Scheduling Parallel Conference Sessions: An Application of a Hybrid Clustering Algorithm for Constrained Cardinality

B. Houlding*, J. Haslett

Discipline of Statistics, Trinity College Dublin, Ireland.

Abstract

The 2011 World Statistics Congress included approximately 1,150 oral and 250 poster presentations over approximately 250 sessions, with as many as 20 sessions running in parallel at any one time. Scheduling a timetable for such a conference is hence a complicated task, as ideally, talks on similar topics should be scheduled in the same session, and similar session topics should not be presented at identical times, allowing participants to easily decide which of the number of sessions to attend. Here we consider a novel hybrid clustering algorithm that allows a solution under a constraint of fixed cardinality, and which is designed to find clusters of highly dense regions whilst forming others from the remaining outlying regions.

Keywords: Cluster Analysis, Non-Parametric, Constrained Cardinality, Timetabling.

1. Introduction

Appropriate scheduling of parallel sessions for a conference can be a difficult task. If talks are scheduled through a random process participant experience could be degraded, as topics that are considered of individual interest may be scheduled at identical times in different locations, or participants may feel the

*Corresponding author, address: Rm 105, Discipline of Statistics, Lloyd's Institute, Trinity College Dublin, Dublin 2, Ireland. E-mail: brett.houlding@tcd.ie, tel: +353 (0) 1896 2933, fax: +353 (0) 1 677 0711.

need to only attend one talk in any given session. A more appropriate scheduling technique would be to ensure that talks on similar topics are held in the same session, mitigating the risk that attendees would not have interest in all of the talks scheduled.

In addition, if sessions are run in parallel, then it would not be appropriate to hold similar sessions at the same time but in different venues, as the likely audience for those sessions would be split. Such problems are further expounded if a conference focuses on a general discipline that has many diverse elements, *e.g.*, statistics, which has elements of theoretical statistical development, applied statistical analysis, collection of data, formulation of national statistics, and the use of statistics in industry *etc.* Moreover, conference contributors are likely to not only include academics, but also government and private sector statisticians, and these distinct groups will be primarily interested in talks given by others working in the same sector.

For small to medium sized conferences this problem may be overcome by manual inspection of all proposed talks by an organizer with detailed knowledge of a broad range of topics. For larger conferences a typical approach is to organize program committees whose task would be to sift through a very large collection of contributed talks. Program chairs would then be responsible for assembling those talks into sessions that are as cohesive as possible, but with other remaining sessions having more of a 'potpourri' feeling. There may then be manual adjustment to improve results.

Hence scheduling parallel conference sessions is an expensive and complicated task. This is not only due to the time that would be required to inspect all proposed talks, but also because, when considering conferences with potentially hundreds of talks, it becomes impossible for an organizer to remember all proposals, and it could well be that, as proposals are considered in sequence, the second talk reviewed is most similar to the last talk reviewed. Furthermore, any given conference organizer will have their own views on what is a similar talk that may not be agreed with by those who are experts in a given sub-discipline. For all of these reasons, a more automated data driven approach is required.

Cluster analysis is the natural statistical framework for assigning singletons into groups or 'clusters'. It involves the use of an unsupervised algorithm with the typical objective of finding a grouping of all data points such that elements within a cluster are 'similar', whilst elements in different clusters are 'dissimilar', with the meaning of 'similarity' being predefined. However, in the application of scheduling parallel sessions for a conference we require an additional constraint that clusters (the sessions themselves) are of a fixed size, *i.e.*, that each session will consist of m talks. This would appear to be a novel (or at least a relatively new) requirement, as usually there is no constraint on the size of any given cluster, nor that clusters must be of the same size.

The remainder is as follows: in Section 2 we review cluster analysis and present the proposed hybrid algorithm that seeks to solve the problem of clustering with constrained cardinality. In Section 3 we discuss the application to the scheduling of the 2011 International Statistical Institute's 58th World Statistics Congress that was held in Dublin and was organized by Ireland's Central Statistics Office. Finally, we conclude with a discussion in Section 4.

