Technical Report

Selection of Relevant Features for Raman Spectroscopy

Michael Kemmler, Joachim Denzler

Chair for Computer Vision
Department of Mathematics and Computer Science
Friedrich Schiller University of Jena
Germany

Abstract

One challenging area of research in analytical chemistry is concerned with the automatic identification of microorganisms. Recent empirical studies show that Raman spectroscopy is suited for this task and enables an accurate categorization of very similar genera, species and strains, even on the single cell level. This work focuses on the problem of wavenumber selection for Raman spectroscopy using supervised classification techniques. In addition to well-known supervised criteria employing state-of-the-art classifiers such as Boosting, Regularized Logistic Regression and Projection to Latent Structures Discriminant Analysis, new relevance measures are derived from Random Decision Forests. We also investigate Automatic Relevance Determination (ARD) in Gaussian process classifiers, using a Bagging paradigm, which has not been used before in Raman spectroscopic feature selection. In experiments, we analyze a total number of 15 different relevance criteria which are applied on a large-scale database comprising 10 different species. In order to yield an unbiased performance estimate of each strategy, an additional independent dataset (comprising 7 species and previously unseen latent strains) is analyzed using four different classification techniques for performance assessment. Compared to unreduced Raman spectra, the majority of proposed feature selection strategies leads to an increase in recognition rate (on species level) which suggest that the proposed measures are suitable for feature selection. The highest accuracy (97.8% overall recognition rate) is achieved using a relevance criterion derived from ARD which highlights the potential of non-parametric Bayesian methods for Raman spectroscopy.

Keywords: Raman spectroscopy, supervised feature selection, Gaussian processes, Automatic Relevance Determination, Random Forests, bacteria recognition

1. Introduction

Bacteria, fungi, and other kinds of microorganisms inhabit nearly any place on earth. Due to this fact, the identification of such microbes is often desirable. Especially in crucial fields such as medical applications [2], food sciences [3] or clean-room environments [4] as in pharmaceutical industry, a reliable and fast method for categorizing particles is needed. There is a whole assortment of techniques which can be used to achieve a differentiation between certain classes of microorganisms, e.g. growth measurements under certain conditions [5] or morphology-based microscopy [6]. While the latter can be used on single-cell level, it is often not possible to distinguish between similar categories (e.g. bacterial strains). Assessing the ability to grow, on the other hand, is a time-consuming process. However, many samples require real-time analyses on very limited amounts of data which highlights the need for methods which achieve both accuracy and sensitivity. In recent years, several techniques such as mass spectroscopy, flow cytometry and fluorescence spectroscopy [7, 5] were developed which achieve this goal.

One alternative are vibrational spectroscopic techniques [8, 9, 10] such as Raman spectroscopy [11, 12]

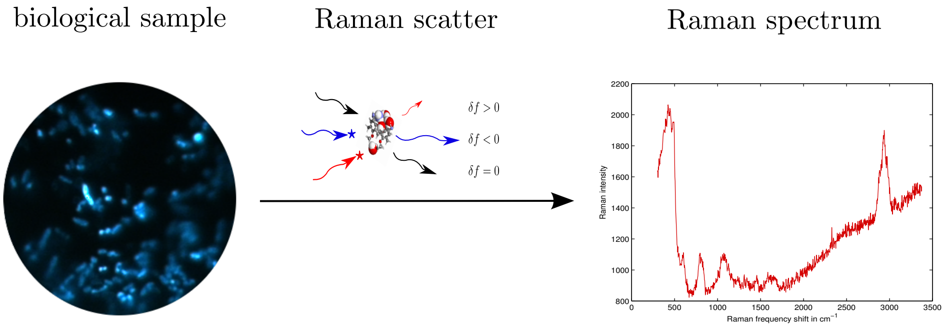

Figure 1: Simplified mechanism behind Raman spectroscopy. Light from a narrow-band laser beam irradiates the sample of interest. Shifts in frequency (wavenumber) due to molecular vibrations [1] are then recorded by means of a Raman spectrometer. The frequency changes are integrated over a specific time interval, generating a fingerprint-like signal that contains information of the molecular decomposition of the whole sample.

which are increasingly used to detect or categorize microorganisms [13, 14, 15, 16], even on the single-cell level [17, 4, 18]. This family of optical techniques aims at measuring the vibration of molecules and generate a fingerprint-like signature of the whole sample of interest (c.f. Figure 1). Based on this encoding of the chemical composition of the sample, the respective category can be learned or predicted by employing state-of-the-art classification techniques [19] such as Support Vector Machines (SVMs) [20, 4, 21] or Gaussian process (GP) classifiers [22, 23].

In many cases, the Raman spectra contain uninformative parts, i.e. regions which have no positive impact on the categorization result and can thus be neglected. Retaining only relevant parts of the spectrum can be advantageous in different ways. First of all, the reduction of features most often has a positive effect on the time complexity of the classifier's learning and prediction phase. Second, since unimportant spectral regions might contain arbitrary signal fluctuations, excluding the corresponding features might lead to a gain in recognition performance. Third, knowledge about the relevance of features also improves the interpretability of the results. In Raman spectroscopy this may enable experts to draw conclusions about the importance of chemical compounds [24] with respect to the categorization task.

The determination of regions which are relevant for a given classification task, however, would require solving a combinatorial optimization problem, since we want to find the smallest subset $\mathcal{I} \subseteq \mathcal{F}$ of features (dimensions) $\mathcal{F} = \{1, \ldots, D\}$ which maximizes a given performance measure (with respect to a given training and test dataset). To obtain the optimal subset $\mathcal{I}$, an exhaustive evaluation of 2^D feature combinations is required which cannot be computed in reasonable time for data points containing more than a few dozen features. Since, unfortunately, the majority of application-relevant tasks in Raman spectroscopy deals with spectra containing at least a few hundred dimensions, heuristics or approximations are needed in order to find important features.

One major branch of feature reduction techniques developed for vibrational spectroscopy circumvent the problem of solving a combinatorial optimization problem by concentrating on a different objective. The most popular criteria focus on extracting explanatory variables which contain high uncertainty or high correlation with their respective response variables, e.g. by projection onto principle components [19] or using related factor models such as *Projection to Latent Structures* [25]. In general, these methods result in a transformation to linear or non-linear subspaces. In spite of the fact that impressive performance gains can be obtained by using these procedures, the direct interpretability of the result gets lost. On the other hand, these criteria are usually related to regression tasks and are thus unsupervised, i.e. they do not consider the underlying categories. Prominent exceptions are Fisher's Linear Discriminant Analysis (FLDA) [19] and related methods [26] which aim at finding a linear subspace which maximizes the data variation between categories (while minimizing the variation within categories). However, the corresponding dimension reduction step for FLDA is also accomplished by projection onto a manifold which makes a direct interpretation of features difficult.

There exists also a large set of supervised algorithms which directly incorporate class-specific information into the feature extraction process. In the literature, classifiers are often used as black boxes in order to score a previously selected subset $\mathcal{I}$. This so-called *wrapper approach* does not eliminate the combinatorial problem, however, it is often encountered employing some feature selection strategy for choosing possibly interesting subsets, e.g. forward selection, backward elimination [27, 28] or genetic algorithms [29].

Other approaches directly utilize supervised learning methods which are capable of generating scores which indicate the relevance of features for the classification task. E.g. linear SVMs with l_1-norm regularization [30] or linear SVMs employing an iterative reweighting approach [31] are used for extracting informative variables. Ensemble methods for classifier combination such as Boosting [32] and Random Decision Forests (RDFs) [33] are also successfully utilized for feature selection [34] and relevance scoring [35, 36].

This work focuses on comparing different supervised techniques which are able to infer relevant features. Along with Regularized Logistic Regression [19] and the ensemble methods RDF [33] and Boosting [32], we also investigate the suitability of Automatic Relevance Determination (ARD) [37] based on GP classifiers [22], which to our knowledge has not been done before in the context of feature selection in vibrational spectroscopy. In addition to utilizing known scoring procedures for RDFs [36], we further introduce new relevance criteria based on this ensemble method.

This paper is structured as follows. Sect. 2 reviews supervised classifiers along with possibilities to extract relevant dimensions. All methods are compared based on a Raman spectroscopic task in Sect. 3. A summary and discussion to future work concludes this work.

