Optimal classification of standoff bioaerosol measurements using evolutionary algorithms

Ragnhild Nyhavn, Hans J. F. Moen, Øystein Farsund, Gunnar Rustad

Ragnhild Nyhavn, Hans J. F. Moen, Øystein Farsund, Gunnar Rustad, "Optimal classification of standoff bioaerosol measurements using evolutionary algorithms," Proc. SPIE 8018, Chemical, Biological, Radiological, Nuclear, and Explosives (CBRNE) Sensing XII, 801806 (3 June 2011); doi: 10.1117/12.883919

Event: SPIE Defense, Security, and Sensing, 2011, Orlando, Florida, United States

Optimal classification of standoff bioaerosol measurements using evolutionary algorithms

Ragnhild Nyhavn, Hans J. F. Moen, Øystein Farsund and Gunnar Rustad[1]
Norwegian Defence Research Establishment (FFI), Instituttveien 20, 2007 Kjeller, Norway

ABSTRACT

Early warning systems based on standoff detection of biological aerosols require real-time signal processing of a large quantity of high-dimensional data, challenging the systems efficiency in terms of both computational complexity and classification accuracy. Hence, optimal feature selection is essential in forming a stable and efficient classification system. This involves finding optimal signal processing parameters, characteristic spectral frequencies and other data transformations in large magnitude variable space, stating the need for an efficient and smart search algorithm. Evolutionary algorithms are population-based optimization methods inspired by Darwinian evolutionary theory. These methods focus on application of selection, mutation and recombination on a population of competing solutions and optimize this set by evolving the population of solutions for each generation. We have employed genetic algorithms in the search for optimal feature selection and signal processing parameters for classification of biological agents. The experimental data were achieved with a spectrally resolved lidar based on ultraviolet laser induced fluorescence, and included several releases of 5 common simulants. The genetic algorithm outperform benchmark methods involving analytic, sequential and random methods like support vector machines, Fisher's linear discriminant and principal component analysis, with significantly improved classification accuracy compared to the best classical method**.**

Keywords: standoff bioaerosol measurements, pattern recognition, feature selection, feature extraction, genetic algorithms, Bayesian normal classifier.

1. INTRODUCTION

Today there are several methods for identifying biological substances using microbiological techniques [1]. Although these are precise techniques, they are quite slow compared to optical techniques which give response within seconds. Efforts to develop equipments for rapid standoff detection of biological substances exist in several nations [2-4]. The Norwegian Defence Research Establishment (FFI) has built an experimental lidar [5-6] for the purpose of near real time standoff detection and classification of biological aerosols. By analyzing laser-induced fluorescence from different biological aerosols, a classifier can detect biological aerosol clouds and potentially classify the threat. A pattern recognition system of low classification error rate is demonstrated for data acquired during a field test where five different simulants were released: BG *(Bacillus Atrophaeus)* and BT *(Bacillus Thuringensis)* as simulants for encapsulated bacteria*,* OV *(Ovalbumin)* for simulating toxins*,* MS2 *(Bacteriophague MS2)* for simulating virus and EH *(Pantoea Agglomerans)* as simulant for vegetative bacteria.

1.1. Feature optimization

The training of a classifier is based on features, i.e. measured characteristics describing the substances in consideration. Hence, the selection of good features is important in terms of classification accuracy. In spectral data analysis the objects to classify are the fluorescence spectra (light measurements spanned by channels representing different frequency ranges across the spectrum of visible light). In this relation, feature selection aims at localizing the spectral ranges of significance, selecting channels able to discriminate the different substances and ignoring the rest. Figure 1

[1] Gunnar.rustad@ffi.no; phone: +47 63807380

Chemical, Biological, Radiological, Nuclear, and Explosives (CBRNE) Sensing XII,
Edited by Augustus W. Fountain III, Patrick J. Gardner, Proc. of SPIE Vol. 8018, 801806
© 2011 SPIE · CCC code: 0277-786X/11/$18 · doi: 10.1117/12.883919

shows normalized fluorescence spectra of the five different simulants used in this study and demonstrates frequency ranges of dissimilar discriminative ability.

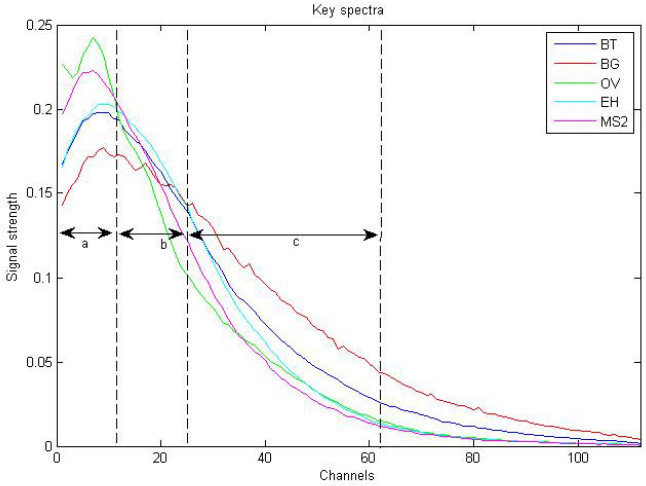


Figure 1: Key fluorescence (normalized) of the five simulants used along with the demonstration of dissimilar discriminative abilities in different frequency ranges, motivating feature selection. In ranges ***a*** and ***c*** the spectra are easier to distinguish than in range ***b*** having several intersecting points. This indicates that the features are more discriminative in ranges ***a*** and ***c*** than in range ***b***.

