A Generative Semi-Supervised Classifier for Datasets with Unknown Classes

Stefan Schrunner∗
Oslo University Hospital
Oslo, Norway
stefan.schrunner@fys.uio.no

Anja Zernig
KAI GmbH
Villach, Austria
anja.zernig@k-ai.at

Bernhard C. Geiger
Know-Center GmbH
Graz, Austria
bgeiger@know-center.at

Roman Kern
Know-Center GmbH
Graz, Austria
rkern@know-center.at

ABSTRACT
Classification has been tackled by a large number of algorithms, predominantly following a supervised learning setting. Surprisingly little research has been devoted to the problem setting where a dataset is only partially labeled, including even instances of entirely unlabeled classes. Algorithmic solutions that are suited for such problems are especially important in practical scenarios, where the labelling of data is prohibitively expensive, or the understanding of the data is lacking, including cases, where only a subset of the classes is known. We present a generative method to address the problem of semi-supervised classification with unknown classes, whereby we follow a Bayesian perspective. In detail, we apply a two-step procedure based on Bayesian classifiers and exploit information from both a small set of labeled data in combination with a larger set of unlabeled training data, allowing that the labeled dataset does not contain samples from all present classes. This represents a common practical application setup, where the labeled training set is not exhaustive. We show in a series of experiments that our approach outperforms state-of-the-art methods tackling similar semi-supervised learning problems. Since our approach yields a generative model, which aids the understanding of the data, it is particularly suited for practical applications.

KEYWORDS
semi-supervised learning, semi-supervised classification, unknown classes, Bayes classifier, S-EM algorithm, Gaussian mixture model

ACM Reference Format:
Stefan Schrunner, Bernhard C. Geiger, Anja Zernig, and Roman Kern. 2020. A Generative Semi-Supervised Classifier for Datasets with Unknown Classes. In The 55th ACM/SIGAPP Symposium on Applied Computing (SAC ’20), March 30-April 3, 2020, Brno, Czech Republic. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3341105.3373890

∗Stefan Schrunner contributed to this work during his employment at KAI GmbH.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SAC ’20, March 30-April 3, 2020, Brno, Czech Republic
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-6866-7/20/03 . . . $15.00
https://doi.org/10.1145/3341105.3373890

1 INTRODUCTION
The task of classification has been tackled by a large number of supervised machine learning approaches, yielding very accurate and impressive results for different types of datasets. These methods prove to be successful if the modelled scenario fulfills an assumption that is implicitly required for the training data: the availability of a sufficiently large amount of labeled samples for training, covering the whole variety of all possible classes. While elementary approaches only handle binary problems (i.e., identify one class against another), especially more recent techniques using deep learning have beaten the state-of-the-art benchmarks regarding classification accuracy on thousands of different categories, but at the price of increasing the need for labeled training data even further. Contemporary big data applications deliver this huge amount of data, but often lack consistency and correct labeling in advance.

In this work, we propose a semi-supervised classifier that is able to deal with unknown classes and that is generative. Semi-supervised classification ranges between classification in a supervised setting and semi-supervised clustering: we assume that we obtain labeled as well as unlabeled data for training and aim to identify specific classes. In contrast to supervised learning, we have unlabeled data available; in contrast to semi-supervised clustering, where the supervised information is usually introduced by partition-based or must-link/cannot-link constraints, we do not allow reordering, merging or splitting of the predefined classes. As a further difference to clustering, we evaluate our results with the F1-measure instead of clustering quality metrics.

A conventional classifier would assume that each class, which needs to be identified, is represented by a sufficiently large number of labeled instances in the training set. In our case, we assume that some classes might not be present in the labeled part of the training dataset at all, but only in the unlabeled part – we refer to them as unknown classes. This restriction is especially challenging, if the number of labeled training data is small compared to the number of unlabeled training data. Similar constraints appear in one-class classification or PU learning (learning from positive and unlabeled examples), see e.g. [8]. Both of these fields assume that classes exist, which are not covered by the labeled part of the training set, but restrict to one class of interest (which is usually represented in the labeled training data by a sufficiently large number of instances). In contrast, we consider that multiple classes are represented in both the labeled and unlabeled training data in this work.
As a third aspect, our method is generative and therefore allows a deeper understanding of the data by a statistical model, which presents an advantage over discriminative approaches.

We present a two-step approach, which comprises Bayesian classifiers as elementary components. In the first step, the S-EM algorithm, introduced in [9] for PU learning, is used to fit a classifier to the known classes and an artificial additional class accounting for unknown classes (classes not present in the labeled training data). Therefore, the first step identifies a subset of unlabeled data points as "likely unknowns", i.e., as being drawn from unknown classes.

In the second step, a Gaussian mixture model is fit to these likely unknowns, combining the likelihoods of the known and unknown classes to obtain an improved Bayesian classifier.

As many similar (but not identical) problems exist in literature (see Section 2), we will demonstrate the ability of our method in comparison to these in two distinct settings: the multiclass classification problem for which the presented method is developed, and the PU learning problem, which is a special case of our multiclass classification problem. We compare the results of the applied methods on different open datasets from the UCI library1, including the two famous dataset MNIST [6] and LETTER [4], as well as on a wafer-test dataset from the semiconductor industry [10]. While these datasets might be considered as "easy" classification problems in the machine learning community, the extension to semi-supervised classification with unknown classes massively complicates the problem since only a subset of the classes is considered for training. Our method achieves a competitive F1-measure when compared to the state-of-the-art algorithms from related fields (see Section 5).

2 RELATED WORK

Similar problem formulations to our setting were addressed in the machine learning literature. Related fields include exploratory learning, semi-supervised clustering, PU learning and open set recognition.

