Machine Learning for Fish Oil Anaylsis

No Author Given

No Institute Given

Abstract. In fish processing worldwide, 60% of fish is not served. This fish wastage can be repurposed into other products such as fish oils and fish feed. It is difficult and requires domain expertise to effectively repurpose fish biomass. Gas chromatography (GC) can be used to identify tissue samples in fish processing. The existing analytical chemistry techniques for processing GC data are manual and time-consuming. To reduce biomass waste we must incentivise biomass re-use, we reduce this barrier of entry through automation. Here, we explore classification algorithms for fish oil data that automate and significantly reduce the time required to process GC data. Visualisation is used to explore the interpretability of the models such that their efficacy can be verified for use in a factory setting. The fish oil data is high-dimensional and low sample size.

Keywords: feature selection · classification · gas chromatography · support vector machine · particle swarm optimization · high-dimensional data

1 Introduction

Omega-3 supplements are a high-value product made by repurposing fish waste in food processing. The waste is further refined into fish oil capsules commonly bought and sold at pharmacies. Certain fish species and parts are richer in omega-3 fatty acids, this makes these samples high value. We can identify valuable fish oils by classifying their species and part.

Gas chromatography (GC) is a chemistry technique used to analyze fish oils [3]. It analyzes the chemical structure of a fish oil sample [17]. The technique is used for quality assurance in food processing. To repurpose fish waste effectively we must know what is in it. Chemists manually compare GC samples to reference data to determine their class.

Classification automates this process by training a model on an existing dataset manually labelled by chemists. Given a fish oil sample, we can identify the fish species (i.e. Bluecod, Tarakihi), and part (Head, Fin). Many classifiers [4,5,6,16,1] can used to indentify the class of an instance from its features. These models can be organized into five categories: instance-based, probabilistic, tree-based, bagging and kernel-based, respectively. In this paper, we evaluate the performance of each classifier on both datasets, then explore the representation for the best performing classifier [7] with visualization.

Many feature selection methods [15,2,10,7] are available to eliminate redundant features. These methods can be grouped into four categories: statistics, information theory, similarity, and swarm intelligence, respectively. In this paper, the SVM classifier [1] is evaluated on the feature subsets selected by each FS method, on both datasets.

By only using important features for classification, it is easier to interpret what features/patterns the SVM model utilises. Interpretable models are capable of troubleshooting and diagnosis by domain expertise in real-world applications. A model that is both interpretable and accurate has the potential to be deployed in a factory setting, this eliminates the need for manual work.

2 Background

2.1 Gas Chromatorgraphy

Gas chromatography (GC) is a technique for the analysis of chemical compounds [3,17]. The process separates compounds based on their boiling point and molecular weight. A compound is injected as a liquid, then heat is applied to vaporize it into a gas. This process is referred to as a phase transition. The speed at which a compound is vaporized depends on its boiling point. The vaporized gases travel through a long coiled tube. That tube has a detector at the end, this detects the rate and intensity at which compounds reach the tube's end.

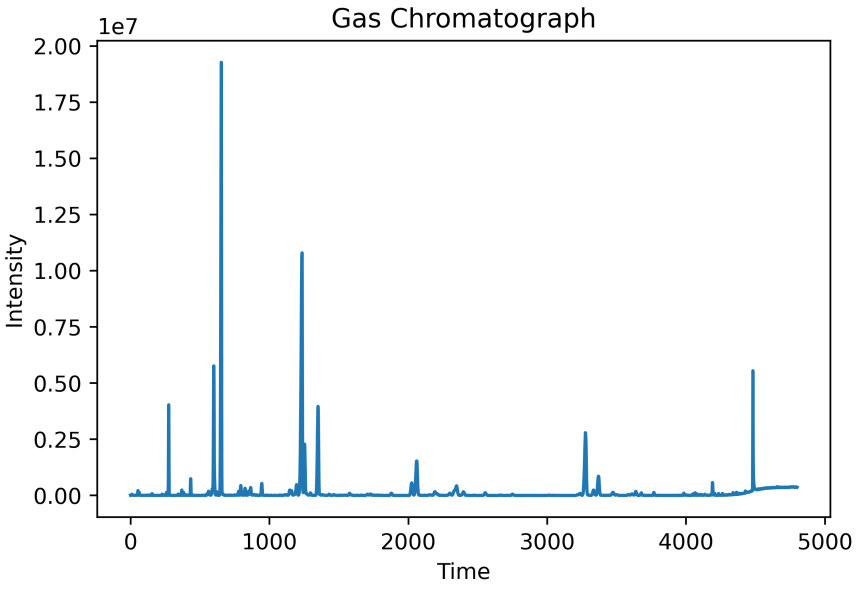
Figure 1 shows a gas chromatograph - the artefact of the GC method - from the dataset. This tissue sample was taken *part* of skin, taken from the Snapper *species*. This chromatograph is the first instance of the fish oil dataset. The detection is used to visualize intensity (y-axis) and time (x-axis) on a chromatograph. The distinguishable peaks are unique signatures where distinct chemical compounds can likely be identified. Chemists match the peaks from known reference samples to classify unknown samples. Analysis can identify an unlabelled sample since they share similar peaks to previously labelled data. There is noise in the peaks caused by time-shift - a known limitation of the measurement technique addressed in previous works [20,22].

