Variable Selection for Clustering and Classification*

Jeffrey L. Andrews† and Paul D. McNicholas

Department of Mathematics & Statistics, University of Guelph, Guelph, ON, Canada, N1G2W1.

Abstract

As data sets continue to grow in size and complexity, effective and efficient techniques are needed to target important features in the variable space. Many of the variable selection techniques that are commonly used alongside clustering algorithms are based upon determining the best variable subspace according to model fitting in a stepwise manner. These techniques are often computationally intensive and can require extended periods of time to run; in fact, some are prohibitively computationally expensive for high-dimensional data. In this paper, a novel variable selection technique is introduced for use in clustering and classification analyses that is both intuitive and computationally efficient. We focus largely on applications in mixture model-based learning, but the technique could be adapted for use with various other clustering/classification methods. Our approach is illustrated on both simulated and real data, highlighted by contrasting its performance with that of other comparable variable selection techniques on the real data sets.

Keywords: Classification; Cluster analysis; High-dimensional data; Mixture models; Model-based clustering; Variable selection.

1 Introduction

Variable selection is an important feature of many types of statistical analyses, including clustering and classification. The use of variable selection techniques can facilitate both model fitting and the interpretation of results. With the continued growth in data dimensionality, the importance of efficient variable selection techniques is increasing. In this paper, we put forward a flexible variable selection technique that can be used in unsupervised, semi-supervised, or fully supervised classification contexts. We focus largely on the unsupervised format (i.e., clustering) and specifically the usage of model-based techniques to guide the algorithm. However, the algorithm's applicability under a classification format will be briefly illustrated in Section 4.

Model-based clustering is often used on high-dimensional data sets, such as those found in the field of bioinformatics. Though clustering on very large data sets is possible, it is difficult to execute for a number of reasons. The most obvious reason is time: as the dimensionality of the data increases, the number of parameters requiring estimation increases, often in a quadratic fashion. Of more importance, however, is that the human brain (accustomed to three-dimensions visually, plus a few other senses) is not prepared to understand dimensionality that can run well into the thousands. Thus, to facilitate interpretation of high-dimensional data sets, determining which variables are most active in cluster formation is important. A final consideration is the cost in creating high-dimensional data sets, which can be enormous; knowing which variables are important for differentiating between groups-of-interest can save both time and money.

Although algorithm efficiency is certainly a worthwhile reason in its own right, variable selection techniques can drastically improve clustering performance. This is achieved through eliminating noisy variables that can cloud the clustering algorithm's ability to distinguish groups. Unfortunately, variable selection techniques do not necessarily improve clustering performance; it is, therefore, important that an inferior reduced-variable solution is not chosen over a solution on the full variable set (cf. Section 5).

In Section 2, we conduct a short review of comparable variable selection techniques and other relevant background material. Then we discuss our methodology (Section 3), before running simulations (Section 4) and real-data examples (Section 5). Finally, we conclude with a summary and suggestions for future work (Section 6).

2 Background

A number of dimensionality reduction techniques are available to researchers interested in clustering data sets. For the purposes of microarray data sets, the `select-genes` procedure from the EMMIX-GENE software (McLachlan et al., 2002) fits multi-component mixture models to each variable and then calculates the likelihood ratio test statistic between these and the one-component model. Unfortunately, fitting mixture models to each individual variable is time consuming and, by relying on random initializations, `select-genes` can be inconsistent.

Another variable selection technique is given by Raftery and Dean (2006), whereby multiple models from the MCLUST family are compared using approximate Bayes factors (Kass and Raftery, 1995). This variable selection technique is readily available via the `clustvarsel` package (Dean and Raftery, 2006) in R (R Development Core Team, 2012). However, because the number of parameters that require estimation in some of the MCLUST models is quadratic in data-dimensionality, the `clustvarsel` package can be very slow in high-dimensions. Furthermore, the application of `clustvarsel` can sometimes lead to inferior results when compared to the use of `mclust` alone (cf. McNicholas and Murphy, 2008). A related approach, denoted `selvarclust`, is taken by Maugis et al. (2009), where the assumptions on the role of variables are relaxed with the potential benefit of avoiding the over-penalization of independent variables.

In addition to these procedures, a number of implicit and explicit variable selection procedures are built into model-based clustering algorithms. Implicit variable selection procedures include approaches such as mixtures of factor analyzers (cf. Ghahramani and Hinton, 1997; Tipping and Bishop, 1999; McLachlan and Peel, 2000; McNicholas and Murphy, 2008, 2010; Andrews and McNicholas, 2011a,b; Montanari and Viroli, 2010; Viroli, 2010). An explicit dimensionality reduction approach is taken in some recent work by Scrucca (2010) and Bouveyron and Brunet (2012); the latter also gives a summary of other work in the area of dimensionality reduction with respect to clustering. Outside of model-based methods, Witten and Tibshirani (2010) recently introduced a dimensionality reduction technique that is based on k-means; we provide a comparison to this technique in Section 5.6.

