Learning with a Nonexhaustive Training Dataset A Case Study: Detection of Bacteria Cultures using Optical-Scattering Technology

Murat Dundar
Computer & Information Science Department
IUPUI
723 W. Michigan St.
Indianapolis, IN 46202
dundar@cs.iupui.edu

E. Daniel Hirleman
School of Mechanical Engineering
Purdue University
585 Purdue Mall
West Lafayette, IN 47907
hirleman@purdue.edu

Arun K. Bhunia
Department of Food Science
Purdue University
1160 Food Science Building
West Lafayette, IN 47907
bhunia@purdue.edu

J. Paul Robinson
Bindley Bioscience Center
Purdue University
1203 West State Street
West Lafayette, IN 47907
jpr@cyto.purdue.edu

Bartek Rajwa
Bindley Bioscience Center
Purdue University
1203 West State Street
West Lafayette, IN 47907
brajwa@purdue.edu

ABSTRACT

For a training dataset with a nonexhaustive list of classes, i.e. some classes are not yet known and hence are not represented, the resulting learning problem is ill-defined. In this case a sample from a missing class is incorrectly classified to one of the existing classes. For some applications the cost of misclassifying a sample could be negligible. However, the significance of this problem can better be acknowledged when the potentially undesirable consequences of incorrectly classifying a food pathogen as a nonpathogen are considered.

Our research is directed towards the real-time detection of food pathogens using optical-scattering technology. Bacterial colonies consisting of the progeny of a single parent cell scatter light at 635 nm to produce unique forward-scatter signatures. These spectral signatures contain descriptive characteristics of bacterial colonies, which can be used to identify bacteria cultures in real time. One bottleneck that remains to be addressed is the nonexhaustive nature of the training library. It is very difficult if not impractical to collect samples from all possible bacteria colonies and construct a digital library with an exhaustive set of scatter signatures.

This report deals with the real-time detection of samples from a missing class and the associated problem of learning with a nonexhaustive training dataset. Our proposed method assumes a common prior for the set of all classes, known and missing. The parameters of the prior are estimated from the samples of the known classes. This prior is then used to generate a large number of samples to simulate the space of missing classes. Finally a Bayesian maximum likelihood classifier is implemented using samples from real as well as simulated classes. Experiments performed with samples collected for 28 bacteria subclasses favor the proposed approach over the state of the art.

The dataset and our implementation of the proposed approach is available on the web via the link:

http://www.cs.iupui.edu/∼dundar/kdd2009.htm

WOODSTOCK '09 Paris, France

Categories and Subject Descriptors

I.5.2 [**Pattern Recognition**]: Design Methodology—*classifier design and evaluation*

Keywords

nonexhaustive training data, bayes classifier, support estimation, support vector domain description, anomaly detection

1. INTRODUCTION

The goal of statistical learning is to build robust models that, when deployed in a real-life application, should generalize well to yet unseen examples of the sample population. Among other factors that influence the generalizability of a learning algorithm, the quality of the training dataset is perhaps the most critical. Although a detailed discussion of what makes a training dataset high quality goes beyond the scope of this study, a *representative* training dataset is essential to the realization of any supervised learning algorithm with high predictive accuracy.

There is not a well-defined definition of what makes a training dataset representative, but two things to consider are: is the list of classes (of informational value) complete, i.e. *exhaustive*? If yes, are there sufficiently large number of samples available from each class? It is very difficult to come up with a clear-cut answer to the second question, as how many samples considered sufficient for each class

depends on many factors: the dimensionality of the data, the type of model being used (discriminative or a generative), and the number of parameters to estimate with the data, to name few. Besides, this would not matter much if the list of classes is not exhaustive. In other words, if the existing set of classes is incomplete, i.e., misses one or more classes of informational value, no matter how many samples we have for the existing classes the training dataset would still have to be considered unrepresentative.

The easiest way to deal with this problem, as most traditional supervised algorithms do, is to ignore it. When this direction is taken, a sample of a class that is not represented in the training dataset would be incorrectly classified to one of the classes available in the training dataset. This could be a reasonable strategy for some domains where the cost of misclassifying a sample is negligible. However, within the framework of our current research, one could better appreciate the criticality of a more rigorous approach, considering the potentially unfortunate consequences of incorrectly classifying a pathogen as nonpathogen.

Learning with a nonexhaustive training dataset is an ill-defined problem, and to our knowledge there are not many studies in the machine-learning literature that explicitly address this issue. One recent work tackles this problem for the sake of identifying samples of land-cover types that are not readily known to exist in the remote sensing imagery of an area and hence are not represented in the training data [8]. The authors consider the well-known support vector domain description technique [11] to deal with a nonexhaustive training dataset and discusses the performance of this approach in comparison to density-based models.

One area of machine learning that has drawn much attention lately is anomaly detection. Both anomaly detection and the current problem of learning with a nonexhaustive training set aim to detect samples that are not represented in the training data, and in that regard they can be considered similar. However, there is a well-defined line between the two. In the dictionary an anomaly is defined as something peculiar, irregular, abnormal, or difficult to classify. So anomalies are outliers and they could be as different from each other as they are from normal cases [12]. More specifically, anomalies do not necessarily have informational value and it is very difficult if not impractical to model them. On the other hand, samples originating from a missing class have informational value and just like any class available in the training set they could be modeled, were they known during training.

Most of the early work in anomaly detection, centered around support estimation and density-based models, can also be applied to the current problem of learning with a nonexhaustive training set. In addition to these, the traditional supervised classification algorithms can be modified to accommodate for learning with a nonexhaustive training set by redefining the decision function to include *an undecisive region*. When tested by a classifier, if a sample falls onto this region, it is considered to originate from an unrepresented/missing class.

After reviewing the state of the art concerning these ideas, we propose a Bayesian approach based on the assumption that all classes are distributed according to a Gaussian distribution with a common covariance matrix. In this approach, we first define a hyperprior over the mean vectors of class distributions and estimate its parameters with samples from known classes. Then, we use this hyperprior to simulate the space of unknown classes. The final classifier is implemented using real as well simulated data and a new sample is rejected, i.e. the sample originates from a class not represented in the training set, when the likelihood is maximized for one of the simulated classes, otherwise the sample is accepted, i.e. the sample originates from one of the classes in the training set.