2. Methods

The following section contains supervised classification methods which are used in this work to infer relevant features.

2.1. Ensemble classifiers

The result of a given classification task strongly depends on the power of the underlying classifier. If a given classifier C_1 is not able to achieve a predefined goal in terms of prediction accuracy, one often resort to a classifier C_2 which exhibits a higher flexibility or generalization ability. One alternative strategy is to use ensemble methods, i.e. to combine multiple classifiers of type C_1 in a certain way in order to increase the expected classification accuracy.

For the ensemble methods used in this paper, the combined classifier type C_1 was chosen to be a Decision Stump [38]. This simple, univariate classifier is often used in this setting, since it is easy to implement, allows for a fast optimization of parameters and can be directly associated with features. The family of Decision Stump classifiers can be defined as

$$h_{\theta,s,k}(\mathbf{x}) = \begin{cases} 1 : & s \cdot x_k > \theta \\ -1 : & \text{otherwise} \end{cases} \tag{1}$$

with threshold parameter θ, polarity s, and index position k. This classifier separates the vector space at threshold θ in two half-spaces parallel to the k-th coordinate axis.

This Decision Stump is commonly used as weak classifier in two ensemble methods which are discussed in the next sections: RDFs and AdaBoost.

2.2. AdaBoost

One straightforward way to combine a set of weak classifiers $h_1, \ldots, h_T$ is to use a linear model:

$$h(\mathbf{x}) = \sum_{i=1}^{T} \alpha_i h_i(\mathbf{x}) \tag{2}$$

where $\boldsymbol{\alpha}$ denotes the vector of weights associated to the respective weak classifiers (c.f. Figure 2). AdaBoost is one representative of this family of ensemble methods which aims at optimizing an exponential loss on the training data. Let $\mathcal{D} = (\mathbf{X}, \mathbf{y})$ be the training set containing inputs $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ and corresponding outputs $\mathbf{y} = \{y_1, \ldots, y_n\}$, then the exponential loss is given by

$$\varepsilon = \sum_{k=1}^{n} \exp\left(-y_k \cdot h(\mathbf{x}_k)\right) \tag{3}$$

In order to optimize this loss function, the standard (binary) AdaBoost algorithm sequentially adds one classifier per round while updating a data distribution. The adjusted distribution is then used to focus on samples which are misclassified in the previous round and thus allows to concentrate on seemingly harder data points. It has been shown that this simple updating mechanisms optimizes the objective in (3). For more detailed insights into the algorithm and mathematical properties, we refer to [32].

The standard AdaBoost algorithm can also be modified in order to allow for multiple classes. There is a variety of different approaches [32] from which we used AdaBoost.MH in our experiments. This variant constructs an internal one-vs-all problem which is straightforwardly solved using the standard AdaBoost formalism described above. Although the prediction step for the weak classifiers is accomplished independently, AdaBoost.MH jointly learns one classifier parameter set

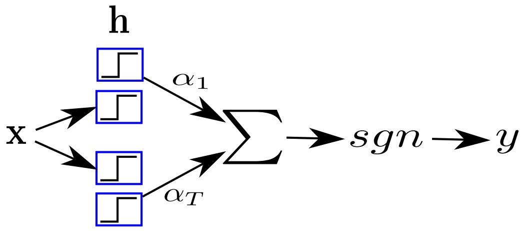

Figure 2: Schematic illustration of AdaBoost using a list of weak classifiers $\mathbf{h} = (h_1, \ldots, h_T)$.

and one weight vector $\boldsymbol{\alpha}$ for all binary subtasks. In order to be able to infer the class of a given test example $\mathbf{x}_*$, a majority vote is generally used:

$$c^* = \operatorname*{argmax}_c h^{(c)}(\mathbf{x}) \tag{4}$$

where $h^{(c)}$ denotes the classifier which separates class c from the rest.

Using a Decision Stump as base classifier, AdaBoost.MH iteratively selects one dimension per iteration (shared by all binary subtasks) along with a weight α_i, $i \leq T$. Since the selected features are ordered by their ability to minimize the (adaptive) exponential loss, one naïve feature selection strategy using Boosting (B) is to concentrate on the first D' disjunctive features. In this context, disjunctive means that we ignore weak classifiers which contain a previously recorded feature.

2.3. Random Decision Forests

Instead of using a sum of weighted classifier outputs, a sequential feed-forward technique can also be employed to build an ensemble classifier. So-called Decision Trees utilize a tree-based topology, where weak classifiers are represented as nodes and connections between classifiers are encoded as edges.

Given a test data point $\mathbf{x}_*$, the output is generated by presenting $\mathbf{x}_*$ to the root of the tree. This data point then traverses the tree on a certain path which is given by the decisions of the weak classifiers with respect to the presented data point. Since the edges are uni-directional, the data point eventually reaches a leaf. Each leaf of the tree contains a class distribution which denotes how probable it is that data points from a given class reach this leaf. This distribution then serves as output of the ensemble classifier, given the presented data point (c.f. Figure 3).

The structure of Decision Trees is learned in an iterative manner. Beginning with the root, a weak classifier is trained in order to optimize a given criterion. Generally, an impurity criterion such as the *Gini Index* or *Information Gain* is minimized to achieve a good separation between classes. In our work, we solely use the Information Gain, which is defined as the difference of information content before and after the classifier is invoked. The information content can be specified by the entropy $E(v)$ with respect to node v. Let $p_c(v)$ be the expected probability that class $1 \leq c \leq C$ occurs within node v, the entropy is defined as

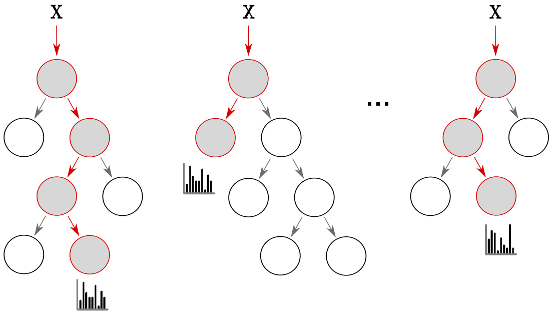

Figure 3: A set of Decision Trees (a forest). A test vector $\mathbf{x}$ is presented at all roots within the forest, traverses the trees and ends up at leaf nodes which are equipped with a class probability distribution.

$$E(v) = -\sum_{c=1}^{C} p_c(v) \log p_c(v) \tag{5}$$

By employing binary classifiers, two children nodes will result which are denoted by left node v_l and right node v_r. The *Information Gain* achieved by performing the according split at node v can then be computed as

$$IG(v) = E(v) - \omega_l E(v_l) + (1 - \omega_l) E(v_r) \tag{6}$$

where ω_l denotes the probability that a data point ends up in the left children node.

RDFs are a collection of Decision Trees, where each tree is learned on a randomly drawn subset of data points, a technique which is known as Bootstrap Aggregation [39]. Standard RDFs [33] utilizes Decision Stumps as weak classifiers which enables a second level of randomization: Instead of searching for the right index k among all features, only a random subset of features is taken into account. This randomization strategies allows for a better generalization ability and bias reduction [33]. A final output can be obtained by averaging the results of all trees of the forest.

Since standard RDF implementations work with *Decision Stumps* as base classifier, each node is directly associated with a feature. It is thus possible to associate features with node properties. This is done by Rogers et al. [35, 40], where the average information gain, augmented by a node complexity criterion, is used as feature relevance score. A similar approach is followed by Menze et al. [36] for feature selection in the context of vibrational spectroscopy, where the sum of *Gini Index* values, averaged over all nodes and trees, serves as relevance measure.

In this work, the following set of properties of the RDF classifier is exploited in order to generate feature relevance scores:

1. **Counting (RDF-C)**. If a feature is selected within a node, the evidence for relevance increases. The resulting histogram (i.e. counts) over selected dimensions can hence be interpreted as importance measure.
2. **Information Gain (RDF-IG)**. As in [36], the relevance of a selected feature is raised by a value proportional to the information gain, since features which are responsible for achieving pure children nodes should receive appropriate rewards.
3. **Inverse Depth (RDF-ID)**. Since nodes of lower tree depth have both a high impact on the structure of the tree and influence nodes in deeper stages, the depth can be incorporated into a relevance measure, e.g. the relevance of a selected feature can be raised proportional to the inverse depth.
4. **Incoming Datapoints (RDF-IDP)**. The number of data points which are effected by a node can vary drastically. Selecting a feature which correctly separates only two training points, however, should be weighted differently from features which effect a large number of data points. We therefore propose to increase the relevance measure proportional to the number of incoming data points that are associated to the node during construction.
5. **Combinations**. A combination of above measures is also possible, e.g. a multiplication of criteria Information Gain, Inverse Depth and Incoming Datapoints (RDF-IG-ID-IDP).