For the purposes of variable selection within a clustering context, the desire is to find variables that show differentiation between the *a priori* unknown groups and eliminate variables that do not. The variable selection method introduced herein seeks precisely this, and is flexible enough to implement using a variety of clustering/classification techniques.

3 Methodology

3.1 Introduction

Variable selection for clustering and classification (VSCC) is intended to find the variables that simultaneously minimize the 'within-group' variance and maximize the 'between-group' variance. The combination of these two criteria will give variables that best show separation between the desired groups. Note that the

within-group variance for each variable $j = 1, \ldots, p$ can be written as

$$\mathcal{W}_j = \frac{\sum_{g=1}^{G} \sum_{i=1}^{n} z_{ig} (x_{ij} - \mu_{gj})^2}{n},$$

where x_{ij} is observation i on variable j, μ_{gj} is the mean of variable j in group g, n is the number of observations, and z_{ig} is a group membership indicator variable defined so that

$$z_{ig} = \begin{cases} 1 & \text{if observation } \mathbf{x}_i = (x_{i1}, \ldots, x_{ip}) \text{ belongs to cluster } g, \\ 0 & \text{otherwise.} \end{cases}$$

The leftover variance within variable j not accounted for by $\mathcal{W}_j$, or $\sigma_j^2 - \mathcal{W}_j$ in the common notation, is then a measure of the variance between groups. In general, calculation of this value will be necessary. However, if the data have been standardized to have equal variance across variables, then any variable minimizing the within-group variance is also maximizing the leftover variance.

The VSCC method utilized in this article will be applied to data that have been standardized to have mean 0 and variance 1 and, as such, calculation of the $\mathcal{W}_j$ is sufficient. In addition to the variance calculations, our method uses the correlation between variables; we let ρ_{ij} denote the correlation between variables i and j. The actual implementation of VSCC procedures depends on the form of the data; in our analyses, we consider examples where no memberships are known (clustering) as well as where some observations have known membership (classification). Specifics regarding the implementation of the VSCC algorithm under each format can be found in Sections 3.5 and 3.6. In the sections that immediately follow, we motivate and describe the VSCC algorithm.

3.2 A Motivating Example

VSCC will proceed in a step-wise fashion after calculating the within-group variances. The first variable selected is the variable with the minimum $\mathcal{W}_j$. One way to select from the remaining variables is by using simple, user-specified thresholding. For example, by sorting the $\mathcal{W}_j$ in ascending order we could consider each variable in a step-wise manner and select those variables with $\mathcal{W}_j$ less than some value w where all $|\rho_{jr}|$ are also less than some value c, $\forall\, r \in V$; here, V is the set of previously selected variables. While this approach could be useful, it requires the user to adjust the algorithm to maximize its effectiveness.

An additional concern behind this type of selection criterion can be shown via a simple example. Consider a three-dimensional data set where the first variable minimizes $\mathcal{W}_j$ and so is already selected. Suppose that the remaining two variables can be summarized as follows.

- Variable 2: $\mathcal{W}_2 = 0.6$ and $|\rho_{12}| = 0.75$.

- Variable 3: $\mathcal{W}_3 = 0.2$ and $|\rho_{13}| = 0.75$.

Suppose we simply use the thresholds described previously. In the current example, if the correlation threshold was set at $c = 0.70$ then both variables would be considered equally 'bad' and neither would be selected. However, if the correlation threshold was set at $c = 0.80$, and assuming both that $|\rho_{23}| < 0.8$ and $w > 0.6$, then both variables would be selected. Under the argument that within-group variance is our primary concern and correlation is a secondary concern, we propose that, in this scenario, retaining Variable 3 and eliminating Variable 2 would be desirable. To this end, we need to go beyond simple, user-specified thresholding.

3.3 The VSCC Method

We have illustrated a desire for a sliding correlation threshold that is more forgiving for small values of $\mathcal{W}_j$ and more stringent for larger values. Thus, we seek to define a relationship between the within-group

variance and between-variable correlation that properly expresses this goal. As a first attempt, we consider a linear relationship between the two quantities. Let V represent the space of currently selected variables, then we select variable j if for all $r \in V$,

$$|\rho_{jr}| < 1 - \mathcal{W}_j.$$

Other potential relationships will be discussed shortly, but utilizing this relationship we can write the VSCC algorithm as follows:

1. Calculate within-group variances $\mathcal{W}_j$.

2. Sort $\mathcal{W}_j$ in ascending order, denote this sorted list $\mathbf{W}_s$.

3. $\mathcal{W}_1$ minimizes $\mathbf{W}_s$ and is automatically selected and placed into the set of selected variables V. Set count $k = 2$.

4. If $|\rho_{kr}| < 1 - W_k$, for all $r \in V$, variable $s = k$ is placed into V.

5. If $k < p$, set $k = k + 1$ and return to Step 4. Else end algorithm.

The linear relationship defined in Step 4 of the VSCC algorithm might be too strong a criterion. For instance, a variable with within-group variation of 0.25 and correlation of 0.76 with one of the previously selected variables would be rejected. Given the interval that correlation values (and the $\mathcal{W}_j$ when the data are standardized) will lie on, a simple fix is to consider relationships of order greater than one (Table 1); a visualization of these criteria is given in Figure 1.