The rest of the paper is organized as follows. In Section 2 we present the detection of bacteria cultures using optical-scattering technology as a case study. Section 3 reviews and discusses the early work performed mainly in the area of anomaly detection. We present the details of the proposed approach in Section 4. Several experiments are performed to compare the state of the art and the proposed approach in Section 5. Finally, we conclude with a brief analysis of our results and by providing future research directions.

2. DETECTION OF BACTERIA

2.1 Optical recognition of bacterial colonies

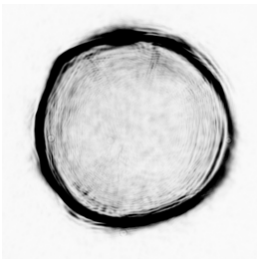
Traditional bacteria recognition methods based on antibodies or genetic matching are labor intensive, time consuming, and involve multiple steps. Moreover, samples are destroyed by these type of tests and are thus unavailable for further confirmatory assessment. The tests rely on the use of specific reporter molecules such as antibodies or nucleic-acid probes coupled with fluorophores or enzymes, thus limiting their broad application for multipathogen detection, or detection of unknown pathogens.

Light scattering is a fundamental optical process whereby electromagnetic waves deviate from a rectilinear path as a result of non-uniformities in the medium that they traverse. Light-scattering technology has been used before to interrogate bacterial cells in suspension [15, 16], as well as in flow [10, 9]. The scope of this approach was very narrow and only a limited number of bacterial species could be detected successfully. Recently, we have discovered that interrogation of bacterial cultures on the surface of agar in a semi-solid state could provide a possible differentiation via distinctive forward-scattering patterns [4, 1, 2]

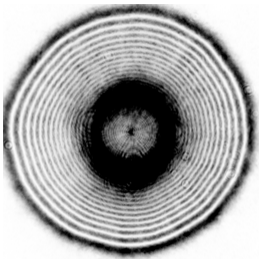
The BARDOT (BActeria Rapid Detection using Optical scattering Technology) system uses a laser (635 nm) to illuminate individual colonies and create a forward-scatter signature that is collected and subsequently analyzed (See Figure 1). In our earlier works we successfully demonstrated that scattering properties of *Listeria, E. coli, Salmonella, Staphylococcus*, and *Vibrio* colonies can be used to differentiate the species occurring in food samples as well as those isolated from experimentally infected animals [3].

2.2 Classification of known bacteria species and strains

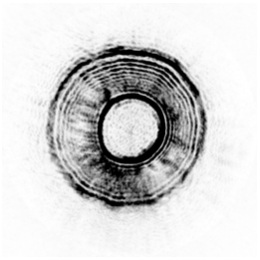
Forward-scattering patterns of bacterial colonies show descriptive characteristics. Zernike moment invariants and Haralick descriptors are used to capture these descriptive features and to construct a scatter-signature image library. We have previously reported examples of BARDOT application for classification and/or recognition of known (previously examined) bacterial species and strains. The classification techniques utilized for classification of BARDOT patterns included nonsupervised methods, partial least squares, linear discriminant analysis, neural networks, and support



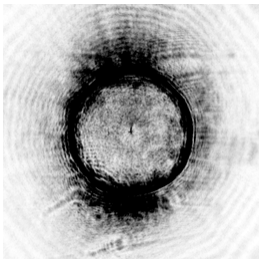
(a) *Salmonella* Typhimurium (Copenhagen)



(b) *Vibrio orientalis* CECT629



(c) *Listeria seeligeri* V45



(d) *Staphylococcus aureus* S-41

Figure 1: Representative examples of scatter pattern from four bacterial strains. The differences in the patterns can be seen with naked eye.

vector machines [4, 3]. The results obtained by classification varied widely and depended not simply on the classification technique used, but mostly on the type, quality, and number of features. However, no systematic study of methods performance has been attempted up to date. Therefore the comparison summarized in 5.2 is the first analysis of this kind reported for BARDOT data.

2.3 Non-exhaustive Nature of the Training Library

One of the main advantages of a label-free classification technology such as BARDOT is the fact that it can potentially recognize and classify bacterial species or strains for which there are no available antibodies or genetic markers. It is well known that some infectious agents are characterized by a high mutation rate, which can influence their pathogenicity. Therefore application of molecular-biology techniques for detection and classification are problematic owing to the dependence of these techniques on very specific reagents.

BARDOT relies only on the physical properties of the bacterial colonies and can acquire scatter patterns of colonies regardless of their genetic makeup; therefore it can be adapted to automatically recognize any new forms of the pathogens of interest, just by retraining the classifier using a new set of scatter patterns formed by colonies. However, the classification approach currently used with BARDOT relies on supervised training. In order to retrain the classifiers employed by BARDOT one would have to acquire a pure isolate of an unknown pathogen, measure the light-scattering characteristic of the colonies, and use the scattering features to define a new class. In practice, isolating a new class of bacteria, and subsequently obtaining a sufficient number of training samples, may turn out to be impractical or even impossible. The described situation also brings a procedural and logical conundrum: one would have to employ an independent testing procedure to find that a certain colony represent a new class, not present in the BARDOT library.

The outlined problem could be solved if the analysis strategy employed by BARDOT allowed for novelty detection prior to or simultaneous with the process of classification. If a novelty detection procedure were successfully implemented, BARDOT methodology would be capable of raising an alarm when a new class of pathogens was detected in tested samples. This crucial advancement in the treatment of BARDOT patterns coupled with the simplicity of BARDOT measurement would make this technique especially attractive for integration with highly automated systems operating for long periods without human supervision.