In [3], a semi-supervised multiclass problem called "exploratory learning" was attacked by the well-known EM algorithm to detect elements from unknown classes. The resulting elements are used to extend the set of classes. As elementary classifiers, methods such as Naive Bayes, seeded K-Means or seeded von-Mises Fisher are suggested. A similar problem is investigated in [2], where a Support Vector Machine (SVM) is generalized to the so-called LACU-SVM, which is able to detect new class by incorporating knowledge from unlabeled data. The approaches of exploratory learning and LACU-SVM additionally consider incremental learning of previously unseen classes, which is not intended in our problem.

[5] presents a statistical physics approach to semi-supervised learning, replacing classifiers minimizing some cost by a Boltzmann distribution (parameterized by a temperature) over all such classifiers. Their approach copes with unknown classes and degenerates to clustering if no labeled data is available. Similarly, the semi-supervised clustering method from [16], which combines model-based cross-entropy clustering with partition-level side information, has the capability to model unknown classes. A generative approach utilizing a constraint-based version of Gaussian mixture models (GMMs) for semi-supervised clustering was presented by [15]. However, semi-supervised clustering does not yield an assignment between the class labels in the training data and the produced clusters. Moreover, the number of produced clusters may be different from the number of classes in the labeled training data.

PU learning and one-class classification solve classification problems, where instances of one positive class of interest must be distinguished from an arbitrary number of other classes, based on labeled data from the positive class only. While one-class classification is restricted to a supervised setup, PU learning additionally considers unlabeled examples from both positive and unknown classes, i.e., a semi-supervised setup. For this purpose, [9] proposed the generative I-EM and the S-EM algorithms for detecting unknown samples in the unlabeled data and used them to train a Naive Bayes classifier. Step 1 of our work is based on their S-EM algorithm. Discriminative approaches towards PU learning exist as well, e.g. [7], where the Rocchio method is combined with support vector machines. Further, a typical representative of one-class classification is the one-class SVM by [14], which aims to identify the closest possible class boundaries of one single class. However, in contrast to our problem, they do not use unlabeled training data, and their algorithm operates in a discriminative way.

Scheirer et al. formulated the problem of open set recognition, i.e. a classification problem with a rejection option for elements from a new class. They developed the 1-vs-set machine [12], as well as the Weibull-calibrated SVM (W-SVM) [13] for this purpose. As a major difference between the presented approach and open set recognition, the latter contains instances from classes that were not present in the training set. Instead, we assume that all classes are present in the unlabeled training set, but unknown classes do not occur in the labeled training set. In general, open set recognition is considered as supervised (probably containing more labeled training data than in our case) and attacked by discriminative methods, while our focus is put on a semi-supervised, generative model.

Furthermore, [18] demonstrated how new classes, which are observed in a data stream (i.e. one-pass learning) can be integrated into a classifier. In contrast to our proposed approach, these methods are discriminative. Another difference is that online learning systems obtain every instance of the training data only once, which is not required in our case.

3 A BAYESIAN PERSPECTIVE ON SEMI-SUPERVISED CLASSIFICATION

Formally, our investigated problem is given as follows: We assume that a set $L \subset \mathbb{F}$ of labeled elements and a larger set $D \subset \mathbb{F}$ of unlabeled elements are provided, where $\mathbb{F} \subset \mathbb{R}^d$ is a continuous feature space. While the elements of $L$ are assigned a class label out of $C \equiv \{c_1, \ldots, c_N\}$, we assume that $D$ contains elements from the same (known) classes $\{c_1, \ldots, c_N\}$, as well as from unknown classes $\{c_{N+1}, \ldots, c_{N+k}\}$, i.e., from $C^* \equiv \{c_1, \ldots, c_{N+k}\}$. In terms of the notation, we aggregate all unknown classes to one residual class, denoted by 0, and therefore, we aim to classify new elements by one of the classes $\{c_1, \ldots, c_N\} \cup \{0\} = C_0$ (classification with unknown classes). In a mathematical sense, our intention is to find a measurable classification function $f : \mathbb{F} \rightarrow C_0$, which, to each element in $\mathbb{F}$, either assigns one of the known classes $C$ or responds with class 0. The scheme of this situation is depicted in Fig. 1.

1http://archive.ics.uci.edu/ml
Given the problem setting stated above, we provide a generative method to solve it. As a starting point, we look at the classification problem from the viewpoint of Bayesian decision theory: we assume that a decision corresponds to assigning a new sample to a known class \( c^* \in C \) or to the unknown class \( c^* = 0 \). Hence, the optimal Bayesian posterior decision given a bounded loss function \( V: C_0 \times C^* \rightarrow [0, K], K < \infty \), is obtained by the class, which minimizes the expected value of the loss function

\[
e^* = \arg \min_{c \in C_0} \mathbb{E}_{\tilde{c} \mid \tilde{c} \mid x} V(c, \tilde{c}) \quad (1)
\]

\[
e^* = \arg \min_{c \in C_0} \int \int V(c, \tilde{c}) dP(\tilde{c} \mid x) \quad (2)
\]

\[
e^* = \arg \min_{c \in C_0} \sum_{\tilde{c} \in C^*} V(c, \tilde{c}) \cdot p(\tilde{c} \mid x) \quad (3)
\]

\[
e^* = \arg \min_{c \in C_0} \sum_{\tilde{c} \in C^*} \sum_{\tilde{c} \in C^*} V(c, \tilde{c}) \cdot p(x \mid \tilde{c}) \cdot p(\tilde{c}) \cdot p(c) \quad (4)
\]

This approach is usually referred to as Bayesian posterior expected loss. We will next adapt the 0-1-loss to our purposes, i.e., we propose the following loss function:

\[
V(c, \tilde{c}) = \begin{cases} 
0, & c = \tilde{c} \\
0, & c = 0, \tilde{c} \in \{c_{N+1}, \ldots, c_N\} \\
1, & \text{else.}
\end{cases} \quad (5)
\]

The loss \( V \) is defined as 0, if either both \( c \) and \( \tilde{c} \) represent the same known class, or \( \tilde{c} \) is any unknown class and \( c = 0 \). Otherwise, the loss is set to 1.