The high number of spectral channels (112 channels) yields high dimensionality data, making analysis of fluorescence a challenging task. This difficulty is shown in Figure 2 by a typical noisy sample of a lidar observation. The optimization of feature selection and feature extraction is expected to improve classification accuracy, extracting characteristic information from the non-discriminative data of already extracted features. This involves searching for characteristic subsets of the channels and other class discriminative transformations of the data sets.

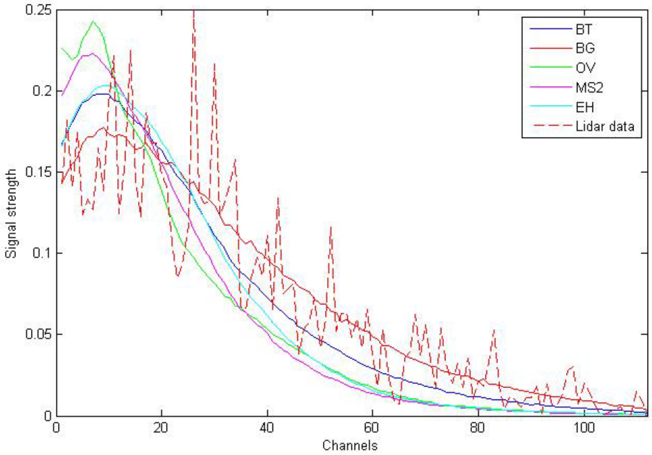


Figure 2: Fluorescence of the five simulants (multiple colors) and one lidar observation (red).

In this paper *feature extraction* is used as a general transformation of a data set; hence, signal processing can be referred to as feature extraction. This can be linear or nonlinear transformations and can result in both down-sampling the dimensionality or projecting the data to a higher-dimensional space. The familiar data transformations Principal Component Analysis (PCA) [7] and Fisher's Linear Discriminant (FLD) [7] focus on linear combinations of features representing each class or discriminating the classes, respectively, and are examples of data transformation methods down-sampling the dimensionality. High-dimensionality projections can be exemplified by Support vector machine (SVM) [7], projecting the data to an often much higher-dimensional space where two classes can be discriminated by a hyper plane with an optimized distance margin to each class.

Feature selection is the selection of a subset of the originally provided features. This set is used to train the classifier with observational data spanned by the features and is essential for the quality of a classifier, both in improving classification accuracy when selecting characteristic features and in down-sampling the dimensionality by ignoring specific dimensions of the extracted feature set. Recently, feature selection has been of increased interest as the growing computational capacity opens for solving high-complexity problems, involving a large amount of features. Moreover, the demand for real-time processing makes limiting of features important for efficient algorithms and are often vital within automatic systems.

1.2. Scope and objective

In this paper we apply genetic algorithms (GAs), an evolutionary search technique, on the combinatorial optimization problem of finding an optimal transformation of the dataset. The study involves classification of the five common simulants. Background (non-harmful substance) is an additional class, resulting in a 6-categorial classification problem. The experimental data used for training and testing the classifier were achieved with a spectrally resolved lidar and consist of 112 channels.

The linear Bayesian normal classifier is used, assuming independently observed samples, multivariate Gaussian class-densities and equally correlated features for all classes which in practice is the use of the Mahalanobis distance [7] to classify a sample. The only known condition as regards the classification problem is that the features of spectral data are correlated [8]. Nevertheless, the adaptive algorithm will adjust to these assumptions and calculate the classification error to each solution.

1.3. Instrument

The FFI-developed lidar is based on ultraviolet laser-induced fluorescence. A UV laser pulse at 355 nm is emitted from the lidar and backscattered light is collected by a $12''$ telescope. The backscattered light collected is focused into an optical fiber and transmitted to the spectrally resolved detection system. This system consists of a spectrograph and an intensified CCD camera which records the received spectrum between 350 nm and 700 nm after each laser pulse (10 Hz pulse rate). The camera has an adjustable gain that can be varied from 0 to 1000 by adjusting the voltage across the intensifier stage, and can be rapidly turned on and off for range gating.