Table 1: List of variance-correlation relationships considered for implementation into Step 4 of the VSCC algorithm.

Linear	$|\rho_{kr}| < 1 - W_k$
Quadratic	$|\rho_{kr}| < 1 - W_k^2$
Cubic	$|\rho_{kr}| < 1 - W_k^3$
Quartic	$|\rho_{kr}| < 1 - W_k^4$
Quintic	$|\rho_{kr}| < 1 - W_k^5$

Many different relationships could be defined between the within-group variance and the between-variable correlation. However, many of these relationships would be intuitively silly. For example, any relationship that will allow a variable with $\mathcal{W}_j = 1$ to be selected should not be allowed. Also, any relationship that results in impossible values (those outside of the interval [0,1]) need not be considered. Obviously, piecewise relationships could be considered that solve some of these issues for more complicated relationships. However, we consider that the relationships in Table 1 constitute a relatively thorough, common-sense handling of the issue of variance-correlation relationships.

3.4 Subset Selection

Using multiple criteria for selecting variables will naturally lead to multiple subsets of variables — as many as five solutions under the current relationship structure. Under a clustering framework, one must define a method for choosing between these subsets without specific knowledge of which subset produces the best classifier. One could develop a Bayes factors framework similar to that used by `clustvarsel` (Raftery and Dean, 2006) to compare variable subsets. However, this approach is complicated by the fact that subsets will not necessarily differ by only one variable, as would happen in a truly step-wise approach. Instead,
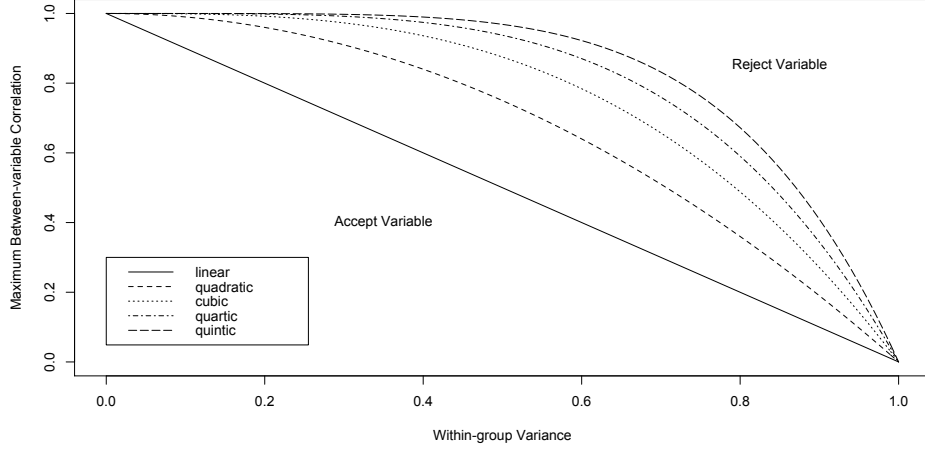
Figure 1: Graphical representation of the selection criteria considered for implementation into Step 4 of the VSCC algorithm.

we introduce a novel approach to selecting variable subsets that relies on one of the major strengths of model-based clustering/classification: measuring the uncertainty of the classification.

The uncertainty for each observation is found simply through the fuzzy classification matrix; i.e., the $n \times G$ matrix containing the $\hat{z}_{ig}$. Each $\hat{z}_{ig}$ element of this matrix is a measure of the strength of evidence indicating observation i belongs to group g. For well-defined clusters, the $\hat{z}_{ig}$ will all be approximately equal to 0, with one entry per row i being approximately equal to 1. We take the uncertainty to be the sum of all the $\hat{z}_{ig}$ entries, except the $\max_g\{\hat{z}_{ig}\}$ entries. This can be expressed as $\sum_{i=1}^{n} \sum_{g=1}^{G} \hat{z}_{ig} - \sum_{i=1}^{n} \max_g\{\hat{z}_{ig}\}$ or equivalently as $n - \sum_{i=1}^{n} \max_g\{\hat{z}_{ig}\}$.

Selecting the variable subset that minimizes the uncertainty in the classification suggests we will be selecting the variables that produce the strongest group structure, so there is some intuitive appeal. In some ways, this is a large departure from information-based criteria; however, the uncertainty is used in the calculation of the integrated completed likelihood by Biernacki et al. (2000), which marries the uncertainty and the Bayesian information criterion (BIC; Schwarz, 1978). The efficacy of using the uncertainty as a relationship selection criterion will be shown in Sections 4 and 5.

The authors note that one concern about using the uncertainty calculations is that, by definition, the uncertainty for any $G = 1$ solution will be 0. It is considered a strength within the clustering field that model-based clustering using the BIC can consider $G = 1$ as a solution and inform the user that there are, in fact, no groups in the data. Unfortunately, given a $G = 1$ solution, the VSCC algorithm cannot be computed as described in this paper. As such, it is an implicit assumption by even running VSCC that $G > 1$ groups exist. Tying into this assumption, it is therefore somewhat reasonable to ignore any variable subsets that produce $G = 1$ as a solution. We recognize this is an unfortunate consequence of utilizing model uncertainty as a subset selection device, and we leave this matter as the subject of future research.