The likelihood and the prior for \( c \in C \) will now be selected in the same way as for the Bayes classifier. In detail, we assume that the likelihood corresponds to a (potentially multivariate) Gaussian distribution, the prior is calculated by the relative frequency of each class \( c \) within the labeled dataset \( p(c) = \frac{|L_c|}{|L|} \), where \( L_c \) is the subset of labeled elements from class \( c \). Alternatively, a uniform prior is also applicable. However, as we need to account for the "unknown" classes as well, the prior distribution is modified by a probability of unknown classes \( p_0 \). The resulting components of the model (restricted to \( c \in C \)) are summarized by

\[
p(c) = (1 - p_0) \frac{|L_c|}{|L|}, \quad (6)
\]

\[
x | c \sim N(\mu(c), \Sigma(c)). \quad (7)
\]

The challenge of the problem is now to select an appropriate prior and likelihood for the "unknown" classes, which should be mapped to the class 0. We propose a model for which the likelihood is the same for all unknown classes, i.e., that \( p(x | c_{N+1}) = \cdots = p(x | c_N) =: p(x | 0) \). Specifically, we model the data for unknown classes using a GMM. It follows that the probability of unknown classes evaluates to \( p(0) = p_0 = \sum_{k=1}^K p(c_{N+k}) \), i.e. the prior probability for the unknown classes. With \( p(0) \) and \( p(x | 0) \) defined accordingly, (4) is solved by what resembles the well-known maximum a-posteriori (MAP) estimator

\[
c^*(x) = \arg \max_{c \in C_0} p(x | c) \cdot p(c). \quad (8)
\]

## 4. A SEMI-SUPERVISED APPROACH TO CLASSIFICATION WITH UNKNOWN CLASSES

We aim at defining a prior \( p(0) \), as well as a likelihood \( p(x | 0) \). At this stage, we depart from a purely supervised setting and take advantage from the information provided by the unlabeled data. Our approach consists of two major steps (see Fig. 2):

1. detect ‘likely unknown’ elements \( U \) in the unlabeled data \( D \) - this will provide us the prior probability,
2. fit an adequate mixture model, which is able to divide these “likely unknowns” into subclasses - this will serve as a likelihood model.

### 4.1 Step 1: Detecting "Likely Unknowns"

The S-EM algorithm by Liu et al. [9] detects a subset \( U \subset D \), which is likely to be in a new class. While originally developed in the context of PU learning, we generalize it for distinguishing multiple classes \( c_1, \ldots, c_N \) against an unknown residual class 0.

As a first step, a random subset \( S \subset L \), including a predefined percentage \( \xi \in [0, 1] \) of the labeled training data, is selected as so-called spies. While \( L \setminus S \) retains its labels \( \{c_1, \ldots, c_N\} \), the labels of \( D \cup S \) are specified as 0. Applying the I-EM algorithm (a version of EM, introduced in [9]) delivers a Bayes classifier for \( \{c_1, \ldots, c_N, 0\} \), which produces a likelihood function \( q(x | c) \). A data point is considered a "likely unknown" if its likelihood falls below a certain threshold \( t \), i.e.,

\[
U = \{x \in D: \forall c \in C: q(x | c) < t \}. \quad (9)
\]

We set the threshold to \( t = \min_{x \in S} q(x | c(x)) \), where \( c(x) \) denotes the correct label of the spy element. In the noise-affected version, Liu et al. suggest to use the empirical 0.1-quantile instead of the min.

Now, instead of selecting a classifier from the iterative I-EM procedure, our interest lies on the set of unknown samples \( U \), which are identified by the procedure. We assume that these are representatives of one or more unknown classes, which are not yet captured by the labeled training data. In case that \( U \) is empty, we conclude
that no unknown class exists and continue with the original multi-class classifier found by the I-EM algorithm. Otherwise, we set the prior for our model as \( p_0 = \frac{|U|}{|L|} \).

### 4.2 Step 2: Fitting a Mixture Model

We next build a generative model for data \( U \) originating from the unknown classes \( C_{N+1}, \ldots, C_{N+k} \). An intuitive choice for this purpose is a GMM, which has proven to be useful in many applications.

The GMM model is commonly defined as a linear combination of \( \ell \) Gaussian distributions, i.e.

\[
U \ni x \sim \sum_{i=1}^{\ell} \phi_i N(\mu_i, \Sigma_i) =: p(x|0),
\]

where \( \sum_{i=1}^{\ell} \phi_i = 1 \). As we are interested in reducing the number of model parameters due to potentially few likely unknown elements, we set \( \phi_i = \frac{1}{\ell} \) for all \( i = 1, \ldots, \ell \). It is worth mentioning that \( \ell \) is not necessarily equal to \( k \), the number of unknown classes. Indeed, \( \ell \) may be strictly smaller (modeling multiple classes with a single Gaussian component) or strictly larger (modelling a single class with multiple Gaussian components) than \( k \).

For a selected number of components \( \ell \), we initialize the partition of \( U \) via k-Means and proceed with an EM algorithm, which provides us the final Gaussian components. However, during EM, we also introduce the labeled data (and their classes) as fixed elements. Hence, elements which were selected as likely unknowns during S-EM although they are part of a known class, can be assigned to their original classes without introducing a "new" Gaussian component.

