	.8605	.8731	.8170	.8967	.8536
MIXLAS	.8260	.9880	**.8996**	.8266	.970	.8912	.9471	.7953	.8610	.8539	.8373	.8427
MIXALS	.8213	.9880	.8962	.8292	.986	.9003	.9038	.8868	**.8920**	.8322	.9153	**.8704**

Best results per component marked in bold.

Next, the same simulation setup is used to study the accuracy of the non-zero coefficient estimates. In order to compare the behavior of the tested models we look at several error metrics. We consider the mean absolute error (MAE), mean squared error (MSE) and root mean squared error (RMSE) of the coefficient estimates. These metrics are calculated based on the p true coefficient estimates β as reported in Table 1 and the p estimated coefficient estimates $\widehat{\beta}$ resulting from the tested models. As reported, we have $p = 10$ covariates in model M_1 and $p = 25$ covariates in model M_2. The used metrics are formulated as

$$\begin{aligned} \text{MAE} &= \frac{1}{p}\sum_{j=1}^{p}|\beta_j - \widehat{\beta}_j|, \\ \text{MSE} &= \frac{1}{p}\sum_{j=1}^{p}\left(\beta_j - \widehat{\beta}_j\right)^2, \\ \text{RMSE} &= \sqrt{\frac{1}{p}\sum_{j=1}^{p}\left(\beta_j - \widehat{\beta}_j\right)^2}. \end{aligned} \tag{43}$$

Every iteration of the simulation results in an error value. Hence, for a more convenient comparison of the models we summarize the metrics by reporting the mean over the 100 simulation repetitions. This yields a single value for each used error metric. This means we report the average of the errors over all $n = 100$ iterations such that

$$\begin{aligned} \overline{\text{MAE}} &= \frac{1}{n}\sum_{i=1}^{n}\text{MAE}_i, \\ \overline{\text{MSE}} &= \frac{1}{n}\sum_{i=1}^{n}\text{MSE}_i, \\ \overline{\text{RMSE}} &= \frac{1}{n}\sum_{i=1}^{n}\text{RMSE}_i. \end{aligned} \tag{44}$$

The component-wise results for the accuracy of non-zero coefficient estimates are reported in Table 3. The numbers are rounded to four decimal points. Figure 1 visualizes the average error and standard deviation over the iterations for model M_1 and Figure 2 for model M_2. The standard deviation of each metric over the repetitions are shown graphically with an error bar. Note that the scales of the error on the y-axis differ per component and prior. RDGMIX is clearly the least accurate method in both simulation models. MIXREG performs well in estimating the coefficients in all cases for the easier problem M_1. The penalized likelihood approaches come with the price of introducing a bias in the estimates which is reflected in this simulation. The difference between the performance of MIXREG and the penalized methods decrease when the problem becomes more difficult. That is, when the amount of observations belonging to component 2 decreases. Note that the range of the error on the y-axis differs per choice of prior π_s. This finding proves even more substantive when the variable selection problem is complicated further by increasing the amount of zero- and non-zero covariates from $p = 10$ to $p = 25$ in M_2 while still using $N = 100$ observations. In general, the deviation of the MIXREG is now considerably larger than the penalized approaches. In this case we find that performance of the lasso and elastic net models approach the MIXREG. Again, MIXALS provides the most accurate solution in the most difficult case for $\pi_1 = 0.6$. This exceptional performance compared to all other tested models is likely explained by the fact that the adaptive lasso possesses the oracle property as discussed in Section 1.3.2 (Zou, 2006).