2. Constrained Cluster Cardinality

Given possibly multi-variate data $\mathbf{x}_1, \ldots, \mathbf{x}_n$, the objective of cluster analysis is to partition the data into clusters $C_1, \ldots, C_k$, where the number of clusters k is not necessarily fixed. Generally one of two approaches is adopted, namely either a 'hierarchical' [10] or an 'iterative' [8] clustering algorithm. Hierarchical algorithms are non-parametric in nature and are exploratory techniques by design. They require both a dissimilarity measure $d(\mathbf{x}_i, \mathbf{x}_j)$ that details the dissimilarity between two data points $\mathbf{x}_i$ and $\mathbf{x}_j$, and a linkage measure $l(C_r, C_s)$, which is a function of the dissimilarity measure, and details the dissimilarity between clusters C_r and C_s. The algorithm then proceeds by initially assigning each data point as a cluster on its own, before repeatedly merging the two least dissimilar clusters under the linkage criteria until a single cluster is formed that consists of all data points. At this stage a point in which the hierarchical algorithm should have stopped is determined.

Alternatively, iterative algorithms such as k-means [8] or partitioning about medoids [6] assume a fixed number of clusters that remains at all steps and instead proceeds by re-assigning all data points to clusters until some convergence criteria has been achieved. They do not require specification of a linkage measure, but instead employ a typical Expectation-Maximization (EM) type algorithm [3]. However, in both hierarchical and iterative methods the implementation of cluster analysis does not usually require anything in particular about the cardinality of cluster sizes, denoted $|C_1|, \ldots, |C_k|$, yet our motivation does require development of a clustering algorithm that ensures $|C_1| = \cdots = |C_k| = m$, for a given m, or more generally, that ensures $|C_1| = m_1, \ldots, |C_k| = m_k$.

Constrained cluster analysis is a relatively recent field of research (see for example [1] for a comprehensive review), yet has tended primarily to focus on constraints that either require two (or more) data points to be in the same cluster, or that two (or more) data points can not be in the same cluster. This is irrelevant for our purpose as we are concerned only with the size of the clusters. Size constraints were, however, considered by Zhu *et al.* [11], where an heuristic algorithm was presented using both prior knowledge on cluster size and the results of an unconstrained iterative (k-means) clustering to find a most-similar alternative clustering with k clusters that do satisfy the constraints. This is achieved by transforming the problem into an integer linear programming problem whose solution is the desired approximation.

Our approach is to instead use the constraints on the cardinality of the clusters directly within the steps of a hybrid analysis, hence avoiding sensitivity to the random initialization of the clustering process (which is a consequence of the EM type algorithms used in iterative clustering methods). By a 'hybrid' analysis we mean one that is neither exactly hierarchical (in that we conserve the same number of clusters at each stage), but is neither iterative (as we do not assign all data points to a cluster at each stage, but gradually build up the size of the clusters, and in doing so, only allow cluster merges that satisfy the constraints of the fixed cardinality size).

Assuming that a symmetric n by n scaled dissimilarity matrix D exists, with

elements $d_{ij} \in [0,1]$ detailing the dissimilarity between data point $\mathbf{x}_i$ and $\mathbf{x}_j$, the following vanilla algorithm forms clusters $C_1, \ldots, C_k$ with $k = n/m$ by first assigning each $C_r := \phi$ and subsequently proceeding by:

Proposed 'Vanilla' Hybrid Constrained Clustering Algorithm

`count:=1`

`While` $\min\{|C_1|, \ldots, |C_k|\} < m$

`Find` (i,j) `such that` $d_{ij} = \min\{d_{ab} | d_{ab} \in D; a \neq b\}$

`1. If` $\mathbf{x}_i \in C_r$; $\mathbf{x}_j \in C_s$; $|C_r| + |C_s| \leq m$; `then` $C_{\min\{r,s\}} := C_{\min\{r,s\}} \cup C_{\max\{r,s\}}$; $C_{\max\{r,s\}} := \phi$

