ORIGINAL RESEARCH ARTICLE

An Efficient and Accurate Numerical Determination of the Cluster Resolution Metric in Two Dimensions

Michael Sorochan Armstrong[1*] | A. Paulina de la Mata, PhD[1*] | James J. Harynuk, PhD[1*]

[1]Department of Chemistry, University of Alberta, Edmonton, Alberta, T6G 2G2, Canada

Correspondence
James J. Harynuk, PhD, Department of Chemistry, University of Alberta, Edmonton, Alberta, T6G 2G2, Canada
Email: james.harynuk@ualberta.ca

Funding information
Natural Sciences and Engineering Research Council of Canada (NSERC), Discovery Grant RGPIN-2016-05445; Canada Foundation for Innovation, Major Science Initiatives Fund (CFI-MSI) and Genome Canada / Genome Alberta, Genomics Technology Platforms (GTP) awards to The Metabolomics Innovation Centre (TMIC)

Cluster resolution (CR) is a useful metric for guiding automated feature selection of classification models. CR is a measure of class separation in a linear subspace for variable subsets via the determination of maximal, non-intersecting confidence ellipses [1][2][3]. Feature Selection by Cluster Resolution (FS-CR) is most commonly used to extract panels of useful, discriminating features from sparsely populated chromatographic peak tables, optimizing models from raw signals, or when working with data sets with many more variables than samples. The absence of a numerical method for calculating cluster resolution necessitates a great deal of dynamic programming and algorithmic complexity [4]. In this work, we present a numerical determination of the cluster resolution metric, which reduces computation time by about 65 times when compared with the dynamic programming approach, and simplifies the operating principles of FS-CR algorithm.

1 | INTRODUCTION

Modern platforms for chemical analysis, particularly chromatographic separations with mass spectral detection, generate an abundance of data. Modern liquid chromatography - tandem mass spectrometry (LC-MS/MS) or comprehensive two-dimensional gas chromatography - time-of-flight mass spectrometry (GC×GC-TOFMS), often reveal thousands of unique compounds in complex samples. Within such data sets, there are a typically a relatively small number of variables which contain the useful information required for discrimination between classes of samples, obscured by many more variables that hinder predictive modeling capabilities. Problems such as these are frequently observed when modelling raw spectroscopic data (FT-IR, Raman, NMR, etc), raw chromatographic signals, and peak table data from a series of complex samples, as frequently encountered in metabolomics data sets.

Feature selection seeks to identify a useful subset of variables in the data for inclusion, while excluding variables that do not contribute to model performance. Ideally, feature selection will identify a subset of variables that yield a rugged model that is resistant to noise and easy to interpret.

A number of variable selection techniques exist, which can be broadly classified as belonging to filter, wrapper, or embedded-type methods. All of these techniques have relative advantages and disadvantages[5] [6]. Briefly, filter methods are easy to apply to many different types of data, but require careful optimization of thresholds. Selection based on a Fisher-ratio threshold is a common example of a filter method [7] [8] [9]. The approach is fast, but since it evaluates each variable independently, it cannot account for relationships between correlated variables. Selectivity ratios and Variable Importance in Projection (VIP) scores are metrics for feature selection which do consider each variable in the context of others - either by examining the weighted variable correlation with the vector of observed values, y, within its projection to the latent variable space (in the case of VIP scores) or the ratio of variance explained to residual variance (in the case of Selectivity Ratios) [10]. Wrapper-type methods reduce user intervention through automated model quality assessment as different combinations of variables and samples are tested and validated, but may still require carefully optimised user parameters. These also require large numbers of iterations in order to find an optimal variable subset. Recursive weighted Partial Least Squares (rPLS) [11] is a modern implementation of a wrapper-type method [12], but methods such as Genetic Algorithms (GA) [13] and Random Forests (RF) [14] have been used as variable selection routines within the framework of wrapper methods as well. Embedded methods incorporate an extra step to make decisions about variable selection during model calculation, independent of model quality assessment. Powered partial least squares discriminant analysis is an example of this technique [15], but embedded methods based on classification by Support Vector Machines (SVM) are also applied [16]. Ideally, this approach makes objective decisions about variable selection, and reduces the dependency on extensive cross-validation, but the extra step increases the computation time required for these techniques. Embedded methods are not as extensively used, perhaps because they assess variable significance based on optimisation criteria instead of a statistical measure of performance [17].

Hybrid variable selection routines incorporate elements of 2 or more classes of variable selection, most commonly to reach a compromise between the simplicity of threshold methods, and the reliability of wrapper methods [18] [19]. An initial subset of variables exceeding a particular threshold are considered, and model performance is evaluated once per iteration or multiple times either by adding (forward-selection) or removing (backwards-elimination) high-ranking variables until the predictive ability of the model is no longer improved. Variables that are consistently retained across multiple cross-validation sets are retained, and others are discarded.

Principal Component Analysis (PCA) is a data reduction technique that can be used to reduce the complexity of the feature selection problem, and a qualitative tool often used by investigators to examine the effect of variable selection on the most significant axes of variance in the data. Classes well-separated along their most significant principal components typically perform well using a targeted discrimination technique such as Linear Discriminant Analysis (LDA), Partial Least Squares - Discriminant Analysis (PLS-DA), or Support Vector Machines (SVM). As such, it is often convenient to examine variable subsets within their principal component space [20] [21].

1.1 | Feature Selection by Cluster Resolution (FS-CR)

Feature selection by cluster resolution (FS-CR) is a supervised learning technique that returns a subset of variables whose linear combination provides the best possible separation between two or more sample classes in principal component (PCA) space as determined by the Cluster Resolution (CR) metric. There are two basic assumptions for the operation of the algorithm: 1 - That a useful, discriminating subset of variables is of relatively low rank, and can be adequately described using only a few principal components, and 2 - that samples resolved along relatively few principal components are trivial to separate using supervised classification methods.