Philosophically, it makes sense to approach variable selection under a "do no harm" mentality. In this vein, note that because we can calculate the uncertainty from the original (non-feature reduced) data set, this solution can be considered as part of the variable selection process. In other words, under VSCC we can select the full data set rather than a reduced set if the full data set results in the minimum uncertainty; we note that for illustrative reasons we will ignore this ability during the simulations in Section 4.

3.5 Clustering

In a clustering scenario, the values of indicator variables z_{ig} are unknown for all $i = 1, \ldots, n$. The VSCC method needs these variables to compute the $\mathcal{W}_j$. However, an initial run of clustering can be used to give 'good' estimates of the component memberships $\hat{z}_{ig}$; any clustering approach could be used (e.g., agglomerative hierarchical, model-based, k-means).

Herein, we focus on the use of a model-based clustering procedure to initialize VSCC, specifically the `mclust` algorithm (Fraley and Raftery, 2002, 2003, 2006). Note that the applications in Section 5 will incorporate 'hard' (0s and 1s in the context of $\hat{z}_{ig}$) initializations from a model-based clustering technique, though 'soft' ('fuzzy' or probabilistic) classifications could be easily incorporated into the procedure instead. Under the clustering format, the algorithm runs as follows:

1. Perform `mclust` under default settings.

2. Use the resultant (hard) $\hat{z}_{ig}$ to initialize VSCC.

3. Perform `mclust` on the (up to) five variable subsets given by VSCC, selecting the best model via the BIC in each case.

4. Select the best variable subset according to the total model uncertainty, and report the results from `mclust` on that subset.

3.6 Classification

In a classification scenario, a subset of the z_{ig} is known and can be utilized by the VSCC method to compute the $\mathcal{W}_j$ in two potential ways. The first option is to use only the known z_{ig} to calculate the $\mathcal{W}_j$. The other option is to calculate the $\mathcal{W}_j$ in a more semi-supervised format, where an original classification algorithm is run to find good estimates of the unknown $\hat{z}_{ig}$. The algorithms for both options follow.

3.6.1 Supervised Algorithm

1. Use only the known z_{ig} to initialize VSCC.

2. Perform model-based classification on the (up to) five variable subsets given by VSCC, selecting the best model via the BIC in each case.

3. Select the best variable subset according to the total model uncertainty, and report the results from model-based classification on that subset.

3.6.2 Semi-Supervised Algorithm

1. Perform model-based classification to estimate the unknown $\hat{z}_{ig}$.

2. Use both the known z_{ig} and the (hard) estimated $\hat{z}_{ig}$ to initialize VSCC.

3. Perform model-based classification on the (up to) five variable subsets given by VSCC, selecting the best model via the BIC in each case.

4. Select the best variable subset according to the total model uncertainty, and report the results from model-based classification on that subset.

3.7 Clustering/Classification Performance

The performance of a clustering algorithm, with respect to known groups present in the data, can be measured in a number of ways. Misclassification rates are often used, but this measure cannot be meaningfully interpreted unless we know the correct number of groups or the clustering algorithm chooses the correct number of groups, which is not always the case. An alternative is to use the Rand index (Rand, 1971), which is calculated as the number of pairwise agreements (between the estimated and known groups) divided by the number of pairs. Because the Rand index is tricky to interpret especially for lower values, the adjusted Rand index (ARI) was introduced by Hubert and Arabie (1985). It essentially accounts for the fact that two random groupings will have some pairwise agreements, thus making the ARI equal to 0 for random clustering and 1 for perfect clustering.

4 Simulations

To determine the performance of VSCC under a variety of scenarios, we introduce several simulation studies. The clusterGeneration package (Qiu and Joe, 2006) from R is used to simulate data sets.

4.1 Increased Dimension

Herein we investigate how VSCC reacts to increased dimension. The genRandomClust function from clusterGeneration is used to generate data sets with four groups, with between 100 to 150 observations per group and where sepVal$= 0.7$ (well separated groups). We generate data sets of dimension 45, 90, 120, and 150, each containing 33% noisy variables. A total of 250 replicates of each data set are generated and analyzed using VSCC and mclust (under default settings). Summary results are given in Table 2.

Table 2: Summary of results from mclust and VSCC on the increased dimension simulations (250 runs per dimension size).

	$d = 45$	$d = 90$	$d = 120$	$d = 150$
mclust Mean ARI	0.79	0.36	0.30	0.23
mclust SD ARI	0.15	0.14	0.17	0.17
mclust Avg Runtime (sec)	5.16	14.74	72.01	157.82
VSCC Mean ARI	0.99	0.84	0.76	0.57
VSCC SD ARI	0.03	0.20	0.30	0.33
VSCC Avg Runtime (sec)	25.95	42.89	160.04	265.62

VSCC performs better than mclust alone on all dimension sizes considered, according to mean ARI. We do, however, note an increase in the standard deviation of the ARI as the dimension size increases. This is due, in large part, to an increased number of $G = 1$ solutions given by mclust for an initialization (which is counted as an ARI of 0 for both mclust and VSCC). Because mclust performance is on average closer to 0, these $G = 1$ examples affect its standard deviation to a lesser extent.