Our first interest concerns the number \( \ell \) of mixture components: Selecting this number via a maximum likelihood approach is not possible, as the joint likelihood of the model will increase with the number of components \( \ell \). Hence, it is necessary to apply an appropriate criterion, which provides a trade-off between the likelihood and the model complexity (number of estimated parameters).

In line with statistical theory, we select a tailored version of the Bayesian information criterion (BIC) to obtain the number of mixture components in the GMM, defined as

\[
\text{BIC}(\hat{p}(:, \ell)) = m \log(|U| + |L|) - \max_{(\mu_i, \Sigma_i)} \sum_{x \in U \cup L} \log \left( \hat{p}(x; \ell) \right),
\]

where \( m = (\ell + N) \cdot d + (\ell + N) \cdot \frac{d(d+1)}{2} \) denotes the number of estimated parameters (i.e., the number entries in the mean vectors and covariance matrices \( \{\mu_i, \Sigma_i\} \)) and

\[
\hat{p}(x; \ell) = \frac{1}{N + \ell} \sum_{c \in C} p(x|c) + \frac{\ell}{N + \ell} p(x|0)
\]

denotes the likelihood of the whole classifier, regarded as a single GMM (each Gaussian component, regardless of being associated with a known or unknown class, is weighted equally). At this point, note that \( p(x|c) = q(x|c) \), for all \( c \in C \), as the likelihood of the known classes remains unchanged. The BIC is therefore always calculated for all known and unknown components, but minimized over models of different complexity in order to retrieve an optimal choice. The BIC is known to overestimate the number of parameters of a model [1]. Observations from experiments, where the number of components selected via BIC are compared to the true number of unknown classes, underline this behavior. From a theoretical perspective, this overestimation is unproblematic in our setting, since it is preferable to model the unknown classes with too many mixture components, rather than modeling them with too few.

By selecting the model with the lowest BIC value, we obtain the model parameters \( \mu^{(c)}_i \) and \( \Sigma^{(c)}_i \), \( i = 1, \ldots, \ell \) and thereby create a generative model for the previously unknown classes. Using this model, we can determine the likelihood \( p(x|0) \) in (10) for a new element \( x \) and finally obtain the posterior probabilities \( p(c|x) \) for all \( c \in C_0 \) in (4). As a result, we can calculate the minimum posterior expected loss (i.e., the best Bayesian decision). At this stage, we have invented a full classification model to solve a multiclass classification problem, while accounting for unknown classes. More formally, the proposed algorithm can be described as Alg. 1. For practical reasons, the number of components \( \ell \) is bounded by a maximum integer \( \ell_{max} \).
Table 1: Dataset properties.

| dataset   | classes | samples | dimension | samples/class |
|-----------|---------|---------|-----------|---------------|
| dna       | 3       | 3186    | 180       | 765-1654      |
| ecoli     | 8       | 336     | 8         | 2-143         |
| glass     | 6       | 214     | 10        | 9-76          |
| iris      | 3       | 150     | 4         | 50            |
| LETTER    | 26      | 20000   | 16        | 734-813       |
| MNIST     | 10      | 70000   | 784       | 6313-7877     |
| pendigits | 10      | 10992   | 16        | 1055-1144     |
| segmentation | 6   | 6435    | 36        | 626-1533      |
| svmguide4 | 7       | 2310    | 19        | 330           |
| user      | 4       | 403     | 5         | 50-129        |
| usps      | 10      | 9298    | 256       | 708-1553      |
| vertebral | 3       | 310     | 6         | 60-150        |
| vowel     | 11      | 990     | 10        | 90            |
| wafer-test| 5       | 5000    | 108       | 1000          |
| wine      | 3       | 178     | 13        | 48-71         |

Algorithm 1 Generative Semi-Supervised Classifier for Unknown Classes

```
sample S \subset D, \text{s.t. } |S| = \zeta \cdot |D|  
\hat{c}(x) \leftarrow \begin{cases}  
c(x) & x \in L \setminus S  
0 & x \in D \setminus S 
\end{cases}  
run EM algorithm with labels \hat{c}(x), x \in L \cup D, and obtain q(\.|c) 
\ell \leftarrow \min_{x \in S} q(c|x)  
U \leftarrow \{x \in D : \forall c \in C : q(x|c) < \ell\}  
\text{for } \ell = 1 : \ell_{\text{max}} \text{ do}  
\quad \text{train GMM for } U \text{ with } \ell \text{ components given } L  
\text{end for}  
\text{return } \hat{p}(.; \ell^*), \text{s.t. } \ell^* = \arg \min_{\ell=1}^{\ell_{\text{max}}} BIC(\hat{p}(.; \ell))
```

With regard to Alg. 1, the main novelty of our work is the extension of S-EM to the multiclass case, so that training with more than one known class is possible. Therefore, we combine two sophisticated approaches, S-EM and a GMM, to a powerful method for performing multiclass classification under the presence of previously unknown classes. As a third aspect, we select the number of mixture components by a version of BIC, which was tailored to the case of both known and unknown classes, where only the number of unknown classes is variable in the model.

5 EXPERIMENTS AND RESULTS

In this section we analyze the sensitivity of our proposed method to its parameters and compare it with state-of-the-art methods. In our experiments, we consider several datasets, which are characterized in Table 1. Specifically, we perform experiments on UCI datasets, as well as a wafer test dataset from semiconductor industry [10].

In order to provide evidence for the performance of our proposed method, we perform two different experiments: the first experiment is used to demonstrate the ability to discriminate between "known" and "unknown" classes and will be compared to state-of-the-art methods on different UCI datasets, including ecoli, glass, iris, segmentation, user, vertebral and wine, as well as the two famous datasets MNIST [6] and LETTER [4] and a wafer test dataset from semiconductor industry [10], represented by tailored image features [11]. In the second experiment, we compare our method to other multiclass algorithms, which are able to handle unknown classes in a semi-supervised way. However, not all datasets are suitable for this experiment, as some contain too few samples for splitting them into labeled, unlabeled and test datasets.

