# Sparse and group-sparse clustering for mixed data An illustration of the `vimpclust` package

Marie Chavent[1] & Jérôme Lacaille[2] & Alex Mourer[2,3] & Madalina Olteanu[4]

[1] *Univ. Bordeaux, CNRS, INRIA, Bordeaux INP, IMB, UMR 5251, marie.chavent@u-bordeaux.fr*
[2] *Safran Aircraft Engines - Datalab, jerome.lacaille@safrangroup.com*
[3] *SAMM EA 4543, Université Panthéon Sorbonne, moureralex@gmail.com*
[4] *CEREMADE, UMR 7534, Université Dauphine PSL, olteanu@ceremade.dauphine.fr*

**Résumé.** Les données en grande dimension contiennent souvent un mélange de variables numériques et catégorielles. De plus, dans de nombreux cas, les variables sont disponibles sous la forme de groupes définis a priori (mesures répétées, catégories d'attributs, ...). En pratique, classifier ces données pose plusieurs problèmes : comment utiliser les données mixtes pour construire des clusters pertinents? comment sélectionner les variables ou les groupes de variables les plus pertinents pour le *clustering*? Avec une approche *k-means*, il est possible d'utiliser une version pénalisée de la variance inter-classes, et de trouver à la fois la meilleure partition des données, et les variables ou groupes de variables les plus informatifs. Le présent manuscrit décrit les méthodes des *sparse k-means* et des *group-sparse k-means* pour des données mixtes, en utilisant le package `R` `vimpclust`. L'exemple d'un petit jeu de données réelles illustre comment la sélection de variables peut être directement combinée avec le *clustering*, en fournissant à la fois une classification pertinente et en préservant la qualité du *clustering*.

**Mots-clés.** clustering, $k$-means parcimonieux, pénalités $L_1$ et $L_1$-groupe, données mixtes, packages `R`.

**Abstract.** High-dimensional data may often contain both numerical and categorical features, and in some cases features may be available as natural groups (repeated measurements, categories of features, ...). Clustering this kind of data raises several issues: how to simultaneously deal with numerical and categorical features? how to build meaningful clusters of the input entities? how to select the most informative features or groups of features for the clustering? In the $k$-means framework, one may rely on a penalised version of the between-cluster variance, and find both the best partitioning of the data, and the most informative features or groups of features. The present manuscript illustrates sparse $k$-means and group-sparse $k$-means for mixed data, using the `vimpclust` package. The example provided on a small real-life dataset shows how feature selection may be directly combined with clustering, and provide a meaningful selection while preserving the quality of the clustering.

**Keywords.** clustering, sparse $k$-means, $L_1$ and group-$L_1$ penalties, mixed data, `R` packages.

# 1 Introduction

While feature selection received a great deal of attention in the supervised learning framework during the past twenty years, it is only much later and relatively recently that it effectively emerged in the unsupervised context. At the same time, with the dimensionality of the collected data constantly increasing, feature selection for clustering becomes a crucial issue, since the presence of many uninformative features could potentially damage the clustering, introduce noise and favour unstable results. Furthermore, features may often be available as priorly known groups (repeated measurements, categories of attributes, ...), and imply that group-wise selection should be considered instead of feature-wise one.

Since model-based approaches are naturally suited for including $L_1$ and related penalties, several contributions were proposed in the literature, and are summarised, for instance, in [1], which contains a detailed review on model-based clustering for high-dimensional data. In the $k$-means context, one may cite the sparse $k$-means procedure, introduced in [9] and based on a $L_1$-penalised version of the between-class variance, or more recent developments in [8], [2] or [6].

That being said, all methods cited above are essentially designed for numerical data, while real data is often made of numerical and categorical features. Some of the authors above touch upon the question of categorical features, by mentioning the possibility of making them numerical using a transformation through dummy variables. However, this processing step is not that immediate, since the Euclidean distance on zero-one vectors is not particularly suited for being mixed with Euclidean distances on numerical variables. Other authors implicitly suggest that the proposed algorithms may be written in terms of distances or dissimilarities between input data only, and hence it suffices to use an appropriate distance for categorical features. Nevertheless, the distance-based approaches may rapidly translate into an increased complexity if the size of the data becomes large.