`2. Else if` $\mathbf{x}_i \in C_r$; $\mathbf{x}_j \notin C_1, \ldots, C_k$; $|C_r| + 1 \leq m$, `then` $C_r := C_r \cup \{\mathbf{x}_j\}$

`3. Else if` $\mathbf{x}_j \in C_r$; $\mathbf{x}_i \notin C_1, \ldots, C_k$; $|C_r| + 1 \leq m$, `then` $C_r := C_r \cup \{\mathbf{x}_i\}$

`4. Else if` $\mathbf{x}_i, \mathbf{x}_j \notin C_1, \ldots, C_k$; $|\{C_s | C_s = \phi\}| \neq 0$; $r = \min\{s | C_s = \phi\}$, `then` $C_r := \{\mathbf{x}_i, \mathbf{x}_j\}$

$d_{ij} :=$ 1+`count`

`count:=count+1`

We describe this as a vanilla algorithm as it is designed only for elucidation of the proposal. It is certainly not the most computationally efficient version which would achieve the same result, *e.g.*, by taking into account knowledge that once a cluster has reached its maximum size it will be fixed, so no consideration of the dissimilarity of any of its elements with any other data point is needed. Further efficiency is gained by noting that if all clusters have data points assigned to them, then the next merger will necessarily involve consideration of the dissimilarity of one of those data points with another point.

The algorithm searches for the two most similar data points that have not thus far been considered, and tries to place them in the same cluster. If the two data points had been previously assigned to separate clusters, then it merges these clusters if the constraints on cluster cardinality remain satisfied. A counter is included as it is possible for nothing to happen to the two data points under consideration, and this will occur if all clusters already have some elements

assigned, but none of these are either of the two data points considered, *i.e.*, there are no empty clusters remaining to which the data points may be assigned. Thus the use of a counter allows memory of the ordering of the dissimilarity rankings should it be necessary to reconsider the two data points at a later stage in the algorithm when possibly one or both of them might have been assigned to a specified cluster, or an empty cluster had been freed-up because of the merging of two other clusters.

Because at each stage the algorithm considers the two most similar data points not so far investigated, it is closely related to hierarchical clustering with single linkage [9], which in turn is closely related to Kruskal's algorithm for minimum spanning trees [7] (see, for example, [5] for details on the relationship). A consequence of this, which we demonstrate below, is that clusters will be formed by capturing dense regions of data points, whilst the remaining data points outside of these regions will be placed in the remaining clusters. This is a very different approach from what traditional cluster analyses would provide, and is also very different to the results that would be found using the constrained clustering approach of Zhu *et al.* In effect, there will be clusters of very compact data points, and other 'potpourri' clusters consisting of the residue with low internal similarity.

The effect of finding clusters of very compact data points, and other clusters consisting of the residue with low internal similarity, is similar in nature to the motivation for the kernel projection classification method of [4], where in order to separate circular classes the data points are projected into a higher-dimensional space where such circular classes are separated by a hyper-plane. Whether or not such a clustering result is of use will depend upon the application. A benefit of the hybrid algorithm is that it will produce clusters much more compact than that found through an approximation to a k-means result, but at the cost of producing others with very low internal similarity.

Typical iterative clustering approaches seek to ensure that the average internal dissimilarity over clusters is minimised, *e.g.*, k-means, whilst the approach outlined here would instead be of use if the objective was not to minimise overall

internal dissimilarity, but to instead ensure that some clusters are as compact as possible (though if a distance measure was used that did not satisfy the triangle inequality, *e.g.*, a quasi-norm such as the p-norm with $p \in (0, 1)$, then the proposed algorithm could reduce average internal cluster dissimilarity). Whichever is the appropriate objective, however, is a choice to be made by the analyst depending on the motivation for the clustering purpose, but in our application of scheduling parallel conference sessions we have found this property to be of benefit.