We will judge the quality of our results by the F1-score, defined as the harmonic mean between precision and recall of the regarded classes and calculate the macro-averaging F1-measure (see, e.g., [17]). If \( c(x) \) represents the real class of \( x \) and \( c^*(x) \) the assigned class, we define the following sets for each known class \( c_i \):

- \( TP_i = \{ x : c(x) = c_i \} \)
- \( FP_i = \{ x : c(x) \neq c_i \} \)
- \( TN_i = \{ x : c(x) \neq c_i \} \)
- \( FN_i = \{ x : c(x) = c_i \} \)

According to these definitions, the F1-score for a class \( c_i, i = 1, \ldots, N \) is defined as follows:

\[
F_{1i} = \frac{2|TP_i|}{2|TP_i| + |FP_i| + |FN_i|},
\]

leading to the macro-averaging F1-score

\[
F_{1,\text{macro}} = \frac{1}{N} \sum_{i=1}^{N} F_{1i}.
\]

We compare our semi-supervised classifier for unknown classes (SSC-UC) to the following methods: the generative Naive Bayes version of exploratory learning by [3] (EL)\(^2\), the S-EM algorithm for PU learning by [9] (S-EM)\(^3\), the GMM-based semi-supervised clustering method with must-link and cannot-link constraints by [15] (cGMM)\(^4\) and the semi-supervised cross-entropy clustering method with information bottleneck regularization by [16] (CEC-IB)\(^5\). For the clustering algorithms cGMM and CEC-IB, the assignment of the class labels to the resulting clusters was done in favor of the F1 scores, as they provided the clusters in an arbitrary order. In the setting where multiple classes are known, we apply a committee of several S-EM algorithms, where each member is trained on a different known class as the positive class. If all committee members decide that a sample belongs to unknown class, then the sample is assigned to class 0; otherwise, the sample is assigned to the positive class with the highest score among all committee members. By this adaption, we directly generalize the S-EM algorithm, representing a state-of-the-art PU learning method.

5.1 Parameter Sensitivity

For our method, the employed S-EM algorithm proved to be highly dependent on a good sampling of the spy elements. This aspect is even more crucial if we investigate small labeled training datasets. In case of a bad selection of the threshold (caused by an unfortunate sampling of the spies), two failure modes can be observed: if the threshold is selected too small, the returned set \( U \) is empty although

\(^2\)implemented by the authors
\(^3\)implemented by the authors
\(^4\)http://www.scharp.org/thertz/code.html
\(^5\)https://github.com/mareksmieja/CEC-IB
unknown classes existed. On the other hand, if the threshold is too high, $U$ contains both, known and unknown elements. To resolve this issue and increase the robustness of the S-EM algorithm, we run the procedure $l$ times for independently sampled spy sets $S_i$ of equal size and obtain sets of likely unknown elements $U_i$. We continue with the set with the median number of elements, i.e., $U = \frac{1}{l} \sum U_i$, where $U_i$ are ordered by increasing cardinality.

To investigate the sensitivity of SSC-UC w.r.t. the parameters, we observe the F1-scores when varying the fraction $\zeta$ of spies in the step 1 in a range from 0.01 to 0.2 and the number $l$ of times how often this step is repeated in a range from 5 to 50. The results of this analysis are shown in Table 2. It can be seen that neither $\zeta$, nor $l$ has a significant influence on the overall performance of the SSC-UC classifier for most datasets. The fact that the parameter $\zeta$ has only minor influence on the result is in accordance with the observations on document classification described in [9], where the authors claimed that, if in a reasonable range, the percentage of spies does not matter to the S-EM step. In the remainder of this

### Table 2: Sensitivity of SSC-UC to distinct parameter settings. In all cases, we set $N = 2$ and select these classes randomly. Bold settings indicate the default setting for the remainder of the experiments.

| dataset   | $l = 10$ | $l = 10$ | $l = 10$ | $l = 10$ |
|-----------|----------|----------|----------|----------|
|            | $\zeta = 1\%$ | $\zeta = 5\%$ | $\zeta = 10\%$ | $\zeta = 15\%$ | $\zeta = 20\%$ |
| dna       | 0.84     | 0.88     | 0.85     | 0.84     | 0.85   |
| ecoli     | 0.85     | 0.86     | 0.86     | 0.85     | 0.86   |
| glass     | 0.55     | 0.54     | 0.64     | 0.64     | 0.62   |
| iris      | 0.8      | 0.8      | 0.91     | 0.77     | 0.91   |
| LETTER    | 0.21     | 0.15     | 0.2      | 0.17     | 0.15   |
| MNIST     | 0.78     | 0.8      | 0.79     | 0.81     | 0.76   |
| pendigits | 0.5      | 0.39     | 0.41     | 0.39     | 0.38   |
| satimage  | 0.63     | 0.65     | 0.64     | 0.65     | 0.63   |
| segmentation | 0.57  | 0.57     | 0.57     | 0.56     | 0.56   |
| svmguide4 | 0.35     | 0.57     | 0.57     | 0.56     | 0.57   |
| user      | 0.5      | 0.8      | 0.79     | 0.78     | 0.82   |
| usps      | 0.53     | 0.49     | 0.47     | 0.51     | 0.5    |
| vertebral | 0.45     | 0.59     | 0.62     | 0.6      | 0.57   |
| vowel     | 0.42     | 0.31     | 0.32     | 0.29     | 0.29   |
| wafer-test| 0.73     | 0.71     | 0.85     | 0.81     | 0.8    |
| wine      | 0.76     | 0.88     | 0.96     | 0.78     | 0.99   |