In a recent contribution [5], we introduced an explicit method for feature selection in a mixed-data framework, by building on a penalized group-$L_1$ criterion. An R package, called `vimpclust`[1] and available on CRAN, has been implemented and allows one to train sparse $k$-means procedures on numerical or mixed features, or on groups of numerical features. In the present manuscript, we extend the above to groups of mixed features and add a new functionality to the package. In the rest of the paper, we briefly recall how to train sparse $k$-means for groups of numerical features, for mixed features, and for groups of mixed features, and illustrate these methods using the `vimpclust` package and a small real-life dataset.

[1] https://cran.r-project.org/web/packages/vimpclust/index.html

## 2 Sparse clustering for groups of numerical features

Group-sparse clustering may be of interest in many real-life situations, where a natural structure of groups is available for the features. In other practical applications, groups of features may be the output of a clustering applied to the features.

In the following, suppose that the $p$ numerical features are divided into $L$ priorly known groups, such that $\mathbf{X} = \left[\mathbf{X}^1 | \dots | \mathbf{X}^L\right] \in \mathbb{R}^{n \times p}$ is the matrix of data, where $\ell = 1, \dots, L$, and $\mathbf{X}^\ell \in \mathbb{R}^{n \times p_\ell}$ represents the features in group $\ell$, and $p_1 + ... + p_\ell = p$.

Group sparse $k$-means searches both a partitioning $C_1, ..., C_K$, and a set of weights $\mathbf{w}^T = (\mathbf{w}^1, \dots, \mathbf{w}^L) \in \mathbb{R}^p$, maximising the between-class variance penalised by the $L_1$-group penalty:

$$\max_{C_1,...,C_K, \mathbf{w} \in \mathbb{R}^p} \mathbf{w}^T \mathbf{b} - \lambda \sum_{\ell=1}^{L} \sqrt{p_\ell} \|\mathbf{w}^\ell\|_2, \tag{1}$$

where $\|\mathbf{w}\|_2 \leq 1$ and $\mathbf{w} \geq 0$, and where $\lambda > 0$ is a tuning hyperparameter.

In the notation above, $\mathbf{b}^T = (b_1, ..., b_p)$ is the vector containing the between-class variance computed per each feature,

$$b_j = b_j(\mathbf{X}, C_1, ..., C_K) = \sum_{k=1}^{K} \frac{n_k}{n} (\overline{X}_{j,k} - \overline{X}_j)^2, \tag{2}$$

with $\overline{X}_j$ being the average of feature $j$ over the entire data set, $\overline{X}_{j,k}$ the average of feature $j$ in cluster $k$, and $n_k$ the size of cluster $C_k$. One should also remark here that each component $\mathbf{w}_\ell$ or $\mathbf{b}_\ell$ in $\mathbf{w}$ and, respectively, $\mathbf{b}$, are actually vectors of size $p_\ell$.

For a fixed number of clusters $K$ and a fixed tuning parameter $\lambda$, the optimisation problem above may be solved using an iterative algorithm, which alternates two steps:

1. For a fixed vector of weights $\mathbf{w}$, find the partitioning $C_1, ..., C_K$ maximising $\mathbf{w}^T \mathbf{b}$. This is actually equivalent to training an usual $k$-means algorithm on the features scaled by the weights $\tilde{X}_j = \sqrt{w_j} X_j$, $j = 1, ..., p$.

2. For a fixed partitioning $C_1, ..., C_K$, find the vector of weights $\mathbf{w}$ maximising the penalised between-class variance, and such that $\|\mathbf{w}\|_2 \leq 1$. The solution may be found analytically as a function of the group soft-thresholding operator, and is described in detail in [5].

From a practical point of view, the group soft-thresholding operator will drop all groups of features which between-class variance norm $\|\mathbf{b}_\ell\|_2$ is smaller than $\sqrt{p_\ell}\lambda$, and shrink by $\sqrt{p_\ell}\lambda$ the between-class variance norm of the remaining groups of variables.

# 3 Sparse clustering for mixed data

In the context of mixed data, the group-sparse principle above may be used for building a procedure for variable selection. Indeed, each categorical feature may be transformed into a group of dummy ones, thus producing a natural structure of groups on the data. If the input data $\mathbf{X}$ is described by $d_1$ categorical features and $d_2$ numerical ones, such that each of the categorical features has $p_j$ possible values, $j = 1, ..., d_1$, she may transform each categorical feature $X_j$ into $p_j$ dummy variables $\tilde{\mathbf{X}}_j = (\tilde{X}_j^1, ..., \tilde{X}_j^{p_j}) \in \{0,1\}^{n \times p_j}$ and thus define a natural group structure on the transformed data $\mathbf{Y} = [\mathbf{Y}_1|...|\mathbf{Y}_{d_1+d_2}]$, where $\mathbf{Y}_\ell = \tilde{\mathbf{X}}_\ell$ for $\ell = 1, ..., d_1$, $\mathbf{Y}_\ell = X_\ell$ for $\ell = d_1 + 1, ..., d_1 + d_2$, with respective group sizes $\mathbf{p}^T = (p_1, ..., p_{d_1}, 1, ..., 1) \in \mathbb{R}^{d_1+d_2}$.