2.4. *Regularized Logistic Regression*

As has been mentioned in the introduction, many algorithms estimate feature relevance based on regression, where linear models are constructed in order to predict certain output values. This unsupervised approach, however, neglects the class information given by the problem at hand. A popular supervised analog is logistic regression, where a logistic model (c.f. Figure 4) is employed to estimate the probability of belonging to a certain class:

$$p(y = 1|\mathbf{x}, \omega) = \frac{1}{1 + \exp\left(-\boldsymbol{\omega}^T \mathbf{x}\right)} \tag{7}$$

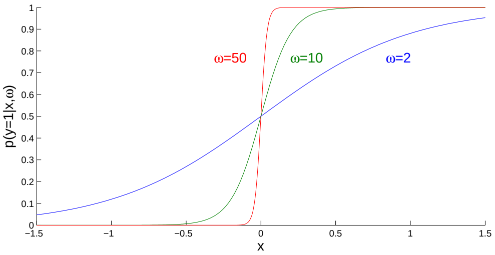

Figure 4: Logistic function on one-dimensional input space with varying parameters of $\omega \in \{2, 10, 50\}$.

The parameter $\boldsymbol{\omega}$ can be learned by means of Maximum Likelihood optimization, i.e. assuming i.i.d. data $\mathcal{D}$:

$$\boldsymbol{\omega} = \underset{\mathbf{w}}{\operatorname{argmax}} \sum_{i=1}^{n} \log p(y_i|\mathbf{x}_i, \mathbf{w}) \tag{8}$$

This approach offers the possibility to directly access the features in the input space, since a weight ω_k is associated to the k-th feature. A value of ω_k close to zero would indicate a highly irrelevant feature and large values might indicate high relevance. It is not sure, however, that weight values have a high spread ranging from zero to high values. This property can be encouraged by placing a prior on the weight vector $\boldsymbol{\omega}$. If we expect the weight vector to contain many values close to zero, we can incorporate this knowledge into the optimization routine, where we end up with a Maximum-A-Posteriori optimization, i.e. $\boldsymbol{\omega} = \operatorname{argmax}_{\mathbf{w}} \log p(\mathbf{w}|\mathbf{X}, \mathbf{y})$. Using a zero-mean Gaussian $\mathbf{w} \sim \mathcal{N}(0, \lambda^{-1}\mathbf{I})$ with inverse variance $\lambda > 0$, this simplifies to

$$\boldsymbol{\omega} \propto \underset{\mathbf{w}}{\operatorname{argmax}} \sum_{i=1}^{n} \log p(y_i|\mathbf{x}_i, \mathbf{w}) - \lambda \cdot ||\mathbf{w}||^2/2 \tag{9}$$

which is a regularized version of the logistic regression, also known as ridge logistic regression.

By tuning the inverse variance parameter λ, we can adjust the impact of the regularization, i.e. large λ lead to solutions where many values ω_k are close to zero and vice versa.

Using this Regularized Logistic Regression approach, we can score features by their associated weight vectors. One approach is to mark features as relevant which have nonzero weights or weights higher than a small threshold ϵ (RLR-C). Since larger weights might have a higher impact on the estimate, one could also use the absolute values $|w_k|$ (RLR-W) as score for the k-th feature. however, the latter approach would introduce a bias, since the dimensions might contain different scalings. To account for this fact, it is also possible to reweight this score by the average feature size, i.e. to use

$|w_k/\mu_k|$, where $\mu_k = n^{-1}\sum_{i=1}^{n} x_{ik}$ and x_{ik} denotes the k-th feature of vector $\mathbf{x}_i$ (RLR-RW).

Since we have multiple classes in our settings, we learn C different one-vs-all subtasks. The final score is then computed by summing over all scores obtained from the respective binary problems.

2.5. Gaussian Process Classifier

In the following section, the theory of GP classification is described. Since the GP classifier is motivated from Bayesian regression, we follow the usual approach of first describing the regression case before discussing necessary changes that allow for classification.

One of the simplest regression model is to assume that outputs are generated by means of a linear model

$$y(\mathbf{x}, \boldsymbol{\omega}) = \Phi(\mathbf{x})^T \boldsymbol{\omega} + \varepsilon, \tag{10}$$

where $\Phi : \mathcal{X} \rightarrow \mathcal{H}$ denotes some deterministic transformation of input arguments to some inner product space and ε specifies a non-deterministic noise term.

The standard frequentist strategy of regression proceeds by tuning the parameters $\boldsymbol{\omega}$ to minimize a loss function between the predicted function $y(\mathbf{x}_i, \boldsymbol{\omega})$ and the observed outcomes y_i. However, this approach is prone to overfitting, i.e. over-adaption to the training data.

Bayesian regression aims to avoid this shortcoming by defining confidences for parameters $\boldsymbol{\omega}$ according to some prior belief. This uncertainty is used to integrate out the nuisance parameter $\boldsymbol{\omega}$ to obtain a prediction which effectively combines infinitely many models.

2.5.1. Gaussian Process Regression

Instead of concentrating on the weights $\boldsymbol{\omega}$, we can directly focus on the latent noise-free function $f(\mathbf{x}) = \Phi(\mathbf{x})^T \boldsymbol{\omega}$. The assumption in GP regression is that the latent function $f(\cdot)$ is drawn from a GP prior. Such a prior, which can be seen as a normal distribution over functions, is solely specified by a mean function $m(\mathbf{x})$ and a positive definite covariance function $\kappa(\mathbf{x}, \mathbf{x}')$. Hence, we assume

$$f(\mathbf{X}) \sim \mathcal{N}(m(\mathbf{X}), \kappa(\mathbf{X}, \mathbf{X})) \tag{11}$$

for any finite collection of random variables $\mathbf{X}$.

The crucial step for predicting an output $y_* = y(\mathbf{x}_*)$ for a previously unseen data point $\mathbf{x}_*$ is to infer $p(f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y})$, i.e. the distribution of the latent function at point $\mathbf{x}_*$ given the training data $\mathcal{D} = (\mathbf{X},\ \mathbf{y})$. By assuming that outputs are independently generated according to (10), this can be accomplished by solving the following integrals

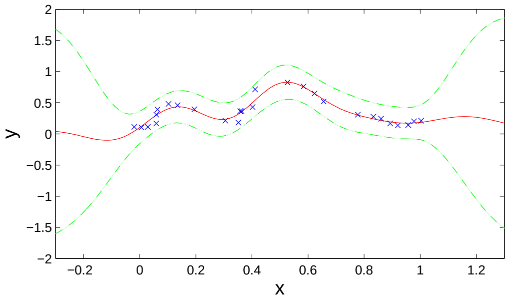

Figure 5: Example for GP regression with SE-kernel ($\boldsymbol{\theta} = (0.9195, 0.1742)^T$, $\sigma_n^2 = 0.1158$). Training points (crosses) are well fitted by the GP mean μ_* (solid line). The 95%-confidence interval $\mu_* \pm 1.96\sigma_*$ (spaced lines) illustrates the confidence of predicted values.