### Table 3: F1-measures for SSC-UC, EL, S-EM, cGMM and CEC-IB, applied to the multiclass semi-supervised classification problem with unknown classes. $\times$ denotes that the algorithm did not converge for the given dataset. For datasets with 3 classes, $N > 2$ is obsolete, for datasets with 4 classes, $N \approx k$ is already covered by $N = 2$ and $k = 1$.

| dataset   | $N = 2$ | $N = k$ | $k = 1$ |
|-----------|---------|---------|---------|
|            | SSC-UC | EL      | cGMM    | CEC-IB  |
|            | SSC-UC | EL      | cGMM    | CEC-IB  |
|            | SSC-UC | EL      | cGMM    | CEC-IB  |
| dna       | 0.85 | 0.57 | 0.12 | × | 0.19 |
| ecoli     | 0.86 | 0.74 | 0.8  | 0.75 | 0.61 |
| glass     | 0.64 | 0.6  | 0.3  | 0.21 | 0.25 |
| iris      | 0.91 | 0.8  | 0.67 | 1   | 0.38 |
| LETTER    | 0.79 | 0.45 | 0.29 | ×   | 0.14 |
| MNIST     | 0.41 | 0.23 | 0.82 | 0.33 | 0.05 |
| pendigits | 0.65 | 0.6  | 0.84 | 0.22 | 0.1 |
| satimage  | 0.57 | 0.34 | 0.72 | 0.06 | 0.06 |
| segmentation | 0.57 | 0.47 | 0.54 | ×   | 0.26 |
| svmguide4 | 0.79 | 0.69 | 0.87 | 0.6  | 0.12 |
| user      | 0.47 | 0.27 | 0.66 | ×   | 0.21 |
| usps      | 0.62 | 0.45 | 0.6  | 0.4  | 0.29 |
| vertebral | 0.32 | 0.29 | 0.74 | 0.15 | 0.24 |
| vowel     | 0.88 | 0.54 | 0.99 | ×   | 0.3 |
| wafer-test| 0.96 | 0.78 | 0.12 | 0.5  | 0.46 |
| wine      | 0.96 | 0.78 | 0.12 | 0.5  | 0.46 |
work, we thus set $\zeta = 0.1$. During the experiments, we observed that for an increasing $l$, the results became more stable (less affected by random initialization). However, as also the runtime increased linearly with this parameter, the setting with $l = 10$ was deployed as a default for further experiments.

Among the datasets, the best results were achieved for the dna, ecoli and iris dataset, while the method performed worst on the LETTER dataset. This fact can be explained by the total number of classes comprised in these datasets: as LETTER contains many unknown classes, modelling these via a GMM is difficult due to inhomogeneity. However, iris and dna are datasets with only 3 classes, i.e. only 1 class is unknown during training.

### 5.2 Multi-Class Classification with Unknown Classes

In order to demonstrate the abilities of our method for multiclass classification with unknown classes, we evaluate the given datasets in comparison with state-of-the-art methods described in the related work section. We select the following experimental setup:

We define an experimental dataset size $\delta \in \mathbb{N}$ for each dataset, which is equal to the full size of the dataset with exception for some datasets (e.g. LETTER), where the selected labeled training classes are not sufficiently populated. Then, we set the number of labeled samples to approximately $0.1 \cdot \delta$, the number of unlabeled training data to $0.6 \cdot \delta$ and the number of test data to $0.3 \cdot \delta$. We train the classifier using a balanced, randomized sample of representatives from the known classes $\{c_1, \ldots, c_N\}$ of each dataset as labeled data and samples of all present classes $\{c_1, \ldots, c_{N+k}\}$ as unlabeled data. We fix the number $N$ of known classes to 2, to half the total number of classes (we set $N = k$), and to the total number of classes minus one (i.e., $k = 1$), respectively. The unlabeled training dataset and test dataset are sampled out of all available classes. Although regarded as baseline datasets for multiclass classification, these datasets are still challenging in the investigated problem setup.

The results of this experiment are shown in Table 3. The cGMM did not converge for all dataset. Furthermore, as datasets containing merely 3 or 4 classes are present, some settings are redundant − hence, only the setup with $N = 2$ is investigated for datasets with 3 classes, while results for $N = 2$ and $k = 1$ are presented for datasets with 4 classes. Note that the distinct settings cannot be ranked w.r.t. their complexities due to two counteracting aspects: on the one hand, increasing the number $N$ of known classes leads to more options in the classification step; while for $N = 2$, a decisions between classes 0, 1 and 2 must be taken, the number of options is much larger if e.g. the predicted class is selected from $\{0, \ldots, 10\}$ for $N = 10$. On the other hand, an increasing $N$ leads to a lower number of unknown classes, i.e. the set of unknown elements is more homogeneous and easier to separate from the known classes.

In the multiclass setting, our method outperformed the clustering approaches, as classification algorithms are effectively able to make use of labeled data in contrast to (semi-supervised) clustering algorithms, especially in a high-dimensional feature spaces, where the clustering relies more on the distance measure than the classification methods. In many cases, the number of clusters was selected very low, which caused bad results. Regarding the EL method, the algorithm assigns an unknown class to many samples belonging to known classes. This leads to a high number of “false negatives” and a low recall. However, the performance of EL tends to improve if the number of known classes $N$ is increased. With regard to the S-EM method, we can observe the contrary behaviour: while for $N = 2$, this method delivers very accurate results, the F1-scores decreases for larger $N$. SSC-UC, however, shows a more stable behavior with regard to the number of known classes.