The existing task [3,17] of identifying compounds through chromatography is laborious. Chemists integrate the area under each peak, and compare this to a reference sample, to classify the sample. GC must be performed slowly to ensure that the peaks are not too broad. This ensures each peak resolves and represents a single compound. For this fish oil data, we classify a sample by two class labels - species and part. With intuition, fish from the same species share similar chemical compositions. Machine learning techniques learn to recognize these patterns automatically, speeding up the process significantly. It can match a chromatograph to a target class, eliminating manual analysis.

2.2 Feature Selection

There are two datasets with the same features. Those features are the high dimensional GC data, the class labels are the species and part. The curse of

Fig. 1: Gas Chromatograph from a Snapper's Skin



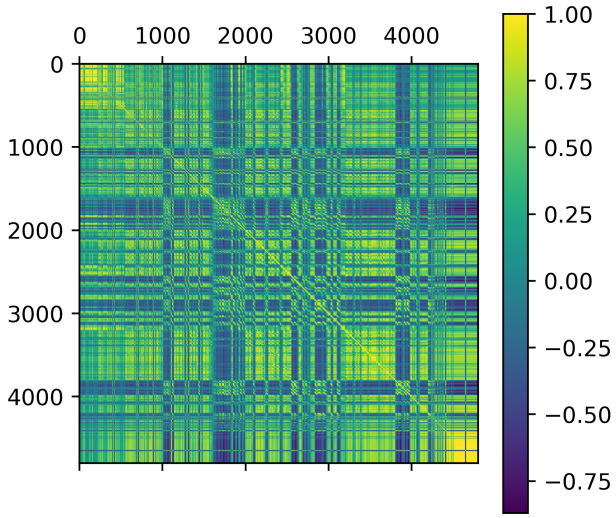
dimensionality [11] introduces problems: (1) computationally inefficient to evaluate the entire 4800 feature problem space. (2) Classification models are likely to overfit, learning noise in the data from irrelevant features. (3) Classifier models that utilize high-dimensional feature spaces are difficult to interpret, and it is difficult for domain experts to perform diagnoses or troubleshoot.

Figure 1 shows at most several points of interest (peaks), and many negligible features, for an average chromatograph from the dataset.

Figure 2 gives the fish oil data pairwise correlation for the feature set using Pearson's correlation coefficient. These figures illustrate the redundancy within the dataset, showing many irrelevant features. Feature selection reduces the size of the feature space by eliminating redundant and correlated attributes. To address these issues of high-dimensionality, we employ feature selection to select a subset of relevant features. Feature selection can (1) improve computational efficiency with a reduced dataset, (2) increase classification performance, and (3) lets classifiers produce simpler models which are in turn easier to troubleshoot. We employ heuristic-based algorithms to address the combinatorial explosion of searching the possible feature subsets. The algorithms used in this paper are introduced in further detail.

Liu et al. proposed Chi2 [15], a feature selection method via discretization. The algorithm is a generalized version of ChiMerge [8] that determines a good χ^2 threshold automatically from the data.

Fig. 2: Feature Correlation Matrix



$$\chi^2 = \sum_{i=1}^{2} \sum_{j=1}^{k} \frac{(A_{ij} - E_{ij})^2}{E_{ij}} \tag{1}$$

The formula for calculating the χ^2 statistic is given by eq. (1). The technique performs both feature selection and discretization, making it ideal for continuous numeric fish oil data. The method can increase predictive performance, and efficiency (time/memory) and simplify models.

Ding and Peng proposed Maximum-Redundancy Maximum-Relevance (MRMR) [2] a feature selection method for gene microarray data. MRMR is a filter-based multi-objective method,

$$I(X;Y) = D_{KL}(P_{(X,Y)}||P_X \otimes P_X) \tag{2}$$

These are both measured in terms of mutual information, given by eq. (2), Let (X, Y) be a pair of random variables, take the KL divergence [13] between their join distribution $P_{(X,Y)}$ and the product of their maginal distribution $P_X \otimes P_X$.

$$f^{mRMR} = I(Y;X_i) - \frac{1}{|S|} \sum_{X_s \in S} I(X_s;X_i) \tag{3}$$

The MRMR feature importance score is given in eq. (3), for full derivation see [15,23]. It balances both, the relevance for predicting outcome $I(Y;X_i)$, and the

redundancy within features $I(X_s; X_i)$, scaled by the size of the feature subset $|S|$. It builds a set of features, adding the X_i with the maximum feature importance to the selected subset.

Robnik proposed Relief-F [10], a feature selection method based on k nearest neighbours. The algorithm extends Relief [9], the extension is noise-tolerant, handles incomplete data and multi-class problems. Relief estimates feature importance based on their ability to separate other nearby instances. With intuition, a good feature can distinguish between classes. Relief-F is noise tolerant by averaging the contribution of the k nearest neighbours. Missing values of attributes are treated probabilistically[18]. Relief-F handles multi-class problems by taking the weighted average of near misses to all classes.

$$W[A] := W[A] - \frac{1}{m}\left(diff(A,R,H) + \sum_{C \neq class(R)} P(C) \times diff(A,R,M(C))\right) \tag{4}$$

The attributes are estimated with the weight update function given by eq. (4).