$$p(y_*|\mathbf{X}, \mathbf{y}, \mathbf{x}_*) = \int p(f_*|\mathbf{X}, \mathbf{y}, \mathbf{x}_*)\ p(y_*|f_*)\ df_* \tag{12}$$

$$p(f_*|\mathbf{X}, \mathbf{y}, \mathbf{x}_*) = \int p(f_*|\mathbf{X}, \mathbf{f}, \mathbf{x}_*)\ p(\mathbf{f}|\mathbf{X}, \mathbf{y})\ d\mathbf{f} \tag{13}$$

where $\mathbf{f} = f(\mathbf{X}) = (f(\mathbf{x}_1), \dots, f(\mathbf{x}_n))^T$ and $f_* = f(\mathbf{x}_*)$. This is often not analytically computable, however, by assuming a GP prior over latent functions and i.i.d. Gaussian noise, i.e. $\varepsilon \sim \mathcal{N}(0, \sigma_n^2)$, both integrals turn out to be normal distributions. Using a zero-mean GP prior (i.e. $m = 0$), it can be shown that (13) has the following moments [41]:

$$\mu_* = \mathbf{k}_*^T \left(\mathbf{K} + \sigma_n^2 \mathbf{I}\right)^{-1} \mathbf{f} \tag{14}$$

$$\sigma_*^2 = k_{**} - \mathbf{k}_*^T \left(\mathbf{K} + \sigma_n^2 \mathbf{I}\right)^{-1} \mathbf{k}_* \tag{15}$$

where μ_* and σ_*^2 denote mean and variance of $p(f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y})$, respectively, and the shorthands $\mathbf{K} = \kappa(\mathbf{X}, \mathbf{X})$, $\mathbf{k}_* = \kappa(\mathbf{X}, \mathbf{x}_*)$, and $k_{**} = \kappa(\mathbf{x}_*, \mathbf{x}_*)$ are used for the sake of readability. Moreover, the distribution (12) is also normally distributed with mean μ_* and variance $\sigma_*^2 + \sigma_n^2$ [22].

2.5.2. From Regression to Classification

The goal in GP classification is to model a function which predicts a confidence for each class $y \in \{-1, 1\}$, given a feature vector $\mathbf{x}$. Since the output space is discrete, rather than continuous, it is not appropriate to assume a Gaussian noise model. There are two common strategies to tackle this issue. We could either ignore the discrete nature of the problem and perform label regression, or choose a more appropriate likelihood function which is suitable for classification. In the following work, the latter approach is adopted where a cumulative Gaussian likelihood $p(y|f(\mathbf{x})) = \Phi(yf(\mathbf{x}))$ is utilized

$$\Phi(yf(\mathbf{x})) \quad = \quad (2\pi)^{-\frac{1}{2}} \int_{-\infty}^{yf(\mathbf{x})} \exp\left(-z^2/2\right)\, dz \quad (16)$$

along with the assumption of conditional independence for outputs $\mathbf{y}$, given their respective latent function values $\mathbf{f}$. This slightly change from Gaussian to cumulative Gaussian likelihood has, however, far reaching consequences with respect to the inference procedure. Due to the non-Gaussian likelihood, integral (13) can no longer be solved in closed form which renders the problem of estimating $p(f_*|\mathbf{X}, \mathbf{y}, \mathbf{x}_*)$ computationally intractable.

Approximate inference methods can be used to tackle this problem. In the GP framework, these methods generally estimate a Gaussian approximation to a non-Gaussian probability distribution such as $p(\mathbf{f}|\mathbf{X}, \mathbf{y})$. Two popular methods following this scheme are Laplace approximation [42] (LA) and Expectation Propagation [43] (EP). While the latter usually achieves very accurate estimations, LA has indisputable speed advantages [44]. Interested readers are referred to [22] for further insights of LA and EP and their application to the GP framework.

Using the likelihood (16) in combination with a Gaussian approximation to $p(f_*|\mathbf{X}, \mathbf{y}, \mathbf{x}_*)$, integral (12) also turns out to be a cumulative Gaussian which is computable in closed form [22].

2.5.3. Model Selection and Relevance Determination

Parameter selection is a crucial step in designing a classifier. In the Bayesian framework those parameters are given as prior and hyperpriors over latent variables. In case of GP classifiers, the appropriate covariance function (which serves as prior over latent functions) needs to be specified. One of the most commonly used covariance function is the isotropic squared exponential (SE) kernel

$$\kappa_{\text{SE}}(\mathbf{x}, \mathbf{x}') = \theta_1^2 \cdot \exp\left(-\theta_2^{-2} \cdot \|\mathbf{x} - \mathbf{x}'\|^2/2\right) \qquad (17)$$

One flexible alternative to the isotropic SE-kernel is the Automatic Relevance Determination (ARD) kernel

$$\kappa_{\text{ARD}}(\mathbf{x}, \mathbf{x}') = \nu_0^2 \cdot \exp\left(-\sum_{k=1}^{D} \nu_k^{-2} \cdot (x_i - x_i')^2/2\right) \qquad (18)$$

Hyperparameters $\boldsymbol{\theta}_{\text{SE}} = (\theta_1, \theta_2)^T$ or $\boldsymbol{\theta}_{\text{ARD}} = (\nu_0, \nu_1, \ldots, \nu_D)^T$ can be optimized via the evidence framework. This strategy aims at finding the hyperparameters $\boldsymbol{\theta}^*$ which maximizes the probability $p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})$ of generating the output given the input, i.e. $\boldsymbol{\theta}^* = \text{argmax}_{\boldsymbol{\theta}}\, p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})$ [22]. This method is often used in practice since it provides an automatic way to obtain parameters, e.g. by using gradient based optimization routines such as Conjugate Gradient Descent [22].

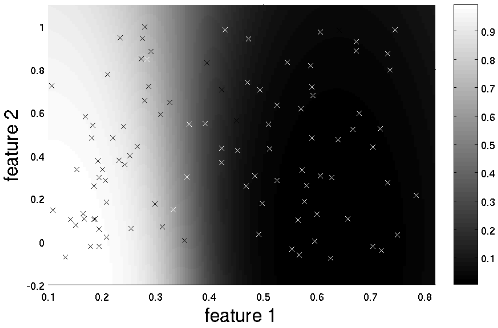

Figure 6: Automatic Relevance Determination using the GP classifier applied on a binary toy dataset with class A (black dots) and class B (white dots). The inferred hyperparameters are $\nu_1 = 0.2868$ and $\nu_2 = 1.7806$ which indicates a high relevance of the first feature. The estimated probability of belonging to class A using above parameters is visualized by different background shading.

The evidence framework, coupled with the ARD-kernel can thus serve as a mechanism to infer the importance of features. A small value of ν_k $(1 \leq k \leq D)$ would lead to an amplification of the k-th feature, whereas a large value attenuates contributions of feature k with respect to the classification result (c.f. Figure 6). Hence, the following relevance criteria can be derived from the GP classifier using ARD-kernel *after* evidence maximization is invoked:

1. **Contribution to kernel (ARD-U)**: Since large values of ν_k^{-1} $(1 \leq k \leq D)$ suggest large contribution of the k-th feature to the kernel, this measure can serve as a relevance criterion.
2. **Normalized Contribution to kernel (ARD-N)**: Unequal scalings within dimensions cannot be inferred by using the ARD kernel. A measure which tries to overcome this shortcoming reweights the inferred contribution with the standard deviation σ_k within the k-th dimension. Since the kernel measures differences, this variance-based correction is used instead of the mean correction as in RLR-RW, i.e. the relevance of feature k is proportional to $(\nu_k \sigma_k)^{-1}$

However, when many parameters are optimized by the evidence framework, which is the case for our Raman spectroscopic problem using the ARD approach, severe overfitting problems can result. In order to avoid over-adaption to the training set, Qi et al. [45] propose to monitor a leave-one-out (LOO) error for intermediate parameter estimates $\boldsymbol{\theta}'$ within the optimization procedure. Since no method for obtaining a LOO error estimate using LA is known to us, the approach of [45] is followed which uses EP, leading to a procedure closely related to the mean field approach of Winther and Opper [46].

Since the GP classifier used in our experiments is a binary classifier, we again utilized a one-vs-all scheme. Instead of optimizing the hyperparameters independently, however, we follow Kapoor et al. [47] and optimize the joint evidence, i.e.

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \sum_{\tau} \log p(\mathbf{y}^{(\tau)}|\mathbf{X}, \boldsymbol{\theta}) \qquad (19)$$

where τ indexes the binary one-vs-all tasks. In addition to a possible speed improvement, this method also achieves a smoothing of hyperparameters and hence avoids overfitting.