Concerning the number $\ell$ of mixture components in step 2 of the SSC-UC method, we observed that, as expected, BIC tends to overestimate $\ell$. While only one class is unknown in the case of $k = 1$, the parameter $\ell$ is estimated to approximately 3 or 4 for most datasets. This effect guarantees that the inhomogeneous set of unknown classes is modelled in a flexible way, improving results.

To investigate how our method is affected by the size of the labeled training dataset, we additionally carried out experiments with increasing proportion of labeled vs. unlabeled training data. As this comparison is not possible with datasets containing sparsely populated classes, we conduct this experiment on few larger datasets, i.e. LETTER, pendigits, satimage and wafer-test. In detail, the proportions of sampling training and test data from the datasets are varied: while the standard setting comprised an $\alpha / (70 - \alpha) / 30$ split (labeled / unlabeled / test) w.r.t. the total experimental dataset size $\delta$ with $\alpha = 0.1$, we investigate the cases of $\alpha \in \{0.23, 0.35\}$ in addition. The results are depicted in Fig. 3.

When increasing the proportion of labeled data, obviously not all methods are able to profit from this additional information. In particular, SSC-UC delivers significantly better results for the pendigits and the satimage dataset, while the results for LETTER and wafer-test remain in a similar range. Unexpectedly, CEC-IB and EL do not benefit from a larger amount of labeled training data, but in contrast, their performance drops. S-EM continuously delivers results of similar quality in all setups.

### 5.3 PU Classification

We finally assess our proposed classification method in a problem setting from PU learning. Therefore, we assume that instances from only one known (positive) class are provided in a labeled training dataset, while instances from the known and multiple unknown classes (positives and negatives) are available in the unlabeled dataset. The goal is to train a classifier, which is able to distinguish between the positives and negatives in the test set. Note that this setting is a special case of the setting presented in the previous experiments. By default, the positive class will be randomly chosen from all classes represented in the dataset. Again, we reduce the number of labeled samples to approx. 10% of the dataset size, as well as we sample 60% as unlabeled training data and 30% for testing. The results are evaluated by the F1-measure, considering the prediction labels “positive” (known) and “negative” (unknown).

Table 4 demonstrates the performance of our proposed algorithm (SSC-UC) compared to those of existing methods. SSC-UC performed equally well or better than other algorithms in most cases, while the performance of the competing methods varies strongly by the datasets. Especially the clustering algorithms have difficulties especially with larger or higher-dimensional datasets (such as MNIST) – as before, no convergence could be achieved for the cGMM method in some cases. While S-EM performed well
in this experiment, EL did not assign any unknown class labels in most datasets, which explains the low F1 scores.

From the perspective of computational complexity and runtime, all compared methods were in an equal range. In detail, cGMM had the lowest, CEC-IB had the longest processing time. SSC-UC was in a medium range, depending on the number of iterations \( l \). The main factor influencing the runtime of the experiments was the number of features contained in the datasets — hence, MNIST and usps had the longest processing times. For all other datasets, results could be obtained within seconds.

6 DISCUSSION
The experiments in the previous section illustrated that our approach to semi-supervised classification with unknown classes successfully competes with the state-of-the-art in semi-supervised clustering, exploratory learning or PU learning. For a small ratio between the numbers of known and unknown classes, a committee of S-EM algorithms achieves top performance; if this ratio is large, then exploratory learning performs well. In both extremes, SSC-UC performs on par with these schemes, indicating the wide applicability of our method. In addition to this, SSC-UC achieves competitive performance in PU learning problems, where it is only outperformed by S-EM, a method that is tailored towards this very problem. To these selling points one can add the fact that SSC-UC requires the setting of only few parameters, and the performance is not very sensitive even to those few.

Compared to related methods, which are able to tackle the problem of semi-supervised classification with unknown classes, we extend the current state-of-the-art by increasing the potential of the Bayes classifier with an option to assign a sample to an unknown class. The generative nature of the method provides not only an appropriate prediction for the test dataset, but also a basic understanding of the distribution of the classes (unknown, as well as previously known classes). Furthermore, due to its low number of parameters, our model is capable of handling low amounts of labeled data, which is a clear benefit in many practical applications. Possible extensions of our model are soft classification and adaptation to use mixtures of other probability distributions or even distinct (generative) classifiers. In contrast to clustering methods, we can conserve the distinction between known classes, but also - in contrast to supervised classifiers - detect unknown elements.
We presented a generative approach to solve the multiclass semi-supervised classification problem with high probability. Then, these likely unknown elements are treated as new classes emerging over time. Open set classifiers such as Open Set SVMs and Open Set GMMs are adapted to other generative frameworks (such as GMMs). The algorithm for semi-supervised classification consists of two steps: first, the unlabeled dataset is screened for so-called likely unknown elements, i.e., samples that are drawn from the unknown classes with high probability. Then, these likely unknown elements are used to set up a new classifier, which fits a mixture of Gaussians to describe the unknown elements.

In the provided experiments, our approach performs on par with the state-of-the-art methods from different branches of semi-supervised learning. Further, we presented a list of notable facts that underline the quality and promising features of our method. The ability to handle many constraints in the problem setup, especially the detection of new classes, paves the way for practical applicability in many domains, such as decision support systems. A future goal is to modify our method towards incremental learning and lifelong learning, i.e., to continuously improve the performance of the method after initial training.

ACKNOWLEDGMENTS

The work has been performed in the project Power Semiconductor and Electronics Manufacturing 4.0 (SemI40), under grant agreement No 692466. The project is co-funded by grants from Austria (BMVIT-IKT der Zukunft, FFG project no. 853338), Germany, Italy, France and Portugal and Electronic Component Systems for European Leadership Joint Undertaking (ECSEL JU).