Kennedy and Eberhart proposed PSO [7] swarm intelligence method for nonlinear optimization. This imitates social behaviour, e.g. birds flocking or fish schooling. Their synchronous movement was modelled as a function of each fish maintaining equal distance to its neighbours. To give intuition, we imagine a simple model of a schooling fish in search of food. With p_{best} an individual fish's best spot, and g_{best} the best spot the school has found. Each fish updates its movement in equal parts towards p_{best} and g_{best}, both multiplied by a stochasticity factor. The stochasticity introduces randomness, which makes the fish overshoot their target about half the time. The overshooting has fundamental to the success of the model, it allows the school to explore uncharted waters (unknown regions of the problem domain). The social model was simplified into a particle swarm, which allowed collisions where two particles share the same location.

The fitness function used to determine p_{best} and g_best, made suitable for feature selection, is given by

$$fitness = C_1 * (1 - accuracy) + C_2 * k_{selected}/k_{total} \tag{5}$$

Where $(1 - accuracy)$ is the error, $k_{selected}/k_{total}$ is the selection ratio. $k_{selected}$ is the number of features chosen, and $k_{total} = 4800$ the size of the full feature set. C_1 and C_2 are constants that scale the error and selection ratio, respectively.

3 Data processing

- Why the raw data is not applicable to existing classification algorithms?
- Extracting datasets that are ready for classification algorithms:
  - Sum up the intensity.
  - Aligning missing packets.
- Overview of extracted data.

4 Classification

In this section, we evaluate the performance of classifiers on the fish species and part dataset. Classification identifies the class label of an instance from its features. An effective classifier has these desirable attributes: (1) high classification accuracy, (2) efficient computation, (3) and interpretable representation. Classification accuracy should be equal to and/or greater than that of humans. Any classifier that can perform the task in minutes rather than hours (how long it takes chemists manually) is preferred. An interpretable representation is required so troubleshooting and diagnoses can be performed by domain experts. A classifier with these desirable attributes (1,2,3) can be utilized in a factory setting.

We compare 5 classifiers [4,5,6,16,1] - k-Nearest Neighbours (KNN), Naive Bayes (NB), Random Forest (RF), Decision Trees (DT) and Support Vector Machines (SVM). The average balanced classification accuracy by each classifier over 30 independent runs. Central Limit Theorem says a sample distribution will approach a normal distribution $\mathcal{N}(\mu, \sigma)$, a sample size of 30 approximates a reliable measure of variance concerning the population. The datasets have a small sample size, so we use ten-fold cross-validation to create the training and test set. K-fold stratification and balanced accuracy measures are used for each run. The balanced accuracy measure ensures results are not biased towards the mode class. In sum, this experiment gives 2 (datasets) x 5 (classifiers methods) x 30 (runs) = 300 sets of empirical performance results.

4.1 Results and Discussion

Table 1: Classification Results

Dataset	Method	AvgTrain ± Std	AveTest ± Std
Species	KNN	83.57 ± 1.80	74.88 ± 12.54
	RF	100.00 ± 0.00	85.65 ± 10.76
	DT	100.00 ± 0.00	76.98 ± 13.12
	NB	79.54 ± 1.60	75.27 ± 4.35
	SVM	**100.00 ± 0.00**	**98.33 ± 5.00**
Part	KNN	68.95 ± 3.49	43.61 ± 13.48
	RF	100.00 ± 0.00	72.60 ± 16.15
	DT	100.00 ± 0.00	60.14 ± 14.57
	NB	65.54 ± 2.69	48.61 ± 12.19
	SVM	**100.00 ± 0.00**	**87.14 ± 8.52**

Table 1 shows the results for KNN, RF, DT, NB, and SVM. It gives balanced classification accuracy with ten-fold cross-validation for each method. Results are

given for fish species (top), and fish part (bottom) datasets. Results are averages across 30 independent runs for statistically significance. The mean and standard deviation of balanced accuracy is given using the fish species and part datasets. For each dataset, the best test accuracy is emphasized in bold.

The SVM classifier outperforms the other classifiers on the test set, with 98.33% and 87.14% for species and part, respectively. It does so for the test set for both the species and part datasets. For classifiers, KNN, DT and NB, for the species test it achieves 23.56%, 21.35%, 23.06% better, and for the part test it achieves 12.26%, 27%, 38.53% better, respectively. NB assumes conditional independence between features. Under this assumption, it cannot model complex relationships between dependent variables. The chromatograph from figure 1 shows a continuous and differentiable curve. The features at timestamps close to either side of a peak are also likely to be high. Conversely, features adjacent to a low intensity zero value are likely to also be low. The NB simply cannot model these feature interactions.

The random forest, decision tree and support vector machine have perfect training accuracy. The decision tree and random forest overfit the training data, achieving 100% training accuracy on both datasets. This accuracy is not reflected in the test data, where accuracy for both models is reduced, they cannot generalize to unseen data. By default, sklearn has no maximum depth for the decision tree. Without a depth limit, the tree creates strict rules on sparse data and perfectly fits the training data. It captures noise in the training set, that does not generalize to unseen data. Setting maximum depths or pruning redundant branches would address the overfitting.