The latest implementation of FS-CR operates within the framework of a hybrid filter/wrapper method with a combination of backwards-elimination and forward-selection to consider individual variables within the context of others. Variables are first ranked, typically through the application of either Fisher ratio [22] or selectivity ratio[23] [3], and the algorithm evaluates candidate variables via a hybrid backwards-elimination / forward-selection routine [4]. The initial population of variables to be included in the preliminary model is determined through analysis of the true and null distributions of ranking metric values [21]. Backwards-elimination proceeds by sequentially removing variables beginning with the lowest-ranked (based on ranking metric) variable and working towards the highest-ranked. If CR improves when a variable is removed, that variable is permanently discarded; otherwise it is returned and permanently retained. This proceeds until the entire initial population of variables has been tested. Forward-selection is then performed, testing as-yet unconsidered variables to see if their inclusion improves the model based on the variables that survived the backwards-elimination step. Forward-selection proceeds from the highest-ranked variable that was not included in the initial population being considered until a stop condition is met [21].

The FS-CR algorithm has several advantages, including the fact that the utility of a variable is evaluated in the context of the information provided by other variables (unlike in methods such as a Fisher-ratio cut-off threshold). Unlike feature selection methods based on a partial least squares regression, variables are considered in an unsupervised projection to their principal component space. This helps to reduce to risk of over-fitting because the model is not seeking to impose a favourable projection on the data, but is seeking to find a group of variables that naturally lend themselves to favourable projections via PCA[1]. The results can also be thoroughly cross-validated by redistributing the samples among the training (data which is used to calculate the principal components), optimization (data for which CR is calculated within the previously calculated principal component space), and validation (data that measures the predictive accuracy of the variables) sets through multiple iterations of the algorithm and retaining only those variables that survive in a given fraction of all iterations (typically 75-90%). Other hybrid feature selection routines are structurally similar to the FS-CR algorithm in this regard, but a metric describing the accuracy of cross-validation results is more commonly employed to assess variable subsets. FS-CR employs the cluster resolution metric to assess variable subsets. Cluster resolution is defined as the maximum confidence interval over which two or more classes can be separated when projected into the principal component space of the candidate feature subset. This is more sensitive to favourable orientations of sample scores, and is much less granular than cross-validation results alone.

FS-CR has been successfully deployed for datasets where the number of features greatly exceeds the number of samples, and in cases where there are many pre-processing artefacts or spurious signals. These situations often arise with weak signals close to the detection limits of analytical instrumentation or in data derived from highly variable populations of samples (often encountered in natural products, petroleum, and metabolomics samples). Two challenges for the application of FS-CR include the need for a relatively large number of samples (30 per class is a typical minimum) so that they can be properly partitioned into training, optimization, and validation sets, and the relative slowness of the cluster resolution calculation which is performed ${}_{n}C_{2}$ times for each variable being tested where n is the number of classes in the optimization problem. Calculation of CR scales poorly for multi-class problems: for a three-class problem, CR is calculated three times for each variable considered, and in datasets with seven classes, CR is calculated 21 times per variable [3], as the overall CR for the iteration is calculated as the product of the individual pair-wise binary combinations of different classes. This puts a practical limitation on the number of classes that can be analyzed within a reasonable time-frame using this technique.

The utility of CR in four or more dimensions is currently unknown. The volumes and surface areas of N-spheres are maximal at 5 and 7 dimensions, respectively [24]. This may have implications for the determination of higher-order cluster resolution, especially when considering that the axes of confidence ellipses decrease as a function of explained variance. However, the proposed mathematical formalisation of the cluster resolution metric does not prohibit similar determinations of CR for any number of dimensions, or principal components. Evaluating the utility of this formalisation in two dimensions is an important first step in addressing these questions, which will be explored in future works.

2 | THEORY

2.1 | Cluster Resolution

Cluster resolution is defined as maximum size of confidence ellipses (or ellipsoids when operating in higher-dimensional space) that can be described about the scores for the samples in a pair of classes projected into a linear space, without the ellipses overlapping, and is bounded between 0 and 1. The determination of cluster resolution has so far utilised extensive dynamic programming. In addition to being slow, dynamic programming is mathematically unsatisfying. The method works by calculating a number of points along confidence ellipses projected within two or three principal components, and uses graphical methods to determine whether or not they intersect. Improvements to the efficiency of this method have been made by "hopping"[4] between intervals where the ellipses do intersect, and where the ellipses do not intersect. This reduces the number of iterations of the algorithm, and solves some issues regarding the granularity of confidence ellipse calculations. However there is still the need to determine the coordinates of many points multiple times for each calculation of cluster resolution, and the accuracy scales with the computational workload required for a properly representative graphical determination.

2.2 | Mathematical Description of Confidence Ellipses

For two uncorrelated score vectors in principal component space, confidence ellipses for one class follow the form [25]:

$$\left(\frac{\mathbf{T}_1}{\sqrt{S_1}}\right)^2 + \left(\frac{\mathbf{T}_2}{\sqrt{S_2}}\right)^2 = \chi^2 \tag{1}$$