Note also the increase in runtime for both procedures. Keep in mind that VSCC runs mclust once on the full data set, and then up to five times on reduced-variable data sets. Thus, a large savings in computation time could be achieved by at least initializing VSCC using a faster clustering technique (e.g., k-means). To illustrate this point, if the initializations were given to VSCC 'free-of-charge' for the $d = 150$ simulations, VSCC's average runtime would be merely 98.8 seconds.

4.2 Increased Number of Groups

In this simulation, we investigate how VSCC reacts to different numbers of groups present in the data. Once again, the `genRandomClust` function is used to generate 250 replicates of each data set. In this study, we simulate under mostly default conditions — which includes `sepVal`=0.01, or not well separated groups — with 10 noisy and 10 non-noisy variables in each data set. Importantly, we generate data sets for each of $G = 2, 4, 6, 8, 15, 20$ that are then analyzed using VSCC and `mclust` (under default settings except for $G = 15$ and $G = 20$ data sets, where `mclust` is forced to consider $G = 10, \ldots, 20$). Summary results are given in Table 3. We also provide Table 4 for more in-depth details on the $G = 2$ simulation to illustrate the specific variance-correlation relationships as well as the performance of choosing relationships via the total model uncertainty.

Table 3: Summary of results from `mclust` and VSCC on the varied number of groups simulations (250 runs per group structure).

	$G = 2$	$G = 4$	$G = 6$	$G = 8$	$G = 15$	$G = 20$
`mclust` Mean ARI	0.88	0.79	0.80	0.75	0.74	0.73
`mclust` SD ARI	0.08	0.17	0.11	0.13	0.06	0.05
`mclust` Avg. Runtime (sec.)	2.47	6.89	10.76	16.59	97.51	197.31
VSCC Mean ARI	0.89	0.82	0.85	0.83	0.84	0.82
VSCC SD ARI	0.06	0.17	0.08	0.08	0.03	0.03
VSCC Avg. Runtime (sec.)	8.95	18.21	22.87	31.82	171.39	302.77

As the number of groups increases, the general trend for `mclust` is a reduction in clustering performance from 0.88 ($G = 2$) to 0.73 ($G = 20$), coupled with an increase in average runtime (2.47 to 197.31 seconds, respectively). Interestingly though, VSCC's performance remains remarkably consistent, and arguably improves (ignoring $G = 2$) as the number of groups increases to 20 (due to a reduction in variation). However, VSCC does suffer a similar fate in runtime due to its reliance on `mclust`.

Several things stand out in the results presented in Table 4. For one, the linear relationship (which is also the most stringent of those considered) performs terribly under this simulation. Fortunately, the rest of the relationships put up solid performances and, in fact, very similar performances in general, as three of the four select six variables most often. Interestingly, no one relationship on its own would outperform `mclust` on the full data set via either mean ARI or standard deviation; by choosing the best relationship via the total model uncertainty, however, the VSCC algorithm does narrowly beat out the full data set in both categories. This lends support to the use of uncertainty as a selection method. This is not the only simulation where

Table 4: Detailed results from `mclust` and VSCC on the 250 $G = 2$ simulations (10 noisy variables and 10 non-noisy variables). 'Mode # Vars' includes the number of occurrences in parentheses.

Analysis	Mean ARI	SD ARI	Mode #Vars	Mean Unc.	SD Unc.
`mclust`	0.88	0.08	20 (250)	5.88	3.81
VSCC Linear	0.40	0.30	2 (194)	14.45	13.27
VSCC Quadratic	0.81	0.09	3 (75)	9.67	5.26
VSCC Cubic	0.86	0.08	6 (74)	7.32	4.45
VSCC Quartic	0.87	0.09	6 (82)	6.67	4.29
VSCC Quintic	0.88	0.09	6 (71)	6.27	4.21
VSCC (min unc.)	0.89	0.06	6 (75)	5.73	2.86

we see results such as this, but it is not universally true across all simulations. We note further support for using the uncertainty while discussing the real data applications in our concluding paragraphs.