3. EARLY WORK

If all samples of the known classes are considered as positive, and all samples of the missing classes are considered as negative, then the problem can be cast as a one-class problem with multiple subclasses with samples available for all subclasses of the positive class and no samples for the negative class. In this setting, the missing samples of the negative class can be considered as anomalies/outliers. One promising approach that has been heavily explored in this domain is the support vector domain description technique (SVDD). This method seeks to fit a tight spherical boundary into the data to include most of the samples and reject possible outliers. We also have more traditional techniques based on density estimation to deal with one-class data. This group usually fits a Gaussian density into the data, and rejects a sample as an outlier if the likelihood is below some threshold. Finally, we would like to include a third group of techniques derived from discriminative classifiers into our discussion to evaluate the performance of these classifiers for identifying future samples.

3.1 Support Vector Domain Description

Support vector domain description (SVDD) is a kernel-based approach for estimating the support of a distribution, by fitting a spherical boundary around the target dataset [11]. To avoid accepting outliers the volume of the sphere is minimized. This problem is formulated as a constrained optimization problem with the data included in the problem in dot-product form. Hence the kernelized form of the problem is readily obtained [14]. Different kernel functions can be used to change the ball-shaped spherical boundary into more flexible boundaries, which allows for moderately precise control of the estimated support of the data. In what follows we review the basic theory for SVDD and discuss its use for learning with nonexhaustive training datasets as outlined in this study.

We assume that each sample for the target class is characterized by a feature vector $x_i \in R^d$, where d is the dimensionality of the feature space, $i = 1, \ldots, n$, and n is the number of training samples available for the target class. The dot product between two samples is denoted by $(x_i \cdot x_j)$. We want to minimize the radius of the sphere while making sure all data remains within the sphere. This is cast as the following constrained optimization problem.

$$\min_{(r,a)\in R^{d+1}} r^2 \quad \text{s.t} \quad \|x_i - a\|^2 \leq r^2 \tag{1}$$

where r is the radius of the sphere and $a \in R^d$ is its center. It is possible that the target dataset contains outliers, in which case extending the boundary of the sphere to include these outliers may unnecessarily increase the volume of the sphere. To deal with the outliers, the distance from each sample x_i to the center a is restricted to less than r^2 plus some positive number ξ_i. The objective function is modified to account for this change to penalize for larger values of ξ_i. With this changes the problem in (1) becomes

$$\begin{aligned} &\min_{(r,a)\in R^{d+1}} r^2 + C\sum_i^n \xi_i \\ &\text{s.t.} \quad \|x_i - a\|^2 \le r^2 + \xi_i, \;\; \xi_i \ge 0 \end{aligned} \tag{2}$$

The parameter C controls the trade-off between the volume of the sphere and the total error induced by the samples left outside the sphere. The Lagrange equation for this problem can be written as follows:

$$\begin{aligned} L(r,a,\xi) = r^2 + C\sum_i^n \xi_i \\ - \sum_{i=1}^n \alpha_i \left(r^2 + \xi_i - \|x_i - a\|^2\right) - \sum_i^n \beta_i \xi_i \end{aligned} \tag{3}$$

Setting the partial derivatives with respect to R, a, and ξ to zero gives the constraints:

$$\sum_i^n \alpha_i = 1, \quad a = \sum_i^n \alpha_i x_i, \quad 0 \le \alpha_i \le C$$

Resubstituting these constraints back into (3) yields the following dual form:

$$\begin{aligned} &\min_{\alpha\in R^n} \sum_i^n \sum_j^n \alpha_i \alpha_j \left(x_i \cdot x_j\right) - \sum_i^n \left(x_i \cdot x_i\right) \\ &\text{s.t.} \quad \sum_i^n \alpha_i = 1, \quad 0 \le \alpha_i \le C \end{aligned} \tag{4}$$

Note that the center of the sphere a is a linear combination of the samples x_i. Therefore only samples x_i with nonzero α_i are needed in expressing a. These samples are called support vectors.

A new sample z is accepted when its distance to the center is smaller than or equal to the radius:

$$\begin{aligned} \|z - a\|^2 = (z \cdot z) - 2\sum_i^n (z \cdot x_i) \\ + \sum_i^n \sum_j^n \alpha_i \alpha_j \left(x_i \cdot x_j\right) \le r^2 \end{aligned} \tag{5}$$

A sphere is certainly not flexible enough to fit data with complex shapes. Since the data x_i are in the dot-product form in both the training and testing phases, using the kernel concept initially introduced for support vector machines (SVM) [14], the dot products in (4) and (5) can be replaced by a predefined kernel function to obtain more flexible models. Replacing dot products with a predefined kernel corresponds to implicitly mapping the data from the input space into a new feature space and evaluating (4) for this new feature space. The spherical shape obtained in the feature space may correspond to arbitrary shapes in the input space. Throughout this study we used the Gaussian kernel, $K(x_i, x_j) = exp(-\|x_i - x_j\|^2 / \sigma^2)$. Adjusting the width of the kernel σ yields arbitrarily different shapes in the input space.

Our discussion of SVDD so far has been limited to one-class problems with only one subclass available. Recall that we are dealing with a one-class problem with samples available from all subclasses of the positive class, where positive class here is defined as the superset of all classes available in the training dataset. One way to approach this problem is to fit all class data with one boundary, but this will lose potentially useful subclass information. A more effective strategy would be to fit one spherical boundary to each subclasses of the positive class, i.e. to each of the classes in the training dataset, and reject a new sample as belonging to the negative class, i.e. a missing class, if the sample does not fall inside the support of any subclasses in the positive set.

3.2 Density Estimation

In this approach the data are distributed according to a probability density function $f(x|\theta)$. For a new sample z the likelihood is computed as $\ell(z) = f(z|\theta)$ and z is considered an outlier if $\ell(z)$ is less than some predefined threshold τ. The density function $f(x|\theta)$ is usually not known. The most popular choice for modeling an unknown distribution is a Gaussian model. The parameters $\theta = \{\mu, \Sigma\}$ are estimated using the training data and after removing the terms in the log-likelihood expression that do not depend on z, the decision function to accept or reject a new sample z is obtained as follows:

$$h(z) = \begin{cases} accept & \text{if } g(z) \le \tau \\ reject & \text{if } g(z) > \tau \end{cases} \tag{6}$$

where $g(z) = (z - \mu)^T \Sigma^{-1} (z - \mu)$.