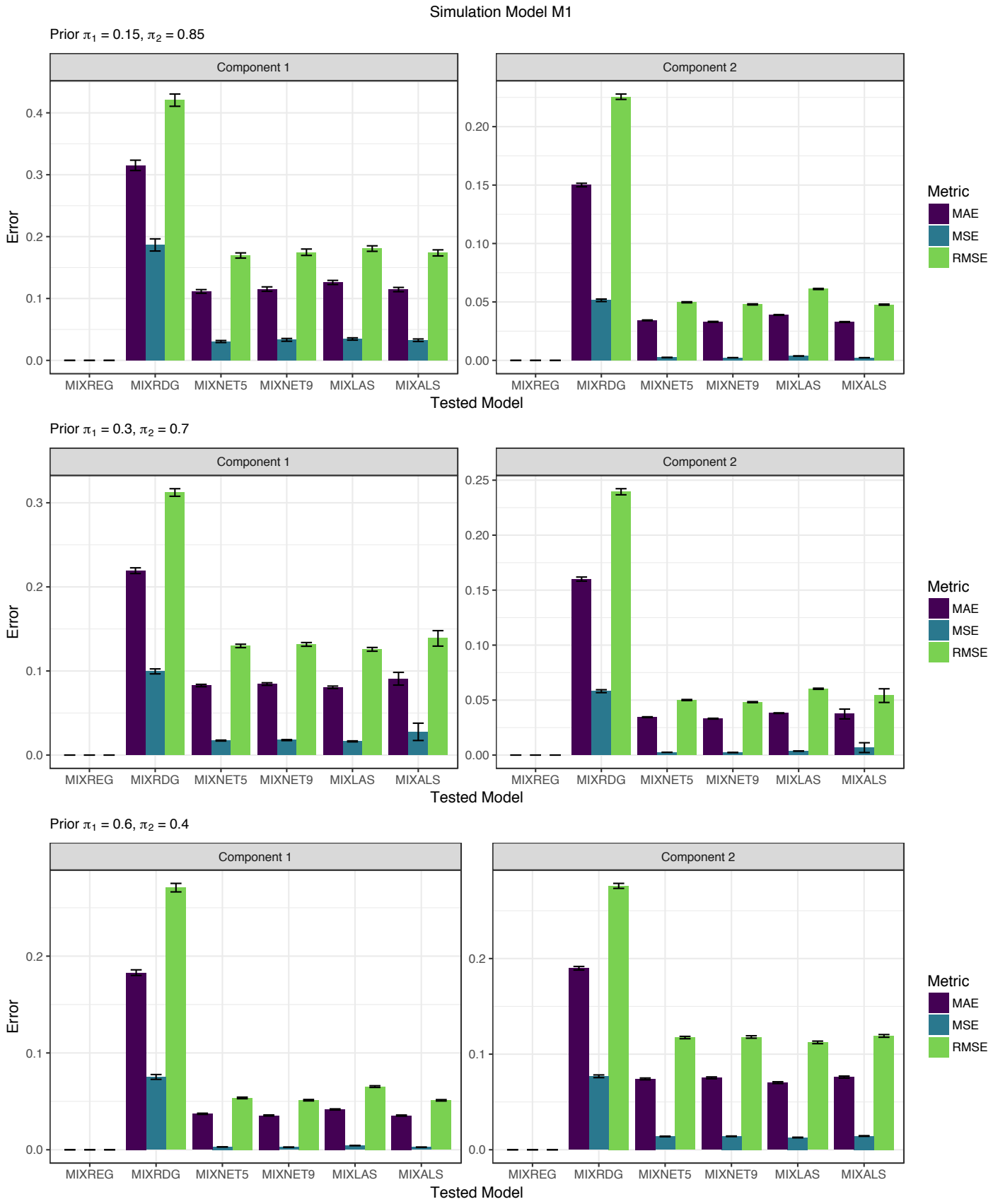

Figure 1 Comparison of the MAE, MSE and RMSE of the coefficients estimated per component by each tested mode in simulation model M_1. Black error bars indicate the standard deviation.