The RF classifier can generalize well, with SVM achieving, 12.68% and 14.54$ better than RF, on the fish species and part, respectively. RF is 8.76% and 12.46% better than DT on the test set, for fish and species respectively. The ensemble mechanism makes RF more tolerant to noise than DT, it averages over many DT classifiers each operating on different subsets of the data. Also, the RF is noise-tolerant in comparison to KNN and NB.

The KNN struggles to notice the similarity in our n-dimensional where $n = 4,800$, supported by train accuracy 83.57% and 68.95% for species and part, respectively. Tho model cannot fit the training data, let alone generalize effectively to the test, with 74.88% and 43.61% for species and part respectively. KNN struggles with high-dimensional data, as the number of dimensions increases the distance between similar instances also increases.

4.2 Discussion

Classification accuracy for all models was better for the fish species than the part. This suggests tissue samples for different species may have distinct chemical compositions. Yet, different fish parts may have fewer underlying structural differences. For GC data the intra-class variation between species provides a larger signal than part variation. For example, we expect there to be more difference between a tarakihi and a bluecod, than there is a similarity between two livers from different species.

Of the techniques evaluated, the SVM outperformed the rest, providing the best test accuracy for both datasets. This model can identify fish species from gas chromatography data with near-perfect accuracy (98.33%). This model was a suitable candidate for automating the task, but can its reasoning be understood? This motivated further investigation into the representation of this technique.

4.3 Weight Analysis

Model interpretability is explored through visualisation. We aim to uncover learnt patterns that can be verified with domain knowledge. The desired algorithm should strike a balance between predictive performance and semantically meaningful features.

What constitutes semantic meaning varies from one domain to another. It is easy to build intuition for semantic meaning in computer vision and natural language processes, they correspond to recognisable images and structured text. In the domains of gas chromatography and fish processing, our meaning is derived from performance on the classification task(s) and similarity to underlying chemical compounds. We expect models that generate knowledge that can be verified with domain expertise. For example, important features will correspond to timestamps of important chemicals in the GC data.

Support Vector Machines Cortes and Vapnik proposed the Support Vector Machine (SVM) [1]. This model creates a hyperplane that can draw distinct class boundaries between classes. We call these class boundaries the support vectors. We are performing multi-class classification, so it used a one-vs-rest (OVR) approach [19]. This creates a divide between one class and the rest, then repeats for the other classes. The l1 regularization term leads to sparse models, these select fewer features. With fewer features, it becomes easier to interpret their meaning.

SVM is generally considered a black box method. This is especially so with complex kernels such as radial base function (RBF), sigmoidal or polynomial. However, an analysis of a Linear kernel - as proven effective in our previous experiments - can provide meaningful interpretations. We explore the hyperplane coefficients to find important features for a given class.

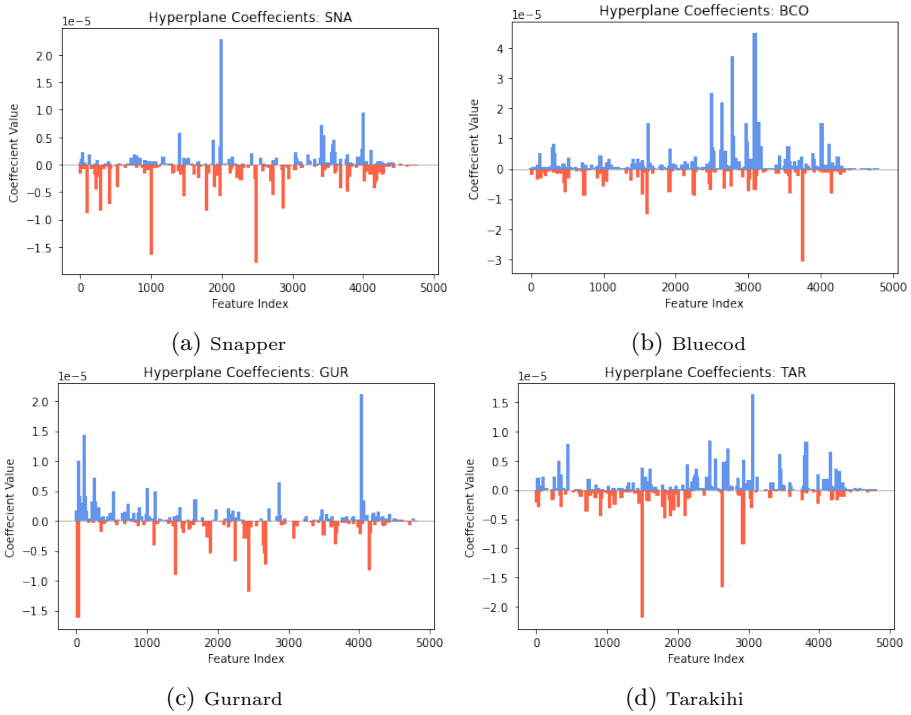
Hyperplane The coefficients β give you a vector in the n-dimensional ($n = 4800$) hyperplane. Let x be a vector for an instance in that same hyperplane. The SVM model takes the dot product $\beta.x$ between the coefficients β and instance x to determine the class of x.

$$SVM = \begin{cases} \beta.x > 0 & Snapper \\ \beta.x < 0 & Snapper \end{cases} \tag{6}$$

Equation (6) gives the piecewise function used to determine the positive class, in this example snapper. When $\beta.x$ is positive, the instance x is classified as a

snapper. Conversely, when $\beta.x$ is negative, the instance x is not a snapper, and then the algorithm proceeds to the next hyperplane. Instances in training data will not lie directly on the hyperplane, i.e. $\beta.x = 0$, as it constructs an optimal hyperplane. The GC data, shown in figure 1 is non-negative, all zero-valued or higher. This simplifies the dot product $\beta.x$ greatly, as only negative weights in β can produce negative values. The $\beta.x$ is the sum of element-wise product of β and x. Positive weights push the sum towards the positive class a snapper. Negative weights push the sum towards the negative class, not a snapper.