Where T_1, and T_2 are vectors containing the first and second principal component scores of the confidence ellipsis, S_1 and S_2 are the variances associated with each principal component, and χ^2 corresponds to the size of the ellipse for a given confidence interval, as defined by the χ^2 distribution. Equation 1 can be rewritten in more general parametric form:

$$\mathbf{T}_1 = T_1^0 + \sqrt{\Lambda\chi^2}\cos(\theta) \tag{2}$$

$$\mathbf{T}_2 = T_2^0 + \sqrt{\lambda\chi^2}\sin(\theta) \tag{3}$$

In Equation 2, T_1^0 and T_2^0 refer to the mean of each cluster along the first and second principal components. Λ, and λ describe the major and minor eigenvalues, which correspond to the variance of the data, and θ encompasses the angle associated for a given point along the ellipse. For the majority of cases, where T_1, and T_2 are not completely uncorrelated, the angular components of Equations 2 and 3 are multiplied by a rotation matrix, R_s, as a function of an angle ϕ. ϕ is calculated as the angle between the major eigenvector of the ellipse, $\boldsymbol{v_1}$, relative to the first principal component of the data:

$$\phi = \arctan\frac{\boldsymbol{v_1}(2)}{\boldsymbol{v_1}(1)} \tag{4}$$

Where R_s for a two-dimensional case follows:

$$R_s = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \tag{5}$$

Previously, CR was calculated by increasing χ^2 by the same factor for each of the two confidence ellipses until the point where a collision between the two ellipses was detected (or χ^2 was decreased similarly until a collision no longer occurred). This requires several hundred points around the ellipse to be calculated for each increase of χ^2. Accuracy is improved by increasing the granularity of the χ^2 expansion and/or the number of points along the ellipse, at the expense of computation time. While this is a reliable method of determining cluster resolution, it is computationally costly. Consequently, in this work a numerical solution through minimization of some cost function is sought.

2.3 | Derivation of a Numerical Solution

The intersection of two confidence ellipses can be described as the intersection of two lines, for a pair of angles that stem from the centre of each confidence ellipse.

$$\begin{bmatrix} T_1^1 \\ T_2^1 \end{bmatrix} + \sqrt{\chi_1^2}\begin{bmatrix} u_1 \\ v_1 \end{bmatrix} = \begin{bmatrix} T_1^2 \\ T_2^2 \end{bmatrix} + \sqrt{\chi_2^2}\begin{bmatrix} u_2 \\ v_2 \end{bmatrix} \tag{6}$$

Where T_j^i refers to the i^{th} confidence ellipse centre for the j^{th} principal component scores. Vector components u, v are shorthand for the following expansions from Equations 2 and 5:

$$u_1 = \sqrt{\Lambda_1}\cos\theta_1\cos\phi_1 - \sqrt{\lambda_1}\sin\theta_1\sin\phi_1 \tag{7}$$

$$u_2 = \sqrt{\Lambda_2}\cos\theta_2\cos\phi_2 - \sqrt{\lambda_2}\sin\theta_2\sin\phi_2 \tag{8}$$

$$v_1 = \sqrt{\Lambda_1}\cos\theta_1 \sin\phi_1 + \sqrt{\lambda_1}\sin\theta_1 \cos\phi_1 \tag{9}$$

$$v_2 = \sqrt{\Lambda_2}\cos\theta_2 \sin\phi_2 + \sqrt{\lambda_2}\sin\theta_2 \cos\phi_2 \tag{10}$$

For almost any pair of θ_1, θ_2, depending on their positions relative to the angle of the ellipses, ϕ_1, ϕ_2, χ_1^2, and χ_2^2 can be solved for by rearranging Equation 6.

$$\frac{1}{-(u_1 v_2) + u_2 v_1}\begin{bmatrix} -v_2 & u_2 \\ -v_1 & u_1 \end{bmatrix}\begin{bmatrix} T_1^2 - T_1^1 \\ T_2^2 - T_1^2 \end{bmatrix} = \begin{bmatrix} \sqrt{\chi_1^2} \\ \sqrt{\chi_2^2} \end{bmatrix} \tag{11}$$

The euclidean norm of Equation 11 can be used to constrain the problem as the minimization of a cost function:

$$\min f(\theta_1, \theta_2) = \sqrt{\left(\sqrt{\chi_1^2}\right)^2 + \left(\sqrt{\chi_2^2}\right)^2} \tag{12}$$

2.4 | Practical Considerations

For an accurate numerical solution to the cluster resolution problem, at the minimum of Equation 11, χ_1^2 ought to be equal to χ_2^2. By minimizing Equation 12, the results often approach this equality. Ideally, a solution is found when $\partial/\partial\theta_1 = 0$ and $\partial/\partial\theta_2 = 0$, such that the intersection of two lines at the minimum of Equation 12 becomes:

$$\sqrt{\chi_1^2}\begin{bmatrix} \partial u_1/\partial\theta_1 \\ \partial v_1/\partial\theta_1 \end{bmatrix} = 0 \tag{13}$$

$$\sqrt{\chi_2^2}\begin{bmatrix} \partial u_2/\partial\theta_2 \\ \partial v_2/\partial\theta_2 \end{bmatrix} = 0 \tag{14}$$

Setting Equations 13 and 14 equal to each other and expanding the differentials yields:

$$\sqrt{\chi_1^2}\begin{bmatrix} -\sqrt{\Lambda_1}\sin\theta_1 \cos\phi_1 - \sqrt{\lambda_1}\cos\theta_1 \sin\phi_1 \\ -\sqrt{\Lambda_1}\sin\theta_1 \sin\phi_1 + \sqrt{\lambda_1}\cos\theta_1 \cos\phi_1 \end{bmatrix} = \sqrt{\chi_2^2}\begin{bmatrix} -\sqrt{\Lambda_2}\sin\theta_2 \cos\phi_2 - \sqrt{\lambda_2}\cos\theta_2 \sin\phi_2 \\ -\sqrt{\Lambda_2}\sin\theta_2 \sin\phi_2 + \sqrt{\lambda_2}\cos\theta_2 \cos\phi_2 \end{bmatrix} \tag{15}$$