The backscattered light is recorded in 141 spectral channels. Elastic scattering and Raman scattering by atmospheric constituents dominate the inelastic scattering (fluorescence) and often lead to saturation. The signal in this spectral range (wavelengths shorter than 420 nm) is therefore discarded before the rest of the spectrum is normalized. A typical spectrum is shown in Figure 3, indicating the discarded wavelengths below the dotted line. Thus, the data sets contain 112 channels, corresponding to 2.5 nm per channel.

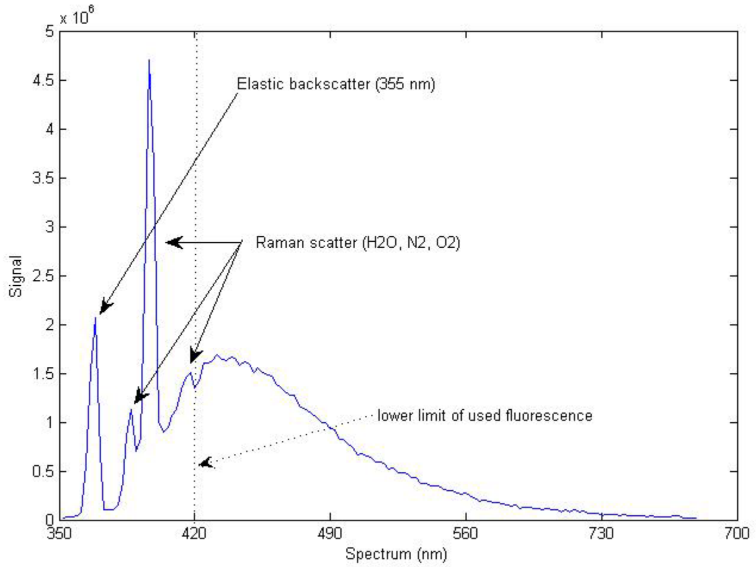


Figure 3: Example of lidar output from 350 to 700 nm. The elastic backscatter signal is damped several orders of magnitude by a CG385 color glass. The dotted vertical line shows the lower limit of used fluorescence in this work.

2. FEATURE OPTIMIZATION USING GENETIC ALGORITHMS

This section aims at explaining methodology concerning feature optimization involving pattern recognition and genetic algorithms, one class of evolutionary algorithms.

2.1. Pattern recognition

Pattern recognition is the act of taking an action based on observed feature measures describing an object. The function controlling these actions is the *classifier* which can be defined based on observations representing the different classes. Using this technique we say that the classifier is *trained*; based on the measured features connected to each observed sample (i.e. a fluorescence spectrum) the classifier calculates a density function for each class.

Let $x \in \mathrm{R}^n$ be one observed sample of n measured features. Given a training set $X_{train} \in \mathrm{M}^{m*n}$ ($m*n$ -matrix where m is the number of samples) containing samples describing each of the classes, a classifier

$$c : X_{train} \to Y$$

is created, where Y is the set of real labels connecting each sample in the training set to its respective class. The classifier is now trained and able to classifying an arbitrary sample $x \in \mathrm{R}^n$, that is

$$c : \mathrm{R}^n \to Y.$$

The selection of a subset of features or a transformed set of features will create a different classifier and will, depending on the discriminative quality of the features, probably have classification accuracy different from the one utilizing all the available features. The classifier using a subset or transformed set of features is similarly created as follows.

Given a subset of the features $s \subseteq \{1, 2, \ldots, n\}$, let $X^{'}_{train} \in \mathrm{M}^{m*|s|}$ be the transformed training set containing the selected features (columns of the original set X_{train}). The classifier is now trained and created as

$$c : X'_{train} \rightarrow Y.$$

After training, this mapping can classify an arbitrary sample $x \in \mathrm{R}^{|s|}$, that is,

$$c : \mathrm{R}^{|s|} \rightarrow Y.$$

By comparing estimated labels to the key labels when classifying a second data set (test data) we get the classification accuracy utilizing the selected features or feature transformation.

2.2. Decision variable space

The decision variable space defines the range and restrictions of the transformation search, which aims at improving classification accuracy.

These are selected to be parameters for

- Channel selection:
  $s \in \{0,1\}^{112}$, where $s_i = 1$ if channel i is selected, and 0 otherwise.
- Channel binding:
  $b \in \{0,1\}^{112}$, where $b_i = 1$ if channel i is to be bound to the next channel, and 0 otherwise. Bounded channels are simply the mean of the channel signals which form new features.
- Time integration:
  $t \in \{0,1\}^k$, where t is a k-bit binary number representing interval size for time integration. Let $t_{real} = l$ be the real number coded by the binary number t. This part of the data transformation integrates the signal in intervals of l samples, smoothing the signal.

Any candidate solution of a transformation is now a specific set (s, b, t) and will aim at selecting characteristic channels and increasing the SNR by integrating the signals in frequency and time. These parameters are assumed to be non-independent; therefore the search for an optimal combination involves a large solution space and most likely a non-analytic problem structure. In search for the optimal combination we have used genetic algorithms, one of few optimization methods able to handle this optimization problem.