For the current problem of learning with a nonexhaustive training dataset, we fit a Gaussian density for each class, ω_k, $k = \{1, \ldots, K\}$ in the training dataset and update the decision function in (6) as follows:

$$h(z) = \begin{cases} accept & \text{if } \min_k \{g_k(z)\} \le \tau \\ reject & \text{if } \min_k \{g_k(z)\} > \tau \end{cases} \tag{7}$$

where K is the total number of known classes.

3.3 Discriminant Functions with an Indecisive Region

Discriminative models of the form $f(x) = w^T x + w_0$ are used for the design of binary classifiers. The classifier coefficient w and the bias term w_0 are learned from the training data by optimizing an objective function usually consisting of two conflicting goals: minimizing the complexity of the classifier and minimizing the error committed on training samples. Support vector machine (SVM) [14], relevance vector machine (RVM) [13], linear Fisher's discriminant (LFD) [6], and logistic regression [7] are all members of this group. The binary decision function for a new sample z is expressed as

$$h(z) = \begin{cases} positive & \text{if } f(z) \ge 0 \\ negative & \text{if } f(z) < 0 \end{cases} \tag{8}$$

In this binary setting the range of the discriminant function $f(x)$ is divided into two regions, positive and negative.

A third region can be introduced to accommodate samples with discriminant values close to the decision boundary. Under this setting the decision function in (8) can be updated as

$$h(z) = \begin{cases} \textit{positive} & \text{if } f(z) \geq \epsilon \\ \textit{indecisive} & \text{if } -\epsilon \leq f(z) < \epsilon \\ \textit{negative} & \text{if } f(z) < \epsilon \end{cases} \quad (9)$$

where ϵ is a designated number.

For the multiclass problem, a one-against-one approach can be taken to design $K(K-1)/2$ binary classifiers, where K is the number of classes. A new sample, z, is classified by all $K(K-1)/2$ and after each classification the sample is assigned to either the current positive/negative class or labeled as indecisive using the decision function in (9). Then the total number of times z is assigned to each of the K classes is computed. If the maximum of these numbers is less than the total number of times a sample is labeled as indecisive, then the sample is rejected as belonging to a potentially missing class.

This concludes our review of the potentially useful methods from the literature to deal with a nonexhaustive training dataset. We will implement these techiques to obtain a baseline performance in Section 5 when evaluating the performance of the proposed work with the bacteria dataset.

4. PROPOSED APPROACH

A training dataset is nonexhaustive when one or more of the classes with informational value are not represented by any samples. When a classifier is trained with this dataset, a sample of a yet unseen class will be misclassified with probability one, making the corresponding learning problem ill-defined. A two stage design of the classifier can help alleviate this problem and make learning with a nonexhaustive training dataset a more reasonable goal to achieve. The first stage is a one-class classifier, which identifies whether the sample is a member of one of the classes in the training set or a member of a yet unseen/missing class. If the sample is confirmed as belonging to one of the classes in the training set, then the sample is fed to the second stage where it is classified to one of the existing classes. If the sample is confirmed as belonging to an unseen class at the first stage, an alert is raised and the sample is saved for follow-up analysis.

classifier of the second stage can be trained using any supervised classification algorithm. For an exhaustive training dataset we implemented four supervised classifiers from the literature for this task and summarized the results in Table 5.2. What makes this problem challenging is the design of the first stage classifier, where we only have samples from the known classes, i.e. the positive class. In what follows we present a Bayesian approach to learning with a nonexhaustive training dataset.

4.1 Bayesian Approach to Modeling Missing Classes

We start with the notation first. The symbols Ω, Ψ, and Γ denotes the set of *all*, *known* and *missing* classes respectively with $\Omega = \Psi \cup \Gamma$; A, K and M are their corresponding cardinalities with $A = K + M$. The conditional distribution of a class $\omega_i \in \Omega$ is defined by the density function $f_i(x|\theta_i)$ with θ_i being distributed according to a common prior $\pi(\theta|\beta)$ shared across all classes. Let $x_{ij} \in R^d$, where d is the dimensionality of the feature space, $j = \{1, \ldots, n_i\}$, n_i be the number of training samples for class ω_i, $\omega_i \in \Psi$. For notational simplicity all samples belonging to class $\omega_i \in \Psi$ are denoted in the matrix form as $X_i = [x_{i1} \ldots x_{in_j}]$.

The decision that minimizes the Bayes risk under a 0/1 loss-function assumption assigns a new sample z to the class with the highest posterior probability. More specifically,

$$z \in \omega_i^* \quad \text{s.t.} \quad p_i^*(\theta_i|z) = \max_i \{p_i(\theta_i|z)\} \quad (10)$$

where $i = \{1, \ldots, A\}$. The classifier obtained by evaluating this decision rule is known as maximum a posteriori classifier (MAP). Using Bayes' rule the above decision rule can be rewritten as follows:

$$z \in \omega_i^* \quad \text{s.t.} \quad p_i^*(\theta_i|z) = \max_i \left\{ \frac{f_i(z|\theta_i)\pi_i(\theta_i|\beta)}{p(z)} \right\} \quad (11)$$

where $f_i(z|\theta_k)$ is the *likelihood* of z, $\pi(\theta_i|\beta)$ is the *prior*, and $p(z)$ is the *evidence*. The evidence $p(z)$ is same for all classes and hence can be removed from the above formulation.

The class distributions $f_i(x|\theta_i)$ are not known. Before we estimate $f_i(x|\theta_i)$ for $\omega_i \in \Psi$, i.e. known classes, using the training dataset, we make some general assumptions that we can also carry to $f_i(x|\theta_i)$ for $\omega_i \in \Gamma$. The most common and also an effective way to deal with data of unknown nature is to assume normal distributions for all classes, $\omega_i \sim N(\mu_i, \Sigma_i)$, $\theta_i = \{\mu_i, \Sigma_i\}$. Here we go one step further and assume that all classes share the same covariance matrix, $\omega_i \sim N(\mu_i, \Sigma)$, $\forall \omega_i \in \Omega$, where μ_i and Σ are estimated using the training samples from the known classes as follows:

$$\bar{\Sigma} = \frac{1}{K} \sum_{i=1}^{K} \frac{1}{n_i - 1} \left(X_i - \bar{\mu}_i e_{n_i}^T \right) \left(X_i - \bar{\mu}_i e_{n_i}^T \right)^T \quad (12)$$

$$\bar{\mu}_i = \frac{1}{n_i} X_i e_{n_i} \quad (13)$$

Here e_{n_i} is a vector of ones with the size of the vector equivalent to the number of training samples in class ω_i and the bar sign on $\bar{\mu}_i$ and $\bar{\Sigma}$ indicates the estimated values.