A further property of the similarity to clustering with single linkage is that it may result in clusters that have a slight overlapping of assignment, meaning that there is no hard border (separating hyper-plane) between cluster membership assignment. This is again in contrast to typical k-means type algorithms. However, should this be found to be problematic for a given application, then it is noted that the situation can be easily resolved by switching the cluster assignments of data points that fell beyond any required hard border either manually, or automatically by calculating the centroid of the dense cluster and by re-assigning its membership to be the closest m data points to this location.

2.1. Artificial Examples

To demonstrate these properties of the algorithm consider data that is two-dimensional and hence may be easily visualized. In particular, consider data that is generated by taking 500 samples from the following bivariate Gaussian distribution:

$$\mathbf{x}_1, \ldots, \mathbf{x}_{500} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 3 & 0.5 \\ 0.5 & 1 \end{pmatrix} \right)$$

Figure 1 (left panel) provides the results of the hybrid clustering algorithm under the fixed constraints that $|C_1| = |C_2| = 250$. The result of a k-means clustering with a value of $k = 2$ is also included for comparison (right panel). As can be observed, the hybrid approach has found a cluster of size 250 from

the densely populated center of the bivariate Gaussian distribution, whilst the residual data points that fall away from the mean of the distribution are assigned to the alternative cluster. In contrast, the k-means approach has split the data by a line running roughly vertically through the highly dense mean of the data (the approach of Zhu *et al.* would then swap assignment of some of the data points close to this boundary so that both clusters were of the same size).

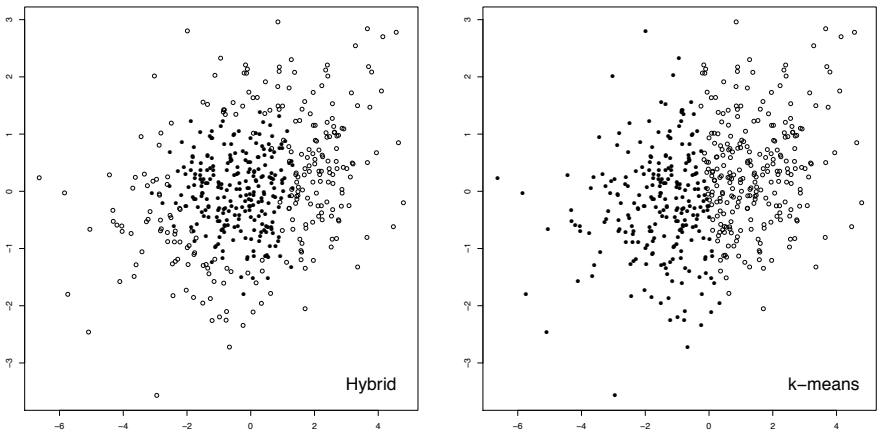

Figure 1: Results of the constrained clustering algorithm (left) requiring 2 clusters of size 250 to be formed from artificial data that had a true structure of 1 cluster from a bivariate Gaussian distribution, and a comparison (right) of the results from a k-means algorithm when $k = 2$.

A more complicated situation would be to consider data generated from multiple distributions, for example:

$$\begin{aligned}
\mathbf{x}_1, \ldots, \mathbf{x}_{50} &\sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 3 & 0.5 \\ 0.5 & 1 \end{pmatrix}\right) \\
\mathbf{x}_{51}, \ldots, \mathbf{x}_{150} &\sim \mathcal{N}\left(\begin{pmatrix} 10 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 & 1.5 \\ 1.5 & 2 \end{pmatrix}\right) \\
\mathbf{x}_{151}, \ldots, \mathbf{x}_{300} &\sim \mathcal{N}\left(\begin{pmatrix} 10 \\ 10 \end{pmatrix}, \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}\right)
\end{aligned}$$

Figure 2 (left panel) plots the results of the hybrid clustering algorithm with constraints $|C_1| = \cdots = |C_6| = 50$ and the results of a k-means clustering with $k = 6$ (right panel). There would appear to be substantial differences in how the two approaches cluster the data. Whilst the hybrid method correctly identifies the cluster size 50 in the bottom left corner, the k-means algorithm splits this into two distinct groups. Indeed, the k-means algorithm splits all three true clusters into half with separating lines running roughly through their means. However, the hybrid algorithm identifies a core dense group around the mode of the bottom-right cluster and another of the residual members of that sample, whilst it finds two core groups in the top-right cluster with a third again consisting of the residual outlying members.