It is clear that a solution for θ_1, θ_2 can be found that satisfies Equation 15 as a system of nonlinear equations. However, there is no guarantee that a solution to this system of equations would minimize Equation 12 nor would a minimum of Equation 12 necessarily satisfy Equation 15. It is possible to find a minimum subject to the constraints of 13 and 14 via Lagrange's method; however, it was shown to be computationally inefficient, and unstable given the complexity of the equations involved, and the potential for undifferentiable points on the optimization surface (i.e. for two lines parallel to each other such that the χ^2 at which they converge is undefined). The accuracy of the algorithm is therefore somewhat limited, but it will be shown that minimizing Equation 12 yields a workable approximation by calculating an intermediate of the upper and lower bounds of χ^2 via the mean:

$$\chi^2_{mean} = \min f^2/2 \tag{16}$$

The confidence interval (defined as cluster resolution (ξ) for this particular problem) is calculated from the cumulative χ^2 distribution function with two degrees of freedom (DOF) using the $chi2cdf$ function in the MATLAB ® Statistics and Machine Learning Toolbox [26]. Where:

$$\xi = F(x|\nu) = \int_0^x \frac{t^{(\nu-2)/2}e^{-t/2}}{2^{\nu/2}\Gamma(\nu/2)}dt \tag{17}$$

In Equation 17, ν refers to DOF, Γ is the Gamma function, and x is the input χ^2_{mean} value from Equation 16.

3 | MATERIALS AND METHODS

3.1 | Implementation

Equation 12 was minimized using an implementation of the Nedler-Mead Simplex algorithm available as *fminsearch* in MATLAB ® 2018b (64 bit)[27]. This algorithm outperformed its equivalent quasi-Newtonian counterparts, both in terms of the reliability and speed of its convergence rate, due in part to its ability to operate without the need for an analytical determination of the gradient at each iteration. The tolerance for convergence was set at 2.5×10^{-6} for the numerical experiments, and 1×10^{-8} for the classification data. Randomly generated two-dimensional data, simulating scores in the first and second principal components for a balanced dataset were generated (See Supporting Information). The original (dynamic programming) and new numerical approach to determination of cluster resolution were applied to the data. All computations were performed on a Lenovo ThinkCentre M700 running Ubuntu 18.04 LTS "Bionic Beaver" with 8 Gb RAM, and an Intel i3-6100T CPU @ 3.20 GHz.

The most recent implementation of the FS-CR algorithm was used for selecting discriminating features in the experimental data using both the current dynamic programming, and proposed numerical method for determining CR. Variables were ranked using Fisher Ratios and populations for backwards elimination and forward selection were calculated using experimental true and null distributions of significant features [21]. The numerical implementation of the CR algorithm has been made freely available online: 10.5281/zenodo.4064280.

3.2 | Experimental Data

A dataset comprising the volatile organic chemical signatures of 162 samples of cotton and polyester fabrics recovered from a wear trial, wherein participants each wore bi-symmetrical shirts comprised of one-half cotton, and one-half polyester fabric was used to compare the algorithms with real data. Details of the wear trial can be found elsewhere [28], but the stated goal of the analysis was to find discriminating chemical signatures between the cotton and polyester samples indicating which compounds were particularly well-retained on the different fabrics following multiple wear-wash cycles. Both washed and unwashed samples were categorised only as belonging to either the cotton or polyester classes. The sampling was performed using Solid Phase Micro-Extraction (SPME) fibres with a “tri-mode” divinylbenzene/carboxen/polydimethylsiloxane (DVB/CAR/PDMS) extraction phase (SUPELCO, Bellefonte, PA). Extractions were performed on the headspace of 2.0 × 2.0 (± 0.2 cm) samples of fabric, sealed within 10 mL crimp-top vials at 30 °C for 21 h. The potentially large in-class variation makes for a somewhat challenging dataset for classification.

4 | RESULTS AND DISCUSSION

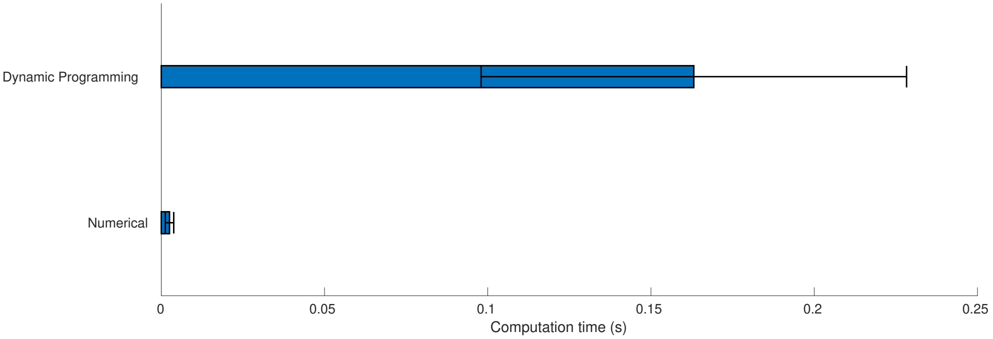

FIGURE 1 Average computation times for numerical and dynamic programming determinations of cluster resolution for two clusters. Error bars indicate ±s.

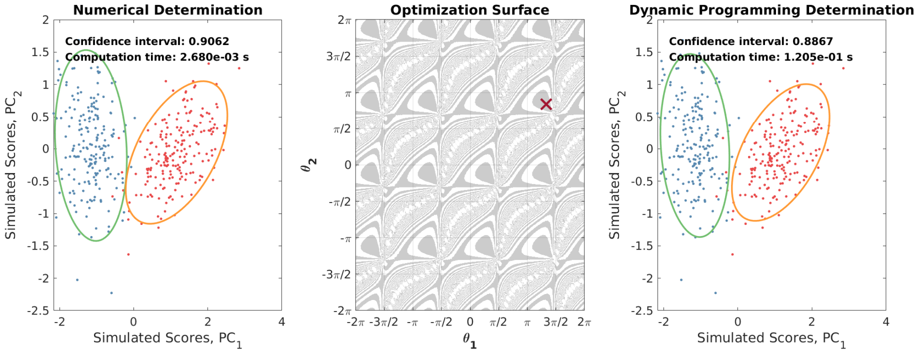

FIGURE 2 Example calculation, comparing the numerical and dynamic programming implementation of the cluster resolution metric, with the corresponding optimization surface. "X" is the location of the optimum values for θ_1, and θ_2 found by the Nedler-Mead Simplex algorithm via the minimization of Equation 12.