When training group-sparse $k$-means on the group structure above, this amounts to performing variable selection in a mixed-data context. Before applying the algorithm described in the previous section, $\mathbf{Y}$ must be properly preprocessed. This prior step is described for example in [4]: numerical variables are scaled to zero mean and unit variance, while the dummy variables are centered and normalized by $1/\sqrt{\frac{n}{n_{j,s}}}$, where $n_{j,s}$ is the number of input data taking the $s$-th value of the $j$th categorical feature, or equivalently the sum over $\tilde{X}_j^s$. The scaling applied to the dummy variables actually leads to using a $\chi^2$ distance on the categorical features, while the numerical features, after scaling, are compared through the usual Euclidean distance.

# 4 Sparse clustering for groups of mixed features

Suppose now that the $p = d_1 + d_2$ categorical and numerical features are divided into $L$ priorly known groups. The sparse clustering procedures described above (for groups of numerical features and for non grouped mixed data) may be combined to select groups of mixed features. Indeed, the input data $\mathbf{X}$ may be transformed into a numerical matrix $\mathbf{Y}$ with $p_1 + \ldots + p_{d_1}$ dummy variables replacing the $d_1$ categorical features, this matrix $\mathbf{Y}$ being pre-processed as described section (3). The $p_j$ dummy variables of each categorical features $X_j$ may then be assigned to the group $\ell$ of $X_j$, thus defining $L$ groups of the $p_1 + \ldots + p_{d_1} + d_2$ transformed features. The sparse clustering method for groups of numerical variables described in Section 2 may then be applied to the pre-processed numerical matrix $\mathbf{Y}$ and these new groups. Because the levels of a categorical feature belong by construction to the same group, if this group is selected, the feature is selected.

# 5 An illustration of `vimpclust`

The `vimpclust` package has been implemented as a general tool for sparse $k$-means clustering. It handles sparse $k$-means for numerical features as introduced in [9], but also

group-sparse $k$-means for numerical features, sparse $k$-means for mixed features, and group sparse $k$-means for mixed features. The package is available on CRAN, and contains several detailed vignettes with use cases.

As an example, sparse $k$-means and group sparse $k$-means for mixed data are illustrated on the `wine` dataset [7]. The `wine` data contains a description of 21 wine varieties, using 31 features, including soil type and wine label (categorical with respectively five and three categories) and different sensory descriptors (numerical). The dimensionality of the data is larger than the number of inputs, and feature selection through sparse clustering is crucial for meaningful interpretation.

First, the data is clustered using the sparse $k$-means procedure for mixed data, using the `sparsewkm` function in `vimpclust`. The number of clusters is fixed equal to four.

```
res <- sparsewkm(X = wine, centers = 4)
plot(res, what="weights.features")
plot(res, what="expl.var")
```

The regularisation paths for the weights of each feature, but also the regularisation path of the ratio of explained variance are illustrated in Figure 1. The hyper-parameter $\lambda$ varies between 0 and 1, as the initial features were all scaled to unit variance. If one decides to keep the last configuration before the ratio of explained variance drops significantly, ten features out of thirty are eventually selected: *Aroma.quality.before.shaking, Quality.of.odour, Balance, Intensity, Overall.quality, Surface.feeling, Aroma.quality, Smooth, Harmony, Typical*.

What about if one is interested in selecting groups of significant features for the clustering? In order to do so, and since one has no prior knowledge on the groups of features for the `wine` dataset, we decide to build groups of features using the `ClustOfVar` procedure as described in [3]. The resulting 7 groups of features are illustrated in Figure 2.

Clustering is achieved using the `groupsparsewkm` function in `vimpclust`, and by providing a supplementary argument `index`, which contains groups labels for the features. Here again, four clusters for the input data were searched for.

```
res <- groupsparsewkm(X, centers = 4, index = groupes)
plot(res, what = "weights.features")
plot(res, what="expl.var")
```

The regularisation paths both in terms of features, and in terms of groups of features, are illustrated in Figure 3.