Fig. 3: SVM Hyperplane Coeffecients



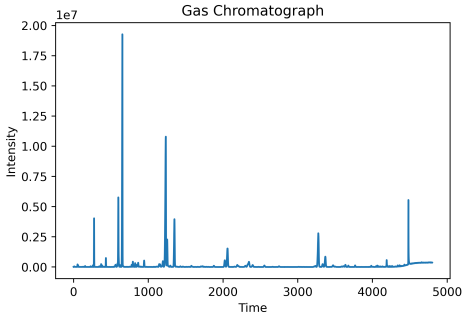
(a) Snapper (b) Bluecod

(c) Gurnard (d) Tarakihi

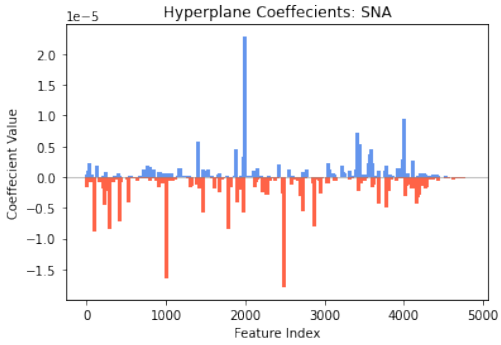
Individual Hyperplane Figure 3 shows the hyperplane coefficients for each class in the fish species dataset. The hyperplane coefficients are the weights assigned to each feature when drawing a support vector. We use colours to denote the sign of weight, a positive weight (blue), and a negative weight (red). Note, that the SVM uses OVR for multi-class problems, for classification each hyperplane is evaluated in sequential order.

In figure 4 we overlay a the snapper chromatograph x (top) - previously from figure 1- with Snapper hyperplane coefficients β (bottom). From a cursory glance,

Fig. 4: Hyperplane coefficients β_t.



(a) Species



(b) Part

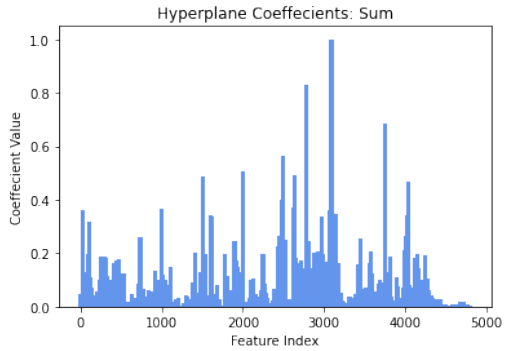
low valued intensities x correspond to the negative weights β. Conversely, if they were peaks instead in x would push the $\beta.x$ negative, away from the snapper class. In general, the higher value intensities x including peaks, correspond to positive weights β. Peaks in x for positive weights β, push $\beta.x$ positive, towards the positive class. Note that the largest peak, at timestamp $k = 653$, does not correspond to the largest weight β. This suggests, that while prominent for this instance, the feature at $k = 653$ on average does not have an important role in determining the snapper class. There are more large negative weights than positive, there are many important features that determine a class is not a snapper and fewer important weights that determine it is one.

Total Hyperplane Feature importance can be measured from the hyperplane coefficients. The absolute size of a coefficient relative to another gives a measure of its importance. Weights with large values have more impact on the position of the support vectors. To approximate the feature importance for an SVM, we can sum the absolute weights of the hyperplanes for each class, giving the *total hyperplane*.

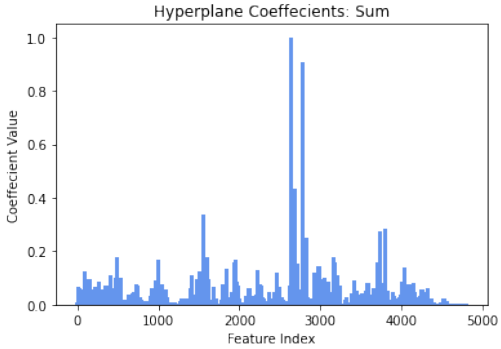
$$\beta_{\mathrm{t}} = \mathrm{minmax}(\sum_{\mathrm{c}\in\mathrm{C}} |\beta_{\mathrm{c}}|) \tag{7}$$

Equation (refeq:hyperplane) defines the total hyperplane. Where c is one class vs the rest hyperplane coefficient, and C is the set of all classes. β_{t} coefficient as the sum of hyperplane coefficients magnitude for each class β_{c}. We normalize the coefficients with a min-max feature scaling.

Fig. 5: Hyperplane coefficients β_t.