In the absence of an analytical solution to the cluster resolution metric, a numerical experiment consisting of 200 randomly generated data sets was used to compare the performance of the numerical implementation of the algorithm with the current version. The dynamic programming implementation requires an initial guess for CR, and 0.75 was used, as this is a typical initial guess used in practice. The numerical solution does not require an initial guess for CR. An example solution and the corresponding optimization surface for the numerical method is shown in Figure 2. For N = 200 sets of randomly generated data, the average time required to calculate cluster resolution was 2 ± 1 ms using the numerical method, and 163 ± 65 ms using the current dynamic programming approach. For two clusters

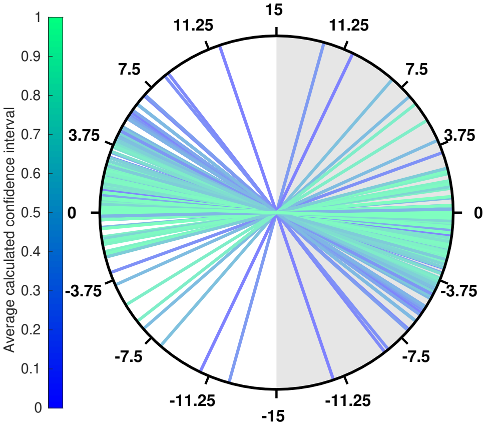

FIGURE 3 Secant plot comparing the relative percent differences in the determination of cluster resolution using the numerical (left, white hemisphere) vs. dynamic programming (right, grey hemisphere) approach for data with sample classes. All calculations agreed within a 15% relative error. Lines intersecting with the edge of the circle at the left side of the circle indicate the % relative difference of the numerical method for the determination of cluster resolution, for an averaged value for cluster resolution indicated by the colour of the line - similar to the intersection with the right hand side of the circle, expect with respect to the % relative difference for the dynamic programming approach.

in two-dimensional space, the numerical method is on average 65 times faster (Figure 1). The two methods provided similar results that agreed within 15% (3) and the ellipses do not appear to significantly overlap in any of the solutions from the numerical method (See: Supporting Information 1). Comparing the two methods with a secant plot (Figure 3) shows the tendency for the the dynamic programming approach to underestimate cluster resolution vs. the new, numerical approach.

4.1 | Calculation of the cluster resolution metric for *N* clusters

The number of times cluster resolution is calculated per evaluated variable depends on the binomial coefficient, ${}_nC_k$ where $k = 2$ and n is the number of sample classes. This is due to the fact that it is necessary to calculate the cluster resolution between each pair of classes to evaluate the overall cluster resolution for the model (Ξ). Consequently, computation time scales poorly for variable selection problems with more than two classes. In general, the overall cluster resolution is calculated as the product of the individual cluster resolutions for each possible combination of clusters:

$$\Xi = \prod_{n=2}^{{}_nC_2} \xi \binom{n}{2} \tag{18}$$

To compare the compounded improvement for multi-class problems offered by the numerical approach vs. the current approach, randomly generated data sets simulating n-class problems were generated as before ($2 < n < 7$). 30 data sets were simulated per value of n. Results are summarized in Figure (4). For the 7-class problem, a single Ξ calculation required 3.3 ± 1 s using the dynamic programming approach, and 0.040 ± 0.005 s using the numerical approach, an 82-fold improvement.

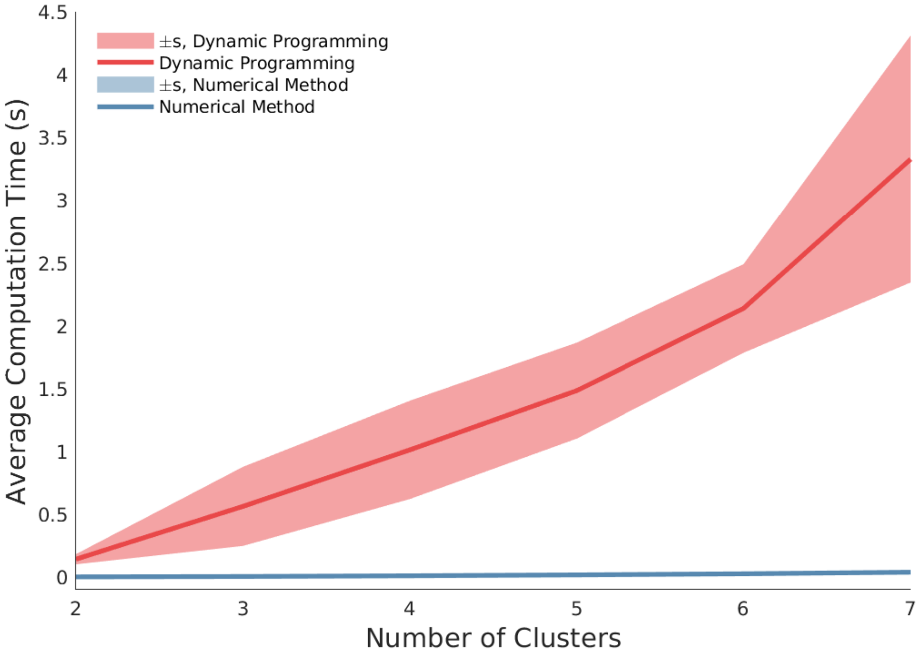

FIGURE 4 Average computation time for Ξ in N-class problems for $2 < N < 7$ for numerical and dynamic programming methods. s refers to sample standard deviation for the dynamic and numerical computation times.

4.2 | Comparison of Predictive Capabilities

Feature Selection by Cluster Resolution has proven to be extremely useful for selecting useful subsets of sparse datasets, typical of peak tables generated GC×GC-TOFMS data, and so one such dataset was used from a previous study [29]. The dataset is available at: `https://doi.org/10.7939/DVN/RLMSRW`.