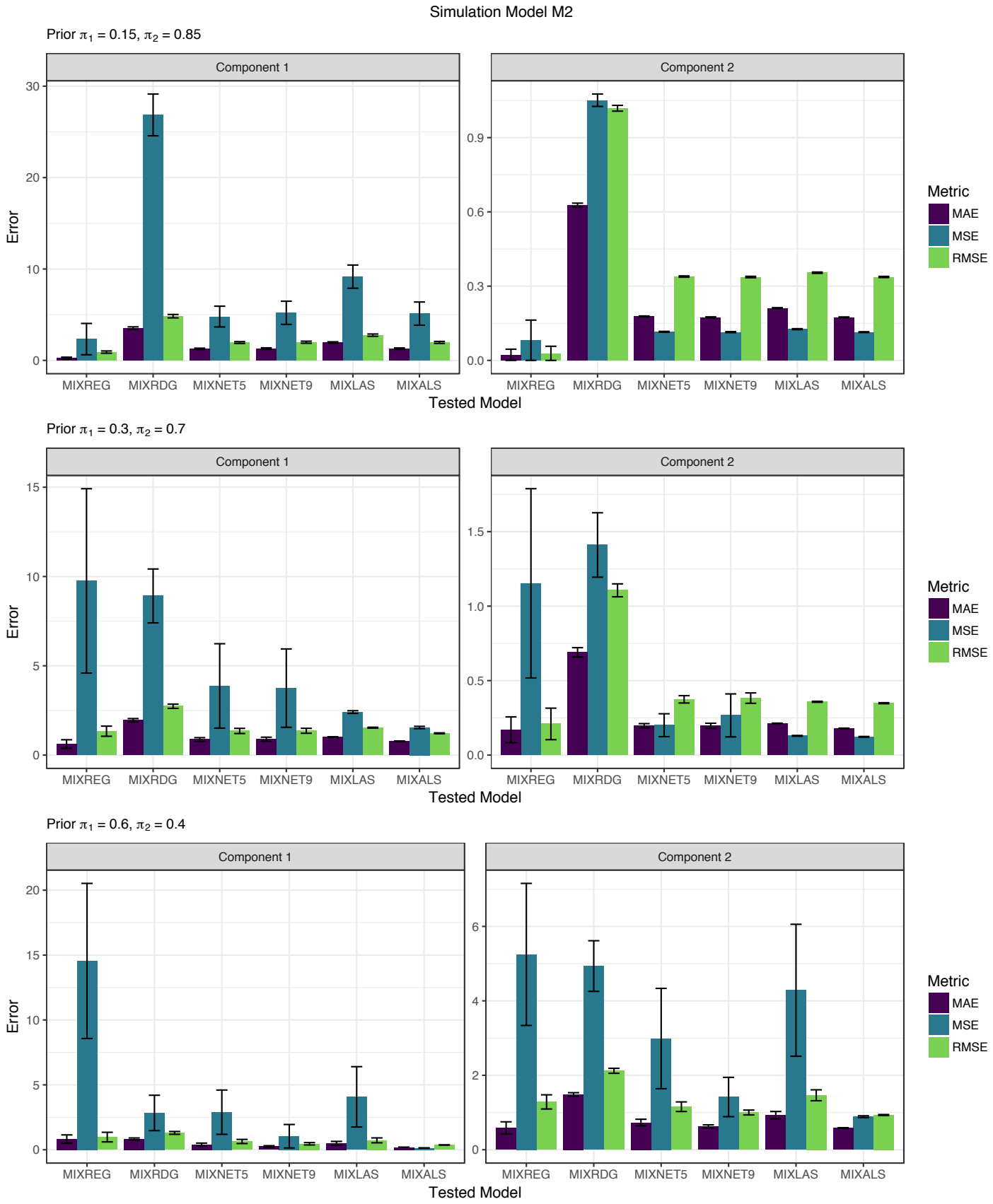

Figure 2 Comparison of the MAE, MSE and RMSE of the coefficients estimated per component by each tested model in simulation model M_2. Black error bars indicate the standard deviation.

Table 3 accuracy of non-zero coefficients based on 100 simulation repetitions.