(a) Species



(b) Part

The total hyperplane for each dataset is given in Figure 5. The normalized sum of the magnitude of the coefficients for each class is given in Eq 7. The absolute value operator removes the precision of the sign, we lose information about if a weight was positive or negative. This lost information makes the total hyperplane an approximation, not the full intuition behind the support vector machine. We can no longer reason on an individual class basis, as we did before. However, this approximation is still useful for reasoning with domain

expertise. The outliers correspond to feature timestamps important for drawing class boundaries. These are chemical compounds that separate the fish part and species, respectively. We see many important features for fish species and far less for the fish part. The SVM model struggles to converge on a set of good features for the part dataset. This could explain why classification accuracy is worse for the fish part.

5 Feature Selection

To find effective feature selection methods, we evaluate the classification performance of FS methods for a range of k - where k is the number of features selected - on the fish species and part dataset. Feature selection reduces the dimensionality of the problem space. FS methods identify and keep important features, and eliminate redundant ones. With fewer features contributing to the model's decisions, it becomes easier to troubleshoot and diagnose with domain expertise. By reducing the feature space we simplify the model, and simpler models become easier to understand. Feature selection can increases performance, both in terms of efficiency and accuracy. We can see if feature selection can improve the poor classification accuracy on the fish part dataset.

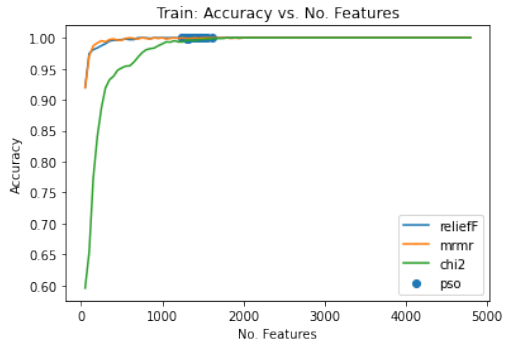
As before with classification, we use ten-fold cross-validation to generate the training and test set. For each method, we measure balanced classification accuracy with an SVM model from [19]. It has linear kernel, l1 regularization [18] and 10,000 maximum iterations. We use 3 feature selection methods [15,2,10] - Chi2, MRMR, Relief-F - with implementation from [14] set to their default parameters. We use PSO from [7] with fitness function described in equation (5). The fitness function balances the SVM classification accuracy and the number of features selected. The PSO parameter settings are: population size 30, iterations 100, $[x]^n$ where $x \in [0, 1], n = 4800$, $v_{max} = 0.2$, $v_{min} = -0.2$.

This gives classification accuracy as a function of feature number k. Where $k \in [0, 50, ..., 4800]$, and k increases by increment of 50. As before with classification, we give balanced classification accuracy with ten-fold cross-validation Due to the nature of the PSO fitness function, given by equation (5), k is selected programmatically. To allow a comparison of PSO and the other methods, we plot the results of 30 independent runs. In sum, this experiment gives 2 (datasets) x 4 (feature selection methods) x 4800 (features) / (50) increment = 768 sets of empircal performance results.

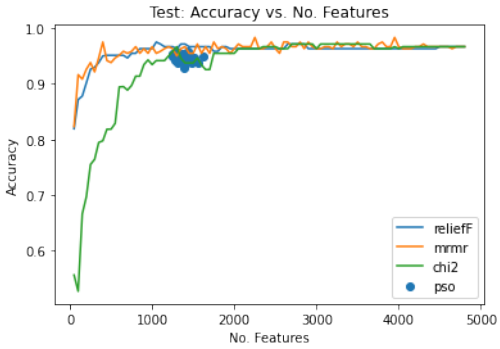
5.1 Fish Species

Figure 6 shows the results for Chi2, Relief-F, MRMR and PSO on the fish part species. We give the balanced classification accuracy for ten-fold cross-validation for each number of features selected k. Where $k \in [0, 50, ..., 4800]$, and k increases by increment of 50. In sum, this experiment gives 4 (feature selection methods) x 4800 (features) / (50) increment = 384 sets of empircal performance results.

Fig. 6: FS Classification Accuracy per k



(a) Species: Training set



(b) Species: Test set

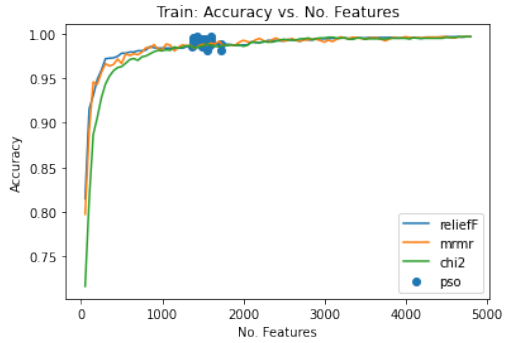
Figure 6a (top) gives the training accuracy for fish species. All methods converge on optimal training accuracy at approximately $k = 1050$. The graph follows a plateau with 100% accuracy, with diminishing returns past $k = 1000$. MRMR, Relief-F and PSO can fit the training data perfectly with approximately $k = 900$ features. Chi2 is the slowest to converge, the algorithm performs poorly for low values of k. PSO achieves consistent train accuracy, it can perfectly fit the data dataset with k features where $k \in [1100, 1800]$.