4.3 Classification Example

To briefly demonstrate the feasibility of VSCC in a classification scenario, we apply the method under the supervised algorithm (cf. Section 3.6.1) to the $G = 15$ simulated data from the previous section. For each data set, we randomly select 50% of the data to have known membership and analyze using model-based classification with the MCLUST family of models, then compare these results to using VSCC (with the same model-based classification on the chosen variables). This is performed on the 250 data sets, and a summary of classification performance can be deduced through Figure 2. Note that the ARI reported only considers the observations with 'unknown' cluster membership.
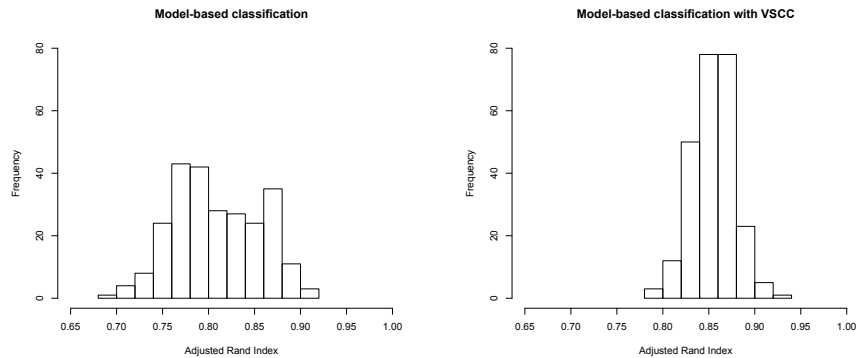

Figure 2: Histograms of classification performance on the $G = 15$ simulated data set by model-based classification and model-based classification with a reduced feature set selected by VSCC.

The mean ARI for analysis on the full data set is 0.81 with a 0.05 standard deviation, while the VSCC reduced data set achieves a mean of 0.85 with a 0.02 standard deviation. Recall the $G = 15$ simulated data set contains 10 non-noisy and 10 noisy variables. VSCC (under the supervised algorithm with 50% known) picks out the 10 meaningful variables 226 times, or for over 90% of the data sets considered.

5 Applications

5.1 Introduction

The VSCC method will now be applied to real data sets under a clustering framework. An introduction to each of the four data sets is given at the beginning of each subsection. To facilitate interpretation, VSCC will be compared to the popular variable selection method introduced by Raftery and Dean (2006), available as the `clustvarsel` package in R, as well as the `selvarclust` technique introduced by Maugis et al. (2009), available as a command-line addition to the MIXMOD software (Biernacki et al., 2006). Recall from Section 3.5, we utilize VSCC under a `mclust` framework, meaning we use `mclust` to initialize the $\hat{z}_{ig}$ and to cluster the feature-reduced data sets. In VSCC, we will utilize `mclust` under the default settings: `mclust` considers $G = 1, \ldots, 9$. Hence, for fairness of comparison, `clustvarsel` and `selvarclust` will be set to consider $G = 1, \ldots, 9$ as well. Note that all methods will be run on standardized variable (mean 0, variance 1) versions of the data discussed. Finally, `selvarclust` will be restricted to the covariance parameterizations available in `mclust`, again for fairness of comparison. Even so, the results presented from `selvarclust` are

not as directly comparable to the other two variable selection methods due to a difference in initializations used by the MIXMOD software versus `mclust`.

5.2 Leptograpsus Crabs

The Leptograpsus crabs data set can be found in the `MASS` library in R. It contains five length measurements on two different colour forms of the crabs, further separated into the two genders. The results for the `mclust` initialization, VSCC, `clustvarsel`, and `selvarclust` on the crabs data set are given in Table 5.

Table 5: Table of results from `mclust`, VSCC, `clustvarsel`, and `selvarclust` on the crabs data set. The relationship chosen for VSCC by the total model uncertainty is given in parentheses.

Analysis	ARI	Time (sec.)	G	#Vars	Unc.
mclust	0.31	3.94	4	5	14.71
VSCC (Quintic)	0.76	12.49	5	4	10.96
clustvarsel	0.76	63.01	5	4	10.96
selvarclust	0.50	256.69	5	4	12.79

On this data set, `clustvarsel` (via approximate Bayes factors) and VSCC (via total uncertainty) agree on the solution that eliminates one variable and increases the ARI from 0.31 to 0.76. In fact, the `selvarclust` algorithm selects the same variables, but the results are very different due to the initializations used. We can, for all intents and purposes, consider the performance of all techniques equivalent. The main item of interest here is that VSCC accomplishes this task five times faster than `clustvarsel` and over 20 times faster than `selvarclust`.

Perhaps a more important aspect of this application is that VSCC manages to increase clustering performance from a poor initialization (increasing the ARI from 0.31 for the correct number of groups to 0.76 for group over-estimation). One argument against an approach such as VSCC could be that one might need 'quite good' initializations for the technique to be useful; the crabs data set, however, shows that this is not necessarily the case. Note that `mclust` actually chooses the correct number of groups, meaning that the ARI is not artificially deflated by the choice of large G; its clustering performance is simply poor.

5.3 Italian Wine

The Italian wine data set is readily available in the `gclus` library in R and contains 13 chemical measurements on 178 samples of wine originating from three different varieties (Barolo, Grignolino, and Barbera). From the results (Table 6), we can see that VSCC outperforms the clustering done by `mclust` alone as well as those done by `clustvarsel` and `selvarclust`. Running `mclust` on the variables selected by `selvarclust` results in the same performance as listed for `selvarclust`. In addition to outperforming `clustvarsel` and `selvarclust` with respect to clustering performance, VSCC runs over 10 and 200 times faster, respectively.