Although a faster inference is possible employing joint maximization of parameters, using EP on large amounts of data is still a lengthy procedure and may easily take a few weeks or months on current computers. In order to achieve a further speed-up, we rely on a Bagging approach. As for RDFs (c.f. 2.3), we construct multiple problems by resampling from the whole dataset. The overall relevance score is then computed as the sum of scores obtained from all subproblems. Instead of estimating the ARD-hyperparameters for each bag separately, a joint optimization over all bags is possible. This strategy avoids high fluctuations among hyperparameters from different bags and gets rid of the rather heuristic summation process. Since all parameters are shared between different bags, the risk of overfitting is once more reduced.

2.6. PLS-DA

Projection to Latent Structures (PLS) is a basic tool in chemometrics which enables to compress information hidden in both inputs $\mathbf{X}$ (explanatory variables) and outputs $\mathbf{Y}$ (response variables) by projecting both to a lower-dimensional latent variable space. While several types of PLS exist, most variants try to iteratively find those latent basis vectors which maximize the covariance between projected input and output vector [25]. To make use of PLS for the classification scenario, PLS Discriminant Analysis (PLS-DA) was recently proposed [26]. In a one-versus-all manner, the multi-label problem is reduced to several subproblems with binary response variables. By collecting all binary response vectors to a response matrix $\mathbf{Y}$, standard PLS regression is then utilized to infer a given number of latent basis vectors. To fix the dimension of the latent space in advance is a difficult problem since a trade-off between flexibility and generalization ability has to be made. In practice, a good estimation can be often accomplished by using k-fold cross validation (CV) on the training data, choosing the dimensionality that leads to the lowest CV error.

Once the dimensionality is fixed, PLS is invoked and the weights w_{ij} used for the projection to latent variables $\mathbf{T} = \mathbf{W}^T\mathbf{X}$ can be employed to estimate variable importance. By construction, w_{ij} describes the contribution of the i-th input to the j-th latent dimension. The sum $\sum_j |w_{ij}|$ of absolute weights over all dimensions of the latent space can hence be used as relevance score for input dimension i. Wold et al. [25] also propose to additionally use the amount of variance explained by latent component j as a weighting factor to further include label information into the relevance estimation procedure. However, the latter is not further mentioned in this paper since it always lead to inferior performance in our experiments.

2.7. Base Classifiers for Performance Assessment

For measuring the suitability of feature extraction methods, quantitative methods based on classification accuracy can be used. Apart from the Gaussian process classifier from Sect. 2.5.2, three standard methods [19] well established in chemometrics literature are used for performance assessment.

2.7.1. Linear Discriminant Analysis

One of the most simple classification strategy is to model each class by a normal distribution in feature space. This parametric assumption can be further constrained by restricting all classes to share the same covariance matrix $\boldsymbol{\Sigma}$. After inferring a mean $\boldsymbol{\mu}_c$ for each class $1 \leq c \leq C$ and shared covariance by using unbiased estimates from the training data, test vectors are assigned to classes by using a maximum likelihood rule, i.e. label c is assigned to test vector $\mathbf{x}$ if $c = \operatorname{argmax}_i p(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma})$. It can be shown that the resulting class boundaries based on above decision process are piecewise linear which is the reason for terming this classifier Linear Discriminant Analysis.

2.7.2. Quadratic Discriminant Analysis

As in LDA, Quadratic Discriminant Analysis (QDA) assumes normally distributed data for each class. However, the class-specific covariance matrix is no longer shared but allowed to vary for each class independently.

Using maximum likelihood estimation as in LDA, this higher flexibility leads to a piecewise quadratic decision boundary.

2.7.3. Nearest Neighbor Classifier

The Nearest Neighbor classifier is an easy but powerful non-parametric decision rule. Rather than learning a distribution of some parametric family, the whole training data is stored as a model. A new test data is then assigned to the class of its nearest neighbor in the training dataset according to a predefined similarity measure. In this work, standard Euclidean distance is used for measuring similarities between samples.

3. Experiments

This section deals with the analysis of the relevance criteria presented in Sect. 2 with respect to a Raman spectroscopic classification problem. Information about the Raman spectra dataset and implementations used for our experiment are provided prior to discussing the results.

3.1. Raman Spectra Datasets

The current study is based on two datasets, measured by a micro-Raman setup (HR LabRam invers, Jobin-Yvon-Horiba, Bensheim, Germany). The spectrometer has an entrance slit of 100 μm, has a focal length of 800 mm, and is equipped with a 300-lines/mm grating. As excitation wavelengths the 532-nm line of a frequency doubled Nd:YAG laser (Coherent Compass, Dieburg, Germany) with a laser power of approx. 2.4 mW incident on the sample were used. The Raman scattered light was detected by a CCD camera operating at 220 K. A Leica PLFluoar 100× objective (NA 0.75) focused the laser light onto the samples ($\approx$0.7 μm focus diameter). The spectrometer was calibrated each day prior to measuring (using titanium dioxide). All cells were recorded from fused silica plates with an integration time of 60 s.

The first (large) Raman spectra dataset $\mathcal{D}_L$ contains 6707 spectra from 10 different bacterial species. The species were chosen according to their occurrence in clean-room environments. The microorganisms were purchased from the German Collection of Microorganisms and Cell cultures (DSMZ, Braunschweig, Germany) and from the Institute for Infectious Biology at the University of Würzburg. The employed cultivation media consisted of NA (nutrition agar), S-1-NA (standard 1 nutrition agar), CA (corynebacterium agar) and CASO (trypticase soy yeast extract medium). The microorganisms were cultured under varying conditions with respect to nutrient medium, growing time and temperature.

Table 1: Contents of the bacterial Raman datasets used in this study: The large dataset $\mathcal{D}_L$ and the independent dataset $\mathcal{D}_I$.

genus	species	number of spectra	
		in $\mathcal{D}_L$	in $\mathcal{D}_I$
Bacillus	B. pumilus	534	0
	B. sphaericus	275	15
	B. subtilis	924	8
	B. megatarium	94	0
Escherichia	E. coli	641	84
Micrococcus	M. luteus	1259	51
	M. lylae	186	10
Staphylococcus	S. cohnii	245	25
	S. epidermidis	1884	27
	S. warnerie	665	0

In addition, an independent test set $\mathcal{D}_I$, consisting of 7 out of 10 species included in the training set, was recorded. For performing Raman measurements on single cells the bacteria were extracted from the agar plates and smeared on a fused silica plate. The composition of both datasets is listed in Table 1.

3.2. Spectral Pre-processing

Both datasets $\mathcal{D}_L$ and $\mathcal{D}_I$ are pre-processed by performing local quadratic interpolation to obtain Raman intensities on a fixed (integer) wavenumber grid. All Raman signals are then cropped to the integer wavenumber range $\mathcal{I} = [540, 3350]$ cm^{-1} which is covered by all spectra. In order to suppress spike noise introduced by cosmic radiation, a running median filter is employed. For numerical stability, all spectra are further normalized to unit length.

3.3. Implementation Details

In this experiment, we used Discrete AdaBoost.MH from the *MultiBoost* [48] package and single Decision Stumps as weak learners.

As in [36], we used 100 trees for RDF construction, employing a high randomization for the resampling scheme (10% of the data is randomly drawn for each tree). Furthermore, 250 features are randomly chosen at each node for optimization of the Decision Stumps.

The RLR classifier was trained using an inverse variance parameter of $\lambda = 10$. This rather large value was chosen to generate weight vectors with many components close to zero. The in-house implementation of

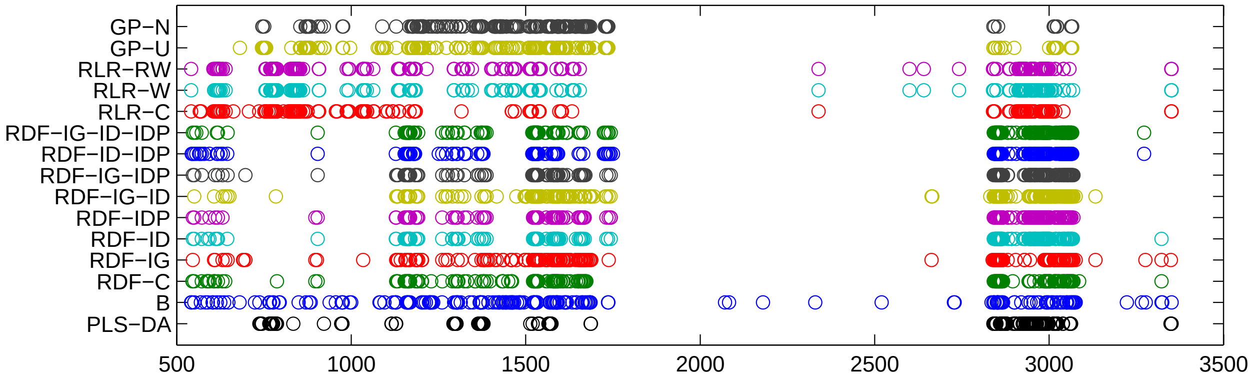

Figure 7: Visual inspection of important features for classification on the species level inferred by all 15 relevance criteria.