	Model $\mathbf{M}_1$ ($p = 10$)						**Model $\mathbf{M}_2$** ($p = 25$)					
	Component 1			Component 2			Component 1			Component 2		
Method	$\overline{\text{MAE}}$	$\overline{\text{MSE}}$	$\overline{\text{RMSE}}$	$\overline{\text{MAE}}$	$\overline{\text{MSE}}$	$\overline{\text{RMSE}}$	$\overline{\text{MAE}}$	$\overline{\text{MSE}}$	$\overline{\text{RMSE}}$	$\overline{\text{MAE}}$	$\overline{\text{MSE}}$	$\overline{\text{RMSE}}$
$\pi_1 = 0.15$												
MIXREG	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.2524**	**2.3295**	**0.9078**	**0.0226**	**0.0814**	**0.0285**
MIXRDG	0.3152	0.1865	0.4205	0.1501	0.0514	0.2256	3.5306	26.8569	4.8393	0.6277	1.0511	1.0186
MIXNET_5	0.1114	0.0305	0.1695	0.0342	0.0025	0.0497	1.2665	4.7945	1.9511	0.1779	0.1154	0.3387
MIXNET_9	0.1152	0.0332	0.1748	0.0330	0.0023	0.0479	1.2928	5.2074	1.9865	0.1741	0.1143	0.3367
MIXLAS	0.1259	0.0346	0.1807	0.0389	0.0038	0.0611	1.9363	9.1635	2.7603	0.2111	0.1262	0.3542
MIXALS	0.1145	0.0326	0.1737	0.0328	0.0023	0.0477	1.9363	5.1196	1.9652	0.1741	0.1141	0.3370
$\pi_1 = 0.3$												
MIXREG	**0.0000**	**0.0000**	**0.0000**	0.0000	**0.0000**	**0.0000**	0.6190	9.7534	1.3362	**0.1701**	1.1529	**0.2091**
MIXRDG	0.2193	0.0995	0.3123	0.1600	0.0581	0.2395	1.9523	8.9116	2.7358	0.6898	1.4105	1.1061
MIXNET_5	0.0827	0.0173	0.1298	0.0345	0.0025	0.0501	0.8724	3.8717	1.3509	0.1969	0.2004	0.3742
MIXNET_9	0.0845	0.0178	0.1316	0.0331	0.0023	0.0481	0.8888	3.7490	1.3590	0.1964	0.2664	0.3825
MIXLAS	0.0806	0.0163	0.1257	0.0381	0.0037	0.0603	1.0077	2.4079	1.5330	0.2125	0.1292	0.3578
MIXALAS	0.0908	0.0276	0.1388	0.0374	0.0068	0.0541	**0.7759**	**1.5442**	**1.2197**	0.1790	**0.1225**	0.3483
$\pi_1 = 0.6$												
MIXREG	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.8306	14.5471	0.9749	0.5863	5.2478	1.2853
MIXRDG	0.1830	0.0752	0.2708	0.1899	0.0769	0.2760	0.8307	2.8373	1.2989	1.4831	4.9359	2.1207
MIXNET_5	0.0371	0.0029	0.0535	0.0741	0.0139	0.1174	0.3936	2.8946	0.6387	0.7319	2.9880	1.1563
MIXNET_9	0.0354	0.0027	0.0512	0.0752	0.0141	0.1180	0.2518	1.0416	0.4594	0.6266	1.4171	1.0005
MIXLAS	0.0417	0.0043	0.0653	0.0702	0.0128	0.1123	0.4948	4.0784	0.7244	0.9285	4.2848	1.4632
MIXALS	0.0353	0.0027	0.0511	0.0760	0.0144	0.1191	**0.1911**	**0.1379**	**0.3691**	**0.5834**	**0.8880**	**0.9342**

Best results per component marked in bold.

Lastly, we consider the performance of the models in terms of recovering the true number of components in the mixture, this is also known as order selection. We define a S-component Gaussian mixture model where we can vary the number of components as

$$\sum_{s=1}^{S} \pi_s \cdot \phi(y_s, x^\mathsf{T}\beta_s, \sigma^2), \tag{45}$$

with $\sigma^2 = 1$. We generate random priors $\pi_s = \{\pi_1, \ldots, \pi_S\}$ by splitting the value 1 into S parts based on a binomial distribution with a restriction on π_s to ensure $\sum_{s=1}^{S} \pi_s = 1$. The p regression coefficients $\beta_s = \{\beta_{s1}, \ldots, \beta_{sp}\}$ per component are drawn from a uniform distribution such that

$$\beta_{sj} \sim \mathcal{U}(-3, 3) \;\; \forall s = 1, \ldots, S, \; \forall j = 1, \ldots p. \tag{46}$$

The covariates x are generated from a multivariate normal distribution with mean 0, variance 1 and correlation structure $\rho_{ij} = 0.6^{|i-j|}$ as described above. In order to resemble a problem where variable selection is of importance we set all regression coefficients with an absolute value smaller than 0.5 to zero. This results in a varying amount of zero-coefficients per component, generally zero to three. We use this framework to simulate a mixture with varying component amounts $S = \{2, 4, 6, 8, 10, 15\}$ and test each modeling approach. A stepwise component selection procedure is used. In short, we fit each model starting with the following initial amount of components $\widehat{S} = \{1, 2, 5, 10, 15\}$ and select the best solution based on the BIC measure. To somewhat decrease computational intensity a limit of 100 iterations is used in the EM algorithm. If the prior probability of a component falls below the value of 0.05 it is removed from the solution after which the EM algorithm continues fitting with $\widehat{S} - 1$ components. This allows the algorithms to perform component selection. Consequently, we compare the performance of the models in terms of selecting the true amount of components present based on the data. Again 100 repetitions are performed. We report the average amount of determined components present in the mixture $\overline{\widehat{S}} = \frac{1}{n}\sum_{i=1}^{n} \widehat{S}$ to examine the solutions over the repetitions. In addition, we look at the hit-rate of the determined amount and true amount of present components, $\widehat{S} = S$. Lastly, the ratio of converged solutions over the repetitions is also reported. A ratio of 1 indicates all solutions over the $n = 100$ repetitions converged.