2.3. Genetic algorithms

GAs is one of the most popular classes of evolutionary algorithms [9-10]. These are stochastic optimization methods which improve a set of solutions influenced by a quality measure called the fitness function. The optimization process is based on the concept of environmental adaptation, the chance for a solution to survive over time.

The GAs are able to handle large search areas without having to search through all solutions. Guided by the fitness function, the evolutionary operators ensure that high performance solutions are preserved over time. These operators are *selection*, *crossover* and *mutation*.

2.3.1. Solution representation

Genetic algorithms use chromosomes for representing solutions, a string (usually) of binary bits coding the parameters spanning the decision variable space. Figure 4 gives an example of a chromosome, consisting of the three parameter vectors s, b and t which code for a data transformation. The genetic algorithm will evolve a set of these candidate solutions over time and seek the optimal parameters which give the best fitness value, in this relation being the classification error rate.

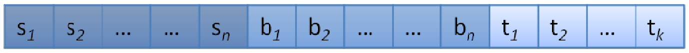


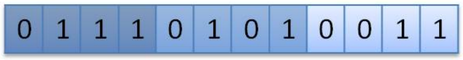
Figure 4: A chromosome representing a solution involving three parameter vectors s, b and t, coding for a data transformation.

Example

Consider a data set of 4 channels and 6 samples, i.e. 6 observed fluorescence spectra spanned by 4 channels,

$$D = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \\ a_{51} & a_{52} & a_{53} & a_{54} \\ a_{61} & a_{62} & a_{63} & a_{64} \end{pmatrix}$$

and a chromosome c coding the signal processing

C =


The first four bits coding for channel selection imply channel 2, 3 and 4 to be selected. The next four bits code for channel binding and in this example channel 2 and 4 are coded to merge with the next channel. Channel 2 is to be bound to channel 3, which is a valid event since channel 2 and 3 are actually part of the selected subset. Since channel 4 is the last channel this binding is ignored. The resulting data after channel selection and binding is

$$D^{'} = \begin{pmatrix} \frac{1}{2}(a_{12}+a_{13}) & a_{14} \\ \frac{1}{2}(a_{22}+a_{23}) & a_{24} \\ \frac{1}{2}(a_{32}+a_{33}) & a_{34} \\ \frac{1}{2}(a_{42}+a_{43}) & a_{44} \\ \frac{1}{2}(a_{52}+a_{53}) & a_{54} \\ \frac{1}{2}(a_{62}+a_{63}) & a_{64} \end{pmatrix} = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \\ b_{41} & b_{42} \\ b_{51} & b_{52} \\ b_{61} & b_{62} \end{pmatrix}.$$

The transformed data set $D^{'}$ consist of only two features. Last, the data set is smoothed over time coded by the last four bits in the chromosome, coding the number 3. The data set is now integrated in intervals of three, with a moving average.

2.3.2. Fitness evaluation

The fitness evaluation of each chromosome solution involve transforming the training data and test data, training the classifier on transformed training data and testing the performance of the classifier by counting the error classifications from the transformed test data set.

For each chromosome, do

1. *Decode* : extract and decode parameters s, b and t
2. *Transform data: training data and test data are transformed separately, based on the parameters s, t and b.*

$$f(s,t,b) : X_{data} \rightarrow X^{'}_{data}$$

3. *Train classifier:* based on transformed training data

$$g(s,t,b) : X^{'}_{traindata} \rightarrow Y$$

4. *Fitness evaluation:* Count number of misclassifications among samples from the transformed test data set. This is given by the error function

$$\varepsilon_{s,t,b} = \sum_{x \in X^{'}_{testdata}} \chi_{g(x)}$$

where $\chi_{g(x)}$ is simply an indicator function returning value 1 if estimated label is not identical to the key label.

2.3.3. Optimization with a genetic algorithm

One iteration of a genetic algorithm is illustrated in Figure 5. A population P with a constant number of chromosomes is evaluated and ranked by calculating fitness. Then, it is reproduced calling evolutionary operators, forming a new population. Over time, this solution set will presumably evolve towards an optimal solution, providing the best parameters evaluated by the fitness function. Genetic algorithms cannot guarantee the finding of an optimal solution, but the presence of elitism (explained in *Operators*) will guarantee weak optimization, i.e. $F(y^*_{T+1}) \geq F(y^*_T)$ where F is the fitness function and y^*_T is the best solution at generation T .