Partial Least Squares Discriminant Analysis (PLS-DA) was used to generate a classification model; the discrimination threshold for predicted Y scores was generated using a Bayesian technique [30]. The external validation set was used to evaluate prediction results, following strict class membership assignment designated by the aforementioned threshold. Results for predictions were made using PLS-DA without feature selection (Figure 5, row 1), with the current dynamic programming (DP-FS-CR) implementation (Figure 5, row 2), and the numerical (NM-FS-CR) implementation (Figure 5, row 3), and summarized using predicted Receiver-Operator Characteristics (ROC) and prediction accuracy, where prediction accuracy is defined as the ratio of the sum of the true positive rate (TP) and true negative rate (TN) over the sum of all prediction rates including the false positive (FP) and false negative (FN) rates:

$$Accuracy = (TP + TN)/(TP + FP + TN + FN) \tag{19}$$

The dataset contains a high number of replicates, and was divided evenly between training and validation sets for a critical analysis of each models' predictive ability. External validation samples were centred and scaled according to values calculated in the training set. All results are presented with respect to Class 1 (cotton samples) versus Class 0 (polyester samples). Within the training set, 200 combinations of training and optimization sets were generated and used for both DP-FS-CR and NM-FS-CR routines, with variables selected at least 90% of the time across all sample combinations being included in the final feature subset. In the training set, there were a total of 81 samples: 63 class 1, and 18 class 0. In the validation set there were also a total of 81 samples, with 61 class 1 and 20 class 0 samples.

Results from the confusion matrices and predicted ROC curves suggest that the variables selected using NM-FS-CR perform better than the much slower DP-FS-CR algorithm in terms of predictive ability. DP-FS-CR selected a total of 32 variables, and NM-FS-CR selected a total of 46, with total computation times of 14760 and 1468.3 seconds respectively. All but 2 variables that were selected using DP-FS-CR were also selected using NM-FS-CR (Table 1), in addition to 16 variables that were unique to the NM-FS-CR method. It was previously shown that using a simple thresholding method for feature selection with this dataset was unsuccessful [8].

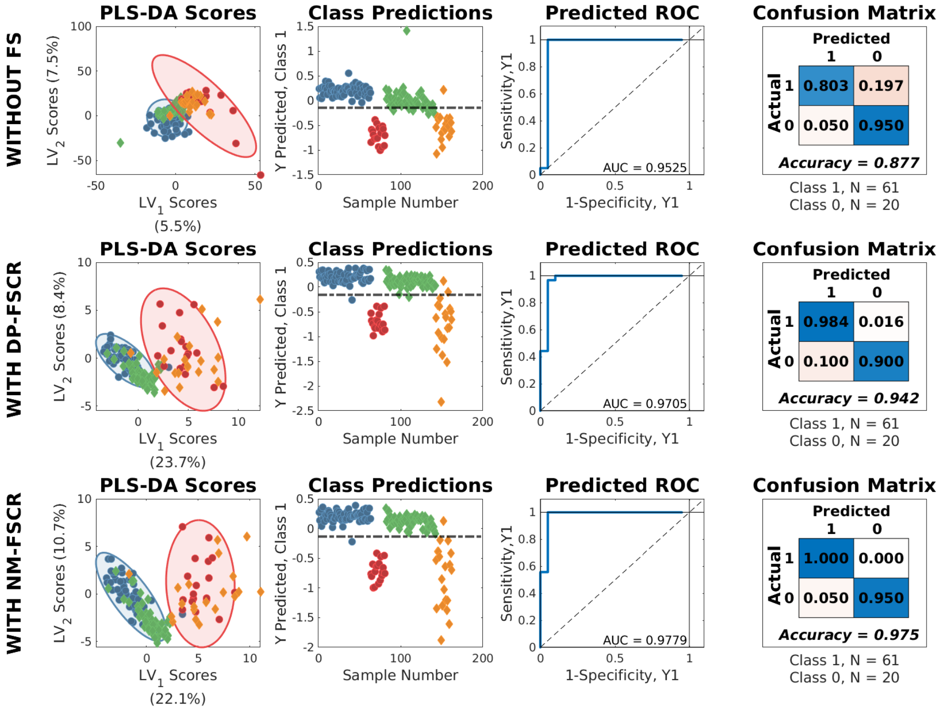

FIGURE 5 Summary of classification results using PLS-DA, DP-FS-CR, and NM-FS-CR. Confidence ellipses are displayed for a confidence interval of 95%

5 | CONCLUSIONS

Cluster resolution is a useful metric for evaluating model quality and guiding variable selection routines. Cluster resolution permits consideration of favourable changes to the relative positions and orientations of score clusters representing the distribution of sample classes in principal component space, without relying solely on cross-validation results as is done with other methods. Previously, there existed no mathematical formalization of the cluster resolution metric, and its determination relied on dynamic programming. The speed and accuracy of the dynamic programming method depends primarily on the number of points used in the confidence ellipse projections, and a reasonable initial guess for the cluster resolution. The numerical solution to the calculation of cluster resolution presented herein has demonstrated a substantial improvement in computation speed, while generally maintaining or improving the accuracy of the calculation. Preliminary results show that the variables selected using the new numerical determination largely encompass the variables selected using the dynamic programming approach, while also identifying additional useful variables. Including these previously hidden variables has been shown to improve the predicted ROC and prediction accuracy of the model.