Table 6: Table of results from `mclust`, VSCC, `clustvarsel`, and `selvarclust` on the wine data set. The relationship chosen for VSCC by the uncertainty is given in parentheses.

Analysis	ARI	Time (sec.)	G	#Vars	Unc.
mclust	0.48	1.91	8	13	5.85
VSCC (Quartic)	0.90	10.25	3	9	0.90
clustvarsel	0.78	113.95	3	5	2.23
selvarclust	0.54	2220.42	7	8	6.35

5.4 Swiss bank notes data

The Swiss bank notes data set is also available in the `gclus` library in R and contains six measurements on 200 monetary bills, of which some are legal tender and others counterfeit. The results (Table 7) show that VSCC again results in the best clustering performance, with an ARI of 0.85 compared to 0.68 and 0.67 for `mclust` and `clustvarsel`, respectively. Running `mclust` on the variables chosen by `selvarclust` results in an ARI of 0.69, leaving it roughly on par with the `mclust` and `clustvarsel` results. VSCC utilizes fewer variables (4 versus 5) and runs eight times faster than `clustvarsel`.

Table 7: Table of results from `mclust`, VSCC, `clustvarsel`, and `selvarclust` on the bank notes data set. The relationship chosen for VSCC by the uncertainty is given in parentheses.

Analysis	ARI	Time (sec.)	G	#Vars	Unc.
mclust	0.68	2.34	4	6	6.16
VSCC (Quadratic)	0.85	8.52	3	4	0.17
clustvarsel	0.67	66.18	4	5	6.10
selvarclust	0.25	357.51	8	3	15.71

5.5 Coffee data

The coffee data set given by Streuli (1973) contains 13 chemical measurements on 43 samples of coffee hailing from one of two species: Arabica or Robusto.These results (Table 8) serve as an example where a variable selection method can negatively affect clustering performance. While `mclust` performs perfect classification on the full data set, `clustvarsel` and `selvarclust` (including under `mclust` analysis of the selected variables) select too many groups.

Table 8: Table of results from `mclust`, VSCC, `clustvarsel`, and `selvarclust` on the coffee data set. The relationship chosen for VSCC by the uncertainty is given in parentheses.

Analysis	ARI	Time (sec.)	G	#Vars	Unc.
mclust	1.00	0.16	2	13	0.00
VSCC (Quadratic)	1.00	1.19	2	2	0.00
clustvarsel	0.41	2.79	3	6	0.42
selvarclust	0.37	404.67	4	7	0.23

Perhaps more importantly, VSCC gives perfect classification while reducing the number of variables from 13 to 2: caffeine and fat content. A visualization of the clusters on these two variables is given in Figure 3.

5.6 Comparison with Correctly Specified G

We conclude the analysis of real data sets by comparing VSCC with results from the `sparcl` package (Witten and Tibshirani, 2011) in R, which contains an implementation of the sparse k-means technique described by Witten and Tibshirani (2010). We use this approach under default settings, allowing `KMeansSparseCluster.permute()` to select the tuning parameter. Because this approach is based on k-means, the number of groups must be specified; this thus differs from the approaches compared in the previous sections. Table 9 contains the results from all four data sets analyzed by `sparcl` and VSCC.

The results across all four data sets show that VSCC can be run in a significantly shorter amount of time and result in better (or equal, in one case) clustering performance. Note also that `sparcl` has reduced the
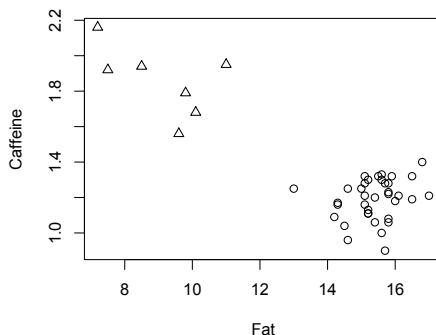
Figure 3: Scatterplot of the Arabica (circles) and Robusto (triangles) coffee beans on the VSCC chosen variables.

Table 9: Table of results from VSCC and `sparcl` on the previously introduced data sets with the correct number of groups pre-specified. The relationship chosen for VSCC by the uncertainty is given in parentheses.

Data Set	Analysis	ARI	Time (sec.)	#Vars
Crabs	VSCC (Quintic)	0.37	1.01	4
	`sparcl`	0.02	6.17	5
Wine	VSCC (Cubic)	0.93	0.33	7
	`sparcl`	0.85	6.36	13
Bank	VSCC (Quintic)	0.98	0.26	5
	`sparcl`	0.96	4.55	6
Coffee	VSCC (Cubic)	1.00	0.12	6
	`sparcl`	1.00	2.71	13

dimensionality of the data set but has not technically removed any variables from consideration (because none of the variables have been weighted to zero in the model). As a small side note, we point out that the VSCC results on the coffee and wine data sets differ slightly from those earlier in Sections 5.3 and 5.5, even though the final number of groups selected is the same. This is due to differences in initializations from the preliminary `mclust` runs, and thus is not particularly surprising.