Since Σ is already estimated from the data and thus known, the prior $\pi(\theta|\beta)$ is only defined over the mean vectors μ_i. Next, we assume a Gaussian prior over μ_i with mean m and covariance matrix S, i.e. $\mu_i \sim N(m, S)$, $\beta = \{m, S\}$. Both m and S are estimated from the samples $\bar{\mu}_i$, $\omega_i \in \Psi$, which were in turn estimated in (13). Once $\bar{m}$ and $\bar{S}$ are obtained we can use the prior distribution $\pi(\mu|\bar{m}, \bar{S})$ of the mean vectors to simulate the space of missing classes, i.e. Γ. More specifically, using this distribution we generate a very large number of samples with each sample corresponding to the mean vector of a supposedly missing class. To differentiate between the mean vectors estimated from the data and the mean vectors simulated from the prior, we use $\tilde{\mu}_i$ for the latter. Similarly, we use $\tilde{\Gamma}$ to differentiate the set of simulated classes from the set of missing classes Γ. The Bayes decision function in (11) is implemented using real as well as simulated classes and a new sample z is classified according to the following decision function

$$h(z) = \begin{cases} \textit{accept} & \text{if } \omega_i^* \in \Psi \\ \textit{reject} & \text{if } \omega_i^* \in \tilde{\Gamma} \end{cases} \quad (14)$$

That is, if the class that maximizes the posterior is a simulated class, then z is rejected as belonging to a potentially missing class.

4.2 Implementation Challenges

We are estimating two sets of parameters. First, $\{\mu_i, \Sigma\}$, $\omega_i \in \Psi$ using $x_{ij} \in R^d$ with n_i samples and then $\{m, S\}$ using $\bar{\mu}_i \in R^d$ with K samples, where K is the number of known classes. For large d, feature selection will help alleviate the problem of the curse of dimensionality [5] and will significantly improve parameter estimation for the first set of parameters. However, the estimation of m and S will continue to suffer from the limited number of samples K. For $d > K$ the inverse of $\bar{S}$ does not exist. Even though the inverse of S is not required to generate samples to simulate the mean vectors of the missing classes, an ill-conditioned matrix $\bar{S}$ will prevent us from evaluating $\pi_i(\theta_i|\beta)$ in (11). One possible solution would be to use simpler covariance models involving the trace or the diagonal forms of $\bar{S}$, but this will eliminate useful correlation information between features. Therefore we restrict our model by assuming that all classes are a priori likely given $\bar{S}$ and $\bar{m}$ and drop $\pi_i(\theta_i|\beta)$ in (11). The maximum a posteriori classifier (MAP) in (11) becomes a maximum likelihood (ML) with this change and the decision function in (11) becomes

$$z \in \omega_i^* \quad \text{s.t.} \quad p_i^*(\theta_i|z) = \max_i \{f_i(z|\theta_i)\} \tag{15}$$

4.3 Algorithm for Learning with Real and Simulated Classes

ALGORITHM 4.1. **Algorithm for Training**

(i) For each class in the training dataset, i.e. $\omega_i \in \Gamma$ estimate Σ and μ_i using equations (12) and (13).

(ii) Then estimate m and S for $\pi(\mu|m, S)$ using estimates of the mean vectors $\bar{\mu}_i$, $i = \{1, \ldots, K\}$.

(iii) Generate M samples from the prior distribution $\pi(\mu|\bar{m}, \bar{S})$.

ALGORITHM 4.2. **Algorithm for Detection**

(i) Evaluate $\ell^*_{real} = \max_i \left\{ f_i \left(z|\bar{\mu}_i, \bar{\Sigma} \right) \right\}$ for $\omega_i \in \Psi$, $i = \{1, \ldots, K\}$.

(ii) Evaluate$\ell^*_{sim} = \max_i \left\{ \tilde{f}_i \left(z|\tilde{\mu}_i, \bar{\Sigma} \right) \right\}$ for $\omega_i \in \tilde{\Gamma}$, $i = \{1, \ldots, M\}$.

(iii) If $\ell^*_{sim} > \ell^*_{real}$ reject z, i.e. z originates from one of the missing classes; otherwise accept z, i.e. z originates from one of the known classes.

5. EXPERIMENTS

5.1 Dataset Used

A total of 28 subclasses from five different bacteria classes were considered in this study. The classes available are *Escherichia coli, Listeria, Salmonella, Staphylococcus* and *Vibrio*. Table 1 shows the list of subclasses and classes considered in this study together with the number of samples collected for each subclass using the BARDOT system detailed in Section 2.

Class	Subclass	Samples
E. coli	O157:H7 01	64
	O25:K98:NM ETEC	67
	O78:H11 ETEC	58
	K12 ATCC 29425	65
	O157:H7 6458	87
	O157:H7 G5295	68
Listeria spp.	*L. monocytogenes* 7644 (1/2c)	91
	L. welshimeri 35897	47
	L. innocua F4248	59
	L. ivanovii 19119	81
	L. monocytogenes 19118 (4e)	94
	L. monocytogenes V7 (1/2a)	98
Salmonella spp.	S. Enteritidis PT28	90
	S. Enteritidis 13096	89
	S. Typhimurium (Copenhagen)	95
	S. Tennessee 825-94	78
Staphylococcus spp.	*S. aureus* S-41	67
	S. hyicus T6346	69
	S. aureus PS103	50
	S. aureus 13301	46
	S. epidermidis PS302	31
	S. *epidermidis* 35547	45
Vibrio spp.	*V. alginolyticus* CECT521	88
	V. campbellii CECT523	71
	V. cincinnatiensis CECT4216	89
	V. hollisae CECT5069	79
	V. orientalis CECT629	96
	V. parahaemolyticus CECT511	92
	Total	2054

Table 1: The 28 subclasses from five genera (classes) considered in this study. The last column lists the number of samples collected for each strain using the BARDOT system.