RLR used in this work also sets small weight values below a threshold $\epsilon < 10^{-4}$ equal to zero in each iteration of the optimization routine. This property and the further restriction that zero-valued weights are re-adjusted only with a probability of 0.5 additionally encourage the formation of sparse weight vectors.

The GP classifier from the source code provided alongside the book of Rasmussen et al. [22] is adapted to allow joint evidence optimization and Bagging. For the latter, resampling is performed with 25 random draws, each containing 50 spectra per class. For the hyperparameter optimization, the Conjugate Gradient Descent optimizer `minimize` was adapted to enable a LOO-error control mechanism. In all cases, the optimizer is used with initial parameter vector $\boldsymbol{\theta} = (1, \dots, 1)^T$ and a predefined number of function evaluations (30 for ARD-kernel, 10 for SE-kernel).

For PLS-DA, the Matlab function `plsregress` from the *Statistical Toolbox* including its built-in CV option was utilized for all PLS-related analyses. By using a 10-fold CV setup with the number of latent basis vectors ranging from one to 100, the minimum CV error was obtained for 34 latent dimensions. All subsequent results concerning PLS-DA are based on this model.

3.4. Results

In this section, relevance criteria are compared based on a feature selection task. First, all 15 relevance criteria discussed in Sect. 2 are employed on the large Raman dataset $\mathcal{D}_L$ in order to generate a relevance ordering of features. Using this ordering, only the $D' = 200$ most relevant features are retained whereas remaining dimensions are discarded from all features. This results in 15 different subsets $\mathcal{I}_1, \dots, \mathcal{I}_{15}$. For a given subset $\mathcal{I}_k$, both datasets $\mathcal{D}_L$ and $\mathcal{D}_I$ are then projected onto the relevant feature in $\mathcal{I}_k$, generating reduced datasets $\mathcal{D}_L(\mathcal{I}_k)$ and $\mathcal{D}_I(\mathcal{I}_k)$. The 200-dimensional spectra of $\mathcal{D}_L(\mathcal{I}_k)$ are then used to train a classifier.

The above procedure results in a number of 15 different classification models, one for each relevance criterion. The suitability of a given relevance criterion generating subset $\mathcal{I}_k$ can thus be estimated by means of the recognition performance of their corresponding classifiers. This performance is estimated on the independent dataset $\mathcal{D}_I(\mathcal{I}_k)$, utilizing two known measures from pattern recognition: average recognition rate (ARR) and overall recognition rate (ORR). The ORR measure is defined as the percentage of correctly classified test points of the whole dataset. The ARR measure, on the other hand, calculates the recognition rate for each class separately and then averages all class-specific accuracies.

Fig. 3.3 illustrates the relevant feature subsets based on all relevance criteria from Sect. 2, plotted according to their wavenumbers. It can be seen, that a clear pattern arises with two clusters at wavenumbers about 700 – 1800 cm^{-1} (fingerprint region) and 2800 – 3100 cm^{-1} (high wavenumber region). While the fingerprint region is usually used for Raman spectroscopic analyses, the high wavenumber region is often discarded. However, it has been shown that this region also contains discriminant information for classification [49, 50, 51], most likely due to C-H stretching vibrations of compounds such as triglycerides ($\approx$ 2850 cm^{-1}) and proteins ($\approx$ 3000 cm^{-1}). The latter two wavenumber regions are highlighted by each single relevance criterion (with the clearest separation of both regions by ARD-U and ARD-N) which indicates that C-H stretching vibrations of fatty acids and proteins can provide useful information for the discrimination of bacterial species.

The lack of relevant features within the range 1800–2800 cm^{-1} is also supported by chemical properties of organic compounds which rarely produce excita-

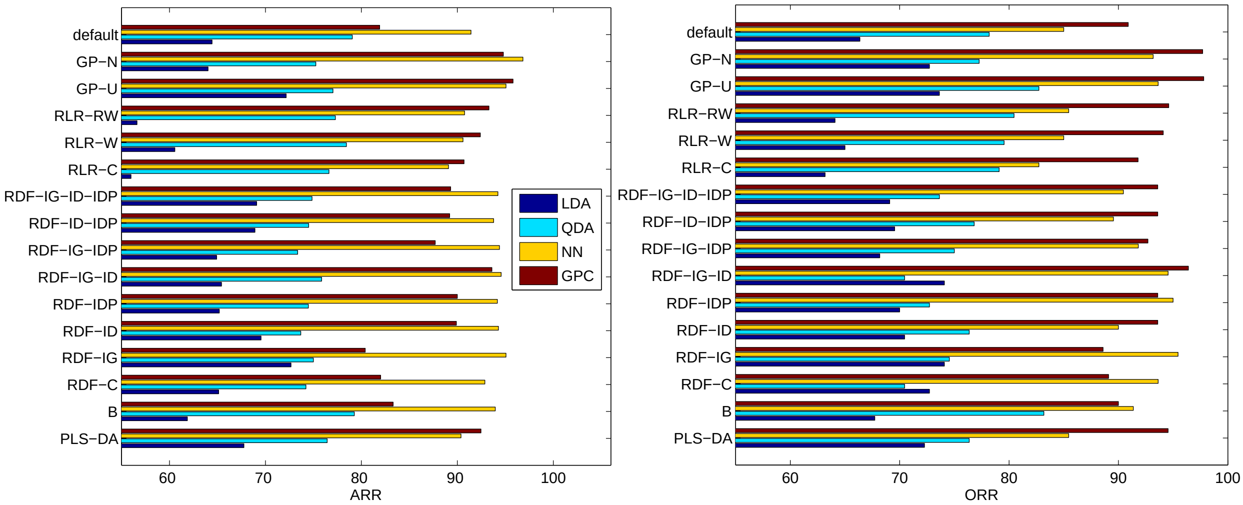

Figure 8: Classification results (ARR and ORR) for different classifier architectures employing the proposed methods using the 200 most relevant features. The tag "default" denotes the standard approach for the respective classifier: while for kernel-based classifiers (NN and GP classifier) no dimension reduction is used, PCA projection onto the first 200 eigenvectors is performed for LDA and QDA in order to avoid numerical issues.

tion peaks in this frequency band. The above observations supports the relevance criteria used in this work. However, notable differences occur in the range 540–650 cm^{-1} and 650–1150 cm^{-1}. The first, which might be attributed to glycosidic bonds (polysaccharides) and disulfide bonds (polypeptides), is not selected by ARD-based criteria. The second range, however, is not included in the relevant set of RDF based methods (with the exception of a small strip at ≈900 cm^{-1}).

The recognition accuracies with respect to different feature subsets were empirically measured by means of the classifiers introduced in Sect. 2.7 (LDA, QDA and NN) and a GP classifier using the SE-kernel (17) which has been recently shown to produce excellent results compared to other state-of-the-art methods [23]. Note that the GP classifier results discussed in the following are obtained using the bag-wise hyperparameter optimization (see Sect. 2.5.3). The joint optimization strategy does not lead to an improvement in our experiments (and is thus omitted).

Apart from the 15 feature subsets mentioned above we also analyzed all classifiers in a default setting. While the whole set of 2811 features is used as default for the non-parametric classifiers (NN and GP classifier), PCA was employed as preprocessing step for the normal distribution based methods (LDA and QDA) to avoid numerical problems by inverting the covariance matrices. The results are graphically depicted in Figure 8. Beside the fact that the GP classifier and NN always outperform LDA and QDA, no clear pattern is followed by all four tested classification methods. While LDA, NN and GP classifier prefer RDF-based and ARD-based measures, QDA yields best results for AdaBoost.MH (B), RLR and ARD-based scores. The well-known PLS-DA method provides useful scores yielding comparable results to RLR-W for most classification techniques. It can be further seen that, throughout all classifiers, ARD-U serves as a method of choice since it constantly leads to high recognition rates measured by both ARR and ORR.