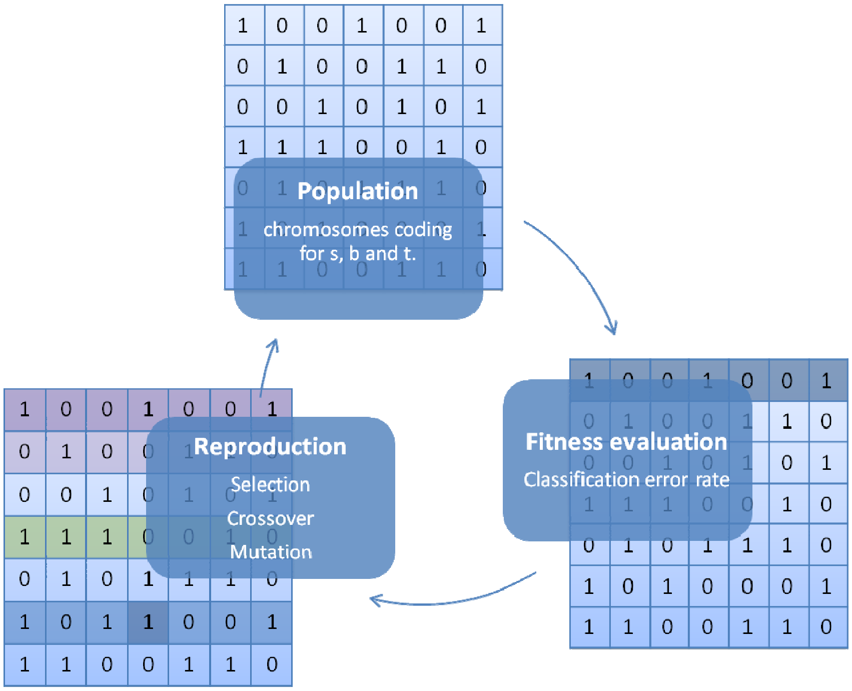


Figure 5: Sketch of one iteration of a genetic algorithm. A population of solutions is evaluated by the fitness function before it is reproduced calling evolutionary operators.

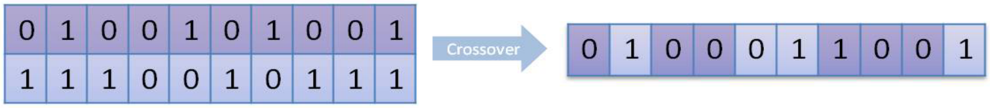
Through testing and learning, the genetic algorithm can simultaneously optimize the three parameters, guided by the fitness function. The pseudo code of the employed optimization algorithm using a genetic algorithm is as follows:

- Generate an initial population P_0 of N chromosomes and an empty population Q_T of $N - N_{elites}$ chromosomes where N_{elites} is the chosen number of elites guaranteed survival for each generation.
- For each generation T, do
  1. *Evaluate fitness* of each chromosome $c \in P_T$.
  2. *Rank* P_T based on fitness.
  3. *Copy* N_{elites} *elites* (best solutions) from P_T to P_{T+1}.
  4. *Perform selection* from P_T to form a parent population of $N - N_{elites}$ chromosomes, based on fitness ranking. Make one copy of the parent population and randomly permute the order of chromosomes in the two identical parent populations.
  5. *Perform bit-wise crossover* by selecting parents in pairs, one from each of the two permutated parent populations, and randomly select bits from the two. Save the offspring in Q_T.
  6. *Mutate* the leftover parent solutions from one parent population not being exposed for crossover, and save the offspring in Q_T.
  7. Fill the rest of the slots in P_{T+1} with the reproduced Q_T.

2.3.4. Operators

Prior to the reproduction of the population, the candidates for reproduction are carefully selected by the *selection* operator. The selection is influenced by the solutions' classification accuracy and ensures that the solutions of high fitness have a greater chance of survival than the solutions of poor fitness. Selection also preserves the best solutions of each generation (elites). This elite-mechanism ensures that the best solution of each generation never is degraded, guaranteeing weak optimization.

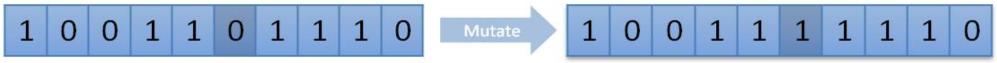
When solutions are selected they are candidates for reproducing the new population. Some of the candidates are used for *crossover;* reproducing an offspring solution by randomly selecting bits from two parent solutions as exemplified in Figure 6.


0 1 0 0 1 0 1 0 0 1
1 1 1 0 0 1 0 1 1 1

0 1 0 0 0 1 1 0 0 1

Figure 6: Crossover of two parent solutions. The offspring's bits have 50% probability of being selected from one parent or the other.

Further, another part of the reproduction candidates are mutated. *Mutation* flips a small part of a solution (usually just a single bit, called 'bit-wise mutation', from 0 to 1 or reversely) as shown in Figure 7.


1 0 0 1 1 0 1 1 1 0

1 0 0 1 1 1 1 1 1 0

Figure 7: Bit-wise mutation of a chromosome, changing one single bit from 0 to 1.