The results of the component selection simulation are reported in Table 4. The numbers are rounded to two decimal points. In general we observe accurate performance when the amount of components in the mixture is small. When the amount of true components in the mixture grows, all approaches prove ineffective in detecting the correct amount. The performance of all models drop in each step from $S = 6$ onwards. This is possibly due to the fact that the number of observations per component decreases when S increases. Very noticeable is the accuracy of the adaptive lasso model in this aspect of the simulation. MIXLAS yields the most accurate result when the amount of true components in the mixture increases. For $S = 8, 10$ and 15 it is the only method that manages to approach the true component amount while all other techniques yield worse results. Even the adaptive lasso is clearly outperformed by the normal lasso in this comparison. Again, we find that MIXRDG does not perform well compared to the other methods. Interestingly, MIXRDG is the only model that manages to obtain good convergence over all simulation repetitions.

Taking all three aspects into account, both the mixture modeling approaches with a penalized likelihood based on the lasso and the elastic net prove to yield good performance in comparison with a traditional mixture model. There is no universally dominant method in this simulation. The added value of the extension of feature selection abilities is most evident when a higher-dimensional variable selection problem is of interest. In this situation, the extended models clearly outperform the traditional mixture model. We find that both approaches based on the lasso yield good results, closely followed by the models based on the elastic net. In terms of selecting the correct subset of variables while also providing an accurate estimate of non-excluded variables in the mixture component, MIXALS is the optimal method. Naturally, this is a golden combination when dealing with a regression problem where heterogeneity and variable selection are of both of importance which is often the case. When determining the optimal amount of components in larger mixtures we find MIXLAS is the only model that performs well.

Table 4 Recovery of true component amounts based on 100 simulation repetitions.

Method	$\widehat{\overline{S}}$	Hit-rate	Converged	**Method**	$\widehat{\overline{S}}$	Hit-rate	Converged	**Method**	$\widehat{\overline{S}}$	Hit-rate	Converged
$S = 2$				$S = 4$				$S = 6$			
MIXREG	2.16	0.84	0.98	MIXREG	**4.00**	**1.00**	**1.00**	MIXREG	5.12	0.12	0.64
MIXRDG	**2.00**	**1.00**	**1.00**	MIXRDG	**4.00**	**1.00**	**1.00**	MIXRDG	4.68	0.02	**1.00**
MIXNET_5	2.04	0.96	**1.00**	MIXNET_5	**4.00**	**1.00**	0.40	MIXNET_5	5.16	0.16	0.34
MIXNET_9	**2.00**	**1.00**	0.99	MIXNET_9	**4.00**	**1.00**	0.46	MIXNET_9	5.26	0.30	0.36
MIXLAS	2.06	0.96	0.94	MIXLAS	4.50	0.50	0.62	MIXLAS	**6.72**	**0.42**	0.28
MIXALS	2.04	0.96	**1.00**	MIXALS	**4.00**	**1.00**	0.50	MIXALS	5.14	0.16	0.38
$S = 8$				$S = 10$				$S = 15$			
MIXREG	5.12	0.00	0.38	MIXREG	4.18	0.00	0.32	MIXREG	4.46	0.00	0.34
MIXRDG	4.66	0.00	**0.96**	MIXRDG	4.56	0.00	**1.00**	MIXRDG	4.66	0.00	**1.00**
MIXNET_5	5.20	0.00	0.49	MIXNET_5	4.64	0.00	0.50	MIXNET_5	5.06	0.00	0.57
MIXNET_5	5.26	0.00	0.58	MIXNET_9	4.62	0.00	0.58	MIXNET_9	5.02	0.00	0.52
MIXLAS	**8.48**	**0.30**	0.46	MIXLAS	**9.36**	**0.90**	0.38	MIXLAS	**9.58**	**0.06**	0.38
MIXALS	5.10	0.00	0.37	MIXALS	4.62	0.00	0.66	MIXALS	5.00	0.00	0.65

Best results per scenario marked in bold.