It should be also noted that the simple NN rule consistently produces high recognition rates, often outperforming all other classifiers. The highest overall recognition rate (97.8%), however, is obtained by using the GP classifier whose detailed recognition results are shown in Table 2. Using the GP classifier based on ARR performance, 14 out of 15 relevance criteria lead to a better recognition performance compared to the default setting (all features). This behavior which is also followed by NN is encouraging, considering that less than 10% of the original features are used. This is an additional evidence for the suitability of the proposed feature selection methods.

What remains to show is that this increase in recognition accuracy does not stem from the mere reduction of dimensionality. To validate this hypothesis, we randomly draw 1000 artificial feature subsets containing 200 wavenumbers and analyze their quality using recognition performance. For this experiment, the NN classification rule was employed due to its good performance using the subsets above and its fast learning and prediction process. The results are visualized in Figure 9

Table 2: Detailed recognition performance on independent Raman dataset $\mathcal{D}_I$, using varying sets of relevant features (each containing $D' = 200$ elements) obtained by relevance criteria from Sect. 2. For the sake of readability, only detailed results from the classifier which lead to the highest overall recognition rate (GP classifier) are shown (c.f. Figure 8). The criterion leading to the highest prediction accuracy is highlighted in boldface.

methods	B.sph.	B.sub.	E.coli	M.lut.	M.lyl.	S.coh.	S.epi.	ARR	ORR
ALL	0	8	82	51	8	24	27	81.9	90.9
ARD-N	11	8	84	51	9	25	27	94.8	97.7
ARD-U	14	8	83	51	9	23	26	**95.8**	**97.8**
RLR-RW	13	8	84	48	10	18	27	93.3	94.6
RLR-W	13	8	84	49	10	16	27	92.4	94.1
RLR-C	13	8	84	45	10	16	26	90.7	91.8
RDF-IG-ID-IDP	8	7	82	50	10	23	26	89.3	93.6
RDF-ID-IDP	6	8	84	49	10	23	26	89.2	93.6
RDF-IG-IDP	7	7	83	49	10	23	25	87.7	92.7
RDF-IG-ID	10	8	84	51	10	23	26	93.6	96.4
RDF-IDP	7	8	83	49	10	23	26	90.0	93.6
RDF-ID	10	7	83	49	10	21	26	89.9	93.6
RDF-IG	0	8	83	51	10	17	26	80.4	88.6
RDF-C	3	8	84	50	10	15	26	82.0	89.1
B	4	8	84	51	10	15	26	83.3	90.0
PLS-DA	13	8	84	51	10	17	25	92.5	94.6
#spectra	15	8	84	51	10	25	27		

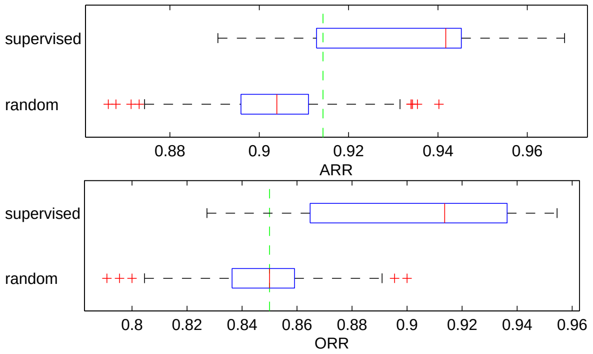

Figure 9: Quantitative analysis of 15 supervised feature selection methods compared to 1000 randomly selected feature subsets using the NN classifier. The result obtained using all 2811 features/wavenumbers is visualized as a vertical spaced line.

via boxplots showing the accuracy for both supervised and randomly generated feature subsets. It can be seen that the median of the random subset accuracies is close to the result using the default setting, i.e. on average there does not seem to be an improvement over using all features. It is also obvious that subsets from supervised selection techniques leads to a significant increase in recognition accuracy compared to randomly generated subsets. This observation additionally justifies the use of supervised feature selection methods proposed in this paper.

The above results suggest that responses in wavenumber regions below 650 cm^{-1} and above 3100 cm^{-1} are not relevant for discriminating between bacterial species contained in the independent Raman spectra dataset $\mathcal{D}_I$. This follows from the fact that ARD-based measures which do not select any wavenumbers in that range are competitive to all other feature selection methods in the quantitative analysis.

4. Conclusions and Future Work

This work aims at comparing different supervised classification techniques for the task of feature selection. In order to extract a subset of features that is relevant for discriminating between bacterial species, 15 relevance criteria based on five different classifiers are employed. Apart from using techniques which are known in the field of Raman spectroscopy (AdaBoost, Random Decision Forests, Regularized Logistic Regression), we also applied a Gaussian process classifier with Automatic Relevance Determination which, to our knowledge, has not been used for feature selection in a vibrational spectroscopic context.

All relevance criteria are applied on a large Raman spectra dataset, consisting of 10 different bacterial species. A visual observation of the most relevant 200 features revealed that all methods selected wavenumbers within reasonable spectral regions (fingerprint and high wavenumber region). All inferred feature subsets are additionally ranked by a classification task, using an independent Raman dataset. Compared to classification without wavenumber selection, the majority of relevance criteria achieved a higher recognition rate using appropriate classification techniques. Among all tested feature selection methods, Automatic Relevance Determination turned out to be particularly useful. Moreover, our results indicate that non-parametric classifiers such as the Nearest Neighbor rule and the GP classifier are methods of choice since they outperform standard parametric techniques by achieving high overall recognition rates up to 97.8%. A further comparison with randomly generated feature subsets showed that using supervised feature selection criteria is beneficial and that the increase in recognition performance is not a mere byproduct of dimension reduction.

Further improvements of the proposed criteria are also possible. E.g. the AdaBoost based selection procedure might take weights α into account for reweighting feature relevance. Regularized Logistic Regression might be used with an l_1 loss instead of an l_2-loss which would lead to very sparse weight vectors ω. Furthermore, the selection procedure used in this paper concentrates on a predefined number ($D' = 200$) of relevant features. A fast procedure for obtaining this number automatically for a given relevance criterion would be advantageous. Last but not least, the visual analysis in Sect. 3 shows that relevant features usually cluster in prominent Raman bands. However, it is often suggested in literature that neighboring spectra share a large amount of information. Hence, methods which choose features from different bands are highly worth investigating in future work.

Acknowledgments

We want to thank Mario Krause and Michaela Harz from the Institute of Physical Chemistry at the FSU Jena for capturing the bacterial datasets as well as Petra Rösch and Jürgen Popp for supervision and outlining goals in the field of Raman spectroscopy. Additional thanks go to Erik Rodner and Björn Fröhlich for their contribution in implementing algorithms and participating in fruitful discussions. This work was funded by the TMBWK ProExzellenz project "MikroPlex" (PE113-1).

References

[1] W. Mark A. Thompson. Planaria Software LLC, Seattle, ArgusLab 4.0.1, http://www.arguslab.com, 2005.

[2] P. Buijtels, H. Willemse-Erix, P. Petit, H. Endtz, G. Puppels, H. Verbrugh, A. van Belkum, D. van Soolingen, K. Maquelin, Rapid identification of mycobacteria by Raman spectroscopy, J. Clin. Microbiol. 46 (3) (2008) 961–965.

[3] D. I. Ellis, R. Goodacre, Rapid and quantitative detection of the microbial spoilage of muscle foods: current status and future trends, Trends Food Sci. Tech. 12 (11) (2001) 414 – 424.

[4] P. Rösch, M. Harz, K.-D. Peschke, O. Ronneberger, H. Burkhardt, H.-W. Motzkus, M. Lankers, S. Hofer, H. Thiele, J. Popp, Chemotaxonomic Identification of Single Bacteria by Micro-Raman Spectroscopy: Application to Clean-Room-Relevant Biological Contaminations, Appl. Environ. Microb. 71 (2005) 1626–1637.