5.2 Experiment 1: Classification of Known Bacteria Subclasses

In this experiment we design and implement several state-of-the-art classifiers for the task of classifying already known bacteria subclasses. The objective here is to train a 28-class classifier to classify new samples at the subclass level, i.e. when a yet unseen sample from one of the 28 subclasses emerges, the classifier accurately assigns the sample to its subclass of origin.

5.2.1 Classifier Validation

Classifiers are validated using a 10-fold cross-validation approach as follows. First, we randomly shuffle the training data and split the data into 10 groups using stratified sampling to make sure that each of the ten folds has roughly an equal number of samples from each subclass. Then at each stage one fold is left out as testing data and a classifier is trained with the remaining 9 groups. Next, the test data are classified with this classifier and class labels assigned to each sample are recorded. Once all 10 groups are tested, the estimated labels are compared with the actual labels and the classifier accuracy is obtained. This process is repeated five times and the classifier accuracies averaged over ten runs are recorded together with the standard deviations as the 10-fold cross-validation performance of the classifier.

5.2.2 Classification Methods Considered

The classification methods considered are support vector machines (SVM) [14], linear Fisher's discriminant (LFD) [6], maximum likelihood classifier implemented with Gaussian distributions (ML), and support vector domain description (SVDD) [11]. The first two (SVM, FLD) uses discriminative

models, SVDD is a one-class classifier and ML Classifier is density-based approach. In what follows we briefly discuss the classifier design and parameter selection for each algorithm.

Support vector machine: SVM is a binary classifier that optimizes a hyperplane of the form $f(x) = w^T x + w_0$ to maximize the margin between the two classes. Multi-class design with SVM can be achieved using either the *one-against-one* or the *one-against-all* scheme. Here we adopted the *one-against-one* approach. For the 28-class problem $(28 \times 27)/2 = 378$ unique pairs of classes exist. Therefore we trained 378 binary classifiers during a single run of the training phase. For each binary classification problem one of the classes is assumed positive and the other one negative. A new test sample z is classified by all 378 classifiers and after each classifier z is assigned to the class denoted positive if $f(z) > 0$, otherwise it is assigned to the class denoted negative. At the end, the number of times z is assigned to each class is counted and z is permanently assigned to the class with the highest number of hits.

The tuning parameters for SVM are the type of the kernel function, its parameter(s) and the cost C of misclassifying a sample. Here we used the popular Gaussian kernel function $K(x_i, x_j) = \exp\left(-\|x_i - x_j\|^2 / \sigma^2\right)$. The width of the kernel function σ and C are chosen jointly to optimize the 10-fold cross-validation performance of the classifier. We considered five different values for $\sigma = 0.5, 0.75, 1, 1.25, 1.5$ and three different values for $C = 10, 100, 1000$. The pair of values with the best 10-fold cross-validation performance is found as $\sigma = 1, C = 100$.

As a standard data-preprocessing step we normalize each feature to between -1 and 1 and implement the classifier with the normalized data.

Linear Fisher's discriminant: LFD is a binary classifier that projects the high-dimensional data onto a line and performs classification in this one-dimensional space. The projection, w, is chosen such that the ratio of the scatter matrices (between and within classes) is maximized. Like SVM, the classifier function can be expressed as a hyperplane of the form $f(x) = w^T x + w_0$. For the multiclass implementation of LFD we follow the same approach outlined above for SVM.

The only design parameter for LFD is the regularization constant. Regularization of the classifier coefficients w is achieved by adding a small scalar times the identity matrix to the within-class scatter matrix, i.e. $\bar{S}_w = S_w + \lambda I$, where I is the identity matrix and λ is the tuning parameter used to regularize the classifier. Here $\lambda = 5 \times 10^{-4}$ is chosen from the set of $\lambda = \{\ 5 \times 10^{-6}, 10^{-6}, 5 \times 10^{-5}, 10^{-5}, 5 \times 10^{-4}, 10^{-4}, 5 \times 10^{-3}, 10^{-3}, 5 \times 10^{-2}, 10^{-2}\ \}$ as the value that optimizes the 10-fold cross-validation performance.

Support vector domain description: The SVDD approach is used to estimate the support of each of the 28 subclass distributions. The details of the SVDD approach for estimating the support of a distribution are presented in Section 2. Here SVDD is used as a supervised classifier. A new sample z is assigned to the class whose center a is closest to z. The tuning parameters for this approach are the width σ of the Gaussian kernel function and the cost C of rejecting a sample as an outlier. As with the SVM classifier above, σ, and C are chosen jointly to optimize the 10-fold cross validation performance of the classifier. We considered five different values for σ={0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 1.5, 2, 5} and three different values for C={1, 10, 100}. The pair of values with the best 10-fold cross-validation performance is found to be ($\sigma = .7,\ C = 10$).

Bayes maximum likelihood classifier: This approach assumes Gaussian distributions with a common covariance matrix for each subclass. The covariance matrix and mean vectors are estimated using equations (12) and (13).

The dimensionality of the dataset is d=240. We are estimating $240^2 = 57600$ parameters for the common covariance matrix and $28 \times 240 = 6720$ parameters for the mean vectors using a total of 2054 training samples. To avoid *the curse of dimensionality* [5], a sequential forward-backward propagation feature-selection algorithm is implemented with this classifier in a wrapper framework. This approach starts with an empty subset and performs a forward selection succeeded by a backward attempt to eliminate a feature from the subset. During each iteration of the forward selection exactly one feature is added to the feature subset. To determine which feature to add, the algorithm tentatively adds to the candidate feature subset one feature that is not already selected and tests the 10-fold cross-validation performance of the classifier built on the tentative feature subset. The feature that results in the highest classification accuracy is added to the feature subset. During each iteration of the backward elimination, the algorithm attempts to eliminate the feature whose elimination results in the highest classification accuracy. This process continues until no improvement is gained.