3 REFERENCES

G. M. Allenby and P. E. Rossi. Marketing models of consumer heterogeneity. *Journal of Econometrics*, 89 (1):57–78, 1998.

S. T. Apple. Deep learning for siri's voice: On-device deep mixture density networks for hybrid unit selection synthesis. *Apple Machine Learning Journal*, 1 (4), 2017.

A. Barron, L. Birgé, and P. Massart. Risk bounds for model selection via penalization. *Probability theory and related fields*, 113(3):301–413, 1999.

L. Breiman. Better subset regression using the nonnegative garrote. *Technometrics*, 37(4):373–384, 1995.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B*, pages 1–38, 1977.

W. S. DeSarbo and W. L. Cron. A maximum likelihood methodology for clusterwise linear regression. *Journal of classification*, 5(2):249–282, 1988.

J. Diebolt and C. P. Robert. Estimation of finite mixture distributions through bayesian sampling. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 363–375, 1994.

J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360, 2001.

J. Fan and J. Lv. A selective overview of variable selection in high dimensional feature space. *Statistica Sinica*, 20(1):101, 2010.

J. Fan, H. Peng, et al. Nonconcave penalized likelihood with a diverging number of parameters. *The Annals of Statistics*, 32(3):928–961, 2004.

L. E. Frank and J. H. Friedman. A statistical view of some chemometrics regression tools. *Technometrics*, 35(2):109–135, 1993.

J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

J. Friedman, T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu. Discussion of boosting papers. Citeseer, 2004.

J. Friedman, T. Hastie, and R. Tibshirani. glmnet: Lasso and elastic-net regularized generalized linear models. *R package version*, 1(4), 2009.

J. C. Gower. Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika*, 53(3-4):325–338, 1966.

C. Hennig. Identifiablity of models for clusterwise linear regression. *Journal of Classification*, 17(2):273–296, 2000.

A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

P. J. Huber et al. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1): 73–101, 1964.

G. James, D. Witten, T. Hastie, and R. Tibshirani. *An introduction to statistical learning*, volume 7. Springer, 2013.

M. I. Jordan and L. Xu. Convergence results for the em approach to mixtures of experts architectures. *Neural networks*, 8(9):1409–1431, 1995.

A. Khalili and J. Chen. Variable selection in finite mixture of regression models. *Journal of the American Statistical Association*, 102(479):1025–1038, 2007.

B. G. Lindsay. Mixture models: theory, geometry and applications. In *NSF-CBMS regional conference series in probability and statistics*, pages i–163. JSTOR, 1995.

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

G. J. McLachlan and K. E. Basford. *Mixture models: Inference and applications to clustering*, volume 84. Marcel Dekker, 1988.

B. Muthén and K. Shedden. Finite mixture modeling with mixture outcomes using the em algorithm. *Biometrics*, 55(2):463–469, 1999.

K. Pearson. Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London. A*, 185:71–110, 1894.

S. Rosset, J. Zhu, and T. Hastie. Boosting as a regularized path to a maximum margin classifier. *Journal of Machine Learning Research*, 5(Aug):941–973, 2004.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

A. N. Tikhonov, V. I. Arsenin, and F. John. *Solutions of ill-posed problems*, volume 14. Winston Washington, DC, 1977.

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

J. W. Tukey. A survey of sampling from contaminated distributions. *Contributions to probability and statistics*, 2:448–485, 1960.

M. Tuma and R. Decker. Finite mixture models in market segmentation: a review and suggestions for best practices. *Electronic Journal of Business Research Methods*, 11(1):2–15, 2013.

C. Van Rijsbergen. Information retrieval. *Dept. of computer science, University of Glasgow*, 14, 1979.

M. Wedel and W. S. DeSarbo. A mixture likelihood approach for generalized linear models. *Journal of Classification*, 12(1):21–55, 1995.

M. Wedel and W. A. Kamakura. *Market segmentation: Conceptual and methodological foundations*, volume 8. Springer Science & Business Media, 2012.

H. Zou. The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476): 1418–1429, 2006.

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67 (2):301–320, 2005.