2.4. Data and optimization parameters

All training data are achieved during (controlled) semi-closed chamber releases of simulants while test data are achieved from both chamber and open air releases. For each solution, the classifier is trained with all together over 14000 samples representing the six different classes of spectra. Furthermore, the trained classifiers are tested on approximately 60 000 samples of test data. All test data contain a number of background samples. Therefore, we do not need a separate test data set for background. A detailed overview of training data and test data is given in Table 1.

Table 1: Number of time samples for training data and test data

Simulant	Training data	Test data
BT	3516	608
BG	1001	13971
OV	2694	17589
Background	2499	-
MS2	2550	17430
EH	2354	11761
Sum	***14614***	***61359***

The parameters used in the genetic algorithm are shown in Table 2. Several runs with different parameters have been performed, resulting in the selected set. The large amount of decision variables involves a large population size so that the random initial population has solutions spread across large parts of the decision variable space. The selection method and low number of elites are chosen to prevent too quick convergence which could lead to local optima. Scattered crossover involves selecting bits randomly from parent chromosomes and the crossover rate involves 80 % of the remaining slots in the next population to be offspring of crossover (after the five elites are copied). The remaining slots in the next population are offspring of mutation where the mutation rate gives the probability of every single bit in a chromosome to be mutated.

Table 2: Evolutionary parameters

Operator	Value/Rate/Method
Cross-over	Scattered
Cross-over rate	0,8
Mutation	Bitwise
Mutation rate	1/chromosomelength
Selection	Rank fitness selection
Elites	5
Population size	1000
Generations	40

All data are normalized with the Euclidean norm, shown in Figure 8 and Figure 9 representing training data and test data for BT, respectively.

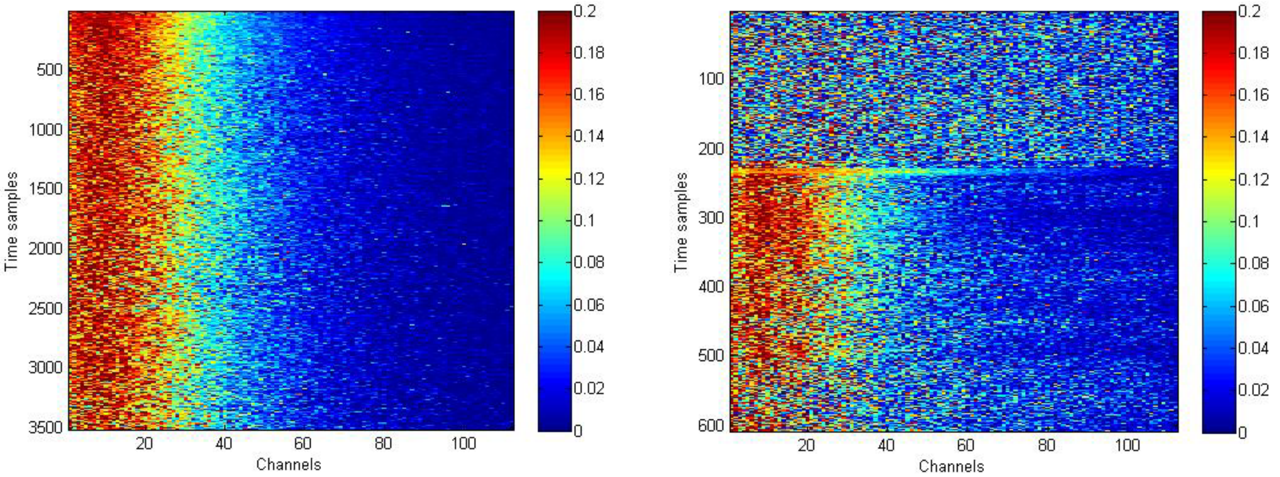


Figure 8: Normalized training data for the simulant BT. The color bar indicates signal strength.

Figure 9: Normalized test data for the simulant BT together with background samples at the top. The color bar indicates signal strength

3. RESULTS

The results show that feature optimization with GAs gives significantly improved classification error rate, from 40 % to only 7 %. The iterative improvement provided by the GA is shown in Figure 10, where the blue points are the best solutions for each generation and the green points are mean fitness of the entire population of solutions. The improvement of the best solution for each generation indicates the finding of a potentially optimal solution already at generation 27 with a classification error of 7 %. The decreasing gap between each generation's best fitness and mean fitness shows continued improvement even after generation 27 and is an indication of convergence for the set of solutions.