Figure 6b (top) gives the test accuracy for fish species. The test graph has more variance than the training, with noise in accuracy for all methods. The graph stabilizes at a plateau with 96% accuracy, with diminishing returns after $k = 1100$. The test performance is worse than the training by 4%, suggesting some overfitting to the training set. The near-perfect 96% test accuracy for fish species demonstrates the model can generalize well with (1100 selected / 4800 total) = 22.91% features selected. MRMR and Relief-F work best on the fish species dataset, they achieve the best test accuracy for $k \in [0, 1200]$. For MRMR and Relief-F, the model achieves 90% test accuracy using only (50 selected / 4800 total) 1% of its features. PSO achieves its best test accuracies, with the smallest

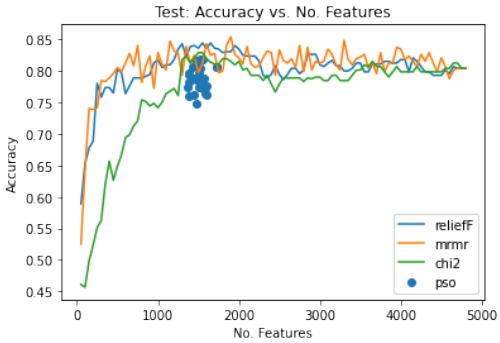
number of features selected. PSO performs worse on average than Relief-F and MRMR, although PSO found comparable results for its best individual runs.

5.2 Fish Part

Fig. 7: FS Classification Accuracy per k



(a) Part: Training set



(b) Part: Test set

Figure 6 shows the results for Chi2, Relief-F, MRMR and PSO on the fish part dataset. We give the balanced classification accuracy for ten-fold cross-validation for each number of features selected k. Where $k \in [0, 50, ..., 4800]$, and k increases by increment of 50. In sum, this experiment gives 4 (feature selection methods) x 4800 (features) / (50) increment = 384 sets of empircal performance results.

Figure 7a (top) gives the training accuracy for fish part. For MRMR, Relief-F and Chi2 the train performance follows a log curve. For these methods, there are sharp improvements between $k \in [0, 1000]$, followed by a gradual incline, this peaks at the perfect fit on the training data when $k - 4800$ (the entire feature set). PSO on average found smaller feature sets (low $k \in [1200, 1600]$)

with better train accuracy much faster than the other methods. PSO sometimes found solutions that fit the training data perfectly with $k = 1200$ features.

Figure 7b (bottom) gives the test accuracy for fish part. PSO test accuracy is the worst, it performs worse than Relief-F, MRMR and Chi2. There is potential overfitting here, future work could address improving the generalization of PSO on test data. MRMR and Relief-F work best on the fish part dataset, they achieve the best test accuracy for $k \in [0, 2000]$. MRMR, Relief-F and Chi2 all achieve their best performance at $k = 1150$, notably better than using the whole feature set. For those methods, test accuracy decreases after $k = 1150$, a contrast to train accuracy which increases, this suggests the model overfits the training set when using too many features. In comparison to fish species, the slope of the fish test contains far more variance, there is a lot of noise. There is no clear signal, no smooth improvement in classification accuracy per k the graph shows the opposite. Feature selection improves the classification performance for the part dataset.

5.3 Disucssion

The MRMR and Relief-F methods performed the best for both datasets on the training data. A normalization term, $\frac{1}{|S|}$ for MRMR and $\frac{1}{M}$, are both used to scale the data, making them tolerant to noise. Chi2 performed the worst for species, this algorithm lacks a normalization term. The $(.)^2$ operator makes this metric especially sensitive to outliers, outliers can often be attributed to noise. The use of normalization terms or lack thereof could explain why MRMR and Relief-F were robust in comparison to Chi2. The normalization term in the feature importance measure makes methods noise-tolerant.

PSO performed worst for the fish part dataset. Figure 7b shows a lot of noise, this would weaken the signal of the accuracy per k features. The accuracy per k features given in Figure 7b, can be seen as an approximation of the fitness function the PSO optimizes given in equation (5). The PSO noise in the part dataset increases the complexity of the fitness landscape where PSO searches for global optima. The accuracy per k features in figure 7a has less noise. For PSO, the fish part has a simpler fitness landscape than the fish species. This explains why PSO can generalize to the test set better on fish species than the part.

The PSO may not have a hyperparameter for feature number k. Instead, it automates the selection of this parameter. Yet, it achieves comparable results to other state-of-the-art methods. This automation may prove useful for automating the classification task for online learning. In a factory, we may want to train a model as new data arrives. PSO requires less human intervention, yet still, for fish species provides competitive performance.

6 Conclusions and Future Work

This paper has demonstrated an interpretable and effective method for fish oil analysis. The method can be understood, domain experts can understand the

important features in the decision-making. Not only have we found effective classification and feature selection techniques, but we have also tried to explain their performance with visualisation and analytical results. We can draw many conclusions from the analytical results and visualisations, but here we recall the most important:

1. Fish species are easier to predict than fish parts - there is more intra-class variation within fish species than there is a similarity between the same part from different fish.
2. The Linear SVM classifier performs better for both classification tasks - the fish oil data is linearly separable on a hyperplane.
3. Weight analysis of the SVM hyperplane serves as a useful approximation to gain insight into the model. This analysis is possible due to the nature of GC data, and its dot product with hyperplane coefficients of the Linear kernel. It is useful for troubleshooting/diagnosis using domain expertise.
4. This model can achieve 90% fish species accuracy using only 1% (k = 50) of its features, enabling efficient training, and showing many redundant and correlated features.
5. Feature selection improves the classification accuracy on the test set for the fish part - removing irrelevant features makes the task simpler.