6   Discussion and Future Work

A novel variable selection technique (VSCC) based on within-group variance was introduced and utilized under a model-based clustering framework. The strengths of the technique lie in the speed at which it can be run as well as its intuitive appeal. It was shown to outperform or equal `clustvarsel`, `selvarclust`, and `sparcl` in clustering performance, and significantly outperform them in speed, on four commonly used data sets. Notably, the VSCC relationship chosen by the total uncertainty was, in every real data set considered, the relationship that resulted in the highest ARI. This, along with several of the simulation studies, lends support for the use of total uncertainty as a subset selection criteria.

The inner workings of VSCC are flexible enough to incorporate clustering/classification algorithms other than the model-based techniques covered in this article. One hurdle to overcome in this respect is an effective

subset selection criterion — as non-model-based methods will not contain uncertainty measures — and this will be a subject of future research. In addition, the development of VSCC software for the R computing environment is intended pending code optimization and further testing.

Acknowledgements

References

Andrews, J. L. and P. D. McNicholas (2011a). Extending mixtures of multivariate t-factor analyzers. *Statistics and Computing 21*(3), 361–373.

Andrews, J. L. and P. D. McNicholas (2011b). Mixtures of modified t-factor analyzers for model-based clustering, classification, and discriminant analysis. *Journal of Statistical Planning and Inference 141*(4), 1479–1486.

Biernacki, C., G. Celeux, and G. Govaert (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*(7), 719–725.

Biernacki, C., G. Celeux, G. Govaert, and F. Langrognet (2006). Model-based cluster and discriminant analysis with the MIXMOD software. *Computational Statistics and Data Analysis 51*(2), 587–600.

Bouveyron, C. and C. Brunet (2012). Simultaneous model-based clustering and visualization in the Fisher discriminative subspace. *Statistics and Computing 22*(1), 301–324.

Dean, N. and A. E. Raftery (2006). *The clustvarsel Package*. R package version 0.2-4.

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.

Fraley, C. and A. E. Raftery (2003). Enhanced software for model-based clustering, density estimation, and discriminant analysis: MCLUST. *Journal of Classification 20*, 263–286.

Fraley, C. and A. E. Raftery (2006). MCLUST: version 3 for R: Normal mixture modeling and model-based clustering. Technical Report 504, Department of Statistics, University of Washington. Revised 2009.

Ghahramani, Z. and G. E. Hinton (1997). The EM algorithm for factor analyzers. Technical Report CRG-TR-96-1, University of Toronto, Toronto.

Hubert, L. and P. Arabie (1985). Comparing partitions. *Journal of Classification 2*, 193–218.

Kass, R. E. and A. E. Raftery (1995). Bayes factors. *Journal of the American Statistical Association 90*, 773–795.

Maugis, C., G. Celeux, and M.-L. Martin-Magniette (2009). Variable selection for clustering with Gaussian mixture models. *Biometrics 65*(3), 701–709.

McLachlan, G. J., R. W. Bean, and D. Peel (2002). A mixture model-based approach to the clustering of microarray expression data. *Bioinformatics 18*(3), 413–422.

McLachlan, G. J. and D. Peel (2000). Mixtures of factor analyzers. In *Proceedings of the Seventh International Conference on Machine Learning*, San Francisco, pp. 599–606. Morgan Kaufmann.

McNicholas, P. D. and T. B. Murphy (2008). Parsimonious Gaussian mixture models. *Statistics and Computing 18*, 285–296.

McNicholas, P. D. and T. B. Murphy (2010). Model-based clustering of microarray expression data via latent Gaussian mixture models. *Bioinformatics 26*(21), 2705–2712.

Montanari, A. and C. Viroli (2010). Heteroscedastic factor mixture analysis. *Statistical Modelling 10*(4), 441–460.

Qiu, W. and H. Joe (2006). Generation of random clusters with specified degree of separation. *Journal of Classification 23*, 315–334.

R Development Core Team (2012). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing.

Raftery, A. E. and N. Dean (2006). Variable selection for model-based clustering. *Journal of the American Statistical Association 101*(473), 168–178.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association 66*, 846–850.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics 6*(2), 461–464.

Scrucca, L. (2010). Dimension reduction for model-based clustering. *Statistics and Computing 20*(4), 471–484.

Streuli, H. (1973). Der heutige stand der kaffeechemie. In *Association Scientifique Internationale pour le Café, 6th International Colloquium on Coffee Chemistry*, Bogatá, Columbia, pp. 61–72.

Tipping, T. E. and C. M. Bishop (1999). Mixtures of probabilistic principal component analysers. *Neural Computation 11*(2), 443–482.

Viroli, C. (2010). Dimensionally reduced model-based clustering through mixtures of factor mixture analyzers. *Journal of Classification 27*(3), 363–388.

Witten, D. and R. Tibshirani (2010). A framework for feature selection in clustering. *Journal of the American Statistical Association 105*(490), 713–726.

Witten, D. M. and R. Tibshirani (2011). *sparcl: Perform sparse hierarchical clustering and sparse k-means clustering*. R package version 1.0.2.