After selecting $\lambda$ such that the best trade-off between the ratio of explained variance and the number of selected groups is achieved, only three groups of features are kept. Their composition in detailed in Figure 4. As one may easily see, all features selected by the sparse $k$-means procedure for mixed data also appear in the selected groups of

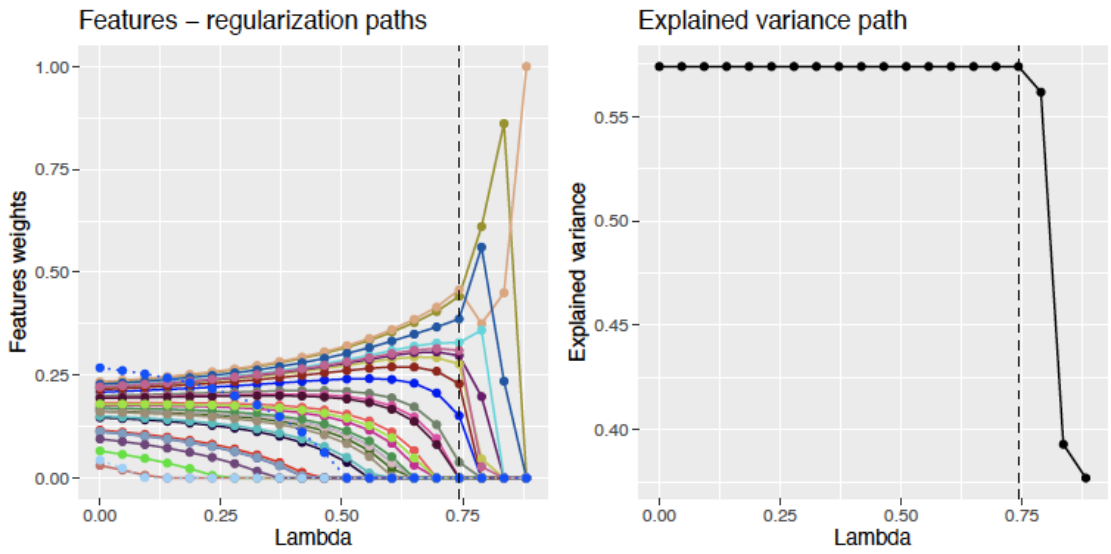


Figure 1: Regularisation paths for different values of $\lambda$. Left: features paths. Right: ratio of explained variance path.

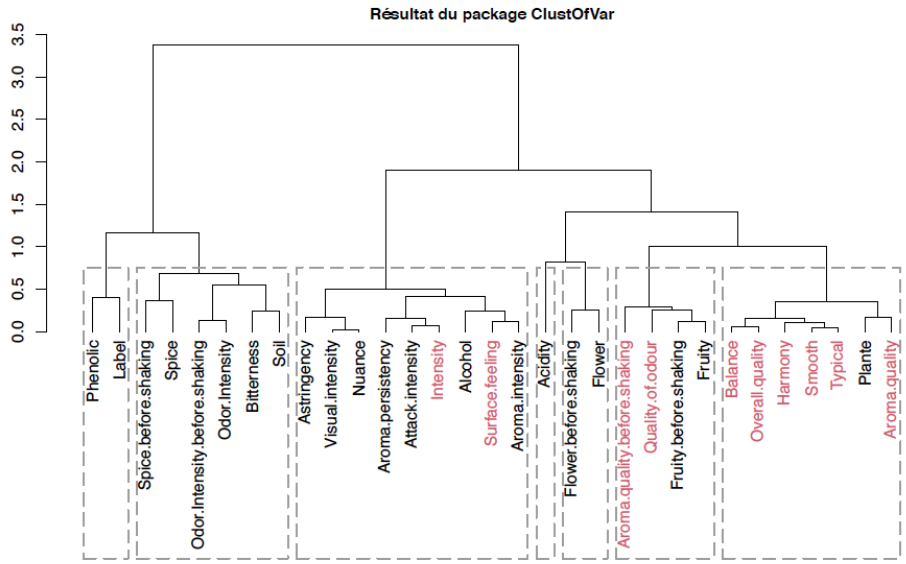


Figure 2: Resulting groups of features after having trained `vimpclust`

features. The results of the two procedures are thus coherent. Furthermore, groups of features are interesting to consider since they provide better insights on the available information in the data.