[5] D. Ivnitski, I. Abdel-Hamid, P. Atanasov, E. Wilkins, Biosensors for detection of pathogenic bacteria, Biosens. Bioelectron. 14 (7) (1999) 599–624.

[6] G. Barrow, R. Feltham, Cowan and Steel's Manual for the Identification of Medical Bacteria, Cambridge University Press, UK, 3 edn., 1993.

[7] S. Al-Khaldi, M. Mossoba, Gene and Bacterial Identification Using High Throughput Technologies: Genomics, Proteomics, and Phenomics., Nutrition 20 (2004) 32–38.

[8] J. Chalmers, P. Griffiths (Eds.), Handbook of Vibrational Spectroscopy, vol. 1-5, Wiley, 2002.

[9] F. Siebert, P. Hildebrandt, Vibrational Spectroscopy in Life Science, Wiley, 2007.

[10] M. Diem, P. Griffiths, J. Chalmers (Eds.), Vibrational Spectroscopy for Medical Diagnosis, Wiley, 2008.

[11] R. McCreery, Raman Spectroscopy for Chemical Analysis, vol. 157, Wiley-Interscience, 2000.

[12] J. Ferraro, K. Nakamoto, C. Brown, Introductory Raman Spectroscopy (Second Edition), Elsevier, 2003.

[13] W. Nelson, J. Sperry, UV resonance raman spectroscopic detection and identification of bacteria and other microorganisms, Modern Techniques for Rapid Microbiological Analysis (1991) 97–143.

[14] S. Chadha, W. Nelson, J. Sperry, Ultraviolet micro-Raman spectrograph for the detection of small numbers of bacterial cells, Rev. Sci. Instrum. 64 (11) (1993) 3088–3093.

[15] A. Berger, Q. Zhu, Identification of oral bacteria by Raman microspectroscopy, J. Mod. Opt. 50 (6) (2003) 2375–2380.

[16] R. M. Jarvis, R. Goodacre, Ultra-violet resonance Raman spectroscopy for the rapid discrimination of urinary tract infection bacteria, FEMS Microbiol. Lett. 232 (2) (2004) 127–132.

[17] G. Puppels, F. de Mul, C. Otto, J. Greve, M. Robert-Nicoud, D. Arndt-Jovin, T. Jovin, Studying single living cells and chromosomes by confocal Raman microspectroscopy, Nature 347 (1990) 301–303.

[18] U. Schmid, P. Rösch, M. Krause, M. Harz, J. Popp, K. Baumann, Gaussian mixture discriminant analysis for the single-cell differentiation of bacteria using micro-Raman spectroscopy, Chemometr. Intell. Lab. 96 (2) (2009) 159 – 171.

[19] C. M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer, 1 edn., 2007.

[20] V. Vapnik, The nature of statistical learning theory, Springer-Verlag New York, Inc., New York, NY, USA, ISBN 0-387-94559-8, 1995.

[21] B. Schölkopf, A. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, MIT Press, Cambridge, MA, USA, ISBN 0262194759, 2001.

[22] C. E. Rasmussen, C. K. I. Williams, Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning), The MIT Press, 2005.

[23] M. Kemmler, J. Denzler, P. Rösch, J. Popp, Classification of Microorganisms via Raman Spectroscopy Using Gaussian Processes, in: Proc. DAGM, 2010.

[24] F. Rossi, A. Lendasse, D. François, V. Wertz, M. Verleysen, Mu-

tual information for the selection of relevant variables in spectrometric nonlinear modelling, Chemometr. Intell. Lab. 80 (2) (2006) 215 – 226.

[25] S. Wold, M. Sjöström, L. Eriksson, PLS-regression: a basic tool of chemometrics, Chemometr. Intell. Lab. 58 (2) (2001) 109–130.

[26] M. Barker, W. Rayens, Partial Least squares for discrimination., J. Chemom. 17 (2003) 166–173.

[27] K. Fukunaga, Introduction to Statistical Pattern Recognition, Academic Press, 2nd ed., 1990.

[28] I. Guyon, A. Elisseeff, An Introduction to Variable and Feature Selection, J. Mach. Learn. Res. 3 (2003) 1157–1182.

[29] R. Jarvis, R. Goodacre, Genetic algorithm optimization for preprocessing and variable selection of spectroscopic data, Bioinformatics 21 (7) (2005) 860–868.

[30] J. Bi, K. Bennett, M. Embrechts, C. Breneman, M. Song, Dimensionality Reduction via Sparse Support Vector Machines, J. Mach. Learn. Res. 3 (2003) 1229–1243, ISSN 1532-4435.

[31] J. Weston, A. Elisseeff, B. Schölkopf, M. Tipping, Use of the Zero-Norm with Linear Models and Kernel Methods, J. Mach. Learn. Res. 3 (2003) 1439–1461, ISSN 1532-4435.

[32] R. E. Schapire, Y. Singer, Improved Boosting Algorithms Using Confidence-rated Predictions, Mach. Learn. 37 (3) (1999) 297–336.

[33] L. Breiman, Random Forests, Mach. Learn. 45 (1) (2001) 5–32.

[34] P. Viola, M. Jones, Rapid Object Detection using a Boosted Cascade of Simple Features, in: Proc. CVPR, 511–518, 2001.

[35] J. Rogers, S. Gunn, Ensemble Algorithms for Feature Selection, in: Sheffield Machine Learning Workshop, 2004.

[36] B. Menze, B. Kelm, R. Masuch, U. Himmelreich, P. Bachert, W. Petrich, F. Hamprecht, A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data, BMC Bioinformatics 10 (2009) 213–228.

[37] R. Neal, Bayesian Learning for Neural Networks, Springer, 1996.

[38] W. Iba, P. Langley, Induction of One-Level Decision Trees, in: Proc. ICML, 1992.

[39] L. Breiman, Bagging Predictors, Mach. Learn. 24 (1996) 123–140.

[40] J. Rogers, S. Gunn, Identifying Feature Relevance Using a Random Forest, in: Subspace, Latent Structure and Feature Selection, 173–184, 2006.

[41] R. von Mises, Mathematical theory of probability and statistics, Academic Press, 1964.

[42] D. MacKay, Information Theory, Inference & Learning Algorithms, Cambridge University Press, 2002.

[43] T. P. Minka, A family of algorithms for approximate bayesian inference, Ph.D. thesis, supervisor-Picard, Rosalind, 2001.

[44] H. Nickisch, C. E. Rasmussen, Approximations for Binary Gaussian Process Classification, J. Mach. Learn. Res. 9 (2008) 2035–2078.

[45] Y. A. Qi, T. P. Minka, R. W. Picard, Z. Ghahramani, Predictive automatic relevance determination by expectation propagation, in: Proc. ICML, ACM, New York, NY, USA, ISBN 1-58113-828-5, 85, doi:http://doi.acm.org/10.1145/1015330.1015418, 2004.

[46] M. Opper, O. Winther, Gaussian Processes for Classification: Mean-Field Algorithms, Neural Comput. 12 (11) (2000) 2655–2684, ISSN 0899-7667, doi: http://dx.doi.org/10.1162/089976600300014881.

[47] A. Kapoor, K. Grauman, R. Urtasun, T. Darrell, Gaussian Processes for Object Categorization, Int. J. Comput. Vision 88 (2) (2010) 169–188.

[48] N. Casagrande, MultiBoost: An open source multi-class AdaBoost learner, http://iro.umontreal.ca/ casagran/multiboost/, 2005.

[49] S. Koljenovic, T. Bakker Schutt, R. Wolthuis, B. de Jong, L. Santos, P. Caspers, J. Kros, G. Puppels, Tissue characterization using high wave number Raman spectroscopy, J. Biomed. Opt. 10 (2005) 031116.

[50] A. Nijssen, K. Maquelin, L. F. Santos, P. J. Caspers, T. C. Bakker Schut, J. C. den Hollander, M. H. A. Neumann, G. J. Puppels, Discriminating basal cell carcinoma from perilesional skin using high wave-number Raman spectroscopy, J. Biomed. Opt. 12 (3) (2007) 034004–+, doi:10.1117/1.2750287.

[51] A. Chau, Development of an intracoronary Raman spectroscopy, Ph.D. thesis, Massachusetts Institute of Technology. Dept. of Mechanical Engineering., 2009.