The improvements in speed make it feasible to analyse many more combinations of training and optimisation sets,

DP-FS-CR	NM-FS-CR	Common
32	46	30
1	1	1
3	3	3
4	4	4
	5	
9	9	9
11	11	11
21	21	21
22	22	22
23	23	23
28		
30	30	30
35	35	35
	54	
	56	
	69	
75	75	75
76	76	76
78	78	78
79	79	79
84	84	84
85	85	85
	109	
123	123	123
141	141	141
148	148	148
	165	
228		
236	236	236
	260	
	280	
308	308	308
336	336	336
458	458	458
483	483	483
610	610	610
	662	
	806	
	912	
1022	1022	1022
1342	1342	1342
1573	1573	1573
	1614	
	1763	
1842	1842	1842
2230	2230	2230
	2531	
	2708	
	2766	

TABLE 1 Variables selected using the DP-FS-CR and NM-FS-CR feature selection routines. NM-FS-CR identified all but 2 variables identified using DP-FS-CR in addition to 16 variables that were not identified using DP-FS-CR.

and a greater number of classes within a reasonable time-frame. As with any hybrid feature selection method, this extensive cross-validation is generally considered to improve the robustness and predictive accuracy of the model.

Although employed here as a feature selection routine, CR is a generally useful metric for model quality that can be used in conjunction with validation and residual analysis in any linear space to describe the expected utility of classification models. It is the authors' hope that the described mathematical formalization and freely-available code will enable its use in a variety of different fields where multivariate classification problems are encountered. Further studies are necessary to derive cost functions for the resolution of N-Dimensional confidence ellipses, and validate the applicability of this method for determining CR in higher dimensional PCA space.

acknowledgements

The authors wish to acknowledge the support of the Natural Sciences and Engineering Research Council (NSERC) Canada and the support given to The Metabolomics Innovation Centre (TMIC) through grants from Genome Alberta, Genome Canada, and The Canada Foundation for Innovation.

An online tutorial for constructing confidence ellipses was instrumental in making this publication possible [31]. Automated colour palette generation made use of $linspecer.m$[32]. Figure 4 also utilised code for visualising error bars as as a translucent background [33].

references

[1] Sinkov, N. A.; Johnston, B. M.; Sandercock, P. M. L.; Harynuk, J. J. Automated optimization and construction of chemometric models based on highly variable raw chromatographic data. *Analytica Chimica Acta* **2011**, *697*, 8–15.

[2] Sinkov, N. A.; Harynuk, J. J. Cluster resolution: A metric for automated, objective and optimized feature selection in chemometric modeling. *Talanta* **2011**, *83*, 1079–1087.

[3] Sinkov, N. A.; Harynuk, J. J. Three-dimensional cluster resolution for guiding automatic chemometric model optimization. *Talanta* **2013**, *103*, 252–259, Cited By :3.

[4] Adutwum, L. A. Data Reduction and Feature Selection for Chemometric Analysis. Ph.D. thesis, University of Alberta, 2017.

[5] Mehmood, T.; Liland, K. H.; Snipen, L.; Sæbø, S. A review of variable selection methods in partial least squares regression. *Chemometrics and Intelligent Laboratory Systems* **2012**, *118*, 62–69.

[6] Mehmood, T.; Sæbø, S.; Liland, K. H. Comparison of variable selection methods in partial least squares regression. *Journal of Chemometrics* **2020**,

[7] Mohler, R. E.; Dombek, K. M.; Hoggard, J. C.; Pierce, K. M.; Young, E. T.; Synovec, R. E. Comprehensive analysis of yeast metabolite GC× GC–TOFMS data: combining discovery-mode and deconvolution chemometric software. *Analyst* **2007**, *132*, 756–767.

[8] de la Mata, A. P.; McQueen, R. H.; Nam, S. L.; Harynuk, J. J. Comprehensive two-dimensional gas chromatographic profiling and chemometric interpretation of the volatile profiles of sweat in knit fabrics. *Analytical and bioanalytical chemistry* **2017**, *409*, 1905–1913.

[9] Pierce, K. M.; Hope, J. L.; Johnson, K. J.; Wright, B. W.; Synovec, R. E. Classification of gasoline data obtained by gas chromatography using a piecewise alignment algorithm combined with feature selection and principal component analysis. *Journal of Chromatography A* **2005**, *1096*, 101–110.

[10] Farrés, M.; Platikanov, S.; Tsakovski, S.; Tauler, R. Comparison of the variable importance in projection (VIP) and of the selectivity ratio (SR) methods for variable selection and interpretation. *Journal of Chemometrics* **2015**, *29*, 528–536.

[11] Rinnan, A.; Andersson, M.; Ridder, C.; Engelsen, S. B. Recursive weighted partial least squares (rPLS): an efficient variable selection method using PLS. *Journal of Chemometrics* **2014**, *28*, 439–447.

[12] Sereshti, H.; Ataolahi, S.; Aliakbarzadeh, G.; Zarre, S.; Poursorkh, Z. Evaluation of storage time effect on saffron chemical profile using gas chromatography and spectrophotometry techniques coupled with chemometrics. *Journal of food science and technology* **2018**, *55*, 1350–1359.

[13] Correa, E.; Goodacre, R. A genetic algorithm-Bayesian network approach for the analysis of metabolomics and spectroscopic data: application to the rapid identification of Bacillus spores and classification of Bacillus species. *BMC bioinformatics* **2011**, *12*, 33.

[14] Speiser, J. L.; Miller, M. E.; Tooze, J.; Ip, E. A comparison of random forest variable selection methods for classification prediction modeling. *Expert Systems with Applications* **2019**, *134*, 93–101.

[15] Liland, K. H.; Indahl, U. G. Powered partial least squares discriminant analysis. *Journal of Chemometrics: A Journal of the Chemometrics Society* **2009**, *23*, 7–18.

[16] Maldonado, S.; López, J. Dealing with high-dimensional class-imbalanced datasets: Embedded feature selection for SVM classification. *Applied Soft Computing* **2018**, *67*, 94–105.

[17] Luts, J.; Ojeda, F.; Van de Plas, R.; De Moor, B.; Van Huffel, S.; Suykens, J. A. A tutorial on support vector machine-based methods for classification problems in chemometrics. *Analytica Chimica Acta* **2010**, *665*, 129–145.