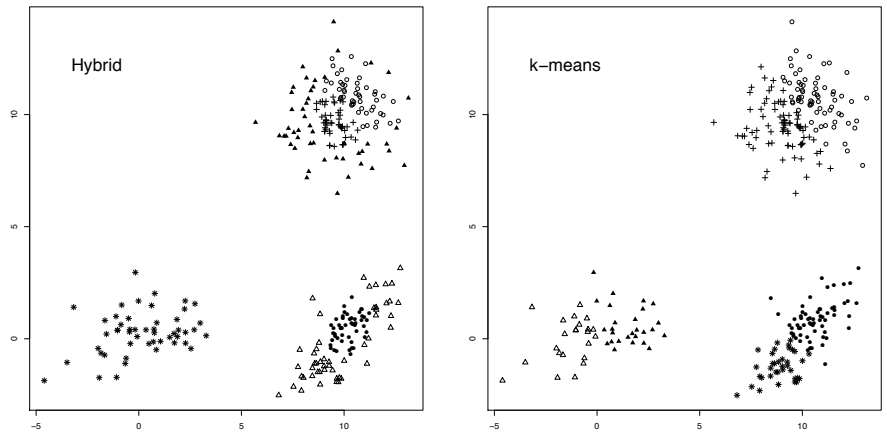

Figure 2: Results of the constrained clustering algorithm (left) requiring 6 clusters of size 50 to be formed from artificial data that had a true structure of 3 clusters from different bivariate Gaussian distributions with respective sizes 50, 100, and 150, and a comparison (right) of the results from a k-means algorithm when $k = 6$.

3. Data and Analysis

The data considered consists of approximately 700 contributed paper applications (approximately 450 oral and 250 poster) that were submitted by potential delegates to the ISI 2011 organizing committee. There were 179 session

topics offered for applicants to consider submitting a presentation to, with each of these being used to describe either an Invited Paper Session (IPS) or a Special Topic Session (STS). For example, IPS 4 concerned 'statistics with high frequency data', whilst IPS 23 concerned 'algebraic statistics'.

As opposed to selecting a single session title, applicants were requested to select up to three possible session titles, hence permitting a measure of similarity between session topics to be determined. For example, one of us (B.H.) submitted a talk titled 'Considerations on the UK Re-Arrest Hazard Data Analysis: How Model Selection can alter Conclusions for Policy Development', and so selected session topics titled 'Social problems, official statistics and social science', 'Risk communication to a lay audience reflecting uncertainty and variability issues', and 'Role of Statisticians in Policy'.

The requirement to submit multiple potential session titles was key to the ability of employing the constrained clustering algorithm of Section 3 for the scheduling of parallel sessions, and would appear to be an important innovation for allowing implementation of an automated data driven clustering of sessions, thus avoiding the requirement of program committees and chairs manually having to timetable the conference from scratch. Feasible alternative strategies that could also have been implemented, and which would also support use of the hybrid constrained clustering algorithm, could include the predefinition of a collection of key words such as 'official statistics', 'clustering methods', or 'industry implementation' *etc.* The restriction to just selecting three possibilities is also not necessary, and indeed, allowing potential participants to select more than this number would have increased reliability of the results.