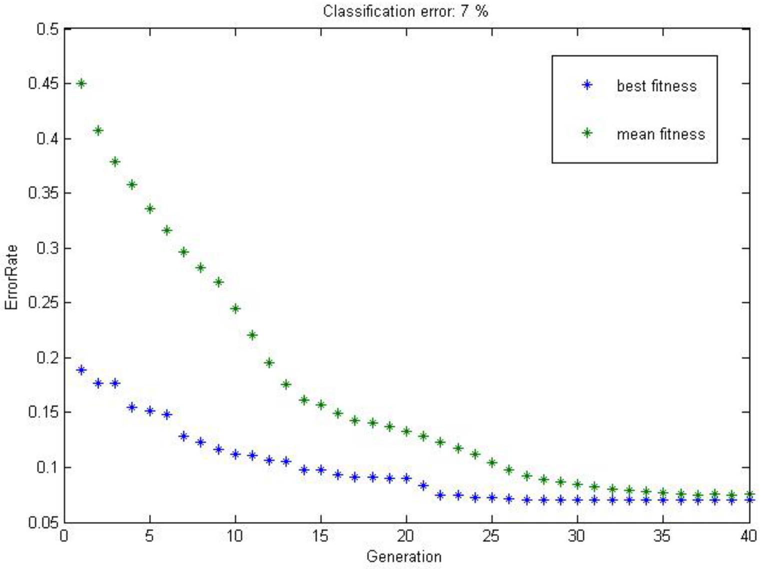


Figure 10: Optimization of classification error rate with genetic algorithms

Figure 11 shows the optimal solution of channel selection with GA, visualized by the gray bars. The solution shows that channels and bound intervals are selected from frequencies spread across the entire spectrum. Nevertheless, we make note of the algorithm's special points of interest, being the most discriminating frequency ranges. These are the ranges around the peak of the amplitude, after the cross point section around channel 25, and the interval from channel 35 to 55. The optimal solution integrates the signal over 56 time samples, near maximal size for smoothing the signal.

Figure 12 visualizes which of the selected channels that are bound in addition to the selected channels. Here, the white bars are the single selected channels (at full resolution) and the white bars followed by gray bars are the bounded intervals (reduced resolution). The bars show that the bound channels are the channels located at the peak, after the cross points and in the mid range frequencies, a possible indication of the significance of integrating the signal where the spectra have more or less equal derivative and selecting characteristic channels solely at frequencies surrounding cross points, preserving its discriminative content.

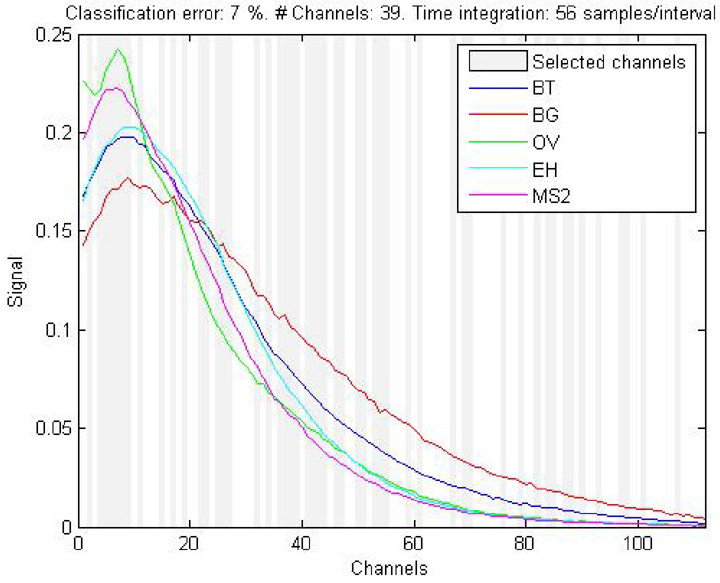


Figure 11: Visualizing channel selection of the optimal solution. The gray bars are the selected channels.

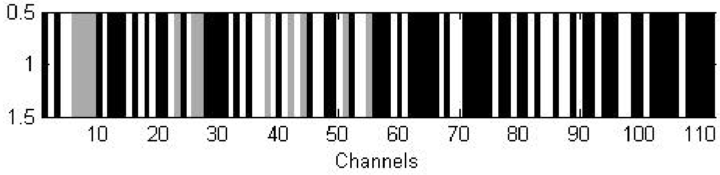


Figure 12: The white bars show selected channels and the white bars followed by gray bars show bound intervals.

The benchmark results are shown in Figure 13, demonstrating advantage for the methods handling solutions of high flexibility (blue), and low performance for methods of more restricted solutions (green). The figure shows performance of the primitive feature selection method *forward feature selection (*FFS*)* (keep adding the best feature until no improvement), FLD*,* PCA and SVM in combination with the Bayesian linear classifier, meaning the methods are used to transform the data while classification is performed with the Bayesian linear classifier*.* For these benchmark methods, the training data and test data are preconditioned because these methods do not allow for signal processing. For this reason several parameters were tested and the most efficient processing in terms of low classification error was time integration of 10 time samples per interval and channel binding over 20 uniformly spread intervals covering the entire fluorescence spectrum.