LinearSVC applied to GC fish oil data can help to reduce waste in fish processing. Feature selection can be used to improve its performance, in terms of accuracy, efficiency and interpretability. Fewer features result in simpler models, which can be interpreted by domain experts to gain insight. This ensures more sustainable eco-friendly practices in fish processing. Waste reduction and repurposing is a rising tide that lifts all boats. Sustainable practices will leave plenty of fish in the sea, preserving resources for future generations.

It is worth noting that the classification and feature selection methods presented in this paper could be extended to potentially improve performance. This is particularly useful for the lower-accuracy fish part dataset. Here we give a non-exhaustive list of possible extensions: (1) Further investigate feature selection by evaluating the FQC MRMR-variant proposed in [23]. (2) Extend classifier to S3VM [21], a semi-supervised SVM that can utilize unlabelled GC data. This uses unlabelled data to ensure the decision boundaries are drawn through low-density areas. (3) NIST has published a Gas Chromatographic Retention dataset [12]. NIST could be applied to S3VM described, as well as pre-training/transfer learning approaches.

References

1. Cortes, C., Vapnik, V.: Support-vector networks. Machine learning **20**(3), 273–297 (1995)
2. Ding, C., Peng, H.: Minimum redundancy feature selection from microarray gene expression data. Journal of bioinformatics and computational biology **3**(02), 185–205 (2005)

3. Eder, K.: Gas chromatographic analysis of fatty acid methyl esters. Journal of Chromatography B: Biomedical Sciences and Applications **671**(1-2), 113–131 (1995)
4. Fix, E., Hodges, J.L.: Discriminatory analysis. nonparametric discrimination: Consistency properties. International Statistical Review/Revue Internationale de Statistique **57**(3), 238–247 (1989)
5. Hand, D.J., Yu, K.: Idiot's bayes—not so stupid after all? International statistical review **69**(3), 385–398 (2001)
6. Ho, T.K.: Random decision forests. In: Proceedings of 3rd international conference on document analysis and recognition. vol. 1, pp. 278–282. IEEE (1995)
7. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of ICNN'95-international conference on neural networks. vol. 4, pp. 1942–1948. IEEE (1995)
8. Kerber, R.: Chimerge: Discretization of numeric attributes. In: Proceedings of the tenth national conference on Artificial intelligence. pp. 123–128 (1992)
9. Kira, K., Rendell, L.A.: A practical approach to feature selection. In: Machine learning proceedings 1992, pp. 249–256. Elsevier (1992)
10. Kononenko, I.: Estimating attributes: Analysis and extensions of relief. In: European conference on machine learning. pp. 171–182. Springer (1994)
11. Köppen, M.: The curse of dimensionality. In: 5th Online World Conference on Soft Computing in Industrial Applications (WSC5). vol. 1, pp. 4–8 (2000)
12. Kovats, E.: Gas chromatographic characterization of organic compounds. i. retention indexes of aliphatic halides, alcohols, aldehydes, and ketones. Helv. Chim. Acta **41**, 1915 (1958)
13. Kullback, S., Leibler, R.A.: On information and sufficiency. The annals of mathematical statistics **22**(1), 79–86 (1951)
14. Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R.P., Tang, J., Liu, H.: Feature selection: A data perspective. ACM Computing Surveys (CSUR) **50**(6), 94 (2018)
15. Liu, H., Setiono, R.: Chi2: Feature selection and discretization of numeric attributes. In: Proceedings of 7th IEEE International Conference on Tools with Artificial Intelligence. pp. 388–391. IEEE (1995)
16. Loh, W.Y.: Classification and regression trees. Wiley interdisciplinary reviews: data mining and knowledge discovery **1**(1), 14–23 (2011)
17. Restek: High-resolution gc analyses of fatty acid methyl esters (fames)
18. Robnik-Šikonja, M., Kononenko, I.: Theoretical and empirical analysis of relieff and rrelieff. Machine learning **53**(1), 23–69 (2003)
19. Sklearn: 1.13. feature selection
20. Tomasi, G., Van Den Berg, F., Andersson, C.: Correlation optimized warping and dynamic time warping as preprocessing methods for chromatographic data. Journal of Chemometrics: A Journal of the Chemometrics Society **18**(5), 231–241 (2004)
21. Zemmal, N., Azizi, N., Dey, N., Sellami, M.: Adaptative s3vm semi supervised learning with features cooperation for breast cancer classification. Journal of Medical Imaging and Health Informatics **6**(4), 957–967 (2016)
22. Zhang, D., Huang, X., Regnier, F.E., Zhang, M.: Two-dimensional correlation optimized warping algorithm for aligning gc× gc- ms data. Analytical Chemistry **80**(8), 2664–2671 (2008)
23. Zhao, Z., Anand, R., Wang, M.: Maximum relevance and minimum redundancy feature selection methods for a marketing machine learning platform. In: 2019 IEEE international conference on data science and advanced analytics (DSAA). pp. 442–452. IEEE (2019)