To create a similarity measure of session topics we considered how frequently the same two topics were included in an applicant's list of potential topics. The motivation here is that, if two session titles are never included in any applicant's list of three potential topics, then they can be considered very dissimilar. In contrast, if whenever one session title is selected it was always included with a specified other session title, then these would be considered very similar (and ideally, should not be held at the same time in different venues). Initially each

pair of session topics was said to have a similarity of 0, but whenever an applicant listed two different topics in their list of three potential topics the similarity between these topics was incremented by 1 (these values could then have been scaled so as to take into account relative popularity of topic titles, though this was not implemented here). Finally, the similarity values were placed on a 0-1 scale.

For example, no applicant simultaneously listed both IPS 100 'copula-based advances in hydrologic engineering', with STS 118 'measuring speculative capital flows (*e.g.* carry trades)', meaning these session topics had a similarity value of 0. The most common duplicate listing, however, was between IPS 7 'inference for stochastic processes' and IPS 115 'recent advances in time series', meaning these session topics were given a similarity score of 1.

Once a similarity score over session titles had been created, a dissimilarity measure between applicants could be estimated by noting the similarity between their suggested session titles and taking the average of these. For example, one applicant had listed IPS 97, IPS 89, and IPS 100 (respectively, 'water - extremes', 'hydrological processes', and 'copula-based advances in hydrologic engineering')[1], whilst another applicant listed IPS 55, IPS 100, and IPS 76 (respectively, 'inference for linked data', 'copula-based advances in hydrologic engineering', and 'fuzzy data and statistics'). This meant the dissimilarity between these two applicants was:

$$\begin{aligned} d(\mathbf{x}_i, \mathbf{x}_j) \quad \propto \quad & d(\text{IPS } 97, \text{IPS } 55)(=0) + d(\text{IPS } 97, \text{IPS } 100)(=0.06) \\ & + d(\text{IPS } 97, \text{IPS } 76)(=0) + d(\text{IPS } 89, \text{IPS } 55)(=0) \\ & + d(\text{IPS } 89, \text{IPS } 100)(=0.12) + d(\text{IPS } 89, \text{IPS } 76)(=0) \\ & + d(\text{IPS100}, \text{IPS } 55)(=0.06) + d(\text{IPS } 100, \text{IPS } 100)(=1) \\ & + d(\text{IPS } 100, \text{IPS } 76)(=0.06) \end{aligned}$$

[1]Note the 2011 World Statistics Congress had a special Theme Day, during which all papers addressed, from various statistical perspectives, issues in 'Water, quality and quantity'.

Using such a dissimilarity measure, the hybrid clustering algorithm outlined in Section 3 was applied to provide a first clustering result (which once provided eases the task of making any manual adjustment). Because the hybrid algorithm has an approach of finding clusters of closely related data points, and others of residual data points, sessions could be organized so that similar topics were included in the same schedule, whilst other sessions would consist of 'what was left', *i.e.*, the collection of talks that may be on niche areas that are not subject to large investigation within either the academic, government, or private sector communities.

Finally, estimates of the internal dissimilarity of a cluster, and the dissimilarity between clusters, can be calculated by considering the average dissimilarity between elements within a cluster (internal dissimilarity), and pairwise combinations of elements between clusters. The latter then allows a visual representation of cluster dissimilarity by performing a Multi-Dimensional Scaling in two-dimensions [2] (*i.e.*, by calculating a two-dimensional co-ordinate configuration of the clusters so that the dissimilarity within this two-dimensional configuration matches that of the original dissimilarity matrix as closely as possible). Figure 3 plots a two-dimensional non-metric Multi-Dimensional Scaling of the resulting clustering results where 679 proposals were clustered into 97 sessions of size 7. Such a visualization tool is of benefit in timetabling sessions, as it allows easy inspection of which cluster sessions should not be run at the same time.

4. Discussion

We have made a novel attempt at solving a little considered problem of cluster analysis under a constraint of cardinality, thus allowing us to approach a particular problem of scheduling parallel sessions within a conference. In general, however, the problem of clustering under constrained cardinality can be investigated much further, and is likely to be applicable and of interest in a wide variety of situations of timetabling problems.