Random search is the best solution of 40 000 randomly generated solutions of the exact same coding for channel selection, channel binding and time integration as the solutions in the genetic algorithm. Compared to FLD as the best classical dimensionality reduction method, the solution found with the genetic algorithm is more than three times more effective in terms of classification error, and compared to random search the same performance is more than twice as effective.

SVM is performed using a radial basis kernel. A plausible reason for the low performance of this classifier is that SVM is only directly applicable for 2-categorial problems. Therefore, the classifier has to reduce the multi-class problem to several binary problems and compare the outcome.

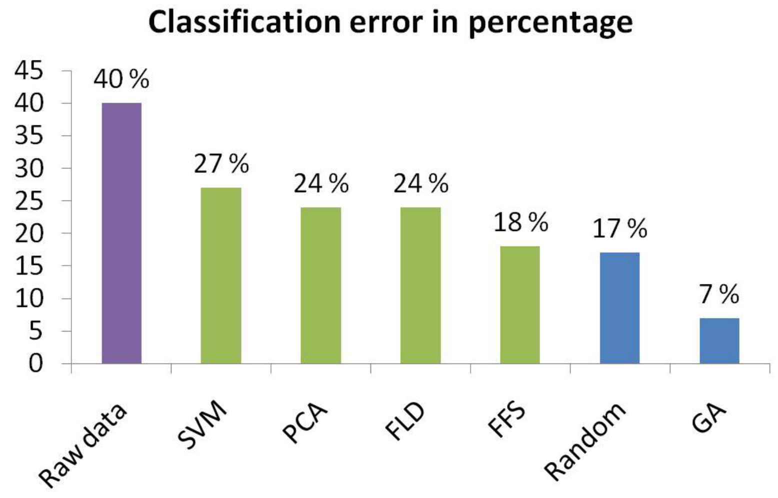


Figure 13: Relative performance of several data transformations combined with the linear Bayesian normal classifier.

4. CONCLUSION

The results show that simultaneous optimization of feature extraction and feature selection employing genetic algorithms gives a significant improvement in performance of the biological aerosol classification system and outperforms all benchmark methods tested. To our knowledge, this is a novel application of feature optimization using evolutionary algorithms in the field of biological warfare standoff detection. Compared to the raw data classification error of 40%, the error is reduced to 7% using the genetic algorithm.

The advantage of GA as an optimization method of finding an optimal data transformation is partly due to the flexibility when coding the decision variable space of the problem. This claim is supported by the good performance of random search having same solution flexibility as the GA and being more accurate than all other benchmark methods (next after GA). This may be an indication that the optimization problem of selecting features combined with feature extraction for spectral data classification is indeed combinatorial with non-independent decision variables, a condition making it difficult for more established optimization methods to handle.

REFERENCES

[1] Lim, D. V., Simpson, J. M., Kearns, E. A. and Kramer, M. F., "Current and developing technologies for monitoring agents of bioterrorism and biowarfare", Clinical Microbiological Reviews 18, 583-607 (2005).

[2] Buteau, S., Lahaie, P., Rowsell, S., Rustad, G., Baxter, K., Castle, M., Foot, V., Vanderbeek, R. and Warren, R., "Final report for task group (RTG-55) on laser based stand-off detection of biological agents", NATO RTO SET-098, NATO, Brussels (2008).

[3] Ho, N., Emond, F., Babin, F., Healy, D., Simard, J. R., Buteau, S. and McFee, J. E., "Short Range LIDAR for Bioagent Detection and Classification," Proc. SPIE 7665, 76650A (2010).

[4] Hopkins, R. J., Barrington, S. J., Castle, M., Baxter, K. L., Felton, N. V., Jones, J., Griffiths, C., Foot, V. and Risbey, K., "UV-LIF LIDAR for Standoff BW Aerosol Detection", Proc. SPIE 7484, 748409 (2009).

[5] Farsund, Ø., Rustad, G., Kasen, I. and Haavardsholm, T.V., "Required Spectral Resolution for Bioaerosol Detection Algorithms Using Standoff Laser-Induced Fluorescence Measurements", IEEE Sensors Journal 10, 655 - 661 (2010).

[6] Rustad, G. and Farsund, Ø., "Standoff detection of biological aerosols by UV-laser induced fluorescence", FFI-rapport 2008/02025, FFI, Kjeller, Norway (2008).

[7] Duda, R. O. , Hart, P. E. and Stork, D. G., [Pattern Classification], John Wiley & Sons (2001).

[8] Villmann, T., Merenyi, E. and Seiffert, U., "Machine Learning Approaches and Pattern Recognition for Spectral Data", Proc. 16th European Symposium on Artificial Neural Networks, ESANN'2008, pp 433-444 (2008).

[9] Mitchell, M., [An introduction to genetic algorithms], MIT press (1998).

[10] Goldberg, D. E., [Genetic algorithms in search, optimization & machine learning], Addison Wesley (1989).