We run this feature-selection algorithm 10 times and record the features selected after each run. At the end of the ten runs we sort the features according the total number of times each feature is selected descending order and choose the top 80 features for the final classifier. A new sample z is classified using the decision function in (15).

5.2.3 Results and Analysis

The 10-fold cross-validation performance obtained by each of the four classifiers is shown in Table 2. Results show that LFD and SVM are equally competitive, with LFD yielding slightly better results. The maximum likelihood classifier looks promising but SVDD is not competitive as a supervised classifier.

	SVM	LFD	ML	SVDD
Acc. (%)	91.7	92.3	89.2	72.8
Std.	0.2	0.3	0.2	0.6

Table 2: The classifier accuracies achieved by the four classifiers averaged over ten runs.

5.3 Experiment 2: Learning with a Nonexhaustive Set of Bacteria Subclasses

In this experiment we implement classifiers to detect samples of known bacteria subclasses, i.e. subclasses represented in the training data, in order to separate them from samples belonging to unknown classes.

5.3.1 Performance Evaluation and Classifier Validation

In the previous experiment we assumed the training dataset to be exhaustive and implemented several classifiers to clas-

sify samples of the bacteria subclasses. That was mainly a classification task and as such we used classifier accuracy as a performance metric to evaluate classifiers. In this experiment we assume that the training dataset is nonexhaustive. The goal is now to design a classifier that accurately detects samples from the known classes and rejects samples of the unrecognoized classes. In this framework, classifiers can be more properly evaluated using sensitivity and specificity metrics. Here sensitivity is defined as the number of samples from known classes accepted by the classifier divided by the total number of samples from known classes, and specificity is defined as the number of samples from missing classes rejected by the classifier divided by the total number samples from missing classes. For each classification method multiple sensitivity and specificity values are obtained at different operating points to plot the receiver operating characteristic (ROC) curves. When comparing different classification methods, the method with the largest area under the curve is considered the best. The 10-fold cross-validation approach of the previous experiment is modified to conform it to the current task as follows.

First, we randomly shuffle the training data and split the data into 10 groups using stratified sampling to make sure that each of the ten folds has roughly an equal number of samples from each strain class. This step is the same as the one in the previous experiment. Then one group is left out as testing data and from the remaining 9 groups, samples of each of the 28 subclasses are removed, one subclass at a time, to create 28 different nonexhaustive training sets. This way all 28 subclasses are considered missing in turn, albeit only one at a time. Next, 28 different classifiers are trained, one for each of the 28 nonexhaustive training sets, and test samples are classified with these classifiers as *accept* or *reject*. Once all ten groups are covered this way, the sensitivity and specificity of the classification method is computed by averaging out the sensitivities and specificities achieved by the 28 individual classifiers. This process is repeated ten times and the sensitivity and specificity values averaged over ten runs are recorded together with the standard deviations as the 10-fold cross-validation performance of the classification method for detecting samples of the known bacteria subclasses.

5.3.2 Classification Methods Considered

The classification methods of Experiment 1 are considered for this experiment as well. Following the discussion in Sections 3 and 4, this time they are cast as one-class classifiers.

SVM and LFD: The binary decision rule used in Experiment 1 for SVM and LFD is now modified to include an indecisive range for $f(x)$ as in (9). Once a test sample z is classified by all $K(K-1)/2 = 378$ binary classifiers using $h(z)$ in (9), it is rejected if the total number of times it is denoted as indecisive outnumbers any individual class assignment. The same set of parameters optimized in Experiment 1 for SVM and LFD are used for this part. Multiple operating points on the ROC curve are obtained by varying ϵ from 0 to 0.2 in increments of 0.01.

SVDD: Previously a new sample z was assigned to the class whose center a is closest to z. This time a new sample z is accepted if it falls within the support of one of the bacteria subclasses; otherwise it is rejected. The parameter C was optimized as 10 in Experiment 1. The same value is used here. Multiple operating points on the ROC curve are obtained by considering the set of values σ={0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.5, 0.75, 1, 3, 5, 10} as the width of the kernel function. Each different value of σ corresponds to a different operating point on the ROC curve.

ML classifier with real and simulated bacteria subclasses: The same set of 80 features selected in Experiment 1 for the ML classifier is also used here. The classifier is trained following the steps in Algorithm 4.1. Since subclasses are considered missing, one at a time, as described in Section 5.3.1), m and S are estimated using $\bar{\mu}_i$ of the remaining 27 bacteria subclasses. Then the steps in Algorithm 4.2 are followed and a new sample z is accepted if the likelihood is maximized for one of the real subclasses and rejected if the likelihood is maximized for one of the simulated subclasses. Multiple operating points on the ROC curve are obtained by generating varying numbers of mean vectors with the prior in step 3 of Algorithm 4.1, i.e. values of M={5, 10, 50, 100, 500, 1000, 5000, 10000, 20000, 35000, 50000} are considered in this experiment.

5.3.3 Results and Analysis

The ROC curves obtained for classification methods are plotted in Figure 2 together with the corresponding error bars. In contrast to Experiment 1, the discriminative classifiers (SVM and LFD) performed very poorly in this experiment. The corresponding ROC curves are only slightly better than a random classifier. SVDD shows some promise but the ML algorithm is by far the best.

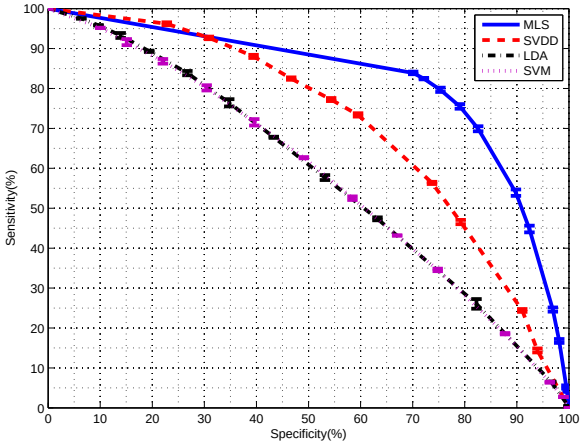


Figure 2: ROC curves obtained for the four classification methods considered in Experiment 2.