[18] Wongravee, K.; Heinrich, N.; Holmboe, M.; Schaefer, M. L.; Reed, R. R.; Trevejo, J.; Brereton, R. G. Variable selection using iterative reformulation of training set models for discrimination of samples: application to gas chromatography/mass spectrometry of mouse urinary metabolites. *Analytical chemistry* **2009**, *81*, 5204–5217.

[19] Cadenas, J. M.; Garrido, M. C.; MartíNez, R. Feature subset selection filter–wrapper based on low quality data. *Expert systems with applications* **2013**, *40*, 6241–6252.

[20] Lukasiak, B. M.; Zomer, S.; Brereton, R. G.; Faria, R.; Duncan, J. C. Pattern recognition and feature selection for the discrimination between grades of commercial plastics. *Chemometrics and intelligent laboratory systems* **2007**, *87*, 18–25.

[21] Adutwum, L.; de la Mata, A.; Bean, H.; Hill, J.; Harynuk, J. Estimation of start and stop numbers for cluster resolution feature selection algorithm: an empirical approach using null distribution analysis of Fisher ratios. *Analytical and Bioanalytical Chemistry* **2017**, *409*, 6699–6708.

[22] Box, G. E. Non-normality and tests on variances. *Biometrika* **1953**, *40*, 318–335.

[23] Rajalahti, T.; Arneberg, R.; Berven, F. S.; Myhr, K.-M.; Ulvik, R. J.; Kvalheim, O. M. Biomarker discovery in mass spectral profiles by means of selectivity ratio plot. *Chemometrics and Intelligent Laboratory Systems* **2009**, *95*, 35–48.

[24] Huber, G. Gamma function derivation of n-sphere volumes. *The American Mathematical Monthly* **1982**, *89*, 301–302.

[25] Takagi, A.; Fujimura, E.; Suehiro, S. *Vestibular and visual control on posture and locomotor equilibrium*; Karger Publishers, 1985; pp 74–79.

[26] MathWorks, Chi-square cumulative distribution function. 2018; `https://www.mathworks.com/help/stats/chi2cdf.html`.

[27] Lagarias, J. C.; Reeds, J. A.; Wright, M. H.; Wright, P. E. Convergence Properties of the Nelder–Mead Simplex Method in Low Dimensions. *SIAM Journal on Optimization* **1998**, *9*, 112–147.

[28] McQueen, R. H.; Harynuk, J. J.; Wismer, W. V.; Keelan, M.; Xu, Y.; de la Mata, A. P. Axillary odour build-up in knit fabrics following multiple use cycles. *International Journal of Clothing Science and Technology* **2014**,

[29] de la Mata, A. P.; McQueen, R. H.; Nam, S. L.; Harynuk, J. J. Comprehensive two-dimensional gas chromatographic profiling and chemometric interpretation of the volatile profiles of sweat in knit fabrics. *Analytical and bioanalytical chemistry* **2017**, *409*, 1905–1913.

[30] Pérez, N. F.; Ferré, J.; Boqué, R. Calculation of the reliability of classification in discriminant partial least-squares binary classification. *Chemometrics and Intelligent Laboratory Systems* **2009**, *95*, 122–128.

[31] Spruyt, V. How to draw a covariance error ellipse. 2014; `https://www.visiondummy.com/2014/04/draw-error-ellipse-representing-covariance-matrix/`.

[32] Lansey, J. C. Beautiful and distinguishable line colors + colormap. 2015; `https://www.mathworks.com/matlabcentral/fileexchange/42673-beautiful-and-distinguishable-line-colors-colormap`.

[33] Martínez-Cagigal, V. Shaded area error bar plot. 2015; `https://www.mathworks.com/matlabcentral/fileexchange/58262-shaded-area-error-bar-plot`.

[34] Tomick, J. J. On convergence of the nelder-mead simplex algorithm for unconstrained stochastic optimization. Ph.D. thesis, 1995.

[35] Norgaard, L.; Saudland, A.; Wagner, J.; Nielsen, J. P.; Munck, L.; Engelsen, S. B. Interval Partial Least-Squares Regression (iPLS): A Comparative Chemometric Study with an Example from Near-Infrared Spectroscopy. *Appl. Spectrosc.* **2000**, *54*, 413–419.

[36] Xu, W.; Zhang, L.; Huang, Y.; Yang, Q.; Xiao, H.; Zhang, D. Fatty acid metabolic profiles and biomarker discovery for type 2 diabetes mellitus using graphical index of separation combined with principal component analysis and partial least squares-discriminant analysis. *Chemometrics and Intelligent Laboratory Systems* **2012**, *118*, 173–179.

[37] Rezzi, S.; Axelson, D. E.; Héberger, K.; Reniero, F.; Mariani, C.; Guillou, C. Classification of olive oils using high throughput flow 1H NMR fingerprinting with principal component analysis, linear discriminant analysis and probabilistic neural networks. *Analytica Chimica Acta* **2005**, *552*, 13–24.

[38] Elliott, G. N.; Worgan, H.; Broadhurst, D.; Draper, J.; Scullion, J. Soil differentiation using fingerprint Fourier transform infrared spectroscopy, chemometrics and genetic algorithm-based feature selection. *Soil Biology and Biochemistry* **2007**, *39*, 2888–2896.

[39] Pierce, K. M.; Hope, J. L.; Johnson, K. J.; Wright, B. W.; Synovec, R. E. Classification of gasoline data obtained by gas chromatography using a piecewise alignment algorithm combined with feature selection and principal component analysis. *Journal of Chromatography A* **2005**, *1096*, 101–110.

[40] Liu, X.-Y.; Liang, Y.; Wang, S.; Yang, Z.-Y.; Ye, H.-S. A hybrid genetic algorithm with wrapper-embedded approaches for feature selection. *IEEE Access* **2018**, *6*, 22863–22874.