REFERENCES

[1] Sakyajit Bhattacharya and Paul D. McNicholas. 2014. A LASSO-penalized BIC for mixture model selection. Advances in Data Analysis and Classification 8, 1 (01 Mar 2014), 45–61. https://doi.org/10.1007/s11634-013-0155-1
[2] Dong, Yang Yu, and Zhi-Hua Zhou. 2014. Learning with Augmented Class by Exploiting Unlabeled Data. In 28th AAAI Conference on Artificial Intelligence. https://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/view/8388
[3] Bhavana Dalvi, Cohen, Callan William W., and Jamie. 2013. Exploratory Learning. "Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2013)" Lecture Notes in Computer Science.
[4] Peter W. Frey and David J. Slate. 1991. Letter recognition using holland-style adaptive classifiers. Machine Learning 6 (March 1991), 161–182. Issue 2.
[5] Gudet, Noam Shental, and Eytan Domany. 2005. Semi-supervised learning - a statistical physics approach. In "Learning with Partially Classified Training Data"–ICML05 workshop.
[6] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. In Proceedings of the IEEE. Vol. 86. 2278–2324. Issue 11.
[7] Xiaoli Li and Bing Liu. 2003. Learning to Classify Texts Using Positive and Unlabeled Data. In "Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence". 587 – 592.
[8] Bing Liu. 2011. Web Data Mining (2nd edition ed.). Springer: 184–204 pages.
[9] Bing Liu, Wee Sun Lee, Philip S Yu, and Xiaoli Li. 2002. Partially supervised classification of text documents. In ICML. Vol. 2. Citeseer. 387–394.
[10] Martin Fleschberger, Michael Scheuer, and Stefan Schranner. 2019. Simulated Analog Wafer Test Data for Pattern Recognition. https://doi.org/10.5281/zenodo.2542504
[11] T. Santos, S. Schranner, B. C. Geiger, O. Pfeifer, A. Zernig, A. Kaestner, and R. Kern. 2019. Feature Extraction From Analog Wafermaps: A Comparison of Classical Image Processing and a Deep Generative Model. IEEE Transactions on Semiconductor Manufacturing 32, 2 (May 2019), 190–198. https://doi.org/10.1109/TSM.2019.2911061
[12] Walter J. Scheuer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E. Boult. 2013. Towards open set recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 35, 7 (July 2013), 1757 – 1772.
[13] Walter J. Scheuer, Lalit P. Jain, and Terrance E. Boult. 2014. Probability models for open set recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 36, 11 (March 2014), 2317 – 2324.
[14] Bernhard Schölkopf, Robert C. Williamson, Alex J. Smola, John Shawe-Taylor, and John C. Platt. 1999. Support vector method for novelty detection. In "Advances in Neural Information Processing Systems 12 (NIPS)". 582–588.
[15] Noam Shental, Aharon Bar-Hillel, Tomer Hertz, and Daphna Weinshall. 2004. Computing Gaussian mixture models with EM using equivalence constraints. In "Advances in Neural Information Processing Systems (NIPS)".
[16] Marek Snieja and Bernhard C. Geiger. 2017. Semi-supervised cross-entropy clustering with information bottleneck constraint. Information Sciences 421 (Dec. 2017), 254 – 271. https://doi.org/10.1016/j.ins.2017.07.016 open-access: arXiv:1705.01681 [cs.AI]
[17] M. Zhang and Z. Zhou. 2014. A Review on Multi-Label Learning Algorithms. IEEE Transactions on Knowledge and Data Engineering 26, 8 (Aug 2014), 1819–1837. https://doi.org/10.1109/TKDE.2013.39
[18] Yue Zhu, Kai-Ming Ting, and Zhi-Hua Zhou. 2017. New Class Adaptation Via Instance Generation in One-Pass Class Incremental Learning. In IEEE International Conference on Data Mining, 1207 – 1212. https://doi.org/10.1109/ICDM.2017.163

Table 4: F1-measures for SSC-UC, EL, S-EM, cGMM and CEC-IB, applied to the PU learning problem (i.e., $N = 1$). The number of classes for the datasets is denoted in brackets. $\times$ denotes that the algorithm did not converge for the given dataset.

| dataset | SSC-UC | EL | S-EM | cGMM | CEC-IB |
|---------|--------|----|------|------|--------|
| dna     | 0.67   | 0.28 | 0.35 | $\times$ | 0.26   |
| ecoli   | 0.95   | 0.6  | 0.95 | 0.76 | 0.6    |
| glass   | 0.59   | 0.46 | 0.59 | 0.54 | 0.54   |
| iris    | 1      | 0.55 | 1    | 1    | 0.74   |
| LETTER  | 0.78   | 0.76 | 0.85 | 0.1  | 0.12   |
| MNIST   | 0.85   | 0.32 | 0.85 | $\times$ | 0.32   |
| pendigits| 0.46   | 0.24 | 0.71 | 0.16 | 0.1    |
| satimage| 0.45   | 0.27 | 0.74 | $\times$ | 0.96   |
| segmentation| 0.84 | 0.09 | 0.62 | $\times$ | 0.2    |
| svmguide4| 0.49   | 0.24 | 0.49 | $\times$ | 0.16   |
| user    | 0.74   | 0.33 | 0.89 | 0.61 | 0.77   |
| usps    | 0.54   | 0.13 | 0.62 | $\times$ | 0.13   |
| vertebral| 0.42   | 0.21 | 0.5  | 0.27 | 0.36   |
| vowel   | 0.3    | 0.26 | 0.68 | 0.13 | 0.21   |
| wafer-test| 0.95  | 0.29 | 0.33 | $\times$ | 0.29   |
| wine    | 0.92   | 0.39 | 0.92 | $\times$ | 0.96   | 0.39   |