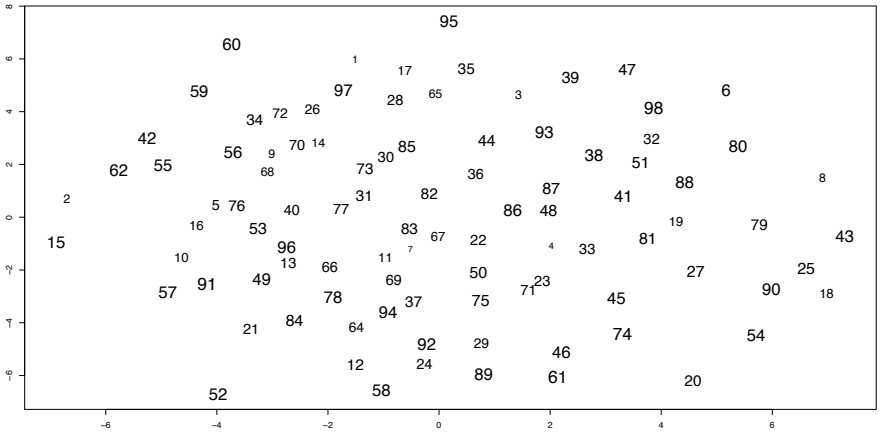

Figure 3: A two-dimensional non-metric Multi-Dimensional Scaling of the resulting clustering results. Group numbers that are closer to each other are more similar, whilst the size of the group number reflects the level of internal dissimilarity within the group, with the smaller the font the more compact the group.

Our approach was to consider a hybrid algorithm that is intuitive and simple. However, it does not guarantee an 'optimal solution', as we have not specified an objective that any discussion of optimality can be considered against. One possibility is that clusters which are meant to be within the core regions of groups have minimal internal dissimilarity. We do not currently guarantee that this is the case as we allow soft cluster borders, so one future direction may be to take this into account. Another further direction maybe to generalize the constraints on cluster cardinality so that, rather than they having to take specific and predefined values, they are only restricted in their orderings, *e.g.*, $|C_1| \leq |C_2| \leq \cdots \leq |C_k|$.

Acknowledgments

The authors would like to thank Ronald Wasserstein for comments on the practice of timetabling meetings of the American Statistical Association. B.H. is funded by the STATICA project, a Principle Investigator program of Science Foundation Ireland (08/IN.1/I1879). This work is subject to a U.S. Patent

application titled "Method and System for Scheduling of Events" (application number 61/526,221).

References

[1] Basu, S., Davidson, I., & Wagstaff, K.L. (eds.) (2009). *Constrained Clustering: Advances in Algorithms, Theory, and Applications*. Chapman & Hall/CRC.

[2] Cox, T.F., & Cox, M.A.A (2001). *Multidimensional Scaling*. Chapman & Hall/CRC.

[3] Dempster, A.P., Laird, N.M. & Rubin, D,B, (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society Series B (Methodology), 39*, 1:38.

[4] Domijan, K. & Wilson, S.P. (2011). Bayesian Kernel Projections for Classification of High Dimensional Data. *Statistics and Computing, 21*, 203:216.

[5] Gower, J.C. & Ross, G.J.S. (1969). Minimum Spanning Trees and Single Linkage Cluster Analysis. *Journal of the Royal Statistical Society Series C (Applied Statistics), 18*, 54:64.

[6] Kaufman, L. & Rousseeuw, P.J. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley.

[7] Kruskal, J.B. (1956). On the Shortest Subtree of a Graph and the Traveling Salesman Problem. *Proceedings of the American Mathematical Society, 7*, 48:50.

[8] Lloyd, S.P. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory, 28*, 129:137.

[9] Sneath, P.H. (1957). Computers in taxonomy. *Journal of General Microbiology, 17*, 201:226.

[10] Ward, J.H. (1963). Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association, 58*, 236:244.

[11] Zhu, S. Wang, D., & Li, T. (2010). Data Clustering with Size Constraints. *Knowledge-Based Systems, 23*, 883:889.