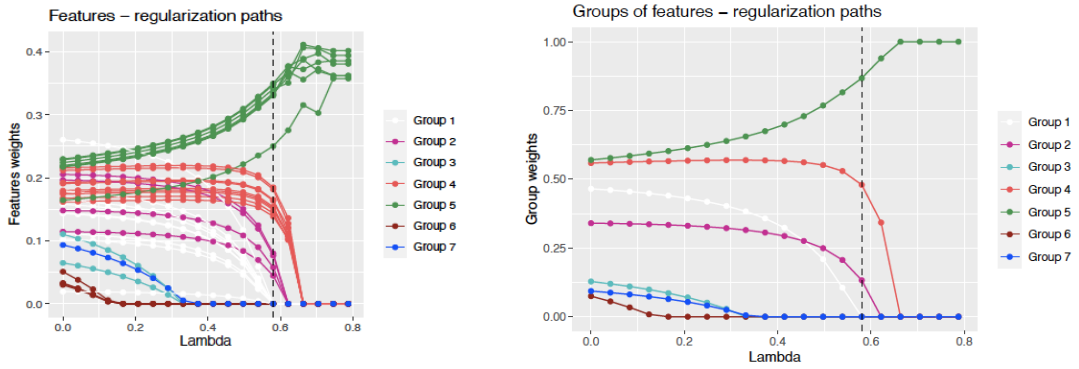


Figure 3: Regularisation paths for different values of $\lambda$. Left: features paths. Right: groups of features paths.

| Group 2 | Group 4 | Group 5 |
|---|---|---|
| Aroma.quality.before.shaking | Visual.intensity | Plante |
| Fruity.before.shaking | Nuance | Aroma.quality |
| Quality.of.odour | Surface.feeling | Balance |
| Fruity | Aroma.intensity | Smooth |
| | Aroma.persistency | Harmony |
| | Attack.intensity | Overall.quality |
| | Astringency | Typical |
| | Alcohol | |
| | Intensity | |

Figure 4: Selected groups of features by the group-sparse $k$-means. Red: features selected individually by the sparse $k$-means procedure.

# 6 Conclusion

Several $k$-means generalisations, allowing to train sparse clusterings on mixed data, and to select important features or groups of important features were described in the above sections. These variants of $k$-means are implemented in a recent package available on CRAN, `vimpclust`. A detailed example on a small real-life data set allowed to illustrate how the proposed methods produce both meaningful clusterings and feature selection, the latter being a crucial issue for high-dimensional data.

# References

[1] Ch. Bouveyron and C. Brunet-Saumard. Model-based clustering of high-dimensional data: A review. *Computational Statistics & Data Analysis*, 71:52–78, 2014.

[2] S. Chakraborty and S. Das. A strongly consistent sparse $k$-means clustering with direct $l\_1$ penalization on variable weights. *arXiv preprint arXiv:1903.10039*, 2019.

[3] M. Chavent, B. Kuentz-Simonet, V.and Liquet, and J. Saracco. Clustofvar: An r package for the clustering of variables. *Journal of Statistical Software*, 50:116, 2012.

[4] M. Chavent, V. Kuentz-Simonet, A. Labenne, and J. Saracco. Multivariate analysis of mixed data: The pcamixdata r package. *arXiv preprint arXiv:1411.4911*, 2014.

[5] M. Chavent, J. Lacaille, A. Mourer, and M. Olteanu. Sparse k-means for mixed data via group-sparse clustering. In *ESANN 2020 - 28th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, volume 978-2-87587-074-2, Bruges / Virtual, Belgium, October 2020.

[6] Z. Huo and G. Tseng. Integrative sparse $k$-means with overlapping group lasso in genomic applications for disease subtype discovery. *Ann. Appl. Stat.*, 11(2):1011–1039, 06 2017.

[7] S. Le, J. Josse, and F. Husson. Factominer: An r package for multivariate analysis. *Journal of Statistical Software*, 25(1):1–18, 2008.

[8] W. Sun, J. Wang, Y. Fang, et al. Regularized k-means clustering of high-dimensional data and its asymptotic consistency. *Electronic Journal of Statistics*, 6:148–167, 2012.

[9] D.M. Witten and R. Tibshirani. A framework for feature selection in clustering. *Journal of the American Statistical Association*, 105(490):713–726, 2010. PMID: 20811510.