We ran the ML algorithm for a varying number of simulated subclasses. The specificity and sensitivity values at M=50,000 are 84% and 70% respectively. Generating specificities above 84% will require increasing M beyond 50,000, which will further increase the computational time.

The training takes place offline. Even though offline training time is not critical, we provide some numbers to serve as comparison between SVDD and ML algorithms. All five runs of the ML algorithm took little less than two days of computer time. The five runs for the SVDD took less than five hours All experiments are run on an Intel-based com-

puter (Intel Core 2 Duo T9300 2.50Ghz). The time taken for classification is negligible for both approaches.

We believe SVDD suffers mainly from independent modeling of each subclass. In the current framework the support for each subclass is estimated independent of other subclasses. As a result, the supports of some classes overlap with each other. A new sample frequently falls within the support of more than one subclasses at the same time. This negatively impacts the prediction accuracy of SVDD.

The number of samples as well as the number of subclasses available in the training dataset is critical to a robust implementation of the proposed ML technique. As we have more samples from each subclass, more reliable estimates of the mean vector and covariance matrix can be obtained for each subclass. In addition to improving the estimate of the conditional distribution for each subclass, this will more importantly help with the modeling of the prior. The parameters of the priors are estimated using the mean vectors estimated with each subclass data. However, this does not address the problem of limited sample size for the estimation of the prior mean and covariance matrix. In the current study these parameters are estimated using only 27 samples (one subclass is considered missing in turn at a time) in an 80-dimensional space.

6. CONCLUSIONS

In this study we proposed an approach for learning with a nonexhaustive training dataset. The technique is based on Bayesian modeling of subclasses with real and simulated data. The training dataset is used to estimate the conditional distributions of each subclass. A Gaussian distribution with a common covariance matrix is assumed for all subclasses. A Gaussian prior is defined on the mean vectors and the parameters of this distribution are estimated using the estimates of the mean vectors obtained for each class conditional distribution. A large number of samples from this prior are generated to simulate the set of missing subclasses. Finally, a Bayesian maximum-likelihood classifier is implemented using real and simulated data.

This research was mainly motivated by the need for real-time detection of bacteria cultures in food chains. We applied our technique to the dataset collected for this purpose, which was composed of samples from 28 bacteria subclasses. Even though the proposed approach is based on relatively strong assumptions, results with this dataset clearly favor the proposed approach over the state of the art, which is mainly centered around support vector domain description and well known discriminative classifiers.

Before this system can be deployed in real time to safely and accurately detect samples of potentially harmful bacteria, the performance of the classifier needs to improve. So far we have used only phenotype of the bacteria, characterized by different morphological properties captured by the light-scattering technology. Our ongoing research efforts are directed toward incorporating genotype as well as the functional characteristics of the bacteria colonies into this learning problem. This new information will provide significant additional evidence to distinguish between bacteria colonies, which we will incorporate into the learning problem to improve the prediction accuracy of the classifier.

7. REFERENCES

[1] E. Bae, P. P. Banada, K. Huff, A. K. Bhunia, J. P. Robinson, and E. D. Hirleman. Biophysical modeling of forward scattering from bacterial colonies using scalar diffraction theory. *Applied Optics*, 46(17):3639–48, June 2007.

[2] P. P. Banada, S. Guo, B. Bayraktar, E. Bae, B. Rajwa, J. P. Robinson, E. D. Hirleman, and A. K. Bhunia. Optical forward-scattering for detection of listeria monocytogenes and other listeria species. *Biosensors & Bioelectronics*, 22(8):1664–71, Mar. 2007.

[3] P. P. Banada, K. Huff, E. Bae, B. Rajwa, A. Aroonnual, B. Bayraktar, A. Adil, J. P. Robinson, E. D. Hirleman, and A. K. Bhunia. Label-free detection of multiple bacterial pathogens using light-scattering sensor. *Biosensors & Bioelectronics*, 24(6):1685–92, Feb. 2009.

[4] B. Bayraktar, P. P. Banada, E. D. Hirleman, A. K. Bhunia, J. P. Robinson, and B. Rajwa. Feature extraction from light-scatter patterns of listeria colonies for identification and classification. *Journal of Biomedical Optics*, 11(3):34006, 2006.

[5] R. E. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[6] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, San Diego, CA, 1990.

[7] T. H. J. Friedman and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, 38:337–374, 2000.

[8] J. Muñoz-Marí, L. Bruzzone, and G. Camps-Valls. A support vector domain description approach to supervised classification of remote sensing images. *IEEE Transaction on Geoscience and Remote Sensing*, 45(8):2683–2692, 2008.

[9] B. Rajwa, M. Venkatapathi, K. Ragheb, P. P. Banada, E. D. Hirleman, T. Lary, and J. P. Robinson. Automated classification of bacterial particles in flow by multiangle scatter measurement and support vector machine classifier. *Cytometry. Part A*, 73(4):369–79, Apr. 2008.

[10] H. B. Steen. Light scattering measurement in an arc lamp-based flow cytometer. *Cytometry*, 11(2):223–30, 1990.

[11] D. M. J. Tax and R. P. W. Duin. Support vector domain description. *Pattern Recognition Letters*, 20(11-13):1191–1199, 1999.

[12] J. Theiler and D. M. Cai. Resampling approach for anomaly detection in multispectral images, 2003.

[13] M. E. Tipping. The relevance vector machine. In S. Solla, T. Leen, and K.-R. Muller, editors, *Advances in Neural Information Processing Systems 12*, pages 652–658. MIT Press, Cambridge, MA, 2000.

[14] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

[15] P. J. Wyatt. Identification of bacteria by differential light scattering. *Nature*, 221(5187):1257–8, Mar. 1969.

[16] P. J. Wyatt and D. T. Phillips. Structure of single bacteria from light scattering. *Journal of Theoretical Biology*, 37(3):493–501, Dec. 1972.